



UNITRADE

WHITEPAPER 1.0

Introduction

UniTrade is a completely decentralized trading platform in the DeFi space, built on top of UniSwap liquidity pools. It requires no KYC, no accounts and has no trading limits. By leveraging UniSwap's vast liquidity pools, UniTrade aims to provide a layer on top of the existing architecture to add substantial functionality, via an improved trading interface with extended functionality.

UniTrade users will be able to automate trades, schedule buys and sells and monitor critical trading data (such as order books, charts, and analytics) for Ethereum and ERC-20 token pairs. Therefore, providing the tools and functionality one could expect from a centralized exchange now in a completely decentralized platform, UniTrade.

Growth of DeFi & Liquidity Pools

Aside from the amazing technology behind cryptocurrency projects, exchanges continue to be one of the most impactful tools and arguably, the catalyst behind the rapid growth of the entire cryptocurrency market. While most of these remain completely centralized, the majority of the assets on their markets are decentralized (Bitcoin, Ethereum, etc).

As development in the cryptocurrency space continues to advance, the rise of decentralized applications are growing exponentially and for good reasons. Decentralized applications provide full transparency, entirely unbiased information, immutability, agility, and are not controlled by any single entity. All of these features of DeFi applications are difficult, or even impossible, to find in traditional centralized exchanges.

People in the community appreciate these features, and consequently, a boom of new DeFi applications has accumulated over \$4 billion of total assets as of writing this document. One of those applications is UniSwap.

UniSwap is currently the leading decentralized exchange, with over \$280M in total liquidity and a daily trading volume c.\$200M per day. Their platform provides a simple to use interface to perform real time swaps between two assets. Shown below, is a chart of the exponential growth in volume, market share, total value locked and unique users on their platform.

UniSwap & Liquidity Pools

Uniswap is a decentralized on-chain smart contract. Built on the Ethereum network, it was initially created and developed by Hayden Adams. It allows for trustless swaps between ERC-20 compatible tokens. As opposed to traditional exchanges, it utilizes “liquidity pools” deployed by smart contracts.

To further explain, markets are created by supplying an initial amount of two assets to form a liquidity pool. The exact exchange rate between these two assets is set by the initial provider, and fluctuates depending on supply/demand of these assets on the open market. On a very small scale, you can imagine a liquidity pool is created by adding 1 ETH and 1 TRADE and therefore, the price per TRADE token is equal to 1 ETH. Now if a new user utilized this liquidity pool to sell 0.5 TRADE, there would now be 0.5 ETH and 1.5 TRADE in this pool, and thus price per TRADE would adjust accordingly.

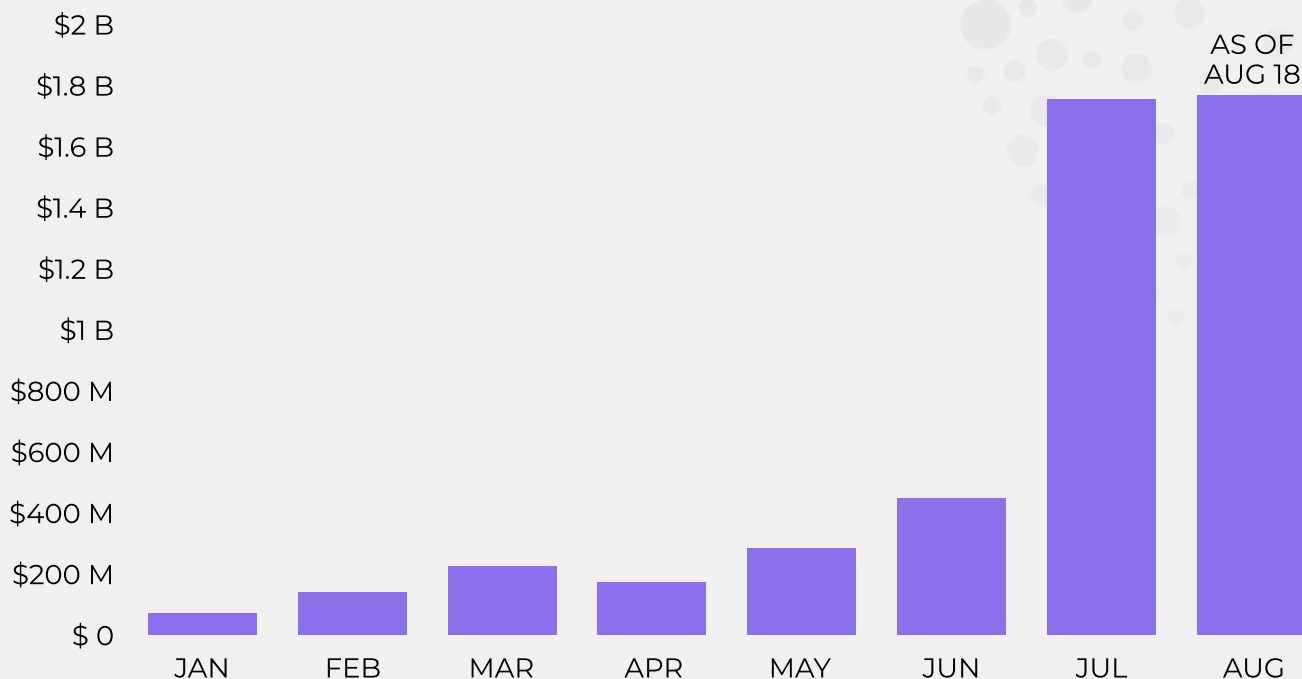
This process occurs automatically as a result of its “Constant Product” mechanism, and is derived from the following mathematical formula:

$$x * y = k$$

x = Quantity of ETH y = Quantity of ERC Token k = Constant Value

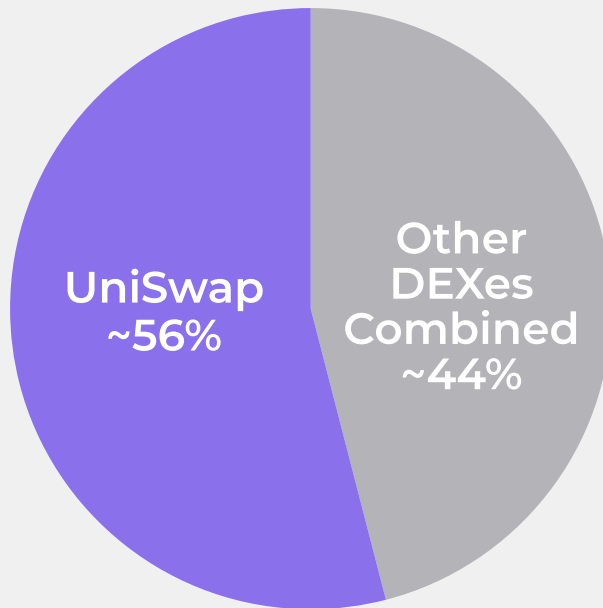
UniSwap Monthly Trading Volume

Except for April, all months have seen volume increase by at least 15% MoM.



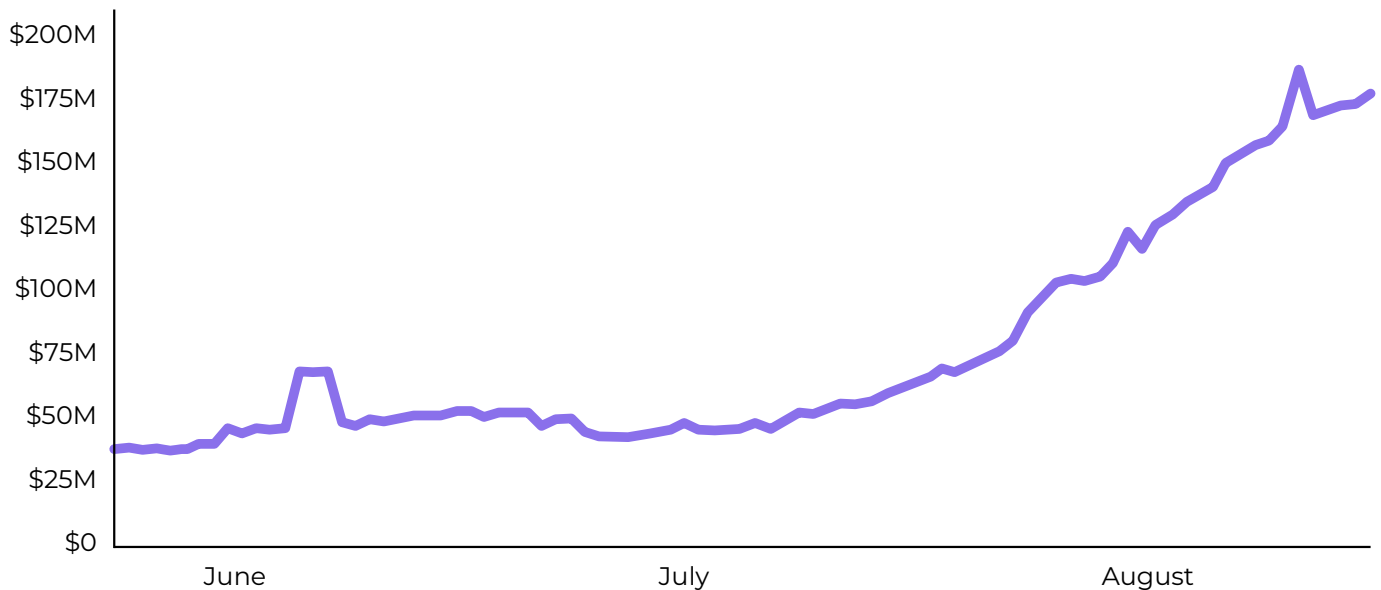
Market Share

As of the 1st August, UniSwap had over half of the market share by volume for DEXes.



Total Value Locked

LTV & ETH have both been simultaneously increasing, meaning their rise is not only attributed to one variable. Furthermore, its recent developments have been exacerbated by the strong “tailwinds” present for DEX’s in the current cryptocurrency environment.



All of the research, development and passion to create UniSwap, is very much respected by our developers. It is our intention to extend this amazing platform, while supporting all their backend developments in any way we can. If you have time, I would strongly recommend reading the incredible journey Mr. Adams took to get UniSwap to where it's at today- <https://uniswap.org/blog/uniswap-history/> - it's an interesting story and very inspiring.

“UniSwap is an amazing platform but it’s so new and there are so many times I wish I could setup buy orders or sell orders at specific prices for tokens. There just wasn’t anything out there to simply automate that. That’s when the UniTrade idea formed.”

@Reborn1002, Founder & Developer

UniTrade Value Proposition

UniSwap, as the name implies, provides users with a way to instantly swap between assets in real time. This however, lacks the advanced functionality that many users coming from existing centralized interfaces are accustomed to. For example, instead of being able to set a limit order on a coin, you instead must continue to monitor the price and manually place an order in real time.

Furthermore, traders have considerably less information on the UniSwap platform. There are not any indicators of selling/buying pressure and market depth for given assets. Users are simply in the dark, and again, must make decisions in real time, as they continue to manually monitor activity.

It's this exact lack of functionality that gives rise to UniTrade. Which we will go into more detail below.

Price point orders

Through its simple user interface, users will now seamlessly be able to set and schedule price point triggers on UniTrade.

Akin to traditional exchanges, once pre-established specific price points are hit on UniSwap, UniTrade will automatically execute these orders.

Trading information

Through various charts and analytics, UniTrade users will also have the information they would typically expect for an advanced trading experience.

Traders will be able to analyze and act, based upon the knowledge they obtain from such data (i.e market depth, volume and live order books).

How do we do this?

User Submits Order

The first step occurs once the order is received from the user. This is handled via web3 on either our desktop site, or our mobile application.

Initially, the user will fill out the input fields determining how much of a token they want to sell/buy, with multiple amounts/price points for one token pairing being an option.

Other choices, such as slippage and Gas to be paid, will also be presented.

Once price points and amounts using the frontend application have been decided, they can hit submit. The contract will then take the total amount of tokens and/or Ether expected to be swapped from the user in one transaction, storing this in a balance mapping of (address user -> (address token -> uint balance)). Knowing, at any one time, how many tokens it has deducted from the user.

Once that mapping is updated, the orders are placed into their respective mappings (address token -> (address price -> [address user, uint amount, bool completed, bool valid, string errorString, uint buyOrSell])) to be executed at a later time.

The user's portion of the process is then complete, they can now await delivery of their order.

Autonomous Checks

Using the Ethereum alarm clock oracle, the contract is called upon on a frequent adjustable basis according to scale and execution estimates. Presently, the frequency has been set to every 60 seconds. This call tells the contract to loop through each token listed and determines current pricing from the UniSwap contract.

Once the price has been determined for that token, it will then look in the token mappings to determine if any orders exist that are either under the price (if a sell order) or above the price (if a buy order). If any of those orders are valid, the contract will execute on them.

Since this is a read only function, no gas is necessary to execute the call and it can happen "behind the scenes" running directly on the chain at the set intervals.

Order Execution

The execution is handled by the contract itself, paying gas from its Ether amount to execute the function. This Ether comes from the user's inbound order if they placed a buy order with Ether. Otherwise, it comes from the contract's total balance. In this case, at the end of execution, the contract will obtain a "refund" for the gas it spent from the Ether that would have been awarded to the user.

The order is then executed, UniSwap is called and tokens or Ether are transferred respectively with the contract being the receiving end of the order.

Once the contract has received the asset it will execute the refund amount explained above, and then send the remainder of the order directly to the users wallet.

How does it scale?

The difficulty with scaling the previously described process lies predominantly in the mapping length, as obtaining all orders ready to be executed on a frequent basis can take longer than expected, particularly once the length of these increases. The main way to deal with this, is to ensure mappings are indexed based on price, so that the contract only ever looks for buy orders above the price and sell orders below it. Therefore, discarding anything that falls outside of that cutting down on the looping required. This allows us to apply a “divide and conquer” algorithm, efficiently finding all orders with the least amount of work possible for the chain.

As things scale, multiple segmented contracts can also be deployed as a solution. Counting already with a master, factory, routing and token pairing contracts, as the platform and it's orders begin to scale, these can be segmented out even further. Therefore, allowing for greater mapping size control, and thus leading to reduced processing times and gas.

What is the contract architecture?

As mentioned previously, the contracts consist of 4 interlinked ones deployed directly on the Ethereum mainnet. Separating out the functionality, allows us to innovate at scale. Meaning, changing something in the token pairing contract, won't affect the routing contract code.

Master Contract

The master contract exists to have oversight over everything occurring at every other level of the contract architecture, and provides response to and from the user.

When the user originally places the order, the master contract takes over to designate what should be done and how.

Factory Contract

The factory contract contains all the information necessary to deploy token pairing contracts, at any one point. Its job is to decide if a token pairing exists, and if not, to deploy a separate one.

Token Pairing Contract

This contract forms the inner workings of the operation. It takes in the tokens from the user, places orders on UniSwap, and handles the tracking of all of the necessary data. It contains orders, users, tokens and calls out to UniSwap.

Routing Contract

This tells every other contract how to get to every other one. It contains giant mappings, which contain the addresses of every contract in the architecture, so that the master one knows where to push tokens and orders for handling.

Gas explanation

Gas is worth a lengthy discussion considering how high it is climbing on the Ethereum network as of right now. There are three main components as to how UniTrade will handle it.

Keeping Gas Cost Low

Right now, the gwei required to get an execution is climbing at an alarming rate. While the Ethereum development team is working diligently to bring this under control, and recent days have seen a slight decrease, it remains an issue to be considered at the forefront of every DAPP development. With that in mind, UniTrade has a 4 step approach towards keeping gas execution costs to an absolute minimum.

Users are allowed to set a “maximum” gas cost when they place their orders. This means the user can determine how much they are willing to spend on their orders to execute.

Multi-contract deployment with only one contract per token pairing. This means that mappings stay small and manageable, less looping means paying less to execute the function.

Storage is kept to an absolute required minimum. As it is the fastest way you can spend gas. Arrays are deleted when an order is executed. Ethereum currently rewards gas refunds to any function execution that is removed from storage. At some points, this function can refund almost 50% of the gas used to execute the call.

Delegating Gas

With UniTrade being an automated system, orders are not executed at the time “Swap” is pressed. It “sits” on the contract, waiting to execute once the price points are met. This means, gas to be paid at a later date, has to be then accounted for without any intervention from the user. Giving rise to gas delegation.

Essentially, as you place an order, the system determines what the expected gas will be at the time of execution. If no Ether is selected in the token pairing, a small upfront fee is charged to cover that gas cost. If Ether is in the token pairing, it will then be taken.

Once the order executes, if a surplus of Ether ends up being taken, it will then be refunded accordingly. UniTrade does not keep it.

This process ensures the user is only paying for gas once, whilst allowing for customization of its chosen amount. Making it simple and concise.

Gas Management

Gas is always going to be around and expensive until at least Ethereum 2.0 rolls out. Unfortunately, it’s part of being a decentralized platform. UniTrade trusts in the development team behind Ethereum and their ability to roll out scalability changes to gas in the future, but it can’t only rely on that.

As part of its commitment to users, UniTrade endeavours to continuously innovate, and try to find new ways to decrease the amount of gas required in every function call the system does.

As new improvements are found, changes will be rolled out pushing cost savings back to users.

Token economics

The TRADE token exists on the Ethereum network as an ERC-20 that is non-mintable with a fixed supply of 50,000,000. The core of the token-economics is that TRADE's are burnt on a consistent basis, causing a deflationary effect on the total circulating supply.

UniTrade Fees

60% of the fees collected by the UniTrade platform will execute a buy back of TRADE tokens directly against the UniSwap TRADE/ETH liquidity pool each day, being destroyed forever. These will be verifiably burnt by the contract automatically. The process will be open source, verifiable and trusted before launch. Noting that as it happens at the contract layer, The UniTrade founders will not have any control over this.

Community TRADE Rewards

30% of the fees (in ETH) collected by the UniTrade platform, will be awarded to TRADE token holders that are actively locking their funds in a smart contract. The duration you choose to lock your TRADE and the amount of tokens you are locking, will determine how much ETH you will earn per day for doing so.

These token economics allow for the community supporting UniTrade, to earn passively from their TRADE holdings.

Founders Reward / Developers Fund

10% of the fees collected by the UniTrade platform will be awarded to the founding members and continued development of the platform.

What fees are charged?

UniTrade will charge a flat fee of 0.2% above the UniSwap platform ones (currently averaging 0.3%). If the total fee is larger than \$50 USD, it will then be capped at this rate.

Fee Discounts

Holders of the TRADE token are also granted a fee discount for their holdings. These are set at the minimum rate mentioned below:

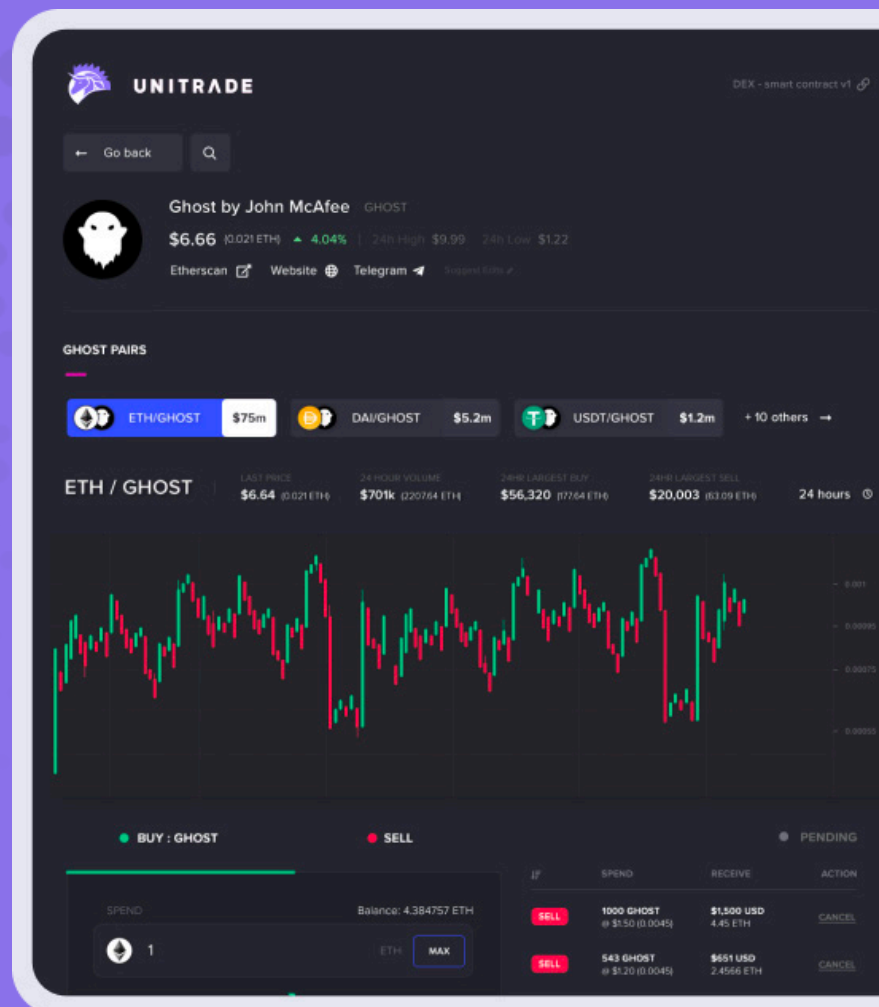
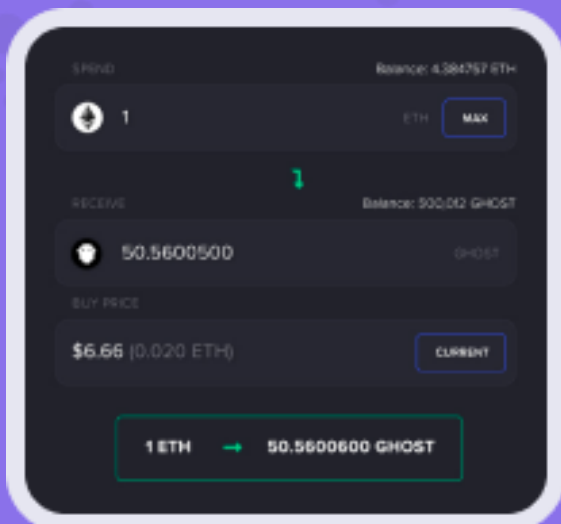
5,000 TRADE	Discounted to 0.15%
10,000 TRADE	Discounted to 0.125%
25,000 TRADE	Discounted to 0.1%

How do I use UniTrade?

The primary interface is the web frontend. It's a web app that allows the user to connect any valid web3 wallet (metamask for example) and interact directly to place orders and cancel them. Users can import any wallet they would like into metamask and use it through the site, including hardware wallets.

Frontend components

The frontend of UniTrade is a react application that will be accessible via any browser, and a mobile application running on any android or IOS device. Both of these access data directly on the chain, with the exclusion of cached pair price, liquidity and gas estimations from the UniSwap API. To access this data, UniTrade will have a small nodejs instance and a mongodb instance, both of which will exist purely to speed up the process of obtaining said data. If at any one time, those instances become unreachable, the site and app will default to accessing directly on-chain.



React Web App

The react web app exists to be operated as the primary access method for users to use UniTrade on a consistent basis. It allows users to connect a web3 capable wallet (MetaMask, Trust etc) and execute their trades directly on-chain.

It provides a simple UI, sliders for pricing, input fields for amounts and the token pairings with some settings around slippage and gas allowance. Therefore, giving users an easy way to complete their trades while also providing an in-depth configuration if wanted.

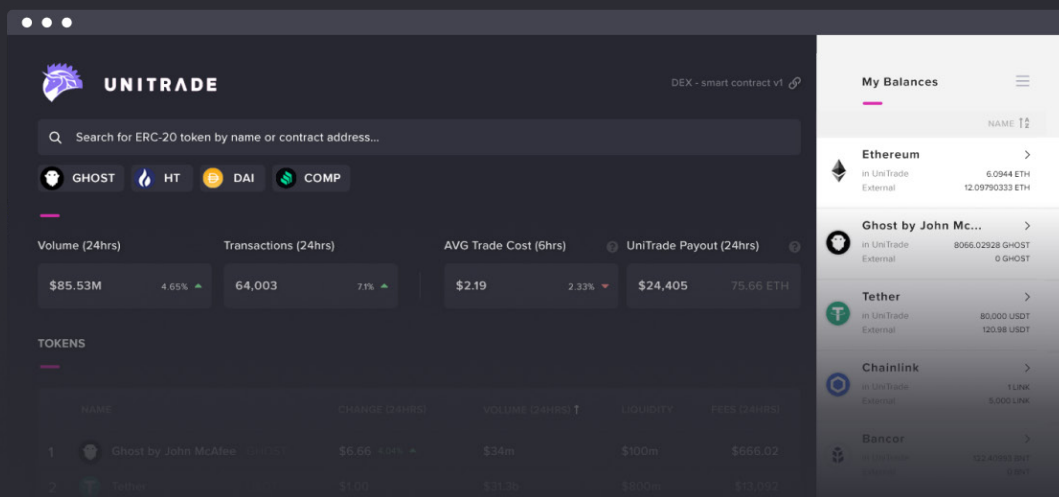
NodeJS and MongoDB

Consistently serving data from the Ethereum main-net can become a bottleneck at high traffic. For this reason, we cache the non-important data such as token pairings, price and gas estimation costs in a mongo database.

We consider this data non-important, as it's superficial and provided on-chain if the cache ever fails to deliver the data. It doesn't take any inbound user data, since it's purely read only from the UniSwap API. It doesn't store anything from UniTrade and has no ability to do so.

The NodeJS instance exists as an express instance to server the data from mongodb up to the React App. It also provides API access for the mobile app, as getting direct on-chain access from mobile applications can provide issues at store registration.

The best way to trade on **UniSwap**.



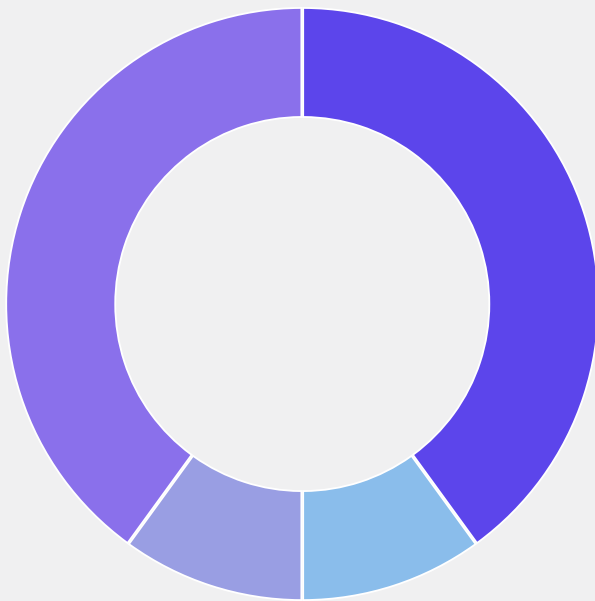
TRADE Token Information

Token Name UniTrade	Symbol TRADE	Type ERC-20
-------------------------------	------------------------	-----------------------

Total Supply: 50,000,000 **Circulating Supply:** ~22,000,000

Contract Address: 0x6f87d756daf0503d08eb8993686c7fc01dc44fb1

Token Distribution



40%
OTC Presale
Now in circulation.



40%
Liquidity & Development
Liquidity is locked.
The rest is vested over a year.



10%
Audits, Marketing, Partnerships.
Locked, and vested over a year.



10%
Reserve & Team
Locked, and vested over a year.

Roadmap

Early August

UniTrade backend prototype development, initial research and testing.

August

UniTrade airdrop and UniSwap exchange listing.

August 14th

UniTrade.app website launch.

August 20th

Whitepaper

August 22nd

UniTrade platform UI designs finalized and front-end development begins.

September 11th

UniTrade.app development beta launch.

September 14th

Third party security audit from reputable firm.

September 30th

UniTrade official platform launch!

