DEXbot White Paper

DEXbot, LLC
Admin@dexbot.io
DEXbot.io

Abstract. DEXbot is a decentralized application that acts as a sidecar to your cryptocurrency wallet, allowing you to automate selling your tokens 24/7. DEXbot has cross chain compatibility and works with any EVM compatible blockchain, side-chain or Layer 2. DEXbot is compatible with any cryptocurrency wallet that uses an Elliptic Curve Digital Signature Algorithm (ECDSA) to generate public/private key pairs.

Users can configure token settings like the amount of tokens to never to sell below, what price to never sell below, the minimum amount of tokens to sell during a transaction, the maximum amount of tokens to sell during a transaction and more. DEXbot takes the legwork and emotions out of selling your tokens so that you don't have to keep an eye on the charts.

DEXbot is designed to sell tokens without a negative impact on the token price by monitoring key stats and taking token health into consideration. DEXbot only sells directly after a buy transaction, allowing you to sell as the price goes up and without affecting the token health making DEXbot optimal for whales, dev/marketing wallets and everyday token holders.

1. Introduction

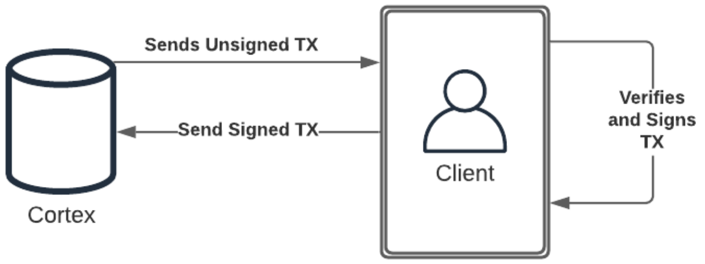
This white paper is designed to be straightforward and easy to read with the purpose of having an individual fully understand how DEXbot works after reading. This document describes the completed version of DEXbot at the end of the roadmap with all upgrades. However, there will be a section at the end explaining the current state of development as well as the roadmap for DEXbot.

2. Overview

DEXbot is a decentralized application that acts as a sidecar to your cryptocurrency wallet, allowing you to automate selling your tokens 24/7. DEXbot has cross chain compatibility with any EVM compatible blockchain, side-chain or Layer 2 and is compatible with any cryptocurrency wallet that uses an Elliptic Curve Digital Signature Algorithm (ECDSA) to generate public/private key pairs. There are four major parts to the DEXbot ecosystem. The Client application, the Scheduler, the Cortex and the Abacus. These four pillars work together to identify opportunities, build, and send transactions autonomously, while keeping the token price healthy and growing. DEXbot takes the legwork and emotions out of selling tokens so that whales, token developer teams and everyday token holders don't have to keep an eye on the charts or worry about the impact of their sells on token price.

3. Transactions

The transaction life cycle goes through a few stages of off chain logic to be built, signed and sent to the blockchain. First, a user starts the Client application, which sends a payload containing token sell settings to the Scheduler. We will go into the Client application and Scheduler in more detail later. The Scheduler then adds the wallet to a queue of user wallets that are actively waiting to receive newly built transactions sent from the Cortex to sell their tokens. Once the user is added to the queue, the Cortex begins to look for opportunities to build a sell transaction for the wallet. The Cortex is designed to prioritize token health and will only build a new sell transaction after a buy transaction has occurred on the token. Once a buy transaction has occurred that meets the user settings, the Cortex will build a new sell transaction and send it to the Client. Upon receiving the built transaction, the Client will verify the transaction parameters and ensure that the sell transaction is within the user specified settings. After verification, the Client will sign and send the transaction back to the Cortex.



Once the signed transaction is received by the Cortex, it will validate the transaction integrity, ensuring that the transaction has not been altered in any way. Following validation, the Cortex broadcasts the transaction to the network.

4. Client Application

The DEXbot Client is an open source, off chain software application that allows users to connect their wallet to DEXbot. The Client is built in the Go programming language, the same language that GETH nodes, Erigon nodes, and other popular decentralized systems are built in. The Client is designed to be lightweight, fast and easy to spin up on any machine. Since the release of Go version 1.17, Go can compile for a wide variety of operating systems and architectures which means that the DEXbot client can run on virtually any machine.

The Client is responsible for configuring wallet/token sell settings, listening for incoming sell transactions sent from the Cortex, verifying transaction integrity, signing transactions and sending signed transactions back to the Cortex. All data being sent to and from the Client goes through an intermediary database acting as an insulator between microservices. This database also keeps track of the wallet queues for tokens and transaction status.

When configuring the Client, users can create unique token settings for every token the user enters into DEXbot. For example, a user could add a single wallet with 10 tokens, and configure the sell settings for each token to be different. Settings include the minimum USD sell price, a minimum token balance to never sell below, the minimum 24 hour price change, the minimum amount of native tokens to sell per transaction, the maximum amount of native tokens to sell per transaction and the minimum wait time between sells. Users can also run multiple wallets simultaneously on DEXbot. For example, if a user wants to connect a Polygon wallet with 5 tokens, a Binance Smart Chain wallet with 3 tokens and another Polygon wallet with 4 tokens, this is possible. There is no limit to how many wallets or tokens that a user can connect to the system.

Once the user tells the Client to start listening for new transactions, they will be prompted to enter their private key for each wallet they have connected. This allows for the Client to autonomously sign and send the transaction back to the Cortex after verification and validation of the transaction integrity. It is crucial to note that there are important steps that DEXbot takes to ensure safety and security of the private key.

First, DEXbot never stores, retrieves, sends, logs, or shares the private key in any capacity, at any point. The private key is completely local to your machine and is only used when signing transactions. Furthermore, after entering your private key, the Client completely wipes the byte slice containing the private key so that it only exists in the data structure that is used to sign transactions. The Client has no persistence and when the application is stopped, this data is destroyed. To ensure maximum safety, make sure you are only using official releases of trusted operating systems and taking proper measures to secure your computer. If you want to take a deep dive into the source code, the entire Client application is open sourced and available on the official DEXbot github repo.

When listening for transactions, once the Client receives a built transaction from the Cortex, the Client takes a few steps to validate and verify the parameters of the transaction. The Client checks that the “to” address is the specified swap router for that chain (Uniswap, Pancakeswap, ect.). In later updates of the system, the “to” address will be a DEXbot smart contract that simplifies the DEXbot transaction life cycle. Then the client checks the transaction data to validate the method signature, the swap path and the amountOutMin. After validation and verification, the transaction is signed and sent back to the Cortex. The Cortex then broadcasts the transaction and the Client is added back into the wallet queue for that token.

If the Client application is stopped, the Scheduler will remove the user from all token queues and transactions will not be sent to the user until they reconnect their Client. The source code will always be open source and available on the official DEXbot github repo. Every line is commented so that anyone can read through and understand what the code is doing, even if you are not a developer.

5. The Scheduler

The Scheduler is the next core pillar of the offchain logic that manages Client connections and token queues. The way the Scheduler works is very straightforward. When a Client connects to the system, the Scheduler checks the payload for the Client's wallet/token configurations and adds the wallet(s) to the respective token queues. When a Client is sent a newly built transaction, they are removed from that queue. After the transaction is broadcasted to the network, the specific wallet is added back into the token queue. Along with adding users to the queue, the Scheduler manages removing the user from all token queues when the Client disconnects from the system.

6. The Cortex

The Cortex is the third core pillar of the offchain logic that is responsible for listening to the blockchain to find sell opportunities, build new transactions and send them to the Client. As previously mentioned, the Cortex will only create a sell transaction directly after there has been a buy transaction on the token. Every sell transaction will be slightly smaller than the buy transaction that triggered the Cortex. This ensures that the sells created by DEXbot do not affect the token health and there is always a net positive effect on the token price. This design allows for whales, token developers and everyday token holders that have a vested interest in seeing the token price rise, create sell transactions in a way that has a net positive effect on the token.

Once a buy transaction trigger is identified, the Cortex identifies the first wallet in the queue and checks the wallet's token sell criteria. If the sell transaction parameters fall within the wallet's sell criteria for that token, a new sell transaction is built and sent to that Client. If the sell transaction parameters do not fall within the wallet's token sell configuration, the Cortex searches the queue for a wallet where the token sell configuration matches the sell transaction parameters. All swap transactions made by the Cortex result in the wrapped native token. For example, if the transaction is on Polygon, the resulting token out will be $WMATIC and if the transaction is on the Binance Smart Chain, the resulting token out will be $WBNB.

After receiving a signed sell transaction from a Client, the Cortex verifies the integrity of the signed transaction against the originally built transaction, making sure that the transaction sent to the Cortex is unaltered and exactly the same as when it was sent. After verifying the signed transaction's integrity, the transaction is broadcasted to the network to be committed to the blockchain. To give an idea of how fast this logic currently executes, once a buy transaction trigger is identified, the Cortex finds an eligible user wallet, builds a new transaction and sends it to the Client in ~1 second on average.

7. The Abacus

The Abacus is the fourth and final core pillar of the DEXbot ecosystem that is responsible for paying the user wallet the amountOut from the swap transaction. The Abacus has two major stages during the development roadmap. The first stage is an off chain Abacus that tracks swap transactions made by DEXbot and pays the amountOut from each transaction to the respective user wallet, minus a 2.5% DEXbot fee. The second stage of the Abacus is an on-chain smart contract that converts the entire swap transaction and payout process into a single trustless transaction.

We have decided to start with an off-chain Abacus to give our development team the time and resources to build a secure and gas efficient contract that will help strengthen the DEXbot ecosystem and benefit DEXbot users. Security and gas efficiency are two of the biggest problems with many smart contracts and we want to take the time and attention to make sure we build a secure, audited smart contract that uses the most efficient approach to save the end user money on gas fees. While we are working on this contract, we wanted to develop a way for users to utilize DEXbot in the meantime, and that solution was an off-chain Abacus.

Taking a look at how the off-chain Abacus works, it can be broken down into a few steps. First, since all transactions are tracked throughout the DEXbot database, the off-chain Abacus knows the second that a swap transaction happens through DEXbot. Upon receiving notification of a new swap transaction from DEXbot, the Abacus looks up the transaction by hash to identify the transaction details including the amountOut that was sent during the transaction. The Abacus then sends the amountOut in the native token minus a 2.5% transaction fee. So for example, if the amountOut is 10 $WBNB, the Abacus will subtract 2.5% and send the user wallet 9.75 $BNB. To give a reference for how fast this off-chain process happens, the average turn around time from the moment that a buy trigger is identified by the Cortex, to the moment that the Abacus payout transaction is broadcasted and confirmed, is about 5-7 seconds on the Binance Smart Chain.

Now lets take a look at how the on-chain Abacus works. Since the on-chain Abacus is a smart contract, that means that everything can happen in one transaction and there is no more need for off-chain logic to listen to swap transactions. Instead of the Cortex building and sending a swap transaction with the decentralized exchange (ex. Uniswap, Pancakeswap, ect.), it will build a transaction for the on-chain Abacus, which wraps the swap function for the decentralized exchange. The on-chain Abacus transaction will take the exact same parameters as the swap transaction as well as a few extra parameters for the Abacus to handle payouts. Then in one single transaction, after broadcasting to the network, the on-chain Abacus will swap the tokens, and payout the amountOut owed to the user wallet, minus the 2.5% DEXbot fee. The result is a trustless process that has zero downtime and completely transparent source code. The trustless Abacus is on the DEXbot roadmap and is expected to be deployed in 2022.

8. Referral Rewards

Each user that creates an account on DEXbot.io will receive a unique referral URL that can be shared to friends, family, and on social media platforms. When a user signs up for an account using your unique referral URL, the referral will be attributed to your account for life. If the referred user makes token sells with DEXbot, you will receive .125% of every transaction they make (this percentage comes out of the DEXbot fee, not the amount sent to the referred user). There are no limits to the amount of rewards that you can receive, and no limit to the amount of users that can sign up with your referral code.

Referral rewards are paid out each time $5.00 USD has been accumulated from total referral rewards, per chain. For example, if you have accumulated $10.00 USD worth of referral rewards in $MATIC (Polygon native token) and $4.00 USD worth of referral rewards in $BNB (Binance Smart Chain native token), you will receive your referral rewards for Polygon, but will not yet receive your rewards for the Binance Smart Chain (BSC). To accumulate referral rewards for each chain, you must enter a wallet address that rewards will be sent to. If you only have a wallet set up for Polygon, you will not receive referral rewards on BSC. You can set a wallet address to receive referral rewards in your account settings on DEXbot.io.

9. Roadmap/DEXbot Ecosystem Updates

In 2022, the development team behind DEXbot plans to roll out a few major updates that will help to build the DEXbot ecosystem and improve the existing codebase. We are always actively listening to the DEXbot community and plan to roll out more updates as DEXbot grows. Below are a few updates that will be implemented in 2022.

- Launch referral rewards.
- Deploy on Polygon.
- Additions to the Client application validation logic before signing incoming transactions.
- Updates and improvements to the Client application to maximize runtime efficiency.
- Runtime efficiency improvements for the Cortex, Scheduler and off-chain Abacus to increase off-chain logic speed.
- Update the Abacus to be trustless by transitioning from off-chain logic to a smart contract that will handle swap and payout transactions.