# dTWAP for Decentralized Exchanges (DEXs)

The past several years have brought new levels of sophistication to the DeFi space, with an increasing number of users accessing DEXs and AMMs to execute trades. At the same time, due to the current limited capabilities of smart contracts, DeFi markets lack features that could help manage the heightened volatility and liquidity issues that characterize the current crypto landscape.

Orbs has pioneered the concept of L3 infrastructure, which utilizes the project's decentralized network to enhance the capabilities of EVM smart contracts. By relying on Orbs validators as off-chain bidders, the project has developed a new decentralized TWAP protocol for DEX/AMMs, which allows for advanced time-spread trades to be executed on these platforms in a decentralized manner.

This lightpaper outlines the concept and architecture behind Orbs' new decentralized TWAP protocol and a prototype UI that will allow DEXs to harness this technology to benefit the platform and its users.

## 1. Intro to TWAP

TWAP (Time-Weighted Average Price) is a common trading executing strategy in CeFi that seeks to minimize a large order's impact on the market by dividing it into a number of smaller trades and executing these trades at regular intervals over a longer period of time.

TWAP trades can be divided into market orders and limit orders. In a market TWAP order, the trader sets parameters, the size of the desired trade, the total duration of the strategy, and the intervals between individual trades. The trades are then executed at the predetermined intervals at the best available current market price.

Limit orders are similar, except that in these strategies, the trader sets a limit on the price at which trades will be executed. If a price within the limit is not available at a given interval, the individual trade will not be executed. In this version of the strategy, the larger total order may only be executed partially by the end of the duration period.

## 2. Use-Cases for TWAP

There are two primary benefits of utilizing TWAP trading strategies.

### 2.1. Price Impact Reduction

TWAP strategies reduce the impact of an order on the general market price.

One scenario where price impact causes significant losses to traders is with large orders. However, even smaller orders can suffer from price impact, particularly when the trading pair is long-tail and low liquidity.

The problem of an individual trade having a disproportionate effect on the market, for either of these reasons, can be especially acute in DeFi where liquidity is much more fragmented than it is in the traditional financial markets. This fragmentation often exists even within individual large DEXs, which offer multiple pools for the same trading pairs.

With this type of fragmented liquidity, even trades that are not large in proportion to the overall market can create outsized price impact within the specific smaller pool in which the trade is executed. In addition, fragmentation may exacerbate problems for assets that are already low-liquidity (such as more obscure altcoins) by dividing the liquidity that does exist among a number of smaller pools, thus rendering each even less liquid.

TWAP strategies, even those with relatively short durations (i.e., executing a trade in intervals of 1-2 minutes over a period of 15-20 minutes), can mitigate this problem by giving arbitrageurs a short window to close any price discrepancies on the affected pools and bring the reserves back to equilibrium (on par with spot price). Optimal TWAP trading requires setting the time period to be relatively short to enable the trader to limit any exposure to price volatility that could occur over a longer period of time, while still giving the arbitrageurs sufficient time to close the gaps. In the context of a TWAP trade on a blockchain system, these factors also need to be balanced against the consideration that increasing the number of trades requires incurring greater gas fees.

In addition to benefiting traders, this type of TWAP strategy also benefits the trading platform itself, as minimizing the risk of price impact can lead to increased liquidity on the pool and thus increased trading volume.

## 2.2. Automation of Dollar-Cost Averaging (DCA)

Dollar-cost averaging (DCA) is an investing strategy where the investor purchases an asset or set of assets having a certain dollar value on a specific schedule (i.e., on the first day of every month). The goal behind DCA is to average out abnormal market conditions, particularly when the trader lacks strong conviction regarding the future direction of the market.

TWAP trades can be used to generate an automated version of this strategy, typically by entering a market order with longer intervals that lasts for a longer period of time or even perpetually. This trade can essentially serve as an automated DCA bot that requires no additional action from the investor.

Note that in a blockchain context, this order may be visible on chain and thus predictable, so to avoid price manipulation it may be better reserved for large cap assets that are not easily prone to price manipulations.

These are only two of the primary benefits of TWAP trading, which can also be used as part of more sophisticated strategies, including short-selling and hedging.

## 3. dTWAP Protocol - Components and Players

While the benefits of TWAP transactions are clear, the current limitations of EVM smart contracts make them impossible to implement in a decentralized manner. Orbs' L3 infrastructure solves this problem using the new dTWAP protocol in a fully decentralized and secure way.

The dTWAP protocol smart contract has been developed to work on-chain with a permissionless set of bidders. A customized version can be designed for any specific DEX to create a user-friendly version that works seamlessly and exclusively within that DEX's current offering.

The components of the dTWAP protocol are as follows:

1. User-friendly and flexible UI - the Orbs team has created a UI that can be modified for integration into any DEX's current UI. Through this UI, DEX users can, after entering their desired parameters and conditions, send their TWAP orders to the dTWAP smart contract. In addition to the user's choice of parameters, orders sent to the dTWAP smart contract via the DEX's specialized UI may also include hard-coded elements that are specific to the specific DEX integration (such as the requirement that all trades be executed in one of the DEX's pools).

2. dTWAP Smart Contract - the backbone of the dTWAP protocol, the dTWAP smart contract holds orders sent by the originating traders (known as "makers"). ). The smart contract then receives bids from takers who find the best path for the next given trade in a given order. The smart contract selects the winning bid, which then triggers the execution of a transaction in which the maker's assets are sold on a DEX through the winning path and the winning bidder takes the fees.

3. Orbs Guardians - (honest bidders) - in order to ensure that the protocol described above functions effectively as close as possible to spot prices, it is sufficient to have one honest bidder who is willing to limit their fee to the amount needed for gas fees. The Orbs Network's validator nodes (known as "Orbs Guardians") will utilize Orbs' new L3 infrastructure to act, in a decentralized manner, as the required honest bidder.

4. Third Party Routers - Dex's will have the option, as part of the integration, to elect whether to utilize the DEX's own default router or to use a third party route optimization service to locate the best given path for TWAP trades on the DEX. If the choice is made to use a third party router, the Orbs Guardians will utilize the applicable API as the first option in generating the paths for the bids they will present while providing the service of acting as an honest bidder. In the event that the third party option is not a viable option, or it fails for any reason to return an appropriate path, the Orbs Guardians will utilize the existing DEX default router UI as a fallback option.

The following sections provide in depth explanations of the proposed generic dTWAP UI template, the dTWAP Smart Contract and the role that will be played by the Orbs Guardians.

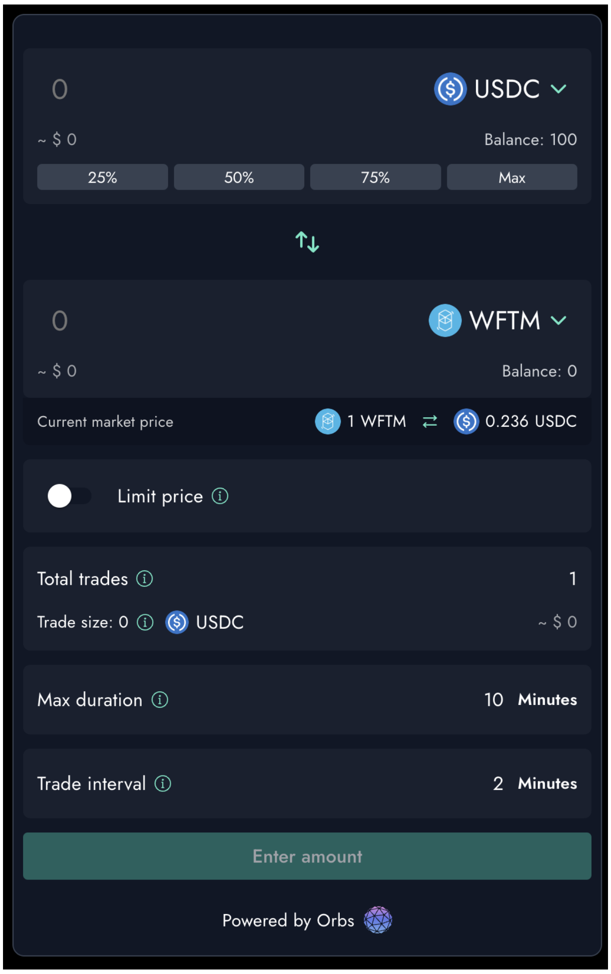
## 4. The dTWAP UI

While the general dTWAP Smart Contract is open-source and can be utilized or integrated with any DEX, the Orbs team has designed a specialized UI template which can be easily integrated by any specific DEX. The UI is designed to seamlessly fit into most DEXs current offering and create an intuitive, user-friendly and informative way for the DEX users to set the parameters for  and initiate TWAP trades.

**The screenshots below were taken from dTWAP integration to SpiritSwap DEX-AMM. However the same can apply to any other DEX-AMM platform.*

The first page of the UI is currently designed as follows for market orders:

When the limit order option is toggled, the front page is designed as follows:



0 USDC

~ $ 0 Balance: 100

25% 50% 75% Max

0 WFTM

~ $ 0 Balance: 0

Current market price 1 WFTM ⇄ 0.236 USDC

○ Limit price ⓘ

Total trades ⓘ 1

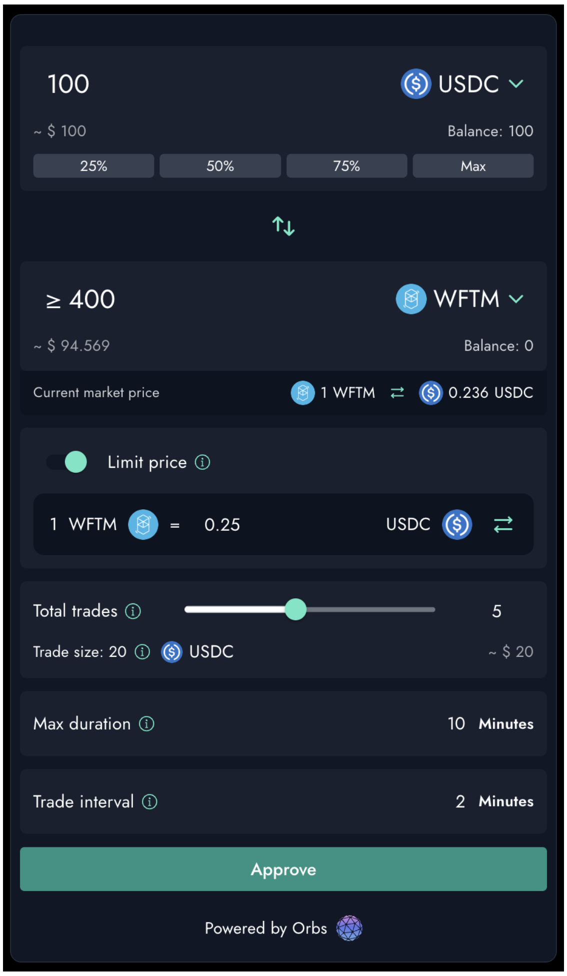
Trade size: 0 ⓘ USDC ~ $ 0

Max duration ⓘ 10 Minutes

Trade interval ⓘ 2 Minutes

Enter amount

Powered by Orbs



100 USDC

~ $ 100 Balance: 100

25% 50% 75% Max

≥ 400 WFTM

~ $ 94.569 Balance: 0

Current market price 1 WFTM ⇄ 0.236 USDC

● Limit price ⓘ

1 WFTM = 0.25 USDC ⇄

Total trades ⓘ 5

Trade size: 20 ⓘ USDC ~ $ 20

Max duration ⓘ 10 Minutes

Trade interval ⓘ 2 Minutes

Approve

Powered by Orbs

This initial page of the UI allows users to set the following parameters for their TWAP order:

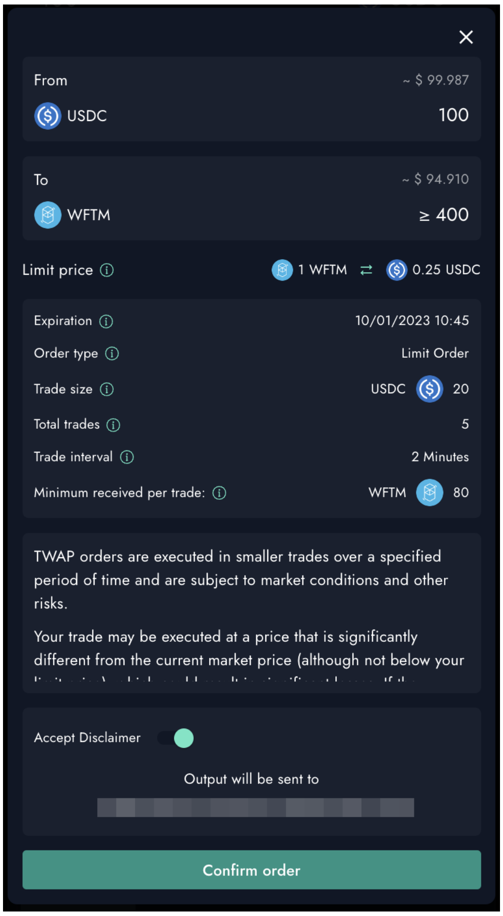
- From: the source token that the user holds and intends to swap. As a default ,it is suggested that the UI constrain a user’s ability to sell their entire holdings of the selected source token at once and to add options to swap 25%, 50% and 75% of their current balance.
- To: the destination token that the user wants to receive in the swap in exchange for providing the source token. This field will also show an estimate based on the current

price (which is also displayed in this field). A “tool tip” providing a clear disclaimer and explanation that this is only an estimate and that actual results will vary depending on the market condition and the other parameters selected by the user.

- Limit Price: Users will have the option to add a “limit price”, thereby turning their order into a limit order. If this option is not selected, the default is a market order with no constraint as to what price trades may execute.
    - If a user selects the limit price option, a new field will open in which they can set the desired lowest price at which transactions will execute. Setting the limit price, as well as any changes to it, will also change the estimate in the “To” column to reflect the new limits.
    - Once a limit price is set, trades will only be executed when the limit price is better than or equal to the market price offered through the bidding mechanism in the smart contract set below.
- Max. Duration: the total time during which the total amount of individual trades making up the full TWAP order may be executed.
- Total Trades: the user can set the number of individual trades that will be scheduled as part of the TWAP order. The UI then automatically calculates the size of each individual trade, the estimated duration and the trade interval.
- Trade Interval - sets the minimum amount of time between each individual trade. In order to allow sufficient time for the taker bidding auction for each individual trade, the trade interval must be at least two minutes.

Once all of these parameters are set, the user will have the ability to Approve the specific source token. This will require signing a blockchain transaction that grants an approval to the dTWAP smart contract to transfer the applicable token out of the user’s wallets (only in accordance with the parameters and conditions set by the user and the smart contract).

Once this approval has been given, the user can elect to place their order, which brings them to the following confirmation screen:



| | |
|---|---|
| From | ~ $ 99.987 |
| USDC | 100 |
| To | ~ $ 94.910 |
| WFTM | ≥ 400 |
| Limit price ⓘ | 1 WFTM ⇄ 0.25 USDC |
| Expiration ⓘ | 10/01/2023 10:45 |
| Order type ⓘ | Limit Order |
| Trade size ⓘ | USDC 20 |
| Total trades ⓘ | 5 |
| Trade interval ⓘ | 2 Minutes |
| Minimum received per trade: ⓘ | WFTM 80 |

TWAP orders are executed in smaller trades over a specified period of time and are subject to market conditions and other risks.

Your trade may be executed at a price that is significantly different from the current market price (although not below your

Accept Disclaimer ●

Output will be sent to

[redacted]

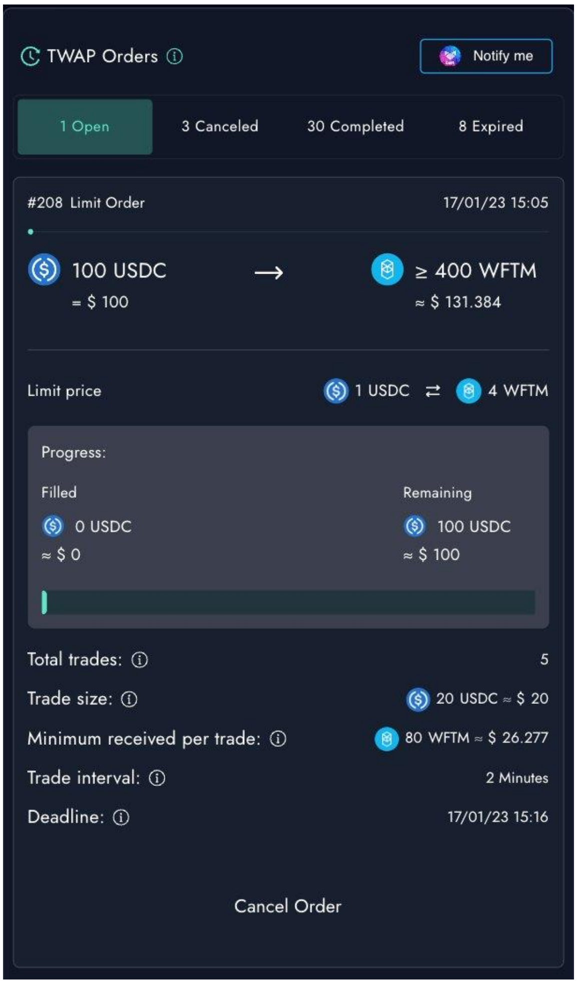
The “confirm order” screen shows the basic components of the user’s proposed trade order, based on the parameters set in the previous screen. These components are:

- From: Reiterates the amount of source tokens the user elected to sell as part of the swap.
- To (estimate): Repeats the estimated amount of destination tokens that the user may receive, together with the explanation and disclaimer noting that actual results may vary.
- Expiration Date: Based on the duration set on the previous page, the UI will calculate the date and time at which the user can expect the dTWAP trade to end (whether the order is fully filled or not).
- Limit Price: If the user previously selected this option, the limit price set.
- Trade Size: Repeats the number of source tokens the user has set to swap in each trade.

- Minimum Received Per Trade: If the user has not set a limit price, this field will say "None", with a tool tip explaining that since they have chosen a market order, there is no set minimum number of tokens they will receive and actual results will depend on market conditions. If a limit order has been chosen, the minimum number of tokens that may be received in a trade will be shown. A tooltip will inform the user that this only refers to executed trades, and that some trades may not be executed if the limit price is worse than the available market prices. This corresponds to the minAmountOut sent to DEX/AMM swap routers.
- Total Trades: Shows the number of individual trades that will be proposed as part of the TWAP order.
- Trade Interval: Shows the amount of time that will elapse between each trade.

A longer text provides additional disclaimers and warnings prior to the user clicking the confirm button. This will result in their being requested to sign using their blockchain wallet, sending an "Ask" transaction that is then sent to the dTWAP smart contract. This transaction initiates the process described in the next section.

Users can continuously monitor their Open/Filled/Expired/Canceled dTWAP orders status in the SpiritSwap UI. Open orders can be canceled at any time by the user. In addition, users can also set up free, on-chain mobile notifications using the Open DeFi Notification Protocol powered by Orbs, for real-time notifications of their dTWAP order status.



TWAP Orders ⓘ

#208 Limit Order 17/01/23 15:05

100 USDC → ≥ 400 WFTM

≈ $ 100 ≈ $ 131.384

Limit price: 1 USDC ⇄ 4 WFTM

Progress:

Filled: 0 USDC ≈ $ 0

Remaining: 100 USDC ≈ $ 100

Total trades: ⓘ 5

Trade size: ⓘ 20 USDC ≈ $ 20

Minimum received per trade: ⓘ 80 WFTM ≈ $ 26.277

Trade interval: ⓘ 2 Minutes

Deadline: ⓘ 17/01/23 15:16

## 5. The dTWAP Smart Contract

The dTWAP smart contract allows the incentivized execution of a TWAP order (either a  Limit Order or a Market Order) on any DEX, with the possibility of partial fills. A TWAP order breaks a larger order down into smaller trades or "chunks", which are executed over a set period of time.

In this smart contract, users (makers) create orders that wait in the contract to be filled. Once made, these orders enable an English Auction bidding war on each chunk at its time interval. Anyone willing to participate can serve as a “taker” by finding the best path to fill the order for the next chunk on any DEX, within the parameters set by the maker.

When integrating the dTWAP protocol to a specific DEX, the DEX's users will be the makers, and the UI can ensure that the parameters set by any order they make specifies that their orders may only be filled on one of the specified DEX's  pools.
Takers submit the best path they have located as part of a bid to the contract, which selects the winner based on criteria described in detail below.
The winning taker receives a portion of the output tokens as a reward for their effort.
One honest taker (i.e., a taker who is willing to set the fee at the minimum amount needed to cover gas costs) is enough to ensure the entire system functions effectively at spot prices.
The contract is set to operate only up to the year 2106 (32bit timestamps), at which point it will no longer be usable.
The dTWAP Smart Contract does not hold any funds, has no owners, administrators or other roles and is entirely immutable once deployed on an EVM blockchain.

**Integrate your DEX: GitHub README**

## Architecture

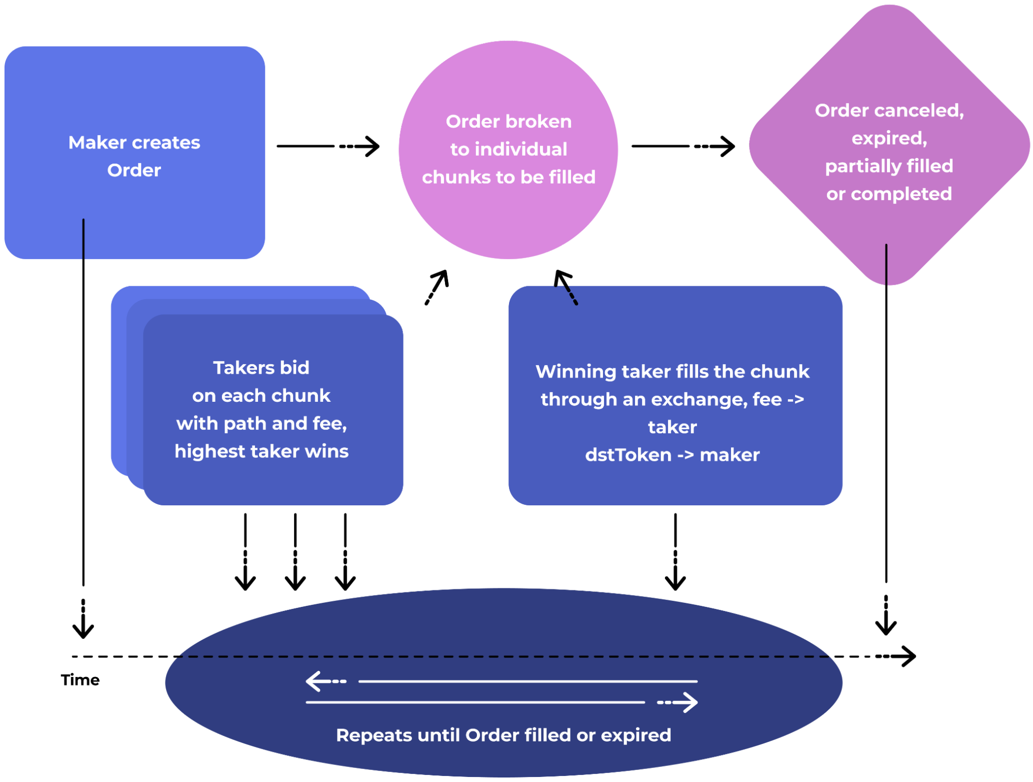
Actors

- Maker: User, the Order creator. Signs srcToken approval and swap transactions by the dTWAP contract on a specific (or any) exchange
    - Controls all Order restriction parameters such as limit price, expiration and the length of delay between chunks
- Taker: Incentivized independant participators that monitor Orders submitted by makers
    - Takers try to find the best path for relevant chunks and submit bids for those chunks, including a fee for the taker

fee: 0 or more of the dstToken output (i.e., the token the maker is purchasing) to be sent to the taker at the time the chunk is filled.

- Spends the effort needed to find the best path, and risks being out-bid in the bidding war by another taker with a better path or lower fee.

State diagram and execution flowchart


Maker creates Order
Order broken to individual chunks to be filled
Order canceled, expired, partially filled or completed
Takers bid on each chunk with path and fee, highest taker wins
Winning taker fills the chunk through an exchange, fee -> taker dstToken -> maker
Time
Repeats until Order filled or expired

The dTWAP contract holds the order book, which is an array of all Orders.

Before an Order can be filled, makers sign a transaction giving approval to the dTWAP contract to remove the applicable source asset from their wallet.

Order is created by a maker, who, after granting the approval, sends an Ask transaction to the dTWAP contract containing certain requested parameters (specified below).

Takers monitor the Orders held by the contract, and can send bid transactions specifying the DEX on which it proposes to execute the next chunk, the output amount it can receive and the requested fee.

As a result of this process, an Order is generated and held in the dTWAP contract, which contains the parameters and constraints specified in the library.

Once an order is created, it waits in the contract to be filled, when viable.

The smart contract checks every bid transaction received to ensure that the following conditions are met:

- The order has not been previously canceled, and the deadline has not passed

- The maker granted the dTWAP contract an approval to swap the srcToken, with the approval covering an amount that is high enough to fill the next chunk.
- The maker has a high enough balance of the srcToken to be swapped (for the next chunk)
- The last chunk of the order was not filled too recently and there has been a sufficient delay between chunks.
  - The delay is set by the order maker
- If the order maker specified that a particular exchange would be utilized for the order, only that exchange can be used to swap. If the value is set to the zero address, any exchange can be used.
- The current bid output after fees and slippage is higher than the previous winning bid, by more than MIN_OUTBID_PERCENT
- The current bid output after fees and slippage is higher or equal to the minimum set by the order maker (limit price)
- A bid can be out-bid by a lower output when that bid is stale. A bid is considered stale when STALE_BID_SECONDS have passed since that bid was placed and wasn't filled. This mechanism avoids denial of service attacks on orders.

Any invalid constraint will revert the bid transaction, so a successful transaction implies the bid has won

A winning bid can be filled (executed) only after a minimum delay of MIN_BID_DELAY_SECONDS, with the specific time period set by the user. This delay gives other bidders an opportunity to make a competing bid, allowing for a bidding war on the applicable chunk.

- Each successful bid allows for another bid delay interval to challenge it
- If no other bid is set as the new winner, the current taker (winning bidder) can fill the bid by sending a fill transaction.
- When receiving the fill transaction, the TWAP contract performs the same verifications as when bidding.
- Once verified, the fill transaction also utilizes the approval previously provided by the maker to perform the actual swap on the requested exchange, resulting in the transfer of the input tokens from the maker to the exchange, and the transfer of the output tokens from the exchange back to the maker.
- If dstFee is set >0, the specified fee is paid out to the winning taker as part of the completion of the order, out of the dstToken amount of that swap.
- All of the above transfers are contained and set within the valid fill transaction.

- OrderFilled event is emitted

This structure creates an incentive for takers to find a path that is advantageous for the bid, as a higher return of output tokens enables the fee to be as high as possible

But, the presence of one honest taker, who is willing to set the fee such that it takes only enough to reimburse it for the gas fees incurred, will ensure that the order will be filled to as close a price to the spot market as possible. This is because this honest taker, by charging a lower fee, will propose a transaction containing the highest output to the maker and will therefore be selected as the winning bid when competing with those charging higher fees, all other things equal. Therefore, if the honest bidder is equally capable of locating the best path as any competing taker, the price received by the maker will be optimal.

Please note the following additional features of the dTWAP contract:

- An order can be canceled any time by the maker, effectively setting deadline to 0.
- An order can be only partially filled, due to market price volatility (unfilled limit orders), or expiration.
- The maker must ensure that it has granted approval and has a sufficient balance of srcToken for each chunk before that chunk is available for bidding. It does not have to have the entire amount available and approved at the time of order creation.

## 6. Orbs Guardians

The Orbs Network is an open, decentralized and public blockchain infrastructure executed by a secure network of permissionless validators (known as "Guardians") using Proof-of-Stake (PoS) consensus. Orbs is optimized to provide "L3" services, working in conjunction with existing L1 and L2 layers and acting as a "decentralized backend" that enhances the capabilities of EVM smart contracts. Orbs Network mainnet is live since 2019 and has dozens of active validators staked with over $100M.

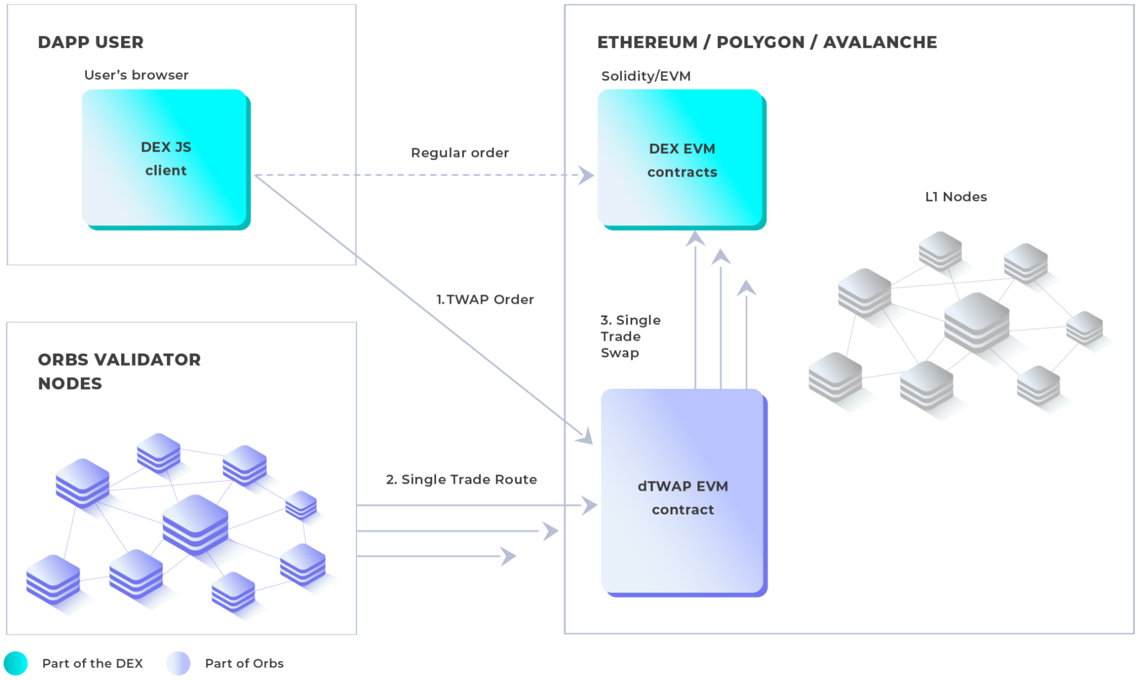
The network provides its L3 services by operating as a decentralized serverless cloud that allows developers to design applications that extend the capabilities of their smart contracts without relying on traditional centralized server solutions. These applications are deployed to be executed by the Orbs Guardians in a decentralized way.

Orbs' L3 offering is divided into two categories - ORBS-VM and ORBS-Lambda. ORBS-VM is a docker-compatible decentralized virtual machine, similar in concept to AWS EC2. This is designed for more complicated functions.

For simpler functions, Orbs Network offers ORBS-Lambda - a decentralized and serverless cloud function solution, similar to AWS-Lambda. Developers can write cloud functions and, when the specified trigger occurs, Orbs Guardians then execute the function on the specified L1 blockchain in a decentralized way.

As part of the dTWAP protocol, Orbs Guardians will run software utilizing ORBS-VM that acts as the single honest bidder that is required for the dTWAP protocol to run optimally and achieve prices that track the spot market price as closely as possible.

The application run by Orbs Guardians will, in a decentralized manner, monitor the dTWAP Smart Contract. When an order has been sent and a chunk is open for bids, Orbs Guardians will automatically calculate and submit an honest bid. The fee component of the bid will include only a request to be compensated for the estimated gas fees.



In determining the path for the transactions that they will propose as part of their bids, Orbs Guardians can utilize either a DEX's default router or a third party service, depending on the parameters set during the integration.

By automatically creating honest bids that are hardcoded to only specify a minimal gas fee and utilizing the best available routing tools selected by the DEX doing the integration, the Orbs Guardians will provide a strong level of assurance that the trades executed through the DEX's UI and the dTWAP contract track spot market prices as closely as possible.

## 7. Conclusion

Harnessing the ability of Orbs' unique L3 infrastructure to expand the capabilities of the EVM smart contracts can allow DEXs to provide highly efficient TWAP orders to their user bases without sacrificing decentralization. With Orbs' decentralized backend ensuring that TWAP orders are executed at an optimal price and at fair fees, this type of trade can become a viable option for DEX-AMMs users, giving them a number of new ways to make their trading activities more sophisticated. In turn, DEXs themselves can benefit from increased liquidity and attract new users by offering features that are unavailable on competing DEXes that do not utilize this technology.