1.1 Introduction

The darkpool is an alternative trading system popular in the traditional financial markets. Its primary purpose is to allow traders to transact with guaranteed privacy of their order and trade information from the public. This is not only of general use to users who value privacy, but also provides economic benefit to traders and marketmakers dealing in large size who do not want to leak information or move the market with their orders. This is in stark contrast with today's decentralized exchanges, typically hosted on on-chain smart contracts, where all the information is publicly traceable on the blockchain.

Phalanx's mission is to faithfully recreate a cross-chain, fully decentralized darkpool environment for crypto traders, built on top of the Phala parachain in the Dotsama ecosystem. Its main features are order/trade privacy and mid-price clearing.

1.2 A Summary of Crypto Exchanges Today

Orderbook exchanges, the predominant and most robust type of exchange in traditional financial markets, are currently problematic to implement in the decentralized world. This is because traders/market-makers need to send their orders as transactions on the blockchain, requiring gas fees everytime they need to send, update, or cancel an order. This is economically infeasible, especially for marketmakers who typically update their orders every second or even fraction of a second in response to market moves.

AMMs solved the transaction fee problem that plagued orderbook DEXes by getting rid of marketmakers and orderbooks in favor of new concepts such as liquidity providers and constant product pricing. However, they are not perfect. Some issues plaguing AMMs include liquidity providers sudden pulling liquidity, sudden price changes or slippage costs during low liquidity, and order frontrunning exploiting to the public mempool of blockchains.

These limitations must be kept in mind when designing a new DEX. Phalanx's *order/trade privacy* handles the frontrunning problem. Phalanx's *mid-price clearing* mechanism handles transaction fee problem. Moreover, Phalanx provides a novel, innovative venue for traders to transact that does not currently exist in the DeFi world.

1.3 Building on Phala

Phala has two primary improvements over current-gen L1 chains. Firstly, Phala utilizes a hybrid blockchain/hardware solution (called TEE, or Trusted Execution Environment) to guarantee privacy of smart contracts and transactions. Second, since smart contract logic is executed natively inside the TEE, Phala smart contracts are able to make HTTP requests natively (as opposed to Ethereum smart contracts, which require on-chain oracles). Phalanx leverages both features heavily for its implementation, making Phala the natural choice on top of which to build.

2.1 Orders and Order Matching

To submit an order on Phalanx, users need to specify four key inputs:

market The pair they wish to trade (e.g., DOT/USDT, PHA/USDT, GLMR/USDT, etc). The system is flexible enough to allow trading of cross-chain native coins, any associated ERC20 tokens, as well as wrapped coins/tokens from other chains not part of the non-Dotsama ecosystem.
orderExpiryTime UNIX timestamp, after which this order expires and is no longer valid
side Boolean representing whether the user is a buyer or seller

size The amount the trader wishes to transact

When a user submits an order into Phalanx, they are entered into a FIFO queue. A buyer submitting an order will enter the bids queue, and a seller submitting an order will enter the asks queue. A trade is executed when there is simultaneously a buyer in the bids queue and a seller in the asks queue. Hence, the bid and ask queues cannot both be non-empty at the same time, since trades would continue clearing until at least one side is empty.

Important to note is that users never need to specify what price they wish to trade at. The price at which trades are executed is determined by Phalanx's mid-price clearing mechanism.

2.2 Mid-Price Clearing

The price at which trades are executed is determined by a set of external reference prices provided via HTTP request, which is made possible due to Phala's unique features. These external reference prices can be a set of centralized and/or decentralized exchanges. The overarching logic is that some theoretical average mid price of the market in real-time will be used for the execution price. However, some care must be taken when using an external source for real-time price data from a security standpoint. We filter the prices through three processes: Size-Weighted Mid Pice, Stale Rate Filter, and Outlier Filter.

2.2a Size-Weighted Mid Price

First, we define the *midPrice* for any particular exchange. We assume the exchange is an orderbook exchange. The *minSize* is a parameter used to weight the bid and ask prices. The weighted bid and ask prices are then averaged together to get the *midPrice* for that exchange.

Example: We represent the orderbook bids/ask as an array of (price, size) pair. Suppose bids = [(5, 5000), (4.9, 5000)] and asks = [(5.1,4000), (5.5,6000)], and *minSize* = 10000. Then, the weighted *bidPrice* will be (5*5000 + 4.9*5000) / (10000) = 4.95 and similarly, the weighted *askPrice* will be (5.1*4000 + 5.5*6000) / (10000) = 5.34. The *midPrice* is then (4.95 + 5.34)/2 = 5.145. This weighted price represents a better picture of where the theoretical mid price is for traders dealing with significant order sizes, and also ensures that malicious traders cannot manipulate the exchange price by flashing orders with erroneous prices for small sizes.

2.2b Stale Rate Filter

We must also ensure that prices from each exchange must be recent. If the time elapsed since the timestamp of the last price update from an exchange is is greater than the *staleTime* parameter, it should be discarded.

2.2c Outlier Filter

Lastly, outlier prices from any exchange should be also discarded. To accomplish this, we cache the *lastGoodMidPrice* from the last instance where a known good price is established on the platform. A +/-% band is applied on the *lastGoodMidPrice*, and any *midPrice* returned from an exchange falling outside of the band is discarded. Suppose the remaining good prices are represented by an array *midPrices* = [midPrice_1, midPrice_2, ...]. The final *executionPrice* is determined as *executionPrice* = median(midPrices). We also store it as the new *lastGoodMidPrice*.

2.3 Exchange of Coins/Tokens

Once a buyer and seller is matched and a trade is executed, the platform will handle the transfer of coins/tokens between the counterparties. This can be done using XCMP. One potential challenge is that while trade executions have privacy guaranteed due to execution via Phala's TEE structure, coin/token transfers via XCMP are not guaranteed the same privacy and can be potentially reconstructed via block explorers. One potential solution to this is users first wrapping their native coins/tokens into a wrapped token on Phala blockchain, and the wrapped tokens are exchanged, where privacy is guaranteed, and then unwrapped by the counterparties. Other potential solutions are actively being researched by the team.

3.1 Future Roadmap

An interesting potential extension of the mid-price clearing feature of Phalanx is that the traded products do not have to be limited to cryptocurrencies. As long as there are liquid, robust price feeds, the platform can offer products such as stock/commodity futures or even options using the analogous price feeds from the traditional finance exchanges (CME, CBOE, etc) and cash-settling the contracts to stablecoin. This means in future, Phalanx can move forward from just a crypto darkpool exchange to a offering a full-suite trading brokerage services across cryptocurrency and traditional financial markets.