SMOO SWAP:
A Cross-chain Liquidity Aggregator

April 7, 2022
DRAFT Version

Contents

1 Introduction

Decentralized exchanges (DEXs) are playing an essential role in the crypto ecosystem, especially after the evolution of automated market maker (AMM) and decentralized finance (DeFi) Legos.

However, the DEX marketplace remains fragmented, with both trad- ing pairs and liquidity in individual DEXs inferior to centralized exchanges (CEXs). One major weakness of DEXs is that they are inherently limited to a specific blockchain, whereas their off-chain centralized counterparts are automatically "cross-chain" and provide access to a full range of cryptoassets.

Cross-chain bridges are introduced to move liquidity across blockchains, but they are typically used independently from DEXs and users have to switch between several DApps to perform a cross-chain swap.

Therefore, a cross-chain liquidity aggregator is needed to make full use of existing DEXs scattered across multiple blockchains.

Smoo Swap is a cross-chain liquidity aggregator that provides access to various DEXs, cross-chain bridges, and other DeFi applications, so as to expand the available market and increase trading volumes.

1.1 An overview of Smoo Swap

As a cross-chain liquidity aggregator, Smoo Swap consists of both on-chain and off-chain components.

The on-chain part of Smoo Swap is a set of smart contracts deployed

on Mina that keeps record of all transaction history and state transitions happened within Smoo Swap. To achieve best security guarantee Smoo Swap leverages zero-knowledge proof techniques (i.e. zk-SNARK) on Mina, which enables a simple validation of the blockchain state. Such verification used to be quite expensive, which requires either running a light node or trusting some third-party, say the blockchain browser. This design also reduces the cost of storage and avoids unnecessary exposure of user actions.

The off-chain part includes frontend user interface, backend core, and the cross-chain coordinators. In short, most computation is done off-chain for better efficiency and user experience, and simultaneously necessary in- formation with proofs will be committed into and verified by the on-chain contracts.

The cross-chain interoperability is implemented with a committee of notaries. They follow the instructions from the on-chain part and run a threshold signature scheme (TSS) to sign transactions to be sent to DEXs on other chains. They will also monitor the execution results and commit receipts together with proofs to the on-chain part of Smoo Swap (indeed any relayer can submit the receipts as long as the attached proof is valid). The committee is also responsible for transferring funds

between different blockchains, which is inevitable after cross-chain swaps.

More details of the design is given in section 2.

1.2 What does Smoo Swap provide?

Smoo Swap provides the following functionalities:

- Built-in AMM algorithm, for both stable and unstable tokens. An order-book mode will be added in the future.
- Exhibit available exchange rate and volume for requested trading pairs.
- Schedule the optimal trading path based on liquidity available on all possible chains, and estimate real-time exchange fee and gas fee.
- Launch transfer and exchange activities.
- Monitor the status of in-process activities.
- Reclaim assets after execution.

1.3 How is Smoo Swap different from a CEX?

The key difference is that Smoo Swap runs under the constraint defined by Smoo Swap smart contracts. That is, every action and every state transition must be verified by contracts, and all cryptoassets are hosted in contracts. Furthermore the exhibited exchange schedule also comes along with a proof validating its truthfulness.

In summary, SmooSwap is a DeFi application with off-chain computation and on-chain verification. It achieves much stronger transparency, trustworthiness, and security than any CEX solutions.

2 Protocol Architecture

The architecture of Smoo Swap is as in Figure 1.

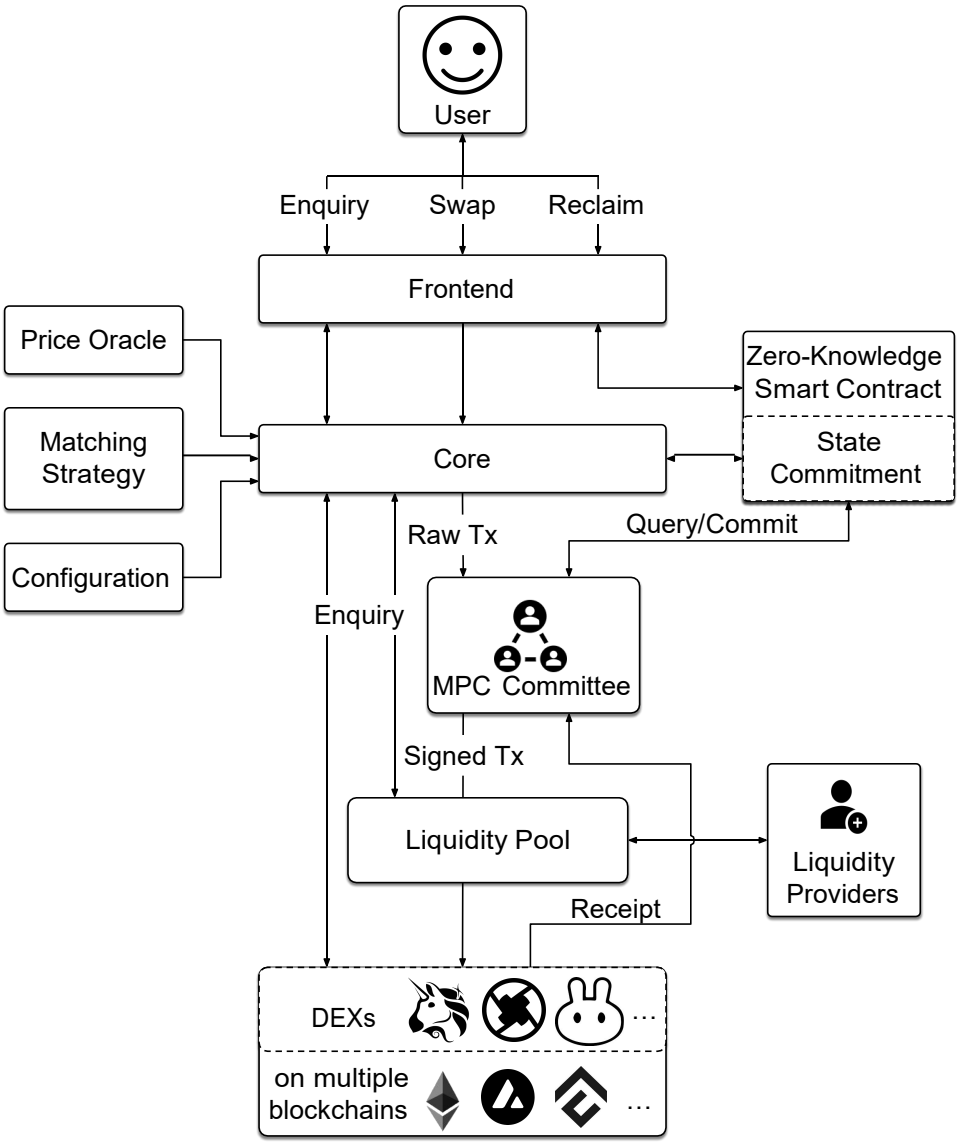

Figure 1: Protocol Architecture.

2.1 Frontend

Frontend is the interface for users to interact with Smoo Swap,mainly through the following three actions:

- **Enquiry**: Before making any exchange, it is necessary that the user first gets the latest exchange rate. *Frontend* passes user enquiries to *Core* for the current best exchange rate, and presents the real-time result to users. More details are given in section 3.1.
- **Swap**: When user decides to make an exchange, *Frontend* composes the raw transaction containing exchange instructions (which the user is supposed to sign on), and sends authenticated instructions to *Core*.
- **Reclaim**: After executing exchange transactions on other chains, the final result (including success/failure, actual rate/volume/fee, *etc.*) is available to the user. The result can be verified through the state commit from the *Zero-Knowledge Smart Contract*.

2.2 Core

Core is the center of Smoo Swap, with three major functionalities:

- Calculate the best trading path on user enquiries. Exchange fees, rates and fluctuations, as well as user-specified restrictions such as slippage tolerance and timeliness, would be taken into account. The calculation may also depend on external data from *Price Oracle*, *Matching Strategy* and *Configuration*. For example, when assessing the cost of using some DEX deployed on Ethereum, the real-time price oracle of ETH would be necessary.
- For every *Swap* action, submit user authenticated transaction to Mina, parse user instructions into raw transactions according to the underlying trading path, and finally send those raw transactions to the *MPC Committee*.
- Keep in sync with the latest state of *Zero-Knowledge Smart Contract* on Mina as well as the *Liquidity Pool* deployed on multiple other blockchains, and generate validity proofs for user verification.

2.3 MPC Committee

The committee consists of multiple (how many?) reputable and trustworthy entities. They run a MPC protocol to verify Smoo Swap user instructions and sign exchange transactions accordingly. The private keys of *Liquidity Pool* are kept using a secure TSS, where every member in the committee only holds a share and hence it requires a consensus of the majority of committee members to move funds hosted in *Liquidity Pool* contracts.

More specifically, Smoo Swap applies Distributed Control Rights Man- agement

(DCRM) technology developed by Fusion[1], which implements an efficient TSS based on ECDSA.

2.4 Liquidity Pool

Liquidity Pool contains cryptoassets on multiple blockchains, and hence empower Smoo Swap with the ability to aggregate DeFi applications from the whole crypto-ecosystem. Liquidity providers may add liquidity to the pool, and get reward for their contribution.

2.5 Zero-Knowledge Smart Contract

This is the contract deployed on Mina to maintain the state of Smoo Swap. Inside the contract there is a state commitment, and every update to that state commitment must come along with a valid zero-knowledge proof for the validity of such state transition.

More specifically, *Zero-Knowledge Smart Contract* contains the verification method of a ZK circuit, which checks the following conditions on input of state commitments PRE_STATE, POST_STATE together with the corresponding ZK proof PROOF:

- **valid state**: the before-transition-state PRE_STATE is consistent with the latest committed state CURRENT_STATE in the contract;
- **valid transition**: there is a sequence of transactions $T[1], \ldots, T[n]$ such that

  $$\delta(\text{PRE\_STATE}, T[1], \ldots, T[n]) = \text{POST\_STATE}$$

  Where $\delta(\cdot)$ denotes the transition function,PRE_STATE and POST_STATE are the state commitments before and after the transition respectively;
- **valid transactions**: every transaction $T[i]$ in the above transition follows the predefined format and carries valid value, fee, *etc.*;
- **valid witness**: every transaction $T[i] = (t[i], \pi[i])$ has witness fields (such as nonce and signature) in $\pi[i]$, where $t[i]$ is a truncation of $T[i]$ without those witness fields.

For data availability, transactions should also be submitted to *Zero-Knowledge Smart Contract*. Here we apply the witness segregation technique to make the update and synchronization more efficient. That is, only the succinct transactions $t[1], \ldots, t[n]$ (without witness $\pi[i]$'s) are submitted. The validation of $\pi[i]$'s are wrapped in the ZK proof and hence omissible thereafter.

After the state update request passing all above validation, *Zero- Knowledge Smart Contract* would update its state commitment accordingly, by setting CURRENT_STATE := POST_STATE.

As a result, the state commitment stored in *Zero-Knowledge Smart Con- tract* is guaranteed correct. And detailed states can be verified by everyone given a valid ZK proof with respect to the latest state commitment.

2.6 Price Oracle

Price Oracle connects to multiple DEXs and CEXs to provide the real-time price of cryptoassets. This component is especially important to sup- port *Core* in finding the best trading path across multiple independent blockchains.

2.7 Matching Strategy

This component holds the algorithm description of the searching algorithm that finds best deals and trading paths for exchange enquiries. Everyone is thus capable to verify the correctness of result exhibited in *Frontend*, i.e. *Core* returns the exact result by truthfully applying the matching strategy on the latest state.

2.8 Configuration

This component provides configuration and governance functions, including:

- charging configuration;
- liquidity pool configuration;
- MPC committee configuration;
- enable/disable the usage of specific blockchains and on-chain DeFi applications.

3 One Entrance to Access Cross-chain Liquidity

As a cross-chain liquidity aggregator, Smoo Swap aggregates various DeFi applications deployed on multiple blockchains, and hence becomes an easy entry point to the whole crypto-ecosystem. Users typically interacts with Smoo Swap with three actions: Enquiry, Swap, Reclaim.

3.1 Enquiry

The Enquiry action allows users to check the exchange rate and related information such as slippage, optimal trading path, estimated latency and so on. The procedure of Enquiry is as follows:

1. User: submit the trading pair and volume that he wants to query.
2. *Frontend*: send the enquiry to *Core*.
3. *Core*: parse the enquiry and then request external information from DeFi applications, *Liquidity Pool*, and *Price Oracle*. More specifically, DeFi applications should return the current exchange rates of potential trading pairs, *Liquidity Pool* returns possible available volume of cryptoassets on each blockchain, *Price Oracle* returns the real-time price of tokens needed for exchange fees and gas consumption.
4. *Core*: run the search algorithm as specified in *Matching Strategy* to find the optimal trading schedule, and add Smoo Swap service fee following *Configuration*. The result (including exchange rate/slippage/trading path/estimated latency/...) will be sent to *Frontend*.
5. *Frontend*: present the result to user.

3.2 Swap

The Swap action allows users to make an exchange. Its procedure is as follows:

1. User: submit the exchange request, with trading pair, volume, slippage tolerance, *etc*.
2. *Frontend*: compose the raw transaction that locks the cryptoasset that user wants to trade out into a smart contract on Mina. This transac- tion also contains necessary information about the exchange request.
3. User: double-check the raw transaction, and sign on it if no objection.
4. *Frontend*: send the signed transaction T to Mina, and inform *Core* about the exchange action.

5. *Core*: after the transaction T is confirmed, compose raw transactions according to the calculated trading schedule, and send them to the *MPC Committee*.
6. *MPC Committee*: verify the committed transaction on Mina and the raw transactions from *Core*, then run the threshold signing protocol to sign on those raw transactions.
7. *MPC Committee & Liquidity Pool*: submit signed transactions to corresponding blockchains through *Liquidity Pool*.
8. *MPC Committee*: fetch execution receipts of submitted transactions and commit the execution results to the *Zero-Knowledge Smart Con- tract* on Mina.
9. *Zero-Knowledge Smart Contract*: upon receiving the execution results, calculate the actual exchange rate and dispose of the locked assets accordingly. For example, user would get the cryptoasset he wants to buy in if the underlying exchanges are all successful, and he would get back his origin cryptoasset (after deducting necessary fees) if the exchange fails.

3.3 Reclaim

The Reclaim action allows users to check the execution results of Swap actions as committed to the *Zero-Knowledge Smart Contract* on Mina and reclaim the assets that he deserves.

This action does not rely on the truthfulness of *Core* (indeed it can be performed independently without interacting with *Core*). The validity of committed states is proved using zk-SNARK technique, and the proof can be verified through any trustworthy zk-SNARK verifier. Therefore, the user no longer needs to trust any centralized party, neither a blockchain browser nor the *Frontend* of Smoo Swap.

4 Mathematical Model Inside

How to compare the prices from different chains? If necessary (or remove this part if not ready).

TBA.

5 Token Economics

TBA.

6 Roadmap

TBA.

References

[1] Fusion: Distributed Control Rights Management.
https://www.fusion.org/tech/dcrm.