DEx Technical White Paper (Short Version)

Introduction

DEx is a smart contract based, decentralized crypto-token exchange which emphasizes two features

A. **Decoupling assets of the users from the exchange.**
The users' assets are never in custody of DEx and therefore can not be lost even if the exchange is hacked.

B. **Instant trading experience.**
Placing, canceling, matching and executing orders are instant.[1]

A. is the core strength of traditional decentralized exchanges, while B. is the core strength of centralized exchanges. DEx achieves **both** with its novel way of designing and driving smart contracts. We name it ***ROC (Replayed on Chain)***.

Technical Background: Two Ledgers

DEx has two trading ledgers: an off-chain ledger responsible for providing instant trading experience and a smart-contract based on-chain ledger to ensure the traders' assets are safe. The two ledgers are closely synchronized, where all trading activities are first registered on the off-chain ledger and later get exactly *replayed* (executed in the same way and the same order) and confirmed on the on-chain ledger. The on-chain ledger can be interpreted as a delayed version of the off-chain ledger, which will eventually catch up. In detail, the two ledgers work as follows:

The off-chain ledger is a virtual trading floor with instant trade settlement, hosted on DEx servers:

- **Off-chain trading:** Each trader has a off-chain trading balance reflecting their on-chain trading balance. Traders can place buy and sell orders, and when orders are matched, the off-chain trading balances are updated immediately. However, **the balances on the off-chain ledger are simply phantom balances and there is no actual transfer of tokens taking place off-chain.** All the activity registered on the off-chain ledger **must be confirmed on the on-chain ledger** where the actual transfer of tokens takes place.
- **Private key signature:** To verify the identity of the traders and to ensure that the off-chain trading activity reflects the traders' will, each trader has to sign all off-chain trading activity with the private key of the trader's on-chain account.
- **Off-chain / on-chain synchronization:** To avoid a mismatch between the off-chain and on-chain trading balances, and to ensure that traders cannot trade off-chain with funds that they have withdrawn from the on-chain trading balance, the on-chain smart contract holds the funds of each trader to match the trader's off-chain trading allowances.

The on-chain smart contract is the key element for the decentralized custody and settlement of assets. The smart contract safeguards the traders' funds on-chain and has a very limited set of functions:

[1] Exceptions are transferring initial tokens to the smart contract and releasing tokens from the smart contract, which require blockchain confirmation.

- **Receipt of tokens / custody:** A trader is required to transfer the tokens the trader wishes to trade to the smart contract, which will be added into an account identified by the trader's public key in the smart contract. Only who has the trader's private key can trade (by signing orders with the private key) with the balances of this account. Next, the updated trading balance of this on-chain account will be reflected in the off-chain ledger and the trader can start trading. The trading allowance held in the smart contract is to ensure that the trader holds a sufficient on-chain trading balance for the off-chain trading activity.
- **Update of trading balance / settlement:** When the off-chain ledger communicates a pair of matching orders to the smart contract, and both orders are correctly signed with the private keys of the traders, the trading balances for both traders will be updated in the smart contract. To ensure trading is done in compliance with the DEx rules (e.g., approved KYC checks), the “PlaceOrder” function of the the smart contract requires signature of the administrator private key, which is held by the DEx team.
- **Release of funds:** Upon request of the trader, and if there are no orders outstanding on the off-chain ledger, the tokens held on the smart contract will be released to the token account of the trader[2]. For verification that there are indeed no outstanding orders, the smart contract function for releasing the tokens again requires the signature of the DEx administrator. The smart contract has a key safety feature: **It is technically impossible for the smart contract to release the assets to any account other than the trader’s account, it is therefore impossible for hackers or creditors of DEx to gain control over the traders’ assets. As an additional safety feature to prevent users' assets from being locked in the smart contract, the smart contract has a “release all” function (callable by the DEx administrator) which stops trading indefinitely and releases all assets of all traders to the traders’ accounts. To resume trading a new smart contract needs to be deployed[3].**

As the smart contract is the key to the trustless trading model, it is:

- **Immutable:** Once deployed, the smart contract is fully on-chain and cannot be changed, neither by DEx nor by anyone else.
- **Open-source:** The traders do not have to trust DEx as DEx has no custody of or access to their assets. The traders simply have to verify that the smart contract is coded correctly. Therefore, the smart contract will be fully open-source and viewable to the public in the beginning.
- **Simple:** Verifying the correctness of the code must be easy. Therefore, the smart contract is kept as simple as possible (only appx. 500 lines of code).

=> The traders’ tokens are never held by DEx, only by the smart contract.

=> The tokens can only be traded directly from trader to trader. An order must be correctly signed by the private key of the trader, which will be verified by the smart contract when executing the order. DEx will not collect or store any private key of the traders.

=> Hacking of DEx’ off-chain trading platform can only interrupt trading, but cannot transfer any trader's assets to an account controlled by the hacker, nor placing an order for any trader. In the

[2] An account identified by the trader's public key, which requires the trader's private key to operate.
[3] The expiry of the old smart contract and the release of a new smart contract is equivalent to creating a completely independent one. A trader has to transfer the released funds again to the new one in order to trade in the new smart contract..

extreme case where the interruption cannot be recovered, DEx can close the on-chain ledger and release all traders' assets to their accounts.

=> The only risk the traders face is leakage / loss of their own private key: DEx does not have a copy of the private key and there is no “I forgot my password” functionality. Also, it is technically impossible to release the assets of a trader to an account controlled by another private key. If the private key is lost, the assets are lost. This is a general property of crypto-currencies, not unique to DEx.

Step-by-Step Trading

To illustrate the functioning of DEx, please find below a step-by-step explanation of how Alice, an avid cryptocurrency trader with a few ETH, can exchange her ETH for the ERC20 token “Token A”. Numbers in brackets such as "(2)" signal a single trading activity that is executed both on-chain and off-chain - matching numbers indicate that it reflects the same trading activity.

Time Points	Off-chain	On-chain
1	Alice registers on DEx and, after passing the KYC requirements, she can add a personal public key as her trader public key in DEX. She must store the corresponding private key securely in order to sign future trading actions and control her assets released from DEx once she is done trading on DEx.	
2		Alice transfers 10 ETH to her trader account in DEx (identified by her trader public key).
3	(1) After Alice's transfer has been well confirmed on the blockchain, the DEx server will add 10 ETH to Alice's off-chain trading allowance.	
4		(1) 10 ETH is added to Alice’s on-chain trading balance.
5	(2) Alice places an order to buy 80 Token A at price 0.1 ETH/Token. The order must be signed by the private key of Alice and will be verified by the DEx server.	
6	(3) DEx matches and executes Alice's order with an order placed earlier by Bob, selling 60 Token A at price 0.1 ETH/Token. Alice's balance is updated to (60 Token A, 4 ETH) and her order's amount is reduced by 60 to 20 Token A. Bob’s balance and orders are updated as well.	

7	(4) Alice cancels the remaining buy order for 20 Token A, and places an order to sell 40 Token A bought just now, at price 0.2 ETH/Token. No matching order in the orderbook for now.	
8		(2) Alice places an order to buy 80 Token A at price 0.1 ETH/Token. The smart contract will verify the signature to ensure that this order is authorized by Alice.
9		(3) The smart contract matches Alice's order with Bobt's order selling 60 Token A at price 0.1 ETH/Token. Update of the trading balances and orders in the same way as the off-chain ledger. The smart contract will check that the prices of the two orders are matching and both sides have enough balances for the trade.
10	(5) Pat places an order that can fully match Alice's selling order.	
11	(6) Alice's and Pat's orders are matched and executed. Alice's order is fully executed. Now Alice's balance is (20 Token A, 12 ETH).	
12	(7) Alice withdraws 5 ETH. Alice's balance is updated to (20 Token A, 7 ETH).	
13		(4) Alice places an order to sell 40 Token A bought just now, at price 0.2 ETH/Token. No matching order in the orderbook for now.
14		(5) Pat places an order that can match Alice's selling order. The smart contract will verify the signatures.
15		(6) Alice's and Pat's orders are matched and executed. Alice's order is fully executed. Alice's balance is (20 Token A, 12 ETH).
16		(7) Alice withdraws 5 ETH. Alice's balance is updated to (20 Token A, 7 ETH). The smart contract will check that Alice has enough balance and make a transfer of 5 ETH to the account identified by Alice's trader public key.