Efficient Onchain CEX Derivatives with UTxO Bilateral Derivative Contracts

Max Brillaugte
brillaugte@proton.me

21st July 2023

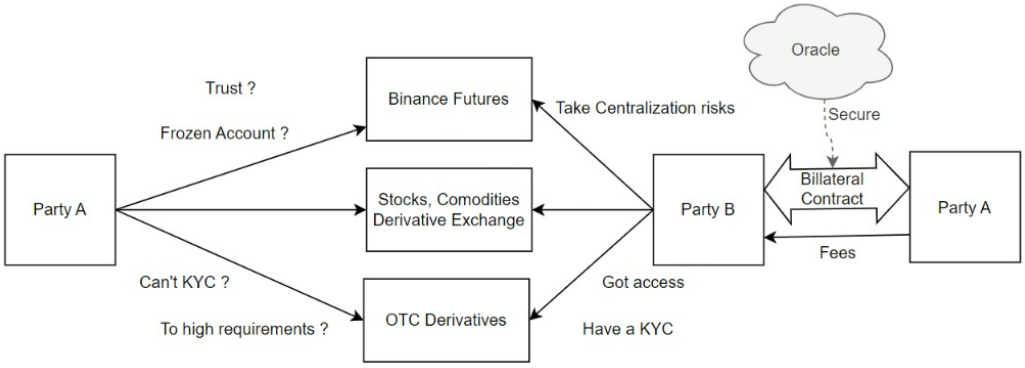
Abstract

This paper introduces how, with the help of UTXO smart contracts and a price oracle, peer-to-peer non-custodial users can trade exchange smart contracts safely, with the same collateral requirements and user experience as centralized exchanges.

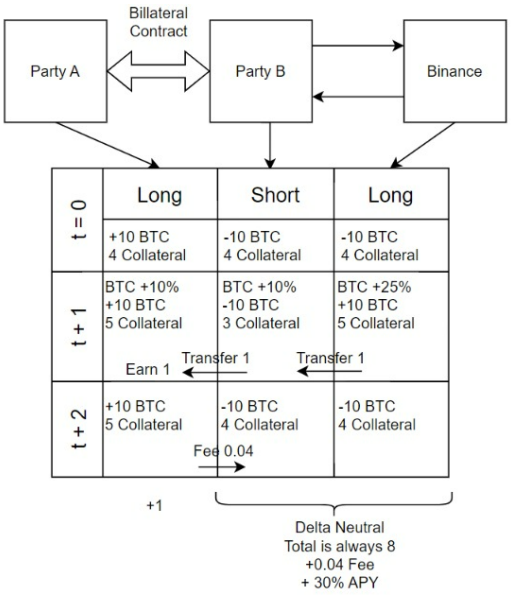
1 Introduction

The 2022 FTX default and the multiple exchange asset freezes reminded the cryptocurrency community that self-custody was a necessity. From a wider perspective, billions of people cannot access certain products due to high requirements, KYC processes, or, as in the case of Binance Futures, outright bans on leverage trading by many countries. We present a solution where user, which we will qualify as party A, wants to buy a derivative on a centralized exchange or intermediary. With the help of an on-chain bilateral contract, a party B, who has access to or trusts these trading exchanges, takes the desired position of party A. Money is sent to party A through the Bilateral contract if they win, and taken if they lose.

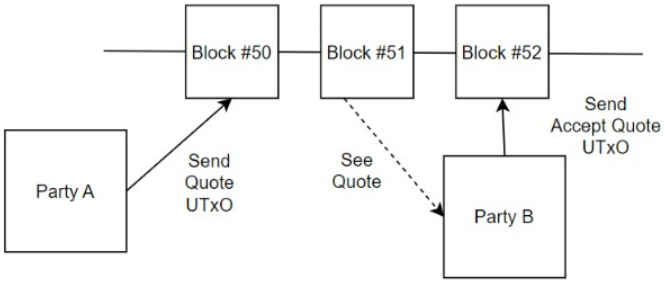
Party B commits to paying the profit and loss PnL on a product but is free to choose where they replicate and hedge the trade.



In Figure 2, we show an example of a party A and party B opening a bilateral contract of 10 BTC long with 4 BTC of collateral. A is long, B is short, and B opens a long on Binance with 4 BTC to replicate the profit and loss of A. At t=1, the BTC position moved by 10 percents, meaning that A owns 5 BTC and B owns 3, but the Binance hedge position also earned 1. Party B then rebalances collateral between the bilateral contract and Binance to have equally 4 BTC in each of them. At t=2, party A closes the contract and pays interest to B of 30 percents APY in this example. APY can be set much higher or much lower depending on party B. When it's difficult to find a delta-neutral strategy with low risk, providing liquidity to party A's can be very profitable as it only locks needed collateral and is almost risk-free. Additionally, being a retail counterparty with high leverage can allow for farming their default fee, allowing a significant increase of APY.



The protocol architecture revolves around party A and party B agreeing off-chain on a front end with reputation or on a protocol like 0x. Then one party sends a UTXO quote on-chain, which is answered or not by party B. Party B is responsible for only accepting quotes that they expect to be profitable. This architecture allows, in case of an oracle hack, for only a few positions to be affected as everything is isolated.



In the literature, solutions to on-chain derivatives are either overcollateralized, limited to extremely liquid assets, or rely on the goodwill of a third party. This paper's architecture is inspired by traditional finance over-the-counter derivatives that are highly customizable and designed to operate in extremely illiquid markets with high latency.

For derivatives, we can consider time efficiency and collateral requirement efficiency. To achieve the latter, contracts are not transferable, so each coun-

terparty only has to pay the difference between the entry and current price of the product. With frequent collateral updates and settlement, the amount needed to be held in the contract represents only a few worst hours for highly liquid assets and five worst days for the most illiquid products.

2 Quote

Party A issues a quote on one side of the contract, whether short or long, and party B will take the opposite side. To help party B lower their spread due to the costs of hedging the asset price feed (as BTCUSDT from two different exchanges can have different prices), an "expireAt" timestamp is provided. This is the last moment when party B can respond to this quote. To prevent party B from taking advantage of price delays and to avoid depending on an oracle, both parties exchange off-chain prices and push the correct price on-chain. If the price is correct, any whitelisted members of the Collateral Agreement (CA) can fill the quote.

In order to lock a significant amount of collateral to the position to cover hedging losses, expected losses based on historical simulated scenarios are calculated. These are referred to as the initial margin of Party A (IM_A) and Party B (IM_B). This system only asks for the minimum required collateral. When the quote initiator sends for A and B, it is the responsibility of party B to inform party A of his requirements, for example, through a frontend interface.

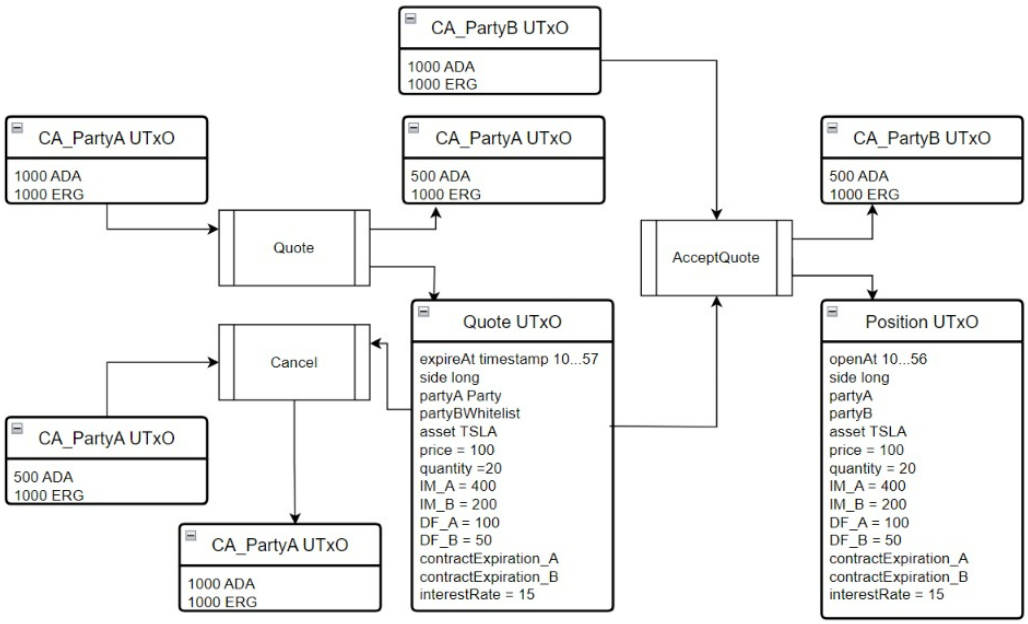
Since the position is under-collateralized and early termination is costly, there are the default funds of party A (DF_A) and party B (DF_B). These fees go to the non-defaulting party, incentivizing them to stay in the trade or to pay the counterparty to take on another trade.

To prevent the contract from being open indefinitely, contract expirations can be set by both parties. This is a specified time when they can optionally or be forced to close the trade. And finally, to cover the cost of capital immobilization for party B who has to hedge the trade, as well as to pay for centralized risk and cost, an annual interest rate is defined and paid at the position's closure.

To avoid exploits, when a quote is issued, the issuer locks IM_A and DF_A. The quote can be canceled if not filled before the expiration, and the collateral is then returned to the CA.

3 acceptQuote

Once the quote is cast on-chain, party B answers it with an acceptQuote script and locks DF_B and IM_B. The timestamp of the position's opening is added for interest rate computation.



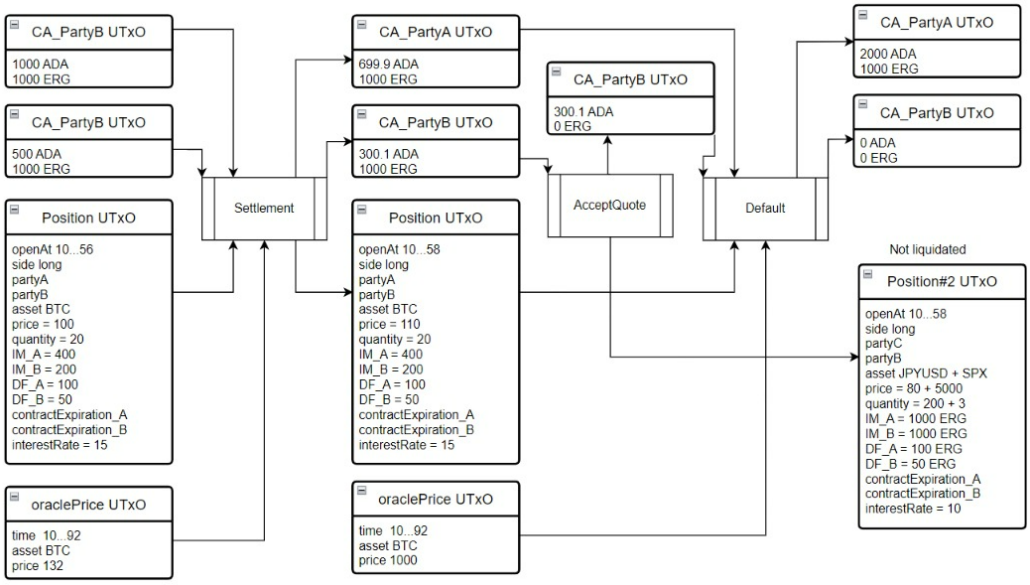
4 Settlement

When a contract is open, the IM is defined for a few days. As soon as the price evolves in a direction, one of the parties must update the IM and return extra collateral to the CA. The settlement can be made without oracles. When a settlement is made without an oracle, no default or withdrawal can be made until a predefined time that allows a liquidator to penalize the settler if the price error is too large compared to the oracle's price. In the case where it doesn't trigger a default event, if one of the parties disagrees, they can call for a new oracle price.

5 Default

When the IM is not enough to cover unrealized PnL ($uPnL$), a default event can be triggered by one of the parties or a liquidator for a fee. A

default event requires a price from an oracle, a position, CA_A, and CA_B. If there is enough collateral in the defaulter's CA to cover the $uPnL$, the trade continues; otherwise, the position is closed and the non-defaulting party receives the remaining collateral of the defaulting party, along with the IM and DF. The IM architecture allows non-whitelisted parties to trade with the same CA, but each party needs to settle on a regular basis to avoid default losses due to the on-chain position closing at a more expensive price than the position hedge.



6 Deposit

The deposit process is straightforward as no attacks can be executed by adding money to the CA.

7 Withdraw

If a party initiates a settlement, a withdrawal can be made to avoid the settlement. If the position's IM has been properly defined and the settlement is made on time, the position defaults and the non-defaulting party doesn't suffer a loss. If it's too expensive to settle regularly, a time-based withdrawal

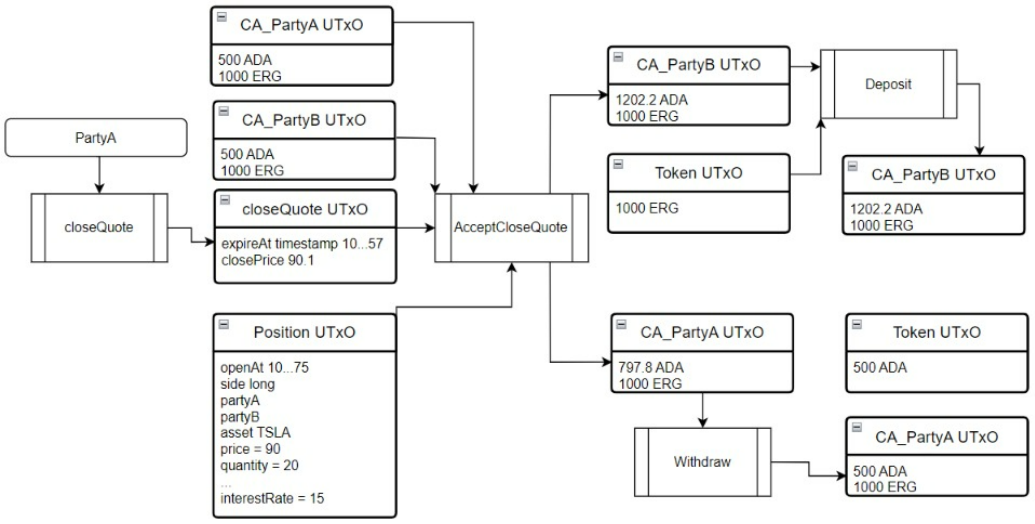
system needs to be employed to give a liquidator time to react and earn a fee.

8 closeQuote

A closeQuote event can occur in two ways: either one party casts a closePrice and their counterparty accepts the closeQuote before the expiration time, or the close is made with the assistance of an oracle, as specified in "marketClose".

9 acceptCloseQuote

When party A or B sends a closeQuote, once party B sees it, they can accept it by sending CA_A, CA_B, closeQuote, and the position to close out the position. If the closePrice is unfavorable, the party can choose not to respond to the quote. For better user experience, the closeQuote price should be defined off-chain with a request for quote mechanism between the two parties. Only the position initiator can close the position; the hedging party must account for potential spread cost when closing the party.



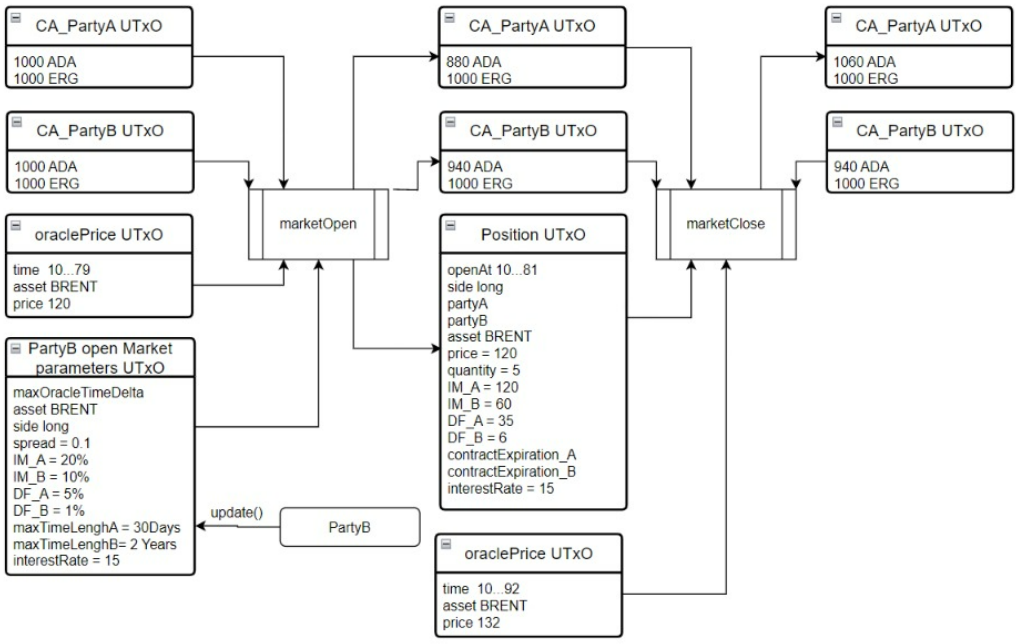
10 marketOpen

We have now looked at the entire process without the use of an oracle. But what if party A wants to open a position instantly without being able to exploit party B? Party B defines parameters in a $UTxO$ (unspent transaction output) that they publish and can update. Each oracle price data point is published into a $UTxO$, so a maxOracleTimeDelta is defined to prevent price exploits by calling oracles and waiting until the price moves enough to take advantage of the positions. The time to open the transaction is limited. In volatile environments, the CA of B can be set to "close only" to prevent any exploit. Each parameter is defined for each asset ID, as described in "quote". Parameters include IM_A, IM_B, DF_A, DF_B, and percentages to adapt based on position size. Other parameters include maxTimeLength$A$, maxTimeLength$B$, and interest rates.

To prevent someone from taking too large a position, an extra parameter can be added to the $UTxO$ for the maximum number of position updates before the $UTxO$ needs updating. It is intended to be reset by the counterparty between each position.

11 marketClose

The trade is secure as long as the closing price of the positions is secured. We've previously seen a method to close a position by agreeing with party B, but a method is also needed to force party B to close at market price. To do this, a $UTxO$ with the oracle price, the position $UTxO$, CA_A, and CA_B $UTxOs$ are required to close the positions. Once again, a limit is defined regarding the freshness of the oraclePrice.



12 Limitations

In the case where gas fees or oracles are expensive, we consider PartyA as a centralized exchange and PartyB as a liquidity provider with a reputation. PartyA deploys a collateral agreement with a list of whitelisted PartyB. Only the selection of the counterparty is centralized; the counterparty cannot exploit PartyA for money, or for a spread in the worst case. Additionally, in the aim of reducing costs, if the UTXO space is sufficient, multiple positions between the same PartyA and PartyB can be set in the same UTXO.

The project is limited to a sort of on-chain CFDs, as transferring to an unknown PartyB can lead to less trust in the capacity of the new party defaulting, as well as having multiple layers of PartyA and B that would lead to an implosion in case of one party in any of them defaulting. New types of dApps would benefit from these advantages of bilateral parties, like inflation-based stablecoins, but counterparty risk needs to be solved first without decentralization. [1] Describes how starting from bilateral positions with the help of a central counterparty, counterparty risk can be managed.

13 Conclusion

We defined a straightforward system to allow non-KYC users to trade on-chain products available to centralized KYC users with the help of a counterparty, off-chain front ends, and trusted bilateral positions. We exposed the centralized points and observed that they have low impacts on the security of the trade. To fully embrace decentralization, bilateral arrangements are not sufficient due to zero-knowledge counterparty risk and require an automated central counterparty.

References

[1] Peer-to-Peer Derivatives Trading vie an Automated Central Counterparty Financial Market Infrastructure, 2023. `https://microderiv.com/whitepaper`