V
D
E
X

Zero Knowledge Exchange

José Betancourt and the VDEX Team
jose@vdex.trade and team@vdex.trade

June 4, 2025

Abstract

Virtual Labs puts forward VDEX, a modular perpetual exchange capable of matching the security of Ethereum with the performance of Binance. This is achieved through our proprietary technological architecture of the last three years, the VIRTUAL ROLLUP. The VIRTUAL ROLLUP is a memory network for streaming Zero Knowledge through local consensus. This is the first scaling solution to achieve higher throughput by splitting STATE and ESCROW. This is able to achieve a modular version of a perpetual exchange. This makes VDEX uniquely capable of (1) being deployed across multiple settlement layers while maintaining a UNIFIED LIQUIDITY LAYER, (2) executing instantly without regard to the blocktime finality latencies of the underlying SETTLEMENT LAYERS, (3) and achieving full self-custodial BFT (Byzantine Fault-Tolerance) with lesser security assumptions than existing blockchain state transfer mechanisms.

1 Introduction

This paper will answer how a trading protocol can achieve trustless, self-custody with the performance and cost of centralized systems, an achievement that has evaded traditional blockchain architectures. It primarily explores the VIRTUAL ROLLUP architecture.

2 Performance of a CEX

2.1 Sub-Millisecond Finality

VDEX overcomes the limitations of blockchain architecture through post-blockchain architecture capable of independently maintaining STATE. The first of these limits is due to the block's innate quality of periodicity–which creates an average minimum latency. VDEX overcomes this latency because VIRTUAL ROLLUPS settle STATE between the users' VIRTUAL NODES. If these VIRTUAL NODES are on the same machine, they will settle in under a millisecond.

If the users are not in the same location, then they will settle in a time equal to the finality on the machine, M, and the time derived from their distance, D, where V is the VIRTUAL ROLLUP settlement time: V = M + D. Exchanges use the terminology “co-geoprically located clients” to refer to this latency measurement. It additionally displays that VIRTUAL ROLLUP maintains the theoretical optimal latency and is comparable to CEXes. The impact of this innovation is reflected best in the self-reported latency scores from decentralized perpDEXs.

Name Latency
VDEX <1ms + D
GMX (Arbitrum + Avalanche) 250ms + D/2,005ms
Hyperliquid <1,000ms + D
dydx <1,000ms + D

Sources: HackMD, Hyperliquid, DyDx

No Slippage Trading

The advantages of ultra-fast trading are obvious. Two that are worth drawing attention are (1) the ability to frontrun other DEXes and (2) the elimination of slippage. VDEX is able to nearly remove all slippage because users will likely see the same price on their “order screen” and “entry price” due to the instant communication between user frontend and VDEX.

2.2 ZeroGas Fees

The second fundamental blockchain limitation, alongside the periodic nature of the block, is the fact that blockspace is finite. This restraint is created by GLOBAL STATE. In VIRTUAL ROLLUP, only LOCAL STATE is maintained. This is because only the relevant users to the transactions must store STATE. Users do not need to store the STATE of other users. Therefore, “blockspace” or “TPS” could both be considered theoretically infinite in VIRTUAL ROLLUP.

In the VIRTUAL ROLLUP, blockspace is VIRTUAL and infinite. Blockspace no longer exists as a constraint, so VDEX does not need to charge for gas tolls.

2.3 MEVZero

VDEX is as performant as CEXes in a final major way: the virtual destruction of MEV, impermanent loss, and sandwich attacks. This is all due to the decision to integrate an orderbook instead of an AMM. This further allows limit orders, open interest calculations, and a more traditional experience to traditional finance and CEXes.

3 Decentralization of a DEX

3.1 What is VIRTUAL ROLLUP?

VIRTUAL ROLLUP is a technical architecture composed of a ZK STATE CHANNEL and SETTLEMENT

LAYER. VIRTUAL ROLLUP makes this distinction in order to separate STATE and ESCROW. Traditionally, STATE and ESCROW are stored and moved together. However, as seen in *Section 2 Performance of a CEX*, this separation results in tremendous benefits. As shall be explained here, the split requires **no additional trust assumptions**.

The VIRTUAL ROLLUP series is designed to Stream Zero Knowledge through Local Consensus. VDEX runs VIRTUAL ROLLUP 1.3 (VR1.3) which is custom-fitted to running perpetual order books. VR1.3 offers the following properties to achieve a decentralized protocol: Full Self-Custody, Trustless State Transitions, and Censorship-Proof ESCROW Disputes,

3.2 Full Self-Custody

ZK STATE CHANNELS are Byzantine Fault-Tolerant ephemeral DAGs which are self-verified. So, it can be said that VDEX offers full self-custody by achieving full self-verification. The basic premise is that if funds can only be moved by the unanimous consent of the owners of these funds, malicious attacks are impossible. This is the idea of LOCAL CONSENSUS: agreement reached by the relevant participants in their own transactions.

VR1.3 achieves this in practice by storing ESCROW in smart contracts on SETTLEMENT LAYERS (1) not controlled by VIRTUAL LABS, (2) transparently viewable and publicly audited, and (3) protected by VAULTS of ISOLATED USERS which are independent pools dedicated to a single user's funds.

3.3 Trustless STATE Transitions

In VR1.3, STATES are considered absolute. This means that the newest STATE always overrides previous STATES. New STATES require unanimous consent among users within the ZK STATE CHANNEL. These two conditions mean that the VIRTUAL NODES will form independent ledgers, where the latest root represents the user's latest balance and positions.

There are three types of state transitions within VIRTUAL ROLLUP:

1) SETTLEMENT LAYER to ZK STATE CHANNEL
2) Within ZK STATE CHANNEL
3) ZK STATE CHANNEL to SETTLEMENT LAYER

1) In order for a user to enter the VIRTUAL ROLLUP, the must DepositAndStartZKSC() by deploying ESCROW in the SETTLEMENT LAYER smart contract, which spins up a ZK STATE CHANNEL by returning the PAYLOAD and SCHNORR SIGNATURE as the "genesis block."
2) The user creates a new STATE by signing a new PAYLOAD with the private key associated with the public key responsible for depositing ESCROW. VDEX will sign this and send it back. A transaction is only finalized if both participants acknowledge the fully signed latest STATE.\

3) If both parties agree on the STATE and are online to allow WithdrawSchnorr(), then they will modify isFinalized to be TRUE so that the user can withdraw immediately. If the participants cannot achieve unanimous agreement or are not all present to attest, then any VIRTUAL NODE can submit the latest STATE to the SETTLEMENT LAYER.

3.4 Censorship-Resistant ESCROW Disputes

The final piece of the puzzle in a decentralized LOCAL CONSENSUS system is GLOBAL ESCROW. In *Section 3.2 Trustless STATE Transitions*, it was outlined how users' VIRTUAL NODES store and create STATE. However

this STATE must have a direct and easy method to claim ESCROW. In VIRTUAL ROLLUP, this is WithdrawAndClosePositionTrustlessly().

This function is called if users submit a STATE where the PAYLOAD does not contain isFinalized==TRUE. This signals that there is disagreement and will not process the ESCROW immediately, but instead begins a 24-hour window which will accept STATE from all users. If the first STATE is the most recent, it will award ESCROW to the user. However if a later STATE is submitted by another user, then this signals that the first disputer submitted a STALE STATE, attempting to overclaim ESCROW. The contract then awards the ESCROW proportional to the more recent STATE and can slash the attacking user.

Disputes can be made via an open-source frontend published at IPFS.

4 Chain Abstraction

4.1 Introduction

The term SETTLEMENT LAYER has been used to refer to the location of ESCROW. This SETTLEMENT LAYER can not only be any blockchain (or any programmable financial system), but it can also be multiple SETTLEMENT LAYERS together. This means that users can deposit on one blockchain and withdraw as another, treating VDEX as a bridging protocol we might call VIRTUAL BRIDGE. Perhaps more importantly, VDEX can facilitate crosschain perpetual trading with users depositing from different chains. This creates a UNIFIED LIQUIDITY LAYER. How is it possible?

4.2 ChainID

In VIRTUAL ROLLUP, all STATE points at corresponding ESCROW, whereby ESCROW==STATE must be true. To keep track of ESCROW across multiple SETTLEMENT LAYERS, VDEX assigns each a ChainID. The smart contracts hosted on each blockchain are then identical with the exception of their ChainID, which is supplied by RPC data.

Because STATE is absolute, the presence of ESCROW attached to one SETTLEMENT LAYER necessarily implies the absences of it on all other blockchains. For instance, imagine Alice deposits 1,000 USDC on Ethereum and changes her default chain to Bitlayer.

	User Balance ChainID
STATE 1	1,000 USDC 0
STATE 2	1,000 USDC 1

If Alice's VIRTUAL NODE did not receive STATE 2, then Alice could submit STATE 1 to Ethereum and win that dispute. If VDEX submitted STATE 2 to Ethereum, that would win the dispute there, but in doing so, VDEX would reveal STATE 2 to Alice, who could submit this to Bitlayer and win the dispute there.

The result of ChainID is a fully trustless, but not permissionless omnichain protocol–it might even be considered

more decentralized than previous attempts which trust the Oracle-Relayer. In VIRTUAL ROLLUP, the user might be thought of as the relayer which eliminates a previous trust assumption.

4.3 VIRTUAL BRIDGE

VDEX could function as a shop used for onramping and offramping liquidity from supported chains, similar to CEXes. While a core feature of VDEX, it might be thought of as its own protocol. VIRTUAL BRIDGE could tap VDEX liquidity across SETTLEMENT LAYERS in order to provide an additional service to users. It could use proceeds to further reward supplied liquidity.

4.4 UNIFIED LIQUIDITY LAYER

The purpose of VDEX is to create a bridgeless future. This is made possible by allowing users from various SETTLEMENT LAYERS to deposit into a unified experience and liquidity layer known as VIRTUAL ROLLUP. Once STATE is created in thin manor, users can trade against each other without fees or finality time. Unlike pre-existing perpetual exchanges where ESCROW is held on only one dedicated SETTLEMENT LAYER, VDEX could feasibly support N chains.

This is as a result of VDEX's modular architecture and represents a large leap forward in capital-efficient markets.

5 Volatile Asset Collateral (VAC)

5.1 Introduction

Bitcoin represents the largest consolidation of liquidity in crypto markets. Yet, these funds remain largely untapped because users must access lending markets to acquire stablecoins. VDEX is uniquely able to accept crypto majors, such as Bitcoin and Ether, as collateral at 100%. This consolidates the user journey, trust assumptions, and loans, from two platforms (Lending market + perpDEX) to one (VDEX). This represents an improvement in UX and security.

5.2 Eliminating Bad Debt

In order to support VAC, VDEX must be prepared to liquidate user accounts based on their positions and collateral. VDEX is able to achieve this by sub-second price updates that ensure that the user's balance is above ½ of the initial margin (IM), even if IM is constantly changing. Other protocols do not have access to ZeroGas STATE transitions, which are required for supporting volatile assets.

VDEX must also be able to ensure that collateral retains its value if forfeited by the user. This is done by shorting assumed ESCROW on the VDEX platform itself. This creates a delta-neutral stablecoin. VDEX could choose to directly sell these assets as well.

5.3 Sustainable Bitcoin Yield

VAC is accepted for both trading and liquidity providing. This means users could deposit Bitcoin and receive BTC-denominated yield while maintaining their principle. This would only be the second sustainable (yield deriving from trading fees), source of Bitcoin yield post-Babylon.

6 Decentral-Limit Orderbook (DLOB)

6.1 Trading

VDEX runs a central-limit order book, dubbed DLOB here as a pun.

6.2 Funding Rate

The difference between the VDEX price, as derived from the depth of BIDs and ASKs, and the decentralized oracle price, will determine the funding rate. If price_VDEX is higher than price_Oracle, then the market is overbought, and longs will pay shorts for the privilege. If price_VDEX < price_Oracle, shorts will pay longs. The funding transfer is entirely P2P with no cut taken by VDEX. Its primary intention is to ensure the stability of the exchange.

6.3 Liquidations

The Virtual Market Maker (VMM) will perform liquidations when user balance reaches below the maintenance margin, defined as ½ of the initial margin. As previously stated volatile assets will be considered in liquidations. If through the act of liquidation, the user's balance arrives at less than ⅔ of the maintenance margin, the VMM will assume all remaining liquidity into the Virtual Insurance Pool (VIP). This is to ensure liquidations are not unprofitable on average.

6.4 Pairs

All pairs are denominated in USD. Initially, majors will be supported. VDEX will soon announce support for assets of all kinds, even those not listed on supported SETTLEMENT LAYERS, such as Solana.

6.5 Isolated Positions

Traders may create isolated margin accounts, whereby liquidations will not affect the cross-margin account.

6.6 Leverage

Users may access large amounts of additional funds at rates established by VIRTUAL DAO. This can occur in the long or short direction.

7 Virtual Market Maker (VMM)

7.1 Introduction

VMM is a community-owned and operated market maker that will use deposited funds from liquidity providers to ensure the stability of VDEX.

7.2 No Impermanent Loss

Yield and principle will be paid out in the original denominated ESCROW, meaning there is no impermanent loss. VMM is also designed to minimize possible drawdowns.

7.3 Sustainable Bitcoin Yield

VMM will accept VAC, including Bitcoin and Ethereum. VMM thus offers yield on a trillion-dollar asset where few other options are available.

7.4 Full Self-Custody

When depositing into the VMM, users access a percentage of net asset value (NAV). NAV syncs onchain every 8 hours. Withdrawal requests will be executed after two syncs to allow the VMM to scale back positions. Funds are then claimable four days after the request has been executed at the previous NAV.

8 Technical Architecture and Flow

8.1 Overview

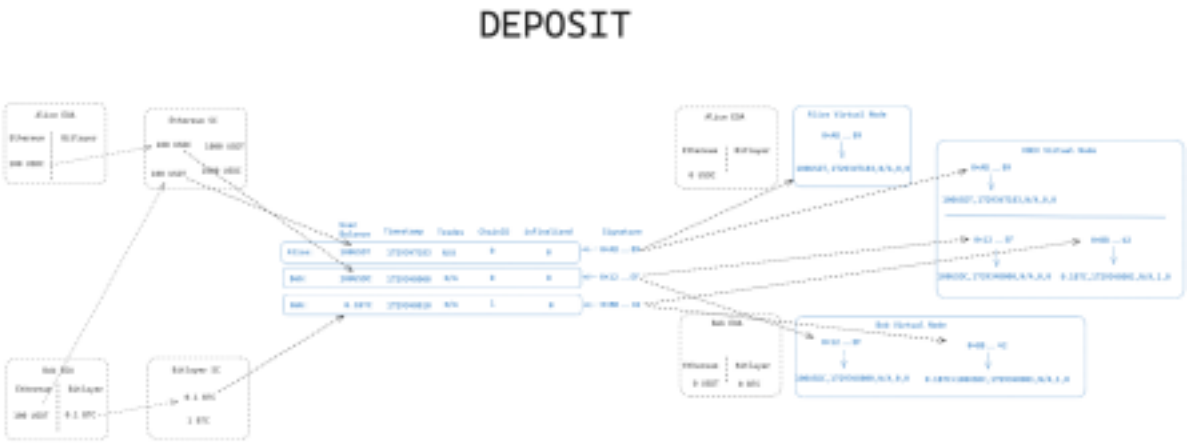
The VIRTUAL ROLLUP is the first to split STATE and ESCROW. In such, the protocol is split into two parallel networks: (1) ZK STATE CHANNELS: whereby users automatically run VIRTUAL NODES that make them the validators and processors of STATE and (2) SETTLEMENT LAYER: underlining blockchain architecture used to arbitrate STATE and store ESCROW.

1. Settlement Layers:
    a. User's SECP256k1 Key-Pair
    b. Blockchain Smart Contracts
    c. ESCROW
        i. ERC-20 Tokens
    d. Functions:
        i. deposit()
        ii. withdrawSchnorr()
        iii. setCombinedPublicKey()
        iv. withdrawAndClosePositionTrustlessly()
            1. settleDisputeResult()
        v. partialLiquidation()
        vi. pause()
        vii. unPause()
2. ZK State Channel
    a. VIRTUAL NODE
    b. P2P Gossip Network
    c. STATE
        i. SCHNORR SIGNATURES
        ii. PAYLOAD
            1. User Balance
            2. Timestamp
            3. Trades
            4. ChainID
            5. isFinalized
    d. Functions:
        i. createTradingSignature()

ii. liquidatePosition()

8.2 Deposit()

Let us examine how the user deposits funds to enter the Virtual Rollup via the SETTLEMENT LAYER'S blockchain smart contracts.



As seen here, Alice and Bob deposit funds from the Ethereum SETTLEMENT LAYER and the Bitlayer SETTLEMENT LAYER. This publicly-viewable Deposit function acknowledges their deposits by returning a SCHNORR SIGNATURE that contains five data fields: User Balance, Timestamp, Trades, ChainID, and isFinalized, thereafter called the PAYLOAD.

Why Does VIRTUAL ROLLUP Use Schnorr Signatures?

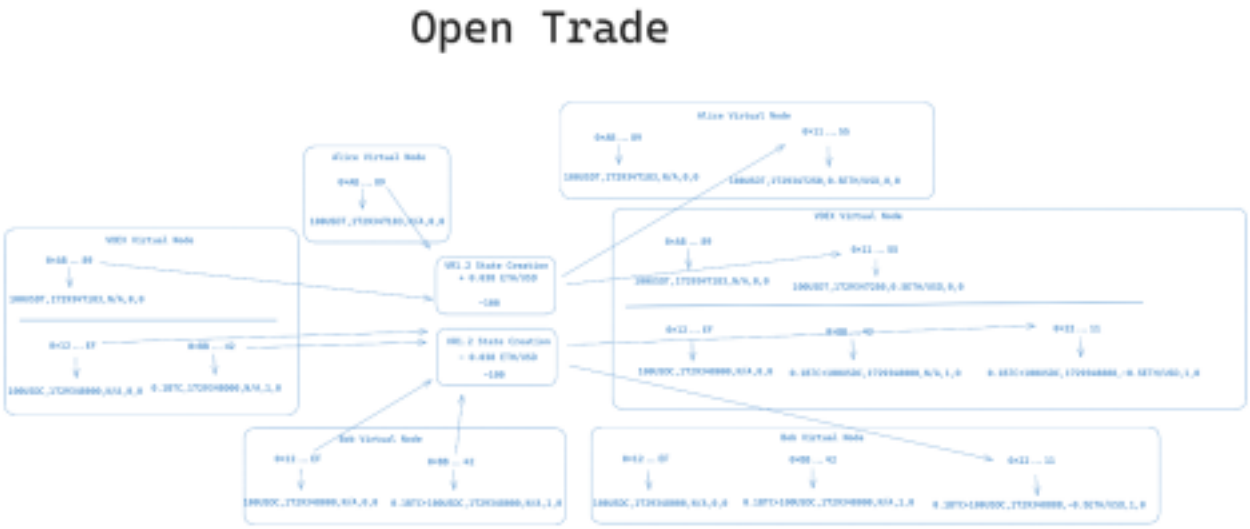
1. Versatile to existing systems:
    a. EMV-Compatible (not true of many other ZKPs).
    b. Gas efficient (520-bit signature size amounts to 6,000 gas per proof verification).
2. Zero Knowledge properties:
    a. SETTLEMENT LAYER can prove the presence of relevant public keys without revealing the public keys themselves.
    b. Number of users (N) is unbounded.
    c. Signature size is constant at 520-bits regardless of the size of PAYLOAD or N.
3. Sub-Millisecond Proof Generation and Verification

The SCHNORR SIGNATURE and PAYLOAD are publicly viewable by everyone, as it exists on the public SETTLEMENT LAYER. The VIRTUAL NODES of Alice, Bob, and VDEX all collect and store the SCHNORR SIGNATURE and PAYLOAD, which together maintain STATE. Now, if at any point hereafter one of the users becomes malicious or goes offline, the opposing VIRTUAL NODE can submit the SCHNORR SIGNATURE and PAYLOAD to the SETTLEMENT LAYER'S smart contract, prove STATE, and collect ESCROW. This property is maintained throughout, ensuring VDEX never assumes custody of user funds on the VIRTUAL ROLLUP.

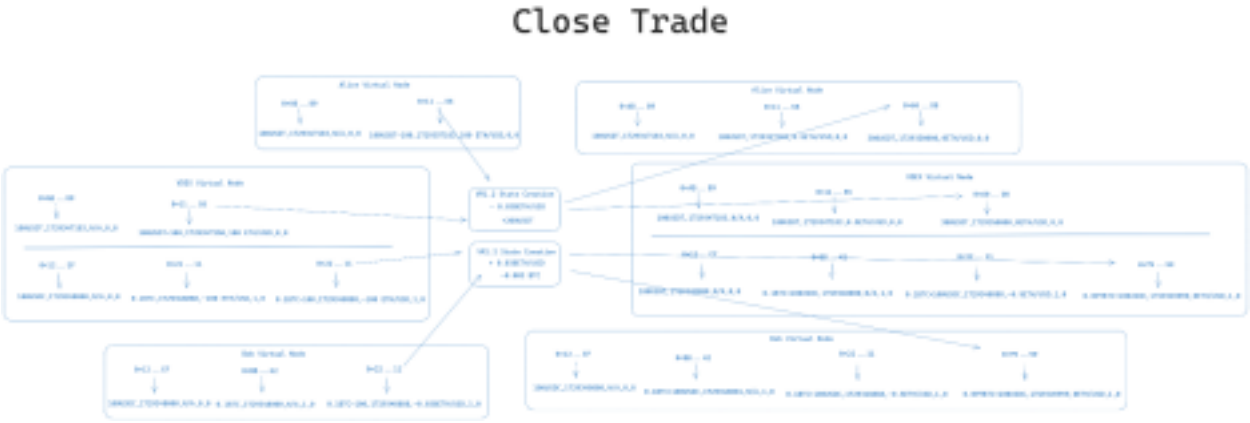
In essence, a SETTLEMENT LAYER can spin up ZK STATE CHANNEL, together forming a VIRTUAL ROLLUP.

8.3 CreateTradingSignature()

Now let us examine how a user makes a trade in the ZK STATE CHANNEL.



As seen here, Alice's and VDEX'S VIRTUAL NODES are creating new STATE alongside Bob's and VDEX's VIRTUAL NODES. This might be thought of as having two, or N, ZK STATE CHANNELS in parallel. On the left, the VIRTUAL NODES only contain the STATE from the Deposit flow, on the right, the VIRTUAL NODES now contain the additional STATE from the 0.038ETH position. Now if VDEX claimed that Alice or Bob did not complete this trade, their VIRTUAL NODES will provide the latest SCHNORR SIGNATURE and PAYLOAD to prove that they did, and collect ESCROW.



This is repeated in the example of closing a trade. Only the last SCHNORR SIGNATURE remains relevant for proving state. Using the ZK STATE CHANNELS, the VIRTUAL NODES can stream STATE over the P2P Gossip Network with sub-millisecond finality and ZeroGas cost.

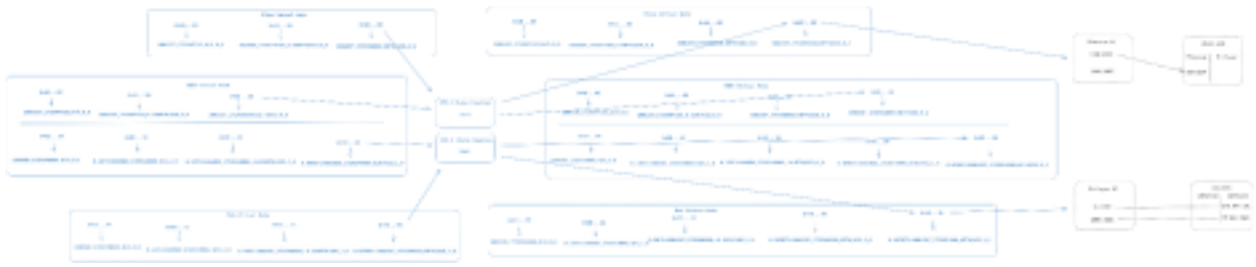
Utilizing Zero Knowledge Proofs in Transmission

VIRTUAL ROLLUPS must support a gas-efficient, EVM compatible signature for the SETTLEMENT LAYER,

however it is able to convert these SCHNORR SIGNATURES to other proof-schemes as needed for each VIRTUAL ROLLUP. This would be possible by allowing any VIRTUAL NODE to convert SCHNORR SIGNATURES into other ZKPs, such as GROTH-16 SNARKS, FROST, or PLONKY3 proofs. Furthermore it is possible to replace SCHNORR SIGNATURES for SVM Rust-based signatures to support new SETTLEMENT LAYERS. These proof changes are not required for the VIRTUAL ROLLUP to function, but helpfully display the modular architecture of the protocol.

8.4 WithdrawSchnorr()

This is the final basic function of the VIRTUAL ROLLUP which unlocks and transfers ESCROW from the smart contract to the user's wallet.



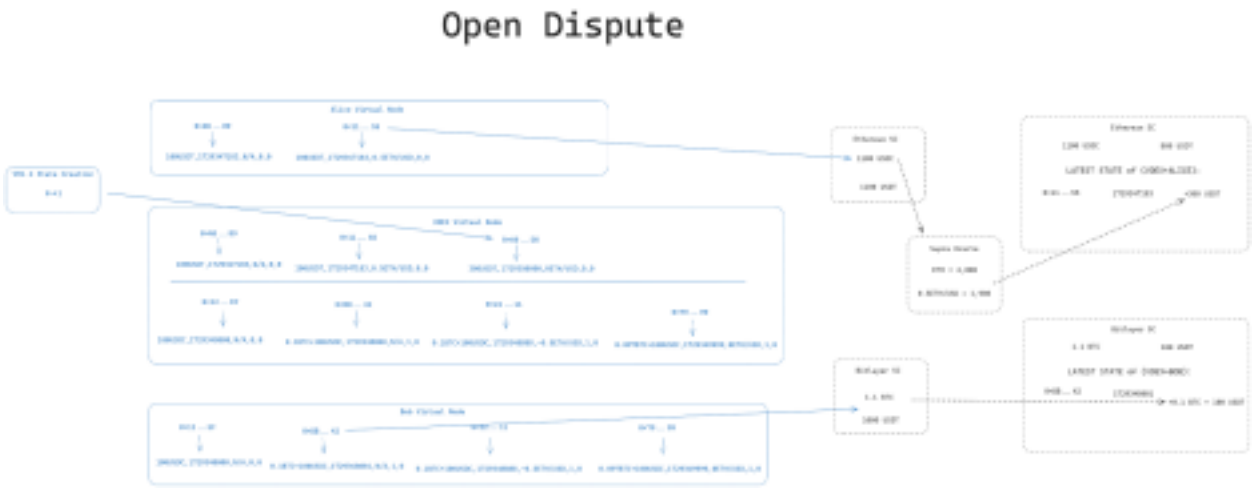
VIRTUAL ROLLUP feature: isFinalized

First, the VIRTUAL NODES agree to create a state where isFinalized is TRUE, which allows the smart contract to unlock the ESCROW immediately. If isFinalized is FALSE, it signals that 1 to N-1 of the users have not agreed to finalize STATE. This calls withdrawAndClosePositionTrustlessly() which triggers a 24-hour fraud proof window which shall be explored in the subsequent section.

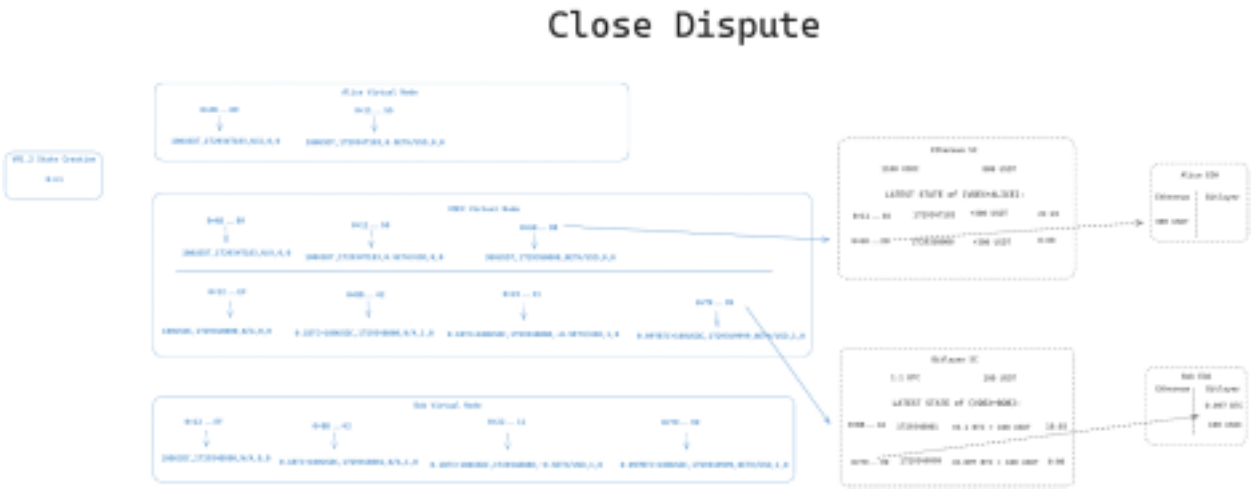
Once the VIRTUAL NODES have created the STATE with isFinalized TRUE, then the user will submit the PAYLOAD and corresponding SCHNORR SIGNATURE to the SETTLEMENT LAYER, which will process the transfer of ESCROW immediately.

8.5 WithdrawAndClosePositionTrustlessly()

If the VIRTUAL NODES cannot (1) disagree on the current STATE, (2) cannot establish secure communication with each other, or (3) suspect the other party to be malicious, they may at any time call this function and submit their latest SCHNORR SIGNATURE and PAYLOAD.



It must be possible to process withdrawals from VIRTUAL ROLLUP in all scenarios. This means powering withdrawals to occur without unanimous consent of the VIRTUAL NODES. This is the purpose of the SETTLEMENT LAYER. As shown above, VDEX went offline or became malicious before relaying the latest STATE to Alice. Additionally, Bob submits a STALE STATE during when he had a higher userBalance. OpenDispute begins a 24-hour period where it will execute the PAYLOAD of the latest STATE submitted.



In both cases VDEX comes back online within a 24-hour window and submits their latest STATE. Because no later STATE exists, the SETTLEMENT LAYERS execute these PAYLOADS, fairly giving Alice her winnings and capturing Bob's true losses. It is worth noting that this process can happen in reverse, so users can CloseDisputes opened by VDEX.

8.6 PartialLiquidation()

PartialLiquidation() is an artifact of VDEX's decision to create a VAULT of ISOLATED USER (VIU).

What is VAULT of ISOLATED USER (VIU)?

When users spin up a ZK STATE CHANNEL in order to enter the VIRTUAL ROLLUP, the SETTLEMENT LAYER moves their funds into a specific pool, dedicated for only their ESCROW. This bifurcation of user funds

from one another ensures Proof of Reserve.

But this presents another challenge. Imagine Alice and Bob are the only two trades and have placed opposite orders. If Alice were to win that bet and collect half of Bob's funds, what would happen if she requested to Withdraw() her funds immediately? How does Alice trustlessly and automatically claim Bob's funds? The solution is that VIUs do not directly trade against each other–and VIUs instead make claims of ESCROW from the VIRTUAL INSURANCE POOL (VIP).

What is VIRTUAL INSURANCE POOL?
VIP is a community-owned and operated pool, run by VIRTUAL DAO. The VIP has three key functions. The first of which is to process VIU transfers. This way, if Bob's ESCROW is still locked, Alice can get an advance on the ESCROW from VIP, and VIP will collect Bob's ESCROW after a possible dispute.

The second of these functions is to insure user funds, so any possible hacks would first take ESCROW from the VIP, leaving user funds untouched. But perhaps even more importantly, the final function of VIP is to protect VIUs from a malicious VDEX. Imagine an evil VDEX that generates Carol, a malicious actor under the protocol's control. VDEX could then process any STATE, and make a claim on the entire SETTLEMENT LAYER'S ESCROW. This massive exploit would jeopardize the connection of STATE and ESCROW. However, only funds in the VIP can ever be withdrawn by a non-owner of the VIP.

This means that a malicious VDEX cannot falsely acquire Alice's funds because they are in isolated, protected pools, not one common cesspool.

However this protection also has the effect that if the VIP is empty, Alice will be unable to retrieve ESCROW from Bob's VIU, despite having the necessary STATE. Therefore, an additional function was added to retrieve already lost funds within the VIRTUAL ROLLUP's STATE, but not yet registered on the SETTLEMENT LAYER. This function partialLiquidation, opens a dispute similar to WithdrawAndClosePositionTrustlessly(). In partialLiquidation() VDEX begins a 24-hour moving process from their VIU to the VIP, allowing Alice to later withdraw.

It should be noted that only VDEX can call this function. This is because only VDEX knows the counterparty's current STATE anyway. This has the effect that a malicious VDEX could freeze funds in VIUs, effectively erasing profits from the system (by first stealing all of the VIP).

This exploit is not too high in severity for two reasons:

(1) User deposits are completely safe. While it's true that possible profits could be stolen, the design of VAULT of ISOLATED USER (VIU)

(2) The maximum exploitable amount would be the associated hacker's VIUs and the VIP. This is deemed low severity because it is not a profitable exploit vector as any funds used for attacking or owned and controlled by VIRTUAL DAO, so an attacker who controls VDEX through the VIRTUAL DAO keys already has access to VIP.

8.7 LiquidatePosition()

VIP actually occurs on the ZK STATE CHANNEL level, and not the SETTLEMENT LAYER. This is because VDEX is able to update STATE without the live consent of the user–who could choose not to be live during a liquidation–through both a limit order from the user beforehand and the updated oracle price (which must be timestamped after the user limit order. This makes it possible for VDEX and the oracle to collude to pre-liquidate

users. But this would require the corruption of the underlying oracle, and would not represent a profitable attack vector because it primarily serves to disturb and censor user trading, not steal ESCROW.

8.8 Pause() and unPause()

Pause() is a novel mechanism devised to allow VIRTUAL DAO time to mobilize the keyholders to upgrade the contract. Pause() can be started by VDEX and will immediately halt withdrawals and all SETTLEMENT LAYER functions. However, it will automatically revert after 24 hours. It can only be used for 24 hours out of a 48 hour period, so VDEX could not immediately pause() again. Additionally, as this mechanism is used to upgrade contracts and also halt VDEX during investigations or suspicions of hacking, VDEX can also unPause() to call off the 24-hour window early.

Upgradable Contract and VIRTUAL DAO

VDEX will begin with an upgradable contract which will allow VIRTUAL DAO to make hot fixes if exploits are detected. Naturally, this will be decentralized with time. During VDEX BETA, VIRTUAL DAO will be controlled by a select few trusted members. VDEX BETA will also begin with a 24-hour withdrawal delay for WithdrawSchnorr(). At full launch, VIRTUAL DAO will comprise a double-digit number of signers with an honest majority assumption. At full launch, WithdrawSchnorr () will now be executed immediately. Finally, once the security of the VIRTUAL ROLLUP in its eternity has been assured, the contract will no longer be upgradable and VIRTUAL DAO becomes collectivized by $VRTL holders.

9 Decentralization Roadmap

9.1 Introduction

As outlined in this paper, VDEX's VIRTUAL ROLLUP architecture has no trust assumptions beyond smart contract risk, the reliability of the oracle, and the decentralization of the SETTLEMENT LAYER. To protect against these risks, VDEX will decentralize according to a stage system, as outlined by Buterin.

9.2 Stage 0

Stage 0 will be in effect during VDEX Beta and will be categorized by large user protection measures able to roll-back smart contract exploits. This includes:

1. Upgradable contract
    a. Controlled by VIRTUAL DAO with core contributors as signers
2. 24-hour withdrawal delay
3. VIRTUAL ROLLUP 1.3 is closed-source
4. Only one VDEX counterparty

9.2 Stage 1

The intention is for VDEX to move past Stage 0 rapidly, as early as Q1-Q2 2025. Stage 1 is categorized by:

1. Upgradable contract
    a. Controlled by VIRTUAL DAO with no less than 11 public signers
2. Instant withdrawals
3. Fully self-custodial and trustless

4. VIRTUAL ROLLUP 1.3 is closed-source
5. Multiple permissioned counterparties

9.3 Stage 2

Stage 2 is full decentralization

1. Frozen contract
2. Instant withdrawals
3. Fully self-custodial and trustless
4. VIRTUAL ROLLUP 1.3 is open-source
5. Unbounded permissionless counterparties

10 Conclusion

We believe that there are no less than five (5) innovations proposed in this paper. This includes the creation of (1) Pause() as a trustless way to give VIRTUAL DAO the proper time to fix exploits and (2) VAULTS of ISOLATED USERS to protect customer funds. It also includes the power of VIRTUAL ROLLUP 1.3, which gives way to (3) sub-millisecond decentralized trading, (4) sustainable Bitcoin yield, and (5) ChainID which creates a UNIFIED LIQUIDITY LAYER and full chain abstraction.

In conclusion, VDEX is certainly the best perpetual exchange protocol, as made possible through its modular design. Moreover, VDEX exemplifies VIRTUAL ROLLUP as a paradigm-shift of post-blockchain architecture and catalyzes other protocols to utilize and dominate through this superior technology.