SwapLand Decentralized Exchange: Protocol for fully decentralized assets trading

Author: Amirhossein Bahrani
SwapLand Foundation

9/17/2023

admin@swapland.exchange

SwapLand is a multi-chain decentralized exchange that offers advanced trading systems with Order-Book Spot and Perpetual markets. SwapLand Chain operates with relayers and data feeders which provide bridges and oracle networks. SwapLand Chain uses its innovative liquidity aggregation and real-world assets trading while providing exact futures as Spot and Perpetual Markets. The SwapLand Liquidity model extracts each token from different Ethereum Virtual Machine (EVM) supported chains and adds it into one single asset pool which delivers higher liquidity compared to other dexs. SwapLand bridge supports transferring assets through zero knowledge channels with anti-money laundering protocol (AML) established via relayers network and token holders.

1. Introduction

Blockchain technology is extending faster than expected, which indicates that its essential purpose must be fully developed and expanded, As a result, digital services based on blockchains that deliver security and decentralization must evolve. One of the most critical services that Blockchains suggest is decentralized trading, which brings safety and speed to asset trading.

There are two types of exchanges: centralized and decentralized. Centralized exchanges give users a variety of options for trading, but at its core, the assets are in a single place that is controlled by the central government, on the contrary, decentralized exchanges provide safety but they don't offer as numerous trading futures as centralized exchanges do. Also, Decentralized exchanges have multiple types, Automated Market Maker (AMM), Order Book, hybrid, etc. In between these, the most classical model is order-book which brings bids and asks to a list and matches them based on their price. AMM model is newly generated with liquidity pools and a hybrid model, which is a combination of the Order Book and AMM model.

SwapLand Exchange powers different kinds of advanced trading systems while keeping decentralization, safety, and velocity at great a level. SwapLand order matching works by batching orders at the beginning of every single block and matching them at the end of the block to remove front-running possibilities. SwapLand also offers trading of real-world assets, holding position with it, and sending it with the capability of Oracle networks. Also, SwapLand bridge accepts assets from different kinds of EVM chains, users can send their ABC token from ETH or any other EVM chain to SwapLand chain and trade it with any other assets.

SwapLand bridge technology allows users to deposit and withdraw their assets from the EVM chain to the SwapLand network or vice versa. Although Bridge offers a zero-knowledge proof channel with an anti-money laundering protocol to offer trading anonymously and safely.

2. Background

Today, Dexs produces great features such as Uniswap, which is an AMM mainly on the Ethereum chain. Also, there are Order Book decentralized exchanges that have on-chain Order Books and off-chain Order Books. Unoptimized On-chain Order Books on EVM chains cost high fees and long waiting times which can't handle significant orders in one block.

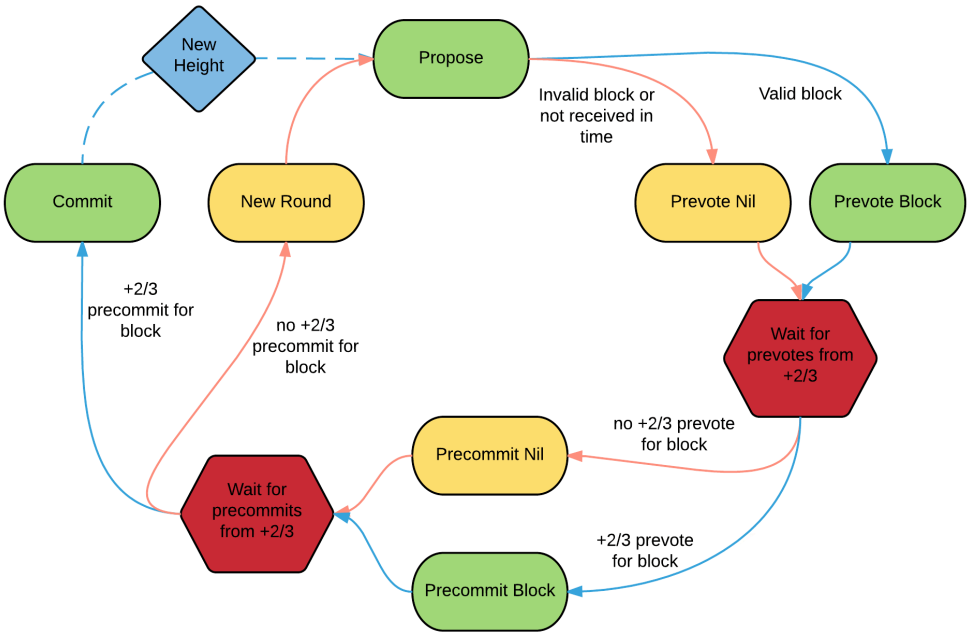
Off-chain order book decentralized exchanges offer incredible speed with lower levels of decentralization. They store orders off-chain and match them using central nodes. After matching, they submit the order results to a chain for validation.

SwapLand stores orders in the chain state but stores each pair's Order Book in the memory store to achieve faster performance and parallel execution.

3. SwapLand Matching Protocol

3.1 Tendermint Overview

SwapLand Chain has built on top of the Tendermint BFT and Cosmos SDK to deliver fast block time and a great rate of transaction per second for a swift user experience while trading.



Figures 1: Tendermint BFT Consensus [1]

Tendermint consensus is an easy state machine. Two voting levels are necessary to commit a block to the chain. Prevote and Precommit which needs more than 2/3 of validator votes to persist. At each stage, if validators can't submit their vote or the proposer validator goes offline and the block didn't get confirmed by 2/3 of the validators on that stage, the consensus mechanism will go for a need round with another validator proposing a new block. [1]

Tendermint introduces a new concept called Application BlockChain Interface (ABCI) which connects the Consensus layer to the Application layer (SwapLand Application). With the development of ABCI, chain builders can focus on their main application purpose instead of networking and consensus issues.

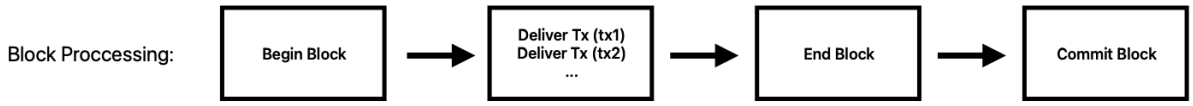


Figure 2: Block Processing Steps

ABCI functions work as Figure 2, At the beginning of the block, the Begin Block function will call, After the timeout period each tx will get verified through stateful checks in Deliver Txs, and on block finalizing, the End Block function will call. Finally, Block will commit to the network and pass through 2/3 of the votes.

Figure 3: CheckTx

When a node receives a transaction it checks the tx with a stateless check that we call CheckTx, after CheckTx succeeds, the tx will go to the node Mempool and gossip into other nodes for getting into the block.

CheckTx has different steps. Firstly, a transaction that was broadcasted needs to decode, after decoding the ValidateBasic function calls and checking the validation of messages that are in the transaction. Finally, AnteHandler is called, which will increase the account sequence number and gas will be reduced from the account using GasKeeper. After verification, Tx will become a candidate for showing into the block so it will reveal in node Mempool and the node relays it to its peer nodes.

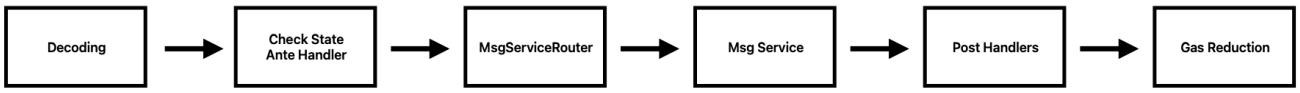


Figure 4: DeliverTx

DeliverTx runs for every transaction in a block, after tx is decoded, CheckTx will get called on that transaction. Then messages will get extracted from transactions and Msg Service will send each message to its module. After messages are executed and the state is changed, Post Handlers run for transactions and gas will be reduced from the transaction's sender. At each step, if the transaction fails the state will be reverted.

3.2 SwapLand Accepting Order

As mentioned before, SwapLand gets orders via messages. After submitting the order transaction, SwapLand will check order parameters via CheckTx in each different message. After CheckTx verifies tx goes for DeliverTx and submitting into Order Book. At the start of each block, (Begin Block) order submission starts, last block orders become open and will be added to the memory Order Book. After orders get into the Order Book and are mixed with remaining orders, sorting of Order Book starts which will sort all bids and asks of Order Book using the following formula:

$$O_{Price} > O_{Quantity} > O_{Block\ Time}$$

It shows orders with higher prices and quantity will get a higher place in the Bids list and the lowest place in the Asks list. It ensures that all nodes in the network who received the same order have the exact Order Book.

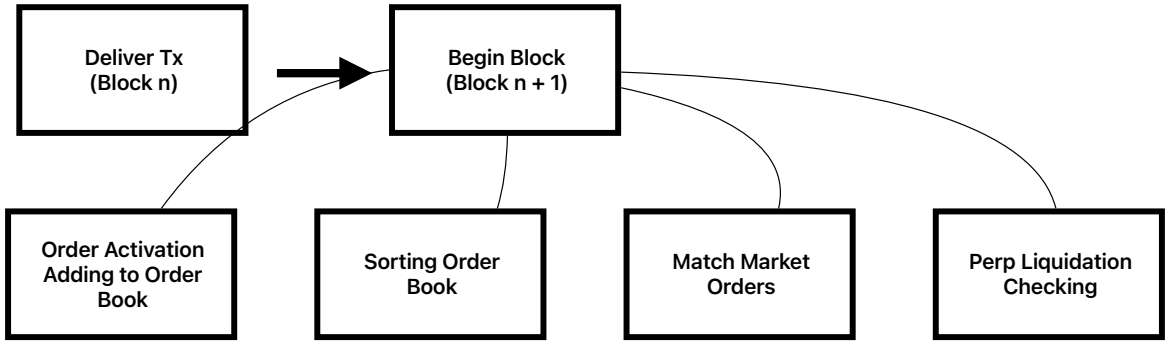


Figure 5: Order Submission

When a new order is broadcasted to the network, After verification and becoming valid in the block, the order will be added to the Order Book on the next block. After a while in the block that TX has been added to the Order Book, the order will get filled with available orders. Orders get packed and batched in the BeginBlock and ready for matching in EndBlock. Spot and Perpetual Markets order submission journeys are the same, Although they are processed in different modules, the progress is the same.

SwapLand supports different types of orders such as Market, Limit, Stop Loss, Take Profit, Trailings, and OCO orders. Market orders will processed at the end of BeginBlock after sorting and adding orders are finished. Firstly Market orders match and fulfill with each other instead of Order Book. At the beginning of matching, the last traded price is determined and will used for filling Market orders. After Market bids and asks are filled with each other, the remaining orders will match with the reversed side of the Order Book. The orders that weren't fulfilled with other Market orders will be matched with the highest bid or lowest ask on the Order Book.

3.3 SwapLand Parallel Order Matching

While the number of assets and tokens is increasing, the trading pairs of these tokens are extending. As a result, matching orders in asynchronous causes high processing time for all exchange pairs. SwapLand processes orders in parallel pairs and applies the changes to the memory state and Order Book and after that, the state storage shifts based on each pair.

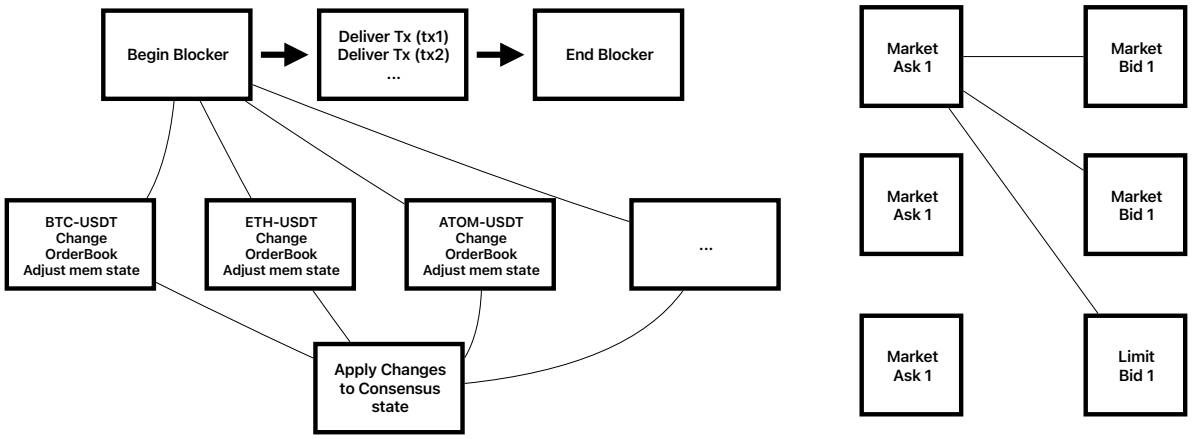
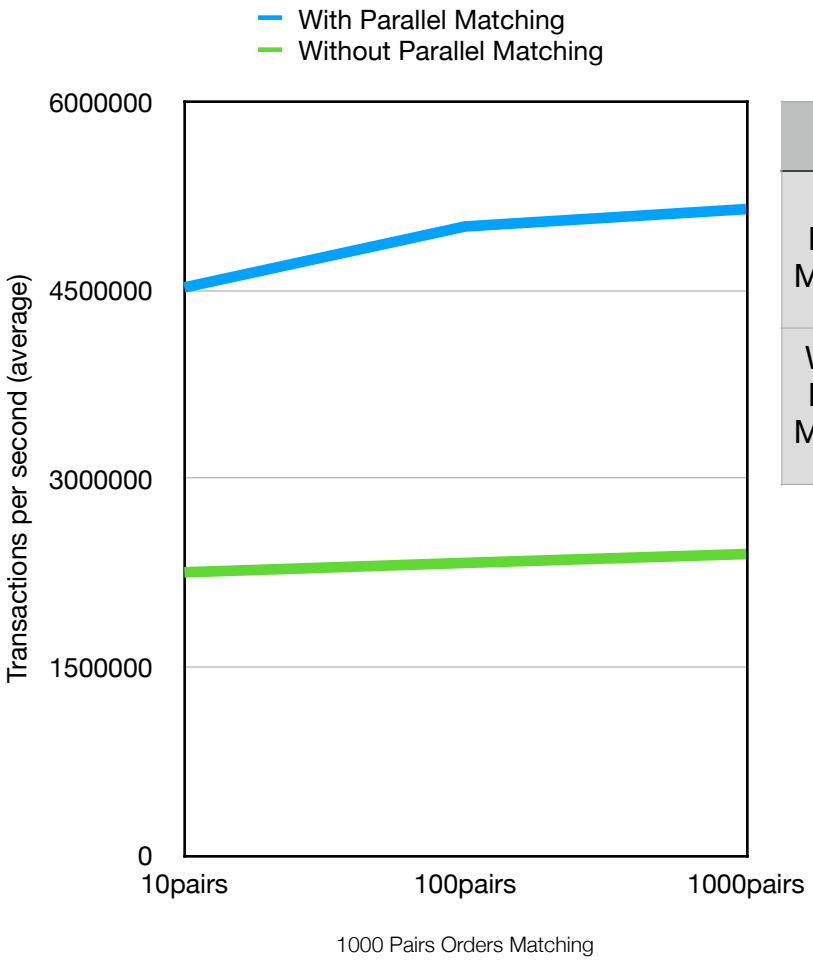


Figure 6: Market Orders Matching - Market Orders matching when they can fill with Limit order

Order Adding to each new pair memory store and order activation work in parallel. Firstly in the beginning block, order IDs that need to become valid will be collected and added to the Order Book, the Order Book will be sorted based on asks and bids in parallel functions, and finally, market orders matching start and filled orders id will be collected. At the end of the process, All of the IDs that have been collected will change the node state store and go for consensus, in this mechanism, the possibility of state collision and other orders of which pair first got matched will be removed and the network stability will become higher.

For EndBlock, orders got matched in a parallel manner like in BeginBlocker but with slight differences. In Perpetual Market, the second step of the liquidation checks will occur in EndBlock and the first step will run on BeginBlock. In EndBlock, after orders are matched with the OCO orders, liquidation for Perpetual markets will be checked.



	10pairs	100pairs	1000pairs
With Parallel Matching	4522410 tps	5008129 tps	5146250 tps
Without Parallel Matching	2254109 tps	2329133 tps	2399763 tps

As a result, in the 1000 pairs test, the SwapLand parallel matching system improves speed by up to **~72.7%** compared to normal matching, and in the 100 pairs test SwapLand parallel matching system improves speed by up to **~73.0%**. In this chart, we can figure out that by growing in number of token pairs, the SwapLand Parallel Matching score will become better than normal matching.

3.4 SwapLand Order Matching And Final Order Fills

As commented out in the previous step, SwapLand's main matching system works in the End Block. In the end block, orders that were sorted and modified in the Begin Block start to match in different pairs simultaneously. Finally, when the matching is finished and the Order Book of each pair has been modified, the state store will be adjusted based on the newly filled and particularly filled orders.

If a checking pair is created by the Oracle module(it has real-world assets on one side of trading), Matching will switch to Oracle mode and real-world asset pairs will have the same order types as all pairs.

In each round of matching (1 block ~1 second) minimum, maximum, open, and close prices are collected to be used for the candle stick chart and to check OCO orders.

SwapLand OCO orders work in both Spot and Perpetual markets, which powers advanced trading choices. SwapLand introduces trailing stop loss order (SL) and trailing take profit (TP) orders in OCO orders that the account can use. In addition, in OCO orders, accounts can submit TP and SL at the same time for both market types.

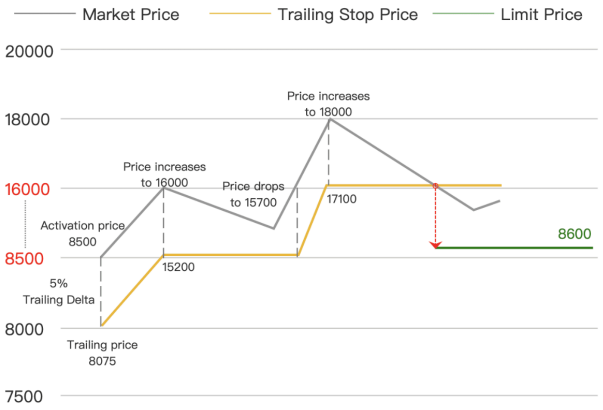


Figure 7: Trailing Stop Loss [2]

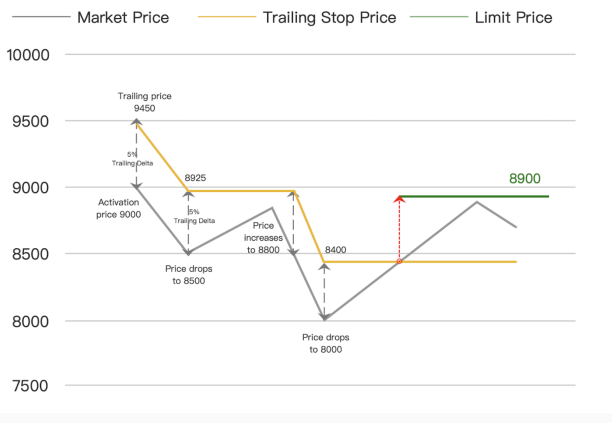


Figure 8: Trailing Take Profit [2]

For trailing orders, SwapLand obtains the following parameters for submitting orders: MinPrice, MaxPrice, MinPricePercent, and MaxPricePercent. MinPricePercent and MaxPricePercent are parameters whose value is from 1 to 100, 1 <= MaxPricePercent <= 100 and 1 <= MinPricePercent <= 100. MinPricePercent is a variable that sets the stop loss price to MaxPricePercent lower than the current market price. If the price goes up, the stop loss price will go up, and if the price goes down, the stop loss point will remain stable. For trailing take profit orders, the process is the same but reverted, which means that if the price goes down, the take profit price will go down, but if the price goes up, the take profit will remain exact.

Spot orders' type in SwapLand Exchange can be normal, All or none (AON), Immediate or cancel (IOK), or Fill or kill (FOK). In the perpetual market, the types are good till cancel (GTC) and good till date (GTD), which will remain active just for 1 day. SwapLand liquidation checks are automatic and will run on each beginning block and ending block for each pair.

The SwapLand perpetual market leverage pool comes from lenders that lend their money into the SwapLand lending pool, and in return, after each position, they receive a particular amount of position value in interest. For taking their loan back, after submission of taking back the loan, the initial amount + interest will be repaid to the lender. The SwapLand lending pool works by creating shares based on the token pool in-position and non-position. Borrowing from each pool occurs only by creating a perpetual order. Each share is created using the following formula:

Let $B_{inPosition}$ = balance inPosition of pool
Let $B_{nonPosition}$ = balance nonPosition of pool
Let $S = B_{inPosition} + B_{nonPosition}$
Let T = Total Available Shares
Let AmountPerShare = S / T | Let LoanAmount = AmountPerShare * RequestedSharesCount

Position can opened if $(B_{inPosition} / 4) <= B_{nonPosition}$ to ensure that every lender can finally take back their loans whenever they want.

4 SwapLand Relayers

4.1 SwapLand Liquidity Extraction

Most of the Order Book-styled decentralized exchanges use only one single chain for their desired liquidity, but when users want to trade their assets they don't need to transfer their assets to the chain on which the exchange is built (like ETH or BSC, etc) and then transfer them to the exchange for trading. Similar to centralized exchanges that accept tokens from different chains, decentralized exchanges also need this technology. This future allows the collection of the liquidity for a coin from different sources into a single location for trading. SwapLand merges tokens from different EVM chains to assemble mixed liquidity for trading. SwapLand makes deposits straightforward with low gas fees, users can directly transfer their assets regardless of their assets' origin to SwapLand Chain instead of bridging into a middle chain and paying extra fees.

SwapLand accepts tokens from different EVM chains, and an account can create a proposal to add support for a chain into SwapLand. In the "AddChain" proposal, 3 relayer account addresses in the SwapLand Chain and the selected chain must be set, and the SwapLand Main Bridge and SystemController must be prepared on the selected chain. In addition, chainId and chainName must be assigned.

After the proposal is approved by the SwapLand community, addresses become default relayers of that chain. After the proposal is passed, every account can submit a join request into the relayer set by locking a certain amount of SwapLand Token. Accounts can become relayer by running the SwapLand Relayer software. Also, tokens can be added using proposal confirmation, and on selected chains, tokens can be used for deposit and withdrawal.

4.2 SwapLand Bridge

SwapLand Protocol contains a set of smart contracts in each EVM chain that will be approved via the SwapLand decentralized government. These contracts contain SystemController, Bridge, Vault, ZkpRouterBridge, ZkpSingleBridge, and Governance. Relayers use the SystemController contract to handle entrance and removal requests of relayers on specified chains.

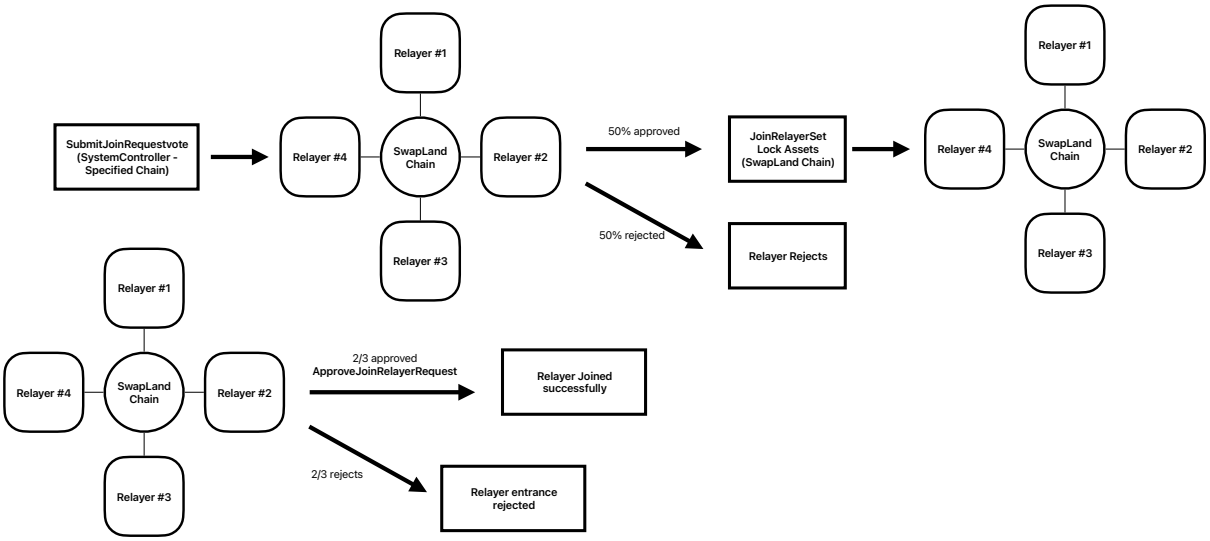


Figure 9: Relayer Diagrams

SwapLand Relayers are observers on each chain on the smart contracts package that was approved. Accounts can join into the SwapLand relayer by submitting SubmitJoinRequestVote into SwapLand SystemController, current relayers witness the request and approve or reject the request, for approving and rejecting relayers, they submit their vote into SystemController with confirmation signature, signature created using the following formula:

$$\text{SHA256}\ (\ P_{\text{Requested Relayer Address}} + P_{\text{Chain Id}} + P_{\text{Voter Address}}\ )$$

After 50% of voters confirm the request, the relayer is eligible to request joining into the relayer set in the SwapLand Chain. After sending a JoinRelayerSet request, 2/3 of the relayers must approve the request and submit their vote into the SwapLand Chain. Then, when verification succeeds relayer joins the set and can participate in deposits and withdraws. RequestToLeaveRelayerSet works like joining, Relayer submits its leave request into the SystemController, and after 50% of the relayers verify the leave request, the relayer status in the SystemController becomes inactive and is eligible to submit a leave request into the SwapLand Chain. 2/3 of the relayers must accept the remove request, and the amount that was locked in JoinRelayerSet minus the slashing amount will transfer to the relayer account. Slashing will occur when 2/3 of the relayers reach consensus on the same vote but the relayer has submitted a different vote to SwapLand Chain.

Deposit and withdrawal works using relayers. For the deposits, users transfer their assets to the SwapLand Vault using the SwapLand Bridge. After 2/3 of the relayers approve the deposit request, the assets will be minted into the SwapLand Chain. Withdrawal works like deposits,

selected tokens burned from SwapLand Chain, and after 2/3 of relayers confirm the request, assets will unlock from the selected chain.

For withdrawal, after the user sends a withdrawal request to SwapLand Chain, Relayers will validate the tx and send a signature with the acceptance tx. After that, the account can submit a batch of signatures to the withdraw function in SwapLand Bridge and withdraw it successfully. Users can choose which chain to withdraw their assets to. Assets in SwapLand Vaults that are pegged in the SwapLand Chain are always synced with their native assets. Users can change their withdrawal selected chain to get their assets based on their preferred chain, which indicates they can bridge their assets from one EVM chain to another EVM chain easily.

4.3 SwapLand Zero Knowledge Proof Bridge

SwapLand's second bridge is the Zero Knowledge Proof Bridge which is only used for depositing into SwapLand Chain and it has an anti-money laundering protocol (AML). Users can trade anonymously after they transfer their assets using ZKP Bridge. SwapLand ZKP disconnects the connection between the sender from the origin chain to the SwapLand Chain. SwapLand ZKP Bridge has one main router which controls ZKP single bridges (gates). SwapLand ZKP Bridge can add a gate, open and close.

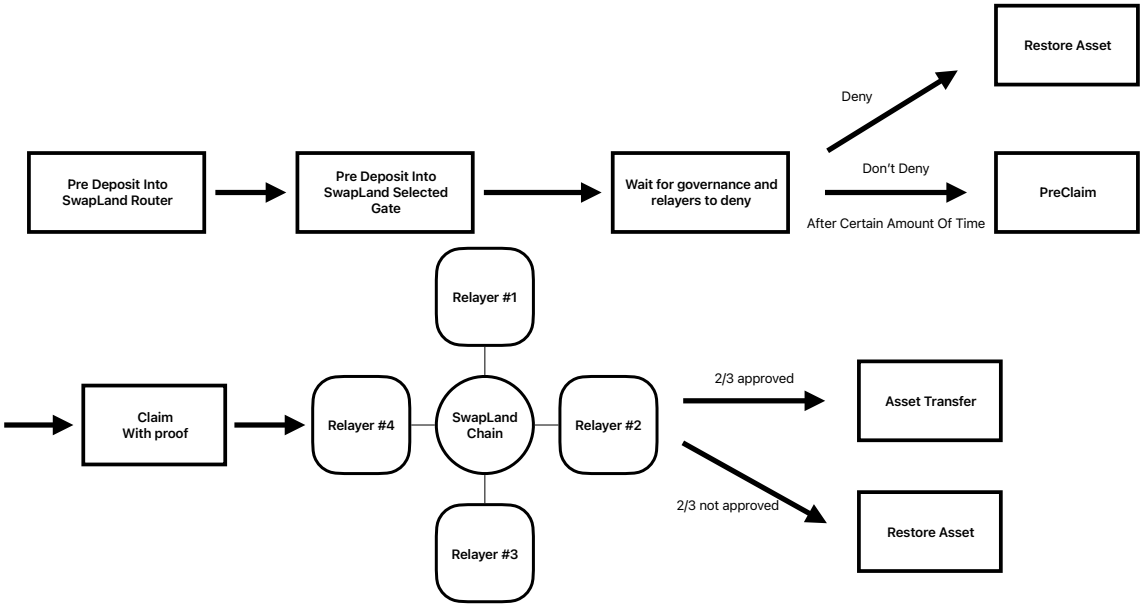


Figure 10: ZKP Bridge

Anti-money laundering works by enabling the community and relayers to interfere in the deposit process. When a potential money laundering case is seen by relayers or the community, they can send a rejection proposal on the chain in the Governance smart contract and cancel the deposit

using the SwapLand ZKP bridge, The remaining balance will be sent back to the account on the chain that was transferred.

Users first assign their SwapLand Chain address with the source chain account address to the ZKP router. Accounts send their request with another account address using the preDeposit function in the ZKP router after that router calls the selected gate and the money is transferred from a user account to SwapLand Vault. After a selected amount of time that can be modified on each chain based on block time, if relayers don't deny the request or the governance deposit rejection proposal about that request doesn't get passed, the user can call a Claim with generated snark proof with the address assigned to the SwapLand Chain address. Then, the request will be broadcast to relayers, and after 2/3 of the relayers approve the request, the asset will be transferred to the SwapLand Chain address.

In this process, users need three addresses, two of which are on their source chain account, and one of them is a SwapLand chain account address. Users sync the SwapLand chain account address with one of the source chain account addresses and then follow the above steps to transfer their assets, in preDeposit step, users create a nullifier and secret and the commitment hash, which is pedersenHash [5] of the concat of buffers of nullifier and secret. Then, send preDeposit with the account address that is not synced with the SwapLand Chain address. For Claim, compute MiMC hash [3] of concat of buffers of nullifier and secret that ends with root R in contract stored roots and computes pedersenHash [5] of nullifier for nullifier-hash. Then, calculate the proof with the synced account using the groth16 [4] proof. Finally, send parameters into the Claim function in the router, and after verification, the user can see assets on the SwapLand Chain.

5 SwapLand Oracle

SwapLand users can trade real-world assets using Oracles. Oracles are data providers that consistently submit pair prices to the SwapLand Chain. At the end of the blocks, votes are counted, and the average of all prices is saved as the final price. If the price submitted by Oracle has a 5% difference from the average of prices, the Oracle address slashes. To join the Oracle network, the addresses must lock a certain amount of SwapLand Chain native token. In addition, new pairs can be generated by creating proposals and passing them on to the community.

Users can buy real-world assets that are on one side of the pair, i.e., for pair EUROUSDT, users can buy EURO based on the price determined by Oracle by calling a static function to burn their USDT and mint EURO. After a while, they can sell their EURO by submitting an order into the Order Book. This has a significant impact on newly generated trading pairs with the forex market, stock market, and more.

Each oracle can configure its feeding pairs and data API from which the price came, but the price must be trustable to concur with the final price submitted by the oracles. Oracle pairs can go to the Spot and Perpetual markets based on the community listing.

After the number of these assets in the chain grows, oracleized tokens can create pairs with each other. For instance, we have EURO-USDT and XAU-USDT, and users can create the XAU-EURO symbol for trading gold with the real euro.

6 References

[1]: docs.tendermint.com. (n.d.). *What is* Tendermint | *Tendermint Core*. [online] Available at: https://docs.tendermint.com/v0.34/introduction/what-is-tendermint.html.

[2]: BinanceUS (2023). Trailing Stop Orders: What They Are, and How To Place One. *Trailing Stop Orders: What They Are, and How To Place One*. Available at: https://support.binance.us/hc/en-us/articles/16129894290071-Trailing-Stop-Orders-What-They-Are-and-How-To-Place-One [Accessed 20 Sep. 2023].

[3]: Albrecht, M., Grassi, L., Rechberger, C., Roy, A. and Tiessen, T. (2016). *MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity*. [online] ePrint IACR. Available at: https://eprint.iacr.org/2016/492 [Accessed 20 Sep. 2023].

[4]: Groth, J. (n.d.). On the Size of Pairing-based Non-interactive Arguments. [online] doi:https://doi.org/10.1007/978-3-662-49896-5.

[5]: iden3-docs.readthedocs.io. (n.d.). *Pedersen Hash — iden3 0.1 documentation*. [online] Available at: https://iden3-docs.readthedocs.io/en/latest/iden3_repos/research/publications/zkproof-standards-workshop-2/pedersen-hash/pedersen.html [Accessed 20 Sep. 2023].