Account Abstraction (AA)

This page talks about one of the main ZeroLend's features - Account Abstraction and how it is integrated in the protocol for better user experience. The DeFi ecosystem can be complex; therefore, enhancing the user experience in interacting with DeFi applications to make them more intuitive and competitive with conventional TradFi applications is crucial.

There are two types of accounts in Ethereum:

1. Smart contract accounts where developers store their code (smart contracts).
2. Externally owned accounts (EOAs) where users store their tokens (wallets like Metamask).
3.

Users generally interact with DApps using EOA wallets, which is the only way to start a transaction or execute a smart contract. This has many downsides like -

- The wallets are tightly coupled with a single key
- They are hard to secure - keys get stolen
- They are hard to recover - keys get lost
- Wallets don't support multi-sig
- No social-logins
- The user must pay the gas fees
- Must have ETH or native chain token to pay gas fees
-

“Abstraction” is the act of pulling or drawing something away. Account abstraction is a change to the current model that aims to abstract away the details of blockchain from users and enable a Web3 user experience that is seamless. From a user's perspective, account abstraction utilizes a smart account that eliminates the need for storing seed phrases/private keys, paying gas for transactions, etc.

Does zkSync support Account Abstraction?

zkSync supports native Account Abstraction. Native Account Abstraction on zkSync fundamentally changes how accounts operate by introducing the concept of Smart Accounts and Paymasters.

The differences between Native Account Abstraction and EIP 4337:

Aspect zkSync Native AA Ethereum EIP 4337 Implementation Integrated at protocol level Avoids protocol-level implementation Account Types All accounts, including EOAs, behave like smart contract accounts; support paymasters Separate transaction flow for smart contract accounts; EOAs lack paymaster support Transaction Processing Unified mempool, Operator bundles transactions for both EOAs and smart contracts, leading to a single flow Separate mempool for smart contract accounts, Bundlers send user operations to EntryPoint contract, resulting in two flows Paymasters Support Supports paymasters for both EOAs and smart contract accounts No paymaster support for EOAs; implemented only for smart contract accounts in a new transaction flow

How Does ZeroLend Implement AA?

ZeroLend integrates various zkSync-native AA features in the following ways:

Paymasters : Allows the protocol to either subsidize or allow users to pay for transaction fees with ERC20 tokens without having any ETH in their wallets.

Social Logins & other authentication options : Allows users to have a wallet that is in control of their secure enclave device (Face ID, Fingerprint scanners, etc..)

Delegated transaction : Allows users to authorize ZeroLend to execute limited actions on their behalf without losing custody of their funds.

Previous Liquidations Next Paymasters Last updated2 months ago On this page *Does zkSync support Account Abstraction? * How Does ZeroLend Implement AA?

Was this helpful? Edit on GitHub