# Introduction

Hi, all!

I’m Tim from Exclave.

We’re building a novel off-chain scaling solution on Ethereum for increasing scalability and reducing the cost, of Exclave. Exclave utilizes zkEVM to implement a universal off-chain coprocessor system and employs Zero-knowledge Fraud Proofs to handle disputes. With Exclave, developers can build data-driven dApps, enabling the low-cost execution of complex computations at the entire chain level while leveraging the security of the entire chain.

# Background

With the increasing number of Ethereum rollups, liquidity, and traffic have become more widely dispersed, leading to increased fragmentation of liquidity. This situation not only significantly raises the difficulty of users’ on-chain operations but also severs the flow and interaction between user communities of different projects.

We are pleased with the efforts of developers in the field of chain abstraction, as they have provided possibilities to reduce the interaction costs for users across different chains. However, there are significant differences in the infrastructure performance and cost requirements among various projects, especially in the case of Fully On-chain Gaming (FOCG).

zk-Rollups require all off-chain operations and validity proofs to interact and synchronize with L1, resulting in significant redundancy for the majority of off-chain undisputed states and executions. Meanwhile, most FOCG startups struggle with the data availability costs imposed by rollups. A dispute protocol based on zk fraud proofs will offer a completely new scaling approach – aiming to build a user-friendly and cost-effective off-chain environment while ensuring security under the protection of Ethereum.

More importantly, based on a loosely decoupled data validity proof mechanism, the off-chain network can easily scale out by running multiple parallel processes. This is determined by the shared-nothing nature of the services it provides. The smallest unit for each service is a game match, and they are entirely independent of each other, allowing them to operate independently. All requests within different services can also be executed in parallel since all services fundamentally store and operate on user data in separate states.

# Designs

## Communication and Synchronization

The interaction model of the zk-Coprocessor RPC aims to mirror Ethereum’s RPC while incorporating additional endpoints specific to the coprocessor network. These endpoints enable users (like Metamask, dApps, AA Wallets, Etherscan, etc.) to interact with off-chain coprocessor nodes. By offering JSON RPC endpoints via an HTTP interface, users can retrieve data, process transactions, and engage with the transaction pool, supporting operations like batches, proofs, L1 verification transactions, and more.

Here, we will use the MUD-based game Sky Strife as an example to further explain the communication process:

- Players initialize the game on-chain (which may involve staking a certain amount of assets).
- During the initialization process, zk-Coprocessor will call the on-chain zkRelayer contract, triggering an event to provide information for the Relayer

to monitor and execute off-chain operations, such as setting up game accounts and initializing off-chain game session contracts.

- The conclusion of the game will be triggered by an event from the off-chain game session contract, generating the corresponding proof, and the dispute resolution mechanism will determine whether to submit it to the chain.
- If submission to the chain is required, the zk-proof will be submitted to EthTxManager

by zkCoordinator

, requesting on-chain verification.

- The results of on-chain verification will be accepted and processed by the zkRelayer

contract, and the on-chain contract of the game will settle the on-chain state based on this result.

[

1280×487 131 KB

](https://ethresear.ch/uploads/default/original/2X/3/39a42261f16cff852355cf24182cd21ac62885e8.png)

## zk-Proof Generation Coordination

The generation of zk-proofs in zkEVM follows a sequential process: Tx → block → batch → proof. However, the production of proofs in Exclave operates per-game sessions, detached from the order of block production. This leads to a parallel proof generation system where the states of multiple batches are decoupled.

Initiating a game session involves on-chain triggers where users initiate gaming activities. Conversely, the conclusion involves off-chain execution and eventual on-chain settlement based on the outcomes derived from off-chain game sessions. This process ensures security under Ethereum's protection while minimizing on-chain operations' costs.

[

1280×843 72.7 KB

](https://ethresear.ch/uploads/default/original/2X/5/565b9853e3e6f0a770d0532ceeb3f3478fff6361.jpeg)

## Dispute Mechanism

The OP + ZK dispute resolution mechanism optimizes effectiveness, security, and cost-efficiency. The protocol closely aligns with OP but differs in generating zk-proofs on L1 only when a challenge is triggered. When a game concludes, the Prover uploads the final state/transactions to L1. A potential challenger reviews the evidence to decide whether to challenge. If challenged, the Prover generates and uploads the proof and corresponding data to L1 for verification.

[

1280×1044 162 KB

](https://ethresear.ch/uploads/default/original/2X/e/e99b3b898c7b4e0bd2ae5187563ac8c2211eb7dc.png)