Problems of traditional zk-rollup

Traditional zk-rollup migrates the computation from on-chain to off-chain, and requires a solidity contract on the chain (ethereum) to verify that the rollup data is valid. However, this architecture causes some problems:

1. Each solidity contract of rollup deployed on ethereum is independent of each other. Users need to deposit their assets to different contracts. Therefore, in fact, users' assets are scattered and cannot be managed and used in a unified way.
2. Solidity is not good at handling the verification of zk proof. In addition, transactions that submit and verify zk proof on ethereum are executed serially.
3. The solidity contract of each rollup does not have a unified specification. This increases the complexity of the whole system, while limiting many things, such as cross-rollup communication.
4. Uploading off-chain data to solidity contracts for verification brings high operational costs. For example, an AMM zk-rollup costs hundreds of thousands of dollars per month to operate, most of that is transaction gas fee.

Native Rollup

Opside proposes a 3-layer scaling architecture, where layer 2 is an EVM-compatible and Rollup-friendly chain, and layer 3 is composed of different zk-rollups. The Opside chain makes many system-level optimizations for zk-rollup, and introduces the concept of native rollup.

[

image

1920×754 50 KB

](https://ethresear.ch/uploads/default/original/2X/4/45167fdb343665827a94c969bffcc54a206f974f.jpeg)

Native rollup is somewhat similar to the architectures of Polkadot and Starknet. Once a rollup registers a slot, the rollup becomes a native rollup. In contrast, the Polkadot architecture has the disadvantage that the consensus of the parachain depends on the assigned set of validators. There is a correlation between security and the number of validators If a few validators went offline, the parachains whose validator groups are too small to validate a block will skip those blocks, or even stop until the situation is resolved. Opside does not have this problem because layer 2 collects data from all native rollups in layer 3 and verifies all zk proofs. Opside chain is more secure and decentralized with all rollups on it as a tighter whole, sharing the same consensus layer.

[

1167×649 43.9 KB

](https://ethresear.ch/uploads/default/original/2X/5/5d25a43f5880df2b6f3012a15d1873a6dc112b3f.png)

Compared with the 3-layer architecture of Starknet, Opside's 3-layer architecture also has obvious advantages. First of all, Starknet's 3-layer architecture, does not address the issue of data availability. Because all L2 and L3 data still have to be uploaded to ethereum eventually, even with some compression, the data volume is still large and the cost is still high. One solution is to use validium as DA layer, which can greatly reduce the cost of data storage, but it will cause the loss of security. Opside's layer 2 is both an execution layer and DA layer, which can provide cheaper data storage than ethereum. Secondly, Starknet's layer 3 network consists of individual independent rollups. The operators (sequencer and prover) between these rollups are independent, and there is no mechanism for cross-rollup communication in the architecture. Opside's architecture solves these problems.

In short, Opside will provide a normalized interface at the system level, which brings many benefits:

- System-level consensus mechanism: native rollup implements the PoVP consensus we proposed earlier. Native rollup is naturally decentralized and permissionless.
- Cross-rollup communication: Opside provides a native cross-rollup communication mechanism for native rollup.
- Precompiled contracts: precompiled contracts will accelerate the verification of zk proof

1. Hybrid consensus of layer 2 & 3 : PoS + PoW

Layer 2 and layer 3 of the Opside architecture share a consensus mechanism with a hybrid of PoS and PoW.

- PoS: On layer 2, anyone can become a validator by staking, then have the opportunity to produce blocks of the

Opside chain. PoS is provable and validators periodically submit PoS proof to layer 1. Validators can get the block reward and staking reward for this part of PoS.

- PoW: The validators of layer 2 will not only produce Opside chain blocks, but also generate zk proof for each native rollup of layer 3 according to the rules of PoVP. Validators will get the IDE reward for successful generating zk proof, which is somewhat similar to PoW. Validators can get an extra bonus for that part of the reward by staking more tokens in system contract.

[

image

1930×1318 209 KB

](https://ethresear.ch/uploads/default/original/2X/4/474e41dccac082abd7899061666337a42cf9d4bf.png)

With the hybrid of PoS + PoW consensus mechanism, Opside unifies validator set of layer 2 and layer 3 to ensure the security, decentralization and permissonlessness of the whole system. More importantly, Opside provides a reliable decentralized solution for each Native Rollup in layer 3 at the system level. Opside defines a unified system contract, and rollup developers only need to implement these standardized interfaces and register rollup slots as native rollups, so that they can focus more on the implementation of business logic and do not need to care about how layer 2 and layer 3 interact at the underlying architecture level.

2. Cross-rollup communication

More significantly, Opside provides a native mechanism for cross-rollup communication for native rollups.

In traditional rollups, each rollup deploys a solidity contract on Ethereum, and the state tree of each rollup is maintained by its own solidity contract. Therefore, the state trees of each rollup are independent from each other, which causes fragmentation of users' assets. In addition, each rollup has its own cross-chain bridge, which is also very complex, slow and expensive to use. These bridges between rollups and Ethereum also rely on smart contracts to complete. In fact, most users now choose to use a third-party liquidity bridge to complete asset transfers between rollups, which can pose a great safety risk.

In Opside, after completing the registration of slots, native rollups share a world state tree with each other and the same global message queue. Therefore, native cross-rollup interoperability is possible in Opside. Imagine you want to lend USDC to a loan contract in Rollup A and then go to DEX in Rollup B to trade to buy BTC. In Opside you no longer need to withdraw assets from Rollup A to L1 and then recharge them from L1 to Rollup B. Instead, you can call the contract method of Rollup B directly in Rollup A. This will make the whole process much faster, cheaper and safer.

3. Precompiled contracts

Another benefit that can be gained by registering a rollup slot is the ability to useprecompiled contracts to speed up the verification of zk proof.

Specifically, the proof verification of a traditional zk-rollup is also a normal contract call transaction that relies on the solidity language to complete. In Opside, the proof verification of a native rollup is handed over to Opside's optimized precompiled contracts to perform. These precompiled contracts are implemented in the rust language and are deeply optimized for cryptography based on Opside's extensive experience in the zk domain.

In addition, it is important to note that the verification of zk proofs for all Native rollups is done by the same system contract. The system contract knows that the proofs of these rollups are not related to each other and are independent, so it will parallelize the computation of the verification. This will increase the verification speed by more than one order of magnitude.

Summary

Opside is a 3-layer scaling solution, with layer 2 being an EVM-compatible and rollup-friendly blockchain. In Opside's architecture, layer 2 and layer 3 are a more tightly integrated whole. The core concept of Opside is Native rollup.

- Layer 2 and layer 3 use the hybrid of PoS + PoW consensus mechanism, sharing a consensus layer and validator set. Under the interface specification of unified system contract, native rollup natively supports the decentralized mechanism of PoVP, and each node is permissionless and trustless, which also greatly solves the problem of MEV.
- Opside provides a native cross-rollup communication mechanism for native rollups. Users' assets can be managed in a unified manner, enabling cross-rollup interoperability.
- As rollups natively supported by Opside, Native rollups has system-level cryptographic optimization and parallel verification computation, resulting in faster proof verification.