

TL; DR

Opside's NCRC (Native Cross Rollup Communication) protocol offers a trustless solution for Rollup interoperability. The NCRC protocol doesn't involve adding an additional third-party bridge to each Rollup; instead, it transforms the native bridge of ZK-Rollup at the system level, allowing direct utilization of the native bridges of different ZK-Rollups for cross-Rollup communication. This approach is more streamlined and comprehensive, inheriting the absolute security of native bridges while avoiding the system complexity and trust costs associated with third-party bridges.

Opside has successfully implemented NCRC on the testnet. Anyone can now experience it on the official website at <https://pre-alpha-assetshub.opside.network/>.

Rollup Recognition Contract (RRC)

As of August 2023, several ZK-Rollups have gone live on the mainnet, including Polygon zkEVM, zkSync era, Linea, and more. However, these ZK-Rollups are independent and unrelated, leading to fragmentation of user assets. The fundamental reason for this issue lies in the fact that their contracts on L1 (Ethereum mainnet) are unrelated. They remain unaware of each other's existence and are unable to directly communicate through native Rollup bridges.

Therefore, the first step we need to take is deploying a specialized contract on L1 to enable Rollups to discover and recognize each other. This is referred to as the RRC (Rollup Recognition Contract). The RRC is responsible for managing all participating ZK-Rollups in the NCRC, including additions, pauses, and exits of Rollups. Each Rollup within the RRC is assigned a dedicated Rollup ID, while the ID for L1 remains fixed at 0.

When initiating cross-Rollup transactions through the native bridge on a Rollup, addresses can specify the target Rollup ID:

- If the Rollup ID is 0, it signifies crossing the message to L1, such as withdrawal.
- If the Rollup ID is not 0, it indicates sending the message to another Rollup.

Opside will deploy an RRC contract on every L1 layer and allow corresponding ZK-Rollups to join or exit without permission. This RRC contract will be used to maintain information for each Rollup ID, including the bridge contract address on L1. It's important to note that the RRC contract solely provides data retrieval services and does not directly interact with cross-chain assets.

Compatibility with Native Bridge Smart Contracts and Bridge Services

Generally, Rollup's native bridge is divided into three components: the bridge contract on L1, the bridge contract on L2, and a bridge service responsible for message relay. The NCRC protocol leverages these components at the underlying level and adds higher-level encapsulation. The main modifications are as follows:

- Bridge contract on L2: While preserving the original methods, a new method named `bridgeAsset` is added. This method allows users to specify the target Rollup's ID in the `destinationNetwork` parameter.
- Bridge contract on L1: A new method is encapsulated to handle the cross-chain messages of the new `bridgeAsset` method. The bridge contract, based on the Rollup ID found in the RRC contract, locates the information of the target Rollup and transfers the cross-chain assets to the bridge contract of the target Rollup. The cross-chain assets are deposited into the target Rollup there.
- Bridge service: Responsible for message relay and charges users fees for cross-Rollup transactions.

Once a Rollup completes the NCRC-related compatibility adaptation mentioned above, it can register with the RRC to join the native cross-Rollup communication network.

Process of Native Cross-Rollup Transactions

For users, the operation of NCRC is entirely consistent with that of Rollup's native bridge. Initiating a cross-Rollup transaction from Rollup1 to Rollup2 is an automated process, including the following steps:

1. The initiator, User1, on Rollup1, invokes the `bridgeAsset` method of the native bridge to initiate the cross-chain transaction. The `destinationNetwork` parameter in this transaction is set to the Rollup ID of Rollup2. This Rollup ID will be used to retrieve the corresponding L1 bridge contract address. If the Rollup ID is 0, it signifies the target network as L1.
2. Subsequently, this transaction is packaged by Sequencer1 of Rollup1. The initiator, User1, bears the cost of the cross-Rollup transaction, paying it to Sequencer1 on Rollup1. Rollup1's Bridge service then transfers the cross-chain asset to the Rollup1 bridge contract on L1. At this point, both Rollup1 and L1 complete the burn and release operations of the asset.
3. To complete the cross-Rollup asset transfer, Rollup1's Bridge service queries the RRC contract to retrieve information about the target Rollup2 corresponding to the `destinationNetwork` parameter. This information provides the L1 bridge

contract address of Rollup2. Then, the bridge contract of Rollup2 takes control of these assets and maps them to Rollup2 through the bridgeAsset method.

4. Finally, once the transaction is successfully packaged and the proof is generated, Rollup2's Bridge service executes the claimAsset operation. Consequently, the cross-chain assets initiated by Rollup1 safely arrive at the designated address on Rollup2.

It's worth mentioning that throughout the cross-chain process, the user's assets flow through the following path: Rollup1 → Rollup1's L1 bridge contract → Rollup2's L1 bridge contract → Rollup2. In other words, the user's assets do not go through any third-party protocol; they leverage Rollup's native bridge. The entire process is secure and trustless

.

[

1193×1193 35 KB

](<https://ethresear.ch/uploads/default/original/2X/c/c198dfa7ab322cc20e0f3e29c36871437d3348ee.png>)

When users execute cross-chain operations on Rollup1, selecting Rollup2 as the destination, the technical process actually involves three entities: Rollup1, L1, and Rollup2. However, users do not need to be aware of the existence of L1 in this process; their experience is simply a direct cross from Rollup1 to Rollup2. The underlying reality is that cross-chain assets undergo two bridging operations on L1, creating a seamless connection from Rollup1 to Rollup2 in the user's perception. During this process, operations on L1 are handled automatically, and users do not need to perform any additional actions. From the user's perspective, their current Rollup can perform cross-chain operations to both L1 and any other Rollup. This design enhances user experience fluidity while concealing underlying complexities.

NCRC is now live on the Opside testnet!

Opside has successfully implemented native cross rollup communication on the testnet. Anyone can now experience it on the official website at <https://pre-alpha-assetshub.opside.network/>. We also welcome users and developers to help us identify potential bugs and security risks and provide any valuable suggestions.

We believe that trustless native cross rollup communication will not only securely share liquidity across all Rollups but also provide robust multi-Rollup interoperability, opening up new possibilities for decentralized applications and DeFi protocols.