rStake 1.0

Shubham Singh, Vatsal Gupta
shubham@routerprotocol.com, vatsal@routerprotocol.com

Abstract

This paper introduces rStake, a powerful primitive built on Router's Cross-chain Intent Framework (CCIF) that unlocks seamless, cross-chain liquid staking possibilities. By leveraging CCIF's modular architecture, rStake empowers developers to integrate diverse liquid staking protocols – Stader, Lido, Ankr, Swell, Metapool, Benqi, and countless others – within their applications, thereby enabling users to effortlessly stake their assets across chains. rStake is one of the earliest primitives built using Router's intent framework tooling, paving the way for a future where developer creativity fuels diverse and innovative blockchain applications.

1 Background

Liquid staking empowers users to earn staking rewards while retaining their asset liquidity. However, within the current ecosystem, liquid staking is viable only for same-chain users, i.e., only users with funds on Ethereum can stake on protocols like Lido, Swell, while only users with funds on Avalanche are able to stake on Benqi. For users with funds on other chains, liquid staking on these protocols is highly inefficient. Users face challenges like navigating multiple platforms, engaging in multiple transactions, huge wait times, high gas costs, among others.

Additionally, there is 5M+ ETH in circulation on L2s like Arbitrum, Optimism, Base, among others, with no viable liquid staking venue. If any user on these L2s wanted to liquid stake on Ethereum within the current setup, it would typically involve around 27 clicks, interaction with 3 different dApps, 6 wallet interactions, and up to 15 minutes of effort. To accomplish a task like staking on Lido, users must be familiar with multiple blockchain networks, possess their respective gas assets, or know the processes for acquiring these assets. Furthermore, a comprehensive understanding of bridges for asset transfer and proficiency in navigating decentralized exchanges or aggregators is imperative. This intricacy and time investment significantly impede user experience and accessibility within the blockchain ecosystem.

2 Introducing rStake

Powered by Router's Cross-chain Intent Framework (CCIF), rStake is a liquid staking primitive that enables developers to build seamless cross-chain liquid staking applications with ease.

2.1 Components

2.1.1 Router Multicall Adapter

The Multicall adapter is a smart contract that can invoke multiple intent adapters. Using Multicall, developers can build applications spanning multiple intent primitives - for example, using rStake Stader adapter to stake into Stader and then using a Lending adapter to lend the resultant ETHx on a lending platform.

2.1.2 rStake Intent Adapters

rStake adapters are specialized smart contracts that connect rStake to individual liquid staking protocols. For example, a Lido intent adapter can be used to stake user funds into Lido. Here are the two functions that handle a liquid staking request:

1. `execute`: This function is present in every rStake adapter and is expected to handle the data received from the Multicall adapter. Within this function, the adapter decodes the data received, receives the funds and calls the `_stake` function.

2. `_stake`: This function is responsible for calling the designated liquid staking pool, staking the asset and transferring the liquid staking token back to the user/recipient.

2.1.3 rStake Intent Solver

The rStake solver is a crucial part of the rStake framework that allows rStake to find the best route for any liquid staking request.

- **Path Discovery:** rStake solver explores all the potential execution routes for a chosen route, like staking on a specific chain.
- **Multi-Criteria Decision Making:** It analyzes and selects the optimal path using advanced algorithms, considering factors like fees, estimated time, and security.
- **Calldata Generation:** Generates transaction data (call data) for each adapter involved in a chosen path, ensuring smooth interactions across different protocols.
- **Future-proof Design:** Built for expandability, rStake solver can easily be configured to index other intent adapters. For example, the rStake Solver can be configured to generate paths for a transaction that involves liquid staking and lending of the liquid staked asset on a lending/borrowing protocol.

2.1.4 Router Nitro

A trustless, ultra-low latency bridge that is used by rStake to transfer user funds across chains. Thanks to Nitro's low latency bridging and asset + message transfer capability, rStake can resolve cross-chain staking requests in less than 1 minute.

2.2 Features

2.2.1 Improved DeFi Accessibility

By enabling seamless cross-chain transactions and interactions, rStake can help in consolidating the currently fragmented ecosystem and facilitate the adoption and development of cross-chain solutions.

2.2.2 Simplified Dapp Development

Building new decentralized applications presents significant complexity and technical barriers, particularly for new developers. To address this, rStake offers streamlined tools (rStake adapters and solver) for creating derived applications.

2.2.3 Composable

By leveraging existing rStake adapters, developers can design applications with custom staking flows and functionalities unique to their users' needs.

2.2.4 Permissionless

Any developer can create adapters for new and emerging liquid staking protocols, expanding the rStake ecosystem and user benefits.

2.3 Leveraging rStake within a User-facing Application

As mentioned in the previous section, any developer can use rStake to generate cross-chain applications with varying use-cases. rStake abstracts things to such a level, that to build a user-facing application, all a developer needs to do is deploy a UI and invoke the rStake solver from that UI. Depending on the use-case, some modifications to the solver might be required (in case you want to add lending on top of liquid staking or add any other feature), but rStake will provide all the essential facets of the application to you.

2.3.1 Workflow

Let's take a look at how the flow will look like for the users of your cross-chain liquid staking application. Suppose a user with USDC on Polygon wants to stake on Benqi (Avalanche) using your platform.

Step 1) User deposits USDC on Polygon.

Step 2) Your custom UI queries the rStake solver with the user's input

Step 3) Based on the user's source asset and desired liquid staking platform, the rStake solver identifies the most optimal route for executing the user's request. The pathfinder generates the execution calldata and sends it to your UI along with the entry point contract.

Step 4) Your custom UI invokes the entry point contract, which in this case is the Router Nitro bridge contract.

Step 5) Router Nitro converts USDC to AVAX and transfers it to Avalanche along with the instructions to stake on Benqi.

Step 6) Upon receiving the instructions, the Multicall adapter delegates AVAX staking to the Benqi liquid staking adapter.

Step 7) Benqi liquid staking adapter stakes AVAX on Benqi's contracts.

Step 8) Benqi sends sAVAX to the Multicall adapter.

Step 9) The Mutlicall adapter provides sAVAX to the user on Avalanche.

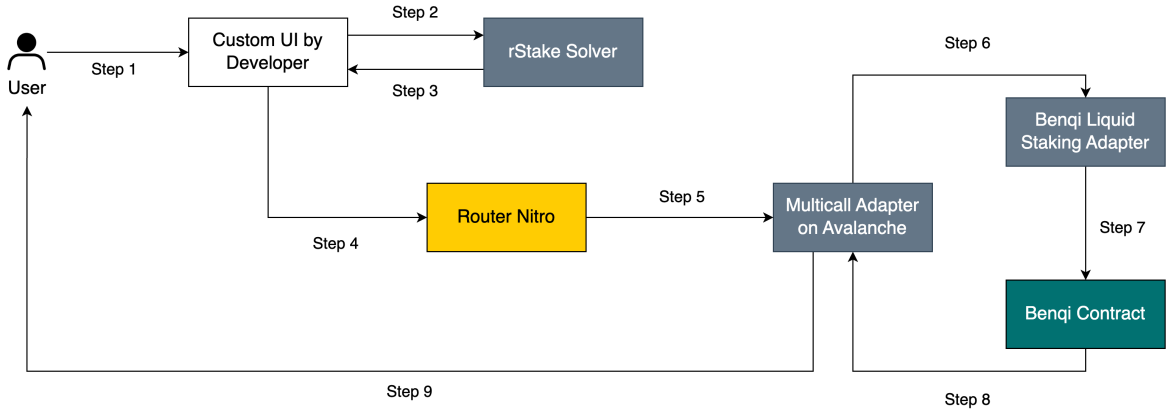

Figure 1: Application Workflow with rStake

3 Security

All the rStake adapters are designed to be stateless, meaning they never hold user funds at any point during the execution process. This minimizes the attack surface and eliminates the risk of funds being compromised within the rStake infrastructure. Furthermore, for unparalleled transparency and trust, all rStake smart contracts, including the Multicall and individual intent adapters, have undergone rigorous security audits. This commitment to secure coding practices helps rStake maintain the highest security standards.

4 Future Work

4.1 Optimizing rStake Solvers

Continuously testing and analyzing the performance of the current multi-criteria decision making algorithm is essential. Identifying areas of inefficiency or inaccuracy can lead to the development of more advanced algorithms, ensuring more optimal path selection for user intents.

4.2 Decentralization of rStake Solvers

The centralization of rStake intent solver systems presents risks like single points of failure and reduced resilience. In the future, we will implement a peer-to-peer (P2P) network for Solver systems that can significantly improve fault tolerance. This network would distribute both the processing load and decision-making, thus enhancing the system's resilience against single-point failures and central control issues, leading to a more robust and reliable framework.

4.3 Non-EVM Adapters

Currently, rStake primarily supports EVM intent adapters out-of-the-box (along with a few Near adapters). In the next few months, we'll develop and add adapters for Cosmos chains and Move-based chains.

4.4 Generalized Solver

We plan to develop a generalized solver that can seamlessly integrate with both EVM (Ethereum Virtual Machine) and non-EVM based adapters. This would significantly broaden the primitive's applicability and interoperability across diverse blockchain ecosystems.

Conclusion

rStake is poised to revolutionize the liquid staking landscape, offering unprecedented flexibility, accessibility, and control. By harnessing the power of Router's CCIF and its open-source adapter toolkit, rStake empowers users and developers to unlock the full potential of cross-chain liquid staking. This is just the beginning of a transformative journey – a journey where user intent drives the execution of powerful blockchain-based actions, ushering in a new era of innovation and user empowerment.