Introduction

Below is a post in response to PERCH: Protocol Evaluation and Request Coordination Hub. Commit-Boost has previously been introduced to the Lido Community (recommended reading if you are not familiar) and we submitted a grant proposal for funding from the Lido Community. Thank you for this opportunity to engage with the Lido Community.

Over the last few months, Commit-Boost has been in development. We are targeting production-ready + audit by the end of October and have been engaged with multiple operators (both that are part of the Lido operator set and outside of the Lido operator set) for testing. We are asking more teams to test for feedback and to ensure Commit-Boost has been designed without unknown limitations across Lido's NOs. This request is scoped to Commit-Boost + the PBS module and follow-on requests for other module testing will be posted on the Lido Community Research Forum.

Below is the information requested in the PERCH post. If more details are needed or other formats are preferred, please let us know.

Details of Request for Testing

- Number of Participants:

We request at least 10 different NOs with a total of 50,000 validators in Holesky to start testing Commit-Boost and the PBS module

- Application Form:

Link here

Testing Outline:

- Software and/or protocols to be used:

Commit-Boost is the software with the repo here, documentation on testing here, and documentation to learn more can be found here. Please start by testing just the PBS module

- Networks:

Holesky

- Duration:

A few weeks

- Expected Impact:

Teams will need to change their / add a sidecar to be Commit-Boost as outlined in the documentation. Please note we also have tools / support different configurations, e.g. docker, native binaries, and k8s to help streamline deployment. The current testing should allow for backward compatibility (i.e. same / better functionality than just running MEV-Boost) and give NOs more modularity / insight to reduce risks and improve their operations

Requirements:

- Alignment with Ethereum Roadmap:

Commit-Boost directly aligns with Etheruem's roadmap with modules that are being created around inclusion lists (censorship resistance) and preconfirmations (scaling). We have also designed Commit-Boost to reduce fragmentation risks to Ethereum and its operator set and not limit who can participate and build Validator Services

- Neutrality and Inclusivity:

Commit-Boost is designed to not limit any innovation around Validator Services and is built to decrease risks for Ethereum and improve safety mechanisms for Lido / Ethereum's NOs

- Adherence to Community Standards:

We have started holding Community calls to hear any concerns / drive consensus around anything contentious and have been working with both validators and module creators to design Commit-Boost from day one. We have also been engaged with the broader Ethereum community for research, design, feedback, and testing around any standards we may support

- Security Best Practices:

This project is open source with the code being broadly engaged by the community with many eyes and different perspectives across teams during the research, design, and feedback / testing phase. We are also getting Commit-Boost + PBS audited by one of the top audit firms in the space

- Complementary Use Cases:

Commit-Boost is designed to be one sidecar with the opportunity for validators to opt into many commitments. We expect things like preconfirmations and inclusion lists to flourish and many other commitments / validator services to compliment each other

Please let us know if there are any questions or concerns or anything we can help with and we appreciate the NOs of the Lido Community helping to test Commit-Boost.