BOLT: MEV-boost Compatible Proposer-Commitments Enabling Trustless Preconfirmations

Proposal

- Bolt allows proposers to make credible commitments,

starting from preconfirmations. This new feature enhances network censorship resistance and user experience, aiming to increase Ethereum usage.

- Bolt will be open-source

and PBS compatible

, with permissionless access on both the demand and supply side. Most importantly, it will be trustless

, primarily relying on staked

capital as economic collateral for commitments.

- Bolt protocol is built by Chainbound, a research and development organization

specializing in optimized infrastructure and networking solutions on Ethereum.

- Chainbound is proposing to consider Bolt for the Lido Alliance.

The team believes that the software aligns with Lido's values and requirements. In addition, the Alliance will support the bootstrapping stages of the new primitive enabled by Bolt, retaining the vision and alignment with Ethereum, Lido, and the community.

Protocol Overview

Bolt Protocol enables Ethereum block proposers to provide credible commitments about their block contents. The system aims to improve Ethereum's UX and censorship resistance by unlocking new primitives such as preconfirmations and inclusion lists. Ultimately, this will increase block-space value and yield for stakers, resulting in a more resilient Ethereum.

Bolt is implemented out-of-protocol, leveraging staked collateral for economic assurances. It is compatible with the existing block production pipeline, making it easy to integrate with existing systems.

Bolt will come to market via the phased approach outlined below:

[

Untitled (1)

1488×684 65.2 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/f/f8a2f87fa2c177aa44595c5dabb5517460928464.png)

Bolt Design Principles:

1. Trust-minimized

: No new trusted entities are introduced. Commitments are backed by economic assurances, not by trusted intermediaries.

1. Credible

: Because Bolt is proposer-centric, they can be fully held accountable for their commitments. In case of breaches, proposers can be penalized, which ensures the credibility of the commitments.

1. Permissionless

: Any proposer can opt-in to the protocol, and any user can request commitments from them. No central authority is needed.

1. Compatible

: Bolt is designed to be compatible with the existing PBS pipeline, and eventually ePBS. From the proposer's perspective, the only change needed is an additional sidecar.

Guided by these principles, we believe Bolt is an effective preconf solution for Lido Node Operators and Ethereum as a whole.

For Ethereum:

1. Bolt accelerates Ethereum’s roadmap

towards stronger censorship resistance properties (Inclusion Lists, PEPC), de-fragmentation (based sequencing), and fast UX (preconfs), without relying on trusted intermediaries.

1. Bolt’s permissionless nature makes it unopinionated

in the current relay and builder market competition. Having an opinionated solution that could potentially favor specific relays or builders is fundamentally unhealthy for Ethereum.

1. Bolt’s proposer-centric credibility

ensures Ethereum does not increase its reliance on trusted entities within the block-production pipeline. This is important so as not to add to existing centralizing pressures.

For Lido Node Operators:

1. Bolt is economical and easy to integrate for node operators.

Bolt is implemented out-of-protocol and is compatible with the existing block production pipeline.

1. Bolt is expressive and future-proof.

Bolt is designed to cover multiple block-space use-cases, while also remaining compatible with the potential transition towards ePBS.

1. Bolt is fault-tolerant.

Bolt has an embedded fallback block building mechanism as a last resort to ensure that relays and builders aren’t critical to the proposers’ operation.

1. Validators do not need to rely on trusted relationships

to make commitments to further monetize their blockspace.

Bolt V1

Bolt V1 will support inclusion preconfirmations. Inclusion preconfirmations are commitments about the inclusion in a certain slot.

Architecture

By default, the software stack for proposers will be extended with a new component called bolt-sidecar

that implements the default builder-API

. The bolt-sidecar

will serve like a proxy for the modified mev-boost

client, which implements the constraints-API

. Users interact with the bolt-sidecar

, turning commitments into constraints and communicating them to the PBS pipeline through the constraints-API

Component Table

Component

Placement

Description

Bolt RPC

Off-chain

RPC proxy server that propagates preconfirmation requests to opted-in proposers in the lookahead

Bolt sidecar

Off-chain

The entrypoint for Bolt. Implements the commitments-API

and turns commitments into constraints

MEV-Boost

Off-chain

Modified MEV-Boost client that implements the constraints-API

and verifies constraint proofs

Registry

On-chain

The registry smart contract that keeps track of the opted-in proposer set and their associated stake

Proposer Software Stack

bolt-sidecar

The Bolt sidecar is the off-chain entrypoint for Bolt. To enable it, proposers must point their beacon node builder-api

to the sidecar’s endpoint. From the perspective of the beacon node, the sidecar will look like a regular external block builder.

The sidecar is responsible for the following tasks:

- Receiving & validating commitment requests from users through the commitments-API
- Turning commitments into constraints and communicating them downstream
- Proposing a safe fallback block in case of any faults

Modified mev-boost

The modified mev-boost

client will implement the constraints-API

and verify those constraints against incoming builder bids, which will have proofs attached. For more information on how proofs and verification will work, see the proofs section.

Registry

The Bolt Registry is the smart contract that keeps track of opted-in proposers and their associated stake. The responsibilities of the registry include:

- Registering new proposers via an EOA, verifying their authenticity cryptographically (see the unfinalizedopt-in procedure below)
- Accounting for the proposer’s stake/collateral, potentially including restaked capital
- Removing proposers that no longer wish to participate in Bolt (with a cooldown period)
- Providing read access to the proposer set (useful for the Bolt RPC and Challenger components)

Bolt RPC

The Bolt RPC will be a public RPC endpoint that will proxy requests from users to opted in proposers according to the lookahead and provide additional functionality like DoS protection and rate limiting. The Bolt RPC will be a separate process from the proposer, and anyone can run one. Proposers can configure the Bolt sidecar to point to their preferred Bolt RPC

Please refer to our technical documentation for more details about Bolt’s architecture, commitment types, use cases, and more!

Delegation

Bolt's architecture allows proposers to delegate the task of providing commitments to a trusted third party. This can be achieved by signing a permit message that allows the third party to submit commitments on their behalf. They then have to point their builder-API

to the third party's bolt-sidecar

endpoint.

We defer to individual validators, larger node operators, and staking pools on their own policies towards delegation. For example, the allowance of delegation can be addressed within Lido's Ethereum Block Proposer Reward Policy. It should be noted that delegation leads to a fully trusted relationship. If the operator commits a fault, the proposerr will get penalized.

Onboarding Process

The onboarding process requires proposers to opt-into the Bolt protocol. This has two parts:

1. Off-chain Validator signature Authentication

Validators need a signature to validate preconfirmation requests. The exact authentication method is still an open research topic, but Bolt plans to support a variety of signing methods, such as the Commit Boost signing manager.

Proposers, and their respective key managers, will interact with a commit-boost's signing manager and proxy key scheme to both authorize respective validators and authenticate commitments.

1. Opt-in Procedure

Once registered, proposers must post some form of collateral to guarantee economic credibility behind their preconfirmations. The specific method in which this is achieved is left as an open point for now.

We are actively looking for validators to take part in Cohort 1 for testing and demoing Bolt. We believe Lido node operators would make ideal partners here. Please reach out if you would like to participate.

Side-note: An experimental (not part of the poc) on-chain registration process could work as follows:

1. The Proposer signs a message with their withdrawal address private key to signify that they are requesting to opt-in. The message will need to contain the Ethereum address that the proposer intends to use as signer to authenticate individual preconfirmation requests. This way, proposers in Bolt will have a separate identity from their validators private key.

2. The Proposer sends a transaction to the Registry, requesting to opt-in to Bolt. The transaction must be sent from the same address that was specified in the signed message, which must be passed in the transaction's data. This way, the ownership of this new ECDSA key-pair is proven on-chain.

Security Culture

The Bolt protocol will be open-sourced and audited allowing for rigorous internal and external examination of all Bolt components. Bolt is designed to be simple and minimal. Along with a modified mev-boost client, Bolt only introduces 2 additional off-chain components: a sidecar and an RPC.

Chainbound will collaborate with external parties for both on-chain and off-chain components. For example, Bolt can utilize existing battle-tested restaking modules.

Additionally, Bolt's interface and standards will be a collaborative output from many teams thinking about similar solutions.

Chainbound will provide maintenance, monitoring and, analytics for continued transparency, integrity, and security of Bolt.

Lido Alliance <> Chainbound

Lido's reGOOSE states:

Internalize validator services that align with Ethereum's roadmap, starting with preconfirmations, without exposing stakers to additional risk

Bolt is not only: (1) a generalized proposer commitments protocol that can facilitate a variety of potential validator services, but also (2) designed with preconfirmations as the primary use case, and (3) aligned with Ethereum's roadmap towards stronger censorship resistance guarantees, better UX, and defragmentation.

Bolt does, however, rely on economic security, placing additional slashing risk on proposers. We find this path necessary when introducing proposer-centric commitments so as not to rely on trusted intermediaries. With that said, there are ways to either reduce proposer risk (which can come in the form of insurance funds) or increase rewards (yield) to compensate for said risk.

Lido Alliance Support for Chainbound

Acceptance into the Lido alliance would be a direct endorsement for Chainbound and Bolt. With this endorsement, Chainbound would appreciate the convergence from Lido stakeholders around Bolt as one of the preconfirmation solutions to test and iterate over. For the Lido working group, Chainbound would appreciate direct collaboration and feedback on Bolt's design choices, suggestions around auditors and firms, and continuous testing of the software.

Helping bootstrap Validator Cohort 1, in particular with Lido Node Operators, would be an incredible help, along with ongoing assistance around validator adoption down the line.

Lido Alliance Alignment Collateral

We at Chainbound are open to discussing the possibilities here.

Towards a Proposer-centric Future

We hope the design decisions around our protocol reflect our desire to support and grow Ethereum, and therefore, help support Lido, a key steward of the Ethereum community. Proposer agency and trustless proposer commitments are core towards making Ethereum more censorship-resistant and usable.

We are excited for Ethereum and Lido's future and hope Bolt can play a part in that!