

As always, humbled by the Ethereum community's willingness to review / provide feedback. Thank you [Barnabe](#), [Chris](#), [Conor](#), [Ellie](#), [Jason](#), [Jonas](#), [Justin](#), [Julian](#), [Kubi](#), [Kydo](#), [Mike](#), [Pascal](#), and [Sam](#) for the feedback and review.

Introduction:

Over the last year, proposer commitments have become more widely discussed. I personally became proposer-commitment-pilled on the back of the [based sequencing](#) / [preconf](#) posts and [weekly calls](#) being held by Justin as well as discussions and [research](#) about blockspace futures. With this interest came the dive down the rabbit hole and as always within the Ethereum community, I found many others already there or willing to join. This post I hope contributes to the ideas already out there and continues to push the discussion around proposer commitments and engagement across the community.

Below we introduce a concept focused on standardizing the last mile of communication between a proposer and a third party and how proposers may register and make / receive commitments. We see proposer commitments as another promising evolution of Ethereum's original core vision that will expand the infinite garden to not just be THE marketplace for credible blockspace, but also THE marketplace for credible proposer commitments. We outline some background and motivation, design principles, an initial high-level design, and some open questions. We plan to continue to expand on this with more detailed specs as we gather more feedback and input!

[

750×500 54.1 KB

](<https://ethresear.ch/uploads/default/original/3X/f/9/f9246140c4556cd087cf9de5c5a0f5e363ada4b4.jpeg>)

TL;DR:

- Proposer commitments have been an important part of Ethereum's history and could continue to be a powerful unlock for Ethereum
- The potential impact of proposer commitments are best captured in a quote from Barnabe's [recent post](#); the "... proposer creates the specs, or the template, by which the resulting block must be created, and the builders engaged by the proposer are tasked with delivering the block according to its specifications"
- Over the last year, there have been multiple ideas around proposer commitments. For instance, even in a short period, we have seen multiple implementations of proposer commitments related to precons
- While powerful, if multiple standards arise around how proposers register and make / receive commitments, we run the risk of fragmentation that could increase risks to Ethereum
- We present a potential design referred to as "Commitment Boost"[1] inspired by research around PEPC, EigenLayer, and the broader Ethereum community
- This envisions leveraging existing pipes to allow proposers to register and make / receive commitments and remain fully backward compatible
- The principles we embrace for this require open source / development in the open, contemplate modularity and self-containment for safety, integrate a robust suite of testing / off-the-shelf alerts / data APIs, and be performant / efficient
- We end with a few items around risks / additional questions we want to engage with the community on

Last, we want to note that this post is focused on the last mile of communication to allow proposers to register / send / receive commitments. We do not discuss how the proposer commitment protocols may work or how they may be enforced.

Related Work:

- [Unbundling PBS](#)
- [PEPC FAQ](#)
- [PEPC-Boost](#)
- [PEPC-DVT](#)
- [Preconfirmation Gateway](#)
- [Based preconfirms](#)

[Presentation](#) on precons and [slides](#)

- [More Pictures about Proposers and Builders](#)
- [Preconfirm protocol](#)

- [Chain Neutrality and Uncrowdable Inclusion Lists](#)
- [Grounded Relay](#)

Background:

Nearly half a decade ago, [Flashboys 2.0](#) was published highlighting how arbitrage bots were challenging the promise of blockchains. On the back of this, some of the authors and community members started a research collective to provide solutions to tackle these challenges. In the end, these efforts created a product more broadly known as [MEV-boost](#).

MEV-boost is a middleware that allows the proposer to make a wholesale commitment that outsources block building to a sophisticated actor called a block builder. In return, these block builders pack a block to earn the highest fee for the proposer. Today, [over 90% of blocks](#) are built with proposer commitments.

Proposer Commitments:

On the back of a few developments[2] and some research by the EF[3], the concept of proposer commitments has begun to be more commonly discussed spurring the question; could proposers make commitments that would unlock a significant design space for Ethereum? And, could this be a mechanism to allow validators "...to provide input into block production, even when they decide to delegate building." [4] In the last year, multiple proposals have come forward that rely on or could greatly benefit from proposer commitments, some examples include:

- Inclusion lists: Proposer commitment where part of the block / a set of transactions will be included / can't be censored or removed by a third party, including the proposer
- Preconf: Proposer commitment to in advance, guarantee inclusion of data / certain transaction or group of transactions in a block
- Partial block auctions: Proposer commitment to auction off the top-of-block and the rest-of-block
- Blockspace / blob futures: Proposer commitment to sell part of their block now, but deliver that part of the block in the future

The proposals range in complexity but are underpinned by the same simple idea—a proposer's commitment to do something with or for a third party. We also note that proposers may not need to make commitments at this level of granularity (i.e., continue to use wholesale block auctions). However, we believe this is an avenue worth exploring as it may help preserve things like chain neutrality "by allowing them to provide input into block production, even when they decide to delegate building" [4] and if they choose, give some autonomy back to the proposer.

Challenge:

On the surface, this all seems great and is an incredibly exciting development. But, in the undercurrents, we are potentially on a perilous path if we can't agree on a standard of how proposers register and make / receive commitments. We see multiple risks including, but not limited to:

- Increased fragmentation: While diversity of standards can create unlock more innovation, multiple standards (particularly in the last mile of communication) could compromise the security integrity of the entire Ethereum network through fragmentation of how proposer commitment protocols speak to proposers (i.e., proposers may need to make client adjustments for each variation of proposer commitments)
- Development complexity: If there is no standard, teams may more commonly make client adjustments to opt into proposer commitments. This could exponentially inflate the burden on core developers tasked with executing / testing major network upgrades increasing risks for the network around hard forks
- Limited transparency: With multiple software and standards, transparency around what proposers are opting into as well as bugs and taking quick actions may be challenging when something does go wrong

These risks are likely to only increase as more and more proposer commitments get proposed and adopted. We also note that longer-term there are potential ideas to enshrine various mechanisms helping to reduce these risks.

Proposal:

We propose an out-of-protocol, open-source public good that is backward compatible to help standardize the communication mechanism for the last mile of proposers' commitments. The goal is to develop, adopt, and then sustain one standard software that will limit fragmentation and reduce complexity for core devs. We currently call this Commitment Boost.

Design Principles:

Below are a few design principles when initially envisioning Commitment Boost.

- Open source / open development: This should be developed in the open and under open source licensing such as MIT / Apache-2.0

- Safety and reducing risks: Should be backwards compatible not changing existing pipes that support current proposer commitments and be built to isolate each proposer commitment. We also note that Commitment Boost should be continuously managed for future forks / upgrades in the Ethereum ecosystem
- Same overhead as today: The software will not have more overhead to run than existing proposer commitments (and ideally it is even more efficient!). Note: there may be additional overhead for facilitating the actual proposer commitments, but it is not the focus of this discussion and will be up to the proposer commitment protocol itself around how to potentially outsource any complexity
- Transparency: There will need to be robust functionality to understand which commitments a proposer is opting into as well as performing rigorous testing to quickly identify bugs. There should also be data APIs to increase information and transparency for the community and to strengthen alerting systems when bugs happen

Initial High-Level Potential Design of Commitment Boost:

- Proposer wishing to register for commitments runs Commitment Boost. This will be backward compatible with consensus clients using the same messaging mechanisms that exist today. Commitment Boost is just focused on standardizing the last mile communication between a proposer choosing to register and then send / receive messages to / from a proposer commitment protocol
- Once running Commitment Boost, the proposer will need to register for each commitment they wish to make
- Each proposer commitment across the Commitment Boost stack is likely to be modular and isolated. The rationale that if there is a bug in one module this will not impact the rest of the proposer commitments / block construction. We also note that safeguards should be put in place to protect consensus (i.e., fallback mechanisms to local block building etc.)

[

1018x592 164 KB

](<https://ethresear.ch/uploads/default/original/3X/4/3/43f7a19652088da9f139f178060df90293cebbf0.png>)

Near-Term Focus:

As noted above, Commitment Boost should lean towards being modular to potentially allow any proposer commitment. However, we initially plan to focus on designing standards for Commitment Boost to be backward compatible and to support commitments such as preconf protocols and inclusion lists. If other projects require proposer commitments and are interested in thinking through designs please reach out.

Open Questions:

Below is a list of questions that we need to consider and engage around. We note that some of these are not specific to Commitment Boost, but proposer commitments more broadly.

- Added risks: While there are general risks and questions on enabling proposer commitment protocols, we are particularly interested to engage on additional risks / the initial high-level design of Commitment Boost and whether the design can reduce risks to any commitment potentially impacting consensus
- Standardization is not always good: While in most cases standardization helps align the market and reduces the risk of fragmentation, it can also limit innovation and create “tech debt” that can confine the design space given how early we are and what we know about proposer commitments today
- How modules are added: It is not clear the best path forward around a process, or lack of, for how new proposer commitment modules are added
- Coordination during Ethereum upgrades: Likely will need to coordinate a group to ensure there is a process to perform testing / changes related to forks
- Upgrades to commitment boost: Similar to the point directly above, if we identify some upgrades needed we will need to test and manage any code changes required
- Economics: Are there any considerations around how proposers are paid for committing and how this could impact Ethereum / could these be internalized to Ethereum in the future
- Centralization: What are the impacts Commitment Boost could have on centralization and does it impact at-home stakers or geographical dispersion of validators

Conclusion:

The garden of Ethereum is infinite and the potential for proposer commitments could set off a wave of ways to use Ethereum. Proposer commitments could enable things like precons that are critical for based sequencing as well as other

applications not currently envisioned. To help the community and entrepreneurs build on Ethereum, Commitment Boost is an initial idea to standardize the way proposers register and send / receive commitments. We look forward to discussing, refining, developing, and working with the community on proposer commitments. We plan to keep gathering feedback on the back of this post and continue to work with the community to push this idea as well as assist in other efforts around proposer commitments.

References:

[1] Fun Fact: The initials for Commitment Boost are shared with [CB Radios](#), “a system allowing short-distance one-to-many bidirectional voice communication among individuals.”

[2] With the excitement around [based sequencing](#) introduced by Justin Drake, there has been a push [by a few teams](#) to build proposer commitments such as precons. Developments like EigenLayer and restaking generally have also expanded developers / teams imagination of what proposers can make commitments around.

[3] Generally, we are referring to PEPC and inclusion lists research as noted in the “Related Work” section of this post.

[4] [Uncrowdable Inclusion Lists: The Tension between Chain Neutrality, Preconfirmations and Proposer Commitments](#)