Relays in a post-ePBS world

[

upload_a79424d246052b2d9f49dc0d2f0821c7

812×656 276 KB

](https://ethresear.ch/uploads/default/original/2X/5/54056db6fa3fd684f1cd3aa6dd41307ceb71968d.png)

\cdot

by mike, jon, hasu, tomasz, chris, & toni

based on discussions with justin, caspar, & stokes

august 4, 2023

\cdot

tl;dr;

Continued ePBS research and the evolving mev-boost

landscape have made it clear that the incentive to use relays will likely remain even if we enshrine a PBS mechanism. This document describes the exact services that relays offer today and how they could change under ePBS.

Post enshrinement, the protocol would serve as a default “neutral relay” while the out-of-protocol relay market continues to develop, offering potential latency optimizations and other ancillary services (e.g., bid cancellations and more flexible payments). This greatly reduces the current dependency on public goods relays.

We also present a new in-protocol unconditional payment design proposed by Caspar and Justin, which we call Top-of-Block (abbr. ToB) Payments. This modification simplifies ePBS meaningfully and further reduces the scope of services that require relays.

Although removing relays has often been cited as the raison d’être for enshrinement, we believe ePBS is still highly beneficial even if relays persist in some (reduced) form. The primary tradeoff is the added protocol complexity.

\cdot

Contents

(1) Why enshrine PBS?

revisits the original question and sets the stage for why we expect the relay market to exist post-ePBS.

(2) Relay roles today

presents the current relay functionality.

(3) A simple ePBS instantiation

outlines the core primitives needed for ePBS and introduces Top-of-Block payments.

(4) Relay role evolution post-ePBS

revisits (2) and presents the advantages that future relays may have over the enshrined mechanism.

(5) The bull case for enshrinement

presents the argument that ePBS is still worth doing despite (4), and also explores the counter-factual of allowing the mev-boost

market to evolve unchecked.

Note

: we continue using the term “relay” for the post-enshrinement out-of-protocol PBS facilitator. It’s worth considering adopting a different name for these entities to not conflate them with relays of today, but for clarity in this article, we continue using the familiar term.

\cdot

Thanks

Many thanks to Justin, Barnabé, Thomas, Vitalik, & Bert for your comments.

acronym

meaning

PBS

Proposer-Builder Separation

ePBS

enshrined Proposer-Builder Separation

PTC

Payload-Timeliness Committee

ToB

Top-of-Block

(1) Why enshrine PBS?

[Why enshrine Proposer-Builder Separation?

](https://ethresear.ch/t/why-enshrine-proposer-builder-separation-a-viable-path-to-epbs/15710) outlines 3 reasons:

(i) relays oppose Ethereum's values, (note: strong wording is a quote from the original

)

(ii) out-of-protocol software is brittle, and

(iii) relays are expensive public goods.

The core idea was that ePBS eliminates the need for mev-boost

and the relay ecosystem by enshrining a mechanism in the consensus layer to facilitate outsourced block production.

While points (i-iii) remain true, it is not clear that ePBS can fully eliminate the relay market. It appears likely that relays would continue to offer services that both proposers and builders may be incentivized to use.

We can't mandate that proposers only use the ePBS mechanism. If we tried to enforce that all blocks were seen in the P2P layer, for example, it's still possible for proposers to receive them from side channels (e.g., at the last second from a latency-optimized relay) before sending them to the in-protocol mechanism. This document presents the case that enshrining is still worthwhile

while being pragmatic about the realities of latency, centralization pressures, and the incentives at play.

(2) Relay roles today

Relays are mutually-trusted entities that facilitate the PBS auction between proposers and builders. The essence of a PBS mechanism is:

(i) a commit-reveal scheme to protect the builder from the proposer, and

(ii) a payment enforcement mechanism to protect the proposer from the builder.

For (i), relays provide two complementary services:

1. MEV-stealing/unbundling protection

–

Relays protect builders from proposers by enforcing a blind signing of the header to prevent the stealing and/or unbundling of builder transactions.

1. Block validity enforcement

–

Relays check builder blocks for validity. This ensures that proposers only commit to blocks that are valid and thus should become canonical (if they are not late).

For (ii), relays implement one of the following:

1. Payment verification

–

Relays verify that builder blocks correctly pay the proposer fee recipient. In the original Flashbots implementation, the payment was enforced at the last transaction in the block. Other relays allow for more flexible payment mechanisms (e.g., using the coinbase transfer for the proposer payment) and there is an active PR in the Flashbots builder repo to upstream this logic.

1. Collateral escrow

–

Optimistic relays remove payment verification and block validity enforcement to reduce latency. They instead escrow collateral from the builder to protect proposers from invalid/unpaying blocks.

Lastly, relays offer cancellations (an add-on feature not necessary for PBS):

1. Cancellation support

–

Relays allow builders to cancel bids. Cancellations are especially valuable for CEX-DEX arbitrageurs to update their bids throughout the slot as CEX prices fluctuate. Cancellations also allow for other builder bidding strategies.

(3) A simple ePBS instantiation

We now present a simple ePBS instantiation, which allows us to consider the relay role post-ePBS. While we focus on a specific design, other versions of ePBS have the same/similar effects in terms of relay evolution. Let's continue using the following framing for PBS mechanisms:

(i) a commit-reveal scheme to protect the builder from the proposer, and

(ii) a payment enforcement mechanism to protect the proposer from the builder.

For (i) we can use the Payload-Timeliness Committee (abbr. PTC) to enforce that builder blocks are included if they are made available on time (though other designs like Two-slot and PEPC are also possible).

[

upload_b7e69764e2a40ab3654c4d27c2165ff0

1400×1018 121 KB

](https://ethresear.ch/uploads/default/original/2X/f/f8eb9cd384c4710e872d2fa2e6c2c24a529b89d2.png)

In the PTC design, a committee of attesters votes on the timeliness of the builder payload. The subsequent proposer uses these votes to determine whether to build on the "full" CL block (which includes the builder's ExecutionPayload

) or the "empty" CL block (which doesn't include the builder's ExecutionPayload

).

For (ii) we present a new unconditional payment mechanism called "Top-of-Block" payments. h/t to Caspar for casually coming up with this neat solution over a bistro dinner in Paris and Justin for describing a very similar mechanism in [MEV burn – a simple design

](https://ethresear.ch/t/mev-burn-a-simple-design/15590/1); c'est parfait

.

Top-of-Block Payments (abbr. ToB)

To ensure that proposers are paid fairly despite committing to the builder's bid without knowing the contents of the ExecutionPayload

, we need an unconditional payment mechanism to protect proposers in case the builder doesn't release a payload on time.

The idea here is simple:

- Part of the builder bid is a transaction (the ToB payment) to the proposer fee recipient; the transaction will likely be a transfer from an EOA (we could also make use of smart contract payments, but this adds complexity in that the amount of gas used in the payment must be capped – since the outcome is the same, we exclude the implementation details).
- The payment is valid if and only if the consensus block commits to the ExecutionPayloadHeader

corresponding to the bid.

- The payment must be valid given the current head of the chain (i.e., it builds on the state of the parent block). In other words, it is valid at the top of the block.
- The ExecutionPayload

from the builder then extends the state containing the ToB payment (i.e., it builds on the state after the unconditional payment).

- If the builder never reveals the payload, the transaction that pays the proposer is still executed.

The figure below depicts the ToB payment flow:

[

diagram-20230804

1728×1462 232 KB

](https://ethresear.ch/uploads/default/original/2X/a/a21663356aae0099361532c10df1ba48bcdc65e1.png)

For slots n

and n+2

, the corresponding ExecutionPayloads

are included and the EL state is updated. In slot n+1

, the builder didn't reveal their payload, but the payment transaction is still valid and included. This is a just-in-time (JIT) payment mechanism with two key points:

- the builder no longer needs to post collateral with the protocol, and
- the builder must still have sufficient liquidity on hand to make the ToB payment (i.e., they're still unable to use the value they would generate within the ExecutionPayload

to fund their bid).

If a builder does not have sufficient capital before the successful execution of the ExecutionPayload

, a relay would still be required to verify the payment to the proposer.

(4) Relay role evolution post-ePBS

Let's revisit how the relays' services evolve if PTC + ToB payments are introduced. We use

to denote that the relay is no longer needed and

to denote that the relay may have some edge over ePBS.

1. MEV-stealing/unbundling protection

–

Relay is no longer relevant

The consensus layer enforces the commit-reveal through the PTC, so the builder is protected in that a proposer must commit to their block before they reveal it.

1. Block validity enforcement

–

Relay is no longer relevant

No block validity check is made, but today's proposers only care about the validity of the block insofar as it ensures their payment is valid. ToB payments give them that guarantee. Note that there is an assumption that proposers only care about their payment attached to a block (and not the block contents itself). While this is generally the case, proposers may make other commitments (e.g., via restaking) that are slashable if not upheld (outside of the Ethereum slashing conditions). In this case, a proposer would need to know that a builder block also fulfills the criteria of their commitment made via restaking (e.g., to enforce some transaction order).

1. Payment verification

–

Relay is superior for high-value blocks

The ToB payment enforces the unconditional payment to the proposer. However, the relay can allow more flexible payments (e.g., the last transaction in the block) and thus doesn't require the builder to have enough liquidity up front to make the payment as the first transaction.

1. Collateral escrow

–

Relay is no longer relevant

Collateral escrow now becomes unnecessary and capital inefficient. If the builder has sufficient liquidity to post collateral, it is strictly better for them to just use ToB payments rather than locking up collateral.

1. Cancellation support

–

Relay is still needed for cancellations

Relays support cancellations, whereas the protocol does not.

Relay advantages over ePBS

Now the critical question: what incentives do proposers and builders have to bypass ePBS through the use of relays?

Key takeaway

– Relays probably will still exist post-ePBS, but they will be much less important and hopefully only provide a marginal advantage over the in-protocol solution.

1. More flexible payments

– Relays can offer flexible payments (rather than just ToB payments) because they have access to the full block contents. Enforcing this requires simulation by the relay to ensure that the block is valid. This adds latency, which may be a deterrent for using relays to bypass the P2P layer in normal circumstances. However, this would be needed for high-value payments which cannot be expressed via ToB payments (i.e., if the builder needs to capture the payment within the ExecutionPayload

to pay at the end of the block). Relays could also allow builders to pay rewards denominated in currencies other than ETH. Note that with zkEVMs, this relay advantage disappears because the builder can post a bid along the encrypted payload with proof that the corresponding block is valid and accurately pays the proposer (VDFs or threshold decryption would be needed to ensure the payload is decrypted promptly).

1. Lower latency connection

– Because relays have direct TCP connections with the proposer and builder, the fastest path between the two may be through the relay rather than through the P2P gossip layer (most notably if the relay is vertically integrated with a builder, implying the builder & proposer have a direct connection). It is not clear exactly how large this advantage may be, especially when compared to a builder that is well-peered in the P2P network.

1. Bid cancellations & bid privacy

– Because relays determine which bid to serve to the proposer, they can support cancellations and/or bid privacy. For a sealed-bid auction, relays could choose not to reveal the value of the bid to other builders and only allow the proposer to call getHeader

a single time. It doesn't seem plausible to support cancellations on the P2P layer, and bid privacy in ePBS is also an unsolved problem. With FHE or other cryptographic primitives, it may be possible to enshrine bid privacy, but this is likely

infeasible in the short term.

These benefits may be enough for some builders to prefer relays over ePBS, so we should expect the relay ecosystem to evolve based on the value that relays add. In short, we expect a relay market

to exist. A few examples of possible entities:

1. Vertically-integrated builder/relay

– Some builders might vertically integrate to reduce latency and overhead in submitting blocks to relays. They will need to convince validators to trust them, the same as any relay.

1. Relay as a Service (RaaS)

– Rather than start a relay, builders may continue to use third-party relays. Relay operators already have trusted reputations, validator connections, and experience running this infrastructure. If the services mentioned above are sufficiently valuable, these relays can begin to operate as viable profit-making entities.

1. Public goods relays
2. Some of the existing third-party relays may remain operational through public goods funding. These would likely be non-censoring relays that are credibly neutral and are supported by ecosystem funds. However, it's not clear that relay public goods funding would be necessary anymore after ePBS. At this point, relays will only provide optional services with a potentially minimal benefit versus the in-protocol option.

A proposer may hook up to multiple entities to source bids for their slot;

consider the example below.

[

upload_098ebda4f1832e6b5f69dccf7e4e68da

1831×985 137 KB

](https://ethresear.ch/uploads/default/original/2X/2/227c3d08a15646273254c2f9bf06de1e80b2c760.png)

Here the proposer is connected to two relays and the P2P bidpool. The proposer may always choose the highest bid, but they could also have different heuristics for selection (e.g., use the P2P bid if it's within 3% of the highest non-P2P bid, which is very similar to the min-bid feature in mev-boost

).

This diagram also presents three different builder behaviors:

- builder A

is part of the vertically-integrated builder/relay, resulting in a latency advantage over the other builders (and their relay may only accept bids from their builder).

- builder B

may be a smaller builder who doesn't want to run a relay, but is willing to pay an independent relay for RaaS to get more payment flexibility or better latency.

- builder C

might not be willing to pay for RaaS or run a relay, but instead chooses to be well connected in the P2P layer and get blocks relayed through the enshrined mechanism.

Note that builder A

and builder B

are sending their bids to the bidpool as well because there is a chance that the proposer is only listening over the P2P layer or that some issue in the relay causes their bid to go undelivered. It is always worth it to send to the bidpool as well (except in the case where the builder may want to cancel the bid).

The obvious concern is that builder A

will have a significant advantage over the other builders, and thus will dominate the market and lead to further builder centralization. This is a possibility, but we note that this is a fundamental risk of PBS, and not something unique to any ePBS proposal. There are still additional benefits of doing ePBS instead of allowing the mev-boost

ecosystem to evolve entirely outside the protocol.

(5) The bull case for enshrinement

Despite the realities of the potential relay advantages over the in-protocol mechanism, we still believe there is value in moving forward with ePBS for the following reasons:

- ePBS may be more efficient than running a relay

– It is possible that instead of running a relay or paying for RaaS, builders are competitive by just having good P2P connectivity. The relay-specific benefits described above may be too marginal to justify the additional operations and associated costs. In this case, it would be economically rational to simply use the enshrined mechanism.

- ePBS significantly reduces the cost of altruism

– Presently, the “honest” behavior is to build blocks locally instead of outsourcing to mev-boost

. However, 95% of blocks are built through mev-boost

because the reward gap between honest and mev-boost

blocks is too high (i.e., altruism is too expensive h/t Vitalik). With ePBS, honest behavior allows for outsourcing of the block production to a builder, whereas side-channeling a block through a relay remains out of protocol. Hopefully, the value difference between the P2P bid and the relay bid will be small enough that a larger percentage of validators choose to follow the honest behavior of using the P2P layer to source their blocks (i.e., altruism is less expensive). Additionally, relying only on the in-protocol mechanism explicitly reduces the proposers’ risks associated with running out-of-protocol infrastructure.

- ePBS delineates in-protocol PBS and out-of-protocol mev-boost

– Currently, with 95% of block share, mev-boost

is de facto in-protocol software (though there is circuit-breaking in the beacon clients to revert to local block building in the case of many missed slots). This leads to issues around ownership of the software maintenance/testing, consensus stability depending on mev-boost

, and continued friction around the integration with consensus client software (seeOut-of-protocol software is brittle). By clearly drawing the line between ePBS and mev-boost

, these issues become less pronounced because anyone running mev-boost

is now taking on the risk of running this sidecar for a much smaller reward. The marginally higher rewards gained from running mev-boost

incur a higher risk of going out of protocol.

- ePBS removes the neutral relay funding issues

– The current relay market is not in a stable equilibrium. A huge topic of discussion continues to be relay funding, which is the tragedy of the commons issue faced in supporting public goods. Through ePBS, the protocol becomes the canonical neutral relay, while allowing the relay marketplace to evolve.

- ePBS is future-compatible with mev-burn, inclusion lists, and L1 zkEVM proof generation

– By enshrining the PBS auction, mev-burn becomes possible through the use of the bidpool as an MEV oracle for each slot (we could use relay bids to set the bid floor, but this essentially forces proposers to use relays rather than relying only on the bidpool, which seems fragile). This constrains the builder blocks that are side-channeled in that they must burn some ETH despite not going through the P2P layer, which may compress the margin of running relays even further. Inclusion lists are also a very natural extension of ePBS (inclusion lists could be implemented without ePBS, but we defer that discussion to an upcoming post). Inclusion lists also constrain builder behavior by forcing blocks to contain a certain set of transactions to be considered valid, which is critical for censorship resistance of the protocol (especially in a regime with a relatively oligopolistic builder market). Once we move to an L1 zkEVM world, having a mechanism in place for proposers to outsource proof generation is also highly desirable (see Vitalik’s Endgame).

- ePBS backstops the builder market in the case of relay outages

– As relays evolve, bugs and outages may occur; this is the risk associated with connecting to relays. If the relays experience an outage, the P2P layer at least allows for the PBS market to continue running without forcing all proposers back into a local block-building regime. This may be critical in high-MEV scenarios where relays struggle under the surge of builder blocks and each slot may be highly valuable.

Overall, it’s clear that relays can still provide services in an ePBS world. What’s not yet clear is the precise economic value of these services versus the associated costs and risks. If the delta is high, it is reasonable to expect that relays would

continue to play a prominent role. If the delta is low, it may be economically rational for many actors to simply follow the in-protocol mechanism. We hope the reality lies somewhere in the middle.

What happens if we don't do ePBS?

It is worth asking the question of what happens if we don't enshrine anything. One thing is very clear – we are not in a stable equilibrium in today's relay ecosystem. Below are some possible outcomes.

- Non-monetizing, public goods relays continue searching for sustainable paths forward

– The continued survival of credibly neutral relays becomes a higher priority because, without ePBS, the only access validators have to the builder market is through the relays. Neutral relays will need to be supported through public goods funding or some sort of deal between builders, relays, and validators.

- Inclusion lists and censorship resistance are prioritized

– If we capitulate and allow the existing market to evolve, censorship resistance becomes increasingly important. We would likely need to enforce some sort of inclusion list mechanism either through mev-boost

or directly in the protocol (again, we think it is possible to do inclusion lists without ePBS – this discussion is forthcoming).

- We give up on mev-burn in the near-term

– Without ePBS, there is no clear way to implement mev-burn.

- We continue relying on mev-boost

software

– Without ePBS, mev-boost

and the relays continue to be de facto enshrined. We would probably benefit from more explicit ownership over the software and its relationship with the consensus client implementations.

Overall, we assess that the benefits of ePBS outweigh the downside (mostly protocol complexity) even if there exists some incentive to bypass it at times. The remaining uncertainty isn't so much if we should enshrine something, but rather what we should enshrine (which is a different discussion that I defer to Barnabé) – c'est la vie.

Appendix – advantages of well-capitalized builders under Top-of-Block Payments or collateralized bidding

In the case of multiple equal or roughly equal bids received, the proposer is always incentivized to select bids that include a ToB payment (rather than a flexible payment later in the block). This is strictly less risky for them than trusting a third party for accurate payments. Additionally, there is a latency advantage with ToB payments versus flexible payments if the relay must simulate the block to validate the payment (though the vertically integrated builder/relay wouldn't check their bids).

Relays would likely support ToB payments, though this is not possible if the builder doesn't have the collateral on hand. This inherently presents some advantage to larger builders with more capital (it's possible for the relay or some other entity to loan money to the smaller builder to make the ToB payment, but this would presumably be accompanied by some capital cost – again making the smaller builder less competitive on the margin).

Note that this same tradeoff exists whether the payment is guaranteed via a ToB payment or an in-protocol collateral mechanism. In general, this advantage is likely to arise very infrequently (i.e., builders will nearly always have capital for ToB payment on hand).

The way to counter this capital advantage for large builders would be to impose some in-protocol cap on the guaranteed payment (either via collateral or ToB). Then, no builder could offer a trustless payment to the proposer above the cap. However, this would simply increase the incentive for proposers to turn to out-of-protocol services during high-MEV periods, and they would more frequently be forced into trusting some third party for their payment verification, so we don't think this is the right path to take.

The frequency and magnitude of this advantage could be diminished if mev-burn is implemented because the only part of the payment that must

be guaranteed is the priority fee of the bid. In mev-burn, it may be reasonable to set two limits:

- Protocol payment (burn)
- Cap the maximum ToB burn transaction (or collateral) for the burn payments. In [MEV burn – a simple design

](https://ethresear.ch/t/mev-burn-a-simple-design/15590/1), Justin discussed the hypothetical example of a 32 ETH cap for

example on collateral here.

- Proposer payment (priority fee)
- This ToB payment can be left unbounded so that the proposer is never forced into trusting out-of-protocol solutions. This still favors well-capitalized builders, but at least it is not the full value of the bid, and the burn portion doesn't need to be guaranteed.

In this case, a failure to deliver the ExecutionPayload

would result in the following:

- Protocol
- the burn is guaranteed up to some capped amount (e.g., 32 ETH), but beyond that, a failure to deliver the ExecutionPayload

would result in the protocol socializing some loss (i.e., the amount of ETH that should have otherwise been burned).

- Proposer
- the priority payment is received in full regardless.

Merci d'avoir lu