This post starts by analyzing the advantages of consensus and execution block timing games both before and after the ePBS setting, focusing on the preparation time available to the next slot's proposer/builder for the subsequent slot. We then dissect the dynamics of timing when the builder block reveals, focusing on altruistic, honest, and greedy rational builder rationale. The second half of the post highlights the greedy rational builder's advantage in controlling the timing of the execution block reveal. Finally, we will list out a series of open questions for further exploration. First, a little meme:

Terminologies, Acronyms & Abbreviations

- ePBS (enshrined proposer-builder separation)

: A protocol enhancement where relayer functionality in mev-boost setting becomes part of the protocol, enabling trustless exchanges between validators and builders. New validator duties: payload timeliness committees is introduced. Unless specified otherwise, references below refer to block auction ePBS.

- CL (Consensus Layer)

: Where consensus activities like votes, slashing, and deposits happen…

- EL (Execution Layer)

: Where execution activities like user transactions and EVM happen.

- Pre-ePBS beacon block

: A consensus block that contains consensus data and a full execution block.

- Post-ePBS beacon block

: A consensus block that contains consensus data and an execution header, the header commits to the execution block which is later revealed. The transactions are hidden.

- Execution block

: A block that contains execution data and transactions. Produced by builders.

- Pre-ePBS Proposer

: Builds and reveals pre-ePBS beacon block.

- Post-ePBS Proposer

: Builds and reveals post-ePBS beacon block.

- Builder

: Builds and reveals execution block in post-ePBS setting.

- Beacon attester

: Votes on the consensus block before BeaconAttestationDeadline

.

- Payload attester

: Votes on the execution block before PayloadAttestationDeadline

.

Settings

Now we'll go over each setting case by case. Focus on the following variations:

- Pre-ePBS setting
- Post-ePBS setting
- Spec definition: assuming no networking and client processing delays.
- Beacon proposer rational behavior: Inclue networking and processing delays and rational proposer and builder behaviors.

Pre-ePBS + Spec definition

- Proposer reveals pre-ePBS beacon block, which contains an execution block at SlotStart

.

- Next slot beacon proposer or builder has SlotDuration

to prepare an execution block for the next slot.

- The execution block preparation time for next slot is SlotDuration

.

[

Screenshot 2024-04-25 at 11.18.56 AM

693×149 12.6 KB

](https://ethresear.ch/uploads/default/original/3X/d/1/d14a473ed9b05e60bbe6c8068e7aaf0083f3e015.png)

Pre-ePBS + Beacon proposer rational behavior

- Proposer reveals pre-ePBS beacon block as close to the BeaconAttestationDeadline

as possible.

- Next slot proposer or builder has SlotDuration-AttestationDeadline

time to prepare an execution block for the next slot.

- The next slot beacon proposer may also play timing game:
- The execution block preparation time for next slot is SlotDuration

w/ timing game.

- The execution block preparation time for next slot is SlotDuration-AttestationDeadline

w/o timing game.

- The execution block preparation time for next slot is SlotDuration

w/ timing game.

- The execution block preparation time for next slot is SlotDuration-AttestationDeadline

w/o timing game.

[

Screenshot 2024-04-25 at 11.19.22 AM

694×159 13.6 KB

](https://ethresear.ch/uploads/default/original/3X/7/4/74da7e206fa90ee82830043bbed4a51b2dac96af.png)

Note: The beacon proposer playing the timing game will consider P2P propagation time and the client processing times.

Post-ePBS + Spec definition

At this point, we assume readers have sufficient knowledge of how the basics of post-ePBS work. If not, refer toAnatomy of a Slot. In most ePBS designs, a slot is divided into two intervals: the first is about consensus that commits to the execution header, and the second is about revealing the execution result. This clear pipelining has several benefits but also introduces new dynamic which we will explore below.

- Proposer reveals post-ePBS beacon block at the SlotStart

.

- Winning builder observes the proposer’s commitment to its submitted execution header.

- If the header is committed and when it’s safe to reveal, the builder reveals the execution block at SlotStart / 2

.

- The execution block preparation time for next slot is SlotDuration / 2

[

Screenshot 2024-04-25 at 11.21.35 AM

732×175 15.6 KB

](https://ethresear.ch/uploads/default/original/3X/4/0/40483990d88767b417907d3987fd554620adce35.png)

Note: No one can start preparing for the next slot's execution block unless the current execution block is revealed.

Post-ePBS x Beacon proposer rational behavior

- Proposer reveals post-ePBS beacon block as close to the new BeaconAttestationDeadlineEPBS

as possible.

- Builder observes the proposer's commitment to its submitted execution header.
- If the header is committed and when it's safe to reveal, the builder reveals the execution block at SlotStart / 2

.

- The next slot beacon proposer may also play timing game:
- The execution block preparation time for next slot is SlotDuration - (SlotStart / 2) + BeaconAttestationDeadlineEPBS

w/ timing game

- The execution block preparation time for next slot is SlotDuration - (SlotStart / 2)

w/o timing game

- The execution block preparation time for next slot is SlotDuration - (SlotStart / 2) + BeaconAttestationDeadlineEPBS

w/ timing game

- The execution block preparation time for next slot is SlotDuration - (SlotStart / 2)

w/o timing game

[

Screenshot 2024-04-25 at 11.22.08 AM

697×207 16.7 KB

](https://ethresear.ch/uploads/default/original/3X/e/e/ee5b4978408abc7a23d6e4a27e3861048c8b3ed9.png)

Notes:

1. At first glance, the beacon block proposer will still play the timing game and try to release as close to BeaconAttestationDeadlineEPBS

as possible. Nothing changes in that regard.

1. BeaconAttestationDeadlineEPBS

is 1s shorter than BeaconAttestationDeadline

to account for payload timeliness committee duty fitting within SlotDuration

.

1. The beacon proposer playing the timing game will consider P2P propagation time and the processing times of both the CL client and the EL client.

[

Screenshot 2024-04-25 at 11.22.24 AM

723×195 20.9 KB

](https://ethresear.ch/uploads/default/original/3X/b/2/b2ba91ddc0a51a9e3757f8907ba98202f2e513aa.png)

Execution block reveal window

So far, we have focused on the beacon block reveal window. What about the execution block reveal window once its header is committed in the beacon block? The consensus spec indicates that the builder should reveal at SlotDuration / 2

, but what is the execution block reveal window? Let’s first consider how early the builder could reveal its execution block once it’s “safe” to do so. The builder monitors the fork choice weight of the beacon block which commits to its header. If the weight is weak, the builder may choose to withhold the execution block by sending a withhold message. Assuming the message is seen by more than half of execution block committee, a withhold boost is granted and gives fork choice weight to the beacon block’s parent, which allows subsequent honest proposer and attesters to reorg the weak beacon block of the canonical view, so the builder has a way out of unconditional payment.

To determine how much the fork weight is sufficient to reveal, consider the attack where the next slot’s proposer is malicious, and wants to reorg the builder’s execution block. We want the next slot’s attesters to recognize the builder’s execution block as head. This requires winning the following equation: assuming X

represents the percentage of current slot attesters who voted for the beacon block, $1-X$

is missing, and you want execution block’s $RevealBoost + X > ProposerBoost + 1 - X$

to be true. With $RevealBoost=40$

and $ProposerBoost=20$

, the builder should reveal when $X>40$

. This means over 40% of the current slot’s beacon committee members voted for the proposer’s beacon block for the builder to feel “safe” to reveal.

[

Screenshot 2024-04-25 at 11.22.47 AM

743×253 22.4 KB

](https://ethresear.ch/uploads/default/original/3X/f/f/ff126f5ba6a4bd8e4f2772ae860558a3469645ea.png)

Note: $RevealBoost + X > ProposerBoost + 1 - X$

assumes the next slot’s proposer wants to reorg the current beacon block + execution block. This is a more powerful attack than just reorg the execution block itself because it gets $1-X$

on its side due to (block, slot) voting.

Now we have determined that the builder may feel safe to reveal the execution block as soon as it observes that more than 40% of beacon attesters have voted for the beacon block. We will refer to this point as the SafeRevealDeadline

. When is the latest point a builder can reveal? The latest point is the PayloadAttestationDeadline

, which is when payload attestations vote on whether the execution block has revealed on time and valid. Thus, the builder’s reveal window spans from SafeRevealDeadline

to PayloadAttestationDeadline

.

Note: Similar to the beacon proposer, the builder will consider P2P propagation time and the execution processing times of the execution block.

Execution block reveal strategy

When modeling rational builder behaviors, consider that builders compete to submit the highest bids at SlotStart

and that having more time to prepare the execution block incurs an advantage to pack a more profitable execution block. The starting time for building the next slot’s execution block depends on knowing the execution result of the current slot’s execution block. Builders can be categorized into two modes: the winning builder of the current slot and all other builders who did not win but want to win next slot. The winning builder can start building for the next slot as soon as its header is committed in the beacon block, while all others must wait until the current’s block is revealed. Below is a table summarizing the three strategies of the winning builder can impose based on reveal time:

Builder Type

Reveal Time

Time for Others to Prepare Next Slot Block

Time for Self to Prepare Next Slot Block

Altruistic

SafeDeadline

SlotDuration-SafeRevealTime

SlotDuration-BeaconAttestationDeadlineEPBS

Spec

SlotDuration/2

SlotDuation/2

SlotDuration-BeaconAttestationDeadlineEPBS

Greedy Rational

PayloadAttestationDeadLine

SlotDuration-PayloadAttestationDeadLine

SlotDuration-BeaconAttestationDeadlineEPBS

[

Screenshot 2024-04-25 at 11.24.03 AM

823×268 20.4 KB

](https://ethresear.ch/uploads/default/original/3X/8/7/87f2dbdcc953b1df4b2bb38b18a6986e543dc350.png)

The builder of the current winning slot influencees when the next slot's builder can start building their block. The dominant strategy for a builder may be to win the current block auction and reveal as late as possible, thus securing more time to build than the competition and such advantage may compound across multiple subsequent slots.

If the next slot proposer plays timing game, all builders receive extra time, equal to the BeaconAttestationDeadline

, to prepare the execution block. This scenario assumes that builders continue to submit their bids and execution headers after SlotStart

, approaching the deadline, via p2p or RPC.

[

Screenshot 2024-04-30 at 9.02.05 AM

889×248 28.2 KB

](https://ethresear.ch/uploads/default/original/3X/1/0/10ddcfc505452ddb63da807366c41bf1605e1c6c.png)

Builder reveal strategy in other ePBS auctions

The advantage which the current slot winning builder gains may be fundamental in other ePBS auction designs, such as slot auction, but the dynamics and advantages slightly differ in other designs. Let's examine the examples in a simple slot auction:

In a simple slot auction, the beacon proposer selects the winning builder without committing to the execution outcome, they only commit to the future right of who can propose the execution block. Here, the beacon proposer timing game is no longer relevant. Under rational strategy, the proposer reveals the beacon block at SlotStart

. A builder following the honest spec definition should reveal at SlotDuration/2

, but under a rational, greedy strategy, the builder may reveal close to the ExecutionAttestationDeadline

.

[

Screenshot 2024-04-25 at 11.24.56 AM

734×268 18.2 KB

](https://ethresear.ch/uploads/default/original/3X/c/3/c3f4abd52ed84e736ba299fe4c5a613650901cab.png)

Revealing close to the deadline now gains more advantages than ePBS block auction:

1. Gives more time to build the execution block.
2. Gives everyone else less time to come up with a confident value bid for the next slot.

The main reason for a builder to reveal the block close to the execution deadline is to maximize the time available to build the execution block and chase more value, which is a different emphasis than in the block auction model. The winning builder may have a better valuation for the next slot, given its consistent view of the mempool but that may be irrelevant.

In the execution ticket model, since the ticket selection occurs far in advance, the execution proposer assignments can be determined and known an epoch before, it becomes less practical to manipulate. To attempt gaming the system, one would need to purchase a lot of tickets at once, but these tickets might have to go through a pending queue before becoming eligible. The specific design details are to be determined.

Open questions

The dominant strategy for the builder discussed above is purely speculative, there are many open questions to answer. One key question is how should the builder bid value correlates by having more time to prepare. How much of the bid value is based on more build time versus more private order flow? If most of the value comes from private order flow, it may be ok for the winning builder to have more time to build than everyone else. But to create a fair and competitive environment, it could be beneficial to define some baseline models that specifiy a minimal build time for all builders, ensuring some equal footings. Could there be incentives or enforcement mechanisms to ensure builders reveal their execution payload on time? One approach could be to make the PayloadAttestationDeadline

random from the perspective of the payload timeliness committee, making it harder for the builder to game the reveal time. Another straightforward solution is to set the PayloadAttestationDeadline

earlier, which requires consideration of several factors such as the propagation times for attestations and execution payloads, and the time needed to verify an execution payload. When a builder wins consecutive slots, such as slots n and n+1, does this provide a compounding advantage for winning subsequent slots like n+2, compared to winning only slot n+1? This aspect is currently unclear and requires more formalized analysis. Finally when a proposer commits to a builder’s block, could an execution state difference be shared and allow other builders to get the peak of the final state and allow other builders to start building on top without winning builder having to leak the payload? Gossiping the entire execution state is not feasible, but perhaps a cryptographic solution like Verkle trees or SNARKs could be used to communicate the state difference efficiently. However, it’s still uncertain whether this is fully safe and what second-order effects might arise.

Acknowledgment

:

Although reviews cannot be viewed as an endorsements, I want to thankBarnabe for his feedback on typos and his suggestion to extend to the n+1

slot. This extension provides a clearer impression of the time available for n+1

builders to construct their block if the n+1

proposer is engaging in timing games.