

Block-auction ePBS versus Execution Ticket

[

DALL-E 2024-04-07 15.26.59 - Design an image featuring the exact text 'ePBS vs ET' in a clear, bold font against a deep space background. The background should showcase the beauty

1024×1024 289 KB

](https://ethresear.ch/uploads/default/original/2X/9/92a53090dcf5ab40d39f98af4bb57e513407add8.jpeg)

This writeup compares ePBS and ET in their current spec forms. It outlines their similarities and differences, including a table below for reference. First and foremost, it presents some disclaimers, related work, and relevant acronyms.

Disclaimers

- Enshrined proposer builder (ePBS) referenced below primarily refers to block auction ePBS. There are various forms of PBS that people will discuss, such as slot auction PBS and others. The comparison below focuses mainly on block auction ePBS versus ET.
- Execution tickets (ET) referenced below are from the post in related work. It's important to remember this was the first post on execution ticket design. The design is volatile, and nothing is final, including ePBS. It is highly subject to change in the coming months.
- The views expressed in this post are my personal views and do not reflect those of my team or company.

Related work

[Payload-timeliness committee \(PTC\) – an ePBS design](#) By Mike - July, 2023

[Execution Tickets](#) By Mike and Justin - Dec, 2023

[Minimal ePBS Beacon Chain Changes](#) By Terence - Feb, 2024

[ePBS Design Constraints](#) By Potuz - Feb, 2024

[Payload Boost](#) By Potuz - Feb, 2024

[ePBS Consensus Spec](#) By Potuz (still work in progress)

Acronyms & abbreviations

ePBS - enshrined proposer builder separation

ET - execution ticket

PTC - payload timeliness committee

Thanks

Many thanks to [Barnabé](#), [Mike](#), and [Potuz](#) for the comments on the draft.

ePBS Vs ET Similarities

Both designs directly address our main issue: the relay as a choking point. Today, a trustless path exists to allocate the execution proposing rights to yourself, that's local building. ePBS creates a trustless path for the beacon proposer to allocate execution proposing rights to someone else

. ET achieves this by excluding the execution proposer role from the consensus validator set. Critics often raise concerns about relay bypassability in the ePBS design, but this point is usually countered by explaining that ePBS is not about removing the relay. Instead, it's about enabling validators and builders to enter into a trust-minimized agreement with one another without the presence of a trusted relay mediating the agreement. Builders can assume the relay role and open an RPC port if needed.

Both designs decouple the delivery of the consensus block from the execution block, ensuring the inclusion of the consensus block even if the execution block is missing or invalid. In ePBS, the proposer receives an unconditional payment. In ET, the protocol gets the payment through ticket burning, or MEV burn.

Both designs enable the resale of block space. In the case of MEV-Boost today and the current ePBS design, the beacon

proposer puts block space up for auction, and the highest bidder among builders wins. Once the bid and transaction root are committed, the block space is locked in, meaning the beacon proposer can sell it only once. In contrast, the slot-auction variant of ePBS only requires the beacon proposer to commit to the bid's value and the execution proposer's ID, offering the execution proposer the chance to resell the block space. When it comes to ET, the reselling rules are protocol-dependent: if the protocol allows for the reselling of the ticket, then the ticket holder gains the right to resell the block space. Otherwise, the ticket holder can only sell the building rights of a block to another party while still retaining the role of the original proposer.

ePBS Vs ET Differences

[Timing game attack](#) occurs when a block proposer delays their proposal until the last second to extract more mev. In ePBS, the dynamics of the timing game remain unchanged, with consensus block proposers still incentivized to propose blocks close to the 4s mark. In ET, consensus blocks no longer commit to execution blocks, removing the incentive for consensus validators to engage in the timing game. Execution proposers, however, are still incentivized to play the timing game, which raises questions of fairness, but this is no longer an issue for consensus.

One of the differences is the timing of the execution block deadline. Currently, the deadline is when the proposer retrieves the header from the relayer, which is the start of the slot. In ePBS, the deadline remains the same. The slot starts when the execution block is committed in the consensus block. After that point, execution can no longer be changed. In ET, because the execution block does not commit to the consensus block, the execution deadline is pushed back to the second half of the slot. This could represent a significant mental shift for applications or MEV-related systems. Both designs present challenges in maintaining slot durations under 12s. In ePBS, the 12s duration is upheld by maintaining a small payload timeliness committee without additional rounds of aggregation. In ET, the current size of the execution committee is unknown.

Both designs feature an inclusion list. In ePBS, the inclusion list for slot n is applied to slot $n+1$, even though the transactions may be available for slot n . In contrast, in ET, since the consensus block does not commit the execution block, the inclusion list for slot n can be directly applied to the execution block at slot n . This design is more advantageous, as there are fewer duplicated transactions over gossip between the inclusion list and the execution block of the same slot.

Consensus blocks that do not commit to execution present specific trade-offs, with the primary concern being the potential for execution block equivocation. In ePBS, importing more than one execution block over p2p networks is only possible if multiple consensus blocks are processed. In this model, only the consensus block is subject to slashing. In ET, it is possible to import more than one execution block over p2p. Each is subject to its slashing rules, increasing the complexity of specs and implementation.

Another trade-off involves the circuit breaker mechanism. Currently, to short-circuit a relayer for producing an invalid block, this mechanism involves an out-of-band design because the relayer's identity is unknown on-chain, and the relayer's address is not part of the consensus block. In contrast, ePBS records the builder's address on the chain, and the builder's public key and signature are included in the block, making it straightforward to bypass or short-circuit a builder if they fail to produce a block or produce an invalid one. The consistent production of execution blocks is less challenging in ePBS. In ET, the execution block proposal operates within its domain, complicating the implementation of a circuit-breaker-like system.

Open Questions

ePBS contains open questions, such as how to address the issue of proposer split view. The current approach is to implement [payload boosts](#), enabling the builder to inform PTC of its intention to withhold the execution block to reorg the current slot consensus block.

ET contains open questions regarding economics, the scheme for selling and reselling tickets, and the slashing of execution blocks. The absence of slashing raises questions about its compatibility with Casper FFG's principle of accountable safety and whether it's just an LMD GHOST fault.

The community's decision is not between ePBS and ET but whether to pause until a satisfactory solution emerges or proceed with ePBS. Meanwhile, we are assessing whether ePBS is a stepping stone towards ET and whether they belong to the same skill trees. The subsequent table helps to clarify the overall picture.

Comparison and Contrast Table

ePBS

ET

Reduce trust between validator and builder

ePBS integrates mev-boost directly into the protocol, allowing beacon proposers to securely obtain bids from builders without the need for trust of an intermediate party. Builders may incorporate the relayer role by opening up its RPC port.

ET achieves this by completely isolating the role of the execution proposer from validators. Validators do not need to propose an execution block at all.

Unconditional payment

In ePBS, if a consensus block stays canonical but the execution block is invalid or not revealed, the beacon proposer still receives payment as part of the protocol if and only if the consensus bid captures the full payment.

In ET, proposing execution blocks is external to the beacon proposer's responsibilities, which becomes irrelevant in this context.

Circuit breaker

In ePBS, if a builder is at fault, this can be observed on the chain. The beacon proposer for the next slot will bypass any submissions from the faulty builder's address after a subjective timeout period.

In ET, there is no circuit breaker mechanism. The beacon proposer does not select the execution builder.

Timing game

In ePBS, a proposer might delay getting the header until the last moment to capture more MEV. This approach can distort consensus and raise questions about fairness for the next slot's proposer.

In ET, an execution proposer might delay broadcasting the execution block until the last possible moment to capture MEV. While this does not distort the consensus process, it still raises questions about fairness for the proposer of the next slot.

Inclusion list

In ePBS, an IL for slot n is enforced for slot $n+1$. Transactions from slot n may already be part of the IL for $n+1$, which is acceptable. Only a summary is committed on chain.

In ET, an IL for slot n could be applied to slot n itself, as the consensus block does not specify the contents of the execution block, only the builder's identity.

Mev burn

In ePBS, implementing MEV burn is difficult due to the complexity of ensuring bid credibility. The protocol cannot prevent collusion between proposers and builders nor guarantee that builders will accurately report the value to be burned.

In ET, the ticket itself is burned. Not burning full amounts could lead to profitable off-chain agreements between parties. Partial auction is hard to do here.

Empty slot

In ePBS, if a consensus block is skipped, no execution block can be produced. However, if the consensus block exists but the execution block is skipped, the slot will contain only consensus information. Empty slot can be made to apply a penalty on the proposer.

The current thinking is the same as ePBS, but it may not be. Assuming SSF and a clean consensus/execution separation, it may be possible to have an execution block if the consensus block is missing.

Slot time

In ePBS, upholding a 12s slot time is highly required.

We need concrete evidence to explain why it cannot be identical to ePBS.

Fork choice rule

(Block, slot) fork choice rule is implemented to enable attesters to vote against late block.

Single slot finality is likely required.

Execution block equivocation

In ePBS, execution block equivocation is prevented because the consensus block commits to one execution block. Any execution block that is not committed is ignored from every node's perspective.

In ET, execution block/ticket equivocation is possible. It may require some form of slashing scheme. As with all slashing designs and implementations, this introduces an element of unknown complexity. The question is whether ticket equivocation is the same as execution block equivocation under ePBS's slot auction scheme.

Bid delivery

In ePBS, if the timing game matters, the bid exchange will happen using RPC over P2P. RPC is faster, and P2P is just the default fallback for liveness.

In ET, the timing game does not involve consensus. Bids can be exchanged directly over P2P. This means builders don't need to open their RPC ports to the beacon proposer, reducing one less concern.

Multi slots attack

In ePBS, acquiring multiple slot proposals is more challenging due to the validator requirements, where each validator has a minimum bound. This is where shuffling becomes relevant.

In ET, the number of ticket participants is unclear, making it increasingly relevant that a single party might propose multiple slots in succession. Multi-slot MEV extraction or types of attacks could become a more significant issue.

Reselling

In ePBS, the proposer requests and receives the header from the builder JIT. The proposer then reallocates this opportunity to the builder. Builders have the option to collaborate with searchers or work alongside other builders.

In ET, the winning round's execution proposer can resell its block space to other entities.

Economics

In ePBS, the builder pays the proposer for the bid.

In ET, the protocol issues the ticket, and the execution proposer pays the proposal cost. The protocol sets the ticket price according to a supply and demand curve.

Hard fork complexity

In ePBS, transitioning to the ePBS regime through a hard fork based on a specific timestamp is relatively straightforward.

In ET, implementing a hard fork necessitates some degree of forward planning regarding proposer execution. There may be ways to start the ticket system without two forks, for example, by conducting a one-off sale or employing a 1559 AMM style market. However, it is too early to make definitive statements at this stage.

Attesting for execution reveal and validity

In ePBS, a PTC committee is utilized to attest for the execution reveal on time and ensure its validity. The committee size is determined to be appropriate where aggregation of attestations is not required. Consensus block of the next slot includes PTC attestations.

In ET, a PTC committee could be employed to verify execution validity. The specifics of how this operates will become clear once the specs are released.

Open questions

In ePBS, a significant design challenge involves proposers splitting attester views, which forces builders to decide whether to reveal their execution block. Payload boost represents a promising effort to address this issue.

In ET, the main challenges include addressing execution block equivocation, developing a viable hard-forking strategy for transitioning to the ET framework, ensuring all processes adhere to the 12s timeframe, and detailing the protocol's ticket issuance mechanism.