Thanks for the post! Interesting idea, here are some of my thoughts:

- The analysis of the economics of MEV becomes more complex when shifting to this model. Currently, the highest bidder wins, but in this new model, the winner is determined by a randomly chosen lottery ticket, moving from a first-price auction to a more complex format. Is there any analysis using mainnet data from the past six months to retrospectively examine outcomes under this new auction model?
- In the previous ePBS design, which Potuz, EPF, and I worked on, builders were staked. This approach faced resistance due to capital inefficiency and a need for compelling reasons. Is it reasonable to consider the purchase of an execution ticket as a form of staking? I will attempt to outline the differences. The introduction of execution tickets expands the design space, balancing research and implementation complexities.
- Buying Execution Tickets:
- Enables slashing builders in the protocol for missing slots.
- Allows for dynamic price adjustments.
- Permits burning of tickets.

- Enables slashing builders in the protocol for missing slots.
- Allows for dynamic price adjustments.
- Permits burning of tickets.

- Staking Builders:
- Only allows for social slashing of builders outside the protocol, though in-protocol slashing can be introduced.

- Only allows for social slashing of builders outside the protocol, though in-protocol slashing can be introduced.

- Side note: I think the separation between the validator and builder is more pronounced here. Currently, anyone can opt for a local builder. In our ePBS design, choosing a local builder requires staking 1024 ETH. In this new design, economic factors make it impractical for validators to use a local builder, as purchasing execution tickets for local block builders is no longer profitable.
- From an implementation perspective, it’s uncertain whether client teams will be responsible for developing the new “execution proposer” software. If not, this could be advantageous in reducing the codebase. However, it’s important to note that creating the execution proposer software will be challenging, especially if slashing conditions are applied. The lookahead aspect is also complex, as it involves caching elements requiring careful analysis.
- I believe the execution attester role should be integrated into the consensus duties, utilizing the current committee structure. For instance, the first index of the committee could perform the execution of attesting duty. Limiting the number of committee members is preferable to avoid an additional aggregation round, thereby saving approximately 4 seconds at the second slot time. An ideal network size would be around 512. This concept necessitates further thought and analysis, particularly regarding the networking aspects.
- The beacon block proposer for the subsequent slot will incorporate the execution attestations from the previous slot into their beacon block. The method for determining rewards and penalties for this action still requires analysis.