Towards Enshrined Proposer-Builder Separation

mike neuder – ethereum foundation

eth cc – july 17, 2023

Outline

- What is Proposer-Builder Separation?
- `mev-boost`, out-of-protocol PBS
- Enshrining PBS
  - Reasons to enshrine
  - Two-slot PBS
  - Payload-Timeliness committee
- Optimistic relaying
  - Current block submission
  - Optimistic v1
  - Optimistic v2
  - Endgame

What is Proposer-Builder Separation

Etymology

Proposer/block builder separation-friendly fee market designs

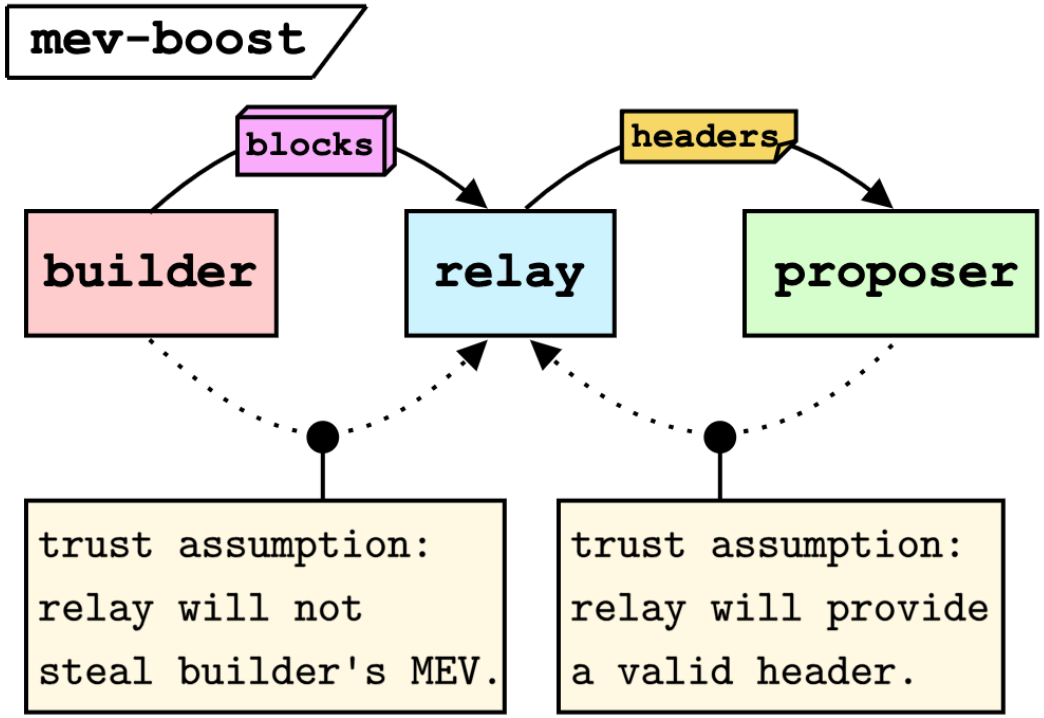
Economics

vbuterin Jun '21

- Proposers = validators selected to propose a block during a slot (unsophisticated)
- Builders = participants capable of constructing high-value blocks (sophisticated)
- Decouple these two roles to avoid centralization pressures

mev-boost

out-of-protocol PBS

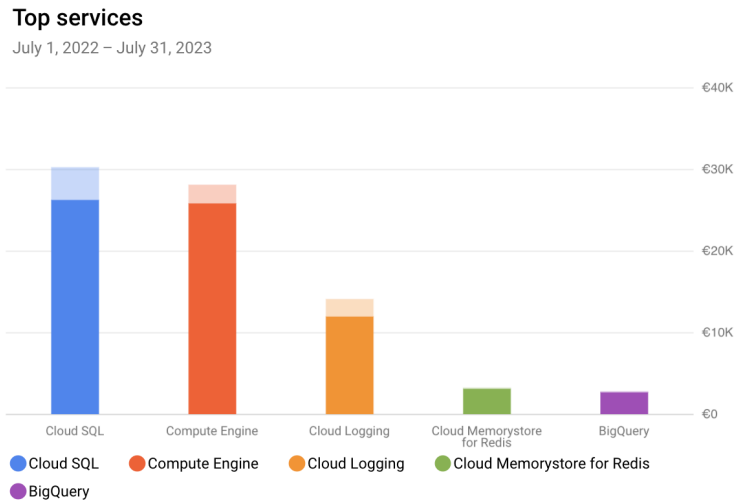


- Relay serves as a mutually trusted auctioneer
- Validators run sidecar software to interact with relays
- Massive adoption: 95% of blocks are built using `mev-boost`

Enshrining PBS

Reasons to enshrine

- Relays erode Ethereum's values
  - Decentralization
  - Censorship resistance
  - Trustlessness
- Out-of-protocol software is brittle
  - Low-Carb Crusader attack
  - Shapella bug
  - Coordination costs
- Relays are expensive
  - No clear funding model
  - Heavily-used public goods infrastructure



Enshrining PBS

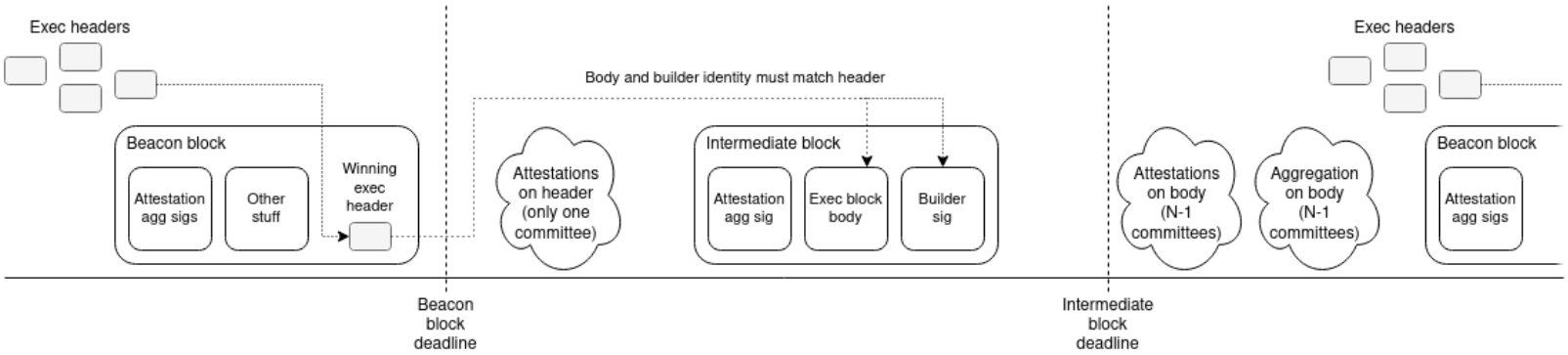
Two-slot PBS

Two-slot proposer/builder separation

Proof-of-Stake ■ proposer-builder-separation

vbuterin 4 Oct '21

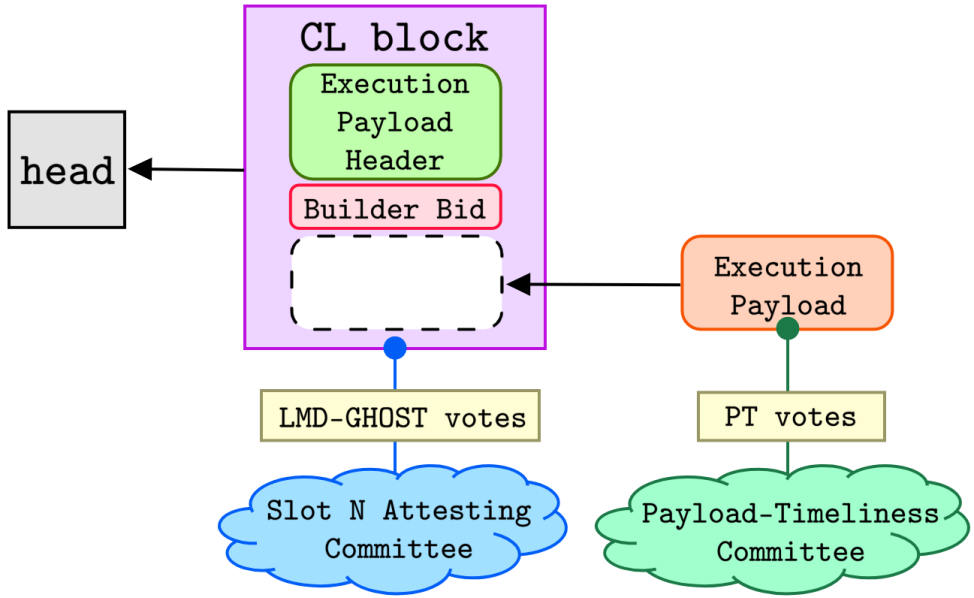
See previous ideas on this topic: Proposer/block builder separation-friendly fee market designs 467



- Gives the builder block attestation weight by partitioning the attesting committee
- Weakens the security properties of the consensus layer

Enshrining PBS

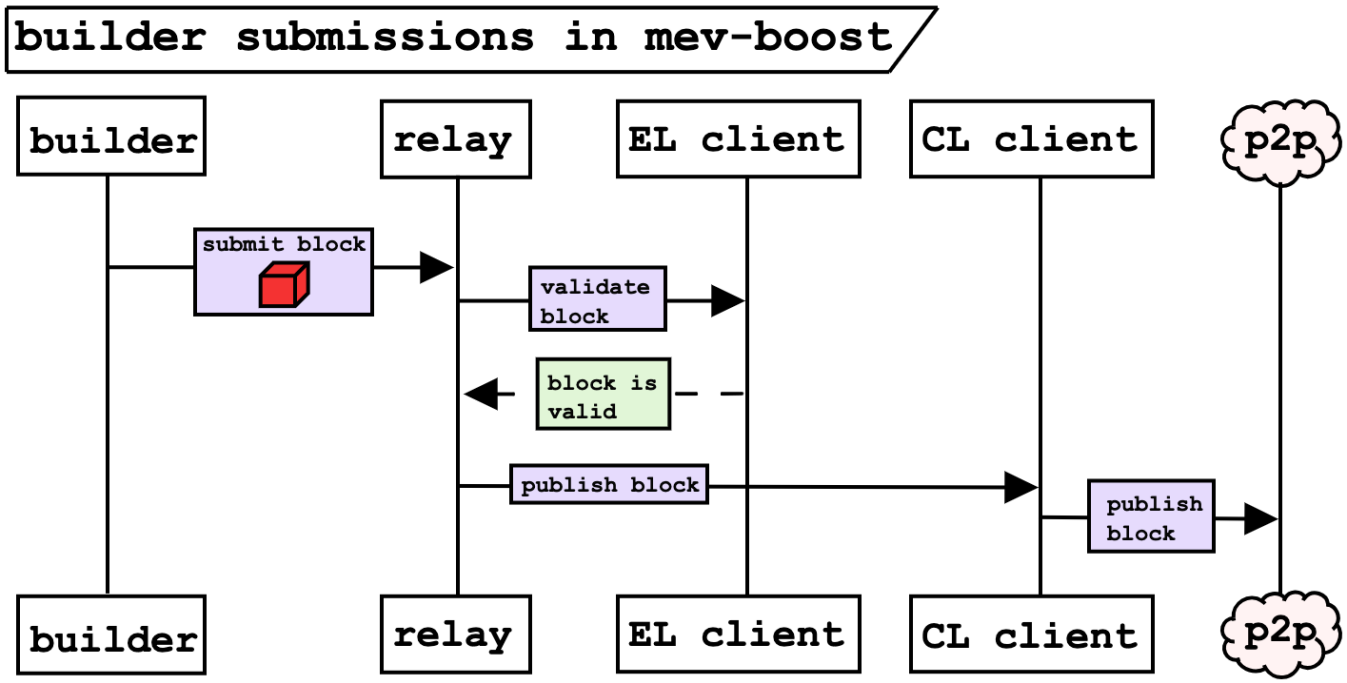
Payload-Timeliness Committee



- Consensus-layer block is produced *without* any transactions
- Consensus-layer attestations remain the same
- Builder reveals the payload (list of transactions)
- Payload-Timeliness Committee votes on if the payload was published
- Limits the impact on the fork-choice rule

Optimistic relaying

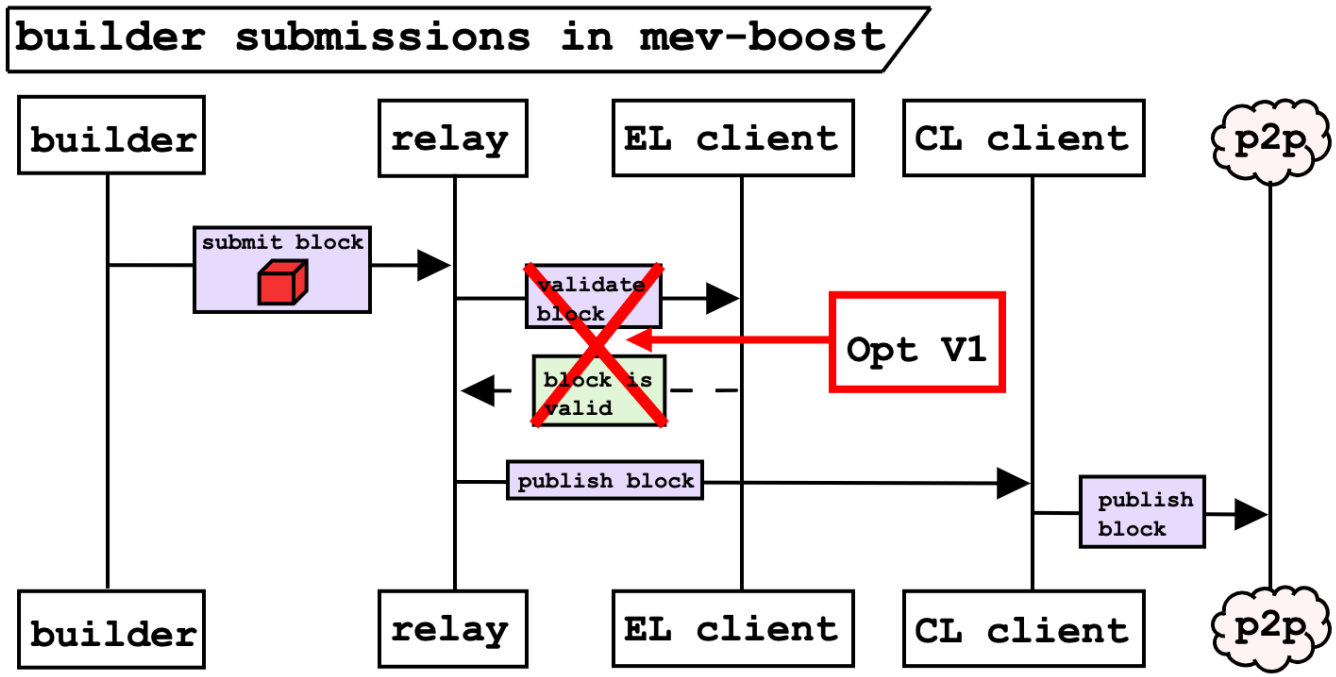
Current block submission



- Lots of latency in the block submission flow
- How can we make this look more like ePBS?
- Work from the bottom-up, removing relay responsibilities!

Optimistic relaying

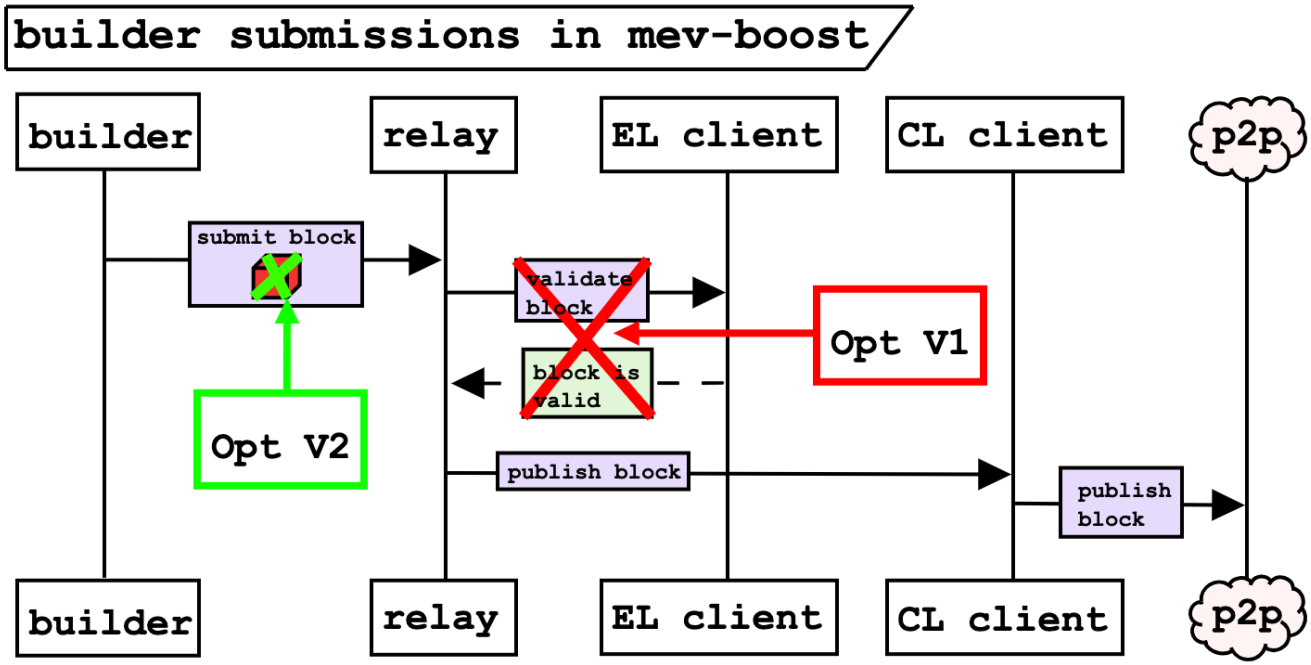
Optimistic v1



- Skip the block simulation
- Immediately saves $\approx 100 - 200$ ms of latency
- Collateralize builders to protect proposers

Optimistic relaying

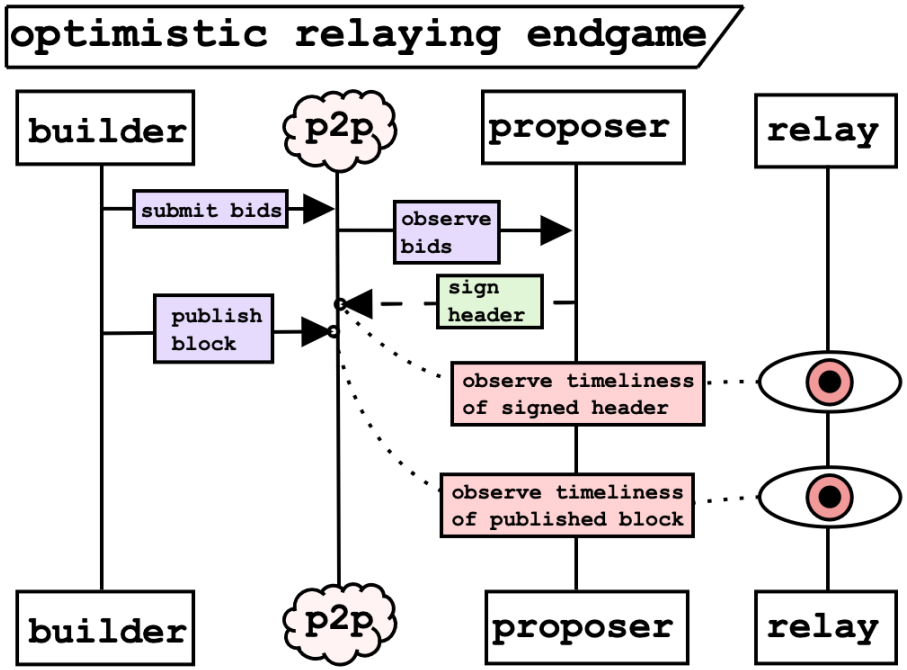
Optimistic v2



- Don't wait for block download
- Saves an additional $\approx 50 - 100$ ms of latency

Optimistic relaying

Endgame



- Relay just observes the timeliness of events
- Sounds a lot like ... Payload-Timeliness Committee

Wrap-up

- We are extremely dependent on `mev-boost` relays
  - Relays are trusted, centralized, and increase censorship surface area
- ePBS eliminates that dependency
  - Iterating on "top-down" approaches (Payload-Timeliness Committee)
  - Concurrently working on the "bottom-up" optimistic relay roadmap

Thanks!