

Builderrelays: Builder-Relay Vertical Integration

tl;dr

builder-relay vertical integration is nearer than we think. It's unclear what to do. While the market incentives for builder-relay integration have been known for a while, recent announcements (discussed below) make renewed discussion feel needed.

Builderrelays Are Coming

The Builder Perspective

The incentives for builders to run their own closely-integrated relays is clear. It cuts down latency by removing the need to send blocks to an extra actor and removes a simulation step (for non-optimistic relays). There's nothing much new here.

While we have seen small or neutral builders run relays in the past (and FB has always done so but in a non-integrated way), [Titan launching a relay](#) seems more substantial, given their position in the market and focus on "additional features". Stephane made a similar pitch not long ago.

PEPC-Boost

[PEPC-Boost](#) was conceived as a PoC for PEPC. The idea is to have a relay enforce some block building rules on behalf of the proposer with the first example being running two separate auctions (top-of-block and rest-of-block). This looks a lot like a primitive version of block building. It seems natural that the "proposer commitment" will evolve to include more than 2 bundles per block as this likely increases profit, leading to the relay basically becoming a builder.

Ultra Sound Relay Monetisation (for PG)

JD has been talking about making a change to the US relay to capture some value. US is currently faster than other relays. This means that US has an additional window for new (presumably higher) bids to flow in. They plan to take the delta between their highest bid and the highest bid that could have been seen by other relays and put that in some funding pool.

The amount the relay is able to skim, not only depends on how much faster it is, but more generally on which builder submissions it receives exclusively.

Staking Yield Maximisation

In the short run, reducing latency by directly connecting to block builders should increase validator payouts. With teams already working on [MEV-related optimisations to maximise staker revenue](#), it would not be surprising to see node service providers opting to connect to builderrelays.

Implications

The extreme case is that competition between builders means that the only way to reasonably be a block builder is to also be a builderrelay. This would turn competitive building into a permissioned role. Since becoming a relay requires doing BD to garner trust and be whitelisted by validators, the barriers to entry into the builder market becomes quite high. This puts builderrelays in a position with significant market power over both validators and users/searchers. I'm not yet quite sure how to think about this.

The less-extreme case is that competitive forces mean that several builderrelays pop up, but difficulty getting new relays adopted and relative efficiency of optimistic relaying (and something like US' monetisation strategy but with kickbacks to builders) means that several builders still share single relays to pick up scraps left by the builderrelays at low-volatility times in which latency plays less of a role.

Wat do?

I don't feel certain about what should be done, but here are a few options.

Reduce inefficiencies in disintegrated system:

Make current relays resemble builderrelays as much as possible to reduce integration incentives. Features like optimistic relaying and relays taking on bidding duties (e.g. US proposal) are examples of this.

Multi-relay queries:

If validators only signed headers that they receive from at least N

relays (assuming relays are still public entities and not sybils), this would significantly hamper the incentive to integrate very

closely with a single relay given that its the \$N\$'th fastest relay that counts. The tradeoff for validators is that they suffer less liveness risk, but also likely receive lower revenue.

SUAVE:

If validators could trust anyone running their relay in a kettle (or just SGX?) then becoming a relay (or builderrelay) would be much lower barriers to entry than doing BD. A validator would simply need to check a registry of keys that correspond to valid kettles/SGX-relays. The problem here is that SGX can enforce that a bid and header correspond to a valid block which does pay the bid, but it cannot enforce that the relay responds to the validator's signed header (e.g. the SGX can be disconnected from the network). An alternative way to address relay trust issues is to have the validator run an SGX node themselves to custody the data to prevent liveness attacks, but this needs a lot more thinking around implementation.

Preserve Open Source Norm For Relays

to prevent more bespoke logic moving to the relay and making integration with the builder more transparent when it happens.

Move Some Building Logic Back To The Validator:

This is an incredibly complex topic, but should be listed here.

I would appreciate it if people weighed in with some thoughts