Abstract

It seems that the prevailing thought is that the order-flow network effects can not allow a fair builder's market, as the order-flow network effects will quickly create advantages for single block builders that will rule the market. Many people conclude that the only solution is a decentralized builder (like SUAVE). This write-up argues that there is a better way: Only decentralize the order-flow-management, but still keep the performance and sophistication of centralized block builders.

Introduction:

Decentralized block builder is a highly anticipated and discussed topic. Likely, they will enable the following advantages:

- Solution to order-flow centralization
- Pre-confirmations features for users
- In-protocol front-running protection
- Being a decentralized infrastructure itself

But likely, decentralized block builders will be accompanied by the following disadvantages:

- Developing decentralized system is generally hard
- Evolution of the decentralized block builder will be slower than many competing centralized block builders
- Decentralized system have performance trade-offs that may hinder the best UX for Ethereum users - e.g., revert protection for txs might require very performant builders.
- Support for custom transaction patterns: Applications like (McAMMs) need custom patterns to enable their full potential.

Given this situation, the following will describe another idea how we can ensure a non-centralised builder ecosystem by controlling only the order-/tx-flow by a decentralized entity - the decentralized order flow distributor (DOFD). This has the potential to lift all the disadvantages for a decentralized block builder mentioned above.

Idea:

The core idea is to build a decentralized order flow distribution system: User post transactions encrypted into an encrypted mem-pool. They use public DKGs-public-keys to encrypt their order/tx flow. The distributed keys are managed by a network of DOFD nodes. Every user will publish 3 pieces of information: the transaction without signature encrypted by a first DKG public key, the signature of their tx encrypted by a second DKG public key and a zk-proof that the encrypted signature is valid for the encrypted transaction.

The DOFD will select per block a small subset of all allowed-listed builders to see the encrypted transactions without signatures, such that this subset can build blocks. The DOFD nodes will make the order-flow visible for the selected builders by publishing the private DKGs-keys encrypted with the public keys of each of the selected builders. These selected builders can then decrypt the private DKG keys, and hence they can also decrypt the encrypted transactions without signatures. With this order-flow, the builders will participate in the usual MEV boost auction. One of the selected builders should win this MEV boost auction against normal/non-participating builders, as the order-flow gives the selected builders a huge advantage. This will result in a final optimized block - minus the signatures. The signatures will be made visible to the block proposer, after they committed to use the winning block. Once the block-proposer has made its commitment to the DOFD network, the DKG nodes will publish their secrets that allow the block-proposer to build the final block with signatures.

If a builder misbehaves, for example, allows sandwiching attacks, then they will be slashed by the DOFD-network and will not get any future order-flow.

This might be implemented via "eigen layer staking mechanisms".

These rules enable an equal playing field for block builders and hence, could serve as a foundation for a healthy competition between builders, as everyone has access to the same order flow. Note that fancy techniques like pre-confirmations or in-protocol front-running protections can be implemented as well:

- Pre-confirmations:
- If every one of the block-builders of the next turn - the block-builders who will be able to decrypt the order-flow of the public mem-pool - give pre-confirmations to a user, then the user would know that his transaction will be included as pre-defined. If pre-confirmation are an important tool, the allow-list of builders could only contain builders that are actually providing this feature.
- If every one of the block-builders of the next turn - the block-builders who will be able to decrypt the order-flow of the public mem-pool - give pre-confirmations to a user, then the user would know that his transaction will be included as pre-defined. If pre-confirmation are an important tool, the allow-list of builders could only contain builders that are

actually providing this feature.

- In protocol front-running protection:
- If the builders have to be in an allow-list and if they have to provide collateral (maybe via eigen layer-systems), then one can easily slash them, if they misbehaved or disregard some rules during block-building.
- If the builders have to be in an allow-list and if they have to provide collateral (maybe via eigen layer-systems), then one can easily slash them, if they misbehaved or disregard some rules during block-building.

Discussion

:

Allowing builders to only see the transactions without the signatures prevents them from publishing the transactions to other builders. But still the trade intends are visible to them and could disclose valuable information. Hence, traders would have to split their trades into smaller pieces, such that the pure information is not very valuable. (If one really does want to prevent the orderflow information from leaking, SGX might be the needed)

Another main disadvantage of the proposed solution is that the list of allow listed builders needs to be maintained by governance. The system is not permission less and will also require security deposits from builders and validators. A more governance minimized system would be preferred. But - for comparison - also a decentralized builder would need some degree of governance to decide about features likes reconfirmation techniques, etc.

Also, another disadvantage is that certain block-builders can still try to gain more influence by temporary operating non-profitable and thereby trying to make in unattractive for others to join.

But maybe all these disadvantages are outweighed by the faster evaluation of builders. Also, block builders can quicker support more sophisticated applications like MEV capturing AMM (McAMMs), etc.

I could also imagine that short-term the DOFD is more valuable for the eco-system as a decentralized block builder - especially due to the fact of its faster evolution and not having SGX as a trust assumption.

I am really curious on your opinions. Which other disadvantages or advances for DOFD approach vs decentralized builder approach do you see?