Block Negotiation Layer

Realigning block building incentives and responsibilities on Ethereum

Author: Łukasz Miłkowski

thanks to the one and only Blocknative team - Chris Meisl, Sajida Zouarhi, Mobeen Chaudhry, Bert Kellerman, Julio Barragan and others for helping with creation of this document and fruitful conversations

Also many thanks to@barnabe for some last minute content suggestions

The Merge was the greatest change ever made to the Ethereum network, resulting in far more than a shift to Proof-of-Stake consensus. The Merge also saw the beginning of Proposer/Builder Separation, which shifted the technical and computationally intensive act of block building from the Block Proposer to specialized actors known as Block Builders - just to increase the incentives. Bridging the gap between these two actors is the critical, yet often overlooked role of Relays. A lot of the network burden is placed on these non-remunerated Relays, who act as trusted, MEV-optimizing parties that protect everyone’s best interests.

As of today, most of Ethereum’s research focuses only on enshrining the existing Proposer/Builder Separation into the protocol. This article, on the other hand, explores the benefits of adding a third layer to the network, known as the Block Negotiation Layer (BNL). Next to Execution and Consensus Layers, the BNL serves to extend the role of Relays. The BNL introduces a new network actor - Block Curators - and Proposer Intent mechanisms to reduce computational inefficiencies affecting both Proposers and Block Builders and improve overall network security.

Some of the currently explored enshrinement mechanisms might be unfavorable to the majority of network Proposers. And because using ePBS would be optional for Validators,to truly optimize their income, it is likely that many Proposers would choose to continue using existing flows, ruining many of the projected enshrinement efforts.

Introducing the Block Negotiation Layer (BNL)

The Block Negotiation Layer (BNL) is meant to be a separate layer of distributed services, responsible for accepting, curating and publishing the next correct block for Ethereum.

The BNL uses a “Proposer Intent" idea - a description of block content that allow every proposer to specify what constitutes a valid block to them (i.e. what type of block they will/won’t propose). To enable proposers to specify specific intents, such as compliance with specific regulations, specific transaction ordering rules, or any number of other intents.

Some form of an intent mechanism is needed not only for block creation, but also for the ability to validate externally created blocks. The offload of some validators’ responsibilities to another layer may in future benefit in better validation performance.

This new layer aims to proportionally reward the efforts and infrastructure costs of block building process providers. It incorporates the notion of the distributed network governance widely used in other blockchains - by allowing ETH holders to express their trust in particular service operators by delegating their assets to its account. The distributed nature of block negotiation guarantees fault tolerance and observability by default - as it’s embedded in the process.Finally, by protecting block contents on the separate layer, BLN allows builders to remain unregulated.

Why anchor Relays into the protocol?

In a post-Merge world, Relays play a pivotal but undervalued role; They enhance the ecosystem’s efficiency and economic security while enabling measurable increases in Validator rewards. Yet, despite the critical role Relays play, they currently lack any economic incentives and are effectively operated as public goods. Relays are 100% risk with 0% reward. This disincentivizes participation in the Relay network resulting in further centralization across the network. Of the new Relays that have popped up in the past few months, none have managed to capture an appreciable share of the network.

On top of that, the MEV-Boost network is highly dependent on the few Relays that currently exist. They require 100% uptime because if they don’t do their job, you have missed slots. And, as intermediaries protecting Validators from malicious Builders and vice versa, even small issues at the Relay network can have serious consequences, as seen in the Sandwich the Ripper malicious Validator attack. Relays matter, but the economics must change dramatically if we’re going to make real progress on decentralizing core infrastructure.

Ongoing ePBS research is reinforced with many other proposed mechanisms for preventing attacks, fraud, or helping with network stability (e.g. MEV Smoothing, MEV Burn). Some edge cases still assume the existence of Relays, but the original goal of ePBS is to remove it.

However, because ePBS would be an optional choice for Validators—to optimize their income, it’s likely that many Proposers would just choose to continue using the existing Relay/Builder flow, ruining any enshrinement efforts.

This is why, instead of further reinforcement of the ePBS ecosystem, this proposal explores the potential of incorporating a separate “block curation” layer into the network. This would allow Builders to remain unregulated by the network, for Proposers to control the contents of their blocks, and for Relays to be fairly incentivized for their work and become even more useful to the ecosystem.

Proposer Intents

Before diving into the BNL model, we should first describe the concept that is essential to this work; Proposer Intents - a list of all allowed block parameters that a given proposer imposes to include in their block.

Without a singular place that holds all validity rules, it would not be possible to validate the block.

In MEV-Boost - Relay’s primary role, after returning the highest blinded bid, is checking for the bid validity. The definition of validity is fluid and varying between Proposers. One Proposer may want to only propose blocks that align with their region’s regulations, while another may like to optimize its revenue by all means. As Intents are meant to be private, they can represent the need of a given proposer.

As the name ‘Proposer’ states, the proposing validator should have a say in what is inside the block that they’re trying to propose. In the currently existing implementations of the MEV-Boost PBS, there is no definitive method for a Proposer to ensure that their intent is executed without just building their own blocks. Instead, the proposer’s role is restricted to blindly signing the block header given to them from MEV-Boost.

Some way of formalizing Proposer Intents on-chain, would be heavily beneficial not only for the proposer (that would be able to state it’s needs), but also for the Block Builders not building needless blocks. As intents are owned by the proposers, there is no way for any other party to impose or enforce any settings upon the network.

Proposer Intent concept is one of the fundamental parts of the BNL, as without it Curators would not be able to validate blocks and present the attestations.

However this concept and the research on the dynamic block validity rules is nothing new in the space.

Great works like PEPC - Protocol-Enforced Proposer Commitments or Eigenlayer’srestaking model for preserving block proposer agency - explore other ways of augmenting proposer ability to define its needs on the proposer-builder scope.

As this work designates some form of intents to be essential, it only presents an abstract concept of a 2-phase intents. Meanwhile, intents can be implemented in many various ways like smart-contracts or as a state on chain. Possibly, even using more sophisticated solutions like PEPC. (research needed).

Regarding Proposer Intents:

1. We define the Root Intent as a set default parameters known from genesis. Parameters that constitute currently valid blocks in the current Ethereum network.
2. Every Proposer Intent should be a derivative of the Root Intent.
3. Every Proposer must create and publish a valid intent - to participate in the block curation process. No intent means that the proposer will create blocks in a different way (e.g. local block building).
4. It should be possible to stack and derive intents. So intent templates (”configurations”) can be created and derived.
5. Intents published must not take effect immediately. There should be a minimal delay time (or number of blocks) for other parts of the ecosystem to read and apply the published information.
6. Intents should be stored on-chain so that the information would be unambiguous and freely accessible to every party.
7. The easiest way to do this would be through smart contract storage that would make it easy to allow intent registration and management.
8. A more complex solution could be based on side-chain mechanisms or consensus layer participation.
9. The easiest way to do this would be through smart contract storage that would make it easy to allow intent registration and management.
10. A more complex solution could be based on side-chain mechanisms or consensus layer participation.
11. Intents allows proposers to state participation and readiness of different block-building techniques and anti-censoring solutions (e.g. forward inclusion lists).
12. As soon as the proposer is able to decide on a state (parent block hash) based on which they would like to propose the next block - they should publish the signed final intent via the gossip layer.
13. Without the final intent, the block cannot be published.

14. Producing different final intents for the same slot should be a slashable behavior.

15. Final intent gives proposer a clear observable method to attach transactions for forward inclusion list based on the previous space left - and at the same time not profit from that forward-inclusion-lists

We can divide proposer intents into two groups: 1) intents that can be stated far ahead of time; and 2) intents that, regardless of the proposer's best intentions, may only be presented during the block building process (we call these 'Final Intents').

Below I have drafted some possible parameters to set for intents and final intents

```
type Intent struct { ValidatorAddress [AddressLength]byte // hex address Version uint64 // Block Parent
[IntentAddressLength]byte // Identifier of a parent intent

BlockCreationMode   // No-choice,MEV,NO-MEV,PartialBlock,MEV-Cancellations etc..

MaximumGasAllowed       big.Int
MaximumBlockSize        uint64
ExpectedProfit          bool

FeeRecipientValidation  FeeRecipientValidation
FeeRecipient            [AddressLength]byte // hex address

InclusionListSpaceLength uint8 // the space for head of the block transactions
InclusionListPosition    // Front, Back

BlacklistedAddresses     [][AddressLength]byte


Active      bool
Time        time.Time // Also Salt

}

type SignedIntent struct { Intent Intent Signautre []bytes }

type FinalIntent struct { ParentHash [HashLength]byte InclusionListTransactions [][]byte Time time.Time // Also Salt }

type SignedFinalIntent struct { FinalIntent FinalIntent Signature []bytes }
```

Curators - A new, better form of relays

Curators are meant to operate between the Proposer and Block Builder - in the same place Relays do now. However, their role is widely different from that of Relays under MEV-Boost. The fundamental difference is in their distribution and the ability for network actors to direct trust on curator nodes via staking.

Among curator nodes, Curators are meant to negotiate the best next block based on the slot's Proposer Intent.

The concept of a Curator is simple:

During the previous slot lifetime, distributed Curators receive potential blocks from Block Builders and test them against the Proposer Intent.

Before the deadline, Curators present the highest valid blinded bids to each other and elect the highest bid.

A small semi-randomly selected group of Curators is chosen to become the slot committee. Curator with the winning bid cannot be part of this committee. The frequency of how often, on average, a Curator is chosen to participate in the attestation committee should be based on the amount it has staked (see: Curator Staking).

After the election, every member of the committee receives the biggest unblinded block and attests to the block validity; This is the deadline for the Proposer to present its Final Intent.

There are only two potential outcomes for the attestation[1]:

1. The Block is valid per the supermajority of attesters.

2. All attesters that voted correctly are rewarded.

3. The Curator that presented the highest bid is rewarded.

4. All attesters that voted correctly are rewarded.

5. The Curator that presented the highest bid is rewarded.

6. The Block is invalid (not compliant with the Proposer Intent)
7. All attesters that participated in attestation are rewarded.
8. The Curator that presented the invalid highest bid is slashed.
9. Every Curator should have the ability to check other bids post-factum and get rewarded if any proposed bid was invalid (dishonest curator).
10. All attesters that participated in attestation are rewarded.
11. The Curator that presented the invalid highest bid is slashed.
12. Every Curator should have the ability to check other bids post-factum and get rewarded if any proposed bid was invalid (dishonest curator).

How a Curator publishes a block after valid attestation could be implemented in one of two ways:

1. Publish the valid attested block to the network without proposer's participation. As the entire process is protected by having Curators stake ETH, dishonest Curators risk being slashed.
2. The proposer joins the committee where it is presented with the attestation results. It can then choose to propagate the signed header to the Curator nodes which should publish the block to the network. Note: Proposers are able to see the signed headers directly after the election phase as the bidding process is public.

Example bid structure:

```
type Bid struct { BlockValue big.Int BlockHash [HashLength]byte

CuratorAddress [AddressLength]byte
BlockAuthorAddress [AddressLength]byte
Sequence uint64 // local to the curator

Slot uint64
ParentHash [HashLength]byte

}
```

[1] For the sake of simplicity let's not explain dishonest attesters - this would be covered by the post-process check

Curator Staking

In the current MEV-Boost system, Relays are trusted not to leak or steal any private block data, not to propose invalid blocks, and to remain performant enough to carry all its functions - but there is nothing enforcing the right behavior. The Block Negotiation Layer designates Curators to carry out many of these same duties (and more), but this time trust is backed up by staking mechanisms.

In BNL, trust materializes in the amount of staked ETH delegated to the organization running the Curator. This model embraces the true meaning of the proof-of-stake model, enabling the wider Ethereum ecosystem to indicate their trusted parties by allowing them to deposit stake into a Curator (and share in their profit). This idea allows ETH holders (i.e. the network) to indicate personal preference of who they believe would be the best party in the community to be Curators.

Curator staking should follow the following:

- Curators protect the ecosystem. It is in the Validators' and Block Builders' best interests to pick trusted parties that they would like to work with.
- The total Curator's stake should be compounded with multiple buckets where only particular parties should be able to stake.
- The model should embrace different levels of participation and engagement from different actors. For example, the amount staked by Validators should weigh much more than any other party.
- The model should embrace different levels of participation and engagement from different actors. For example, the amount staked by Validators should weigh much more than any other party.
- Curator should not be able to propose a bid higher than its own stake — the mechanism should protect both validators and builders from different fallacies. This is why the Curator needs to hold more stake than the block it would propose to cover for potential misbehavior.
- Validators should have a bigger say in who's going to create their blocks than any other group. Whatever mechanism would be chosen, the frequency of how often a Curator is elected to be the attester must be proportional to its level of trust (i.e. to the delegations for this particular curator).

- Every other participant of the ecosystem should also have the ability to express their trust in a particular Curator and to directly or indirectly increase the amount of stake on a Curator. To prevent the hegemony of validator favorable curators, the network should proportionally allow the rest of the ecosystem to allocate the assets onto their favorable Curators.

As Curators are rewarded for their work, part of their reward should also be shared with their delegators as dividends for helping the network elect honest actors.

Pay Day - Understanding curator incentives and rewards

In comparison to the current PBS model, BNL drives innovation and begins Curator competitiveness. Current ecosystem metrics are heavily gameable and there are no incentives to drive innovation and improved performance for Relays because they're not currently rewarded.

BNL solves that problem by rewarding the Curator that presented the highest bid with the same amount as the other attesters for the slot. This means being the best, fastest Curator, could result in you winning every slot. This would result in the ecosystem seeing better quality and value of blocks, much faster as incentives push Curators to constantly seek improvement.

There are several potential sources for these rewards. Even though I can't cover all of them in this article, I believe the most obvious and organic one would be a percentage of the block value as payment for the services. This should go to the curation layer, and be distributed among the participants.

The economics of this distribution could look something like:

1. Curators are rewarded for:

2. With the same amount - Block validity attestations (committee participation) and returning the best bid

3. With varying amounts - For finding bad blocks that were proposed or included on auction.

4. With the same amount - Block validity attestations (committee participation) and returning the best bid

5. With varying amounts - For finding bad blocks that were proposed or included on auction.

6. Rewards should depend on the amount of work curator needs to do to validate Proposer Intent

7. E.g. A Proposer's need for complex computational work such as the filtering of OFAC designated addresses should be rewarded more (i.e. more value would be taken from the block)

8. E.g. A Proposer's need for complex computational work such as the filtering of OFAC designated addresses should be rewarded more (i.e. more value would be taken from the block)

9. Rewards should depend on the bid effectiveness

10. The model should prefer having less bids - the less bids curator present the more reward it should get

11. The model should prefer having less bids - the less bids curator present the more reward it should get

12. Rewards should depend on bid value over time

13. The model should prefer higher bids faster - the earlier winning block should be presented the bigger reward

14. The model should prefer higher bids faster - the earlier winning block should be presented the bigger reward

The Big Bad Wolf - How we negate malicious curators from abusing the BNL

Incentives attract bad actors. There are several mechanisms that can be put in place to mitigate this such as attestation and slashing mechanisms. Slightly different versions of these for block validation already exist in the ecosystem, so the concept is nothing new.

Although it's not the goal of this article to cover all possible scenarios in detail, I would note that the heaviest punishment should be carried out only in situations where a Curator willingly acts against the ecosystem. Hence, slashing should be conducted, when:

- A Curator presented an invalid block;

- A Curator willingly presented an invalid attestation.

Another threat is malicious acts against the gossip layer itself. As many parties may participate in the Curator network, it would be important to create some p2p connectivity guidance. No one should supervise connections, but some parameters of BLN can be used for attack prevention.

1. Staking itself is a carrier of great trust—Curators then may prefer to connect to Curator nodes with higher stake, organically excluding the ones with none.
2. For Attestations, committees may use an additional secure pubsub channel, joinable only by Attestants and Proposers.
3. Curators may prefer local peering for settling the highest value faster.
4. Depending on the chosen algorithm, the selection of maximum value is possible to reasonably distribute. However the nature of the system also allows the implementation of sharding or partitioning of the bidding process.

Some of the processes may also be utilized by the beacon network's added features.

Block Builders in This Brave New World

This proposal only formalizes the layer of Curators that guard the different aspects of block validity. We purposely left the role of Block Builders unspecified as the ecosystem's only concern should be to have valid blocks in a timely manner. The Block building process may benefit from many unforeseen improvements. Putting any constraints on the Block Builder layer makes it much less technologically diverse.

Instead of block proposals, Block Builders should focus on the block building process, delivering the highest, diverse, non-exclusionary blocks. As much as the Block Negotiation Layer encourages Curator competitiveness through a 'highest bid reward' - it also motivates the development of different block building and block submission solutions, over varying Curator codebases.

The Proposer Intent mechanism featured in this document not only gives the ability to state what kind of blocks that a proposer is interested in, but at the same time saves a lot of work for Block Builders. By not doing unnecessary operations, Block Builders are able to save processing power, lowering cost and their carbon footprint. For example, Block Builder would not waste resources building blocks containing OFAC designated addresses if the current Proposer Intent calls for OFAC compliance. This is happening right now, wasting valuable time and resources on all sides because those payloads are never considered.

The bidding process in BNL is public (distributed, gossiped). By listening to the channel Block Builders gain visibility, giving them the ability to openly and actively adjust their bids. Creating such a process on the current system is extremely hard as a single Relay would not have information about the global highest bid. This encourages centralization and is problematic for the Relay (http request flooding, websocket maintenance).

Enabling the future

Having a distributed block validation layer of trusted parties that are able to actively verify themselves opens up a lot of new directions in the growth of the ecosystem.

For example, it enables the possibility of block building segmentation by creating multiple committees. Some possible scenarios:

1. Block size - depending on the demand in future it may make sense for Block Builders to only build a part of the block.
2. Latency + Inclusion - for bigger network dispersion and high inclusion of transactions originating from different regions - Curators may like to form per-region committees - building parts of the block. This way we can create an inclusive environment also supporting less developed regions and fighting builder-relay latency clustering.

The intent mechanism can be created in a way that allows the Proposer to decide what kind of building process rules (and ethics) it may like to use for a particular task. And it is completely possible because the Proposer Intent is known well ahead of time.

As in the BNL where almost all block building activities are moved away from Proposers, It may make sense to also move the public mempool to the Curator layer, as well. This way all the hard work of processing mempool transactions may be shifted into a Curator layer that may also benefit from having closer access to public information. The leak of a mempool's private or public transactions in such a case could also be a slashable action to protect the wider ecosystem.

The Block Negotiation Layer in Practice

Several fundamentally different bidding processes and algorithms may be developed for this proposal. The goal of the one presented below is to show a general idea in an uncomplicated way. It's not meant to solve all the nuances or shortcomings of the current process.

The process should consists of following steps:

1. Prior to the slot:
2. The Proposer should publish its intent to the network.

3. The algorithm should choose a subset of curators to become slot attesters. Similarly to Validators, there should be a changing committee of curators that would be responsible for attestations. The per slot committee should be picked semi-randomly. The frequency of how often a curator is being picked for attestation should be relative to the amount of stake it has.

4. The Proposer should publish its intent to the network.

5. The algorithm should choose a subset of curators to become slot attesters. Similarly to Validators, there should be a changing committee of curators that would be responsible for attestations. The per slot committee should be picked semi-randomly. The frequency of how often a curator is being picked for attestation should be relative to the amount of stake it has.

6. Block Builders and Curators should read the next slot's Proposer Intent.

7. Based on the agreed state[2] - Block Builders should start building better blocks that comply with the Proposer's intent.

8. Based on the agreed state[2] Curators should only accept blocks that comply with the Proposer Intent.

9. Curators connect with each other using gossip protocol

10. Bidding Phase

11. Curators are listening for better blocks, and store Local View of all received messages.

12. As soon as the Curator successfully validates that a block complies to the Proposer Intent and is bigger than any previous bids, it publishes the bid to the gossip layer.

13. Bidding finishes after a hard stop at a precise time in the slot.

14. (optionally) - The best encrypted and signed blocks are being sent to other Curators publicly. The strength of encryption doesn't matter as it only needs to hold for a few seconds. Even after bruteforce it should not be possible to apply logic based on any intel gathered.

15. Upon finishing - The Curator that won the auction must publish the best bid to the attesting committee or return the block upon request.

16. (optionally) - upon the signed requests from the committee members, the Curator returns a decryption key for the encrypted block.

17. (optionally) - upon the signed requests from the committee members, the Curator returns a decryption key for the encrypted block.

18. Curators are listening for better blocks, and store Local View of all received messages.

19. As soon as the Curator successfully validates that a block complies to the Proposer Intent and is bigger than any previous bids, it publishes the bid to the gossip layer.

20. Bidding finishes after a hard stop at a precise time in the slot.

21. (optionally) - The best encrypted and signed blocks are being sent to other Curators publicly. The strength of encryption doesn't matter as it only needs to hold for a few seconds. Even after bruteforce it should not be possible to apply logic based on any intel gathered.

22. Upon finishing - The Curator that won the auction must publish the best bid to the attesting committee or return the block upon request.

23. (optionally) - upon the signed requests from the committee members, the Curator returns a decryption key for the encrypted block.

24. (optionally) - upon the signed requests from the committee members, the Curator returns a decryption key for the encrypted block.

25. Attestation phase

26. The Curator that won the auction cannot be a part of the attestation committee for that block.

27. Every Curator in a committee needs to test if a block is correct based on the Proposer Intent.

28. During the attestation phase, the Proposer should supply the valid parent hash; it may not be mandatory if a block was built on multiple possible parents.

29. After testing, Curators will vote on block inclusion.

30. (optionally) - There can be multiple committees for top bids in case the first one was missed or ambiguous.

31. The Curator that won the auction cannot be a part of the attestation committee for that block.

32. Every Curator in a committee needs to test if a block is correct based on the Proposer Intent.

33. During the attestation phase, the Proposer should supply the valid parent hash; it may not be mandatory if a block was built on multiple possible parents.

34. After testing, Curators will vote on block inclusion.

35. (optionally) - There can be multiple committees for top bids in case the first one was missed or ambiguous.

36. Proposal phase

37. By the proposal phase the Proposer MUST send its Final Intent with the parent block hash.

38. The top, valid attested block should be chosen.

39. Block propagation possibilities:

40. Community publish - because the block was attested by a committee, and processed (approved) by many nodes under the potential for a slashable event, it should be safe to treat the block as a safe next block.

41. Proposer publish - keeping a similar flow to how things work right now, a Proposer may receive signable blinded block header through the gossip protocol. The Proposer should also be presented with the attestations and, based on that, should broadcast a signed header to the members of the committee. Then, all the Curator nodes should publish the block to the beacon chain.

42. Community publish - because the block was attested by a committee, and processed (approved) by many nodes under the potential for a slashable event, it should be safe to treat the block as a safe next block.

43. Proposer publish - keeping a similar flow to how things work right now, a Proposer may receive signable blinded block header through the gossip protocol. The Proposer should also be presented with the attestations and, based on that, should broadcast a signed header to the members of the committee. Then, all the Curator nodes should publish the block to the beacon chain.

44. By the proposal phase the Proposer MUST send its Final Intent with the parent block hash.

45. The top, valid attested block should be chosen.

46. Block propagation possibilities:

47. Community publish - because the block was attested by a committee, and processed (approved) by many nodes under the potential for a slashable event, it should be safe to treat the block as a safe next block.

48. Proposer publish - keeping a similar flow to how things work right now, a Proposer may receive signable blinded block header through the gossip protocol. The Proposer should also be presented with the attestations and, based on that, should broadcast a signed header to the members of the committee. Then, all the Curator nodes should publish the block to the beacon chain.

49. Community publish - because the block was attested by a committee, and processed (approved) by many nodes under the potential for a slashable event, it should be safe to treat the block as a safe next block.

50. Proposer publish - keeping a similar flow to how things work right now, a Proposer may receive signable blinded block header through the gossip protocol. The Proposer should also be presented with the attestations and, based on that, should broadcast a signed header to the members of the committee. Then, all the Curator nodes should publish the block to the beacon chain.

51. Post process

52. There should be another committee designated (and picked post-factum) for checking on-chain blocks, after the block publishing. This committee should check for block validity with the proposer intent. If the block is invalid, the Curator who presented the block and all Curators who attested incorrectly should get slashed. This round should only be rewarded if misbehavior is found.

53. (Optionally) - Every Curator participating in the bidding process has to publish its encrypted block (or random encrypted block) to other nodes. Those blocks may also be checked for bid validity so the bidding process is not artificially bloated.

54. There should be another committee designated (and picked post-factum) for checking on-chain blocks, after the block publishing. This committee should check for block validity with the proposer intent. If the block is invalid, the Curator

who presented the block and all Curators who attested incorrectly should get slashed. This round should only be rewarded if misbehavior is found.

55. (Optionally) - Every Curator participating in the bidding process has to publish its encrypted block (or random encrypted block) to other nodes. Those blocks may also be checked for bid validity so the bidding process is not artificially bloated.

[2] The agreed state means the state that everyone should consider valid ahead of time for the new block creation. It can be the first seen state, it can be all the possible states.

TLDR: Why BNL?

- The model allows Curators to be fairly incentivized for the honest work that they're doing now.
- It's distributed, so it's impossible for one curator to do something bad to the ecosystem.
- It's observable, the whole blinded bidding process is actually "public".
- It enables codebase competitiveness - the best curator with the best bid would be rewarded additionally to the randomly chosen attesters.
- There are no rules, guards or regulations on the block builders - a distributed layer of curators would also keep an eye on all of that and make sure that everyone is getting paid for their work being correctly done.
- Because of the Intents everyone knows what kind of block and state proposer expects. The carbon footprint is also smaller as builders do not waste resources building blocks in vain.
- It is a non-exclusionary scheme for Proposers that must comply with certain regulations, such as OFAC sanctions. Now Proposers can clearly state their will. By bounding the Proposer block reward with Curator reward, we can create a mechanism where Proposers may waive a part of their reward for Curators as compensation for the sanctions check. This also motivates bigger rewards for Proposers that don't need computationally intensive blocks as these Proposers would not need to wave part of their reward.
- Because intents exist, it would be possible to place most of the rewarding logic into the smart contract - that would programmatically distribute rewards to the proper parties.

Related work

After reading about BNL, You may also be interested in learning about some other, great work that's happening right now to improve the ePBS ecosystem:

- PEPC Unbundling PBS: Towards protocol-enforced proposer commitments (PEPC)
- PTC Payload-timeliness committee (PTC) – an ePBS design
- MevBoost++ MEV-Boost+/++: Liveness-first Relay Design - MEV - EigenLayer Research