Thank you Justin Drake and Ryan Sproule for the help.

tl;dr:

Allow users to specify their preferred penalty when requesting a preconf, enabling the market to naturally establish preconf cryptoeconomic security parameters, rather than setting parameters upfront.

As the community settles on a design for preconfs, a critical choice arises: how can we ensure crypto-economic security for preconfs? Specifically, what incentives exist to prevent safety or liveness faults? I'll present a high-level overview of the current solutions before proposing an alternative.

Here are the current mechanisms (they can be used in combination):

1. Basic Slashing:

If a proposer is responsible for a safety or liveness fault, they are slashed. * Open Question:

How much should be slashed, and what amount of stake should the proposer put up?

1. Open Question:

How much should be slashed, and what amount of stake should the proposer put up?

1. Freezing:

The proposer's stake is frozen, causing them to lose the time value of their money. *Justin Drake suggested during the Ethereum sequencing call #7 that this approach could help ease the adoption of the preconf protocol since preconfs introduce new behaviors the market needs to adjust to.

- Open Question:

How much stake should be frozen, and for how long?

1. Justin Drake suggested during the Ethereum sequencing call #7 that this approach could help ease the adoption of the preconf protocol since preconfs introduce new behaviors the market needs to adjust to.

2. Open Question:

How much stake should be frozen, and for how long?

1. Dynamic Reputation Slashing:

Each fault by a validator results in a progressively stricter penalty; for instance, they might be slashed more or locked up longer. * Open Question:

What should the penalty curve look like? Should it be time-based and reset after a period of honest behavior?

1. Open Question:

What should the penalty curve look like? Should it be time-based and reset after a period of honest behavior?

1. Insurance:

Proposers must compensate users whose preconfs fail due to faults. Effectively, users' preconfs are insured. * Open Question:

How much insurance should be offered?

1. Open Question:

How much insurance should be offered?

All these mechanisms require us to know ex ante what preconf users want. Inevitably, this will be opinionated and lead to deadweight loss, as some users who might want a preconf could feel uncomfortable with the setup. Moreover, some proposers might feel uncomfortable with the parameterization and choose not to offer preconfs. The best solution is to allow users to work with proposers to agree on the appropriate level of crypto-economic security.

My Solution: User-Defined Penalties

Users should be able to specify their desired level of crypto-economic security by attaching a penalty structure to their preconf. This structure will detail the consequences of a fault.

For instance:

// Here, users can define a penalty associated with any specific fault, // and the system is generic enough to allow for arbitrarily complex rules. struct PreconfAgreement { faults: Vec>, }

struct Fault { condition: C, penalties: Vec

, }

trait Condition { fn should_penalize(...) -> bool; }

trait Penalize { fn penalize(...); }

Note: There is a DoS vector associated with unbounded compute when evaluating conditions. Some gas metering should be used, or conditions should be constructed as succinct statements (e.g., a SNARK).

This solution is unbiased and allows the market to determine the appropriate parameters naturally. Users can decide the level of security they want rather than leaving it up to the protocol to estimate, while proposers can choose their risk-reward profile. Heavier penalties will likely result in higher costs for users.

Complexity Concerns:

- Proposers' Perspective:

With preconfs, we already assume that proposers (or their gateways) are sophisticated, and giving them the ability to manage their own risk profiles should benefit them. Inexperienced proposers can set a simple threshold for the maximum penalty they are willing to incur and, as they gain experience, adjust it more systematically.

- Users' Perspective:

This approach shouldn't add complexity, as wallets can easily abstract the penalty decision, much like they abstract gas fee choices. Fine-grained choices can be offered as an opt-in feature for more advanced users.