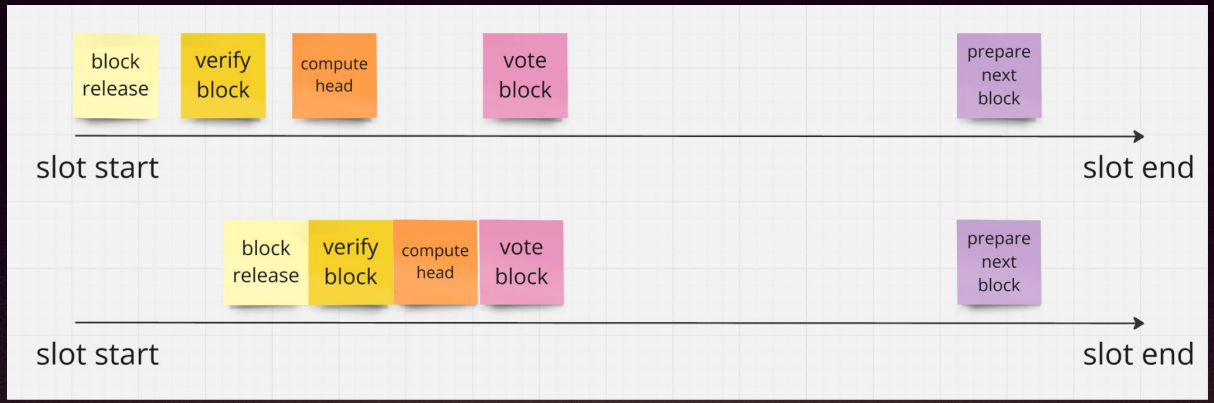
Inevitable Tradeoffs of Inclusion Lists

Terence

Offchain Labs

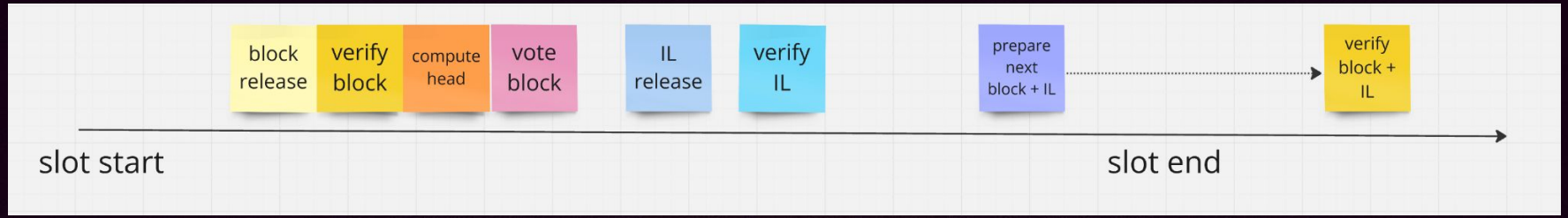
Ethereum Slot Interval Constraints

- Slot: 12 seconds
- Proposer: propose new block on top of head
- **Everyone**: verify block and compute head
- Attesters: attest to head block
- Aggregators: aggregate attestations
- Next proposer: build on top of head



Additional constraints with Inclusion List

- Setup: Same slot + FOCIL
- Inclusion list proposers: propose inclusion lists on top of head
- Everyone: gossip and verify inclusion lists
- Next slot builder: construct inclusion lists in the block
- Attesters: verify the inclusion list is satisfied



Inclusion List Parameters

- **What?** Inclusion list message size. How much overlapping?
- **Who**? Size of the inclusion list committee
- **How**? Satisfactory rule
- **When**? Timing of the releases
- **Concerns**?
  - Increase bandwidth and compute for nodes
  - Proposer insufficient resource to build
  - Attester insufficient time to verify

Open Questions

- How does the inclusion list interact with other protocol upgrades (ePBS block/slot auction, peerDAS)? Uniform fork choice rule?
- How does the inclusion list align with account abstraction?
- How can we support inclusion lists with blob transactions?
- How can we better utilize the execution mempool for inclusion list purposes?
- How do we forsee IL-boost or out-of-market constructions?

Thank you!

Terence

Offchain Labs

@terencechain