# Week 8 Research - Protocol Services

## Guest speaker

Barnabé Monnot, from Robust Incentives Group (RIG), Ethereum Foundation

- Joined the RIG at EF in Jan 2020
- Since then, work on
  - PoS consensus
  - EIP 1559/ fee market mechanisms
  - MEV/ PBS
  - General mechanism design & side quests

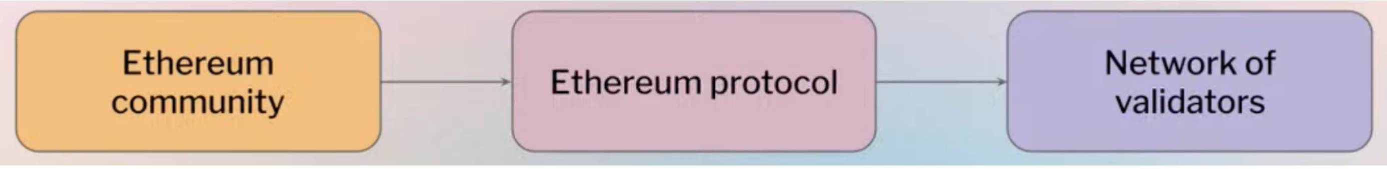
Slides linlk: https://docs.google.com/presentation/d/1TY5-7wTDer4vonYoEaej2aBMukAbXk8xQjYV7vW56RU/edit?usp=sharing

## Summary notes

- Edited by Chloe Zhu
- Online version: https://ab9jvcjkej.feishu.cn/docx/CUTFdy7Yao7lP5xpTkxc0AuknPe

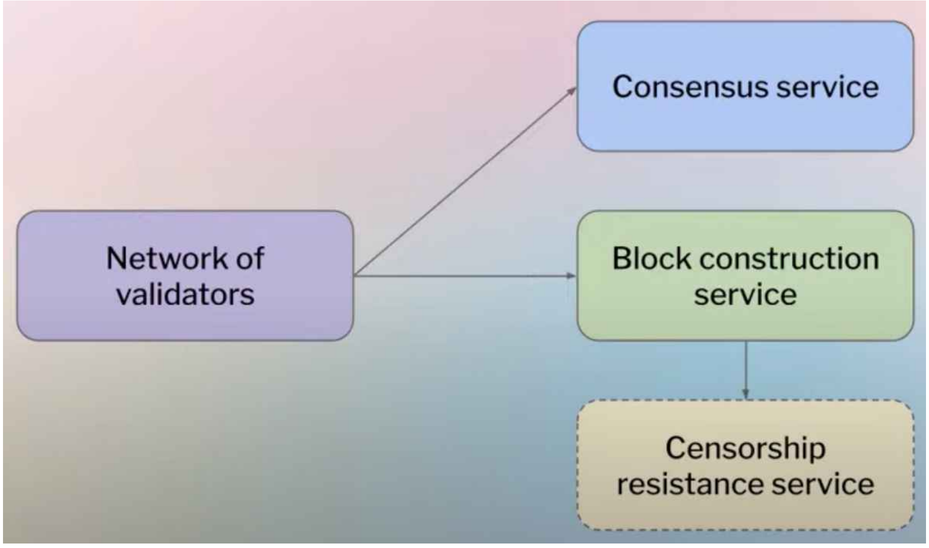
## Seeing like a protocol

- Reference blog: Seeing like a protocol
- Key questions to address
  - Where does protocol credibiility come from?
  - How far does the protocol extend?
  - What should the protocol see?
- TDLR



  - The Ethereum protocol is set up by a diffuse "community", i.e. stakerholders of the protocol. And it uses "rough consensus" for governance.

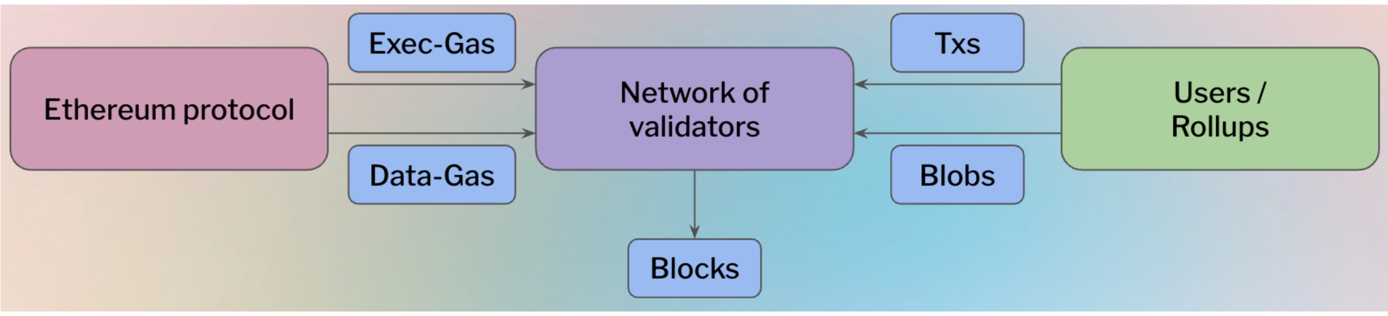
    - The protocol's goal is to **decentralize provision of blockspace for users to achieve max welfare and min rents**.
    - Validators run the protocol.
- Validator services



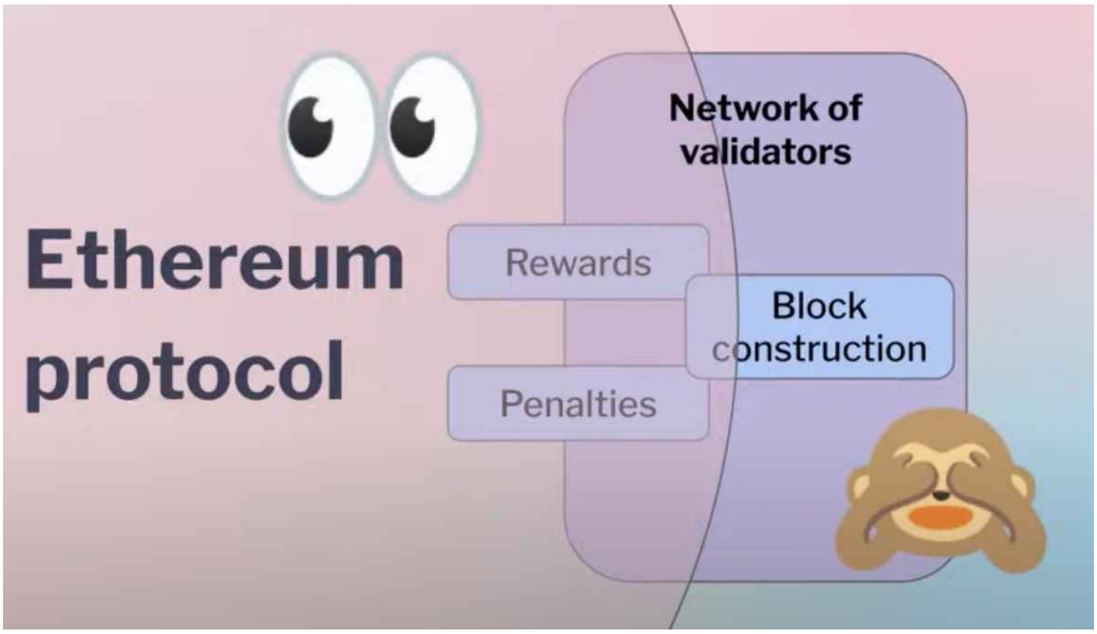
    - Consensus service: Give finality to the chain, make sure everyone agrees on the state & history of the chain
    - Block construction service: Keep adding useful info (eg. txs) to the chain
    - Censorship resistance service: Remain neutral for different parties trying to build blocks on the chain
- The protocol-validator problem
    - Key question: How to make validators achieve the goals of the protocol?
    - Main solution to direct how validators behave
        - Protocol introspection: Obtain credible signals of the environment in protocol state
        - Protocol agency: Respond to signals with updates, rewards, or punishments

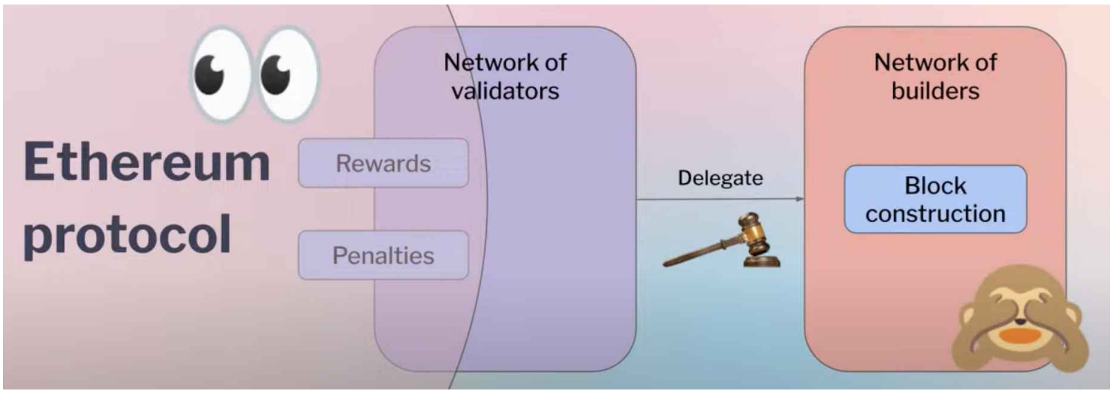
## The block production service

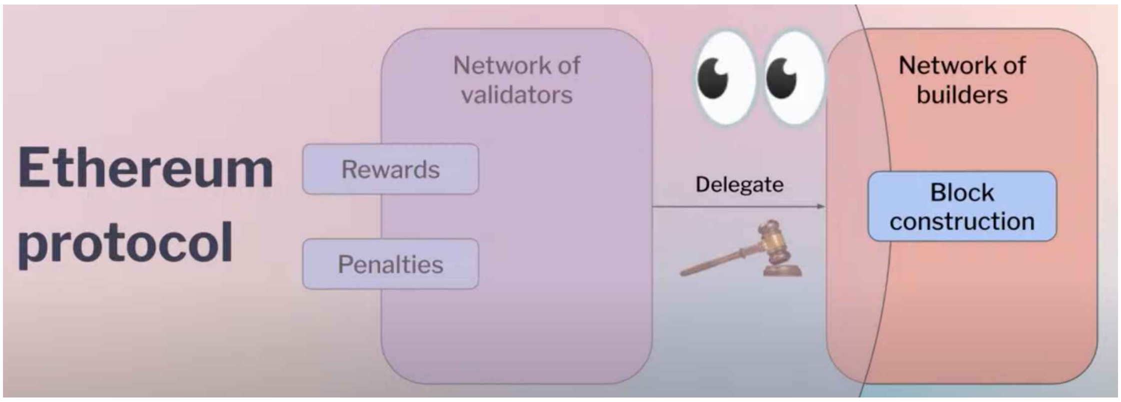
- Validators as resource allocators: Validators can be viewed as the meeting point between the protocol and the users



    - Protocol:
        - Regulate the supply of gas
        - Supply of gas is constrained to guarantee low verification costs
    - Users:
        - Demand for block space
        - The demand is represented by txs
    - Validators:
        - Produce blocks, meeting demand for txs with supply of resources
        - Try to allocate the best use of gas into these txs, and validate th txs
- EIP 1559: Resource introspection
    - A fee market mechanism, where the protocol can know how much demand there is
        - Protocol quotes a reserve price, **dynamic congestion pricing**
    - Demand signal
        - Block target 15 million gas use
        - Block limit 30 million
    - Pricing update rule
        - If gas use > target value, then reserve price increases
- 2-dimensional EIP 1559
    - The problem of EIP 1559: It doesn't bring more people onchain
    - EIP 4844: Separate "exec-gas" and data availability (DA) gas, to max the capacity of the scalling
        - DA: consumed by rollups, L2 solutions secured by Ethereum
- What Ethereum protocol can see and not

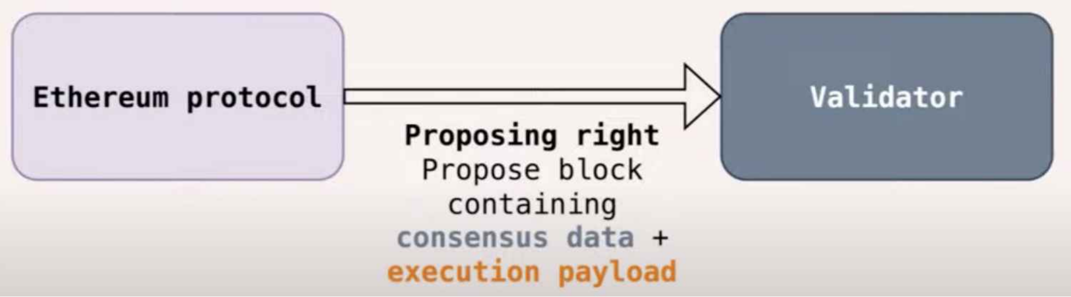






- **What the protocol can see/ control**
  - See what validators do on the consensus layer
  - Has the ability to give reward/ penality to keep validators in check
  - Partially control over the way validators build the block, through EIP 1559 and gas limit
- **What the protocol cannot see/ control**
  - How validators actually build the block
  - How blocks are sequenced -> gives extreme power of validators, which results in MEV (max extractable value)

            - Validators could delegate the responsibility of finding out the most valuable block to a network of entities, known as Builders.
            - Nowadays, c. 90% of the block construction is done by the network of builders.
    - **Debate on the protocol boundary**
        - Should the protocol move forward the boundary to see the dynamics and potentially enshrine parts of the delegation
- Focus on **proposing rights**

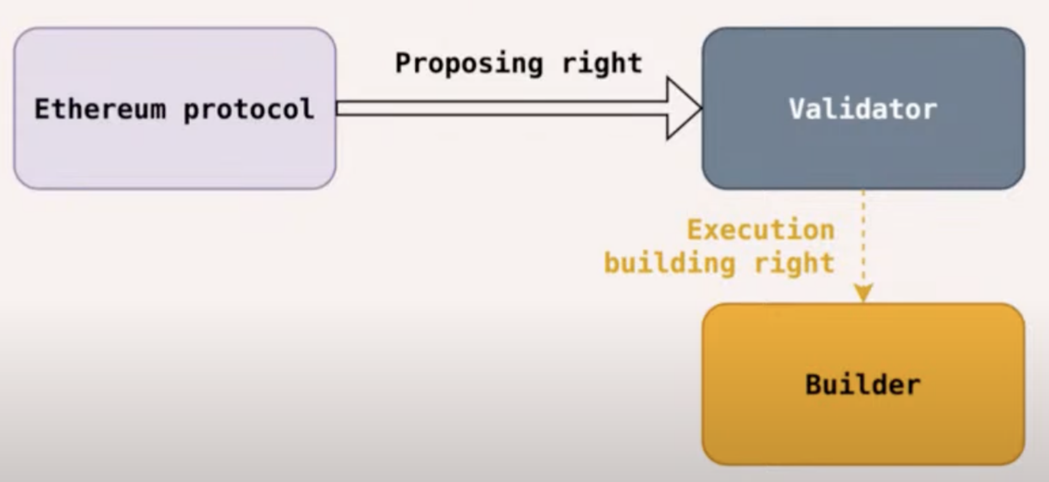


Ethereum protocol

Validator

**Proposing right**
Propose block
containing
consensus data +
execution payload

    - Key question: What rights does the protocol give to the validators?
        - **Ability to propose blocks that contain**: consensus data + execution payload
    - Current dynamics: MEV-Boost (validators call upon builders)
        - Nowadays, **validators outsource the execution building right to builders**, where builders will sequence/ build the blocks



**Proposing right**

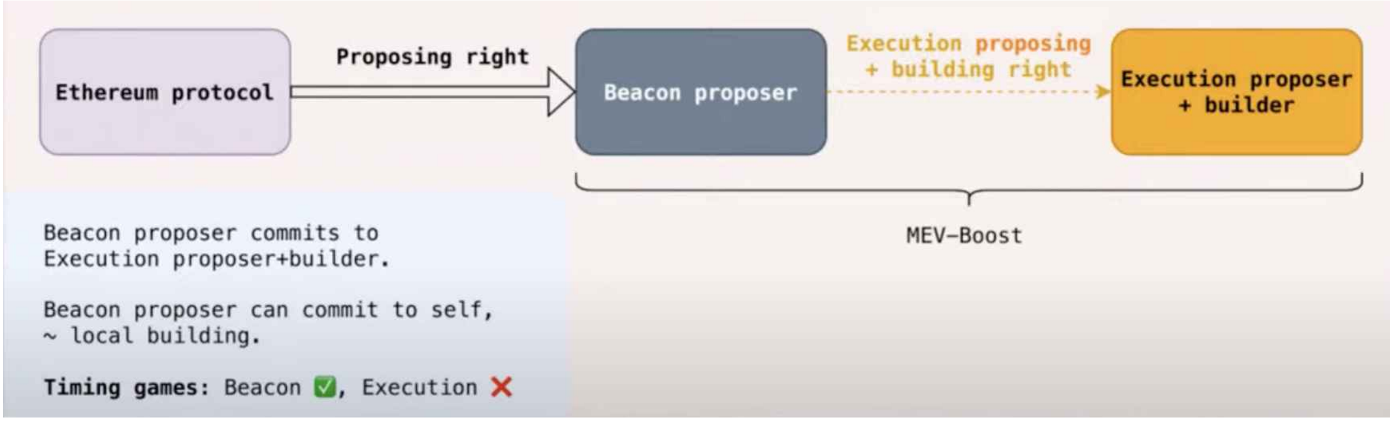
Ethereum protocol

Validator

Execution
building right

Builder

- Block-auction ePBS: Execution-Consensus separation



Ethereum protocol

Proposing right

Beacon proposer

Execution proposing + building right

Execution proposer + builder

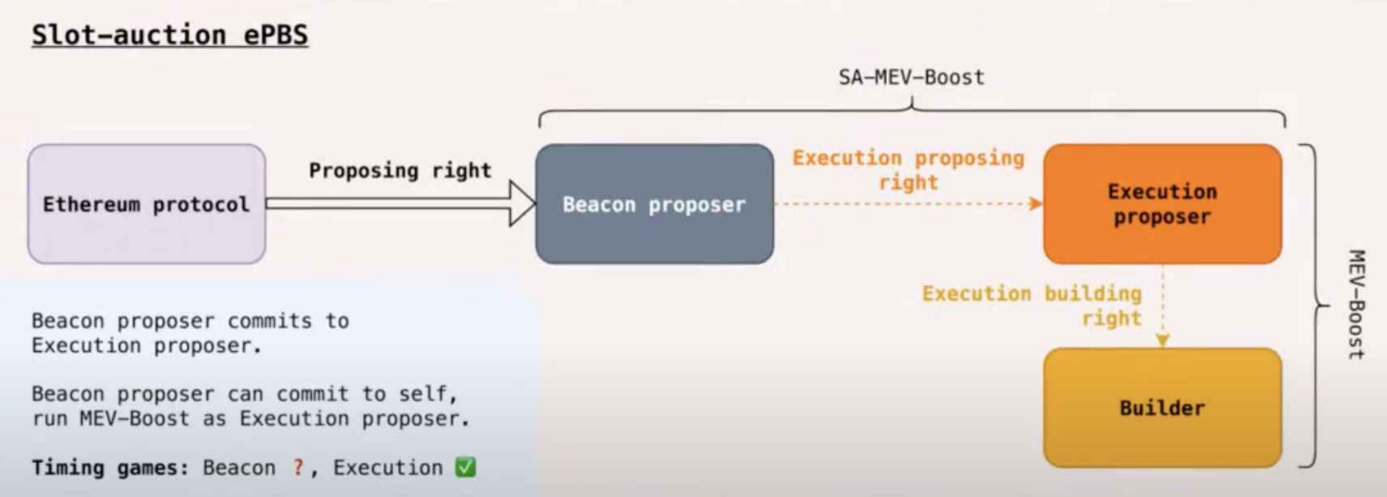
MEV-Boost

Beacon proposer commits to
Execution proposer+builder.

Beacon proposer can commit to self,
~ local building.

**Timing games:** Beacon ✓, Execution ✗

  - Instead of giving the right to propose blocks in one step incl. Consensus data & execution payload, ePBS separates the process into 2 steps
    - The protocol will still give the **block proposing right** to the **validators.**
    - Validators then are able to **commit to an execution proposal** that is going to review the execution payload.
  - The commitment between validators and execution proposer & builder is only known by the latter. **The validator will only get a hash of the tx and won't know what exactly the block contains.**
  - Pros of this approach
    - If the execution proposer & builder goes offline, we can still contain the consensus data, which is not the case today.
    - The commitment the beacon proposer gives to the execution proposer is guranted by the protocol.
  - Cons of this approach
    - It's still possbile for the beacon proposer and execution proposer & builder to use MEV boost and bypass the whole protocol infra
- Slot-auction ePBS



Slot-auction ePBS

SA-MEV-Boost

Ethereum protocol

Proposing right

Beacon proposer

Execution proposing right

Execution proposer

Execution building right

Builder

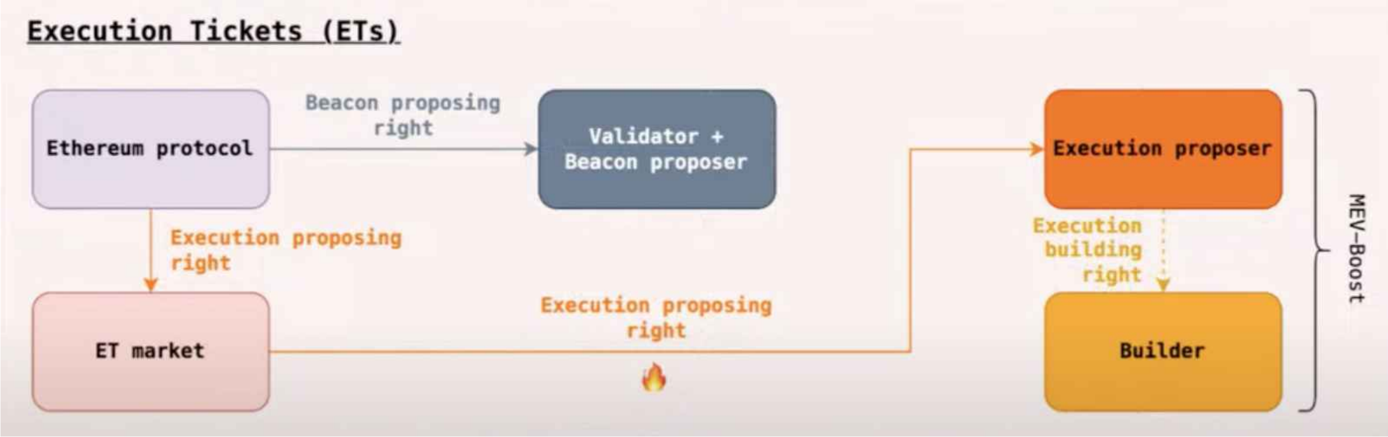
MEV-Boost

Beacon proposer commits to
Execution proposer.

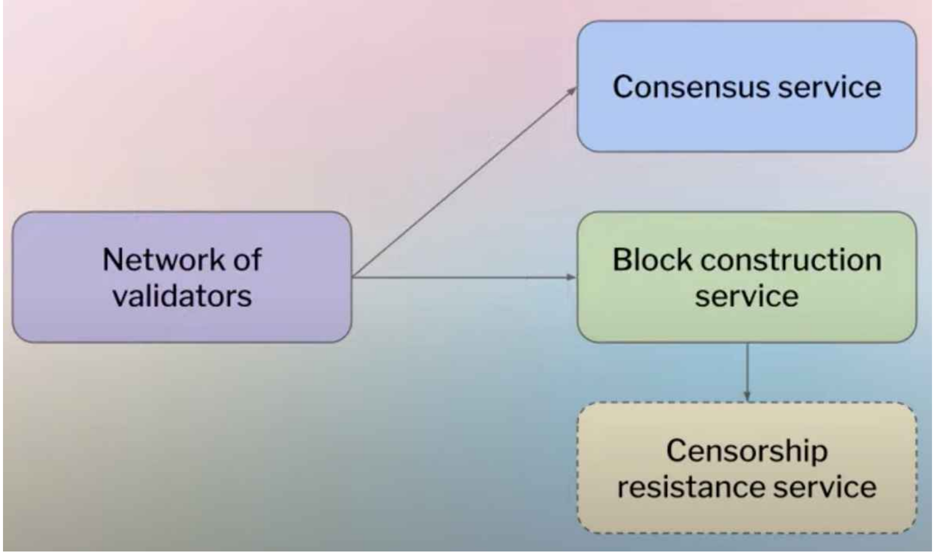
Beacon proposer can commit to self,
run MEV-Boost as Execution proposer.

**Timing games:** Beacon ?, Execution ✓

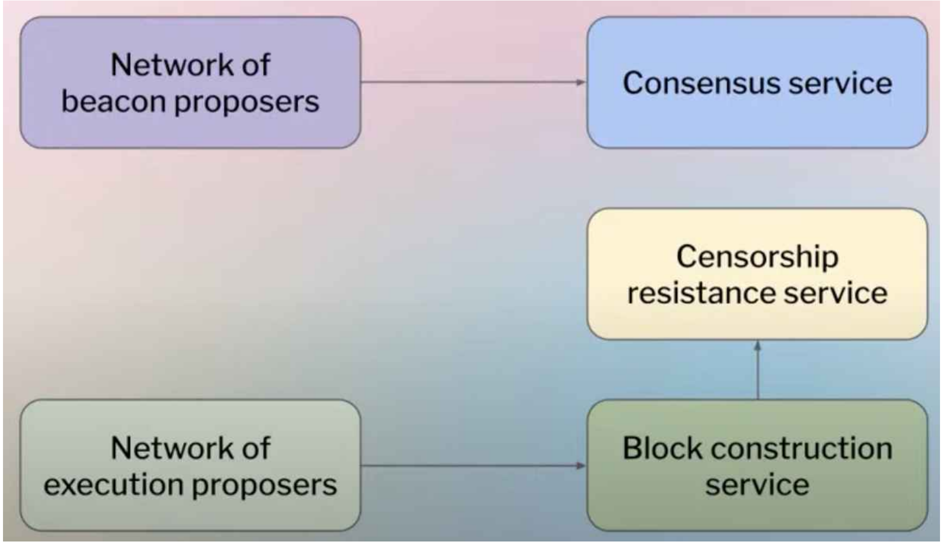
  - The beacon proposer will only commit to the public key of the execution proposer, but it won't commit to the content of the block.
  - When the execution proposer wins the ticket to make the execution payload, it can still resell it to builders
- Execution tickets (ETs)



  - Validators are no longer choosing who the execution proposer is.
  - The exeuction proposer will be chosen by the execution ticket market, which is permissionless.
  - Since the ET market is part of the protocol gadget, the protocol will have the introspection over who comes to the market and how much they are willing to pay.
- Recap of the validator services
  - Originally



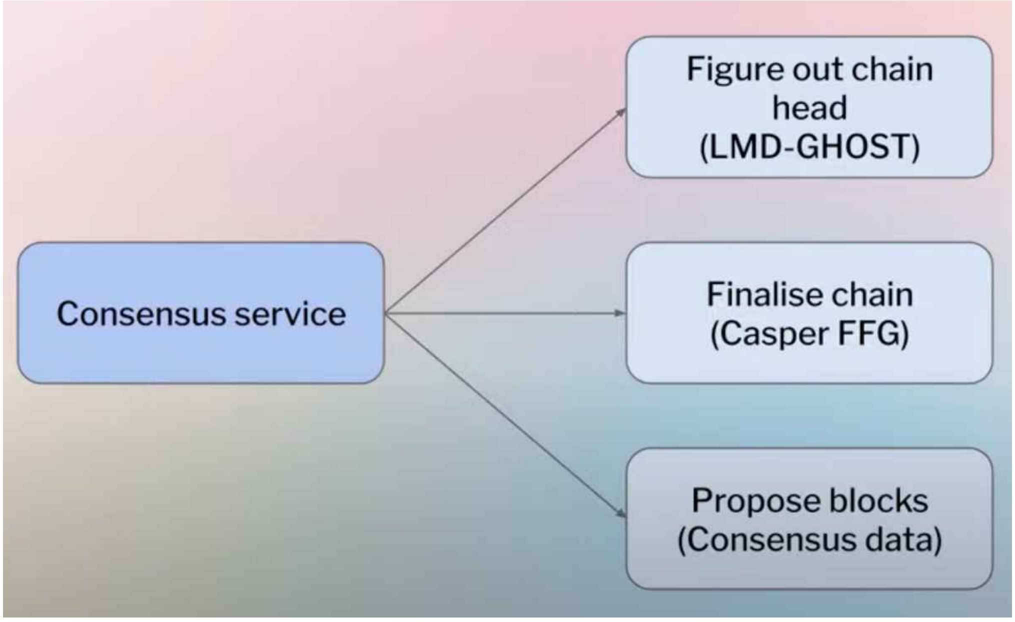
  - Nowadays


Network of beacon proposers
Consensus service
Censorship resistance service
Network of execution proposers
Block construction service

- The issue
    - The execution proposers/ builders are typically more centralized. Currently, they are responsible for providing censorship resistance, which is not ideal.
    - We want a decentralized set of operators for diverse preference
        - When making a block, incl. txs that others dislike
        - When participating in consensus, decoorelation -> stronger resilience

## The consensus service

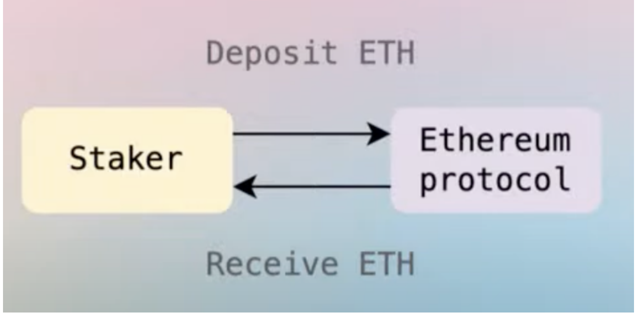
- Consensus service


Consensus service
Figure out chain head (LMD-GHOST)
Finalise chain (Casper FFG)
Propose blocks (Consensus data)

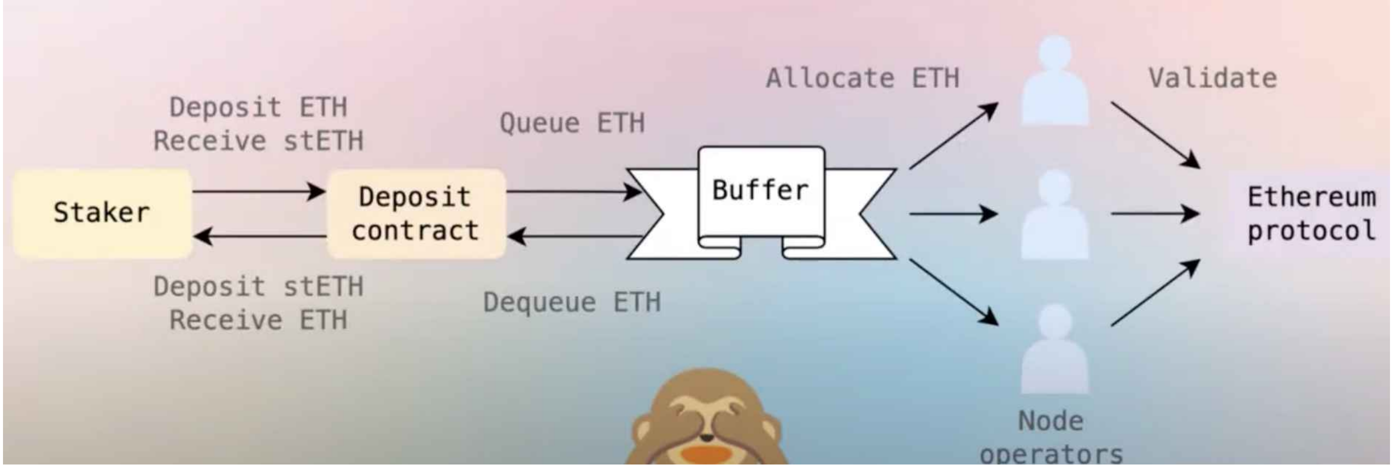
- LMD-GHOST: Tell the head of the chain
- Casper FFG: Finalize the block with 2/3 supermajority
- Consensus data: incl. casting the finality, giving weights to blocks etc.

- The world according to ETH

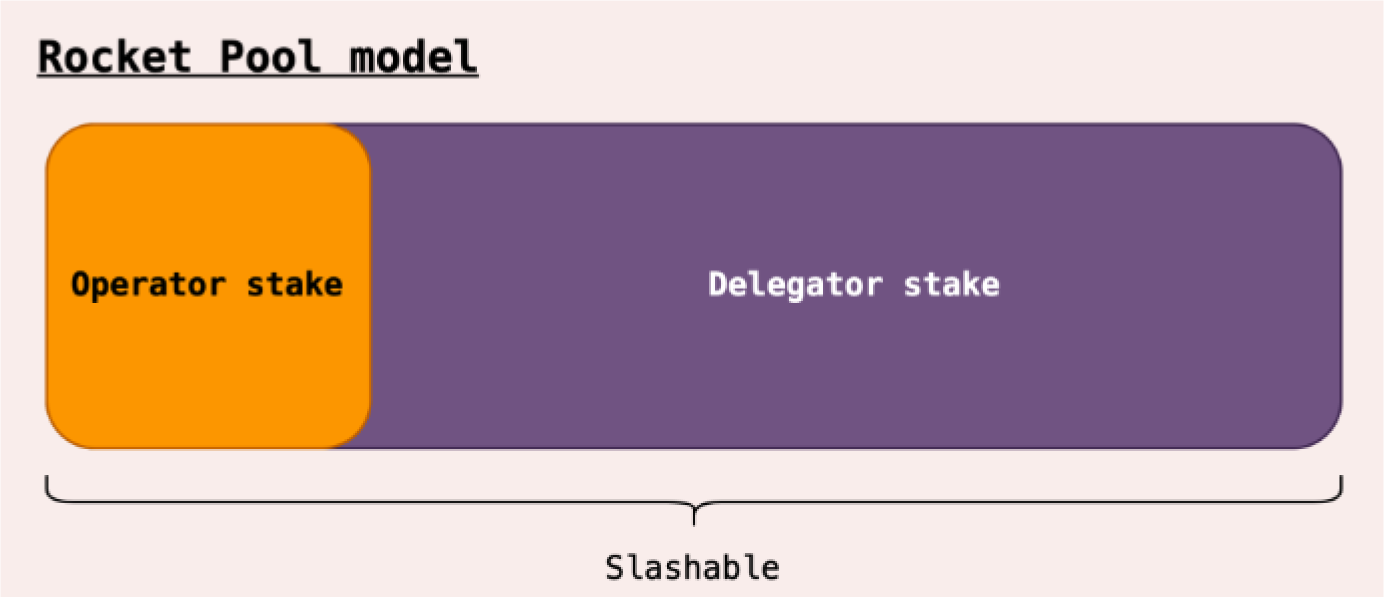
    - People deposit 32 ETH to become validators, or withdraw it to cease the operation.



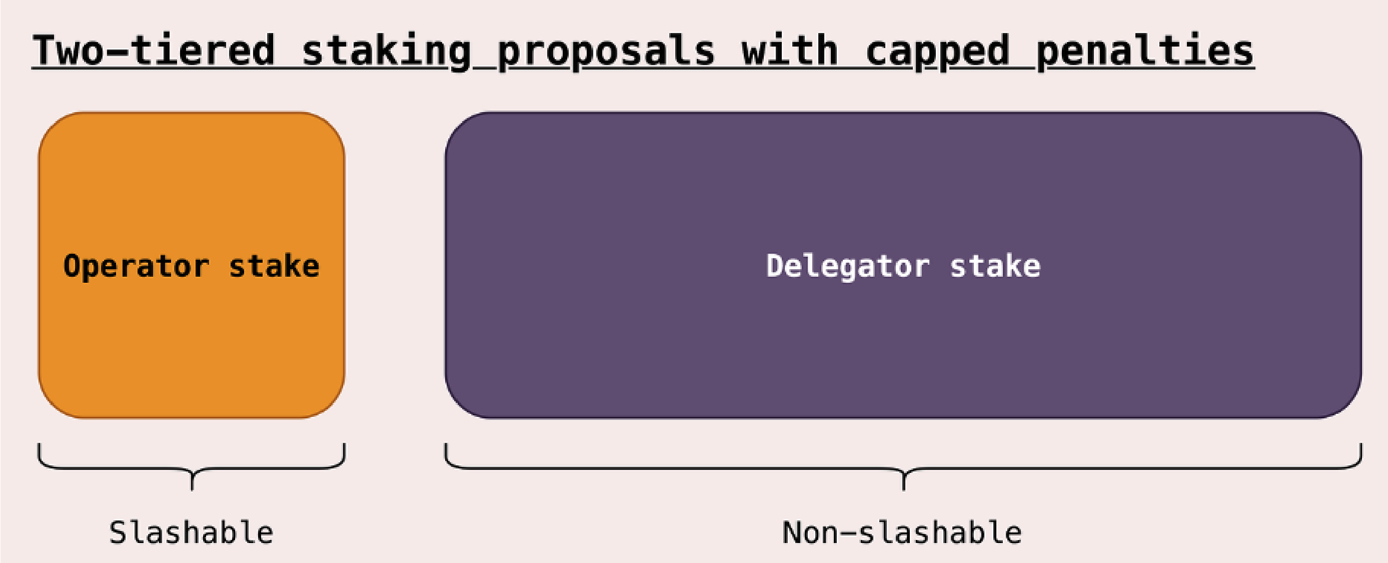
- The world out there
    - There is a very long chain of intermediates that outside of the control of the protocol



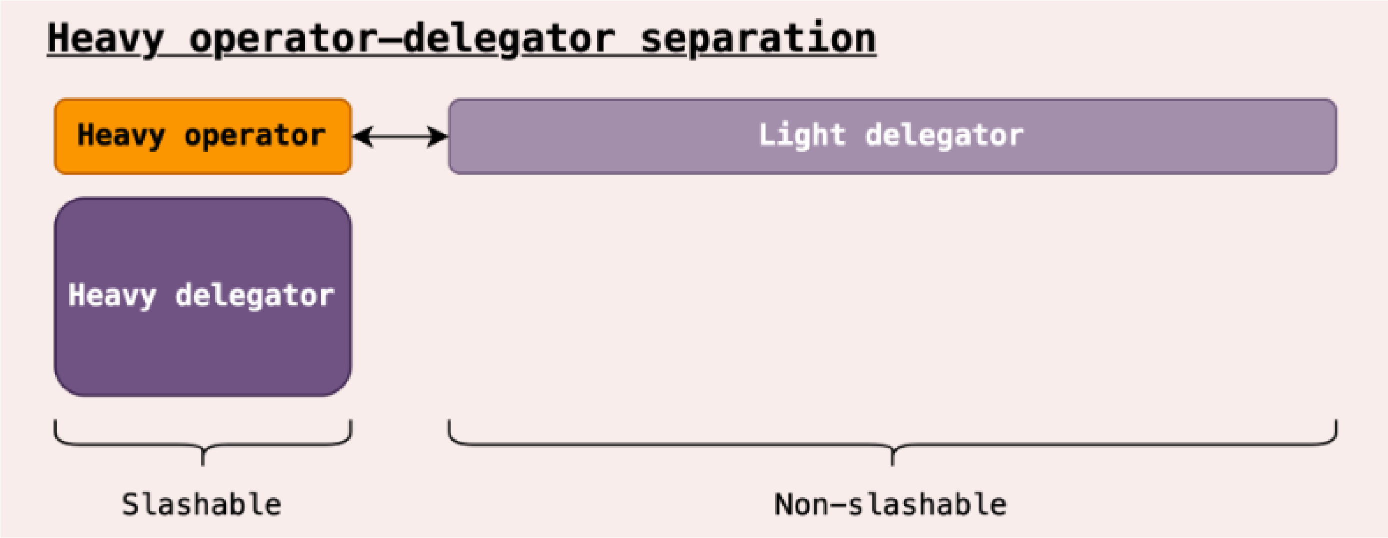
- Why stake?
    - Credible commitment to good service provision can provide a economic safety
    - Commitment requires capital, and capital wants to be free -> Liquid staking protocols (LSP)
    - The statement above doesn't specify whose dollar it belongs to -> The protocol doesn't see delegations
- Protocol service providers
    - 2 classes of providers
        - Solo operators: a priori untrusted, think living room validators, solo stakers (operators + own capital)
        - Professional operators: a priori trusted, think registered companies, big staking providers
    - (Liquid) staking protocols may employ a mixture of both solo and pro operators
- Rocket pool model



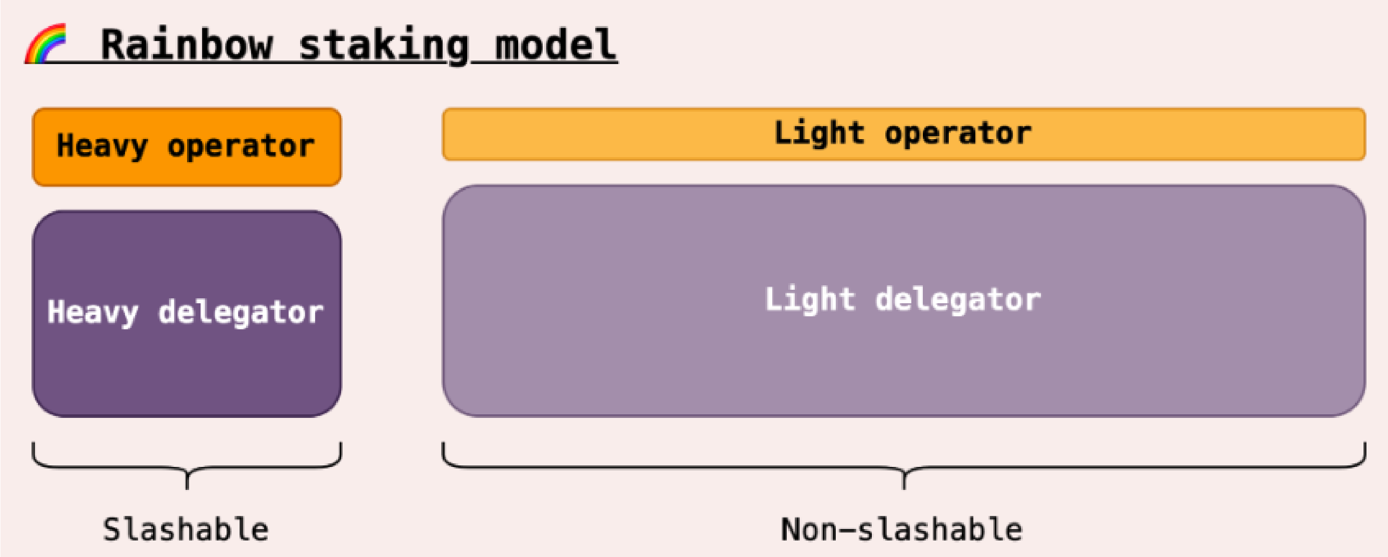
  - Operator can be solo staker: The solo staker needs to put up some ETH as collateral, and other delegators will fill the remainder. All the stakes are slashable.
  - Pro: Allow more opportunities to onboard solo operators
  - Con:
    - There is trust issue between untrusted solo operators and delegators
    - Due to capital efficiency + cost pressure, it means that LSPs rely on pro operators significantly
- Strawman model: two-tiered staking proposals with capped penalties



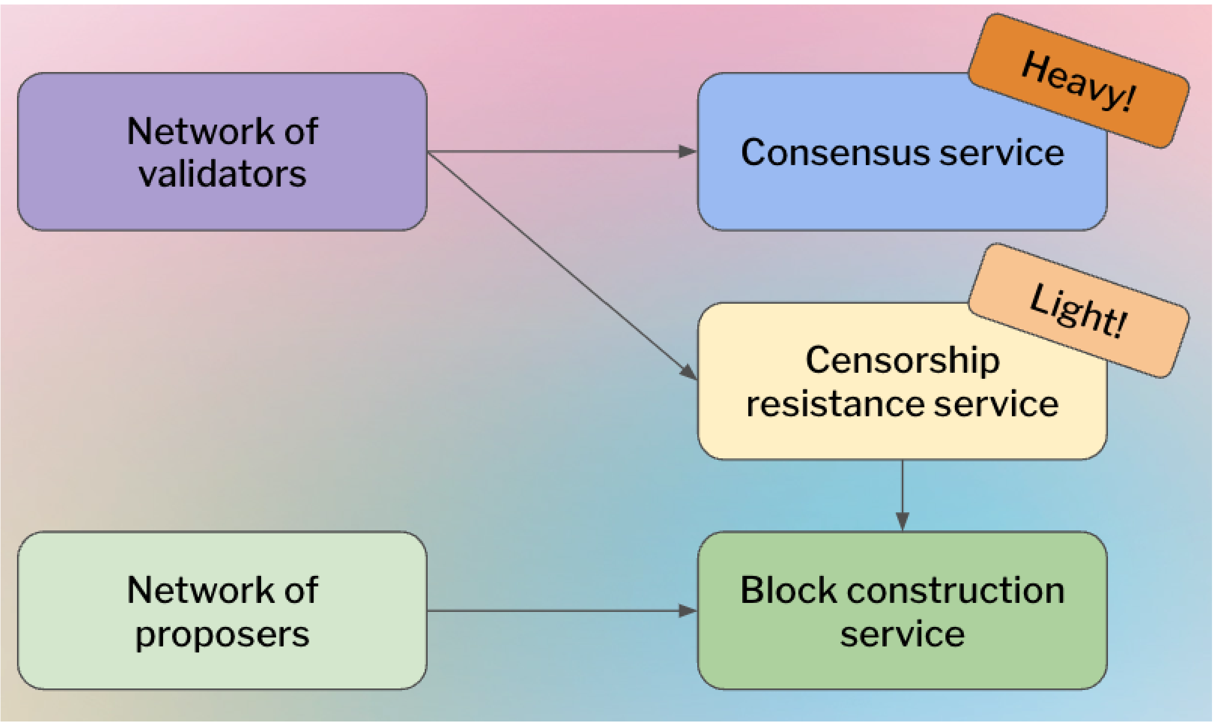
  - Premise 1: People want to stake aka do something with their ETH
  - Premise 2: Make only the people who actually perform validation (i.e. the operators) liable (i.e. slashable)
- Heavy operator-delegator separation



  - Heavy operator & delegator: The slashable stake should be people who are chasing yield, therefore should still incl. both operator and delegators
  - Light delegator: The small-staking roles can be non-slashable as they can contribute to prevent a 51% majority of node operators from engaging in transaction censorship
  - Reference: Protocol and staking pool changes that could improve decentralization and reduce consensus overhead
- Rainbow staking model



  - Heavy operator & delegator: Manage the chain validation, make credible commitments etc.
  - Light operator & delegator: Let light delegators delegate to a distinct set of operators, i.e. light operator.
    - None of the light stack is slashable, so no pressure to rebuild the LSP stack with centralizing forces and it could be solo operator friendly.
  - Reference: Unbundling staking: Towards rainbow staking
- Validator services under Rainbow staking model



- About inclusion lists (IL)
    - The real definition: a way for the most decentralised set of Ethereum to input their preferences into the make-up of the chain
    - Key goal: Block co-creation
    - Reference
        - EIP 7547: https://eips.ethereum.org/EIPS/eip-7547
        - ROP 9 multiplicity gadgets: https://efdn.notion.site/ROP-9-Multiplicity-gadgets-for-censorship-resistance-7def9d354f8a4ed5a0722f4eb04ca73b
        - Committee-based inclusion lists (COMIS): https://ethresear.ch/t/the-more-the-less-censored-introducing-committee-enforced-inclusion-sets-comis-on-ethereum/18835

## Q&A

- Referring to the protocol's goal, do you feel that unbundling validator's roles into different sub-roles eg. Proposer, tester, delegator etc., would help the community better exercise control over validtaors on the LT?
    - It can surely give more control as a community as fitting a big object into only one specific channel is harder.
- With the execution tickets, does that change the frequency of a validator propose a block? What's the mechanism for redeeming the ET?
    - There is a proposal written by Justin & Mike.

  - For the validators, if they want to propose a payload, they need to go the market and buy the execution ticket, instead of directly giving themselves the execution proposing right in slot-auction ePBS.
  - The ET market may be similar to an AMM that supplies ET. But there are still lots of questions about the details of mechanism design, eg. how the transactions flow, if the tx lives in the CL etc.
- Would offline validators and double voting be slashed retroactively with the ET?
  - It will be pretty much the same mechanism as of today.
- Any existing mechanisms for addressing the bypass ability in PBS?
  - MEV burn mechanism: The idea is to constrain the beacon proposer so that it cannot choose whatever execution proposer it wants. The approach is to use attesters to observe the bids in the network and force the beacon proposer to choose a bid that maximizes the value.
  - However, the mechanism is quite difficult to execute. One of the difficulties is the auction for bidding rights as it happens at the same time when the block value increases. Fundamentally, the bypass ability may not possible to be fully eliminated.
- What are the other implications of rainbow staking, apart from addressing the issues of rocket pool, eg. restaking?
  - The idea of adding more protocol services in a permissionless way is the key part here. The rainbow staking model can be a nice interface for plugging in new services to the protocol, which mirrors the restaking services that are out of the protocol.
- On the heavy services, what do you think about them?
  - Compared to other chains, Ethereum doesn't have in-protocol-delegation. That's why we have leaders, and big pools living on top. But that doesn't mean we need to stick to that path.
  - We could enshrine gadgets that try to lower the entry barrier for LSP to prevent dominance of one single protocol.
- How does the protocol enforce how the capital is used? Why wouldn't a heavy operator continue to take in capital and run it all as heavy, slashable capital if it's higher yield? What is the incentive to run it as "light" since we know that disinterested delegators often don't pay heed to (relatively) small differences in risk factors?
  - People can do both heavy & light delegating, eg. Restake their heavy delegation into light services, and surface who are the operators doing a good job on censorship resistance.

    - When choosing delegation to heavy or light operators, the selection criteria might differ. For large amounts of ETH, people will be inclined to choose pro credible operator eg. Lido, coinbase etc. For smaller amounts, the criteria could be more flexible, eg. someone's twitter content quality.
- How to think about these open problems, how to study further on these topics?
    - Read and watch relevant blogs/ videos as much as possible
    - Make your own way into these topics and don't hesitate to reach out to people