



RIG Bouillabaisse: Incentives in eth2 and beyond

Barnabé Monnot






Ethereum Foundation

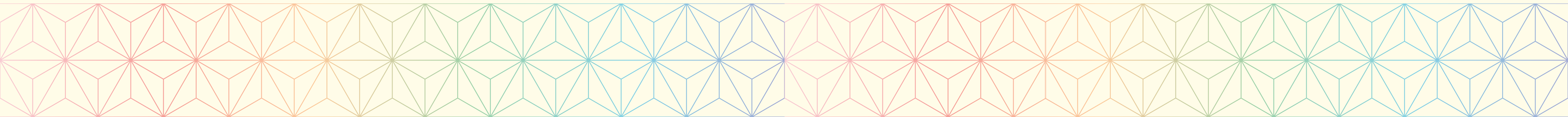
Robust Incentives Group

These slides: <https://github.com/ethereum/rig>






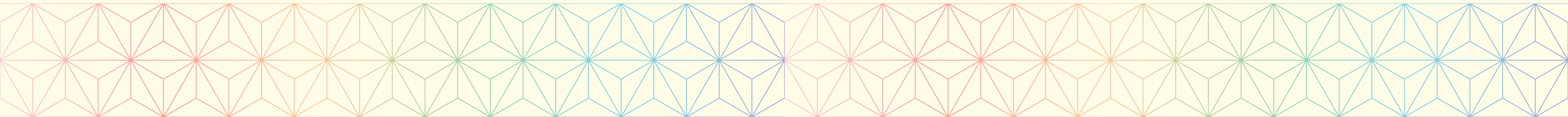
Robust Incentives Group (RIG)

-  Ethereum research team focused on everything with *strategic* flavour.
-  Strategic in the sense:
 -  Stakeholders make decisions to optimise their payoffs.
 -  Their payoffs are affected by decisions of other stakeholders.
-  Extremely general model to study *protocol security* and *decentralised applications*.










What do we mean by incentives?

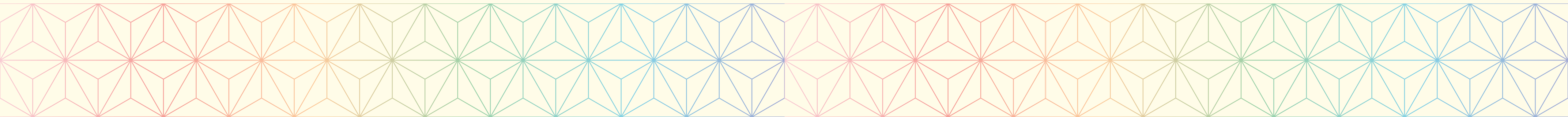
-  **Game theory:** Specify players, action spaces, payoffs, then analyse *what people ought to do* => More descriptive.
-  **Mechanism design:** “Inverse” game theory. Given some outcome we like, *tune the game* so that players achieve that outcome.
-  **Algorithmic game theory:** We also care about *computationally tractable mechanisms* and usually assume agents are bounded.







What do we mean by robust?

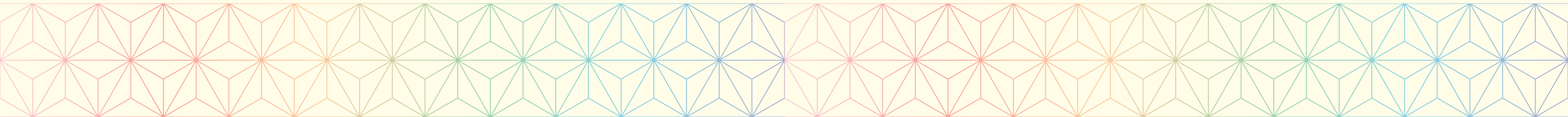
-  Decentralised protocols are **complex**.
-  Complexity compounds! Can lead to various emergent behaviours.
-  We want our protocols to survive in any (or most) conditions.

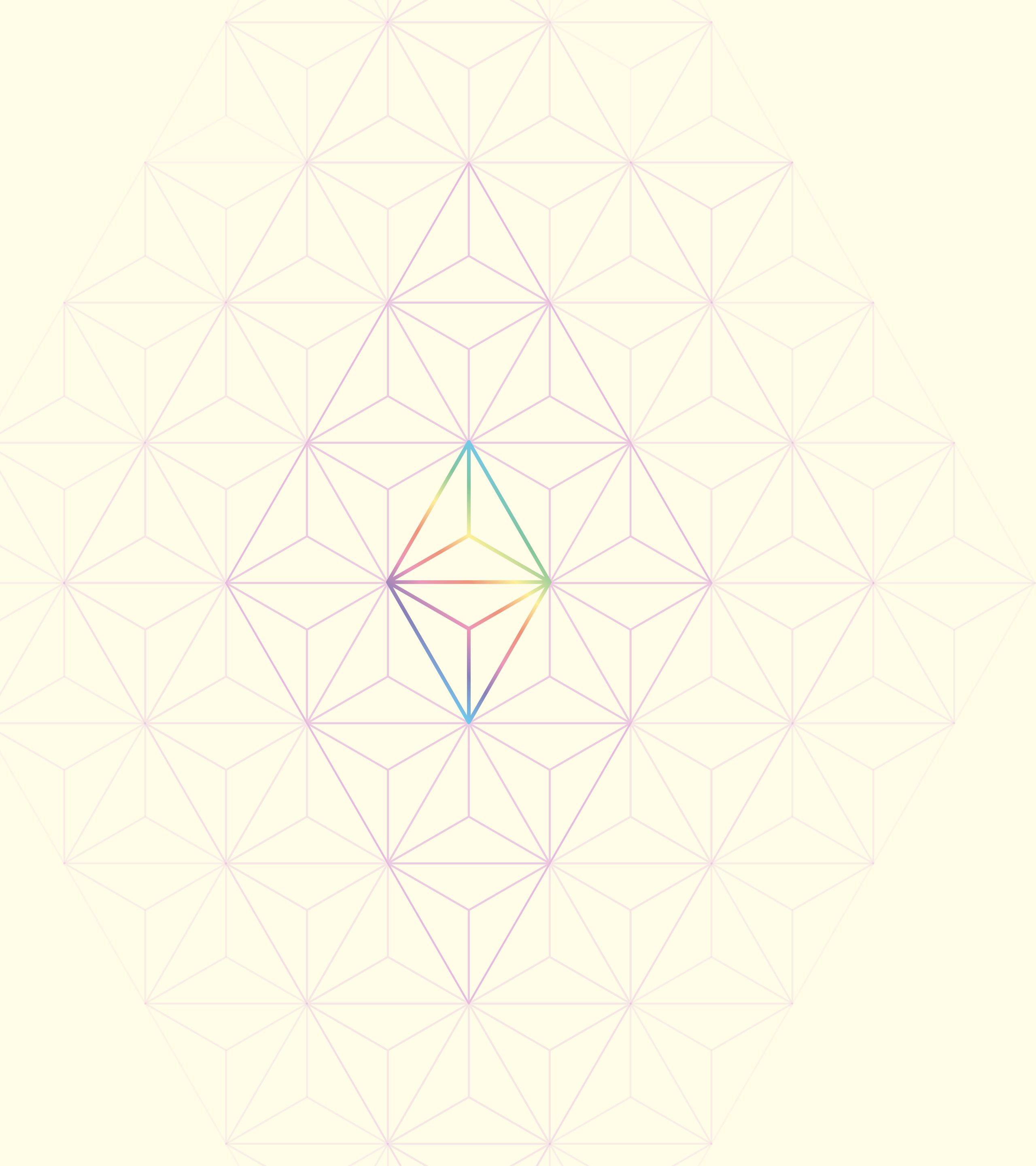
-  Decentralised protocols are very different from “in-a-vacuum” game theory.
-  **Which are our players?** Robust against Sybil behaviour.
-  **Which is the game?** Robust against network delays, adversarial manipulations.
-  **Which are the payoffs?** Do we care only about crypto-denominated incentives?



What do we mean by group?








-  Additional RIG focus: **develop outreach, education and awareness** of strategic questions.
-  Help gather resources to study these questions, exhibit good patterns.
-  Organise existing research and study applicability for Ethereum.
-  Talk to a broad audience about open questions and onboard incentives-oriented people.

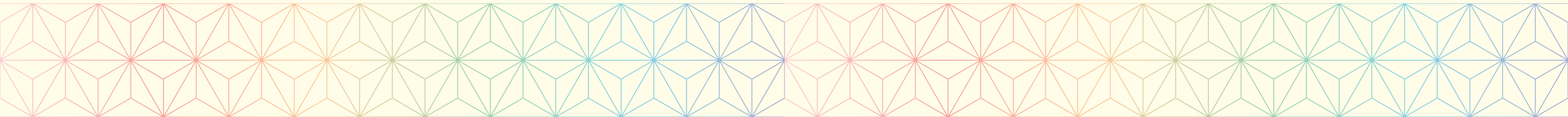




Incentives in eth2




eth2 incentives 101

-  Validators make the network run! We want to incentivise **good validator behaviour**.
-  Validators are expected to vote correctly and on time.
-  Validators are expected to create blocks correctly and on time.
-  Validators are expected to look for slashable offences.
-  (later phases) Validators are expected to finalise shards.
-  Incentives are provided as a return on initial deposit (“**stake**”).
Bad behaviour should hurt this deposit.
-  **What can we expect?**





Economic analysis

Protocol-level stability

-  We specify some desired validator behaviour.
-  We specify rewards depending on outcome.
-  We hope that rewards induce the desired behaviour.

Asset-level security

-  A more intangible (and hotly debated) notion.
-  We want attacks on the chain to be **optimally** painful to the attacker.

Economic analysis

Protocol-I

- 🎨 We specify behaviour
- 🎨 We specify
- 🎨 We hope behaviour



debated)

n to be
ker.

The protocol space



We have a big protocol space, **P**.

P = { all eth2 PoS protocols with FFG }



An instance of **P** (call it *P*) is determined by its parameters.

(e.g., minimum deposit, reward schedule, time to exit, issuance curve....)

Monnot, Saint-Leger, "[The economic incentives of staking in eth2](#)", hackingresearch.ch, 2019

Assumptions (you can change assumptions in green)

Base reward quotient = 348

ETH in circulation = 104,000,000 ETH

Validators per shard = 610

The number of validators per shard is a function of the network stake.

Number of validators = 624,640

If the total network stake is 20,000,000 ETH

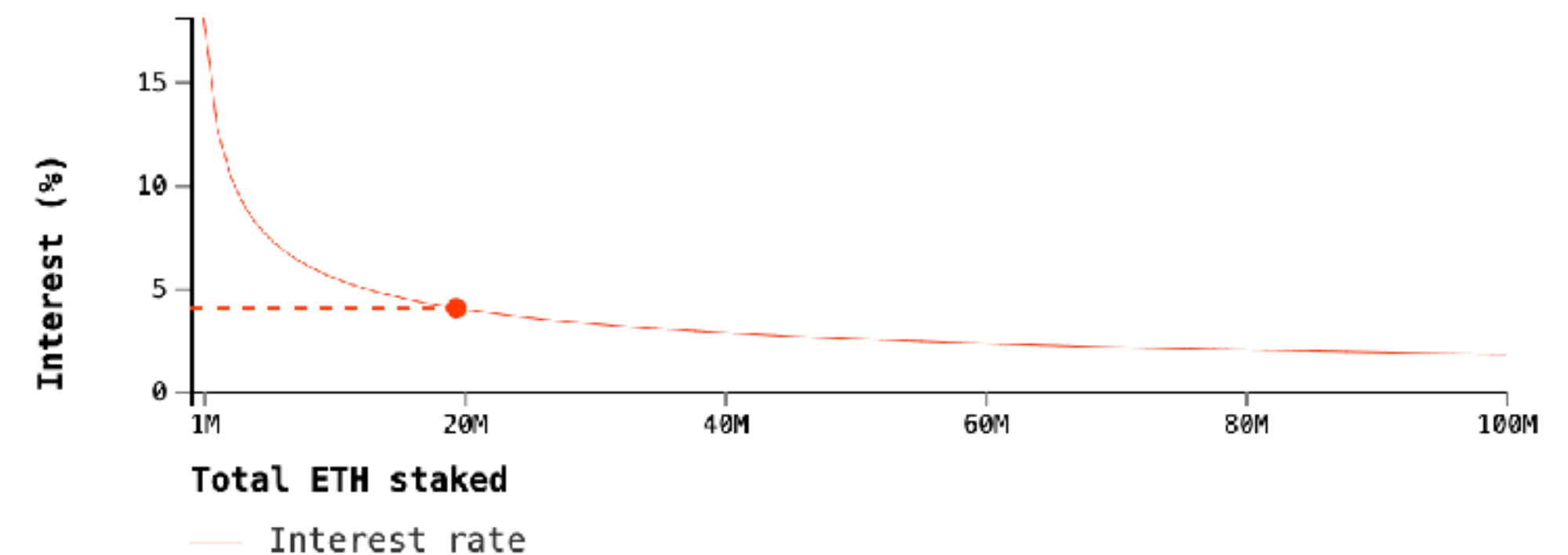
- Validators stand to earn 4.04% interest a year.
- The network issuance rate is 0.78% a year.

Remember, you can drag the green numbers above to adjust them. And click on the blue numbers to reveal how they were calculated.

Image. Interest rate as a function of network stake.

Validator interest decreases with the amount of stake in the network.

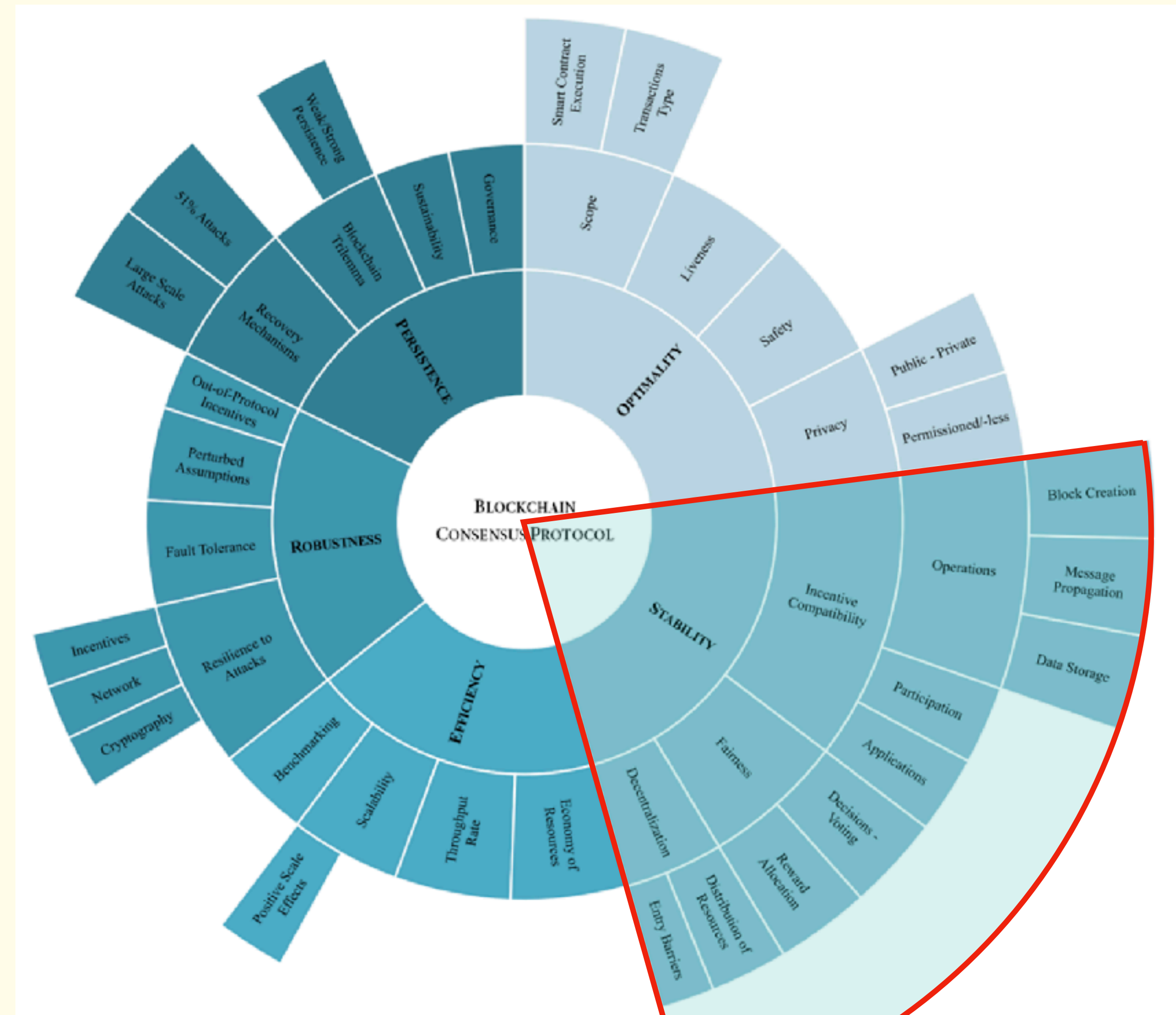
With 20,000,000 ETH staked, the validator interest rate is 4.04%.



Stable protocols

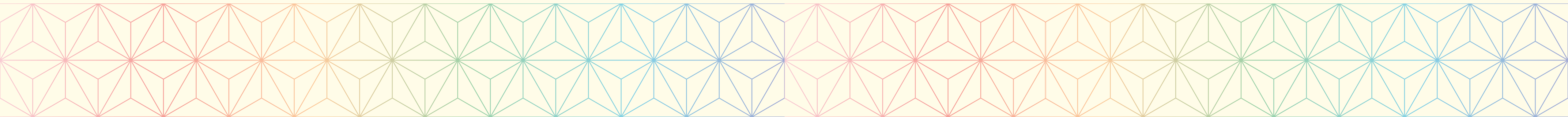
- 🎨 A protocol is stable if **expected validator behaviour matches desired behaviour**.
=> Incentive compatibility.
- 🎨 In other words, it is rational for validators to follow the protocol.
- 🎨 Restrict **P** to internally stable protocols, **P_r**.

Leonardos et al., "**PREStO: A systematic framework for blockchain consensus protocols**", arxiv, 2019






Some examples of instability

- 🎨 **Selfish mining** ([Eyal, Gün Sirer, 2014](#)) is one such deviation.
 - 🎨 We would like miners to release the blocks they find.
 - 🎨 But given enough mining power, **better response** to increase your payoffs is to mine privately.
- 🎨 A more recent example, in Tezos ([Neuder et al., 2019](#))
 - 🎨 Timing of block release can increase payoffs.



Protocol stability to asset security

-  Protocol stability lives in the “native asset” world.
-  But protocols live in a larger economic space.
-  So rational behaviour cannot be fully understood from “in-protocol” behaviour.

Concentration is harmless

The answer is that bitcoin's design doesn't assume mining power is widely distributed. It's simply not a requirement. Instead, it only assumes miners are rational, which is something completely different. Rationality means agents do what is best for them, even if that means colluding with other miners to attack the system.

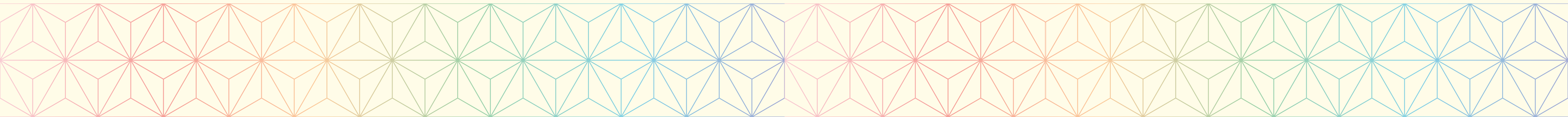
Satoshi addressed this matter directly [in the white paper](#):

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favor him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

Let's unpack this a bit. It is the incentive in the form of new coins and transaction fees that motivate the majority to “stay honest.” Satoshi realized the only way to prevent a “greedy attacker” from taking over is to make it more profitable to play by the rules than to attack the system.

This is the key to bitcoin's assurances and at the same time the most widely misunderstood aspect of bitcoin's design.

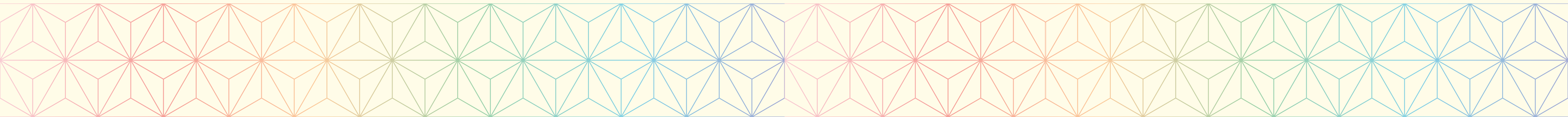
Hasu, **No, Concentration Among Miners Isn't Going to Break Bitcoin**, coindesk.com, 2020







What we would like to have

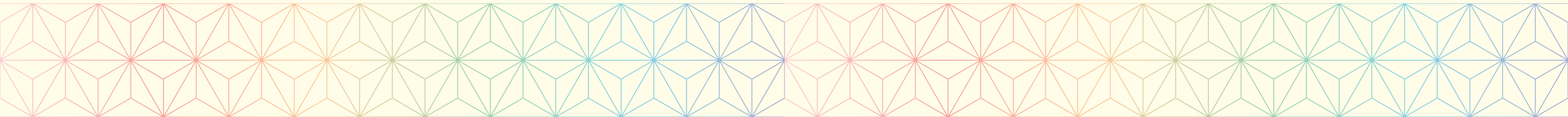
- 🎨 We restricted \mathbf{P} to \mathbf{P}_r .
- 🎨 We seem to like “minimum possible issuance”.
Intuitively, we already know that there are **tradeoffs**.
- 🎨 Now we would like to optimise “something” (**security**) over \mathbf{P}_r to decide protocol parameters.
- 🎨 We need to write down these tradeoffs.

$$\max_{P \in \mathcal{P}_r} S(P)$$









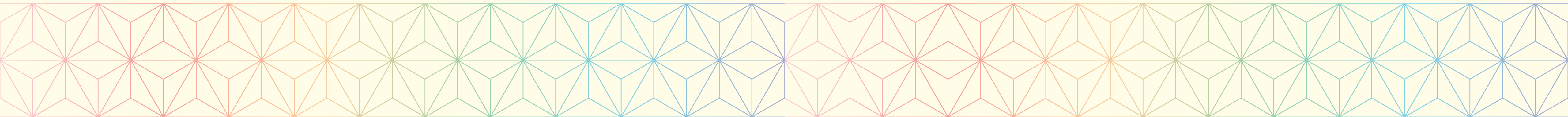
RIG efforts

-  More focused on **protocol stability** so far.
-  **Two axes:**
 -  Pin down a “mathematical specification” of the validation game.
 -  Produce a simulation environment to test validator behaviours (“*Beacon Runner*”).



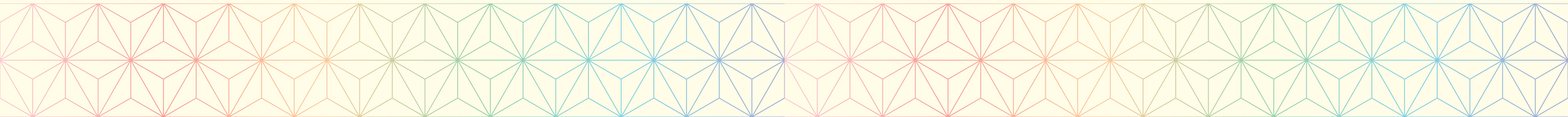
Beacon Runner

-  Adopt the view of **blockchains as controlled dynamical systems**.
-  **State:** A chain of blocks.
-  **State update:** Adding a new block to the chain.
-  **Control:** Users and validators participate in block formation.
-  The [Beacon Runner](#) is a cadCAD wrap of the eth2 specs, focused on validator behaviour and incentives.
-  [cadCAD](#) (by BlockScience) is a new framework focused on simulation of complex token dynamics (but also more general than that).
Check out Griff's workshop on Thursday morning!










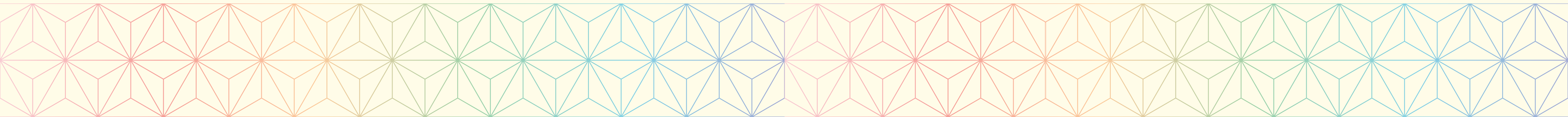
Why do we simulate?

- Complex behaviours often have simple micro-rules.
We want to define these rules and see what emerges => **agent-based models**.
- We also want our simulations to be **as close as possible** to the real system.
This cadCAD wrap takes the specs as is.
- And we want our environment to be used as a part of more complex phenomena:
 - Competition between on-chain lending and staking (see [Chitra, 2020](#)).
 - Formation of competitive validation opportunities (pools, exchanges...)
 - Dynamics of recovery from chain attacks.






Beacon Runner states and policies

-  **State updates:** Given by the specs.
-  **Validator policies:** Up to us!
 -  Honest validation: Given a state, do expected behaviour on time.
 -  Offline validation: Given a state, do nothing.
 -  Kamikaze validation: Given a state, do a slashable offense.
 -  Malicious validation: Given a state, try to finalise a different fork.
 -  ???








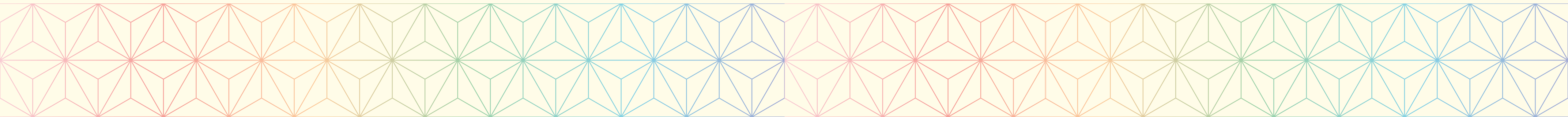
Beacon Runner states and policies

-  Policies are functions of the state.
`policy : state → action`
-  cadCAD allows for an extensible implementation.
-  Plug your own behaviour and see what happens!

```
In [25]: block_attestation_psub = [  
  # Step 1  
  {  
    'policies':{  
    },  
    'variables': {  
      'beacon_state': state_update_slot  
    }  
  },  
  # Step 2+3  
  {  
    'policies': {  
      'action': honest_attest_policy  
    },  
    'variables': {  
      'current_slot_attestations': update_current_slot_attestations  
    }  
  },  
  # Step 4+5  
  {  
    'policies': {  
      'action': propose_block  
    },  
    'variables': {  
      'beacon_state': state_update_block  
    }  
  }  
]
```

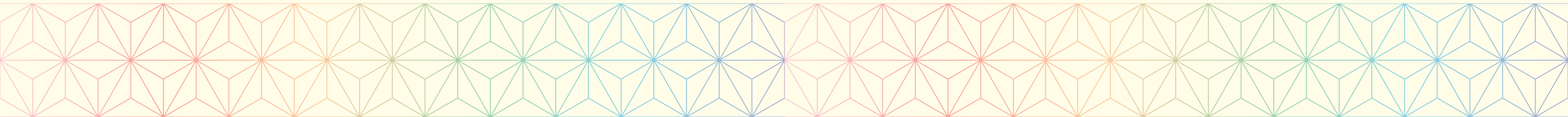
Beacon Runner outcomes and payoffs

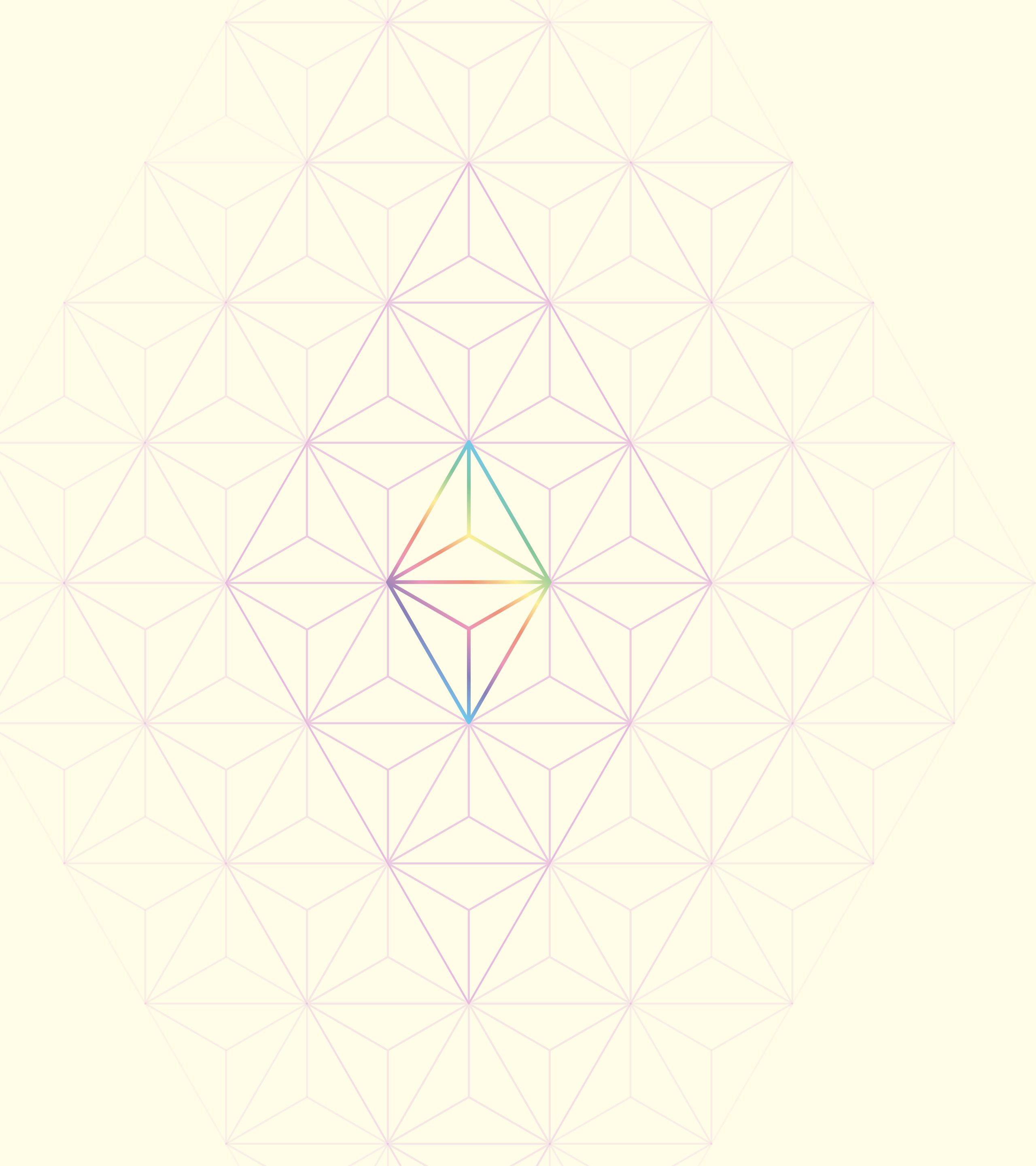
-  **Payoff schedule** is given by the specs.
 -  Your vote is included in the “GHOST”-approved chain: get reward.
 -  Your block is included in the “GHOST”-approved chain: get reward.
 -  Your block has x votes in it: get $x\%$ of rewards.
-
-  Payoffs are functions of outcomes.
`schedule : outcome → payoff`



Beacon Runner 2049

- Validators observe some state and do something.
 $\text{policy} : \text{state} \rightarrow \text{action}$
- Meanwhile, some outcome is realised and they are getting a payoff.
 $\text{schedule} : \text{outcome} \rightarrow \text{payoff}$
- How do we get from actions to outcomes?
This is the space to model uncertainty, e.g., network delays.
 $\text{realisation} : \text{actions} \rightarrow \text{outcome}$
- We can check whether validator strategies are robust to realisations!

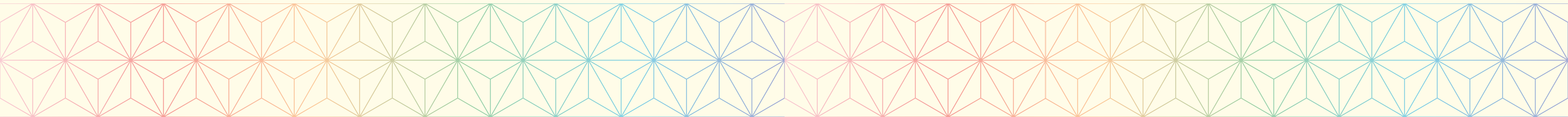








And beyond

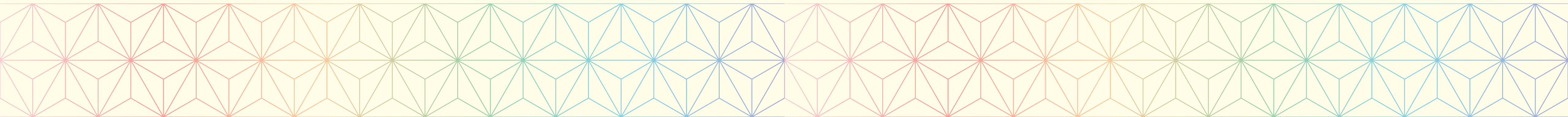
EIP-1559: A new transaction fee market

- 🎨 The current first price auction for tx fee is complicated and suboptimal.
- 🎨 EIP-1559 proposes to replace it with a variable fee based on block demand.
- 🎨 Recently, [Vulcanize developed a client](#) to handle EIP-1559.
- 🎨 We can run tests, simulate execution and theorise about expected market behaviours!



Light client incentives

-  Sharding relies on a network of full nodes and light nodes exchanging data.
-  [Al-Bassam et al., 2018](#): We can use fraud and data availability proofs to keep everyone in check.
-  But who produces these proofs? Who pays for their dissemination?
-  [Light client subprotocol](#) introduces market dynamics.
Are they efficient? Are they robust?



Thank you!



Check out <https://github.com/ethereum/rig>



Find me on Twitter [@barnabemonnot](https://twitter.com/barnabemonnot) (or hiding in Singapore...)

