Endgame: Proof of Governance

Originally posted to Mirror.

Thank you to Hasu, Mike Neuder, Konstantin Lomashuk, Sacha, Josh Bowen, Tomasz Stańczak, Chris Hager, and Toni Wahrstätter for great conversations around this topic.

Introduction

This post expands upon my conversation with Hasu in ourfirst episode of Uncommon Core 2.0 as well as my talk at Modular Summit. I'll cover the following:

- Proof-of-stake (PoS)
- The key motivations for PoS and its relevant properties are often deeply misunderstood, particularly "economic security" and "decentralization."
- Delegated proof-of-stake (DPoS)

- I'm using this to refer to anyPoS system with native stake delegation (e.g., typical Cosmos chains). Simple DPoS by itself has severe shortcomings, including generally poor delegate selection (topheavy stake delegation) and capital inefficiency.

- Liquid staking tokens (LSTs)

- LSTs are often the best tool for achieving a decentralized operator set on top of a permissionless PoS mechanism. It's essential for these LST protocols to be properly aligned with the underlying chain (e.g., Lido's proposed dual governance) or even more directly monitored by the chain's governance (e.g., Cosmos Hub's proposed liquid staking module).

- Proof-of-governance (PoG)
- A chain's governance could just directly elect operators without

staking. I'll argue that this can be more secure and decentralized than PoS in many circumstances. An implicit form of PoG may be inevitable as LSTs continue to proliferate (i.e., it may be the LST's governance which controls delegation, with some dual governance backstop).

- Restaking & PEPC

- If PoS economic security may not be exactly what it seems at the surface, then what is restaking really

for?

I'll primarily analyze this landscape from the perspective of how to choose rollup operators (e.g., sequencers and provers) vs. how to choose Ethereum validators. I intentionally use these opposite ends of the spectrum (e.g., a rollup can be secure even with a single default sequencer) to reveal the underlying fundamentals applicable to all

chains. The mechanisms I describe may also be viable for certain traditional "monolithic chains" (i.e., not just rollups).

Arguments in crypto often take the following narrow forms:

- Is PoS better or worse than PoW?
- Is PBS better or worse than POB?
- Is FCFS better or worse than explicit price auctions?

The list goes on. The simple answer - they're tools which are neither inherently good nor bad. They may be more or less effective in different scenarios depending on the desired outcome. Different protocols have different desired outcomes, so different mechanisms often make sense. This should be plainly obvious, but I feel that we frequently look past this.

Rollups generally serve a very different purpose vs. Ethereum, and sequencers play a very different role vs. Ethereum validators. I often notice an inclination to take the mechanisms we see on Ethereum → try to strap them onto rollups without sufficiently reconsidering them from first principles. Unique problems require independent analysis.

Consensus Protocols & Sybil Resistance Mechanisms

We'll build up from the absolute basics here because I'm going to be rather contrarian on a lot of fundamentals later on. For more background, I recommend Tim Roughgarden's awesome video series on PoS.

For starters, proof-of-work (PoW) and proof-of-stake (PoS) are examples of sybil resistance mechanisms

. They are not

consensus algorithms/protocols (despite frequently being misnomered as such).

- Sybil resistance mechanism (e.g., PoW or PoS)

- Dictates who

can participate in the consensus protocol (e.g., proposing and voting on blocks).

- Consensus algorithm/protocol (e.g., BFT-type or longest-chain)

- Dictates how

the participants reach consensus. It instructs which blocks will be committed to honest nodes' local view of the chain (and in what order).

Consensus protocols must protect themselves against sybil attacks. If 1 identity = 1 vote and it's costless to fake identities, then a single attacker could overwhelm the network:

We can view blockchains in two deployment settings regarding who can participate in consensus:

- Permissioned

- We avoid sybil attacks with a trusted setup to pick all consensus nodes. New entrants cannot join permissionlessly. This permissioned setting is generally called proof-of-authority (PoA).

- Permissionless

- Consensus nodes are not known in advance, and they can change over time unpredictably. We need to get more creative here to solve the sybil attacks.

In the permissionless setting, we generally tie voting power to some real economic cost to prevent sybils:

- PoW

- Probability of proposing a block = % of total hash power

- PoS

- Probability of proposing a block = % of total assets staked

Staking should always be thought first and foremost as “how do I rate limit who can participate” and “how can I make sure the people who do participate care?”

It isn’t really about accountability or slashing, that is just an add-on feature to further encourage good behaviour.

— Patrick McCorry (,) (@stonecoldpat0) July 6, 2023

Different sybil resistance mechanisms may also have varying degrees of accountability and punishment for undesirable behavior:

- PoW

- You could hard fork to change the hashing algorithm, but this punishes all

miners whether honest or dishonest. The hash rate would completely reset as well (e.g., go back to CPU mining), potentially allowing for further attacks.

- PoS

- This allows for targeted punishment, slashing the stake of malicious actors and removing them from the consensus set.

There are countless nuanced tradeoffs between PoW vs. PoS, but both share at least one issue:

Assigning consensus voting power based purely on economics is often suboptimal.

Proof-of-Stake vs. Proof-of-Governance

Proof-of-Stake

We often take PoS as a given nowadays, but we need to question whether "staking derivatives have just been turning them into effectively Proof of Authority networks with hand-selected validators"?

Proof of Stake was a mistake. https://t.co/E14g78Yv9C

— sunnya97.osmo ⚡ (@sunnya97) June 27, 2023

Let's consider the options here broadly:

1) DPoS (Delegated Proof of Stake)

- Token holders all delegate their tokens to a validator who votes for them. This is capital inefficient, and LSTs inevitably arise. There's also heightened demand for LSTs when in-protocol delegation does not exist (as is the case with Ethereum).

2) LSTs (Liquid Staking Tokens)

- LST protocols (e.g., Lido) manage stake delegation behind the scenes and give users a liquid token (e.g., stETH) representing their share.

3) LSTs + Dual Governance

- The LST protocol cedes some control for incentive alignment. For example, the LST protocol's governance (e.g., LDO token holders) may give veto rights to LST holders (e.g., stETH).

Proof-of-Governance

I'll provide a fourth option here as well:

4) PoG (Proof of Governance)

- The chain's governance mechanism elects delegates. No staking is required. There's a critical difference between this and the PoA I mentioned earlier:

- PoA

- Generally used to describe private blockchains (e.g., a bank's). Also reasonable for public RWA blockchains like Noble (mints USDC) where the issuer (e.g., Circle) has effective "authority."

- PoG

- Clear and credible decentralized governance mechanisms in place to elect delegates.

Consider the following example:

- A rollup whose validating bridge controls most of its assets
- Decentralized onchain governance is in place to manage the bridge's upgradeability
- This governance mechanism has the power to elect whichever sequencers they choose (or force their removal) at any time

On Arbitrum, the "escape hatch" (force inclusion) not only allows you to escape but also allows the DAO to elect a new sequencer

— Lee Bousfield (plasmapower.eth) (@PlasmaPower0) July 17, 2023

The difference between the above scenario (PoG) vs. some company running a permissioned blockchain where they're the sole validator (PoA) should be clear. The rollup has governance mechanisms in place to choose delegates which are:

- Clear

- In this case, the mechanism is clearly defined onchain (the bridge contract).

- Credible

- In this case, the bridge's governance is likely to sway everyone to follow (it controls most of the assets).

This contrasts to the PoA example where there's no process in place to choose another operator. Technically, anyone can fork even a corporate chain (if the code is open-source) and put in themself as the delegate, but nobody would follow it.

Whatever mechanism will practically dictate a contentious scenario (e.g., a fork) is in control. This gets into the recent discussion at Modular Summit where Anatoly offered the following hypothetical:

"You can take Ethereum L1 data right now, dump it into Solana, dump the state root that's computed from that Ethereum data, and then run an Optimism style proof to check if fraud has occurred... Does that make Ethereum a Solana L2?

Mechanically yes, socially no."

Vitalik's response described this situation incredibly well:

"The sovereign is the one that decides the exception, and in blockchains the exception is bugs and 51% attacks. And the question is always what happens if Ethereum hard forks for example. If Ethereum hard forks then there's a few possibilities.

One possibility is that the Solana bridge, and if we assume that it's a perfect ZK bridge, then according to the old ZK rules the Ethereum chain will just start being invalid, and that bridge will just basically create its own version of Ethereum Classic 2.0.

But then the question is what do the assets follow, and realistically in this case the assets would follow Ethereum, but if you get into a world where the majority of assets are rooted onto Solana and then bridged onto Ethereum then that would look very different. And in that case then Ethereum would not actually be capable of hard forking unless there's some kind of onchain governance.

Or a possible third world would be a world where the assets are all based on Solana but at the same time the Solana community is willing to hard fork whenever Ethereum is willing to hard fork. And in that case that's a construction based off of I guess some kind of deeper version of onchain governance. And in that case you would still be able to call Ethereum an L2, but it would be part of this ecosystem where things are willing to be part of the governance of each other.

You basically just have to look at what happens if one chain or the other chain gets 51% attacked... what happens if one chain gets 51% attacked and the community decides to do a user-activated soft fork to kill the attackers. So not like an easy slashing hard fork, but like a censorship hard fork where the community responds by picking the minority chain, and on the minority chain the majority would have to either skip being on that chain or get slashed, and if they skip being on the chips they get leaked."

Similarly to the L2 discussion above, Ethereum social consensus has the ultimate power over the consensus layer delegates today. This was clearly displayed in Dankrad's recent comments around restaking alignment:

"We should establish a social norm here that building on Ethereum, like building smart contract protocols on top, we should almost never touch that. I mean it would be in a very extreme, very rare situation.

But messing with the staking layer is a different matter. For example if lots of validators start censoring we do want the ability to intervene. So I think it would be a good signal to send, yes we might be opinionated on what you do on the staking layer, and that might include messing with your protocol and destroying it."

While social consensus is always the ultimate power, the mechanisms enshrined onchain (e.g., PoS vs. PoG, rollup validating bridge vs. no bridge, etc.) have profound consequences. Social fork coordination can be messy and difficult. Conversely, the momentum will generally be to follow a credible onchain governance mechanism where one is present (e.g., a bridge which may control most of the chain's assets).

I'll now walk through the merits of each approach. The table below shows some of the major considerations from the perspective of a "typical" rollup

. This is of course inherently subjective.

Again, I'm not saying either PoG or PoS is universally "better" than the other. We need to assess the problem from different vantage points. Ethereum vs. your typical rollup likely have very different priorities, and different mechanisms may better serve those goals. PoS clearly makes sense for Ethereum validators. My goal is to assess the opposite end of the spectrum (sequencers on rollups with active and opinionated governance) to understand the full tradeoff space. Let's go through it now.

Capital Efficiency

- DPoS

- Inefficient (unable to use the liquid collateral + long unbonding period) → LSTs likely arise.

- LSTs

- Far more capital efficient. Most stake can be represented synthetically and used freely. LSTs trying to support untrusted delegates (e.g., not KYCing and whitelisting operators) may require delegates to place some proportional stake themselves. The unbonding period also introduces some inefficiency.

- LSTs + Dual Governance

- Same as above.

- PoG -

Optimal. Capital isn't locked at all. All assets stay in their native form and retain full utility and value capture. Value captured can be burned, sent to a treasury, etc. (whatever governance chooses). Productive capital remains liquid. Strike economic arrangements directly with operators.

Allowing for separation of capital and labor vs. slashing incentive compatibility is the fundamental tradeoff:

- Require delegates to stake

- You could try to impose a limit where delegates must put up some personal stake as skin in the game proportional to the stake allocated to them. (E.g., the Cosmos Hub liquid staking module proposal recommends 250:1). This is capital inefficient, and it's not possible to know if out-of-protocol delegation is helping the validator fulfill their bond. The norm today is requiring no bond.

- Don't require delegates to stake

- This is the most capital efficient scenario, but it maxes out theprincipal-agent problem (PAP) for delegates as they do not bear the direct slashing cost.

There's inherently a pressure to separate capital and labor here, so we see PoS systems gravitate in this direction over time. The strength of this pressure will vary based on several factors including native delegation (e.g., Ethereum does not support in-protocol delegation) and usefulness of the asset (e.g., ETH is top-shelf collateral to be used throughout DeFi).

While LSTs continue to grow in major markets (e.g., Ethereum), it's possible that chains with highly inefficient markets gravitate less towards LSTs. Maybe nobody cares enough about them (e.g., not worth LSTs' time to deploy there) and the asset kinda sucks anyway so nobody wants to use it. There are still benefits to the liquidity of LSTs (e.g., don't need to wait for the unbonding period to be able to sell), but these factors could slow their growth at least.

Permissionless

Bitcoin and Ethereum have permissionless consensus - anyone who meets the specified requirements may start proposing blocks. Note that the degree of "permissionless-ness" is sometimes viewed as a spectrum based on these requirements. Higher minimum stake limits, caps on the total amount of validators, high hardware requirements to mine (e.g., ASICs), etc. are sometimes viewed as "less permissionless".

Overall though, anyone can join based on some clearly defined conditions within your control. If you meet the requirements, you can join. You don't need permission from some external body or governance process (e.g., if a protocol requires governance whitelisting to become a validator).

While Ethereum seeks permissionless validator entry, I don't view this as necessary for rollup sequencers in many circumstances. Indeed, it may even be very positive

for rollups to have some form of permissioning here. Hand-picking delegates allows you to select them based on their willingness and ability to enforce conditions which aren't necessarily programmatically enforceable.

Real-time Censorship Resistance & MEV Protection

Following on that last point, reputable validators can provide the following:

MEV Protection

- Centralized sequencers are currently trusted to keep user transactions private, preventing manipulation such as front-running. Chains with permissionless and untrusted validators (e.g., Ethereum) just shift this problem to other trusted third parties (TTPs) such as builders, matchmakers, and fillers. Chains with smaller validator sets can try to hold them accountable for undesirable MEV behavior, as is the plan for dYdX to start. This is even easier if you can explicitly hand-pick each validator.

Real-time Censorship Resistance (CR)

- Block proposers who are trusted with seeing builder block contents can modify the blocks. For example, mining pool operators could see megabundles in the clear sent by Flashbots' MEV-Relay. This allowed them to fill in the rest of the block (preventing censorship) even though Flashbots was censoring OFAC transactions in the PoW days.

Relays and builders only became a censorship chokepoint after the Merge, when PBS moved to a full-block commit-reveal to allow for untrusted validators to participate. While the long-tail of validators can now participate in MEV rewards, they're unable to enforce CR if they receive these rewards. This is a challenging balance that Vitalik has written about before.

While theoretically these issues can be fixed by the likes of inclusion lists and proposer suffixes, they come with a host of other issues. One is the simple desire to keep the long-tail of Ethereum validators low-resourced (e.g., some of these proposals are incompatible with statelessness). Additionally, it's unclear if something like publishing an inclusion list with OFAC transactions could weaken the legal argument that proposers are simply neutral infrastructure providers akin to internet service providers (ISPs).

Encrypted mempools and distributed building (e.g., with SUAVE) are other incredibly valuable long-term pursuits here. They can preserve privacy and minimize the power of any individual actor.

Overall, having a long-tail of permissionless operators shifts the MEV problem to unaccountable TTPs and introduces censorship chokepoints at least for the foreseeable future. While both of these can theoretically be addressed via better mechanisms over time (e.g., inclusion lists, encrypted mempools, and decentralized building, etc.), we're certainly not there today. Electing validators that are willing and able to enforce your desires around censorship resistance may be the clearest path today for some chains.

"Permissionless-ness" and "decentralization" are both means to an end. In particular, we're generally looking for:

Permissionless & decentralized operators → user guarantees (e.g., CR)

The end guarantees for users are what really matter. I worry that we're missing the forest for the trees here:

Permissionless operators → centralized TTPs arise → worse guarantees for users (e.g., censorship chokepoints and front-running)

Our end goal is always the properties that the system offers to users. We need to keep that in mind then ask the best way to get it today. The operators are just service providers that should be accountable to users and serve their needs. Always remember what validators are supposed to be:

Here we go https://t.co/xiTiEGBftO pic.twitter.com/43YNTgCZ3d

— Jon Charbonneau (@jon_charb) July 22, 2023

Governance-elected and accountable operators may be the easiest way to achieve some of these goals, particularly as the infrastructure matures. Governance can intentionally choose geo-distributed operators who are willing to meet their desired criteria (e.g., not censor transactions or front-run trades).

Governance-elected accountable operators → out-of-protocol actors (e.g., builders) powerless → user guarantees preserved (e.g., CR & no front-running)

I'll note a subtle but critical point here - it's not necessarily clear who the "user" is across different domains. While the "user" of a typical rollup is indeed a literal human being who wants to do stuff (make a trade, buy an NFT, etc.), the target "user" of Ethereum may be very different. Specifically, Ethereum may design itself with the assumption that rollups are its primary "users" rather than actual people in the long-run. This would result in a very different set of goals to optimize for (e.g., less concern about handling MEV protection in-protocol).

Any protocol must decide who are the "users" they're optimizing for and what guarantees they require. We generally want open access (e.g., censorship resistance) for those users

. Permissionless block producers

are simply a tool to further that goal.

Protocols aren't made for stakers. They're made for users.

Governance

This is closely tied to the point above regarding permissionless-ness, so I'll keep it short. Permissionless-ness is required if you don't want opinionated and active governance. Ethereum is run by a very rough offchain social consensus that's moving towards ossification over time. Managing an elected set of delegates to serve as validators clearly wouldn't be viable.

However, it appears likely that most rollups will have relatively opinionated and active governance mechanisms whether they like it or not. In particular, onchain governance will be necessary to manage upgradeability for their validating bridges. The minimal viable form here would be something like a fast path for emergency halts and a slow path for upgrades. These DAOs are already even electing security council members anyway.

If rollups will have this governance in place anyway, they now have the option to select their operators as well. This governance mechanism could be tasked with electing sequencers in much the same way that stakers or an LST protocol's governance would.

The exception would be if you wanted a rollup with an immutable bridge contract that holds most of your rollup's money. Then PoG wouldn't seem plausible. However, I don't expect that to be common in practice.

Another scenario is potentially a relative weakening of onchain rollup governance in some cases. Alternatives to the "canonical" bridge such as CCTP (native mint and burn USDC) or adoption of the EIP-7281 (aka xERC20) standard would increase native mint-and-burn. Reducing the tokens passively stored in bridges can meaningfully reduce risk.

Minimize tokens passively stored in bridges

— hayden.eth (@haydenzadams) July 20, 2022

This just sets up the interesting question of what is the real "rollup governance" at that point (as opposed to just the "bridge governance"). To my point earlier, you need to consider whether a governance mechanism is "credible" (i.e., who decides a fork).

In any case, I think it's very likely that this onchain bridge governance will be

the de facto rollup governance for most of these chains. These bridges will often hold a sizable portion of rollup assets, and some form of onchain governance will likely manage their upgradeability. You can read more about this power dynamic here and here.

This highlights how important figuring out onchain governance will be. This is especially critical for shared governance, bridge contracts, and upgrades as with Optimism's plan for the Superchain. The Superchain bridge will have governance powers over all OP Chains in the Superchain with the Optimism Collective controlling upgrades.

Settlement Assurances & Economic Security

Nic Carter had a great post a while back which included the below excerpt:

"Ledger costliness

is

the most profound and direct variable available to us to evaluate a blockchain's settlement guarantees. Put simply, it is equivalent to the amount paid to validators/transaction selectors per unit of time

.

At the time of winning a block, the miner necessarily has to have burned resources roughly equivalent to the value of the block

(typically with a small margin), unless they are extraordinarily lucky. Because of this, miners are incentivized to create valid and rule-following blocks.

So why does more ledger costliness per unit time mean more security for transactors? Because a greater salary to miners (who are presumed honest) means you need a larger army of mercenaries to defeat them

.

These resources have to come from somewhere: you need to marshal resources and hardware capable of producing hashes, electricity, and so on."

Settlement assurances are a system's ability to grant recipients confidence that a transaction will not be reversed. "Economic security" is generally viewed as the cost to reorg the chain:

- PoW

- Creating blocks to reorg has a real cost (burn energy) ≈ average block rewards. As more blocks pass, the security builds up (cost to reorg increases).

- PoS

- Costless to simulate blocks and create a fork. However, staked validators may be slashed for malicious behavior. The cost to reorg is generally described as the amount of stake you could have slashed.

However, delegation explicitly reduces this cost for the operators

. Now it's the delegators' stake (not the validators' stake) that gets slashed. Would you

accept $1,000 out of thin air right now if the cost was burning $2,000 of my

money? It's a classic principal agent problem.

[PAP

](https://en.wikipedia.org/wiki/Principal%E2%80%93agent_problem) - The conflict in interests and priorities that arises when one person or entity (the "agent") takes actions on behalf of another person or entity (the "principal").

Note that a PAP exists in both PoS and PoG. Delegates may act contrary to the interests of the delegators (stakers or

governance). Maybe a crazy MEV opportunity comes along and the delegate can make a bunch of money by acting maliciously. They could deviate no matter what we use (note that designs such as MEV-burn can actually help mitigate this "rug-pooling"). The difference is that PoG accepts this PAP and can choose not to extend it to delegator funds.

So, is there any real "economic security" in PoS if it's not even the operator's money at stake? Well, it just depends what you mean exactly. It just isn't quite a 1:1 parallel with the "economic security" in the PoW context.

It's reasonable to argue that PoS slashing (even if it's all delegated) is still "costly" in the sense that validators don't want to lose money for their delegators. But let's be clear - that's very different from actually slashing their own

money. Anatoly recently posed this question:

What's massively under explored in crypto is why there are so few quorum attacks (if any) on pos networks. Empirical evidence suggests that honestly majority assumptions are actually fine. So why is that? My theories

1. Quorum threshold is high, 67%
2. Every member votes on…

— toly (@aeyakovenko) January 27, 2023

This resurfaces many of the discussions around why we still haven't seen reorg-geth.

There's broadly two somewhat related incentives for the PoS delegates here to act honestly:

1) Legal Liability

- There's an argument that validator delegates could have legal liability. Although there isn't necessarily a clear legal contract between the delegator (staker) and delegate (operator), double-signing would provably violate the rules of the protocol which the user was expecting them to follow. (Not legal advice, I'm not a lawyer, I really don't know). I personally wouldn't feel so great using a PoS chain if the validators weren't publicly known.

2) Reputation

- This is a repeated game. Validators have long-term businesses which stand to profit from their continued services. If they nuke their reputation, that business is dead. If they are purely rational actors, they will act honestly so long as:

Reputational value > Profit from corruption

If we view "reputational value" as a strict calculation, then you could rephrase this as:

DCF of future validation profits > Profit from corruption

This gets even trickier to overcome because it's not isolated. In reality, you would need many operators to collude (enough to comprise the majority of voting power), all of whom lose their reputation. This increases the cost of attack to the sum of their reputational value. There can also be related business lines which would suffer. I imagine it wouldn't be great for Coinbase's businesses even outside of staking (and thus their stock price) if they started re-orging chains and nobody wanted them as a validator anymore. I'd probably take my trading elsewhere.

This is all to say that reputation is real. We often treat it like some handwavy thing that doesn't exist vs. treating delegated stake as perfectly secure and real. The reality is that it can be quite the opposite. Reputation and associated future cashflows are real assets with real value today, and those are the assets that the operator actually has at stake personally.

This was a point that Nic also touched on in the PoW context

To sum up, outbidding the set of honest miners dutifully producing blocks on Bitcoin is very expensive. They collectively take a salary of $6.9 billion dollars per year right now, and many of them have presumably invested in their businesses in anticipation of future cashflows (meaning that the hardware active on the network might be even higher than current miner revenue would imply).

So Bitcoin is protected not only by the daily salary that the protocol pays its miners, but by the discounted rewards these miners expect to earn in the future. This means Bitcoin isn't just protected by the reality on the ground today, but miner expectations about rewards in the future.

We don't have an easy way to model expectations, so the easiest thing to do is to simply take the miner salary per unit time and compare blockchains on that basis.

In summary (didn't include legal liabilities, because I'm not a lawyer):

- PoW Miners

- There's an upfront energy cost to revert blocks, and the assumption is that re-orging the chain would also reduce their

future revenue (i.e., the chain would become unstable and not valuable).

- PoS Validators

- There's no upfront energy cost to revert blocks, but re-orging the chain will burn stake (though this is likely delegated and thus not their money). Similarly, validators will lose out on future cashflows they would have otherwise earned (they can be removed from the validator set).

This is my general subjective opinion:

- Explicit Upfront Cost to Re-org

- A lot of the mining pool operator's hash power can be outsourced, but PoW is pretty cool here and can be pretty responsive on removing delegation. PoS is almost entirely delegated in practice and removing delegation can be more difficult. Burning this stake would only be a cost for the operator to the extent it results in legal and/or reputational consequences.

- Long-term Cost to Re-org

- PoS & PoG are really cool here, especially for publicly-known operators (most of them). Double-signing is a uniquely attributable and cryptographically provable fault. This is a huge plus. This is an iterated game where reputation and future value matter a lot. However, this is not the case for the long-tail of operators (e.g., solo stakers) where the upfront profit from corruption can easily outweigh long-term reputational loss (if any). PoW lacks this level of provable attributability, and there have been accusations of large miners playing games such as messing with timestamps or re-orging blocks on short time horizons where there's plausible deniability of who was first.

Overall, I don't think that PoG's lack of slashing is actually the loss it may seem at first glance. It seems completely unrealistic in practice that the best way to attack a PoS network is to literally go and try to buy up potentially billions of dollars of stake. PoS slashing just becomes a PAP with separation of capital and labor. If PoS funds are all delegated, then the cost of performing a re-org in PoS is:

- Reputational

- Very real and valuable. Attackers lose all future cashflows. Given many reputable operators, this is a very high cost and difficult to coordinate.

- Potentially legal liability

- If you buy the argument that operators are legally liable for funds that they get slashed on behalf of their users (again not legal advice, I literally have no clue), then getting slashed gives them an eventual legal bill.

- Coordination

- It's simply hard to get a bunch of people together to agree on colluding for something like this if the majority of voting power is distributed well.

Reputation exists equally in both PoS and PoG. Legal liability can be implicit or explicit in either scenario as desired (you can imagine simply having operators sign onchain legal agreements with offchain recourse if they ever double-sign). Subjective operator selection increases the coordination cost if it's able to get a better distribution needed to reach a majority of voting power.

If it makes you feel better, you can have operators put up a relatively small bond (maybe somewhere in the thousands of dollars, to the point where it's painful to lose but not worth them raising delegated funds out-of-protocol). If the required bond is too large or uncapped, then you just push stake delegation out-of-protocol. LSTs can pop up exactly as they have played out on Ethereum.

Overall, having ~$40bn currently staked or a trillion dollars staked makes no difference to these.

The cost of corruption

is one half of the equation, but now let's consider the other half - the profit from corruption

. I also recommend a great post byViktor Bunin from a few years back - Proof of Stake's security model is being dramatically misunderstood

.

Some of the often-cited profit motives for reorg-ing a chain tend to include:

- MEV

- there was a gigantic arbitrage in one block, and you want a piece of it

- Shorting

- open a huge short position, attack the network, price falls

- Exchange double-spending

- send money to an exchange, cash out, then reorg

- Honest majority bridge exploit

- lie to the other side of the bridge and print money (e.g., this may have been rational as the price of LUNA plummeted)

In practice, these attacks tend to be unrealistic in most cases. Even more importantly in the context of rollups - sequencers are simply relied upon for less than base layer validators. Rollups are not susceptible to these reorgs once the data is on the base layer. At that point, the base layer (e.g., Ethereum) would need to reorg for the rollup blocks to revert. The ability to reorg (and profit from doing so) is constantly being reset from the rollup's perspective. Rollups can also prevent honest majority bridge exploits.

Lastly, we might consider an attacker who's simply trying to break shit. They control delegated funds, and they're willing to self-sabotage to get everyone else's stake slashed.

An attacker could manage to take control of a large portion of stake and perform a ransom attack.As Justin Drake recently described on the latest EF Research AMA:

Staking comes with risks that should be taken into account. In September 2021 David Hoffman asked "Why aren't you staking all of your ether?" and my answer (see here) was "When you make the sausage you know how it's made." One of my main worries is ransom attacks (see more detailed explanation below). Ransom attacks are just as relevant today and I'm not sure my 3% APR is worth the risks. (I should note that Puffer's secure signer dramatically reduces the risk of ransom attacks for solo stakers and I may start using that to stake most of my ETH.)

I do believe staking is riskier than most people perceive it to be and that being aware of the risks is healthy. Tail risks are especially easy to underestimate because years can go by with stakers happily earning a substantial yield and then suddenly, out of nowhere, many stakers see themselves losing a large portion of their stake.

The top staking risk is IMO a so-called "ransom attack". Suppose an attacker gets hold of X% of staking keys. (Staking keys are "hot keys", i.e. private keys connected essentially 24/7 to the internet---they are significantly more exposed than withdrawal keys held in cold storage.) The attacker can now setup a smart contract which will trustlessly slash 3

X% of the stake of the compromised validators unless a ransom is paid.*

Let's consider a concrete example. Imagine there's a rogue sysadmin within Coinbase Cloud who manages to get hold of the staking keys for the 10% of Coinbase Cloud validators. (Notice that the rogue sysadmin doesn't need access to the withdrawal keys held in cold storage.) The sysadmin is now in a position to slash 3

10% = 30% of Coinbase's validators (roughly 0.63M ETH, or $1.2B). The attacker now sets up a smart contract to trustlessly slash the ETH unless a $1B ransom is paid. The rational move for Coinbase Cloud (and possibly its fiduciary duty) is to pay the $1B ransom (recuperating $200M of the $1.2B), and Coinbase Cloud users see themselves losing 25% of their stake in one fell swoop.*

There are other scary scenarios where an attacker can get hold of a large percentage of staking keys and perform a similar ransom attack:

1. An inside job in one of the top consensus clients. For example, a rogue Prysm or Lighthouse dev could insert some subtle bug or backdoor.

2. A supply chain attack targetting one of the libraries used by Prysm or Lighthouse. This could be combined with an inside job for plausible deniability.

3. An accidental remote code execution in a particular operating system. Apple now routinely posts Rapid Security Responses, often in response to actively exploited bugs. Linux and Windows likely also suffer from crippling 0-days.

Ransom attacks turn Ethereum staking into a multi-billion dollar bug bounty program. 0-days previously sold for millions of dollars on the dark web could now be weaponised for hundreds of millions.

These tail-risks increase when slashing risk is packaged up and distributed with financial products (i.e., LSTs) which lay the foundation of DeFi. This risk is further heightened with the advent of restaking, which can subject this stake to many additional slashing conditions external to the core protocol. These harmful actions may even be accidental - simple errors resulting in slashing are more likely when adding on new services. This is precisely why safeguards will be used, such as councils with the ability to veto accidental slashing and other mechanisms to prevent accidental double-signing.

Overall, it seems that reputational and legal risk are the primary assets really at "stake" here for operators (i.e., the cost of corruption). The profit from corruption is generally too difficult to achieve in practice regardless. Self-inflicted attacks (e.g.,

ransom attacks) may be more likely than the more often-cited profit motives for a majority attack (e.g., double spending). While staking is intended to increase the cost of corruption

, it can also increase the profit from corruption

. Removing staking and associated financial products layered on top reduces that leverage and tail-risk when they aren't needed.

Accountability - Slashing vs. Removal

I would then argue that the more important accountability mechanism in PoS > PoW is not the ability to burn funds, but rather the ability to remove the attacker in a targeted manner

. PoW requires the nuclear option (brick the hashing algorithm). PoS allows you to more precisely remove the attacker. PoG retains the same ability to remove only the attacker. You don't light their delegates' money on fire, you just kick the bad actor out.

This also makes socially forking to remove these censoring validators meaningfully easier in practice. The classic hypothetical example was the debate that followed the onset of the OFAC-related censorship issues last year:

Question for the Ethereum community. If a majority of stake chooses A in this poll, will you:

X) Consider the censorship an attack on Ethereum and burn their stake via social consensus

Y) Tolerate the censorship https://t.co/Mf48co37jK

— Eric Wall (@ercwl) August 15, 2022

There was strong consideration that if the majority of validators began actively re-orging blocks with OFAC transactions, they should be forked out. This presumably would've been done via a user-activated soft fork (UASF) toinactivity leak them out (though a hard fork to simply burn their funds would also be possible).

However, this of course would be a highly contentious debate with severe consequences. Would we actually be willing to destroy the majority of users' funds on this fork? Should we really be penalizing every random user who delegated their ETH to a large validator that complied with regulators if they were mandated to censor?

If large validators weren't just custodying entirely users' delegated funds, I would argue that a UASF would be far cleaner and less contentious. Simply fork and remove the censoring operators if you don't approve of them any longer. Removing stake from the equation (effectively held hostage) empowers governance (offchain social consensus in Ethereum's case) to easily hold the operators alone accountable.

In the same way that PoS enables more targeted accountability than PoW, I think the same can be said for PoG over PoS. You can simply excise the attackers' personal assets

(their reputation, i.e. future cashflows) rather than also inflicting collateral damage on their delegators' assets

(i.e., slashing users' funds).

This can increase the credibility of such a threat.

The credibility of what the protocol is willing to defend was explored more broadly inBarnabé's phenomenal article Seeing like a protocol - Where does protocol credibility come from?

:

To create credibility here, casting the spell of extending the protocol's boundaries with further introspection and agency is not enough, if other institutions work towards eroding credibility by shifting incentives of the protocol's agents. Credibility is fully obtained whenever it is clear that the community is willing to defend the protocol's preferred outcomes in the last resort

, forcing the outcomes it cares about even at the cost of forking out part of the validator set.

Removing Censoring Delegates

Let's now look at the specific mechanics of removing operators guilty of double-signing - the fundamental in-protocol violation for a consensus protocol. There are two subsets of this - proposing multiple blocks in a single slot (equivocation) and submitting contradictory attestations.

- PoS

- Anyone can present evidence onchain (conflicting signatures on two blocks at the same height) to get the attacker slashed This is programmatically slashable in PoS, and the attacker may be exited from the active validator set.

- PoG

- Similarly, delegates could simply be forcibly exited from the validator set in PoG if evidence of double-signing is presented onchain. For example, this "tombstoning" would be the punishment on Noble's PoA chain in the event of double-signing

This penalty provides varying degrees of weight depending on the scenario:

- Proposer equivocation

- The proposer commits to one block (e.g., sign a header sent by a relay) then equivocates to sign another block in the same slot. Proposer slashing for equivocation (e.g., to perform an unbundling attack) is a rather weak deterrent unfortunately. The profit from attack (millions in the case of the Low-Carb Crusader) can easily outweigh the small slashing penalty against some of their 32 ETH balance. This can be addressed by more robust mechanism design which makes these games nearly impossible.

- Consensus double-signing

- Once the rest of the consensus set signs on a proposer's commitment, you now have a much stronger commitment. Now you can hold many validators accountable if all attesters sign off on conflicting blocks. If two forks independently meet the ⅔ quorum typically needed to finalize, then at least ⅓ of all stake must have signed on both forks. This ⅓ balance is potentially slashable. The signatures can be presented onchain to prove double-signing.

Social consensus generally enforces censorship resistance (CR) on a long time horizon via the threat of forking. This could be a hard fork to explicitly remove the offenders, or it may simply be a user-activated soft fork (UASF) to inactivity leak them out as mentioned earlier. In any case, the big difference here is that social consensus must step in to enforce long-term CR vs. double-signing is automatically slashable because it is provable in-protocol.

However, an often overlooked point is that a malicious majority of delegates can simply censor the transactions which prove double-signing

. If an attacker controls ⅔ of voting power (in either PoS or PoG), they could simply ignore and build around any blocks which would include the transactions to prove double-signing and have them slashed. Automatic slashing for double-signing is most valuable in the event of a malicious majority attack, but that's also unfortunately when it's most likely to fail.

Similarly, the malicious majority could censor the slower paths to remove them:

- PoS

- Attackers can refuse to include any transactions from delegators attempting to remove their delegated funds from the operator.

- PoG

- Attackers can refuse to include any transactions from governance attempting to vote them out as operators.

This is where one of the superpowers of rollups comes in - you can forcibly remove the malicious operators via the rollup's base layer

. Rollup forced inclusion mechanisms are often discussed in the user mass exit scenario (where they bridge out all funds because the sequencer is censoring them). However, this forced inclusion method could be used to remove the sequencer

.

- Monolithic blockchain

- Majority censorship would require social consensus to kick in and remove them

- Rollups

- Because rollups can delegate their CR to another layer (their DA layer), they can use this to forcibly remove their operators onchain (even if they're malicious)

Decentralization

Regardless of the staking design, we see a clear separation of capital (stake) and labor (validator operators). So let's ask what PoS is providing then:

- ✅ Sybil Resistance / Delegation

- Users elect operators they trust by delegating stake to them. If this is managed by an LST provider, then the LST's governance in turn is actually choosing the delegates.

- ✖ Slashing Penalty

- The slashing penalty no longer directly applies to the operators themselves if funds are delegated. The delegators' money is at risk.

We're back to that point I made earlier - PoS is fundamentally used for electing delegates. The ability to burn funds is an optional feature. Staking forces delegators to have skin in the game though, incentivizing them to pick good operators so they don't get slashed. If PoS is primarily about picking delegates, then we need to ask what will yield the better decision:

- DPoS

- Token holders vote for themselves

. They directly vote on personal funds. Tragedy of the commons (i.e., top-heavy stake distribution).

- Governance (PoG or LST)

- Governance votes for the collective interest

. Voters do not put personal funds at stake. They vote for the best overall composition, not just the best delegate for their personal funds.

WTF is decentralization though? Simon Brown recently gave a great presentation at Flashbots' PBS Salon onMeasuring the Concentration of Control in Ethereum.

For our purposes here, I'll keep it simple and look at two high level points:

1) Number of total validators

- The total number of consensus participants. This says nothing about their distribution of voting power.

By this measure, the below voting power distribution would be very "decentralized":

2) Distribution of voting power

- This is mostly commonly measured by something like the Nakamoto coefficient, but broadly we're talking about how distributed is the top chunk of stake (whether ⅓, ½, ⅔, etc.). We don't consider the bottom minority stake here, we just look at how many people need to collude to cause a liveness/safety failure.

By this measure, the below voting power distribution would be very decentralized:

Ethereum is undoubtedly decentralized on #1 - there areover 700k active validators. Even though the vast majority of them are run by shared entities, the number of actual operators running Ethereum consensus is still in the single-digit thousands. That's still incredibly high. "However, "solo stakers" (running <200 validators and not part of a major organization) comprised ~6.5% of Ethereum's total stake from the Merge through the end of 2022. That number also seems likely to decrease over time in my opinion.

The importance of long-tail validators is a point of common debate, but it's generally viewed as desirable in Ethereum. The scenario where this long-tail may be particularly valuable is for a liveness-favoring chain that's going through a doomsday-type scenario. Ethereum of course optimizes for these conditions.

For example, let's say there's a big war going on and major countries partition the network. Or maybe the minority of validators actively choose to initiate a UASF to remove a censoring majority of validators. In either case, this long-tail of validators would continue to progress the network while the majority gets inactivity leaked out. The network would remain live at all times, and it would begin to finalize once the majority of stake has been leaked out. This affords Ethereum the option of removing attackers via UASF while retaining liveness (as opposed to needing to resort to a hard fork).

While the above is a fun point of debate, this scenario doesn't apply for the vast majority of chains. For any chain that favors consistency (safety), your chain will just halt if a quorum isn't reached. It doesn't make any difference if you have some at-home stakers comprising a few percent of the total voting power. Rollups will generally fall into this bucket (as will most chains).

Rollups are likely to make the opposite tradeoffs,complementing Ethereum's consensus. They will tend to favor fast finality (optimistic responsiveness) rather than dynamic availability (which Gasper achieves). Think something that looks more like Tendermint.

A long-tail of validators may also be helpful for turning "strong censorship" (never included) into "weak censorship" (included, but with a delay). For example, even if 9 out of 10 validators are censoring, then you'll on average be included by the 10th block. However, note that this can be prevented if the censoring majority decides to simply ignore the minority-produced blocks, as they have enough voting power to finalize without them. Additionally, rollups should already inherit "eventual" censorship resistance (CR) from their base layer. Sequencers are primarily tasked with real-time

guarantees, so the distribution of the majority of voting power is what actually matters.

Overall, I don't think this long tail of at-home sequencers would add much to most rollups, and they're not likely to even exist often (especially in a world with many rollups). Rollups are already paying Ethereum consensus for this long-tail decentralization and disaster scenario recovery among other things.

Back to those pie charts I showed you earlier, that second one was actually the distribution of stake under the hood allocated amongst Lido operators:

This is exactly the kind of distribution that's optimized for real-time

security. While PoS can easily support a long-tail permissionless validator entry, it tends to result in a top-heavy validator selection as described above. Subjective governance processes which decide as a collective can optimize for the best overall set of delegates. Protocols like Lido can and do impose strict requirements on decentralization within their operator set (e.g., stake distribution, geographical distribution, client diversity, cloud services, etc.).

PoG is pretty cool. once you except that the quorum is only there to provide liveness/censorship resistance, and that CR is observable, it changes the design space. Pick the fastest quorum that is still censorship resistant.

— toly (@aeyakovenko) June 22, 2022

The endgame for decentralized Ethereum validators is likely to take the shape of something like Lido's Staking Router. This upgrade allows for running any arbitrary staking "module" under the hood for Lido, including those which would allow for long-tail bonded validators.

Historically, Lido has been very simple and scalable. Users' ETH deposits are delegated to a curated Node Operators Registry (NOR) - that's the pie chart you saw above. While this is highly capital efficient, it's not very flexible in allowing for different staking models.

With the introduction of the Staking Router, the NOR becomes the first module within Lido. Anyone can make a proposal to the Lido DAO to onboard any other module. For example, this may be a module where stakers run DVT. These proposals would come with the associated technical plans and business cases for the DAO to approve. They can even have independent fee mechanisms (rather than the standard 90% / 5% / 5% split between stETH / operator / Lido DAO for the NOR). The Lido DAO will assess these proposals and decide on the stake delegation between different modules.

As with dual governance, LST protocols have a balance of incentives here:

- Economic efficiency

- It's simple and scalable to run all stake under one entity (e.g., Coinbase's cbETH)

- Decentralization

- It's more complicated to manage a large set of trusted operators, a module for untrusted DVT stakers, etc.

However, a decentralized LST protocol that's aligned with the underlying chain may be able to safely grow further. As Hasu described on a recent episode of Bell Curve:

"Decentralization for many systems can be seen as a form of sacrifice... but in staking protocols, this does not apply... The more decentralized you can make it, the better it gets, because it's so much about neutrality and trust. And this is why for Lido decentralization is not a sacrifice of anything, it's extremely good... It's the number one priority in the sense that it's not a defensive move. Decentralization for Lido is offense."

LST governance is able to handle the functions that Ethereum is unable to. In particular, LST governance can manage the additional subjective incentives required to decentralize operators (e.g., different modules may receive different fee rates). Free market economics do not lead to a long-tail of solo stakers or uniform stake distribution in the long-run.

The Ethereum core protocol is largely built on the notion that it should be objective and un-opinionated whenever possible. However, subjective management and incentives will be required to achieve a decentralized operator set.

While governance minimization is often desirable, some minimal form of governance will always likely be necessary for LSTs. Some process is needed to match the demand for staking with the demand to run validators. LST governance will always be needed to manage the objective functions of the node operator set (e.g., targets on stake distribution, different module weighting, geography targets, etc.). This fine-tuning can be infrequent, but this high-level goal setting is critical for monitoring and maintaining the decentralization of the operator set.

Delegation Control

All of these are mechanisms for electing delegates, so we need to consider who

is given the power to choose the delegates here:

- DPoS

- Token holders vote with their feet via direct delegation. This is decent alignment of interest - you're at least incentivized to vote for a good operator. However, it's often incredibly centralizing - users tend to select the largest operators.

- LSTs

- LST governance will now decide how to delegate the funds to stakers instead of the actual LST holders. This is problematic - LST governance and rollup governance may be misaligned.

- LSTs + Dual Governance

- The issue above can be greatly mitigated with a dual governance proposal as has been discussed for Lido. This proposal would provide some mechanism for Ethereum-aligned participants (e.g., stETH holders) to veto actions taken by the Lido DAO. This provides LST holders meaningful agency to protect the consensus layer against a potentially misaligned LST DAO. Sacha had a great presentation on the topic here.

- PoG

- Rollup governance just elects the delegates themselves. This gives them the highest control and incentive alignment. This also preserves the ability for a rollup to use a governance mechanism other than simple majority token voting. For example, Optimism is experimenting with creative governance mechanisms that go beyond simple token voting.

I view some form of dual governance as necessary for an LST protocol to be viable long-term. However, it's still imperfect, as noted by Danny Ryan in The Risks of LSD

:

"It is important to note here that ETH holders are not by definition Ethereum users, and in the long run, we expect that there are massively more Ethereum users than ETH holders (people with ETH held beyond the amount needed to facilitate TXs). This is a critical and important fact that informs Ethereum governance -- there is no on-chain governance granted to ETH holders or stakers. Ethereum is the protocol that users choose to run.

ETH holders in the long run are just a subset of users, so staked ETH holders are even a subset from there. In the extreme of all ETH becoming staked ETH under one LSD, governance vote weights or aborts by staked ETH do not protect the Ethereum platform for users.

Thus even if the LSD protocol and the LSD holders are aligned on subtle attacks and capture, users are not and can/will react."

However, as Hasu pointed out in Do stakers represent users?

:

First, the observation "users and block producers are not the same group" is completely orthogonal to whether stake is delegated or not. It's simply a property of PoS and all other known Sybil resistance mechanisms (except maybe proof of humanity which is unworkable for other reasons).

TLDR: "Stakers don't represent users" is a true observation but one that exists in PoS and all other consensus systems. LSD-PoS is not to blame and in fact makes the problem better, not worse.

PoS gives the delegation control to token holders, not the superset of users or the chain's community. I think most of us would agree that simple token holder majority governance isn't the ideal solution either:

Decentralized governance is necessary, but coin voting governance in its current form has many acknowledged and unacknowledged dangers. Augmenting or moving beyond coin voting is a key part of the solution:https://t.co/pZQ4sLAbEy

— vitalik.eth (@VitalikButerin) August 16, 2021

However, PoG allows your chain to use any

governance process to make the decision. Optimism recently announced its plans to give its Token House powers over sequencer selection while the Citizens' House (identity-based governance) has veto rights.

Governance should also have the power to quickly swap out the elected sequencers if needed (e.g., if they start acting counter to governance's desires).

Allowing rollup governance to directly control the sequencers appears meaningfully beneficial. This also allows rollups to experiment with creative (less plutocratic) governance mechanisms for delegation other than simple token voting.

Even if simple rollup token voting governance is used to select operators, this will often still be preferable to staking. Stakers are incentivized to pick the biggest validator when it's their own money in native delegation. Subjective governance

processes such as what Lido does are far better for operator decentralization.

The question then is whether rollup governance should handle operator selection or implement PoS and let LST protocols take the reins. You could also get creative with dual governance if you go the PoS route. While Ethereum has no onchain governance that Lido can give veto rights to, LSTs elsewhere may instead grant veto rights to their onchain governance mechanism (rather than just the LST holders). One argument for PoS over PoG is that LSTs at least "free-market" and there can be several, though in practice LSTs are likely to be winner takes all/most.

For a rollup with competent enough governance mechanisms, simply bringing the delegation in-house seems desirable. Outsourcing power over consensus layer operations seems unnecessary. Go pick a bunch of people you trust in your community, go handpick a bunch of geographically decentralized ones, etc. The absolute minimum is likely to be a single default sequencer live at a given time with a list of pre-elected hot backups ready to automatically swap in if there's any CR or broader liveness issues.

The real value-add of LST protocols here in my mind is the ability of a specialized protocol (e.g., Lido) to pick all your operators for you in the event that your own governance simply doesn't want to or is too incompetent. This is a valuable service that can be replicated without staking and delegation of funds though.

PoS vs. PoG Economics 101

Let's consider two hypothetical scenarios for a rollup:

- LST + Dual Governance

- An LST protocol captures all stake, and they pick delegates to operate the rollup. 90% of rewards go to LST token holders (e.g., stOP), 5% goes to LST governance (e.g., controlled by LDO), and 5% goes to the operators (e.g., Coinbase).

- PoG

- Governance elects the exact same delegates that the LST protocol would have selected. They make a direct agreement to pay out inflationary rewards equivalent to what they would have made as LST operators. All fees and MEV are burned.

Here are some hypothetical numbers to illustrate the major differences:

In both scenarios:

- Income

- $50mm of fees and other MEV in excess of what it pays to the DA layer

- Operator Costs

- Operator set requires $30mm p.a. (LST scenario - paid as 5% of all rewards. PoG scenario - paid via lower inflationary rewards going only to them).

For the differences, PoG comes out ahead. LST governance takes 5% of all rewards here ($30mm). This is their payment needed to manage a responsible operator set, fund operations such as possibly incentivizing LST liquidity, take a profit, etc. If the rollup has its own governance though, it could internalize this work rather than leak value out to LST protocol governance. Other staking operators like Coinbase (for cbETH) charge a whopping 25% fee rate. There's no reason for a protocol to be subsidizing this amount of value for large operators.

Overall, the PoS scenario pays out $30mm to operators + $30mm to LST governance = $60mm. They only capture $50mm in revenue in excess of DA layer fees. The $10mm shortfall is borne by all token holders via excess issuance.

Rather than distributing fees and MEV to stakers, the PoG scenario burns all of them (you could equally imagine them sending it to a rollup treasury, funding public goods, or whatever else they want). Because this is $20mm greater than the inflationary rewards paid to operators, the rollup token is net deflationary at the end of the year. Assuming a flat market cap, the token price increases.

Additionally, high payment of issuance rewards consistently leaks value.Staking rewards may be treated as income in many jurisdictions (not tax advice). To the extent that they are, you'd consistently leak value to taxes and associated sell pressure. Whatever is paid out will potentially be taxed at a high rate. Minimum viable issuance is maximally tax efficient for keeping value within a given ecosystem.

Finally, staking rewards overall lead to concerning distributions of wealth. They unnecessarily hand value to large capital holders, tax collectors, and institutional staking operators. These are exceedingly plutocratic mechanisms with even worse wealth distribution than present-day systems in many ways. We often don't like to hear this in crypto, but a more creative and potentially subjective distribution of capital is an absolute must (e.g., Optimism's RetroPGF). Blindly giving rewards to large stakers who don't actually contribute to the economic growth of the system is simply a poor use of capital.

It's also an odd incentive. I am clearly incentivized by a PoS protocol to go give my money to a staking provider that then

gives me an LST in return. They will hand my stake to some delegate with no skin in the game, and I go play with my financial product. Did I actually contribute to the chain's "economic security" in this case? Or am I just being paid for taking on smart contract risk and levering up the system?

The primary value-add from PoS here is in my mind not the "economic security of slashing," but rather the services offered by LST protocols. And I'm not talking about the liquid staking token

. That's not their product here. Their product is outsourced management of a decentralized and robust set of operators. Ethereum has no governance mechanism in place to do this. LST protocols can do a far better job than free market economics would lead to, and they'd potentially do a far better job than many chains' own governance mechanisms would as well. Governance is hard.

Yudkowsky was right we're all gonna diehttps://t.co/5SOeHcI6IX

— Jon Charbonneau (@jon_charb) April 9, 2023

Endgame - PoS vs. PoG Summary

PoG and PoS end up with striking similarities if you play out the economic incentives to their logical conclusion:

- Total separation of capital (stake) and labor (operators)
- Operator selection and delegation is managed by governance aligned with the chain (e.g.,dual governance or PoG)

Let's walk through that table again and recap the merits of each approach here (from the perspective of a "typical" rollup):

- Capital Efficient

- DPoS is super capital inefficient. LSTs alleviate the vast majority of this inefficiency. The free market tends toward economic efficiency over time. PoG is optimal on all dimensions.

- Permissionless

- PoS allows for "permissionless" entry in that anyone can buy up stake and become a validator (this is a spectrum though, subject to validator caps, staking queue, etc.). PoG is "permissioned" in that you need governance approval to become a block producer.

- Real-time CR & MEV Protection

- Known operators can be hand-picked to meet your requirements then held accountable. This empowers them to enforce guarantees such as real-time CR and MEV protection which may not be attributable in-protocol. Completely permissionless and untrusted validators tend to push these chokepoints offchain to unaccountable actors.

- Governance

- Some form of governance is of course required in PoG to manage the operator set. This is undesirable for chains like Ethereum that try to be as hands-off and neutral as possible. This seems to be less of a concern for most other chains though which are likely to have some form of governance whether onchain or offchain, however minimal.

- Accountability

- PoS allows you to slash operators' money who are acting in an undesirable manner vs. PoG only allows you to remove them. The latter seems less "economically secure" on the face of it, but the reality is that funds are effectively all delegated → operators' real personal "stake" is primarily reputational (future cash flows) and possibly legal (but idk).

- Decentralization

- It's possible for certain PoS designs to more naturally support a long-tail of validators which is potentially helpful for CR and liveness. However, subjective targets set by governance (either an LST's or the underlying chain's) can enforce far better stake distribution on various metrics, increasing real-time CR and liveness. Free market economics otherwise consolidate stake. Note that it would also be technically possible to build functionality similar to Lido's staking router into a given chain's own core functions.

- Delegation Control

- You can give this power to stakers (DPoS), external LST governance, or your own governance (which can be whatever you want!). The latter often seems preferable to me. If external LSTs are to be used though, they require significant attention to minimize and align their governance with the underlying chain.

- Economic Overview

- Staking can lead to excess value leaked to delegates, LST providers, and the tax man. This seems like a generally

inefficient and poor distribution of wealth. PoG gives your chain a chance to get more creative and less plutocratic.

Restaking & PEPC

So, if I'm questioning the fundamentals of staking

, what does that mean for restaking

?

The Use Cases for Restaking

In this section, I'll first summarize Sreeram's recent description of EigenLayer as a "marketplace for decentralized trust" on another great episode of Bell Curve. He describes three elements of decentralized trust that restaking can offer to Actively Validated Services (AVSs = services provided by EigenLayer restakers):

1) Economic security

- Typical PoS staking economics. You could restake say $1bn to secure some application. If the restakers misbehaved, this can be slashed. You don't care much whether this is 1 person with $1bn staked or 1 million people with $1,000 each staked.

2) Decentralized operators

- You want many distributed nodes that make collusion difficult. This could be useful for something like Shamir secret sharing - you distribute pieces of a secret to many participants, and a majority of them is needed to put it back together. You can't slash them because this is a non-attributable fault if they collude.

Note that "economic security" and "decentralization" aren't at all unique to Ethereum restakers. You could equally bootstrap these in some other way, even using assets like RWAs as collateral.

The specific relation is primarily due to the fact that Ethereum happens to have the largest amount of staked assets sitting around, so it's easiest to piggy-back off of it. If I've got some stETH sitting around, you probably won't have to twist my arm to go restake it. Some delegate can take my money, use it to validate an AVS, and make me more money. If you want to bootstrap some capital for securing your AVS, you can get it pretty easily.

Pooling security is often viewed as beneficial. The argument is that it's much harder to attack a committee with $100bn at stake vs. a committee with $50bn at stake.

While we can see that restaking could help bootstrap staked assets

, it's less clear that restaking helps to easily bootstrap decentralized operators

. As a delegate with some stETH, I just care about capital costs. My additional capital cost is 0 if I'm already holding some stETH. However, the operator's additional cost is unchanged.

It's just as much work for an Ethereum validator to integrate some random unrelated new AVS using restaking vs. if that application was launched as a self-secured app-chain. They need to get a team up to speed, get the software running, monitor it over time, etc. There are real overhead costs and ongoing work associated with this. Whether it's restaked or self-secured makes effectively no difference to their operational burden.

3) Proposer commitments

- This third point now is unique to Ethereum (or more generically, unique to the chain of the restaker making the commitment). Restaking is clearly valuable when you want a credible commitment from a specific chain's proposer. For example, you may want the Ethereum proposer to commit to follow a certain transaction ordering in their block, sell you the first N% of the block, commit to triggering liquidations on a lending protocol under certain conditions, etc.

Protocol-Enforced Proposer Commitments (PEPC)

Those proposer commitments via restaking are also precisely the idea behind Barnabé's PEPC (protocol-enforced proposer commitments). Note that PEPC is very much still in the research phase without a concrete spec, so the below is rather exploratory thinking at this stage.

Rather than having Ethereum validators opt into out-of-protocol commitments (e.g., via EigenLayer), the idea here would be to enshrine a mechanism that allows proposers to do this natively. Proposers would be able to opt into binding arbitrary commitments over the blocks they propose. This can be viewed as a generalization of PBS.

The critical difference between PEPC vs. restaking is the first half of that acronym - these proposer commitments would be protocol-enforced

. Attesters would reject a block as invalid if a proposer doesn't fulfill a commitment they opted into.

This is a much stronger enforcement vs. out-of-protocol restaking where the only recourse is to slash the proposer. The validity of the Ethereum block wouldn't be dependent on the proposer commitment with restaking. As mentioned with the Low Carb Crusader, the profit from deviating can outweigh the associated slashing costs. Note that restaking applications can circumvent Ethereum's 32 ETH limit though, requiring pooling of much higher stake to form a single operator of an AVS (e.g., maybe 10,000 ETH is required to be a validator on my new chain secured by restaking).

For more details, please see Barnabé's amazing new post on PEPC.

The Real Use Case for Restaking

With the background in place, I'll now provide my own opinion here:

1. Economic security

- As you can probably imagine based on my earlier arguments, I think this economic security point gets way too much attention. I don't get the sense that many AVSs actually care about this aspect that much either. You can also pledge any asset, not just ETH.

1. Decentralized operators

- This is clearly valuable, but as described above this has nothing to do with Ethereum. You can sign up any set of decentralized operators without them committing to also being Ethereum stakers. Ethereum is once again just a natural place to find a bunch of operators.

1. Proposer commitments

- This is clearly a valuable concept, as evidenced by exploration of PEPC. In the absence of an in-protocol mechanism, this makes a ton of sense for restaking.

So, I see the real use cases for restaking actually coming down to two buckets for the most part:

1. Technical (proposer commitments)

- There's no way around this - if you want a credible commitment to the ordering of an Ethereum block, it's gotta come from Ethereum validators. MEV-Boost++ is one example of a hypothetical application that could fall into this bucket.

1. Social

- I view Ethereum alignment

as the primary use case for most restaking applications today, not pooling of economic security or decentralization. It's getting to say "look we're an Ethereum project!" It's much the same reason why chains keep calling themselves Ethereum L2s regardless of the architecture.

you found the real use case for restaking

— Jon Charbonneau (@jon_charb) August 3, 2023

Being an "Ethereum-aligned" project or an "Ethereum L2" is seemingly more of a vibe than anything else at this point. That's not even a criticism (at least for the well-intentioned ones who are clear about it). Incentive alignment can be a legitimate point. These are incredibly social systems we're building, and everyone wants to sit at the cool kids table (especially when there's roughly one lunch table of users in crypto right now).

As an aside, this actually gives me quite a bit of comfort regarding the whole "restaking alignment" fear going around lately. This potentially self-selecting nature of restaking could mitigate the probability of tail risks. If AVSs use restaking primarily for the Ethereum alignment, then you're not going to use it if the whole Ethereum community would freak out at you for using it.

The centralization fear is that someone could launch an AVS that's super valuable but difficult to run operationally (e.g., a Solana fork). However, I think a lot of use cases would naturally filter out. I don't believe most of these AVSs mentioned today benefit from the "economic security" as much as we keep touting. It appears far more social. You'd be launching Solana on staking to say "hey look we're Ethereum aligned." If that's the case, this ceases to be a benefit (and indeed becomes detrimental) if using restaking would be viewed as misaligned with Ethereum.

This self-selection doesn't address technical cases where proposer commitments are necessary, or if some other application intentionally uses restaking maliciously despite negative social pressure. So it's not a solution by any means, but I do think that it directionally mitigates some of the fear of a creeping problem.

Looking further ahead, I believe it would be incredibly valuable to explore various forms of dual governance in the context of restaking.

Conclusion

PoG is far from perfect, and it's certainly not a fit for everyone. However, it's instructive to consider where it may be reasonable so that we can better understand the properties applicable to all chains.

I realize that saying "oh yea just let governance decide" isn't exactly the most confidence-inspiring thing given the state of most crypto governance.

However, the alternative is "oh yea just let free market economics hopefully result in decentralization" or "let someone else's governance do it for you."

I generally favor governance minimalism when possible. PoG is not an attempt to expand the reach of governance. It is rather an acknowledgment that this governance must necessarily happen regardless

.

The choice is simply whether that governance managing a chain's consensus operators should be outsourced to the free market (i.e., between LST protocols' governance) or in-housed within a chain's own governance. I think both approaches are reasonable under different circumstances.

Some minimal governance will always be needed to manage decentralization targets among operators and provide a fallback response in extreme scenarios. From there, the free market can handle the specific implementations and associated economics (e.g., as in the case of Lido's Staking Router).

I worry that many rollup teams' currently expressed plans for "permissionless" mechanisms optimize for the nice word but may fall very short in practice. Many designs such as variations of MEV auctions or prover races are likely to be extremely

centralizing in practice with entirely unaccountable out-of-protocol actors. The free-market economics left unchecked do not tend towards decentralized participants, and they may be hard-to-impossible to remove once in place.

Some of these designs appear to choose a poor place on the tradeoff spectrum - they take on Ethereum's current weaknesses (e.g., no MEV protection by default + relatively centralized out-of-protocol and unaccountable actors who present censorship chokepoints) without gaining the benefits that Ethereum derives from having made this tradeoff for the sake of permissionless-ness and neutrality.

Yes, the long-term answer to many of these issues is all the fancy ZK encrypted mempool threshold FHE one-shot signatures for distributed intent matching and blah blah. But we're not there today.

Plus, if we can figure out all this crypto governance stuff maybe we can use it to replace our governments IRL.

The author of this report has material personal investments in stETH, ETH, and EigenLayer.