Don't overload Ethereum's consensus

Special thanks to Karl Floersch and Justin Drake for feedback and review

The Ethereum network's consensus is one of the most highly secured cryptoeconomic systems out there.18 million ETH (~$34 billion) worth of validators finalize a block every 6.4 minutes, running many different implementations of the protocol for redundancy. And if the cryptoeconomic

consensus fails, whether due to a bug or an intentional 51% attack, a vast community of many thousands of developers and many more users are watching carefully to make sure the chain recovers correctly. Once the chain recovers, protocol rules ensure that attackers will likely be heavily penalized.

Over the years there have been a number of ideas, usually at the thought experiment stage, to also use the Ethereum validator set, and perhaps even the Ethereum social consensus, for other purposes

:

- The ultimate oracle

: a proposal where users can vote on what facts are true by sending ETH, with aSchellingCoin mechanism: everyone who sent ETH to vote for the majority answer gets a proportional share of all the ETH sent to vote for the minority answer. The description continues: "So in principle this is an symmetric game. What breaks the symmetry is that a) the truth is the natural point to coordinate on and more importantly b) the people betting on the truth can make a credible thread of forking Ethereum if they loose."

- Re-staking

: a set of techniques, used by many protocols includingEigenLayer, where Ethereum stakers can simultaneously use their stake as a deposit in another protocol. In some cases, if they misbehave according to the other protocol's rules, their deposit also gets slashed. In other cases, there are no in-protocol incentives and stake is simply used to vote.

- L1-driven recovery of L2 projects

: it has been proposed on many occasions that if an L2 has a bug, the L1 could fork to recover it. One recent example isthis design for using L1 soft forks to recover L2 failures.

The purpose of this post will be to explain in detail the argument why, in my view, a certain subset

of these techniques brings high systemic risks to the ecosystem and should be discouraged and resisted.

These proposals are generally made in a well-intentioned way, and so the goal is not to focus on individuals or projects; rather, the goal is to focus on techniques. The general rule of thumb that this post will attempt to defend is as follows: dual-use of validator staked ETH, while it has some risks, is fundamentally fine, but attempting to "recruit" Ethereum social consensus for your application's own purposes is not.

Examples

of the distinction between re-using validators (low-risk) and overloading social consensus (high-risk)

- Alice creates a web3 social network where if you cryptographically prove that you control the key of an active Ethereum validator, you automatically gain "verified" status. Low-risk

.

- Bob cryptographically proves that he controls the key of ten active Ethereum validators as a way of proving that he has enough wealth to satisfy some legal requirement. Low-risk

.

- Charlie claims to have disproven thetwin primes conjecture, and claims to know the largest p

such that p

and p+2

are both prime. He changes his staking withdrawal address to a smart contract where anyone can submit a claimed counterexample q > p

, along with a SNARK proving that q

and q+2

are both prime. If someone makes a valid claim, then Charlie's validator is forcibly exited, and the submitter gets whatever of Charlie's ETH is left. Low-risk

.

- Dogecoin decides to switch to proof of stake, and to increase the size of its security pool it allows Ethereum stakers to "dual-stake" and simultaneously join its validator set. To do so, Ethereum stakers would have to change their staking withdrawal address to a smart contract where anyone can submit a proof that they violated the Dogecoin staking rules

. If someone does submit such a proof, then the staker's validator is forcibly exited, and whatever of their ETH is left is used to buy-and-burn DOGE. Low-risk

.

- eCash does the same as Dogecoin, but the project leaders further announce: if the majority of participating ETH validators collude to censor eCash transactions

, they expect that the Ethereum community will hard-fork to delete those validators. They argue that it will be in Ethereum's interest to do so as those validators are proven to be malicious and unreliable. High-risk

.

- Fred creates an ETH/USD price oracle, which functions by allowing Ethereum validators to participate and vote. There are no incentives. Low-risk

.

- George creates an ETH/USD price oracle, which functions by allowing ETH holders to participate and vote. To protect against laziness and creeping bribes, they add an incentive mechanism where the participants that give an answer within 1% of the median answer get 1% of the ETH of any participants that gave an answer further than 1% from the median. When asked "what if someone credibly offers to bribe all the participants, everyone starts submitting the wrong answer, and so honest people get 10 million of their ETH taken away?", George replies: then Ethereum will have to fork out the bad participants' money. High-risk

. * George conspicuously stays away from making replies. Medium-high risk

(as the project could create incentives to attempt such a fork, and hence the expectation that it will be attmpted, even without formal encouragement)

- George replies: "then the attacker wins, and we'll give up on using this oracle". Medium-low risk

(not quite "low" only because the mechanism does create a large set of actors who in a 51% attack might be incentivized to indepently advocate for a fork to protect their deposits)

- George conspicuously stays away from making replies. Medium-high risk

(as the project could create incentives to attempt such a fork, and hence the expectation that it will be attmpted, even without formal encouragement)

- George replies: "then the attacker wins, and we'll give up on using this oracle". Medium-low risk

(not quite "low" only because the mechanism does create a large set of actors who in a 51% attack might be incentivized to indepently advocate for a fork to protect their deposits)

- Hermione creates a successful layer 2, and argues that because her layer 2 is the largest, it is inherently the most secure, because if there is a bug that causes funds to be stolen, the losses will be so large that the community will have no choice but to fork to recover the users' funds. High-risk

.

- George conspicuously stays away from making replies. Medium-high risk

(as the project could create incentives to attempt such a fork, and hence the expectation that it will be attmpted, even without formal encouragement)

- George replies: "then the attacker wins, and we'll give up on using this oracle". Medium-low risk

(not quite "low" only because the mechanism does create a large set of actors who in a 51% attack might be incentivized to indepently advocate for a fork to protect their deposits)

If you're designing a protocol where, even if everything completely breaks, the losses are kept contained to the validators and users who opted in to participating in and using your protocol, this is low-risk. If, on the other hand, you have the intent to rope in the broader Ethereum ecosystem social consensus to fork or reorg to solve your problems, this is high-risk, and I argue that we should strongly resist all attempts to create such expectations.

A middle ground is situations that start off in the low-risk category but give their participants incentives to slide into the higher-risk category; SchellingCoin-style techniques, especially mechanisms with heavy penalties for deviating from the majority, are a major example.

So

what's so wrong with stretching Ethereum consensus, anyway?

It is the year 2025. Frustrated with the existing options, a group has decided to make a new ETH/USD price oracle, which works by allowing validators to vote on the price every hour. If a validator votes, they would be unconditionally rewarded with a portion of fees from the system. But soon participants became lazy: they connected to centralized APIs, and when those APIs got cyber-attacked, they either dropped out or started reporting false values. To solve this, incentives were introduced: the oracle also votes retrospectively on the price one week ago, and if your (real time or

retrospective) vote is more than 1% away from the median retrospective vote, you are heavily penalized, with the penalty going to those who voted "correctly".

Within a year, over 90% of validators are participating. Someone asked: what if Lido bands together with a few other large stakers to 51% attack the vote, forcing through a fake ETH/USD price value, extracting heavy penalties from everyone who does not participate in the attack? The oracle's proponents, at this point heavily invested in the scheme, reply: well if that happens, Ethereum will surely fork to kick the bad guys out.

At first, the scheme is limited to ETH/USD, and it appears resilient and stable. But over the years, other indices get added: ETH/EUR, ETH/CNY, and eventually rates for all countries in the G20.

But in 2034, things start to go wrong. Brazil has an unexpectedly severe political crisis, leading to a disputed election. One political party ends up in control of the capital and 75% of the country, but another party ends up in control of some northern areas. Major Western media argue that the northern party is clearly the legitimate winner because it acted legally and the southern party acted illegally (and by the way are fascist). Indian and Chinese official sources, and Elon Musk, argue that the southern party has actual control of most of the country, and the international community should not try to be a world police and should instead accept the outcome.

By this point, Brazil has a CBDC, which splits into two forks: the (northern) BRL-N, and the (southern) BRL-S. When voting in the oracle, 60% of Ethereum stakers provide the ETH/BRL-S rate. Major community leaders and businesses decry the stakers' craven capitulation to fascism, and propose to fork the chain to only include the "good stakers" providing the ETH/BRL-N rate, and drain the other stakers' balances to near-zero. Within their social media bubble, they believe that they will clearly win. However, once the fork hits, the BRL-S side proves unexpectedly strong. What they expected to be a landslide instead proves to be pretty much a 50-50 community split.

At this point, the two sides are in their two separate universes with their two chains, with no practical way of coming back together. Ethereum, a global permissionless platform created in part to be a refuge from nations and geopolitics, instead ends up cleaved in half by any one of the twenty G20 member states having an unexpectedly severe internal issue.

That's

a nice scifi story you got there. Could even make a good movie. But what can we actually learn from it?

A blockchain's "purity", in the sense of it being a purely mathematical construct that attempts to come to consensus only on purely mathematical things, is a huge advantage. As soon as a blockchain tries to "hook in" to the outside world, the outside world's conflicts start to impact on the blockchain too. Given a sufficiently extreme political event - in fact, not that

extreme a political event, given that the above story was basically a pastiche of events that have actually happened in various major (>25m population) countries all within the past decade - even something as benign as a currency oracle could tear the community apart.

Here are a few more possible scenarios:

- One of the currencies that the oracle tracks (which could even be USD) simply hyperinflates, and markets break down to the point that at some points in time there is no clear specific market price.
- If Ethereum adds a price oracle to another cryptocurrency

, then a controversial split like in the story above is not hypothetical: it's something that has already happened, including in the histories of both Bitcoin and Ethereum itself.

- If strict capital controls become operational, then which price to report as the legitimate market price between two currencies becomes a political question.

But more importantly, I'd argue that there is a Schelling fence at play: once a blockchain starts

incorporating real-world price indices as a layer-1 protocol feature, it could easily succumb to interpreting more and more real-world information. Introducing layer-1 price indices also expands the blockchain's legal attack surface: instead of being just

a neutral technical platform, it becomes much more explicitly a financial tool.

What

about risks from examples other than price indices?

Any

expansion of the "duties" of Ethereum's consensus increases the costs, complexities and risks of running a validator. Validators become required to take on the human effort of paying attention and running and updating additional software to make sure that they are acting correctly according to whatever other protocols are being introduced. Other communities gain the ability to externalize their dispute resolution needs onto the Ethereum community. Validators and the Ethereum community as a whole become forced to make far more decisions, each of which has some risk of causing a community split. Even if there is no split, the desire to avoid such pressure creates additional incentives to externalize the decisions to centralized entities through stake-pooling.

The possibility of a split would also greatly strengthen perverse too-big-to-fail mechanics. There are so many layer-2 and application-layer projects on Ethereum that it would be impractical for Ethereum social consensus to be willing to fork to solve all

of their problems. Hence, larger projects would inevitably get a larger chance of getting a bailout than smaller ones. This would in turn lead to larger projects getting a moat: would you rather have your coins on Arbitrum or Optimism, where if something goes wrong Ethereum will fork to save the day, or on Taiko, where because it's smaller (and non-Western, hence less socially connected to core dev circles), an L1-backed rescue is much less likely?

But

bugs are a risk, and we need better oracles. So what should we do?

The best solutions to these problems are, in my view, case-by-case, because the various problems are inherently so different from each other. Some solutions include:

- Price oracles

: either not-quite-cryptoeconomic decentralized oracles, or validator-voting-based oracles that explicitly commit to their emergency recovery strategies being something other than appealing to L1 consensus

for recovery (or some combination of both). For example, a price oracle could count on a trust assumption that voting participants get corrupted slowly, and so users would have early warning of an attack and could exit any systems that depend on the oracle. Such an oracle could intentionally give its reward only after a long delay, so that if that instance of the protocol falls into disuse (eg. because the oracle fails and the community forks toward another version), the participants do not get the reward.

- More complex truth oracles

reporting on facts more subjective than price: some kind of decentralized court system built on anot-quite-cryptoeconomic DAO.

- Layer 2 protocols

: * In the short term, rely on partial training wheels (what this post callsstage 1)

- In the medium term, rely on multiple proving systems. Trusted hardware (eg. SGX) could be included here; I strongly anti-endorse SGX-like systems as a sole

guarantor of security, but as a member of a 2-of-3 system they could be valuable.

- In the longer term, hopefully complex functionalities such as "EVM validation" will themselves eventually be enshrined in the protocol

- In the short term, rely on partial training wheels (what this post callsstage 1)

- In the medium term, rely on multiple proving systems. Trusted hardware (eg. SGX) could be included here; I strongly

anti-endorse SGX-like systems as a sole

guarantor of security, but as a member of a 2-of-3 system they could be valuable.

- In the longer term, hopefully complex functionalities such as "EVM validation" will themselves eventually be enshrined in the protocol
- Cross-chain bridges

: similar logic as oracles, but also, try to minimize how much you rely on bridges at all: hold assets on the chain where they originate and use atomic swap protocols to move value between different chains.

- Using the Ethereum validator set to secure other chains

: one reason why the (safer) Dogecoin approach in thelist of examples above might be insufficient is that while it does protect against 51% finality-reversion

attacks, it does not protect against 51% censorship

attacks. However, if you are already relying on Ethereum validators, then one possible direction to take is to move away from trying to manage an independent chain entirely, and become a validium with proofs anchored into Ethereum. If a chain does this, its protection against finality-reversion attacks becomes as strong as Ethereum's, and it becomes secure against censorship up to 99% attacks (as opposed to 49%).

- In the short term, rely on partial training wheels (what this post callsstage 1)
- In the medium term, rely onmultiple proving systems. Trusted hardware (eg. SGX) could be included here; I strongly anti-endorse SGX-like systems as a sole

guarantor of security, but as a member of a 2-of-3 system they could be valuable.

- In the longer term, hopefully complex functionalities such as "EVM validation" will themselves eventually be enshrined in the protocol

Conclusions

Blockchain communities' social consensus is a fragile thing. It's necessary - because upgrades happen, bugs happen, and 51% attacks are always a possibility - but because it has such a high risk of causing chain splits, in mature communities it should be used sparingly. There is a natural urge to try to extend the blockchain's core with more and more functionality, because the blockchain's core has the largest economic weight and the largest community watching it, but each such extention makes the core itself more fragile.

We should be wary of application-layer projects taking actions that risk increasing the "scope" of blockchain consensus to anything other than verifying the core Ethereum protocol rules. It is natural for application-layer projects to attempt such a strategy, and indeed such ideas are often simply conceived without appreciation of the risks, but its result can easily become very misaligned with the goals of the community as a whole. Such a process has no limiting principle, and could easily lead to a blockchain community having more and more "mandates" over time, pushing it into an uncomfortable choice between a high yearly risk of splitting and some kind of de-facto formalized bureaucracy that has ultimate control of the chain.

We should instead preserve the chain's minimalism, support uses of re-staking that do not look like slippery slopes to extending the role of Ethereum consensus, and help developers find alternate strategies to achieve their security goals.