TLDR

: EigenLayer can be used to create a data availability layer that is superior to Danksharding. I argue that we should cancel EIP-4844 (proto-danksharding), instead implementing EIP-4488 (calldata gas reduction) and focusing resources into more crucial tasks.

What is EigenLayer?

EigenLayer is a programmable slashing protocol for Ethereum. In other words, it allows Ethereum validators to voluntarily submit themselves to extra slashing risks in order to provide services. Before if there was some feature that a developer wanted to add to Ethereum, there was only two options:

1. Try to get it enshrined in the protocol. This would result in your feature having the highest trust level (being secured by the entirety of the Ethereum network), but is extremely hard to accomplish (and for good reason, protocol changes should be difficult).

2. Implement it out-of-protocol as a layer 2. This is completely permissionless and fast (you only need to deploy smart contracts) but would require you to bootstrap a network from scratch (most likely creating your own token and trying to start a community).

So most features either end up either abandoned or with fragmented trust. EigenLayer provides a third option. Anyone is still able to add a new feature permissionlessly, but now Ethereum validators can be asked to “restake” their 32 ETH in order to secure that new feature. They’ll run the necessary middleware to provide that feature and if they act maliciously they’ll be slashed. There’s no cost of capital for the validators to provide these extra services, just the cost of running the middleware, and they can collect extra fees from doing so. If a feature is really popular, the entirety of the Ethereum validator set might opt in into it, thus giving that feature the same trust as if it was enshrined in the protocol.

Summarizing, EigenLayer allows permissionless feature addition to the Ethereum protocol.

Materials:

- Sreeram Kannan’s talk at ETHconomics

- Explainer Blog Post

Why use EigenLayer for Data Availability?

EigenLayer is working on a data availability layer for Ethereum, called EigenDA. It is supposed to be similar to the current Danksharding specs (with data availability sampling, proof of custody, etc). Except that it is an opt-in middleware instead of being a part of the core protocol. They have a testnet running now with 100 validators at 0.3 Mb/s each, which results in 15 MB/s total capacity (with a code rate of 1/2). Of course, the main problem with building a DA layer isn’t increasing the total capacity but rather the number of nodes. But I digress.

By itself, EigenDA doesn’t have any advantage over Danksharding, they do basically the same thing. But because it is built on top of the protocol and not as a part of it, it gains two very important properties:

1. Anyone can experiment with different DA layer designs and parameters.

2. Validators and users can opt into the DA layer that they prefer.

This means that we can let the free market converge on the best designs and that we can seamlessly update those designs in the future without needing to coordinate on a hard fork. New research will for sure appear on the data availability topic and the rollups needs will evolve over time (as rollups themselves will also evolve). By settling into a particular design for DA now, we are running the risk of getting stuck with a suboptimal design for many years.

If we have already accepted that the responsibility of scaling execution will be on the layer 2 protocols, it makes sense that we also delegate the responsibility of scaling data availability to the layer 2 protocols. Otherwise, we might be stifling the rate of innovation on the rollup space by forcing those same rollups to be constrained by an inflexible DA layer.

Another advantage of EigenLayer-based DA layers is that we can have many heterogeneous layers working at the same time. Different users (and apps and rollups) have different requirements for data availability, as can be gathered from all the talk about Validiums and alternative DA layers (like zkPorter, Celestia, etc). Polynya even wrote about this. By using EigenLayer, we can have DA layers with different security levels (by varying the number of validators or the erasure code rate), bandwidth capacities, latencies and prices. All of these secured by Ethereum validators with zero capital cost. Instead of letting another generation of “Ethereum-killers” appear (now for DA), we can let that innovation happen directly on top of Ethereum.

The final advantage that I want to mention is that an EigenDA could be done much faster than Danksharding and without requiring any resources from the Ethereum Foundation. This would free up the core developers and researchers to work on the much more pressing issue of censorship-resistance.

What could be done now?

The most obvious item would be to stop EIP-4844 inclusion in the Shanghai upgrade. It is a good EIP, I personally have been a vocal supporter of it, but EigenLayer based DA is just superior. The other items are more speculative and opinionated.

It is still probably a good idea to somehow increase the data capacity for rollups, the best candidate for this is EIP-4488 (which might need EIP-4444 to also be implemented). It is very easy to implement and rollups don't need to change anything in order to benefit from it. A recent post from Vitalik goes over why we might not want to do EIP-4488. Although, if we are to move sharding to L2, then points 2 and 3 no longer apply.

We might also want to protocolize EigenLayer in order to make it more functional. There's not a lot of research on this, but the post on PEPCs describes a possible way to do it.