Found the following take on eigenlayer a few times :

“Understood the high level of eigenlayer, and while it seems like a great idea theoretically, I’m not sure if teams would be willing to use it from an economic perspective, as it may hinder a team’s growth with a token”

Let’s now discuss potential business models for teams building on eigenlayer, and I’ll provide examples for that:

1. The first business model could be considered less crypto-native. In this model, you build your service on eigenlayer, drop the software stakers will have to run and take a cut of the fees generated by the service that go to a private wallet, while the rest of the fees go to the restakers. This approach could be suitable for web 2.0 companies that want to provide participants with digital property rights but prefer not to be decentralized themselves.

2. The second business model could be applied to a social network, for instance. Building a social network on Ethereum can be challenging due to the competition with DeFi transactions that have higher intrinsic values than social transactions. However, some stakers may want to help secure this social network because, on a 10-year horizon, the social network may succeed. With eigenlayer, you could build a data availability (DA) service dedicated to a social network, where restakers secure the DA layer and are compensated with your social network token.

3. The third business model could be applied to oracles for example. In a non-eigenlayer world, starting an oracle would involve launching a token and asking people to stake it to secure the service. However, the incentives may not be sufficient initially, resulting in stakers leaving and causing a security risk for the service, which is known as a “death spiral.” This is where eigenlayer solves the bootstrapping problem.

The power of the dual staking model comes into play here. A service can launch its own token, which adds an additional layer of security on top of the security inherited from ETH. This means you would have ETH restakers running the software of your service, as well as token service (e.g., $tokenservice) stakers running the software. A portion of the fees generated would be distributed to both sets of stakers. You could consider delaying the token launch until after the service has gained traction, so that the initial ETH restakers are rewarded with maybe an airdrop, acknowledging their early participation and ownership in the network. The service token could also be used for governance upgrades.

One could envision increasing the rewards and the weight of the service token once it has gained significant value, while still maintaining a minimum level of security with ETH restakers as a backstop in case the service token’s value decreases. This way, you don’t have to worry about achieving the same level of economic security as Chainlink, which has hundreds of millions staked, as even with just 1% of ETH stakers, you could have a substantial amount at stake (around 200M) to launch your service. This approach ensures the security of your service, attracts more activity and fees, and allows for the development of utility for your token.

As always, hope that makes sense,

Cheers !