EIGEN Token Overview

What is the EIGEN token?

The EIGEN token is a universal intersubjective work token. We will break down the terms here. First, by work token, we mean that a token that can be staked to perform some work within a blockchain platform. This work could be an execution task or validation task. Existing work tokens often have the following limitations:

- Special-purpose:
- Existing work tokens are designed to provide cryptoeconomic security to only one enshrined specific digital task. For example, before ETH restaking in EigenLayer, ETH was specialized for securing only Ethereum's consensus.
- Objective:
- ETH restaking expanded the scope of ETH. With ETH restaking, the ETH staked can now be used to secure those services that have objectively attributable faults and can be proven onchain by means of an optimistic dispute resolution mechanism. Resolution of faults is done by means of mathematics and cryptography.
- However, there is a much wider class of digital tasks where the faults in their execution are not provable onchain in a smart contract but are observable by any observer outside the chain and a wide agreement can be achieved among the honest observers. We call this category of faultsintersubjectively attributable faults
- . By having staking with EIGEN, the goal is to secure AVSs that haveany
- intersubjectively attributable faults and penalize the EIGEN stake of the operators who participate in these faults. This lends a universal nature to the EIGEN token.

What are intersubjectively attributable faults?

The core property of intersubjectively attributable faults is that in the event of such a fault, there will be wide agreement among observers outside the chain about the occurrence of the fault. Unlike objectively attributable faults, intersubjectively attributable faults might not be provable mathematically and cryptographically onchain. Examples of intersubjectively attributable faults are:

- Is data available in the Data Availability layer at a certain point in time? This can be observed externally by means of data availability sampling, but any unavailability or otherwise can't be proven in a smart contract.
- Is an AI inference result accurate within a margin of error? Given the description of hardware spec, random seeds used and input data, it can verify whether the inference is within a confidence interval or not.

An important note is that any objectively attributable fault is also an intersubjectively attributable fault. For example, any fault in executing a deterministic VM such as EVM is an objectively attributable fault that all honest observers outside the chain will agree with.

Intersubjectively attributable faults are different from subjective faults . The set of intersubjectively attributable faults doesn't include subjectivity in the observer's response. In digital tasks with subjective faults, it implies that no wide agreement can be achieved among the observers about the occurrence of fault. Examples of subjective faults are: Is Paris the most beautiful city? What will be the price of a particular NFT in 1 year? And so on.

What are the core ideas behind how staking with EIGEN enables resolution of intersubjectively attributable faults?

Resolving intersubjectively attributable fault using EIGEN staking utilizes three core ideas: (1) setup and execution phase, (2) slashing and (3) token forking.

The first core idea is the concept of setup phase and execution phase that is applicable to any coordination system. The setup phase represents the process of discussing, agreeing on and codifying (a) the set of execution rules that will be used for execution of any digital task in the system and, (b) the set of monitoring and verification rules among the observers about any fault while executing. Once the setup phase is over, it is the execution phase where these pre-agreed rules are executed. These rules should enable detection of faults to be self-evident and self-verifiable by any user in the social consensus such that it precludes meeting in-person or in a discord channel. An example of the setup phase is the ratification of the US constitution and now in the execution phase any law passed has to be compliant with the constitution. In the context of blockchains, an example would be the agreement to use the longest-chain consensus rule which is then used for determining the latest block in bitcoin.

The second core idea is slashing. Slashing allows any adversarial participant in a proof-of-stake network to be punished for their misbehavior during the execution phase by taking away their stake. This lends to cryptoeconomic benefits to any PoS system in the form of karma. You can read more about the benefits of slashing inthis blog post .

The third core idea is token forking. Most of the current systems attempt to resolve intersubjectively attributable faults either by slashing the stake of operators whose response to the task diverged from the majority response, or by using a committee to slash the operators whose claims are not concordant with the "true" answer. However, both mechanisms are vulnerable to the tyranny-of-majority. A third way is forking of the chain state itself, which is used in the context of resolving intersubjective fault of lack of chain growth in Ethereum. This involves empowering social consensus of the chain to slash the adversarial validators in the event of majority corruption of the validator set. Token forking uses this fact to fork just the token, without forking the chain state, to induce cryptoeconomic penalties on malicious stakers. With EIGEN staking, if the majority of EIGEN stakers were to turn malicious and cause an intersubjectively attributable fault, then a new fork of EIGEN can be started by anyone where the malicious stakers can be penalized by restricting them from being able to redeem the tokens from the new fork. As the new fork comes to be considered as canonical by the social consensus, the malicious stakers end up being penalized.

What are the key features of the EIGEN staking?

There are four key features in EIGEN staking: (1) universality, (2) isolation, (3) metering, and (4) compensation.

- Universality
- . The setup phase of EIGEN stipulates that EIGEN can be universally used for resolving any intersubjectively attributable fault, not just one specific fault. Any AVS that wants to use the cryptoeconomic benefits of EIGEN must encode their rules of coordination that they have agreed on in their respective setup phase. These AVS-specific rules, such as slashing conditions, are akin to amendments to EIGEN. Furthermore, these slashing conditions of the AVSs must ensure that any intersubjectively attributable fault is self-verifiable beyond a reasonable doubt.
- Isolation
- . The second core feature of EIGEN is isolation. To appreciate this feature, consider an alternate design where any fork in the token would require DeFi markets to be aware of the fork, and the token is no longer usable for going into long-term DeFi positions. To avoid this undesirable externality on DeFi, EIGEN has a two-token model. The first token, bEIGEN, is used for staking and can be subjected to forking. As for the other token, EIGEN, our design offers the benefit of remaining unaware of forks in bEIGEN to any holder of EIGEN who is using it for DeFi or any non-staking applications.
- Metering
- . Resolving any intersubjective fault incurs a cost to social consensus for switching from one token to another, or rejecting a malicious fork. Therefore, any claim to fork the token should require depositing a bond in bEIGEN to deter malicious challenges. This bond needs to be higher than the cost incurred by the social consensus (users and AVSs) to consider and reject a malicious fork. Additionally, any successful challenge results in a significant cost to the broader ecosystem in the form of contract upgrades for incorporating the new fork in daily operations. A challenge, therefore, should only be raised if a sufficient amount of staked EIGEN can be considered malicious and burnt, resulting in a lower token supply for the remaining EIGEN.
- Compensation
- . The protocol for intersubjective staking ensures that if an AVS is attacked due to a malicious quorum of EIGEN stakers, that AVS is able to slash and redistribute the malicious stake back to the AVS users. If the AVS ensures that this “attributable security” is greater than the harm done to its users, then it achieves “strong cryptoeconomic security,” which specifies that no honest user suffers any harm. Strong cryptoeconomic security is a user-centric characterization and does not need to make any assumptions on the adversary or even the other users’ incentive structure. For example, when there are multiple AVSs, as long as a given AVS ensures that it has enough attributable security, this particular AVS is protected even if the system is, in sum total, not secure because other AVSs do not have enough security.
- Furthermore, using EIGEN for resolving intersubjectively attributable fault ensures that Ethereum’s social consensus is not overloaded.

How does EIGEN staking complement ETH restaking?

EIGEN staking and ETH restaking play complementary roles within EigenLayer. In many mature AVS protocols, safety properties are secured through objective slashing, while liveness and censorship-resistance, which previously relied on majority-assumptions and stake decentralization, are achieved through EIGEN staking. For a service that uses restaking for safety and EIGEN staking for liveness, fees can be split between the two quorums. Furthermore, for core services provided to the Ethereum ecosystem, we envision many services that will use dual staking between ETH and EIGEN. The ETH restaking absorbs the decentralization/collusion resistance and operator alignment that comes with it, and the EIGEN staking can support cryptoeconomic slashing. In this model, the former serves as a mechanism to obtain majority trust from the Ethereum participants, and the latter serves as a mechanism to obtain economic security.

In keeping with the spirit of open innovation, EigenLayer allows AVSs to mix and match these two modalities of objective staking from ETH and intersubjective staking from EIGEN, in addition to utilizing the native AVS tokens for providing additional validation from an aligned community of AVS token stakers.

How does EIGEN staking accelerate innovation in AVSs with objectively attributable faults?

EIGEN staking can support digital tasks that could potentially be secured via objective fraud proofs, but doing so would involve significant technical complexity and associated risk. Specifically, when envisioning the lifecycle of a new objective AVS, we can identify a progression that leverages EIGEN staking for security early in the protocol's bootstrapping phase, transitioning to restaking or even native protocol adoption as the protocol matures, ossifies, and more faults become objective.

What are some examples of Actively Validated Services (AVS) that could be secured using the

EIGEN token?

- Some of the foundational primitives that can now be secured using EIGEN staking are:
- Censorship-resistance: Ensuring that propagated transactions are eventually included in the - ledger.
- Ledger growth: Ensuring that new transactions keep getting added to the ledger.
- Data Availability: Guaranteeing the availability and accessibility of data across the network.
- Oracles: Providing reliable and verifiable real-world data to blockchain applications.

With these foundational modules, one can build a plethora of new services:

- New Chains: Building new blockchain networks with customizable security and features for different modules.
- Intents and MEV (Miner Extractable Value): Managing transaction ordering and preventing malicious activities related to MEV.
- AI Training, Benchmarking, and Inference: Ensuring the validity and security of AI models and their execution.
- Prediction Markets: Creating decentralized prediction markets.
- Storage Services: Building secure and decentralized storage solutions.
- Cloud Microservices: Migrating traditional cloud services like Kafka to blockchain with cryptoeconomic security and slashability.
- Gaming Virtual Machines: Securing the execution of gaming environments on the blockchain.
- Databases: Such execution environments have complex dispute resolution which can be substituted with the intermediate step of being secured by EIGEN staking.

EIGEN Token Restaking FAQ

Please see theEIGEN Token Restaking FAQ page for more information.

Previous Key Terms Next Community and EIGEN Claim Support