0. Background

Currently, there is no unified term for the middleware protocols (The Graph, Gelato Network, Chainlink) used in DApp development, and most people usually categorize them as “Infra.” This term is vague and causes confusion as the underlying layer networks like Ethereum are also Infra.

Therefore, we at Hyper Oracle define the middleware involved in DApp development as “Oracle”. The term “Oracle” has been used to describe certain parts of the blockchain: We expand on its notion and propose the concept and design of a ZK-based Oracle for Ethereum.

More details about zkOracle and its components can be found in our whitepaper:Hyper Oracle: A Programmable zkOracle Network.

1. Framing of Oracles

When people hear the term “oracle,” they often associate it with the price feed oracle, which provides off-chain data to on-chain smart contracts. However, this is just one type of oracle among many.

A straightforward explanation of the Oracle concept, as outlined in thiseducational resource, divides it into two main types:

- Input Oracle: delivers off-chain data to on-chain context (ex: Chainlink Price Feeds).
- Output Oracle: delivers on-chain data to off-chain context for advanced computation (ex: The Graph Protocol, Hyper Oracle zkIndexing).

[

截屏2023-02-21 下午3.04.20

2560×1440 149 KB

](https://ethresear.ch/uploads/default/original/2X/9/92d36d21b7aae5c40e3f4eaacc397025327fad43.png)

In the realm of blockchain, the terminology “input” and “output” are used to distinguish between two types of oracles: input oracles and output oracles. In addition, Hyper Oracle is defining the I/O oracle, a specialized type of oracle that integrates both input and output oracles by first following the output oracle’s flow and then the input oracle’s. Each oracle can be further broken down into three components: data source, computation, and output.

- Input Oracle
- Data Source: Off-chain data (e.g. CEX price feeds, real-world weather data)
- Computation: Aggregation of off-chain data and “uploading” of data
- Output: On-chain data (equivalent to off-chain data, but stored on-chain)
- Data Source: Off-chain data (e.g. CEX price feeds, real-world weather data)
- Computation: Aggregation of off-chain data and “uploading” of data
- Output: On-chain data (equivalent to off-chain data, but stored on-chain)
- Output Oracle
- Data Source: On-chain data (e.g. smart contract interactions or events like ERC-20 transfers or ERC-721 minting)
- Computation: Indexing, aggregation, filtering, or other complex computation
- Output: Off-chain data in an organized and easy-to-use form
- Data Source: On-chain data (e.g. smart contract interactions or events like ERC-20 transfers or ERC-721 minting)
- Computation: Indexing, aggregation, filtering, or other complex computation
- Output: Off-chain data in an organized and easy-to-use form
- I/O Oracle
- Combines Input Oracle and Output Oracle with Output flow first, then Input flow.
- Combines Input Oracle and Output Oracle with Output flow first, then Input flow.

[

截屏2023-03-14 07.18.42

2560×1440 227 KB

](https://ethresear.ch/uploads/default/original/2X/8/88ee180f9e736f929505eefd069bf8696f5804ec.jpeg)

2. zkOracle

A zkOracle has advantages over a traditional oracle:

- Providing a unstoppable autonomous network
- Math as the consensus
- Safeguarding the security of the base layer
- A 1-of-N trust model
- Optimal cryptography-native decentralization
- Efficient computing power allocation (ideally no excess wasted)

As a component that processes data, an oracle must ensure both the accuracy and security of computation. It is important to confirm that the output is valid and correct and that the verification process is fast.

[

截屏2023-03-14 07.31.08

2560×1440 113 KB

](https://ethresear.ch/uploads/default/original/2X/1/1c048aa885e153d2f016abc1086bd539aff12d40.png)

To achieve a trustless and secure oracle, we need to make it a zkOracle.

Hyper Oracle zkOracle is natively categorized as output zkOracle and I/O zkOracle.

I. Output zkOracle

An output zkOracle is an output oracle that uses zk to prove its computation's validity. An example of this is Hyper Oracle zkIndexing Meta App.

[

截屏2023-03-14 07.32.25

2560×1440 137 KB

](https://ethresear.ch/uploads/default/original/2X/6/60551b3eb89cbd4779f1797bbdc95feaa925d7bd.png)

- Data Source: On-chain Data

The straightforward solution is to use on-chain data as the source. This data has already been verified and secured by the blockchain. Off-chain data sources cannot efficiently reach the trust level of on-chain data (at least not yet, according to this source). The on-chain data source solution requires zkOracle to act as an output oracle.

- Computation: Execution and ZK Proof Generation

The solution is to create a zk proof of the computation (typically indexing, aggregation, and filtering…) and enable the step of accessing the data source in a zero knowledge fashion. This adds a layer of validity and trustlessness to the computation. The output will now be accompanied by a zk proof, making the computation and output verifiable.

- Output: Execution Output and On-chain Verifiable zk Proof

The output of the computation will be both the execution output and a verifiable zk proof. The proof can be easily verified in a smart contract or any other environment. The verification component can confirm the validity of the execution of the zkOracle.

II. I/O zkOracle (Output + Input)

An I/O zkOracle is an output oracle and an input oracle both with ZK as computation. An example is Hyper Oracle

zkAutomation Meta App.

[

截屏2023-03-14 20.29.17

2560×1440 165 KB

](https://ethresear.ch/uploads/default/original/2X/0/085518c92189247a046941b15bfb80688519ea6f.png)

In this case, a zkOracle will function as a combination of two oracles that operates in two stages:

- Data Source: On-chain Data

The data source for I/O zkOracle is identical to the output zkOracle.

- Computation: Execution and ZK Proof Generation

The computation of I/O zkOracle includes the output zkOracle (which involves indexing, aggregation, and filtering) as well as the input zkOracle (which involves setting up off-chain computation results as calldata for smart contract calls). The combination of both parts makes it feasible to automate smart contracts with complex off-chain computation.

- Output: On-chain data and On-chain Verifiable zk Proof

The output for this stage includes on-chain data which is the execution output provided on-chain as calldata, and a verifiable zk proof. This proof is easily verifiable in smart contracts or any other environment. The verification component can confirm the validity of the execution of I/O zkOracle.

III. Definitions

Technically, zkOracle is an oracle with verifiable pre-commit computation.

Functionally, zkOracle utilizes zk to ensure the computation integrity of the oracle node for the oracle network's security, instead of staking and slashing mechanism.

In essence, zkOracle is an oracle that utilizes zk for computation and data access, while also using on-chain data for the data source to secure the oracle in a trustless manner.

IV. Comparisons

The advantages of the zkOracle network compared to traditional networks are similar to those of the zk rollup network compared to traditional distributed networks.

1. Security

The trust model of the zkOracle network is 1 of N, meaning the system remains functional as long as at least one node behaves as expected. Securing the network only requires one honest zkOracle node. In contrast, traditional oracle networks typically operate under a trust model of N/2 of large N, or 1 of 1.

[

Vitalik's definition on Trust Models (https://vitalik.ca/general/2020/08/20/trust.html)

640×567 27.9 KB

](https://ethresear.ch/uploads/default/original/2X/f/fb55877a18ea5156715a0dd88ac1e5862c316e09.png)

Image Source: Vitalik's definition on Trust Models (Trust Models)

It's important to note that traditional oracle networks cannot be fully trusted when there's only one node (either a data provider or an oracle node). This has significant implications for the following points.

1. Decentralization

The traditional oracle network may be difficult for entry due to its high staking requirement, but the zkOracle network will be more accommodating to nodes as it only requires hardware that can be further optimized through innovative proof systems and other cryptographic designs related to zk technology.

1. Performance

Performance is a crucial factor when it comes to oracle services, especially those that involve output oracles such as indexing protocols. The latency of request and response is highly dependent on the geographical distance between the node and the requester. Although requesters can rely on the results from the entire traditional oracle network, they cannot

rely on a single node (that serves fastest), which can have an impact on performance. In contrast, a zkOracle node that is geographically closest and fastest can be trusted to provide better performance due to its computation verifiability.

3. zkOracle Network for Ethereum

zkOracle = zkPoS + zkGraph run in zkWASM

Hyper Oracle is designing a zkOracle network operates solely for the Ethereum blockchain. It retrieves the data from every block of the blockchain as a data source with zkPoS and processes the data using programmable zkGraphs that run on zkWASM, all in a trustless and secure manner.

Here is the zkOracle design for the Ethereum blockchain. This serves as a foundational design for a zkOracle, complete with all of the essential components.

[

截屏2023-03-13 20.15.58

2560×1440 146 KB

](https://ethresear.ch/uploads/default/original/2X/1/1f88b8cfa14430bbe06564a6537e03e4ac4712f9.png)

zkPoS verifies Ethereum consensus with a single zk proof that can be accessed from anywhere. This allows zkOracle to obtain a valid block header as a data source for further processing.

zkWASM (zkVM in the graph) is the runtime of zkGraph, providing the power of zk to any zkGraph in the Hyper Oracle Network. It is similar to the kind of zkEVM used in ZK Rollups.

zkGraph (run in zkWASM) defines customizable and programmable off-chain computation of zkOracle node’s behaviors and Meta Apps. It can be thought of as the smart contract of the Hyper Oracle Network.