Decentralized Oracles

February 2021

Abstract

Decentralized Oracle (DEOR) is a fully decentralized network, unlike existing oracles; it is not a centralized or semi-centralized network, and therefore is more secure. This paper details the connection of on-chain components for smart contracts and the (randomly and appropriately) selected off-chain world. We also describe about reputation scoring algorithm which incorporates randomness to improve security and integrity of the oracle output and the security monitoring system for DEOR which help people make informed data provider selections.

Contents

1 Introduction

Smart contracts provide an infrastructure for creating Dapps (decentralized applications) and DAOs (Decentralized Autonomous Organizations) in order to automate jobs on top of the blockchain.Such jobs are triggered algorithmically and cannot be modified after deployment. However, smart contracts cannot access external data. Today, the solution to this problem is called an oracle. An oracle connects the off-chain world with smart contracts. This paper details the design of DEOR, the problems with current oracles and how DEOR improves on these solutions. Then, the details of how on-chain components for smart contracts connecting with the random selectable off-chain world are described. We also describe about reputation and security monitoring system for DEOR which help people make informed provider selections. Finally, at the end of the paper, Roadmap of the project is declared which is consisted of adding various data sources, responses with various kind of proofs and developing decentralized oracle services.

Compared to traditional contracts, Smart contracts are more secure, as anyone (including the author) has the same authority Smart contracts are automatically executed when they meet the requirements; all parties of the contracts can reach an agreement without trust.This feature turns smart contracts into a superior tool for realizing and implementing digital agreements. It should be noted that smart contracts are trying to digitalize real-world agreements. Consequently, in order to carry out such a task, they need to access real-world data. However, due to the specific nature of the underlying consensus protocols, blockchains are unable to connect external data sources, therefore they cannot access data from outside the blockchain. Thus, smart contract developers encounter a connectivity issue according to which majority of smart contracts are not able to function practically. Decentralization reduces the needs of trust among parties of the contracts, and DEOR ensures the security of the entire procedure of smart contracts execution, including obtaining data from off-chain sources. This is the prerequisite of connecting smart contracts to the real world and taking the place of traditional digital contracts. DEOR can output data securely to off-chain systems, which implements the connection to the real world and ensure the temper-proof of smart contracts.

In this paper, Section 2 provides the technical overview how DEOR links the on-chain and off-chain worlds in terms of both off-chain and on-chain components. Section 3 describes how DEOR is secured, with Section 4 concluding discussion.

2 Technical Overview

DEOR aims to link the on-chain and off-chain worlds in a secure manner. We describe the architecture and technical innovations of each DEOR component below. A pictographic overview can be found in Fig. 2

2.1 On-chain

- **Custom Oracle Selection with Randomizer** Oracle services purchasers evaluate their specific requirements, then select nodes and services who can fulfil their requirements (Node related data is available in the list for consumers to choose appropriate nodes and services); however manual matchin is not possible in all cases. In this case, Oracle selects the voting node with a randomizer. 70 pct of oracles are randomly selected per transactions; not all oracles participate into voting, so DEOR can respond effectively to Sybil attacks. In a Sybil attack, the attacker subverts the reputation system of a network service by creating a large number of pseudonymous identities and uses them to gain a disproportionately large influence. A reputation system's vulnerability to a Sybil attack depends on how cheaply identities can be generated, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically. An entity on a peer-to-peer network is a piece of software that has access to local resources. An entity advertises itself on the peer-to-peer network by presenting an identity. More than one identity can correspond to a single entity. In other words, the mapping of identities to entities is many to one. Entities in peer-to-peer networks use multiple identities for purposes of redundancy, resource sharing, reliability, and integrity. In peer-to-peer networks, the identity is used as an abstraction so that a remote entity can be aware of identities without necessarily knowing the correspondence of identities to local entities. By default, each distinct identity is usually assumed to correspond to a distinct local entity. In reality, many identities may correspond to the same local entity. An adversary may present multiple identities to a peer-to-peer network in order to appear and function as multiple distinct nodes. The adversary may thus be able to acquire a disproportionate level of control over the network, such as by affecting voting outcomes.In this paper, DEOR is implemented with Randomizer which has a great robustness from the sybil attacks.

- **Data Reporting** Once the new oracle record has been created, the off-chain oracles execute the agreement and report back to on-chain.

- **Data Aggregation** Once the oracles have revealed their results to the oracle contract, their results will be fed to the aggregating contract. There will be no universal aggregator contract, for every demand can be different. DEOR will include a standard (i.e. a template) for users to customize their contracts. There two types of data aggregation in current DEOR: price feed and data query.

2.2 Off-chain

The off-chain component of DEOR network is the oracle node. It is connected to the Ethereum network, and it will support all leading smart contract networks. The DEOR oracle nodes are powered by the standard open-source core implementation, which handles standard blockchain interactions, scheduling, and connecting with common external resources.

- **DEOR Core** The core of DEOR oracle off-chain node is responsible for interacting with the blockchain, assignment scheduling and balancing work across its various external services. Each assignment is a set of smaller job specifications, known as subtasks, which are processed as a pipeline. Each subtask passes its result to the next subtask; they run in tandem to get the final result.
- **External Adapters** Users can customize subtasks within an external adapter. Beyond the built-in subtask types, custom subtasks can be defined by creating adapters. Adapters are external services with a minimal REST API.
- **Subtask Schemas** With the applications of DEOR becoming wider, there can be more open-source adapters; thus they will be available for public review. DEOR currently operates with a schema system based on JSON, to specify what inputs each adapter needs and how they should be formatted.

3 Oracle Security

In order to explain DEOR's security architecture, we must first explain why security is important and what it means.If a smart contract security gets a false data feed, it will likely behave payout in an undesired manner, for instance, paying the incorrect party or not on agreed terms, insurance fraud (if smart contract insurance data feeds can be tampered with by the insured party), etc. More generally, a well-functioning blockchain, with its ledger or bulletin-board abstraction, offers very strong security properties. An oracle too must therefore serve users as an effective trusted third party, providing correct and timely responses with very high probability. There are four ways by which DEOR provides security.

3.1 Reputation System

The reputation system proposed for DEOR would record and publish user ratings of oracle providers and nodes, offering a means for users to evaluate oracle performance holistically. Reputation metrics should be easily accessible off-chain where larger amounts of data can be efficiently processed and more flexibly weighted. For a given oracle operator, the reputation system is initially proposed as supporting the following metrics, both at the granularity of specific assignment types, and also in general for all types supported by a node:

- Total number of assigned requests: The total number of past requests that an oracle has agreed to, both fulfilled and unfulfilled.
- Total number of completed requests: The total number of past requests that an oracle has fulfilled. This can be averaged over number of requests assigned to calculate completion rate.
- Total number of accepted requests: The total number of requests that have been deemed acceptable by calculating contracts when compared with peer responses. This can be averaged over total assigned or total completed requests to get insight into accuracy rates.

Reputation score doesn't only affect the voting, but also affects the award for true voting oracles. The weighting system uses the idea of levels as can be seen in Table 1.

In this, oracle's level can be easily upgraded from Level 5 to Level 4. But it is very difficult to be upgraded from Level 2 to Level 1. We define 5 Levels for use levels and set 10.0 as the max reputation score

Level	Reputation Score
5	1.0$\leq x < 5.09$
4	5.09$\leq x < 7.7$
3	7.7$\leq x < 9.18$
2	9.8$\leq x < 9.83$
1	9.83$\leq x \leq 10.0$

Table 1: Reputation Table

according to equation:

$$\beta \sum_{i=1}^{i \leq MaxLevel} (\frac{i}{2})^2 = Score_max - Score_{min}, Score_{max} = 10.0, Score_{min} = 1.0 \tag{1}$$

Reputation score line is divided into 5 levels according to .

$$Low_level = Score_{max} - \beta \sum_{i=1}^{i \leq MaxLevel} (\frac{i}{2})^2, \sigma = 0.01 \tag{2}$$

The additional score point σ is the point which is added to the score when the oracle did the true voting, σ increases as the score increases.When the oracle did the false voting, its reputation score is set to the low score point of under level. An example of reweighting is shown in Fig. 3.

3.2 Secure Data Aggregation

We can obtain data from several different data sources to mitigate the impact of an abnormal data source. An aggregating function can aggregate the results into a single output. There can be many ways to do aggregation, such as calculating the weighted average after removing abnormal data.

3.3 Secure Oracle Node

Several nodes form the oracle network, and each node has its data source set, which may overlap with the others'. An oracle aggregates data from its data sources and sends the aggregated result to the request. A request may choose several nodes to ensure accuracy. As faulty nodes may exist, there should be a plan to mitigate the influence. In our case, voting oracles are selected by randomizer and their votes have its own weight-reputation score. If the sum of upvoting weights is over 70pct of the total weight, it is verified as true and the answer is verified data by oracles.There is a security monitoring system for oracle nodes, which calculates average response time and last active time. And also each oracles must have to pay penalty payments. If penalty payments were locked in to assure a node operator's performance, the result would be a financial metric of an oracle provider's commitment not to engage in an "exit scam" attack, where the provider takes users' money and doesn't provide services. This metric would involve both a temporal and a financial dimension.

3.4 Contract-upgrade service

As recent smart contract hacks have shown, coding bulletproof smart contracts is an extremely challenging exercise. No one can control the actions of smart contracts once deployed; this makes the security of the oracle important. A decentralized exchange can suffer a massive loss if it receives incorrect data from an oracle.DEOR proposes a contract-upgrade service for security reasons. The service will be run by the organizations who launch DEOR nodes and follow DEOR's philosophy of decentralized design. Contract-upgrade service is non-mandatory; users decide whether to turn this on according to their demand. Migration of users to new oracle contracts functions as a kind of "escape hatch," something long advocated for by blockchain researchers as a mechanism to fix bugs and remediate hacks without resorting to such cumbersome approaches as 20 white hat hacking or hard forks.

4 Conclusion

We introduce the fully decentralized oracle network DEOR, describe its on and off-chain components. We interpret our schemes of security and decentralization and describing them in the context of existing design flaws of Oracles and give plans for future developments.

Tokenomics

- DEOR Tokens DEOR oracle has issued its dedicated tokens dubbed DEOR in the Ethereum blockchain.The purpose of this token is creating a more affordable payment method for clients to pay for DEOR oracle services with a significant discount in comparison to pay with ERC20 tokens.
- Token Distribution Policy The basic principle which has been observed in distribution policy is implementing maximum dispersion and avoiding centralization.
- In this procedure 20 pct of tokens will be distributed in a fair distribution event that lasts for 7 days. 20 pct of the tokens are for the continued development and updating of DEOR. The 40 pct will be used for vote mining and propagation of network effects from various Oracles and aggregators. The final 20 pct is for incentivization and fair distribution by traditional pool methodology.

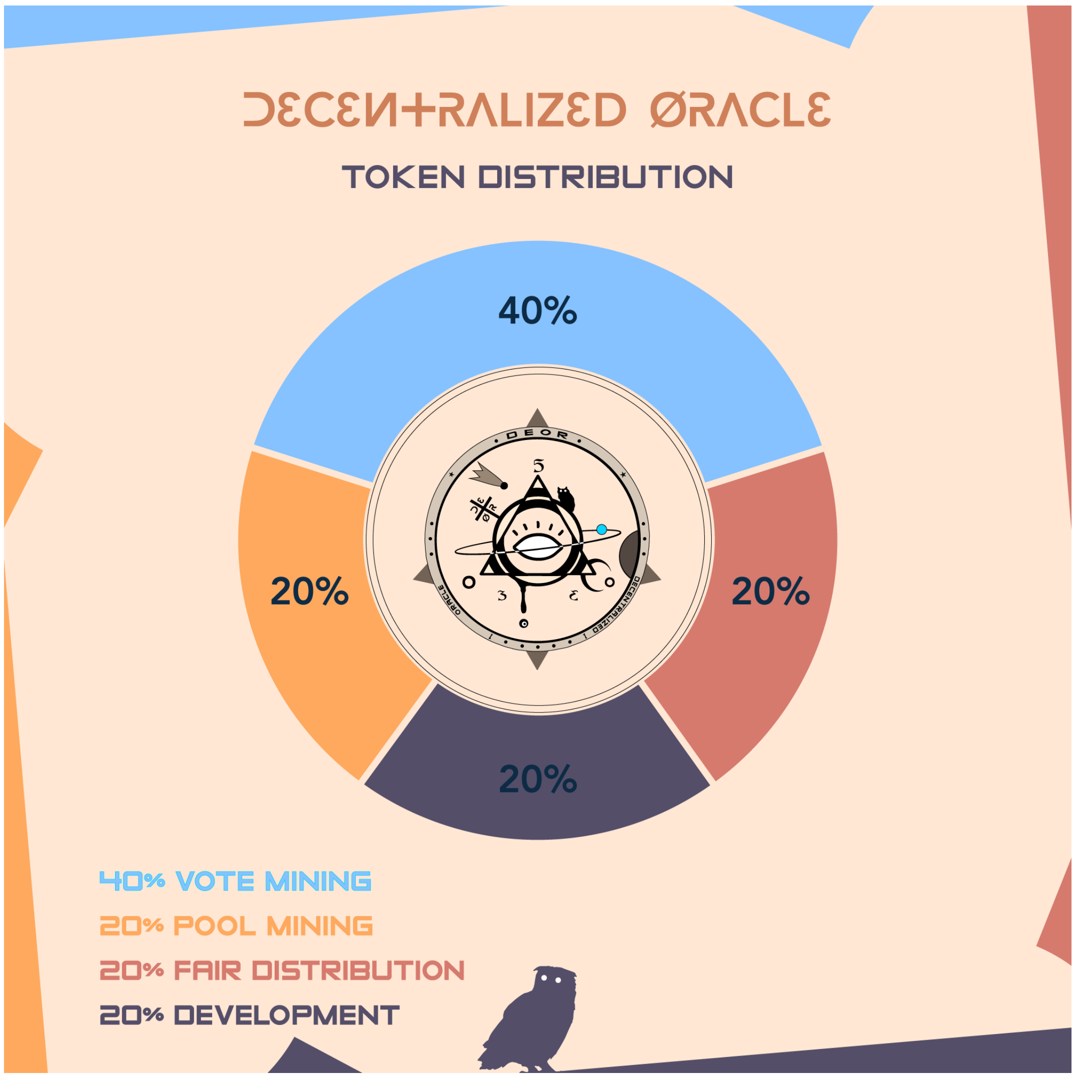

Figure 1: Tokenomics

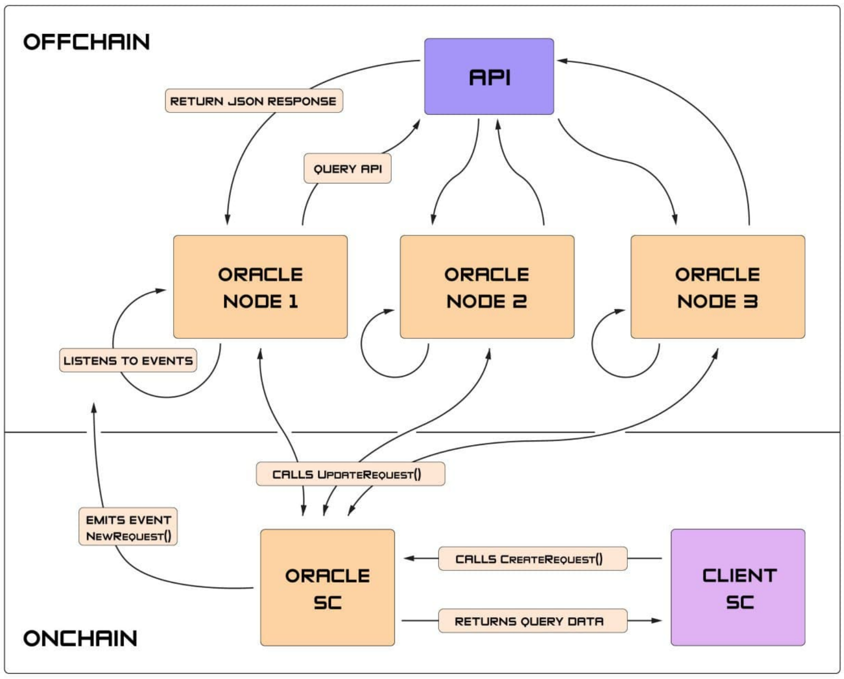

Figure 2: Overview

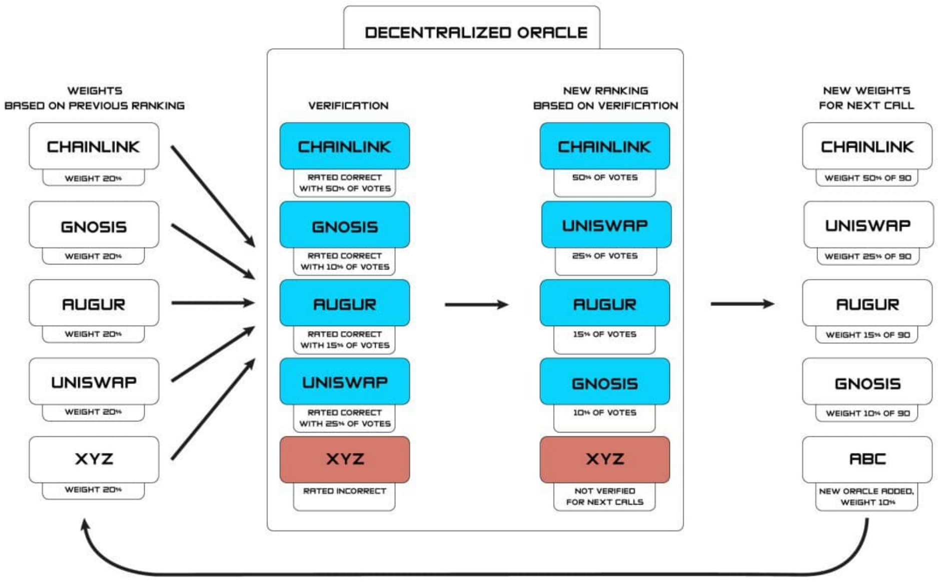

Figure 3: