# Themis—A Distributed Oracle Solution

**Introduction:**

Oracle is the system which provides out-of-chain data source for in-chain smart contracts. The word of “oracle” comes from Greek mythology, representing the people who can communicate with gods, and see the future vision. In the context of blockchain, oracle is the system which can answer the external problems of blockchain and the bridge connecting in-chain and out-of-chain data. In ideal conditions, oracle is a free-of-trust system, which means that they do not need to be trusted as they operate according to the principle of decentration. Themis also comes from Greek mythology, who is the goodness of justice. We hope to provide fair, just, accurate and timely data service for in-chain smart contracts through Themis solution.

## 1. Why do we need oracle?

Generally, oracle refers to the mechanism writing in the information out of the blockchain into the blockchain. We can understand it as the bridge connecting the real world and the blockchain world, which writes the external information into the blockchain and complete the data connectivity between the blockchain and the real world. It allows determinate smart contracts to react to the indeterminate external world. It is the only approach of smart contracts in data interaction with outside and the interface of data interaction between the blockchain and the real world.

Why do we need oracle?

In the blockchain, only in-chain data can be acquired, but the out-of-chain data of the real world cannot be accessed. It means that the blockchain world is isolated from the outside. Circulation can be conducted inside the blockchain, but it cannot get in touch with the outside. When certain functions or smart contracts need to read external information, no signal can be received. In other words, smart contracts cannot initiatively acquire out-of-chain data, but can only passively accept data, which is the major reason for the birth of oracle.

If certain result will be triggered after meeting certain condition, it is the execution process of smart contracts. However, smart contract cannot read out-of-chain data by itself. There must be an external data source telling it what happens so that it can execute corresponding contents. For example, if we need to make some computation based on the total global population M in the smart contract,

such M is an uncertain information outside the system. It is possible that the results with such data M got from outside at different nodes are different, and even the results got from outside at the same node but different time are different. Thus, the correctness cannot be proved mutually between nodes.

As more nodes join the network, new nodes need to replay all the transactions on the blockchain before. The total global population M got at this time may also be completely different. New nodes cannot confirm whether the data in the chain before is correct. The consensus mechanism of such blockchain will collapse. Thus, the blockchain cannot open the network port which initiatively and synchronously acquire external data.

The function of oracle is the agent in the chain which provides the data of the real world. The most important character is that it needs to guarantee that such agent of oracle does nothing bad itself or does not tamper the data. Smart contract is only the procedure which meets the reaction state when corresponding conditions are met. In other words, the reaction and operation of smart contracts cannot do without data source, and the next step of operation can only be proceeded when smart contracts accept these determinate data. Without oracle, the blockchain will be isolated from the external world. No application scene requiring interaction with the external world will be realized, which will greatly limit the development of blockchain ecology.

## 2. Main application scenes of oracle at present

In ideal conditions, oracle provides a free-of-trust or at least almost free-of-trust way to acquire the information in the real world, such as the result of sports events, the price of gold or the authentic random number, for smart contract on the platform of Ethereum (ETH). They can also be used to directly and safely relay the data to DApp front-end. Thus, oracle can be regarded as the mechanism closing up the gap between the out-of-chain world and smart contracts. It allows smart contracts to enforce contractual relationship based on the incidents and data of the real world, so as to greatly expand their scope.

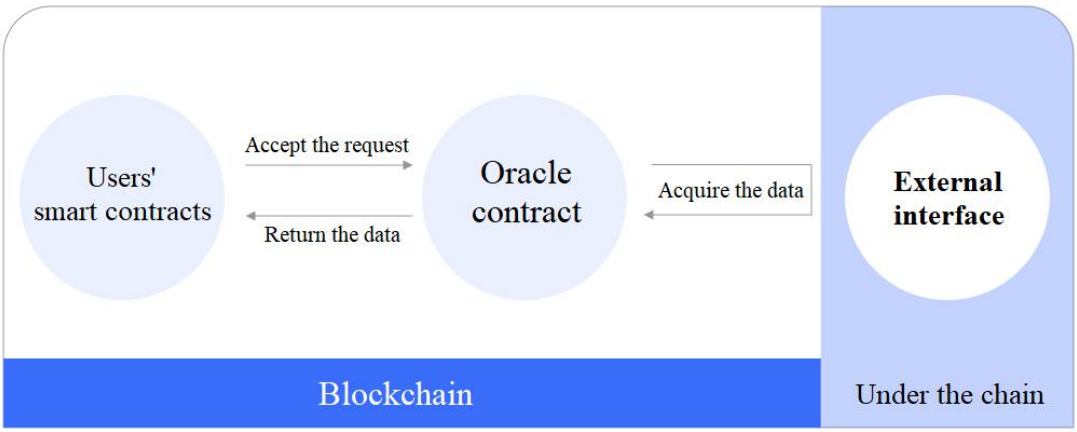


Figure 1. Oracle—Connecting Blockchain and Out-of-chain Data

More data examples which may be provided by oracle include:

Physical random number source or entropy source (such as quantum phenomenon or thermal phenomenon): e.g. selecting the winner fairly in smart contracts of lottery.

Exchange rate data: e.g. precisely linking encrypted currency to lawful currency.

Capital market data: e.g. package tokenized capital or security pricing.

Indicator reference data: e.g. including interest rate into smart contracts of financial derivative instruments.

Time and interval data: the event trigger abased on the accurate SI

(international system of units) time measurement.

Weather data: e.g. the premium calculator based on weather forecast.

Sports events: relevant contracts predicting market behavior and sports betting.

Events occurring on other blockchains: interoperable functions.

Market price of ETH: e.g. oracle of gas price.

Themis mainly focuses on random number oracle, in-chain asset price oracle, computational oracle and other application scenes.

## 3. Solution of Themis

Themis provides a whole set of technical solutions, which mainly focuses on random number oracle, in-chain asset price oracle, computational oracle and other application scenes, including mortgage assets becoming the provider of data, identification verification, preventing attack algorithm, verifiable random function VRF, arbitration protocol and other modules, which constitute a set of complete Themis-Protocol.

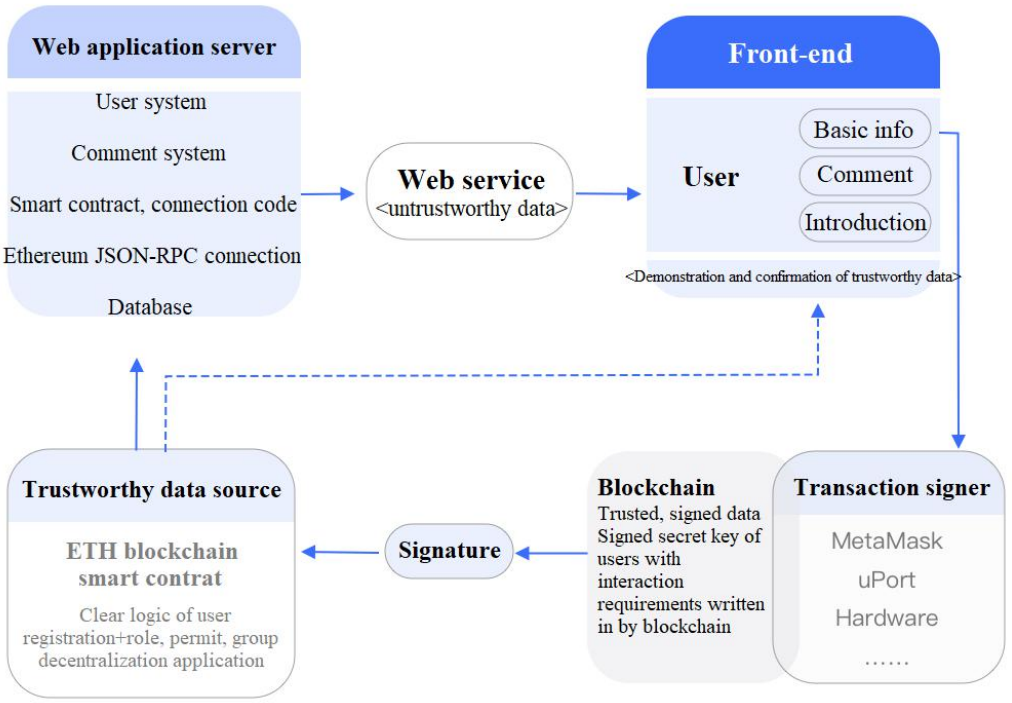


Figure 2. Schematic Diagram of Themis

### 3.1 The definitions of participants of Themis are as follows

**Data provider**: The participant providing various data in the protocol, including providers of verifiable random number, in-chain asset price, and verifiable hashrate. Anyone holding a certain amount of Themis can become data provider. When providing data, they need to pledge certain Themis (ERC-20). If the data provided finally takes effect, the data provider will acquire Themis

(ERC-20) award, whose quantity is in direct proportion to the Themis (ERC-20) under pledge.

**Verifier**: If the data provided by certain data provider is wrong, verifiers can identify such data. Verifiers can verify the data identified by VRF, verifiable computing equipment, standard in-chain asset price, and submit the verification result to arbitration nodes. Arbitration nodes will arbitrate the identification. If the arbitration is successful, verifiers will acquire the Themis (ERC-20) pledged by the other party, so as to acquire benefits. After identifying data, verifiers need to provide a new data, and such transaction does not need to provide Themis (ERC-20) for pledge.

**Data caller:** After calling Themis oracle contract and paying certain Themis (ERC-20) as the expense, any contract and account can become price caller. The Themis paid by data callers will also be distributed to data providers according to certain proportion.

### 3.2 Data provision and data verification

According to different specific application scenes, the ways in which data providers provide data are slightly different.

**Verifiable random number**

The data provider uses Verifiable Random Function (VRF) to generate verifiable random numbers and then input the random number and pledged Themis (ERC-20) into the smart contract. The verification contract on the chain verifies the uploaded random number. If the uploaded random number cannot be verified, the Themis (ERC-20) pledged by the data provider will enter the rewards pool (see: Ecological Design).

If the data provided by the data provider is verified, the data provider will be rewarded Themis (ERC-20), and the pledged Themis (ERC-20) will also be returned to the data provider later.

To better understand the principle of VRF, it is a good idea to understand hash functions such as SHA256, SHA3, etc. first. For an ideal hash function, its value range should be discrete and uniformly distributed. When given different input values, its output value should be irregular, and randomly scattered and distributed in the value range.

There is a simple variant of the hash function, hash function combined with the secret key, such as result = SHA256 (secret, info). In this function, in order to get the result, both info and secret need to be inputted. This function is also called hash function with key.

Simply speaking, Verifiable Random Function (VRF) is a hash function combined with asymmetric key, such as result = VRF_Hash (SK, info), in which SK is a private key that is kept secret, while the PK paired with SK is a public key and needs to be disclosed to the verifier. The specific operation process is as follows:

1. The data provider generates a pair of keys, PK and SK;

2. The data provider calculates result = VRF_Hash (SK, info);

3. The data provider calculates proof = VRF_Proof (SK, info);

4. The data provider submits the result and proof to the data verifier;

5. The data provider submits PK and info to the verifier;

6. The data verifier calculates whether result = VRF_P2H (proof) is established. If it is established, continue with the following steps, otherwise, the process ends;

7. The data verifier calculates True/False = VRF_Verify(PK, info, proof), True means the verification passed, False means the verification failed.

The so-called verification passed refers to whether the proof is generated through info, and whether the result can be calculated through the proof, so as to

deduce whether the info and result correspond to a match, and whether the data provided by the data provider is correct. In the entire operation process, the private key SK is never disclosed, but the verifier can deduce whether info and result match.

From the perspective of the evolution of the hash function:

The original hash function: info --> result

Hash function with key: (info, secret) --> result

Hash function with public key (aka VRF): (info, SK)-->(proof, verified by PK)--> result

Based on the above VRF principle, Themis designed the entire random oracle process that can be divided into two parts: random number provision and random number consumption:

**Random number provision**

1. The data provider uses its own private key to calculate the random number: result = VRF_Hash (SK, infoPro); where infoPro is the latest block number on the chain obtained in real time

2. The data provider calculates proof = VRF_Proof (SK, infoPro);

3. The data provider submits the PK, infoPro, result, proof and pledged Themis to the random number smart contract.

4. The smart contract first makes sure that infoPro is less than or equal to the current latest block number, infoPro is greater than the infoPro provided last time when the random number was provided, and the interval between infoPro and the latest block number is not greater than 300.

5. There can only be one random number provided by the address in the available random number sequence.

6. The smart contract performs VRF verification, and if the verification passes, the random number will be stored in the available random number sequence.

**Random number consumption:**

1. The data consumer requests Themis random number oracle and provides an infoReq as a variable in the subsequent budget

2. If the currently available random number contains a usable random number, then calculate proof = VRF_Proof(SK, info);

3. SK is the sender address of the requested random number. info = SHA256 (infoReq + available random number)

4. If there is no random number available, the data consumer will be informed and can conduct a new request after a certain period of time.

**In-chain asset price**

Take BTC/USDT as an example, if certain data provider intends to offer 1BTC=10000USDT, he needs to introduce the asset class offered "BTC" and the asset price 10000USDT (here, 10000USDT is only introduced as the price, and does not need to be transferred to smart contracts) and the Themis (ERC-20) under pledge into oracle contracts as parameters. The whole process is complete open. Anyone can become data provider, and the price and pledge scale are set by themselves. The more pledges, the more rewards. The specific formula is result=Max (m, x*Themis), in which x is the variable parameter which changes with time, and m is the cap if every reward.

After asset, price and Themis (ERC-20) under pledge of the data provider are submitted to offer contract, if any verifier considers such price as questionable, it can identify such price. Afterwards, arbitration node will execute arbitration, determine the offer time according to the block of offer, and inquire the true price of current head exchange at such time. If the difference between the price

provided by the data provider and the true price is greater than the threshold value, the data provider will lose the Themis (ERC-20) under pledge. Such mechanism ensures that the offer is the fair price on the market.

**Verifiable computational hashrate**

1. The initiator of computational task uploads the codes to be executed and provide the commission.

2. It the data provider discovers such task and considers the commission is acceptable, it will execute the computational task and introduce the computed result and the Themis (ERC-20) under pledge into smart contracts.

3. The verifier re-executes the task. If it is found that the data provider conducts fraud, the verifier can submit its own computed result and the Themis (ERC-20) under pledge to the arbitration node.

4. The arbitration node will distribute the Themis (ERC-20) under pledge of the fraud party to the verifier and the arbitration node by re-executing the computational task or truly calculating the result.

5. If nobody can provide evidence to prove that the data provider conducts fraud within a certain period of time, the data provider will acquire the Themis (ERC-20) commission provided by the task initiator.

**Data verification period**

From the time that the data is submitted to the oracle, the verification period of every data submission is limited. Upon the expiry of the verification period, the data which has not been challenged will be called valid data, including the data provided and the Themis (ERC-20) under pledge.

For in-chain asset price, the valid data will be calculated according to certain algorithm to generate the queryable price, so as to provide inquiry service of price.

**Anti-attack algorithm**

If the scale of Themis called is comparatively large, there may be attacker. Attackers will tamper certain normal data or launch malicious challenges, expecting no upgrade of data (as once the data is challenged, it will be impossible to be adopted and updated). Attackers are willing to sacrifice the Themis (ERC-20) under pledge to exchange for greater benefits.

We prevent the attack by improving the cost of attackers:

Firstly, when verifiers challenge, they have to pledge certain Themis (ERC-20). After that, they have to leave a new data or asset, which means that, after verifiers challenge, they must leave correct data or lose more Themis (ERC-20). It is inevitable that there will be other verifiers on the market for interest arbitrage and data correction.

Secondly, in order to increase attackers' cost, the pledge scale for all the verifiers is arranged as follows: if the pledge scale for data providers is n1, then the pledge scale for verifiers is n2=mx1, in which m>1, meaning that verifiers must challenge with at least twice the pledge scale. Take m=2 as an example, the initial pledge is n=100 Themis (ERC-20). With continuous challenge, n1=200, n2=400, n3=800... and so on. Attackers need to pay extremely great cost to distort the market price for a certain period of time.

**Ecological design**

Data providers acquire Themis (ERC-20) by paying the in-chain service fee and pledging certain Themis (ERC-20). When verifiers make profits by challenge, as long as it is not a malicious challenge, verifiers will have no arbitrary risk. Thus, for verifiers, the cost benefit is relatively clear. And for data providers, the pledge scale determines the benefits.

It is not enough to complete the logic closed loop by the Themis (ERC-20) awarded only, which returns to our original intention of constructing price oracle:

out-of-chain data is the rigid demand of various smart contracts and the most important infrastructure. Thus, the developer or user of any smart contract should pay corresponding expense when calling Themis, and the benefit of such part should also be distributed to data providers in proportion.

Because of the punishment mechanism of Themis, most data providers will provide correct data, which will result in a result that verifiers are unprofitable. Verifier is an important link in the entire ecology to ensure security. If nobody is wiling to play such role, the security of the entire system will be destructed.

In order to guarantee the ecology, Themis design a mechanism called accumulative bonus.

The basic principle is that arbitration nodes will randomly select some proposals as the wrong data provided by data providers (called forced error). When the verifier launches a challenge, the data provider will not be punished and the verifier will acquire accumulative bonus as the reward.

## 4. Application of Themis

When we own **verifiable random numbers**, the products depending on random number can be designed, such as:

1) Social public service: The application of random number can be seen everywhere in daily social life. For example, school admission lottery, license plate lottery, house-purchase lottery etc. all need fair and transparent random number system to reduce the management cost of the society.

2) Lottery activity: Almost all the electronic lottery activities that we can see will use random number, and the generation of verifiable random number is critical. The generation mechanism of random number is unpredictable and unattackable, which is the fundamental core of lottery activities.

3) Gambling products: By true random number, various in-chain gambling games can be designed, such as lottery, DuoBao, 777 etc. which gamble in the way of decentralization, to make the whole process more just and transparent, avoid hacker attack, artificial cheating and other behaviors.

4) Decentralized games: Various on-line card games depend on the generation of random number. Effective random number mechanism can ensure the randomness of the game to the largest extent and avoid system attack.

When we own **in-chain asset price**, more DeFi products can be applied based on Themis, such as:

1) Decentralized transactions: Current decentralized transactions are mainly transactions matched by point-to-point offer. However, the new decentralized transactions with free market-maker system are more suitable for users' demand. Market makers offer prices bilaterally and participate in transactions bilaterally. Decentralized exchanges require to acquire objective and fair in-chain asset price to balance their own offer system.

2) Self-settled mortgage loan: It is also the most widely applied Defi service.

As it owns in-chain price, loan contracts involving liquidation or automatic settlement can quota such price to complete the trigger of certain conditions, so as to automatically complete loan behavior system.

3) Products of futures and options: Similar loan products, products of futures and options with general meaning need decentralized institution to conduct forced liquidation etc., and decentralized platforms own in-chain price oracle mechanism so that they can capture in-chain price in real time, so as to automatically realize product contracts without undertaking the risk of centralization.

When we own verifiable hashrate, products depending on hashrate can be designed, such as:

1) Computational tasks exceeding the upper limit of in-chain service fee. Take ETH as an example, there is upper limit for the gas of single ETH transaction. In other words, when the computation burden of a smart contract exceeds the upper limit, such smart contract cannot be executed completed. Similarly, within the limit of gas, if gas consumes a lot, the transaction service fee of calling single smart contract is excessively high, resulting in the phenomenon of wasted service fee. Verifiable out-of-chain hashrate can well solve such problem. Doge Ethereum bridging is an example. The algorithm proving the workload of Dogecoin—Scrypt—is a function with strong memory requirement and intensive computation. It cannot finish the computation within the upper limit of gas in ETH block.

2) Iterative computation consuming a lot of computation time. Some iterative computations consume a lot of time like SPV demonstration. If all the computations re conducted in the chain, it needs to compute all the nodes to accomplish the agreement. Thus, the operation efficiency of the entire chain will be reduced. If these computations are put out of the chain, the problem above will be solved by callback the type via event listening mechanism.

3) Mass-data computation which cannot be conducted in the chain as the

in-chain storage capacity is limited. For example, some computations need the support of mass data, but these data cannot be stored on ETH limited by the storage capacity of ETH. These problems can be effectively solved by putting these computations and computational data out of the chain and verifying by verifiable mechanism.

## 5. Themis Token

Themis Token is named as MIS with the total amount of one billion. It is expected to use 10% for preliminary project promotion. The remaining 90% are produced by mining, in which 75% are directly awarded to data providers, 10% to developers, and 5% as reward for arbitration nodes and ecological incentive. The production of mining will be progressively decreased and released with ETH block.

The release plan of developer and arbitration node and ecological incentive is as follows:

| Height of ETH block | Themis released per block |
|---|---|
| 10514999~12514999 | 25 |
| 12514999~14514999 | 20 |
| 14514999~16514999 | 15 |
| 16514999~76514999 | 0.5 |

The release plan of data provider incentive is as follows:

| Height of ETH block | MIS released per block |
|---|---|
| 10514999~14514999 | 20 |
| 14514999~18514999 | 18 |
| 18514999~22514999 | 16.2 |
| ... | ... |

Every 4 million blocks, the MIS awarded per block reduces by 10%. The reward per block at present is 20 MIS.

Miners acquire MIS by providing verifiable random number or offering the price of in-chain assets. Whenever miners call mining contracts, the system will charge no service fee (excluding the service fee of ETH). Miners need to pledge certain amount of, at least 100, MIS whenever calling the contract.

The computation of MIS mining quantity of miners' every mining transaction:

At first, the number of MIS mining reward N included in the block of package mining transaction shall be worked out. If the height difference of such block from the last block including mining transaction is y, then:

$N=y \times 20$

It means that, if no mining transaction occurs within a certain period of time, the first new block including mining transaction will acquire all the MIS rewards before. In such way, miners will be motivated to keep mining, so as to maintain the stability of Themis ecology.

The MIS mining quantity of such mining transaction is M:

$$M = \frac{X_i}{x_1 + x_2 + x_3 + \ldots + x_n} \times N$$

In which, X is the rank of MIS pledge quantity in such block. People holding the same quantity of MIS rank the same.

Assume that there are 12 mining transactions in a block, the rank according to the MIS under pledge of every transaction is:

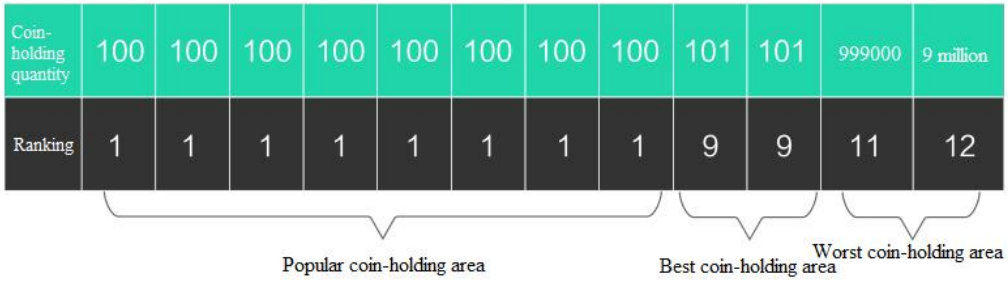


Figure 3. Schematic Diagram of MIS Pledge Ranking

Coin-holding ranking is based on jumping ranking weighting algorithm other than the weighted average of users' coin-holding quantity, with the purpose of avoiding MIS from controlled by the minority, monopoly, and breaking up major

clients, as well as realizing community win-win of Themis with best efforts.

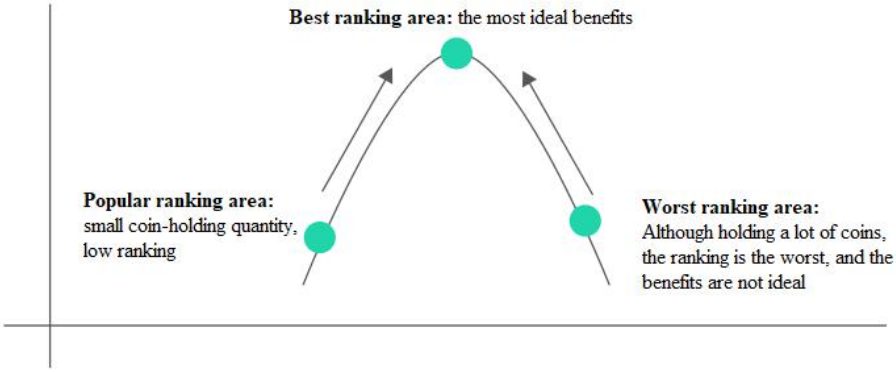


Figure 4. Schematic Diagram of Every Ranking Area

Users ranking in the best ranking area will acquire the most benefits, which provides good mechanism guarantee to attract more users to participate in mining. Meanwhile, it is good for making data providers more scattered, so as to ensure the decentralization of oracle system.