1 Dividends

The dividendable fund pool will contain a fraction of X per ten thousand, which can be used for dividend distribution. For the total amount T in the dividendable fund pool and the sum of dividend weights of various token levels within the current DARC protocol, we can determine the dividend amount for each dividend unit as follows:

$$u = \frac{XT}{10000W}$$

Here, u represents the dividend amount for each dividend unit, T is the total amount in the dividendable fund pool, and W represents the sum of dividend weights for all dividendable tokens across different token levels within the current DARC protocol.

For each user, assuming that the sum of dividend weights from various token levels that user i holds is D_i, the total dividend amount U_i that user i can receive in this round is given by:

$$U_i = \frac{D_i}{10000}u$$

For the DARC protocol, three dividend distribution methods can be employed:

Transaction-Based Dividend Distribution: The first approach involves distributing dividends based on the quantity of dividendable transactions. This is achieved by setting the dividend transaction cycle counter, N, to a reasonable value. Upon receiving at least N dividendable transactions, the instruction `OFFER_DIVIDENDS` becomes executable. Upon execution, the dividendable fund pool and dividendable transaction cycle counter are reset to zero, initiating a new count. This process repeats, waiting for the next N transactions and subsequent dividend distribution operation.

Time-Based Dividend Distribution: The second method entails dividend distribution based on time. Setting N to 1, an additional plugin can be introduced, allowing `OFFER_DIVIDENDS` to be invoked no less than S seconds after the previous call within the entire DARC instance. In this manner, dividends can be scheduled every 2 weeks, 4 weeks, 3 months, 6 months, or 1 year. This approach aligns more closely with traditional corporate dividends.

Fund-Pool-Based Dividend Distribution: The third method involves distributing dividends based on the total amount in the dividendable fund pool. Setting N to 1, an additional plugin enables the execution of `OFFER_DIVIDENDS` whenever the amount in the dividendable fund pool exceeds a specified threshold. Following this design, a dividend distribution can be triggered after obtaining Y dividendable native tokens.

Users can choose any of the above methods or design plugins to create alternative, practical, and rational dividend distribution patterns based on different DARC organizational structures and methods.

It is crucial to note that performing `OFFER_DIVIDENDS` for every received dividendable transaction may result in significant gas fee wastage. This is due to the potential time complexity of $O(MNP)$, where M represents the number of token levels, N is the number of tokens in each level, and P is the total number of token holders. Therefore, implementing an efficient dividend distribution mechanism to save on gas fees is imperative.

Additionally, funds in the dividendable fund pool are not locked within the smart contract by the DARC protocol. The protocol does not safeguard dividends in cash, and `OFFER_DIVIDENDS` only performs a calculation, storing each token holder's upcoming dividends in a withdrawable dividend balance. When the operator withdraws dividends, it is impossible if the DARC protocol lacks a sufficient quantity of native tokens. To protect the dividend rights of specific or all token holders, additional plugins must be designed.