Resolve Token Distribution

Tokenized Debt as Trust

source code:https://etherscan.io/address/0x91683899ed812c1ac49590779cb72da6bf7971fe#code

Fair distribution of a token supply can be accomplished through a smart contract that functions as a pyramid scheme which inverts a centralizing force as an incentive engine for distribution. I believe this method of distribution can be used to enhance the delegate layer of (adaptive) D/PoS sidechains, sharding, and other protocols, or function as the fundraising & governance distribution mechanism for DAOs. Oracles even.

How is this distribution accomplished? By inverting a pyramid scheme contract. This is an example of a pyramid scheme contract:
https://test.jochen-hoenicke.de/crypto/ponzitoken/

The contract has an internal balance of “bonds” that it generates and burns on buys and sells respectively. The price of these bonds is controlled by the contract with the bancor formula. Every time someone buys into the contract using ETH, the price of bonds increases as the supply increases, and every time someone sells, the price decreases as the sold bonds are burned and the supply decreases. Some people may know this as “proof of weak hands”.

In the example above, there is also a 5% fee. A fee is necessary, but that will be addressed later.

Like in all pyramid schemes, there are many more losers than winners, but this is what creates the opportunity for this “perfect” distribution. The contract will be distributing Resolve tokens every time someone sells their bonds back into the contract. They get the ETH value of the bonds they sold back and they also get resolve tokens. The amount of resolve tokens they receive is based on this formula:

```
Resolve tokens = input ETH * ( input ETH / output ETH )
```

There are three assets to note: ETH, contract bonds, and Resolve tokens.

For example: They invested 10 ETH into the contract and received bonds. Later, they sell their bonds for ETH again, but the price of the bonds has since gone down twofold, so they get less ETH, represented in the above formula as `10 ETH * (10 ETH / 5 ETH)= 20 Resolve tokens`. If the price had gone up twofold, it would

look like this: `10 ETH * (10 ETH / 20 ETH) = 5 Resolve tokens`. Therefore, ETH spent in the contract is either multiplied by its loss in value or divided by it's increase in value. Here are some more examples. Let "in" denote ETH paid in and "out" denote ETH paid out.

- `50 in * (50 in / 10 out) => 250 resolves and 10 ETH received`
- `16 in * (16 in / 256 out) => 1 resolve and 256 ETH received`
- `100 in * (100 in / 25 out) => 400 resolves and 25 ETH received`
- `10 in * (10 in / 50 out) => 2 resolves and 50 ETH received`

Note: All the above examples do not take a fee into account. Below you will find an example with a fee included.

As you can see, those who benefit from the upward price change of the pyramid don't receive as many Resolve tokens. Those who take the hardest hit in loss of value receive the most Resolve tokens.

The input and output happen in between the buy and sell fee. Assume the fee is 5%. The input is recorded after the first fee is subtracted, and the output is recorded before the final fee is subtracted. For example, if you put `20 ETH` in, the 5% fee is subtracted and your input is recorded as `19 ETH`. Assuming no price change/volatility in between the input and output, your output is recorded as `19 ETH` as well and only then is the final 5% fee subtracted which gives you `18.525 ETH`. This fee is paid in ETH to those who stake Resolve tokens into the contract and the fee percentage can also be changed by resolve-weighted votes.

However, this is not the entirety of the ecosystem. This fee exists in the first layer of the ecosystem to start rewarding those who have lost but is also necessary to prevent contracts from spamming rapid buys and sells. If there wasn't a fee then someone could multiply their resolve tokens without actually taking a loss.

What are resolve tokens used for? They can be staked in the core contract for dividends or staked in a D/PoS sidechain or other extensions to the ecosystem. In theory, they are for weighted votes applied to functions that span the scope of the ecosystem. These votes can be applied to configurations that are typically static in traditional PoS networks. Resolve holders can dictate the transaction fee, staking requirement, delegate requirements, number of delegates and other features.

This is a proof-of-stake token that whales can't buy out. They can't centralize the supply, because the moment they create demand for resolve tokens (purchasing in mass on an exchange), the supply inflates at a disproportionate rate as bonds are sold to match that demand. Also with any attempt to control the majority of the supply the whale is transfering value to previous holders while leaving themselves with nothing.

Finally no whale would want to generate resolve tokens by buying at the top of the pyramid just for everyone to dump on them.

When minting resolve tokens this way, there are 2 problems that arise. Inflation and manipulation.

Without a deflationary mechanism, the resolve token supply will indefinitely inflate, and distribution of that token can become very lopsided as the inflation becomes more disproportionate through time. To counterbalance this, resolve tokens can be invested back into the core contract to collect fees. When earnings on those fees are withdrawn, the resolve tokens used to earn them are burned, or dissolved.

The second problem comes with manipulation. Someone could buy into the contract and then immediately sell out just to get resolve tokens. It's possible that it may be worth paying the fee to do this. This would also give short term holders an advantage over long term holders, because they would get the same resolve reward. So there's a time multiplier put in place. This component looks at the average cashout time across the contract and factors that into the amount of resolves minted. If a buy and sell is immediate, 0 time is spent in the contract and therefore a 0 is multiplied when calculating resolve tokens are rewarded. If the length between the buy and sell is the average length most bonds are held for, then the multiplier is 1. If it's twice as long, the multiplier is 2, etc.

These things keep the supply from hyper inflating, ensuring the token supply distribution remains stable.

The final mechanic is the "flux fee". The fee percentage is the percentage of resolve tokens not being staked in the core contract. This says that "the ecosystem needs to be useful to invite growth".

```
Resolves = input ETH * ( input ETH / output ETH ) * (
                Relative Holding Time )
```