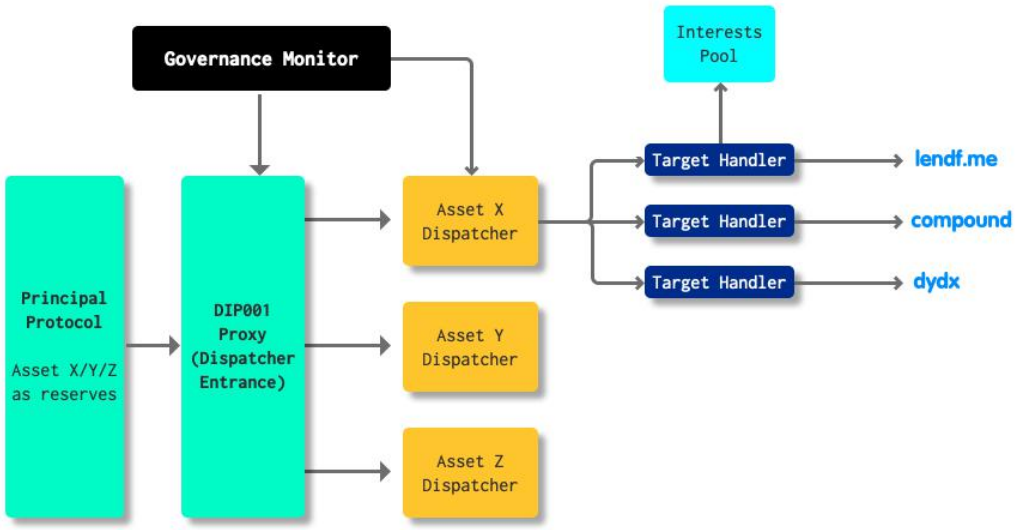
dForce Improvement Proposal (DIP001)

Summary:

DIP001 aims to provide a general framework to generate interest income out of reserves that can be implemented in any collateral-based protocols, for instance, with USDx protocol (a synthetic indexed stablecoin with a basket of stablecoins including USDC, PAX and TUSD locked as reserves). For introduction in details, please click here.

DIP001 governance framework serves to facilitate unlocking of collaterals from the initiated protocol to designated yielding protocols (i.e. Lendf.Me, and etc) and redistributing those earned interest to USDx holders. As a decentralized protocol, DIP001 is governed by DF holders through governance contract.

Architecture:



There are 6 modules under DIP001：

1. **Governance Contract (yet to be implemented)**

   The governance contract functions to administer DIP001, enabling governance token holders to decide on collateralization ratio, collateral re-utilization ratio (percentage of assets to be unlocked and re-supplied to lending protocols), interest pool (accrued interest for redistribution) and other governance issues through voting.

2. **Principal Protocol**

   Principal protocols are collateral-based DeFi protocols, i.e. USDx protocol, S-Swap (a Uniswap-like DEX with stablecoin as reserves), with the right of asset redistribution reserved. DIP001 can be authorized to generate interest out of reserves from initiated DeFi protocols by unlocking and re-supplying collaterals to yielding protocols at a pre-determined ratio, which can only be adjusted via on-chain governance. In the event the percentage of lent-out collaterals fail to satisfy, anyone can bring the funding pool back to a healthy level by triggering DIP001. The Governance Contract accommodates to terminating and withdrawing assets unlocked and re-supplied (in part or in whole) for interest generation.

3. **Dispatcher Entrance**

   Dispatcher entrance contract serves to manage the registration, destruction and other functions of the Dispatcher Contract.

4. **Dispatcher Contract**

   Dispatcher deals with assets with respect to collateralization ratio, designated lending protocols for asset allocation, percentage of asset allocation, data analysis, profits and other interfaces.

5. **Target Handler**

   Target Handler encapsulates interfaces interacting with lending contracts, including supply, withdraw, and etc. It also tracks data in relation to the principal, deposit amount, profits and other information.

6. Lending Protocol

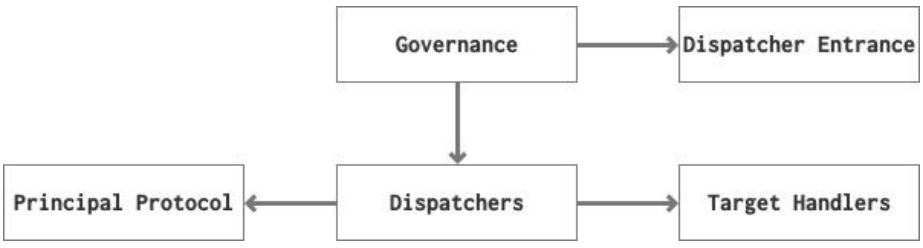
Lendf.Me, Compound.Finance, dYdX.exchange, for example.

Modules:

1. Authority Constraints

OnlyOwner: can be called by contract owner only

Auth: can be called by owner, and admin upon authorization by owner.



Note: The direction of arrow indicates module with inferior level of permission.

2. Key Functions

a) **Governance** (yet to be implemented)

Temporarily administration by the owner and admin before the launch of Governance.

b) Dispatcher Entrance

registDispatcher(address _fund, address _token, address _dispatcher)

Descriptions: to register for dispatchers

Parameters:

_*fund*: address of principle contract

_*token*: address of asset

_*dispatcher*: address of dispatcher

getDispatcher(address _fund, address _token)

Descriptions: to get the address of dispatcher

Parameters:

_*fund*: address of principle contract

_*token*: address of asset

c) Dispatcher

trigger()

Descriptions: to check and trigger the action of supply or withdraw if preconditions are satisfied, or remain still vice versa.

withdrawProfit()

Descriptions: to withdraw profits from all designated lending protocols

setAimedPropotion(uint256[] calldata _thPropotion)

Descriptions: to determine the percentage for asset allocation to different lending protocols

Parameters:

_thPropotion: proportion array

function drainFunds (uint256 _index)

Descriptions: to withdraw all principal plus interest gained for the TargetHandler

Parameters:

_index: index of TargetHandler

addTargetHandler(address _targetHandlerAddr, uint256[] calldata _thPropotion)

Descriptions: to add TargetHandler to the dispatcher

Parameters:

_targetHandlerAddr: address of new TargetHandler

_thPropotion: proportion array after adding new TargetHandler

removeTargetHandler(address _targetHandlerAddr, uint256 _index，uint256[] calldata _thPropotion)

Descriptions: to remove one TargetHandler

Parameters:

_targetHandlerAddr: address of TargetHandler to be removed

_index: index of the TargetHandler

_thPropotion: proportion array after removing the TargetHandler

setReserveUpperLimit(uint256 _number)

Descriptions: to determine the maximum of collateralization rate

Parameters:

_number: number, number / 1000 indicates percentage

setReserveLowerLimit(uint256 _number)

Descriptions: to determine the minimum of collateralization rate

Parameters:

_number: number, number / 1000 indicates percentage

setExecuteUnit(uint256 _number)

Descriptions: to determine the smallest amount of executing unit for an asset

Parameters:

_number: number

setProfitBeneficiary(address _profitBeneficiary)

Descriptions: to set the address of interest funding pool

Parameters:

_profitBeneficiary: address of interest funding pool

d) TargetHandler

setDispatcher (address _dispatcher)

Descriptions: to set current dispatcher

Parameters:

_dispatcher: address of dispatcher

deposit(uint256 _amounts)

Descriptions: to deposit principal to lending protocols

Parameters:

_amounts: amount of depositing

withdraw(uint256 _amounts)

Descriptions: to withdraw principal from lending protocols

Parameters:

_amounts: amount of withdrawal

withdrawProfit()

Descriptions: to withdraw all profits

function drainFunds ()

Descriptions: to withdraw principal plus interest gained