(ENG) BiFi Smart Contract: Code Architecture

1. Introduction

BiFi is a DeFi project on Ethereum that offers a financial service, which enables users to deposit various digital assets and borrow other digital assets against the deposits as collateral. BiFi is implemented by Solidity smart contracts.

Each supported digital asset on BiFi has a dedicated Market Handler Contract ("Market Handler"). All Market Handlers are intermediated by Market Manager Contract ("Market Manager").

Service Incentive Handler ("SI Handler") manages the reward program that rewards participants (service users and operators) that contributed in the growth and operation of BiFi with Service Incentive Tokens ("SI Tokens"). Liquidation

Manager Contract ("Liquidation Manager") manages the liquidation process for users whose Loan-To-Value (LTV) Ratio has exceeded a set threshold.

The BiFi components interact each other. For example, a user may borrow up to 75% of the value of previously deposited assets, and the value of any digital asset can be expressed as the product of its amount and current price. For this user (borrowing) operation, each Market Handler provides the user balance and the Oracle contract provides the current prices of each Token. Market Manager then validate and execute the action. The user also gets service incentive for this action; The SI Handler will provide SI Tokens when the user claims. When a price of the user token significantly decreases, the user's debt can be liquidated by anyone via the Liquidation Manager contract.

The logic contracts (Market Handler, SI Handler, Market Manager) are designed so that they can be updated without changing the data storage. Each logic contract has its own data storage contracts to store the data.

2. Architecture

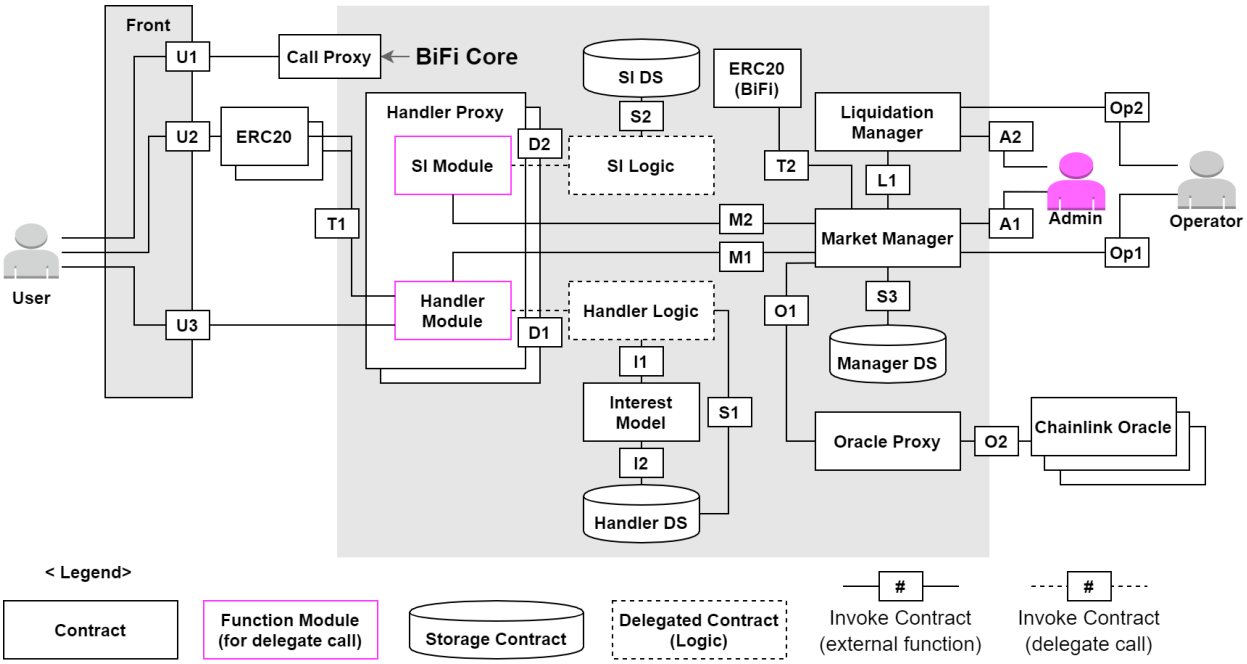
2.1. Notation

- DS: Data Storage
- SI: Service Incentive

2.2. Overview

BiFi service has a BiFi Core contracts and interface component contracts. BiFi Core consists of basic service logic, HandlerProxy contract (proxy for Token Handler that supports upgrades), and CallProxy contract (collection of view functions to support web front-end). Interface components integrate third party contracts (e.g., Third Party Token Contract and Chainlink Oracle Contract).

The diagram below outlines the overall structure of BiFi service. Boxes on the line show the names of transitions (call flows).



Participants

- **User**: Participant who deposits or borrows digital assets through BiFi Core
- **Operator**: Participant who updates service parameters
  - Refer to *Appendix II* and *Appendix III*
- **Admin**: Administrator who updates contracts and key parameters
- **Front**: Front-end service and MetaMask

Interface Component Contracts

- **ERC-20**: Third Party Token Contract (Tokens supported on BiFi)
- **Chainlink Oracle**: Price Feed Contract from Chainlink providing price information

2.3. BiFi Core Contract

2.3.1. Handler Proxy

Handler Proxy link the logic and data storage (DS) of Market Handler (2.3.2) and SI Handler (2.3.3). This allows the logic and DS to stay separate and the logic to be updated while preserving the data.

Each supported digital asset is assigned its own Handler Proxy. Each Handler Proxy connects Handler Logic and Handler DS (both for Market Handler and SI Handler). Users interact with Handler Proxy, which uses Market Handler Logic and SI Handler Logic and stores data in Handler DS.

The implementations for ERC-20 tokens and ETH (Ethereum Native Coin) are functionally identical, but are slightly different in implementation for sending and receiving coins.

Market Module (function module)

- Delegates user request to Market Handlers
  - Calls Market Handler Logic (via `delegatecall()`)
  - Actions: deposit, withdraw, borrow, repay
- Delegates Market Manager requests to Market Handlers
  - Updates interest amount
  - Executes liquidation

SI Module (function module)

- Calls on SI Handler Logic (via `delegatecall()`)
  - Calculate reward amount for user (SI Tokens)
  - Update reward parameter for operator

2.3.2. Market Handler

Market Handler provides deposit and borrow service for the digital asset it manages. The methods of Market Handler is called by Handler Proxy running `delegatecall()`.

Admin can change the Market Handler Logic while preserving the data of Handler Proxy and Handler DS.

Handler Logic

- BiFi Service Logic
  - Updates deposit and borrow amounts, based on user action
  - Calls Interest Model

- Logic for Liquidation Process

Handler DS

- Stores deposit amount and borrow amount of the user
- Stores total deposit amount and borrow amount of the digital asset
- Stores interest condition variables (e.g., `minimumInterestRate`, `liquiditySensivitiy`)
- Stores collateral condition variables (e.g., `borrowLimit` , `marginCallLimit`)

For more details on the stored data, refer to *BiFi Smart Contract Interest Model Design*.

2.3.3. SI Handler

SI Handler updates reward amounts of SI Tokens for users who deposit or borrow digital assets. Generally, the methods of SI Handler is called by Handler Proxy via `delegatecall()`.

Admin can change the SI Handler Logic while preserving the data of Handler Proxy and Handler DS.

SI Logic

- Updates SI Token reward amounts, based on the amount of assets of the user
- Updates reward allocation parameter determined by Market Manager

SI DS

- Stores reward amount for the user
- Stores reward parameter

For more details, refer to *BiFi Smart Contract Service Incentive Model*.

2.3.4. Interest Model

Interest Model provides the logic to calculate the interest for the user. In particular, it calculates the interest upon user action.

For more details, refer to *BiFi Smart Contract Interest Model Design*.

2.3.5. Market Manager

Market Manager calculates the information for all digital assets in the market, as well as manages other informations and functions for the entire market.

For example, if a user requests a borrow, the Market Manager accesses and confirms the asset information of the user across all Market Handlers.

Manager Logic

- Aggregates asset information of all Market Handlers
- Updates interest amount, by sending request to each Market Handler
- Updates reward amount, by sending request to each SI Handler
- Updates reward parameter, by sending request to each SI Handler
  - Refer to *BiFi Smart Contract Service Incentive Model*.
- Provides functions for the admin
- Provides function to claim SI Tokens

Manager DS

- Stores Handler information
- Stores reward parameter
  - Refer to *BiFi Smart Contract Service Incentive Model*.

Liquidation Manager

- Executes liquidation of assets of liquidation targets (`delinquentBorrower`), by calling Market Manager, which in turn calls the Market Handler
  - Gives liquidators the collateral (deposit) of liquidation targets

2.3.6. Oracle Proxy

Oracle Proxy accesses the Price Feed Oracle Contract of Chainlink.

- Register Chainlink Oracle
- Provides price information to BiFi Core

2.3.7. Call Proxy

Call Proxy collects all the information from BiFi and sends to the web front-end (e.g., view functions).

3. Basic Action Flows

Below provides a broad overview of the action flow for users, operators, and liquidators on BiFi, in terms of transitions (call flows) between components outlined in the diagram in *2.2 Overview*.

For indexed connection among components, refer to the **Appendix: transition**.

3.1. User Flow: Deposit, Withdraw, Borrow, Repay

While different actions are implemented differently for optimization, below outlines the shared action flows.

- `U2`: Approve
- `U3`: Identify action (e.g., Deposit, Withdraw, Borrow, Repay)
- `D1`: Execute identified action logic
- `M1`: Request update of interest and reward amounts
  - `M2`: Request update of user reward amount
    - `D2`: Run user reward calculation logic
    - `S2`: Update user reward amount
  - `M1`: Request update of interest
    - `D1`: Run interest update logic
    - `I1`: Calculate interest
    - `I2`: Update asset information for the user and the entire market.
  - `O1,O2`: Get price information for each Handler and calculate aggregate asset information of the user
- `S1`: Check conditions for user action and update asset amount of the user
- `T1`: Transfer (`transfer` or `transferFrom`) tokens

3.2. Operator Flow: Reward Parameter Update

- `Op1`: Request update of Handler reward parameter
- `S3`: Determine global reward parameter
- `O1,O2`: Determine reward parameter for each Handler and calculate α-score, based on asset information of each Handler
- `M1`: Request update reward parameter of each Handler
  - `D1`: Run Handler logic
  - `S1`: Request asset information of each Handler
- `M2`: Request update reward parameter
  - `D2`: Run SI Handler parameter update logic
  - `S2`: Update reward parameter

3.3. Operator Flow: Liquidation

- `Op2`: Initiate liquidation process, determine eligibility for liquidation
  - `L1`: Request update of reward and interest
    - `O1,O2`: Get token price information
    - `M1,M2`: Update reward and interest
- `L1`: Send liquidation target, liquidation target token, liquidation reward token, liquidator, and liquidation amount
  - `M1`: Execute liquidation, and get asset information of liquidation target and liquidator
    - `D1`: Run liquidation logic
    - `S1`: Update asset information of liquidation target and liquidator
- `L1`: Determine liquidation reward amount and send liquidation amount
  - `M1`: Execute liquidation reward process
    - `D1`: Run liquidation reward logic
    - `S1`: Update asset information of liquidation target and liquidation

Appendix: Transitions

Below explains the transitions in the diagram shown in *2.2 Overview*.

User Interaction

- `U1`: Display information collected by Call Proxy to user through the front-end
- `U2`: Get token balance information of the user, or approve token
- `U3`: Actions (deposit, withdraw, borrow, repay)

Token Interaction

- `T1`: Transfer tokens according to the user action
- `T2`: Transfer SI Token to user

Delegate Call Interaction

- `D1`: Delegate call Market Handler Logic
- `D2`: Delegate call SI Handler Logic

Interest Model Interaction

- `I1`: Execute interest calculation according to the user action
- `I2`: Update asset information of Handler based on the calculated interest amount

Storage Interaction

- `S1`: Update asset amounts based on user action
- `S2`: Update reward amounts based on user action
- `S3`: Update Handler information and Global Reward Parameter

Manager Interaction

- `M1`
  - Request interest update
  - Access asset information for each Handler and of the entire market
  - Request Liquidation Process
- `M2`

    - Request reward update
    - Update reward parameter

Liquidation Manager Interaction

- `L1`
    - Request liquidation process
    - Determine eligibility for liquidation
    - Calculate liquidation reward

Oracle Interaction

- `O1`: Provide token price
- `O2`: Access Chainlink Price Feed Oracle Interface

Admin Interaction

- `A1`
    - Update Contract
    - Update interest parameter
    - Update global reward parameter
- `A2`
    - Manage contract

Operator Interaction

- `Op1`
    - Update interest for the entire market
    - Update reward parameter for Handler
- `Op2`
    - Execute liquidation process