

(KOR) BiFi Smart Contract: Code Architecture

1. Introduction

2. Architecture

2.1. Notation

2.2. Overview

Participants

Interface Component Contracts

2.3. BiFi Core Contract

2.3.1. Handler Proxy

2.3.2. Market Handler

2.3.3. SI Handler

2.3.4. Interest Model

2.3.5. Market Manager

2.3.6. Oracle Proxy

2.3.7. Call Proxy

3. Basic Action Flows

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

3.2. Operator's Flow: Reward Parameter Update

3.3. Operator's Flow: Liquidation

Appendix: transitions

1. Introduction

BiFi 서비스는 가상화폐를 예치하여 이자를 얻거나, 예치금을 담보로 가상화폐를 대출할 수 있는 금융서비스이다. BiFi 서비스는 Solidity로 작성된 여러 Ethereum Smart Contract로 구성된다.

BiFi가 지원하는 여러 가상화폐는 고유의 Market Handler Contract(Market Handler)를 가지고 있으며, 모든 Market Handler Contract는 Market Manager Contract(Market Manager)에 연결되어 Handler 간 연계 작업이 가능하다.

Service Incentive Handler(SI Handler)는 BiFi 서비스 유지에 공헌한 참여자에게 BiFi Token을 보상으로 지급하고, Liquidation Manager Contract(Liquidation Manager)는

총 예치 자산 대비 총 대출 자산의 비율(LTV, Loan-To-Value)이 정해진 수준을 넘긴 사용자의 자산을 청산하는 기능을 제공한다.

BiFi의 컴포넌트는 각각의 기능이 서로 상호작용한다. 예를 들어 사용자는 이전에 예치한 모든 가상화폐 가치의 75%만큼 대출을 실행할 수 있는데, Market Manager는 Oracle Contract에서 제공하는 각 가상화폐의 가격정보를 이용하여 사용자가 예치한 가상화폐 가치의 합계를 도출한다. 사용자의 요청("대출")이 발생하면, 각각의 Market Handler는 사용자의 잔액을 제공하고 Oracle 컨트랙트는 각 토큰의 가격을 제공한다.

한편, BiFi 서비스의 로직 Contract(Market Handler, SI Handler, Market Manager)들은 Data를 유지한 상태로 업데이트 할 수 있도록 설계되어 있다. 로직 Contract들은 데이터를 저장하는 각각의 Data Storage Contract를 보유하며, Handler Proxy를 이용하여 효과적으로 로직 Contract는 교체될 수 있다.

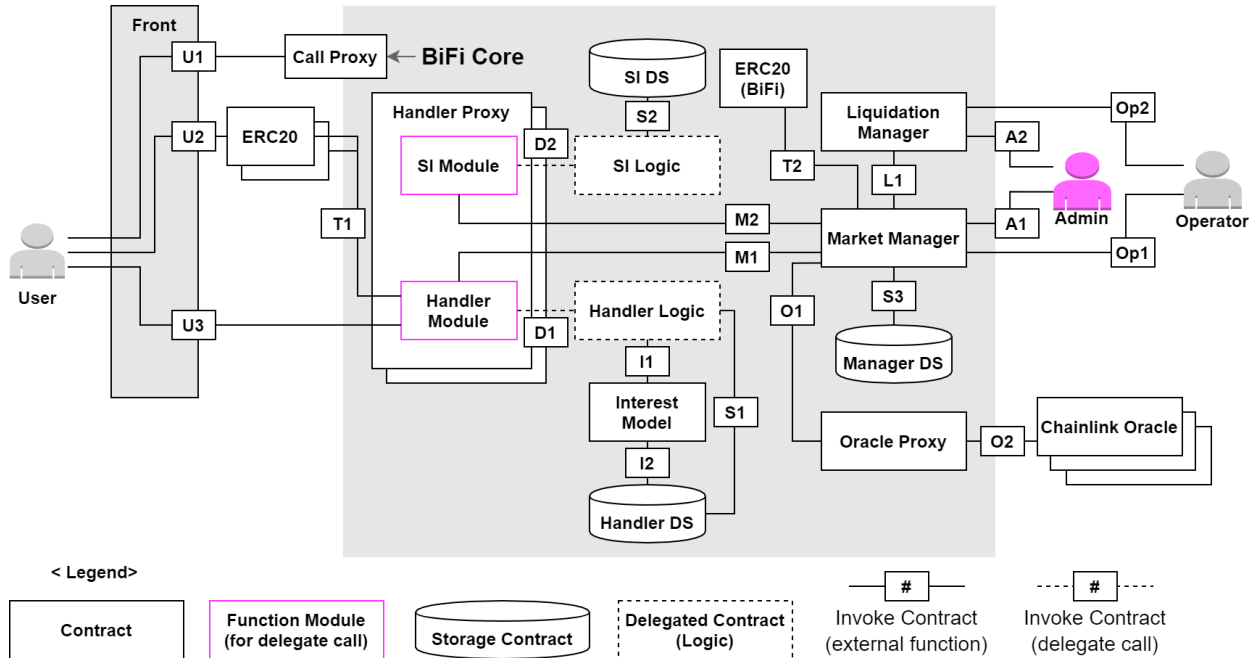
2. Architecture

2.1. Notation

- Abbreviation
 - DS: Data Storage
 - SI: Service Incentive

2.2. Overview

BiFi 서비스는 BiFi core와 interface contract들로 외부 기능과 연결하기 위한 components(Third-party Token contract, chainlink Oracle contract)로 구성된다. BiFi Core에는 기본적인 logic 외에 upgrade를 지원하기 위한 HandlerProxy (token handler를 위한 proxy) 와 web front-end의 표시를 지원하는 view function의 모음인 CallProxy contract가 있다. 아래 그림은 BiFi 서비스의 전체적인 구성도를 보여준다. 화살표 선상의 박스는 transition (call flow) 이름을 표시한다.



Participants

- **User:** BiFi core를 통해 가상화폐를 예치하거나 대출받은 사용자
- **Operator:** BiFi 서비스 파라미터를 업데이트 하는 임의의 네트워크 참여자
- **Admin:** BiFi 마켓 관리자 또는 컨트랙트 운영 관련 로직 실행자
- **Front:** front-end 서비스 및 메타마스크

Interface Component Contracts

- **ERC20:** Third-party 토큰 contract (BiFi 서비스에서 이용할 수 있는 token)
- **Chainlink Oracle:** 토큰 가격을 제공하는 Chainlink의 price-feed contract

2.3. BiFi Core Contract

2.3.1. Handler Proxy

Handler Proxy는 Market Handler Logic(section 2.3.2)과 SI Handler Logic(section 2.3.3)을 정보가 저장되는 data storage와 분리하여, logic을 교체할 수 있도록 한다.

지원하는 암호화폐마다 별도의 Handler Proxy가 할당된다. 하나의 Handler Proxy는 정보를 저장하는 Handler DS와 logic이 저장되어 있는 Market Handler와 연결하여 서비스를

제공한다. 즉, Handler Proxy는 사용자가 서비스를 접근하는 entry point로서 logic이 포함된 Market Handler와 SI Handler를 library처럼 이용하고 정보는 Handler DS에 저장한다. 단, ERC20 토큰을 위한 Contracts와 ETH(Ethereum Native Coin)을 위한 Contracts는 구현상 약간의 차이가 있음을 유의해야 한다.

- **Market Module(function module)**

- 사용자 서비스 요청 처리를 위해 Market Handler 로직을 delegate call
 - 액션 유형: 예금(deposit), 출금(withdraw), 대출(borrow), 상환(repay)
- 매니저 요청 처리를 위해 Market Handler 로직을 delegate call
 - 이자 업데이트
 - 청산 실행

- **SI Module(function module)**

- 사용자의 서비스 보상량 계산 처리를 위해 SI Handler 로직을 delegate call
- 운영 사용자의 보상 파라미터 업데이트 처리를 위해 SI Handler 로직을 delegate call

2.3.2. Market Handler

Market Handler는 자신이 담당하는 Token에 대한 예치 및 대출 서비스를 제공한다. Market Handler의 method들은 Handler Proxy에 의해 `delegatecall()` 을 통해 호출된다. Admin은 Handler Proxy와 SI DS를 변경하지 않고 SI Handler의 logic 만을 변경할 수 있다.

Handler Logic

- BiFi 서비스 로직
 - 사용자 액션에 따른 예금, 대출 양 업데이트
 - 이자 모델 호출
- 청산 프로세스를 위한 로직

Handler DS

- 사용자의 예금, 대출양 저장
- 전체 예금, 대출양 저장

- 이자 파라미터 저장(e.g., `minimumInterestRate`, `liquiditySensitivity`)
- 사용자의 자산 활용에 대한 조건 변수 저장

상세한 내용은 *BiFi Smart Contract Interest Model Design* 문서 참조

2.3.3. SI Handler

SI Handler는 자신이 담당하는 Token에 대한 예치 및 대출을 보유한 사용자에게 BiFi 토큰을 보상으로 지급하는 로직을 가진다. 일반적으로 SI Handler의 method들은 Handler Proxy에 의해 `delegatecall()` 로 호출된다.

Admin은 Handler Proxy와 SI DS를 변경하지 않고 SI Handler의 logic 만을 변경할 수 있다.

SI Logic

- 사용자 액션에 따른 보상 양 업데이트
- Market Manager에서 결정된 핸들러별 보상 분배 파라미터 업데이트

SI DS

- BiFi 사용자 보상량 저장
- 보상 파라미터 저장

더 자세한 설명은 *BiFi Smart Contract Service Incentive Model* 참조

2.3.4. Interest Model

BiFi 사용자에게 지급될 이자를 산출하는 로직을 제공한다. 특히 이자 지급 시점의 이자율 산출 기능을 제공한다.

더 자세한 설명은 *BiFi Smart Contract Interest Model Design* 문서 참조

2.3.5. Market Manager

BiFi에서 지원하는 여러 종류의 자산을 보유한 BiFi 사용자의 종합 자산 정보들을 산출하는 기능을 제공한다.

BiFi에서 지원하는 digital asset를 포함한 서비스의 종합적인 정보 및 기능을 관리한다.

예를 들어 사용자의 대출 요청이 있는 경우에 Market Manager는 연결된(지원하는) Market Handler들에서 해당 사용자의 자산 정보를 조회하여 검증한다.

Manager Logic: BiFi 컨트랙트 서비스 로직

- 핸들러별 자산 정보 통합
- 이자 업데이트 서비스 로직 (핸들러 기능 이용)
- 리워드 업데이트 서비스 로직 (SI 핸들러 기능 이용)
- 리워드 파라미터 업데이트 서비스 로직 (SI 핸들러 기능 이용)
 - *BiFi Smart Contract Service Incentive Model* 문서 참조
- 서비스 운영 사용자(Admin)의 기능 제공
- 보상 토큰 클레임 기능 제공

Manager DS

- 핸들러 정보 저장
- 리워드 파라미터 저장
 - *BiFi Smart Contract Service Incentive Model* 문서 참조

Liquidation Manager

- 청산 대상자(`delinquentBorrower`)의 자산을 청산(매니저를 통해 Handler 로직 실행)
 - 청산 대상자의 담보 자산(deposit)을 청산자에게 제공

2.3.6. Oracle Proxy

BiFi core는 Oracle Proxy contract를 통해 Chainlink 등의 Price Feed Oracle contract에 접근할 수 있다.

- Chainlink의 Price Oracle 등록
- price 정보 제공

2.3.7. Call Proxy

- BiFi 컨트랙트들을 통해 BiFi 시스템 정보를 수집하고 프론트에 전달

3. Basic Action Flows

BiFi core를 활용하는 BiFi user, operator와 liquidator의 개략적인 action flow를 설명한다. 이후 설명들은 "2.2 Overview"의 도식도에서 component 간 transition (call flow) 이름으로 표시한다.

Indexed Connection Among Components는 **Appendix: transition**을 참고하세요.

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

사용자 액션 유형별로 최적화에 따라 차이점이 있지만, 공통적인 흐름에 대해서 설명합니다.

- **U2**: Token Approve
- **U3**: Action 구분(e.g., Deposit, Withdraw, Borrow, Repay)
- **D1**: 구분된 Action에 맞는 Handler Logic 실행
- **M1**: 전체 이자, 리워드 업데이트 요청
 - **M2**: 사용자 보상 업데이트 요청
 - **D2**: 사용자 보상 계산 로직 실행
 - **S2**: 보상량 업데이트
 - **M1**: 사용자 이자 업데이트 요청
 - **D1**: 이자 업데이트 로직 실행
 - **I1**: 이자 계산
 - **I2**: 이자 결과에 따른 사용자, 글로벌 자산 업데이트
 - **O1, O2**: 핸들러 가격 획득 및 사용자 전체 자산 정보 계산
- **S1**: 사용자 액션 실행 조건 검사 및 변화량 업데이트
- **T1**: 토큰 전송

3.2. Operator's Flow: Reward Parameter Update

- **Op1**: 핸들러 리워드 파라미터 업데이트 요청
- **S3**: 글로벌 리워드 파라미터 결정
- **O1, O2**: 핸들러별 자산 정보로 Alpha 계산 및 핸들러별 리워드 파라미터 결정
- **M1**: 전체 핸들러 자산 정보 요청
 - **D1**: 핸들러 로직 실행
 - **S1**: 핸들러 전체 자산 정보

- **M2**: 리워드 파라미터 업데이트 요청
 - **D2**: SI 핸들러 파라미터 업데이트 로직 실행
 - **S2**: 리워드 파라미터 업데이트

3.3. Operator's Flow: Liquidation

- **Op2**: 청산 실행, 청산 가능 결정
 - **L1**: 보상, 이자 업데이트
 - **O1, O2**: 토큰 가격정보 획득
 - **M1, M2**: 보상, 이자 업데이트
- **L1**: 청산 대상자, 대상 토큰, 보상 토큰, 청산자, 청산양 전달
 - **M1**: 청산 실행 및 청산자의 자산 정보 획득
 - **D1**: 청산 로직 실행
 - **S1**: 청산자, 청산 대상자 자산 정보 업데이트
- **L1**: 청산에 대한 보상량 결정 및 보상량 전달
 - **M1**: 청산에 대한 보상 실행
 - **D1**: 청산 보상 로직 실행
 - **S1**: 청산자, 청산 대상자 자산 정보 업데이트

Appendix: transitions

"2.2 Overview"의 도식도의 comment사이 transition을 설명한다.

User Interaction

- **U1**: CallProxy에서 수집된 정보를 프론트를 통해 사용자에게 제공
- **U2**: 사용자 토큰 잔액 정보 및 토큰 approve
- **U3**: BiFi 시스템의 사용자 액션(예금, 출금, 대출, 상환)

Token Interaction

- **T1**: 사용자 액션에 따른 토큰 전송

- **T2**: SI Token 전송

Delegate Call Interaction

- **D1**: BiFi 마켓의 주요 로직인 Market Handler 로직을 Delegate Call
- **D2**: BiFi 시스템의 보상 로직인 SI Handler 로직을 Delegate Call

Interest Model Interaction

- **I1**: 사용자 액션에 의해 이자 계산 실행
- **I2**: 이자 계산 결과를 핸들러 자산 정보에 업데이트

Storage Interaction

- **S1**: 사용자 액션에 따른 자산 업데이트
- **S2**: 사용자 액션에 따른 보상 업데이트
- **S3**: BiFi의 핸들러 정보 업데이트, 전체 보상 파라미터 업데이트

Manager Interaction

- **M1**
 - 이자 업데이트 요청
 - 핸들러 별, BiFi 전체 자산 정보 전달
 - 청산 프로세스 요청
- **M2**
 - 보상 업데이트 요청
 - 보상 파라미터 업데이트

Liquidation Manager Interaction

- **L1**
 - 청산 실행 요청
 - 청산 가능성 검사
 - 청산 보상량 계산

Oracle Interaction

- **O1**: 토큰가격 제공

- **02**: Chainlink 토큰 가격 오라클 인터페이스

Admin Interaction

- **A1**
 - 컨트랙트 업데이트
 - 이자 파라미터 업데이트
 - 글로벌 보상 파라미터 업데이트
- **A2**
 - 컨트랙트 관리

Operator Interaction

- **Op1**
 - BiFi 전체 이자 업데이트
 - 핸들러 보상 파라미터 업데이트
- **Op2**
 - 청산 프로세스 실행