

Phân Tích Kỹ Thuật và Hướng Dẫn Triển Khai Solomon NFT Lending Protocol (SNL)

Người thực hiện: Vũ Văn Đăng
Email: dang19082004@gmail.com

Ngày 12 tháng 12 năm 2025

Tóm tắt nội dung

Tài liệu này trình bày phân tích chi tiết về kiến trúc, các công thức tài chính cốt lõi, và hướng dẫn triển khai cho Dự án Solomon NFT Lending Protocol (SNL). Giao thức sử dụng mô hình lãi suất thuật toán (Kink Model) và cơ chế thế chấp NFT (SCNFT) để quản lý rủi ro và thanh khoản.

1 Kiến trúc Hợp đồng (Contract Architecture)

Hệ thống được xây dựng từ ba hợp đồng chính:

1.1 CycloneERC20Token (Tài sản Vay và Cơ sở)

Là token ERC20 chuẩn (Symbol: CLT) dùng làm tài sản vay và tài sản cơ sở được khóa bên trong NFT.

- **Chức năng chính:** Cung cấp nguồn cung cấp ban đầu (initialSupply) và token miễn phí (FREE_TOKEN_AMOUNT) cho người dùng mới.

1.2 CycloneERC721Token (SCNFT - Tài sản Thế chấp)

Đóng gói một lượng ERC20 thành một NFT duy nhất (tokenId).

- **Khóa Thế chấp:** Mapping collateralApprover[tokenId] theo dõi địa chỉ Pool đang giữ quyền thế chấp (lock). Các hàm transferFrom và approve tiêu chuẩn bị chặn nếu NFT đang bị khóa.
- **Giá trị:** depositValue[tokenId] lưu trữ lượng token ERC20 bị khóa.

1.3 SolomonTreasury (Lãi Giao thức)

Quản lý Pool thanh khoản, lãi suất, và logic Vay/Trả nợ/Thanh lý.

2 Logic Tài chính và Công thức Cốt lõi

2.1 Tỷ lệ Sử dụng và Lãi suất (Kink Model)

Lãi suất Vay (Borrow Rate – R_B) phụ thuộc vào Tỷ lệ Sử dụng (U).

2.1.1 Tỷ lệ Sử dụng (U)

$$U = \frac{\text{totalBorrowed} \times \text{ONE}}{\text{totalDisposited}}$$

Với $\text{ONE} = 10^{18}$.

2.1.2 Công thức Lãi suất Vay (R_B)

- Trường hợp 1: Sử dụng tối ưu ($U < U_{kink}$)

$$\text{Slope}_1 = \frac{(R_{kink} - R_0) \times \text{ONE}}{U_{kink}}$$

$$R_B = R_0 + \frac{U \times \text{Slope}_1}{\text{ONE}}$$

- Trường hợp 2: Quá tải ($U \geq U_{kink}$)

$$\text{Slope}_2 = \frac{(R_{max} - R_{kink}) \times \text{ONE}}{(\text{ONE} - U_{kink})}$$

$$R_B = R_{kink} + \frac{(U - U_{kink}) \times \text{Slope}_2}{\text{ONE}}$$

(Trong đó, $R_0 = \text{baseRate}$, $R_{kink} = \text{kinkRate}$, $R_{max} = \text{maxRate}$, $U_{kink} = \text{kinkUtilization}$).

2.1.3 Lãi suất Cung cấp (R_S)

$$R_S = \frac{\left(\frac{R_B \times U}{\text{ONE}}\right) \times (\text{ONE} - \text{reserveFactor})}{\text{ONE}}$$

2.2 Tích lũy Lãi suất và Tính Nợ

2.2.1 Cập nhật Chỉ số Vay (borrowIndex)

$$\Delta t = \text{block.timestamp} - \text{lastUpdateTimestamp}$$

$$\text{New Borrow Index} = \text{Old Borrow Index} + \frac{(\text{Old Borrow Index} \times R_B \times \Delta t)}{\text{SECONDS PER YEAR} \times \text{ONE}}$$

2.2.2 Lãi suất Gộp và Dự trữ

$$\text{Gross Interest} = \frac{\text{totalBorrowed} \times (\text{New Borrow Index} - \text{Old Borrow Index})}{\text{Old Borrow Index}}$$

$$\text{Reserve Amount} = \frac{\text{Gross Interest} \times \text{reserveFactor}}{\text{ONE}}$$

2.2.3 Tổng Nợ Hiện tại (Total Debt)

$$\text{Total Debt} = \text{Principal} \times \frac{\text{Current borrowIndex}}{\text{Index At Borrow}}$$

2.3 Cơ chế Thanh lý (Liquidation)

Hàm `checkLiquidationStatus` sử dụng công thức so sánh Nợ hiện tại với một ngưỡng xác định từ Vốn gốc ban đầu.

$$\text{Condition} : \frac{(\text{Principal Ban Đầu} \times 95 \times \text{ONE})}{70 \times \text{ONE}} < \text{Total Debt}$$

3 Hướng dẫn Triển khai Chi tiết

3.1 Trình tự Triển khai

1. **Deploy CycloneERC20Token** → Nhận `ERC20_Address`
2. **Deploy CycloneERC721Token** (Dùng `ERC20_Address`) → Nhận `SCNFT_Address`
3. **Deploy SolomonTreasury** (Dùng `ERC20_Address` và `SCNFT_Address`) → Nhận `Treasury_Address`

3.2 Ví dụ Tham số Khởi tạo cho SolomonTreasury

(Đơn vị giá trị được scale 10^{18})

- `IERC20 _token: ERC20_Address`
- `CycloneERC721Token _scNFT: SCNFT_Address`
- `uint256 _baseRate (1%): 10000000000000000`
- `uint256 _kinkRate (5%): 50000000000000000`
- `uint256 _maxRate (20%): 200000000000000000`
- `uint256 _kinkUtilization (80%): 800000000000000000`
- `uint256 _reserveFactor (10%): 100000000000000000`

3.3 Các Bước Cấu hình Bắt buộc (Post-Deployment)

1. **Cấp Vốn Pool:** Admin gọi `ERC20.transfer(Treasury_Address, Amount)` để cấp thanh khoản cho Pool.
2. **Thiết lập Tham số Lãi suất:** Admin có thể điều chỉnh các tham số $R_0, R_{kink}, R_{max}, U_{kink}$ bằng hàm `setRateParameters` và `reserveFactor` bằng `setReserveFactor` (nếu cần thay đổi sau triển khai).

4 Tài liệu Liên quan và Trích dẫn

1. **OpenZeppelin Contracts Documentation:** Nguồn tham khảo cho các thư viện và tiêu chuẩn cơ sở (`ERC20`, `ERC721`, `AccessControl`, `SafeERC20`).
2. **Mô hình Lãi suất Aave (Aave Interest Rate Model):** Công thức tính lãi suất Kink Model được lấy cảm hứng từ các giao thức cho vay hàng đầu trong DeFi.
3. **ERP-20 & ERP-721:** Tiêu chuẩn kỹ thuật của Ethereum cho các loại token được sử dụng trong dự án.
4. **Mã nguồn Hợp đồng (Solidity Codebase):** Toàn bộ logic nghiệp vụ được định nghĩa trong các tệp `CycloneERC20Token.sol`, `CycloneERC721Token.sol`, và `SolomonTreasury.sol`.