PROJECT

Warp Protocol

CLIENT

Warp Finance

REVIEWERS

AkrAn

DATE

February 2022

Table of Contents

- Details
- Issues Summary
- Executive summary
- Scope
- Recommendations
  - Increase the number of tests
- Issues
  - acceptOwnerTransfer emits event with nil address
  - setDistributorStatus emitted event name is confusing and lacks complete data
  - acceptOwnerTransfer / acceptOwnership should include previous owner address in the emitted events
  - Ownership transfer process is inconsistently named across different contracts
  - Typo in RewardDistirbutorManagerStorageV1
  - onlyEOA modifier might not work in future Ethereum versions
- Artifacts
  - Surya
  - Files Description Table
  - Contracts Description Table
  - Legend
  - Coverage
  - Tests
- License

Details

- **Client** Warp Finance
- **Date** February 2022
- **Reviewers** AkrAn
- **Repository**: Warp Protocol
- **Commit hash** `36ed4ac6a4b2c500096423359e73a8eded107cda`
- **Technologies**
  - Solidity
  - TypeScript

Issues Summary

SEVERITY	OPEN	CLOSED
Informational	0	4
Minor	0	2
Medium	0	0
Major	0	0

Executive summary

This report represents the results of the engagement with **Warp Finance** to review **Warp Protocol**.

The review was conducted over the course of **2 weeks** from **January 31st to February 11th, 2022**. A total of **10 person-days** were spent reviewing the code.

Scope

The initial review focused on the Warp Protocol repository, identified by the commit hash `36ed4ac6a4b2c500096423359e73a8eded107cda`.

After the initial review, I updated the code to git hash `41b184094ef39939777063f906f1893f6037af8f` to pull in fixes made by the Warp Finance team.

I focused on manually reviewing the codebase, searching for security issues such as, but not limited to, re-entrancy problems, transaction ordering, block timestamp dependency, exception handling, call stack depth limitation, integer overflow/underflow, self-destructible contracts, unsecured balance, use of origin, costly gas patterns, architectural problems, code readability.

Includes:

- code/contracts/token/ERC20.sol
- code/contracts/token/ERC20Permit.sol
- code/contracts/token/EIP712.sol
- code/contracts/DebtToken.sol
- code/contracts/LendingPairFactory.sol
- code/contracts/interest/JumpRateModelV2.sol
- code/contracts/util/Initializable.sol
- code/contracts/util/ReentrancyGuard.sol
- code/contracts/oracle/PriceOracleAggregator.sol

- code/contracts/oracle/adapter/ChainlinkUSDAdapter.sol
- code/contracts/rewards/RewardDistributorManager.sol
- code/contracts/rewards/RewardDistributor.sol
- code/contracts/rewards/RewardDistributorFactory.sol
- code/contracts/VaultFactory.sol
- code/contracts/BorrowWrapperToken.sol
- code/contracts/LendingPair.sol
- code/contracts/math/Exponential.sol
- code/contracts/Vault.sol
- code/contracts/DataTypes.sol
- code/contracts/VaultBase.sol
- code/contracts/CollateralWrapperToken.sol
- code/contracts/WrapperToken.sol
- code/contracts/helper/LendingPairHelper.sol
- code/contracts/helper/FeeWithdrawal.sol

Excludes:

- code/contracts/upgradability/UUPSProxy.sol
- code/contracts/upgradability/UUPSProxiable.sol
- code/contracts/upgradability/UUPSUtils.sol

Recommendations

I identified a few possible general improvements that are not security issues during the review, which will bring value to the developers and the community reviewing and using the product.

Increase the number of tests

A good rule of thumb is to have 100% test coverage. This does not guarantee the lack of security problems, but it means that the desired functionality behaves as intended. The negative tests also bring a lot of value because not allowing some actions to happen is also part of the desired behavior.

Issues

`acceptOwnerTransfer` emits event with nil address

Status Fixed Severity Minor

Description

`commitOwnerTransfer` can be called by the *owner* of the contract to transfer ownership to a new address:

code/contracts/rewards/RewardDistributorManager.sol#L123-L127

```
    function commitOwnerTransfer(address _newOwner) external onlyOwner {
        require(_newOwner != address(0), "INVALID_NEW_OWNER");
        newOwner = _newOwner;
        emit TransferControl(_newOwner, block.timestamp);
    }
```

`acceptOwnerTransfer` can then be used by the `newOwner` address to accept the ownership transfer:

code/contracts/rewards/RewardDistributorManager.sol#L129-L134

```
    function acceptOwnerTransfer() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }
```

The issue however is that the `newOwner` variable is reset to nil address *before* the event is emitted.

Recommendation

Update `acceptOwnerTransfer` function to emit the `OwnershipAccepted` event before the `newOwner` state variable is reset to nil address;

Please note that the above-mentioned issue and the recommendation are applicable for similar functions present in `PriceOracleAggregator` and `Vault` contracts (see issue #6):

code/contracts/Vault.sol#L230-L239

```
    function acceptOwnership() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }

    /// @notice Transfer control from current owner address to another
    /// @param _newOwner The new team
    function transferToNewOwner(address _newOwner) external onlyOwner {
```

code/contracts/oracle/PriceOracleAggregator.sol#L101-L110

```
    function acceptOwnership() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }

    /// @notice Transfer control from current owner address to another
    /// @param _newOwner The new team
    function transferToNewOwner(address _newOwner) external onlyOwner {
```

`setDistributorStatus` emitted event name is confusing and lacks complete data

Status Fixed Severity Minor

Description

`setDistributorStatus` can be called by the owner of the contract to set the status of a rewards distributor:

code/contracts/rewards/RewardDistributorManager.sol#L57-L66

```
    /// @dev approves a distributor contract for a token
    /// @param _distributor The distributor contract address
    /// @param _approve the status of the distributor contract
    function setDistributorStatus(IRewardDistributor _distributor, bool _approve)
        external
        onlyOwner
    {
        approvedDistributors[_distributor] = _approve;
        emit ApprovedDistributor(_distributor, block.timestamp);
    }
```

The `ApprovedDistributor` event emitted is named in such a way that implies that the distributor has been approved. This is not the case since `_approve` is a *bool* and therefore can be set to `false` to represent that a `_distributor` has been disapproved by the contract owner.

Recommendation

Update the name of the event to closer reflect the action of setting the distributor status (something along the lines of `DistributorStatusUpdated` comes to mind). Add another parameter to the event to reflect the status value that is being set.

`acceptOwnerTransfer` / `acceptOwnership` should include previous owner address in the emitted events

Status Fixed Severity Informational

Description

`RewardDistributorManager`, `Vault` and `PriceOracleAggregator` contracts have a process of transferring the ownership of the contract to a new owner:

code/contracts/rewards/RewardDistributorManager.sol#L129-L134

```
    function acceptOwnerTransfer() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }
```

code/contracts/Vault.sol#L230-L239

```
    function acceptOwnership() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }

    /// @notice Transfer control from current owner address to another
    /// @param _newOwner The new team
    function transferToNewOwner(address _newOwner) external onlyOwner {
```

code/contracts/oracle/PriceOracleAggregator.sol#L101-L110

```
    function acceptOwnership() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }

    /// @notice Transfer control from current owner address to another
    /// @param _newOwner The new team
    function transferToNewOwner(address _newOwner) external onlyOwner {
```

Recommendation

Only the new owner data is included when the `OwnershipAccepted` event is emitted. However, I recommend including the previous owner address since this is directly

relevant to this operation and could be helpful to external systems.

Ownership transfer process is inconsistently named across different contracts

Status Fixed Severity Informational

Description

In `RewardDistributorManager` the methods for transferring and accepting ownership of the contract are named `commitOwnerTransfer`, `acceptOwnerTransfer`:

code/contracts/rewards/RewardDistributorManager.sol#L123-L129

```
    function commitOwnerTransfer(address _newOwner) external onlyOwner {
        require(_newOwner != address(0), "INVALID_NEW_OWNER");
        newOwner = _newOwner;
        emit TransferControl(_newOwner, block.timestamp);
    }

    function acceptOwnerTransfer() external {
```

In `Vault` and `PriceOracleAggregator` they're named `transferToNewOwner` and `acceptOwnership`:

code/contracts/Vault.sol#L230-L239

```
    function acceptOwnership() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }

    /// @notice Transfer control from current owner address to another
    /// @param _newOwner The new team
    function transferToNewOwner(address _newOwner) external onlyOwner {
```

code/contracts/oracle/PriceOracleAggregator.sol#L101-L110

```
    function acceptOwnership() external {
        require(msg.sender == newOwner, "invalid owner");
        owner = newOwner;
        newOwner = address(0);
        emit OwnershipAccepted(newOwner, block.timestamp);
    }

    /// @notice Transfer control from current owner address to another
    /// @param _newOwner The new team
```

```
    /// @param _newOwner The new team
    function transferToNewOwner(address _newOwner) external onlyOwner {
```

Recommendation

For consistency purposes and developer experience, it is worth naming these methods the same.

Typo in `RewardDistirbutorManagerStorageV1`

Status Fixed Severity Informational

Description

`Distirbutor` -> `Distributor`

code/contracts/rewards/RewardDistributorManager.sol#L10

```
abstract contract RewardDistirbutorManagerStorageV1 is UUPSProxiable, IRewardDistributorManager {
```

---

onlyEOA modifier might not work in future Ethereum versions

Status Acknowledged Severity Informational

Description

`onlyEOA` modifier is used to ensure that the caller is an Externally Owned Account or a pre-approved account that is allowed to perform certain actions.

> In general, there are two types of accounts: externally owned accounts, controlled by private keys, and contract accounts, controlled by their contract code. An externally owned account has no code, and one can send messages from an externally owned account by creating and signing a transaction; in a contract account, every time the contract account receives a message its code activates, allowing it to read and write to internal storage and send other messages or create contracts in turn.

code/contracts/helper/FeeWithdrawal.sol#L45-L49

```
    modifier onlyEOA() {
        // Try to make flash-loan exploit harder to do by only allowing externally-owned addresses.
        require(msg.sender == tx.origin, "MUST_BE_EOA");
        _;
    }
```

The EIP-3074, although still in the works, might be implemented in a future version of Solidity which will make the `msg.sender == tx.origin` type of re-entrancy guards inappropriate:

> Allowing authorized to equal tx.origin has the possibility to:
>
> - Break atomic sandwich protections which rely on tx.origin;
> - Break re-entrancy guards of the style require(tx.origin == msg.sender).

Edit: Marking this issue as Acknowledged. While there is a small risk that EIP-3074 will be released, when and if this happens the team can upgrade the contract to use a different method to ensure that this is an EOA account.

References

Ethereum Accounts

Open Zeppelin - Provide function(s) that verify an address is an EOA

Bypassing Smart Contract Timelocks

Artifacts

Surya

Sūrya is a utility tool for smart contract systems. It provides a number of visual outputs and information about the structure of smart contracts. It also supports querying the function call graph in multiple ways to aid in the manual inspection and control flow analysis of contracts.

Files Description Table

File Name	SHA-1 Ha
code/contracts/token/ERC20.sol	5be2ea83eed48ce1642c6a0
code/contracts/token/ERC20Permit.sol	a25e03bcf8e303f6aca3452c
code/contracts/token/EIP712.sol	cdae803db537ea0dd02a096
code/contracts/DebtToken.sol	c8d6497af63c071d6db4419
code/contracts/LendingPairFactory.sol	89e2e629303ca547e4fddd6
code/contracts/interest/JumpRateModelV2.sol	ebc2c83444b003ba645d604
code/contracts/util/Initializable.sol	574648268f1a14267082146
code/contracts/util/ReentrancyGuard.sol	a21e799fd83ab0c208c71ec
code/contracts/oracle/PriceOracleAggregator.sol	67d17cbd777d0b6a4a811fb
code/contracts/oracle/adapter/ChainlinkUSDAdapter.sol	dff36c2efa2cd5cbb6745749
code/contracts/rewards/RewardDistributorManager.sol	35c4e38d7cc766ede6aec9b
code/contracts/rewards/RewardDistributor.sol	72b440652be851967fba0c6
code/contracts/rewards/RewardDistributorFactory.sol	d0f0d8d238d07c90074e08d
code/contracts/VaultFactory.sol	e7c2a8aafcd398e463813320
code/contracts/BorrowWrapperToken.sol	c67725f5c381c57cbb93214
code/contracts/LendingPair.sol	95be5b5671d70db3476626f
code/contracts/math/Exponential.sol	8006e0f173da2cdd20474d5
code/contracts/Vault.sol	e0c8b36f5ea308f775e47d52
code/contracts/DataTypes.sol	73db8a6a7cdb3f6acd0111d
code/contracts/VaultBase.sol	0ba23476e56d6defc8445f50
code/contracts/CollateralWrapperToken.sol	4641123eb90183eb9eef186
code/contracts/WrapperToken.sol	400c23d639dec132fcb10a9
code/contracts/helper/LendingPairHelper.sol	45a1ea4b3d36a2125a33ebc
code/contracts/helper/FeeWithdrawal.sol	f4ea71d94670a7d084e5779

Contracts Description Table

Contract	Type	
└	**Function Name**	
ERC20	Implementation	
└	initializeERC20	
└	balanceOf	
└	totalSupply	
└	transfer	
└	allowance	
└	approve	
└	transferFrom	
└	increaseAllowance	
└	decreaseAllowance	
└	_mint	
└	_approve	
└	_transfer	
└	_burn	
ERC20Permit	Implementation	
└	initializeERC20Permit	
└	permit	
└	nonces	
└	DOMAIN_SEPARATOR	
EIP712	Implementation	
└	initializeEIP712	
└	_domainSeparatorV4	
└	_buildDomainSeparator	
└	_hashTypedDataV4	
└	_getChainId	
DebtToken	Implementation	
└	initialize	

Contract	Type	
└	principal	
└	balanceOf	
└	mint	
└	_mint	
└	owner	
└	burn	
└	borrowAllowance	
└	delegateBorrow	
└	_delegateBorrowInternal	
└	delegateBorrowWithSignedMessage	
└	_decreaseBorrowAllowance	
└	increaseTotalDebt	
└	transfer	
└	approve	
└	allowance	
└	transferFrom	
└	increaseAllowance	
└	decreaseAllowance	
└	permit	
LendingPairFactory	Implementation	
└		
└	pause	
└	unpause	
└	updatePairImpl	
└	updateCollateralWrapperImpl	
└	updateDebtTokenImpl	
└	updateBorrowAssetWrapperImpl	
└	updateRewardManager	

Contract	Type	
└	createIR	
└	disableIR	
└	createLendingPairWithProxy	
└	initWrapperTokensWithProxy	
└	initializeWrapperTokens	
JumpRateModelV2	Implementation	
└		
└	updateJumpRateModel	
└	utilizationRate	
└	getBorrowRateInternal	
└	getSupplyRate	
└	updateJumpRateModelInternal	
└	getBorrowRate	
Initializable	Implementation	
└	_isConstructor	
ReentrancyGuard	Implementation	
└	__init_ReentrancyGuard	
PriceOracleAggregator	Implementation	
└		
└	setOracleForAsset	
└	removeOracleForAsset	
└	addStable	
└	getPriceInUSD	
└	getPriceInUSDMultiple	
└	acceptOwnership	
└	transferToNewOwner	
ChainlinkUSDAdapter	Implementation	

Contract	Type	
└		
└	latestAnswer	
RewardDistirbutorManagerStorageV1	Implementation	
RewardDistributorManager	Implementation	Rev
└	initialize	
└	accumulateRewards	
└	setDistributorStatus	
└	activateReward	
└	removeReward	
└	findRewardDistributor	
└	commitOwnerTransfer	
└	acceptOwnerTransfer	
└	proxiableUUID	
└	updateCode	
RewardDistributorStorageV1	Implementation	
RewardDistributor	Implementation	
└		
└	accumulateReward	
└	initialize	
└	add	
└	activatePendingRewards	
└	set	
└	getMultiplier	
└	pendingRewardToken	
└	calculatePoolReward	
└	massUpdatePools	
└	updatePool	

Contract	Type	
└	withdraw	
└	updateEndTimestamp	
└	withdrawUnclaimedRewards	
└	safeTokenTransfer	
└	getTokenPoolID	
└	calculatePendingReward	
└	updatePoolAndDistributeUserReward	
└	createPool	
RewardDistributorFactory	Implementation	
└		
└	setDistributorImplementation	
└	createRewardDistributor	
VaultFactory	Implementation	
└		
└	updateVaultLogic	
└	createUpgradableVault	
BorrowWrapperToken	Implementation	
LendingPair	Implementation	
└		
└	initialize	
└	pause	
└	unpause	
└	depositCollateral	
└	depositBorrowAsset	
└	borrow	
└	repay	
└	redeem	

Contract	Type	
└	warp	
└	select	
└	calculateLiquidationFee	
└	exchangeRateCurrent	
└	getCashPrior	
└	totalBorrows	
└	accrueInterest	
└	borrowBalanceCurrent	
└	borrowBalancePrior	
└	withdrawFees	
└	withdrawCollateral	
└	collateralOfAccount	
└	getMaxWithdrawAllowed	
└	getTotalAvailableCollateralValueInUSD	
└	getTotalAvailableCollateralValue	
└	getPriceOfCollateral	
└	getPriceOfBorrowAsset	
└	getPriceOfToken	
└	calcBorrowLimit	
└	calcCollateralRequired	
└	getBorrowLimitInUSD	
└	getBorrowLimit	
└	liquidate	
└	_repayLiquidatingLoan	
└	_liquidate	
└	getBlockNumber	
└	borrowRatePerBlock	
└	supplyRatePerBlock	
└	normalize	

Contract	Type	
└	denormalize	
Exponential	Implementation	
└	getExp	
└	mulScalar	
└	mulScalarTruncate	
└	mulScalarTruncateAddUInt	
└	divScalar	
└	divScalarByExp	
└	divScalarByExpTruncate	
└	truncate	
Vault	Implementation	
└		
└	initialize	
└	proxiableUUID	
└	updateCode	
└	allowContract	
└	approveContract	
└	pause	
└	unpause	
└	deposit	
└	withdraw	
└	transfer	
└	acceptOwnership	
└	transferToNewOwner	

Contract	Type	
└	_transfer	
└	maxFlashLoan	
└	flashFee	
└	flashLoan	
└	updateFlashloanRate	
└	toShare	
└	toUnderlying	
└	rescueFunds	
DataTypes	Library	
└	validBorrowAssetConfig	
VaultStorageV1	Implementation	Re
VaultBase	Implementation	
└		
└	_buildDomainSeparator	
└	_domainSeparatorV4	
└	_getChainId	
CollateralWrapperToken	Implementation	
└	_transfer	
WrapperTokenBase	Implementation	
└	_rewardHook	
WrapperToken	Implementation	
└	initialize	
└	mint	
└	burn	
└	owner	

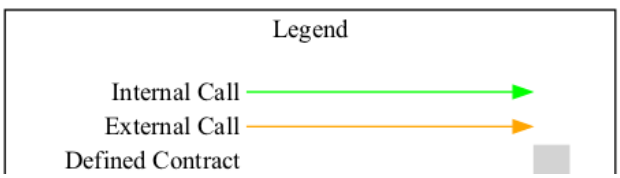
Contract	Type	
└	_transfer	
LendingPairHelper	Implementation	
└		
└	viewBorrowedValue	
└	viewBorrowedValueInUSD	
└	viewBorrowLimit	
FeeWithdrawal	Implementation	
└		
└	withdrawFees	
└	swapFees	
└	transferToReceiver	
└	updateAdmin	
└	rescueFunds	
└	getPath	
└	_convertToWarp	
└	initialize	
└	proxiableUUID	
└	updateCode	

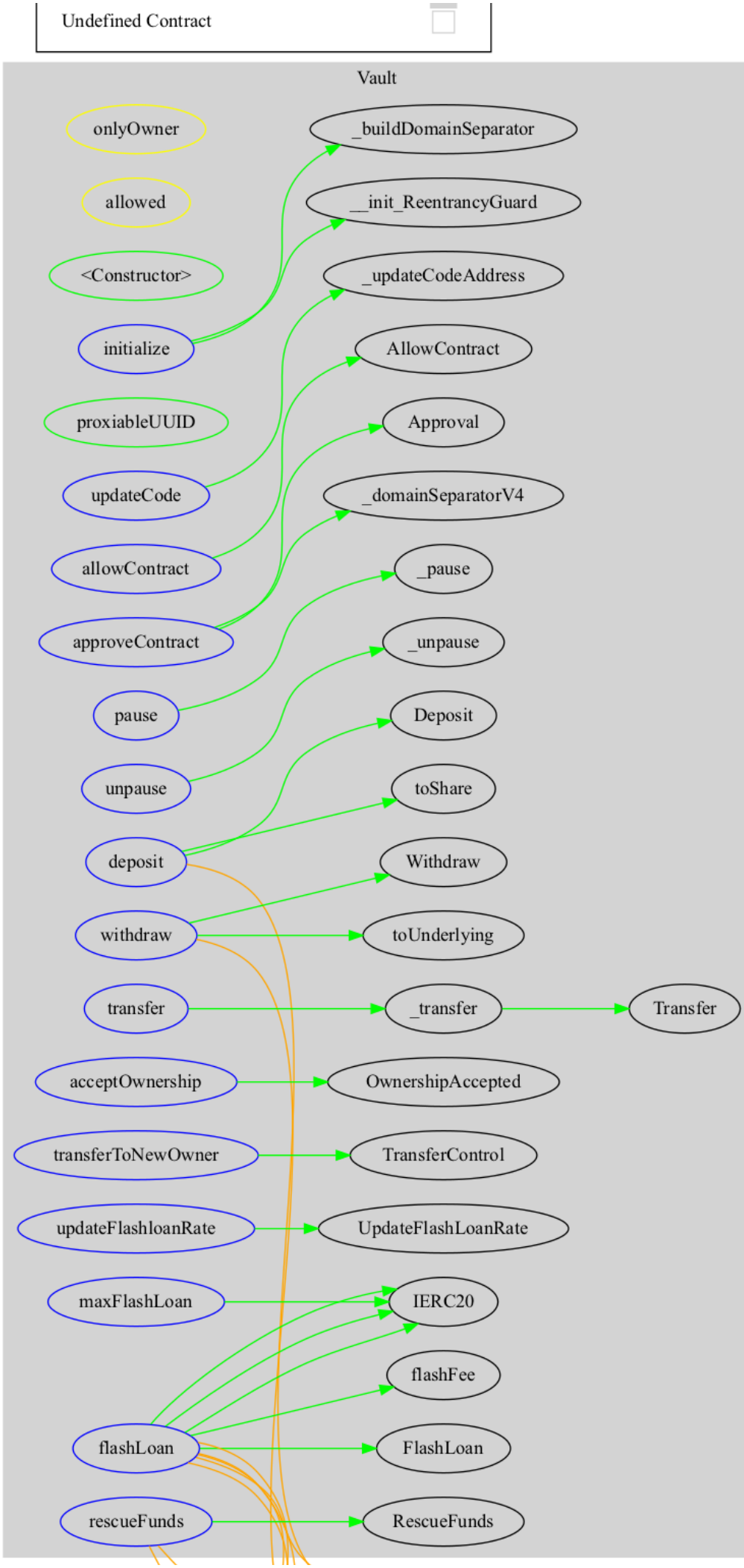
Legend

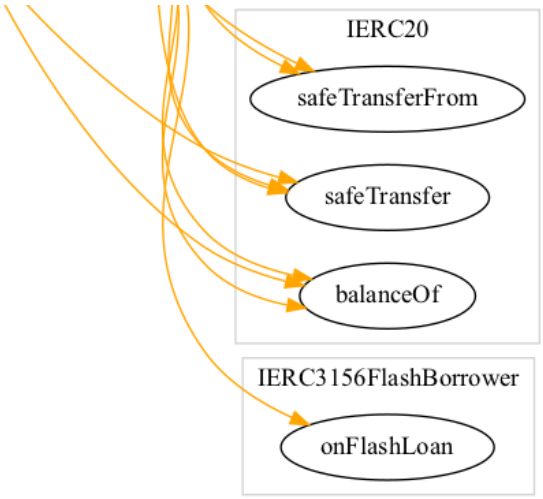
Symbol	Meaning
	Function can modify state
	Function is payable

Graphs

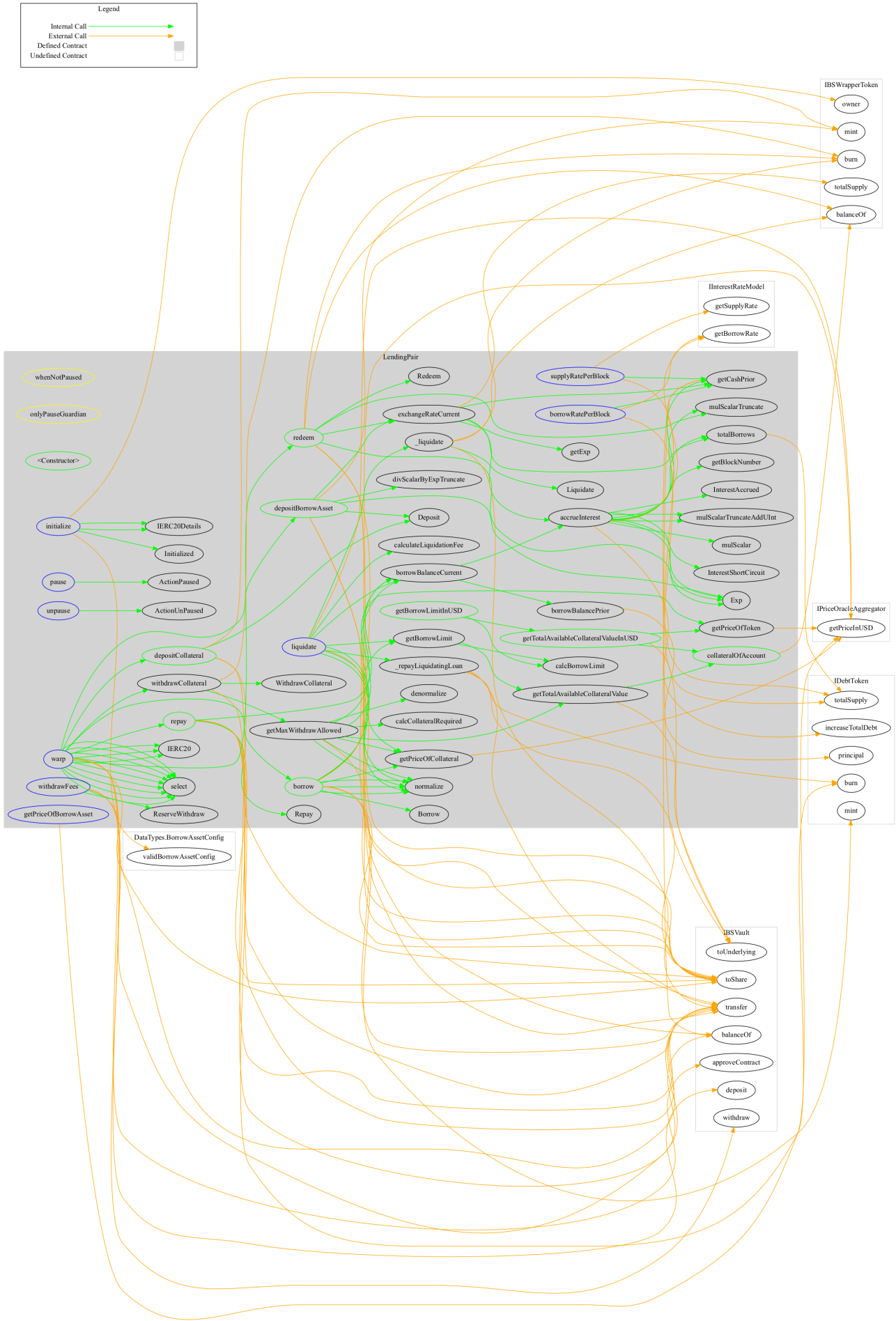
Vault



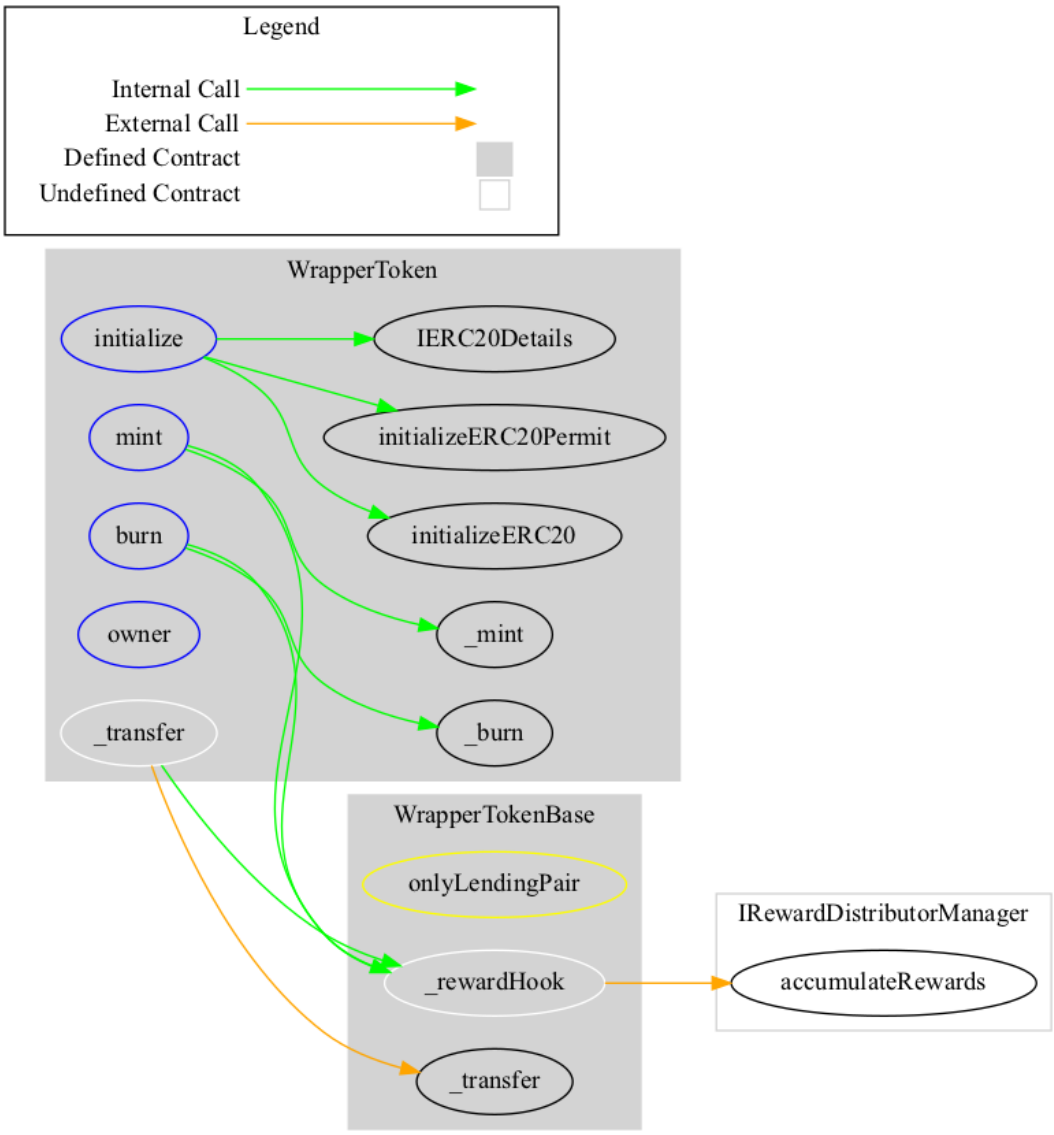

Undefined Contract
Vault
onlyOwner
allowed
<Constructor>
initialize
proxiableUUID
updateCode
allowContract
approveContract
pause
unpause
deposit
withdraw
transfer
acceptOwnership
transferToNewOwner
updateFlashloanRate
maxFlashLoan
flashLoan
rescueFunds
_buildDomainSeparator
__init_ReentrancyGuard
_updateCodeAddress
AllowContract
Approval
_domainSeparatorV4
_pause
_unpause
Deposit
toShare
Withdraw
toUnderlying
_transfer
Transfer
OwnershipAccepted
TransferControl
UpdateFlashLoanRate
IERC20
flashFee
FlashLoan
RescueFunds



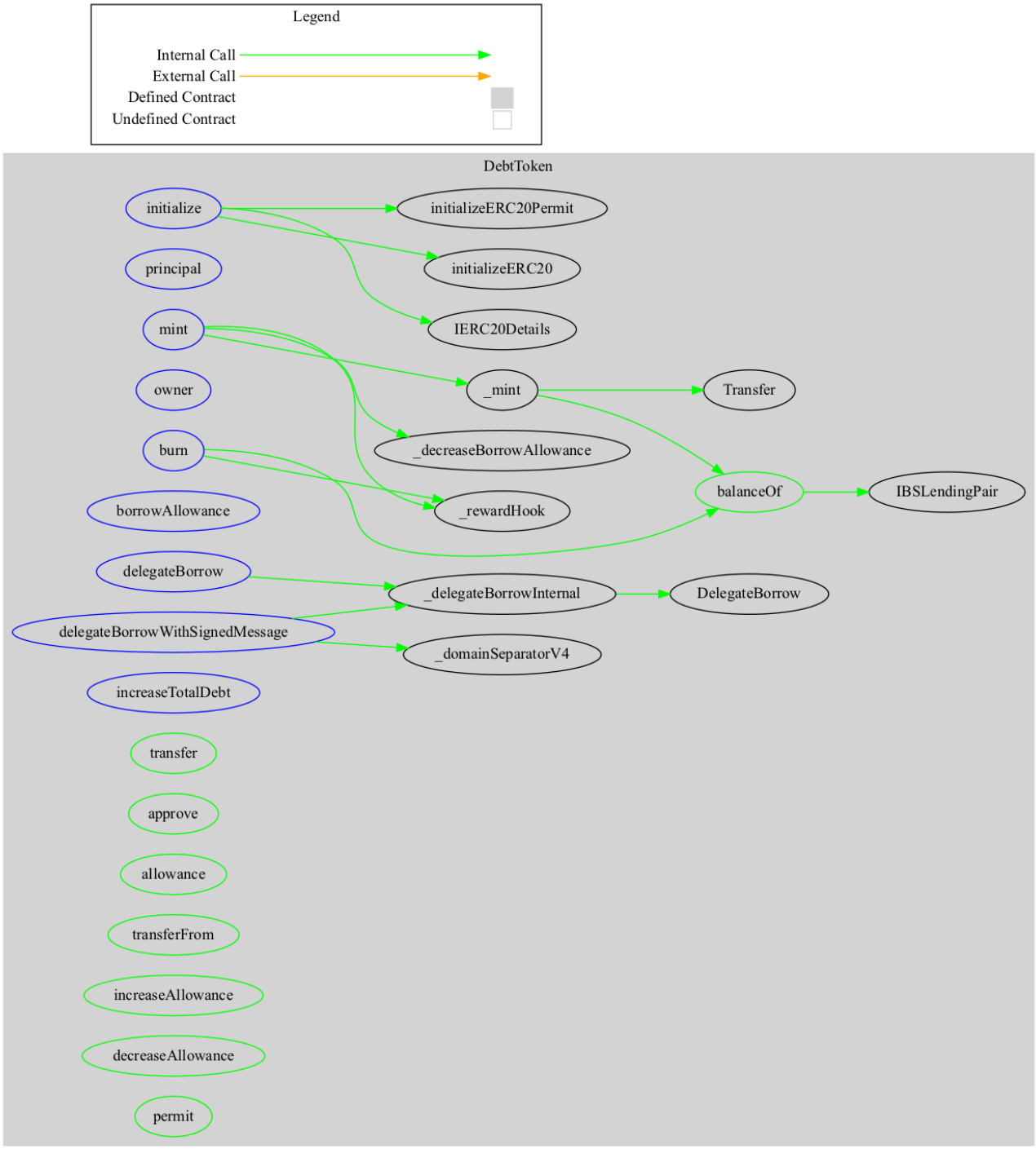
LendingPair



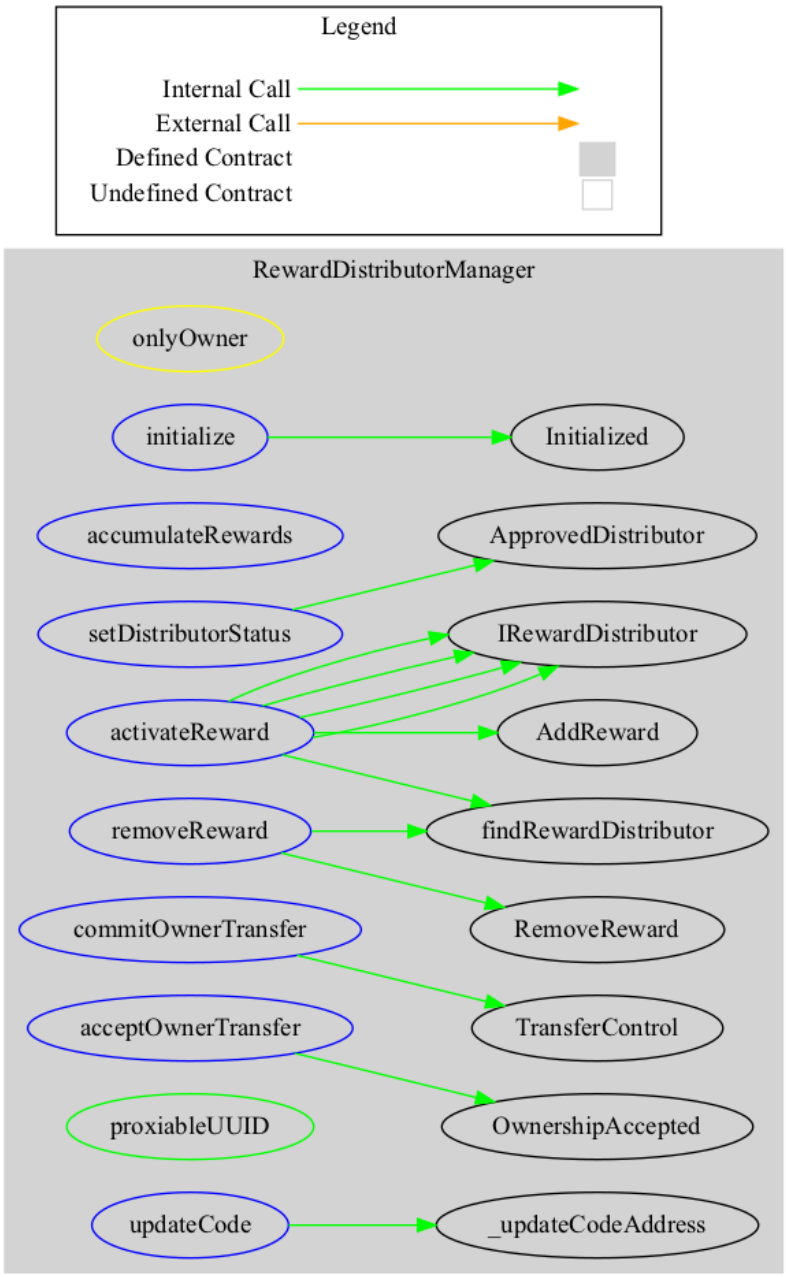
WrapperToken


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
WrapperToken
initialize
IERC20Details
initializeERC20Permit
mint
initializeERC20
burn
_mint
owner
_burn
_transfer
WrapperTokenBase
onlyLendingPair
IRewardDistributorManager
_rewardHook
accumulateRewards
_transfer

DebtToken



RewardDistributorManager



Describe

```
+  Vault (VaultBase)
   - [Pub] <Constructor> #
      - modifiers: VaultBase
   - [Ext] initialize #
      - modifiers: initializer
   - [Pub] proxiableUUID
   - [Ext] updateCode #
      - modifiers: onlyOwner
   - [Ext] allowContract #
      - modifiers: onlyOwner
   - [Ext] approveContract #
   - [Ext] pause #
      - modifiers: onlyOwner
   - [Ext] unpause #
      - modifiers: onlyOwner
   - [Ext] deposit #
```

```
      - modifiers: whenNotPaused,allowed,nonReentrant
   - [Ext] withdraw #
      - modifiers: whenNotPaused,allowed,nonReentrant
   - [Ext] transfer #
      - modifiers: whenNotPaused,allowed
   - [Ext] acceptOwnership #
   - [Ext] transferToNewOwner #
      - modifiers: onlyOwner
   - [Int] _transfer #
   - [Ext] maxFlashLoan
   - [Pub] flashFee
   - [Ext] flashLoan #
      - modifiers: nonReentrant
   - [Ext] updateFlashloanRate #
      - modifiers: onlyOwner
   - [Pub] toShare

   - [Pub] toUnderlying
   - [Ext] rescueFunds #
      - modifiers: nonReentrant,onlyOwner

+  RewardDistirbutorManagerStorageV1 (UUPSProxiable, IRewardDistributorManager)

+  RewardDistributorManager (RewardDistirbutorManagerStorageV1)
   - [Ext] initialize #
      - modifiers: initializer
   - [Ext] accumulateRewards #
   - [Ext] setDistributorStatus #
      - modifiers: onlyOwner
   - [Ext] activateReward #
   - [Ext] removeReward #
      - modifiers: onlyOwner
   - [Int] findRewardDistributor
   - [Ext] commitOwnerTransfer #
      - modifiers: onlyOwner
   - [Ext] acceptOwnerTransfer #
   - [Pub] proxiableUUID
   - [Ext] updateCode #
      - modifiers: onlyOwner

+  LendingPair (IBSLendingPair, Exponential, Initializable)
   - [Pub] <Constructor> #
   - [Ext] initialize #
      - modifiers: initializer
   - [Ext] pause #
      - modifiers: onlyPauseGuardian
   - [Ext] unpause #
      - modifiers: onlyPauseGuardian
   - [Pub] depositCollateral #
      - modifiers: whenNotPaused
   - [Pub] depositBorrowAsset #
      - modifiers: whenNotPaused
```

```
    - modifiers: whenNotPaused
  - [Pub] borrow #
    - modifiers: whenNotPaused
  - [Pub] repay #
  - [Pub] redeem #
  - [Ext] warp #
  - [Int] select
  - [Pub] calculateLiquidationFee
  - [Pub] exchangeRateCurrent #
  - [Pub] getCashPrior
  - [Pub] totalBorrows
  - [Pub] accrueInterest #
  - [Pub] borrowBalanceCurrent #
  - [Pub] borrowBalancePrior
  - [Ext] withdrawFees #
  - [Pub] withdrawCollateral #
  - [Pub] collateralOfAccount

  - [Pub] getMaxWithdrawAllowed #
  - [Pub] getTotalAvailableCollateralValueInUSD
  - [Pub] getTotalAvailableCollateralValue
  - [Pub] getPriceOfCollateral
  - [Ext] getPriceOfBorrowAsset
  - [Pub] getPriceOfToken
  - [Pub] calcBorrowLimit
  - [Pub] calcCollateralRequired
  - [Pub] getBorrowLimitInUSD
  - [Pub] getBorrowLimit
  - [Ext] liquidate #
  - [Int] _repayLiquidatingLoan #
  - [Int] _liquidate #
  - [Int] getBlockNumber
  - [Ext] borrowRatePerBlock
  - [Ext] supplyRatePerBlock
  - [Int] normalize
  - [Int] denormalize

+ WrapperTokenBase (ERC20Permit, Initializable)
  - [Int] _rewardHook #

+ WrapperToken (IBSWrapperToken, WrapperTokenBase)
  - [Ext] initialize #
    - modifiers: initializer
  - [Ext] mint #
    - modifiers: onlyLendingPair
  - [Ext] burn #
    - modifiers: onlyLendingPair
  - [Ext] owner
  - [Int] _transfer #

+ DebtToken (IDebtToken, WrapperTokenBase)
  - [Ext] initialize #
    - modifiers: initializer
```

```
    - [Ext] principal
    - [Pub] balanceOf
    - [Ext] mint #
       - modifiers: onlyLendingPair
    - [Int] _mint #
    - [Ext] owner
    - [Ext] burn #
       - modifiers: onlyLendingPair
    - [Ext] borrowAllowance
    - [Ext] delegateBorrow #
    - [Int] _delegateBorrowInternal #
    - [Ext] delegateBorrowWithSignedMessage #
    - [Int] _decreaseBorrowAllowance #
    - [Ext] increaseTotalDebt #
       - modifiers: onlyLendingPair
    - [Pub] transfer

    - [Pub] approve #
    - [Pub] allowance
    - [Pub] transferFrom #
    - [Pub] increaseAllowance #
    - [Pub] decreaseAllowance #
    - [Pub] permit #


 ($) = payable function
 # = non-constant function
```

Coverage

NB: some tests failed under coverage report but passed during normal tests run (see below).

```
❯ npm run coverage

> blacksmith@1.0.0 coverage
> npx hardhat coverage


Version
=======
> solidity-coverage: v0.7.16

Instrumenting for coverage...
=============================

> BorrowWrapperToken.sol
> CollateralWrapperToken.sol
> DataTypes.sol
```

```
> DebtToken.sol
> helper/FeeWithdrawal.sol
> helper/LendingPairHelper.sol
> interest/JumpRateModelV2.sol
> interfaces/IBSLendingPair.sol
> interfaces/IBSVault.sol
> interfaces/IBSWrapperToken.sol
> interfaces/IChainlinkV3Aggregator.sol
> interfaces/IDebtToken.sol
> interfaces/IERC3156FlashBorrower.sol
> interfaces/IERC3156FlashLender.sol
> interfaces/IInterestRateModel.sol
> interfaces/IKeeperOracle.sol
> interfaces/IOracle.sol
> interfaces/IPriceOracleAggregator.sol
> interfaces/IRewardDistributor.sol
> interfaces/IRewardDistributorManager.sol
> LendingPair.sol
> LendingPairFactory.sol
> math/Exponential.sol
> oracle/adapter/ChainlinkUSDAdapter.sol
> oracle/PriceOracleAggregator.sol
> rewards/RewardDistributor.sol
> rewards/RewardDistributorFactory.sol
> rewards/RewardDistributorManager.sol
> test/MockChainlinkUSDAdapter.sol
> test/MockFeeWIthdrawUniswapRouter.sol
> test/MockFlashBorrower.sol
> test/MockLendingPair.sol
> test/MockPriceOracle.sol
> test/MockRewardDistributorManager.sol
> test/MockToken.sol
> test/MockUniswapV2Router02.sol
> test/MockVault.sol
> test/MockVaultUser.sol
> test/VaultStorageLayoutTester.sol
> token/EIP712.sol
> token/ERC20.sol
> token/ERC20Permit.sol
> token/IERC20Details.sol
> upgradability/UUPSProxiable.sol
> upgradability/UUPSProxy.sol
> upgradability/UUPSUtils.sol
> util/Initializable.sol
> util/ReentrancyGuard.sol
> Vault.sol
> VaultBase.sol
> VaultFactory.sol
> WrapperToken.sol

Compilation:
```

```
============

Compiling 67 files with 0.8.1
Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment
--> @uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol


Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment
--> @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol


Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment
--> @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol


Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.
  --> contracts/test/MockFeeWIthdrawUniswapRouter.sol:10:9:
   |
10 |         uint256 amountIn,
   |         ^^^^^^^^^^^^^^^^


Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.
  --> contracts/test/MockFeeWIthdrawUniswapRouter.sol:11:9:
   |
11 |         uint256 amountOutMin,
   |         ^^^^^^^^^^^^^^^^^^^^


Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.
  --> contracts/test/MockFeeWIthdrawUniswapRouter.sol:12:9:
   |
12 |         address[] calldata path,
   |         ^^^^^^^^^^^^^^^^^^^^^^^


Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.
  --> contracts/test/MockFeeWIthdrawUniswapRouter.sol:13:9:
   |
13 |         address to,
   |         ^^^^^^^^^^


Warning: Unused function parameter. Remove or comment out the variable name to silence this warning.
  --> contracts/test/MockFeeWIthdrawUniswapRouter.sol:14:9:
   |
14 |         uint256 deadline
   |         ^^^^^^^^^^^^^^^^


Warning: Function state mutability can be restricted to pure
```

```
 --> contracts/test/MockFeeWIthdrawUniswapRouter.sol:9:5:
  |
9 |     function swapExactTokensForTokens(
  |     ^ (Relevant source part starts here and spans across multiple lines).


Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may
  --> contracts/LendingPair.sol:27:1:
   |
27 | contract LendingPair is IBSLendingPair, Exponential, Initializable {
   | ^ (Relevant source part starts here and spans across multiple lines).


Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may
  --> contracts/LendingPairFactory.sol:18:1:
   |
18 | contract LendingPairFactory is Pausable {
   | ^ (Relevant source part starts here and spans across multiple lines).


Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may
  --> contracts/Vault.sol:24:1:
   |
24 | contract Vault is VaultBase {
   | ^ (Relevant source part starts here and spans across multiple lines).


Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may
  --> contracts/rewards/RewardDistributor.sol:68:1:
   |
68 | contract RewardDistributor is RewardDistributorStorageV1 {
   | ^ (Relevant source part starts here and spans across multiple lines).


Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may
 --> contracts/test/MockVault.sol:9:1:
  |
9 | contract MockVault is Vault {
  | ^ (Relevant source part starts here and spans across multiple lines).


Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may
 --> contracts/test/VaultStorageLayoutTester.sol:8:1:
  |
8 | contract VaultStorageLayoutTester is Vault {
  | ^ (Relevant source part starts here and spans across multiple lines).


Compilation finished successfully
Creating Typechain artifacts in directory types/ for target ethers-v5
Successfully generated Typechain artifacts!
```

```
The Hardhat Network tracing engine could not be initialized. Run Hardhat with --verbose to learn more.

Network Info
============
> HardhatEVM: v2.5.0
> network:    hardhat

Creating Typechain artifacts in directory types/ for target ethers-v5
Successfully generated Typechain artifacts!


  Borrow Delegation
    ✓ borrow delegation (2174ms)

  DebtToken
    ✓ underlying
    ✓ mint - fails if not owner
    ✓ burn - fails if not owner
    ✓ transfer
    ✓ approve
    ✓ allowance
    ✓ transferFrom
    ✓ increaseAllowance
    ✓ decreaseAllowance
    ✓ increaseTotalDebt - fails if not owner
    ✓ increaseTotalDebt (38ms)

  FeeWithdrawal
    ✓ proxiableUUID & updateCode (190ms)
    scenarios
      ✓ withdrawFees (95ms)
      ✓ swapFees (169ms)
      ✓ transferToReceiver (148ms)
      ✓ rescueFunds (117ms)

  LendingPairFactory
    ✓ updatePairImpl (75ms)
    ✓ updateCollateralWrapperImpl (40ms)
    ✓ updateDebtTokenImpl (46ms)
    ✓ updateBorrowAssetWrapperImpl (41ms)
    ✓ pause
    ✓ unpause
    ✓ disableIR (139ms)
    ✓ createIR (40ms)
    ✓ createLendingPairWithProxy (229ms)

  LendingPair - Warp
    ✓ warp - actions (473ms)
    ✓ warp - pass parameters (949ms)

  LendingPair
```

```
    ✓ initialize (176ms)
    ✓ depositBorrowAsset - fails without enough vault balance (357ms)
    ✓ depositBorrowAsset (622ms)
    ✓ depositCollateral (574ms)
    ✓ borrow - fails when you try to borrow more than allowed (862ms)
    ✓ borrow - fails when you try to borrow more than cash available in vault (501ms)
    ✓ borrow (880ms)
    ✓ collateral transfer - can not transfer more than allowed that puts loan at risk (958ms)
    ✓ collateral transferFrom - can not transferFrom more than allowed that puts loan at risk (1433ms)
    ✓ withdrawCollateral (1585ms)
    ✓ repay - fails if you are trying to pay more than owed (474ms)
    ✓ repay (1034ms)
    ✓ redeem - fails if you are trying to redeem more than your account (853ms)
    ✓ redeem (727ms)
    ✓ liquidate - cannot liquidate self (301ms)
    ✓ liquidate & withdrawFees - correctly (1576ms)
    ✓ withdrawFees - fails without enough balance (293ms)
    ✓ pause (518ms)
    ✓ unpause (1012ms)

  PriceOracleAggregator
    ✓ correct team address
    ✓ updateOracleForAsset - fails for non admin
    ✓ updateOracleForAsset - fails for invalid oracle adddress
    ✓ getPriceInUSD - fails for non existent oracle
    ✓ setOracleForAsset

  RewardDistributorFactory
    ✓ setDistributorImplementation
    ✓ createRewardDistributor (7036ms)

  RewardDistributorManager
    ✓ initialize (57ms)
    ✓ setDistributorStatus
    ✓ activateReward (81ms)
    ✓ removeReward (183ms)
    ✓ acceptOwnerTransfer & transferOwnership (44ms)
    ✓ proxiableUUID & updateCode (153ms)

  RewardDistributor
========= NOTICE =========
Request-Rate Exceeded  (this message will not be repeated)

The default API keys for each service are provided as a highly-throttled,
community resource for low-traffic projects and early prototyping.

While your application will continue to function, we highly recommended
signing up for your own API keys to improve performance, increase your
request rate/limit and enable other perks, such as metrics and advanced APIs.

For more details: https://docs.ethers.io/api-keys/
```

```
==========================
    ✓ initialize (8577ms)
    ✓ add & set (7470ms)
    ✓ should not accumulate reward if current time < startTimestamp (9879ms)
    1) reward calculation
    ✓ withdrawUnclaimedRewards & reward calculation - should allocate previous pending rewards (7222ms)
    ✓ withdraw calculates the reward and disburses it (4004ms)

  scenarios
    ✓ lending pair with different decimal places borrow & collateral asset (1292ms)
    ✓ multiple users

  UUPSProxy
    initializeProxy
      ✓ initializeProxy - fails when address is 0
      ✓ initializeProxy - fails when already initialized

  Vault Factory
0x3Aa5ebB10DC797CAC828524e59A333d0A371443c
    ✓ updateVaultLogic
    ✓ createUpgradableVault (72ms)

  Vault
    ✓ initialize fails with 0 team
    ✓ initialize - correctly (101ms)
    ✓ version
    ✓ name
    ✓ proxiableUUID
    ✓ pause & unpause - fails incorrect user (41ms)
    ✓ pause (90ms)
    ✓ allowContract (47ms)
    ✓ approveContract - contract (136ms)
    ✓ approveContract - fails invalid to / fails with wrong signature  (70ms)
    ✓ approveContract (346ms)
    ✓ deposit fails with invalid `to` address (41ms)
    ✓ deposit - correctly with correct user balance (121ms)
    ✓ deposit fails with incorrect approve deposit (40ms)
    ✓ transfer - correctly & fails with invalid `to` address (231ms)
    ✓ maxFlashLoan
    ✓ flashFee (40ms)
    ✓ flashLoan - correctly (145ms)
    ✓ withdraw fails with invalid `to` address (74ms)
    ✓ user cannot withdraw more than balance (73ms)
    ✓ withdraw - correctly (119ms)
    ✓ withdraw - fails if user tries to withdraw past minimum share balance (86ms)
    ✓ updateFlashloanRate (50ms)
    ✓ transferOwnership & acceptOwnership (85ms)
    ✓ toShare/toUnderlying - convert to appropriate shares & underlying (433ms)
    ✓ rescueFunds (376ms)

  Vault - Upgradable Layout
```

```
0x322813Fd9A801c5507c9de605d63CEA4f2CE6c44
    ✓ updateCode (198ms)
    ✓ test storage layout (56ms)

  BorrowToken/CollateralToken
    ✓ underlying (199ms)
    ✓ only owner can mint (197ms)
    ✓ only owner can burn (197ms)
    ✓ permit (233ms)
    ✓ transfer/transferFrom (799ms)
    ✓ increaseAllowance/decreaseAllowance (67ms)


  105 passing (3m)
  1 failing

  1) RewardDistributor
       reward calculation:
     AssertionError: Expected event "Withdraw" to be emitted, but it wasn't
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	97.06	73.45	93.91	97.06	
BorrowWrapperToken.sol	100	100	100	100	
CollateralWrapperToken.sol	100	100	100	100	
DataTypes.sol	100	50	100	100	
DebtToken.sol	90.91	55.56	85	90.91	102,113,209,221
LendingPair.sol	97.92	81.82	92.68	97.85	... 725,752,753
LendingPairFactory.sol	94.74	63.89	92.86	94.83	99,100,101
Vault.sol	98.86	82	100	98.89	347
VaultBase.sol	85.71	50	100	88.89	97
VaultFactory.sol	100	62.5	100	100	
WrapperToken.sol	100	66.67	100	100	
contracts/helper/	81.94	50	88.24	82.43	
FeeWithdrawal.sol	87.5	50	92.31	87.93	... 146,147,148
LendingPairHelper.sol	62.5	100	75	62.5	... 39,40,41,42
contracts/interest/	74.36	36.67	85.71	74.36	
JumpRateModelV2.sol	74.36	36.67	85.71	74.36	... 135,136,192
contracts/interfaces/	100	100	100	100	
IBSLendingPair.sol	100	100	100	100	
IBSVault.sol	100	100	100	100	
IBSWrapperToken.sol	100	100	100	100	
IChainlinkV3Aggregator.sol	100	100	100	100	
IDebtToken.sol	100	100	100	100	
IERC3156FlashBorrower.sol	100	100	100	100	
IERC3156FlashLender.sol	100	100	100	100	
IInterestRateModel.sol	100	100	100	100	

```
  IKeeperOracle.sol                |      100 |      100 |      100 |      100 |                |
  IOracle.sol                      |      100 |      100 |      100 |      100 |                |
  IPriceOracleAggregator.sol       |      100 |      100 |      100 |      100 |                |
  IRewardDistributor.sol           |      100 |      100 |      100 |      100 |                |
  IRewardDistributorManager.sol    |      100 |      100 |      100 |      100 |                |
 contracts/math/                   |     87.5 |      100 |     87.5 |     87.5 |                |
  Exponential.sol                  |     87.5 |      100 |     87.5 |     87.5 |          65,66 |
 contracts/oracle/                 |    46.88 |    61.11 |    44.44 |    46.88 |                |
  PriceOracleAggregator.sol        |    46.88 |    61.11 |    44.44 |    46.88 |... 114,115,116 |
 contracts/oracle/adapter/         |        0 |        0 |        0 |        0 |                |
  ChainlinkUSDAdapter.sol          |        0 |        0 |        0 |        0 |... 30,31,36,37 |
 contracts/rewards/                |    88.27 |    63.95 |    91.43 |    88.61 |                |
  RewardDistributor.sol            |    83.49 |       55 |       85 |    83.02 |... 349,350,352 |
  RewardDistributorFactory.sol     |      100 |      100 |      100 |      100 |                |
  RewardDistributorManager.sol     |    97.62 |    81.82 |      100 |      100 |                |
 contracts/test/                   |    73.91 |       50 |    57.14 |    71.67 |                |
  MockChainlinkUSDAdapter.sol      |       50 |      100 |       50 |       50 |             12 |
  MockFeeWIthdrawUniswapRouter.sol |      100 |      100 |        0 |      100 |                |
  MockFlashBorrower.sol            |       75 |       50 |       50 |       75 |          25,36 |
  MockLendingPair.sol              |        0 |      100 |        0 |        0 |          10,14 |
  MockPriceOracle.sol              |       75 |      100 |       75 |       75 |             21 |
  MockRewardDistributorManager.sol |       20 |      100 |       20 |       10 |... 23,24,25,33 |
  MockToken.sol                    |       80 |      100 |       80 |       80 |             21 |
  MockUniswapV2Router02.sol        |      100 |      100 |      100 |      100 |                |
  MockVault.sol                    |        0 |      100 |       50 |        0 |             12 |
  MockVaultUser.sol                |      100 |      100 |      100 |      100 |                |
  VaultStorageLayoutTester.sol     |      100 |       50 |      100 |      100 |                |
 contracts/token/                  |    96.43 |       50 |    95.45 |    96.61 |                |
  EIP712.sol                       |    92.31 |       50 |      100 |    93.33 |             70 |
  ERC20.sol                        |      100 |       50 |      100 |      100 |                |
  ERC20Permit.sol                  |    88.89 |       50 |       75 |       90 |             78 |
  IERC20Details.sol                |      100 |      100 |      100 |      100 |                |
 contracts/upgradability/          |      100 |       75 |      100 |      100 |                |
  UUPSProxiable.sol                |      100 |       50 |      100 |      100 |                |
  UUPSProxy.sol                    |      100 |      100 |      100 |      100 |                |
  UUPSUtils.sol                    |      100 |      100 |      100 |      100 |                |
 contracts/util/                   |      100 |       50 |      100 |      100 |                |
  Initializable.sol                |      100 |       50 |      100 |      100 |                |
  ReentrancyGuard.sol              |      100 |       50 |      100 |      100 |                |
-----------------------------------|----------|----------|----------|----------|----------------|
All files                          |    89.58 |    64.06 |    86.56 |    89.38 |                |
-----------------------------------|----------|----------|----------|----------|----------------|

> Istanbul reports written to ./coverage/ and ./coverage.json
Error in plugin solidity-coverage: ❌ 1 test(s) failed under coverage.

For more info run Hardhat with --show-stack-traces
```

Tests

```
❯ npm run test

> blacksmith@1.0.0 test
> npx hardhat test

Compiling 67 files with 0.8.1
Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment
--> @uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol


Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment
--> @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol


Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment
--> @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol


Compilation finished successfully
Creating Typechain artifacts in directory types/ for target ethers-v5
Successfully generated Typechain artifacts!


  Borrow Delegation
The Hardhat Network tracing engine could not be initialized. Run Hardhat with --verbose to learn more.
    ✓ borrow delegation (630ms)

  DebtToken
    ✓ underlying
    ✓ mint - fails if not owner
    ✓ burn - fails if not owner
    ✓ transfer
    ✓ approve
    ✓ allowance
    ✓ transferFrom
    ✓ increaseAllowance
    ✓ decreaseAllowance
    ✓ increaseTotalDebt - fails if not owner
    ✓ increaseTotalDebt

  FeeWithdrawal
    ✓ proxiableUUID & updateCode (121ms)
    scenarios
      ✓ withdrawFees (65ms)
      ✓ swapFees (85ms)
      ✓ transferToReceiver (48ms)
      ✓ rescueFunds (171ms)

  LendingPairFactory
    ✓ updatePairImpl (53ms)
    ✓ updateCollateralWrapperImpl (121ms)
```

```
  ✓ updateDebtTokenImpl (54ms)
  ✓ updateBorrowAssetWrapperImpl (90ms)
  ✓ pause
  ✓ unpause
  ✓ disableIR (70ms)
  ✓ createIR
  ✓ createLendingPairWithProxy (100ms)

LendingPair - Warp
  ✓ warp - actions (168ms)
  ✓ warp - pass parameters (262ms)

LendingPair
  ✓ initialize (127ms)
  ✓ depositBorrowAsset - fails without enough vault balance (144ms)
  ✓ depositBorrowAsset (218ms)
  ✓ depositCollateral (243ms)
  ✓ borrow - fails when you try to borrow more than allowed (252ms)
  ✓ borrow - fails when you try to borrow more than cash available in vault (165ms)
  ✓ borrow (344ms)
  ✓ collateral transfer - can not transfer more than allowed that puts loan at risk (350ms)
  ✓ collateral transferFrom - can not transferFrom more than allowed that puts loan at risk (697ms)
  ✓ withdrawCollateral (445ms)
  ✓ repay - fails if you are trying to pay more than owed (148ms)
  ✓ repay (441ms)
  ✓ redeem - fails if you are trying to redeem more than your account (224ms)
  ✓ redeem (332ms)
  ✓ liquidate - cannot liquidate self (119ms)
  ✓ liquidate & withdrawFees - correctly (473ms)
  ✓ withdrawFees - fails without enough balance (166ms)
  ✓ pause (220ms)
  ✓ unpause (327ms)

PriceOracleAggregator
  ✓ correct team address
  ✓ updateOracleForAsset - fails for non admin
  ✓ updateOracleForAsset - fails for invalid oracle adddress
  ✓ getPriceInUSD - fails for non existent oracle
  ✓ setOracleForAsset

RewardDistributorFactory
  ✓ setDistributorImplementation
  ✓ createRewardDistributor (2801ms)

RewardDistributorManager
  ✓ initialize
  ✓ setDistributorStatus
  ✓ activateReward (38ms)
  ✓ removeReward (69ms)
  ✓ acceptOwnerTransfer & transferOwnership
  ✓ proxiableUUID & updateCode (94ms)
```

```
  RewardDistributor
========= NOTICE =========
Request-Rate Exceeded  (this message will not be repeated)

The default API keys for each service are provided as a highly-throttled,
community resource for low-traffic projects and early prototyping.

While your application will continue to function, we highly recommended
signing up for your own API keys to improve performance, increase your
request rate/limit and enable other perks, such as metrics and advanced APIs.

For more details: https://docs.ethers.io/api-keys/
==========================
    ✓ initialize (6468ms)
    ✓ add & set (6198ms)
    ✓ should not accumulate reward if current time < startTimestamp (9231ms)
    ✓ reward calculation (19237ms)
    ✓ withdrawUnclaimedRewards & reward calculation - should allocate previous pending rewards (13722ms)
    ✓ withdraw calculates the reward and disburses it (5556ms)

  scenarios
    ✓ lending pair with different decimal places borrow & collateral asset (565ms)
    ✓ multiple users

  UUPSProxy
    initializeProxy
      ✓ initializeProxy - fails when address is 0
      ✓ initializeProxy - fails when already initialized

  Vault Factory
0x3e786a510517a8051D57f57dbd157e3eb405563E
    ✓ updateVaultLogic
    ✓ createUpgradableVault (38ms)

  Vault
    ✓ initialize fails with 0 team
    ✓ initialize - correctly
    ✓ version
    ✓ name
    ✓ proxiableUUID
    ✓ pause & unpause - fails incorrect user
    ✓ pause (51ms)
    ✓ allowContract
    ✓ approveContract - contract (84ms)
    ✓ approveContract - fails invalid to / fails with wrong signature
    ✓ approveContract (141ms)
    ✓ deposit fails with invalid `to` address
    ✓ deposit - correctly with correct user balance (39ms)
    ✓ deposit fails with incorrect approve deposit (63ms)
    ✓ transfer - correctly & fails with invalid `to` address (78ms)
```

```
    ✓ maxFlashLoan
    ✓ flashFee
    ✓ flashLoan - correctly (88ms)
    ✓ withdraw fails with invalid `to` address
    ✓ user cannot withdraw more than balance
    ✓ withdraw - correctly (71ms)
    ✓ withdraw - fails if user tries to withdraw past minimum share balance (40ms)
    ✓ updateFlashloanRate
    ✓ transferOwnership & acceptOwnership (51ms)
    ✓ toShare/toUnderlying - convert to appropriate shares & underlying (200ms)
    ✓ rescueFunds (150ms)

  Vault - Upgradable Layout
0x4A679253410272dd5232B3Ff7cF5dbB88f295319
    ✓ updateCode (100ms)
    ✓ test storage layout

  BorrowToken/CollateralToken
    ✓ underlying (66ms)
    ✓ only owner can mint (75ms)
    ✓ only owner can burn (73ms)
    ✓ permit (178ms)
    ✓ transfer/transferFrom (318ms)
    ✓ increaseAllowance/decreaseAllowance


  106 passing (2m)
```

License

This report falls under the terms described in the included LICENSE.