Quantstamp Security Assessment Certificate

QUANTSTAMP VERIFIED
SECURITY CERTIFICATE

January 5th 2021 — Quantstamp Verified

Mainframe

This security assessment was prepared by Quantstamp, the leader in blockchain security.

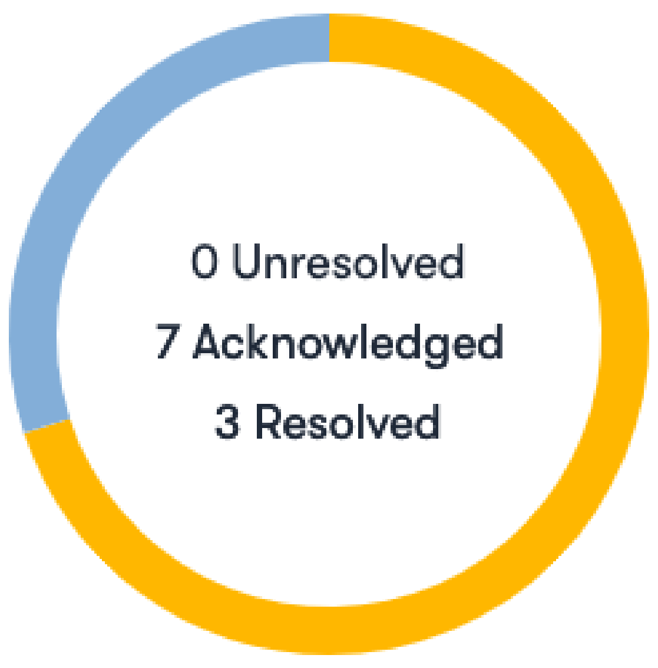
Executive Summary

Type	Fixed-rate lending protocol
Auditors	Jan Gorzny, Blockchain Researcher Ed Zulkoski, Senior Security Engineer Kacper Bąk, Senior Research Engineer
Timeline	2020-10-26 through 2020-11-18
EVM	Muir Glacier
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Whitepaper
Documentation Quality	High
Test Quality	High
Source Code	

Repository	Commit
mainframe-lending-protocol	72d6bb4
contracts	6c3ee4a

Goals	• Look for issues which may cause lost or locked funds, denial of service, or unintended behaviour.

Total Issues	**10**	(3 Resolved)
High Risk Issues	0	(0 Resolved)
Medium Risk Issues	0	(0 Resolved)
Low Risk Issues	**5**	(1 Resolved)
Informational Risk Issues	**3**	(0 Resolved)
Undetermined Risk Issues	**2**	(2 Resolved)



High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
? Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

Quantstamp has reviewed the Mainframe lending protocol and found several issues, most of which are of low or informational severity. The code has excellent test coverage and most functions are accompanied by descriptive comments.

ID	Description	Severity	Status
QSP-1	Potential Misuse of `decimals`	⌄ Low	Acknowledged
QSP-2	Assumed Constant Number of Decimals	⌄ Low	Acknowledged
QSP-3	Privileged Roles and Ownership	⌄ Low	Acknowledged
QSP-4	Admin Must Check Token Behaviour	⌄ Low	Mitigated
QSP-5	Possible Malicious Interactions With Other Contracts	⌄ Low	Acknowledged
QSP-6	Race Conditions / Front-Running	○ Informational	Acknowledged
QSP-7	Unlocked Pragma	○ Informational	Acknowledged
QSP-8	Clone-and-Own	○ Informational	Acknowledged
QSP-9	Missing `require` Statements	? Undetermined	Fixed
QSP-10	Possible Bad Input	? Undetermined	Fixed

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
    i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
    i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Findings

QSP-1 Potential Misuse of `decimals`

Severity: Low Risk

Status: Acknowledged

File(s) affected: `FyToken.sol`

Description: Lines 60, 67: although `decimals()` usually returns the correct expected value, according to the ERC20 spec "This method can be used to improve usability, but interfaces and other contracts MUST NOT expect these values to be present" From EIPs/eip-20.md at master · ethereum/EIPs.

Recommendation: Hard-code the value of expected decimals, or record this information elsewhere.

QSP-2 Assumed Constant Number of Decimals

Severity: Low Risk

Status: Acknowledged

File(s) affected: `FyToken.sol`

Description: It is important to state explicitly that `FyToken` expects the underlying and collateral to have a constant values of `decimals()`. It's a reasonable assumption, but there may exist tokens that have a variable value of `decimals()`.

Recommendation: Update the documentation, or enforce the necessity of constant decimals somehow.

QSP-3 Privileged Roles and Ownership

Severity: Low Risk

Status: Acknowledged

File(s) affected: `Fintroller.sol`

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract. However, this centralization of power needs to be made clear to the users, especially depending on the level of privilege the contract allows to the owner.

Exploit Scenario: - If the admin invokes `setCollateralizationRatio` and increases the ratio (say from 1 to 1.5), the token can be immediately liquidated.

- The owner can turn off/on any functionality related to fyTokens (e.g., repayment, deposits, etc.).
- Prices are dependent on an oracle that can be changed at any time.

Recommendation: Ensure that users understand that this role exists and the actions it may perform. Consider using a TimeLock multisig for the admin role.

QSP-4 Admin Must Check Token Behaviour

Severity: Low Risk

Status: Mitigated

Description: The admin should vet each fy token that wraps another token to ensure the wrapped token behaves as expected.

Recommendation: Ensure that admins are aware of this responsibility.

QSP-5 Possible Malicious Interactions With Other Contracts

Severity: Low Risk

Status: Acknowledged

File(s) affected: `BalanceSheet.sol`

Description: Malicious interactions with other protocols like Uniswap may be possible. For example, the contract may be susceptible to flash loans.

Recommendation: Warn users of such possibilities or build guards into the contract.

QSP-6 Race Conditions / Front-Running

Severity: Informational

Status: Acknowledged

Description: A block is an ordered collection of transactions from all around the network. It's possible for the ordering of these transactions to manipulate the end result of a block. A miner attacker can take advantage of this by generating and moving transactions in a way that benefits themselves.
Specifically, if an account can be liquidated, only the first mined transaction(s) to deplete the total debt will succeed.

Recommendation: Quantstamp has no recommendations at this time.

QSP-7 Unlocked Pragma

Severity: Informational

Status: Acknowledged

File(s) affected: `SimpleUniswapAnchoredView.sol`, `TestOraclePriceUtils.sol`, `GodModRedemptionPool.sol`, `GodModeFyToken.sol`, `GodModeBalanceSheet.sol`, `Erc20Mintable.sol`, `UniswapAnchoredViewInterface.sol`, `OraclePriceUtils.sol`, `BaseInvariants.sol`, `FintrollerInvariants.sol`, `BalanceSheet.sol`, `BalanceSheetInterface.sol`, `BalanceSheetStorage.sol`, `Fintroller.sol`, `FintrollerInterface.sol`, `FintrollerStorage.sol`, `FyToken.sol`, `FyTokenInterface.sol`, `FyTokenStorage.sol`, `RedemptionPool.sol`, `RedemptionPoolInterface.sol`, `RedemptionPoolStorage.sol`

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.4.*`. The caret (^) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, it is recommended to remove the caret to lock the file onto a specific Solidity version.

QSP-8 Clone-and-Own

Severity: Informational

Status: Acknowledged

Description: The clone-and-own approach involves copying and adjusting open source code at one's own discretion. From the development perspective, it is initially beneficial as it reduces the amount of effort. However, from the security perspective, it involves some risks as the code may not follow the best practices, may contain a security vulnerability, or may include intentionally or unintentionally modified upstream libraries.
Several files, particularly libraries, are cloned for this project.

Recommendation: Rather than the clone-and-own approach, a good industry practice is to use the Truffle framework for managing library dependencies. This eliminates the clone-and-own risks yet allows for following best practices, such as, using libraries.

QSP-9 Missing `require` Statements

Severity: Undetermined

Status: Fixed

File(s) affected: `Fintroller.sol`, `BalanceSheet.sol`

Description: `Fintroller.sol`: Line 171 may be a no-op, since its return value is not checked.
`BalanceSheet.sol`: Line 540 may be a no-op, since its return value is not checked.

Recommendation: These lines should be wrapped in a `require()` call.

QSP-10 Possible Bad Input

Severity: Undetermined

Status: Fixed

File(s) affected: `Fintroller.sol`

Description: The function `setDebtCeiling` could potentially set the ceiling below the current debt.

Recommendation: Prevent this scenario if it is undesirable.

Code Documentation

`Fintroller.sol`: line 346 is missing a function description. **Update:** This has been fixed.

Adherence to Best Practices

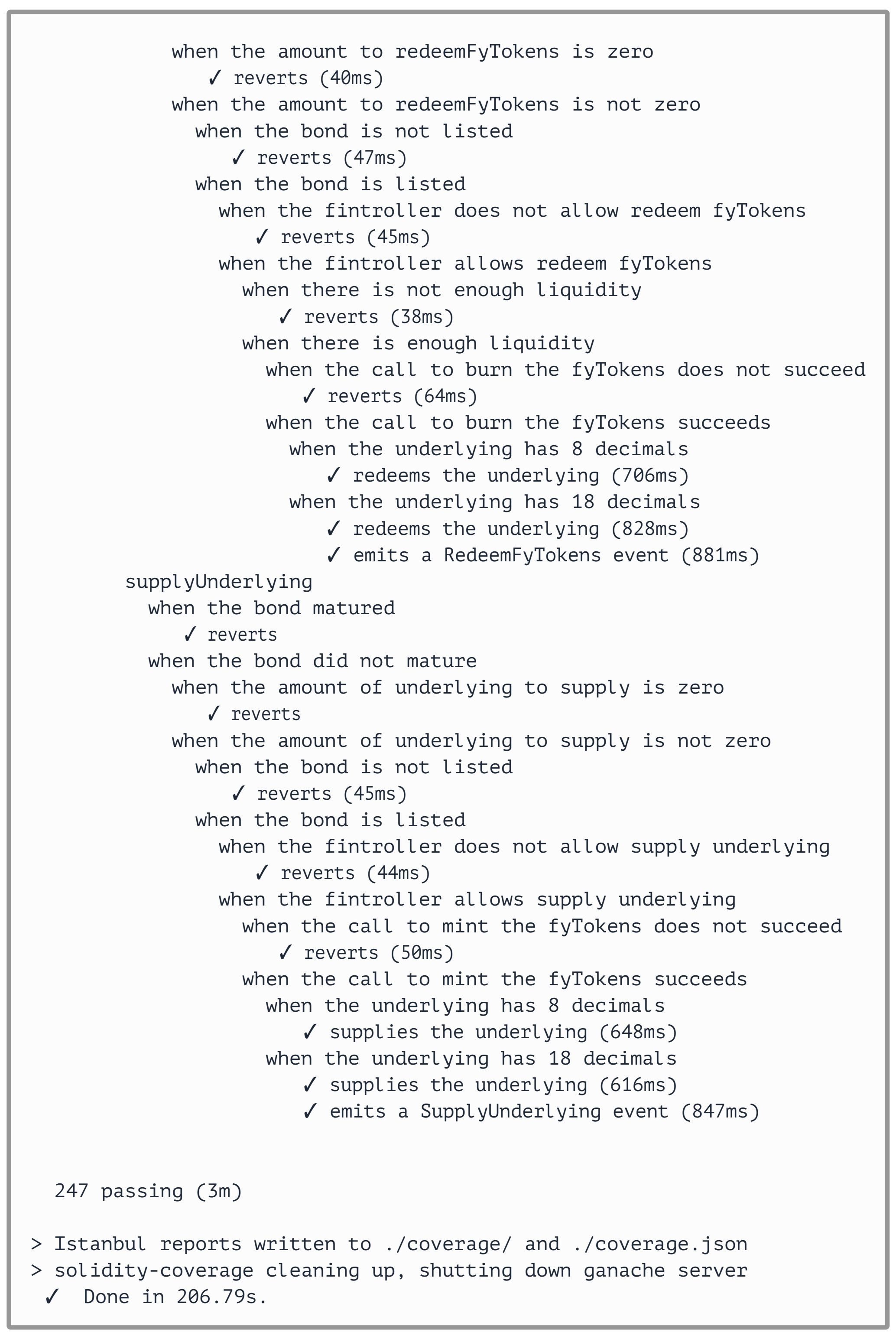
`Fintroller.sol`: line 353 is not implemented. `BaseInvariants.sol`: The listed addresses are likely not those to be used in the production deployment.

Test Results

Test Suite Results

```
Integration Tests
    FyToken
      Effects Functions
        borrow
           ✓ borrows fyTokens (1811ms)
           ✓ increases the debt of the caller (1948ms)
           ✓ emits a SetVaultDebt event (1750ms)
        burn
          when the amount to burn is zero
             ✓ reverts
        liquidateBorrow
          when there is not enough locked collateral
             ✓ reverts (202ms)
          when there is enough locked collateral
             ✓ liquidates the borrower (3129ms)
             ✓ reduces the debt of the borrower (2989ms)
             ✓ reduces the locked collateral of the borrower (3288ms)
             ✓ transfers the clutched collateral to the liquidator (3045ms)
             ✓ emits a ClutchCollateral event (2860ms)
        mint
          when the amount to mint is zero
             ✓ reverts
        repayBorrow
           ✓ repays the borrow (726ms)
           ✓ reduces the debt of the caller (727ms)
           ✓ emits a SetVaultDebt event (692ms)
    RedemptionPool
      Effects Functions
        redeemFyTokens
          when the bond matured
             ✓ redeems the fyTokens (232ms)
             ✓ burns the fyTokens (290ms)
             ✓ emits a Burn event (228ms)
        supplyUnderlying
           ✓ supplies the underlying (1036ms)
           ✓ mints the new fyTokens (1360ms)
           ✓ emits a Mint event (883ms)

  Unit Tests
    BalanceSheet
      View Functions
        fintroller
           ✓ retrieves the address of the fintroller contract
        getClutchableCollateral
          when the amount to repay is zero
             ✓ reverts (48ms)
          when the amount to repay is not zero
            when the liquidation incentive is zero
               ✓ retrieves zero
            when the liquidation incentive is not zero
              when the collateral has 18 decimals
                 ✓ retrieves the clutchable collateral amount (116ms)
              when the collateral has 8 decimals
                 ✓ retrieves the downscaled clutchable collateral amount (103ms)
        getCurrentCollateralizationRatio
           ✓ returns the current collateralization ratio mantissa (303ms)
        getHypotheticalCollateralizationRatio
          when the vault is not open
             ✓ reverts (47ms)
          when the vault is not open
            when the locked collateral is zero
               ✓ reverts
            when the locked collateral is not zero
              when the debt is zero
                 ✓ reverts
              when the debt is not zero
                when the collateral price from the oracle is zero
                   ✓ reverts (62ms)
                when the collateral price from the oracle is not zero
                  when the underlying price from the oracle is zero
                     ✓ reverts (127ms)
                  when the collateral price from the oracle is not zero
                    when the collateral has 8 decimals
                       ✓ retrieves the hypothetical collateralization ratio mantissa (122ms)
                    when the collateral has 18 decimals
                       ✓ retrieves the hypothetical collateralization ratio mantissa (98ms)
        getVault
          when the bond is not open
             ✓ retrieves the default values
          when the vault is open
             ✓ retrieves all the storage properties of the vault
        getVaultDebt
          when the bond is not open
             ✓ retrieves the default value
          when the vault is open
             ✓ retrieves the default value
        getVaultLockedCollateral
          when the bond is not open
             ✓ retrieves the default value (39ms)
          when the vault is open
             ✓ retrieves the default value
        isAccountUnderwater
          when the vault is not open
             ✓ retrieves false
          when the vault is open
            when the debt is zero
               ✓ retrieves false
            when the debt is non-zero
              when the user is safely collateralized
                 ✓ retrieves false (86ms)
              when the user is dangerously collateralized
                 ✓ retrieves true (91ms)
        isBalanceSheet
           ✓ retrieves true
        isVaultOpen
          when the vault is not open
             ✓ retrieves false
          when the vault is open
             ✓ retrieves true
      Effects Functions
        clutchCollateral
          when the caller is not the fyToken contract
             ✓ reverts (48ms)
        depositCollateral
          when the vault is not open
             ✓ reverts (39ms)
          when the vault is open
            when the amount to deposit is zero
               ✓ reverts (41ms)
            when the amount to deposit is not zero
              when the bond is not listed
                 ✓ reverts (44ms)
              when the bond is listed
                when the fintroller does not allow deposit collateral
                   ✓ reverts (169ms)
                when the fintroller allows deposit collateral
                  when the call to transfer the collateral does not succeed
                     ✓ reverts (63ms)
                  when the call to transfer the collateral succeeds
                     ✓ makes the collateral deposit (576ms)
                     ✓ emits a DepositCollateral event (759ms)
        freeCollateral
          when the vault is not open
             ✓ reverts (42ms)
          when the vault is open
            when the collateral amount to free is zero
               ✓ reverts (39ms)
            when the collateral amount to free is not zero
              when the caller did not deposit any collateral
                 ✓ reverts
              when the caller deposited collateral
                when the caller did not lock the collateral
                   ✓ reverts (40ms)
                when the caller locked the collateral
                  when the caller does not have a debt
                     ✓ it frees the collateral (147ms)
                  when the caller has a debt
                    when the caller is dangerously collateralized
                       ✓ reverts (119ms)
                    when the caller is safely over-collateralized
                       ✓ it frees the collateral (348ms)
                       ✓ emits a FreeCollateral event (299ms)
        lockCollateral
          when the vault is open
            when the collateral amount to lock is not zero
              when the caller deposited collateral
                 ✓ it locks the collateral (137ms)
                 ✓ emits a LockCollateral event (179ms)
              when the caller did not deposit any collateral
                 ✓ reverts
            when the collateral amount to lock is zero
               ✓ reverts
          when the vault is not open
             ✓ reverts
        openVault
          when the vault is open
             ✓ reverts
          when the vault is not open
            when the fyToken is not compliant
               ✓ reverts
            when the fyToken is compliant
               ✓ opens the vault (100ms)
               ✓ emits an OpenVault event (140ms)
        setVaultDebt
          when the caller is not the fyToken contract
             ✓ reverts
        withdrawCollateral
          when the vault is not open
             ✓ reverts (39ms)
          when the vault is open
            when the amount to withdraw is zero
               ✓ reverts
            when the amount to withdraw is not zero
              when the caller did not deposit any collateral
                 ✓ reverts
              when the caller deposited collateral
                when the caller locked the collateral
                   ✓ reverts (60ms)
                when the caller did not lock the collateral
                   ✓ makes the collateral withdrawal (420ms)
                   ✓ emits a WithdrawCollateral event (515ms)
    Fintroller
      View Functions
        getBond
          when the bond is listed
             ✓ retrieves the default values after listing
          when the bond is not listed
             ✓ retrieves the default values
        getBondCollateralizationRatio
          when the bond is not listed
             ✓ retrieves zero
          when the bond is listed
             ✓ retrieves the default collateralization ratio
        getBondDebtCeiling
          when the bond is not listed
             ✓ retrieves zero
          when the bond is listed
             ✓ retrieves the default debt ceiling
        getBorrowAllowed
          when the bond is not listed
             ✓ reverts (39ms)
          when the bond is listed
             ✓ retrieves the default value
        getDepositCollateralAllowed
          when the bond is not listed
             ✓ reverts
          when the bond is listed
             ✓ retrieves the default value
        getLiquidateBorrowAllowed
          when the bond is not listed
             ✓ reverts
          when the bond is listed
             ✓ retrieves the default value
            getRedeemFyTokensAllowed
              when the bond is not listed
                ✓ reverts
              when the bond is listed
                ✓ retrieves the default value
            getRepayBorrowAllowed
              when the bond is not listed
                ✓ reverts
              when the bond is listed
                ✓ retrieves the default value
            getSupplyUnderlyingAllowed
              when the bond is not listed
                ✓ reverts
              when the bond is listed
                ✓ retrieves the default value
            isFintroller
              ✓ retrieves true
            liquidationIncentiveMantissa
              ✓ retrieves the default value
            oracle
              ✓ retrieves the address of the oracle contract
            oraclePricePrecisionScalar
              ✓ retrieves the oracle precision scalar
          Effects Functions
            listBond
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the contract to be listed is non-compliant
                  ✓ rejects
                when the contract to be listed is compliant
                  ✓ lists the bond (142ms)
                  ✓ emits a ListBond event (143ms)
            setBondDebtCeiling
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the bond is not listed
                  ✓ reverts
                when the bond is listed
                  when the debt ceiling is zero
                    ✓ reverts
                  when the debt ceiling is not zero
                    when the debt ceiling is below the current debt
                      ✓ reverts
                    when the debt ceiling is not below the current debt
                      ✓ sets the new debt ceiling (116ms)
                      ✓ emits a SetBondDebtCeiling event (106ms)
            setBorrowAllowed
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the bond is not listed
                  ✓ rejects
                when the bond is listed
                  ✓ sets the value to true (99ms)
                  ✓ sets the value to false (94ms)
                  ✓ emits a SetBorrowAllowed event (89ms)
            setCollateralizationRatio
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the bond is not listed
                  ✓ reverts
                when the bond is listed
                  when the collateralization ratio is not valid
                    when the collateralization ratio is higher than 10,000%
                      ✓ reverts
                    when the collateralization ratio is lower than 100%
                      ✓ reverts
                    when the collateralization ratio is zero
                      ✓ reverts
                  when the collateralization ratio is valid
                    ✓ sets the new collateralization ratio (101ms)
                    ✓ emits a SetCollateralizationRatio event (110ms)
            setDepositCollateralAllowed
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the bond is not listed
                  ✓ rejects (38ms)
                when the bond is listed
                  ✓ sets the value to true (100ms)
                  ✓ sets the value to false (108ms)
                  ✓ emits a SetDepositCollateralAllowed event (107ms)
            setLiquidateBorrowAllowed
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the bond is not listed
                  ✓ rejects
                when the bond is listed
                  ✓ sets the value to true (110ms)
                  ✓ sets the value to false (102ms)
                  ✓ emits a SetLiquidateBorrowAllowed event (100ms)
            setLiquidationIncentive
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the liquidation incentive is not valid
                  when the liquidation ratio is zero
                    ✓ reverts
                  when the liquidation incentive is higher than 150%
                    ✓ reverts
                  when the liquidation incentive is lower than 100%
                    ✓ reverts
                when the liquidation incentive is valid
                  ✓ sets the new liquidation incentive (95ms)
                  ✓ emits a SetLiquidationIncentive event (91ms)
            setOracle
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when oracle address is not the zero address
                  ✓ sets the new oracle (98ms)
                  ✓ emits a SetOracle event (93ms)
                when the oracle address is the zero address
                  ✓ reverts
            setRedeemFyTokensAllowed
              when the caller is not the admin
                ✓ reverts (39ms)
              when the caller is the admin
                when the bond is not listed
                  ✓ rejects
                when the bond is listed
                  ✓ sets the value to true (108ms)
                  ✓ sets the value to false (136ms)
                  ✓ emits a SetRedeemFyTokensAllowed event (88ms)
            setRepayBorrowAllowed
              when the caller is not the admin
                ✓ reverts
              when the caller is the admin
                when the bond is not listed
                  ✓ rejects
                when the bond is listed
                  ✓ sets the value to true (106ms)
                  ✓ sets the value to false (128ms)
                  ✓ emits a SetRepayBorrowAllowed event (113ms)
            setSupplyUnderlyingAllowed
              when the caller is not the admin
                ✓ reverts (38ms)
              when the caller is the admin
                when the bond is not listed
                  ✓ rejects
                when the bond is listed
                  ✓ sets the value to true (114ms)
                  ✓ sets the value to false (88ms)
                  ✓ emits a SetSupplyUnderlyingAllowed event (88ms)
        FyToken
          Constructor
            when the underlying has zero decimals
              ✓ reverts (223ms)
            when the underlying has more than 18 decimals
              ✓ reverts (163ms)
            when the collateral has zero decimals
              ✓ reverts (148ms)
            when the collateral has more than 18 decimals
              ✓ reverts (321ms)
            when the expiration time is in the past
              ✓ reverts (207ms)
          View Functions
            balanceSheet
              ✓ retrieves the address of the balance sheet contract
            collateral
              ✓ retrieves the contract address of the collateral
            collateralPrecisionScalar
          when the collateral has 18 decimals
             ✓ retrieves 1
          when the collateral has 8 decimals
             ✓ retrieves 1.0e10 (316ms)
        expirationTime
           ✓ retrieves the expiration time
        fintroller
           ✓ retrieves the address of the fintroller contract
        isFyToken
           ✓ retrieves true
        redemptionPool
           ✓ retrieves the address of the redemption pool contract
        underlying
           ✓ retrieves the contract address of the underlying
        underlyingPrecisionScalar
          when the underlying has 18 decimals
             ✓ retrieves 1
          when the underlying has 8 decimals
             ✓ retrieves 1.0e10 (248ms)
      Effects Functions
        borrow
          when the vault is not open
             ✓ reverts (46ms)
          when the vault is open
            when the bond matured
               ✓ reverts (49ms)
            when the bond did not mature
              when the amount to borrow is zero
                 ✓ reverts (45ms)
              when the amount to borrow is not zero
                when the bond is not listed
                   ✓ reverts (47ms)
                when the bond is listed
                  when the fintroller does not allow borrows
                     ✓ reverts (47ms)
                  when the fintroller allows borrows
                    when the borrow overflows the debt ceiling
                       ✓ reverts (57ms)
                    when the borrow does not overflow the debt ceiling
                      when the caller did not deposit any collateral
                         ✓ reverts (67ms)
                      when the caller deposited collateral
                        when the caller did not lock the collateral
                           ✓ reverts (60ms)
                        when the caller locked the collateral
                          when the user is dangerously collateralized
                             ✓ reverts (77ms)
                          when the user is safely collateralized
                            when the call to set the new vault debt does not succeed
                               ✓ reverts (76ms)
                            when the call to set the new vault debt succeeds
                               ✓ borrows fyTokens (945ms)
                               ✓ emits a Borrow event (898ms)
                               ✓ emits a Mint event (782ms)
                               ✓ emits a Transfer event (860ms)
        burn
          when the caller is not the fyToken contract
             ✓ reverts (41ms)
        liquidateBorrow
          when the vault is not open
             ✓ reverts (39ms)
          when the vault is open
            when the caller is the borrower
               ✓ reverts (46ms)
            when the caller is not the borrower
              when the amount to repay is zero
                 ✓ reverts (48ms)
              when the amount to repay is not zero
                when the bond is not listed
                   ✓ reverts (48ms)
                when the bond is listed
                  when the fintroller does not allow liquidate borrow
                     ✓ reverts (48ms)
                  when the fintroller allows liquidate borrow
                    when the borrower does not have a debt
                       ✓ reverts (69ms)
                    when the borrower has a debt
                      when the account is not underwater
                        when the bond did not mature
                           ✓ reverts (54ms)
                        when the bond matured
                           ✓ liquidates the borrower (938ms)
                      when the account is underwater
                        when the caller does not have enough fyTokens
                           ✓ reverts (69ms)
                        when the caller has enough fyTokens
                           ✓ liquidates the borrower (783ms)
                           ✓ emits a Burn event (792ms)
                           ✓ emits a Transfer event (824ms)
                           ✓ emits a RepayBorrow event (778ms)
                           ✓ emits a LiquidateBorrow event (754ms)
        mint
          when the caller is not the fyToken contract
             ✓ reverts (46ms)
        repayBorrow
          when the vault is not open
             ✓ reverts (42ms)
          when the vault is open
            when the amount to repay is zero
               ✓ reverts (44ms)
            when the amount to repay is not zero
              when the bond is not listed
                 ✓ reverts (52ms)
              when the bond is listed
                when the fintroller does not allow repay borrow
                   ✓ reverts (50ms)
                when the fintroller allows repay borrow
                  when the caller does not have a debt
                     ✓ reverts (57ms)
                  when the caller has a debt
                    when the caller does not have enough fyTokens
                       ✓ reverts (52ms)
                    when the caller has enough fyTokens
                       ✓ repays the borrowed fyTokens (598ms)
                       ✓ emits a Burn event (725ms)
                       ✓ emits a Transfer event (554ms)
                       ✓ emits a RepayBorrow event (603ms)
        repayBorrowBehalf
          when the vault is not open
             ✓ reverts (39ms)
          when the vault is open
            when the amount to repay is zero
               ✓ reverts (42ms)
            when the amount to repay is not zero
              when the bond is listed
                when the fintroller allows repay borrow
                  when the user does not have a debt
                     ✓ reverts (62ms)
                  when the user has a debt
                     ✓ repays the borrowed fyTokens (839ms)
                     ✓ emits a Burn event (792ms)
                     ✓ emits a Transfer event (593ms)
                     ✓ emits a RepayBorrow event (760ms)
        setFintroller
          when the caller is not the administrator
             ✓ reverts (51ms)
          when the caller is the administrator
            when the new Fintroller is not compliant
               ✓ reverts (40ms)
            when the new Fintroller is compliant
               ✓ sets the new Fintroller (111ms)
    OraclePriceUtils
      getAdjustedPrice
        when the oracle does not have price data for the symbol
           ✓ reverts
        when the oracle has price data for the symbol
          when the precision scalar multiplication overflows
             ✓ reverts
          when the precision scalar multiplication does not overflow
             ✓ retrieves the adjusted price
    RedemptionPool
      View Functions
        fyToken
           ✓ retrieves the address of the fyToken contract
        isRedemptionPool
           ✓ retrieves true
        totalUnderlyingSupply
          when the underlying supply is zero
             ✓ retrieves zero
          when the total underlying supply is not zero
             ✓ retrieves the correct amount
      Effects Functions
        redeemFyTokens
          when the bond did not mature
             ✓ reverts
          when the bond matured
```



```
            when the amount to redeemFyTokens is zero
              ✓ reverts (40ms)
            when the amount to redeemFyTokens is not zero
              when the bond is not listed
                ✓ reverts (47ms)
              when the bond is listed
                when the fintroller does not allow redeem fyTokens
                  ✓ reverts (45ms)
                when the fintroller allows redeem fyTokens
                  when there is not enough liquidity
                    ✓ reverts (38ms)
                  when there is enough liquidity
                    when the call to burn the fyTokens does not succeed
                      ✓ reverts (64ms)
                    when the call to burn the fyTokens succeeds
                      when the underlying has 8 decimals
                        ✓ redeems the underlying (706ms)
                      when the underlying has 18 decimals
                        ✓ redeems the underlying (828ms)
                        ✓ emits a RedeemFyTokens event (881ms)
        supplyUnderlying
          when the bond matured
            ✓ reverts
          when the bond did not mature
            when the amount of underlying to supply is zero
              ✓ reverts
            when the amount of underlying to supply is not zero
              when the bond is not listed
                ✓ reverts (45ms)
              when the bond is listed
                when the fintroller does not allow supply underlying
                  ✓ reverts (44ms)
                when the fintroller allows supply underlying
                  when the call to mint the fyTokens does not succeed
                    ✓ reverts (50ms)
                  when the call to mint the fyTokens succeeds
                    when the underlying has 8 decimals
                      ✓ supplies the underlying (648ms)
                    when the underlying has 18 decimals
                      ✓ supplies the underlying (616ms)
                      ✓ emits a SupplyUnderlying event (847ms)


  247 passing (3m)

> Istanbul reports written to ./coverage/ and ./coverage.json
> solidity-coverage cleaning up, shutting down ganache server
 ✓  Done in 206.79s.
```

Code Coverage

File	Statements		Branches		Functions		Lines	
BalanceSheet.sol	100%	**130**/130	78.57%	**66**/84	100%	**17**/17	100%	**131**/131
BalanceSheetInterface.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
BalanceSheetStorage.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
Fintroller.sol	100%	**84**/84	100%	**44**/44	100%	**21**/21	100%	**84**/84
FintrollerInterface.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
FintrollerStorage.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
FyToken.sol	100%	**83**/83	93.55%	**60**/64	100%	**11**/11	100%	**84**/84
FyTokenInterface.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
FyTokenStorage.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
RedemptionPool.sol	100%	**36**/36	86.67%	**26**/30	100%	**3**/3	100%	**36**/36
RedemptionPoolInterface.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
RedemptionPoolStorage.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0
OraclePriceUtils.sol	**100%**	13/13	**100%**	8/8	**100%**	3/3	**100%**	13/13
UniswapAnchoredViewInterface.sol	100%	**0**/0	100%	**0**/0	100%	**0**/0	100%	**0**/0

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

dffba7a13053aef1463adddff7fcc6fbf2c30ee40acfa8e8f6e1a5ca01747a8c ./contracts/BalanceSheet.sol

2ba14b0398d12c6f047533c5dffa8000c09d1a253c1cc1646f13a4e671d746cf ./contracts/BalanceSheetInterface.sol

9a6c3c6707fad861228cd6b5f807964cd5473f6dc10b2bc32966a8e24922e839 ./contracts/BalanceSheetStorage.sol

463b00d9e19d257fef27173f57d9f3be2ce9da410b09f138e1dcd17ecc94378b ./contracts/Fintroller.sol

50c25a6c1a814ac8a45616a95dd6b63e090205ecc475b7b53116841c602ffd3d ./contracts/FintrollerInterface.sol

06f43e94446ea1797f0c8684efbe08444468c0a5587a36984f3c3e895a9e1bb1 ./contracts/FintrollerStorage.sol

3592caba8613b887995714eec804065d827595a8beb20be895778b8dd6b1c1ab ./contracts/FyToken.sol

1c0d45d35b0d1746c85c1ac9cbc8057efab48bada35e1a90517274f19310f34c ./contracts/FyTokenInterface.sol

a5063ecb28e7ae08e8f9d86d3ac1890e1510aba5c06a2b86393060555d5e812f ./contracts/FyTokenStorage.sol

df1b8702c4c29e680913b6225a8d38b46e863f9c2d5dc8566c670525b9e46a8d ./contracts/RedemptionPool.sol

2e0142ae8533091abb00c46fbde5c1e21083108b25b178edd731f0f430f64c04 ./contracts/RedemptionPoolInterface.sol

103922d8aba9b2f9dd80504f0c3b4dbaf1e5e3b0d00a05f514c06b68df8902a5 ./contracts/RedemptionPoolStorage.sol

```
5b7b0033cacdba6a81495624efa462300f079a8c88d8448b1bd74a9c7be87dd8 ./contracts/test/Erc20Mintable.sol
ed2377831614442a5496b1d9d12dc8d3a5dbfae068062eda13238320a860ff4c ./contracts/test/GodModeBalanceSheet.sol
5a92f0b9296d96375afa40ea180014db6fa57b5e5a8ba697b3405488ec560283 ./contracts/test/GodModeFyToken.sol
395e0fc3ad7bec4d6b2459ea57985b274871c189f57b86f7004f9388d4a6e93a ./contracts/test/GodModRedemptionPool.sol
14060d10b5c7f05c78279310d314ac8af9666c3494400f26e810d56cd9c00241 ./contracts/test/SimpleUniswapAnchoredView.sol
5b1a971405cc47ab57269df3d01f75fea1a52b3ebfb21d2e98bb8fd026c9408e ./contracts/test/TestOraclePriceUtils.sol
001655961ac9565166431cca6a051cf4643b2324643d7da2e48e9362e5b1a1ca ./contracts/oracles/OraclePriceUtils.sol
ac0096ef77bf093a1f5d45777e0790e7378f93a87c04afc263b827809d3aac33 ./contracts/oracles/UniswapAnchoredViewInterface.sol
b81b2f85ee69cbac99ead306a864a4c1d79aec3fdef3fd08de1ae53c8db37776 ./contracts/invariants/BaseInvariants.sol
2090abad321d8d82ad9e4c1c23a7751a0cb617dacfdeb744a1c696a87b69c3c5 ./contracts/invariants/FintrollerInvariants.sol
```

Tests

```
075821f3f41cc36d97627db9aee379b3dac532118ea060776a6e8ea6c6fca8e1 ./test/contexts.ts
439c6a7d0cc09b93fdc422416d648eb4a7238795a23c0117049a0ca20c948417 ./test/deployers.ts
ee3067fb9e474ab28aed0de46b0d1ab9b354571cd1edf86c089a4d7230b3997b ./test/jsonRpc.ts
ad46677e55f036415472cfb3b24c761cb07dd3ddd8c0bce37a2a1f2f47915632 ./test/scenarios.ts
b3ac12fcf9fe6ce38aa7b5c1565c97d5d90e5622168390712c1e08bd81aef933 ./test/unit/fixtures.ts
d09f0d30709614ddf09a604a7b036cbc71a0bdb85bb4d187d6bc91819e4da45e ./test/unit/index.ts
c82ddf4bef26165fcc3657776ff9fb8b86d4d70eb54390b1deb8d948b2c08797 ./test/unit/stubs.ts
33291d13e324bc8e133daa52ae7a2e3dd30aa5e2e2f8e90e8ada7f5ae08829ab ./test/unit/redemptionPool/RedemptionPool.behavior.ts
4898db9942ace537300fbd8fa49754626fea3cf7e2c14395081e1c7f42783d14 ./test/unit/redemptionPool/RedemptionPool.ts
c004e71b32e6f0f54abf43a6c5930ecfc32951eb6fa2a0f31433554bcfc01ffd ./test/unit/redemptionPool/view/fintroller.ts
b1f2d556052f8b13c18c40f50ec9044c943712db4958dc2e2f3e745814c5207b ./test/unit/redemptionPool/view/fyToken.ts
c1f2238a36238e2a06c69545d101c58c1fa9428fb6054799d238a54c0ca2d50f ./test/unit/redemptionPool/view/isRedemptionPool.ts
993bf992ba548e56780cd205195c63a9fa5a89b5385e11b1b0f6cbb1022b57b2 ./test/unit/redemptionPool/view/totalUnderlyingSupply.ts
eb4de6db8b36066d60677601fbce2cf88042283c52bec19b47c09c7c43f77ab2 ./test/unit/redemptionPool/effects/redeemFyTokens.ts
6057983aba73246f1e3bc284d093eb54829a618ce258ad9c6a75a32502d78420 ./test/unit/redemptionPool/effects/supplyUnderlying.ts
290e234cc1b8434f76a8668536f691cc29ba777e054b0ec6906043fc3e2d5869 ./test/unit/oraclePriceUtils/OraclePriceUtils.behavior.ts
4ed7e1d1eba20be90005e22f8b8ead70011232992073c68613bb95f59c0f6bdb ./test/unit/oraclePriceUtils/OraclePriceUtils.ts
d3f09f01fda82399562a9f9f0553c95b69ab9753a5b93d71333d94e0bb43f49b ./test/unit/oraclePriceUtils/view/getAdjustedPrice.ts
d2a94a02c43dbb64293d5b37882587096ccc4e297e879065d621a78e274169a9 ./test/unit/fyToken/FyToken.behavior.ts
efcd81a2c5498c740476453da7e4212868e48bf3575ec6ff574be5bc178386ef ./test/unit/fyToken/FyToken.ts
0e764637fa3a279ff227e4c90f1f0bc743f619e862eb9b00bb92df3fa59247dc ./test/unit/fyToken/view/balanceSheet.ts
2a0366b8e62f0754282d115cf37ceca6d48f4cd7ed50f2999c234933677ae004 ./test/unit/fyToken/view/collateral.ts
77682cda0f602465c4d014a7c96b642e2dc442e57e2a44f194613dbd455e35d5 ./test/unit/fyToken/view/collateralPrecisionScalar.ts
c1917d2694f4832b59717576d28f9cd2e35f167a99c5afa5142cd0e6d0b2c31a ./test/unit/fyToken/view/expirationTime.ts
964043b4af3fbe727048c3c491a846d7881dbb0efa25e4b8ac5ab6efd13b0df1 ./test/unit/fyToken/view/fintroller.ts
cdfc88b197915461e200617d3ab9fe7da575781fc6af26a5c7eb5e136b205f0e ./test/unit/fyToken/view/isFyToken.ts
a644f3213deca43de7f38ded7ee78b8ec8e0173ca6083c38f4eaae3f7614ae51 ./test/unit/fyToken/view/redemptionPool.ts
ec9ebda998847f11003761ec034f9f1b23600a78122c9b35d1e40906810e0e93 ./test/unit/fyToken/view/underlying.ts
1d96d5940b1e4a3cad4cdc90e471a4f497b116bc4d85066346b35adba1d52f79 ./test/unit/fyToken/view/underlyingPrecisionScalar.ts
485255dac39afee9cac0bf75fae3757febf0cdc5e853861f1c5ff4720d2155a4 ./test/unit/fyToken/effects/borrow.ts
260934d98690e133c389ef6948a9aa8846b73e9501af2cf3cc9355bd36959b65 ./test/unit/fyToken/effects/burn.ts
025b50aa915bcf458d67683b8a8c75d601e53d5c7ce46936c6a15237255489d8 ./test/unit/fyToken/effects/liquidateBorrow.ts
f82df5204598611101bf3a8799228235ab8eb2ce99c1a6d21a13374104f9ed0d ./test/unit/fyToken/effects/mint.ts
0757d4ee522fa85afcd8de49eed4c94e49b32331bce72abeeeeb69ee5d2654b7 ./test/unit/fyToken/effects/repayBorrow.ts
9e993d7aaf518170d891063eb4001dbf9069343a4b180522fc6bdcd2412a507c ./test/unit/fyToken/effects/repayBorrowBehalf.ts
88422faa157c047a2e978f27d9a3be0e0c22109fe5c904428912236698851024 ./test/unit/fyToken/effects/setFintroller.ts
de27d577b0a1ffb070ebb38bc5d0a267d5c331877c2e6fac07f336943ce1e6d8 ./test/unit/fyToken/constructor/index.ts
9018287b99651c7d00d079472d42e535e2a909350f65ce3a5a07c2b437ffc37d ./test/unit/fintroller/Fintroller.behavior.ts
9c8632e403b861ce7e2aba525154d8118310ff876c467df04cf0ca19c3c31ea5 ./test/unit/fintroller/Fintroller.ts
0cd309bf7695f481521ec75fa2badac586671e2403833ce38e9daba06cf88ff0 ./test/unit/fintroller/view/getBond.ts
d096922b229963512d4507753531a523acdea94fae2a7192dcc2430d61e1ac1f ./test/unit/fintroller/view/getBondCollateralizationRatio.ts
e316f68155f4e700801e4b2f219cfe3735ae64155c48f21698c6f31c96e6476f ./test/unit/fintroller/view/getBondDebtCeiling.ts
c5fa8d56be777109ff1bfce81fde64a4e32b534ac48205eb11277a8220c35aaf ./test/unit/fintroller/view/getBorrowAllowed.ts
cbe4b4be085a198404d371a0590fd1cefe05a7d000593de2ff38ebea2a492347 ./test/unit/fintroller/view/getDepositCollateralAllowed.ts
3cd8ca289c3ca066ef7d04e55cb289dc28c5a0fd3ace6f52ab55f3d98b7665e3 ./test/unit/fintroller/view/getLiquidateBorrowAllowed.ts
9a4132a28dd1d5e772292bdecd041c8aca528d665ebbe7659832d3d23fd3e891 ./test/unit/fintroller/view/getRedeemFyTokensAllowed.ts
b33d42b381d1534867fb42176639811e77a9e87ef3ea3892e20661302e453455 ./test/unit/fintroller/view/getRepayBorrowAllowed.ts
00d8d58376ad64ba62122a5f2467e6b180d0f86bce8d73545598d5c0ffb8980d ./test/unit/fintroller/view/getSupplyUnderlyingAllowed.ts
55a8ea47aeedf43ea9bda2f19353fe7b4d3f233af0d1e644d429be7ec6a622ec ./test/unit/fintroller/view/isFintroller.ts
```

```
e99970676160a2aa61e589b185f5404ead3d99fc0a0f1e67c670e86d55c21d5a ./test/unit/fintroller/view/liquidationIncentiveMantissa.ts
ceb8e88ff9c2e265a1dc18af9b163ca5fce83240f7e4ff403b6e6d9db7824226 ./test/unit/fintroller/view/oracle.ts
b313fa3e9caf0d34330d0c073594e3d4ec6707bda631dac5b2022b970b9d4b1d ./test/unit/fintroller/view/oraclePricePrecisionScalar.ts
cf0bc705e8a04f78e4103841f4647beadfe047149fdfa14443a24a02d934887e ./test/unit/fintroller/effects/listBond.ts
0d19e8630ca16f130451159ce582c292c07015b5298e30744161fb7c3b5c1354 ./test/unit/fintroller/effects/setBondDebtCeiling.ts
95a72f2ae61883b18300cc4de3ca8030064621d84b07f2e8a41a355dce357330 ./test/unit/fintroller/effects/setBorrowAllowed.ts
a347f7960c2da13a2db37376d336e7110c42432ab2a2f847cd2fbca025d32ef9 ./test/unit/fintroller/effects/setCollateralizationRatio.ts
8ee1a7c7fbeedff876e480acbd89258614b6aaa38146c8cdb7c251a49254f4fc ./test/unit/fintroller/effects/setDepositCollateralAllowed.ts
32f6d2e5edb690c76ee34435378680381ed4b9e42427efe5db1c8ae0821c7fd0 ./test/unit/fintroller/effects/setLiquidateBorrowAllowed.ts
d0eb2d714e44294fc8a4618face982e66db93ea2c8f172fe8c70b7f82c62f844 ./test/unit/fintroller/effects/setLiquidationIncentive.ts
9f860487e5eee47850deec45db17c3c6d1bfe4634f9683d36edf3915d1ae66a9 ./test/unit/fintroller/effects/setOracle.ts
89e7ab39604107dac2b7a2ae5b84d85a88434dbe77a9b691fa320306aab250c8 ./test/unit/fintroller/effects/setRedeemFyTokensAllowed.ts
4ca8c786ebd09d071148497c0cb2767e52e73d1b010d47fda7c5c64db76a3a2c ./test/unit/fintroller/effects/setRepayBorrowAllowed.ts
f19296c1ac7ffb701eecbcac7e128e7df1bd1914694437e6682bcf99cc54a87c ./test/unit/fintroller/effects/setSupplyUnderlyingAllowed.ts
3874932e98a53b826c38cbed7ea7372888b393fd4dbdf5f29db9b3d8b07171d1 ./test/unit/balanceSheet/BalanceSheet.behavior.ts
7cc57a74a6eba6453558dacfebc83f6b8568202704442e67b05970595e153312 ./test/unit/balanceSheet/BalanceSheet.ts
94d55863be3c2bc0ac9c9232e76cd3f7c63df529d5c71bc7ffa9cef22efc662e ./test/unit/balanceSheet/view/fintroller.ts
53aa22aa6d58db9b1409819ab11d7bf8ea613b3ef2bc33b8a846051f8a645594 ./test/unit/balanceSheet/view/getClutchableCollateral.ts
e628c42de807f6a07abec49b8246e3e4e96fa879de6ae85e9d223a3765f7fad7 ./test/unit/balanceSheet/view/getCurrentCollateralizationRatio.ts
48a45e716bcb5d0cf1acb95da971e77897031193cd2106ec4ec8d48b68299ac9 ./test/unit/balanceSheet/view/getHypotheticalCollateralizationRatio.ts
c2f5eb2c8e5574675617c34ea3ecd503d8f04f0375cb69c7259b7d9df57bd452 ./test/unit/balanceSheet/view/getVault.ts
32653c11e59409643ec9fa23a2477d3cae124d6d132d4aa01b781f92d608d9cf ./test/unit/balanceSheet/view/getVaultDebt.ts
77fb8fa92b6c8c92850a174d9ba08bb021919f60e047f47a670f8dc5c0cd9fca ./test/unit/balanceSheet/view/getVaultLockedCollateral.ts
d513258e92df077b864b6a02b668895e6241dec8e344c8145e2fb495a274718c ./test/unit/balanceSheet/view/isAccountUnderwater.ts
377b70450c86db2dbe63539e9a0bfa15c82b6c8acaada798b1468a2dc7b3797e ./test/unit/balanceSheet/view/isBalanceSheet.ts
1be1b1a0cca75caebe53af7f1f495b60035320e493319c5c4c7de7b577c6c889 ./test/unit/balanceSheet/view/isVaultOpen.ts
e0d385dc16c56c9bc12d2cfd8773ff9228a124231ab19dd5fb1c15dc6bd660eb ./test/unit/balanceSheet/effects/clutchCollateral.ts
6c8aacb9d374bcac3e68cf55137c7806274ce5ca7e2457b188d08388342dc50b ./test/unit/balanceSheet/effects/depositCollateral.ts
529a14443172ab9029c2601179c1fbd87c83231764967cd5a85ab1689d297585 ./test/unit/balanceSheet/effects/freeCollateral.ts
254ab41e521bf835f3638ae01c09895d80260e9aef1a1cc7ef04a064f7244e60 ./test/unit/balanceSheet/effects/lockCollateral.ts
f7572f28ea13476ee77788b2155492e679ec9702216c264f15d402c3395c34f0 ./test/unit/balanceSheet/effects/openVault.ts
3297fe2b32909b71d33c9be354fb6b2575eaa906ff75d2a7fd3a51cb927ad319 ./test/unit/balanceSheet/effects/setVaultDebt.ts
115f3f4628f28e97a036021e14277d57ed15b635c38b7d0a2919057796e31e75 ./test/unit/balanceSheet/effects/withdrawCollateral.ts
a70e0b8f28ae07ab8e39875e5d1eb82851df335857f78eda1e72bb787e406e39 ./test/integration/fixtures.ts
b38b98d782bce176e44051407b857920999c786c82180331163468995c2bbddd ./test/integration/index.ts
5fcf03f10b8733c2229daa3b32cca2e858b37b138c6aa61bf15ca9a13f52f645 ./test/integration/redemptionPool/RedemptionPool.behavior.ts
2e9514fbbd4de7b87723f9e68f89d6726afcdbbc9e939868680ff4f4269fda7b ./test/integration/redemptionPool/RedemptionPool.ts
5cee8859ed195c63a15db275c98e6c9e220224b3d3a8f6a7dc3f72b84fdb8efa ./test/integration/redemptionPool/effects/redeemFyTokens.ts
ed8352613f2bab55e28151943e71c10f02994e2a29a4d055792b0b2728b81eee ./test/integration/redemptionPool/effects/supplyUnderlying.ts
c7bf8e996aced830214cc181e24c5380f61a02b689581e09b43133fa6e31c2d9 ./test/integration/fyToken/FyToken.behavior.ts
e23f709b0bd8ed8e1365f7fad517ff02276e3496550a14bcd96f796fca73df40 ./test/integration/fyToken/FyToken.ts
453cb4a4b647a5f4b606b8f058b47d8456f688311800364ad316d693a21df85e ./test/integration/fyToken/effects/borrow.ts
6c1c605daf9584bb2ec794db1b523eeadf25a2fe5d0162b8fdc6c15fddfe435e ./test/integration/fyToken/effects/burn.ts
63c497abfdfdfb14675f07f51cd7ba058221dd8fc266164cc2f84179a6d0b945 ./test/integration/fyToken/effects/liquidateBorrow.ts
eab166babd4beb4261c1beb6fcf77c971b2b7fb7a00ce5fd5300c71a5cffabbd ./test/integration/fyToken/effects/mint.ts
627f02d31c9200db7f657c07f49e649539f195c15420001a0f0024db91fe28e9 ./test/integration/fyToken/effects/repayBorrow.ts
```

Changelog

- 2020-11-09 - Initial report [f77b7f3, 6c3ee4a]
- 2020-11-18 - Revised report [e1ea9b7, 6c3ee4a]
- 2020-11-24 - Revised report [abbc9fe, 6c3ee4a]
- 2020-12-02 - Revised report [abbc9fe, 6c3ee4a]

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.