Quantstamp Security Assessment Certificate

April 15th 2022 — Quantstamp Verified

Voltz Protocol

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

Executive Summary

Type	AMM for Interest Rate Swaps
Auditors	Kacper Bąk, Senior Research Engineer Poming Lee, Senior Research Engineer Roman Rohleder, Research Engineer
Timeline	2022-02-22 through 2022-04-13
EVM	London
Languages	Solidity, TypeScript
Methods	Architecture Review, Unit Testing, Computer-Aided Verification, Manual Review
Specification	Voltz v1 Technical Specification Voltz Protocol: An Automated Market Maker for Interest Rate Swaps
Documentation Quality	High
Test Quality	High

Source Code

Repository	Commit
voltz-core	3d87a27

Goals

• Are calculations implemented correctly and follow the specification?

• Are liquidations and margin requirements implemented correctly?

Total Issues	**8** (4 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	**1** (0 Resolved)
Low Risk Issues	**3** (2 Resolved)
Informational Risk Issues	**2** (1 Resolved)
Undetermined Risk Issues	**2** (1 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

During the course of this audit we have found a few issues spanning medium, low, informational, and undetermined severity levels. Notably, there is one issue that we classified as medium severity and it's related to the condition that triggers a margin requirement check. Furthermore, we included some of the findings reported by Slither. We recommend addressing all the findings. Overall, the code and the test suite appear to be of high quality, however, the project is very complex due to the large number of non-trivial calculations. Although we haven't found high-severity issues, we still consider the project risky in the sense that it is novel and contains intricate logic.
Update: the team has addressed all of the issues. During the course of the reaudit the team asked us to review `PR#110` and `PR#112` which are included in commit `56d1da5`. It is important to note, however, that besides the two PRs, no other new code has been reviewed.

ID	Description	Severity	Status
QSP-1	Incorrect Trigger Condition for `checkPositionMarginAboveRequirement()`	^ Medium	Acknowledged
QSP-2	Privileged Roles and Ownership	⌄ Low	Acknowledged
QSP-3	No Checks If Arguments Are Non-zero	⌄ Low	Fixed
QSP-4	Owner Can Set Arbitrary Fees	⌄ Low	Fixed
QSP-5	Unlocked Pragma	○ Informational	Fixed
QSP-6	Anyone Can Add Margin to Any Position	○ Informational	Acknowledged
QSP-7	The Function `grow()` Does Not Check that `next` Is < 65535	? Undetermined	Fixed
QSP-8	Updating a Position Without Making an Actual Token Transfer	? Undetermined	Acknowledged

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.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.8.2

Steps taken to run the tools:

1. Installed the Slither tool: `pip install slither-analyzer`
2. Run Slither from the project directory: `slither .`

Findings

QSP-1 Incorrect Trigger Condition for `checkPositionMarginAboveRequirement()`

Severity: *Medium Risk*

Status: Acknowledged

File(s) affected: `MarginEngine.sol`

Description: On `L409` in `MarginEngine.sol`, the function `checkPositionMarginAboveRequirement()` is invoked when the caller is trying to add some liquidity (i.e., when `params.liquidityDelta > 0`) to their `position` instead of when the liquidity gets removed (i.e., when `params.liquidityDelta < 0`). Failing to correctly check the required margin could lead to un-recoverable fund loss on the platform.

Recommendation: We recommend changing the condition.

Update: The team explained that they want to check the position margin requirement only after a mint (scenario where the liquidity delta is positive) since in that scenario the LP is effectively agreeing to enter into future swaps that cross their active tick range. Burning liquidity means the position will not enter into future swaps (will have no active liquidity in the respective tick range), however their past positions will still mean that they need to have sufficient margin to support them.

QSP-2 Privileged Roles and Ownership

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `rate_oracles/BaseRateOracle.sol`, `VAMM.sol`, `MarginEngine.sol`, `Factory.sol`

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract. Specifically, there are some actions that could have important consequences for end-users:

1. The owner of `contracts/Factory.sol` can call owner-only functions in `contracts/MarginEngine.sol` and manipulate all the parameters at any time at will.
2. The owner of `contracts/Factory.sol` can call owner-only functions in `contracts/VAMM.sol` and manipulate all the parameters at any time at will.
3. The owner of `contracts/Factory.sol` can call owner-only functions in `contracts/MarginEngine.sol` and change the external contracts used by the `MarginEngine` at any time at will.
4. The owner of `rate_oracles/BaseRateOracle.sol` can call owner-only functions.

Recommendation: 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.

Update: The team explained that the intention is to initially start off with a multisig being the owner, but then eventually transition into a DAO which would manage critical protocol decisions via a Governor contract coupled with a Timelock. They will be clearly communicating this to the community and users.

QSP-3 No Checks If Arguments Are Non-zero

Severity: *Low Risk*

Status: Fixed

File(s) affected: `rate_oracles/BaseRateOracle.sol`, `rate_oracles/AaveRateOracle.sol`, `AaveFCM.sol`, `Factory.sol`

Description: There are no checks if arguments are non-zero in the following functions:

1. `BaseRateOracle.constructor()`,
2. `AaveRateOracle.constructor()`,
3. `AaveFCM.initialize()`,
4. `Factory.constructor()`.

Recommendation: Add relevant checks.

QSP-4 Owner Can Set Arbitrary Fees

Severity: *Low Risk*

Status: Fixed

File(s) affected: `VAMM.sol`

Description: Owner can set arbitrary fees via the functions `setFeeProtocol()` and `setFee()`.

Recommendation: We recommend adding an upper limit on how high the fees can be.

QSP-5 Unlocked Pragma

Severity: *Informational*

Status: Fixed

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-6 Anyone Can Add Margin to Any Position

Severity: *Informational*

Status: Acknowledged

Description: On `L281` of `MarginEngine.sol`, the code allows anyone to add margin to any position. There is a possibility that this could be utilized by an attacker to manipulate the system in order to conduct a more complex attack.

Recommendation: Apply the principle of least privilege. We recommend leaving as little room as possible for random users to have impact on system's components. Unless there is a good reason not to, restrict the function called to be either the `_owner` or `factory.isApproved(_owner, msg.sender)`.

Update: The team informed us that they made the decision to allow anyone to add margin to any position in order to enable, e.g., an insurance eligible by Voltz to re-collateralise undercollateralised positions.

QSP-7 The Function `grow()` Does Not Check that `next` Is < 65535

Severity: *Undetermined*

Status: Fixed

File(s) affected: `rate_oracles/OracleBuffer.sol`

Description: The function `grow()` does not check that `next` is < 65535. Consequently, an out of bounds error may occur.

Recommendation: Add a relevant check to handle this boundary case.

QSP-8 Updating a Position Without Making an Actual Token Transfer

Severity: *Undetermined*

Status: Acknowledged

File(s) affected: `VAMM.sol`

Description: The functions `mint()` and `burn()` can be used to modify position's margin without calling `updatePositionMargin()` that transfers in or out the required tokens. This could be used by attackers to potentially manipulate the internal state of the contract which may lead to unexpected results.

Recommendation: Unless this is an intended design, perhaps a fix would be to call `updatePositionMargin()` from `mint()` and `burn()` to complete the process in a single transaction.

Update: The team explained that they already do token transfers in the periphery (Periphery.sol). They have a preference for keeping margin related logic in the margin engine and pricing logic in the vAMM.

Automated Analyses

Slither

Slither reported the following:

1. Uninitialized fields in `AaveFCM.sol#41` and `rate_oracles/BaseRateOracle.sol#40`. We classified it as a false positive.
2. Multiplication before division in:
    1. `core_libraries/MarginCalculator.sol#66-71`,
    2. `core_libraries/Tick.sol#42-45`, and
    3. `FullMath.mulDiv()`.

3. Reentrancy in from externally called functions:
    1. `MarginEngine.getPositionMarginRequirement()`,
    2. `BaseRateOracle.increaseObservarionCardinalityNext()`, and
    3. `VAMM.swap()`.

4. Ignored return values in:
    1. `AaveFCM.sol#302`, and
    2. `Periphery.sol#123`.

Code Documentation

1. `MarginEngine.sol`: `L445`: typo `trads`.
2. `MarginEngine.sol`: `L445`: typo `positioin`.
3. `MarginEngine.sol`: `L447`: typo "`hsi`.
4. `MarginEngine.sol`: `L518`: typo `positon`.

5. `core_libraries/FixedAndVariableMath.sol`: `L109`: the comment `/// @param excessBalanceWad Any excess balance from the variable side of the position` appears incorrect since this is for interests accumulated from both fixed and variable, instead of only from variable side.
6. `AaveFCM.sol`: `L129`: the comment `(in terms of the underlyingToken e.g. aUSDC)` should be `(in terms of the underlyingToken e.g. 'USDC')` instead.
7. `interfaces/IFCM.sol`: `L12`: typos in `100aTokens` and `vashflows`.
8. `MarginCalculator.computeApyBound()` missing NatSpec comment for parameter `_marginCalculatorParameters`.
9. `MarginCalculator.worstCaseVariableFactorAtMaturity()` missing NatSpec comment for parameter `_marginCalculatorParameters`.
10. Inconsistent NatSpec parameter comments for `MarginCalculator.getAbsoluteFixedTokenDeltaUnbalancedSimulatedUnwind()` (missing parameter `fixedRateDeviationMinWad` and missing return comment).
11. `SwapMath.computeSwapStep()` missing NatSpec comment for return value `feeAmount`.
12. `Tick.update()` missing NatSpec comments for parameters `fixedTokenGrowthGlobalX128` and `variableTokenGrowthGlobalX128`.
13. `Tick.cross()` missing NatSpec comments for parameters `fixedTokenGrowthGlobalX128` and `variableTokenGrowthGlobalX128`.
14. `AaveRateOracle.writeRate()` missing NatSpec comments for return values `indexUpdated` and `cardinalityUpdated`.
15. `OracleBuffer.initialize()` missing NatSpec comment for parameter `observedValue`.

Adherence to Best Practices

1. Instead of using boolean values passed as function arguments use enumerations to improve readability. For example, `Tick.update()` could use `LOWER` and `UPPER` instead of `true` and `false` to improve readability in `VAMM.sol`. **Update:** acknowledged.
2. Remaining TODO items in (**Update:** fixed):
    1. `MarginEngine.sol`: `L214`,
    2. `core_libraries/Tick.sol`: `L172` and `L223`,
    3. `VAMM.sol`: `L432`,
    4. `rate_oracles/AaveRateOracle.sol`: `L121`,
    5. `rate_oracles/BaseRateOracle.sol`: `L57`.
3. `core_libraries/FixedAndVariableMath.sol`: `L109` says that `excessBalanceWad` should be `Any excess balance from the variable side of the position`; however, when function `calculateFixedTokenBalance()` is called in `L223`, the `excessBalanceWad` passed into the function, is the sum of both interests accrued from fix token side and variable token side. Please make sure that this is intended. **Update:** fixed.
4. As solidity version >= 0.8.0 is used, which implicitly implements safe arithmetic, the import and use of OpenZeppelins `SafeMath` library in `AaveRateOracle.sol` may be omitted. **Update:** fixed.

Test Results

Test Suite Results

Apart from `scenario*` tests, all other tests executed successfully.

```
  - scenario 12 (apy sims)
  - scenario 12 (apy sims)
  - scenario 12 (apy sims)
  - scenario 0
  - scenario 0
  - scenario 1
  - scenario 1
  - scenario 10
  - scenario 11
  - scenario 13
  - scenario 14
  - scenario 2
  - scenario 3
  - scenario 4
  - scenario 6
  - scenario 7
  - scenario 8
  FixedAndVariableMath
    #calculateSettlementCashflow
       ✓ correctly calculates the settlement cash flow
       ✓ correctly calculates the settlement cash flow
excess: 19788219178082192000
fixedFactor: 383561643835616
cashflow:  19596438356164384000
       ✓ scenario (42ms)
    #accrualFact
       ✓ takes 4000000000000000000 and returns the correct value
       ✓ takes 5000000000000000000 and returns the correct value
       ✓ takes 345360000000000000000000000 and returns the correct value
       ✓ takes 345370000000000000000000000 and returns the correct value
    #fixedFactor
       ✓ returns the correct fixed factor at maturity
       ✓ returns the correct fixed factor before maturity
       ✓ returns the correct fixed factor at maturity for 2 weeks
       ✓ returns the correct fixed factor before maturity for 2 weeks
       ✓ reverts: end <= start
       ✓ reverts: current < start
    #calculateFixedTokenBalance
       ✓ reverts unless the end timestamp is after the start timestamp
fixedFactor:  575342465753424
       ✓ correctly calculates the fixed token balance
       ✓ correctly calculates the fixed token balance
       ✓ correctly calculates the fixed token balance
       ✓ balance = amount0 when excess is 0
    #getExcessBalance
       ✓ correctly calculates the excess balance
       ✓ correctly calculates the excess balance
       ✓ correctly calculates the excess balance
       ✓ correctly calculates the excess balance
    #getFixedTokenBalance
       ✓ correctly gets the fixed token balance
    full scenarios
currentBlockTimestamp 1649868086000000000000000000
termEndTimestamp 1651077686000000000000000000
amount0Rebalanced TS -265714269179894495459 16
realizedCashflow 19808219812278031000
       ✓ scenario 1

  Position
    #updateLiquidity
       ✓ reverts if liquidity delta is zero (163ms)
       ✓ correctly updates the liqudity of a position (41ms)
    #updateMargin
       ✓ correctly updates the margin of a position
    #updateBalances
       ✓ correctly updates the variable and fixed token balances of a position
    #updateFixedAndVariableTokenGrowthInside
       ✓ check the inside last balances are correctly updated
    #feeGrowthInside
       ✓ check feeGrowthInsideLast correctly updated
    #calculateFixedAndVariableDelta
```

```
      ✓ check fixed and variable deltas are correctly calculated
    #calculateFeeDelta
      ✓ check fee delta correctly calculated

  SwapMath
    #computeFeeAmount
      ✓ computeFeeAmount
      ✓ no notional
      ✓ no time until maturity
      ✓ no percentage fee
      ✓ overflow if notional >= 10**60
    #computeSwapStep
      ✓ exact amount in that gets capped at price target in one for zero
      ✓ exact amount out that gets capped at price target in one for zero
      ✓ exact amount in that is fully spent in one for zero
      ✓ exact amount out that is fully received in one for zero
      ✓ amount out is capped at the desired amount out
      ✓ target price of 1 uses partial input amount

  Tick
    #checkTicks
      ✓ returns checks of both ticks
    #getFeeGrowthInside
      ✓ two unitialized ticks
      ✓ between two initialized ticks
      ✓ after two initialized ticks
    #getVariableTokenGrowthInside
      ✓ two unitialized ticks
      ✓ between two initialized ticks
      ✓ after two initialized ticks
      ✓ before two initialized ticks
    #getFixedTokenGrowthInside
      ✓ two unitialized ticks
      ✓ between two initialized ticks
      ✓ after two initialized ticks
      ✓ before two initialized ticks
    #cross
      ✓ flips the growth variables
      ✓ two flips no op
    #update
      ✓ does not flip from nonzero to greater nonzero
      ✓ flips from zero to nonzero
      ✓ flips from nonzero to zero
      ✓ exceed max liquididty
      ✓ tick <= tickCurrent
      ✓ tick > tickCurrent
      ✓ do not update globals if initialized
      ✓ subtract net liquidity when upper tick
    #clear
      ✓ deletes all the data in the tick

  TickBitmap
    #isInitialized
      ✓ is false at first
      ✓ is flipped by #flipTick
      ✓ is flipped back by #flipTick
      ✓ is not changed by another flip to a different tick
      ✓ is not changed by another flip to a different tick on another word
    #flipTick
      ✓ flips only the specified tick (65ms)
      ✓ reverts only itself (63ms)
      ✓ gas cost of flipping first tick in word to initialized
      ✓ gas cost of flipping second tick in word to initialized
      ✓ gas cost of flipping a tick that results in deleting a word
    #nextInitializedTickWithinOneWord
      lte = false
        ✓ returns tick to right if at initialized tick
        ✓ returns tick to right if at initialized tick
        ✓ returns the tick directly to the right
        ✓ returns the tick directly to the right
        ✓ returns the next words initialized tick if on the right boundary
        ✓ returns the next words initialized tick if on the right boundary
        ✓ returns the next initialized tick from the next word
        ✓ does not exceed boundary
        ✓ skips entire word
        ✓ skips half word
        ✓ gas cost on boundary
        ✓ gas cost just below boundary
        ✓ gas cost for entire word
      lte = true
        ✓ returns same tick if initialized
        ✓ returns tick directly to the left of input tick if not initialized
        ✓ will not exceed the word boundary
        ✓ at the word boundary
        ✓ word boundary less 1 (next initialized tick in next word)
        ✓ word boundary
        ✓ entire empty word
        ✓ halfway through empty word
        ✓ boundary is initialized
        ✓ gas cost on boundary
        ✓ gas cost just below boundary
        ✓ gas cost for entire word

  Factory
      ✓ Cannot deploy if not the owner
      ✓ Deploys and initializes proxies successfully (193ms)
      ✓ sets master margin engine
      ✓ upgrades the implementation of a proxy

  FCM
    #fcm
traderBalanceInATokens 100.0
traderBalanceInUnderlyingTokens 0.000001
traderOverallBalanceInUnderlyingTokens 100.000001
allowance 100000000000000000000
marginInScaledYieldBearingTokens:  100.0
fixedTokenBalance:  100.498644104512717841
variableTokenBalance:  -100.0
isSettled:  false

historicalApyCached 0.081289727264422592
aTokenBalanceOfFCM 100.15
aTokenBalanceOfTrader 100.15
marginInScaledYieldBearingTokens:  0.0
fixedTokenBalance:  0.0
variableTokenBalance:  0.0
isSettled:  true

traderBalanceInATokens 100.019277683783450319
traderBalanceInUnderlyingTokens 0.000001
traderOverallBalanceInUnderlyingTokens 100.019278683783450319
traderAbsoluteReturn 0.019277683783450319
traderReturn 0.000192776837834503
traderAPY 0.010108392141464796
finalPositionMargin 1.252573107796277129
here?
traderBalanceInATokens 0.0
traderBalanceInUnderlyingTokens 1000000000.130722316216549678
traderOverallBalanceInUnderlyingTokens 1000000000.130722316216549678
traderAbsoluteReturn 0.130722316216549678
traderReturn 0.001307223162165496
traderAPY 0.070539892935308311
positionAPY 0.070539892935308311
       ✓ scenario1 (417ms)

  MarginCalculator
    #computeTimeFactor
      ✓ reverts if currentTimestamp is larger than termEndTimestamp
      ✓ correctly computes the time factor
    #computeApyBound
      ✓ correctly computes the Upper APY Bound
      ✓ correctly computes the Upper APY Bound
      ✓ correctly computes the Lower APY Bound
    #worstCaseVariableFactorAtMaturity
      ✓ correctly calculates the worst case variable factor at maturity FT, LM
      ✓ correctly calculates the worst case variable factor at maturity FT, IM
      ✓ correctly calculates the worst case variable factor at maturity VT, LM
      ✓ correctly calculates the worst case variable factor at maturity VT, IM
    #getTraderMarginRequirement
      ✓ correctly calculates the margin requirement: FT, LM (42ms)
      ✓ correctly calculates the margin requirement: FT, LM with <0 (44ms)
      ✓ correctly calculates the margin requirement: FT, LM with <0 (46ms)
      ✓ correctly calculates the margin requirement: VT, LM (44ms)
      ✓ correctly calculates the margin requirement: FT, IM (44ms)
    #getPositionMarginRequirement
position margin requirement 0.110927648378902124
       ✓ current tick < lower tick: margin requirement for upper tick | no minimum (66ms)
position margin requirement 0.288637791189893257
       ✓ current tick < lower tick: margin requirement for lower tick | minimum (72ms)
position margin requirement 0.332524547647427475
       ✓ lower tick < current tick < upper tick: margin requirement for upper tick | no minimum (105ms)
position margin requirement 0.132907391350678334
       ✓ lower tick < current tick < upper tick: margin requirement for lower tick | minimum (288ms)
position margin requirement 0.132907391350678334
       ✓ upper tick < current tick: margin requirement for lower tick | minimum (69ms)
position margin requirement 0.221596901319562816
```

```
       ✓ upper tick < current tick: margin requirement for upper tick | no minimum (94ms)
       ✓ case 1 of mc and fcm pdf (52ms)
position margin requirement 376.682544773587003569
       ✓ case 2 of mc and fcm pdf (67ms)
       ✓ correctly checks for the fact the position is liquidatable (66ms)
       ✓ correctly checks for the fact the position is not liquidatable (67ms)

  MarginEngine
 min_tick : -69100
min_price : 2503036416286949174936592462
 max_tick : 69100
max_price : 250779481055183781714411595774O
    #updatePositionMargin
       ✓ reverts if margin delta is zero
    #positions
       ✓ returns empty position by default
    #checkPositionMargin
       ✓ check position margin above requirement reverted (49ms)
       ✓ check position margin can be updated reverted (82ms)
    #checkTraderMargin
       ✓ check position margin above requirement
       ✓ check trader margin can be updated
    #updatePositionMargin
       ✓ check position margin correctly updated scenario 1
       ✓ check trader margin correctly updated scenario 2 (38ms)
       ✓ check token balance correctly updated
    #updatePositionMargin
       ✓ correctly updates position margin (internal accounting)
       ✓ check token balance correctly updated
    #unwindPosition
variable token balance: -10.000000000000000002
variable token balance: 0.0
       ✓ liquidate and unwind LP (277ms)
       ✓ not enough liquidity to unwind position (223ms)
       ✓ unwinds Trader (112ms)
    #collectProtocol
       ✓ checkOwnerPrivilege
       ✓ checkCollectProtocol
    #updatePositionTokenBalances
       ✓ correctly updates position token balances (growth inside last)
       ✓ correctly updates position token balances (fixed and variable deltas)
    #settleTrader
       ✓ reverts before maturity
       ✓ correctly updates position balances (38ms)
       ✓ correctly updates position margin
    settle position
       ✓ scenario 1: position settlement (266ms)
    liquidations
       ✓ scenario1: simple trader liquidation (194ms)
oldBalanceOther 100000000000000000000000000099989999900000100000000000O
balabalanceWalletnceOther 1000000000000000000000000000999899999100000999999999990
       ✓ scenario 2: simple position liquidation (283ms)
    #getHistoricalApy[ReadOnly]
       ✓ correctly computes historical apy without cache (45ms)
       ✓ correctly caches historical apy (92ms)

  Aave Rate Oracle
    #initialize
Deploy rate oracle for mocked {token, aave}: {0x2f8D338360D095a72680A943A22fE6a0d398a0B4, 0x21915b79E1d334499272521a3508061354D13FF0}
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 101
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 201
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 301
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 401
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 501
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 601
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 701
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 801
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 901
Increased size of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011's buffer to 1000
Updated minSecondsSinceLastUpdate of 0x1c39BA375faB6a9f6E0c01B9F49d488e101C2011 to 3600
       ✓ rateIndex, rateCardinality, rateCardinalityNext correctly initialized
    #grow
       ✓ increases the cardinality next for the first call
       ✓ is no op if oracle is already gte that size
    #write
       ✓ single element array gets overwritten
       ✓ grows cardinality if writing past (50ms)
       ✓ does not grow cardinality if insufficient time has passed (53ms)
    #getRateFromTo
       ✓ correctly calculates rate from one timestamp to the next (50ms)
       ✓ correctly calculates rate from one timestamp to the next when there was no recent write (529ms)
    #getApyFromTo
       ✓ correctly calculates APY between two known, consecutive data points
       ✓ correctly calculates APY between two known, NON-consecutive data points
       ✓ correctly calculates APY for intervals IN BETWEEN consecutive data points
       ✓ correctly calculates APY for intervals across data points
    #interpolateRateValue
       ✓ interpolateRateValue(1, 1.00000001, 86400)
       ✓ interpolateRateValue(1, 1.0000000001, 3600)
       ✓ interpolateRateValue(1, 1.000000000001, 60)
       ✓ interpolateRateValue: 1 second at insanely low interest
       ✓ interpolateRateValue: zero interest
       ✓ interpolateRateValue: zero interest (one second)
       ✓ interpolateRateValue(1, 1.000000001, 1)
       ✓ interpolateRateValue(50, 1.000000001, 86400)
       ✓ interpolateRateValue(50, 1.000000001, 1)
       ✓ interpolateRateValue: 3 months
       ✓ interpolateRateValue(2, 1.000001, 1)
       ✓ interpolateRateValue: 3 months @ insanely high interest
    #binarySearch
       ✓ binary search works as expected (47ms)
    #getSurroundingRates
       ✓ target is beforeOrAt
       ✓ target is atOrAfter
       ✓ target is in the middle
       ✓ fails if target is too old
    #computeApyFromRate
       ✓ correctly computes apy
    #variableFactor
       ✓ calculate variable factor at maturity
       ✓ calculate variable factor after maturity
       ✓ calculates variable factor before maturity
       ✓ reverts if termStartTimestamp is too old

  VAMM
    #swapAndBurn
PFTB -1005982702720117618
PVTB 999999999999999997
TFTB 1005982702720117617
TVTB -1000000000000000000
sumOfTraderFixedTokenBalanceAndPositionFixedTokenBalance -1
BigNumber { _hex: '-0x01', _isBigNumber: true } sumOfTraderFixedTokenBalanceAndPositionFixedTokenBalance
       ✓ scenario1 (214ms)
PFTB 2 994003528608217623
PVTB 2 -1000000000000000001
TFTB 2 -994003528608217626
TVTB 2 1000000000000000000
sumOfTraderFixedTokenBalanceAndPositionFixedTokenBalance 2 -3
       ✓ scenario 2:  (269ms)
FATLP 958880327245098
       ✓ scenario 3: check fees (no protocol fees) (196ms)
FATLP 639252494503665
       ✓ scenario 4: check fees (with protocol fees) (206ms)
FATLP 958880327245098
FIBT 958880327245100
       ✓ scenario 5: check fees accrued = fees incurred (233ms)

  VAMM
    #quickChecks
       ✓ check the current tick
    #initialize
       ✓ fails if already initialized
       ✓ fails if starting price is too low
       ✓ fails if starting price is too high
       ✓ can be initialized at MIN_SQRT_RATIO
    #mint
       ✓ fails if not initialized
      after initialization
        failure cases
          ✓ fails if tickLower greater than tickUpper
          ✓ fails if tickLower less than min tick
          ✓ fails if tickUpper greater than max tick
          ✓ fails if amount exceeds the max (115ms)
          ✓ fails if total amount at tick exceeds the max (115ms)
        success cases
          ✓ initial tick
          ✓ adds liquidity to liquidityGross (419ms)
          ✓ removes liquidity from liquidityGross (186ms)
    #checkCurrentTimestampTermEndTimestampDelta
       ✓ check checkCurrentTimestampTermEndTimestampDelta
       ✓ check checkCurrentTimestampTermEndTimestampDelta
       ✓ check checkCurrentTimestampTermEndTimestampDelta
    #updateProtocolFees
       ✓ check MarginEngine privilege
       ✓ check not enough Protocol Fees
       ✓ check updateProtocolFees (41ms)
```

```
    #setFeeProtocol
       ✓ check owner privilege
       ✓ check setFeeProtocol
    #setFee
       ✓ check owner privilege
       ✓ check setFee
    #burn
       ✓ fails if not initialized
      after initialization
        failure cases
           ✓ fails if tickLower greater than tickUpper
           ✓ fails if tickLower less than min tick
           ✓ fails if tickUpper greater than max tick
        success cases
           ✓ initial tick
           ✓ adds liquidity to liquidityGross (282ms)

  Periphery
     ✓ set lp notional cap works as expected with margin engine owner
     ✓ set lp notional reverts if invoked by a non-owner
     ✓ check can't mint beyond the notional cap (135ms)
     ✓ check can't mint beyond the notional cap (286ms)
     ✓ alpha state can only be set by the owner of the vamm
     ✓ can't mint or burn with vamm when alpha but can mint with periphery
VM Exception while processing transaction: reverted with reason string 'TLU'
          margin requirement: 0
                 tick before: 0
                  tick after: 0
           fixed token delta: 0
        variable token delta: 0
                 fee incurred: 0
fixed token delta unbalanced: 0
     ✓ swap quoter on revert: margin requirement not met (154ms)
          margin requirement: 1.8717709376472416
                 tick before: 0
                  tick after: 0
           fixed token delta: -10009.745194930401
        variable token delta: 10000
                 fee incurred: 0
fixed token delta unbalanced: -10010.01001001001
     ✓ swap quoter on success (272ms)
VM Exception while processing transaction: reverted with reason string 'TLU'
          margin requirement: 0
                 tick before: 0
                  tick after: 0
           fixed token delta: 0
        variable token delta: 0
                 fee incurred: 0
fixed token delta unbalanced: 0
     ✓ swap quoter on different revert (129ms)
          margin requirement: 5.600456495684336
     ✓ mint quoter on revert (118ms)
on success
          margin requirement: 5.600437970708825
     ✓ mint quoter on success (126ms)
on success
          margin requirement: 5.600446389197259
     ✓ mint quoter from vamm (113ms)
on revert
Error: VM Exception while processing transaction: reverted with custom error 'LiquidityDeltaMustBePositiveInMint(0)'
    at TestVAMM.checkIsAlpha (contracts/VAMM.sol:57)
    at TestVAMM.mint (contracts/VAMM.sol:348)
    at VoltzERC1967Proxy._delegate (@openzeppelin/contracts/proxy/Proxy.sol:31)
    at VoltzERC1967Proxy._fallback (@openzeppelin/contracts/proxy/Proxy.sol:60)
    at Periphery.mintOrBurn (contracts/periphery/Periphery.sol:146)
    at runMicrotasks (<anonymous>)
    at processTicksAndRejections (node:internal/process/task_queues:93:5)
    at runNextTicks (node:internal/process/task_queues:62:3)
    at listOnTimeout (node:internal/timers:524:9)
    at processTimers (node:internal/timers:498:7)
    at async HardhatNode.runCall (/Users/kbak/Downloads/voltz_protocol-voltz-core-3f22c5e-github/node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:534:20)
    at async EthModule._callAction (/Users/kbak/Downloads/voltz_protocol-voltz-core-3f22c5e-github/node_modules/hardhat/src/internal/hardhat-network/provider/modules/eth.ts:353:9)
          margin requirement: 0
     ✓ mint quoter on different revert
on revert
          margin requirement: 5.600437126695596
     ✓ update position margin on revert (203ms)
     ✓ approvals work as expected
lpVariableTokenBalance 5.007499619400846835
traderVariableTokenBalance -5.007499619400846838
     ✓ minting via periphery (237ms)
lpVariableTokenBalance 2.503749809700423415
traderVariableTokenBalance -2.503749809700423419
     ✓ burning via periphery (266ms)
lpVariableTokenBalance 5.007499619400846835
traderVariableTokenBalance -5.007499619400846838
     ✓ swapping via periphery (248ms)

  LiquidityMath
    #addDelta
       ✓ 1 + 0
       ✓ 1 + -1
       ✓ 1 + 1
       ✓ 2**128-15 + 15 overflows
       ✓ 0 + -1 underflows
       ✓ 3 + -4 underflows

  SqrtPriceMath
    #getAmount0Delta
       ✓ returns 0 if liquidity is 0
       ✓ returns 0 if prices are equal
       ✓ returns 0.1 amount1 for price of 1 to 1.21
       ✓ works for prices that overflow
    #getAmount1Delta
       ✓ returns 0 if liquidity is 0
       ✓ returns 0 if prices are equal
       ✓ returns 0.1 amount1 for price of 1 to 1.21
    #getAmountsSequentially
      below amount 0 3004.354062741925653978
accumulated amount 0 3004.354062741925653948

      below amount 1 2995.354955910780937674
accumulated amount 1 2995.354955910780937647

full amount 0 5999.709018652706591653
full amount 1 5999.709018652706591653

       ✓ returns 0 if liquidity is 0 (513ms)

  TickMath
    #getSqrtRatioAtTick
       ✓ throws for too low
       ✓ throws for too low
       ✓ min tick
       ✓ min tick +1
       ✓ max tick - 1
       ✓ max tick
      tick -50
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick 50
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick -100
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick 100
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick -250
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick 250
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick -500
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick 500
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick -1000
         ✓ is at most off by 1/100th of a bips
         ✓ result
         ✓ gas
      tick 1000
         ✓ is at most off by 1/100th of a bips
         ✓ result
```

```
      ✓ gas
    tick -2500
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick 2500
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick -3000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick 3000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick -4000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick 4000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick -5000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick 5000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick -50000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
    tick 50000
      ✓ is at most off by 1/100th of a bips
      ✓ result
      ✓ gas
  #MIN_SQRT_RATIO
    ✓ equals #getSqrtRatioAtTick(MIN_TICK)
  #MAX_SQRT_RATIO
    ✓ equals #getSqrtRatioAtTick(MAX_TICK)
  #getTickAtSqrtRatio
    ✓ throws for too low
    ✓ throws for too high
    ✓ ratio of min tick
    ✓ ratio of min tick + 1
    ✓ ratio of max tick - 1
    ✓ ratio closest to max tick
    ratio 2503036416286949174936592463
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 990352031428304219919299379 2
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 280113854873930699593659691 13
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 560227709747861399187319382 27
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 792281625142643375935439503 36
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 112045541949572279837463876454
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 224091083899144559674927752909
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 633825300114114700748351602688
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas
    ratio 250779481055183781714411595773 9
      ✓ is at most off by 1
      ✓ ratio is between the tick and tick+1
      ✓ result
      ✓ gas


Snapshot Summary
 › 39 snapshots obsolete from 4 test suites. To remove them all, re-run mocha with `--update` to update them.
   ✓ test/core_libraries/swapMath.ts
     • SwapMath #computeSwapStep gas swap one for zero exact in capped 1
     • SwapMath #computeSwapStep gas swap one for zero exact in partial 1
     • SwapMath #computeSwapStep gas swap one for zero exact out capped 1
     • SwapMath #computeSwapStep gas swap one for zero exact out partial 1
     • SwapMath #computeSwapStep gas swap zero for one exact in capped 1
     • SwapMath #computeSwapStep gas swap zero for one exact in partial 1
     • SwapMath #computeSwapStep gas swap zero for one exact out capped 1
     • SwapMath #computeSwapStep gas swap zero for one exact out partial 1
   ✓ test/main_contracts/marginCalculator.ts
     • MarginCalculator #computeApyBound correctly computes the Upper APY Bound 1
   ✓ test/utils/liquidityMath.ts
     • LiquidityMath #addDelta gas add 1
     • LiquidityMath #addDelta gas sub 1
   ✓ test/utils/tickMath.ts
     • TickMath #getSqrtRatioAtTick tick -150000 gas 1
     • TickMath #getSqrtRatioAtTick tick -150000 result 1
     • TickMath #getSqrtRatioAtTick tick -250000 gas 1
     • TickMath #getSqrtRatioAtTick tick -250000 result 1
     • TickMath #getSqrtRatioAtTick tick -500000 gas 1
     • TickMath #getSqrtRatioAtTick tick -500000 result 1
     • TickMath #getSqrtRatioAtTick tick -738203 gas 1
     • TickMath #getSqrtRatioAtTick tick -738203 result 1
     • TickMath #getSqrtRatioAtTick tick 150000 gas 1
     • TickMath #getSqrtRatioAtTick tick 150000 result 1
     • TickMath #getSqrtRatioAtTick tick 250000 gas 1
     • TickMath #getSqrtRatioAtTick tick 250000 result 1
     • TickMath #getSqrtRatioAtTick tick 500000 gas 1
     • TickMath #getSqrtRatioAtTick tick 500000 result 1
     • TickMath #getSqrtRatioAtTick tick 738203 gas 1
     • TickMath #getSqrtRatioAtTick tick 738203 result 1
     • TickMath #getTickAtSqrtRatio ratio 79228162514264337593543 gas 1
     • TickMath #getTickAtSqrtRatio ratio 79228162514264337593543 result 1
     • TickMath #getTickAtSqrtRatio ratio 792281625142643375935439 50 gas 1
     • TickMath #getTickAtSqrtRatio ratio 792281625142643375935439 50 result 1
     • TickMath #getTickAtSqrtRatio ratio 2503036416286949174936592462 gas 1
     • TickMath #getTickAtSqrtRatio ratio 2503036416286949174936592462 result 1
     • TickMath #getTickAtSqrtRatio ratio 792281625142643375935439503 36000 gas 1
     • TickMath #getTickAtSqrtRatio ratio 792281625142643375935439503 36000 result 1
     • TickMath #getTickAtSqrtRatio ratio 792281625142643375935439503 36000000 gas 1
     • TickMath #getTickAtSqrtRatio ratio 792281625142643375935439503 36000000 result 1
     • TickMath #getTickAtSqrtRatio ratio 1461446703485210103287273052203988822378723970341 gas 1
     • TickMath #getTickAtSqrtRatio ratio 1461446703485210103287273052203988822378723970341 result 1

  367 passing (31s)
  17 pending
```

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

abd45dd9843ca8eb14b688114fa944a367b23bf6fa81c319a2142d9f4cedafc2 ./contracts/Factory.sol

9fb0bdcee2ee261d8d9d3675c23d2ecc6bf297045c5898d248dce6b92534fd52 ./contracts/VAMM.sol

8c5802208c819ebfa1c3ceeb14d7321aa8d28af29a1eb24f4c1e2051530a47b0 ./contracts/AaveFCM.sol

8f4b2bbb39db11dfa37865292b6642f71292ef5d13812f0ebcfde0120ff9298c ./contracts/MarginEngine.sol

48137a4e5904fb8bc48ad8a7fbcd4fd0f794bc891f87593f01b05a488c5a3ee4 ./contracts/interfaces/IFactory.sol

8f86b56a3e0b8262fa832491c4112c0f3342cac2848674d7f4165eb2824e84b0 ./contracts/interfaces/IPositionStructs.sol

b521403d93d69ec5350259824bbccdc3ca5a23c0d8d36bde2c93abd77d6c7087 ./contracts/interfaces/IERC20Minimal.sol

f64aa0885794f0dc6e0381a57851e32f8b21b144b3382d037c401ba4b8633c03 ./contracts/interfaces/IMarginEngine.sol

8ec05db311fc465b343f3a23c915efc54e08f2635f563e0fb55120a07e0f81c5 ./contracts/interfaces/IPeriphery.sol

0546cd3ebc98187a1490e3c578d9ffd5e7bca969bd9dcb67c447b75f34974475 ./contracts/interfaces/IVAMM.sol

52b06589101e8c40f8a46d90339920d58b4f3157af248dab1bbff40b55cc26c8 ./contracts/interfaces/rate_oracles/IAaveRateOracle.sol

b8085231e1ef584a6099d7b27acdb2eba215a90d2112e03ee2ba2ddf16fc519a ./contracts/interfaces/rate_oracles/IRateOracle.sol

18534fd5bed5d3330e82104d0cb3a2e008b2a0b77f53f11951758e1aa23cbba7 ./contracts/interfaces/aave/IAToken.sol

43e4fd109be05dfa9856ee2e49fd46a769c36afed6d64be28350ce84796154f3 ./contracts/interfaces/aave/IAaveV2LendingPool.sol

b8c8a6a495859ac4f4ddcddaa02e50264cf87925efede60229abcaa5588c4c36 ./contracts/interfaces/fcms/IFCM.sol

35d674b3615ff882620565330b5561cd1b435418863da74441a3e0aeba742663 ./contracts/interfaces/fcms/IAaveFCM.sol

812ed99122e1fe545dc55c80eaa6681bce22ce7523777a1b0be33bc2e5504de7 ./contracts/rate_oracles/OracleBuffer.sol

77f504b7a8eb14dce42057e7235c73e8eeb35315eb64090fb155f83f23dc90c4 ./contracts/rate_oracles/AaveRateOracle.sol

096fc2a1a0fa266ce6f36e4cd205e149de5a2a8124f5eec2069dd8bb2f2f741e ./contracts/rate_oracles/BaseRateOracle.sol

3bc678c4fc3bf4fbb9e4192c69b96265f551ad0b0c183d577855f18ac670f109 ./contracts/aave/AaveDataTypes.sol

1aed33d8eccbb6fc3bb8733292711a0f97a707bf459410ccba159166fa812e1d ./contracts/storage/FCMStorage.sol

a13845d1c806c1c8b9a4514249d49a4963b160614b3ae2f1140b762d7bbce0e8 ./contracts/storage/VAMMStorage.sol

cb2ae694e041766519e5a0a9b55aeea6c8924129ee42ca28696b2bcb4f2b115a ./contracts/storage/MarginEngineStorage.sol

8036a883296a938470a0ab588124f5f90e70f2b9af73e4975faa4779da16eedd ./contracts/periphery/Periphery.sol

debbc8c405eda725da9a098238312b9f6460a5be7fb4b72a19c41793cb3f473b ./contracts/periphery/peripheral_libraries/LiquidityAmounts.sol

e5924ff5d29838a2c13e52c86f2a9ed37b16577b13ba667952daf79d2bb36194 ./contracts/utils/SafeCastUni.sol

d2a63cc7488db6d867d2c5b74235b3ec3ec44129d75f94cd4b8cfd1b1a8e0c44 ./contracts/utils/LiquidityMath.sol

423b1d65c6a24e6e42c53b56aa0740f57aed1adbaa3611b178a0fdfb482f84f0 ./contracts/utils/CustomErrors.sol

b0c5cbdaebb5a6ac740a3b75d3e26972aae6e0fc8baf9b1e347ffc688eb5406c ./contracts/utils/Errors.sol

4cd6cd3df7c4a4e4c0ff56a184c3d3bff6b87e17918cc4a3e0e243a7401f1cfc ./contracts/utils/BitMath.sol

d4ee43e9bde4b3b869b3dc11f6dbfe6637f204f5633acbf77dfeb860d7e066fd ./contracts/utils/TickMath.sol

7f15775bfb981eb1dbcee25393e4cb8c0b920183059c84d139a4efa64faa181d ./contracts/utils/FixedPoint128.sol

c65ab22c521312a3fcb53426aac3d05e6bab5c3bbbb2130a71f47eaf14c75661 ./contracts/utils/WadRayMath.sol

b01a38f6e3c9a56257cb0ce62dbf11fa330577d3b6623816a99e6f98abb560fa ./contracts/utils/SqrtPriceMath.sol

417f6a19d7f2b51dca80d6823066c79f0e47d00db01d484a0916a1d6aa4b1b14 ./contracts/utils/Printer.sol

2cfe1427ff52485b98fbd16fe3019db0cda7d01e067787d65cd4f7078ae26764 ./contracts/utils/FullMath.sol

c2e7e60da5430bdbf8edfbce2d58d0be7b4448adc5bc75724a239d704c1e9442 ./contracts/utils/FixedPoint96.sol

9cb678d7dbbbc19276033252d9adaf146ccea34e6e0bcf0e58179b0febc4cf67 ./contracts/utils/UnsafeMath.sol

f4c5f452fda019d191e2ac19cbe7e4b1816c012d33f2b79ad7631180af2d2659 ./contracts/test/SqrtPriceMathTest.sol

e2dc0af611560b605dd5c756a37fcd97d85ba240248e953366095727d0cd7216 ./contracts/test/TickBitmapTest.sol

67a9a25a313b189db79d12c604064433f244eb6a605ef2fe64b77a211dbd448a ./contracts/test/ERC20.sol

924fe49e5337874bdca97ddb564496a43aa8fedfa4f885e310c10ea1e256d613 ./contracts/test/MockAToken.sol

b7ca0297748bc98c9ef1db0ec28bf36b79599f9ae42834105c3b8e1f931997c9 ./contracts/test/MockAaveLendingPool.sol

7571f8401f3ec0b47a7c1f064c551252fcdcf7b2425b67b4dcb082d0a730b48c ./contracts/test/E2ESetup.sol

1f1e5efc3e60459a226858edfc37bc5bfedcf852949ad57a3eff280bbe4fa24b ./contracts/test/LiquidityMathTest.sol

cd44a36004a2cc57b03d5c5bfdc8548efdda469aef37f873f73b8fc5d516ed82 ./contracts/test/ERC20Mock.sol

1d9541ecd7c8833715d741f0fe8730d73a86544a793d4ce39a5da32f39a75945 ./contracts/test/SwapMathTest.sol

017291ee62113f779c961a69b1d4ae16aed70dc7c55680a84b5685b3bc2fb2d8 ./contracts/test/PositionTest.sol

26652343580defbb7db04976d03e670710df2dbcb0d00f75fd6be17084d60c43 ./contracts/test/TickMathTest.sol

abe40fc7507edec29508e2e2071673c2a297eba40fff3909fc53046a35ca7c1a ./contracts/test/TickTest.sol

998be454b97f0e97e5d289e035dbe19155a109aba8eeea75456e50dafe47fc92 ./contracts/test/TestMarginEngine.sol

7d569f95c600375f28442ccb85abafae27bd01ebdb3b9c93fec1245c7727c8d4 ./contracts/test/TestAaveFCM.sol

c1fdea878c3f1fd034173b14bcce854d50cdd68f25dcfbfb1ab47fd522f5e79b ./contracts/test/MarginCalculatorTest.sol

1feb65c2f98a1f6b5fa2e057fcad04247ee7ba93955272604cc4bbb2f9eeabf3 ./contracts/test/Actor.sol

ceb9a491cae17d5f70968a2532b7b55f27934b058dcacf08d89f0780f8ba2174 ./contracts/test/TestRateOracle.sol

13035aeba7d222df15b3c3744ff19a642b5337ac7a01451de66dd735d2343392 ./contracts/test/FixedAndVariableMathTest.sol

196859ccb0427103c02dca1ba5542144c86b2adc57d91e60522e6aa64f4e9e0c ./contracts/test/TimeTest.sol

5167475fce3a8c2779063c36cc2d17193eed6887e747a93a73094858ce0a5b83 ./contracts/test/TestVAMM.sol

fd3e00a0ad3f44533e2b80a3eda60e4a620775d53e35de32c0cc4050e785dc44 ./contracts/core_libraries/Time.sol

237f099ff587781f4ed811930f9c7b7387da03b18c4b015a9b24fb18082fdf2d ./contracts/core_libraries/Position.sol

0c3a36ed100ee2837f1758ba38a6db2366e515663f16a0351441ec9937370b04 ./contracts/core_libraries/TickBitmap.sol

2785335f11bf998d0bef8322028eadd0ef425d33ffa93b443518399315545736 ./contracts/core_libraries/Tick.sol

6ca04c66e1581b8afccff8b070b959031ec1abbe4a26ac93391e89c7c26f5071 ./contracts/core_libraries/FixedAndVariableMath.sol

22746c98c83aaad60f64008cf01405465018596dd4a6d8ad79630500cc2dadc7 ./contracts/core_libraries/MarginCalculator.sol

3128657047f200c26a825bf2b59260140038894b8049c8c5a2fd372ddcebf2d9 ./contracts/core_libraries/SafeTransferLib.sol

f411cda967fe29b178130fe0d7960158f1381c7e7e7472d0ff6865793d368f65 ./contracts/core_libraries/TraderWithYieldBearingAssets.sol

b1f0b07cfa52bfac0376216a695f7289addc30af3157ba8e2dcb0bc30ae5a189 ./contracts/core_libraries/SwapMath.sol

Tests

d29db46f00eaa951e391d5ac0e6e2d8e96372c86c424af54460c8814b2359e26 ./test/helpers/toBn.ts

689f181b4f8cdc89ab2d05a0514b02d4411fce5381c0775e8a487afe5293e19c ./test/helpers/constants.ts

e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 ./test/helpers/errorMessages.ts

ff5923d598760479d5630f9698dfc39a323d9b3f32a8e4f6c2c13cde7e4fca79 ./test/helpers/time.ts

96b72e3d70e28ecdaecb4c8c1b58eb5a026725664f3ca694dcde223ec4ff6d3b ./test/fixtures/aaveV2.ts

40accec979affd5a76e61e52f261571383432c801ccd019d9958a57dd3918d14 ./test/parameters/mc_parameters.py

a3d65e78ca3f49d664341c885f460fdfe87ede8f3999b75a418fea131d2f508d ./test/periphery/periphery.ts

a9ed836a8152b59f2f547688dcdc26357c8ca0ebed17fefdaaba1fda10404bfc ./test/main_contracts/marginCalculator.ts

bd990b5f0a4b2a82ca6e29f25549f1121bccae9754e106c7d390ab84441fc3e5 ./test/main_contracts/factory.ts

a5419638099b5793b48dab6141527316a2fcfce4c804682123588a1ab1e441dd ./test/main_contracts/vamm_swaps.ts

a9ad62cb80116407d658d88e87e1f34a43b5fdaeacdd4964bb6c0484a789c2a6 ./test/main_contracts/marginEngine.ts

6fb03838945d41431e26d032cf25176f57c90d276e939a9a6799df7c68967dbb ./test/main_contracts/fcmAave.ts

9cdc2c89fa887d754eba7ecb6dccf262a4015b876ab2a216da90a8ad00efb403 ./test/main_contracts/vamm.ts

8335bf6d88529c268fabfc82da8a55e4683aeb82c92e25a49eaa6a9635bd0679 ./test/main_contracts/rate_oracles/aaveRateOracle.ts

8047c4ff3dc1381e2e4ff22ae3b9d3c1c6066d48cb11534ef7c1e380b2cf3c77 ./test/main_contracts/gas_spec/marginEngine.ts

e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 ./test/main_contracts/gas_spec/vamm.ts

4555cb00d74ae7b2e1a74ea293bc90890599aa03ba160dd8908d72ab8a7036fe ./test/shared/snapshotGasCost.ts

4e67e99a490e44255a15d75f0bd61b303b82d59ff9e4bc2de664dc69eddbd1e3 ./test/shared/fixtures.ts

4bb3e1bbc6fe1c22e695989f282132a9aa49f3c75d93350d2ffdefb2d64c06df ./test/shared/helpers.ts

256ba91ec2ad952a19ec3668b53280c7e957e8b3c79d0cb86b10447c70f63115 ./test/shared/near.ts

81957548803c638c84b8150f805dd07933b2be7b4c6157e57f596ec1443c9b64 ./test/shared/utilities.ts

38f57671122445096ac3c4fa0728cc900923880172eb0ec6c1f7b6a651289c2a ./test/shared/math.ts

352d47613c8eb8e3bf3e53fecba081bb62ffa596c6922898e4fc93f2b28819b4 ./test/shared/errors.ts

9ac4abf0c28aede01ec233dfdb5ef3355d94065431e89906d437e5afbae8227a ./test/shared/fullMath.ts

a6971a71800f5d5a0f2b7c50ed7f0ee2ebcaf4bdb0d23f5d2e979f558bf579de ./test/shared/sqrtPriceMath.ts

0334f3fe0d65152b2357782a371b2e7582844f9e1014fd86d1d9519c83f2991d ./test/shared/aaveMath.ts

c5ba6a1db1d99528b20e68c8a4cfccc80b4e4609f2b350b91f194103a0f18350 ./test/shared/format.ts

6feb82a9df7c976e6920a56107e04423fedb179c7e311169fd7dc20e8cd69a7d ./test/shared/constants.ts

66565efd7c7d9ffb5ef9ae2664ecaa8ea82859ab659cf92c4c28dee759fc3839 ./test/shared/sqrt.ts

af06d1745899454c2d562f539ca5d4e6dd119bf75ccc038794a86145fdf363dc ./test/shared/expect.ts

7735691c0beae20a989fc60565f72b39c9de4fd20336e2c169723f64f62fe3a3 ./test/shared/tickMath.ts

cdd0b910d32acf47dbfa0512f89c5105b83fb0eacc36a4634151fa015c1e2ae1 ./test/shared/functions.ts

2c87b8e0f57c2dd6e3b50ed52b0b6b02fc7de41a2e819d4f2ebea5825f1c9caa ./test/utils/liquidityMath.ts

4f0a15c96f0fc261ae9cc7299c4c6d9d980cdc2e564d7d4e80244d8270e2018b ./test/utils/fullMath.ts

7e63c2d711d1d0c962b20a48d250a3fddca96f3a24fed1960e8a90b695445e74 ./test/utils/sqrtPriceMath.ts

4f0a15c96f0fc261ae9cc7299c4c6d9d980cdc2e564d7d4e80244d8270e2018b ./test/utils/bitMath.ts

638691b0474c0c7fe2f83411d4d2fb604ca43a7e439982627a64450d4386bc04 ./test/utils/tickMath.ts

26d5bb5f2c18f93fe87ae0a707864920e5aee05411965d99db012f7010bd3045 ./test/utils/extractErrorMessage.ts

eda09f985321682d52f95d44afdca1c69a378272fcc47487999c5f2adff4132d ./test/end_to_end/general_setup/general.ts

7491e54e01cd5d7edb63f1492e36c9d5274f85685e7fe17751459c8d331c6442 ./test/end_to_end/general_setup/e2eSetup.ts

47f2d980867f2c1dab7009b2ccd5550331a5f3ed47fea8ca29650ec173018d94 ./test/end_to_end/general_setup/scenario13/scenario13.ts

ad37ef8d94d4a6df9a2057da97259d1d001843b929c5e5ceb43fb989b0a469c0 ./test/end_to_end/general_setup/scenario14/scenario14.ts

98126750b0ef1222c6fcb16c856084f2d72ac7d833cddc5a78351ed63fe43533 ./test/end_to_end/general_setup/scenario6/scenario6.ts

f0cf7f1611584ed7cdfda38c17631cfa717358c9fba295bc4e10f2fd73d2e207 ./test/end_to_end/general_setup/scenario1/scenario1ViaPeriphery.ts

1925efb024c88bad3c93c8b1b45065374ee50ad8d45d67cf86e0a12dd6f01534 ./test/end_to_end/general_setup/scenario1/scenario1.ts

0d4961e06955b733a5acb06f4d9612c53f88a0310575d0ef7c500355e88e59f5 ./test/end_to_end/general_setup/scenario8/scenario8.ts

b622e7eee08cddc7267670fb4a919c7559ac8ccfcf7d5c26676eb937625664c6 ./test/end_to_end/general_setup/scenario0/scenario0ViaPeriphery.ts

718250a5f5e62fcdbae55597047c9dde893bfd2bd3dc23598dfee3566fe85648 ./test/end_to_end/general_setup/scenario0/scenario0.ts

69ee14b7b2cec0eb257db4e7a52a40fddc068fb05322ca2dbde05ad3de41107b ./test/end_to_end/general_setup/scenario7/scenario7.ts

7c57af7dc4d627238c90d26facddc1e017d9ee62d47143430a252d51f7ec5718 ./test/end_to_end/general_setup/scenario11/scenario11.ts

ee8fa14ceb72bf41c7190ee55cb04f25e3c7e2f19542dc90a6dc0c69fba5c2d5 ./test/end_to_end/general_setup/scenario10/scenario10.ts

a8fe5f80e83f04e4f9b8fed6cecdfa9b2f7da52ae5b8376374391925cd107f82 ./test/end_to_end/general_setup/scenario2/scenario2.ts

3d404d6c9e293d7777337f7185eb4962ab329e9e4627f7576b7fe1a7a5129f66 ./test/end_to_end/general_setup/scenario4/scenario4.ts

8e46a5b92547eff8a271bd62e6712c0afa87f7b466e302b393d51802b2c90321 ./test/end_to_end/general_setup/scenario3/scenario3.ts

2fa182c0833712177190962086 83e8b64a91eab5f73493ad3285e2971397179a ./test/end_to_end/apySims/iteration1/apySims.py

0aede9bf6c824bbf3ab674f62fb7e85d3c331df225a17db5054dcca142b631e8 ./test/end_to_end/apySims/iteration1/plotter.py

fe607c2bb9c141fd64405f65ea7a3686f85dca22a7ea34fa8ad1cf70d3f877ce ./test/end_to_end/apySims/iteration1/apySims.ts

bbe528f5ab5b041db55772fd0a1182d5e6eab6e9d17a248760f76ae441a2da16 ./test/end_to_end/apySims/iteration0/apySims.py

4c9f177a5b95ed717090bdc82d40c8c52e9f7a54853039aaf8e96a9b28e7e049 ./test/end_to_end/apySims/iteration0/plotter.py

09bc2f93e797eb3862bedf8df67b3ccdf4be236ec840de82288b4cc0ad401a45 ./test/end_to_end/apySims/iteration0/apySims.ts

d0de1413ce3796142663fccf0614d510fa330dee1132637225e1224544b83317 ./test/end_to_end/apySims/iteration2/apySims.py

0d2c0aeb9e54dba9832f7b0c62be145fa3a10d7a86da3a6d76c6eb5ab36d0294 ./test/end_to_end/apySims/iteration2/plotter.py

7be8b3a9298757ffa934ce1cc62dab47a5bfa4a098dc96bc5ff56b5423bee2de ./test/end_to_end/apySims/iteration2/apySims.ts

1d35611ea3495515f74bf0c6ef25070c613af717a909d58dd51746a17230d7b6 ./test/core_libraries/tick.ts

3c87454085c89ce5805d8b3cac534d814c1bb85527a9f21ac93975c3b645c3c0 ./test/core_libraries/position.ts

d8c6e2cde2d9deeddc9fd879207c9937842016908bbaa1953ecca4611ade68ea ./test/core_libraries/swapMath.ts

cf4f89d986d6fa5739962f923cab2185091080e1c3cbf96fcddf35f1e9102bea ./test/core_libraries/fixedAndVariableMath.ts

ef51b1480a3d53efbb16b1c89140c9a57448226274ccd7cd8568f6c1710d690f ./test/core_libraries/tickBitmap.ts

e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 ./test/core_libraries/time.ts

Changelog

- 2022-03-25 - Initial report
- 2022-04-07 - Reaudit based on commit `3f22c5e`
- 2022-04-13 - Reaudit of `PR#110` and `PR#112` included in commit `56d1da5`

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.