Quantstamp DRAFT

Migration Core (Venus)

Executive Summary

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

Type	DeFi
Timeline	2025-08-26 through 2025-08-29
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Client provided internal documentation
Source Code	• VenusProtocol/venus-protocol #41eaa04
Auditors	• Leonardo Passos Senior Research Engineer • Mustafa Hasan Senior Auditing Engineer • Nikita Belenkov Senior Auditing Engineer

Documentation quality	High
Test quality	Low
Total Findings	3 Fixed: 1 Acknowledged: 2
High severity findings	0
Medium severity findings	0
Low severity findings	1 Fixed: 1
Undetermined severity findings	1 Acknowledged: 1
Informational findings	1 Acknowledged: 1

Summary of Findings

This security audit aimed to verify the migration of various Solidity 0.5.x contracts to the 0.8.25 version. Special attention was given to potential storage collisions and to language-breaking changes.

The performed audit did not identify any critical vulnerabilities. The code is well-written, and care has been taken to limit the audited pull request (PR) to only Solidity version changes. However, we did note that the current test coverage is not in line with industry standards. It is currently at 59%; we expected it to be at least 90%. Improving coverage is key in identifying potential issues when migrating contracts and ensuring the implementation follows the expected protocol behaviour.

Fix-Review Update 2025-09-12:

Repository: `https://github.com/VenusProtocol/venus-protocol/` Commit: `3cacf4606313e7fecf73a6db1ae60a414a85e773` MIG-1 has been remediated. MIG-2 and MIG-3 are acknowledged but will not be addressed due to dependencies and native-token design. Suggestion S1 is partially covered by pull requests 610 and 613 and remains out of scope.

ID	DESCRIPTION	SEVERITY	STATUS
MIG-1	`VTokenstorage` **Does Not Keep Extra Slots**	Low	Fixed
MIG-2	`IPrimeV5` **Still Relies on Old Solidity Version**	Informational	Acknowledged
MIG-3	**Fallback Function Can No Longer Receive Native Tokens**	Undetermined	Acknowledged

Assessment Breakdown

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

Disclaimer

Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

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

- Storage collisions
- Change in semantics
- Breaking changes

Methodology

1. Code review that includes the following
    1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    3. 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:
    1. 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.
    2. 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, clarity, 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.

Scope

The scope of this audit was limited to the analysis of the https://github.com/VenusProtocol/venus-protocol/pull/607/ pull request.

Files Included

contracts/Admin/VBNBAdmin.sol contracts/Admin/VBNBAdminStorage.sol contracts/Comptroller/Diamond/facets/FacetBase.sol
contracts/Comptroller/Diamond/facets/MarketFacet.sol contracts/Comptroller/Diamond/facets/PolicyFacet.sol
contracts/Comptroller/Diamond/facets/RewardFacet.sol contracts/Comptroller/Diamond/facets/SetterFacet.sol
contracts/Comptroller/Diamond/facets/XVSRewardsHelper.sol contracts/Comptroller/Diamond/interfaces/IDiamondCut.sol
contracts/Comptroller/Diamond/interfaces/IFacetBase.sol contracts/Comptroller/Diamond/interfaces/IMarketFacet.sol
contracts/Comptroller/Diamond/interfaces/IPolicyFacet.sol contracts/Comptroller/Diamond/interfaces/IRewardFacet.sol
contracts/Comptroller/Diamond/interfaces/ISetterFacet.sol contracts/Comptroller/Diamond/Diamond.sol
contracts/Comptroller/Diamond/DiamondConsolidated.sol contracts/Comptroller/ComptrollerInterface.sol
contracts/Comptroller/ComptrollerLensInterface.sol contracts/Comptroller/ComptrollerStorage.sol contracts/Comptroller/Unitroller.sol
contracts/external/IProtocolShareReserve.sol contracts/external/IWBNB.sol contracts/Lens/ComptrollerLens.sol
contracts/Lens/SnapshotLens.sol contracts/Lens/VenusLens.sol contracts/Liquidator/Liquidator.sol contracts/Oracle/PriceOracle.sol
contracts/Tokens/Prime/IPrime.sol contracts/Tokens/Prime/IPrimeV5.sol contracts/Tokens/VAI/IVAI.sol contracts/Tokens/VAI/VAIController.sol
contracts/Tokens/VAI/VAIControllerInterface.sol contracts/Tokens/VAI/VAIControllerStorage.sol contracts/Tokens/VAI/VAIUnitroller.sol
contracts/Tokens/VTokens/VBep20.sol contracts/Tokens/VTokens/VBep20Delegate.sol contracts/Tokens/VTokens/VBep20Delegator.sol
contracts/Tokens/VTokens/VBep20Immutable.sol contracts/Tokens/VTokens/VBNB.sol contracts/Tokens/VTokens/VToken.sol
contracts/Tokens/VTokens/VTokenInterfaces.sol contracts/Tokens/XVS/IXVS.sol contracts/Utils/CarefulMath.sol contracts/Utils/ErrorReporter.sol
contracts/Utils/Exponential.sol contracts/Utils/ExponentialNoError.sol contracts/XVSVault/XVSVault.sol contracts/XVSVault/XVSVaultStorage.sol
contracts/InterfacesV8.sol

Findings

MIG-1 `VTokenstorage` Does Not Keep Extra Slots

Low Fixed

Update
Marked as "Fixed" by the client.
Addressed in: `3cacf4606313e7fecf73a6db1ae60a414a85e773` .

File(s) affected: `contracts/Tokens/VTokens/legacy/VTokenStorageR1.sol`

Description: The new `VTokenInterfaceR1` implementation now inherits from `VTokenStorageR1` instead of `VTokenStorage` . The previous `VTokenStorage` contract reserved an additional 50 storage slots through a `uint256[50] __gap` placeholder, ensuring room for future variable additions without altering the storage layout.

After the migration, this explicit reservation is no longer visible to developers. In subsequent upgrades, they may overlook the previously intended expansion buffer or inadvertently add storage variables beyond the implicit 50-slot allowance, risking storage collisions or corrupting the upgrade path.

Recommendation: Document in `VTokenStorageR1` how many additional slots may be added safely. Consider keeping a `__gap` variable for all unused slots (50-23) as a means to prevent accidental storage layout collisions in future upgrades.

MIG-2 `IPrimeV5` Still Relies on Old Solidity Version

• Informational Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
We still need IPrimeV5 for XVSVault. Changing XVSVault implementation was out of the scope for this migration.
```

File(s) affected: `contracts/Tokens/Prime/IPrimeV5.sol`

Description: `IPrimeV5.sol` still targets Solidity ^0.5.16 and enables the experimental ABIEncoderV2, which is inconsistent with the PR's goal to standardize on Solidity 0.8.25.

Recommendation:

- Update the pragma to `pragma solidity 0.8.25`
- Remove `pragma experimental ABIEncoderV2;` (ABI encoder v2 is default in 0.8.x).

MIG-3
Fallback Function Can No Longer Receive Native Tokens

• Undetermined Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

- `Diamond and vBep20 should not receive native tokens, so it's intended`
- `vBNB is still able to receive native tokens via the payable receive() function`

File(s) affected: `contracts/Comptroller/Diamond/Diamond.sol`, `contracts/Comptroller/Unitroller.sol`, `contracts/Tokens/VAI/VAIUnitroller.sol`, `contracts/Tokens/VTokens/VBNB.sol`, `contracts/Tokens/VTokens/VBep20Delegator.sol`

Description: Previously, the fallback functions in `Diamond`, `Unitroller`, `VAIUnitroller`, `VBNB`, and `VBep20Delegator` were payable. The pull request changes them to non-payable. As a result, any delegated calls that expect to forward native tokens will now revert.

Recommendation: From an upgradability standpoint, keeping the fallback payable preserves flexibility. If the intention is to disallow receiving native tokens, add code comments explaining the rationale so future maintainers understand the design choice.

Auditor Suggestions

S1 Interface Naming and File Organization Inconsistencies

Unresolved

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
Mostly fixed by PRs 610, 613, but out of the scope for this upgrade
```

File(s) affected: `venusprotocol/oracle/contracts/interfaces/OracleInterface.sol`, `contracts/Comptroller/ComptrollerLensInterface.sol`, `contracts/Comptroller/ComptrollerInterface.sol`, `contracts/Tokens/VTokens/VTokenInterfaces.sol`, `contracts/Tokens/VAI/VAIControllerInterface.sol`, `contracts/InterfacesV8.sol`

Description: Several interfaces do not follow common Solidity conventions:

- Interface names not prefixed with I (e.g., `ComptrollerLensInterface`, `VAIControllerInterface`).
- Filenames don't match primary type names or case (e.g., `IVtoken.sol` vs `IVToken`).

- Multiple interfaces aggregated in a single file (e.g., `InterfacesV8.sol`, `ComptrollerInterface.sol`, `VTokenInterfaces.sol`), reducing discoverability and tooling clarity.

Recommendation:

- Rename interfaces to use `I*` prefix and align filenames: e.g., `ComptrollerLensInterface` → `IComptrollerLens.sol`; `ResilientOracleInterface` → `IResilientOracle.sol`.
- Fix filename/type-case mismatches (e.g., `IVtoken.sol` → `IVToken.sol`).
- Split aggregated interface files so each file contains a single primary interface matching its filename.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – 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 pose an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **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).

Test Suite Results

While the suite has over 700 tests, it is ineffective in capturing various execution flows (see coverage results). Except for 1, the other 722 tests passed.

```
Network Info
============
> HardhatEVM: v2.22.18
> network:    hardhat



  VBNBAdmin
    ✔ set VBNBAdmin as vBNB admin
  harvest income
    ✔ reduce BNB reserves
  set interest rate model
    ✔ setInterestRateModel

Comptroller
  _initializeMarket
    ✔ Supply and borrow state after initializing the market in the pool
  _setVenusSpeeds
    ✔ Revert on invalid supplySpeeds input
    ✔ Revert on invalid borrowSpeeds input
    ✔ Revert for unlisted market
    ✔ Revert on invalid borrowSpeeds input
    ✔ Updating non-zero speeds after setting it zero (49ms)

Comptroller
  _setAccessControlManager
    ✔ Reverts if called by non-admin
    ✔ Reverts if ACM is zero address
    ✔ Sets ACM address in storage
    ✔ should revert on same value
  Access Control
    setCollateralFactor
```

```
      ✔ Should have AccessControl
      ✔ Should revert for same values
    setLiquidationIncentive
      ✔ Should have AccessControl
    setMarketBorrowCaps
      ✔ Should have AccessControl
    setMarketSupplyCaps
      ✔ Should have AccessControl
    setProtocolPaused
      ✔ Should have AccessControl
    setActionsPaused
      ✔ Should have AccessControl
    _supportMarket
      ✔ Should have AccessControl
    supportMarket
      ✔ Should have AccessControl
    seizeVenus
      ✔ Should have AccessControl

Comptroller: assetListTest
  enterMarkets
    ✔ properly emits events (43ms)
    ✔ adds to the asset list only once (89ms)
    ✔ the market must be listed for add to succeed (51ms)
    ✔ returns a list of codes mapping to user's ultimate membership in given addresses (49ms)
  exitMarket
    ✔ doesn't let you exit if you have a borrow balance (73ms)
    ✔ rejects unless redeem allowed (145ms)
    ✔ accepts when you're not in the market already (79ms)
    ✔ properly removes when there's only one asset (117ms)
    ✔ properly removes when there's only two assets, removing the first (149ms)
    ✔ properly removes when there's only two assets, removing the second (150ms)
    ✔ properly removes when there's only three assets, removing the first (182ms)
    ✔ properly removes when there's only three assets, removing the second (175ms)
    ✔ properly removes when there's only three assets, removing the third (180ms)
  entering from borrowAllowed
    ✔ enters when called by a vtoken (63ms)
    ✔ reverts when called by not a vtoken
    ✔ adds to the asset list only once (85ms)
  unlistMarkets
    ✔ properly emits events and unlist market (96ms)
    ✔ reverts when unlisting not a listed market (79ms)

Comptroller
  constructor
    ✔ on success it sets admin to creator and pendingAdmin is unset (1234ms)
  _setLiquidationIncentive
    ✔ fails if incentive is less than 1e18
    ✔ accepts a valid incentive and emits a NewLiquidationIncentive event
    ✔ should revert on same values
  _setVenusVAIVaultRate
    ✔ should revert on same values
  _setVAIVaultInfo
    ✔ should revert on same values
  _setVAIController
    ✔ should revert on same values
  _setVAIMintRate
    ✔ should revert on same values
  _setLiquidatorContract
    ✔ should revert on same values
    ✔ should revert on zero address
  _setPauseGuardian
    ✔ should revert on same values
  _setVenusSpeeds
    ✔ ensure non zero address for venus speeds
  _setPriceOracle
    ✔ fails if called by non-admin
    ✔ accepts a valid price oracle and emits a NewPriceOracle event
    ✔ setPriceOracle is alias for _setPriceOracle
    ✔ Should revert on same values
  _setComptrollerLens
    ✔ fails if not called by admin
```

```
    ✔ should fire an event
    ✔ should revert on same value
  _setCloseFactor
    ✔ fails if not called by admin
    ✔ should revert on same values
    ✔ fails if factor is set out of range
  _setCollateralFactor
    ✔ fails if asset is not listed
    ✔ fails if factor is set without an underlying price
    ✔ succeeds and sets market
    ✔ succeeds and sets market using alias
    ✔ should revert on same values
  _setForcedLiquidation
    ✔ fails if asset is not listed
    ✔ fails if ACM does not allow the call
    ✔ sets forced liquidation
    ✔ should alias setForcedLiquidation to _setForcedLiquidation
    ✔ sets forced liquidation for VAI, even though it is not a listed market (42ms)
    ✔ emits IsForcedLiquidationEnabledUpdated event
  _setForcedLiquidationForUser
    ✔ fails if asset is not listed
    ✔ fails if ACM does not allow the call
    ✔ sets forced liquidation for user
    ✔ sets forced liquidation for VAI, even though it is not a listed market (44ms)
    ✔ emits IsForcedLiquidationEnabledForUserUpdated event
  _supportMarket
    ✔ fails if asset is not a VToken
    ✔ succeeds and sets market (48ms)
    ✔ cannot list a market a second time (93ms)
    ✔ can list two different markets (153ms)
  updateDelegate
    ✔ should revert when zero address is passed
    ✔ should revert when approval status is already set to the requested value
    ✔ should emit event on success
  Hooks
    mintAllowed
      ✔ allows minting if cap is not reached
      ✔ reverts if supply cap reached
      ✔ reverts if market is not listed
    redeemVerify
      ✔ should allow you to redeem 0 underlying for 0 tokens
      ✔ should allow you to redeem 5 underlyig for 5 tokens
      ✔ should not allow you to redeem 5 underlying for 0 tokens
    liquidateBorrowAllowed
      Forced liquidations enabled for user
        ✔ enables forced liquidation for user
        ✔ reverts if borrowed market is not listed (79ms)
        ✔ reverts if collateral market is not listed
        ✔ does not revert if borrowed vToken is VAIController (75ms)
        ✔ allows liquidations without shortfall (38ms)
        ✔ allows to repay 100% of the borrow
        ✔ fails with TOO_MUCH_REPAY if trying to repay > borrowed amount
        ✔ checks the shortfall if isForcedLiquidationEnabledForUser is set back to false (66ms)
      Forced liquidations enabled for entire market
        ✔ reverts if borrowed market is not listed (69ms)
        ✔ reverts if collateral market is not listed (55ms)
        ✔ does not revert if borrowed vToken is VAIController (78ms)
        ✔ allows liquidations without shortfall
        ✔ allows to repay 100% of the borrow
        ✔ fails with TOO_MUCH_REPAY if trying to repay > borrowed amount
        ✔ checks the shortfall if isForcedLiquidationEnabled is set back to false (54ms)
      Forced liquidations disabled
        ✔ reverts if borrowed market is not listed (58ms)
        ✔ reverts if collateral market is not listed
        ✔ does not revert if borrowed vToken is VAIController (50ms)
        ✔ fails if borrower has 0 shortfall
        ✔ succeeds if borrower has nonzero shortfall
    borrow
      ✔ allows borrowing if cap is not reached
      ✔ reverts borrowing if borrow cap is reached
      ✔ reverts borrowing if borrow cap is 0
0xdF38AdD570290E3765E55dBd865a22622bB3C4B9
```

```
      ✔ getBorrowingPower is an alias for getAccountLiquidity

  Comptroller
    ✔ Revert on check for the function selector (200ms)
    ✔ Add Facet and function selectors to proxy (88ms)
    ✔ Get all facet function selectors by facet address
    ✔ Get facet position by facet address
    ✔ Get all facet addresses
    ✔ Get all facets address and their selectors
    ✔ Get facet address and position by function selector
    ✔ Remove function selector from facet mapping (38ms)
    ✔ Replace the function from facet mapping (67ms)
    ✔ Remove all functions (43ms)

  Comptroller
    liquidateCalculateAmountSeize
      ✔ fails if borrowed asset price is 0
      ✔ fails if collateral asset price is 0
      ✔ fails if the repayAmount causes overflow
      ✔ fails if the borrowed asset price causes overflow
      ✔ reverts if it fails to calculate the exchange rate
      ✔ returns the correct value for
100000000000000000,100000000000000000,100000000000000000,100000000000000000,100000000000000000
      ✔ returns the correct value for
200000000000000000,200000000000000000,142000000000000000,130000000000000000,245000000000000000
      ✔ returns the correct value for
278900000000000000,523048084200000000,771320000000000000000,130000000000000000,1.000245e+22
      ✔ returns the correct value for
7.009232529961056e+24,2.5278726317240445e+24,2.6177112093242585e+23,1179713989619784000,7.790468414639561
e+24
      ✔ returns the correct value for
2.538124957495297e+24,2.696894902112628e+24,4.716636974273603e+22,1407196317860701700,7.887135858770727e+
24

  ComptrollerMock
    _setActionsPaused
      ✔ reverts if the market is not listed
      ✔ does nothing if the actions list is empty (54ms)
      ✔ does nothing if the markets list is empty
      ✔ can pause one action on several markets (60ms)
      ✔ can pause several actions on one market (67ms)
      ✔ can pause and unpause several actions on several markets (176ms)

  MoveDebtDelegate
    setBorrowAllowed
      ✔ fails if called by a non-owner
      ✔ fails if called with zero address for vTokenToBorrow
      ✔ sets borrowAllowed to the specified value
      ✔ emits an event
      ✔ does not emit an event if no-op
    setRepaymentAllowed
      ✔ fails if called by a non-owner
      ✔ fails if called with zero address for vTokenToRepay
      ✔ sets borrowAllowed to the specified value
      ✔ emits an event
      ✔ does not emit an event if no-op
    moveDebt
      ✔ fails if called with a token that is not allowed to be borrowed
      ✔ fails if called with a token that is not allowed to be repaid
      ✔ fails if called with a borrower who is not in the repayment allowlist
      ✔ succeeds if repayments are allowed for ANY_USER (95ms)
      ✔ fails if comptrollers don't match (48ms)
      ✔ fails if repayBorrowBehalf returns a non-zero error code
      ✔ fails if borrowBehalf returns a non-zero error code (72ms)
      ✔ transfers repayAmount of vTokenToRepay.underlying() from the sender (82ms)
      ✔ approves vToken to transfer money from the contract (84ms)
      ✔ calls repayBorrowBehalf after transferring the underlying to self (81ms)
      ✔ converts the amounts using the oracle exchange rates (83ms)
      ✔ uses the actually repaid amount rather than specified amount (83ms)
      ✔ transfers the actually borrowed amount to the owner (87ms)
```

```
    sweepTokens
      ✔ fails if called by a non-owner
      ✔ transfers the full balance to the owner

  assetListTest
    swapDebt
      ✔ fails if called by a non-owner
      ✔ fails if comptrollers don't match (66ms)
      ✔ fails if repayBorrowBehalf returns a non-zero error code (47ms)
      ✔ fails if borrowBehalf returns a non-zero error code (95ms)
      ✔ transfers repayAmount of underlying from the sender (101ms)
      ✔ approves vToken to transfer money from the contract (115ms)
      ✔ calls repayBorrowBehalf after transferring the underlying to self (113ms)
      ✔ converts the amounts using the oracle exchange rates (114ms)
      ✔ uses the actually repaid amount rather than specified amount (117ms)
      ✔ transfers the actually borrowed amount to the owner (119ms)
    sweepTokens
      ✔ fails if called by a non-owner
      ✔ transfers the full balance to the owner

  Evil Token test
Duplicate definition of Log (Log(string,address), Log(string,uint256))
Duplicate definition of Log (Log(string,address), Log(string,uint256))
Duplicate definition of Log (Log(string,address), Log(string,uint256))
    ✔ Check the updated vToken states after transfer out (949ms)

  BUSDLiquidator
    setLiquidatorShare
      ✔ should set liquidator share (40ms)
      ✔ should emit NewLiquidatorShare event
      ✔ should revert if caller is not owner
      ✔ should revert if new liquidator share is < 1
      ✔ should revert if new liquidator share is > (liquidation incentive - treasury percent)
      ✔ should succeed if new liquidator share is = (liquidation incentive - treasury percent) (44ms)
    liquidateEntireBorrow
      ✔ should repay entire borrow (749ms)
Bal Prev BigNumber { _hex: '0x00', _isBigNumber: true }
Bal After BigNumber { _hex: '0x00', _isBigNumber: true }
      ✔ should seize collateral (1012ms)
    liquidateBorrow
      ✔ should repay a part of the borrow (928ms)
      ✔ should seize collateral (881ms)

  TokenRedeemer
    redeemAndTransfer
      ✔ should fail if called by a non-owner
      ✔ should fail if redeem fails (47ms)
      ✔ should succeed with zero amount (121ms)
    ✔ should redeem all vTokens (185ms)
    ✔ should transfer all underlying to the receiver (182ms)
  redeemUnderlyingAndTransfer
    ✔ should fail if called by a non-owner
    ✔ should revert if redeemer does not have vToken balance (91ms)
    ✔ should redeem and transfer successfully (292ms)
  redeemUnderlyingAndRepayBorrowBehalf
    ✔ should revert if redeemer does not have vToken balance (74ms)
    ✔ should redeem and repay successfully (636ms)
  redeemAndBatchRepay
    Generic
      ✔ fails if called by a non-owner
    Full repayment
      Native asset
        ✔ redeems just the required amount of vTokens (323ms)
        ✔ repays all borrows in full (372ms)
        ✔ transfers the excess vTokens to the receiver (309ms)
        ✔ transfers the excess BNB to the receiver (313ms)
      Tokens
        ✔ redeems just the required amount of vTokens (469ms)
        ✔ repays up to specified caps (425ms)
        ✔ repays all borrows in full (444ms)
        ✔ transfers the excess vTokens to the receiver (426ms)
        ✔ transfers the excess underlying to the receiver (456ms)
```

```
    Partial repayment
      Native asset
        ✔ redeems all available vTokens, up to 1 vToken wei (259ms)
        ✔ repays the three borrows: [in full, partially, no repayment] (314ms)
        ✔ uses the excess BNB to repay the debt in full (399ms)
        ✔ does not keep any vBNB or BNB balance (340ms)
      Tokens
        ✔ redeems all available vTokens, up to 1 vToken wei (366ms)
        ✔ repays the three borrows: [in full, partially, no repayment] (406ms)
        ✔ uses the excess underlying to repay the debt in full (446ms)
        ✔ does not keep any vToken or underlying balance (447ms)
  batchRepayVAI
    ✔ fails if called by a non-owner
    ✔ repays one borrow successfully (308ms)
    ✔ repays multiple borrows successfully and transfers refund to treasury (765ms)
    ✔ repays up to caps (800ms)
    ✔ partially repays borrows if insufficient VAI (684ms)
    ✔ can repay small amounts without failure (879ms)
  sweepTokens
    ✔ fails if called by a non-owner
    ✔ sweeps tokens to destination if called by owner (50ms)
    ✔ sweeps native asset to destination

Two Kinks Interest Rate Model Tests
    ✔ Utilization rate: borrows is zero
    ✔ Utilization rate
    ✔ Borrow Rate: below kink1 utilization
    ✔ Borrow Rate: above kink1 and below kink2 utilization (42ms)
    ✔ Borrow Rate: above kink2 utilization (47ms)
    ✔ Borrow Rate: above kink2 utilization and negative multipliers (80ms)
    ✔ Supply Rate

  VenusLens: Rewards Summary
    ✔ Should get summary for all markets (293ms)

  Liquidator
    splitLiquidationIncentive
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2690 blockNumber3694)
network block skew detected; skipping block events (emitted=2693 blockNumber3695)
network block skew detected; skipping block events (emitted=2693 blockNumber3695)
network block skew detected; skipping block events (emitted=2693 blockNumber3695)
network block skew detected; skipping block events (emitted=2693 blockNumber3695)
network block skew detected; skipping block events (emitted=2693 blockNumber3695)
      ✔ splits liquidationIncentive between Treasury and Liquidator with correct amounts
    distributeLiquidationIncentive
      ✔ distributes the liquidationIncentive between Treasury and Liquidator with correct amounts (83ms)
      ✔ reverts if transfer to liquidator fails
      ✔ reverts if underlying transfer to protocol share reserves fails (55ms)

  Liquidator
    liquidateBorrow
      liquidating BEP-20 debt
        ✔ fails if borrower is zero address
        ✔ fails if some BNB is sent along with the transaction (48ms)
        ✔ transfers the seized collateral to liquidator and protocolShareReserve (159ms)
        ✔ transfers tokens from the liquidator (187ms)
        ✔ approves the borrowed VToken to spend underlying (149ms)
        ✔ calls liquidateBorrow on borrowed VToken (165ms)
        ✔ emits LiquidateBorrowedTokens event (167ms)
      liquidating VAI debt
        ✔ transfers VAI from the liquidator (167ms)
        ✔ approves VAIController to spend VAI (136ms)
        ✔ calls liquidateVAI on VAIController (145ms)
    liquidating BNB debt
      ✔ fails if msg.value is not equal to repayment amount (95ms)
      ✔ transfers BNB from the liquidator (104ms)
```

```
      ✔ calls liquidateBorrow on VBNB (96ms)
      - forwards BNB to VBNB contract
    setTreasuryPercent
      ✔ updates treasury percent in storage (43ms)
      ✔ fails when permission is not granted
      ✔ fails when the percentage is too high
      ✔ uses the new treasury percent during distributions (201ms)
    Force VAI Liquidation
      ✔ Should able to liquidate any token when VAI debt is lower than minLiquidatableVAI (126ms)
      ✔ Should not able to liquidate any token when VAI debt is greater than minLiquidatableVAI (47ms)
      ✔ Should able to liquidate any token when VAI debt is greater than minLiquidatableVAI but forced
liquidation is enabled
      ✔ Should able to liquidate VAI token when VAI debt is greater than minLiquidatableVAI (163ms)
      ✔ Should able to liquidate any token and VAI token when force Liquidation is off (203ms)

  Liquidator
    Restricted liquidations
      addToAllowlist
        ✔ fails if not allowed to call (72ms)
        ✔ adds address to allowlist (44ms)
        ✔ fails if already in the allowlist (40ms)
        ✔ emits LiquidationPermissionGranted event
      removeFromAllowlist
        ✔ fails if not allowed to call
        ✔ fails if not in the allowlist
        ✔ removes address from allowlist (67ms)
        ✔ emits LiquidationPermissionRevoked event (42ms)
      restrictLiquidation
        ✔ fails if not allowed to call
        ✔ restricts liquidations for the borrower
        ✔ fails if already restricted (47ms)
        ✔ emits LiquidationRestricted event
      unrestrictLiquidation
        ✔ fails if not allowed to call
        ✔ removes the restrictions for the borrower (64ms)
        ✔ fails if not restricted
        ✔ emits LiquidationRestricted event (44ms)
      liquidateBorrow
        ✔ fails if the liquidation is restricted (38ms)
        ✔ proceeds with the liquidation if the guy is allowed to (65ms)

  PrimeScenario Token
    setMaxLoopsLimit()
Warning: Potentially unsafe deployment of
contracts/Tokens/Prime/PrimeLiquidityProvider.sol:PrimeLiquidityProvider

    You are using the `unsafeAllow.internal-function-storage` flag.
    Internal functions are code pointers which will no longer be valid after an upgrade.
    Make sure you reassign internal functions in storage variables during upgrades.

Warning: Potentially unsafe deployment of contracts/test/PrimeScenario.sol:PrimeScenario

    You are using the `unsafeAllow.internal-function-storage` flag.
    Internal functions are code pointers which will no longer be valid after an upgrade.
    Make sure you reassign internal functions in storage variables during upgrades.

      ✔ Revert when maxLoopsLimit setter is called by non-owner
      ✔ Revert when new loops limit is less than old limit
      ✔ maxLoopsLimit setter success (40ms)
    protocol setup
      ✔ markets added
      ✔ borrow balance
      ✔ get markets in prime
    mint and burn
      ✔ stake and mint (347ms)
      ✔ stake and unstake (255ms)
      ✔ stake manually (269ms)
      ✔ burn revocable token (745ms)
      ✔ cannot burn irrevocable token (669ms)
      ✔ manually burn irrevocable token (523ms)
      ✔ issue (685ms)
      ✔ upgrade (496ms)
```

```
      ✔ stake, issue and unstake (913ms)
      ✔ issue, stake and burn (816ms)
    boosted yield
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
network block skew detected; skipping block events (emitted=3760 blockNumber7779765)
      ✔ calculate score (141ms)
network block skew detected; skipping block events (emitted=3766 blockNumber7779767)
network block skew detected; skipping block events (emitted=3766 blockNumber7779767)
network block skew detected; skipping block events (emitted=3766 blockNumber7779767)
network block skew detected; skipping block events (emitted=3766 blockNumber7779767)
network block skew detected; skipping block events (emitted=3766 blockNumber7779767)
network block skew detected; skipping block events (emitted=3765 blockNumber7779770)
      ✔ accrue interest - prime token minted after market is added (483ms)
      ✔ claim interest (298ms)
      update score
        ✔ add existing market after issuing prime tokens - update score gradually (838ms)
        ✔ add existing market after issuing prime tokens - update score manually (1553ms)
    PLP integration
      ✔ claim interest (441ms)
      ✔ APR Estimation (94ms)
      ✔ Hypothetical APR Estimation (325ms)

  PrimeLiquidityProvider: tests
    Testing all initalized values
Warning: Potentially unsafe deployment of
contracts/Tokens/Prime/PrimeLiquidityProvider.sol:PrimeLiquidityProvider

    You are using the `unsafeAllow.internal-function-storage` flag.
    Internal functions are code pointers which will no longer be valid after an upgrade.
    Make sure you reassign internal functions in storage variables during upgrades.

      ✔ Tokens intialized
      ✔ Distribution Speed
    Testing all setters
      ✔ Revert on invalid args for initializeTokens
      ✔ Revert on re-intializing token
      ✔ initializeTokens success
      ✔ pauseFundsTransfer
      ✔ resumeFundsTransfer (57ms)
      ✔ Revert on invalid args for setTokensDistributionSpeed
      ✔ Revert on non initialized token
      ✔ Revert on invalid distribution speed for setTokensDistributionSpeed (56ms)
      ✔ setTokensDistributionSpeed success with default max speed (56ms)
      ✔ setTokensDistributionSpeed success (66ms)
      ✔ setMaxTokensDistributionSpeed success
      ✔ Reverts on setting prime address same as previous
      ✔ Revert on invalid prime token address
      ✔ Revert when prime token setter is called by non-owner
      ✔ setPrimeToken success
      ✔ Revert when maxLoopsLimit setter is called by non-owner
      ✔ Revert when new loops limit is less than old limit
      ✔ maxLoopsLimit setter success
    Accrue tokens
      ✔ Revert on non initialized token
      ✔ Accrue amount for tokenA (63ms)
      ✔ Accrue amount for multiple tokens (431ms)
    Release funds to prime contract
      ✔ Revert on funds transfer Paused
      ✔ Revert on invalid caller
      ✔ Release funds success (74ms)
    Sweep token
      ✔ Revert on insufficient balance
      ✔  Sweep token success (54ms)

  Swap Contract
    ✔ revert if vToken address is not listed
    Setter
```

```
    ✔ should reverted if zero address
    ✔ should reverted if vToken not listed
    ✔ setting address for VBNBToken  (38ms)
  Swap
    ✔ revert if path length is 1
    ✔ revert if deadline has passed
    ✔ revert if output amoutn is below minimum
    ✔ should be reverted if tokenA == tokenB
    ✔ should swap tokenA -> tokenB (54ms)
    ✔ revert if deadline has passed
    ✔ revert if address zero
    ✔ should reverted if first address in not WBNB address
    ✔ should reverted if output amount is below minimum (42ms)
    ✔ should swap BNB -> token (58ms)
    ✔ revert if deadline has passed
    ✔ should swap tokenA -> tokenB  at supporting fee
    ✔ should reverted if deadline passed
    ✔ should swap BNB -> token  at supporting fee
    ✔ should swap EXact token -> BNB at supporting fee  (81ms)
    ✔ should swap tokesn for Exact BNB
    ✔ should swap tokens for Exact Tokens
    ✔ should swap tokens for Exact BNB
    ✔ should swap BNB for Exact Tokens
  Supply
    ✔ revert if deadline has passed
    ✔ swap tokenA -> tokenB --> supply tokenB (102ms)
    ✔ swap BNB -> token --> supply token (109ms)
    ✔ revert if deadline has passed at supporting fee
    ✔ swap tokenA -> tokenB --> supply tokenB at supporting fee (105ms)
    ✔ swap BNB -> token --> supply token at supporting fee (101ms)
    ✔ swap tokenA -> exact tokenB (105ms)
    ✔ swap bnb -> exact tokenB (116ms)
    ✔ Exact tokens -> BNB and supply
    ✔ Exact tokens -> BNB and supply at supporting fee
  Repay
    ✔ revert if deadline has passed
    ✔ swap tokenA -> tokenB --> supply tokenB (106ms)
    ✔ swap BNB -> token --> supply token (107ms)
    ✔ revert if deadline has passed at supporting fee
    ✔ swap tokenA -> tokenB --> reapy tokenB at supporting fee (104ms)
    ✔ swap BNB -> token --> repay token at supporting fee (111ms)
    ✔ swap tokenA -> exact tokenB (98ms)
    ✔ swap tokenA -> full debt of tokenB (111ms)
    ✔ swap bnb -> exact tokenB (115ms)
    ✔ swap bnb -> full tokenB debt (118ms)
    ✔ Exact tokens -> BNB at supporting fee (80ms)
    ✔ Exact tokens -> BNB (62ms)
    ✔ Tokens -> Exact BNB (62ms)
    ✔ Tokens -> Exact BNB and supply
    ✔ Tokens -> full debt of BNB
  Sweep Token
    ✔ Should be reverted if get zero address
    ✔ Sweep ERC-20 tokens (76ms)
  library function
    ✔ Quote function
    ✔ getAmoutIn function
    ✔ getAmoutout function
    ✔ getAmoutout function
    ✔ getAmoutout function

admin / _setPendingAdmin / _acceptAdmin
  admin()
    ✔ should return correct admin
  pendingAdmin()
    ✔ should return correct pending admin
  _setPendingAdmin()
    ✔ should only be callable by admin
    ✔ should properly set pending admin
    ✔ should properly set pending admin twice
    ✔ should emit event
  _acceptAdmin()
    ✔ should fail when pending admin is zero
```

```
    ✔ should fail when called by another account (e.g. root)
    ✔ should succeed and set admin and clear pending admin
    ✔ should emit log on success

Unitroller
  constructor
    ✔ sets admin to caller and addresses to 0
  _setPendingImplementation
    Check caller is admin
      ✔ emits a failure log
      ✔ does not change pending implementation address
    succeeding
      ✔ stores pendingComptrollerImplementation with value newPendingImplementation
      ✔ emits NewPendingImplementation event
  _acceptImplementation
    Check caller is pendingComptrollerImplementation  and pendingComptrollerImplementation ≠ address(0)
      ✔ emits a failure log
      ✔ does not change current implementation address
    the brains must accept the responsibility of implementation
      ✔ Store comptrollerImplementation with value pendingComptrollerImplementation
      ✔ Unset pendingComptrollerImplementation
      ✔ Emit NewImplementation(oldImplementation, newImplementation)
      ✔ Emit NewPendingImplementation(oldPendingImplementation, 0)
    fallback delegates to brains
      ✔ forwards reverts
      ✔ gets addresses
      ✔ gets strings
      ✔ gets bools
      ✔ gets list of ints

CheckpointView tests (using interest rate models as data sources)
  ✔ should revert if dataSource1 address is zero
  ✔ should revert if dataSource2 address is zero
  ✔ should use old rate model before checkpoint (42ms)
  ✔ should use new rate model after checkpoint (42ms)
  ✔ should return the correct current data source

Peg Stability Module
  PSM: 18 decimals
    initialization
      ✔ should revert if contract already deployed
      ✔ should initialize sucessfully
      reverts if init address = 0x0:
        ✔ acm
        ✔ treasury
        ✔ stableToken
      reverts if fee init value is invalid
        ✔ feeIn
        ✔ feeOut
    Admin functions
      pause()
        ✔ should revert if not authorised
        ✔ should pause if authorised
        ✔ should revert if already paused
      resume()
        ✔ should revert if not authorised
        ✔ should resume if authorised
        ✔ should revert if already resumed
      setFeeIn(uint256)
        ✔ should revert if not authorised
        ✔ should revert if fee is invalid
        ✔ set the correct fee
      setFeeOut(uint256)
        ✔ should revert if not authorised
        ✔ should revert if fee is invalid
        ✔ set the correct fee
      setVAIMintCap(uint256)
        ✔ should revert if not authorised
        ✔ should set the correct mint cap
      setVenusTreasury(uint256)
        ✔ should revert if not authorised
        ✔ should revert if zero address
```

```
    ✔ should set the treasury address
  setOracle(address)
    ✔ should revert if not authorised
    ✔ should revert if oracle address is zero
    ✔ should set the oracle (62ms)
Pause logic
  ✔ should revert when paused and call swapVAIForStable(address,uint256)
  ✔ should revert when paused and call swapStableForVAI(address,uint256)
Swap functions
  swapVAIForStable(address,uint256)
    ✔ should revert if receiver is zero address
    ✔ should revert if sender has insufficient VAI balance  (52ms)
    ✔ should revert if VAI transfer fails  (62ms)
    ✔ should revert if VAI to be burnt > vaiMinted  (48ms)
    should sucessfully perform the swap
      Fees: 10%
        ✔ stable token = 1$  (70ms)
        ✔ stable token < 1$  (73ms)
        ✔ stable token > 1$  (93ms)
      Fees: 0%
        ✔ stable token = 1$  (63ms)
        ✔ stable token < 1$  (58ms)
        ✔ stable token > 1$  (64ms)
  swapStableForVAI(address,uint256)
    ✔ should revert if receiver is zero address
    ✔ should revert if VAI mint cap will be reached  (61ms)
    ✔ should revert if amount after transfer is too small   (100ms)
    should sucessfully perform the swap
      Fees: 10%
        ✔ stable token = 1$  (160ms)
        ✔ stable token > 1$  (96ms)
        ✔ stable token < 1$  (91ms)
      Fees: 0%
        ✔ stable token = 1$  (84ms)
        ✔ stable token > 1$  (89ms)
        ✔ stable token < 1$  (78ms)
PSM: 8 decimals
  initialization
    ✔ should revert if contract already deployed
    ✔ should initialize sucessfully
    reverts if init address = 0x0:
      ✔ acm
      ✔ treasury
      ✔ stableToken
    reverts if fee init value is invalid
      ✔ feeIn
      ✔ feeOut
  Admin functions
  pause()
    ✔ should revert if not authorised
    ✔ should pause if authorised
    ✔ should revert if already paused
  resume()
    ✔ should revert if not authorised
    ✔ should resume if authorised
    ✔ should revert if already resumed
  setFeeIn(uint256)
    ✔ should revert if not authorised
    ✔ should revert if fee is invalid
    ✔ set the correct fee
  setFeeOut(uint256)
    ✔ should revert if not authorised
    ✔ should revert if fee is invalid
    ✔ set the correct fee
  setVAIMintCap(uint256)
    ✔ should revert if not authorised
    ✔ should set the correct mint cap
  setVenusTreasury(uint256)
    ✔ should revert if not authorised
    ✔ should revert if zero address
    ✔ should set the treasury address
  setOracle(address)
```

```
    ✔ should revert if not authorised
    ✔ should revert if oracle address is zero
    ✔ should set the oracle (67ms)
Pause logic
  ✔ should revert when paused and call swapVAIForStable(address,uint256)
  ✔ should revert when paused and call swapStableForVAI(address,uint256)
Swap functions
  swapVAIForStable(address,uint256)
    ✔ should revert if receiver is zero address
    ✔ should revert if sender has insufficient VAI balance  (57ms)
    ✔ should revert if VAI transfer fails  (66ms)
    ✔ should revert if VAI to be burnt > vaiMinted  (53ms)
    should sucessfully perform the swap
      Fees: 10%
        ✔ stable token = 1$  (93ms)
        ✔ stable token < 1$  (90ms)
        ✔ stable token > 1$  (95ms)
      Fees: 0%
        ✔ stable token = 1$  (84ms)
        ✔ stable token < 1$  (84ms)
        ✔ stable token > 1$  (91ms)
  swapStableForVAI(address,uint256)
    ✔ should revert if receiver is zero address
    ✔ should revert if VAI mint cap will be reached  (84ms)
    should sucessfully perform the swap
      Fees: 10%
        ✔ stable token = 1$  (98ms)
        ✔ stable token > 1$  (137ms)
        ✔ stable token < 1$  (127ms)
      Fees: 0%
        ✔ stable token = 1$  (136ms)
        ✔ stable token > 1$  (99ms)
        ✔ stable token < 1$  (89ms)
PSM: 6 decimals
  initialization
    ✔ should revert if contract already deployed
    ✔ should initialize sucessfully
    reverts if init address = 0x0:
      ✔ acm
      ✔ treasury
      ✔ stableToken
    reverts if fee init value is invalid
      ✔ feeIn
      ✔ feeOut
  Admin functions
    pause()
      ✔ should revert if not authorised
      ✔ should pause if authorised
      ✔ should revert if already paused
    resume()
      ✔ should revert if not authorised
      ✔ should resume if authorised (38ms)
      ✔ should revert if already resumed
    setFeeIn(uint256)
      ✔ should revert if not authorised
      ✔ should revert if fee is invalid
      ✔ set the correct fee (38ms)
    setFeeOut(uint256)
      ✔ should revert if not authorised
      ✔ should revert if fee is invalid
      ✔ set the correct fee (40ms)
    setVAIMintCap(uint256)
      ✔ should revert if not authorised
      ✔ should set the correct mint cap
    setVenusTreasury(uint256)
      ✔ should revert if not authorised
      ✔ should revert if zero address
      ✔ should set the treasury address (39ms)
    setOracle(address)
      ✔ should revert if not authorised
      ✔ should revert if oracle address is zero
      ✔ should set the oracle (66ms)
```

```
Pause logic
  ✔ should revert when paused and call swapVAIForStable(address,uint256)
  ✔ should revert when paused and call swapStableForVAI(address,uint256)
Swap functions
  swapVAIForStable(address,uint256)
    ✔ should revert if receiver is zero address
    ✔ should revert if sender has insufficient VAI balance  (64ms)
    ✔ should revert if VAI transfer fails  (84ms)
    ✔ should revert if VAI to be burnt > vaiMinted  (61ms)
    should sucessfully perform the swap
      Fees: 10%
        ✔ stable token = 1$  (116ms)
        ✔ stable token < 1$  (113ms)
        ✔ stable token > 1$  (125ms)
      Fees: 0%
        ✔ stable token = 1$  (118ms)
        ✔ stable token < 1$  (101ms)
        ✔ stable token > 1$  (101ms)
  swapStableForVAI(address,uint256)
    ✔ should revert if receiver is zero address
    ✔ should revert if VAI mint cap will be reached  (100ms)
    should sucessfully perform the swap
      Fees: 10%
        ✔ stable token = 1$  (135ms)
        ✔ stable token > 1$  (141ms)
        ✔ stable token < 1$  (145ms)
      Fees: 0%
        ✔ stable token = 1$  (122ms)
        ✔ stable token > 1$  (128ms)
        ✔ stable token < 1$  (111ms)

VAIController
  ✔ check wallet usdt balance (39ms)
  #getMintableVAI
    ✔ oracle
    ✔ getAssetsIn
    ✔ getAccountSnapshot
    ✔ getUnderlyingPrice (45ms)
    ✔ getComtroller
    ✔ success (184ms)
  #mintVAI
    ✔ success (333ms)
    ✔ fails if there's not enough collateral (259ms)
    ✔ fails if minting beyond mint cap (405ms)
    ✔ fails if can't set the minted amount in comptroller (361ms)
    ✔ puts previously accrued interest to pastInterest (709ms)
  #repayVAI
    ✔ reverts if the protocol is paused (40ms)
    ✔ success for zero rate (210ms)
    ✔ success for 1.2 rate repay all (303ms)
    ✔ success for 1.2 rate repay half (319ms)
    ✔ fails if can't set the new minted amount in comptroller (208ms)
  #repayVAIBehalf
    ✔ reverts if called with borrower = zero address
    ✔ reverts if the protocol is paused
    ✔ success for zero rate (200ms)
    ✔ success for 1.2 rate repay all (289ms)
    ✔ success for 1.2 rate repay half (268ms)
  #getHypotheticalAccountLiquidity
    ✔ success for zero rate 0.9 vusdt collateralFactor (299ms)
    ✔ success for 1.2 rate 0.9 vusdt collateralFactor (383ms)
  #liquidateVAI
    ✔ liquidationIncentiveMantissa
    ✔ reverts if the protocol is paused (38ms)
    ✔ success for zero rate 0.2 vusdt collateralFactor (1151ms)
network block skew detected; skipping block events (emitted=7780131 blockNumber100000000)
network block skew detected; skipping block events (emitted=7780131 blockNumber100000000)
network block skew detected; skipping block events (emitted=7780131 blockNumber100000000)
network block skew detected; skipping block events (emitted=7780131 blockNumber100000000)
network block skew detected; skipping block events (emitted=7780131 blockNumber100000000)
network block skew detected; skipping block events (emitted=7780131 blockNumber100000000)
    ✔ success for 1.2 rate 0.3 vusdt collateralFactor (1097ms)
```

```
    #getVAIRepayRate
      ✔ success for zero baseRate
      ✔ success for baseRate 0.1 floatRate 0.1 vaiPirce 1e18 (158ms)
      ✔ success for baseRate 0.1 floatRate 0.1 vaiPirce 0.5 * 1e18 (150ms)
    #getVAIRepayAmount
      ✔ reverts if the protocol is paused
      ✔ success for zero rate (39ms)
      ✔ success for baseRate 0.1 floatRate 0.1 vaiPirce 1e18 (205ms)
      ✔ success for baseRate 0.1 floatRate 0.1 vaiPirce 0.5 * 1e18 (281ms)
    #getVAICalculateRepayAmount
      ✔ success for zero rate (62ms)
      ✔ success for baseRate 0.1 floatRate 0.1 vaiPirce 1e18 (334ms)
      ✔ success for baseRate 0.1 floatRate 0.1 vaiPirce 0.5 * 1e18 (355ms)
    #getMintableVAI
      ✔ include current interest when calculating mintable VAI (369ms)
    #accrueVAIInterest
      ✔ success for called once (138ms)
      ✔ success for called twice (192ms)
    #setBaseRate
      ✔ fails if access control does not allow the call
      ✔ emits NewVAIBaseRate event (38ms)
      ✔ sets new base rate in storage
    #setFloatRate
      ✔ fails if access control does not allow the call
      ✔ emits NewVAIFloatRate event (38ms)
      ✔ sets new float rate in storage
    #setMintCap
      ✔ fails if access control does not allow the call
      ✔ emits NewVAIMintCap event (42ms)
      ✔ sets new mint cap in storage (38ms)
    #setReceiver
      ✔ fails if called by a non-admin
      ✔ reverts if the receiver is zero address
      ✔ emits NewVAIReceiver event
      ✔ sets VAI receiver address in storage
    #setAccessControl
      ✔ reverts if called by non-admin
      ✔ reverts if ACM is zero address
      ✔ emits NewAccessControl event (53ms)
      ✔ sets ACM address in storage (48ms)
    #prime
      ✔ prime integration (2024ms)

  VAIVault
    ✔ claim reward (714ms)
    setVenusInfo
      ✔ fails if called by a non-admin
      ✔ fails if XVS address is zero
      ✔ fails if VAI address is zero
      ✔ disallows configuring tokens twice (39ms)

    VRTVault
      unit tests
        setLastAccruingBlock
          ✔ fails if ACM disallows the call
          ✔ fails if trying to set lastAccuringBlock to some absurdly high value
          ✔ fails if lastAccuringBlock has passed (63ms)
          ✔ fails if trying to set lastAccuringBlock to some past block
          ✔ fails if trying to set lastAccuringBlock to the current block
          ✔ correctly sets lastAccuringBlock to some future block (62ms)
          ✔ can move lastAccuringBlock to a later block (96ms)
          ✔ can move lastAccuringBlock to an earlier block (94ms)
          ✔ fails if trying to move lastAccuringBlock to a block in the past (75ms)
      scenario
        ✔ deposit (129ms)
        ✔ should claim reward (81ms)
        ✔ should not claim reward after certain block (130ms)

    VToken
      _setReserveFactorFresh
        ✔ rejects change by non-admin (40ms)
network block skew detected; skipping block events (emitted=7780133 blockNumber7790294)
```

```
network block skew detected; skipping block events (emitted=7780137 blockNumber7790312)
network block skew detected; skipping block events (emitted=7780137 blockNumber7790312)
network block skew detected; skipping block events (emitted=7780137 blockNumber7790312)
network block skew detected; skipping block events (emitted=7780137 blockNumber7790312)
network block skew detected; skipping block events (emitted=7780137 blockNumber7790312)
network block skew detected; skipping block events (emitted=7780137 blockNumber7790313)
      ✔ rejects change if market not fresh
      ✔ rejects newReserveFactor that descales to 1 (88ms)
      ✔ accepts newReserveFactor in valid range and emits log (91ms)
      ✔ accepts a change back to zero (176ms)
    _setReserveFactor
      ✔ emits a reserve factor failure if interest accrual fails (119ms)
      ✔ returns error from setReserveFactorFresh without emitting any extra logs (94ms)
      ✔ returns success from setReserveFactorFresh (127ms)
    _reduceReservesFresh
      ✔ fails if called by non-admin (61ms)
      ✔ fails if market not fresh (61ms)
      ✔ fails if amount exceeds available cash (469ms)
      ✔ if there isn't enough cash, reduces with available cash (219ms)
      ✔ increases admin balance and reduces reserves on success (228ms)
    _reduceReserves
      ✔ emits a reserve-reduction failure if interest accrual fails (110ms)
      ✔ returns error from _reduceReservesFresh without emitting any extra logs (207ms)
      ✔ returns success code from _reduceReservesFresh and reduces the correct amount (209ms)

  XVSVault
    setXvsStore
      ✔ fails if XVS is a zero address
      ✔ fails if XVSStore is a zero address
      ✔ fails if the vault is already initialized
    add
      ✔ reverts if ACM does not allow the call
      ✔ reverts if xvsStore is not set (40ms)
      ✔ reverts if a pool with this (staked token, reward token) combination already exists (50ms)
      ✔ reverts if staked token exists in another pool (40ms)
      ✔ reverts if reward token is a zero address (42ms)
      ✔ reverts if staked token is a zero address (40ms)
      ✔ reverts if alloc points parameter is zero (39ms)
      ✔ emits PoolAdded event (56ms)
      ✔ adds a second pool to an existing rewardToken (70ms)
      ✔ sets pool info (76ms)
      ✔ configures reward token in XVSStore (80ms)
    set
      ✔ reverts if ACM does not allow the call
      ✔ reverts if pool is not found (40ms)
      ✔ reverts if total alloc points after the call is zero (60ms)
      ✔ succeeds if the pool alloc points is zero but total alloc points is nonzero (228ms)
      ✔ emits PoolUpdated event (60ms)
  setRewardAmountPerBlockOrSecond
    ✔ reverts if ACM does not allow the call (39ms)
    ✔ reverts if the token is not configured in XVSStore (79ms)
    ✔ emits RewardAmountPerBlockUpdated event (77ms)
    ✔ updates reward amount per block (94ms)
  setWithdrawalLockingPeriod
    ✔ reverts if ACM does not allow the call
    ✔ reverts if pool does not exist
    ✔ reverts if the lock period is 0 (42ms)
    ✔ reverts if the lock period is absurdly high
    ✔ emits WithdrawalLockingPeriodUpdated event (77ms)
    ✔ updates lock period (123ms)
  pendingReward
    ✔ includes the old withdrawal requests in the rewards computation (240ms)
    ✔ excludes the new withdrawal requests from the rewards computation (335ms)
  deposit
    ✔ reverts if the vault is paused (62ms)
    ✔ reverts if pool does not exist
    ✔ transfers pool token to the vault (110ms)
    ✔ updates user's balance (98ms)
    ✔ fails if there's a pre-upgrade withdrawal request (151ms)
    ✔ succeeds if the pre-upgrade withdrawal request has been executed (512ms)
    ✔ uses the safe _transferReward under the hood (295ms)
  executeWithdrawal
```

```
      ✔ fails if the vault is paused (61ms)
      ✔ only transfers the requested amount for post-upgrade requests (285ms)
      ✔ handles pre-upgrade withdrawal requests (296ms)
network block skew detected; skipping block events (emitted=7790880 blockNumber7792385)
      ✔ handles pre-upgrade and post-upgrade withdrawal requests (478ms)
    requestWithdrawal
      ✔ fails if the vault is paused (59ms)
      ✔ transfers rewards to the user (282ms)
      ✔ uses the safe _transferReward under the hood (286ms)
      ✔ fails if there's a pre-upgrade withdrawal request (136ms)
    claim
      ✔ fails if there's a pre-upgrade withdrawal request (71ms)
      ✔ succeeds if the pre-upgrade withdrawal request has been executed (313ms)
      ✔ excludes pending withdrawals from the user's shares (389ms)
      ✔ correctly accounts for updates in reward per block (253ms)
network block skew detected; skipping block events (emitted=7792385 blockNumber7800889)
      ✔ uses the safe _transferReward under the hood (167ms)
    _transferReward
      ✔ sends the available funds to the user (140ms)
      ✔ emits VaultDebtUpdated event if vault debt is updated (89ms)
      ✔ does not emit VaultDebtUpdated event if vault debt is not updated (104ms)
      ✔ records the pending transfer (108ms)
      ✔ records several pending transfers (236ms)
      ✔ sends out the pending transfers in addition to reward if full amount <= funds available (392ms)
      ✔ sends a part of the pending transfers and reward if full amount > funds available (369ms)
    pendingWithdrawalsBeforeUpgrade
      ✔ returns zero if there were no pending withdrawals
      ✔ returns zero if there is only a new-style pending withdrawal (154ms)
      ✔ returns the requested amount if there is an old-style pending withdrawal (39ms)
      ✔ returns the total requested amount if there are multiple old-style pending withdrawals (72ms)
      ✔ returns zero if the pending withdrawal was executed (167ms)
    Scenarios
network block skew detected; skipping block events (emitted=7790882 blockNumber7791884)
network block skew detected; skipping block events (emitted=7790882 blockNumber7791884)
network block skew detected; skipping block events (emitted=7790882 blockNumber7791884)
network block skew detected; skipping block events (emitted=7790882 blockNumber7791884)
network block skew detected; skipping block events (emitted=7790882 blockNumber7791884)
network block skew detected; skipping block events (emitted=7790882 blockNumber7791884)
      ✔ works correctly with multiple claim, deposit, and withdrawal requests (1154ms)

  Prime Token
    mint and burn
network block skew detected; skipping block events (emitted=7791884 blockNumber7794710)
network block skew detected; skipping block events (emitted=7791884 blockNumber7794710)
network block skew detected; skipping block events (emitted=7791884 blockNumber7794710)
network block skew detected; skipping block events (emitted=7791884 blockNumber7794710)
network block skew detected; skipping block events (emitted=7791884 blockNumber7794710)
network block skew detected; skipping block events (emitted=7791884 blockNumber7794710)
Warning: Potentially unsafe deployment of
contracts/Tokens/Prime/PrimeLiquidityProvider.sol:PrimeLiquidityProvider

    You are using the `unsafeAllow.internal-function-storage` flag.
    Internal functions are code pointers which will no longer be valid after an upgrade.
    Make sure you reassign internal functions in storage variables during upgrades.

Warning: Potentially unsafe deployment of contracts/Tokens/Prime/Prime.sol:Prime

    You are using the `unsafeAllow.internal-function-storage` flag.
    Internal functions are code pointers which will no longer be valid after an upgrade.
    Make sure you reassign internal functions in storage variables during upgrades.

      ✔ should alias setPrimeToken to _setPrimeToken
      ✔ stake and mint (1355ms)
network block skew detected; skipping block events (emitted=7794710 blockNumber15570745)
network block skew detected; skipping block events (emitted=7794710 blockNumber15570745)
network block skew detected; skipping block events (emitted=7794710 blockNumber15570745)
network block skew detected; skipping block events (emitted=7794710 blockNumber15570745)
network block skew detected; skipping block events (emitted=7794710 blockNumber15570745)
network block skew detected; skipping block events (emitted=7794710 blockNumber15570745)
      ✔ burn revocable token (3094ms)
      ✔ cannot burn irrevocable token (2848ms)
      ✔ issue and stake token concurrently (2224ms)
```

```
    boosted yield
network block skew detected; skipping block events (emitted=7800889 blockNumber15570748)
network block skew detected; skipping block events (emitted=15570748 blockNumber23346763)
network block skew detected; skipping block events (emitted=15570752 blockNumber23346769)
network block skew detected; skipping block events (emitted=15570752 blockNumber23346769)
network block skew detected; skipping block events (emitted=15570752 blockNumber23346769)
network block skew detected; skipping block events (emitted=15570752 blockNumber23346769)
network block skew detected; skipping block events (emitted=15570752 blockNumber23346769)
network block skew detected; skipping block events (emitted=15570752 blockNumber23346769)
      ✔ claim interest for multiple users (7551ms)


  722 passing (8m)
  1 pending
```

Code Coverage

Overall coverage is low (under 60%). We strongly recommend improving coverage a means to potentially catch errors and assumptions that could break when migrating to a newer Solidity version (now or in the future).

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/	100	100	100	100	
InterfacesV8.sol	100	100	100	100	
contracts/Admin/	90.48	40.91	85.71	88.46	
VBNBAdmin.sol	90.48	40.91	85.71	88.46	71,72,73
VBNBAdminStorage.sol	100	100	100	100	
contracts/Comptroller/	100	90	100	100	
ComptrollerInterface.sol	100	100	100	100	
ComptrollerLensInterface.sol	100	100	100	100	
ComptrollerStorage.sol	100	100	100	100	
Unitroller.sol	100	90	100	100	
contracts/Comptroller/Diamond/	97.26	61.36	100	95.35	
Diamond.sol	97.26	61.36	100	95.35	109,228,229,230
DiamondConsolidated.sol	100	100	100	100	
contracts/Comptroller/Diamond/facets/	75.6	63.67	80.91	76.22	
FacetBase.sol	62.22	55.88	86.67	59.18	... 132,215,228
MarketFacet.sol	98.8	66.67	94.12	98.98	67
PolicyFacet.sol	85.5	72.86	100	85.93	... 385,406,407

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
RewardFacet.sol	1.67	0	10	1.52	... 236,237,248
SetterFacet.sol	83.44	75	79.55	83.89	... 575,576,659
XVSRewardsHelper.sol	94.12	80	100	95.45	79,108
contracts/Comptroller/Diamond/interfaces/	100	100	100	100	
IDiamondCut.sol	100	100	100	100	
IFacetBase.sol	100	100	100	100	
IMarketFacet.sol	100	100	100	100	
IPolicyFacet.sol	100	100	100	100	
IRewardFacet.sol	100	100	100	100	
ISetterFacet.sol	100	100	100	100	
contracts/DelegateBorrowers/	100	89.47	100	100	
MoveDebtDelegate.sol	100	91.67	100	100	
SwapDebtDelegate.sol	100	85.71	100	100	
contracts/Governance/	73.15	44.87	68.18	68	
TokenRedeemer.sol	97.53	70	100	91.4	... 169,176,180
VTreasury.sol	0	0	0	0	... 65,67,70,72
VTreasuryV8.sol	0	0	0	0	... 90,91,98,99
contracts/InterestRateModels/	72.55	59.09	64.29	74.03	
InterestRateModel.sol	100	100	100	100	
InterestRateModelV8.sol	100	100	100	100	
JumpRateModel.sol	71.43	100	75	78.95	114,115,116,117
TwoKinksInterestRateModel.sol	100	56.25	100	93.33	100,104,173
WhitePaperInterestRateModel.sol	0	0	0	0	... 88,89,90,91
contracts/Lens/	43.75	36.54	33.33	45.23	
ComptrollerLens.sol	91.89	81.25	100	94.64	99,158,167
SnapshotLens.sol	0	0	0	0	... 122,124,146

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
VenusLens.sol	35.83	20	29.41	37.33	... 465,498,562
contracts/Liquidator/	83.95	60.47	86.05	83.25	
BUSDLiquidator.sol	97.56	64.29	91.67	98.08	99
Liquidator.sol	79.34	59.72	83.87	78.34	... 505,506,507
LiquidatorStorage.sol	100	100	100	100	
contracts/Oracle/	100	100	100	100	
PriceOracle.sol	100	100	100	100	
contracts/PegStability/	87.91	84.48	85	88.1	
IVAI.sol	100	100	100	100	
PegStability.sol	87.91	84.48	85	88.1	... 424,425,428
contracts/Swap/	92.68	57.14	96.49	87.38	
IRouterHelper.sol	100	100	100	100	
RouterHelper.sol	98.75	70.83	100	92.52	... 308,312,323
SwapRouter.sol	89.76	54.13	95.35	84.65	... 942,943,944
contracts/Swap/interfaces/	100	100	100	100	
CustomErrors.sol	100	100	100	100	
IPancakePair.sol	100	100	100	100	
IPancakeSwapV2Factory.sol	100	100	100	100	
IPancakeSwapV2Router.sol	100	100	100	100	
IVBNB.sol	100	100	100	100	
IVtoken.sol	100	100	100	100	
IWBNB.sol	100	100	100	100	
InterfaceComptroller.sol	100	100	100	100	
contracts/Swap/lib/	100	52.63	100	81.03	
PancakeLibrary.sol	100	50	100	82.61	... 121,141,167
TransferHelper.sol	100	62.5	100	75	18,33,62

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/Tokens/	100	100	100	100	
EIP20Interface.sol	100	100	100	100	
EIP20NonStandardInterface.sol	100	100	100	100	
contracts/Tokens/Prime/	95.61	71.59	95.65	96.38	
IPrime.sol	100	100	100	100	
IPrimeV5.sol	100	100	100	100	
Prime.sol	94.87	68.94	95.83	96.35	... 4,1024,1133
PrimeLiquidityProvider.sol	97.65	79.55	95.24	96.49	123,215,309,351
PrimeLiquidityProviderStorage.sol	100	100	100	100	
PrimeStorage.sol	100	100	100	100	
contracts/Tokens/Prime/Interfaces/	100	100	100	100	
IPoolRegistry.sol	100	100	100	100	
IPrime.sol	100	100	100	100	
IPrimeLiquidityProvider.sol	100	100	100	100	
IVToken.sol	100	100	100	100	
IXVSVault.sol	100	100	100	100	
InterfaceComptroller.sol	100	100	100	100	
contracts/Tokens/Prime/libs/	90.38	75.76	100	89.86	
FixedMath.sol	100	50	100	100	
FixedMath0x.sol	88.24	76	100	88.52	... 211,217,223
Scores.sol	90	90	100	100	
contracts/Tokens/VAI/	78.7	52.73	85.96	81.45	
IVAI.sol	100	100	100	100	
VAI.sol	57.69	30	66.67	65.85	... 156,157,158
VAIController.sol	84.96	59.3	97.14	87.16	... 575,576,578
VAIControllerInterface.sol	100	100	100	100	
VAIControllerStorage.sol	100	100	100	100	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
VAIUnitroller.sol	44	25	66.67	48.48	... 123,124,126
lib.sol	100	100	100	100	
contracts/Tokens/VRT/	21.32	8.87	25.64	19.66	
VRT.sol	36.71	18.97	52.63	38.04	... 305,306,309
VRTConverter.sol	0	0	0	0	... 153,158,159
VRTConverterProxy.sol	0	0	0	0	... 168,178,180
VRTConverterStorage.sol	100	100	100	100	
contracts/Tokens/VTokens/	61.39	47.29	53.6	64.01	
VBNB.sol	0	0	0	0	... 180,181,183
VBep20.sol	63.33	0	62.5	64.52	... 140,141,181
VBep20Delegate.sol	50	25	66.67	42.86	29,40,41,44
VBep20Delegator.sol	26.92	50	26.32	32.14	... 460,498,501
VBep20Immutable.sol	100	100	100	100	
VToken.sol	72.75	51.3	80	74.57	... 9,1670,1675
VTokenInterfaces.sol	100	100	100	100	
contracts/Tokens/VTokens/legacy/	0	0	0	0	
ComptrollerInterface.sol	100	100	100	100	
IProtocolShareReserveV5.sol	100	100	100	100	
VBep20DelegateR1.sol	0	0	0	0	... 30,38,39,42
VBep20DelegatorR1.sol	0	0	0	0	... 522,523,528
VBep20R1.sol	0	0	0	0	... 259,275,284
VTokenInterfaceR1.sol	100	100	100	100	
VTokenR1.sol	0	0	0	0	... 3,1687,1691
VTokenStorageR1.sol	100	100	100	100	
contracts/Tokens/VTokens/legacy/Utils/	0	0	0	0	
CarefulMath.sol	0	0	0	0	... 76,78,79,82

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
ErrorReporter.sol	0	100	0	0	... 272,279,281
Exponential.sol	0	0	0	0	... 168,170,179
ExponentialNoError.sol	0	0	0	0	... 188,189,193
contracts/Tokens/XVS/	19.08	8.46	23.26	18.52	
IXVS.sol	100	100	100	100	
IXVSVesting.sol	100	100	100	100	
XVS.sol	36.71	18.97	52.63	38.04	... 305,306,309
XVSVesting.sol	0	0	0	0	... 218,223,224
XVSVestingProxy.sol	0	0	0	0	... 151,161,163
XVSVestingStorage.sol	100	100	100	100	
contracts/Utils/	51.07	30.43	50.91	52.21	
Address.sol	42.86	0	33.33	50	44,66,70,71
CarefulMath.sol	80	66.67	100	84	35,46,77,88
CheckpointView.sol	100	100	100	100	
Context.sol	0	100	0	0	19,23,24
ECDSA.sol	0	0	0	0	... 6,97,98,101
ErrorReporter.sol	50	100	50	50	... 273,280,282
Exponential.sol	54	40.91	53.85	54	... 169,171,180
ExponentialNoError.sol	76.09	62.5	67.65	76.09	... 173,177,181
IBEP20.sol	100	100	100	100	
Ownable.sol	0	0	0	0	... 63,70,71,72
Owned.sol	0	0	33.33	20	13,14,18,19
SafeBEP20.sol	53.85	33.33	50	53.85	... 41,42,46,50
SafeCast.sol	8.33	4.17	8.33	8.33	... 193,204,205
SafeMath.sol	85	58.33	77.78	85	145,160,161
Tokenlock.sol	33.33	16.67	33.33	33.33	18,20,24,26
contracts/VAIVault/	45.95	45.16	51.85	50.49	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
VAIVault.sol	75.56	51.85	73.68	78.79	... 217,218,236
VAIVaultErrorReporter.sol	0	100	0	0	26,28,35,37
VAIVaultProxy.sol	0	0	0	0	... 125,135,137
VAIVaultStorage.sol	100	100	100	100	
contracts/VRTVault/	47.66	36.29	53.33	48.59	
VRTVault.sol	62.2	40.18	72.73	64.49	... 277,304,305
VRTVaultProxy.sol	0	0	0	0	... 144,154,156
VRTVaultStorage.sol	100	100	100	100	
contracts/XVSVault/	60.67	50	55.71	63.37	
XVSStore.sol	60	46.15	66.67	60.71	... 1,93,94,125
XVSVault.sol	67.73	53.13	62.26	71.2	... 868,921,922
XVSVaultErrorReporter.sol	0	100	0	0	26,28,35,37
XVSVaultProxy.sol	0	0	0	0	... 125,135,137
XVSVaultStorage.sol	100	100	100	100	
contracts/external/	100	100	100	100	
IProtocolShareReserve.sol	100	100	100	100	
IWBNB.sol	100	100	100	100	
contracts/lib/	100	71.43	100	88.89	
Currency.sol	100	90	100	92.86	58
approveOrRevert.sol	100	25	100	75	25
All files	59.39	44.57	56.91	60.91	

Changelog

- 2025-08-26 - Initial report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

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 or other making available of the report to you by Quantstamp.

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. 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 any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed 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. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. 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. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. 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.