



Dahlia Finance Security Review Public Report

PROJECT: Dahlia Finance Security Review
July 2021

Prepared For:

Dahlia Finance

robert@dahlia.finance

Prepared By:

Jonathan Haas | Bramah Systems, LLC.

jonathan@bramah.systems



Table of Contents

Executive Summary	3
Scope of Engagement	3
Timeline	3
Engagement Goals	3
Contract Specification	3
Overall Assessment	4
Timeliness of Content	5
General Recommendations	6
Solidity version out of date	6
Multiple TODO items remain	6
Unused functions should be removed	6
Specific Recommendations	7
Protocol forks a compromised protocol	7
References to Ethereum and ETH should be updated	7
Rounding errors could lead to potential concern	7
Numerous typographical errors	8
Hardcoded oracle address could have setter function	8
Unbound for loops can lead to resource exhaustion	8
Toolset Warnings	8
Overview	9
Compilation Warnings	9
Test Coverage	9
Static Analysis Coverage	9
Directory Structure	10



Dahlia Finance Protocol Review

Executive Summary

Scope of Engagement

Bramah Systems, LLC was engaged in July of 2021 to perform a comprehensive security review of the Dahlia Finance smart contracts (specific contracts denoted within the appendix). Our review was conducted over a period of five business days by both members of the Bramah Systems, LLC. executive staff.

Bramah Systems completed the assessment using manual, static and dynamic analysis techniques.

Timeline

Review Commencement: July 12, 2021

Report Delivery: July 16, 2021

Engagement Goals

The primary scope of the engagement was to evaluate and establish the overall security of the Dahlia Finance protocol, with a specific focus on trading actions. In specific, the engagement sought to answer the following questions:

- Is it possible for an attacker to steal or freeze tokens?
- Does the Solidity code match the specification as provided?
- Is there a way to interfere with the contract mechanisms?
- Are the arithmetic calculations trustworthy?

Contract Specification

Specification was provided in the form of code comments. The contracts were provided via GitHub (commit hash **e0f68c18f415cbea589d0f553b2ecdb979d64a68** and **1d1e72d82eb8bb7183c14de5ff4db2ed321993a0**)



Overall Assessment

Bramah Systems was engaged to evaluate and identify any potential security concerns within the codebase of the Dahlia Finance protocol. During the course of our engagement, Bramah Systems found multiple instances wherein the team deviated materially from established best practices and procedures of secure software development within DLT. The team has since addressed these issues with a resolution or risk acceptance.



Disclaimer

As of the date of publication, the information provided in this report reflects the presently held, commercially reasonable understanding of Bramah Systems, LLC.'s knowledge of security patterns as they relate to the Dahlia Finance Protocol, with the understanding that distributed ledger technologies ("DLT") remain under frequent and continual development, and resultantly carry with them unknown technical risks and flaws. The scope of the review provided herein is limited solely to items denoted within "Scope of Engagement" and contained within "Directory Structure". The report does NOT cover, review, or opine upon security considerations unique to the Solidity compiler, tools used in the development of the protocol, or distributed ledger technologies themselves, or to any other matters not specifically covered in this report.

The contents of this report must NOT be construed as investment advice or advice of any other kind. This report does NOT have any bearing upon the potential economics of the Dahlia Finance protocol or any other relevant product, service or asset of Dahlia Finance or otherwise. This report is not and should not be relied upon by Dahlia Finance or any reader of this report as any form of financial, tax, legal, regulatory, or other advice.

To the full extent permissible by applicable law, Bramah Systems, LLC. disclaims all warranties, express or implied. The information in this report is provided "as is" without warranty, representation, or guarantee of any kind, including the accuracy of the information provided. Bramah Systems, LLC. makes no warranties, representations, or guarantees about the Dahlia Finance Protocol. Use of this report and/or any of the information provided herein is at the users sole risk, and Bramah Systems, LLC. hereby disclaims, and each user of this report hereby waives, releases, and holds Bramah Systems, LLC. harmless from, any and all liability, damage, expense, or harm (actual, threatened, or claimed) from such use.

Timeliness of Content

All content within this report is presented only as of the date published or indicated, to the commercially reasonable knowledge of Bramah Systems, LLC. as of such date, and may be superseded by subsequent events or for other reasons. The content contained within this report is subject to change without notice. Bramah Systems, LLC. does not guarantee or warrant the accuracy or timeliness of any of the content contained within this report, whether accessed through digital means or otherwise.

Bramah Systems, LLC. is not responsible for setting individual browser cache settings nor can it ensure any parties beyond those individuals directly listed within this report are receiving the most recent content as reasonably understood by Bramah Systems, LLC. as of the date this report is provided to such individuals.



General Recommendations

Best Practices & Software Development Guidelines

Solidity version out of date

The Solidity version used within the contracts is out of date (**0.5.16**) and should be updated in order to include new compiler optimizations and security improvements.

Resolution: The team does not intend to update the software version.

Multiple TODO items remain

A number of **TODO** items remain in the source code, particularly related to any Python related tooling. As they appear to have material impact upon the function of the protocol, these items should be resolved prior to deployment.

Resolution: Remove and complete TODO items.

Unused function parameters should be removed

A number of function parameters reference that they are unused. Where present, such parameters should be removed altogether in order to promote overall clarity and remove potential exploit vectors.

Resolution: Compounded functions unused, now removed.



Specific Recommendations

Unique to the Dahlia Finance Protocol

Protocol forks a compromised protocol

The basis for the protocol, while having received prior audits, was [compromised](#) via flash-loan vulnerabilities. These concerns should be taken into account when assessing overall security of the protocol.

In particular, the team should acknowledge that the following concerns have been addressed.

- Rounding error has been resolved in borrow code
- Allowance for use of custom spells has been removed
- Public access to resolveReserve function has been removed

Resolution: These issues have all been addressed and resolved.

References to Ethereum and ETH should be updated

In order to avoid generalized confusion, as the protocol is not being deployed on an Ethereum chain, references to Ethereum should be removed or otherwise augmented.

Resolution: All references to eth from the homora_v2_contracts and fountain_of_youth repos have been removed

Numerous typographical errors



Throughout the protocol, numerous typographical errors exist in code comments and documentation. This should be addressed before publication to mainnet.

Resolution: Addressed and resolved.

Hardcoded oracle address could have setter function

There are multiple instances of hardcoded oracle addresses. Where applicable, these addresses could be modified to be set through the usage of a setter function.

Resolution: We do this for transparency.

Unbound for loops can lead to resource exhaustion

Throughout the protocol, there are a number of for loops which do not possess an upper bound on the number of iterations. This can lead to resource (gas) exhaustion if there are too many iterations through the loops. An upper bound where appropriate should be applied to avoid these potential concerns.

Resolution: An external function that only looks at debt positions.



Toolset Warnings

Unique to the Dahlia Finance Protocol

Overview

In addition to our manual review, our process involves utilizing static analysis and formal methods in order to perform additional verification of the presence of security vulnerabilities (or lack thereof). An additional part of this review phase consists of reviewing any automated unit testing frameworks that exist.

The following sections detail warnings generated by the automated tools and confirmation of false positives where applicable.

Compilation Warnings

No compilation warnings were encountered during the course of our audit.

Test Coverage

The contracts possess a number of functional unit tests encompassing various stages of the application lifecycle.

Static Analysis Coverage

The contract repository underwent heavy scrutiny with multiple static analysis agents, including:

- [Securify](#)
- [MAIAN](#)
- [Mythril](#)
- [Oyente](#)
- [Slither](#)



Directory Structure

At time of review, the directory structure of the Dahlia Finance smart contracts repository appeared as it does below. Our review, at request of Dahlia Finance, covers the Solidity code (*.sol) as of commit hash **e0f68c18f415cbea589d0f553b2ecdb979d64a68** and **1d1e72d82eb8bb7183c14de5ff4db2ed321993a0**

```
.
├── Dockerfile
├── LICENSE
├── Makefile
├── README.md
├── audits
│   └── trailofbits-CREAMSummary.pdf
├── check
├── contracts
│   ├── CErc20.sol
│   ├── CErc20Immutable.sol
│   ├── CToken.sol
│   ├── CTokenInterfaces.sol
│   ├── CarefulMath.sol
│   ├── Comptroller.sol
│   ├── ComptrollerInterface.sol
│   ├── ComptrollerStorage.sol
│   ├── EIP20Interface.sol
│   ├── EIP20NonStandardInterface.sol
│   ├── ErrorReporter.sol
│   ├── Exponential.sol
│   ├── InterestRateModel.sol
│   ├── JumpRateModelV2.sol
│   └── SafeMath.sol
```



- | | — Unitroller.sol
- | | — tests
 - | | — ERC20.sol
- | — deploy
- | | — 001_init.ts
- | | — 002_dtoken.ts
- | — deployments
 - | | — mainnet
 - | | — ComptrollerImpl.json
 - | | — JumpRateModelV2GS.json
 - | | — JumpRateModelV2MS.json
 - | | — Unitroller.json
 - | | — fBTC.json
 - | | — fCELO.json
 - | | — fUBE.json
 - | | — solcInputs
 - | | — 7210072da418b33cbdd6abb9c6b4cf38.json
- | — docs
 - | | — CompoundProtocol.pdf
- | — hardhat.config.ts
- | — jest.config.js
- | — package.json
- | — reporterConfig.json
- | — saddle.config.js
- | — scenario
 - | | — Grammar.pegjs
 - | | — SCENARIO.md
 - | | — package.json
 - | | — script



- | | | — generate_parser
- | | | — repl
- | | | — tsc
- | | | — webpack
- | | — src
- | | | — Accounts.ts
- | | | — Action.ts
- | | | — Assert.ts
- | | | — Builder
- | | | | — CTokenBuilder.ts
- | | | | — CTokenDelegateBuilder.ts
- | | | | — CompBuilder.ts
- | | | | — ComptrollerImplBuilder.ts
- | | | | — Erc20Builder.ts
- | | | | — InterestRateModelBuilder.ts
- | | | | — PriceOracleBuilder.ts
- | | | | — PriceOracleProxyBuilder.ts
- | | | | — UnitrollerBuilder.ts
- | | | — Command.ts
- | | | — Completer.ts
- | | | — Contract
- | | | | — CErc20Delegate.ts
- | | | | — CErc20Delegator.ts
- | | | | — CToken.ts
- | | | | — Comp.ts
- | | | | — CompoundLens.ts
- | | | | — Comptroller.ts
- | | | | — ComptrollerImpl.ts
- | | | | — Counter.ts



- | | | └─ Erc20.ts
- | | | └─ InterestRateModel.ts
- | | | └─ Pot.ts
- | | | └─ PriceOracle.ts
- | | | └─ PriceOracleProxy.ts
- | | | └─ Unitroller.ts
- | | | └─ Vat.ts
- | | | └─ builder.js
- | | └─ Contract.ts
- | | └─ ContractLookup.ts
- | | └─ CoreEvent.ts
- | | └─ CoreValue.ts
- | | └─ Encoding.ts
- | | └─ ErrorReporter.ts
- | | └─ ErrorReporterConstants.ts
- | | └─ Event
- | | | └─ AssertionEvent.ts
- | | | └─ CTokenDelegateEvent.ts
- | | | └─ CTokenEvent.ts
- | | | └─ CompEvent.ts
- | | | └─ ComptrollerEvent.ts
- | | | └─ ComptrollerImplEvent.ts
- | | | └─ Erc20Event.ts
- | | | └─ ExpectationEvent.ts
- | | | └─ InterestRateModelEvent.ts
- | | | └─ InvariantEvent.ts
- | | | └─ PriceOracleEvent.ts
- | | | └─ PriceOracleProxyEvent.ts
- | | | └─ TrxEvent.ts



- | | | └─ UnitrollerEvent.ts
- | | └─ Event.ts
- | | └─ EventBuilder.ts
- | | └─ Expectation
- | | | └─ ChangesExpectation.ts
- | | | └─ RemainsExpectation.ts
- | | └─ Expectation.ts
- | | └─ File.ts
- | | └─ Formatter.ts
- | | └─ Help.ts
- | | └─ HistoricReadline.ts
- | | └─ Hypothetical.ts
- | | └─ Invariant
- | | | └─ RemainsInvariant.ts
- | | | └─ StaticInvariant.ts
- | | | └─ SuccessInvariant.ts
- | | └─ Invariant.ts
- | | └─ Invokation.ts
- | | └─ Macro.ts
- | | └─ Networks.ts
- | | └─ Parser.ts
- | | └─ Printer.ts
- | | └─ Repl.d.ts
- | | └─ Repl.ts
- | | └─ Runner.ts
- | | └─ Settings.ts
- | | └─ Utils.ts
- | | └─ Value
- | | | └─ CTokenDelegateValue.ts



- | | | | — CTokenValue.ts
- | | | | — CompValue.ts
- | | | | — ComptrollerImplValue.ts
- | | | | — ComptrollerValue.ts
- | | | | — Erc20Value.ts
- | | | | — InterestRateModelValue.ts
- | | | | — PriceOracleProxyValue.ts
- | | | | — PriceOracleValue.ts
- | | | | — UnitrollerValue.ts
- | | | | — UserValue.ts
- | | | — Value.ts
- | | | — Verify.ts
- | | | — Web.ts
- | | | — World.ts
- | | — tsconfig.json
- | | — webpack.config.js
- | | — yarn.lock
- | — script
- | | — README.md
- | | — build_scenarios
- | | — compile
- | | — coverage
- | | — lint
- | | — saddle
- | | | — contractSizer.js
- | | | — deployToken.js
- | | | — flywheelInit.js
- | | | — matchToken.js
- | | | — support



- | | | └─ tokenConfig.js
- | | └─ verifyToken.js
- | └─ scen
- | | └─ deploy.scen
- | └─ test
- | └─ verify
- └─ spec
- | └─ certora
- | | └─ CErc20
- | | | └─ accrueInterest.cvl
- | | | └─ admin.cvl
- | | | └─ borrowAndRepayFresh.cvl
- | | | └─ eip20.cvl
- | | | └─ exchangeRate.cvl
- | | | └─ frame.cvl
- | | | └─ liquidateFresh.cvl
- | | | └─ mintAndRedeemFresh.cvl
- | | | └─ rounding.cvl
- | | └─ CErc20Delegator -> CErc20
- | | └─ Comp
- | | | └─ search.cvl
- | | | └─ transfer.cvl
- | | └─ Comptroller
- | | | └─ comptroller.cvl
- | | └─ Math
- | | | └─ int.cvl
- | | └─ Maximillion
- | | | └─ maximillion.cvl
- | | └─ cDAI



- | | | └─ mcd.cvl
- | | └─ contracts
- | | └─ CDaiDelegateCertora.sol
- | | └─ CErc20DelegateCertora.sol
- | | └─ CErc20DelegatorCertora.sol
- | | └─ CErc20ImmutableCertora.sol
- | | └─ CEtherCertora.sol
- | | └─ CTokenCollateral.sol
- | | └─ CompCertora.sol
- | | └─ ComptrollerCertora.sol
- | | └─ InterestRateModelModel.sol
- | | └─ MathCertora.sol
- | | └─ PriceOracleModel.sol
- | | └─ SimulationInterface.sol
- | | └─ UnderlyingModelNonStandard.sol
- | | └─ UnderlyingModelWithFee.sol
- | └─ scenario
- | | └─ AddReserves.scen
- | | └─ Borrow.scen
- | | └─ Borrow.scen.old
- | | └─ BorrowBalance.scen
- | | └─ BorrowCap.scen
- | | └─ BorrowEth.scen
- | | └─ BorrowWBTC.scen
- | | └─ BreakLiquidate.scen
- | | └─ CTokenAdmin.scen
- | | └─ ChangeDelegate.scen
- | | └─ CollateralCap.scen
- | | └─ Comp



- | | | └─ Comp.scen
- | | └─ CoreMacros
- | | └─ EnterExitMarkets.scen
- | | └─ ExchangeRate.scen
- | | └─ Fee.scen
- | | └─ Gas.scen
- | | └─ HypotheticalAccountLiquidity.scen
- | | └─ InKindLiquidation.scen
- | | └─ Liquidate.scen.old
- | | └─ Mint.scen
- | | └─ MintEth.scen
- | | └─ MintWBTC.scen
- | | └─ ReEntry.scen
- | | └─ Redeem.scen
- | | └─ RedeemEth.scen
- | | └─ RedeemUnderlying.scen
- | | └─ RedeemUnderlyingEth.scen
- | | └─ RedeemUnderlyingWBTC.scen
- | | └─ RedeemWBTC.scen
- | | └─ ReduceReserves.scen
- | | └─ RepayBorrow.scen
- | | └─ RepayBorrowEth.scen
- | | └─ RepayBorrowWBTC.scen
- | | └─ Seize.scen
- | | └─ SetComptroller.scen
- | | └─ Supply.scen.old
- | | └─ SupplyCap.scen
- | | └─ Tether.scen
- | | └─ TokenTransfer.scen



- | | └─ Withdraw.scen.old
- | └─ sim
- | └─ 0001-comp-distribution-patch
- | | └─ deploy.scen
- | | └─ hypothetical_upgrade.scen
- | | └─ hypothetical_upgrade_post_deploy.scen
- | | └─ hypothetical_upgrade_post_propose.scen
- | └─ 0003-borrow-caps-patch
- | | └─ deploy.scen
- | | └─ hypothetical_upgrade.scen
- | | └─ hypothetical_upgrade_post_deploy.scen
- | | └─ hypothetical_upgrade_post_propose.scen
- └─ test
- | └─ lend.test.ts
- └─ tests
- | └─ CompilerTest.js
- | └─ Comptroller
- | | └─ accountLiquidityTest.js
- | | └─ adminTest.js
- | | └─ assetsListTest.js
- | | └─ comptrollerTest.js
- | | └─ pauseGuardianTest.js
- | | └─ unitrollerTest.js
- | └─ Contracts
- | | └─ CErc20Harness.sol
- | | └─ ComptrollerHarness.sol
- | | └─ ComptrollerScenario.sol
- | | └─ Const.sol
- | | └─ Counter.sol



- | | └─ ERC20.sol
- | | └─ EvilToken.sol
- | | └─ FalseMarker.sol
- | | └─ FaucetToken.sol
- | | └─ Fauceteer.sol
- | | └─ FeeToken.sol
- | | └─ InterestRateModelHarness.sol
- | | └─ MathHelpers.sol
- | | └─ MockAggregator.sol
- | | └─ MockMCD.sol
- | | └─ MockMasterChef.sol
- | | └─ MockSushiBar.sol
- | | └─ Structs.sol
- | | └─ TetherInterface.sol
- | | └─ WBTC.sol
- | | └─ WrappedNative.sol
- | └─ Errors.js
- | └─ Jest.js
- | └─ Matchers.js
- | └─ Models
- | | └─ InterestRateModelTest.js
- | └─ Scenario.js
- | └─ SpinaramaTest.js
- | └─ Tokens
- | | └─ accountPrivilegeTest.js
- | | └─ accrueInterestTest.js
- | | └─ adminTest.js
- | | └─ borrowAndRepayTest.js
- | | └─ cTokenTest.js



- | | └─ mintAndRedeemTest.js
- | | └─ reservesTest.js
- | | └─ setComptrollerTest.js
- | | └─ setInterestRateModelTest.js
- | | └─ transferTest.js
- | └─ Utils
- | └─ Compound.js
- | └─ EIP712.js
- | └─ Ethereum.js
- | └─ InfuraProxy.js
- | └─ JS.js
- └─ tsconfig.json
- └─ yarn.lock

43 directories, 283 files

- .
- └─ LICENSE
- └─ README.md
- └─ audits
- | └─ Alpha-Homora-v2-Peckshield-audit-report.pdf
- | └─ Alpha-Homora-v2-Quantstamp-audit-report.pdf
- | └─ PeckShield-Audit-Report-AlphaStaking-AggregatorOracle-v1.0.pdf
- └─ brownie-config.yaml
- └─ contracts
- | └─ Governable.sol
- | └─ HomoraBank.sol
- | └─ IbETHRouterV2.sol
- | └─ SafeBox.sol



- | └─ mock
 - | | └─ MockCErc20.sol
 - | | └─ MockCErc20_2.sol
 - | | └─ MockERC20.sol
 - | | └─ MockUniswapV2Factory.sol
 - | | └─ MockUniswapV2Router02.sol
 - | | └─ MockWETH.sol
- | └─ oracle
 - | | └─ AggregatorOracle.sol
 - | | └─ BalancerPairOracle.sol
 - | | └─ BandAdapterOracle.sol
 - | | └─ BaseKP3ROracle.sol
 - | | └─ ChainlinkAdapterOracle.sol
 - | | └─ CoreOracle.sol
 - | | └─ CurveOracle.sol
 - | | └─ ERC20KP3ROracle.sol
 - | | └─ ProxyOracle.sol
 - | | └─ SimpleOracle.sol
 - | | └─ UbeswapV1Oracle.sol
 - | | └─ UniswapV2Oracle.sol
 - | | └─ UniswapV2OracleLibrary.sol
 - | | └─ UsingBaseOracle.sol
- | └─ spell
 - | | └─ BalancerSpellV1.sol
 - | | └─ BasicSpell.sol
 - | | └─ CurveSpellV1.sol
 - | | └─ HouseholdSpell.sol
 - | | └─ SushiswapSpellV1.sol
 - | | └─ UniswapV2SpellV1.sol



- | | └─ WhitelistSpell.sol
- | └─ utils
- | | └─ BConst.sol
- | | └─ BNum.sol
- | | └─ ERC1155NaiveReceiver.sol
- | | └─ FixedPoint.sol
- | | └─ HomoraMath.sol
- | └─ wrapper
 - | └─ WERC20.sol
 - | └─ WLiquidityGauge.sol
 - | └─ WMasterChef.sol
 - | └─ WStakingRewards.sol
- └─ flake.lock
- └─ flake.nix
- └─ interfaces
 - | └─ IAny.sol
 - | └─ IBalancerPool.sol
 - | └─ IBank.sol
 - | └─ IBaseOracle.sol
 - | └─ ICerc20.sol
 - | └─ ICerc20_2.sol
 - | └─ ICEtherEx.sol
 - | └─ IComptroller.sol
 - | └─ ICurvePool.sol
 - | └─ ICurveRegistry.sol
 - | └─ IERC20Ex.sol
 - | └─ IERC20Wrapper.sol
 - | └─ IERC20WrapperOld.sol
 - | └─ IInterestRateModel.sol



- | └─ IKeep3rV1Oracle.sol
- | └─ ILiquidityGauge.sol
- | └─ IMasterChef.sol
- | └─ IOracle.sol
- | └─ IStakingRewards.sol
- | └─ IStakingRewardsEx.sol
- | └─ IUbeswapV1Oracle.sol
- | └─ IUniswapV2Factory.sol
- | └─ IUniswapV2Pair.sol
- | └─ IUniswapV2Router01.sol
- | └─ IUniswapV2Router02.sol
- | └─ IWERC20.sol
- | └─ IWETH.sol
- | └─ IWLiquidityGauge.sol
- | └─ IWMasterChef.sol
- | └─ IWStakingRewards.sol
- └─ package-lock.json
- └─ package.json
- └─ requirements.txt
- └─ scripts
 - | └─ balancer_oracle_test.py
 - | └─ balancer_spell_add_remove_1x_test.py
 - | └─ balancer_spell_add_remove_test.py
 - | └─ balancer_spell_wstaking_add_remove_1x_test.py
 - | └─ balancer_spell_wstaking_add_remove_test.py
 - | └─ balancer_spell_wstaking_add_twice_test.py
 - | └─ balancer_spell_wstaking_harvest_test.py
 - | └─ check_output.py
 - | └─ curve_oracle_test.py



- | └─ curve_spell_wgauge_2_add_remove_1x_test.py
- | └─ curve_spell_wgauge_2_add_remove_test.py
- | └─ curve_spell_wgauge_2_add_twice_test.py
- | └─ curve_spell_wgauge_2_harvest_test.py
- | └─ curve_spell_wgauge_3_add_remove_1x_test.py
- | └─ curve_spell_wgauge_3_add_remove_test.py
- | └─ curve_spell_wgauge_3_add_twice_test.py
- | └─ curve_spell_wgauge_3_harvest_test.py
- | └─ curve_spell_wgauge_4_add_remove_1x_test.py
- | └─ curve_spell_wgauge_4_add_remove_test.py
- | └─ curve_spell_wgauge_4_add_twice_test.py
- | └─ curve_spell_wgauge_4_harvest_test.py
- | └─ dahlia_add_oracle_params.py
- | └─ dahlia_add_uni_spell.py
- | └─ dahlia_addresses.json
- | └─ dahlia_deploy_oracle_and_bank.py
- | └─ dahlia_deploy_oracle_config.py
- | └─ dahlia_deploy_safeboxes.py
- | └─ dahlia_deploy_uniswap_spell.py
- | └─ dahlia_deploy_wrapped_staking.py
- | └─ dahlia_test_oracle.py
- | └─ dahlia_test_safebox.py
- | └─ dahlia_test_uni_spell_add_remove.py
- | └─ dahlia_test_uni_spell_add_remove_1x.py
- | └─ dahlia_test_uni_spell_more_add_remove.py
- | └─ dahlia_test_uniswap_wstaking.py
- | └─ dahlia_update_oracle.py
- | └─ deploy_agg_oracle.py
- | └─ deploy_oracle.py



- | └─ deploy_to_mainnet.py
- | └─ deploy_upgrade.py
- | └─ deploy_v2.py
- | └─ deploy_v2_1.py
- | └─ deploy_v2_2.py
- | └─ deploy_v2_add_bank.py
- | └─ deploy_v2_add_bank_2.py
- | └─ deploy_v2_oracles.py
- | └─ deploy_v2_oracles_setting.py
- | └─ deploy_v2_oracles_setting_2.py
- | └─ deploy_v2_oracles_setting_3.py
- | └─ deploy_v2_safeboxes.py
- | └─ deploy_v2_safeboxes_2.py
- | └─ fake_credit_limit.py
- | └─ ibeth_v2_router_test.py
- | └─ open_position.py
- | └─ query_ibs.py
- | └─ replace_v1_to_agg_oracle.py
- | └─ safebox_dai_2_test.py
- | └─ safebox_dai_test.py
- | └─ safebox_eth_2_test.py
- | └─ safebox_eth_test.py
- | └─ safebox_usdt_2_test.py
- | └─ safebox_usdt_test.py
- | └─ sushiswap_spell_add_remove_test.py
- | └─ sushiswap_spell_wmasterchef_add_remove_1x_test.py
- | └─ sushiswap_spell_wmasterchef_add_remove_test.py
- | └─ sushiswap_spell_wmasterchef_add_twice_test.py
- | └─ sushiswap_spell_wmasterchef_harvest_test.py



- | | — tokens.py
- | | — uniswap_lp_oracle_test.py
- | | — uniswap_spell_add_remove_1x_test.py
- | | — uniswap_spell_add_remove_test.py
- | | — uniswap_spell_eth_test.py
- | | — uniswap_spell_more_add_remove_test.py
- | | — uniswap_spell_usdc_usdt_test.py
- | | — uniswap_spell_wstaking_add_remove_1x_test.py
- | | — uniswap_spell_wstaking_add_remove_test.py
- | | — uniswap_spell_wstaking_add_twice_test.py
- | | — uniswap_spell_wstaking_harvest_test.py
- | | — utils.py
- | | — utils_fork.py
- | | — wliquidity_gauge_test.py
- | | — wmasterchef_test.py
- | | — wstaking_rewards_test.py
- | — shell.nix
- | — tests
 - | | — conftest.py
 - | | — helper_uniswap.py
 - | | — setup_basic.py
 - | | — setup_safebox.py
 - | | — setup_uniswap.py
 - | | — setup_user.py
 - | | — test_core_oracle.py
 - | | — test_homora_bank_1.py
 - | | — test_homora_bank_2.py
 - | | — test_liquidate.py
 - | | — test_safebox.py



```
| |— test_uniswap_add.py
| |— utils.py
|— yarn.lock
```

10 directories, 179 files