SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Asymetrix Protocol
Date: April 19, 2023

This report may contain confidential information about IT systems and the intellectual property of the Customer, as well as information about potential vulnerabilities and methods of their exploitation.

The report can be disclosed publicly after prior consent by another Party. Any subsequent publication of this report shall be without mandatory consent.

Document

Name	Smart Contract Code Review and Security Analysis Report for Asymetrix Protocol
Approved By	Noah Jelich | Lead Solidity SC Auditor at Hacken OU
Type	ERC20 token; Staking; Yielding
Platform	EVM
Language	Solidity
Methodology	Link
Website	https://asymetrix.io
Changelog	17.03.2023 - Initial Review 10.04.2023 - Second Review 14.04.2023 - Third Review 19.04.2023 - Fourth Review

Table of contents

Introduction

Hacken OÜ (Consultant) was contracted by Asymetrix Protocol (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contracts.

Scope

The scope of the project includes the following smart contracts from the provided repository:

Initial review scope

Repository	https://bitbucket.ideasoft.io/projects/PBON/repos/solidity/browse
Commit	a3fca450f638a0ad9aa2cd1627c1e10d27c71e01
Whitepaper	Link
Functional Requirements	https://asymetrix-protocol.gitbook.io/untitled/unDU3flVeFkvWUF7NgRW/asymetrix-protocol-guide/how-it-works
Technical Requirements	https://docs.google.com/document/d/1YxQVv7g1sWD88Up4eVJdyk1M0xKXdvLw5AaDSHI6uCo/edit#heading=h.bgfxjkm73x92
Contracts	File: ./contracts/ASX.sol SHA3: 469aa30d25a3480e9940b44f5a089024202b001d1de65ab24ecd0e9241342f56 File: ./contracts/core/ControlledToken.sol SHA3: 62cac81c8c84021de109d1df615cb3ab178ca86918bb0cd9b393f08e7b3aa316 File: ./contracts/core/DrawBeacon.sol SHA3: 64b421e0d206b0cbe14b4b0b3be413146a3d26e14dfc3dceeaac9daba67d0b65 File: ./contracts/core/DrawBuffer.sol SHA3: 48b38285fb9f230b46ec219656e383c4f96df65ab3d119c689ebafd3cf86dce4 File: ./contracts/core/DrawCalculator.sol SHA3: b324ee55aa66a382d4c65cb117669df02cf6041bd55737bd23b130e5103f79b8 File: ./contracts/core/external/compound/ICompLike.sol SHA3: b0312b8913395d10900842b207eb46c4f64a983b370069e0f2918f21dd4e340d File: ./contracts/core/interfaces/IControlledToken.sol SHA3: affdb914c24c548781aed05c57185b68ec721593d4d74d100b6d7ad52b4c481d File: ./contracts/core/interfaces/IDrawBeacon.sol SHA3: 4c83b3cd08233f1c8a031ca3d7e70f8a759762619cc14ca107a5a8dca353f0ff File: ./contracts/core/interfaces/IDrawBuffer.sol SHA3: 587142b85e6521cd44aaccec46869b7712dd744cb4e22ca799e8d8367f9ed71a File: ./contracts/core/interfaces/IDrawCalculator.sol SHA3: 25f9b6659f9640004a6fd11663fe3b84a414a5b00690d8c521c0b1e6d3abf862 File: ./contracts/core/interfaces/IPrizeDistributionBuffer.sol

	SHA3: 39ba2c60d2b9888e30e0a1bdab3ecd705970d1c92c90d1f1393a0f5390c8a1ab File: ./contracts/core/interfaces/IPrizeDistributionSource.sol SHA3: e9869e3e18f16f412561d24d7fd39081b0d923d468f437055b791706f93aaa39 File: ./contracts/core/interfaces/IPrizeDistributor.sol SHA3: 87ce01696ce86cf48623dd9686a4659175e5613e0c3fbf4f5f740583e6e96adb File: ./contracts/core/interfaces/IPrizePool.sol SHA3: a17a25e19ee7d3120445c802b2105275f9f5b7bbec147d3a4b7ce28a321110fb File: ./contracts/core/interfaces/IReserve.sol SHA3: 137f231a064bfca82bb86047d800a2f51c68e59d78b31fe8cb52ca14cb5383ce File: ./contracts/core/interfaces/ITicket.sol SHA3: 2654d0ac2c2eb0687f80f70e1cffdd7a99c314f6245dde8cdc947715d4e23d79 File: ./contracts/core/libraries/DrawRingBufferLib.sol SHA3: 98c26591feb674064d7e3018973f31be2f1257d39685ab6de2ec1d4dab78d8ae File: ./contracts/core/libraries/ExtendedSafeCastLib.sol SHA3: 606ea61492a0d69077ca1c10d8c395597728cf82d74db41c65b8f894ab58a511 File: ./contracts/core/libraries/ObservationLib.sol SHA3: 2bbdf5147147da06f62b79fe7fb4fb51f2d4e5505b9ef27fe8de1c04316b4c16 File: ./contracts/core/libraries/OverflowSafeComparatorLib.sol SHA3: 7254e240f1be5ade2efac32ca834fa8c6dce1707ae99c861ba12e4ca8720cd91 File: ./contracts/core/libraries/RingBufferLib.sol SHA3: 3a75a2c0ad0a0d9f504249a4330b12dea13562f50211fe092d7e984bc6191a8d File: ./contracts/core/libraries/TwabLib.sol SHA3: 30aa2fe0a87af75e4920bcfe133ea82de6e924ac8ef29fd18bb63878b41ec318 File: ./contracts/core/permit/EIP2612PermitAndDeposit.sol SHA3: 2af005bb68597e86d3246c9f465d2df7307857bdb7ac013bf19f6e6c47d683e7 File: ./contracts/core/prize-pool/PrizePool.sol SHA3: 97635a67ae7404d44ce96b7dbd98cecf9dec91b3cd5f4f5360d168fdfbc33d7c File: ./contracts/core/prize-pool/StakePrizePool.sol SHA3: 962586eb9bb16e8daa145d623a5ee3de38969af3a268f657e7729b3c7bca608b File: ./contracts/core/PrizeDistributionBuffer.sol SHA3: f56c5c367c182616ec728b69a44d5923d414f9db041af80a90cbd47bf31bcac5 File: ./contracts/core/PrizeDistributor.sol SHA3: be619bff745dc79588bf8913ea5473e9308c09dc810449ef0980ef4af97c46da File: ./contracts/core/Reserve.sol SHA3: ed50fcd3d6e31a38eb0c84153c509ba7e49050880386989a0c913f985abb5e07 File: ./contracts/core/Ticket.sol SHA3: 8bf25f9712d6c73f629d577dc74cd3cad22fe8e318b1e68b7acb4045224ccd5d File: ./contracts/owner-manager/Manageable.sol SHA3: c945cd18351e30b7ac875779c7fe9bde4ee84b8b774a934badbbbe07813d2996 File: ./contracts/owner-manager/Ownable.sol SHA3: f4900d2614be6f2b7e906d528a8e10d3a891cc0916deb4e6b888446808ccad7e File: ./contracts/periphery/interfaces/IPrizeFlush.sol

	SHA3: b8ccafac0582451986f265ffc0944ad976429dfab36afa01b5ec34a35a878c4e File: ./contracts/periphery/interfaces/ITwabRewards.sol SHA3: 9cc41ef30198eaeab37a3f3a0bff02d2344ca1b56ac95192778763afbea779a9 File: ./contracts/periphery/libraries/BinarySearchLib.sol SHA3: 0901cc159fcde8dc376edd1086fc4f1842f5cf87382ac40aa96dd2ed8f4a1629 File: ./contracts/periphery/PrizeDistributionFactory.sol SHA3: 0794713a9f3fa4300d9e939a63a301d37e91863a0be658f0a5cbb4d682cb8f0e File: ./contracts/periphery/PrizeFlush.sol SHA3: ba0240ec85f9e8987e6141973dc4d1ea078e69296ba3ab06b5b8b15102311270 File: ./contracts/periphery/TwabRewards.sol SHA3: a888f3e99816fe9e4d8cf455f4a5499924b524f85fed3c6c33984cce38cda4c9 File: ./contracts/timelocks/BeaconTimelockTrigger.sol SHA3: d60d993c0a7121112ad3eaa1ae72b7ccae7a7c52c23d6d62ad0050f56be68d8e File: ./contracts/timelocks/DrawCalculatorTimelock.sol SHA3: 0919cad2a366f6810d0ee006fca0338256d2a200d63d9291103a1381419babaa File: ./contracts/timelocks/interfaces/IBeaconTimelockTrigger.sol SHA3: 1804910f23a2d60c26aeb21b1eb9fdf3ed240f83c8e998e6f0e85c35d30a3600 File: ./contracts/timelocks/interfaces/IDrawCalculatorTimelock.sol SHA3: c2f75575e9f547cfbf6c84d403641aa41474e10d7da357ce55a0955c413211f1 File: ./contracts/timelocks/interfaces/IPrizeDistributionFactory.sol SHA3: 39b59c52f808d8560ef5986e92e70f528d77e55bb5eedd0f67d425b8bd5062c7 File: ./contracts/twab-delegator/Delegation.sol SHA3: 28abd3fd672b0d073252dfdda39bb8149d97dc16fdea3e8a583bd82fa0addfba File: ./contracts/twab-delegator/LowLevelDelegator.sol SHA3: 8beeaac0ca8c463e54292961576d2507dfd629719d16557530b2375214c1d876 File: ./contracts/twab-delegator/PermitAndMulticall.sol SHA3: a49700058c9c861a1c80becbcdeed11899a93de94d87494ee6a0a6eb62e2df6e File: ./contracts/twab-delegator/TWABDelegator.sol SHA3: 76e4083efbe9ba717912181faca0451b6415a1ae0e8be9a8b0df48366566d2f0

Second review scope

Repository	https://bitbucket.ideasoft.io/projects/PBON/repos/solidity/browse
Commit	8068c4bf7a0c54bf191eb5b0ea8eecf5c4d8f911
Functional Requirements	https://docs.asymetrix.io/what-is-asymetrix-protocol/overview
Technical Requirements	https://docs.asymetrix.io/what-is-asymetrix-protocol/overview
Contracts	File: contracts/ASX.sol SHA3: 469aa30d25a3480e9940b44f5a089024202b001d1de65ab24ecd0e9241342f56

	File: contracts/Constants.sol SHA3: 7cd8d36e86065d0bacbe425106afefe5ea188184f41fa3584fe5775f93a16cd0 File: contracts/core/ControlledToken.sol SHA3: c3c0be29760eee1349378f15458e92b1404739c06cfa2b60f84986e1ae69650a File: contracts/core/DrawBeacon.sol SHA3: 9e103025a9904ff283d8643c46c6ec6627379063a7ffdd0b42810f8c260b0ae6 File: contracts/core/DrawBuffer.sol SHA3: 7d94ef53665fc3bb27a5281f2e3c3cd0c0a4f1e83db4a6a5c01be9d0df485ba5 File: contracts/core/DrawCalculator.sol SHA3: 63ffcfc1aad12b7318dbe23a48035aa943cb00fe5bc20a54e3c5427c20b9fced File: contracts/core/PrizeDistributionBuffer.sol SHA3: a1fe01d0a1468afedf141f56ce42ec0c161e18771c06e5f2de16578131445adb File: contracts/core/PrizeDistributor.sol SHA3: 01fb0e369eca2372f28d2fcc8252441baef8448f93f0fe06e416c63b838e2b54 File: contracts/core/Reserve.sol SHA3: e30acf57b0131e37ae606b7b386a5c3b2499f7c6cf903c6ec3dcec4834924c94 File: contracts/core/Ticket.sol SHA3: 94a061ddd072f112bcd81fd8b9e42a5a95ac8767a6779d21e816b1b0184b36fa File: contracts/core/external/compound/ICompLike.sol SHA3: b0312b8913395d10900842b207eb46c4f64a983b370069e0f2918f21dd4e340d File: contracts/core/interfaces/IControlledToken.sol SHA3: affdb914c24c548781aed05c57185b68ec721593d4d74d100b6d7ad52b4c481d File: contracts/core/interfaces/IDrawBeacon.sol SHA3: 3be016b888fd51d0d13746a002ff6752257b8224080c9078a1c0ee6562bde6ee File: contracts/core/interfaces/IDrawBuffer.sol SHA3: 587142b85e6521cd44aaccec46869b7712dd744cb4e22ca799e8d8367f9ed71a File: contracts/core/interfaces/IDrawCalculator.sol SHA3: 25f9b6659f9640004a6fd11663fe3b84a414a5b00690d8c521c0b1e6d3abf862 File: contracts/core/interfaces/IPrizeDistributionBuffer.sol SHA3: 39ba2c60d2b9888e30e0a1bdab3ecd705970d1c92c90d1f1393a0f5390c8a1ab File: contracts/core/interfaces/IPrizeDistributionSource.sol SHA3: 125f1f219be344d895695975258d6b2a918efdea66e5814d08849eb2cd4e2b31 File: contracts/core/interfaces/IPrizeDistributor.sol SHA3: eea3a6c282b215f93b431ef050389060bc175d41ceba28d523441b6ad0bef476 File: contracts/core/interfaces/IPrizePool.sol SHA3: ccdc2a8540c06b09c2536a7252b8e0d5c30ee224dfdd2ff663789c1c046bf097 File: contracts/core/interfaces/IReserve.sol SHA3: 137f231a064bfca82bb86047d800a2f51c68e59d78b31fe8cb52ca14cb5383ce File: contracts/core/interfaces/ITicket.sol SHA3: 2654d0ac2c2eb0687f80f70e1cffdd7a99c314f6245dde8cdc947715d4e23d79 File: contracts/core/libraries/DrawRingBufferLib.sol SHA3: 98c26591feb674064d7e3018973f31be2f1257d39685ab6de2ec1d4dab78d8ae

	File: contracts/core/libraries/ExtendedSafeCastLib.sol SHA3: 606ea61492a0d69077ca1c10d8c395597728cf82d74db41c65b8f894ab58a511 File: contracts/core/libraries/ObservationLib.sol SHA3: 2bbdf5147147da06f62b79fe7fb4fb51f2d4e5505b9ef27fe8de1c04316b4c16 File: contracts/core/libraries/OverflowSafeComparatorLib.sol SHA3: 7254e240f1be5ade2efac32ca834fa8c6dce1707ae99c861ba12e4ca8720cd91 File: contracts/core/libraries/RingBufferLib.sol SHA3: 3a75a2c0ad0a0d9f504249a4330b12dea13562f50211fe092d7e984bc6191a8d File: contracts/core/libraries/TwabLib.sol SHA3: 30aa2fe0a87af75e4920bcfe133ea82de6e924ac8ef29fd18bb63878b41ec318 File: contracts/core/permit/EIP2612PermitAndDeposit.sol SHA3: 11372ceda0628178412f295b2c32fd8c96af59d51169a137ac77c8eb75a21963 File: contracts/core/prize-pool/PrizePool.sol SHA3: b0f455aa91f4559a2381eb9b8da80a2cd9305b24faf8d270eb75aec71918c414 File: contracts/core/prize-pool/StakePrizePool.sol SHA3: 30335c47f67156704752e6f9eff1c4d27cc2e7e4eef233b8b9c3fcfcb634cf4a File: contracts/owner-manager/Manageable.sol SHA3: 442c444a082b77f501a69cb25445525fa090803bffb2940e9cfa80d9d85b8c7a File: contracts/owner-manager/Ownable.sol SHA3: 8b60473d42610f5f104e6d71305bd92ff4bd09f732f875accc51a332dc166491 File: contracts/periphery/PrizeDistributionFactory.sol SHA3: 5bc9b20f786dd75469e057372d159e21b6b210bec73a11d558078ea1beb26485 File: contracts/periphery/PrizeFlush.sol SHA3: 07e14d378bfdd4407b1361331c7c0869990e2a6aca7e6990023acd2e38bbc444 File: contracts/periphery/TwabRewards.sol SHA3: bc24a6b8cf5afacfbbc1a943431c6fd1c181ef04557c674f3b4f19634f616524 File: contracts/periphery/interfaces/IPrizeFlush.sol SHA3: fcbe0795237e6587906214941295e629fc8c6280c1c896e9a957a7d4b1fec56c File: contracts/periphery/interfaces/ITwabRewards.sol SHA3: 9cc41ef30198eaeab37a3f3a0bff02d2344ca1b56ac95192778763afbea779a9 File: contracts/periphery/libraries/BinarySearchLib.sol SHA3: 6c6ad9969b3b801ce65122a232e3aae16f28a2ccdb788be1012658545857bc47 File: contracts/rng-service/RNGServiceChainlinkV2.sol SHA3: 86a5a56daae5bdb387e4cbaf00f359bd66341e1ef3afed5a2c62f3a787d0d9e4 File: contracts/rng-service/chainlink/VRFConsumerBaseV2.sol SHA3: f8bf2814e10e47afa6a4605921c6dfc35850b91d158d03062b99a42bfb6466a3 File: contracts/rng-service/interfaces/IRNGService.sol SHA3: e01bfbac7ce3700d3e68dea662485d3e93dc679180e0a07d707fcefbbb10318c File: contracts/rng-service/interfaces/IRNGServiceChainlinkV2.sol SHA3: 18f4e7e4969a751a14a13ad8c3ffa78fc17c69f87a8a1192f72b20b9c274b6fa File: contracts/timelocks/BeaconTimelockTrigger.sol SHA3: 81fdce08b826dff69c6524e56f240e885699a15cd560e8e1719c3f1c200de4f0

	File: contracts/timelocks/DrawCalculatorTimelock.sol SHA3: 8e829b5f3ccf57f5df2894df619ade8554d45ab2d0508911e4bcbf6884e2cf6d File: contracts/timelocks/interfaces/IBeaconTimelockTrigger.sol SHA3: d931006ac0085832211c4bb4706bc59d6c8c69c98723ebf23965dc58139e421d File: contracts/timelocks/interfaces/IDrawCalculatorTimelock.sol SHA3: b0763ae4f93653fe956e4f176d14b3385245ab95ce52c16a9af5e7fb886dbe66 File: contracts/timelocks/interfaces/IPrizeDistributionFactory.sol SHA3: 2bd9f1c3d2fbf7a1e5d8d848d79f2c6817b70a347680d79e02da17f68a4eb44e File: contracts/twab-delegator/Delegation.sol SHA3: 9445ed24b334af9cd76d2b2309f316071c064265470a65c3f84c5503ad67b9e6 File: contracts/twab-delegator/LowLevelDelegator.sol SHA3: 83c1914e4a8c594760eb03cfd31c47eb06f6303fde092a3421615988f6413b1f File: contracts/twab-delegator/PermitAndMulticall.sol SHA3: 1d27d1b4e4c8dd02d0ab07030c6425937bb5146bea7d425738c6f063f1ed98b8 File: contracts/twab-delegator/TWABDelegator.sol SHA3: 55ab2220cff8ea31b6156ee568bc35674eed65744b1656ea605c1c046731f956

Third review scope

Repository	https://bitbucket.ideasoft.io/projects/PBON/repos/solidity/browse
Commit	dc7979fcc57daf2edaed4192ce9faf6c1e813326
Contracts	File: contracts/ASX.sol SHA3: 469aa30d25a3480e9940b44f5a089024202b001d1de65ab24ecd0e9241342f56 File: contracts/Constants.sol SHA3: 7cd8d36e86065d0bacbe425106afefe5ea188184f41fa3584fe5775f93a16cd0 File: contracts/core/ControlledToken.sol SHA3: c3c0be29760eee1349378f15458e92b1404739c06cfa2b60f84986e1ae69650a File: contracts/core/DrawBeacon.sol SHA3: 5c8154f055ae7cbd5ece9b4b7bbe88a0439a6d292e207ae06213a72594c0f36f File: contracts/core/DrawBuffer.sol SHA3: 7d94ef53665fc3bb27a5281f2e3c3cd0c0a4f1e83db4a6a5c01be9d0df485ba5 File: contracts/core/DrawCalculator.sol SHA3: 63ffcfc1aad12b7318dbe23a48035aa943cb00fe5bc20a54e3c5427c20b9fced File: contracts/core/PrizeDistributionBuffer.sol SHA3: a1fe01d0a1468afedf141f56ce42ec0c161e18771c06e5f2de16578131445adb File: contracts/core/PrizeDistributor.sol SHA3: 01fb0e369eca2372f28d2fcc8252441baef8448f93f0fe06e416c63b838e2b54 File: contracts/core/Reserve.sol SHA3: e30acf57b0131e37ae606b7b386a5c3b2499f7c6cf903c6ec3dcec4834924c94 File: contracts/core/Ticket.sol SHA3: 94a061ddd072f112bcd81fd8b9e42a5a95ac8767a6779d21e816b1b0184b36fa

	File: contracts/core/external/compound/ICompLike.sol SHA3: b0312b8913395d10900842b207eb46c4f64a983b370069e0f2918f21dd4e340d File: contracts/core/interfaces/IControlledToken.sol SHA3: affdb914c24c548781aed05c57185b68ec721593d4d74d100b6d7ad52b4c481d File: contracts/core/interfaces/IDrawBeacon.sol SHA3: 6575375e0935311c58ad46dea1cac8594269524358cb7b73e14be91353a406f8 File: contracts/core/interfaces/IDrawBuffer.sol SHA3: 587142b85e6521cd44aaccec46869b7712dd744cb4e22ca799e8d8367f9ed71a File: contracts/core/interfaces/IDrawCalculator.sol SHA3: 25f9b6659f9640004a6fd11663fe3b84a414a5b00690d8c521c0b1e6d3abf862 File: contracts/core/interfaces/IPrizeDistributionBuffer.sol SHA3: 39ba2c60d2b9888e30e0a1bdab3ecd705970d1c92c90d1f1393a0f5390c8a1ab File: contracts/core/interfaces/IPrizeDistributionSource.sol SHA3: 125f1f219be344d895695975258d6b2a918efdea66e5814d08849eb2cd4e2b31 File: contracts/core/interfaces/IPrizeDistributor.sol SHA3: eea3a6c282b215f93b431ef050389060bc175d41ceba28d523441b6ad0bef476 File: contracts/core/interfaces/IPrizePool.sol SHA3: ccdc2a8540c06b09c2536a7252b8e0d5c30ee224dfdd2ff663789c1c046bf097 File: contracts/core/interfaces/IReserve.sol SHA3: 137f231a064bfca82bb86047d800a2f51c68e59d78b31fe8cb52ca14cb5383ce File: contracts/core/interfaces/ITicket.sol SHA3: 2654d0ac2c2eb0687f80f70e1cffdd7a99c314f6245dde8cdc947715d4e23d79 File: contracts/core/libraries/DrawRingBufferLib.sol SHA3: 98c26591feb674064d7e3018973f31be2f1257d39685ab6de2ec1d4dab78d8ae File: contracts/core/libraries/ExtendedSafeCastLib.sol SHA3: 606ea61492a0d69077ca1c10d8c395597728cf82d74db41c65b8f894ab58a511 File: contracts/core/libraries/ObservationLib.sol SHA3: 2bbdf5147147da06f62b79fe7fb4fb51f2d4e5505b9ef27fe8de1c04316b4c16 File: contracts/core/libraries/OverflowSafeComparatorLib.sol SHA3: 7254e240f1be5ade2efac32ca834fa8c6dce1707ae99c861ba12e4ca8720cd91 File: contracts/core/libraries/RingBufferLib.sol SHA3: 3a75a2c0ad0a0d9f504249a4330b12dea13562f50211fe092d7e984bc6191a8d File: contracts/core/libraries/TwabLib.sol SHA3: 30aa2fe0a87af75e4920bcfe133ea82de6e924ac8ef29fd18bb63878b41ec318 File: contracts/core/permit/EIP2612PermitAndDeposit.sol SHA3: 11372ceda0628178412f295b2c32fd8c96af59d51169a137ac77c8eb75a21963 File: contracts/core/prize-pool/PrizePool.sol SHA3: b0f455aa91f4559a2381eb9b8da80a2cd9305b24faf8d270eb75aec71918c414 File: contracts/core/prize-pool/StakePrizePool.sol SHA3: 30335c47f67156704752e6f9eff1c4d27cc2e7e4eef233b8b9c3fcfcb634cf4a File: contracts/owner-manager/Manageable.sol SHA3: 442c444a082b77f501a69cb25445525fa090803bffb2940e9cfa80d9d85b8c7a

	File: contracts/owner-manager/Ownable.sol SHA3: 8b60473d42610f5f104e6d71305bd92ff4bd09f732f875accc51a332dc166491 File: contracts/periphery/PrizeDistributionFactory.sol SHA3: 5bc9b20f786dd75469e057372d159e21b6b210bec73a11d558078ea1beb26485 File: contracts/periphery/PrizeFlush.sol SHA3: 07e14d378bfdd4407b1361331c7c0869990e2a6aca7e6990023acd2e38bbc444 File: contracts/periphery/TwabRewards.sol SHA3: bc24a6b8cf5afacfbbc1a943431c6fd1c181ef04557c674f3b4f19634f616524 File: contracts/periphery/interfaces/IPrizeFlush.sol SHA3: fcbe0795237e6587906214941295e629fc8c6280c1c896e9a957a7d4b1fec56c File: contracts/periphery/interfaces/ITwabRewards.sol SHA3: 9cc41ef30198eaeab37a3f3a0bff02d2344ca1b56ac95192778763afbea779a9 File: contracts/periphery/libraries/BinarySearchLib.sol SHA3: 6c6ad9969b3b801ce65122a232e3aae16f28a2ccdb788be1012658545857bc47 File: contracts/rng-service/RNGServiceChainlinkV2.sol SHA3: 86a5a56daae5bdb387e4cbaf00f359bd66341e1ef3afed5a2c62f3a787d0d9e4 File: contracts/rng-service/chainlink/VRFConsumerBaseV2.sol SHA3: f8bf2814e10e47afa6a4605921c6dfc35850b91d158d03062b99a42bfb6466a3 File: contracts/rng-service/interfaces/IRNGService.sol SHA3: e01bfbac7ce3700d3e68dea662485d3e93dc679180e0a07d707fcefbbb10318c File: contracts/rng-service/interfaces/IRNGServiceChainlinkV2.sol SHA3: 18f4e7e4969a751a14a13ad8c3ffa78fc17c69f87a8a1192f72b20b9c274b6fa File: contracts/timelocks/BeaconTimelockTrigger.sol SHA3: 81fdce08b826dff69c6524e56f240e885699a15cd560e8e1719c3f1c200de4f0 File: contracts/timelocks/DrawCalculatorTimelock.sol SHA3: 8e829b5f3ccf57f5df2894df619ade8554d45ab2d0508911e4bcbf6884e2cf6d File: contracts/timelocks/interfaces/IBeaconTimelockTrigger.sol SHA3: d931006ac0085832211c4bb4706bc59d6c8c69c98723ebf23965dc58139e421d File: contracts/timelocks/interfaces/IDrawCalculatorTimelock.sol SHA3: b0763ae4f93653fe956e4f176d14b3385245ab95ce52c16a9af5e7fb886dbe66 File: contracts/timelocks/interfaces/IPrizeDistributionFactory.sol SHA3: 2bd9f1c3d2fbf7a1e5d8d848d79f2c6817b70a347680d79e02da17f68a4eb44e File: contracts/twab-delegator/Delegation.sol SHA3: 9445ed24b334af9cd76d2b2309f316071c064265470a65c3f84c5503ad67b9e6 File: contracts/twab-delegator/LowLevelDelegator.sol SHA3: 83c1914e4a8c594760eb03cfd31c47eb06f6303fde092a3421615988f6413b1f File: contracts/twab-delegator/PermitAndMulticall.sol SHA3: 1d27d1b4e4c8dd02d0ab07030c6425937bb5146bea7d425738c6f063f1ed98b8 File: contracts/twab-delegator/TWABDelegator.sol SHA3: 55ab2220cff8ea31b6156ee568bc35674eed65744b1656ea605c1c046731f956

Fourth review scope

Repository	https://bitbucket.ideasoft.io/projects/PBON/repos/solidity/browse
Commit	5dcc0ba3c17ab271ad624cbf5a9a8d16c737b448
Contracts	File: contracts/ASX.sol SHA3: 469aa30d25a3480e9940b44f5a089024202b001d1de65ab24ecd0e9241342f56 File: contracts/Constants.sol SHA3: 7cd8d36e86065d0bacbe425106afefe5ea188184f41fa3584fe5775f93a16cd0 File: contracts/core/ControlledToken.sol SHA3: c3c0be29760eee1349378f15458e92b1404739c06cfa2b60f84986e1ae69650a File: contracts/core/DrawBeacon.sol SHA3: 50ab028b71f5602cacd456759c2f974c94e05a11076241c4247c2f7d0ebc7916 File: contracts/core/DrawBuffer.sol SHA3: 7d94ef53665fc3bb27a5281f2e3c3cd0c0a4f1e83db4a6a5c01be9d0df485ba5 File: contracts/core/DrawCalculator.sol SHA3: 63ffcfc1aad12b7318dbe23a48035aa943cb00fe5bc20a54e3c5427c20b9fced File: contracts/core/PrizeDistributionBuffer.sol SHA3: a1fe01d0a1468afedf141f56ce42ec0c161e18771c06e5f2de16578131445adb File: contracts/core/PrizeDistributor.sol SHA3: 01fb0e369eca2372f28d2fcc8252441baef8448f93f0fe06e416c63b838e2b54 File: contracts/core/Reserve.sol SHA3: e30acf57b0131e37ae606b7b386a5c3b2499f7c6cf903c6ec3dcec4834924c94 File: contracts/core/Ticket.sol SHA3: 94a061ddd072f112bcd81fd8b9e42a5a95ac8767a6779d21e816b1b0184b36fa File: contracts/core/external/compound/ICompLike.sol SHA3: b0312b8913395d10900842b207eb46c4f64a983b370069e0f2918f21dd4e340d File: contracts/core/interfaces/IControlledToken.sol SHA3: affdb914c24c548781aed05c57185b68ec721593d4d74d100b6d7ad52b4c481d File: contracts/core/interfaces/IDrawBeacon.sol SHA3: 708372fc8520214500c291f261f802342e7d38e448ff939a017615f3f0f9a925 File: contracts/core/interfaces/IDrawBuffer.sol SHA3: 587142b85e6521cd44aaccec46869b7712dd744cb4e22ca799e8d8367f9ed71a File: contracts/core/interfaces/IDrawCalculator.sol SHA3: 25f9b6659f9640004a6fd11663fe3b84a414a5b00690d8c521c0b1e6d3abf862 File: contracts/core/interfaces/IPrizeDistributionBuffer.sol SHA3: 39ba2c60d2b9888e30e0a1bdab3ecd705970d1c92c90d1f1393a0f5390c8a1ab File: contracts/core/interfaces/IPrizeDistributionSource.sol SHA3: 125f1f219be344d895695975258d6b2a918efdea66e5814d08849eb2cd4e2b31 File: contracts/core/interfaces/IPrizeDistributor.sol SHA3: eea3a6c282b215f93b431ef050389060bc175d41ceba28d523441b6ad0bef476 File: contracts/core/interfaces/IPrizePool.sol SHA3: ccdc2a8540c06b09c2536a7252b8e0d5c30ee224dfdd2ff663789c1c046bf097

	File: contracts/core/interfaces/IReserve.sol SHA3: 137f231a064bfca82bb86047d800a2f51c68e59d78b31fe8cb52ca14cb5383ce File: contracts/core/interfaces/ITicket.sol SHA3: 2654d0ac2c2eb0687f80f70e1cffdd7a99c314f6245dde8cdc947715d4e23d79 File: contracts/core/libraries/DrawRingBufferLib.sol SHA3: 98c26591feb674064d7e3018973f31be2f1257d39685ab6de2ec1d4dab78d8ae File: contracts/core/libraries/ExtendedSafeCastLib.sol SHA3: 606ea61492a0d69077ca1c10d8c395597728cf82d74db41c65b8f894ab58a511 File: contracts/core/libraries/ObservationLib.sol SHA3: 2bbdf5147147da06f62b79fe7fb4fb51f2d4e5505b9ef27fe8de1c04316b4c16 File: contracts/core/libraries/OverflowSafeComparatorLib.sol SHA3: 7254e240f1be5ade2efac32ca834fa8c6dce1707ae99c861ba12e4ca8720cd91 File: contracts/core/libraries/RingBufferLib.sol SHA3: 3a75a2c0ad0a0d9f504249a4330b12dea13562f50211fe092d7e984bc6191a8d File: contracts/core/libraries/TwabLib.sol SHA3: 30aa2fe0a87af75e4920bcfe133ea82de6e924ac8ef29fd18bb63878b41ec318 File: contracts/core/permit/EIP2612PermitAndDeposit.sol SHA3: 11372ceda0628178412f295b2c32fd8c96af59d51169a137ac77c8eb75a21963 File: contracts/core/prize-pool/PrizePool.sol SHA3: b0f455aa91f4559a2381eb9b8da80a2cd9305b24faf8d270eb75aec71918c414 File: contracts/core/prize-pool/StakePrizePool.sol SHA3: 30335c47f67156704752e6f9eff1c4d27cc2e7e4eef233b8b9c3fcfcb634cf4a File: contracts/owner-manager/Manageable.sol SHA3: 442c444a082b77f501a69cb25445525fa090803bffb2940e9cfa80d9d85b8c7a File: contracts/owner-manager/Ownable.sol SHA3: 8b60473d42610f5f104e6d71305bd92ff4bd09f732f875accc51a332dc166491 File: contracts/periphery/PrizeDistributionFactory.sol SHA3: 5bc9b20f786dd75469e057372d159e21b6b210bec73a11d558078ea1beb26485 File: contracts/periphery/PrizeFlush.sol SHA3: 07e14d378bfdd4407b1361331c7c0869990e2a6aca7e6990023acd2e38bbc444 File: contracts/periphery/TwabRewards.sol SHA3: bc24a6b8cf5afacfbbc1a943431c6fd1c181ef04557c674f3b4f19634f616524 File: contracts/periphery/interfaces/IPrizeFlush.sol SHA3: fcbe0795237e6587906214941295e629fc8c6280c1c896e9a957a7d4b1fec56c File: contracts/periphery/interfaces/ITwabRewards.sol SHA3: 9cc41ef30198eaeab37a3f3a0bff02d2344ca1b56ac95192778763afbea779a9 File: contracts/periphery/libraries/BinarySearchLib.sol SHA3: 6c6ad9969b3b801ce65122a232e3aae16f28a2ccdb788be1012658545857bc47 File: contracts/rng-service/RNGServiceChainlinkV2.sol SHA3: 86a5a56daae5bdb387e4cbaf00f359bd66341e1ef3afed5a2c62f3a787d0d9e4 File: contracts/rng-service/chainlink/VRFConsumerBaseV2.sol SHA3: f8bf2814e10e47afa6a4605921c6dfc35850b91d158d03062b99a42bfb6466a3

	File: contracts/rng-service/interfaces/IRNGService.sol SHA3: e01bfbac7ce3700d3e68dea662485d3e93dc679180e0a07d707fcefbbb10318c File: contracts/rng-service/interfaces/IRNGServiceChainlinkV2.sol SHA3: 18f4e7e4969a751a14a13ad8c3ffa78fc17c69f87a8a1192f72b20b9c274b6fa File: contracts/timelocks/BeaconTimelockTrigger.sol SHA3: 81fdce08b826dff69c6524e56f240e885699a15cd560e8e1719c3f1c200de4f0 File: contracts/timelocks/DrawCalculatorTimelock.sol SHA3: 8e829b5f3ccf57f5df2894df619ade8554d45ab2d0508911e4bcbf6884e2cf6d File: contracts/timelocks/interfaces/IBeaconTimelockTrigger.sol SHA3: d931006ac0085832211c4bb4706bc59d6c8c69c98723ebf23965dc58139e421d File: contracts/timelocks/interfaces/IDrawCalculatorTimelock.sol SHA3: b0763ae4f93653fe956e4f176d14b3385245ab95ce52c16a9af5e7fb886dbe66 File: contracts/timelocks/interfaces/IPrizeDistributionFactory.sol SHA3: 2bd9f1c3d2fbf7a1e5d8d848d79f2c6817b70a347680d79e02da17f68a4eb44e File: contracts/twab-delegator/Delegation.sol SHA3: 9445ed24b334af9cd76d2b2309f316071c064265470a65c3f84c5503ad67b9e6 File: contracts/twab-delegator/LowLevelDelegator.sol SHA3: 83c1914e4a8c594760eb03cfd31c47eb06f6303fde092a3421615988f6413b1f File: contracts/twab-delegator/PermitAndMulticall.sol SHA3: 1d27d1b4e4c8dd02d0ab07030c6425937bb5146bea7d425738c6f063f1ed98b8 File: contracts/twab-delegator/TWABDelegator.sol SHA3: 55ab2220cff8ea31b6156ee568bc35674eed65744b1656ea605c1c046731f956

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation by external or internal actors.
High	High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation by external or internal actors.
Medium	Medium vulnerabilities are usually limited to state manipulations but cannot lead to asset loss. Major deviations from best practices are also in this category.
Low	Low vulnerabilities are related to outdated and unused code or minor Gas optimization. These issues won't have a significant impact on code execution but affect code quality

Executive Summary

The score measurement details can be found in the corresponding section of the scoring methodology.

Documentation quality

The total Documentation Quality score is **10** out of **10**.

- Functional requirements are provided.
- Technical description is provided.

Code quality

The total Code Quality score is **9** out of **10**.

- The code is heavily based on the following project: pool together
- The project follows style guidelines and best practices
- The development environment is configured.

Test coverage

Code coverage of the project is **99.17%** (branch coverage).

- Deployment and basic user interactions are covered with tests.
- PrizeDistributor.sol throws exceptions in draws and randomness.
- Two tests throw an “TypeError: ethers.getContract is not a function” exception.

Security score

As a result of the audit, the code contains **no** issues. The security score is **10** out of **10**.

All found issues are displayed in the “Findings” section.

Summary

According to the assessment, the Customer's smart contract has the following score: **9.7**.

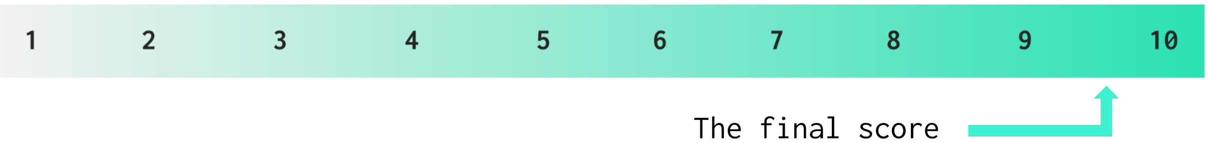

Table. The distribution of issues during the audit

Review date	Low	Medium	High	Critical
13 March 2023	2	13	1	1
10 April 2023	0	2	1	1
14 April 2023	0	0	0	0
19 April 2023	0	0	0	0

Risks

- The project uses different types of uint variables, and performs conversions between them, this may cause reverts due to their upper bounds.
- The **prize distribution process depends on an out of scope off-chain system**. The **admin is allowed to pass any index of their choosing** to IPFS. Thus Hacken cannot fully guarantee the security for the randomness in the prize distribution process. However, **the validity of the prize distribution can be validated at each round**, and the intermediary calculation step is stored on IPFS. (The win chances for each address are calculated based on the contract state, but the **costly calculation** is **done off-chain**, and its **results** are **pinned to IPFS for external verification**). Random numbers are generated using **on-chain VRF** and the **admin supplies** the **random addresses** to the contract **based on those**. This **process can be tracked via the protocol website**.

System Overview

Asymetrix protocol is the decentralized, non-custodial protocol for asymmetric yield distribution generated from staking. The files in the scope:

- **PrizePool.sol** - Escrows assets and deposits them into a yield source. Exposes interest to Prize Flush. Users deposit and withdraw from this contract to participate in the Prize Pool.
- **Ticket.sol** - The Ticket extends the standard ERC20 and ControlledToken interfaces with time-weighted average balance functionality. The average balance held by a user between two timestamps can be calculated, as well as the historic balance. The historic total supply is available, as well as the average total supply between two timestamps.
- **TwabRewards.sol** - Reward distributing contract. Distributes rewards to depositors in a pool.
- **TWABDelegator.sol** - Delegator contract allows accounts to easily delegate a portion of their tickets to multiple delegates.
- **TwabLib.sol** - Adds on-chain historical lookups to a user(s) time-weighted average balance.
- **PrizeDistributor.sol** - Holds Tickets (captured interest) and distributes tickets to users with winning draw claims.
- **DrawBeacon.sol** - Manages pushing Draws onto DrawBuffer.
- **Reserve.sol** - Provides historical lookups of a token balance increase during a target time range.
- **PrizeDistributionFactory.sol** - Populates a Prize Distribution Buffer for a prize pool.
- **PrizeFlush.sol** - The PrizeFlush contract helps capture interest from the PrizePool and move collected funds to a designated PrizeDistributor contract.
- **DrawBuffer.sol** - Provides historical lookups of Draws via a circular ring buffer. Historical Draws can be accessed on-chain using a drawId to calculate ring buffer storage slot.
- **DrawCalculator.sol** - Calculates the amount of user picks based on the user average weighted balance (during each draw period)
- **EIP2612PermitAndDeposit.sol** - Allows users to approve and deposit EIP-2612 compatible tokens into a prize pool in a single transaction.
- **ControlledToken.sol** - ERC20 contract with a controller for minting & burning.
- **ASX.sol** - ERC20 contract with voting and burning feature.
- **PermitAndMulticall.sol** - Allows a user to permit token spend and then call multiple functions on a contract.
- **DrawRingBufferLib.sol** - Library for creating and managing a draw ring buffer.

- **ObservationLib.sol** - Library allows one to store an array of timestamped values and efficiently binary search them.
- **BeaconTimelockTrigger.sol** - Passes the information about the current draw to the prizeDistributionFactory for the creation of a prizeDistribution.
- **DrawCalculatorTimelock.sol** - Pushes Draws to a DrawBuffer and routing claim requests from a PrizeDistributor to a DrawCalculator.
- **Ownable.sol** - Provides a basic access control mechanism, where there is an account (an owner) that can be granted exclusive access to specific functions.
- **StakePrizePool.sol** - The Stake Prize Pool is a prize pool in which users can deposit an ERC20 token.
- **Delegation.sol** - A Delegation allows his owner to execute calls on behalf of the contract.
- **Manageable.sol** - Abstract manageable contract which provides a basic access control mechanism.
- **BinarySearchLib.sol** - BinarySearchLib uses binary search to find a parent contract struct with the drawId parameter.
- **LowLevelDelegator.sol** - Allows users to create delegations very cheaply.
- **OverflowSafeComparatorLib.sol** - OverflowSafeComparatorLib library to share comparator functions between contracts.
- **RingBufferLib.sol** - Library for managing the TWAB index.
- **ExtendedSafeCastLib.sol** - Downcasting from uint256/uint224, uint256/uint208, uint256/uint104.
- **ICompLike.sol** - An interface contract for CompLike token.
- **IPrizePool.sol** - An interface contract for PrizePool.sol
- **ITicket.sol -** An interface contract for Ticket.sol
- **IPrizeDistributor.sol** - An interface contract for PrizeDistributor.sol
- **IPrizeDistributionBuffer.sol** - An interface contract for PrizeDistributionBuffer.sol
- **IDrawBeacon.sol** - An interface contract for DrawBeacon.sol
- **IPrizeFlush.sol** - An interface contract for PrizeFlush.sol
- **IDrawCalculator.sol** - An interface contract for DrawCalculator.sol
- **IDrawBuffer.sol** - An interface contract for DrawBuffer.sol
- **IBeaconTimelockTrigger.sol** - An interface contract for BeaconTimelockTrigger.sol
- **IDrawCalculatorTimelock.sol** - An interface contract for DrawCalculatorTimelock.sol
- **IReserve.sol** - An interface contract for Reserve.sol
- **IControlledToken.sol** - An interface contract for ControlledToken.sol
- **IPrizeDistributionSource.sol** - An interface contract for PrizeDistributionSource.sol

- **IPrizeDistributionFactory.sol** - An interface contract for PrizeDistributionFactory.sol

Privileged roles

- Owner:
  - The PrizePool and StakePrizePool owner has the authority to set balance and liquidity caps, reward per second, claim interval, free exit duration, Lido APR, as well as set Ticket, DrawBeacon, and PrizeFlush addresses.
  - The DrawBeacon owner can set a new DrawBuffer address and adjust the beaconPeriodSeconds.
  - The DrawBuffer owner can create and set a new Draw, and can also set a new prizeDistributor address.
  - The PrizeDistributionBuffer owner can create and set a new prizeDistribution.
  - The PrizeDistributor owner can distribute rewards to winner addresses, withdraw ERC20 token from the contract, and set new DrawBuffer and prizeDistributionBuffer addresses and distribution percentages.
  - The Reserve owner can withdraw ERC20 token from the contract.
  - The Manageable owner can set a new manager address.
  - The Ownable owner can renounce or transfer ownership.
  - The PrizeDistributionFactory owner can create and set a prize distribution, as well as set minimum pick cost and end timestamps offset.
  - The PrizeFlush owner can set destination, reserve, prizePool, protocolFeeRecipient addresses, and ProtocolFeePercentage. The owner can withdraw the reserve balance to the destination address.
  - The BeaconTimelockTrigger owner can lock and push new prize distribution.
  - The DrawCalculatorTimelock owner can set and edit the timelock.
  - The Delegation owner can set delegation lock timestamp and execute low-level calls.
  - The TWABDelegator owner can set minimum and maximum lock durations.
- PendingOwner:
  - The Ownable pendingOwner can claim ownership.
- OnlyController:
  - The controller address for the ControlledToken contract can mint and burn tokens.
- Manager:
  - The DrawBuffer manager can push a new Draw.

- ○ The PrizeDistributionBuffer manager can create and set a new prizeDistribution.
- ○ The PrizeDistributor manager can distribute rewards to winner addresses.
- ○ The Reserve manager can withdraw ERC20 token from the contract.
- ○ The PrizeDistributionFactory manager can create and set a new prize distribution.
- ○ The PrizeFlush manager can withdraw the reserve balance to the destination address.
- ○ The BeaconTimelockTrigger manager can lock and push a new prize distribution.
- ○ The DrawCalculatorTimelock manager can set a new timelock.

Checked Items

We have audited the Customers' smart contracts for commonly known and specific vulnerabilities. Here are some items considered:

Item	Type	Description	Status
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	Passed
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	Passed
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	Passed
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	Passed
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	Passed
Access Control & Authorization	CWE-284	Ownership takeover should not be possible. All crucial functions should be protected. Users could not affect data that belongs to other users.	Passed
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to users.	Not Relevant
Check-Effect-Interaction	SWC-107	Check-Effect-Interaction pattern should be followed if the code performs ANY external call.	Passed
Assert Violation	SWC-110	Properly functioning code should never reach a failing assert statement.	Passed
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	Passed
Delegatecall to Untrusted Callee	SWC-112	Delegatecalls should only be allowed to trusted addresses.	Passed
DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	Passed
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	Passed

Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	Not Relevant
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	Passed
Signature Unique Id	SWC-117 SWC-121 SWC-122 EIP-155 EIP-712	Signed messages should always have a unique id. A transaction hash should not be used as a unique id. Chain identifiers should always be used. All parameters from the signature should be used in signer recovery. EIP-712 should be followed during a signer verification.	Passed
Shadowing State Variable	SWC-119	State variables should not be shadowed.	Passed
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	Passed
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order.	Passed
Calls Only to Trusted Addresses	EEA-Level-2 SWC-126	All external calls should be performed only to trusted addresses.	Passed
Presence of Unused Variables	SWC-131	The code should not contain unused variables if this is not justified by design.	Passed
EIP Standards Violation	EIP	EIP standards should not be violated.	Passed
Assets Integrity	**Custom**	Funds are protected and cannot be withdrawn without proper permissions or be locked on the contract.	Passed
User Balances Manipulation	**Custom**	Contract owners or any other third party should not be able to access funds belonging to users.	Passed
Data Consistency	**Custom**	Smart contract data should be consistent all over the data flow.	Passed
Flashloan Attack	**Custom**	When working with exchange rates, they should be received from a trusted source and not be vulnerable to short-term rate changes that can be achieved by using flash loans. Oracles should be used.	Not Relevant
Token Supply Manipulation	**Custom**	Tokens can be minted only according to rules specified in a whitepaper or any other documentation provided by the Customer.	Not Relevant

Gas Limit and Loops	**Custom**	Transaction execution costs should not depend dramatically on the amount of data stored on the contract. There should not be any cases when execution fails due to the block Gas limit.	Passed
Style Guide Violation	**Custom**	Style guides and best practices should be followed.	Passed
Requirements Compliance	**Custom**	The code should be compliant with the requirements provided by the Customer.	Passed
Environment Consistency	**Custom**	The project should contain a configured development environment with a comprehensive description of how to compile, build and deploy the code.	Passed
Secure Oracles Usage	**Custom**	The code should have the ability to pause specific data feeds that it relies on. This should be done to protect a contract from compromised oracles.	Not Relevant
Tests Coverage	**Custom**	The code should be covered with unit tests. Test coverage should be sufficient, with both negative and positive cases covered. Usage of contracts by multiple users should be tested.	Passed
Stable Imports	**Custom**	The code should not reference draft contracts, which may be changed in the future.	Passed

Findings

■■■■ Critical

C01. Weak Source of Randomness

The distribution of prizes is accomplished through the *payWinners()* function, which requires the owner to provide a *_winners* parameter. The owner selects the winning addresses off-chain, from “random.org”, and subsequently executes the *payWinners()* function in accordance with the off-chain algorithm.

Since this algorithm is off-chain, there exists a risk of malicious actors utilizing fraudulent algorithms to select winning addresses.

Path: ./contracts/core/PrizeDistributor.sol: payWinners()

Recommendation: Implement verifiable randomness oracle, rather than relying solely on off-chain randomness algorithm.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status:Mitigated. The random number selection is now being done with Chainlink VRF. However, the system still accepts admin-supplied random addresses to determine the winners. See the risks section for more details.

■■■ High

H01. Highly Permissive Role Access

Owner can withdraw the user prizes via withdrawERC20() without notifying the users.

Paths: ./contracts/core/prize-pool/PrizePool.sol: withdrawERC20(),

./contracts/core/Reserve.sol: withdrawTo()

Recommendation: Either provide a detailed explanation of the functionality for users in the public documentation or limit the privileges of the owner.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Updated doc link: https://docs.asymetrix.io/asymetrix-protocol-guide/faq)

■■ Medium

M01. Contradiction - Missing Validation

The current implementation allows the _liquidityCap of a prizePool to be lower than the total supply.

This can lead to imbalances.

Path: ./contracts/core/prize-pool/PrizePool.sol: _setLiquidityCap()

Recommendation: Implement check before setting the new value or document this as a feature. In addition to that, oracles can be used to gather caps from off-chain.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M02. Inconsistent Data - Variable Is not Limited

The *claimInterval* variable currently lacks a defined limit. Setting this variable beyond its intended range may result in the blocking of the reward claim process.

Path: ./contracts/core/prize-pool/PrizePool.sol : _setClaimInterval(),

Recommendation: Consider limiting the claimInterval value in order to prevent unexpectedly blocking the reward claim process.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M03. Best Practice Violation - Uninitialized Function

The code contains unimplemented virtual functions.

Path: ./contracts/core/permit/EIP2612PermitAndDeposit.sol : initialize()

Recommendation: Either explain the intention of uninitialized functions or initialize the function.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M04. Inefficient Gas Model - Uncontrolled Num of Loop Iterations

The numbers of iterations of the loop in the functions are uncontrolled as it depends on given input data.

Paths: ./contracts/core/DrawCalculator.sol: _getNormalizedBalancesAt()

./contracts/core/PrizeDistributionBuffer.sol : getPrizeDistributions()

./contracts/core/PrizeDistributior.sol : _setDistribution(), payWinners()

./contracts/core/Ticket.sol : getBalancesAt(), getTotalSuppliesAt(), _getAverageBalancesBetween(),

./contracts/core/DrawBuffer.sol : getDraws(),

./contracts/core/DrawCalculator.sol : calculateNumberOfUserPicks(),

./contracts/core/prize-pool/PrizePool.sol: awardExternalERC721(),

./contracts/periphery/TwabRewards.sol : claimRewards(),

./contracts/periphery/TwabRewards.sol : getRewardsAmount(),

./contracts/twab-delegator/Delegation.sol : executeCalls(),

./contracts/twab-delegator/PermitAndMulticall.sol: _multicall()

Recommendation: Implement loop length limitation.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M05. Contradiction - Missing Validation

It is possible for individuals to generate promotions in the past.The absence of a check to verify that *_startTimestamp > block.timestamp* can result in the creation of redundant inactive promotions.

Path: ./contracts/periphery/TwabRewards.sol : createPromotion()

Recommendation: Implement *require(_startTimestamp > block.timestamp)* check for the function.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M06. Contradiction - Missing Validation

It is possible to establish a minimum lock duration that exceeds the maximum lock duration. This can cause reverts on create and update delegation functions.

Path: ./contracts/twab-delegator/TWABDelegator.sol : _setMinLockDuration(), _setMaxLockDuration()

Recommendation: Implement minimum and maximum lock duration controls to setter functions.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M07. Inconsistent Data - Natspec Mismatch

The functions NatSpec states the function can be called “owner or manager” but the functions can only be called from the owner.

Path: ./contracts/periphery/PrizeDistributionFactory.sol : setPrizeDistribution(), setMinPickCost(), setEndTimestampOffset()

Recommendation: Fix the commented part according to the logic.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M08. Inconsistent Data - Unused Return Value

The functions push(), pushPrizeDistribution(), startDraw() perform function calls but ignore the return values.

This can lead to wrong assumptions on the call results.

Paths: ./contracts/core/DrawBeacon.sol :startDraw()

./contracts/periphery/PrizeDistributionFactory.sol: pushPrizeDistribution()

./contracts/timelocks/BeaconTimelockTrigger.sol : push()

Recommendation: Implement the return value check.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: dc7979fcc57daf2edaed4192ce9faf6c1e813326)

M09. Best Practice Violation - Checks-Effects-Interactions Pattern Violation

The Checks-Effects-Interactions pattern is violated. During the startDraw(), payWinners() and claim() function calls , some state variables are updated after the external calls.

This can lead to reentrancies, denial-of-service attacks, or race conditions.

Paths: ./contracts/core/DrawBeacon.sol :startDraw()

./contracts/periphery/PrizeDistributor.sol: payWinners()

./contracts/core/prize-pool/PrizePool.sol : claim()

Recommendation: Implement the functions according to the Checks-Effects-Interactions pattern

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

M10. Tautology

While creating a promotion, the createPromotion function gets the current balance, then transfer _amount of tokens to the contract. After that, it checks if the new balance is equal to the sum of _amount + previous balance. This check will always return true.

Path: ./contracts/periphery/TwabRewards.sol : createPromotion()

Recommendation: Remove the redundant check.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Mitigated. A commonly used practice.

M12. Best Practice Violation - Redundant Calculation

While calculation the next beacon period's start time the following calculations are made;

- uint64 elapsedPeriods = (_time - _beaconPeriodStartedAt) / _beaconPeriodSeconds;
- return _beaconPeriodStartedAt + (elapsedPeriods * _beaconPeriodSeconds);

These calculations return _time, which is taken as a parameter.

Path: ./contracts/core/DrawBeacon.sol :_calculateNextBeaconPeriodStartTime()

Recommendation: Remove the redundant calculation.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Mitigated. The transaction can be mined in a few seconds, minutes, or even hours (in case of defender failures) after finishing the previous draw. And because of this, the next draw will be started not immediately after the previous one. It will start after the gap between the finishing of the previous one and the transaction mining time. Thus the calculation is necessary.

M13. Contradiction - Missing Validation

While getting normalized balances for a draw, the _getNormalizedBalancesAt takes two arrays as parameters. It iterates over them using only one of the parameters' lengths.

This can lead to out-of-bounds reaches.

Path: ./contracts/core/DrawCalculator.sol : _getNormalizedbalancest(),

Recommendation: Implement array length checks.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

■ Low

L01. Missing Zero Address Validation

Address parameters are being used without checking against the possibility of 0x0.

This can lead to unwanted external calls to 0x0.

Paths: ./contracts/core/Reserve.sol : initialize(),

./contracts/timelocks/BeaconTimelockTrigger.sol : __BeaconTimelockTrigger_init_unchained()

Recommendation: Implement zero address checks.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

L02. Functions That Can Be Declared External

In order to save Gas, public functions that are never called in the contract should be declared as external.

Paths: ./contracts/twab-delegator/TWABDelegator.sol : initialize()

./contracts/timelocks/BeaconTimelockTrigger.sol : initialize()

./contracts/timelocks/DrawCalculatorTimelock.sol : initialize()

./contracts/periphery/TwabRewards.sol : initialize()

./contracts/periphery/PrizeFlush.sol : initialize()

./contracts/periphery/PrizeDistributionFactory.sol : initialize()

./contracts/core/Ticket.sol : initialize()

./contracts/core/Reserve.sol : initialize()

./contracts/core/PrizeDistributionBuffer.sol : initialize()

./contracts/core/DrawCalculator.sol : initialize()

./contracts/core/DrawBeacon.sol : initialize()

./contracts/core/ControlledToken.sol : initialize()

./contracts/core/prize-pool/StakePrizePool.sol : initialize()

Recommendation: Use the external attribute for functions never called from the contract.

Found in: a3fca450f638a0ad9aa2cd1627c1e10d27c71e01

Status: Fixed (Revised commit: 8068c4bf7a0)

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed based on best industry practices at the time of the writing of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The report contains no statements or warranties on the identification of all vulnerabilities and security of the code. The report covers the code submitted and reviewed, so it may not be relevant after any modifications. Do not consider this report as a final and sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements.

While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

English is the original language of the report. The Consultant is not responsible for the correctness of the translated versions.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the Consultant cannot guarantee the explicit security of the audited smart contracts.