SynesisOne

Kanon

SMART CONTRACT AUDIT

19.01.2022

Made in Germany by Chainsulting.de

Table of contents

1. Disclaimer

The audit makes no statements or warrantees about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Synesis One LLC. If you are not the intended receptor of this document, remember that any disclosure, copying or dissemination of it is forbidden.

Major Versions / Date	Description
0.1 (11.01.2022)	Layout
0.2 (14.01.2022)	Test Deployment
0.5 (15.01.2022)	Automated Security Testing Manual Security Testing
0.7 (18.01.2022)	Verify Claims
0.9 (20.01.2022)	Summary and Recommendation
1.0 (22.01.2022)	Final document
1.1 (TBA)	Added deployed contract addresses

2. About the Project and Company

Company address:

Synesis One LLC.
P.O. Box 698 George Town
Grand Cayman KY1-1107
Cayman Islands

Website: https://www.synesis.one

Twitter: https://twitter.com/synesis_one

Medium: https://medium.com/@synesisone

Telegram: https://t.me/Synesis_One

Discord: https://discord.gg/fEut7WBnY9

LinkedIn: https://www.linkedin.com/company/synesisone

Documentation: https://synesis-one.gitbook.io/synesisone

Telegram: https://t.me/Synesis_One

2.1 Project Overview

Synesis One is a Web3 data utility and NFT marketplace for AI. It is the symbiosis of SynesisDAO and Kanon Exchange, powered by our governance token, Synesis, and semi-fungible data token, Kanon.

SynesisDAO is a decentralized autonomous organization (DAO) of ontology contributors, data traders, and ontology consumers. It crowdsources ontologies in natural language format and aggregates them immutably as a permissionless public data utility accessible by any AI system. Mind AI, the world's first natural language reasoning AI, with its novel data structures, canonicals, will be its first beneficiary and ontology consumer.

Kanon Exchange is the native NFT data marketplace for Kanon. A Kanon represents a unique ontology primitive of the mental map of Mind AI. Kanon ownership incentivizes a supply of ontology miners to SynesisDAO because ontology mining is a prerequisite to claim rewards in each period. Kanon enables a vastly more approachable monetization of data for everyone, not just for data scientists or computing specialists. Kanon can be staked as collateral and converted into a composable DeFi asset for more sophisticated transactions involving lending, borrowing, insurance, swaps, futures, and derivatives in an open financial system. The Atomic Split function will provide an option to divide an original, non-fungible (parent) Kanon into a set number of fungible (child) Kanons like stocks split in a stock market.

Synesis One will operate in a cross-chain environment. Smart contracts for Synesis One V1 will be deployed on the Solana Blockchain (SOL) to conserve gas costs, with future deployments on other chains to follow.
Expected revenue source for Synesis One is fivefold: a) mining fee paid by AI companies, b) trading fees from Kanon Exchange (2.5% per trade), c) creator royalties from all Kanons minted (2.5% per trade), d) DeFi related service fees, and e) its treasury at genesis. The treasury will fund software development, business development, marketing, and more. The Synesis Foundation will fund an Ethics Board and grants.

The utility and economics of the dual token system is gamified to recruit, retain, and reward skilled ontology miners.

3. Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 – 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

4. Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

4.1 Methodology

The auditing process follows a routine series of steps:

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

4.2 Dependencies

Dependency / Import Path	Source
anchor-lang	https://crates.io/crates/anchor-lang
anchor-spl	https://crates.io/crates/anchor-spl
spl-token	https://crates.io/crates/spl-token

4.3 Tested Files

The following are the MD5 hashes of the reviewed files. A file with a different MD5 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different MD5 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review

File	Fingerprint (MD5)
./src/admin/_auction/create_auction.rs	ea7584e7a88c2dbe08c324a53518a811
./src/admin/_auction/mod.rs	5643f7390c00f1bab60541db9d5aa4be
./src/admin/_auction/open_auction.rs	ef1c26cbd3540604e1e3c5526d4696a1
./src/admin/_game/mod.rs	ef1c26cbd3540604e1e3c5526d4696a1
./src/admin/_rewards/mod.rs	fde4658c209c85939ef309c13e2f7046
./src/admin/collections/_end_season_and_close_account.rs	68b329da9893e34099c7d8ad5cb9c940
./src/admin/collections/_set_premint_amount_per_time.rs	4c3bd5858dcbb0377a0947d26a7358a3
./src/admin/collections/_set_step.rs	9a0262ac71225389fc2b983653bb8a39
./src/admin/collections/claim_admin_escrow.rs	6deca6b729b3e40fd3a49acba04d445b
./src/admin/collections/mod.rs	0ec035974db0f0cc087e6878b16e5474

./src/admin/initialization.rs	3984c342795eed2498fd4ce7209a3c75
./src/admin/mod.rs	63212e043d2557057eb6cbc6a41312de
./src/admin/season/_cli_manual_mint_or_admin_airdrop_check_possibility_later.rs	9d7c235cce9296f7643ee814815c94ad
./src/admin/season/_test_admin_set_values.rs	4f47992ec85006e88108e9cc6671367d
./src/admin/season/create_season.rs	09bc13e22a9e58c97bb2bc2ce9e3d536
./src/admin/season/mod.rs	a8b559bfd01a85679b81ac9e70f8f17f
./src/admin/season/open_season.rs	2cfd55e39a1302c41c149a6729068a7c
./src/globals/accounts.rs	05e1bd0f7d89cc9fd296286de4ca7b59
./src/globals/constants.rs	74cd06f1705b6380a50b29c1602d559a
./src/globals/enums.rs	059b4f42df921b7fdb6ac6b047de623c
./src/globals/errors.rs	35c0ea4a4fb190202278dd23a140c69f
./src/globals/event.rs	9fd95e27fb640d6f438563e9176ef1dc
./src/globals/mod.rs	df5480edda18fc94d9a5de3f5e8a86ec
./src/globals/traits.rs	f7d0cc70a36e817e58b8f7553fca5fe8
./src/globals/utility.rs	fe5485e975bb797a4571ea148efef24a
./src/lib.rs	38515405e8c88f514779eca75e0fedd3
./src/marketplace/_auction/mod.rs	68b329da9893e34099c7d8ad5cb9c940
./src/marketplace/_game/mod.rs	68b329da9893e34099c7d8ad5cb9c940
./src/marketplace/_rewards/apply_reward.rs	f0563e882f69852f8feaac0a93a0e9c8
./src/marketplace/_rewards/mod.rs	fde4658c209c85939ef309c13e2f7046
./src/marketplace/_season/mod.rs	68b329da9893e34099c7d8ad5cb9c940
./src/marketplace/collections/_upgrade_nft_metadata.rs	0bf478fbc9440bf1d5216a1509d0fc43
./src/marketplace/collections/claim_airdrop_reserved_nfts.rs	467cce5ea998e606c477e8213ee5d9c9
./src/marketplace/collections/freely_mint_nft_one.rs	0e49c72f4c9d46c6be3f1d16020bb642
./src/marketplace/collections/initialize_user_reserved_account.rs	95c82ce49d7cdb7bd74a8ae05a51c390
./src/marketplace/collections/mint_whitelist_nft_one.rs	b59ed4353684cfdfe552d7ed1599d169
./src/marketplace/collections/mod.rs	baae10d9bc6d2b99bd095f761333723c
./src/marketplace/mod.rs	8bbed041f309e5c3e11166d75621e2c9

Commit: https://github.com/Synesis-One/Kanon-prototype (afb14c3e7d248749d3d587a1aa1039966b55ed5d)

4.4 Metrics / Source Unites in Scope

Total : 38 files, 1032 codes, 424 comments, 243 blanks, all 1699 lines

filename	code	comment	total
./src/admin/_auction/create_auction.rs	1	0	2
./src/admin/_auction/mod.rs	2	0	4
./src/admin/_auction/open_auction.rs	1	20	22
./src/admin/_game/mod.rs	0	0	1
./src/admin/_rewards/mod.rs	2	0	4
./src/admin/collections/_end_season_and_close_account.rs	0	0	2
./src/admin/collections/_set_premint_amount_per_time.rs	3	36	47
./src/admin/collections/_set_step.rs	3	62	74
./src/admin/collections/claim_admin_escrow.rs	58	10	78
./src/admin/collections/mod.rs	2	0	4
./src/admin/initialization.rs	30	12	49

filename	code	comment	total
./src/admin/mod.rs	6	0	8
./src/admin/season/_cli_manual_mint_or_admin_airdrop_check_possibility_later.rs	1	20	22
./src/admin/season/_test_admin_set_values.rs	44	8	59
./src/admin/season/create_season.rs	82	29	128
./src/admin/season/mod.rs	4	2	8
./src/admin/season/open_season.rs	47	12	68
./src/globals/accounts.rs	111	23	153
./src/globals/constants.rs	17	0	22
./src/globals/enums.rs	23	12	39
./src/globals/errors.rs	22	3	28
./src/globals/event.rs	12	0	16
./src/globals/mod.rs	14	0	16
./src/globals/traits.rs	7	0	10
./src/globals/utility.rs	12	33	50

filename	code	comment	total
./src/lib.rs	54	17	86
./src/marketplace/_auction/mod.rs	0	0	2
./src/marketplace/_game/mod.rs	0	0	1
./src/marketplace/_rewards/apply_reward.rs	1	6	7
./src/marketplace/_rewards/mod.rs	2	0	4
./src/marketplace/_season/mod.rs	0	0	2
./src/marketplace/collections/_upgrade_nft_metadata.rs	1	1	2
./src/marketplace/collections/claim_airdrop_reserved_nfts.rs	117	35	175
./src/marketplace/collections/freely_mint_nft_one.rs	138	34	199
./src/marketplace/collections/initialize_user_reserved_account.rs	57	3	71
./src/marketplace/collections/mint_whitelist_nft_one.rs	148	38	214
./src/marketplace/collections/mod.rs	8	0	10
./src/marketplace/mod.rs	2	8	12

5. Scope of Work

The Synesis One Team provided us with the files that needs to be tested. The scope of the audit is the kanon program for solana network.

The team put forward the following assumptions regarding the security, usage of the program:

- The program is coded according to the newest standards and in a secure way

The main goal of this audit was to verify these claims. The auditors can provide additional feedback on the code upon the client's request.

5.1 Manual Vulnerability Test

CRITICAL ISSUES

During the audit, Chainsulting's experts found **no Critical issues** in the code of the smart contract.

HIGH ISSUES

During the audit, Chainsulting's experts found **no High issues** in the code of the smart contract.

MEDIUM ISSUES

5.1.1 Sanitize your codebase
Severity: MEDIUM
Status: ACKNOWLEDGED
Code: NA
File(s) affected: All

Attack / Description	Code Snippet	Result/Recommendation
No sanitization of the code has been run before.	NA	It's recommended to use sanitization tools to keep the codebase clean and tidy. 1. Run cargo clippy and fix all the warnings 2. Add #![warn(clippy::pedantic)] and fix all the new warnings https://github.com/rust-lang/rust-clippy https://brson.github.io/2017/07/10/how-rust-is-tested

5.1.2 Use u64 instead of f64

Severity: MEDIUM
Status: ACKNOWLEDGED
Code: NA
File(s) affected: accounts.rs

Attack / Description	Code Snippet	Result/Recommendation
A Duration type to represent a span of time, typically used for system timeouts. Each Duration is composed of a whole number of seconds and a fractional part represented in nanoseconds. If the underlying system does not support nanosecond-level precision, APIs binding a system timeout will typically round up the number of nanoseconds. Durations implement many common traits, including Add, Sub, and other ops traits. It implements Default by returning a zero-length Duration.	Line: 65 `pub fn get_current_step(&self, current_timestamp: i64) -> SeasonStep {`	It’s recommended In “CollectionAccount::get_current_step”: make “duration” a “u64” and use “checked_sub” instead of the comparison to “0 as f64”. There's no real reason to use “f64” here, so it should be avoided.

LOW ISSUES

5.1.3 Missing rustdoc documentation

Severity: LOW
Status: ACKNOWLEDGED
Code: CWE-1056
File(s) affected: All

Attack / Description	Code Snippet	Result/Recommendation
rust can use a special form of comments to provide rich documentation for functions, return variables and more. This special form is named as rustdoc.	NA	It is recommended to include rustdoc documentation and follow https://doc.rust-lang.org/rustdoc/the-doc-attribute.html Rustdoc makes it easier to review and understand your codebase. There needs to be way more inline documentation for types, traits, functions. The small overviews have been useful and need to get added at each file kanon_program/programs/kanon_program/src/marketplace/collections/mint_whitelist_nft_one.rs Some of the struct fields has been documented (eg in kanon_program/programs/kanon_program/src/globals/accounts.rs:17) with normal comments, these should be doc comments (tripple slash).

5.1.4 Visibility and privacy

Severity: LOW
Status: ACKNOWLEDGED
Code: NA
File(s) affected: All

Attack / Description	Code Snippet	Result/Recommendation
Some of your files are prefixed by _file. Probably to indicate it's a private file, which doesn't look common to use for us.	_filename.rs	It's recommended to manage visibility and privacy by using 'pub(crate)'. https://doc.rust-lang.org/reference/visibility-and-privacy.html

5.1.5 Commented code

Severity: LOW
Status: ACKNOWLEDGED
Code: NA
File(s) affected: _set_premint_amount_per_time.rs, _set_step.rs, enums.rs

Attack / Description	Code Snippet	Result/Recommendation
Some of your files are including commented unused code	NA	It's recommended to remove dead / unused code for better readability

5.1.6 Unused arguments

Severity: LOW
Status: ACKNOWLEDGED
Code: NA
File(s) affected: claim_airdrop_reserved_nfts.rs, freely_mint_nft_one.rs, mint_whitelist_nft_one.rs

Attack / Description	Code Snippet	Result/Recommendation
Some of your arguments seems to be unused.	ClaimAirdropReservedNftsArgs::reserved_whitelist_proof` ClaimAirdropReservedNftsArgs::user_mint_reserve_bump FreelyMintNftOneArgs::whitelist_proof FreelyMintNftOneArgs::user_mint_reserve_bump MintWhitelistNftOneArgs::user_mint_reserve_bump	It’s recommended to remove unused code for better readability

5.1.7 Test coverage

Severity: LOW
Status: ACKNOWLEDGED
Code: NA
File(s) affected: ./tests

Attack / Description	Code Snippet	Result/Recommendation
Tests are Rust functions that verify that the non-test code is functioning in the expected manner. The bodies of test functions typically perform some setup, run the code we want to test, then assert whether the results are what we expect.	NA	We recommend to add more tests to marketplace.ts, you don't have any tests for ClaimAirdropReservedNfts, FreelyMintNftOne and MintWhitelistNftOne.

INFORMATIONAL ISSUES

5.1.8 Project architecture
Severity: INFORMATIONAL
Status: ACKNOWLEDGED
Code: NA
File(s) affected: All

Attack / Description	Code Snippet	Result/Recommendation
It's not typical to have in the folder hierarchy a subdirectory under programs	./kanon_program/programs/kanon_program/src	It's recommended move the files/folder from kanon_program directly into the root of programs

5.2 Automated Vulnerability Tests

Test	Result	Recommendation
Cargo Geiger (Dependency Checker)	NA	-
RustSec Advisory Client	scanning vulnerabilities (139 crate dependencies)	No findings
Clippy	`kanon_program` (lib) generated 83 warnings	Run cargo clippy and fix all warnings
prusti	NA	NA

6. Executive Summary

Our Chainsulting experts performed an audit of the program codebase (rust). The final debriefs took place on the January 21, 2022.

The main goal of the audit was to verify the claims regarding the security of the program. During the audit, no critical issues were found, after the manual and automated security testing and the claim have been successfully verified. The developers need to increase documentation of the codebase and sanitize probably.

7. Deployed Program

PENDING