KÖRAlabs

Handles Decentralized Minting (DeMi)

Smart Contract(s) Audit
April 2025

Eric Lee **Jesse Anderson**

Contents:

Project Overview

Handles (formerly ADA Handle) are like blockchain usernames—human-readable NFT identifiers (e.g., `$alice`) that map to a Cardano address. This makes transactions more user-friendly, replacing long bech32 addresses with something intuitive.

Historically, minting these Handles has been **centralized**, handled by a single authority using a **native script with a single signer**. This project aims to move Handles minting to a **Decentralized Minting (DeMi)** "order and mint" system, where anyone can submit an order for a named Handle to an orders contract.

Order and Mint System

- Instead of Handles being minted directly by a central authority, users can now **submit an order** to a **smart contract** that acts as an order book.
- The smart contract records user requests for specific Handle names.

Authorized Batching

- A decentralized set of actors known as **"batchers"** monitor the orders contract.
- They fulfill requests by minting the requested Handle using a **minting contract**.
- While this is still semi-permissioned (batchers are "authorized"), it's a step toward more decentralization than the single-signature model and can later be managed by a DAO.

Key Mechanics

- On Cardano, the minting policy ID (used to track and verify NFT authenticity) is tied to the script that mints the token.
- If you upgrade the script in a traditional way, you'd **change the policy ID**, which breaks backward compatibility.

So to avoid this:

- The system uses the **"withdraw zero" technique**—this allows the smart contract to **trigger additional validation logic without affecting the core minting script**.
- This lets them **update internal contract logic** while **preserving the same minting policy**, ensuring consistent integration with wallets and dApps.

Merkle Patricia Forestry & Merkle Root Management

- The **Merkle root** is a compact cryptographic summary of all existing Handle names.
- It's managed by the **Merkle Patricia Forestry** project (by the Aiken team), which enables efficient, tamper-proof state tracking.
- This root can be used to verify whether a Handle has been minted or not, in constant time.
- Since the team is launching DeMi in **beta**, they'll still maintain the **legacy centralized minter** to keep this Merkle root updated during the transition

Auditor Introduction

Eric Lee is an independent contractor that has been working in the Cardano space on smart contracts written in both Aiken and Helios and has experience with the testing mechanisms of both DSLs and off-chain tooling.

Jesse Anderson has worked in the fintech industry as a Master Platform Engineer with a threat analysis focus for over a decade and has software engineering experience that spans three decades.

Full disclosure - Jesse is also an owner of Kora Labs and a founder of Ada Handle. Jesse helped with the specification and threat analysis portions of this audit, but ultimately Eric is the approving auditor as an independent contractor.

Audit Overview

Our approach in auditing is to first understand the specification, whether it is written or unwritten. We then reiterate what we believe the specification to be in the Project Overview (above). The overview may not contain every detail, but provides enough information to analyze the application in purpose, security, and performance.

We also take a look at the test cases the project has - whether manual or automated, if any. These test cases not only teach us about the intended functionality of the application, but they offer a layer of assurance that the application is performing the intended function and will continue to perform as the application matures.

We then perform our own testing and analysis of the results.

This project was successful in its objectives in moving Handles minting to a Decentralized protocol. While decentralized minting does not yet have feature parity with current “legacy” minting, releasing in a “beta” phase while keeping legacy minting open helps to delineate and mitigate user expectation.

We did find a few issues that should either be addressed, or are informational in nature and don’t necessarily need to be addressed. The project does have automated testing in place with minimal tests. We also listed a few test cases that should be added into the repository to keep project objectives in check as well as our own test results for the suggested tests.

It should be noted that this audit was performed on a limited contract with a limited budget and as such should be considered a cursory audit. It is recommended that the project delves deeper into the auditing process as the project progresses.

Project Test Cases

Test Name	Test Result
Should mint single DeMi Handle via order	SUCCESS
Should mint multiple Handle DeMi via orders	SUCCESS
Should mint multiple Legacy Handles	SUCCESS
DeMi Handles and Legacy Handles should coexist in Merkle root	SUCCESS
Should deny DeMi SubHandles as they are not yet supported	SUCCESS
DeMi SubHandle order cancel is successful	SUCCESS
Should validate Handle length restrictions	SUCCESS
DeMi Handle order cancel is successful	SUCCESS
Should deny Legacy mints if “minted” value doesn’t contain correct tokens	SUCCESS
Should mint Legacy SubHandles	SUCCESS
Should mint Legacy Virtual SubHandles	SUCCESS
Should validate Legacy SubHandle length restrictions	SUCCESS
Should deny Legacy Virtual SubHandles if minted value is incorrect	SUCCESS

Suggested Test Cases

These are the tests that we performed that were not already part of the project repository. It is suggested that they be added and made part of the regular automated testing process.

Test Name	Status
Should deny DeMi mints if minted assets aren't correct	ACKNOWLEDGED
Should deny DeMi mints if minter KeyHash isn't in controls	ACKNOWLEDGED
Should deny if treasury fees aren't paid to treasury	ACKNOWLEDGED
Should deny if minter fees aren't paid to minter	ACKNOWLEDGED
Should calculate correct return amount on fulfilled order	ACKNOWLEDGED
Deny when reference token is not sent to `pz_script_address`	ACKNOWLEDGED
Deny if Merkle hash output doesn't contain expected root hash	ACKNOWLEDGED

Smart Contract Controls

Policy

f0ff48bbb7bbe9d59a40f1ce90e9e9d0ff5002ec48f232b49ca0fb9a

NFTs

Asset Name	Purpose
(222) demi@handle_settings	Defines SettingsV1 (see below)
(222) handle_root@handle_settings	Holds the Merkle Patricia Trie Root Hash

SettingsV1

Property Name	Purpose
policy_id	Current Handles minting policy
allowed_minters	List of KeyHashes of approved mint batchers
treasury_address	Address to send the `treasury_fee` to
treasury_fee	Amount the Handles DAO treasury collects per Handle mint
minter_fee	Amount an approved minter collects per Handle mint
pz_script_address	The destination address of the (100) Reference Token of the Handle
order_script_hash	Where order UTxOs are located
minting_data_script_hash	Where the Merkle Patricia Trie Root Hash is located

Smart Contract Interdependencies

Contract	Execution Dependency
mint_proxy.ak	mint_v1.ak (by way of withdraw)
orders.ak	mint_v1.ak (by way of withdraw)
mint_v1.ak	minting_data.ak (by way of spent input with NFT control)

Findings

KOR-100: No guarantee that Legacy Handle mints will execute the contract

Reference Identifier	Severity	Status
KOR-100	MAJOR	MITIGATED

Recommendation: Close the DeMi “beta” phase as soon as possible - minting only on the DeMi policy - to eliminate the chance of accidental double-mints due to legacy native script policy.

Project Mitigation: The minting engine that controls legacy minting has multiple double-mint controls that have prevented double-mints for 260K mints. That minting engine has retained all double-mint controls and also only executes legacy minting through the new DeMi scripts.

KOR-101: Contract controls are not locked at a smart contract

Reference Identifier	Severity	Status
KOR-101	MAJOR	MITIGATED

Recommendation: Best practice dictates locking contract controls at another smart contract that disallows improper spend and enforces control values. Disregarding this could result in loss of controls or invalid values.

Project Mitigation: Smart contract controls are currently locked at a mult-sig wallet, requiring 2 of 4 signatures to spend. The multi-sig wallet UX allows for review of datum changes.

KOR-200: Multiple mint batchers can cause UTxO contention on Orders

Reference Identifier	Severity	Status
KOR-200	MINOR	ACKNOWLEDGED

Recommendation: Devise a system for Order distribution amongst approved mint batchers.

Project Acknowledgement: DeMi Handles are being released in a “beta” phase. During the “beta” phase, only Kora will be an approved batcher. This will be addressed before exiting the “beta” phase and accepting additional mint batchers.

KOR-201: Multiple mint batchers can cause UTxO contention on the Merkle root

Reference Identifier	Severity	Status
KOR-201	MINOR	ACKNOWLEDGED

Recommendation: Devise a system for sharing the spend of the Merkle root.

Project Acknowledgement: DeMi Handles are being released in a “beta” phase. During the “beta” phase, only Kora will be an approved batcher. This will be addressed before exiting the “beta” phase and accepting additional mint batchers.

KOR-202: Price controls don’t allow for dynamic pricing or batcher unique pricing

Reference Identifier	Severity	Status
KOR-202	MINOR	PENDING

Handles are priced by a combination of rarity-by-length and recent ADAUSD price quotes. The current price controls do not account for this. Additionally, if a new batcher wishes to have different pricing, the current controls don’t allow for it.

Recommendation: Allow pricing to be updated in a separate UTxO with an NFT control specific to each approved batcher.

KOR-203: Price controls don’t account for SubHandle pricing set by owners

Reference Identifier	Severity	Status
KOR-203	MINOR	ACKNOWLEDGED

The current price controls don’t acknowledge pricing of root Handle owner settings for SubHandles.

Recommendation: DeMi Handles should acknowledge the same pricing controls as Legacy Handles - the paired CIP-67 token with the (001) `asset_name_label` that houses the root Handle owner’s pricing tiers.

Project Acknowledgement: DeMi Handles are being released in a “beta” phase and this feature is not currently in scope of delivery. Feature limitations are communicated and agreed to at time of ordering.

KOR-300: Project test cases can be increased

Reference Identifier	Severity	Status
KOR-300	INFORMATIONAL	ACKNOWLEDGED

Recommendation: Increase test coverage per the *Suggested Test Cases* section above.

Project Acknowledgement: Suggested test cases will be added, but not currently in scope of delivery.

KOR-301: DeMi SubHandles aren't supported

Reference Identifier	Severity	Status
KOR-301	INFORMATIONAL	ACKNOWLEDGED

Recommendation: Informational only. Add the feature before DeMi Handles are mainstream for parity with current Handles and re-audit at that time.

Project Acknowledgement: DeMi Handles are being released in a "beta" phase and this feature is not currently in scope of delivery. Feature limitations are communicated and agreed to at time of ordering.

KOR-302: DeMi Personalization isn't supported

Reference Identifier	Severity	Status
KOR-302	INFORMATIONAL	ACKNOWLEDGED

Recommendation: Informational only. Add the feature before DeMi Handles are mainstream for parity with current Handles and re-audit at that time.

Project Acknowledgement: DeMi Handles are being released in a "beta" phase and this feature is not currently in scope of delivery. Feature limitations are communicated and agreed to at time of ordering.

KOR-303: Project should provide a standard interface for Order creation

Reference Identifier	Severity	Status
KOR-303	INFORMATIONAL	ACKNOWLEDGED

Recommendation: Provide a CIP-57 compliant "blueprint" for the Order smart contract.
=
Project Acknowledgement: Suggested blueprint will be added, but not currently in scope of delivery.

KOR-304: Project should provide a standard interface for mint batching

Reference Identifier	Severity	Status
KOR-304	INFORMATIONAL	ACKNOWLEDGED

Recommendation: Provide a CIP-57 compliant "blueprint" for the set of minting smart contracts.

Project Acknowledgement: Suggested blueprint will be added, but not currently in scope of delivery.

File List

Repository: https://github.com/koralabs/decentralized-minting
Commit: 5660d96519f6499140b4fff239a9cf565361d828

Main Contracts	Hash (SHA1)
smart-contract/validators/mint_proxy.ak	88fde33f42566c218f2d7ae7ed7a305b617f76b3
smart-contract/validators/mint_v1.ak	08617a4684d389e1a9d19cb0d350590c29a15a0c
smart-contract/validators/minting_data.ak	da405e55e0805fd927e1b94d11df46785207045b
smart-contract/validators/orders.ak	2862ab690a1b21bfc5ab0d93125d1c01eba94634

Contract Dependencies	Hash (SHA1)
smart-contract/lib/common/cip68.ak	55dbe44dcdbda4ae1fb5b4bec7660485c5e7aa2c
smart-contract/lib/common/hashes.ak	29fe5e3e05de611064ba6757349620a3513e63b3
smart-contract/validators/minting_data.ak	da405e55e0805fd927e1b94d11df46785207045b
smart-contract/validators/orders.ak	2862ab690a1b21bfc5ab0d93125d1c01eba94634
smart-contract/lib/decentralized_minting/minting_data.ak	409dc305413bb82ed7f19acdee2d303ffe044fca
smart-contract/lib/decentralized_minting/orders.ak	fe34de6dfec278a27f7054d68f99b7b61f2eb614
smart-contract/lib/decentralized_minting/settings.ak	b872be156866868f845eb4df078bf47c99f54d6c
smart-contract/lib/decentralized_minting/settings_v1.ak	1fc66d79ea4390217c07e94c94a72c65721c5175
smart-contract/lib/tests/mint_v1.test.ak	2879a351de77b0b2b31c07853866236e4ca1213a
smart-contract/lib/tests/minting_data/utils.test.ak	6d73606988ec90bfa6751e01f87e84dee1827b5c
smart-contract/lib/tests/mocks/fake_constants.ak	fb7f7db7b96677eafeae168529177931d2c13abe
smart-contract/lib/tests/mocks/fake_tx.ak	5db62ece1760fdd1bf231c93f21787f0d159f699
smart-contract/lib/validations/mint_v1/utils.ak	c499a8d285e1e99733f67765169222ce88d301c2
smart-contract/lib/validations/mint_v1/validation.ak	a26fc140c4c48d265a1de48df436d577bf4cf229
smart-contract/lib/validations/minting_data/types.ak	85817fe56100b45edb189ee3d34997b835e4497a
smart-contract/lib/validations/minting_data/utils.ak	53ec1648ba47d89a8b5f20da906e4e80bb9d4b99
smart-contract/lib/validations/minting_data/validation.ak	1216118189401b44c8ae4f35bf6ee3f1130bfbbb

Test Files	Hash (SHA1)
tests/mint.test.ts	34b64770cab5580027585c5480f52ffcc078c6de
tests/setup.ts	b8d94916cb55085158b368e63dd8d8cae060d2a4
tests/utils.ts	70d0bc64046259713cc7361b17a14d0a4bb5eb74
smart-contract/lib/tests/mint_v1.test.ak	2879a351de77b0b2b31c07853866236e4ca1213a
smart-contract/lib/tests/minting_data/utils.test.ak	6d73606988ec90bfa6751e01f87e84dee1827b5c
smart-contract/lib/tests/mocks/fake_constants.ak	fb7f7db7b96677eafeae168529177931d2c13abe
smart-contract/lib/tests/mocks/fake_tx.ak	5db62ece1760fdd1bf231c93f21787f0d159f699

Classifications

Severity

Level	Description
CRITICAL	Attack vectors that compromise user or asset security
MAJOR	Failure to meet application objectives or possible chain-wide performance implications
MINOR	Non-critical or user expectation issue
INFORMATIONAL	Best practice advisory, suggestions for improvement, or observation of application behavior

Status

Level	Description
RESOLVED	The finding is fixed
MITIGATED	The finding has a temporary fix that is satisfactory, with the understanding that a more complete fix is on the way
ACKNOWLEDGED	The project acknowledges the finding, but it is currently out of scope, will be addressed at a later date, or is an acceptable behavior according to the project
PENDING	No official response from the project

Test Results

Level	Description
SUCCESS	The test was successful
FAILURE	The test failed

Revisions

v1.0: Initial audit
Revision date: 2024-04-17
Final commit: 5660d96519f6499140b4fff239a9cf565361d828

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the agreement between the listed auditors on the title page of this audit (AUDITORS) and Kora Labs (CLIENT) (the AGREEMENT), or the scope of services, and terms and conditions provided to the CLIENT in connection with the Agreement, and shall be used only subject to and to the extent permitted by such terms and conditions.

THIS REPORT MAY NOT BE TRANSMITTED, DISCLOSED, REFERRED TO, MODIFIED BY, OR RELIED UPON BY ANY PERSON FOR ANY PURPOSES WITHOUT THE AUDITORS' PRIOR WRITTEN CONSENT.

THIS REPORT IS NOT, NOR SHOULD BE CONSIDERED, AN ENDORSEMENT, APPROVAL OR DISAPPROVAL of any particular project, team, code, technology, asset or anything else. This report is not, nor should be considered, an indication of the economics or value of any technology, product or asset created by any team or project that contracts the AUDITORS to perform a smart contract assessment. THIS REPORT DOES NOT PROVIDE ANY WARRANTY OR GUARANTEE REGARDING THE QUALITY OR NATURE OF THE TECHNOLOGY ANALYZED, nor does it provide any indication of the technology's proprietors, business, business model or legal compliance.

To the fullest extent permitted by law, the AUDITORS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, AND THE RELATED SERVICES AND PRODUCTS AND YOUR USE THEREOF, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. This report is provided on an as-is, where-is, and as-available basis. The AUDITORS do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by CLIENT or any third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services, assets and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and the AUDITORS WILL NOT BE A PARTY TO OR IN ANY WAY BE RESPONSIBLE FOR MONITORING ANY TRANSACTION BETWEEN YOU AND CLIENT AND/OR ANY THIRD-PARTY PROVIDERS OF PRODUCTS OR SERVICES.

THIS REPORT SHOULD NOT BE USED IN ANY WAY BY ANYONE TO MAKE DECISIONS AROUND INVESTMENT OR INVOLVEMENT WITH ANY PARTICULAR PROJECT, services or assets, especially not to make decisions to buy or sell any assets or products. This report provides general information and is not tailored to anyone's specific situation, its content,

access, and/or usage thereof, including any associated services or materials, shall not be considered or relied upon as any form of financial, investment, tax, legal, regulatory, or other advice. This report is based on the scope of materials and documentation provided for a limited review at the time provided. The AUDITORS prepared this report as an informational exercise documenting the due diligence involved in the course of development of the CLIENT's smart contract only, and THIS REPORT MAKES NO CLAIMS OR GUARANTEES CONCERNING THE SMART CONTRACT'S OPERATION ON DEPLOYMENT OR POST-DEPLOYMENT. This report provides no opinion or guarantee on the security of the code, smart contracts, project, the related assets or anything else at the time of deployment or post deployment. Smart contracts can be invoked by anyone on the internet and as such carry substantial risk. The AUDITORS HAVE NO DUTY TO MONITOR CLIENT'S OPERATION OF THE PROJECT AND UPDATE THE REPORT ACCORDINGLY.

THE INFORMATION CONTAINED IN THIS REPORT MAY NOT BE COMPLETE NOR INCLUSIVE OF ALL VULNERABILITIES. This report is not comprehensive in scope, it excludes a number of components critical to the correct operation of this system. You agree that your access to and/or use of, including but not limited to, any associated services, products, protocols, platforms, content, assets, and materials will be at your sole risk. On its own, it cannot be considered a sufficient assessment of the correctness of the code or any technology. This report represents an extensive assessing process intending to help CLIENT increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology, however blockchain technology and cryptographic assets present a high level of ongoing risk, including but not limited to unknown risks and flaws.

While the AUDITORS have conducted an analysis to the best of its ability, it is the AUDITORS' recommendation to commission several independent audits, a public bug bounty program, as well as continuous security auditing and monitoring and/or other auditing and monitoring in line with the industry best practice. The possibility of human error in the manual review process is highly real, and the AUDITORS recommend seeking multiple independent opinions on any claims which impact any functioning of the code, project, smart contracts, systems, technology or involvement of any funds or assets. The AUDITORS POSITION IS THAT EACH COMPANY AND INDIVIDUAL ARE RESPONSIBLE FOR THEIR OWN DUE DILIGENCE AND CONTINUOUS SECURITY.