DcentraLab Diligence

Audit Report

HyperCycle - ShareTokens

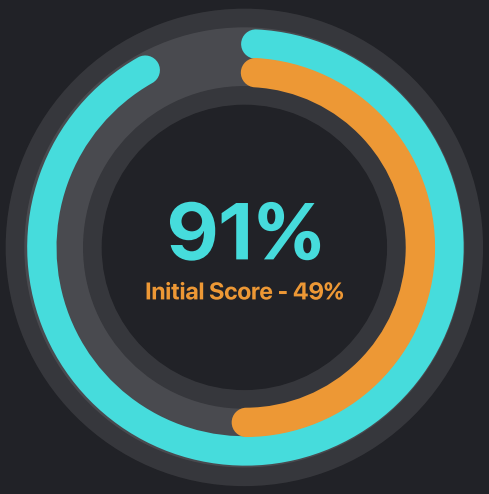
https://www.hypercycle.ai

Provided By DcentraLab Diligence on December 03, 2023

Security Audit Score

Low Risk

DcentraLab Diligence team has conducted an extensive audit on HyperCycle Contracts and has found the code to be in minimal risk level given proper deployment and multi-sig permissioning.



- Minimal Risk
- Low Risk
- Medium Risk
- High Risk
- Critical Risk

Score Guidelines

Scope

Audited Repository:

https://github.com/hypercycle-development/hypercycle-contracts

Audited Branch:

hypercycle-development

Audited Commit Hash:

f374c91b89c7b2422a43a031bc270e363a4adfbd

Fix Branch:

hypershare_development

Fixes Commit Hash:

101de7515a6d50bf8c3db04f3a89697327c0544f

Audited Contracts:

HyperCycleShareTokens.sol

Reviewed For Context:

- CrowdFundHYPCPoolV2.sol
- HyperCycleLicense.sol
- HYPCSwapV2.sol
- HYPC.sol

Nomenclature Of Issues:

E - Environmental
A - Contract HyperCicleShareTokens.sol

Contracts Architecture Overview

HyperCycleShareTokens Analysis

HyperCycleShareTokens is a ERC-1155 contract made primarily to enables users to deposit a license NFT and a cHyPC NFT, generating wealth tokens and revenue tokens.The hardware manager, through the depositRevenue() method, can add revenue to HypeCycleShareTokens, which revenue token holders can claim via claimRevenue() method, ensuring a fair distribution of income.

To maintain fungibility revenue tokens, a claim Revenue hook is triggered whenever revenue tokens are transferred. This prevents imbalances in revenue distribution when tokens change hands. Additionally, there are wealth tokens, they are not currently utilized, they are designed for potential governance functions in future contracts.

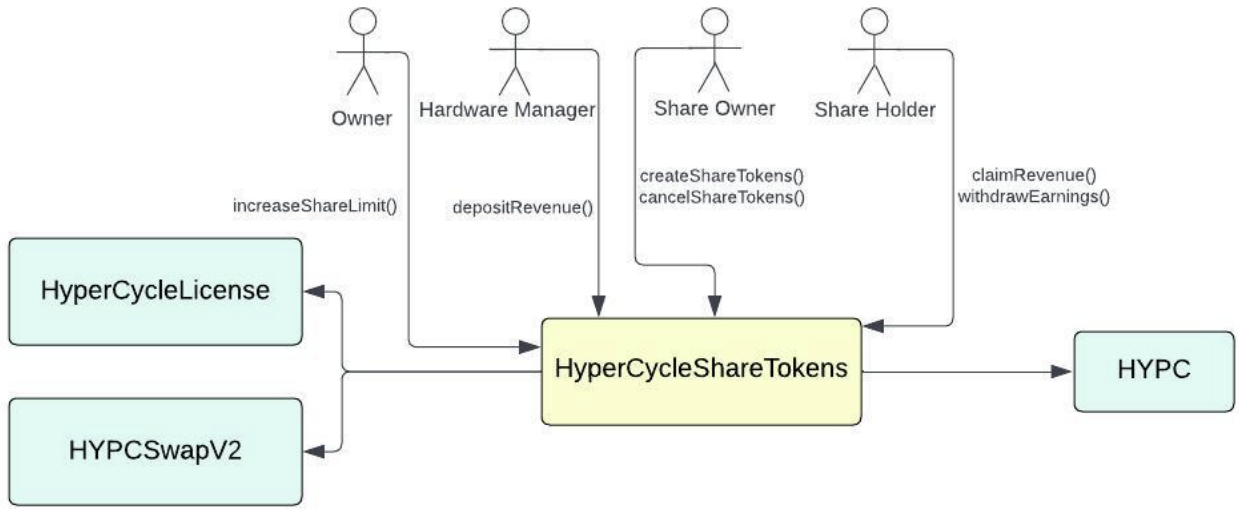
The contract is best managed via a multi-sig wallet, allowing license and CHyPC holders to create and allocate wealth and revenue tokens among themselves or other parites.

Roles:

- Share owner: The user or contract who created the share tokens
- Owner: The deployer of the HyperCycleShareTokens Contract
- Share Holder: holders of revenue tokens
- Hardware Operator: depositors of revenue to be shared for a specific set of share tokens

External Functions:

- createShareTokens: Allows users to create a new share by depositing a HyperCycle License NFT and a cHyPC NFT, resulting in the creation of wealth and revenue tokens.
- transferOwner: Allows the owner of a share to transfer ownership to another address.
- cancelShareTokens: Allows the owner of a share to cancel it, returning the deposited license and cHyPC NFTs.
- setShareMessage: Allows the owner of a share to set a message associated with the share.
- depositRevenue: Allows users to deposit revenue (HyPC tokens) into a share.
- claimRevenue: Allows users to claim their proportional revenue from a share.
- withdrawEarnings: Allows users to withdraw their claimed revenue from a share.



Issues Severity Reference Table

Type

Discussion

The issue severity is dependent on design, centralization, and product specifications of the project.

Informational

This issue is not critical and does not pose an immediate threat to the functionality or security of the smart contract. It is simply an informational item that the auditors have identified and recommends addressing for best practices or to improve the overall performance of the contract.

Low

This issue is relatively minor and does not pose a significant risk to the functionality or security of the smart contract. While it is recommended to address these issues to ensure the highest level of quality and security, they are not likely to cause significant problems if left unaddressed.

Medium

This issue poses a moderate risk to the functionality or security of the smart contract. While it may not be immediately exploitable, it has the potential to cause problems in the future if left unaddressed. It is recommended to address these issues as soon as possible to prevent any potential negative impact on the contract.

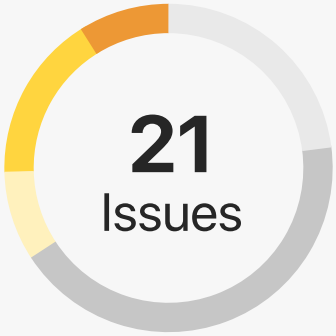
High

This issue poses a significant risk to the functionality or security of the smart contract. Addressing these issues as soon as possible is recommended to prevent any potential negative impact on the contract. Failure to address these issues could result in significant problems and potential loss of funds or other assets.

Critical

This issue poses an immediate and severe risk to the functionality or security of the smart contract. It is recommended to address these issues immediately to prevent any potential negative impact on the contract. Failure to address these issues could result in catastrophic problems and significant loss of funds or other assets.

Findings Summary



- Discussion
- Informational
- Low Risk
- Medium Risk
- High Risk
- Critical Risk

ID	Title	Severity	Status	Risk Points
A.1	Typographical Errors in Contract Documentation	Informational	Resolved	
A.2	Adding a Message on Share Creation	Informational	Resolved	
A.3	Creating Share Tokens Without Deposit CHyPC NFT	High	Resolved	-20
A.4	Limitation of `shareOwner` Modifier for Contract-Owned Shares	Discussion	Acknowledged	-1
A.5	Inability to Transfer an NFT Not Owned by the Contract	High	Resolved	-20
A.6	missing distinction between max and total supply for revenue and wealth tokens	Informational	Resolved	
A.7	Users Can Claim Revenue When There's Nothing to Claim	Low	Resolved	-2
A.8	Simplify Implementation in `shareCreated`	Informational	Resolved	

Findings Summary

ID	Title	Severity	Status	Risk Points
A.9	Potential Arithmetic Overflow	Low	Resolved	-2
A.10	Poor Function and Event Naming	Informational	Resolved	
A.11	Missing Fields In Event	Discussion	Resolved	-1
A.12	Lack of Validity Check for Address	Medium	Resolved	-5
A.13	Transferability of Canceled Shares	Discussion	Resolved Acknowledged	-1
A.14	Missing Field in Event	Informational	Acknowledged	
A.15	Lack of Existing Balance Check	Medium	Resolved	-5
A.16	Missing Field in Event	Informational	Resolved	
A.17	Typo In Event Name	Informational	Resolved	
A.18	Adding A Utility Function	Discussion	Resolved	-1
A.19	Creating more than one set of share tokens is not possible in the contract's initial state	Medium	Resolved	-5
A.20	Missing Fields in Event Description	Informational	Resolved	
A.21	Lack of mechanism for users to obtain, release, or transfer share tokens.	Discussion	Resolved	-1

Total risks score: 36/100

Quality Score Card

Factor	Positive Score Points rage	Final Score
Presence of up to date documentation	1-5	5
Code readability	1-5	5
Operational security of the team	1-5	5
Code reliance on third-party software	1-10	8
Test Coverage	1-10	3

Total quality score: 26/35

Total Score

	Risk Score	Quality Score	Total Score
Pre-Audit	36 (*0.65)	26	**49**
Final Report	100 (*0.65)	26	**91**

Complete Analysis

Local Contract Findings:
Contract: HyperCicleShareTokens.Sol

ID A.1: **Status: Resolved**

Informational | Typographical Errors in Contract Documentation

Present at: Below the imports @ L18-81

Description: Typographical errors have been identified in the contract documentation.

Recommendation:
Line 71 - change "ther" to "there"
Line 73 - change "contractto" to "contract to"
Line 75 - change "garauntee" to "guarantee"

ID A.2: **Status: Resolved**

Informational | Adding a Message on Share Creation

Present at: 'createShareTokens' function @ L329

Description: The 'createShareTokens' function currently has a hardcoded empty string as a message when creating a share

Recommendation: Consider enhancing the 'createShareTokens' function by adding a message parameter with an empty string as the default value.This improvement would provide users with the flexibility to include a message in the same transaction, potentially saving on gas costs.

Complete Analysis

ID A.3: **Status: Resolved**

High | Creating Share Tokens Without Deposit CHyPC NFT

Present at: `createShareToken` function @ L337

Description: The `createShareToken` function includes an `if` statement that checks if the `chypcNumber` parameter is greater than 0. If this condition is not met, the function still allows the creation of shares. However, as indicated in the documentation:

```
/*
...
This could be the case where
this creator is a smart contract that used the CrowdFundHYPCPoolV2 contract
to get an assignment pointed to the license Id. In this case, the smart
contract would have a guarantee that the license has backing cHyPC, but does
not own the cHyPC itself
...
*/
```

The function's logic will only work with the HyperCycleSwapV2 contract. In case the CHYPC was received via the HyperCycleSwap contract independently there is currently no proper transfer of the CHYPC to the HyperCycleShareTokens.sol contract. And in case the CHYPC was received via the HyperCycleSwap contract using the CrowdFundHYPCPoolV2 mechanism, there is currently no check in place that the license has backing CHyPC on the CrowdFundHYPCPoolV2 contract.

Recommendation: Perform additional checks or make a different function for creating shares to ensure backwards compatibility with the HyperCycleSwap and CrowdFundHYPCPoolV2.sol contracts.

Complete Analysis

ID A.4: **Status:** Acknowledged

Discussion | Limitation of `shareOwner` Modifier for Contract-Owned Shares

Present at: `transferOwner`, `cancelShareTokens`, `setShareMessage` @ L355, L364, L385

Description: The `shareOwner` modifier, present in `transferOwner`, `cancelShareTokens`, and `setShareMessage` functions, has a limitation when it comes to shares created by contracts. According to the documentation, there may be cases where the share owner is another contract that has called the `createShareTokens` function and owns the shares. In such cases, the contract developer must fully implement the interface of this contract in order to use the 3 functions that are using this modifier, or otherwise implement a general delegate call mechanism with proper permissions/proposal/voting etc..

Recommendation: Consider introducing an additional role called "share manager", designated for externally owned accounts (EOAs). Modify the `shareOwner` modifier to allow this share manager role to call the `transferOwner`, `cancelShareTokens`, and `setShareMessage` functions.

ID A.5: **Status:** Resolved

High | Inability to Transfer an NFT Not Owned by the Contract

Present at: `cancelShareTokens` function @ L374

Description: if a share was created with `chypcId` equal to 0 (as described in issue A.3), the contract will not own the HYPCSwapV2 NFT. In this scenario, the `cancelShareTokens` function will revert, preventing the cancellation of share tokens.

Recommendation: Prioritize addressing risk number 3, as the resolution of this issue depends on the fix implemented for that risk. Once risk number 3 is resolved, the `cancelShareTokens` function should be updated to handle shares created with `chypcId` equal to 0 appropriately.

Complete Analysis

ID A.6: **Status: Resolved**

Informational | **missing distinction between max and total supply for revenue and wealth tokens**

Present at: REVENUE_TOKEN_MAX_SUPPLY @ L132, WEALTH_TOKEN_MAX_SUPPLY @ 134

Description: The constant `REVENUE_TOKEN_MAX_SUPPLY` is used in the code to define the maximum supply of revenue and wealth tokens, but there is no clear way to receive the total supply at any given point in time, making the implementation inconsistent with standard expectations from fungible tokens (as per the semi-fungibility of erc1155, the fungible token id supplies should behave in a manner consistent with regular fungible tokens, i.e. having standard getters for max and total supply).

Recommendation: Consider adding getters and management for total vs max supply (e.g. in case of burning or other supply changes in the fungible token ids within the contract)

ID A.7: **Status: Resolved**

Low | **Users Can Claim Revenue When There's Nothing to Claim**

Present at: `_claimRevenue` and `claimRevenue` functions @ L410, L424

Description: Users can interact with `claimRevenue` function multiple times, even if there are no funds to claim. Currently, there is no validation or error message for this scenario. In such cases, an `ClaimRevenue` event is emitted with zero as the amount.

Recommendation: Consider making the following adjustments:
Modify the internal `_claimRevenue` function to return the `amountToGiveAddress` as its return value.
In the `claimRevenue` function, add a `require` statement to check if the `amountToGiveAddress` is greater than 0. If not, return a relevant error message to inform users that there is no revenue to claim.
These changes will provide a more user-friendly experience.

Complete Analysis

Recommendation 2: The `canClaimRevenue` modifier you have added to the claimRevenue()` function is calling the internal `_claimRevenue()` function. After it is called in the modifier, it will be called again inside the function. This will result in a double call to the `_claimRevenue()` function, which is a state-changing function that can lead to incorrect calculations (e.g., `withdrawableAmounts` will be twice its expected value).

To resolve this issue, consider removing the `canClaimRevenue` modifier and replacing it with two require statements:

Check that the user's share balance is greater than 0.
Check that the returned value from the `_claimRevenue() `function is greater than 0.

ID A.8: **Status: Resolved**

Informational | Simplify Implementation in `shareCreated`

Present at: `shareCreated` function @ L508

Description: The current implementation checks if a share number is created using a complex approach. A more straightforward approach would be to check if the share's status is different from `NOT_CREATED`.

Recommendation: Simplify the function by checking if the specific share's status by its share number using the condition:
return shareData[shareNumber].status != Status.NOT_CREATED

```
return shareData[shareNumber].status != Status.NOT_CREATED
```

Complete Analysis

ID A.9: **Status: Resolved**

Low | Potential Arithmetic Overflow

Present at: constructor @ L275

Description: The contract defines the `startNumber` and `endNumber` parameters that determine the range of share numbers to be used for creating shares. The wealth tokens are calculated as 2 * `shareNumber` + 1, which means they depend on the share numbers. There is a risk of potential overflow if `endNumber` exceeds half of the maximum value of `uint256`. Overflow in this context can lead to unexpected and undesirable behavior.

Recommendation: On constructor, ensure that `endNumber` will not exceed half of the maximum value of `uint256` minus 2 (2**255 - 2)

Recommendation 2: The fix is not solving that issue. It is solving the issue of A.19. Add a require statement that is checking that `endNumber` is less than 2*255 - 2. That will make sure there will not be an overflow on share creation (remember that wealth token number is 2*shareNumber + 1)

ID A.10: **Status: Resolved**

Informational | Poor Function and Event Naming

Present at: `transferShare` , `transferOwner` @ L210 , L355

Description: `trasnferOwner` might suggest that the function is transferring the ownership of the contract itself.
The event `transferShare` is misleading as it implies the transfer of shares themselves. However, this event is actually describing the transferring of ownership of a share, not the share itself.

Recommendation: To improve clarity consider the following: 1. renaming the function to `transferShareOwnership` 2. Renaming the event to shareOwnershipTransferred() or something similar. This will make the function's functionality more evident to developers and users.

Complete Analysis

ID A.11: **Status: Resolved**

Discussion | Missing Fields In Event

Present at: `createShareTokens` @ L342

Description: The `createShare` event is missing four fields: `status`, `revenueDeposited`, `message` and `block.timestamp`.`status`, `revenueDeposited`, `message` are static fields, and `block.timestamp` is a field that can be derived from the transaction itself. Not including these fields in the event can result in gas savings.

Recommendation: Ensure that the omission of these fields from the `createShare` event is intentional. If there are any changes to the static fields, remember to update the event accordingly.

ID A.12: **Status: Resolved**

Medium | Lack of Validity Check for Address

Present at: `transferOwner` @ L355

Description: The `transferOwner` function does not include a validity check for the address of which the ownership is being transferred. This means that ownership could potentially be transferred to the zero address, which may not be the intended behavior.

Recommendation: To prevent users from transferring ownership to the zero address, consider adding a `require` statement. If the provided address is the zero address, the function should revert with a relevant error message.

Complete Analysis

ID A.13:

Status Recommendation 1: Resolved

Status Recommendation 2: Acknowledged

Discussion | Transferability of Canceled Shares

Present at: `cancelShareTokens` @ L364

Description: Currently, shares that have been canceled can still be transferred, potentially leading to trading in shares that have no value or are worth less due to their "ENDED" status.

Recommendation: If the intention is to prevent the transfer of shares with the "ENDED" status, you can consider implementing an override of the transfer functions to disallow the transfer of shares in this status. This would help avoid situations where canceled shares are mistakenly or intentionally traded.

Recommendation2: Consider burning revenue tokens after their last claim has been made (for an ended/canceled share.

ID A.14:

Status: Acknowledged

Informational | Missing Field in Event

Present at: `setShareMessage` @ L385

Description: The MessageChanged modifier includes only the new message that was set but does not include the previous message that was changed from.

Recommendation: Consider enhancing the event by adding two fields: fromMessage, which represents the current message before the change, and toMessage, which represents the new message that was changed to. This would provide a more complete record of the message change event.

Complete Analysis

ID A.15: **Status: Resolved**

Medium | Lack of Existing Balance Check

Present at: `_claimRevenue` @ L410

Description: The _claimRevenue function does not validate whether the user actually has any ownershipRatio above 0.

Recommendation: Consider adding a require statement to validate that the user has ownershipRatio > 0 before allowing them to claim revenue. This will help prevent users from attempting to claim revenue for shares they do not own and provide a more robust validation process, while avoiding cases of silent failures when user tries to claim revenue for tokens they don't possess.

Recommendation 2: See ID A.7 recommendation 2.

ID A.16: **Status: Resolved**

Informational | Missing Field in Event

Present at: `withdrawEarnings` @ L430

Description: The EarningWithdrawl event currently lacks the amt field, which represents the amount that was withdrawn.

Recommendation: Consider adding the amt field to the EarningWithdrawl event to provide additional information about the earnings withdrawal transaction.

Complete Analysis

ID A.17: **Status: Resolved**

Informational | Typo In Event Name

Present at: `EarningsWithdrawl` @ L232

Recommendation: Should be "EarningsWithdrawal"

ID A.18: **Status: Resolved**

Discussion | Adding A Utility Function

Description: You could possibly add another utility function to combine the two functions claimRevenue and withdrawEarnings, allowing users to claim and withdraw earnings in a single transaction.

Recommendation: Consider adding a separate function called claimAndWithdraw that will call the relevant internal functions, providing users with a more convenient way to claim and withdraw earnings in one step.

Recommendation 2: Same problem as as ID A.7. The `canClaimRevenue` modifier is calling the internal `_claimRevenue()` function and than it will call that function again inside. Consider using the same method as you use to solve A.7.

ID A.19 : **Status: Resolved**

Medium | Creating more than one set of share tokens is not possible in the contract's initial state

Present at: `constructor` @ L285

Description: In the constructor, `shareLimitNumber` is initialized to be equal to `startNumber`. This configuration permits only the creation of a single set of share tokens. If multiple users wish to create share tokens, the contract owner must first increase the limit using `increaseShareLimit()`.

Complete Analysis

Considering the owner should be a sufficiently distributed cold wallet multisig congress contract or vault, and considering the docs of the increaseShareLimit
Which detail its purpose as a more TBD future one, it seems contradictory to the implementation, as with current code, the owner would have to manage the contract continuously to re-set the shareLimit to enable more users to create shares, which seems highly impractical for production setting, and would result in very bad UX if users actually try to create share tokens and would not succeed and have to call out the owner/app/protocol support for help and wait for the multisig to advance the shareLimit each time.

Recommendation: Review whether this behavior is intentional. If not, consider initializing shareLimitNumber to be equal to `endNumber` instead, or to some other number which would enable normal contract operation for a feasible iteration of time, e.g. a week, a month etc..

ID A.20: **Status: Resolved**

Informational | **Missing Fields in Event Description**

Present at: `CreateShare` @ L228

Description: `startingMessage` and `startTime` fields do not exist in the @param description above the event.

Recommendation: Make sure you describe all of the event and function params.

Complete Analysis

ID A.21: **Status: Resolved**

Discussion | **Lack of mechanism for users to obtain, release, or transfer share tokens**

Present at: `depositRevenue` @ L528

Description: It is important to prevent potential frontrunning issue, where a user can frontrun a deposit revenue transaction, buy share tokens, claim the revenue right after and sell the share tokens

Recommendation: Consider implementing a time-lock mechanism for deposited revenue

Disclaimer:

DcentraLab Diligence (DD) has provided the code to the client as is and assumes no responsibility nor legal liability for any use client may do with the code. Any and all usage and/or deployment of the code provided by DcentraLab Diligence will be done solely by the client, at the sole discretion, responsibility, risk, and legal liability of the Client, and DD will not be held accountable or liable for any loss of funds, security exploits or incidents, or any other unintended or negative outcome that may occur in relation to the code provided by DD.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts DD to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model, or legal compliance.

This report and the provided code or services as part of the SOW pertaining to this report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should it be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. DD's position is that each company and individual are responsible for their own due diligence and continuous security. DD's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by DD are subject to dependencies and are under continuing development. You agree that your access and/or use, including but not limited to any services, code, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, DcentraLab Diligence (DD) HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, DD SPECIFICALLY DISCLAIMS

ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, DD MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT / VERIFICATION REPORT, WORK PRODUCT, CODE OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

WITHOUT LIMITATION TO THE DISCLAIMER HyperCycle Contracts FOREGOING, DD PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET THE CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR-FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER DD NOR ANY OF DD'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION, CODE OR CONTENT PROVIDED THROUGH THE SERVICE. DD WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT OR CODE, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, CODE, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS," AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN THE CUSTOMER AND THE THIRD-PARTY
OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS. THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO THE CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT DD'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST DD WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS. THE REPRESENTATIONS AND WARRANTIES OF DD CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF THE CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST DD WITH RESPECT TO SUCH
REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.
FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS, CODE, OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL,
REGULATORY, OR OTHER ADVICE.

dcentralab.com/diligence

DcentraLab Diligence

Provided By DcentraLab Diligence on December 03, 2023