Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

Instructions for Deploy:

(Allostasis DAO smart contracts)

- Token Allocator
- ssAllo Token
- Platforms Allocator

Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

Token Allocator Smart Contract:

This is a complex Ethereum smart contract written in Solidity. The purpose of this contract is to manage and distribute tokens based on stages of a project. Here's an overview of what it does:

1. It uses OpenZeppelin contracts for managing access control (AccessManaged), pausability (Pausable) and reentrancy guarding (ReentrancyGuard).

2. It defines an interface IERC20 for interacting with the ERC-20 token contract, which it assumes is at a certain address when the contract is deployed.

3. The TokenAllocator contract has several public variables and mappings that store data about various stages of the project, recipient addresses, locked amounts, etc.

4. It defines an enum Recipient to categorize different recipients of tokens (Users, Creators, Nodes, Dev Team, Treasury).

5. Several functions are defined for handling different phases or "stages" of a project. Each function transfers a specific amount of tokens to each recipient's address and handles locked amounts. The logic behind the distribution of these tokens might be complex depending on stage.

6. It emits several events, such as when new tokens are minted for a stage or when rewards are transferred to recipients.

7. Some functions (like `changeStage` and `mintStageTokens`) can only be called by the contract owner due to the use of the `restricted` modifier.

8. The contract uses the ReentrancyGuard from OpenZeppelin, which helps prevent re-entrancy attacks by making it impossible for a function to call itself again before its execution is finished.

Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

ALLOSTASIS

Follow the instructions to deploy TokenAllocator contract:

Deploy TokenomicsManager contract
- Give an address as "initialAdmin" to the input

Deploy "ALLOToken" contract
- Set TokenomicsManager address as "initialAuthority" to the input

Deploy "TokenAllocator" contract (This contract has two inputs)
- Set "initialAuthority" to the "TokenomicsManager" address
- Set "ALLOadr" to the "ALLOToken" address

Setting the "TokenAllocator" smart contract as a minter in "ALLOToken" :

Go to the tokenomics contract and call the "grantRole" function with the following parameters:

- Set "roleId" to 1
- Set "account to the "TokenAllocator" address
- Set "executionDelay" to 0

After this go and call "setTargetFunctionRole" with the following parameters:

- Set "target" to the "ALLOToken" address
- Set "selectors" to "kecake256('mint(address,uint256)')" which is 0x40c10f19
- Set roleId to 1

now initialAdmin address can call all the functions

Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

ssAllo Token Smart Contract:

This smart contract is responsible for minting ERC721 tokens (ssAllo NFTs) for each user, representing 90% of their rewards. During this period, users can participate in DAO voting.

After two awarding stages, users can burn their ssAllo NFTs to claim their staked tokens, along with the platform's profits earned over the staking period.

Follow the instructions to deploy ssAlloToken contract:

Deploy TokenomicsManager contract

- Give an address as "initialAdmin" to the input

Deploy "ALLOToken" contract

- Set TokenomicsManager address as "initialAuthority" to the input

Deploy "TokenAllocator" contract (This contract has two inputs)

- Set "initialAuthority" to the "TokenomicsManager" address
- Set "AlloAddr" to the "ALLOToken" address

Deploy "ssAlloToken" contract (This contract has three inputs)

- Set "initialAuthority" to the "TokenomicsManager" address
- Set “TokenAllocatorAddr” to the “TokenAllocator” address
- Set "AlloAddr" to the "ALLOToken" address

Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

Setting the "TokenAllocator" smart contract as a minter in "ALLOToken" :

Go to the tokenomics contract and call the "grantRole" function with the following parameters:

- Set "roleId" to 1
- Set "account to the "TokenAllocator" address
- Set "executionDelay" to the 0

After this go and call "setTargetFunctionRole" with the following parameters:

- Set "target" to the "ALLOToken" address
- Set "selectors" to "kecake256('mint(address,uint256)')" which is 0x40c10f19
- Set roleId to 1

Now Token Allocator admin should stage up and send reward to content and infrastructure's admin.

One of the content or infrastructure's admin should follow the below steps to stake Allo token for recipients: (In this case other users can stake their Allo token too)

- call "approve" function in AlloToken contract: set "spender" to the "ssAllotoken" address ; and set value to the number of needed to be approve.

Now admin or staker can stake AlloToken and mint a NFT and burn it after two stage of staking.

Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

Platforms Allocator Smart Contract:

In this smart contract, the Allostasis admin sets up the platform information and their admin addresses. This allows the platform admins to stage up their platforms, and the Allostasis admin can mint the Allo tokens as determined for them.

Follow the instructions to deploy Platform Allocator contract:

Deploy TokenomicsManager contract
- Give an address as "initialAdmin" to the input

Deploy "ALLOToken" contract
- Set TokenomicsManager address as "initialAuthority" to the input

Deploy "PlatformAllocator" contract (This contract has two inputs)
- Set "initialAuthority" to the "TokenomicsManager" address
- Set "ALLOadr" to the "ALLOToken" address

Defining a new role, the Platforms Allocator admin to add a new platform to the Allostasis ecosystem and mint tokens for them: (Be careful DO NOT REUSE role Ids)

Step one, defining role:
Function name: grantRole()
Parameters:
- roleId: 1
- account: The PlatformsAllocator address
- executionDelay: 0

Step two, assigning the function:
Function name: mint()
Parameters:
- Target: The AlloToken address
- Selectors: ["kecake256('mint(address,uint256)')"] which is 0x40c10f19
- roleId:1

Document name	Centeria	Date	24/Feb/2025
Author	M.Navid Jalalian	Approved by	

The Platforms Allocator admin has the authority to add a new platform to the Allostasis ecosystem to receive funds. To do this, the admin can call the registerPlatform() function with the following parameters:

- Name: platform's name
- Contract address: address of the platform's smart contract, which tokens will be transferred to
- Admin: address of the platform's admin, who can stage up the platform
- Number of stages: all the allocated tokens to this platform are not released immediately, and the number of stages that tokens are going to be released.
- Release percentage: an array of each stage percentage. PLEASE notice the last element in this array is for the decimal that is used for turning to int.

Once a new platform is registered, the Platforms Allocator admin can mint a portion of the new platform's tokens and transfer them to the registered address. To do this, the admin must call the mintPlatformsTokens() function with the platform ID of the new platform. This function automatically calculates the amount of tokens, mints them, and transfers them to the receiver address (the platform's smart contract address).

Each platform admin can stage up their platform and request additional funds. The platform admin can call the stageUpPlatform() function with the platform ID to increase the platform's stage.