Smart Contract White Paper RemitoHormiga

Introduction The RemitoHormigaNFT smart contract is a customized implementation of a non-fungible token (NFT) on the Ethereum blockchain. Below, I provide a technical description of the contract's key features and functions:

Imports and Extensions: The contract imports interfaces and contracts from the OpenZeppelin library, which is a popular and proven library for secure smart contract implementations on Ethereum. The contract inherits from ERC721Enumerable (an NFT token standard that allows token enumeration) and ReentrancyGuard (a modifier to prevent reentrancy attacks).

Structure and Mappings: Remito: A structure defining a "Remito" object with related fields such as valorDeclarado (declared value), recompensa (reward), tiempoLimite (deadline), liberatingWallet, among others. remitos: A mapping that associates a token ID with its corresponding Remito object. originalMinters: A mapping that associates a token ID with the address that originally minted it.

Contract Variables: hormigaToken: Represents the ERC20 token used for rewards and declared values. FEVPoolBalance and FERPoolBalance: Represent the balances of pools for declared value and reward, respectively.

Events: Created: Emitted when a new NFT token is minted. Delivered: Emitted when a Remito is delivered. RefundClaimed: Emitted when a refund is claimed.

Functions: constructor: Initializes the contract, setting the name and symbol of the NFT token and configuring the address of the hormigaToken. mint: Function to mint a new NFT token and associate it with a Remito object. deliver: Function that allows the liberatingWallet address to confirm the delivery of the Remito and receive the declared value and reward. claimRefund: Function that allows the liberatingWallet address to claim a refund if the Remito has not been delivered after the tiempoLimite. tokenURI: Overrides the standard ERC721 function to return the URI of the image associated with the NFT. walletOfOwner:

Returns a list of token IDs owned by a specific address. getNFTs: Filters and returns a list of token IDs based on various filtering criteria.

Modifiers: onlyLiberatingWallet: Restricts access to functions only for the address specified as liberatingWallet for a given token ID.

This smart contract implementation is quite detailed and relates to a specific use case of Remitos (shipments) associated with NFT tokens. Remitos can be "delivered" or "refunded" based on certain conditions, and the contract uses an external ERC20 token to manage payments related to these processes.

Use Case Example: Step 1: Remito Creation Actor: Sender The sender wants to send an item to the receiver and creates a Remito, specifying a declared value of, for example, 30,000 HTC for the item and a reward of 5,000 HTC for the carrier who will make the delivery. The sender calls mintRemito, defining relevant parameters, including the liberatingWallet address, which in this case will be that of the receiver. The sender transfers 30,000 HTC to the FEV contract and 5,000 HTC to the FER contract. An NFT representing the Remito is created, and the sender becomes the first owner of this token. Step 2: Item Transportation Actor: Carrier The sender physically hands over the item to the carrier and transfers the Remito NFT, giving the carrier the right to receive the reward once the delivery is completed. Step 3: Item Delivery and Reward Actor: Receiver (Liberating Wallet) When the item arrives at its destination, the receiver (who may be the liberatingWallet) calls the deliver function, passing the corresponding token ID. The contract verifies that the call has been made from the liberatingWallet and that the Remito has not been marked as delivered previously. If everything is in order, the contract transfers 30,000 HTC and 5,000 HTC from FEV and FER, respectively, to the current holder of the NFT (the carrier), marking the Remito as delivered. Alternative Case: Non-Delivery and Refund Actor: Receiver (Liberating Wallet) If the item is not delivered before the deadline, the address specified as the liberatingWallet can call claimRefund to receive the declared value of the item and the reward, which are transferred from FEV and FER, respectively, to the liberatingWallet address.

Conclusion In this system, the Remito represented by an NFT acts as a binding contract that ensures the carrier will be rewarded once the delivery is satisfactorily completed, provided it occurs before the defined deadline. It also guarantees that in case of non-delivery, the receiver can recover the declared value of the item and the reward.