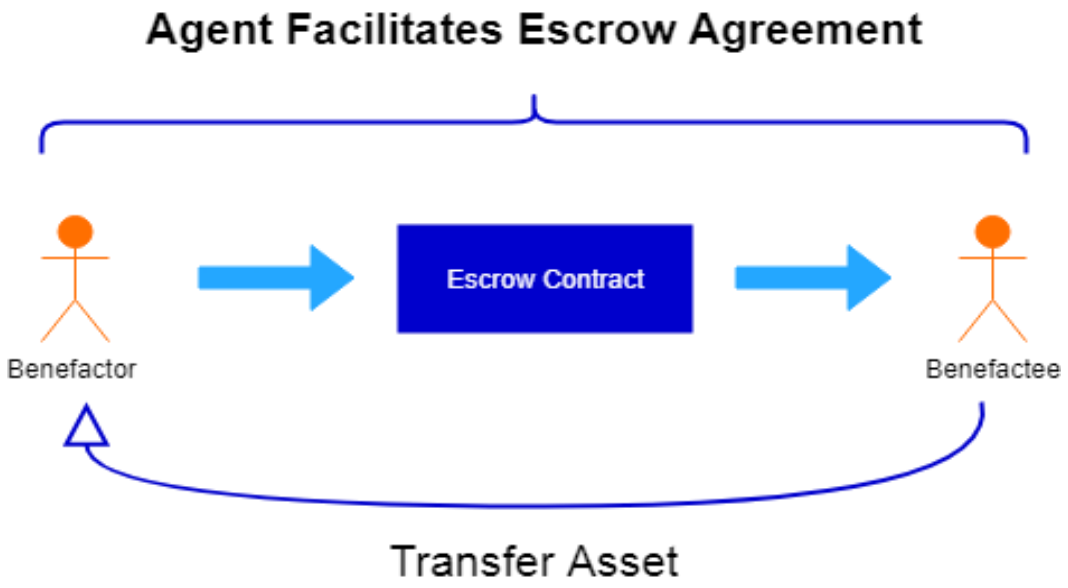
AETHER

What is an Escrow Contract?

An escrow agreement is a contract that specifies the terms and conditions of the parties involved, as well as their respective responsibilities. Escrow agreements often require an independent third party, known as an escrow agent, who holds a valuable asset until the contract's specified conditions are satisfied. They should, however, fully define the terms for all parties involved.



How does Escrowx.sol work?

We create a contract called Escrowx that inherits from another contract ReentracyGuard which helps us secure our contract from Reentrancy attacks.

Constructor()

The contract starts with a constructor that defines the agent aka the owner and the rate (commission) which he will get for initiating the contract.

DepositTokens()

Then comes the benefactor (msg.sender) who will call the function depositTokens() where he will deposit a certain amount of ETH and specify the address of the benefactee. The amount (msg.value) will be stored in the contract using contractData[id].escrowBalance = amount;. We also save all the detail like bool sign1, sign2, isActive to specify if the contract is still active in a struct.

Sign()

To be able to withdraw the tokens, benefactor should sign with the right number and the benefactee should also sign with the same number of the benefactor so we can fetch the right data from the array using a mapping. We defined the sign1() and sign2() to be false in the function depositTokens() so we can change them to true. If the integer Conid of sign1 and sign2() are the same as the id = 1, we can update them to true.

ReleaseTokens()

If benefactor and benefactee signed properly, anyone of them can withdraw the tokens to the benefactee's address and then deactivate the contract.

Mediate()

If something went wrong, the owner can send the tokens back to the benefactor or send the tokens to the benefactee and then deactivate the contract.

getRate()

Any address involved can return the rate that the owner will get.

Viewtotal()

Returns the balance of the address.

Unit Testing

Constructor()

- Only the owner of the contract can initiate Escrowx.sol and can specify the rate.
- The rate cannot be more than 100% or it will throw an error.
- The rate cannot be a string.

DepositTokens()

- The benefactor can be the owner too but he can't be the benefactor and benefactee at the same time thus he can't send tokens to himself.
- The benefactor can't send 0 tokens.
- The benefactor should input a number not a string.
- The benefactor can't send more tokens than he has.

Sign()

- The benefactor and benefactee can only sign with a uint Conid of 1.
- The signer should only be the benefactor or the benefactee.
- The benefactor or the benefactee cannot sign twice.

ReleaseTokens()

- Only the benefactor or the benefactee can release the tokens after they both sign.
- They can release the tokens with a uint Cid of 1.

mediate()

- Can't invoke the mediate function after the releaseTokens() is called and vice versa.
- Only the owner can use the mediate function.