# 166.https://stackoverflow.com/questions/72601947/solidity-call-a-function-of-nft-contract-from-marketplace-contract-and-only-ma

**T:**Solidity: call a function of NFT contract from Marketplace contract, and only Marketplace has access to call it

**Q:**I'am doing a NFT marketplace, and i have to implement that everytime a NFT is sold the images shuffle.The Marketplace has the functionallity to buy the NFTs so we have referenciate tha NFTs smart contract. When the function buy() is called, it calls the transfer() and suffle() from NFT smart contract. But i want only the marketplace to call the suffle(), so I decided to add a require that checks if the address that is calling is equal to the Marketplace address.The problem is that for deploying the Marketplace i need the NFT address and the same for deployig the NFT contract i need the Marketplace address.How can i solve this? Is a problem of the logic of the smart contract or there is a way to solve it in this way?Thank you!!  
  
Smart Contracts Logic Diagram

2 **Answer**

**A1:**You can use a setter function to set the dependency address after deployment.  
  
Example:  
  
contract NFT { address marketplace; function setMarketplace(address \_marketplace) public onlyOwner { marketplace = \_marketplace; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
contract Marketplace { address nft; function setNft(address \_nft) public onlyOwner { nft = \_nft; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**A2:**I wrote a simple framework that should help you. :)  
  
 //SPDX-License-Identifier: UNLICENSEDpragma solidity 0.6.12;contract NFT { //Ensure that only the owner can call important functions address owner; constructor() public { owner = msg.sender; } mapping (address => bool) public \_onlyIfMarketplace; function mint() public { require(owner == msg.sender); } function transfer() public { } function shuffle() public { require(\_onlyIfMarketplace[msg.sender] == true); } //You can always add an address that can call this function, //and you can also write another one to remove the address which can call this function function setMarketplaceContract(address MarketplaceContract) public { require(owner == msg.sender); \_onlyIfMarketplace[MarketplaceContract] = true; }}contract Marketplace { //Call NFT contract NFT nftContract; //Ensure that only the owner can call important functions address owner; constructor(address nftAddress) public { owner = msg.sender; nftContract = NFT(nftAddress); } // You can change the NFT contract you want to call at any time function changeNFTAddress(address newNFTAddress) public { require(owner == msg.sender); nftContract = NFT(newNFTAddress); } //Call the function of NFT contract function buy() public { nftContract.transfer(); nftContract.shuffle(); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Is there any way that the marketplace Smart Contract asigns it self to the \_onlyMarketplace? Because in the solution you gave me, the NFT owner could change the address to another one and run the shuffle withount buying the NFT in the marketplace. I don't know if this implementation could bring some security issues.