# Binding Royaltys for ERC721

by granola

### On Chain Royalties

By owning an NFT from a collection, one is bound to send ETH to transfer an NFT

All NFT transfers require ETH, breaking ERC721 standards to pay artists secondary royalties

### On Chain Royalties

Opensea breaks

Blur breaks

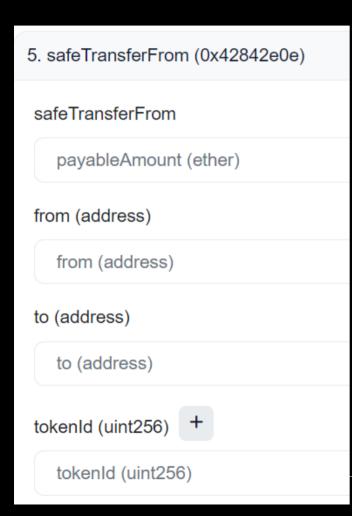
Any contract that does not use a payable version of safeTransferFrom breaks therefore enforcing royalties

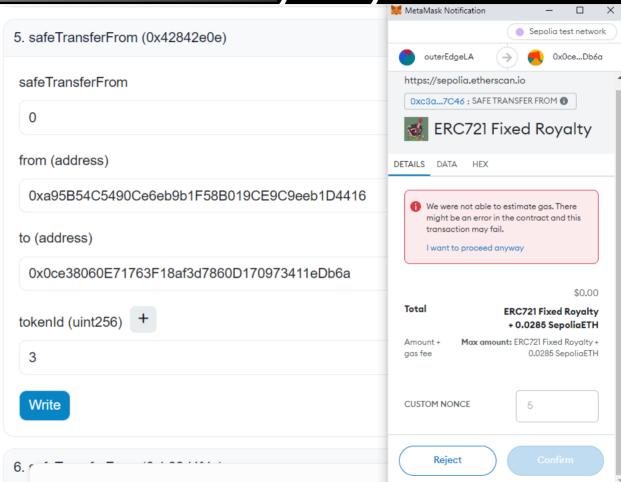
Disallowing Wrapping of NFTs stop subversion of royalty payments

## On Chain Royalties (solidity)

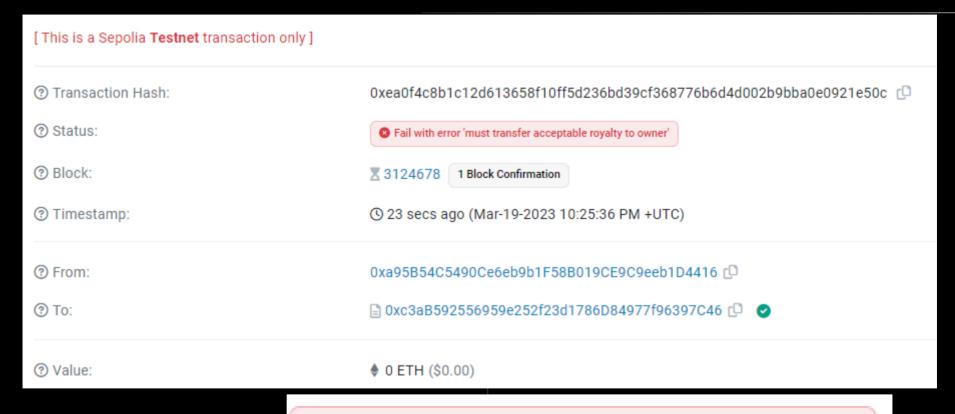
```
* Override transfer function to enforce fixed unit royalty
function safeTransferFrom(
    address from.
   address to.
   uint256 tokenId,
    bytes memory data
 public payable override {
    require( isApprovedOrOwner( msgSender(), tokenId), "ERC721: transfer caller is not owner nor approved");
   require(!to.isContract(), "cant send to contract");
    require(msg.value >= royaltyPrice, "must transfer acceptable royalty to owner");
    (bool success, ) = owner().call{value: msg.value}("");
    require(success, "Transfer failed.");
    safeTransfer(from, to, tokenId, data);
```

#### <u>Fail to Transfer – No Royalty Offered</u>



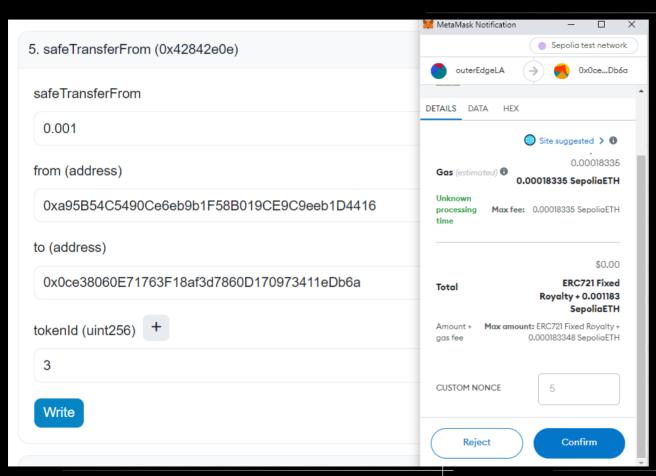


# Fail to Transfer – No Royalty Offered



Fail with error 'must transfer acceptable royalty to owner'

# <u>Transfer Success – Royalty Paid</u>



### <u> Transfer Success – Royalty Paid</u>

