Token Contract

//SPDX-License-Identifier: MIT

pragma solidity ^0.4.24;

// Token Contract

contract Token {

function totalSupply() public constant returns (uint256 supply) {}

function balanceOf(address \_owner) constant returns (uint256 balance) {}

function transfer(address \_to, uint256 \_value) returns (bool success) {}

function transferFrom (address \_from, address \_to, uint256 \_value) returns (bool success) {}

function approve(address \_spender, uint256 \_value) returns (bool success) {}

function allowance (address \_owner, address \_spender) constant returns (uint256 remaining) {}

// indexed keyword for searching logged events

event Transfer (address indexed \_from, address indexed \_to, uint256 \_value);

event Approval (address indexed \_owner, address indexed \_spender, uint256 \_value);

}

contract CT7Token is Token {

// defining Data Types

string public name;

uint8 public decimals;

string public symbol;

uint256 public unitsOneEthCanBuy;

uint256 public totalEthInWei;

address public fundsWallet;

mapping (address => uint256) balances;

mapping (address => mapping (address => uint256)) allowed; // amount of unit allowed by address1 to be used by address 2

uint256 public totalSupply;

// Constructor

// 1 ETH = 10^18 Weis

// 1 CT7 = 10^18

// Balances can be equal or less than totalSupply

constructor() public {

balances[msg.sender] = 1000000000000000000;

totalSupply = 1000000000000000000;

name = "CT7Token";

decimals = 18;

symbol = "CT7";

unitsOneEthCanBuy = 10;

fundsWallet = msg.sender;

}

// Main Function with Safe Math

function transfer(address \_to, uint256 \_value) returns (bool success) {

if (balances[msg.sender] >= \_value && balances[\_to] + \_value > balances [\_to]) {

balances[msg.sender] -= \_value;

balances[\_to] += \_value;

emit Transfer(msg.sender, \_to, \_value);

return true;

}

else {

return false;

}

}

function transferFrom(address \_from, address \_to, uint256 \_value) public returns (bool success){

if (balances[\_from] >= \_value && allowed[\_from][msg.sender]>= \_value && balances[\_to]+ \_value > balances [\_to]) {

balances[\_to] += \_value;

balances[\_from] -= \_value;

allowed[\_from][msg.sender] -= \_value;

emit Transfer (\_from, \_to, \_value);

return true;

}

else {

return false;

}

}

// constant is similar to view in the older version of Solidity

function balanceOf (address \_owner) constant returns (uint256 balance) {

return balances[\_owner];

}

function approve (address \_spender, uint256 \_value) returns (bool success) {

allowed[msg.sender][\_spender] = \_value;

emit Approval(msg.sender, \_spender, \_value);

return true;

}

//fallback function

function() payable{

totalEthInWei = totalEthInWei + msg.value;

uint256 amount = msg.value \* unitsOneEthCanBuy;

require(balances[fundsWallet] >= amount);

balances[fundsWallet] = balances[fundsWallet] - amount;

balances[msg.sender] = balances[msg.sender] + amount;

emit Transfer (fundsWallet, msg.sender, amount);

fundsWallet.transfer(msg.value);

}

}

Transaction:

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Contract: 0x83fF719287C49cFC50BD02a01c3EDa4bb7da41f0

User 1: 0x83fF719287C49cFC50BD02a01c3EDa4bb7da41f0

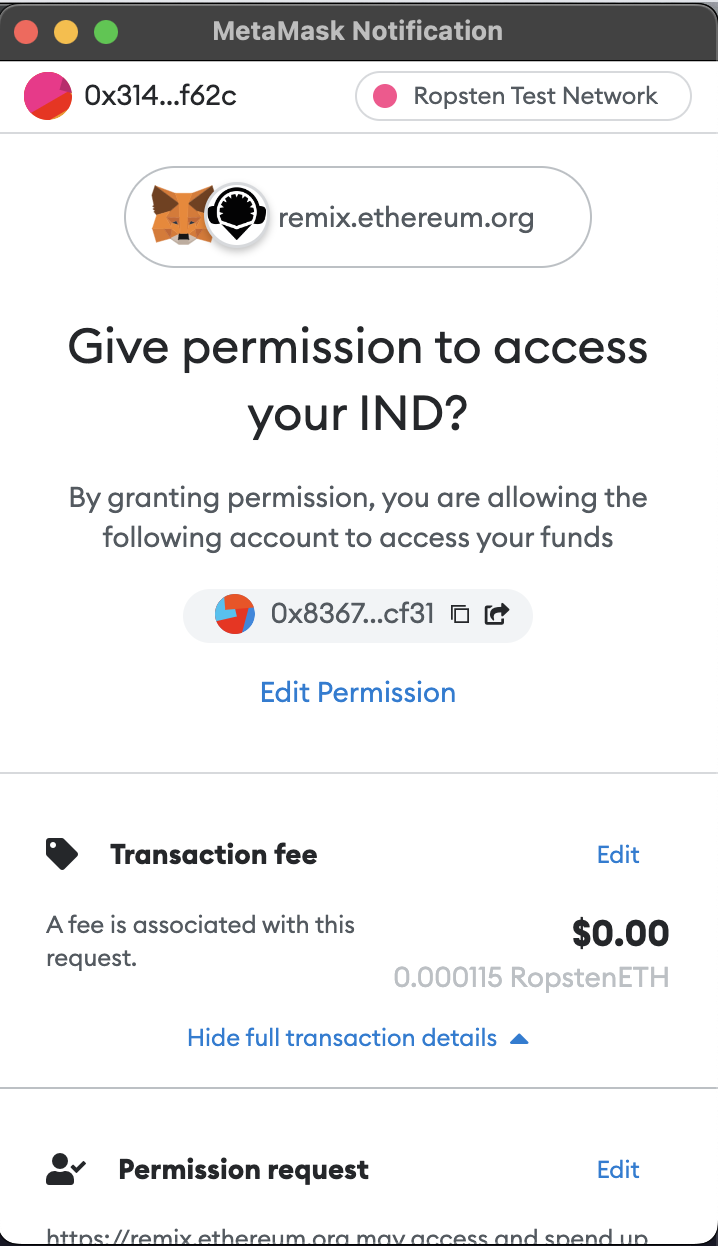
User 2: 0x83675E0cC7f1f55cEFeEb0E794eFD45EfFc3cf31

User 3: 0x314aF8A911e39CA773F0ff9A94a2B4FcfE83f62C

Graphical user interface, text, application

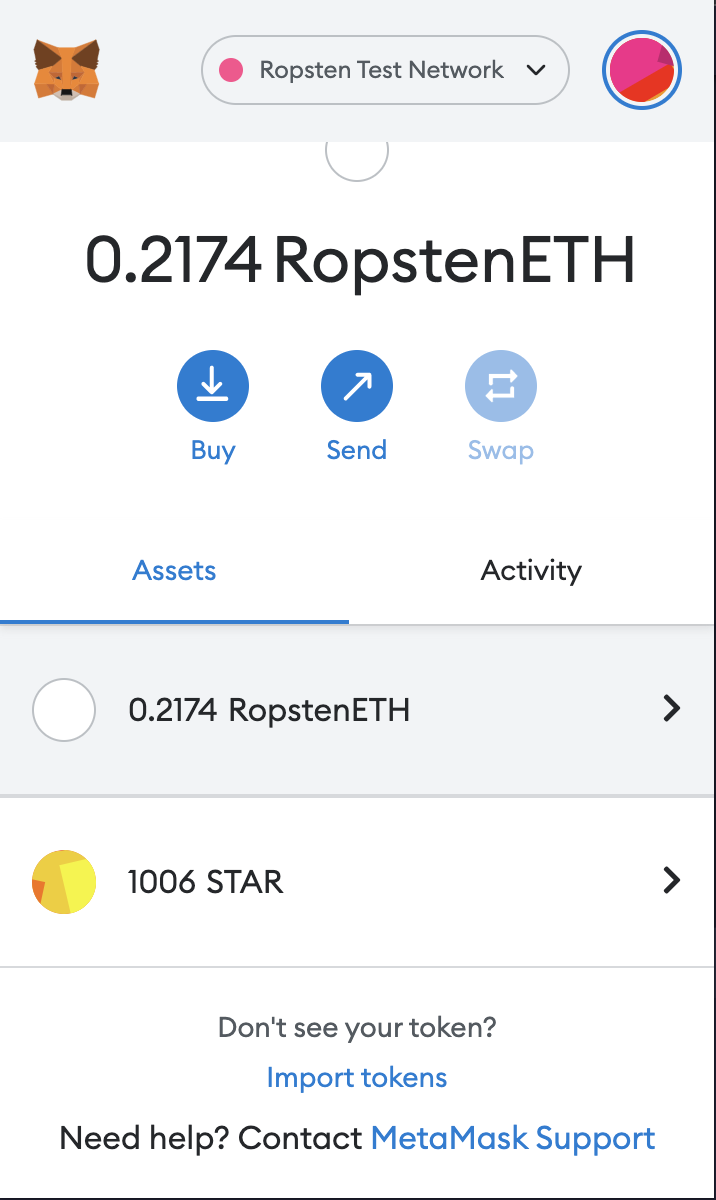
Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

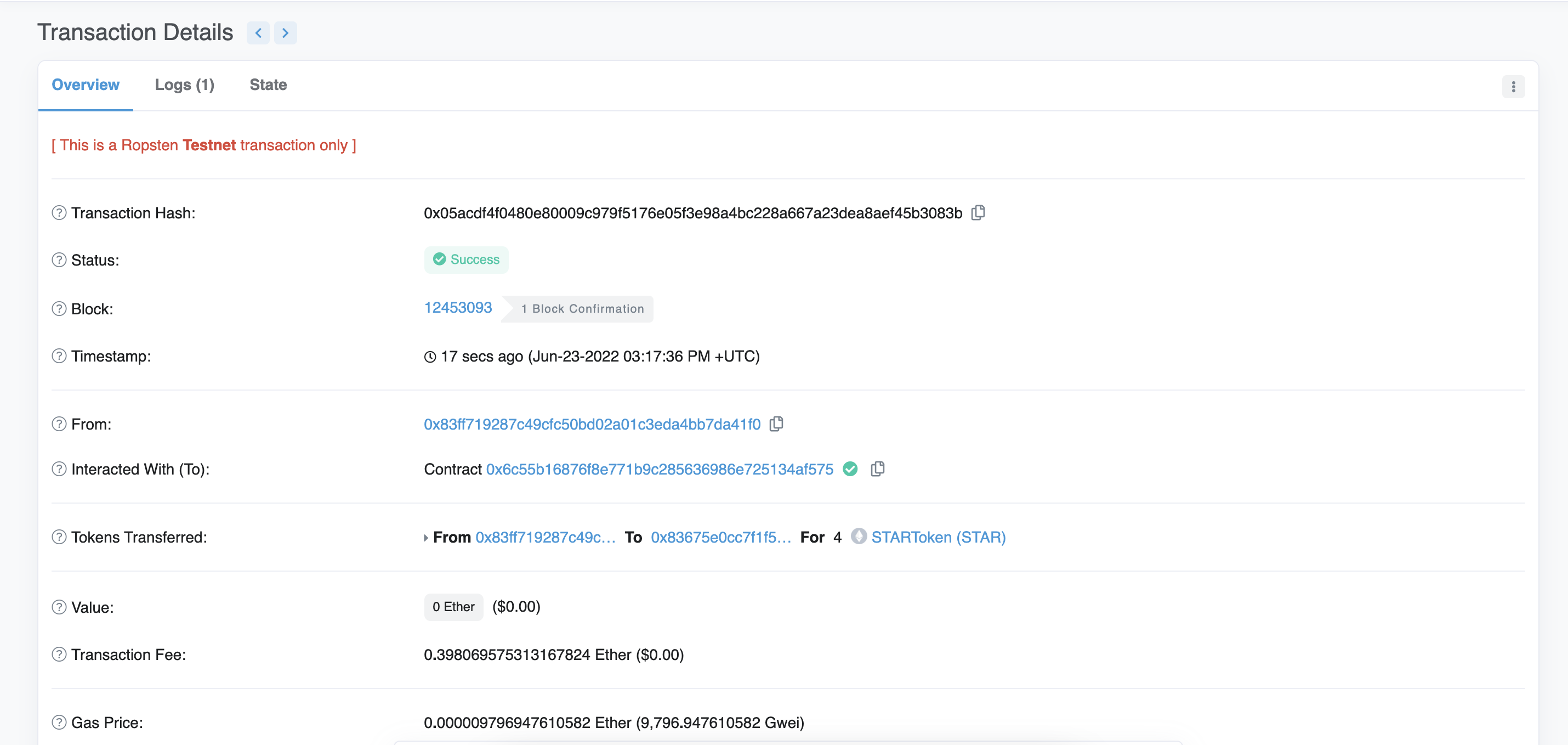
Description automatically generated

Additional Testing with STAR Tokens:

 Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated 

Graphical user interface

Description automatically generated