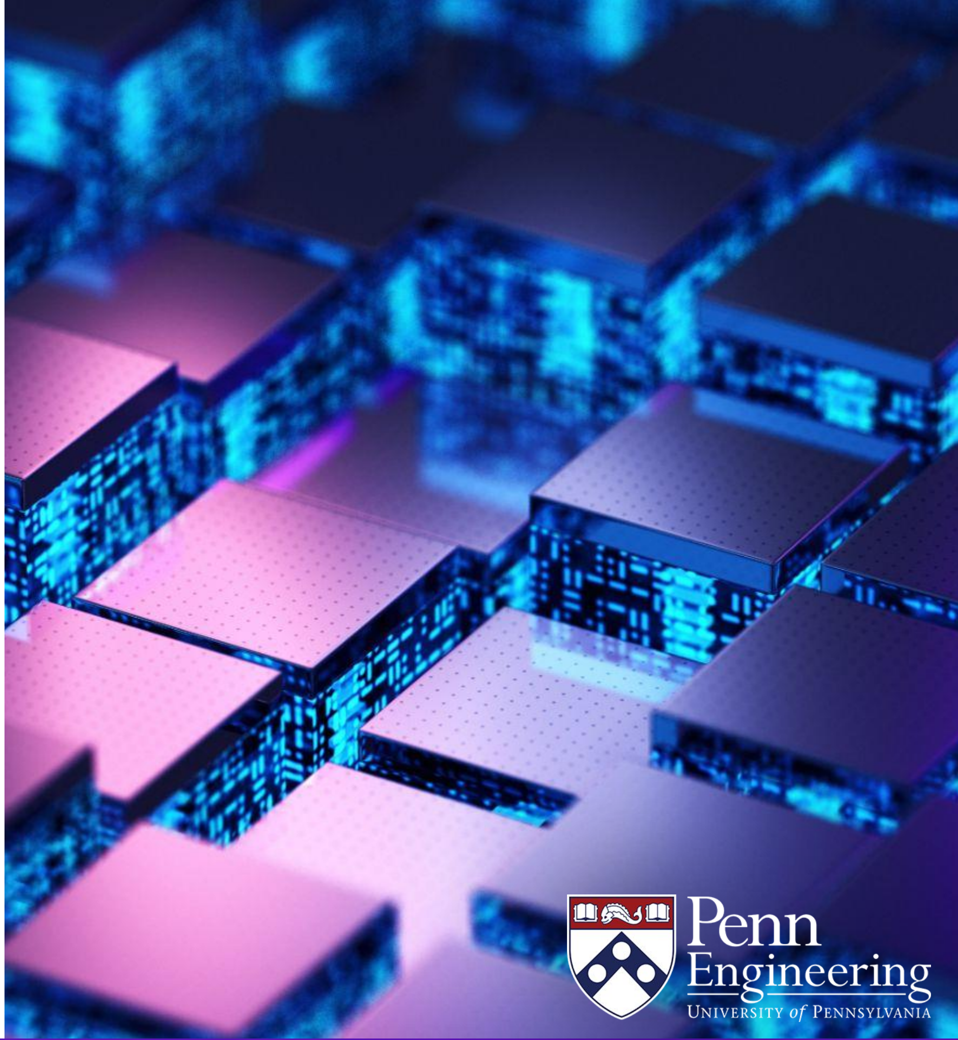
EAS 5830: BLOCKCHAINS

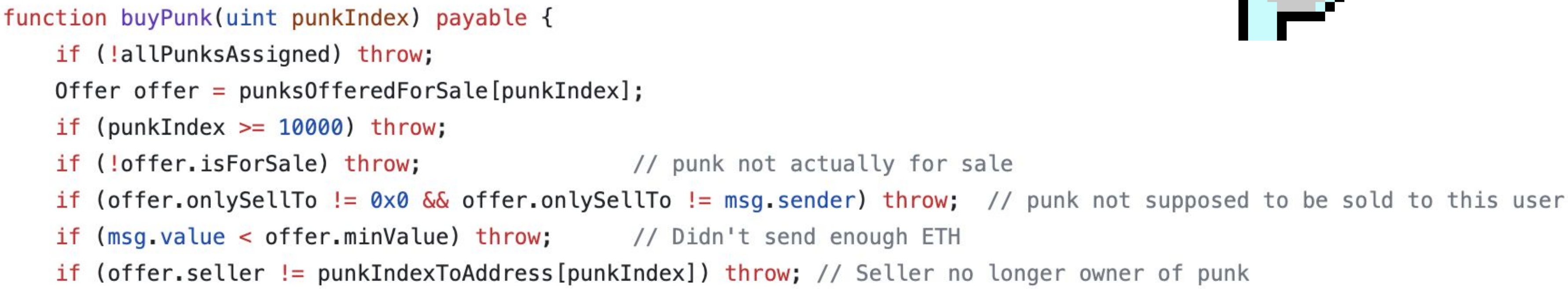
WETH

Professor Brett Hemenway Falk

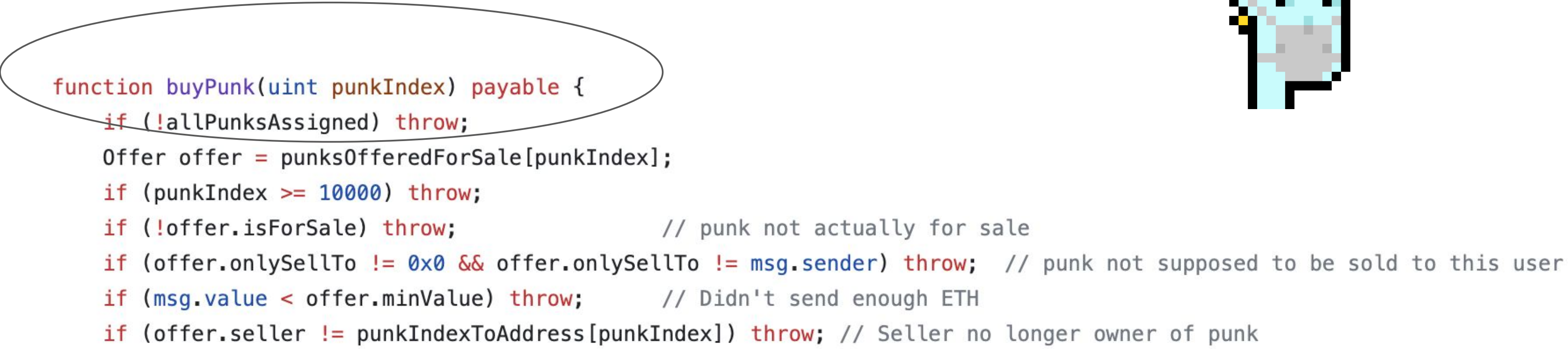


Wrapped Ether

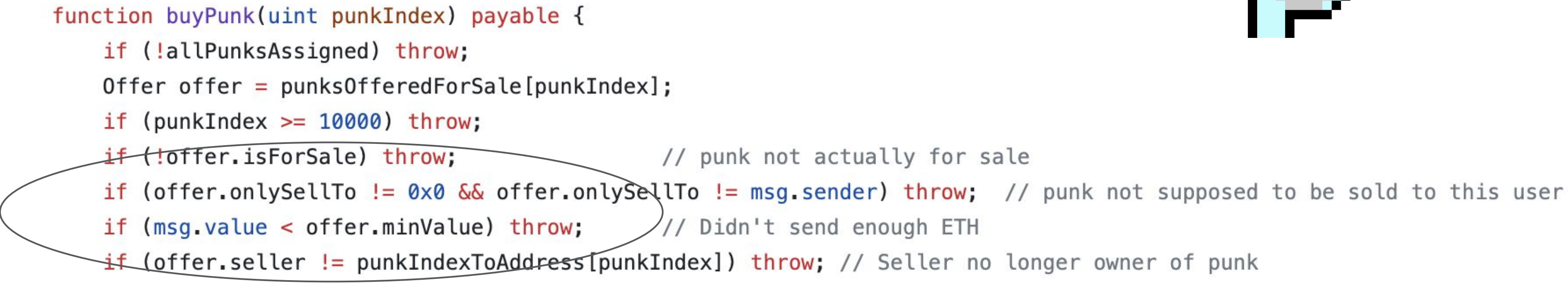
- Problem:
  - ETH is transferred differently than ERC-20s
- Solution
  - Make an ERC-20 version of ETH



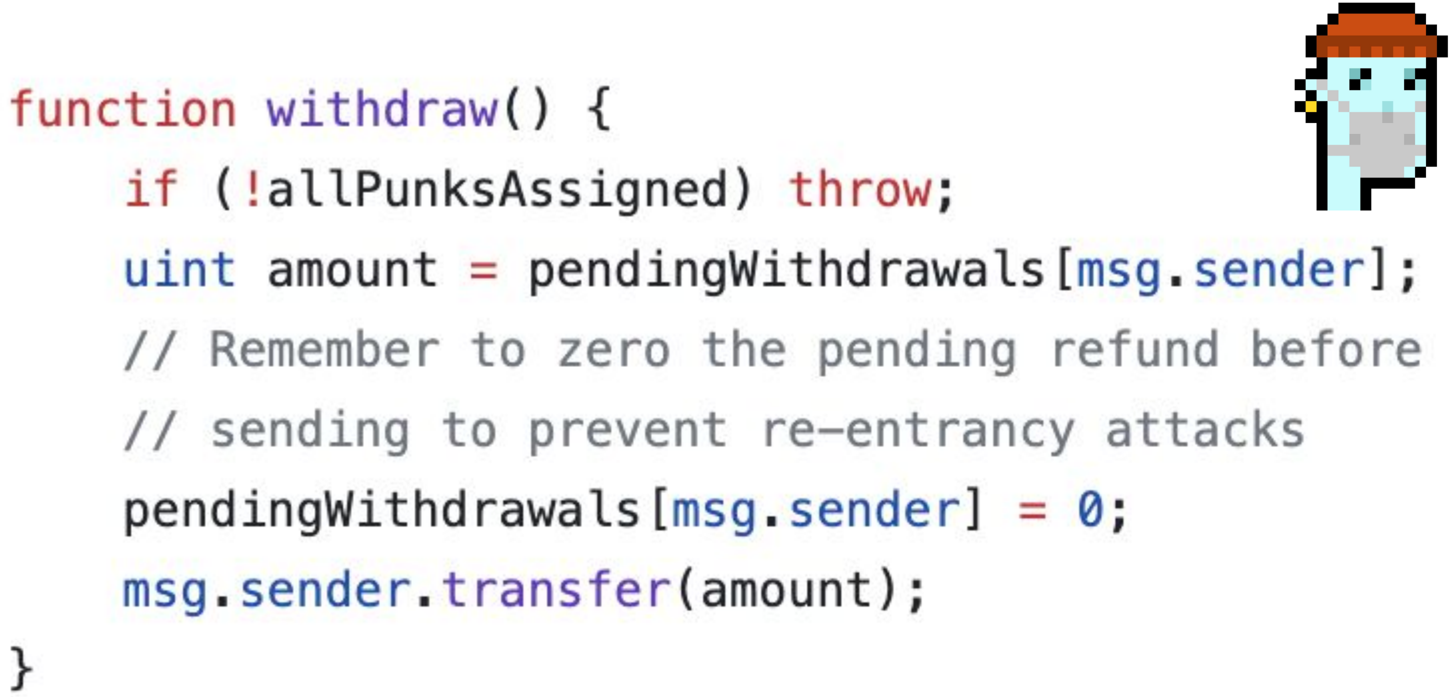
```
function buyPunk(uint punkIndex) payable {
    if (!allPunksAssigned) throw;
    Offer offer = punksOfferedForSale[punkIndex];
    if (punkIndex >= 10000) throw;
    if (!offer.isForSale) throw;                // punk not actually for sale
    if (offer.onlySellTo != 0x0 && offer.onlySellTo != msg.sender) throw;  // punk not supposed to be sold to this user
    if (msg.value < offer.minValue) throw;      // Didn't send enough ETH
    if (offer.seller != punkIndexToAddress[punkIndex]) throw; // Seller no longer owner of punk
```



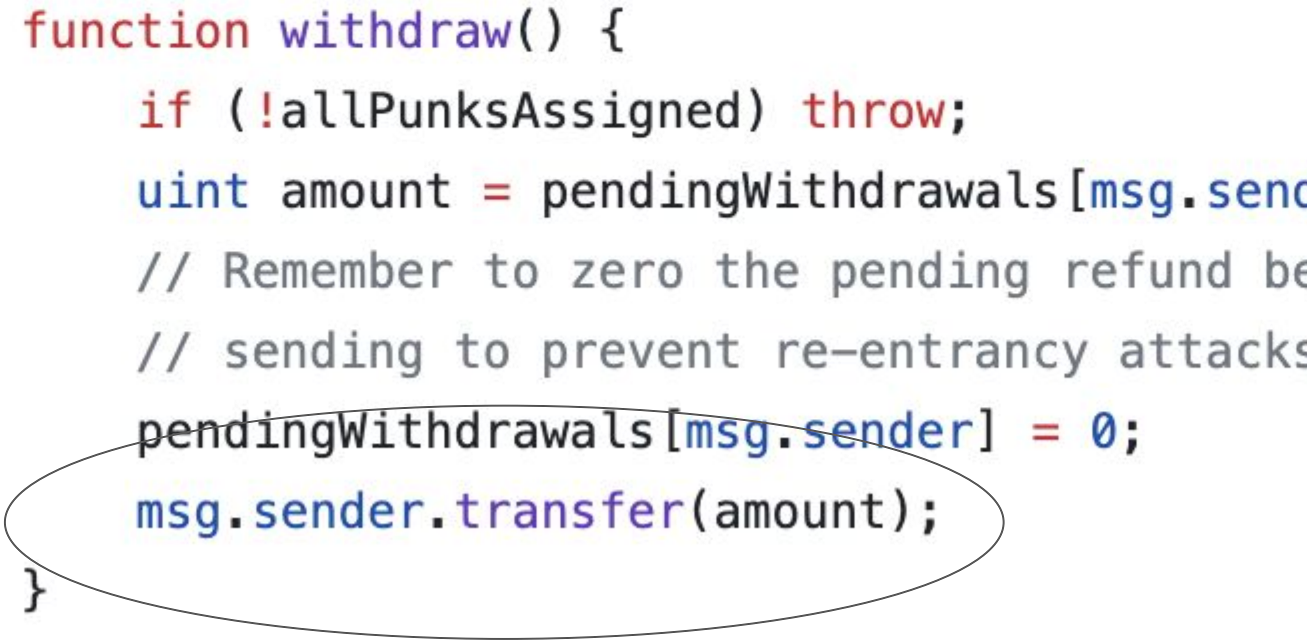
```
function buyPunk(uint punkIndex) payable {
    if (!allPunksAssigned) throw;
    Offer offer = punksOfferedForSale[punkIndex];
    if (punkIndex >= 10000) throw;
    if (!offer.isForSale) throw;                // punk not actually for sale
    if (offer.onlySellTo != 0x0 && offer.onlySellTo != msg.sender) throw;  // punk not supposed to be sold to this user
    if (msg.value < offer.minValue) throw;      // Didn't send enough ETH
    if (offer.seller != punkIndexToAddress[punkIndex]) throw; // Seller no longer owner of punk
```



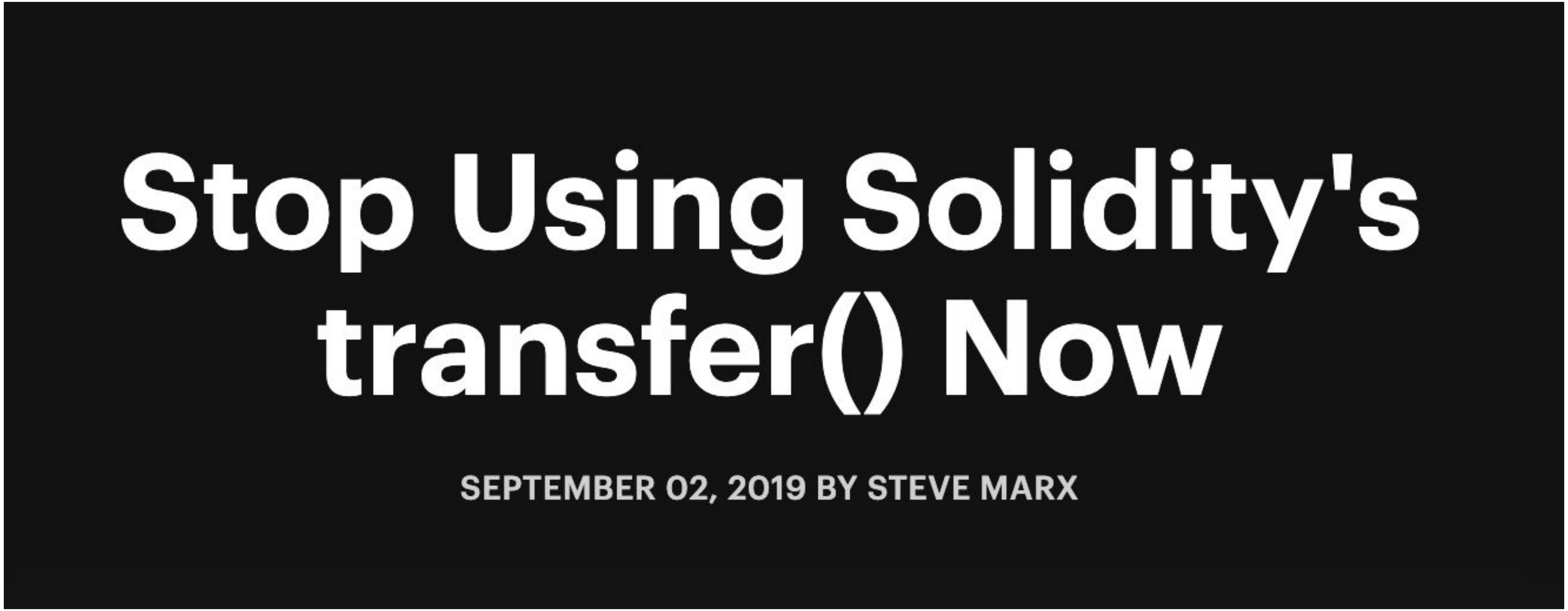
```
function buyPunk(uint punkIndex) payable {
    if (!allPunksAssigned) throw;
    Offer offer = punksOfferedForSale[punkIndex];
    if (punkIndex >= 10000) throw;
    if (!offer.isForSale) throw;                // punk not actually for sale
    if (offer.onlySellTo != 0x0 && offer.onlySellTo != msg.sender) throw;  // punk not supposed to be sold to this user
    if (msg.value < offer.minValue) throw;      // Didn't send enough ETH
    if (offer.seller != punkIndexToAddress[punkIndex]) throw; // Seller no longer owner of punk
```



```
function withdraw() {
    if (!allPunksAssigned) throw;
    uint amount = pendingWithdrawals[msg.sender];
    // Remember to zero the pending refund before
    // sending to prevent re-entrancy attacks
    pendingWithdrawals[msg.sender] = 0;
    msg.sender.transfer(amount);
}
```



```
function withdraw() {
    if (!allPunksAssigned) throw;
    uint amount = pendingWithdrawals[msg.sender];
    // Remember to zero the pending refund before
    // sending to prevent re-entrancy attacks
    pendingWithdrawals[msg.sender] = 0;
    msg.sender.transfer(amount);
}
```



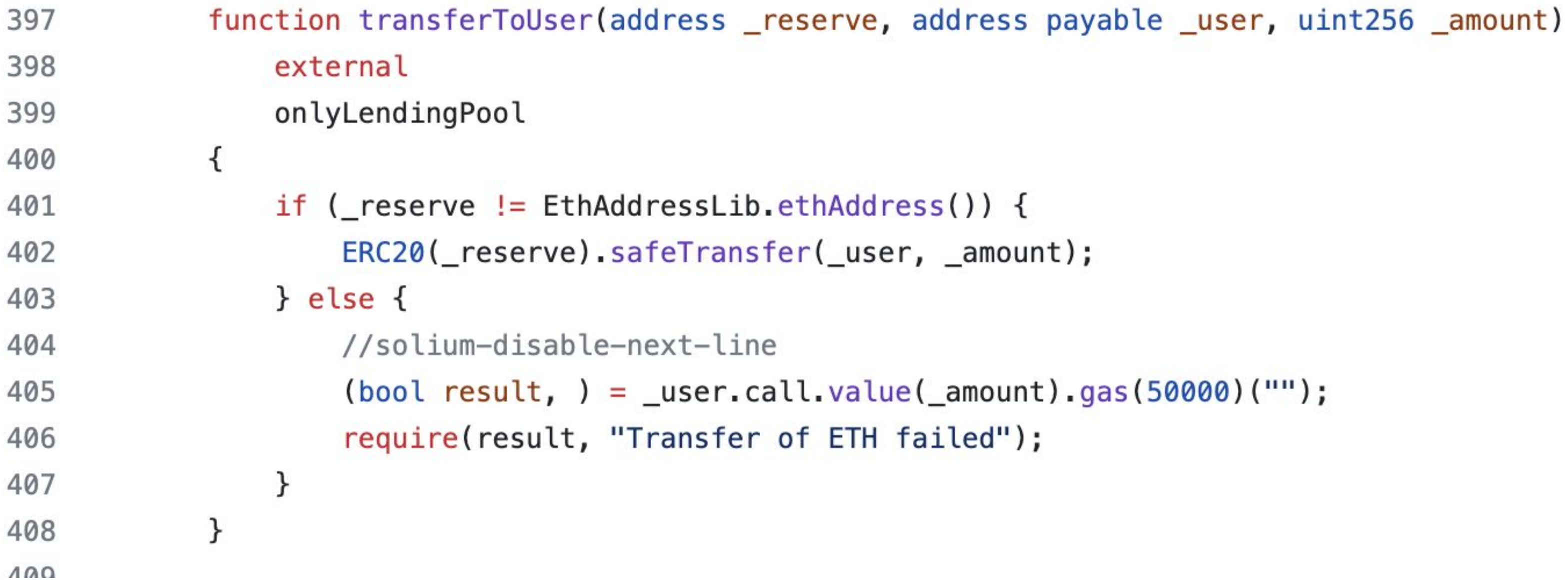
Stop Using Solidity's transfer() Now

SEPTEMBER 02, 2019 BY STEVE MARX

Transferring ETH

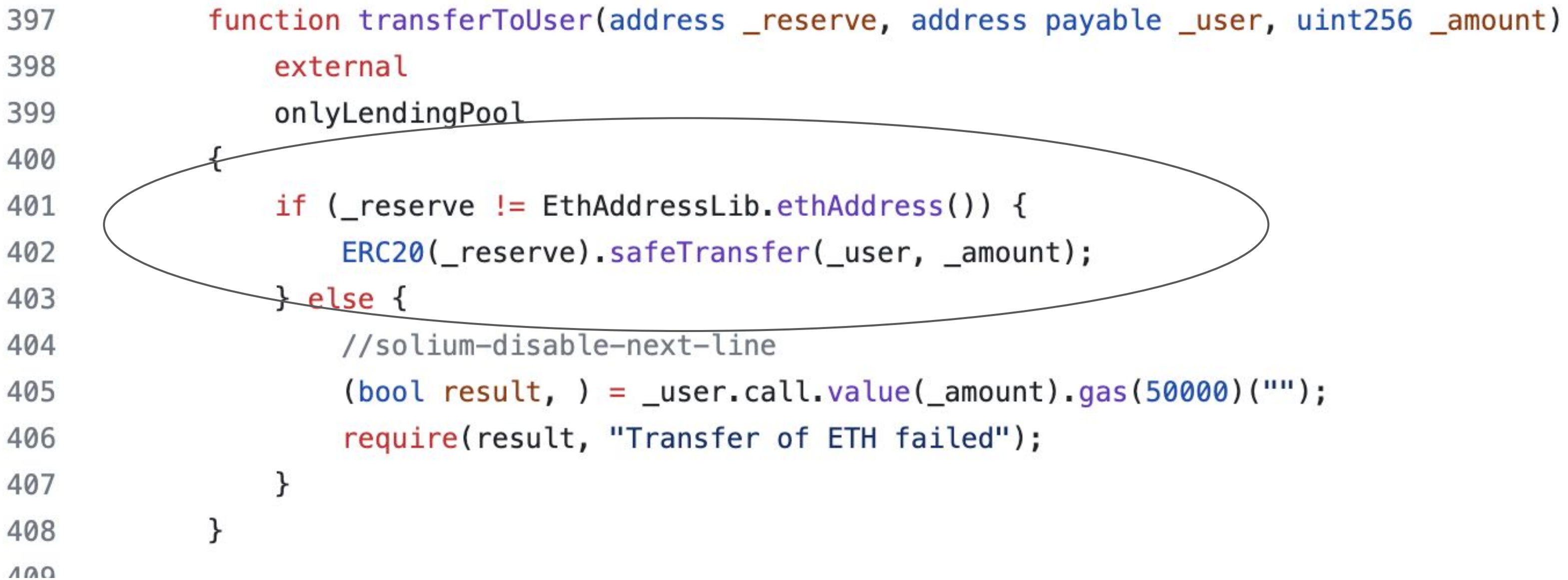
- `receivingAddress.transfer(amount);`
  - Reverts on fail
  - Only sends 2300 gas
- `receivingAddress.send(amount);`
  - Returns false on fail
  - Only sends 2300 gas
- `(bool success, bytes memory data) = receivingAddress.call{value: amount}("");`
  - Allows for re-entrancy
  - Transfer all gas to receiving contract

ERC20s transfer differently



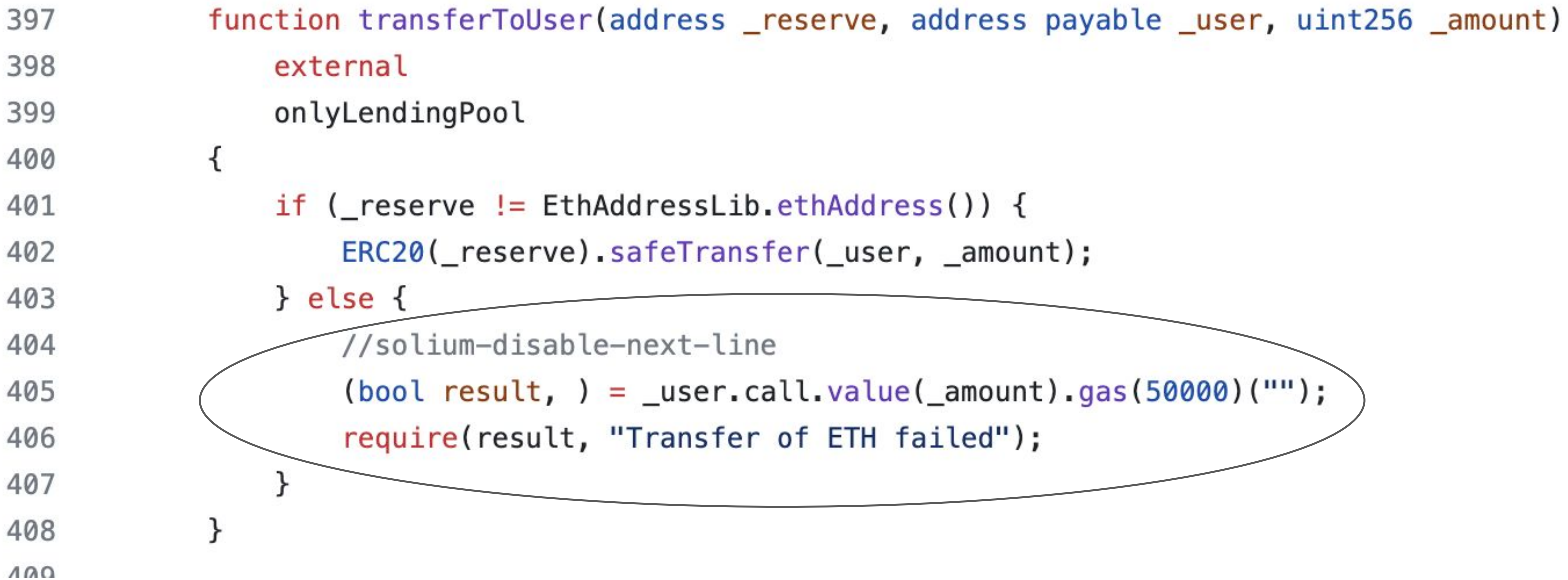
```
function transferToUser(address _reserve, address payable _user, uint256 _amount)
    external
    onlyLendingPool
{
    if (_reserve != EthAddressLib.ethAddress()) {
        ERC20(_reserve).safeTransfer(_user, _amount);
    } else {
        //solium-disable-next-line
        (bool result, ) = _user.call.value(_amount).gas(50000)("");
        require(result, "Transfer of ETH failed");
    }
}
```

ERC20s transfer differently



```
    function transferToUser(address _reserve, address payable _user, uint256 _amount)
        external
        onlyLendingPool
    {
        if (_reserve != EthAddressLib.ethAddress()) {
            ERC20(_reserve).safeTransfer(_user, _amount);
        } else {
            //solium-disable-next-line
            (bool result, ) = _user.call.value(_amount).gas(50000)("");
            require(result, "Transfer of ETH failed");
        }
    }
```

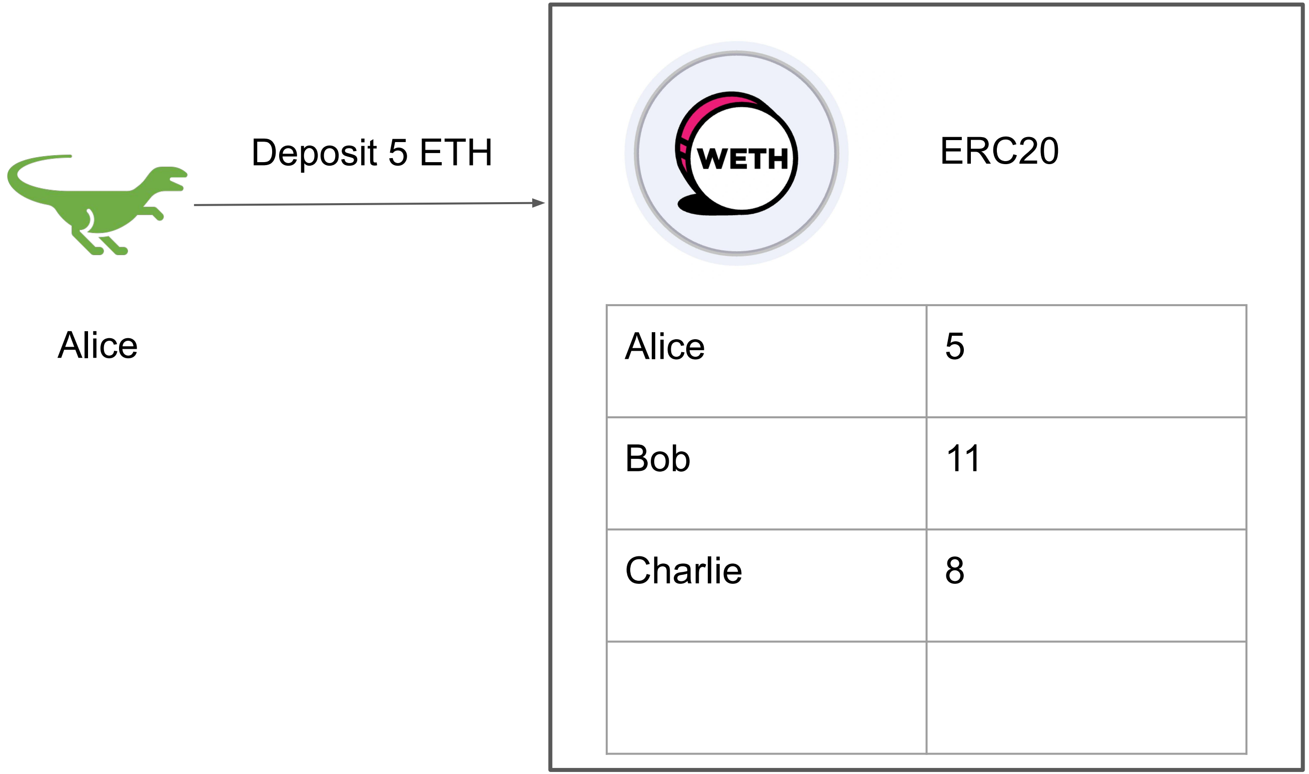
ERC20s transfer differently



```
    function transferToUser(address _reserve, address payable _user, uint256 _amount)
        external
        onlyLendingPool
    {
        if (_reserve != EthAddressLib.ethAddress()) {
            ERC20(_reserve).safeTransfer(_user, _amount);
        } else {
            //solium-disable-next-line
            (bool result, ) = _user.call.value(_amount).gas(50000)("");
            require(result, "Transfer of ETH failed");
        }
    }
```

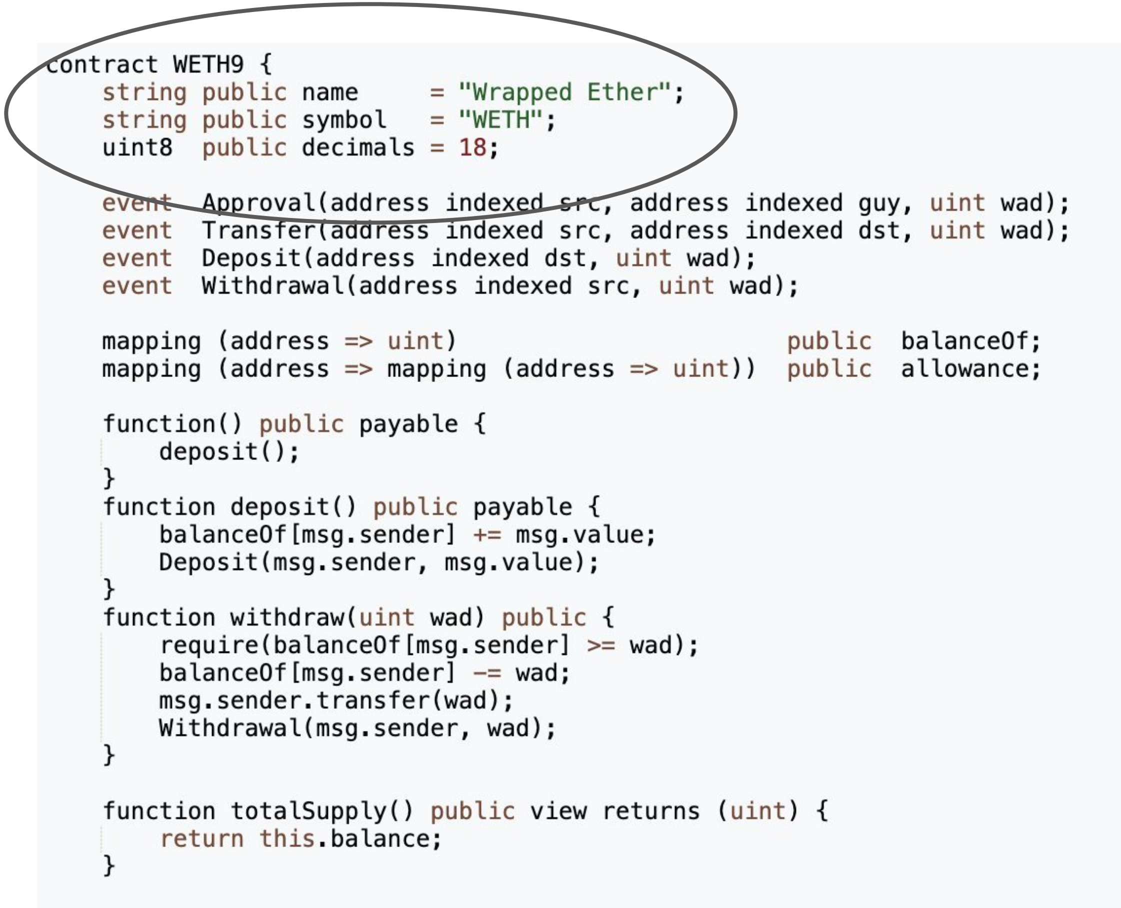
Handling ETH and ERC-20s

- Uniswap v1 only allowed ETH ↔ ERC-20
  - No ERC-20 ↔ ERC-20 trades
- Uniswap v2 only allows ERC-20 ↔ ERC-20
  - No ETH on the platform



Alice	5
Bob	11
Charlie	8

Deployed by 0xLabs in 2017



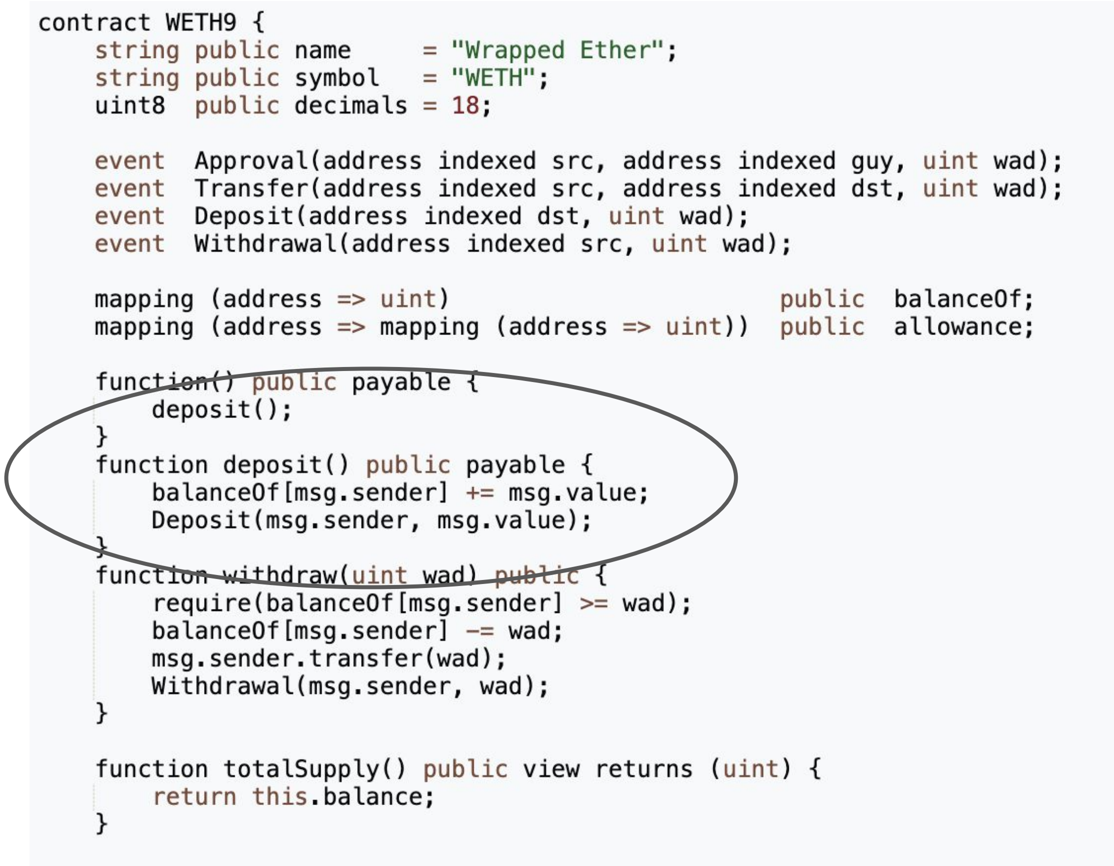
```
contract WETH9 {
    string public name     = "Wrapped Ether";
    string public symbol   = "WETH";
    uint8  public decimals = 18;

    event  Approval(address indexed src, address indexed guy, uint wad);
    event  Transfer(address indexed src, address indexed dst, uint wad);
    event  Deposit(address indexed dst, uint wad);
    event  Withdrawal(address indexed src, uint wad);

    mapping (address => uint)                       public  balanceOf;
    mapping (address => mapping (address => uint))  public  allowance;

    function() public payable {
        deposit();
    }
    function deposit() public payable {
        balanceOf[msg.sender] += msg.value;
        Deposit(msg.sender, msg.value);
    }
    function withdraw(uint wad) public {
        require(balanceOf[msg.sender] >= wad);
        balanceOf[msg.sender] -= wad;
        msg.sender.transfer(wad);
        Withdrawal(msg.sender, wad);
    }

    function totalSupply() public view returns (uint) {
        return this.balance;
    }
```



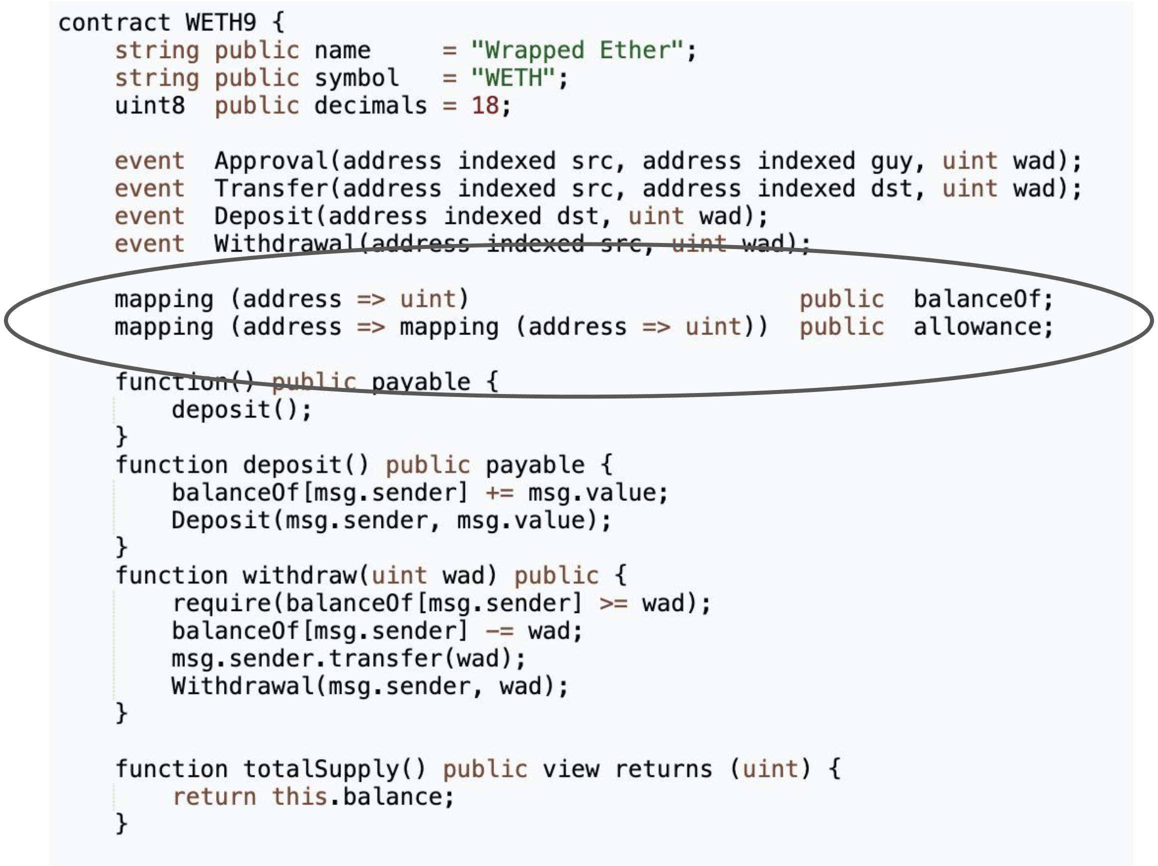
```
contract WETH9 {
    string public name     = "Wrapped Ether";
    string public symbol   = "WETH";
    uint8  public decimals = 18;

    event  Approval(address indexed src, address indexed guy, uint wad);
    event  Transfer(address indexed src, address indexed dst, uint wad);
    event  Deposit(address indexed dst, uint wad);
    event  Withdrawal(address indexed src, uint wad);

    mapping (address => uint)                       public  balanceOf;
    mapping (address => mapping (address => uint))  public  allowance;

    function() public payable {
        deposit();
    }
    function deposit() public payable {
        balanceOf[msg.sender] += msg.value;
        Deposit(msg.sender, msg.value);
    }
    function withdraw(uint wad) public {
        require(balanceOf[msg.sender] >= wad);
        balanceOf[msg.sender] -= wad;
        msg.sender.transfer(wad);
        Withdrawal(msg.sender, wad);
    }

    function totalSupply() public view returns (uint) {
        return this.balance;
    }
```



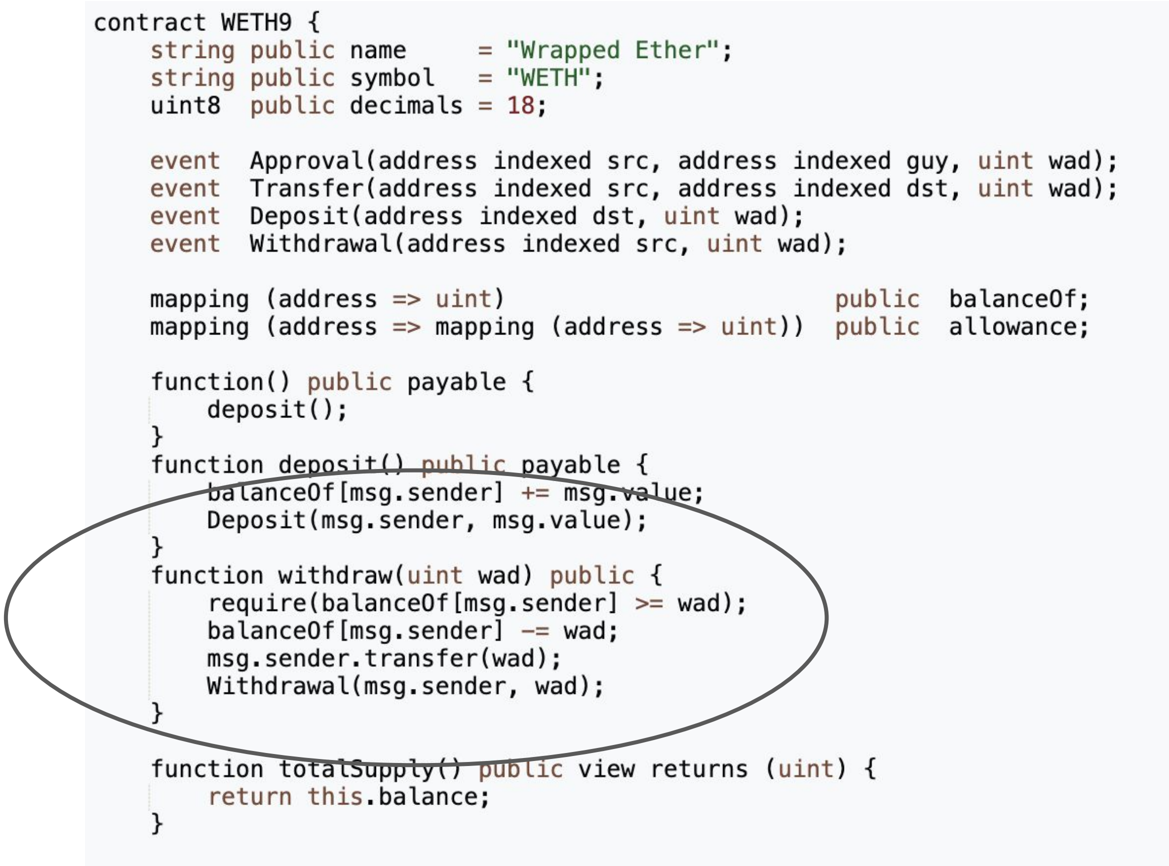
```
contract WETH9 {
    string public name     = "Wrapped Ether";
    string public symbol   = "WETH";
    uint8  public decimals = 18;

    event  Approval(address indexed src, address indexed guy, uint wad);
    event  Transfer(address indexed src, address indexed dst, uint wad);
    event  Deposit(address indexed dst, uint wad);
    event  Withdrawal(address indexed src, uint wad);

    mapping (address => uint)                       public  balanceOf;
    mapping (address => mapping (address => uint))  public  allowance;

    function() public payable {
        deposit();
    }
    function deposit() public payable {
        balanceOf[msg.sender] += msg.value;
        Deposit(msg.sender, msg.value);
    }
    function withdraw(uint wad) public {
        require(balanceOf[msg.sender] >= wad);
        balanceOf[msg.sender] -= wad;
        msg.sender.transfer(wad);
        Withdrawal(msg.sender, wad);
    }

    function totalSupply() public view returns (uint) {
        return this.balance;
    }
```



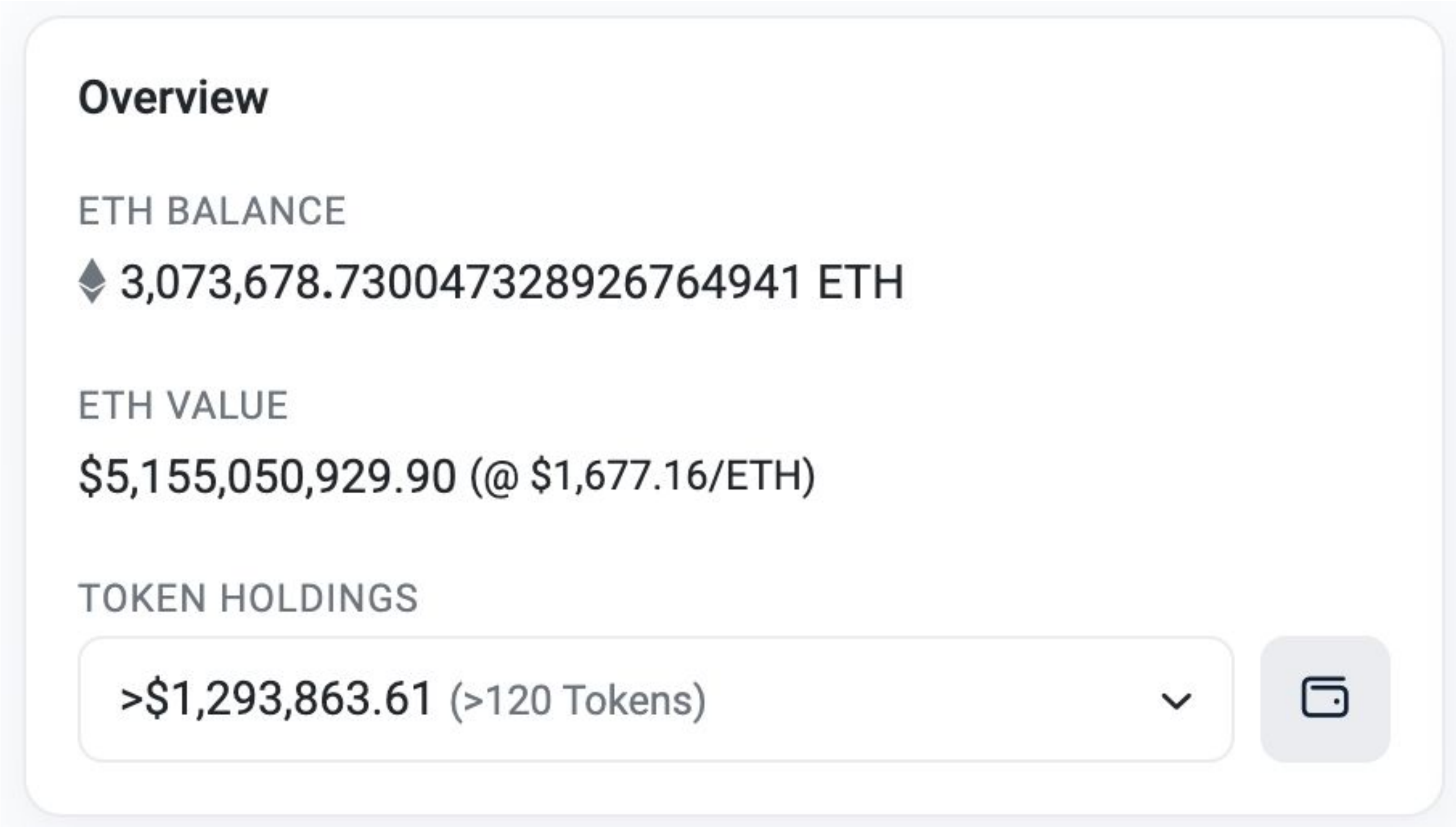
```
contract WETH9 {
    string public name     = "Wrapped Ether";
    string public symbol   = "WETH";
    uint8  public decimals = 18;

    event  Approval(address indexed src, address indexed guy, uint wad);
    event  Transfer(address indexed src, address indexed dst, uint wad);
    event  Deposit(address indexed dst, uint wad);
    event  Withdrawal(address indexed src, uint wad);

    mapping (address => uint)                       public  balanceOf;
    mapping (address => mapping (address => uint))  public  allowance;

    function() public payable {
        deposit();
    }
    function deposit() public payable {
        balanceOf[msg.sender] += msg.value;
        Deposit(msg.sender, msg.value);
    }
    function withdraw(uint wad) public {
        require(balanceOf[msg.sender] >= wad);
        balanceOf[msg.sender] -= wad;
        msg.sender.transfer(wad);
        Withdrawal(msg.sender, wad);
    }

    function totalSupply() public view returns (uint) {
        return this.balance;
    }
```



Overview

ETH BALANCE

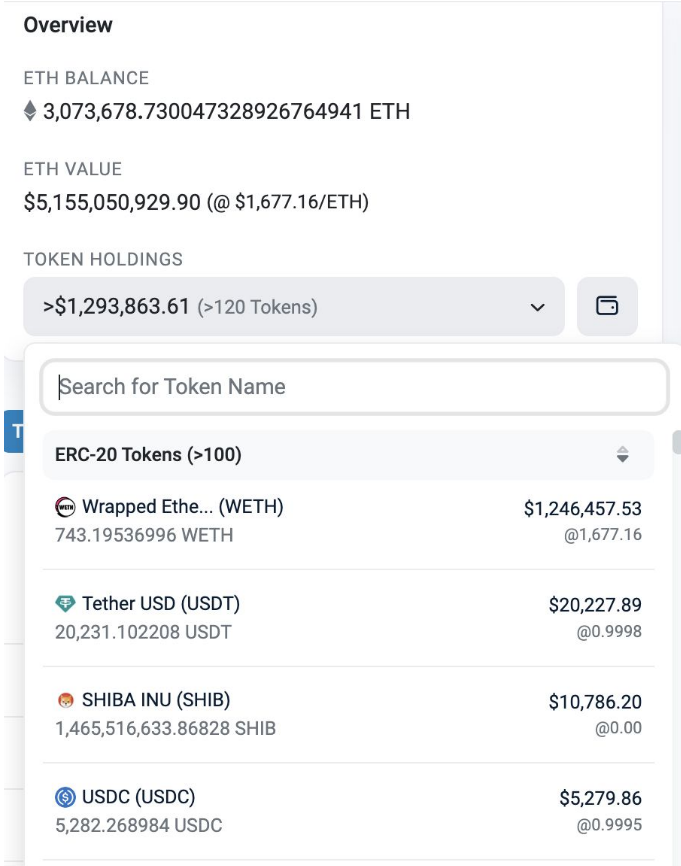
♦ 3,073,678.730047328926764941 ETH

ETH VALUE

$5,155,050,929.90 (@ $1,677.16/ETH)

TOKEN HOLDINGS

>$1,293,863.61 (>120 Tokens)



Overview

ETH BALANCE

3,073,678.730047328926764941 ETH

ETH VALUE

$5,155,050,929.90 (@ $1,677.16/ETH)

TOKEN HOLDINGS

>$1,293,863.61 (>120 Tokens)

Search for Token Name

ERC-20 Tokens (>100)

Wrapped Ethe... (WETH) 743.19536996 WETH	$1,246,457.53 @1,677.16
Tether USD (USDT) 20,231.102208 USDT	$20,227.89 @0.9998
SHIBA INU (SHIB) 1,465,516,633.86828 SHIB	$10,786.20 @0.00
USDC (USDC) 5,282.268984 USDC	$5,279.86 @0.9995