

Double entry point issues - From breaking Compound to Uniswap v4



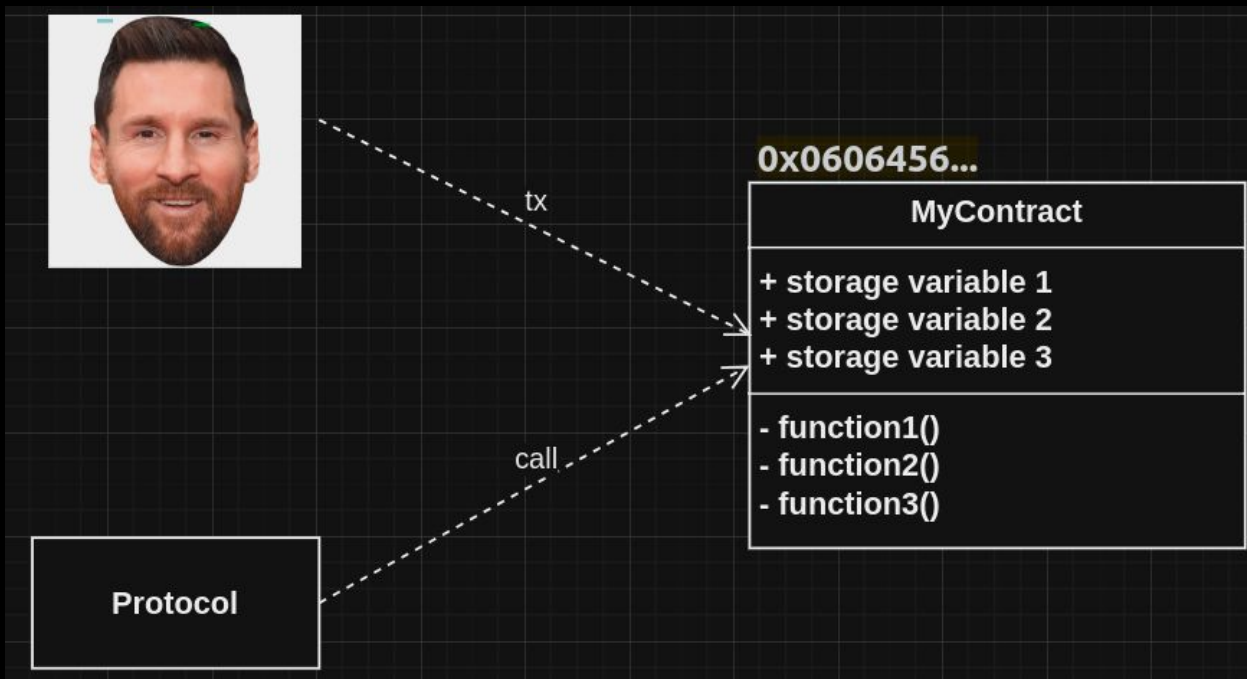
Jota Carpanelli
Head of Security



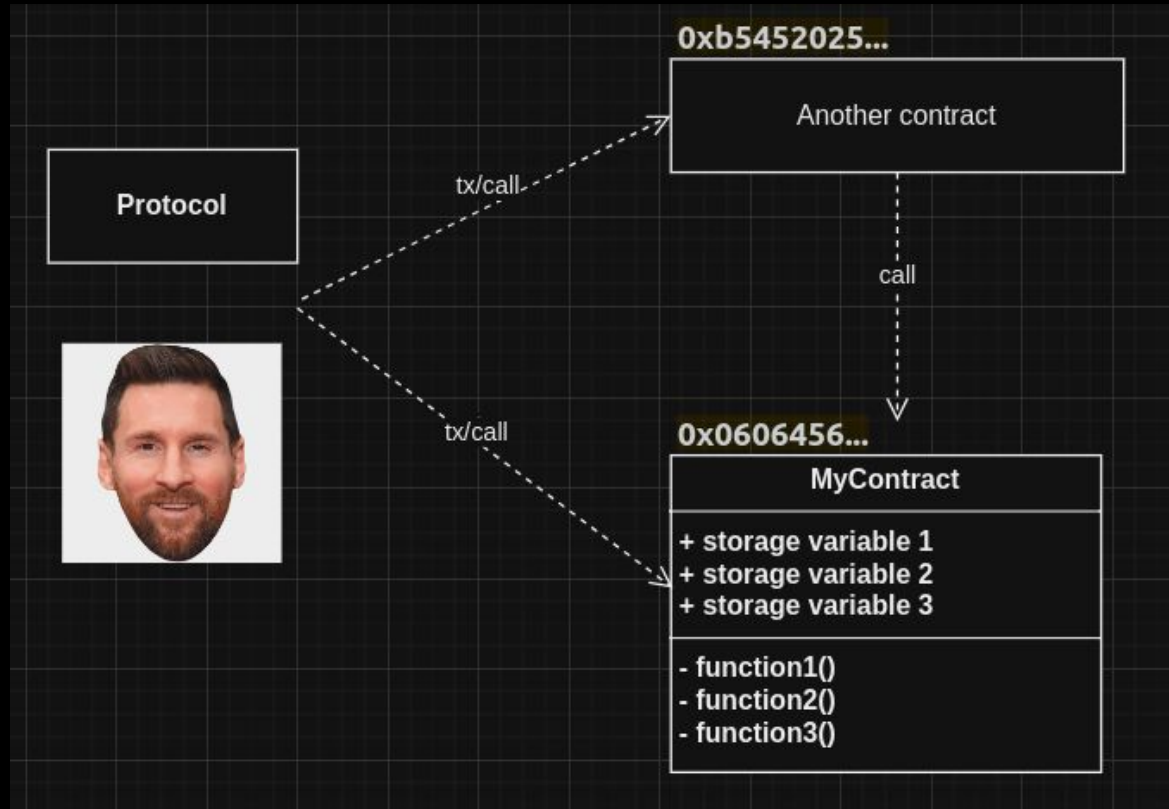
@holajotola

What is a double entry point?

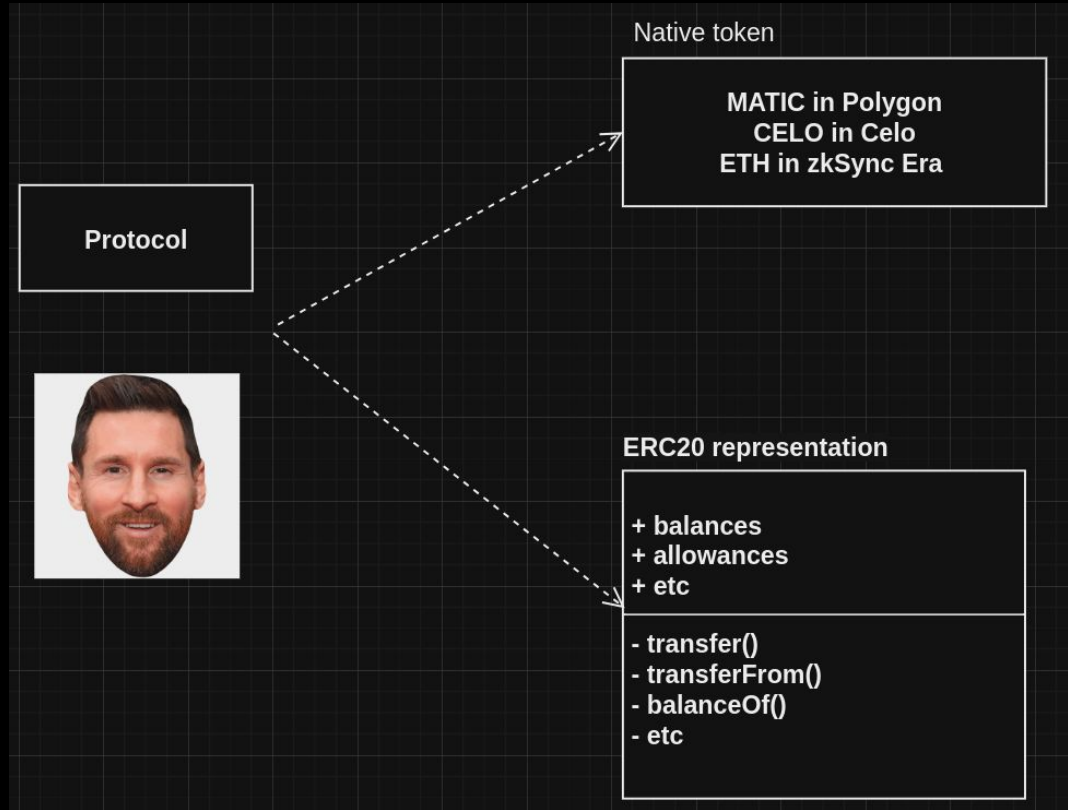
Double entry point - What's that?



Double entry point - What's that?



Double entry point - What's that? - Native + ERC20 Token representation

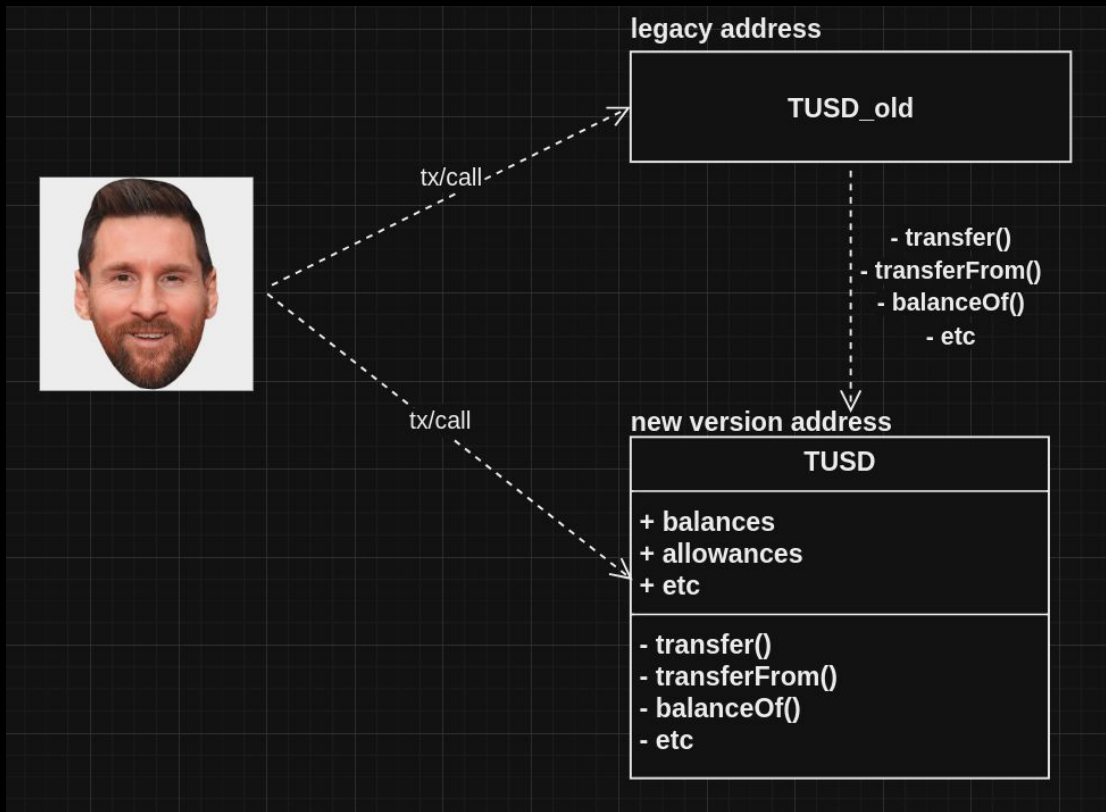


Double entry point - What's the problem?

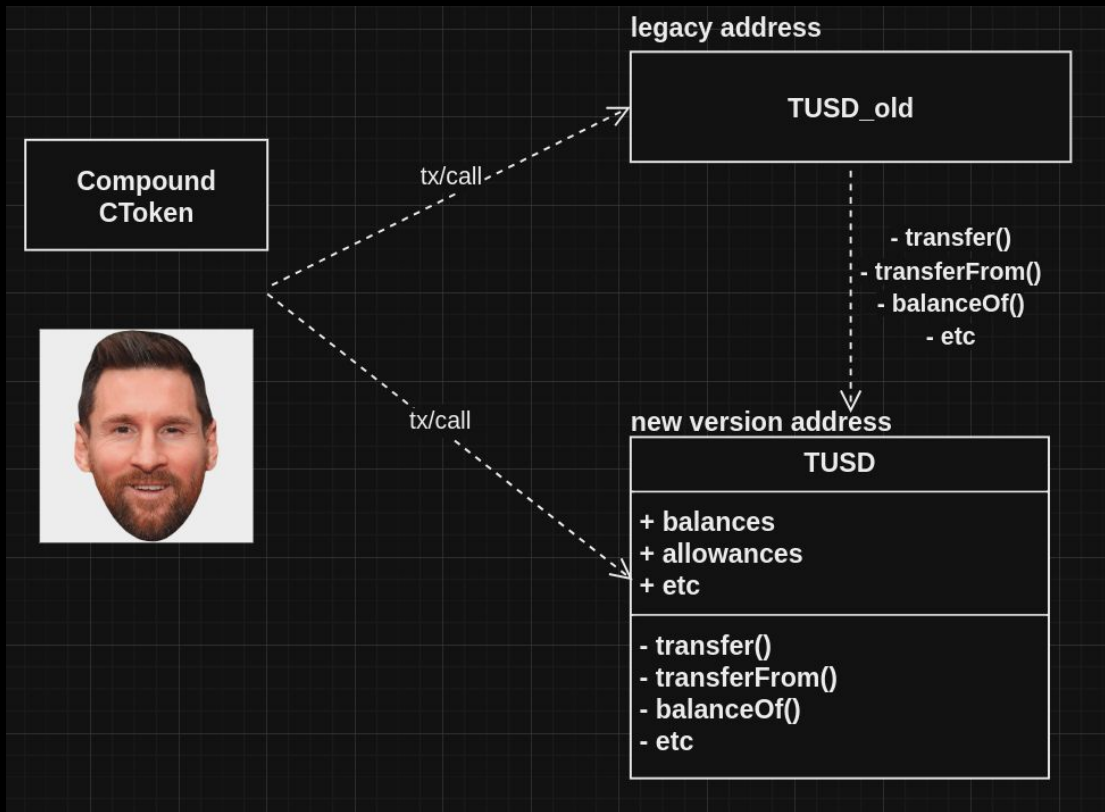
- Unclear to end users which contract is the correct one (trust issues)
- Unclear to protocols which contract to interact with when integrating.
- Introduces numerous security caveats.

Case 1: Compound - 2022

Case 1: Compound



Case 1: Compound



Case 1: Compound

```
function sweepToken(EIP20NonStandardInterface token) external {  
    require(address(token) != underlying, "Cannot sweep underlying token");  
    uint256 balance = token.balanceOf(address(this));  
    token.transfer(admin, balance);  
}
```


Case 1: Compound

```
function sweepToken(EIP20NonStandardInterface token) external {  
    require(address(token) != underlying, "Cannot sweep underlying token");  
    uint256 balance = token.balanceOf(address(this));  
    token.transfer(admin, balance);  
}
```

Case 1: Compound

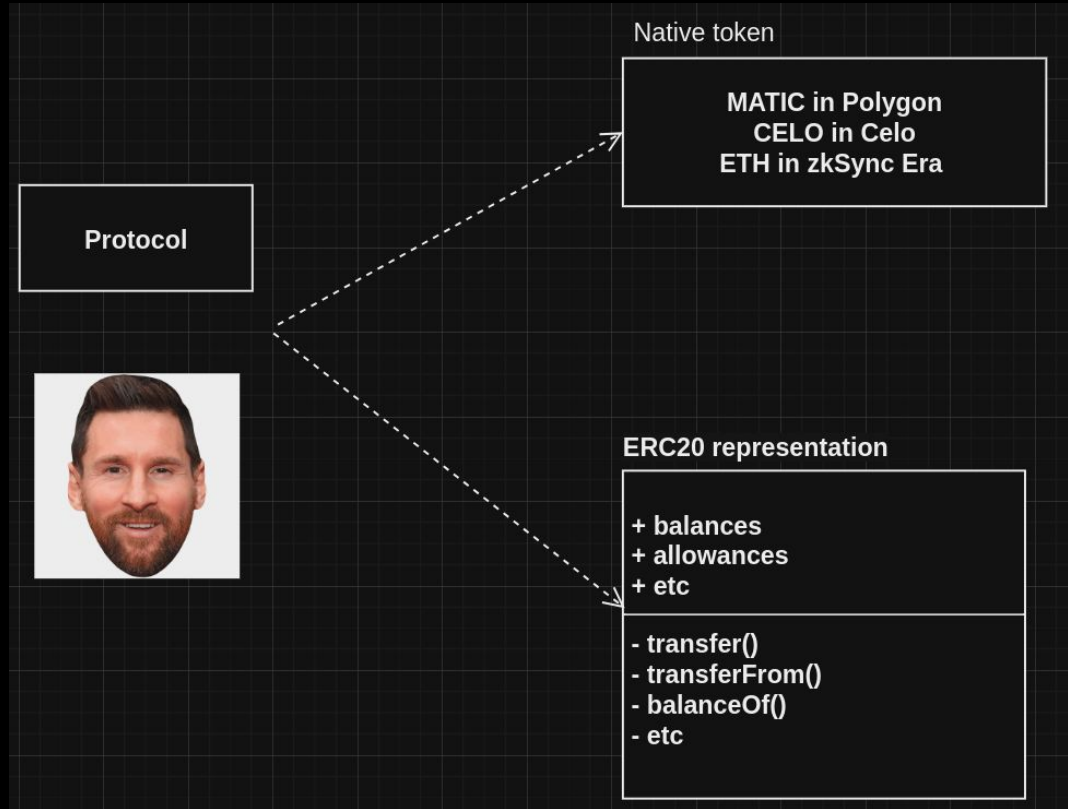
- Wrong CToken:Token price
- Wrong accounting of future minting/burning of CToken
- Possibility to steal funds from the pool

Case 1: Compound

- Wrong CToken:Token price
- Wrong accounting of future minting/burning of CToken
- Possibility to steal funds from the pool \Rightarrow **~50M TUSD** 

Case 2: Uniswap v4 - 2024 🔥

Case 2: Uniswap v4



Case 2: Uniswap v4

```
function settle(Currency currency) external payable ... {
    if (currency.isNative()) {
        paid = msg.value;
    } else {
        if (msg.value > 0) NonZeroNativeValue.selector.revertWith();
        uint256 reservesBefore = currency.getReserves();
        uint256 reservesNow = sync(currency);
        paid = reservesNow - reservesBefore;
    }

    _accountDelta(currency, paid.toInt128(), msg.sender);
}
```


Case 2: Uniswap v4

```
function settle(Currency currency) external payable ... {  
    if (currency.isNative()) {  
        paid = msg.value;  
    } else {  
        if (msg.value > 0) NonZeroNativeValue.selector.revertWith();  
        uint256 reservesBefore = currency.getReserves();  
        uint256 reservesNow = sync(currency);  
        paid = reservesNow - reservesBefore;  
    }  
  
    _accountDelta(currency, paid.toInt128(), msg.sender);  
}
```

Case 2: Uniswap v4

```
function settle(Currency currency) external payable ... {
    if (currency.isNative()) {
        paid = msg.value;
    } else {
        if (msg.value > 0) // reverts
            uint256 reservesBefore = currency.getReserves();
            uint256 reservesNow = sync(currency);
            paid = reservesNow - reservesBefore;
    }
    _accountDelta(currency, paid.toInt128(), msg.sender);
}
```

1. sync(celo_addr)

Case 2: Uniswap v4

```
function settle(Currency currency) external payable ... {  
    if (currency.isNative()) {  
        paid = msg.value;  
    } else {  
        if (msg.value > 0) // reverts  
            uint256 reservesBefore = currency.getReserves();  
            uint256 reservesNow = sync(currency);  
            paid = reservesNow - reservesBefore;  
    }  
    _accountDelta(currency, paid.toInt128(), msg.sender);  
}
```

1. `sync(celo_addr)`
2. `settle{value: 10}(address(0x0))`

Case 2: Uniswap v4

```
function settle(Currency currency) external payable ... {  
    if (currency.isNative()) {  
        paid = msg.value;  
    } else {  
        if (msg.value > 0) // reverts  
            uint256 reservesBefore = currency.getReserves();  
            uint256 reservesNow = sync(currency);  
            paid = reservesNow - reservesBefore;  
        }  
    }  
    _accountDelta(currency, paid.toInt128(), msg.sender);  
}
```

1. `sync(celo_addr)`
2. `settle{value: 10}(address(0x0))`
3. `settle(celo_addr)`

Case 2: Uniswap v4

```
function settle(Currency currency) external payable ... {
    if (currency.isNative()) {
        paid = msg.value;
    } else {
        if (msg.value > 0) // reverts
            uint256 reservesBefore = currency.getReserves();
            uint256 reservesNow = sync(currency);
            paid = reservesNow - reservesBefore;
    }
    _accountDelta(currency, paid.toInt128(), msg.sender);
}
```

1. `sync(celo_addr)`
2. `settle{value: 10}(address(0x0))`
3. `settle(celo_addr)`

~ 15M USD at stake 

Takeaways?

Takeaways

- 2022-2024 ⇒ issue still around
- Never trust any protocol or token you interact with.
- Double check that the protocol/token you interact with doesn't have a double entry point
- Even if it doesn't, don't assume it will never have a double entry point.

Thank You



Jota Carpanelli - Head of Security
@holajotola



We're hiring! 🙋

