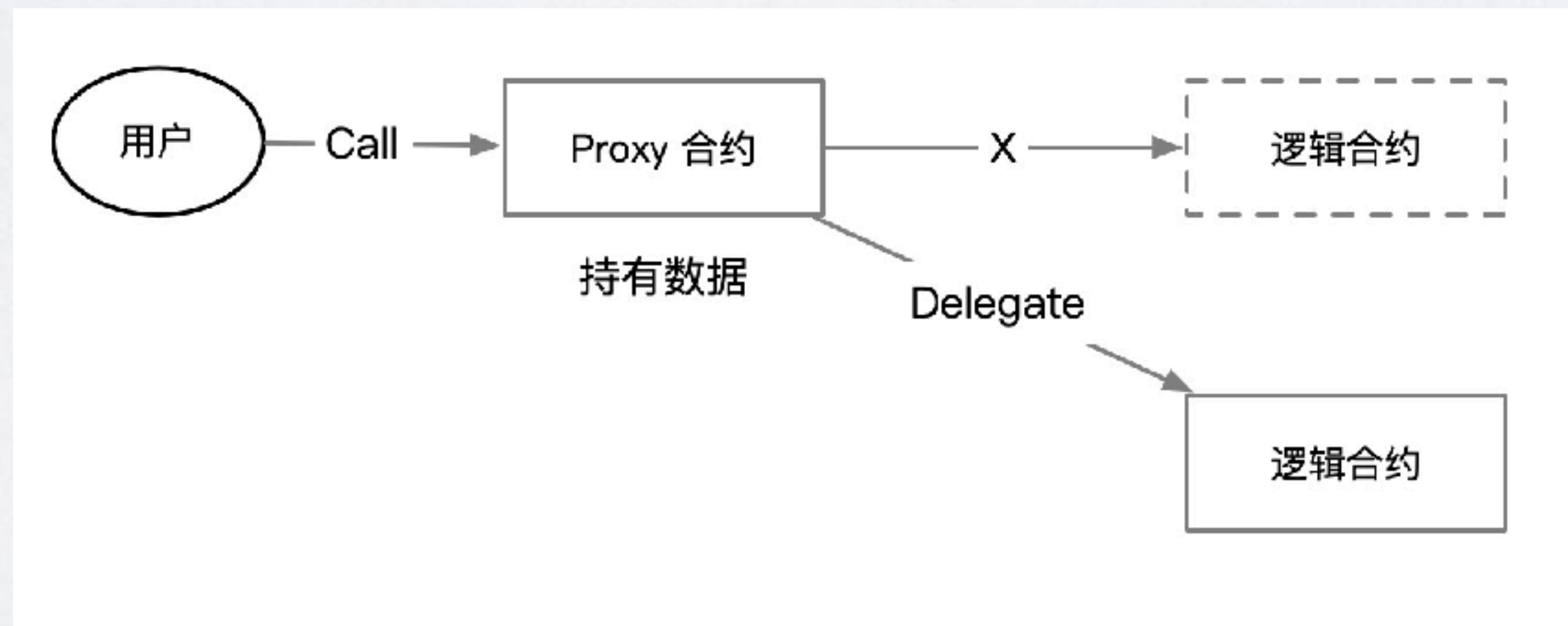


练习题

- 使用 EIP2612 标准 Token , 使用签名的方式 deposit
- 离线授权的白名单地址才可购买 NFT

合约升级

- 合约一旦部署，便不可更改
- 合约中有错误如何修复？
- 要添加额外功能，要怎么办？



Call 与 delegateCall 回顾



Msg.sender = 用户

Msg.sender = A
使用 B 自己的存储



Msg.sender = 用户

Msg.sender = 用户
使用 A 的存储

testCall_Delegate.sol

尝试给Counter升级

```
pragma solidity ^0.8.0;

contract Counter {
    uint private counter;

    function add(uint256 i) public {
        counter += 1;
    }

    function get() public view returns(uint) {
        return counter;
    }
}
```


升级尝试 1 – 使用代理

- 代理和逻辑合约的存储布局不一致发生无法预期的错误

CounterProxy

Storage	
slot	变量
0	impl
1	counter

CounterProxy.sol

Counter.sol

Storage	
slot	变量
0	counter

升级添加的变量，必须在末尾添加

```
bytes32(uint(keccak256("eip1967.proxy.implementation")) - 1)
```


升级尝试 2 – 统一委托

- 如何委托未来添加的函数及获取返回值？

- fallback 统一委托

- 用汇编魔法获取返回值

CounterFallback.sol

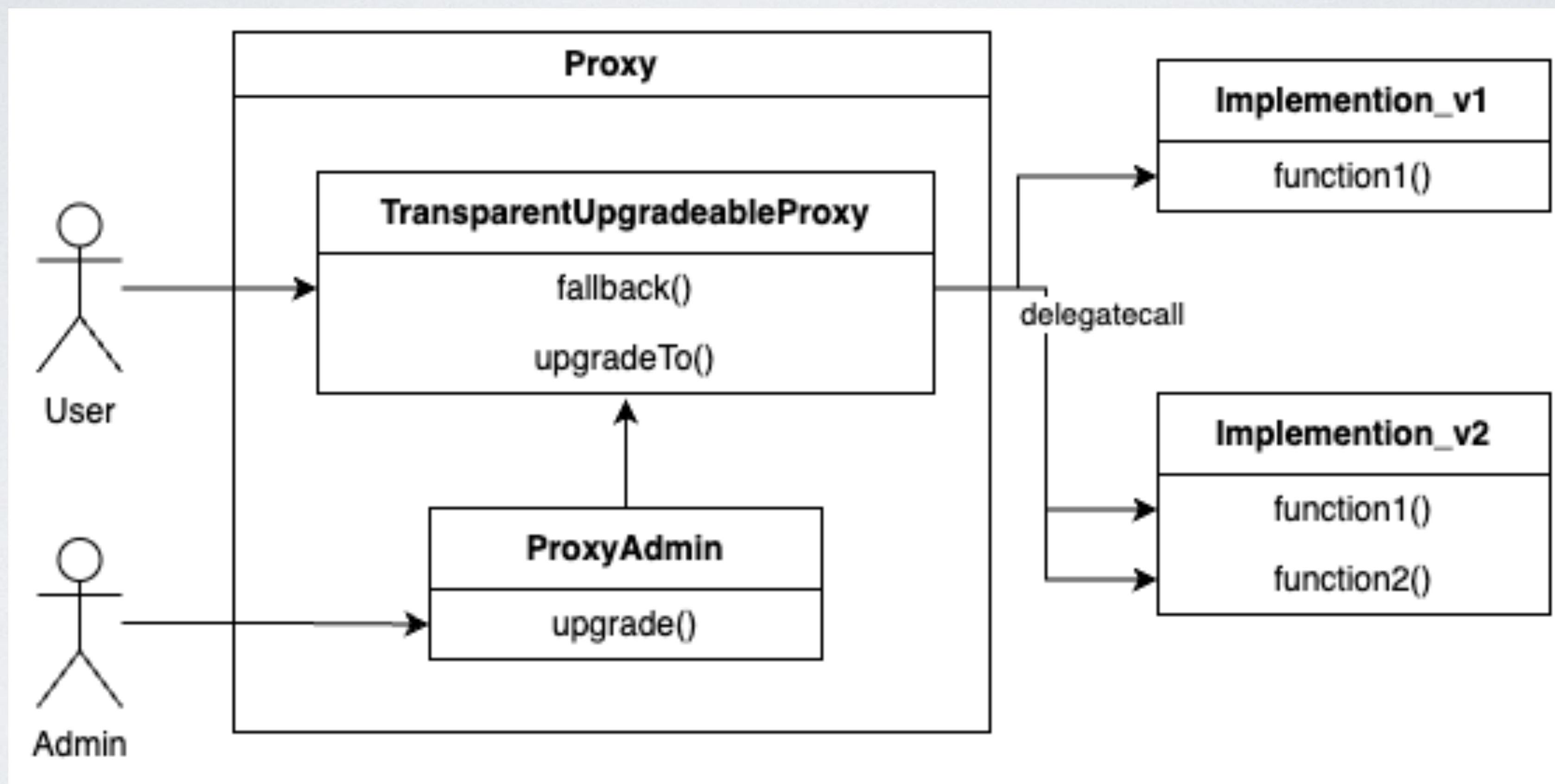
```
assembly {
    //获得自由空闲指针
    let ptr := mload(0x40)
    //将返回值从返回缓冲去copy到指针所指位置
    returndatacopy(ptr, 0, returndatasize())

    //根据是否调用成功决定是返回数据还是直接revert整个函数
    switch success
    case 0 { revert(ptr, returndatasize()) }
    default { return(ptr, returndatasize()) }
}
```

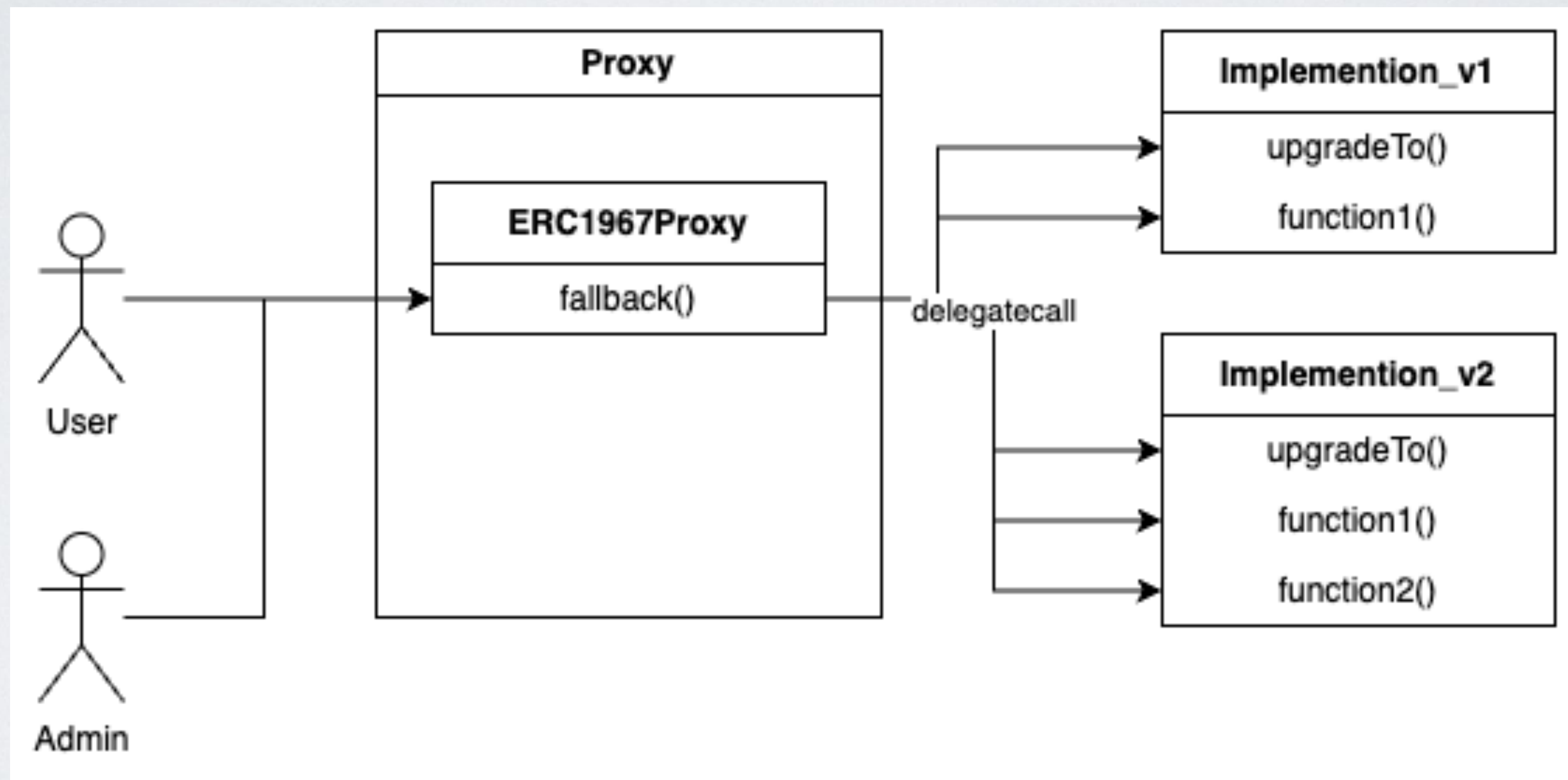

升级尝试 – 通用升级

- 函数冲撞问题，若某个函数与upgradeTo() 函数选择器一样会出现什么问题？
- 初始化问题？
- 透明代理（Transparent Proxy） - ERC1967Proxy
- UUPS（universal upgradeable proxy standard） - ERC-1822

透明代理



UUPS



使用 DelegateCall 要注意的点

- 代理和逻辑合约的存储布局需要一致。
- delegateCall 返回值
 - (bool success, bytes memory returnData) = address.delegatecall(payload);
 - Bytes 需转化为具体的类型
- 不能有函数冲撞
- 初始化问题？

合约升级

合约升级



重温 Call/DelegateCall

解决1. 代理和逻辑合约存储布局问题

解决2: 函数调用及返回值泛化

解决3: 函数冲撞

开发中使用升级 (Hardhat)

- hardhat-upgrades
 - `npm install --save-dev @openzeppelin/hardhat-upgrades`
- contracts-upgradeable
 - `npm install --save-dev @openzeppelin/contracts-upgradeable`

<https://docs.openzeppelin.com/contracts/5.x/upgradeable>

https://github.com/xilibi2003/training_camp_2/tree/main/w3_2_code

开发中使用升级Foundry

- foundry-upgrades
 - forge install OpenZeppelin/openzeppelin-foundry-upgrades
- openzeppelin-contracts-upgradeable
 - forge install OpenZeppelin/openzeppelin-contracts-upgradeable

<https://github.com/OpenZeppelin/openzeppelin-foundry-upgrades>

https://github.com/OpenSpace100/blockchain-tasks/tree/upgrade/w3_permit

练习题

- 部署一个可升级的 NFT 市场合约
 - 第一版本普通合约
 - 第二版本，加入离线签名上架 NFT 功能方法（签名内容：tokenId， 价格），实现用户一次性 setApproveAll 给 NFT 市场合约，每次上架时使用签名上架。
- 部署到测试网，并开源到区块链浏览器（在 Readme.md 中备注代理合约及两个实现的合约地址）

MultiCall

- 如何同时调用合约里的多个（次）函数？

MultiCall

```
function multical(bytes[] calldata data) external virtual returns (bytes[]  
memory results) {  
    results = new bytes[](data.length);  
    for (uint256 i = 0; i < data.length; i++) {  
        results[i] = this.delegateCall(data[i]);  
    }  
    return results;  
}
```