

以太坊合约升级





目录

01 概述

02 合约存储

03 Call和DelegateCall

04 原理和实现



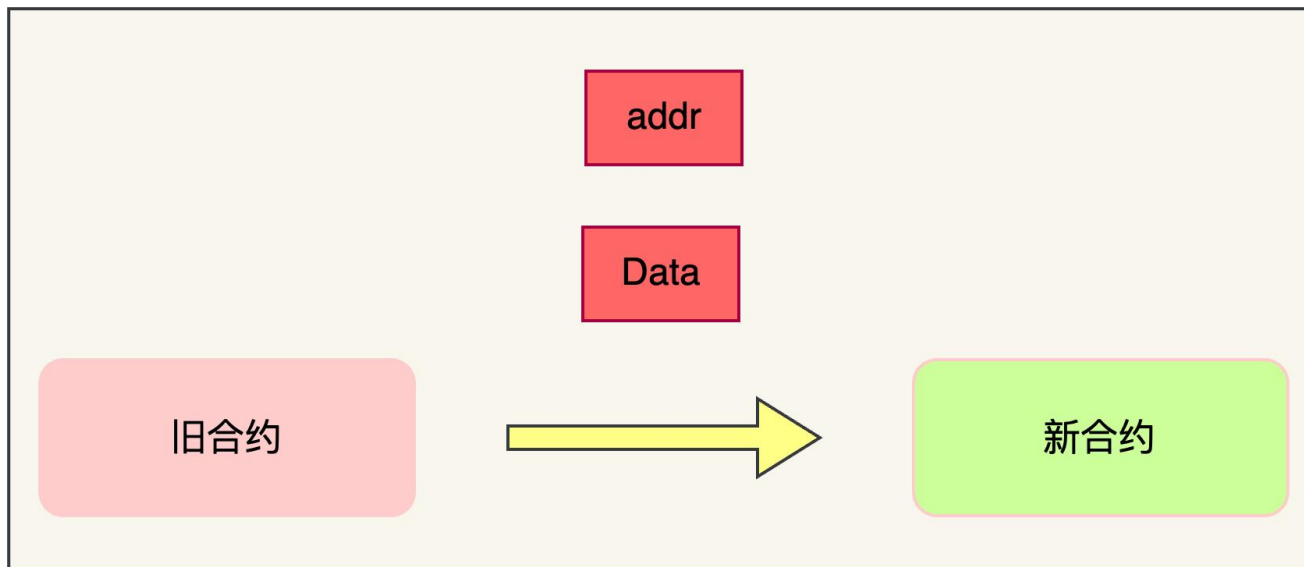
05 常见升级模式

06 代码演示

01 概述

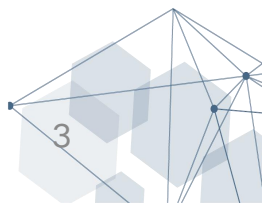
- Code Is Law

- 源码公开
- 合约不可更改



- 合约升级

- 逻辑增加
- Bug Fix

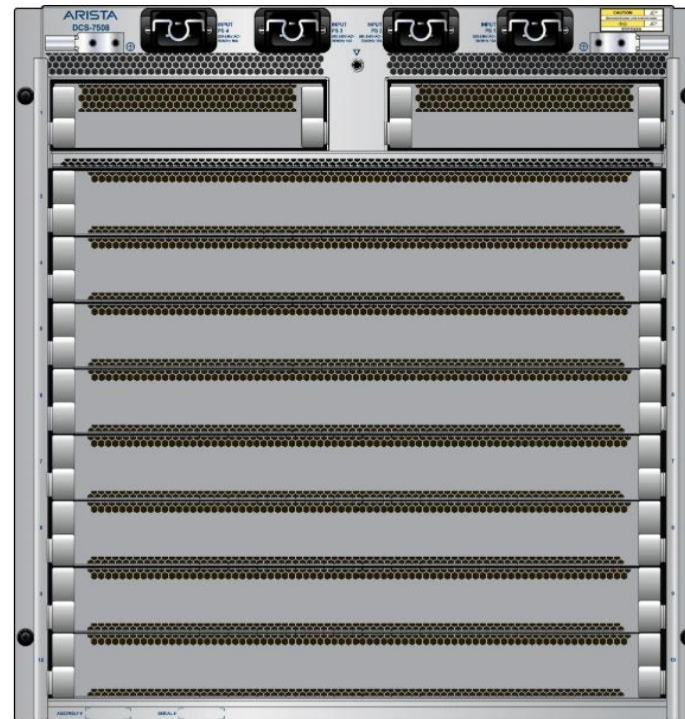
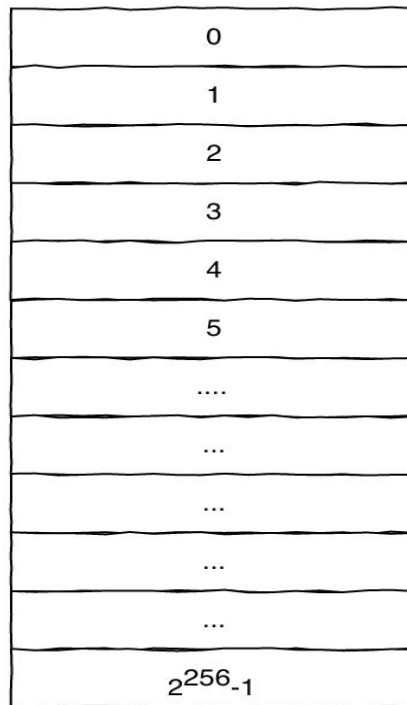




02合约存储

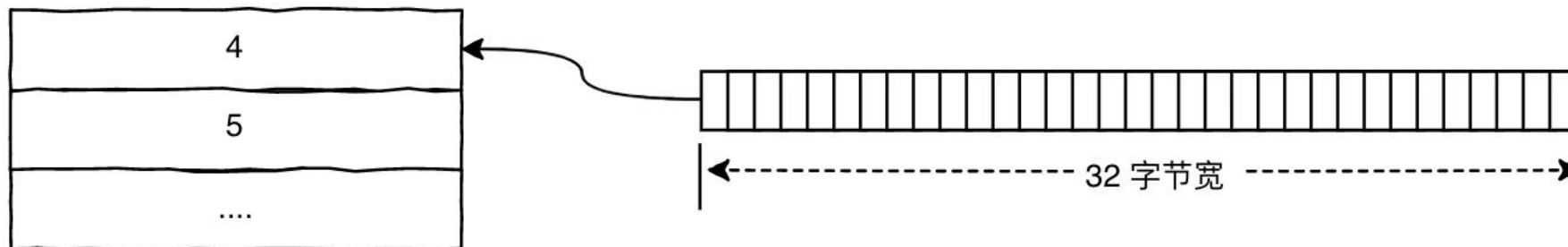
$2^{256} * 32$ 字节

2^{32} 次方 ≈ 40 亿



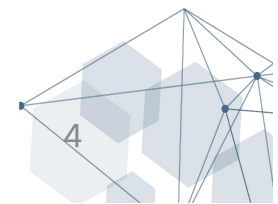
图：插槽式数组存储

以太坊技术与实现

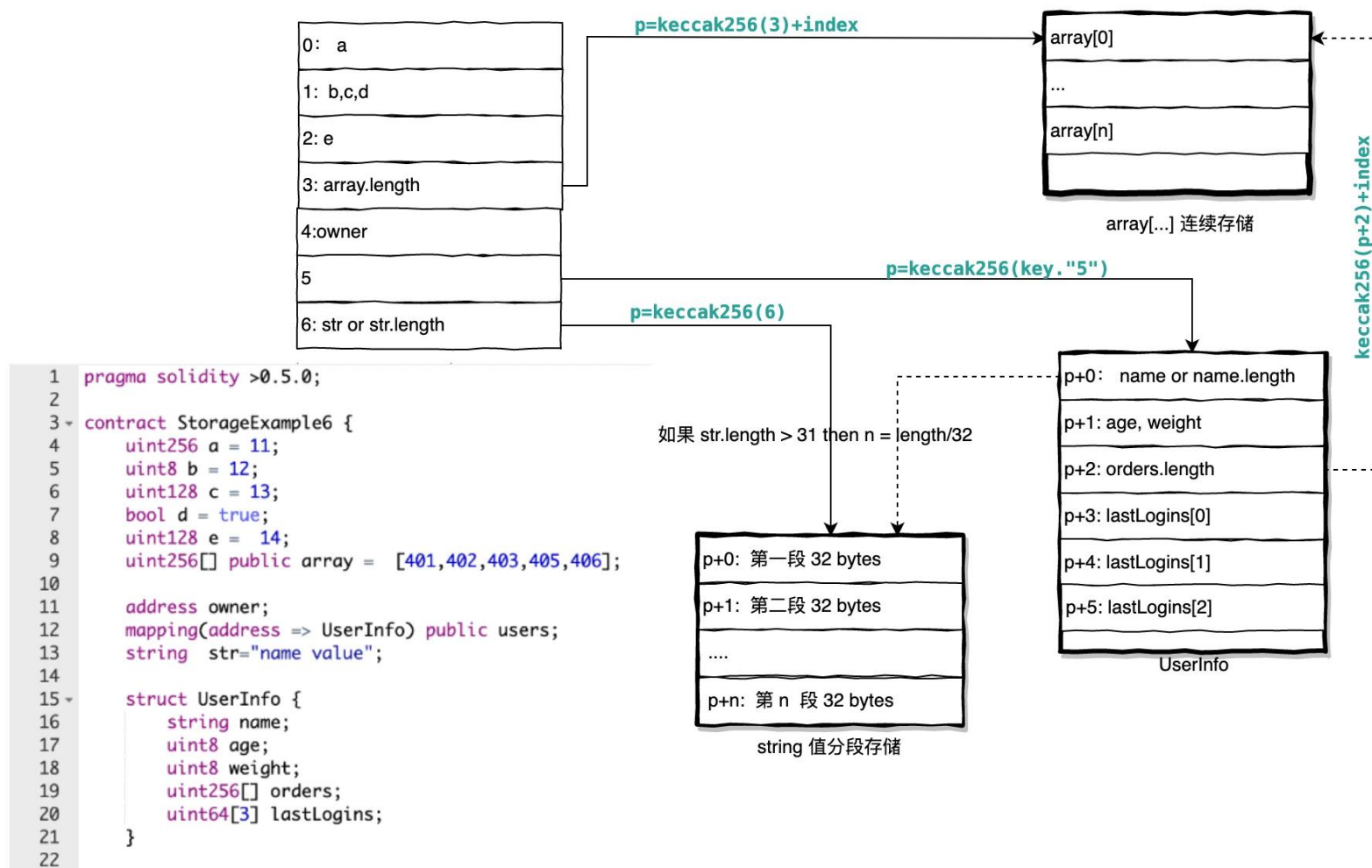


图：每个插槽 32 字节宽

以太坊技术与实现



02合约存储

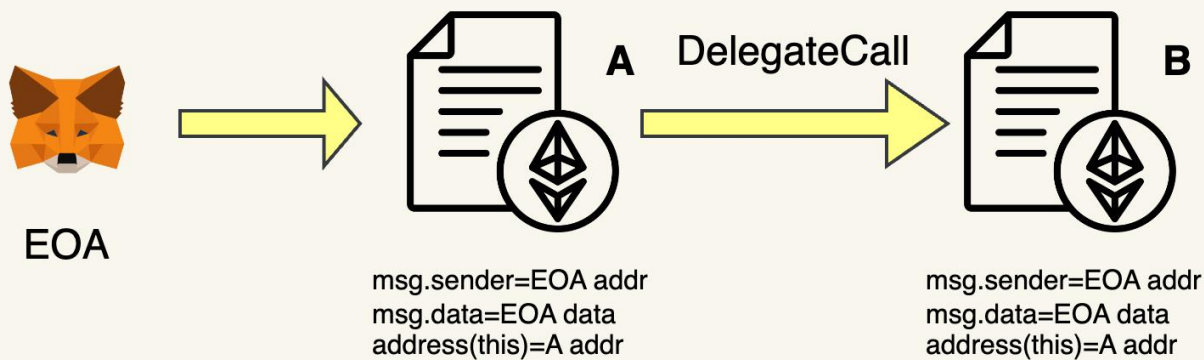
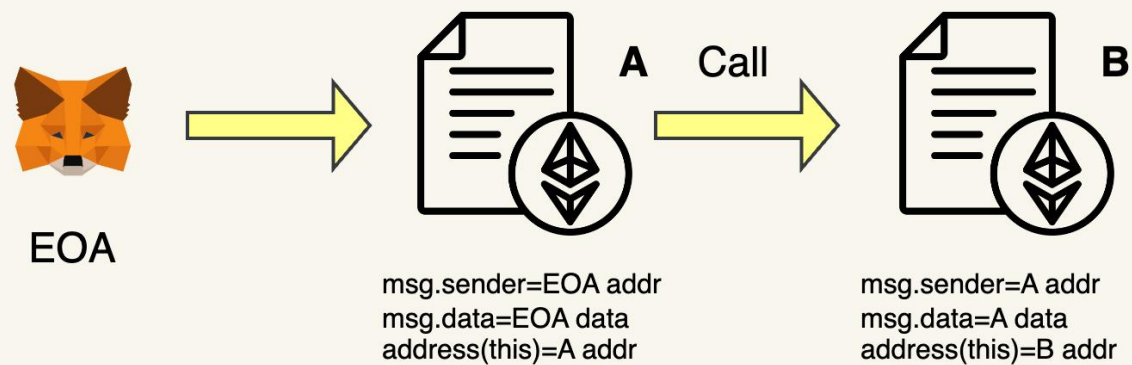


图：一个合约的存储布局

03Call和DelegateCall

```
function callFoo(uint256 _alice, uint256 _bob) external {
    (bool success, bytes memory data) =
        addressB.call(abi.encodeWithSignature("foo(uint256,uint256)",
            _alice, _bob));
    require(success, "Tx failed");
}

function delegateCallFoo(uint256 _alice, uint256 _bob) external {
    (bool success, bytes memory data) =
        addressB.delegatecall(abi.encodeWithSignature("foo(uint256,uint256)",
            _alice, _bob));
    require(success, "Tx failed");
}
```



A Call B: B的环境信息是B

A DelegateCall B: B的环境信息是A
相当于A合约importB合约的代码逻辑

03Call和DelegateCall

属性名称	slot	value
Alice	0	0
Bob	1	0
b	2	b addr
...	3	...

属性名称	slot	value
Alice	0	0
Bob	1	0
...	2	...
...	3	...

属性名称	slot	value
Alice	0	0
Bob	1	0
b	2	b addr
...	3	...

属性名称	slot	value
Alice	0	0
Bob	1	0
...	2	...
...	3	...



属性名称	slot	value
Alice	0	0
Bob	1	0
b	2	b addr
...	3	...

属性名称	slot	value
Alice	0	1
Bob	1	2
...	2	...
...	3	...

属性名称	slot	value
Alice	0	1
Bob	1	2
b	2	b addr
...	3	...

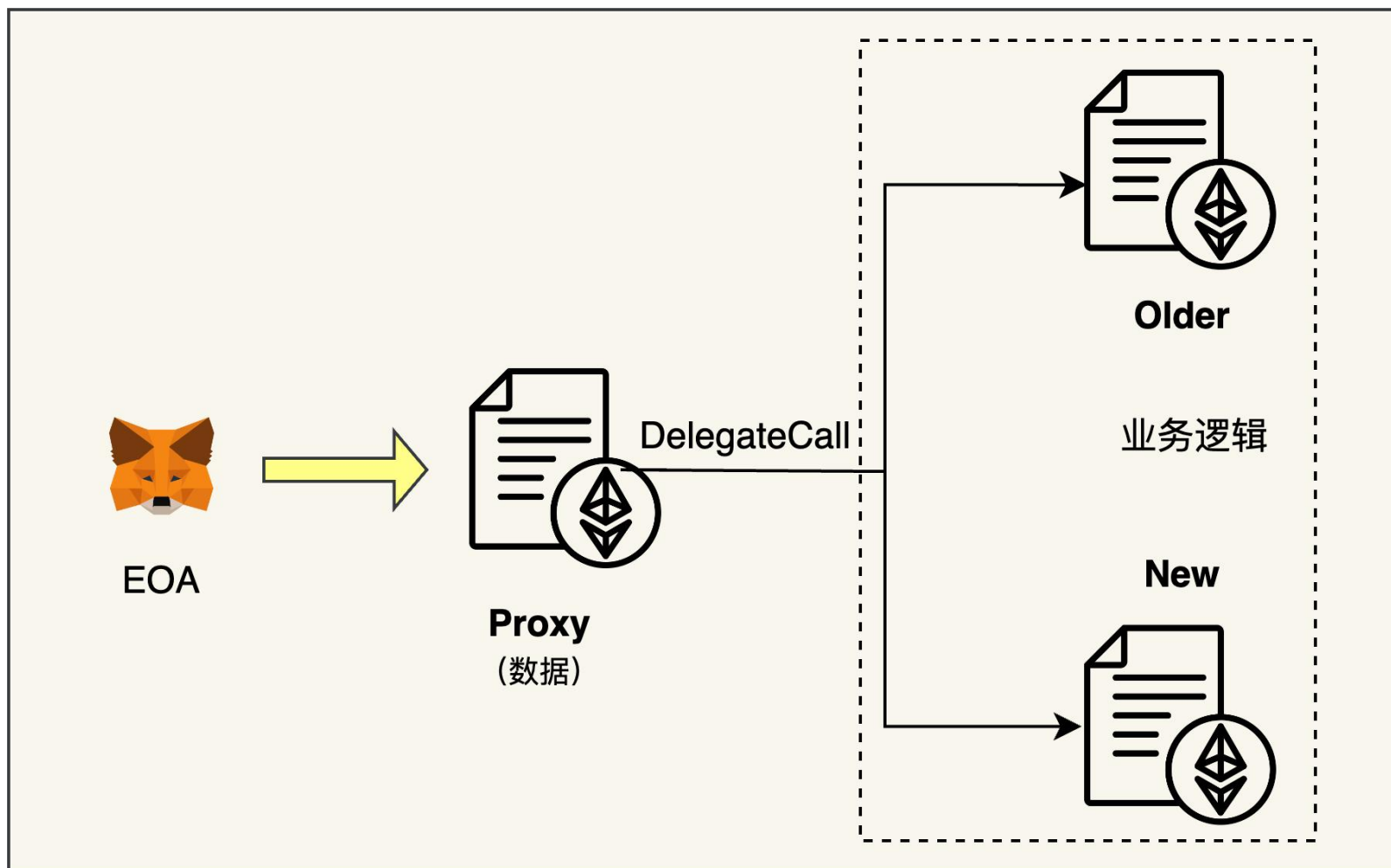
属性名称	slot	value
Alice	0	0
Bob	1	0
...	2	...
...	3	...

A存储数据

B业务逻辑



04原理和实现



f1: a=a+1

f1: a=a+2

04原理和实现

触发fallback() 还是 receive()?



```
4 contract Proxy {
5     address public implementation;
6     address public admin;
7     constructor() public {
8         admin = msg.sender;
9     }
10    function setImplementation(address newImplementation) external {
11        require(msg.sender == admin, "must be admin");
12        implementation = newImplementation;
13    }
14    function changeAdmin(address newAdmin) external {
15        require(msg.sender == admin, "must be admin");
16        admin = newAdmin;
17    }
18    function _delegate(address implementation) internal virtual {
19        assembly {
20            calldatacopy(0, 0, calldatasize())
21            let result := delegatecall(gas(), implementation, 0, calldatasize(), 0, 0)
22            returndatacopy(0, 0, returndatasize())
23            switch result
24            case 0 {
25                revert(0, returndatasize())
26            }
27            default {
28                return (0, returndatasize())
29            }
30        }
31    }
32    function _fallback() internal virtual {
33        _delegate(implementation);
34    }
35    fallback() external payable virtual {
36        _fallback();
37    }
38 }
```

04原理和实现

slot 冲突

方法冲突

```
contract ProxyFixTwoBug {
    // bytes32(uint256(keccak256('eip1967.proxy.implementation')) - 1)
    bytes32 private constant _IMPLEMENTATION_SLOT = 0x360894a13ba1a3210667c828492db98dca3e2076cc3735a920a3ca505d382bbc;
    // bytes32(uint256(keccak256('eip1967.proxy.admin')) - 1)
    bytes32 private constant _ADMIN_SLOT = 0xb53127684a568b3173ae13b9f8a6016e243e63b6e8ee1178d6a717850b5d6103;

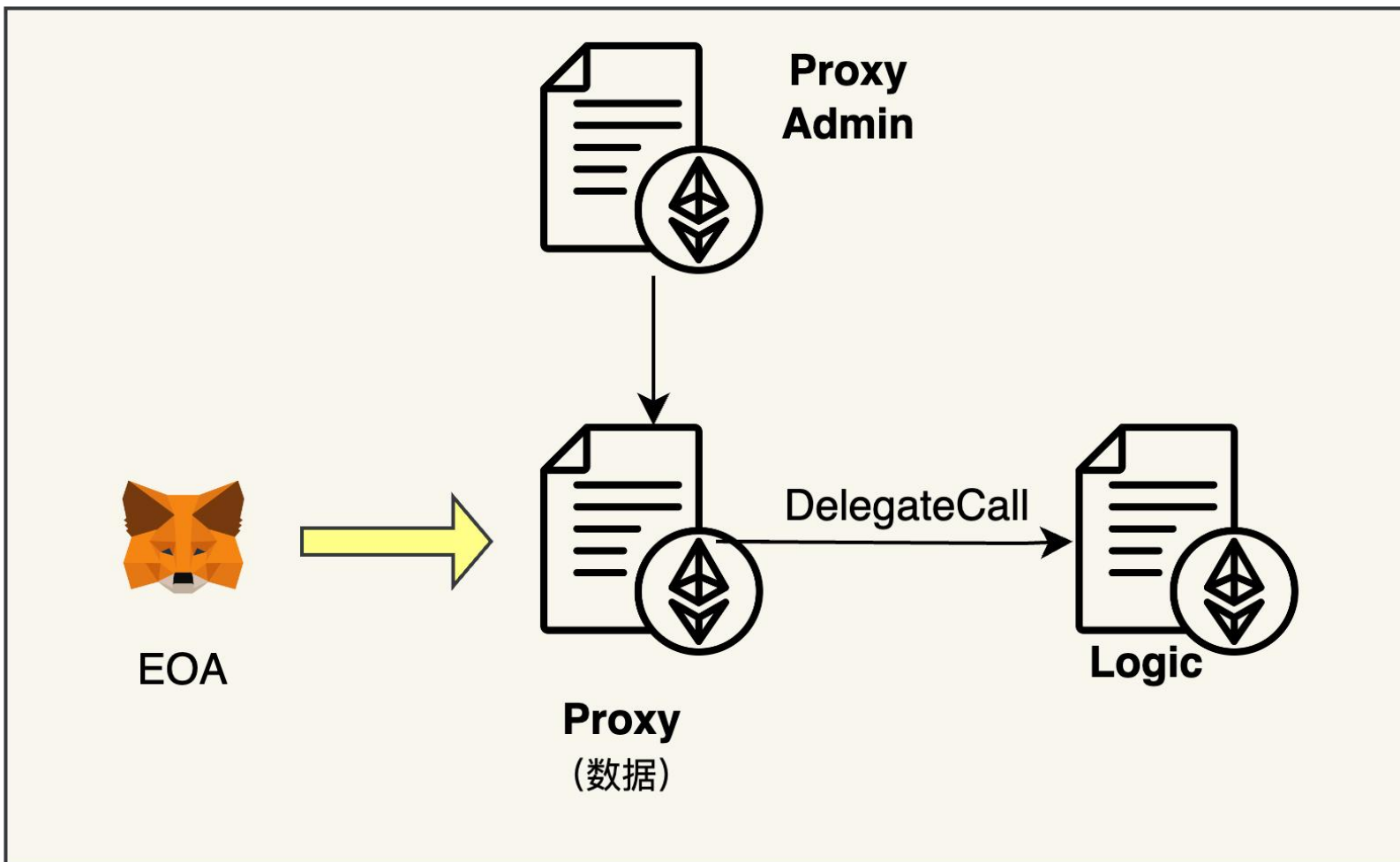
    modifier ifAdmin() {
        if (msg.sender == admin()) {
            _;
        } else {
            _delegate();
        }
    }

    constructor() public {
        address admin = msg.sender;
        bytes32 slot = _ADMIN_SLOT;
        assembly {
            sstore(slot, admin)
        }
    }

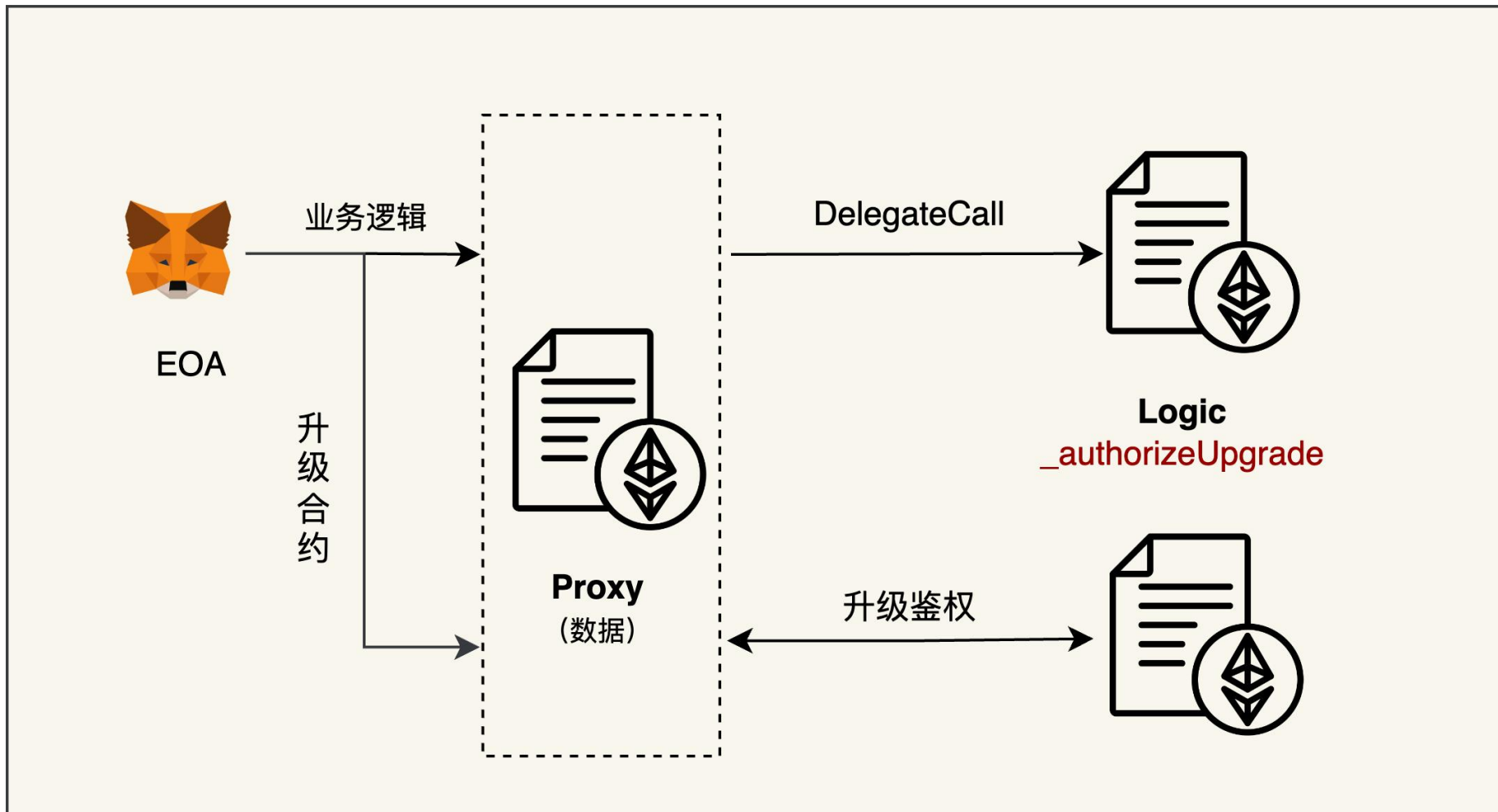
    function implementation() public ifAdmin returns (address) {
        return _implementation();
    }

    function _implementation() internal view returns (address impl) {
        bytes32 slot = _IMPLEMENTATION_SLOT;
        assembly {
            impl := sload(slot)
        }
    }
}
```

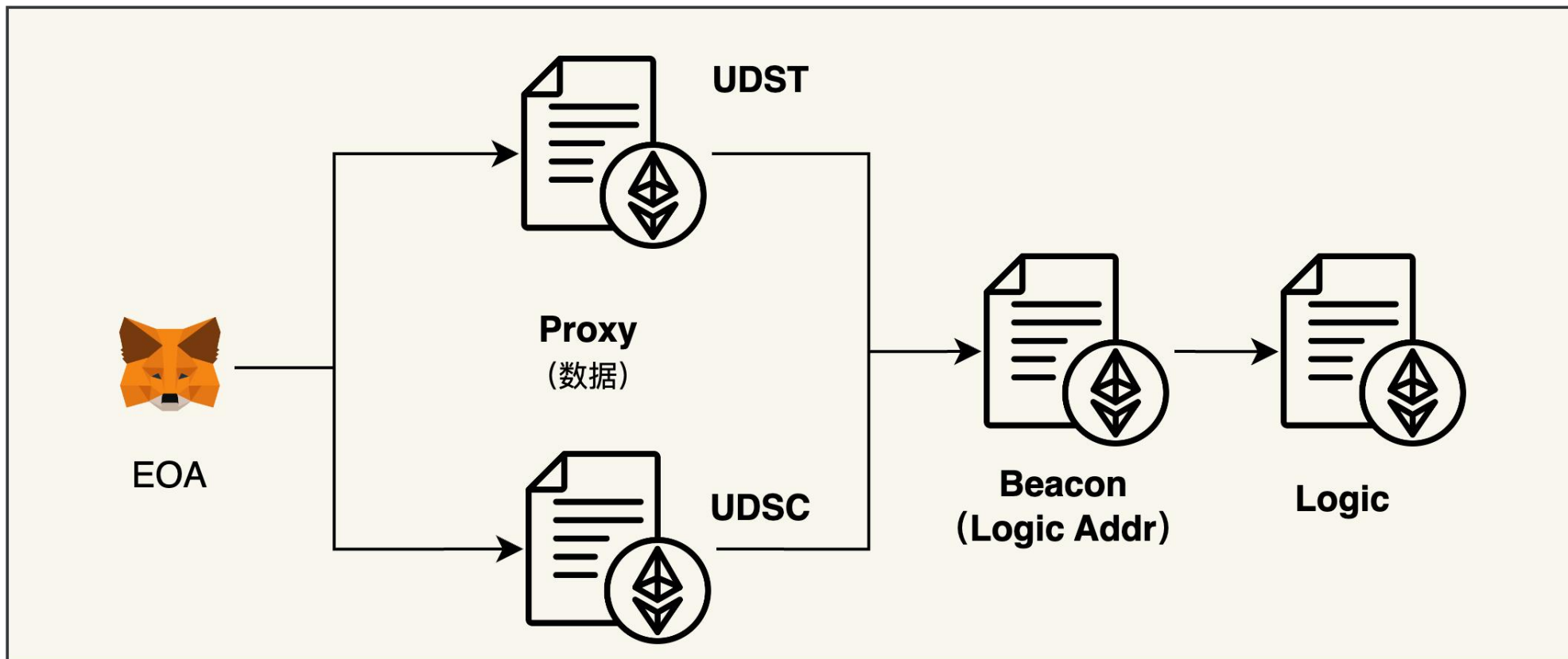
05透明代理



05UUPS



05信标代理



06代码演示

```
pragma solidity ^0.8.24;

import "@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol";
import "@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol";
import "@openzeppelin/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol";
import "@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol";

contract Example is Initializable, OwnableUpgradeable, UUPSUpgradeable {
    string public name;
    string public symbol;
    ////////// SYSTEM //////////
    function initialize(string memory _name, string memory _symbol) public initializer {
        __Ownable_init(msg.sender);
        name = _name;
        symbol = _symbol;
    }

    ///////////USER ACTION//////////
    function setName(string memory name) public {
        name = name;
    }

    function setSymbol(string memory symbol) public {
        symbol = symbol;
    }

    ////////// UPGRADE //////////
    function _authorizeUpgrade(address newImplementation) internal onlyOwner override {
    }
}
```

```
contract ExampleV1 is Initializable, OwnableUpgradeable, UUPSUpgradeable {
    string public name;
    string public symbol;
    mapping(address account => uint256) private _balances;
    ////////// SYSTEM //////////
    function initialize(string memory _name, string memory _symbol) public initializer {
        __Ownable_init(msg.sender);
        name = _name;
        symbol = _symbol;
    }

    ///////////USER ACTION//////////
    function setName(string memory name) public {
        name = name;
    }

    function setSymbol(string memory symbol) public {
        symbol = symbol;
    }

    function balanceOf(address account) external view returns (uint256){
        return _balances[account];
    }

    function transfer(address to, uint256 value) external returns (bool){
        require(to != address(0), "transfer to the zero address");
        require(value > 0, "transfer value must be greater than zero");
        require(_balances[msg.sender] ≥ value, "transfer amount exceeds balance");
        _balances[msg.sender] -= value;
        _balances[to] += value;
        return true;
    }

    ////////// UPGRADE //////////
    function _authorizeUpgrade(address newImplementation) internal onlyOwner override {
    }
}
```




感谢观看

汇报人：张伟

