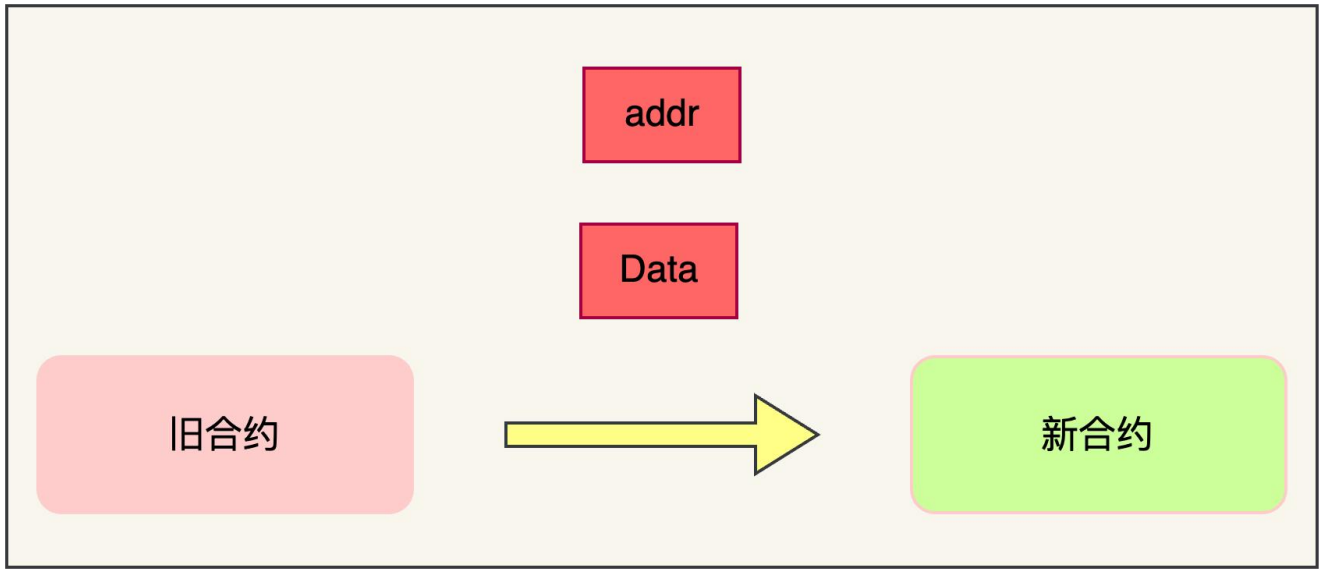
Solidity合约升级

目录

01概述

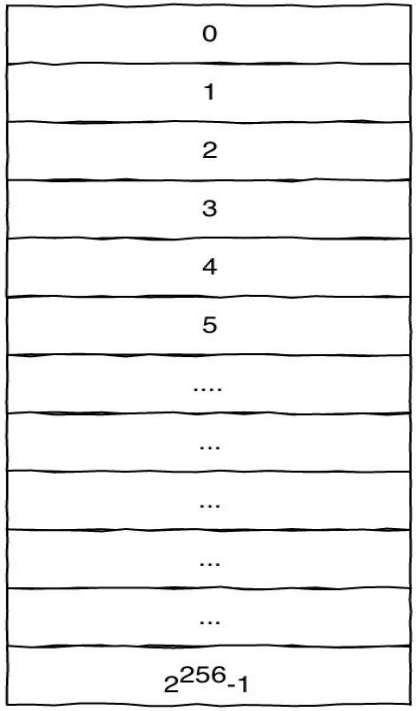
- Code Is Law
  - 合约不可更改
- 合约升级
  - 逻辑更改

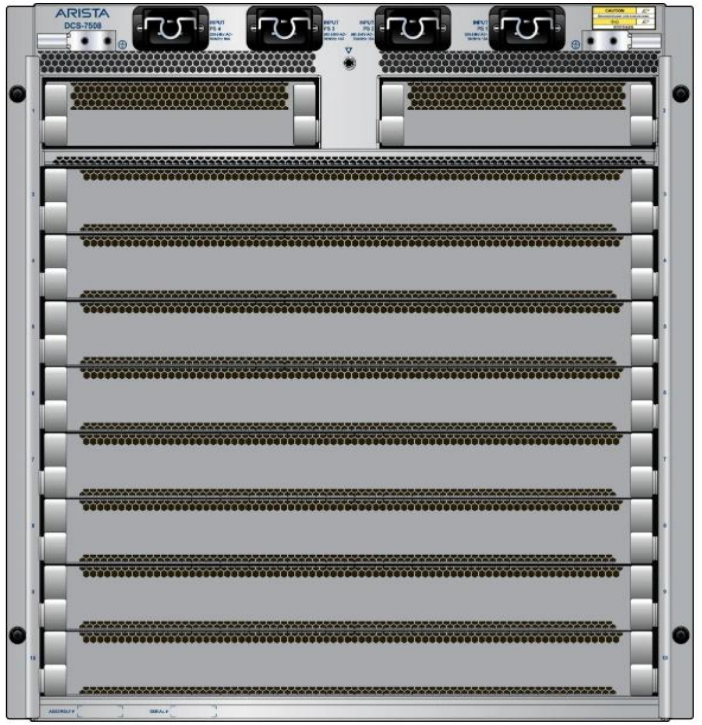


02合约存储

2 ^ 256 *32 字节

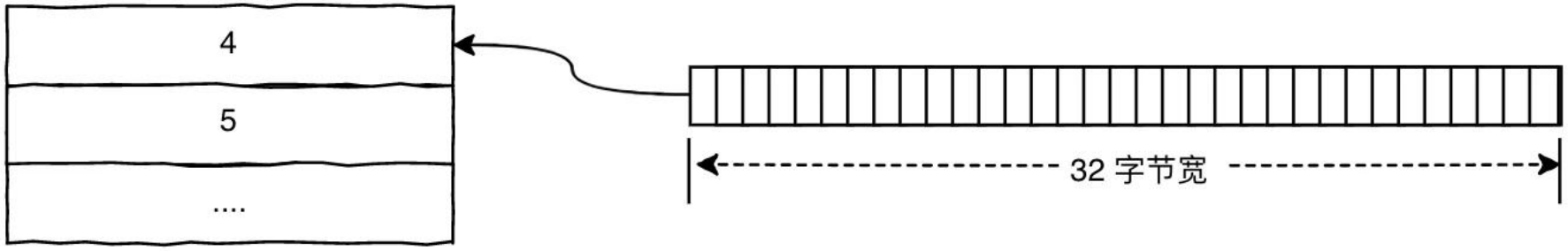
2^32次方≈40 亿





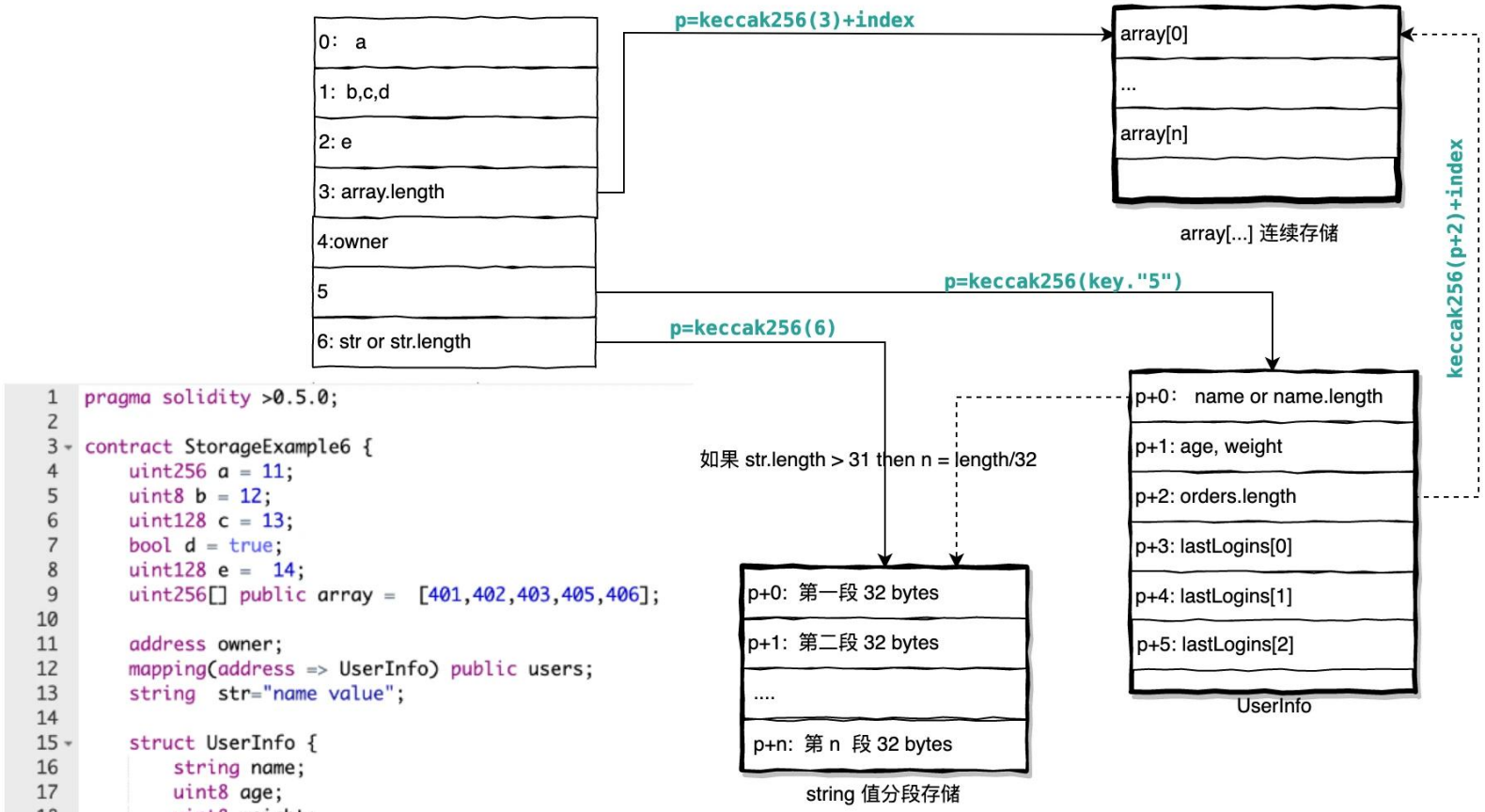
图： 插槽式数组存储

以太坊技术与实现



图： 每个插槽 32 字节宽

02合约存储

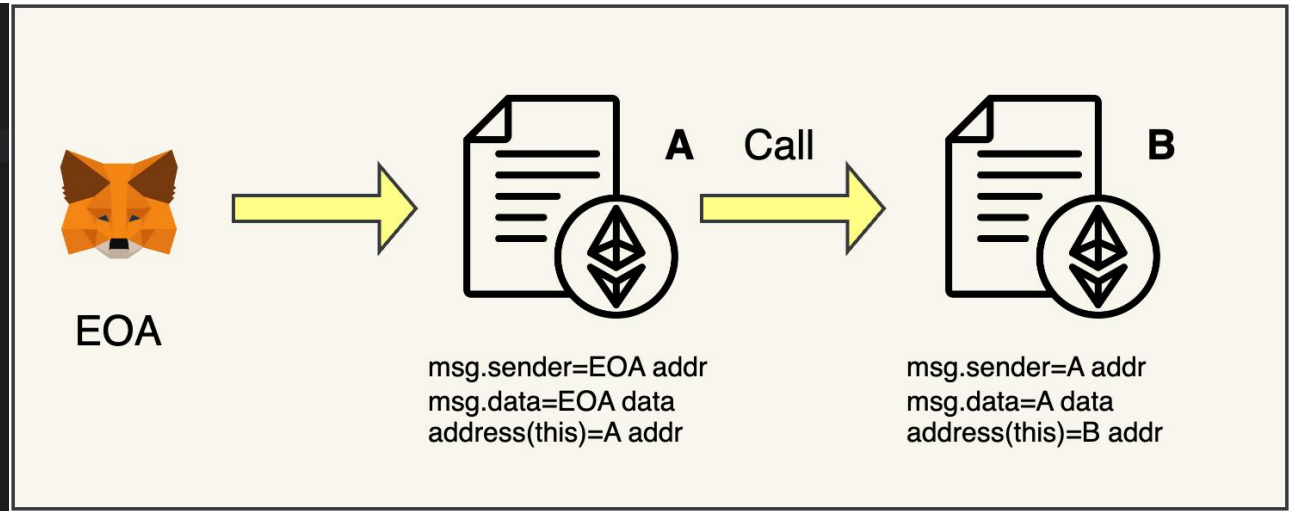


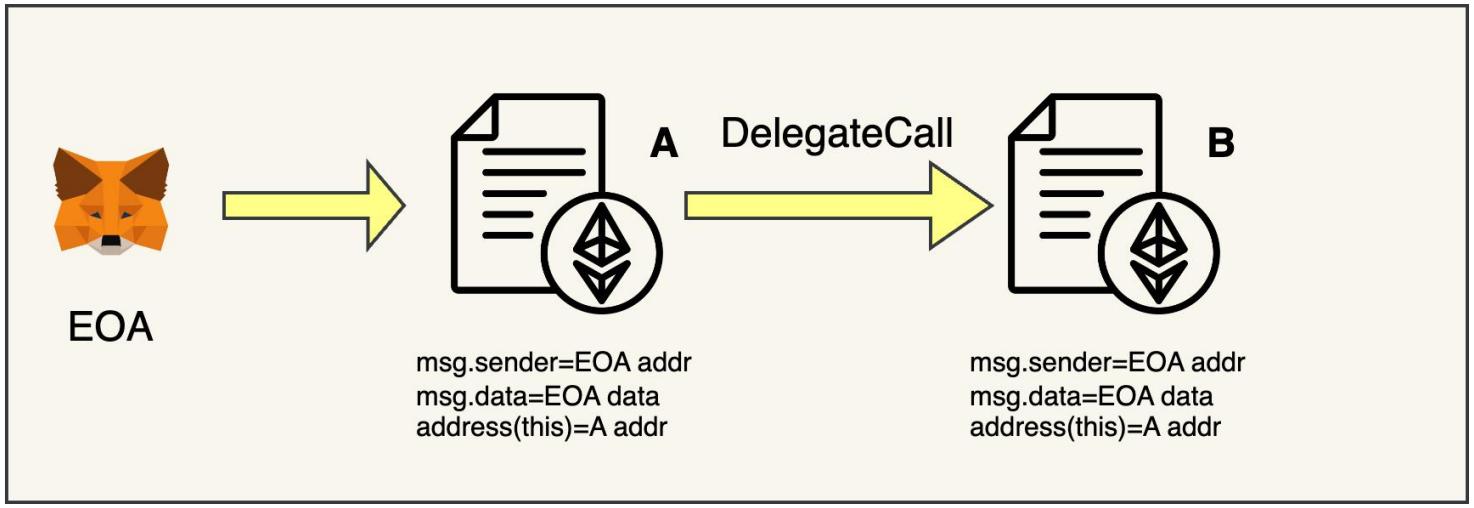

```solidity
pragma solidity >0.5.0;

contract StorageExample6 {
    uint256 a = 11;
    uint8 b = 12;
    uint128 c = 13;
    bool d = true;
    uint128 e =  14;
    uint256[] public array =  [401,402,403,405,406];

    address owner;
    mapping(address => UserInfo) public users;
    string  str="name value";

    struct UserInfo {
        string name;
        uint8 age;
        uint8 weight;
        uint256[] orders;
        uint64[3] lastLogins;
    }

```

图： 一个合约的存储布局

03Call和DelegateCall

```
12
13      function callFoo(uint256 _alice, uint256 _bob) external {
14          (bool success, bytes memory data) =
15                              b.call(abi.encodeWithSignature("foo(uint256,uint256)",
16                  _alice, _bob));
17          require(success, "Tx failed");
18      }
19
20      function delegateCallFoo(uint256 _alice, uint256 _bob) external {
21          (bool success, bytes memory data) =
22                              b.delegatecall(abi.encodeWithSignature("foo(uint256,uint256)",
23                  _alice, _bob));
24          require(success, "Tx failed");
25      }
26  }
```





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	...

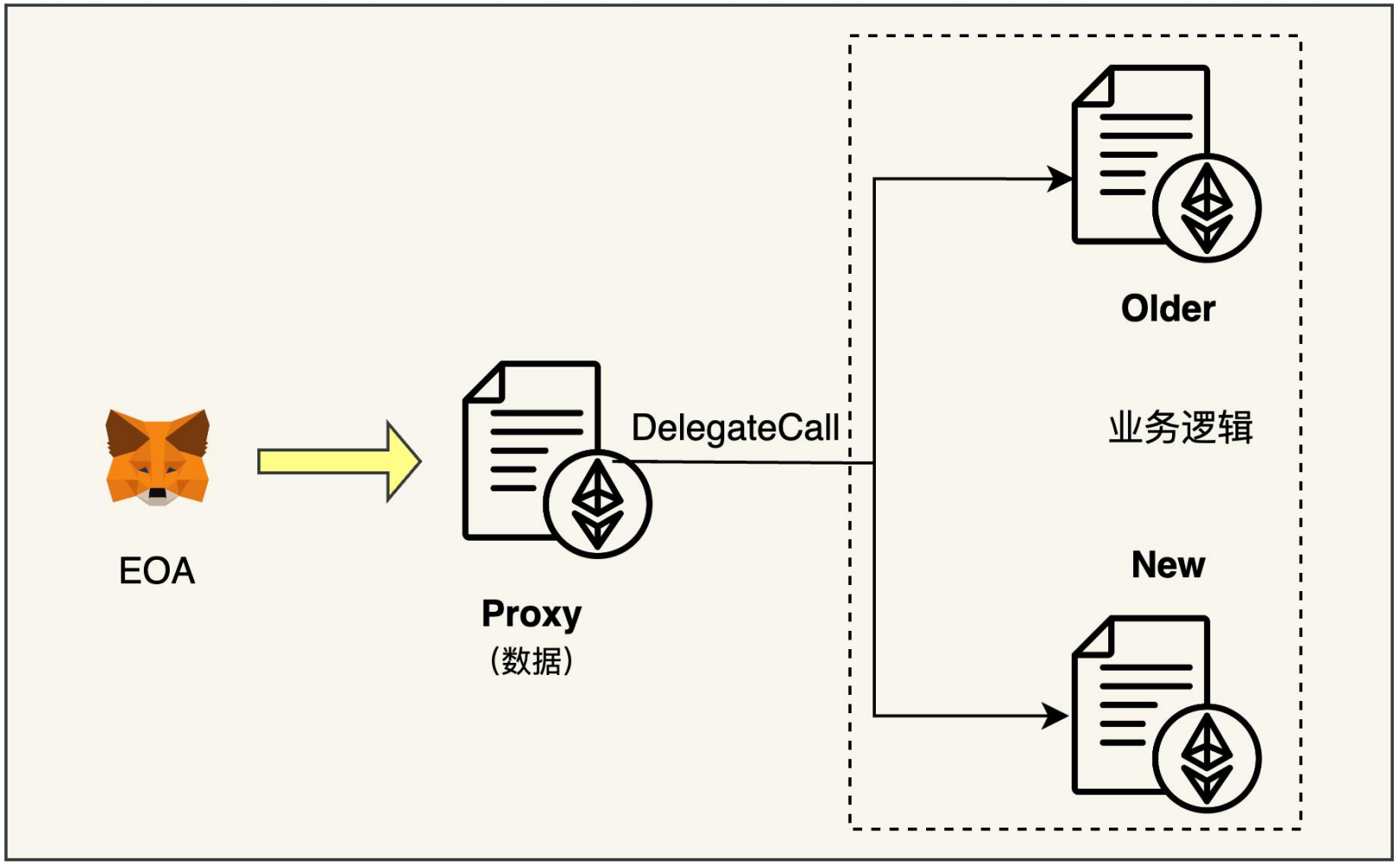
属性名称	slot	value
Alice	0	1
Bob	1	2
...	2	...
...	3	...

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



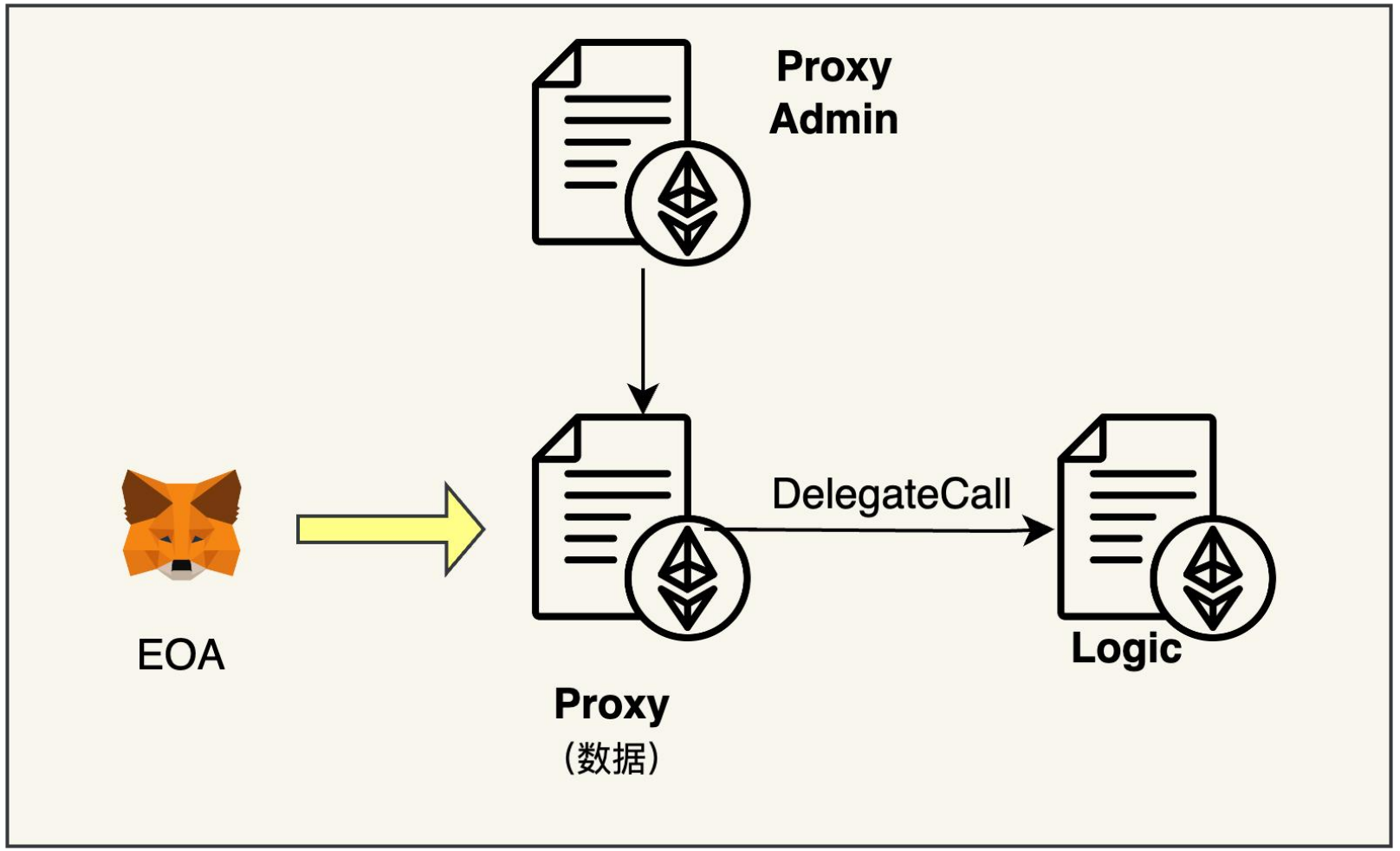
04原理和实现


```solidity
contract Proxy {
    address public implementation;
    address public admin;
    constructor() public {
        admin = msg.sender;
    }
    function setImplementation(address newImplementation) external {
        require(msg.sender == admin, "must be admin");
        implementation = newImplementation;
    }
    function changeAdmin(address newAdmin) external {
        require(msg.sender == admin, "must be admin");
        admin = newAdmin;
    }
    function _delegate(address implementation) internal virtual {
        assembly {
            calldatacopy(0, 0, calldatasize())
            let result := delegatecall(gas(), implementation, 0, calldatasize(), 0, 0)
            returndatacopy(0, 0, returndatasize())
            switch result
            case 0 {
                revert(0, returndatasize())
            }
            default {
                return (0, returndatasize())
            }
        }
    }
    function _fallback() internal virtual {
        _delegate(implementation);
    }
    fallback() external payable virtual {
        _fallback();
    }
}
```

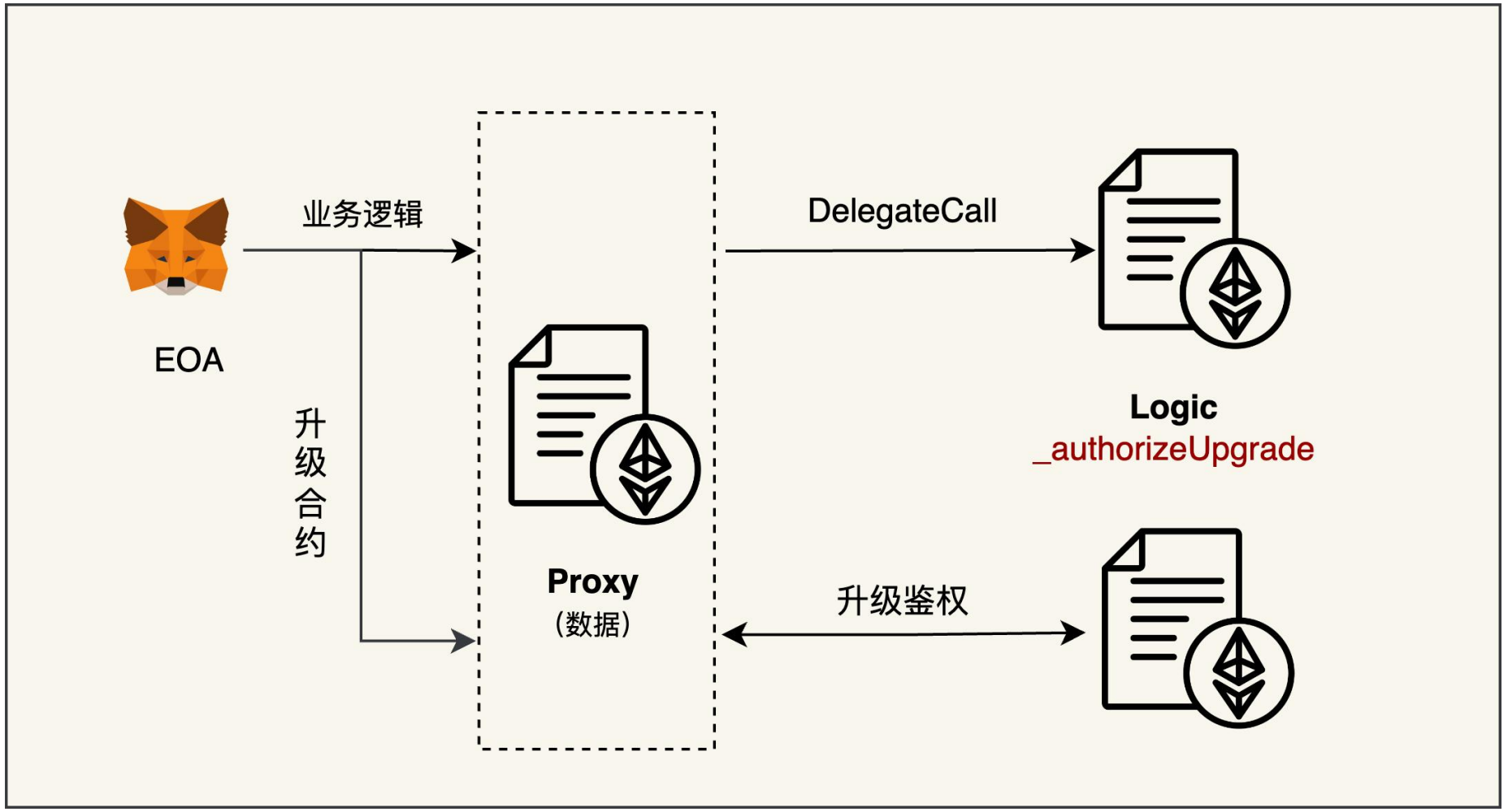
04原理和实现


```
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)
        }
    }
```

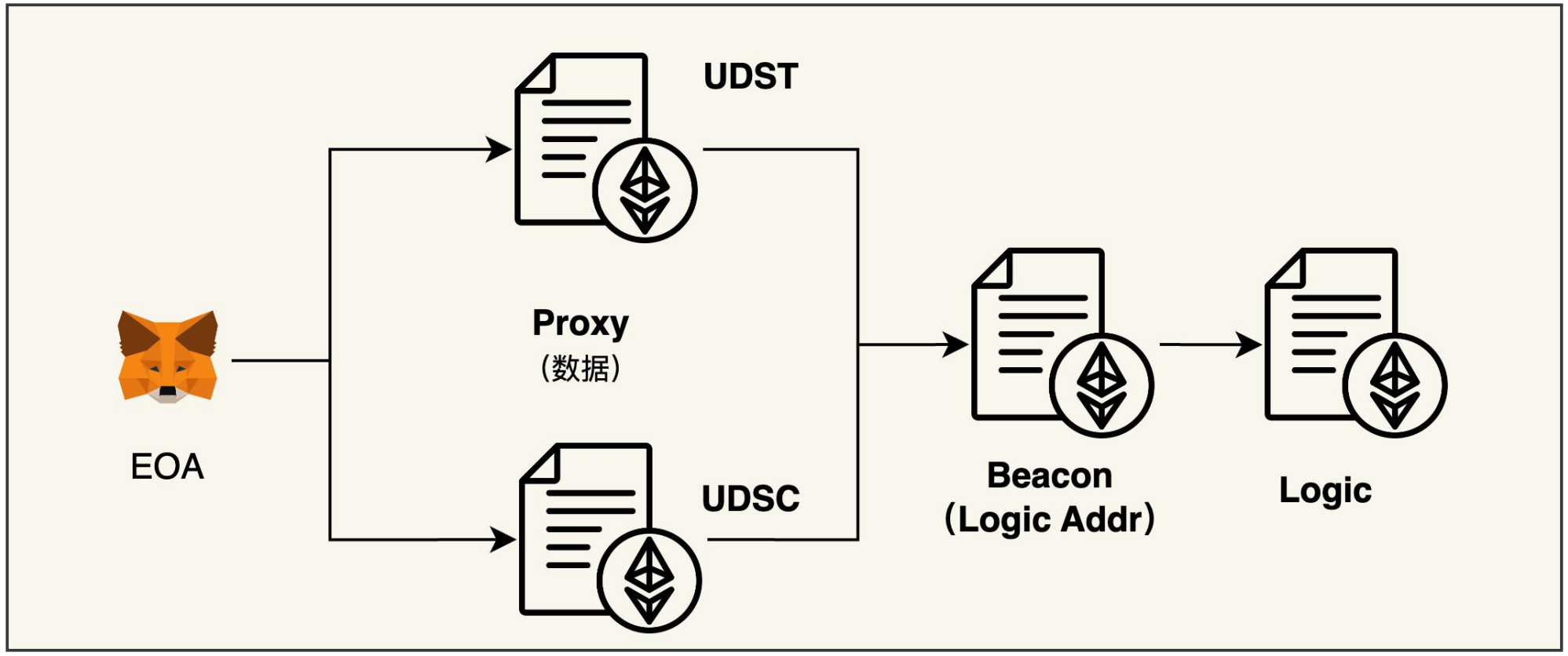
05透明代理



05UUPS


业务逻辑
DelegateCall
EOA
升级合约
Logic
_authorizeUpgrade
Proxy
（数据）
升级鉴权

05信标代理



06代码演示

```solidity
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 {
    }
}
```

```solidity
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