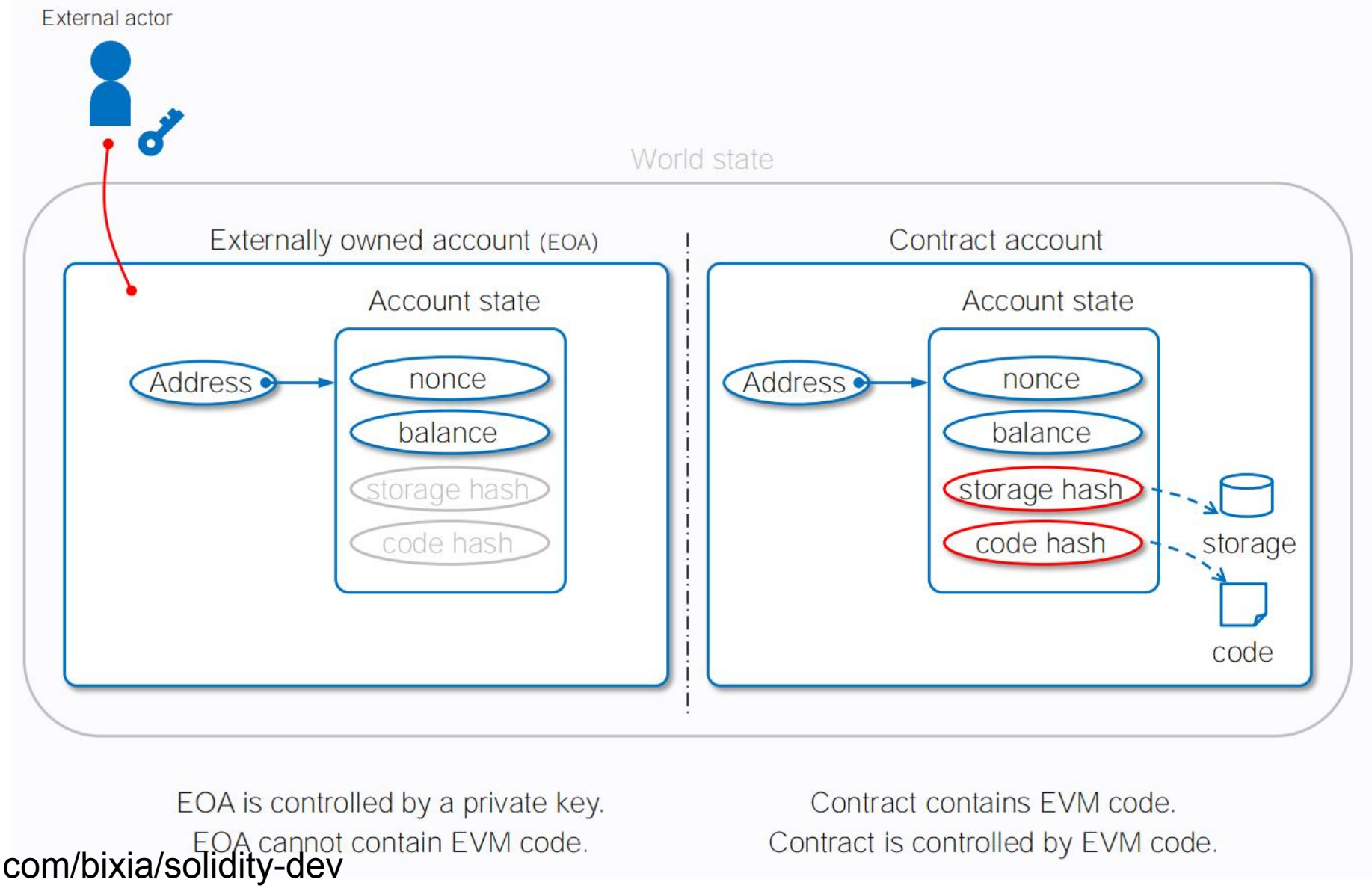
Solidity 智能合约的开发流程

开发，测试与部署

bugWriter 20220326

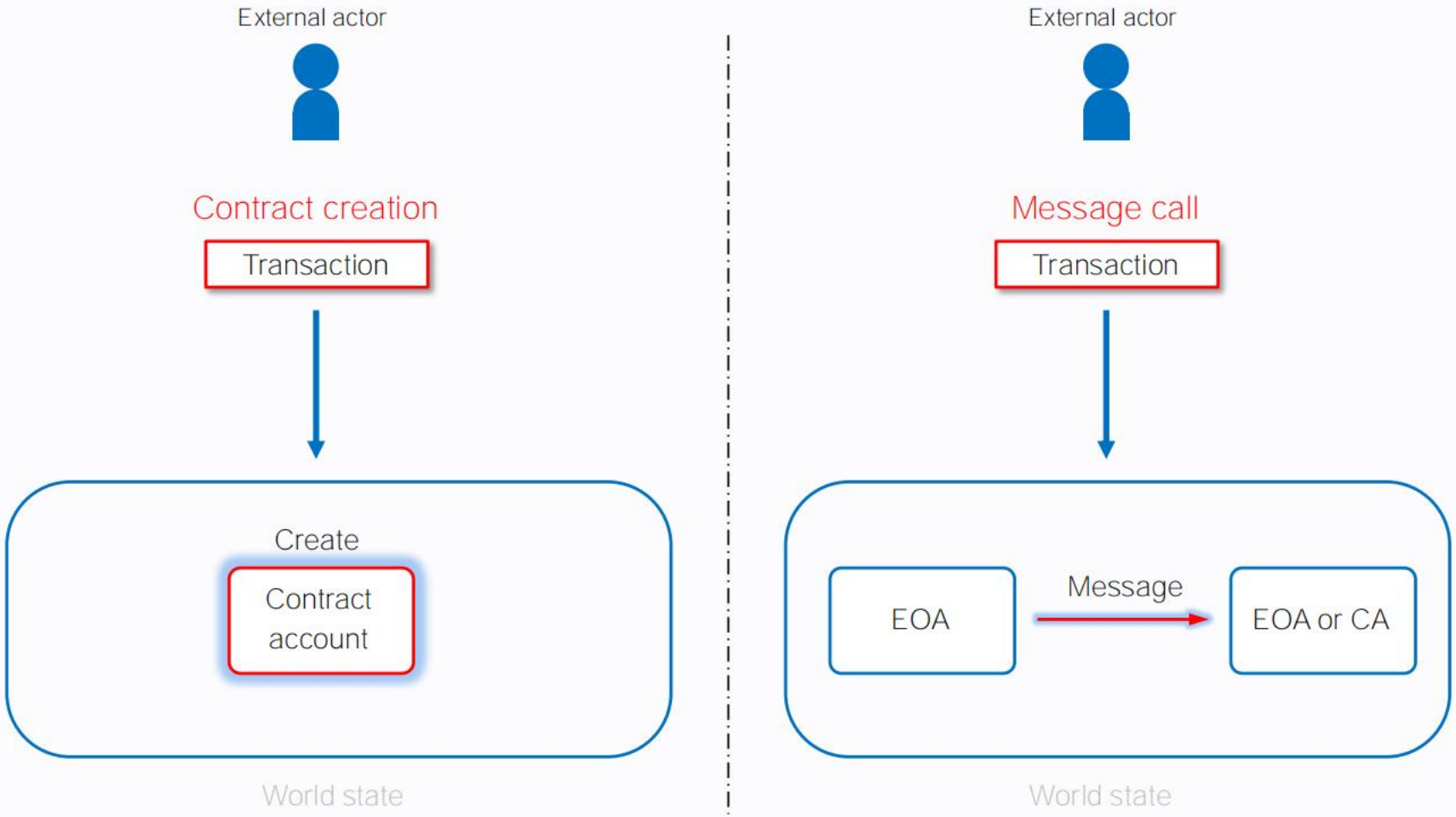
Ref: https://github.com/bixia/solidity-dev

以太坊的账户模型



Ref: https://github.com/bixia/solidity-dev

以太坊的交易



There are two practical types of transaction, contract creation and message call.

Ref: https://github.com/bixia/solidity-dev

认识Solidity智能合约

- 在EVM兼容的链上执行
- 先编译后执行
- Solidity与opcode关系

Ref: https://github.com/bixia/solidity-dev

智能合约组成部分

- SPDX-License
- pragma solidity 0.8.0;
- contract 关键字
- constructor 关键字
- public,private,external,internal
- 全局变量
- 只读函数view
- 状态更改函数

官方文档链接:
https://docs.soliditylang.org/en/v0.8.13/
中文文档链接:
https://learnblockchain.cn/docs/solidity/

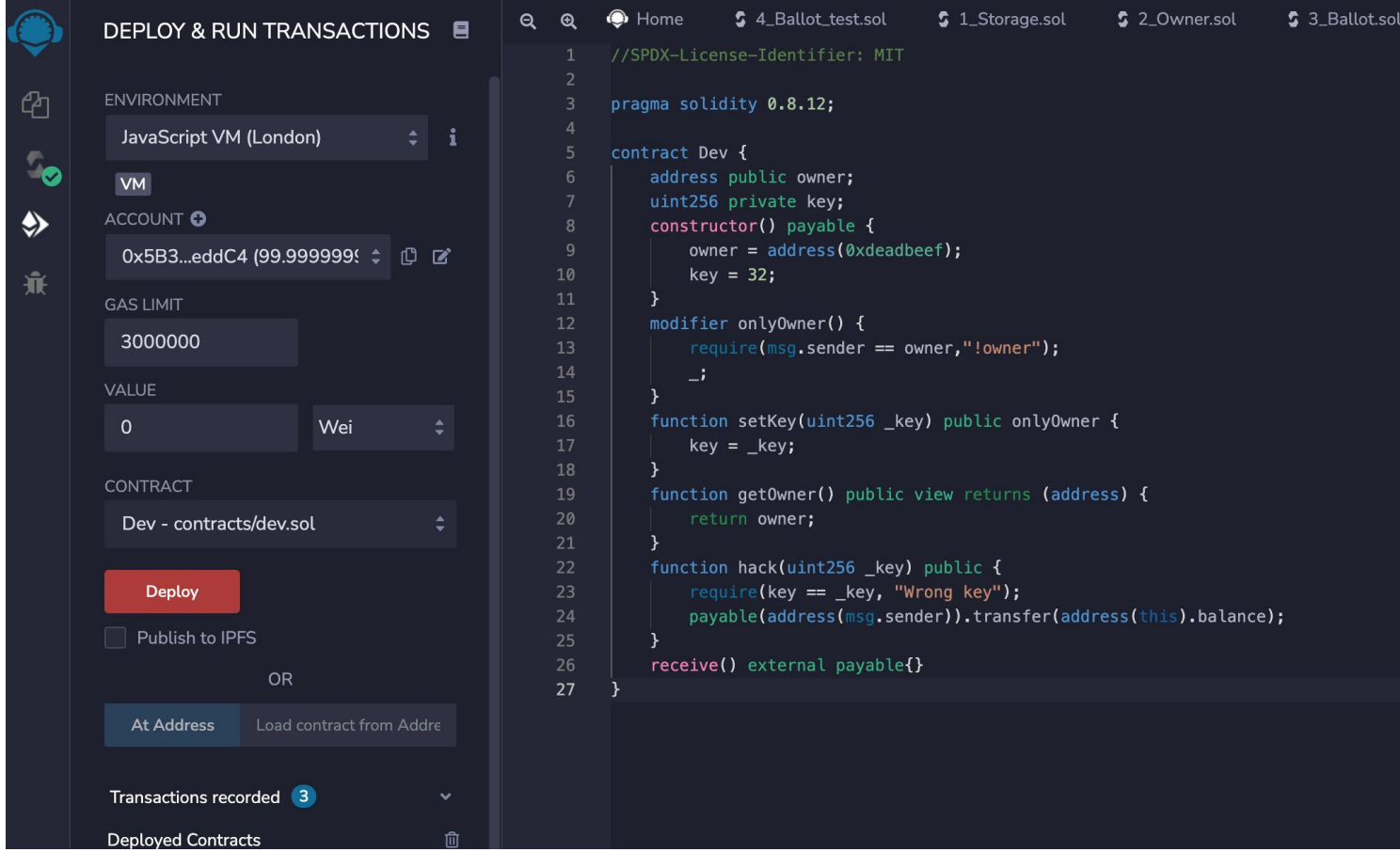
```
//SPDX-License-Identifier: MIT

pragma solidity 0.8.12;

UnitTest stub | dependencies | uml | draw.io
contract Dev {
    address public owner;
    uint256 private key;
    ftrace
    constructor() payable {
        owner = address(0xdeadbeef);
        key = 32;
    }
    modifier onlyOwner() {
        require(msg.sender == owner,"!owner");
        _;
    }
    ftrace | funcSig
    function setKey(uint256 _key) public onlyOwner {
        key = _key;
    }
    ftrace | funcSig
    function getOwner() public view returns (address) {
        return owner;
    }
    ftrace | funcSig
    function hack(uint256 _key) public {
        require(key == _key, "Wrong key");
        payable(address(msg.sender)).transfer(address(this).balance);
    }
    ftrace
    receive() external payable{}
}
```

Remix

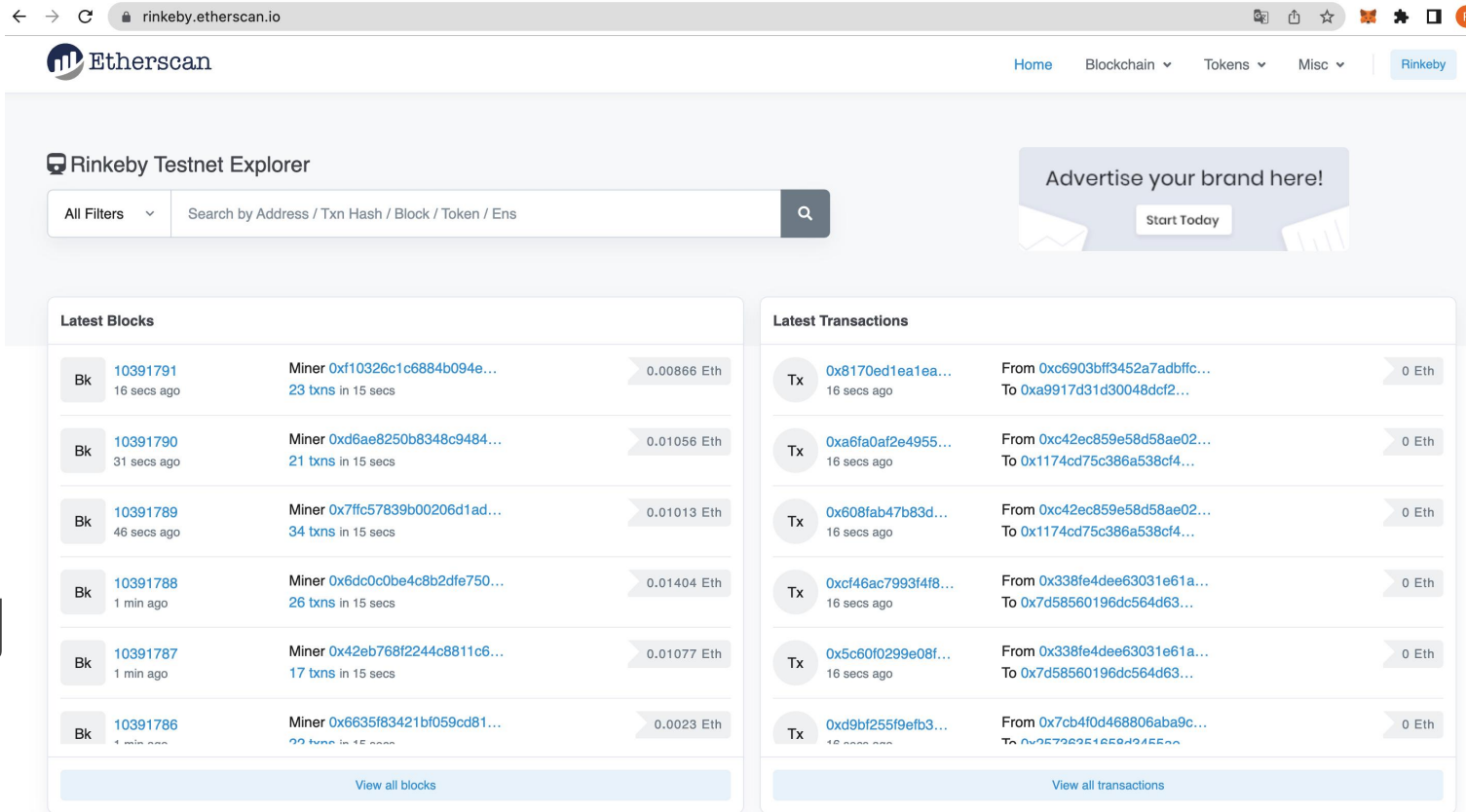
- 网页版IDE，无需环境配置
- 一键编译
- Debug模式
- 部署到本地网络
- 部署到测试网络



官方链接：https://remix.ethereum.org/

测试网络

- 获取测试币
- 开源源码
- 使用etherscan浏览器调用
  - 调用view方法
  - 调用非view方法



官方链接：
获取测试币：https://faucet.rinkeby.io/
测试网浏览器：https://rinkeby.etherscan.io/

Foundry

官方链接：
foundry的官方仓库：https://github.com/gakonst/foundry
foundry的官方文档：https://onbjerg.github.io/foundry-book/config/vscode.html

- 现代化Solidity开发框架
- 下载
- 安装
- 编译
- 测试

```
curl -L https://foundry.paradigm.xyz | bash
```

```
foundryup
```

```
$ forge init hello_foundry
```

```
$ forge build
compiling...
success.
```

```
$ forge test
compiling...
no files changed, compilation skipped.
Running 1 test for ContractTest.json:ContractTest
[PASS] testExample() (gas: 254)
```

Foundry fork主网

- RPC节点
- 区块高度
- 在foundry中重放一笔交易

```
//SPDX-License-Identifier: MIT
pragma solidity 0.8.12;

import "ds-test/test.sol";
import "forge-std/Vm.sol";

//forge test --match-contract ForkTest --fork-url https://eth-mainnet.alchemyapi.io/v2/7Brn0mxZn
/// https://etherscan.io/tx/0x5cfed0b8664dfbc7bc74d83e4d425ce47ebf13e964c50b51e928f4a88c0b7577

UnitTest stub | dependencies | uml | draw.io
contract ForkTest is DSTest {
    Vm public vm = Vm(HEVM_ADDRESS);
    address EOA = 0x108F0eC0D7F05490b7Fa284de5F1D4b1dfb64Dbf;
    address uniswapV3Router = 0x68b3465833fb72A70ecDF485E0e4C7bD8665Fc45;
    address weth = 0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2;
    address token = 0xF1cA9cb74685755965c7458528A36934Df52A3EF;
    address pool = 0x27a9ff745cf1Dd366d94267Cb4aDE2350588a187;

    ftrace | funcSig
    function setUp() public {
        vm.label(EOA, "EOA");
        vm.label(uniswapV3Router, "uniswapV3Router");
        vm.label(weth, "weth");
        vm.label(token, "token");
        vm.label(pool, "pool");
    }
    ftrace | funcSig
    function test_Reply() public {
        vm.startPrank(EOA);
        address(uniswapV3Router).call(hex"5ae401dc0000000000000000000000000000000000000000000000000000
        vm.stopPrank();
    }
}
```

Ref: https://github.com/bixia/solidity-dev