几种创建合约的方法

登链社区 - Tiny熊

将介绍

- 使用 Remix/Foundry 创建的合约地址是如何生成的?
- 如何合约部署到一个确定的地址上
- 大规模部署相同功能的合约如何节省 Gas

创建合约 – 复习

- 外部部署（Remix / Hardhat /Foundry） / Ethers.js / viem.js / Web3.js
- 合约使用 New 关键字： new Counter()
- 在 EVM 中，以上是通过 create 操作码创建的 （https://www.evm.codes/）

```
function createContract1() public returns (address) {
    C  c = new C();
    return address(c);
}
```

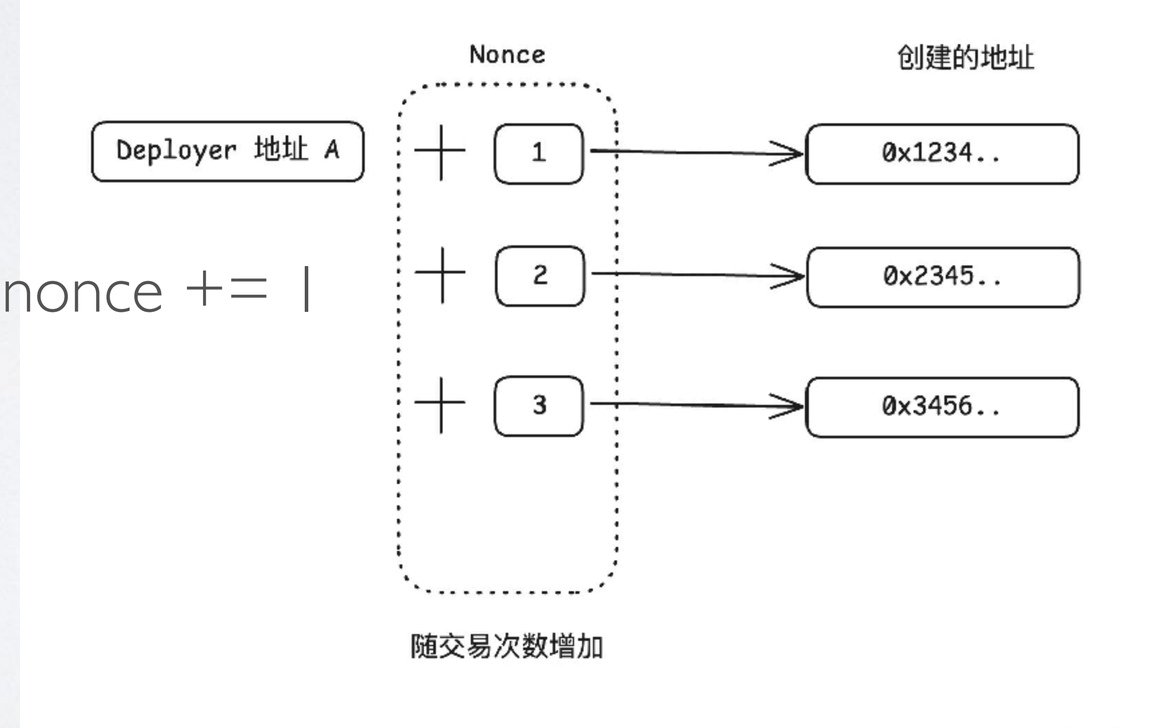
CreateC.sol

CREATE – 地址计算

- 根据创建者（sender）的地址以及创建者发送过的交易数量（nonce）来计算确定
  - keccak256(rlp.encode([normalize_address(sender), **nonce**]))[12:]
  - 不易控制合约地址，多链要保持一致的合约地址麻烦， 适合于不需要预测的情况

EOA： TX nonce += 1

合约工厂：每创建一个合约 nonce += 1



CREATE2 创建合约

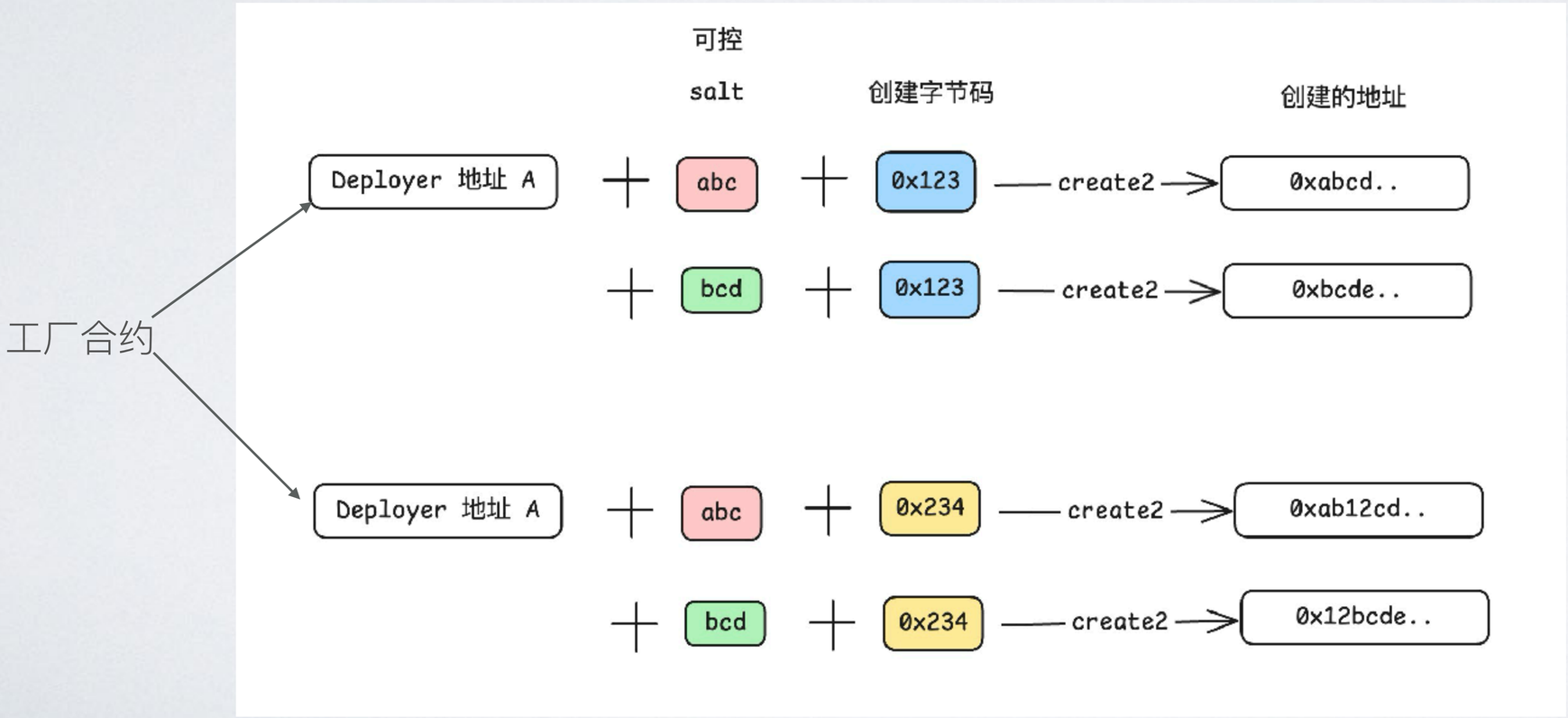
- Create2 使用 salt 控制合约地址， 确定性的创建合约
- Create2 使用方法
    - C c = new C{ salt: _salt }();

```
function create2Contract(uint _salt) public returns (address) {
    C c = new C{salt: keccak256(abi.encode(_salt))}();
    return address(c);
}
```

CreateC.sol

CREATE2 – 地址计算

- 根据创建者（工厂合约地址）的地址 + 用户设置的salt及要创建的合约自己码计算确定
  - keccak256(0xff ++ sender ++ **salt** ++ keccak256(**init_code**))[12:]

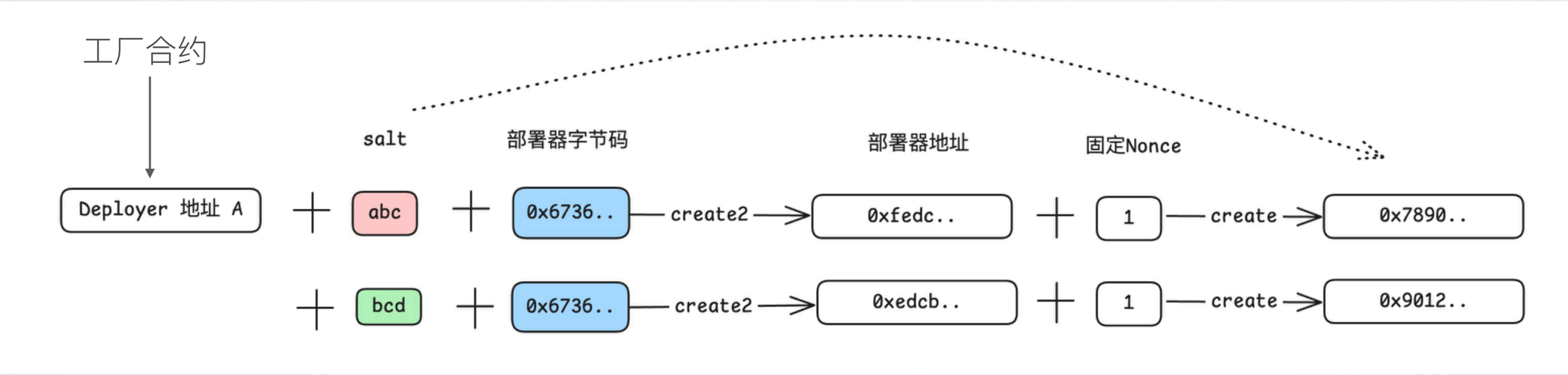


CREATE2 思考

- 用一样的 salt 再次创建会如何?
- 提前确定地址有什么好处或使用场景
  - 流动性池、合约钱包等
  - 多链一致

CREATE3 库

- 创建 不与特定合约字节码绑定的确定性合约地址
  - 场景：更容易实现多链地址一致性
  - 在Cancun 升级之前，可在一个地址上部署不同的的合约代码?
- 原理：工厂先用 create2 部署一个部署器，再用部署器 create 方式部署用户合约
  - 参考实现：https://github.com/transmissions11/solmate/blob/main/src/utils/CREATE3.sol



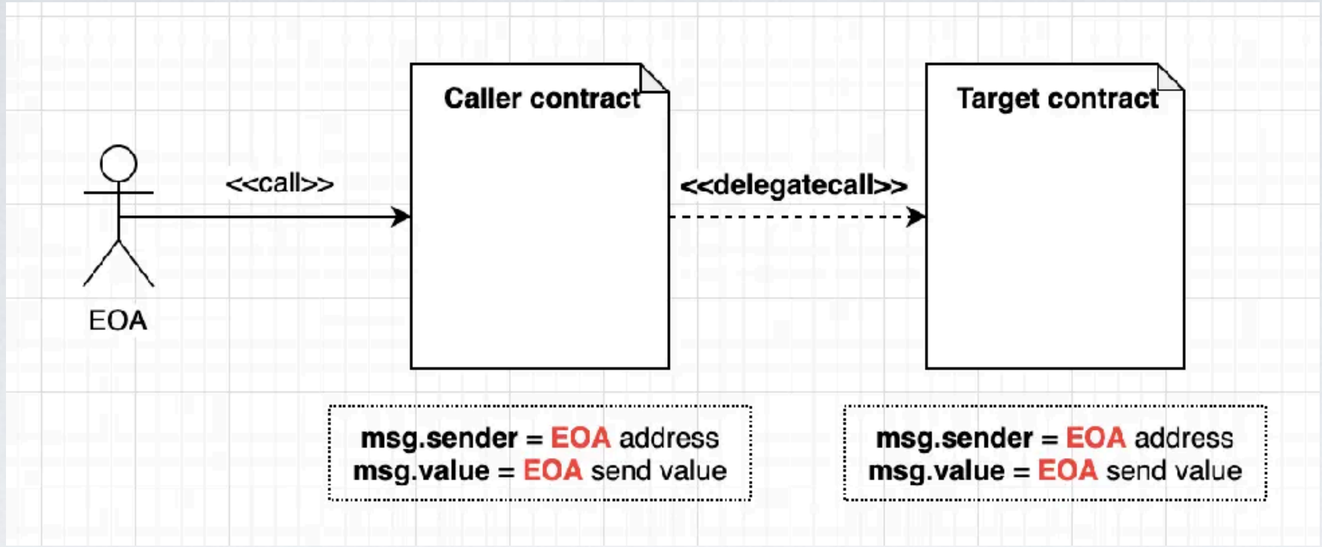
DEMO

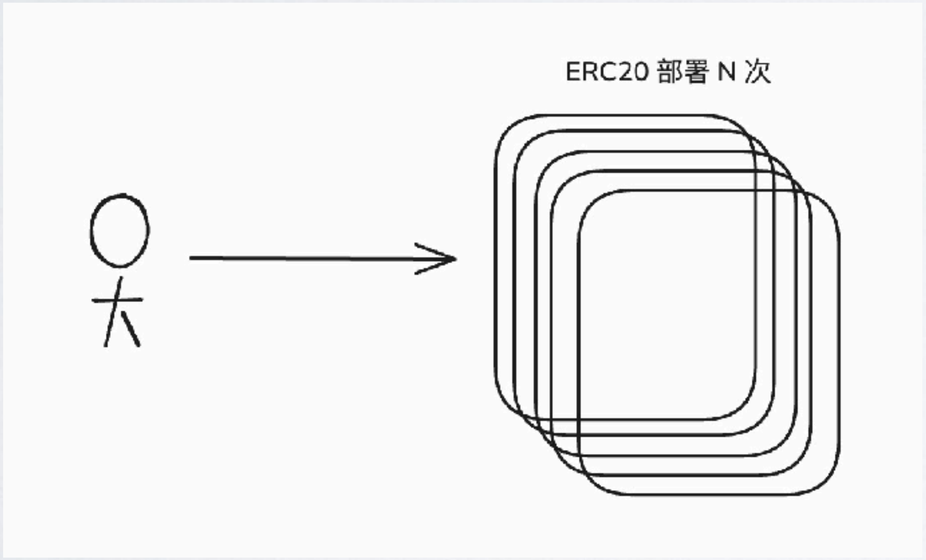
CreateC.sol

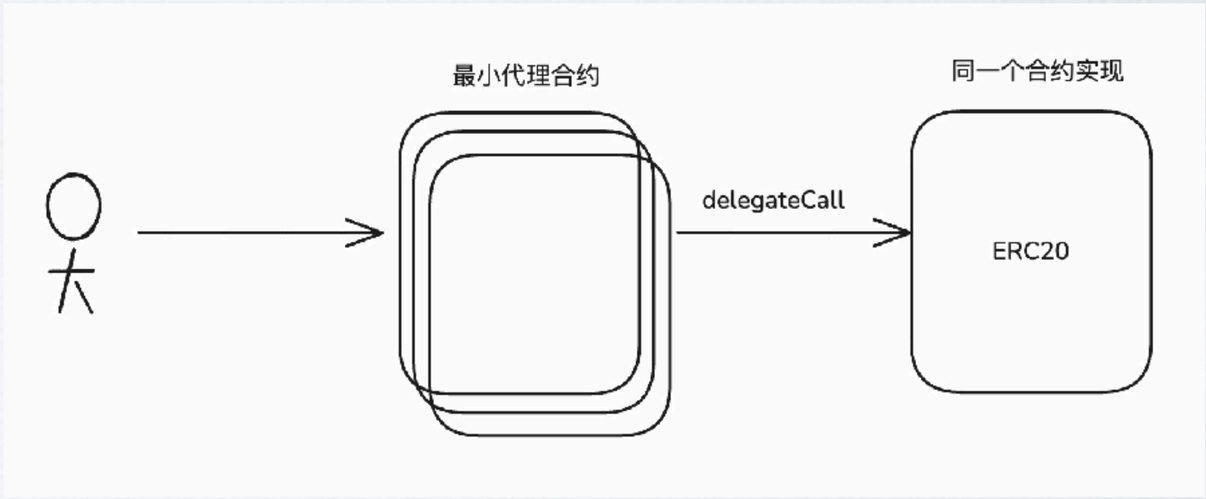
```
function create3Contract(uint _salt, bytes memory creationCode) external returns (address deployed) {
        deployed = CREATE3.deploy(
            keccak256(abi.encode(_salt)),
            creationCode,
            0
    );
    }
```

最小代理 创建合约

- delegatecall 委托调用
  - 多个代理可以委托给同一个实现合约的
- 最小代理：
  - 代理仅保留最小的功能（仅有代理功能本身），所有的逻辑都委托给实现合约（复用实现代码）。
  - 最小代理提案：https://eips.ethereum.org/EIPS/eip-1167
  - 实现：https://github.com/optionality/clone-factory







testCall_Delegate.sol

DEMO

CreateC.sol

```
function createMinContract(address impl) public returns (address) {
    return createClone(impl);
}
```

小结

- create 简单直接
- Create2 可预测的合约地址
- Create3 可预测的合约地址、不依赖部署合约的字节码（gas 高）
- 最小代理：大量重复部署节约 gas

练习题

- 实现一个工厂合约，在以太坊上用 ERC20 模拟 meme 铸造（用最小代理）：
  - 方法 1：deployInscription(string symbol, uint totalSupply, uint perMint)
  - 方法 2：mintInscription(address tokenAddr)

https://decert.me/quests/75782f22-edb8-4e82-9b68-0a4f46fcaadd

练习题（不用写）

- 在以太坊上用 ERC20 模拟铭文铸造，创建可升级的工厂合约：
  - 方法1：deployInscription(string symbol, uint totalSupply, uint perMint)
  - 方法 2：mintInscription(address tokenAddr)
  - 第 2 个版本加入铸造费用（price）

https://decert.me/quests/ac607bb0-53b5-421f-a9df-f3db4a1495f2