OpenZeppelin

How to use OpenZeppelin Contracts Account Abstract framework

Ernesto García

Agenda

- Introduction to Account Abstraction framework
- Repository Setup
- ECDSA Account
- Replayability protection
- Bonus: Factory Setup

Introducing OpenZeppelin

Account Abstraction framework

OpenZeppelin

OpenZeppelin Comunity Contracts

An extension of OpenZeppelin Contracts to host experimental implementations and rapidly changing ERCs.

Community Contracts repository

Goals of our framework

Secure

Our go-to recommendation is a **solid base layer** for developers who're writing their own Solidity implementations of accounts.

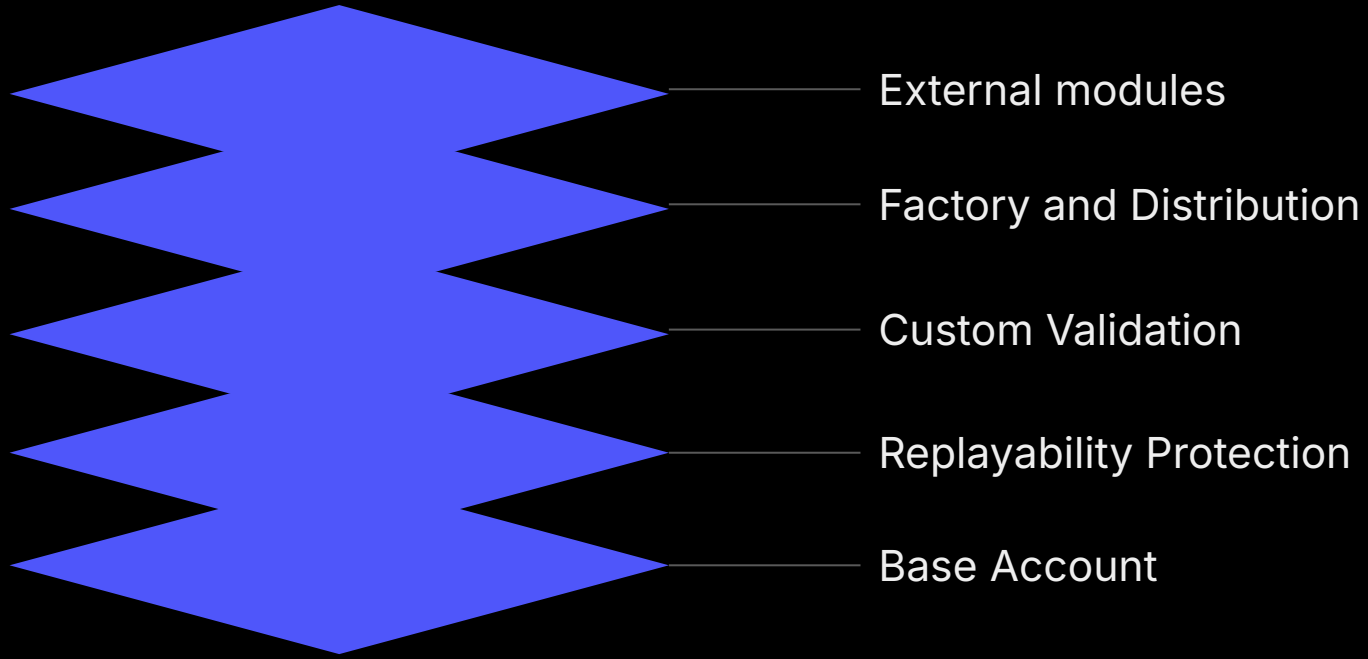
Layered

To accomodate developers building with our libraries, the Account contracts **must be consumed in layers.**

Extensible

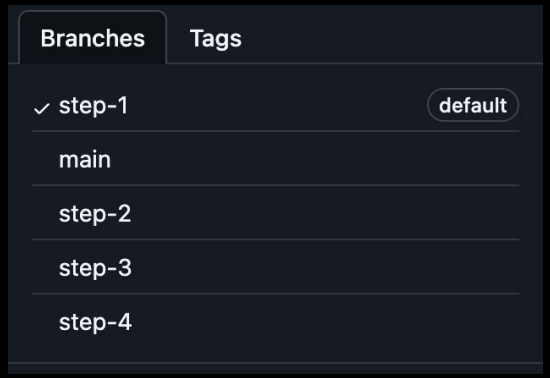
Modularity has become a **massive source of innovation** that developers should have easy access to.

How does it work?



Let's setup

```
git clone https://github.com/ernestognw/account-abstraction-workshop
cd account-abstraction-workshop
forge install
```



Workshop Repository

Step 1

Repository Setup

What's in here?

The basic Foundry template with the following 2 dependencies:

1. OpenZeppelin Contracts
2. OpenZeppelin Community Contracts

Step 2

ECDSA Account

Filling custom validation

```solidity
abstract contract AccountBase is IAccount, IAccountExecute {
    // ...

    /**
     * @dev Validation logic for {validateUserOp}.
     *
     * IMPORTANT: Implementing a mechanism to validate user operations is a security-sensitive operation
     * as it may allow an attacker to bypass the account's security measures. Check out {AccountECDSA},
     * {AccountP256}, or {AccountRSA} for digital signature validation implementations.
     */
    function _validateUserOp(
        PackedUserOperation calldata userOp,
        bytes32 userOpHash
    ) internal virtual returns (uint256 validationData);

    // ...
}
```

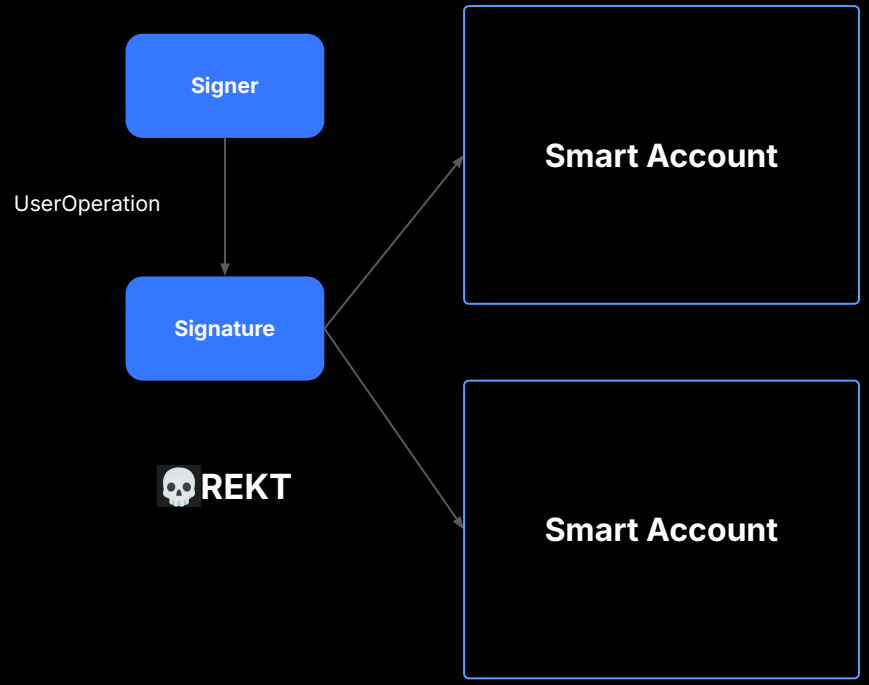
Step 3

Protecting against replayability

Replayability across same-key accounts

Given 2 accounts controlled by the same private key, **an user operation could be replayed*** on the other account unless the signature is tied to the **contract address** and **chain id.**

Best way to do this is with EIP-712.

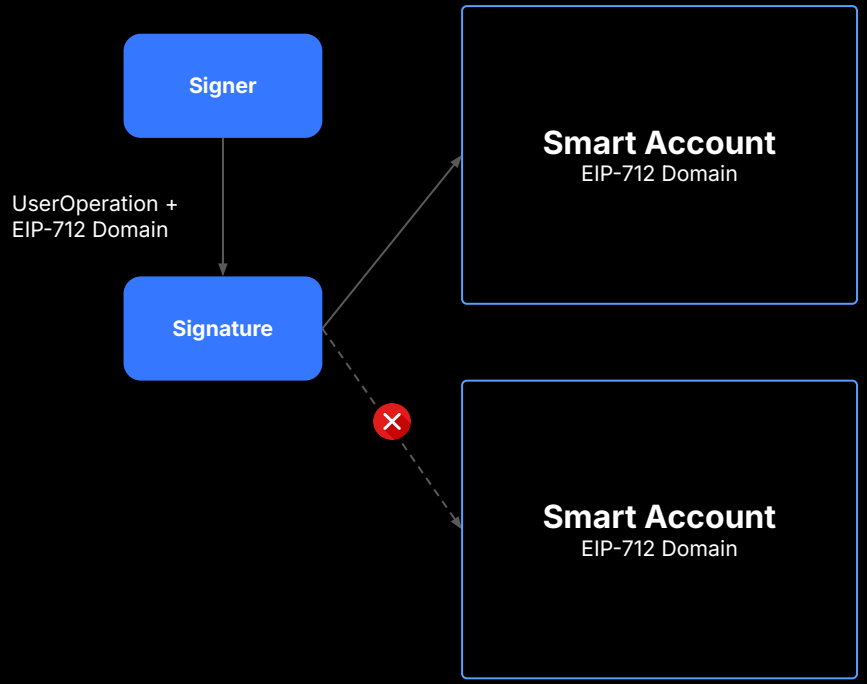


*Issue initially discovered by *curiousapple.eth*

Replayability across same-key accounts

Given 2 accounts controlled by the same private key, **an user operation could be replayed*** on the other account unless the signature is tied to the **contract address** and **chain id.**

Best way to do this is with EIP-712.



Issue initially discovered by* ***curiousapple.eth*

Step 4

Bonus: Creating a factory

Thank You

Ernesto García
@ernestognw

OpenZeppelin

We're hiring! 👋