

# CrossSpaceCall

CrossSpaceCall    conflux   coreSpace   ESpace                    coreSpace           ESpace

- 1. conflux   CoreSpace   ESpace                    conflux
- 2. CoreSpace           ESpace

- Conflux

# createEVM

espace                    call

call    createEVM   nce+1                    call

- init

- 

# transferEVM

PoW      Espace

- conflux

- to: PoW      Espace

- 

# callEVM

ESpace                      call

- to: PoW      Espace

- data: encode

•

## staticCallEVM

ESpace

- to: CroeSpace ESpace
- data: encode

•

## withdrawFromMapped

ESpace

CoreSpace

- value:

# mappedBalance

CoreSpace      CoreSpace

- address: CoreSpace

- 

# mappedNonce

CoreSpace      CoreSpace    nonce

- address: CoreSpace

- nonce

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.5.0;
```

```
interface CrossSpaceCall {

    event Call(bytes20 indexed sender, bytes20 indexed receiver, uint256 value, uint256 nonce,
bytes data);

    event Create(bytes20 indexed sender, bytes20 indexed contract_address, uint256 value,
uint256 nonce, bytes init);

    event Withdraw(bytes20 indexed sender, address indexed receiver, uint256 value, uint256
nonce);

    event Outcome(bool success);

    function createEVM(bytes calldata init) external payable returns (bytes20);

    function transferEVM(bytes20 to) external payable returns (bytes memory output);

    function callEVM(bytes20 to, bytes calldata data) external payable returns (bytes memory
output);

    function staticCallEVM(bytes20 to, bytes calldata data) external view returns (bytes
memory output);

    function withdrawFromMapped(uint256 value) external;

    function mappedBalance(address addr) external view returns (uint256);

    function mappedNonce(address addr) external view returns (uint256);
}
```

---

Revision #7

Created 14 June 2022 00:38:45 by Pana

Updated 28 June 2022 01:58:50 by Xianqi