Upgradable Contract

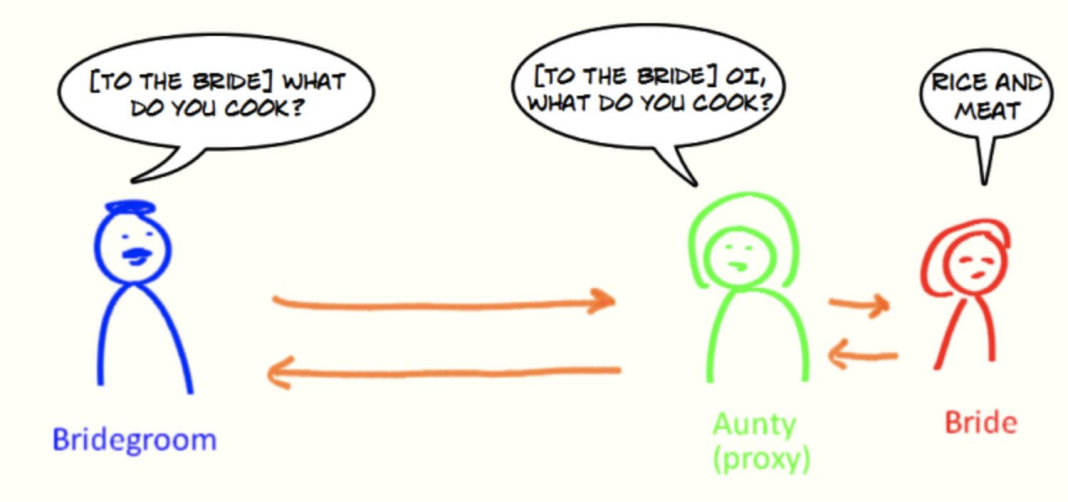
조성은

Upgradable Contract

- 이더리움 스마트 컨트랙트는, 한번 배포되면 수정이 불가능
- EVM의 Delegate Call / Fallback function을 활용.
- Proxy Pattern을 이용하여 수정 가능한 컨트랙트를 구현 가능

Proxy Pattern

- 외부로 명령을 요청받은 컨트랙트가 직접 작업을 수행하지 않고,
  다른 컨트랙트로 위임하여 처리
- Proxy 컨트랙트는 실제 작업 컨트랙트와 트랜잭션간의 연결과 데이터 저장 담당



Delegate Call

- 이더리움 스마트 컨트랙트 Call 방법은 두가지를 제공
- Static Call
  - 가장 일반적인 컨트랙트 호출방식으로, 컨트랙트가 외부 컨트랙트를 호출할 때 Internal Transcation을 생성하여 처리.
  - 각 컨트랙트가 고유의 상태값을 유지
- Delegate Call
  - 위임 호출 방식으로, Caller가 자신의 권한을 Callee 컨트랙트에 전임.
  - 별도의 Internal Tx 없이, 내부에서 처리된 것처럼 기록
  - Callee의 상태에 변화없이 Caller의 상태값만 변경
  - “Callee 가 Caller이것 처럼 작업을 처리한다”

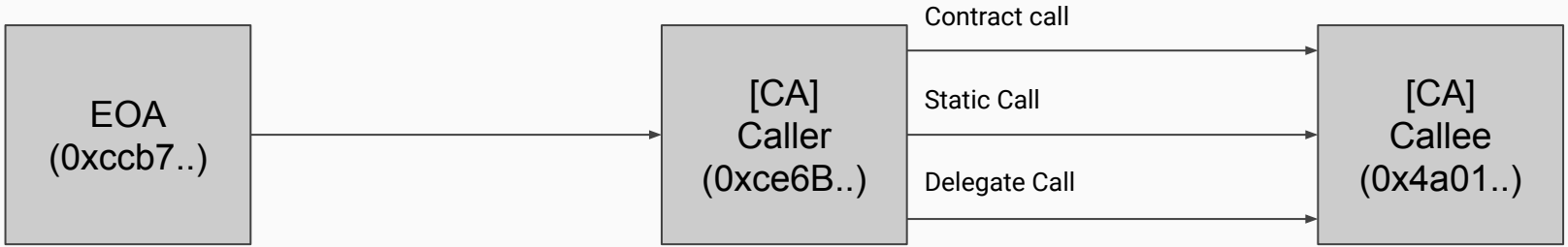
Static Call / Delegate Call

```
pragma solidity ^0.5.0;


contract Callee {
    address public caller;

    event Called(address indexed);

    constructor() public {
    }

    function func() public returns (bool result){
        caller = msg.sender;
        emit Called(caller);
        result = true;
    }
}
```

Static Call / Delegate Call

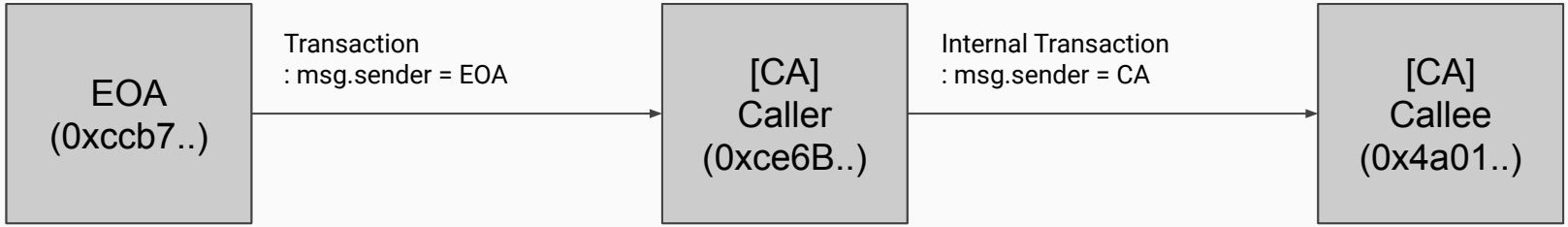
```
contract Caller {
    Callee public _callee;
    constructor(address _ca) public {
        _callee = Callee(_ca);
    }
    function funcDefaultCall() public returns (bool result){
        result = _callee.func();
    }
    function funcStaticCall() public returns (bool){
        //result = address(_callee).call(bytes4(keccak256("func()")));
        (bool result,) = address(_callee).call(abi.encodePacked(bytes4(keccak256("func()"))));
        return result;
    }
    function funcDelegateCall() public returns (bool ){
        //result = address(_callee).delegatecall(bytes4(keccak256("func()")));
        (bool result,) =
address(_callee).delegatecall(abi.encodePacked(bytes4(keccak256("func()"))));
        return result;
    }
}
```

Static Call / Delegate Call



Static Call / Delegate Call

Contract call / Static Call

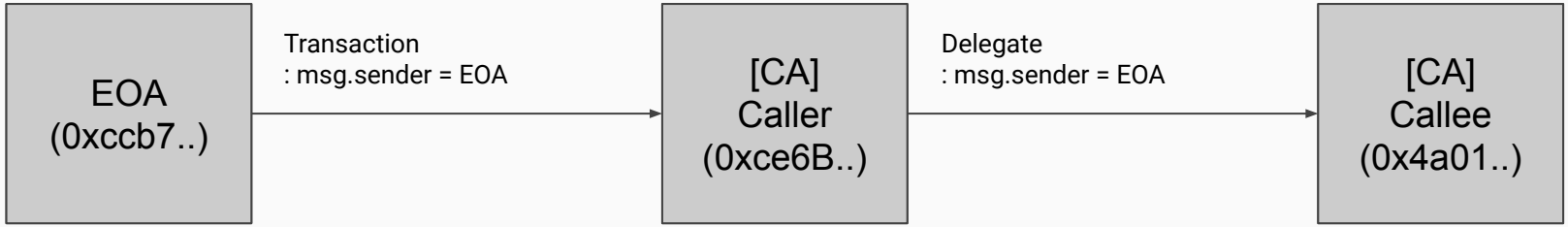


logs


```
[
    {
        "from": "0x4a01a9b4302f5ea7675eb081f8d1da2297e00e36",
        "topic": "0x9ea40c3ed94275424a632b11f10bf8d53c4cdb4164e47ef497b3e664131e4b2a",
        "event": "Called",
        "args": {
            "0": "0xce6B09A54258A7Dc9679D26c8b0dEfA7BF8Ef5A5",
            "length": 1
        }
    }
]
```

Static Call / Delegate Call

Delegate Call



logs

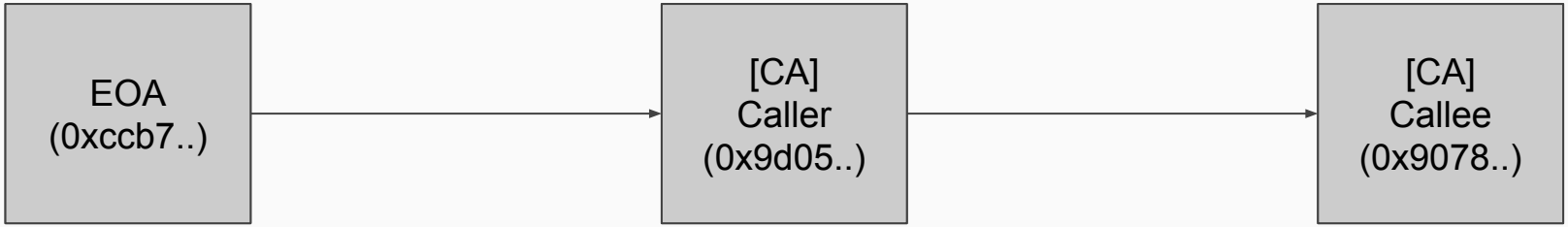

```
[
	{
		"from": "0xce6b09a54258a7dc9679d26c8b0defa7bf8ef5a5",
		"topic": "0x9ea40c3ed94275424a632b11f10bf8d53c4cdb4164e47ef497b3e664131e4b2a",
		"event": "Called",
		"args": {
			"0": "0xcCB70E8C26e6949feF7DDfDB8a8bDf07192acC80",
			"length": 1
		}
	}
]
```

Static Call / Delegate Call

```
pragma solidity ^0.5.0;



contract Callee {
    address public caller;

    event Called(address indexed);
    event WhoAmI(address indexed);

    constructor() public {
    }

    function func() public returns (bool result){
        caller = msg.sender;
        emit Called(caller);
        emit WhoAmI(address(this));
        result = true;
    }
}
```

Static Call / Delegate Call

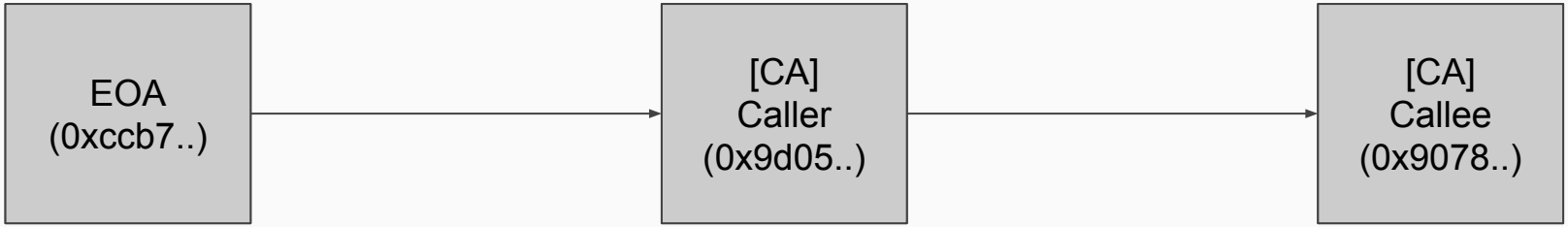
Contract call / Static Call




```
{
        "from": "0x9078ec4ce6a75f9755ce465f9e3e1163c0be35ad",
        "topic": "0x8b942f0f01b471f5d33e04ce59312a9ced116900bcdc1b21604313be28015c84",
        "event": "WhoAmI",
        "args": {
                "0": "0x9078Ec4CE6a75F9755ce465f9e3E1163c0Be35aD",
                "length": 1
        }
}
```

Static Call / Delegate Call

delegateCall




```
{
        "from": "0x9d05dbd8bf27f48c60c1da581b3f94ad7a25ef52",
        "topic": "0x8b942f0f01b471f5d33e04ce59312a9ced116900bcdc1b21604313be28015c84",
        "event": "WhoAmI",
        "args": {
                "0": "0x9d05dbD8bF27f48C60c1DA581B3F94aD7A25EF52",
                "length": 1
        }
}
```

Context

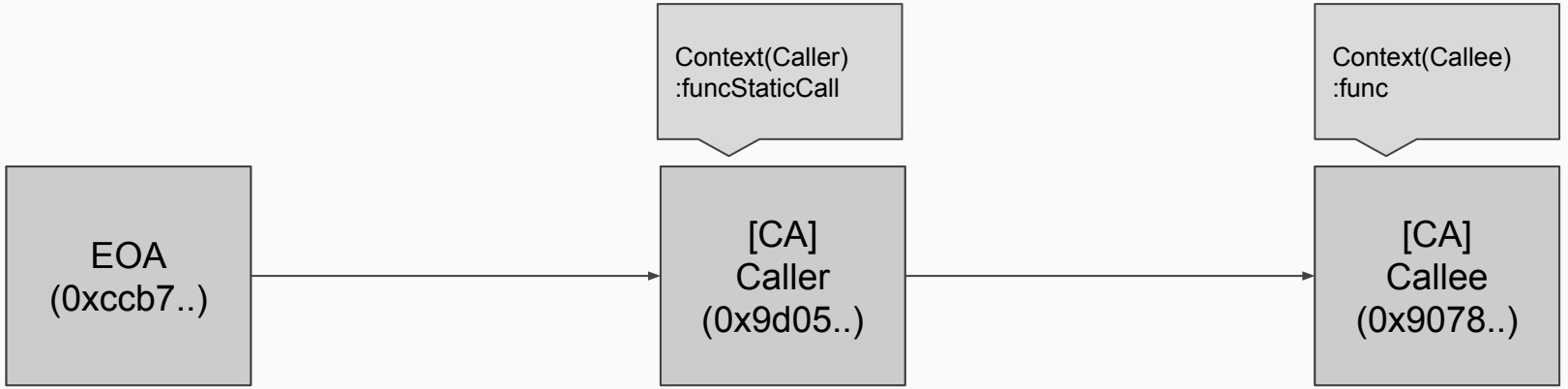
- EVM에서 동작하고 있는 컨트랙트 위치 및 상태
- 현재의 컨택스트 위에서 작업이 이뤄질 경우, 현재 컨택스트에 해당하는 컨트랙트의 상태값이 수정
- Delegate Call을 호출한다는 의미

  현재 컨택스트를 유지한 상태에서, 외부 컨트랙트의 기능을 호출

  -> 외부 컨트랙트의 기능을 현재의 컨트랙트로 불러와서 수행.

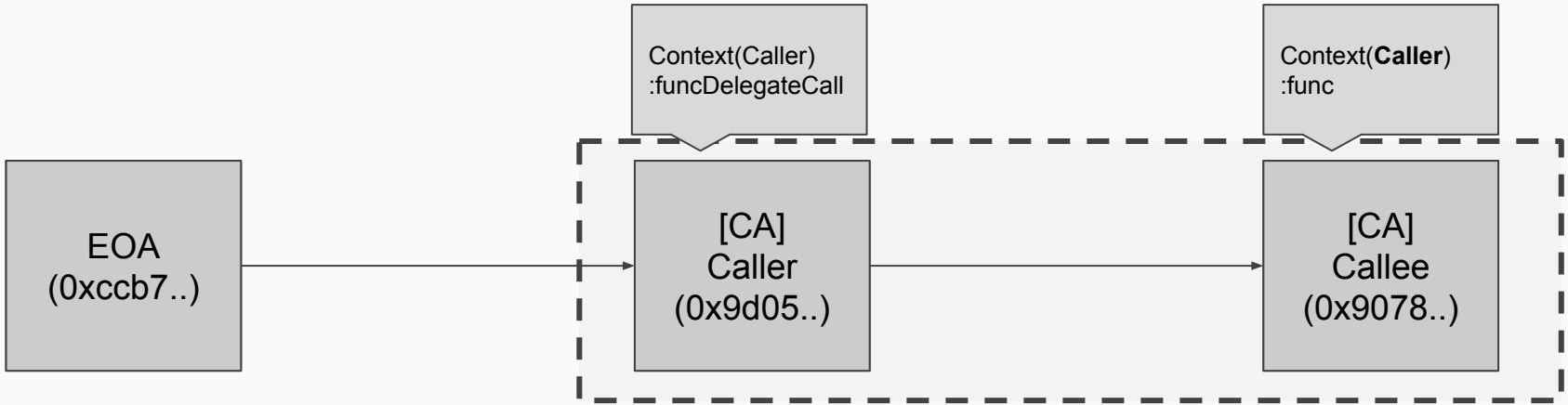
Context

- Static call에서의 컨택스트



Context

- Delegate call에서의 컨택스트



Storage

- Storage는 컨트랙트의 상태값을 저장하기 위한 저장소
- Delegate Call을 통해 Caller의 상태값을 Callee의 함수를 통해 수정 가능

Upgradable Example

- MovieBloc 컨트랙트 코드
- 향후 추가될 서비스 로직을 위해 업그레이더블 컨트랙트로 개발

https://etherscan.io/address/0xb879da8b24c9b8685de8526cf492e954f165d74b#code

Upgradable Example

```
address public implementation;
function _setImplementation(address _newImp) internal {
   implementation = _newImp;
}
function () payable external {
   address impl = implementation;
   require(impl != address(0));
   assembly {
       let ptr := mload(0x40)
       calldatacopy(ptr, 0, calldatasize)                          // (1) Copy incoming call data
       let result := delegatecall(gas, impl, ptr, calldatasize, 0, 0)    // (2) forward call to logic contract
       let size := returndatasize
       returndatacopy(ptr, 0, size)                                // (3) retrieve return data

       switch result                                               // (4) forward return data back to caller
       case 0 { revert(ptr, size) }
       default { return(ptr, size) }
   }
}
```

Upgradable Example

```
pragma solidity ^0.5.0;

contract Person {

  string public name = 'Alice';

  address public addFunctionContract;

  function _addContract(address _newImp) public {
    addFunctionContract = _newImp;
  }

  function () payable external {
    address impl = addFunctionContract;
    require(impl != address(0));
    assembly {
      let ptr := mload(0x40)
      calldatacopy(ptr, 0, calldatasize)
      let result := delegatecall(gas, impl, ptr, calldatasize, 0, 0)
      let size := returndatasize
      returndatacopy(ptr, 0, size)

      switch result
      case 0 { revert(ptr, size) }
      default { return(ptr, size) }
    }
  }
}
```

최초 배포한 컨트랙트

```
contract IWantToChangePerson is Person {

    function changeName(string memory _name) public {
        name = _name;
    }
}
```

업그레이드하여 배포한 컨트랙트

Upgradable Example

- Openzeppelin-labs 에서 제공된 프록시 컨트랙트

https://github.com/OpenZeppelin/openzeppelin-labs/tree/master/upgradeability_using_inherited_storage