

## 8. call, delegatecall, static call의 차이점을 EVM관점에서 서술해주세요.

### 1) 외부 컨트랙트 호출

스마트 컨트랙트에는 기능의 확장성을 위해 외부의 다른 컨트랙트를 호출할 수 있는 방법이 있는데, 이 때 3가지 호출 방법이 있다. 바로 `call`, `delegatecall`, `staticcall` 이다.

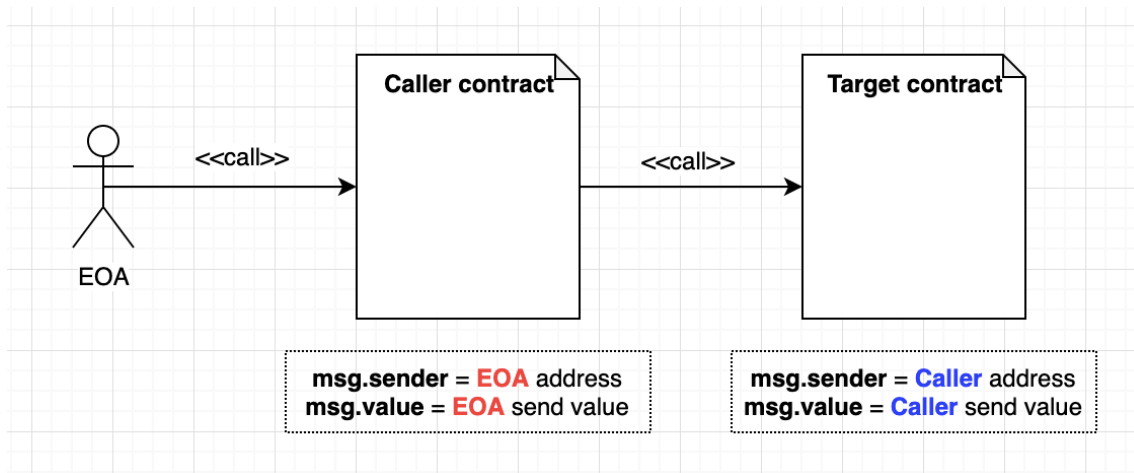
EVM 상에서 코드를 실행하면 각각 `CALL`, `DELEGATECALL`, `STATICCALL` opcode가 호출되는데, 이를 이용하여 외부 컨트랙트의 함수를 호출하고 코드를 실행할 수 있다.

이 때 각 3가지 opcode의 기능에는 차이점이 있는데, `caller` 컨트랙트의 컨텍스트(context)를 계속 유지하는지 아닌지 또는 `state variables` 변경가능하지 아닌지 등이 차이가 있다.

### 2) Call, DelegateCall and StaticCall

#### (1) Call

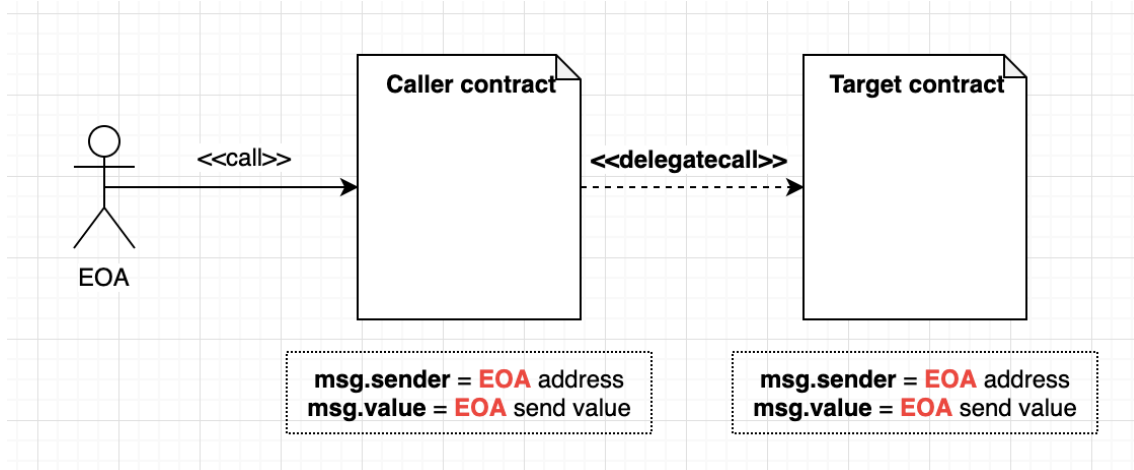
`call` 을 사용하여 외부의 컨트랙트를 호출할 경우 호출받은 컨트랙트는 자신의 컨텍스트(context) 안에서 코드를 실행한다. 따라서 만약 `state variables` 가 변경된다면 호출을 받은 컨트랙트의 `storage` 에 있는 상태가 변경되게 된다. 즉 `caller` 컨트랙트의 `storage` 상태에는 어떠한 변화도 발생하지 않는다. 그리고 호출받은 컨트랙트 입장에서는 `msg.sender` 는 `caller` 가 되고, `msg.value` 는 `caller` 가 전송한 value가 된다. 따라서 최초 트랜잭션을 전송한 사용자에게 대한 컨텍스트가 유지되지 않는다.



#### (2) DelegateCall

`delegatecall` 을 사용하여 외부 컨트랙트를 호출할 경우 호출받은 컨트랙트는 `caller` 컨트랙트의 컨텍스트 안에서 코드를 실행한다. 즉 EVM 이 코드를 실행할 때 사용하는 `storage` 의 컨텍스트는 `caller` 컨트랙트의 `storage` 컨텍스트가 그대로 유지되어 사용된다. 따라서 만약 호출된 컨트랙트에서 `state variables` 를 변경할 경우 실제로는 `caller` 컨트랙트의 `state variables` 가 변경된다. 게다가 `delegatecall` 을 사용하면 최초 트랜잭션을 생성한 `msg.sender` 와 `msg.value` 가 그대로 동일하게 전달된다. 따라서 호출받은 컨트랙트에서 사용하는 `msg.sender` 와 `msg.value` 는 최초 트랜잭션을 발생시킨 사용자의 컨텍스트가 그대로 유지된다.

이러한 이유로 `delegatecall` 은 외부 컨트랙트를 런타임에서 동적 라이브러리 로 사용하기 위해 쓰일 수 있다. 외부 컨트랙트의 실행 코드만 빌리고 실제로 모든 상태는 현재 컨트랙트의 `storage` 에서 처리되기 때문이다. 하지만 외부 컨트랙트에게 권한을 상태 변경 권한을 위임(delegation)하는 것이기 때문에 각별한 주의를 기울이지 않을 경우 심각한 취약점이 발생할 수 있다.



### (3) StaticCall

`staticcall` 을 사용하여 외부 컨트랙트를 호출할 경우 `call` 과 동일하게 동작하지만 호출되는 컨트랙트에서 `state variables` 를 변경할 수 없다. 그리고 해당 호출받은 컨트랙트가 다른 컨트랙트를 호출할 때도 마찬가지로 `state variables` 를 변경할 수 없다.

EVM에서는 `STATIC flag` 를 통해 상태 변경을 모니터링하는데, `STATICCALL` 을 사용하면 자식 컨트랙트를 실행하는 EVM 상에서 `STATIC flag` 가 `true`로 설정되고, 이 flag 값은 다른 서브호출(sub-call)을 할 때도 동일하게 전달된다. 그리고 만약 `STATIC flag` 가 `true`인 상태에서 `state` 를 변경하는 어떠한 오퍼레이션(e.g. `SSTORE`)이 발생하면 `exception` 을 출력한다.

---

## References

- [1] Vaibhav Saini. (2018, Aug 16). [Getting Deep Into EVM: How Ethereum Works Backstage](#) [Hacker Noon]
- [2] zeroFruit. (2019, Sep 1). [DelegateCall: Calling Another Contract Function in Solidity](#) [Medium]