

# **Upgradable Contract**

## **Proxy contract**

**분산기술개발팀**

# 목차

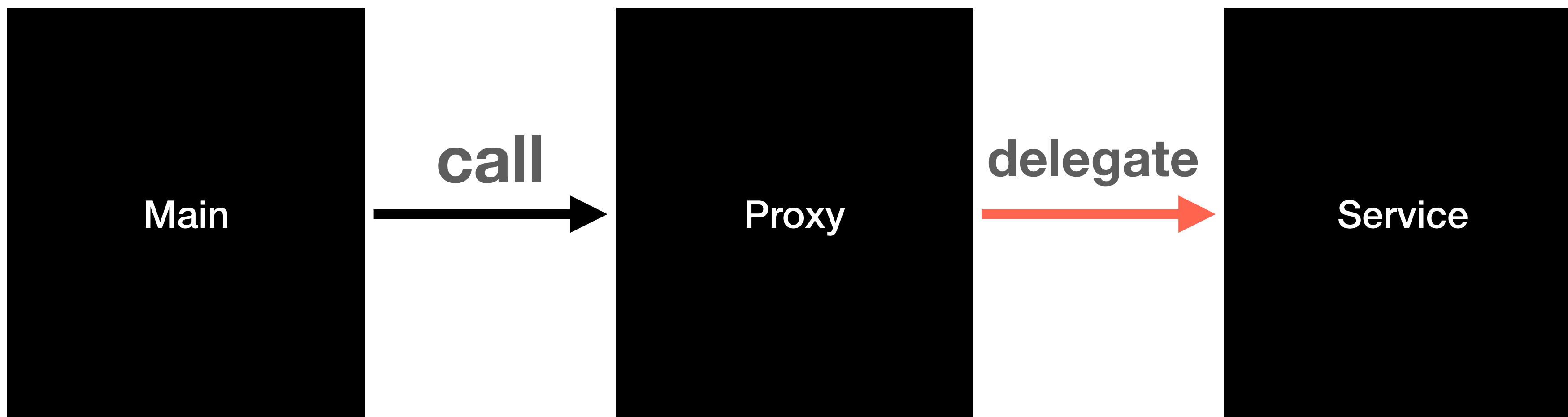
- Proxy Pattern
- Upgradable Contract Logic
- 시나리오
- 테스트

# Proxy Pattern

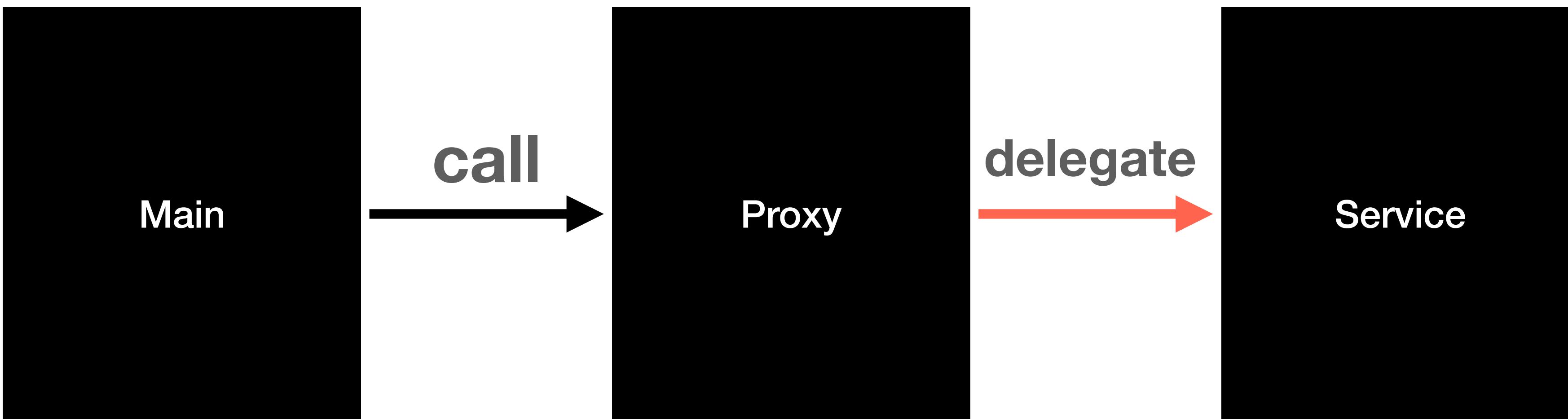
Proxy ; 대리자, 대변인

→ 누군가를 대신해서 그 역할을 수행하는 존재

## Proxy Pattern

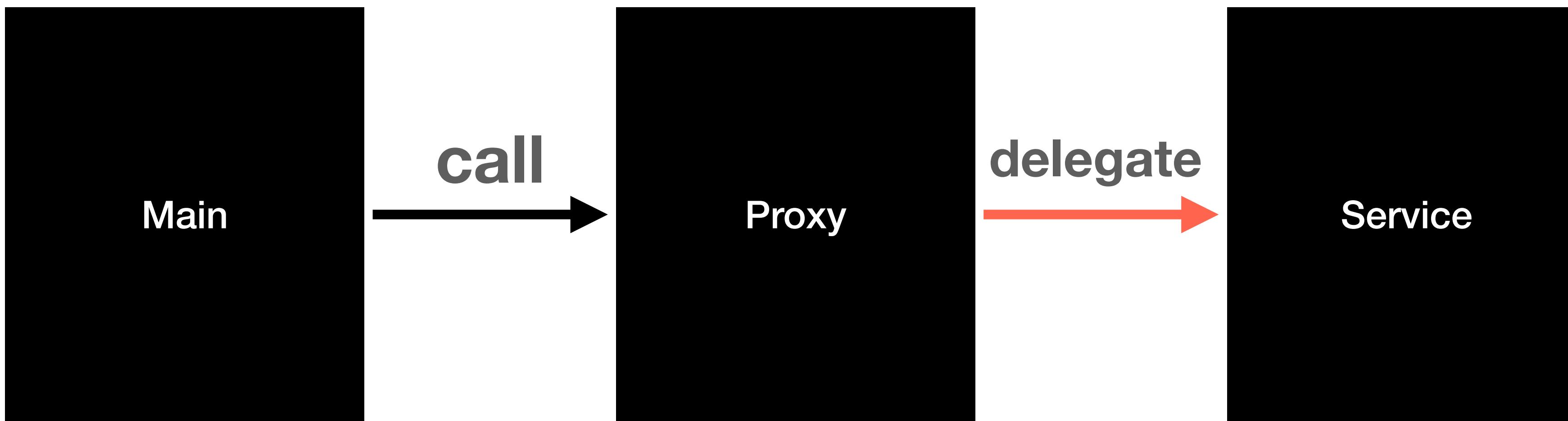


# Proxy Pattern



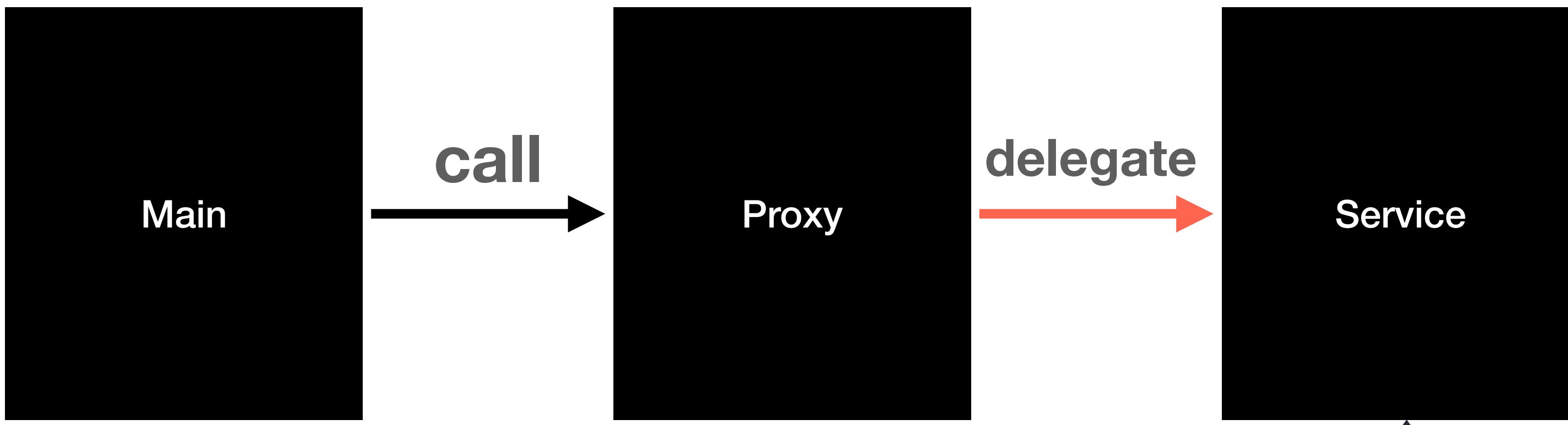
```
public class Main {  
    public static void main(String[] args) {  
        // 직접 호출하지 않고 프록시를 호출한다.  
        IService proxy = new Proxy();  
        System.out.println(proxy.runSomething());  
    }  
}
```

# Proxy Pattern



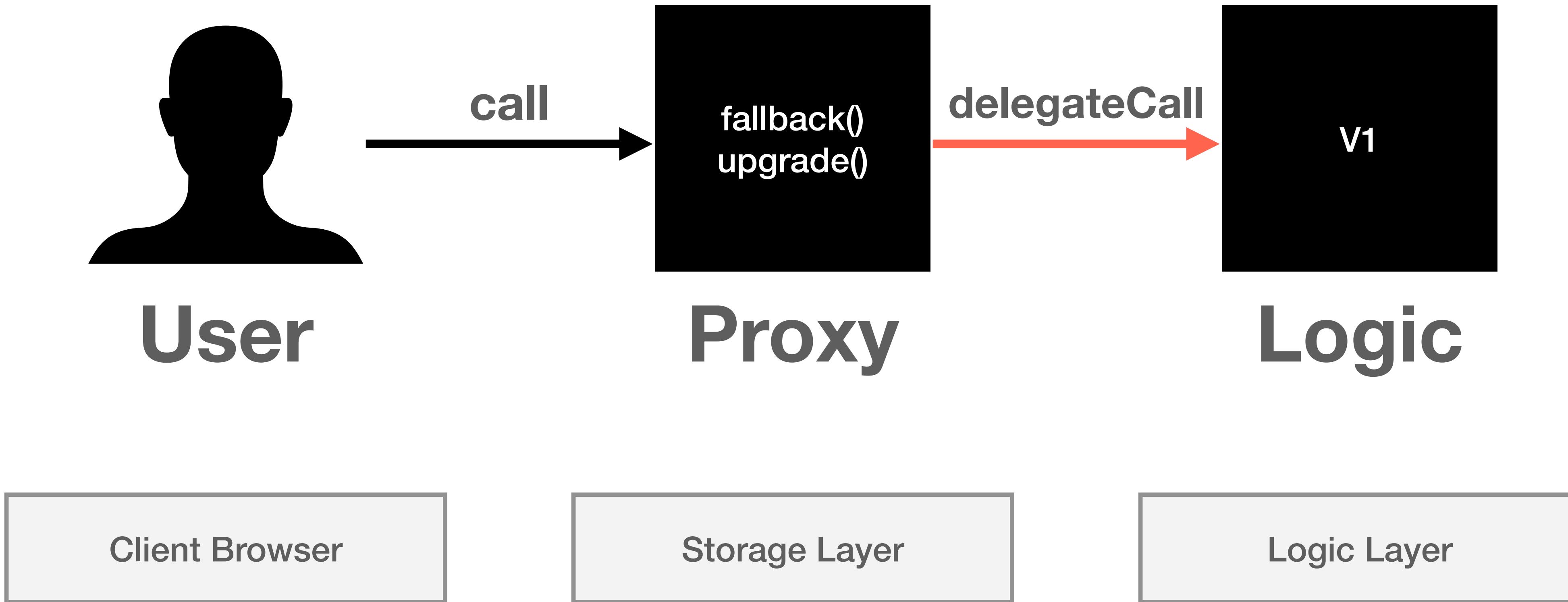
```
public class Proxy implements IService {  
    IService service1;  
  
    @Override  
    public String runSomething() {  
        System.out.println("호출에 대한 흐름 제어가 주목적, 반환 결과를 그대로 ?  
        service1 = new Service();  
        return service1.runSomething();  
    }  
}
```

# Proxy Pattern

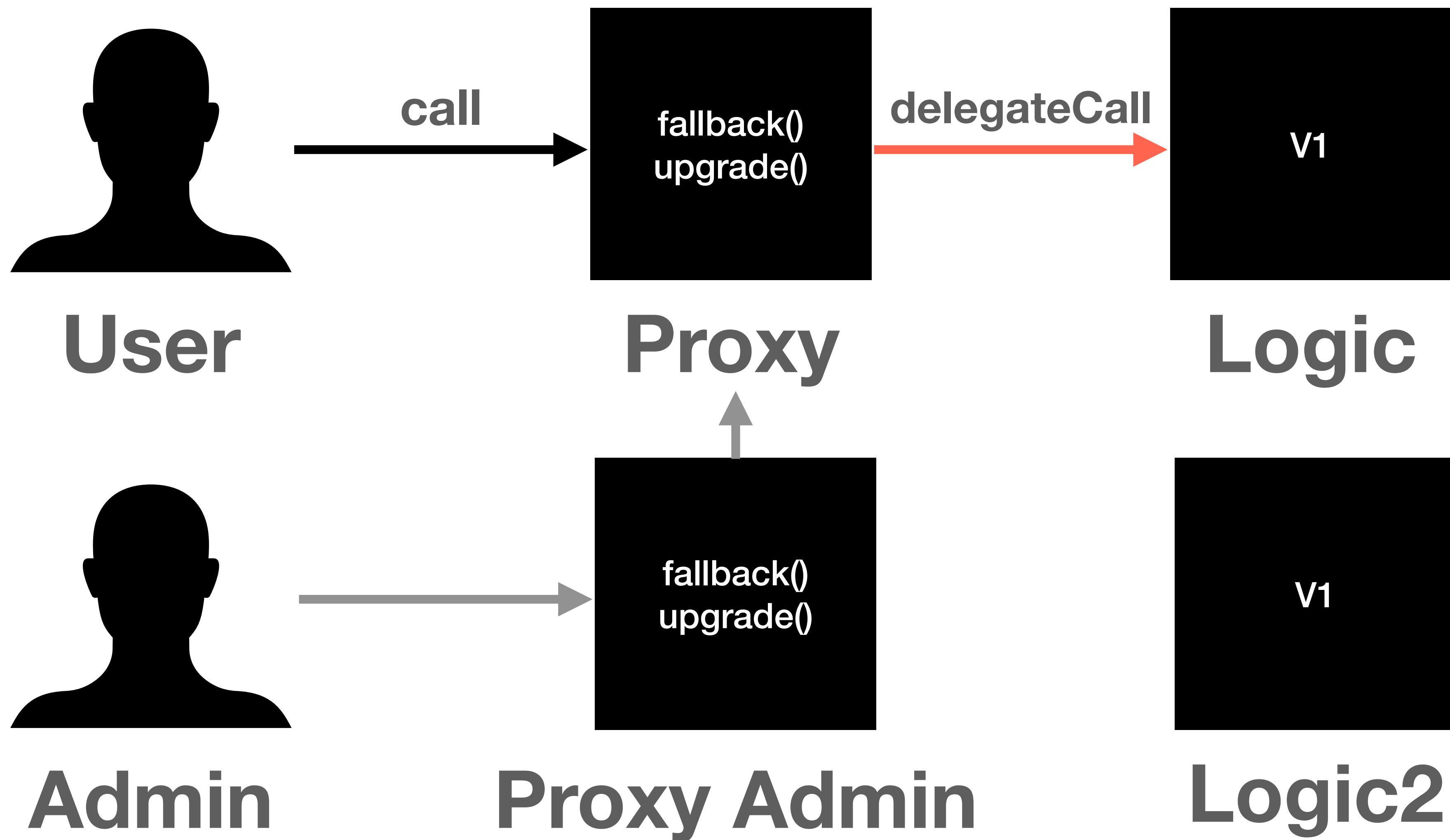


```
public class Service implements IService {  
    @Override  
    public String runSomething() {  
        return "서비스 짱!!!";  
    }  
}
```

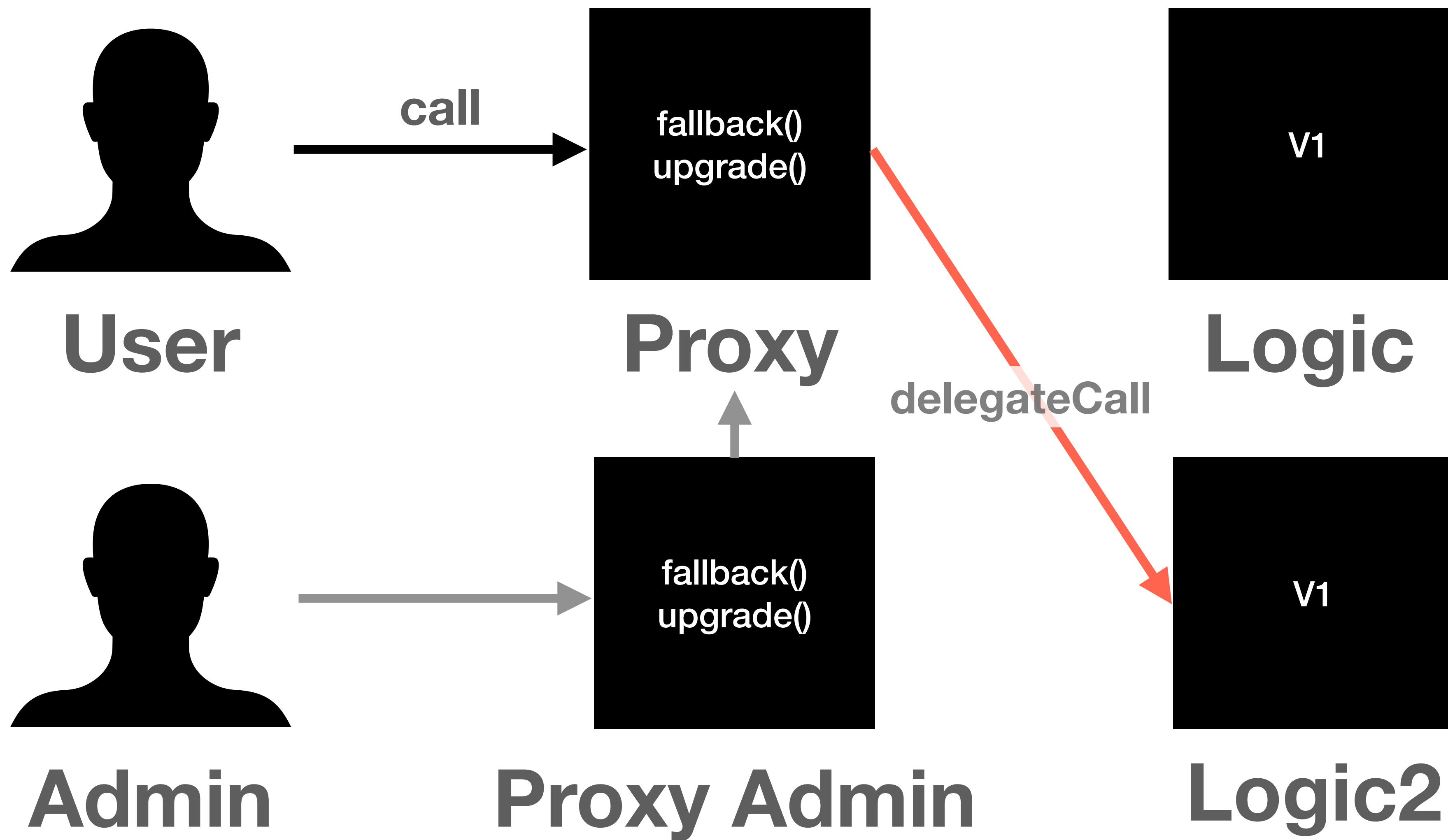
# Upgradable Contract Logic



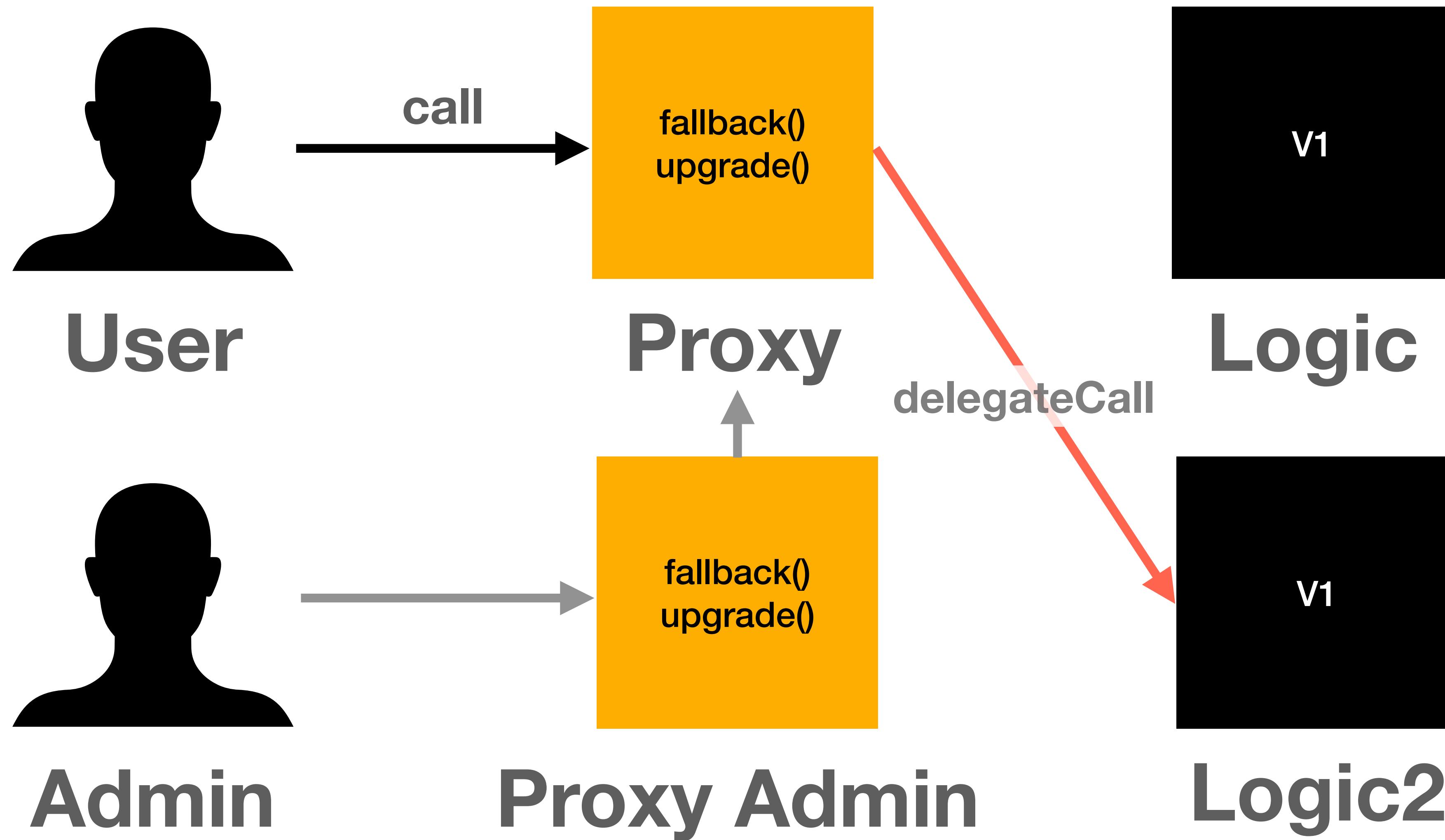
# Upgradable Contract Logic



# Upgradable Contract Logic



# Upgradable Contract Logic



# 시나리오

1. 첫번째 로직 컨트랙트 작성
2. Upgradable Contract의 함수를 이용해서 배포 스크립트 작성
3. 첫번째 배포
4. Value 값 확인
5. 두번째 로직 컨트랙트 작성
6. Upgradable Contract의 함수를 이용해서 업그레이드 배포 스크립트 작성
7. 두번째 배포
8. Value 값 확인

# 시나리오

## 1. 첫번째 로직 컨트랙트 작성

### Box.sol (solidity)

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity 0.8.11;
3
4 contract Box {
5     uint256 public val;
6
7     // constructor(uint _val) {
8     //     val = _val;
9     // }
10
11    ► function initialize(uint256 _val) external {
12        val = _val;
13    }
14 }
```

# 시나리오

## 2. Upgradable Contract의 함수를 이용해서 배포 스크립트 작성

### deploy\_box\_v1.js (Javascript)

```
1  const { ethers, upgrades } = require("hardhat");
2
3  async function main() {
4      const Box = await ethers.getContractFactory("Box");
5      console.log("Deploying Box...");
6
7      const box = await upgrades.deployProxy(Box, [42], {
8          initializer: "initialize",
9      });
10
11     await box.deployed();
12
13     console.log("Proxy contract address (Box deployed to) : ", box.address);
14
15 }
16 main();
```

# 시나리오

## 3. 첫번째 배포

\$ npx hardhat run --network rinkeby scripts/deploy\_box\_v1.js

```
Terminal: Local + ▾
→ Upgrading-test2 git:(main) ✘ npx hardhat run --network rinkeby scripts/deploy_box_v1.js
Deploying Box...
Proxy contract address (Box deployed to) : 0x3cc3df0208fc222894475929F90c4C7677D3397b
```

## etherscan에서 확인

Txn Hash	Method ⓘ	Block	Age	From ↴	To ↴	Value	Txn Fee
<a href="#">0x639aca1b783d41a808...</a>	0x60806040	11106761	1 min ago	0x105c060a0dca4a6286...	OUT  Contract Creation	0 Ether	0.00061786 \$
<a href="#">0x138b8b3908d5cabecb...</a>	0x60806040	11106736	7 mins ago	0x105c060a0dca4a6286...	New Contract 0x3cc3df0208fc222894475929f90c4c7677d3397b	0 Ether	0.00061786 \$

# 시나리오

## 3. 첫번째 배포

```
1  const { ethers, upgrades } = require("hardhat");
2
3  async function main() {
4      const Box = await ethers.getContractFactory("Box");
5      console.log("Deploying Box...");
6
7      const box = await upgrades.deployProxy(Box, args: [42], opts: {
8          initializer: "initialize",
9      });
10
11     await box.deployed();
12
13     console.log("Proxy contract address (Box deployed to) : ", box.address);
14
15 }
16 main();
```

# 시나리오

## 4. Value 값 확인 Proxy contract

Contract [0x3cc3df0208fc222894475929F90c4C7677D3397b](#)

**Contract Overview**

Balance: 0 Ether

**More Info**

My Name Tag: Not Available

Contract Creator: [0x105c060a0dca4a6286...](#) at txn [0x639aca1b783d41a808...](#)

**Transactions Internal Txns Erc20 Token Txns Contract ✓ Events**

[Code](#) [Read Contract](#) [Write Contract](#) [Read as Proxy NEW](#) [Write as Proxy NEW](#)

\_ABI for the implementation contract at [0x24f8676093a589800374c685be3dda34ab7df653](#), using the [EIP-1967 Transparent Proxy](#) pattern.

[Read Contract Information](#) [\[Expand all\]](#) [\[Reset\]](#)

<u>1. val</u>	<small>🔗 ↴</small>
42 uint256	

# 시나리오

## 4. Value 값 확인 Box contract

The screenshot shows the Etherscan interface for a Solidity smart contract named "Box".

**Contract Overview:**

- Balance: 0 Ether
- My Name Tag: Not Available
- Contract Creator: 0xb809be9c4172e560ad... at tx 0x6859f2804fa93e450c8...

**Contract Tab:** Transactions, Internal Txns, Erc20 Token Txns, Contract (selected), Events.

**Code Section:**

- Contract Source Code Verified (Exact Match)
- Contract Name: Box
- Compiler Version: v0.8.13+commit.abaa5c0e
- Optimization Enabled: No with 200 runs
- Other Settings: default evmVersion

**Contract Source Code (Solidity Standard Json-Input format):**

```
File 1 of 1 : Box.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract Box {
5     uint256 public val;
6
7     constructor(uint _val) {
8         val = _val;
9     }
10
12     function initialize(uint256 _val) external {
13         val = _val;
14     }
15 }
```

# 시나리오

## 4. Value 값 확인 Box contract

The screenshot shows a blockchain explorer interface for a specific Ethereum contract. The contract address is highlighted with a red box: `0x24F8676093A589800374c685bE3ddA34ab7DF653`. The interface includes sections for Contract Overview, More Info, Transactions, Internal Txns, Erc20 Token Txns, Contract (selected), and Events. Under the Contract tab, there are buttons for Code, Read Contract (which is selected and highlighted in grey), and Write Contract. The Read Contract Information panel displays the variable `1. val` with the value `0 uint256`, which is also highlighted with a red box.

Contract `0x24F8676093A589800374c685bE3ddA34ab7DF653`

**Contract Overview**

Balance: 0 Ether

**More Info**

My Name Tag: Not Available

Contract Creator: [0xb809be9c4172e560ad...](#) at txn [0x6859f2804fa93e450c8...](#)

**Transactions** **Internal Txns** **Erc20 Token Txns** **Contract** **Events**

**Code** **Read Contract** **Write Contract**

**Read Contract Information** [\[Expand all\]](#) [\[Reset\]](#)

1. val	<input type="button" value=""/>
<code>0 uint256</code>	<input type="button" value=""/>

# 시나리오

## 5. 두번째 로직 컨트랙트 작성

### BoxV2.sol (solidity)

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity 0.8.11;
3
4 contract BoxV2 {
5     uint256 public val;
6
7     // function initialize(uint _val) external {
8     //     val = _val;
9     // }
10
11    function inc() external {
12        val += 1;
13    }
14}
```

# 시나리오

## 6. Upgradable Contract의 함수를 이용해서 업그레이드 배포 스크립트 작성

### upgrade\_box\_v2.js (Javascript)

```
1 // scripts/upgrade_box_v2.js
2
3 const { ethers, upgrades } = require("hardhat");
4
5 const PROXY = "0x3cc3df0208fc222894475929F90c4C7677D3397b"; // 프록시 컨트랙트
6
7 async function main() {
8     const BoxV2 = await ethers.getContractFactory("BoxV2");
9     console.log("Upgrading Box...");
10    await upgrades.upgradeProxy(PROXY, BoxV2);
11    console.log("Box upgraded");
12 }
13
14 main();
```

# 시나리오

## 7. 두번째 배포

\$ npx hardhat run --network rinkeby scripts/upgrade\_box\_v2.js

```
Terminal: Local × + ▾
→ Upgrading-test2 git:(main) ✘ npx hardhat run --network rinkeby scripts/upgrade_box_v2.js
Upgrading Box...
Box upgraded
```

etherscan에서 확인

Txn Hash	Method ⓘ	Block	Age	From ↴	To ↴	Value	Txn Fee
<a href="#">0xdb9946d471ed0ef34a...</a>	Upgrade	11106863	1 min ago	0xb809be9c4172e560ad...	<span>OUT</span> <a href="#">0x4404a87e78b0f7c554...</a>	0 Ether	0.00009703 ⚡

# 시나리오

## 8. 확인 Proxy Admin Contract

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0xb809be9c4172e560ad... at txn 0xab2b60eb443882b5e6...

Transactions Internal Txns Erc20 Token Txns Contract Events

Code Read Contract Write Contract

Read Contract Information [Expand all] [Reset]

1. getProxyAdmin ↗ →

2. getProxyImplementation ↗ ↓

proxy (address)  
0x3cc3df0208fc222894475929F90c4C7677D3397b

Query ↴ ↵

address ↴ ↵

[ getProxyImplementation(address) method Response ]  
» address : 0x70Ec12d89E4d2781da27eF70Cb84467FE3663867

3. owner ↗ →

# 시나리오

## 8. 확인

### BoxV2 Contract

The screenshot shows a blockchain contract verification interface. At the top, it displays the contract address: `0x70Ec12d89E4d2781da27eF70Cb84467FE3663867`, which is highlighted with a red box. Below this, there are two main sections: "Contract Overview" and "More Info".

**Contract Overview:**

- Balance: 0 Ether

**More Info:**

- My Name Tag: Not Available
- Contract Creator: `0xb809be9c4172e560ad...` at tx `0x43a8c9d5de28035a87...`

Below these sections is a navigation bar with tabs: Transactions, Internal Txns, Erc20 Token Txns, Contract (selected), and Events. Under the Contract tab, there are three buttons: Code (selected), Read Contract, and Write Contract. To the right of the tabs is a search bar labeled "Search Source Code" with a dropdown arrow.

Under the "Contract" tab, there is a message: **Contract Source Code Verified (Exact Match)**. This message is preceded by a green checkmark icon and followed by a yellow warning icon.

Contract details:

- Contract Name: `BoxV2` (highlighted with a red box)
- Compiler Version: `v0.8.13+commit.abaa5c0e`
- Optimization Enabled: `No with 200 runs`
- Other Settings: `default evmVersion`

At the bottom, there is a section titled "Contract Source Code (Solidity Standard Json-Input format)". It shows the Solidity code for the `BoxV2.sol` file, with line numbers from 1 to 17. The entire code block is highlighted with a large red box. The code itself is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract BoxV2 {
5     uint256 public val;
6
7     // function initialize(uint _val) external {
8     //     val = _val;
9     // }
10
11
12     function inc() external {
13         val += 1;
14     }
15
16 }
17 }
```

# 시나리오

## 8. Value 확인 BoxV2 Contract

The screenshot shows the Etherscan interface for a specific Ethereum contract. The top navigation bar includes links for Home, Search, and Help. The main content area displays the following information:

**Contract Address:** Contract 0x3cc3df0208fc222894475929F90c4C7677D3397b (highlighted with a red box)

**Contract Overview:**

- Balance: 0 Ether

**More Info:**

- My Name Tag: Not Available
- Contract Creator: 0x105c060a0dca4a6286... at tx 0x639aca1b783d41a808...

**Tab Navigation:** Transactions, Internal Txns, Erc20 Token Txns, **Contract** (selected), Events

**Action Buttons:** Code, Read Contract, Write Contract, **Read as Proxy** (NEW), Write as Proxy (NEW)

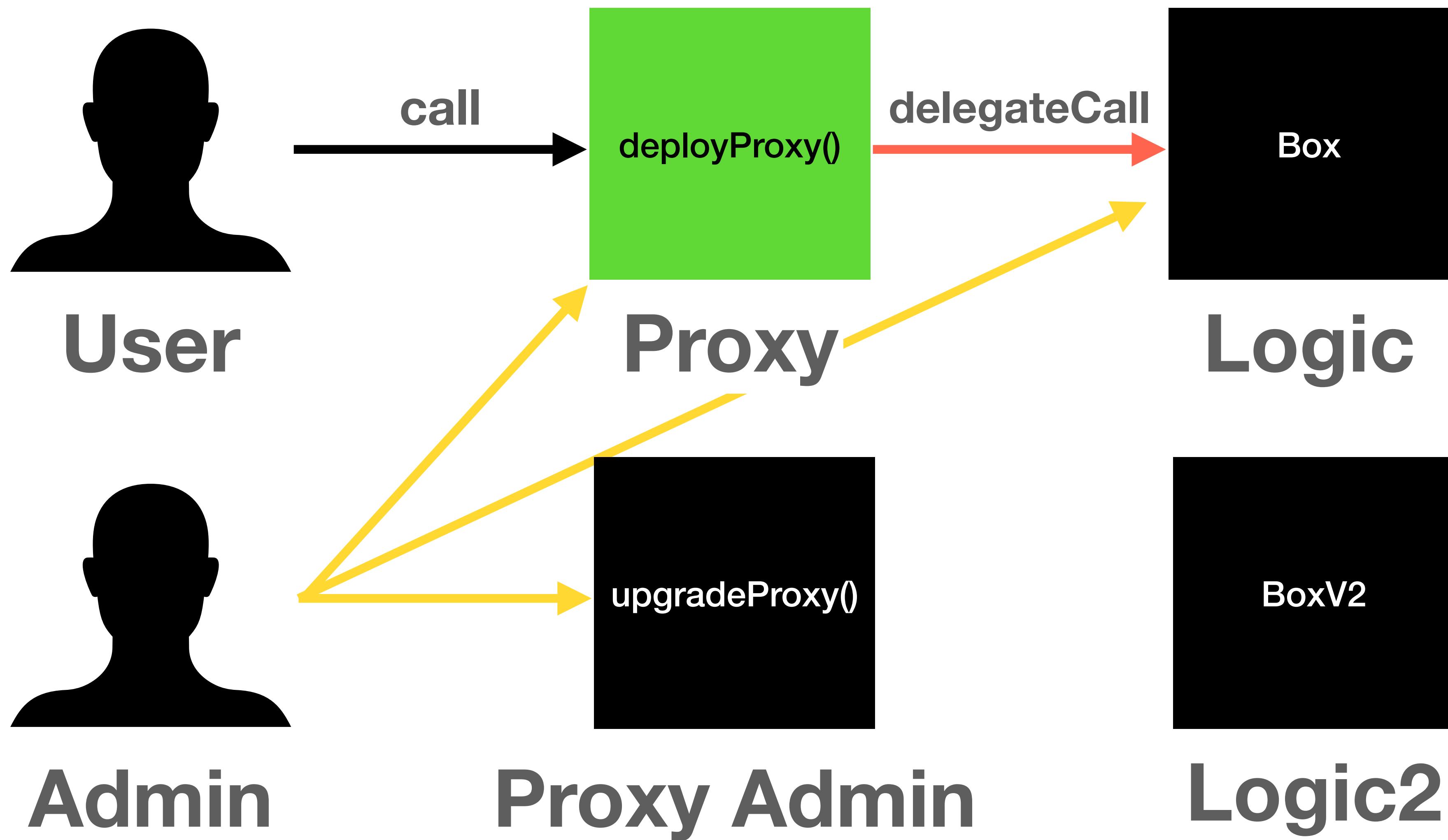
**ABI Information:** ABI for the implementation contract at 0x70ec12d89e4d2781da27ef70cb84467fe3663867, using the EIP-1967 Transparent Proxy pattern. Previously recorded to be on 0x24f8676093a589800374c685be3dda34ab7df653.

**Contract State:**

- 1. val: 43 uint256

[Expand all] [Reset]

# Summary



# Summary

