

041. Client/Server 리팩토링 (계속)

순차적으로 여러 클라이언트 요청 처리

ServerApp

```
ServerSocket ss = ...;
```

```
while (true) {
```

```
    Socket s = ss.accept();
```

```
    ...
```

```
}
```

ClientApp

```
Socket s = new Socket(IP, port);
```

```
...
```

```
s.close();
```



클라이언트

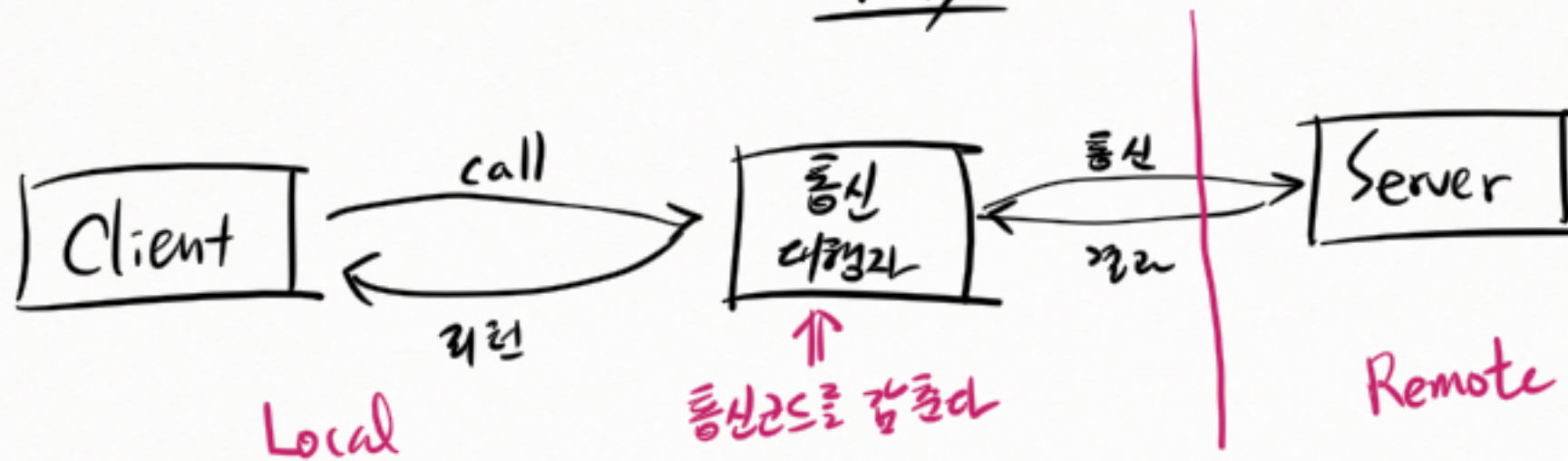


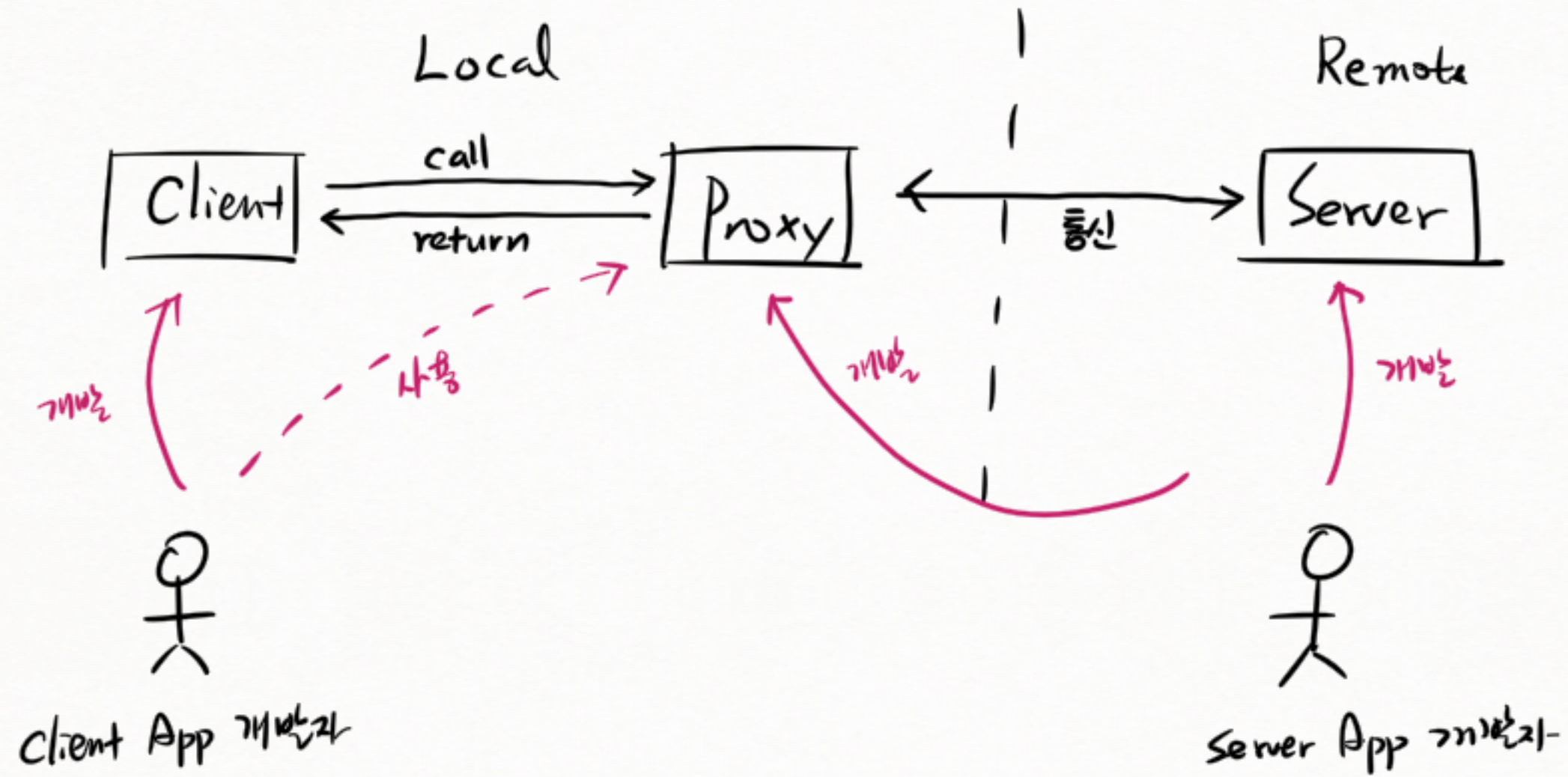
다들

* 042. Proxy 패턴을 활용하는 네트워킹 코드 캐싱

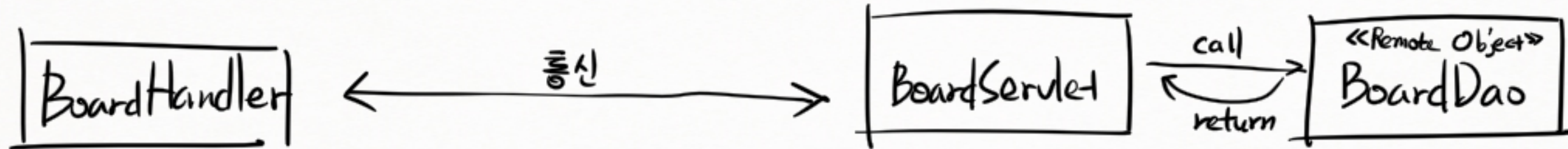
↓
복잡한 코드가 안 보이도록 클래스 안의 메서드로 감춘다
↓
해당 기능이 필요한 경우 간단히 메서드 호출로 처리.

① Proxy 패턴

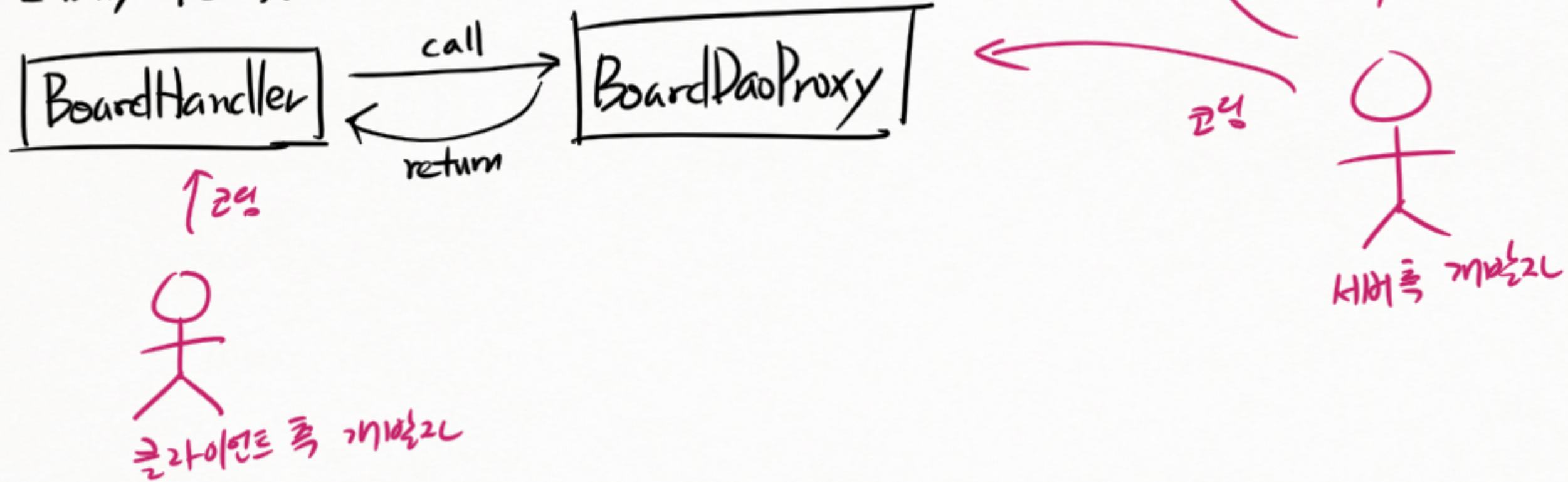




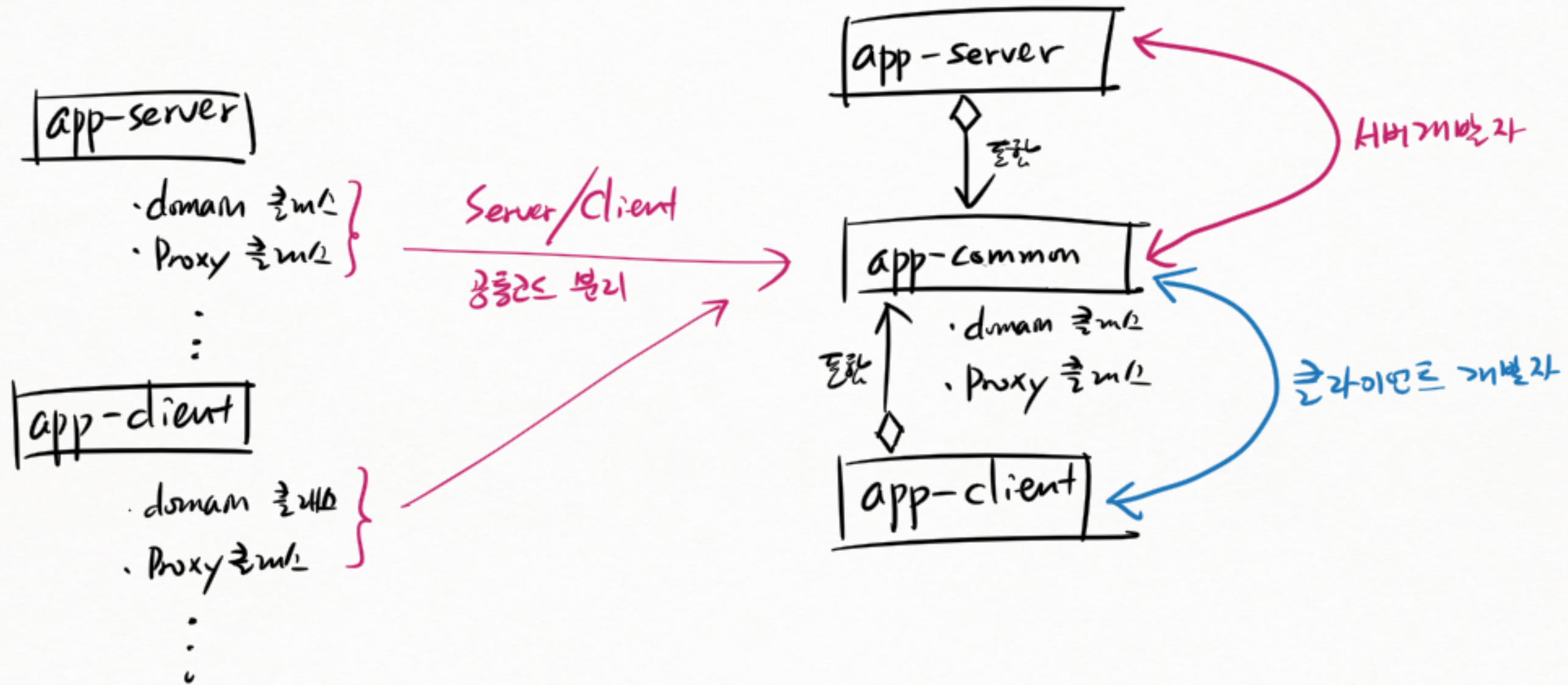
[이전 구조]



[Proxy 패턴 적용]

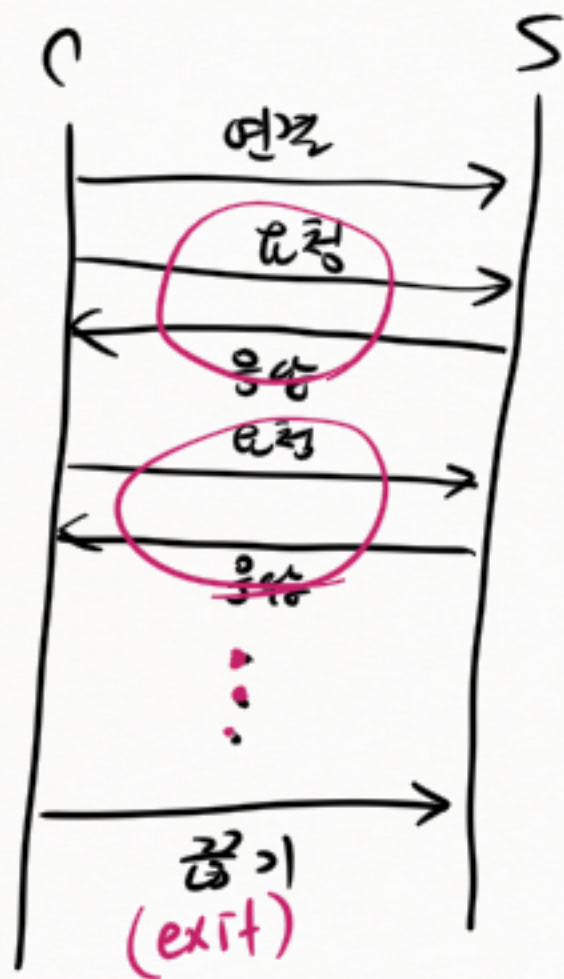


* 043. 서버 프로젝트를 분리 및 공유



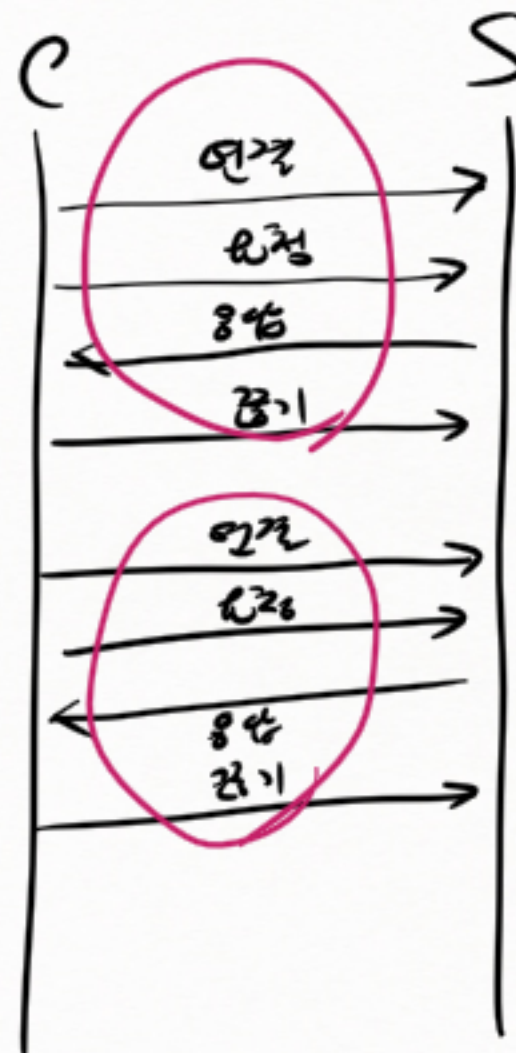
* 044. Stateful → Stateless
(Connection-Oriented)

① Stateful



일상적 예)
- 은행 업무

② Stateless



일상적 예)

- ARS 인증
- 114 전화번호 안내

