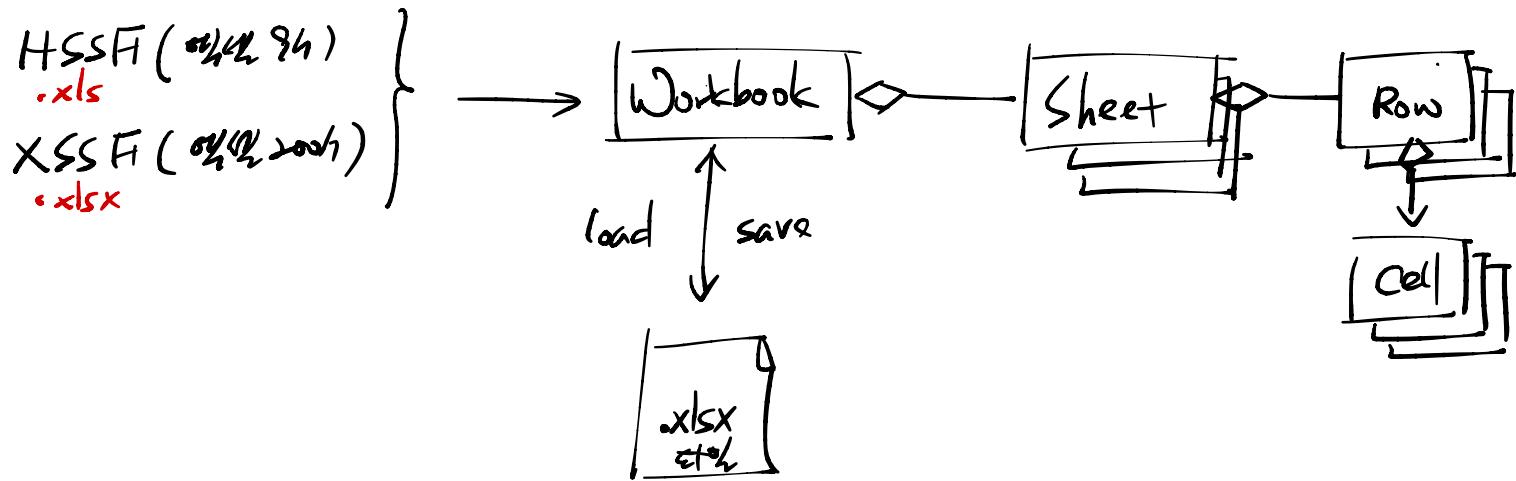


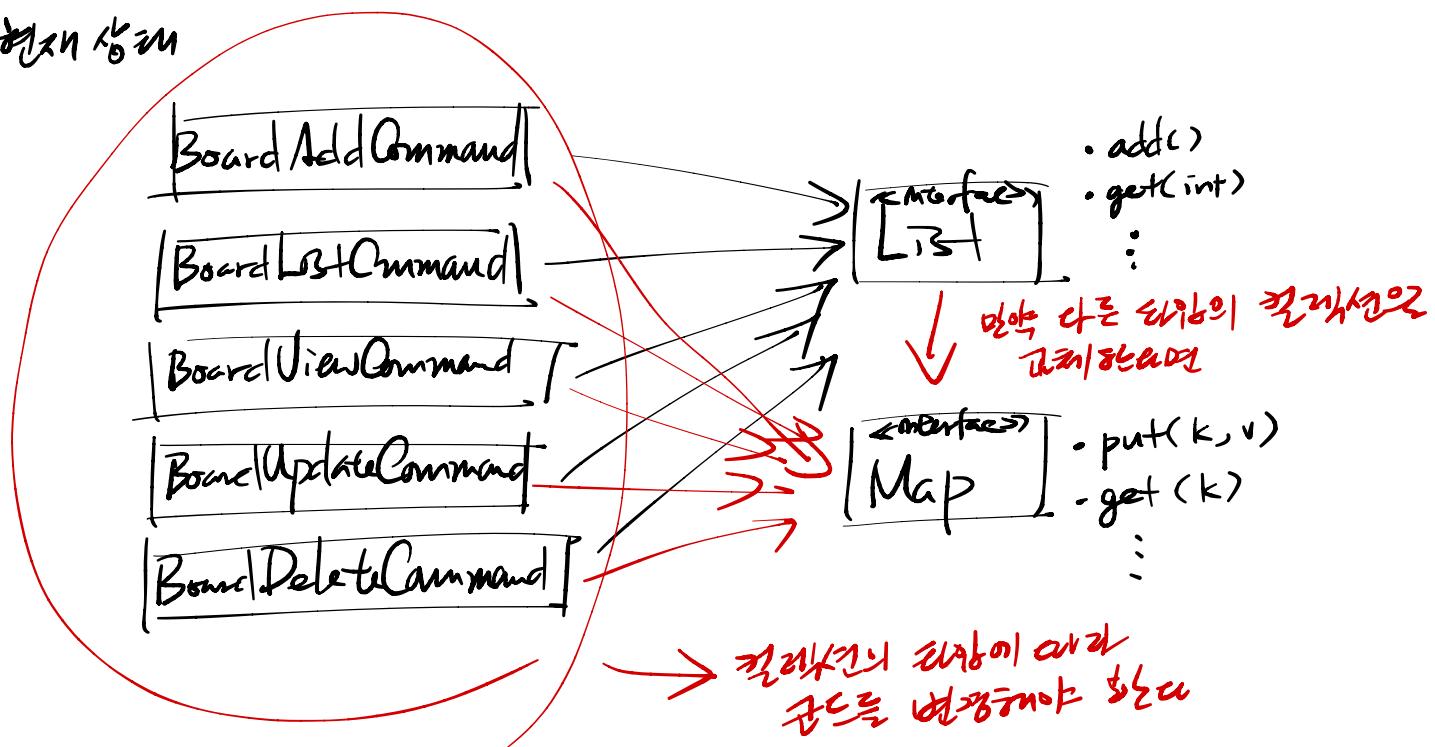
33. 콜렉터는 어떤 원칙을 따른다? : Apache POI 소개



34. DAO 가 필요한 이유

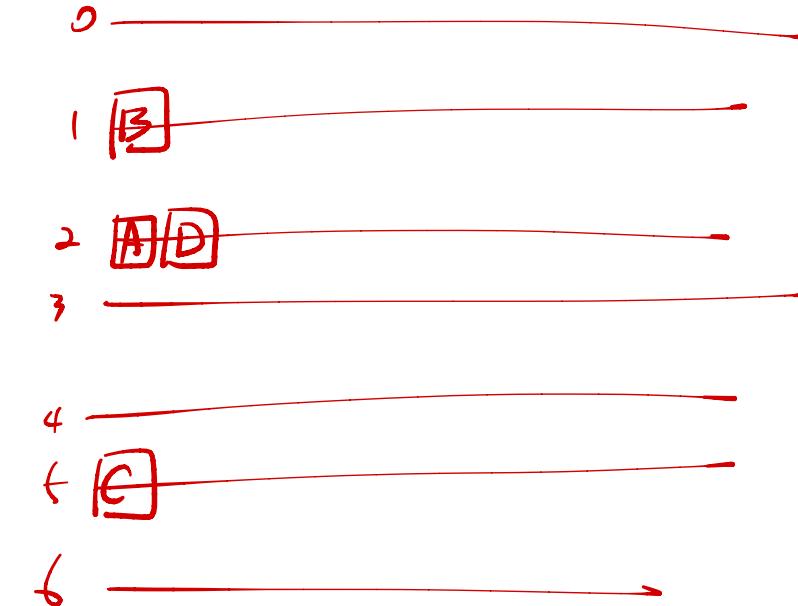
↳ 데이터 베이스를 관리하기

① 테이블



* Map의 데이터를 순회할 때는 항상
꺼내서 접근해야!

Map



② A

8-B

5-C

23-D

values()

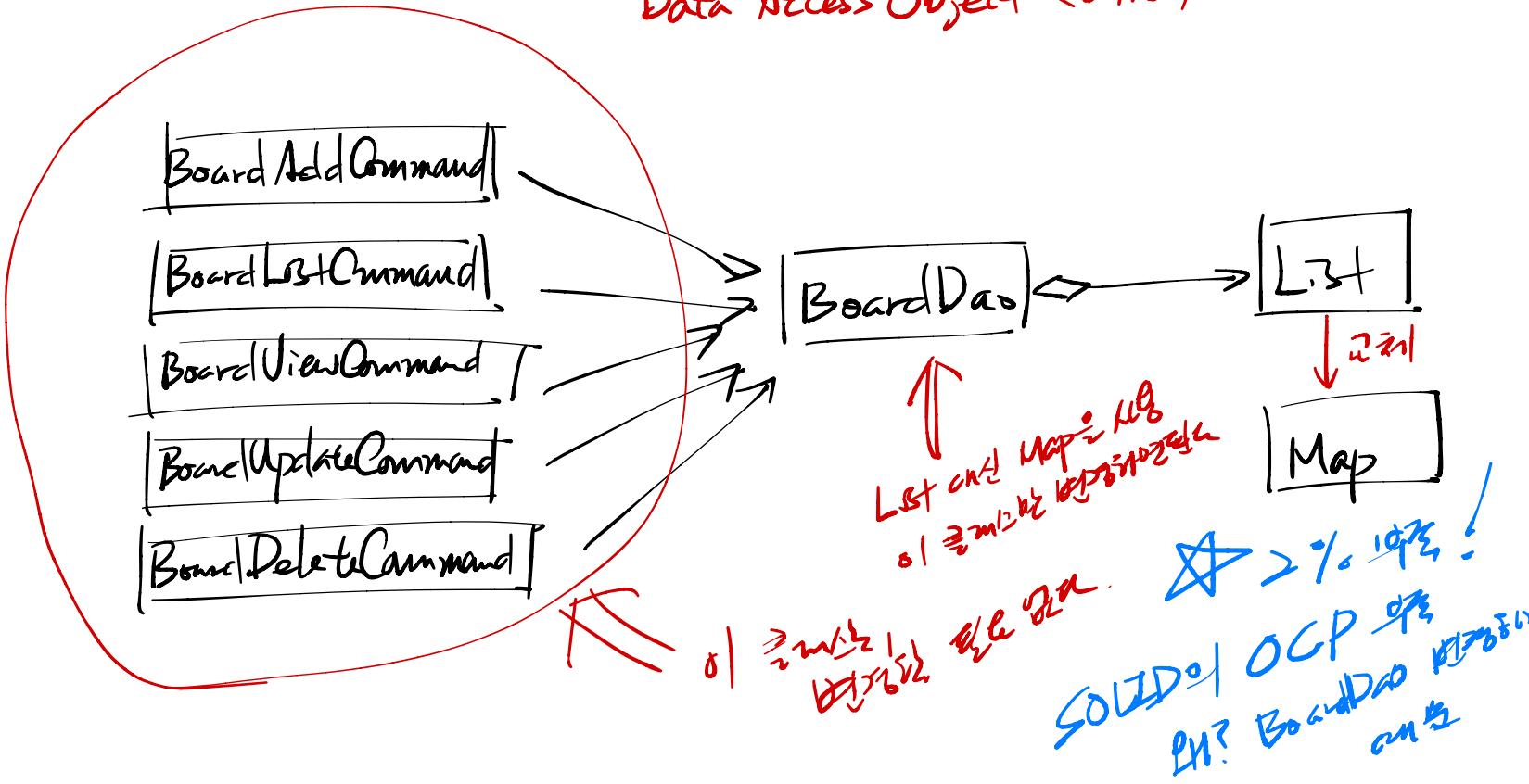
B A D C

① A B C D E
key의 hash 값을 가지고
인덱스 위치를 계산하는 방법

35. 글로벌 키워드를 제거해보자

↳ 제거한 키워드는 뭘까? → 함수가 기반이다.

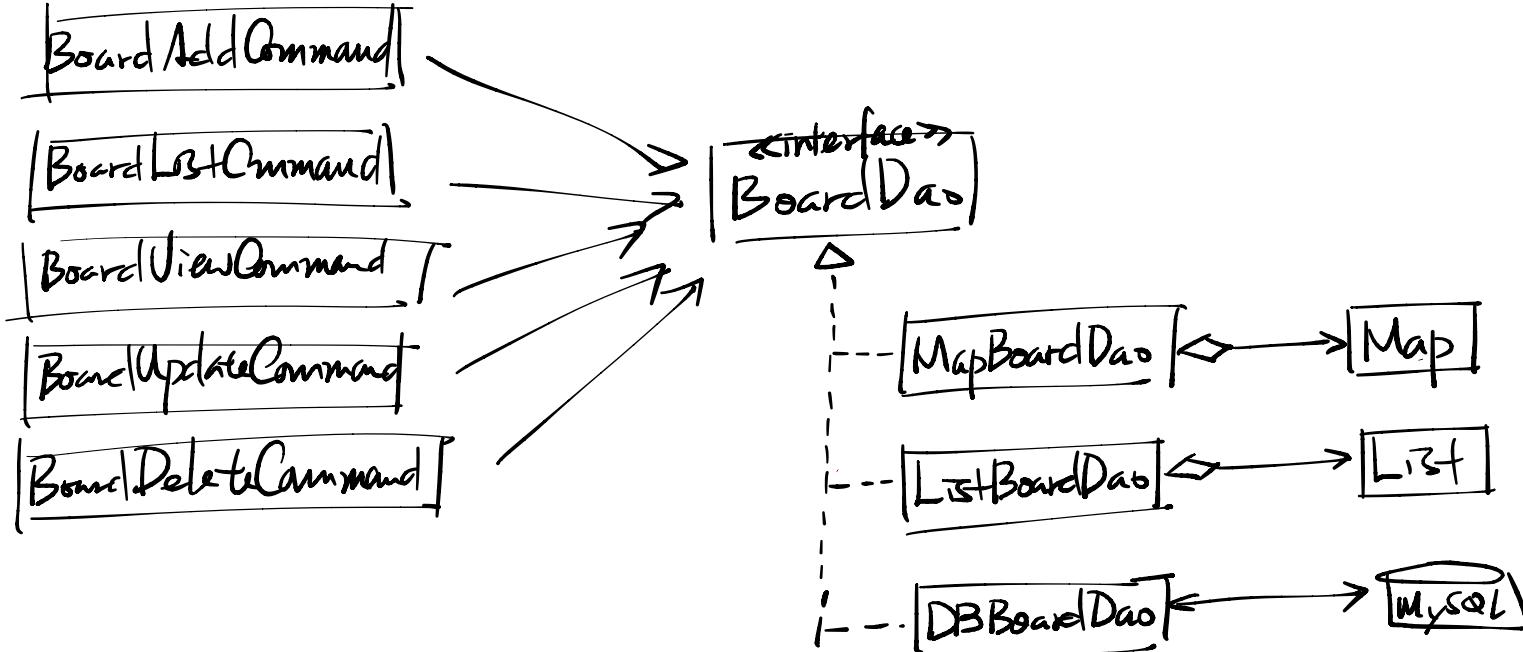
Data Access Object (DAO)



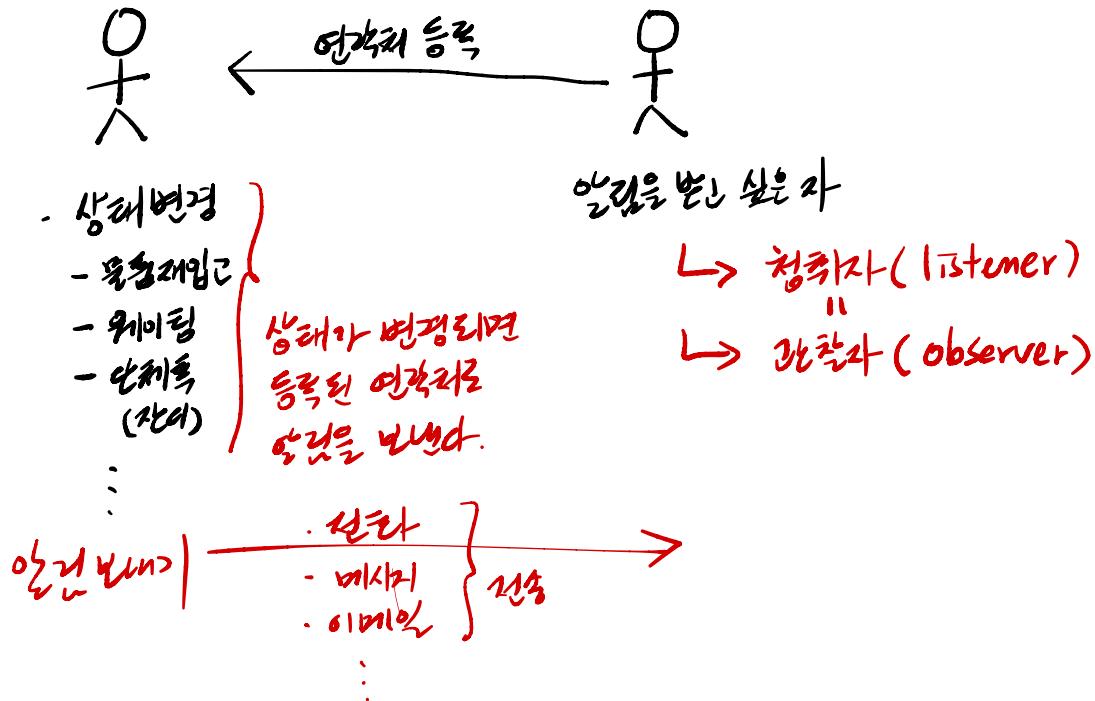
35. 글로벌 키워드는 final이면 됨

↳ 제한된 개수의 경우 → 제한된 개수의 제한된 개수.

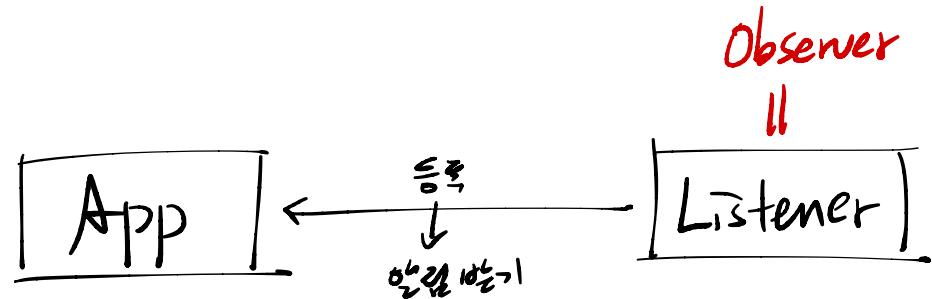
Data Access Object (DAO)



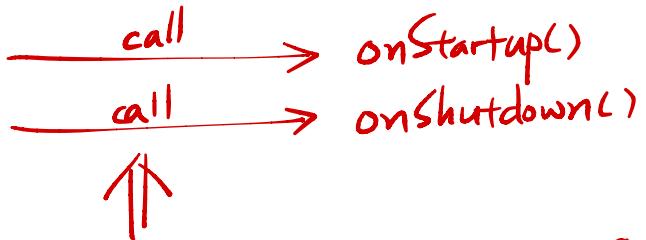
* 36. 알림 패턴 : GOF의 Observer 패턴
(설명문서)



$\exists \subseteq \text{Invert}$)

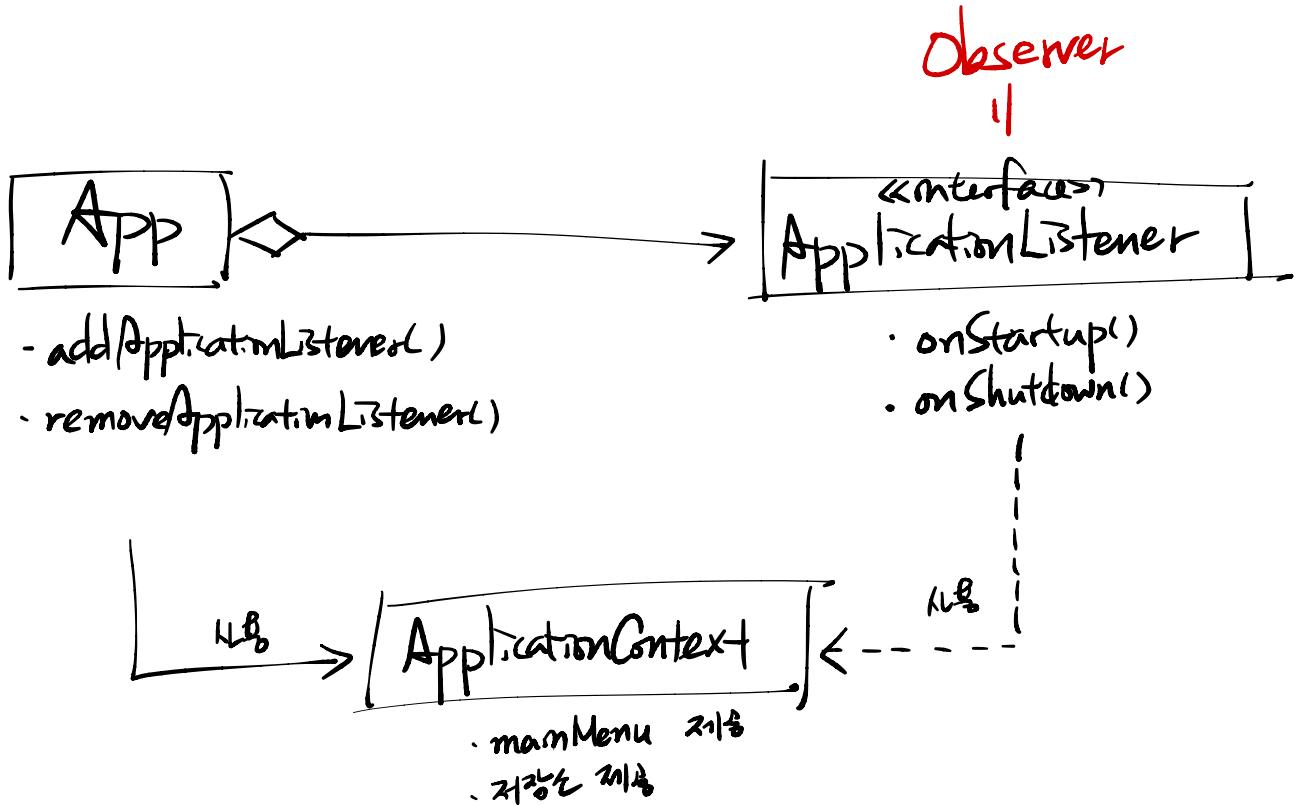


- 설정에 따라
- 사용자
- 풍경



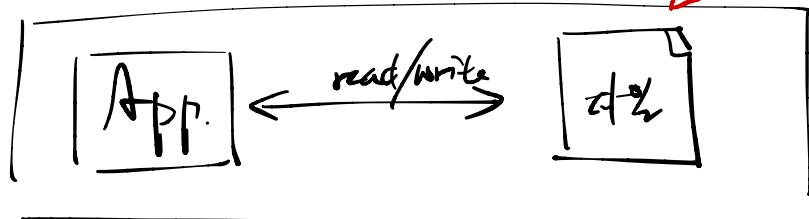
설정에 따라 설정에 따라 설정에 따라
설정에 따라 설정에 따라 설정에 따라

* GOF의 Observer 패턴 대처



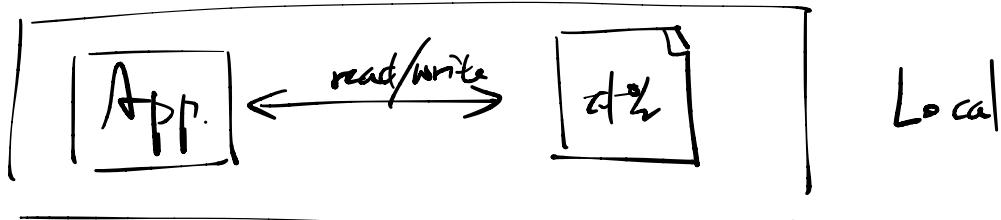
31. Application 간에 데이터 공유

현재 상태



App.의
데이터로 데이터를
Local

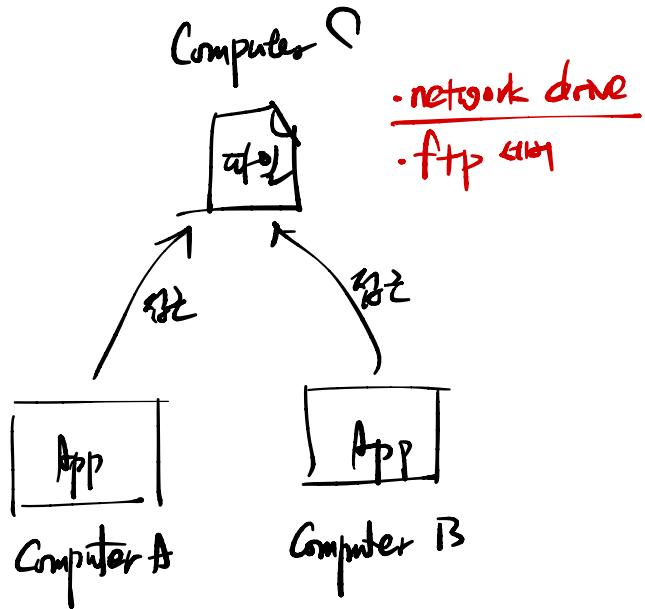
App.-간의
데이터 공유는!



•
•
•

31. Application 간에 파일 교환

① 파일로드된 파일 공유



파일

- 동시에 여러 App. 파일을
교환할 수 있다



파일을 쉽게 쓸 수 있다.

31. Application 간에 데이터 교환

② 멀티프로세스 애플리케이션

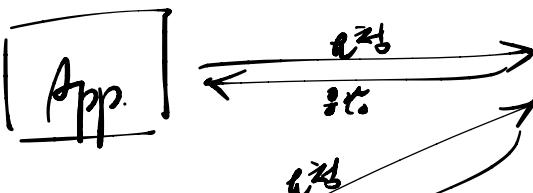
- networking prog.
- multi-tasking prog.
 - ↳ multi-threading
 - ↳ synchronous

애플리케이션
Data는 쓰고 읽을 때
Data를 주고 받음

Computer A



Computer B

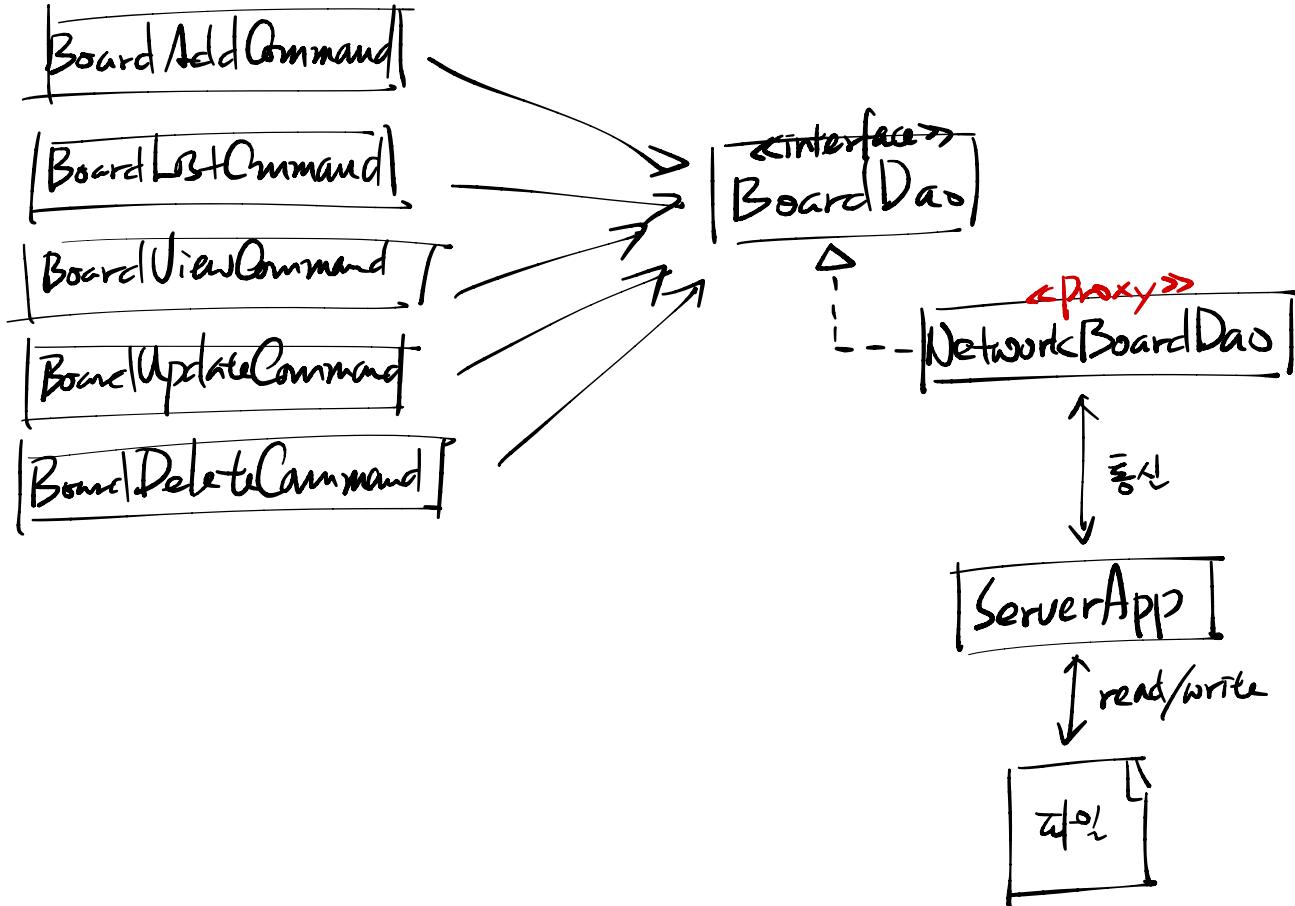


Computer C

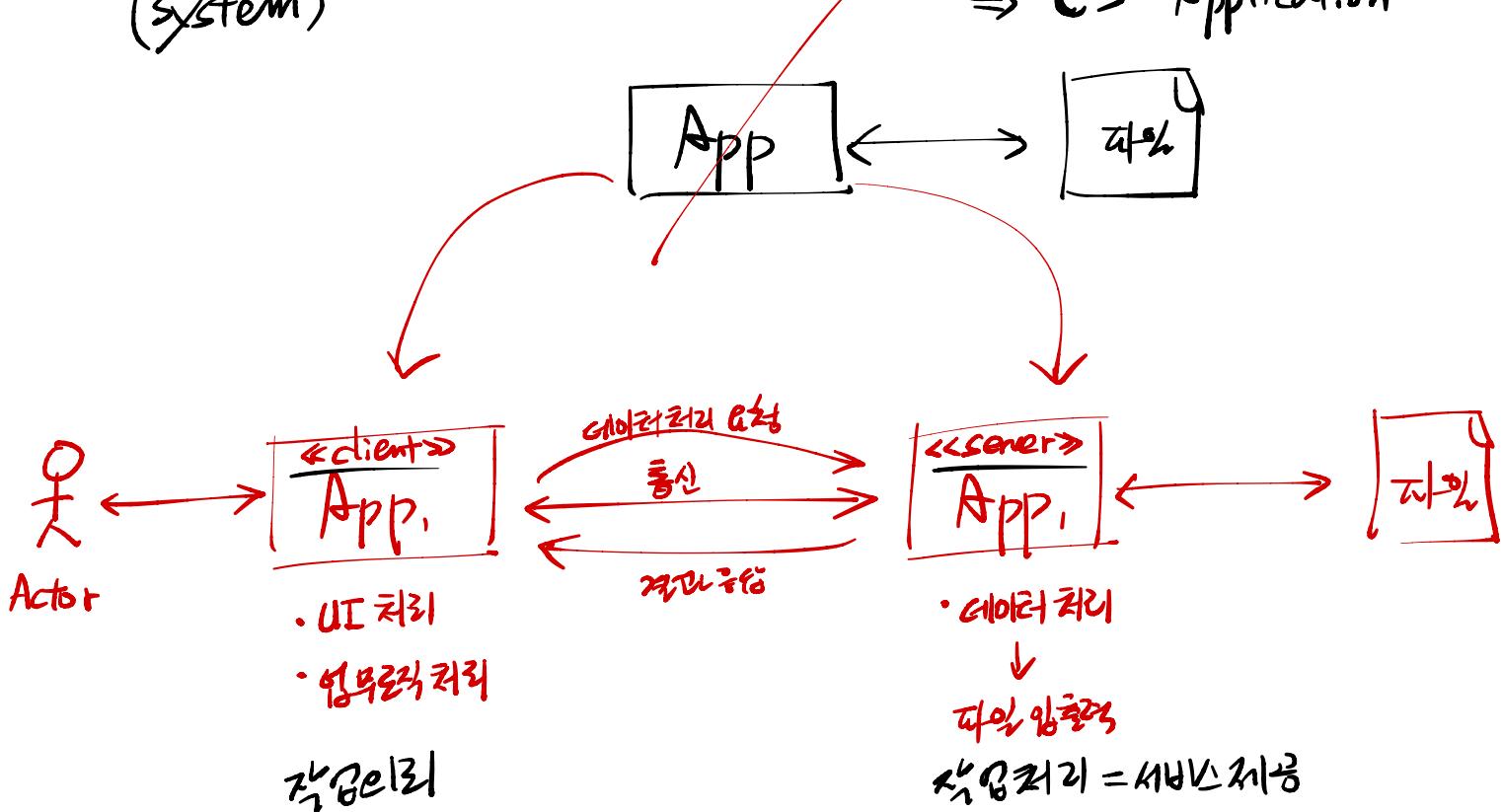


networking

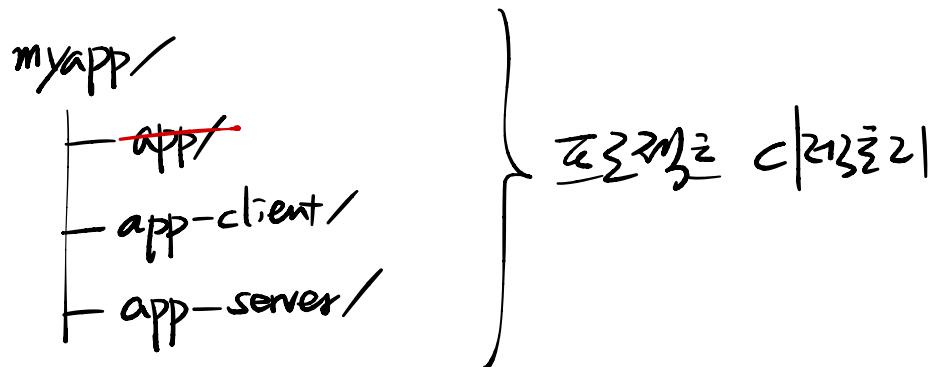
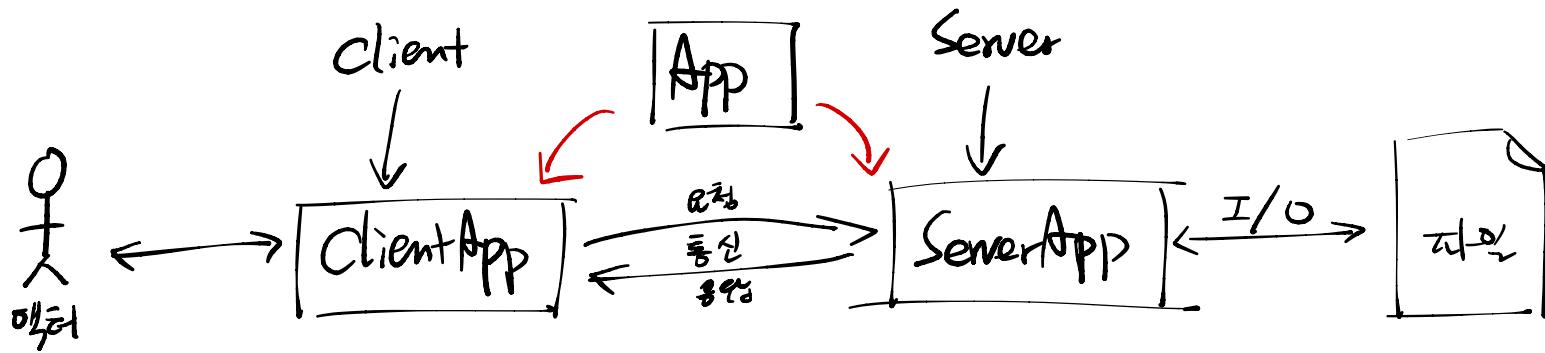
단계 32번의 단계 32번 일정 시스템
(process) (process) 단계 32번
단계 32번의 단계 32번 일정 시스템
(process) (process) 단계 32번



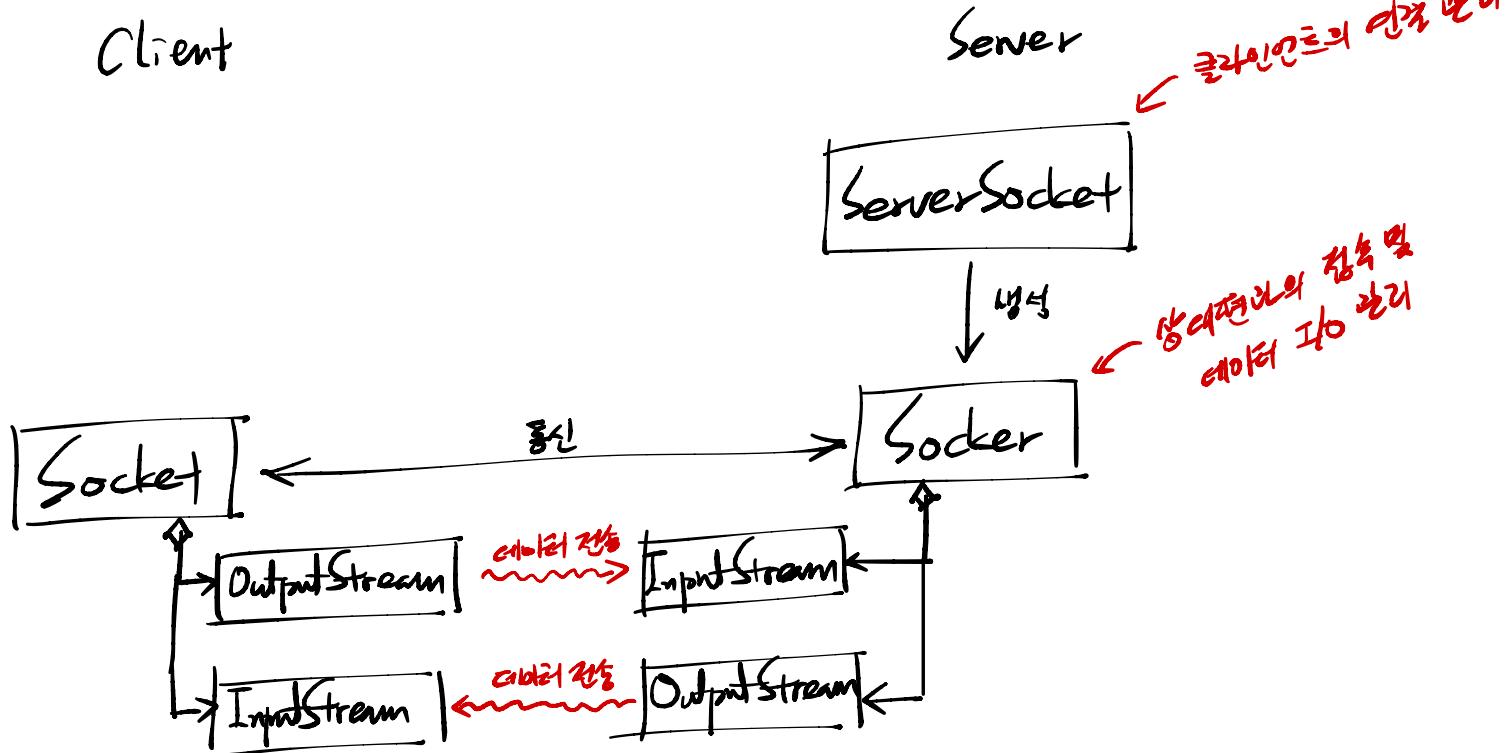
* Software Architecture : Client / Server Architecture
(System) ⇒ CS Application



* CS Architecture 구조

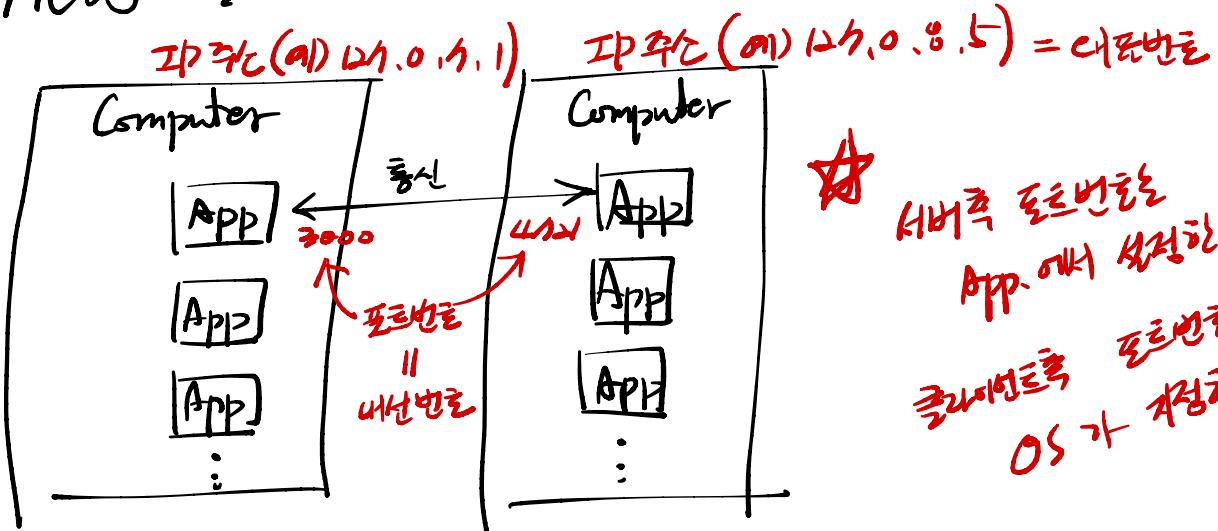


* Networking API API



* ServerSocket

new ServerSocket(포트번호, 대기열크기)



☆
내부적 포트번호는
App.에서 설정된다
클라이언트는 포트번호
OS가 지정한다.

통신내선번호
포트번호
클라이언트포트번호

* Socket

Client IP 주소



Client IP 주소



new Socket (IP 주소, 포트번호)

* 나의 포트번호?
(로컬 IP)

OS가 제공합니다.

* 특별한 IP 주소

127.0.0.1

Local
주소
||
loopback 주소

||
"localhost"

* Protocol : 데이터 송수신 규칙

client

컬렉션 - "users"

작동 명령 - "insert" | "list" | "get" |
"update" | "delete"

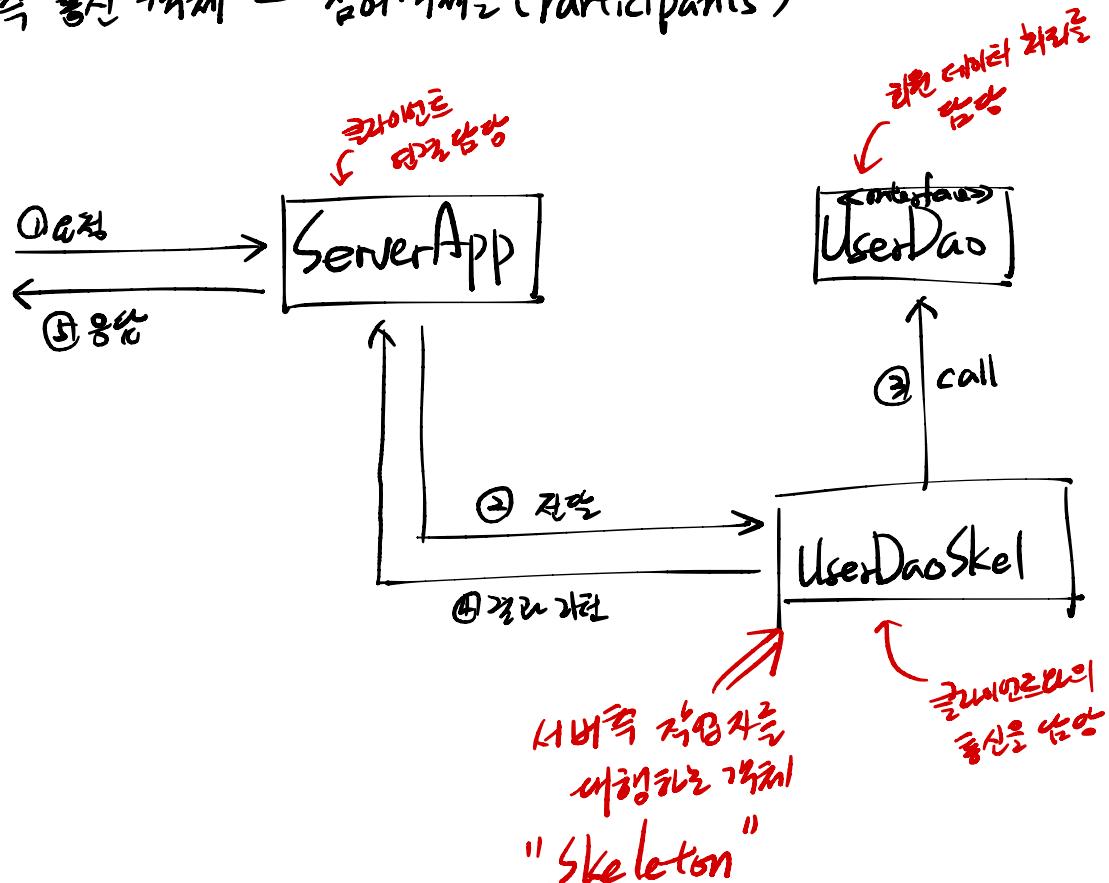
전송 데이터 - *

Server

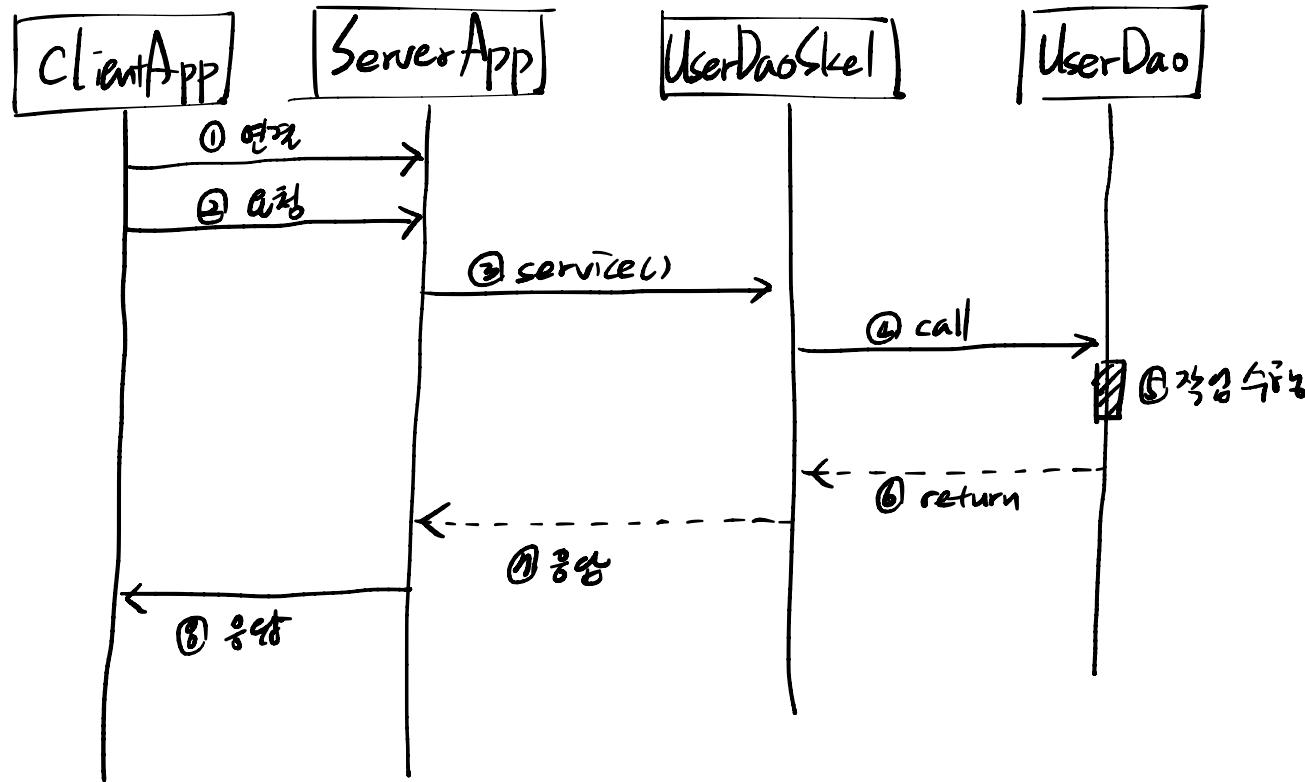
응답 상태 - "success" | "failure"

응답 데이터 - *

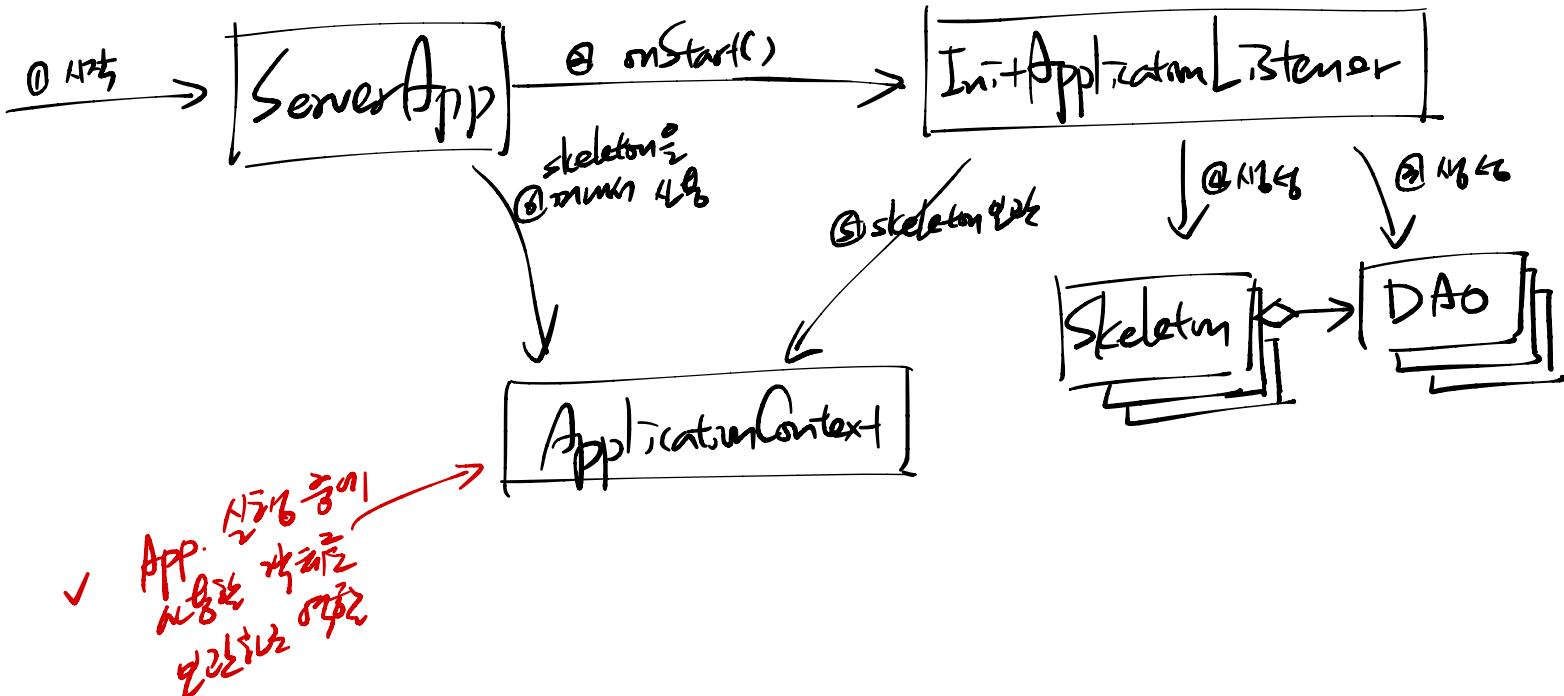
* 서버측 통신 구조 - 참여자들 (Participants)



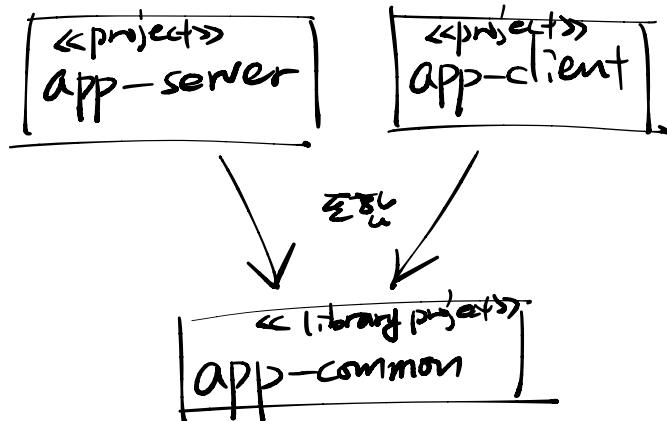
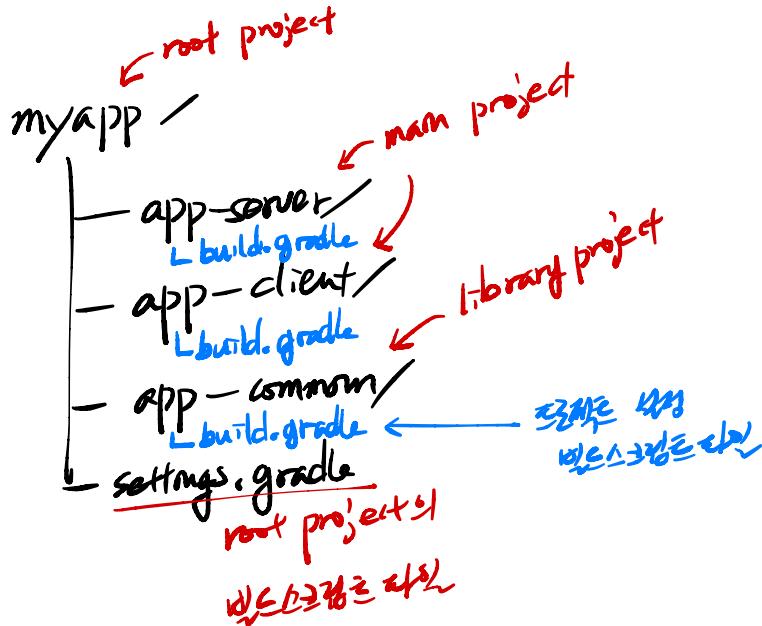
* 서버측 통신 구조 - Sequence Diagram (클라이언트가 서버의 서비스를 호출하는 과정)



* ServerApp in Skeleton

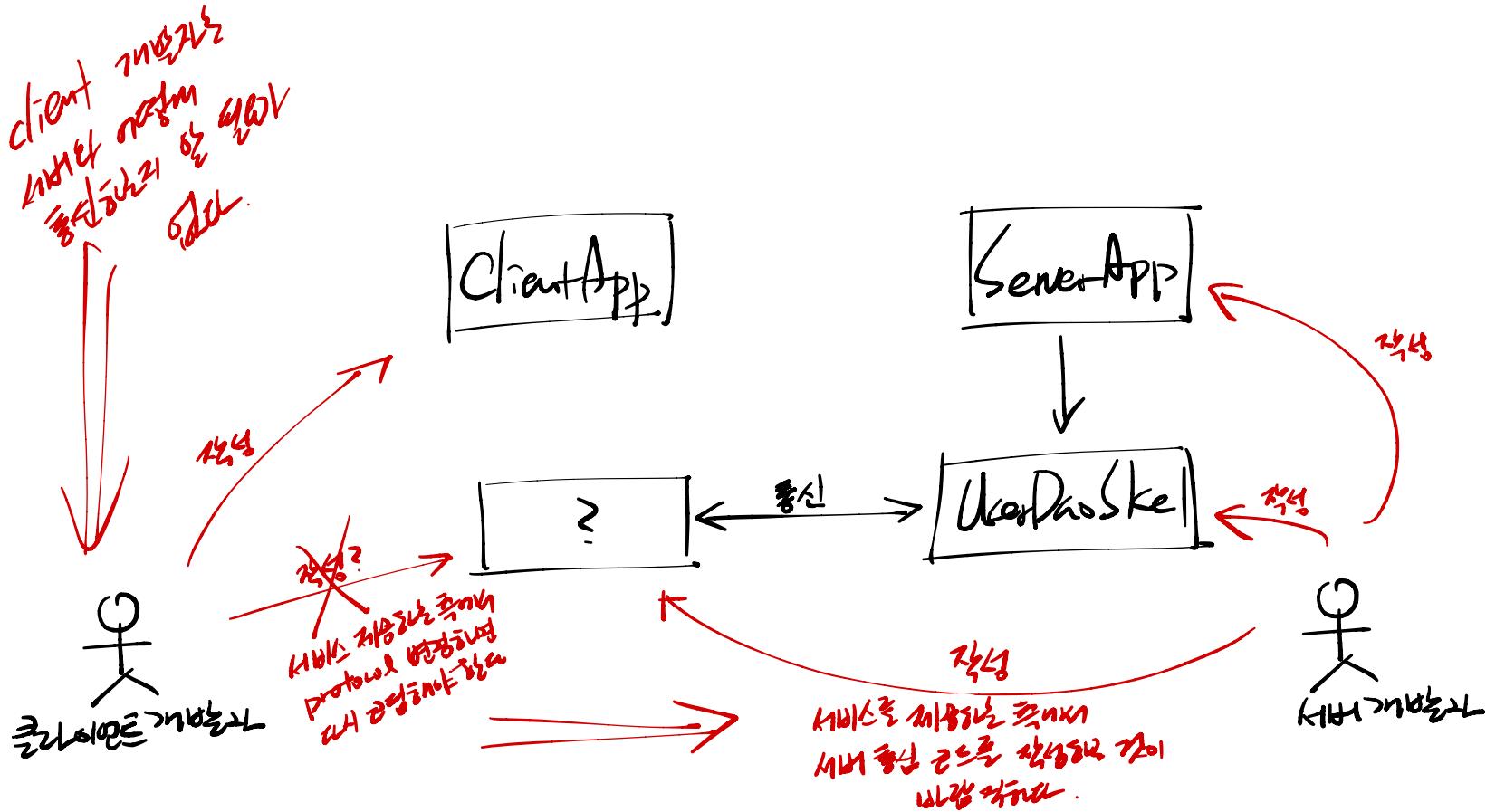


* 2019-2021 အချက် - အကျဉ်းချုပ်၏ ရေးဆွဲမှုပါမ်း

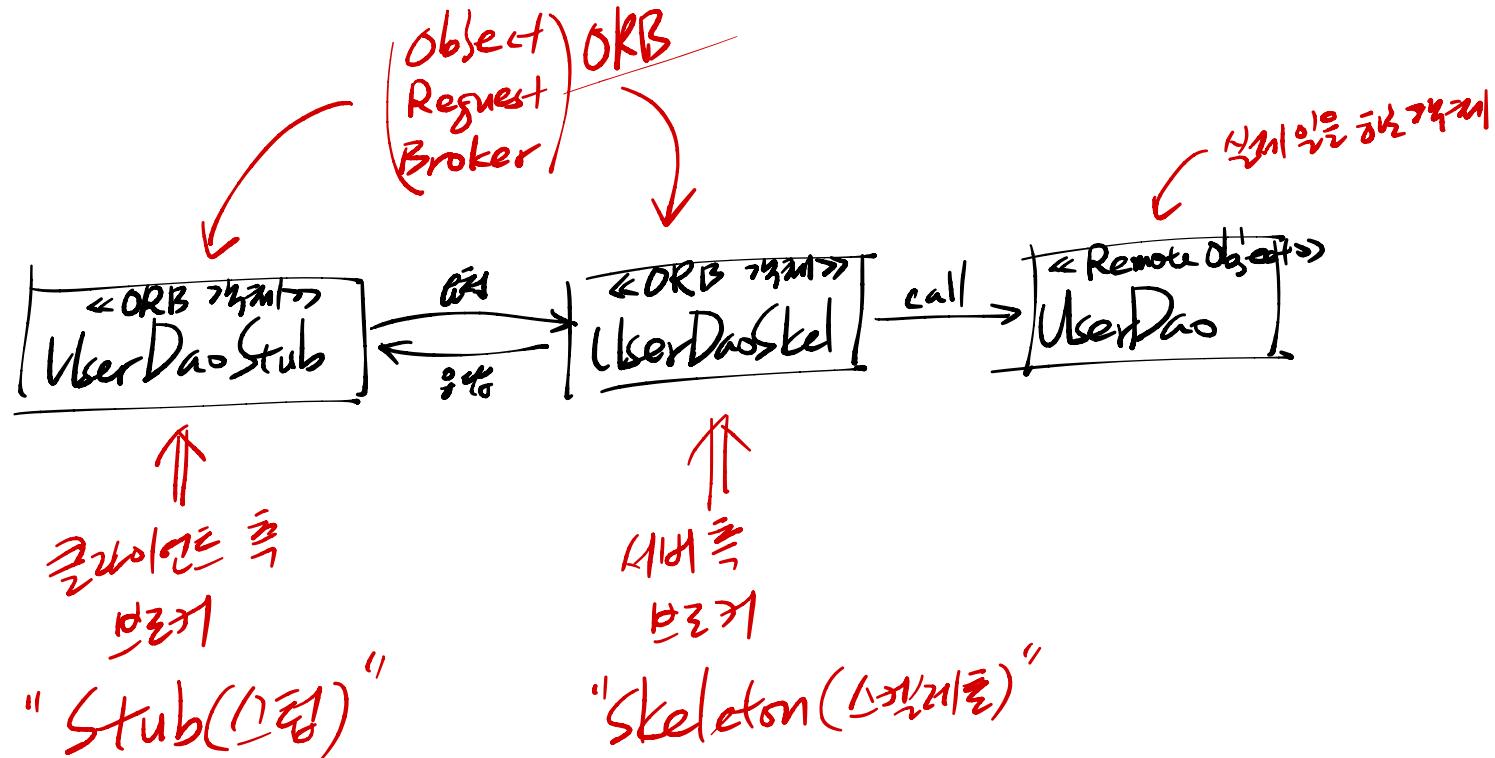


- VO ၁၃၈
- Net ၁၃၈

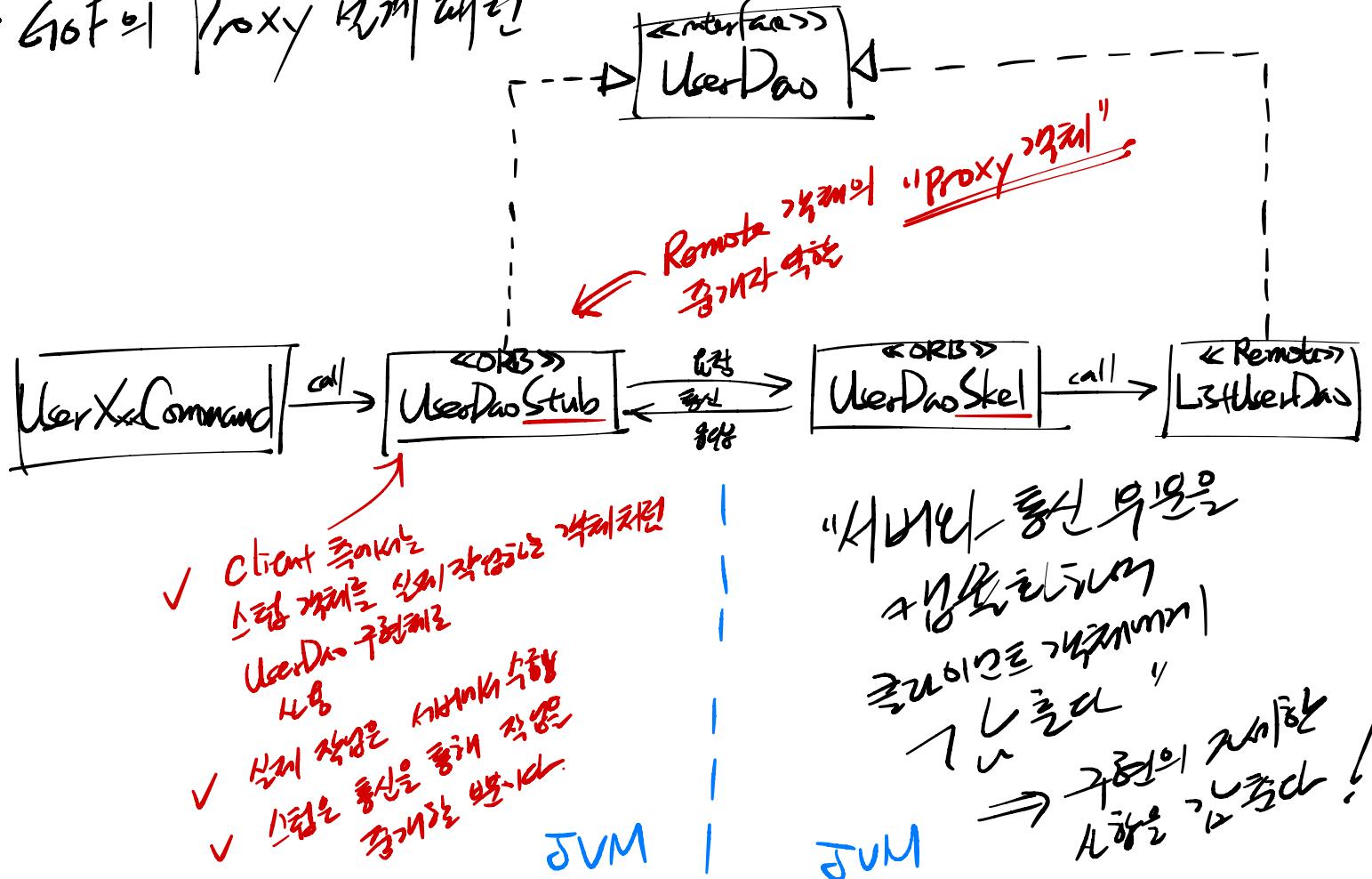
* Server 및 Client



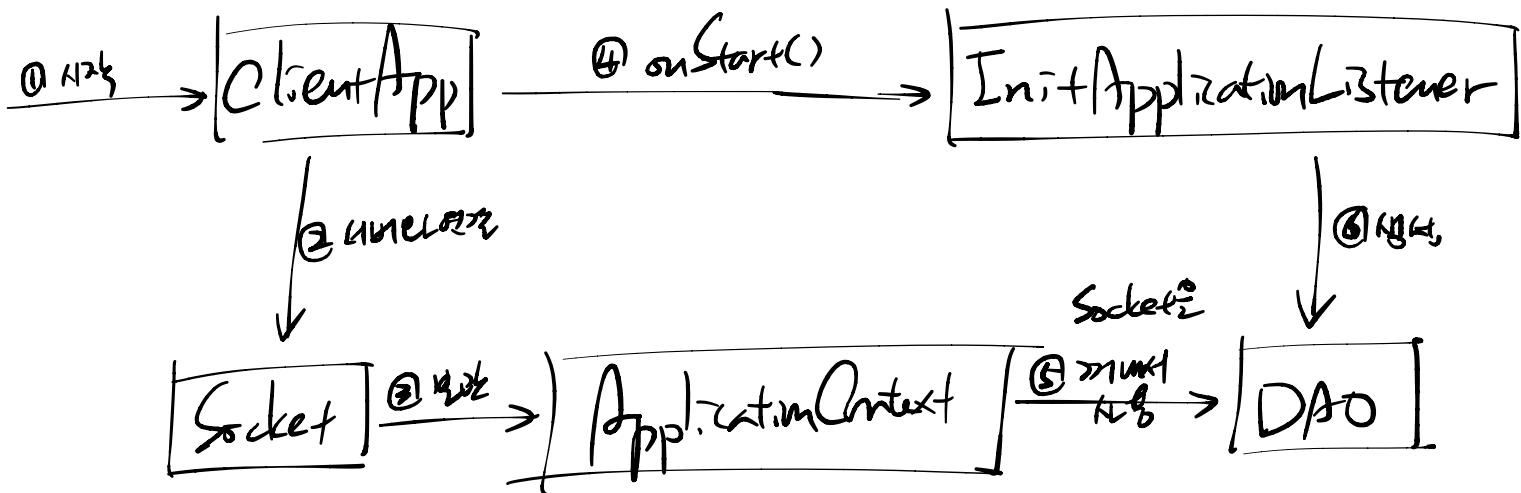
* Skeleton in Stub



* GOF의 Proxy 패턴 구현

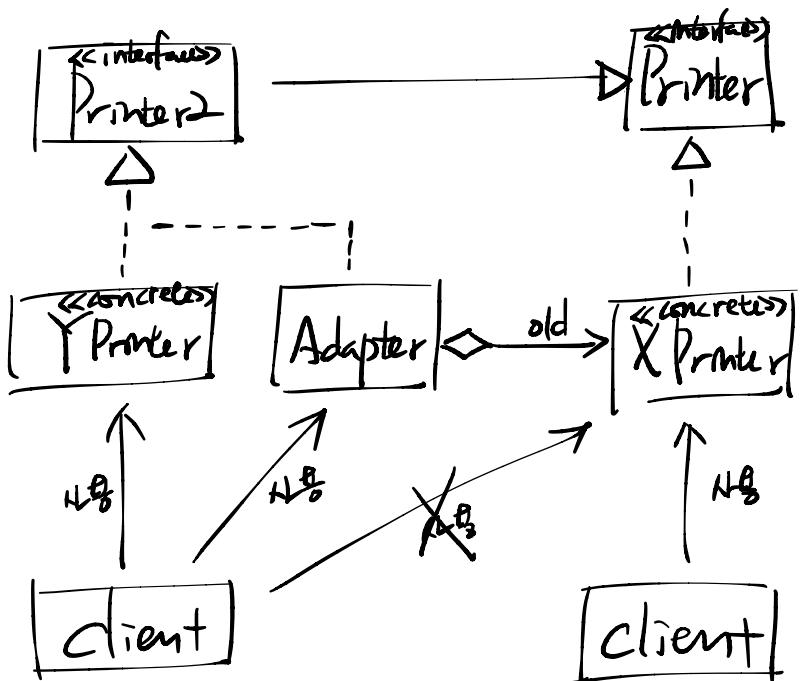


* Client Application & Stub

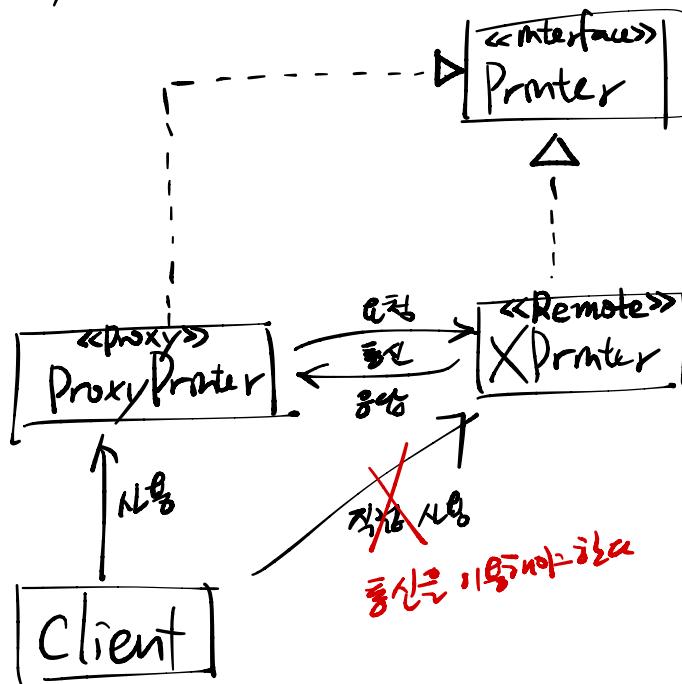


* Adapter 퍼런 vs Proxy 퍼런

① Adapter

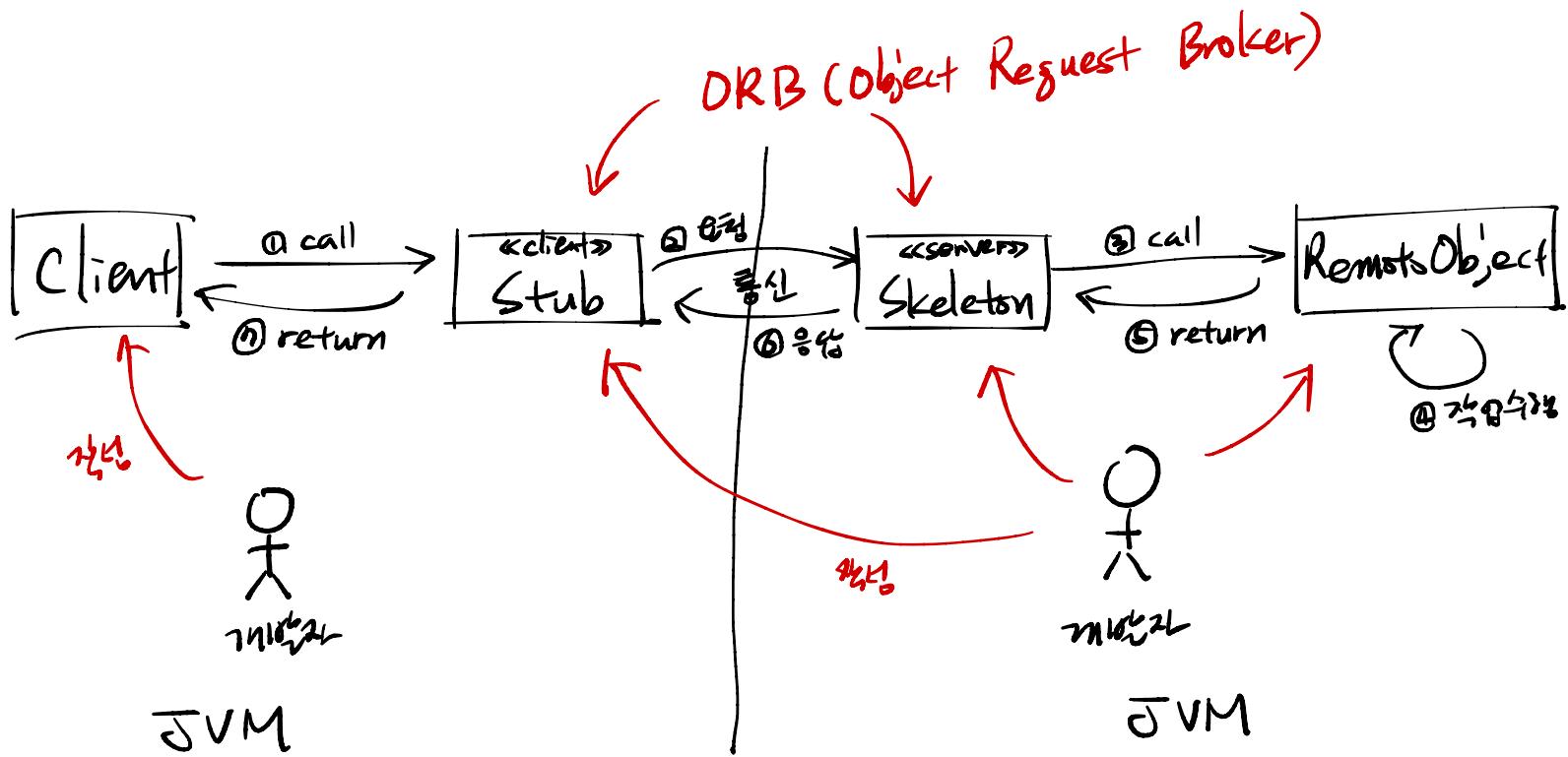


③ Proxy

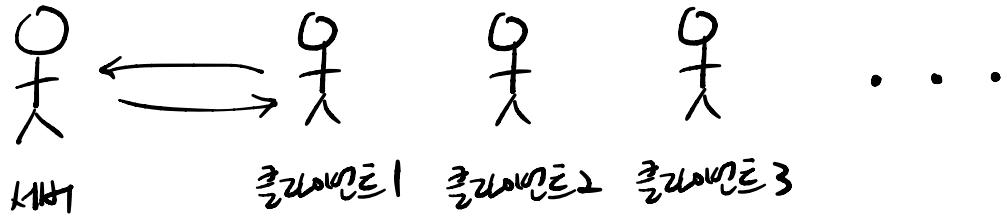


* Remote Object 구조

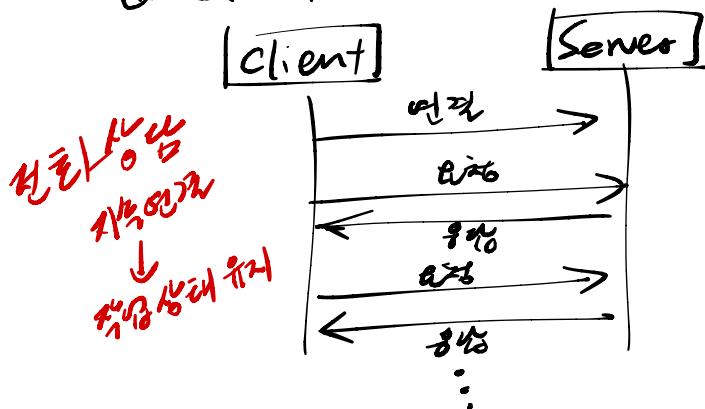
↳ 다른 JVM에서 만든 객체
(프록시)



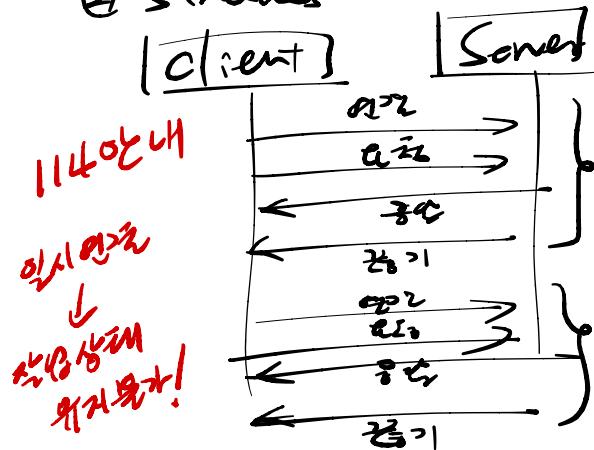
- * Stateful 웹서버 - 한번 연결한 후 여러번 요청/응답 가능
- Stateless 웹서버 - 한번 연결한 후 한번 요청/응답 가능



① Statefull

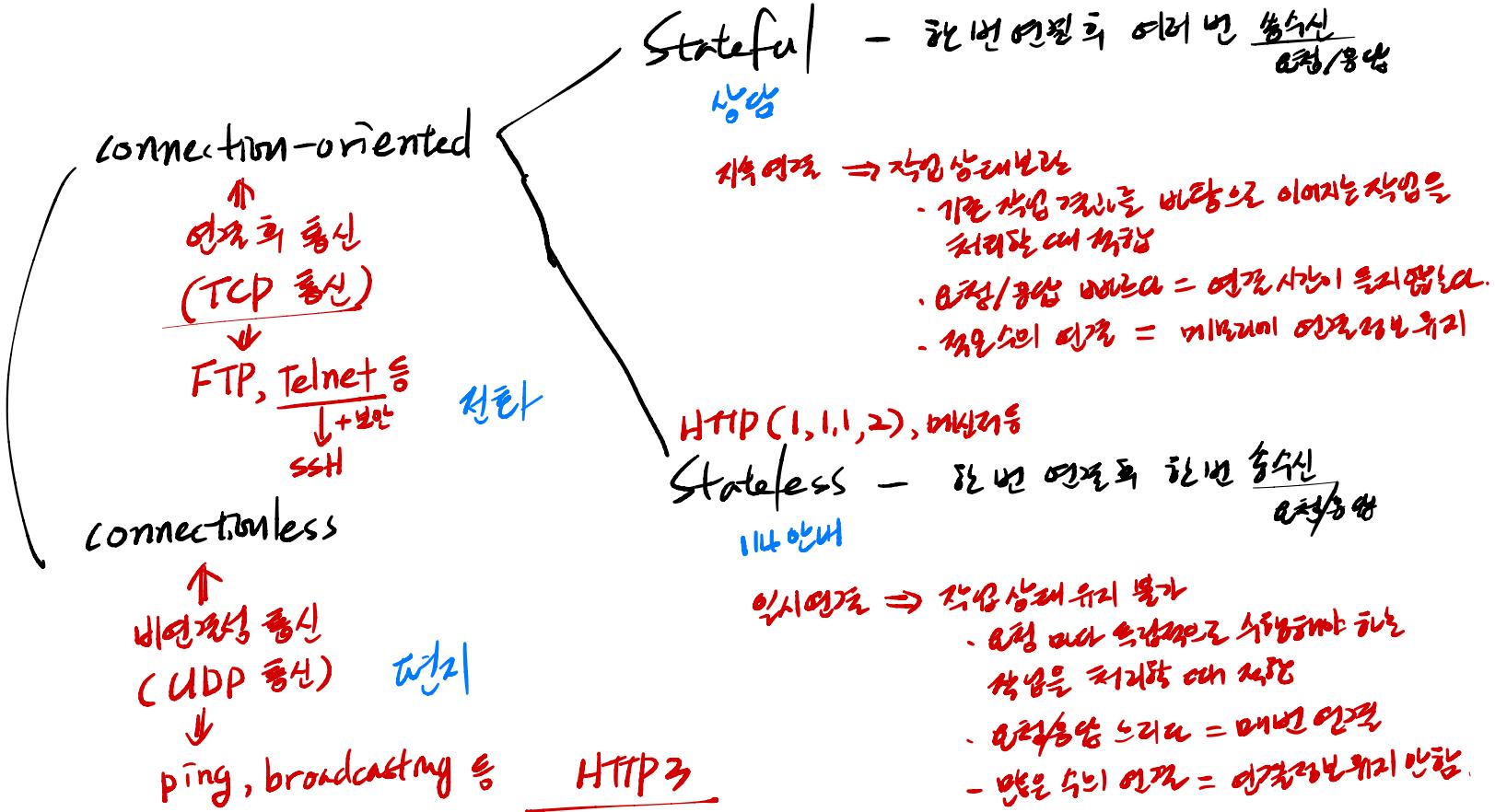


② Stateless



* 통신 18'시'

FTP, Telnet, SSH, 스트리밍, 채팅 등



* Socket API

