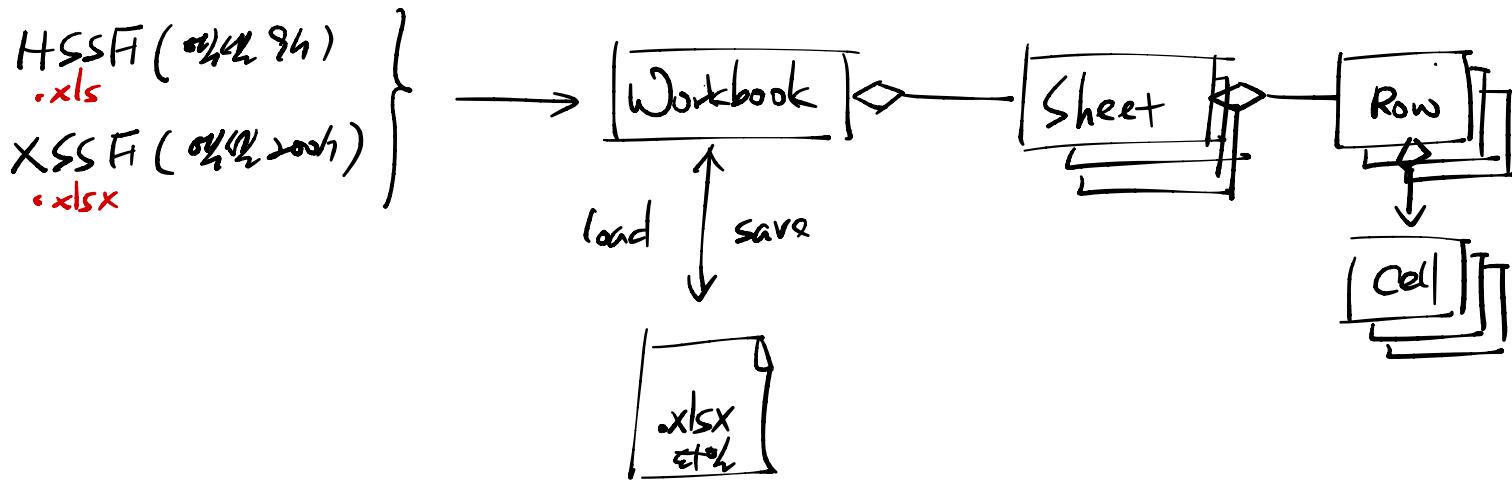


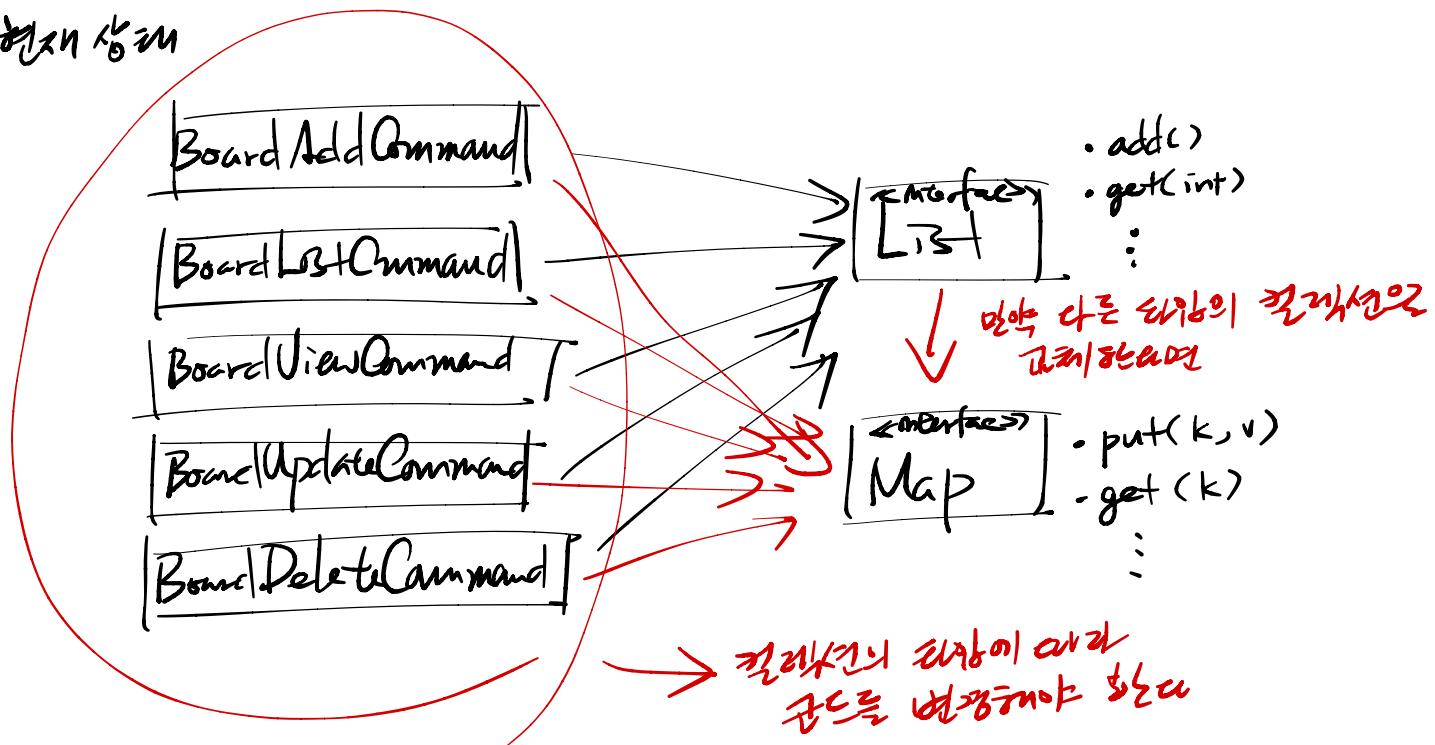
33. 콜렉터는 어떤 원칙을 따른다? : Apache POI 소개



34. DAO 가 필요한 이유

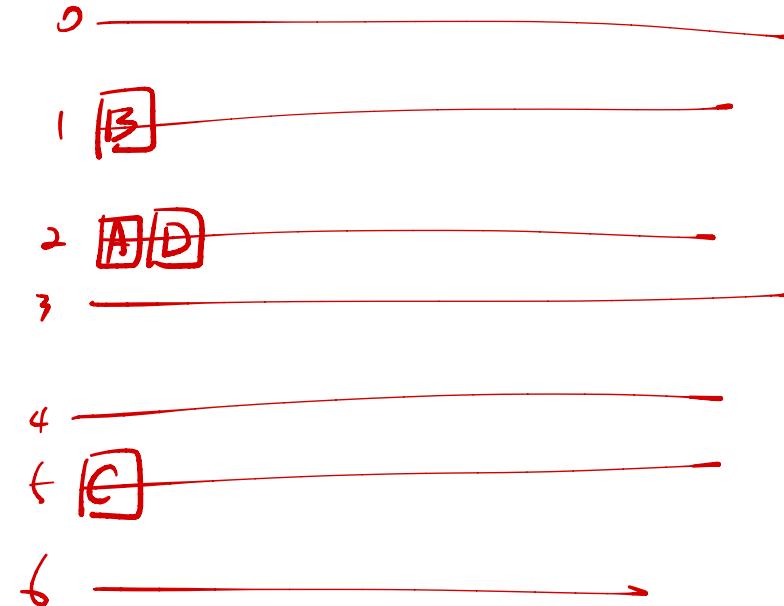
↳ 데이터 베이스를 관리하기

① 테이블



* Map의 데이터를 순회할 때는 항상
꺼내서 접근해야!

Map



② A

8-B

5-C

23-D

values()

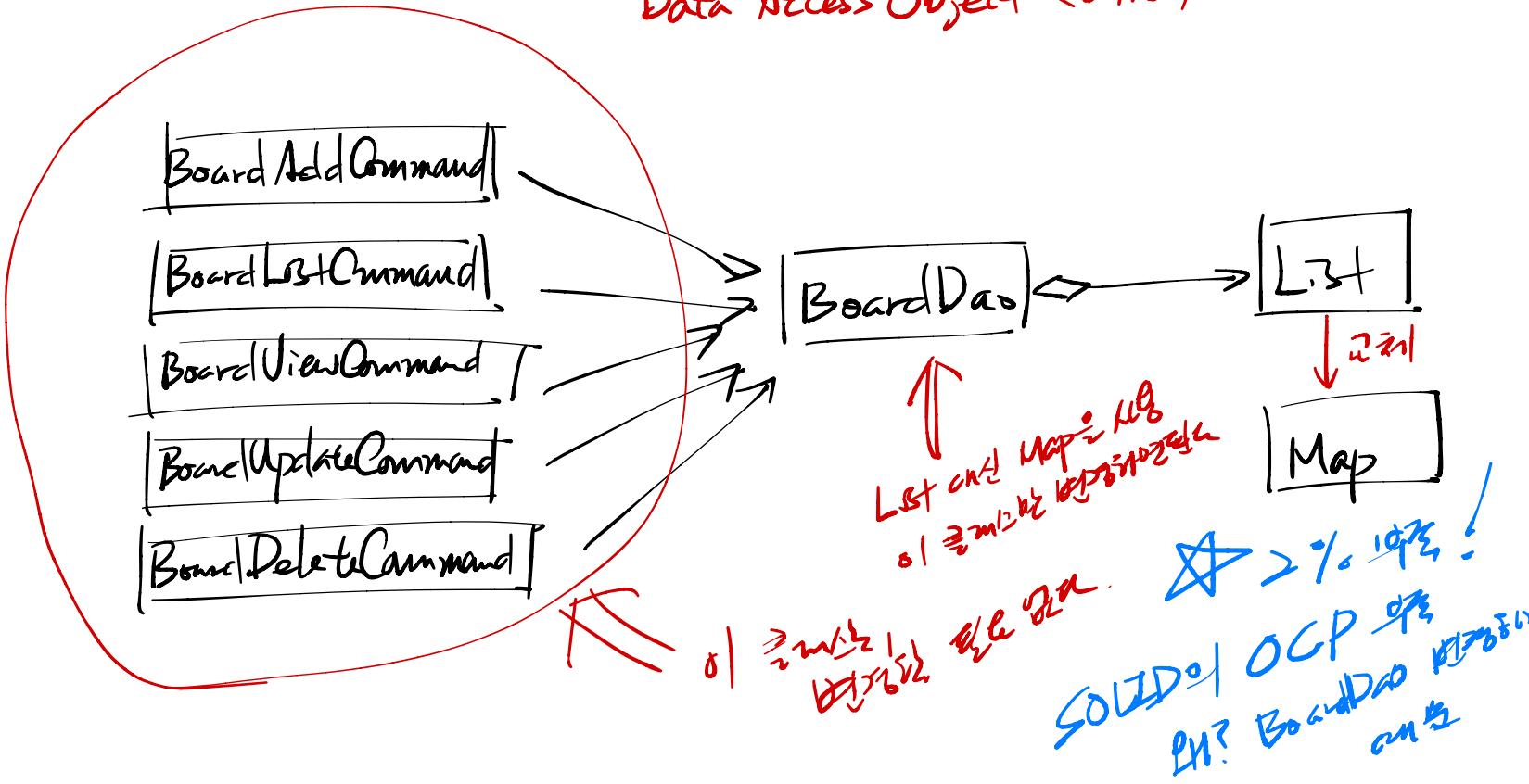
B A D C

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

35. 글로벌 키워드를 제거해보자

↳ 제거한 키워드는 뭘까? → 함수가 기반이다.

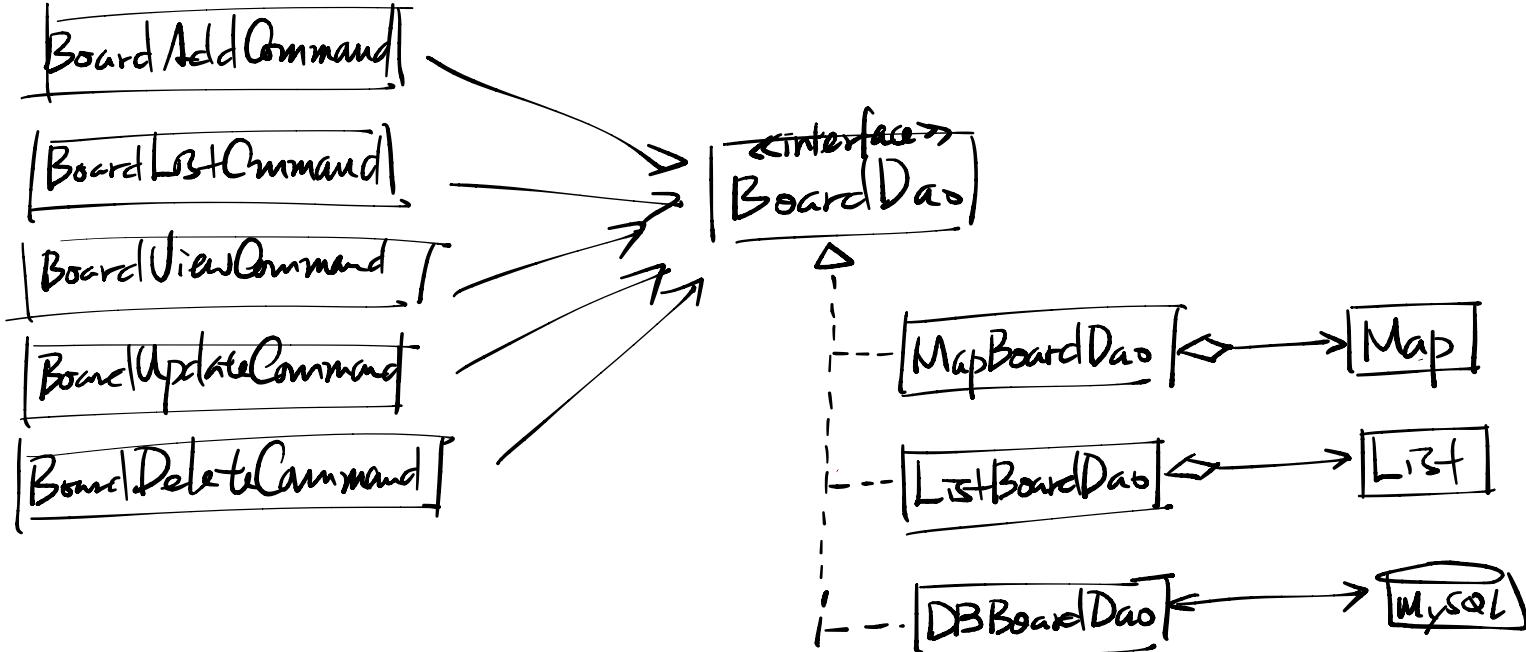
Data Access Object (DAO)



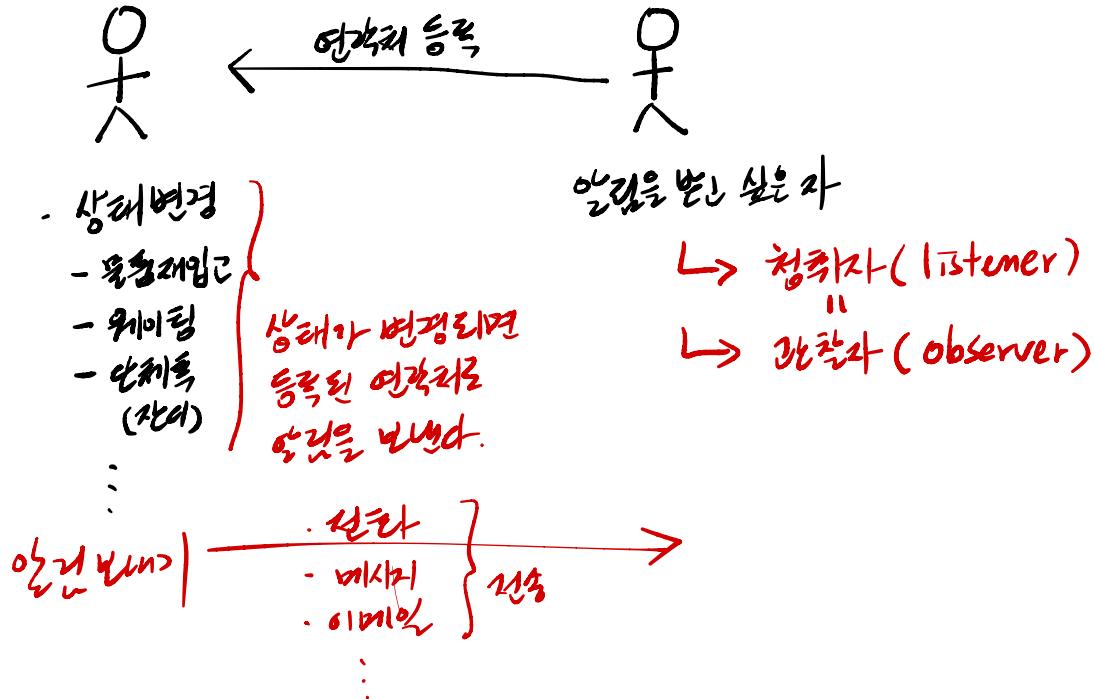
35. 글로벌 키워드는 final이면 됨

↳ 제한된 개수의 경우 → 제한된 개수의 제한된 개수.

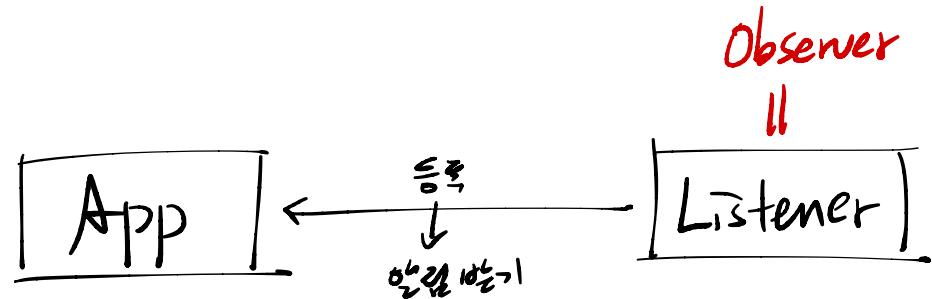
Data Access Object (DAO)



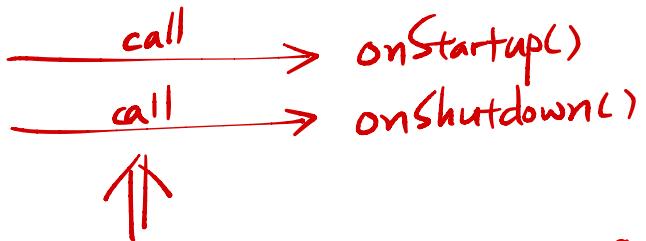
* 36. 알림 패턴 : GOF의 Observer 패턴
실무 활용 예)



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

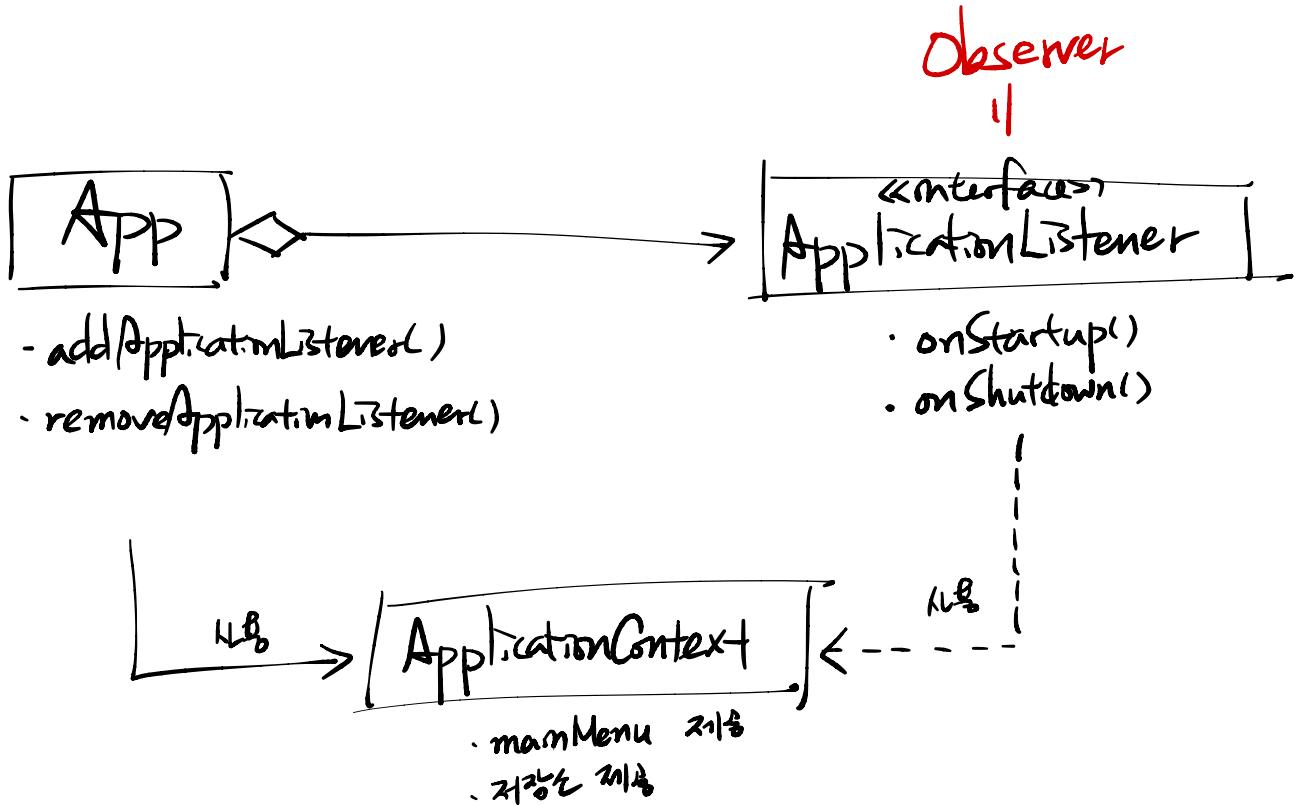


- 설정에 따라
- 사용자
- 풍경



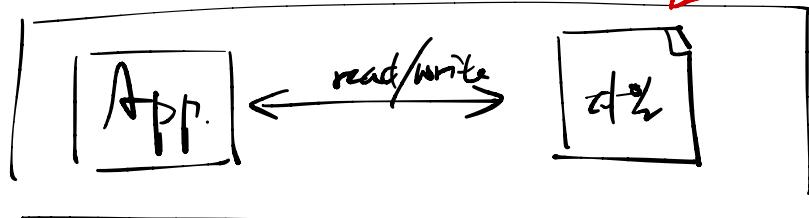
설정에 따라 설정에 따라 설정에 따라
설정에 따라 설정에 따라 설정에 따라

* GOF의 Observer 패턴 대처



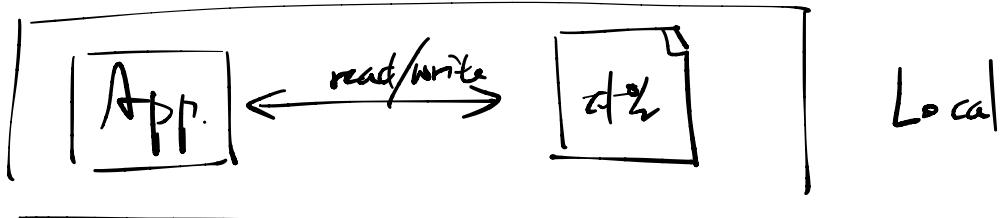
31. Application 간에 데이터 공유

현재 상태



App.의
데이터로 데이터를
Local

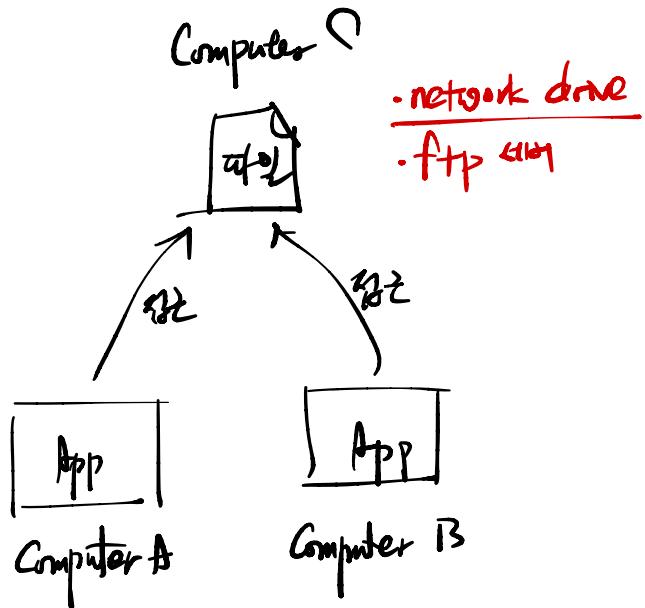
App.-간의
데이터 공유는!



•
•
•

31. Application 간에 파일 교환

① 파일로드된 파일 공유



※ 문제점

- 동시에 여러 App. 실행
문제점이 있다



파일을 쉽게 쓸 수 있다.

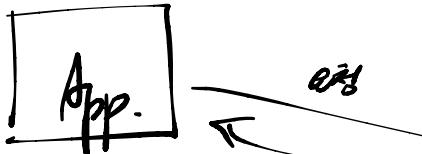
31. Application 간에 데이터 교환

② 멀티프로세스 애플리케이션

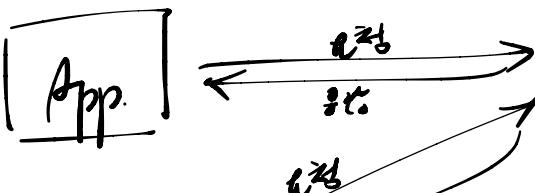
- networking prog.
- multi-tasking prog.
 - ↳ multi-threading
 - ↳ synchronous

애플리케이션
Data는 쓰고 읽을 때
Data를 주고 받음

Computer A



Computer B

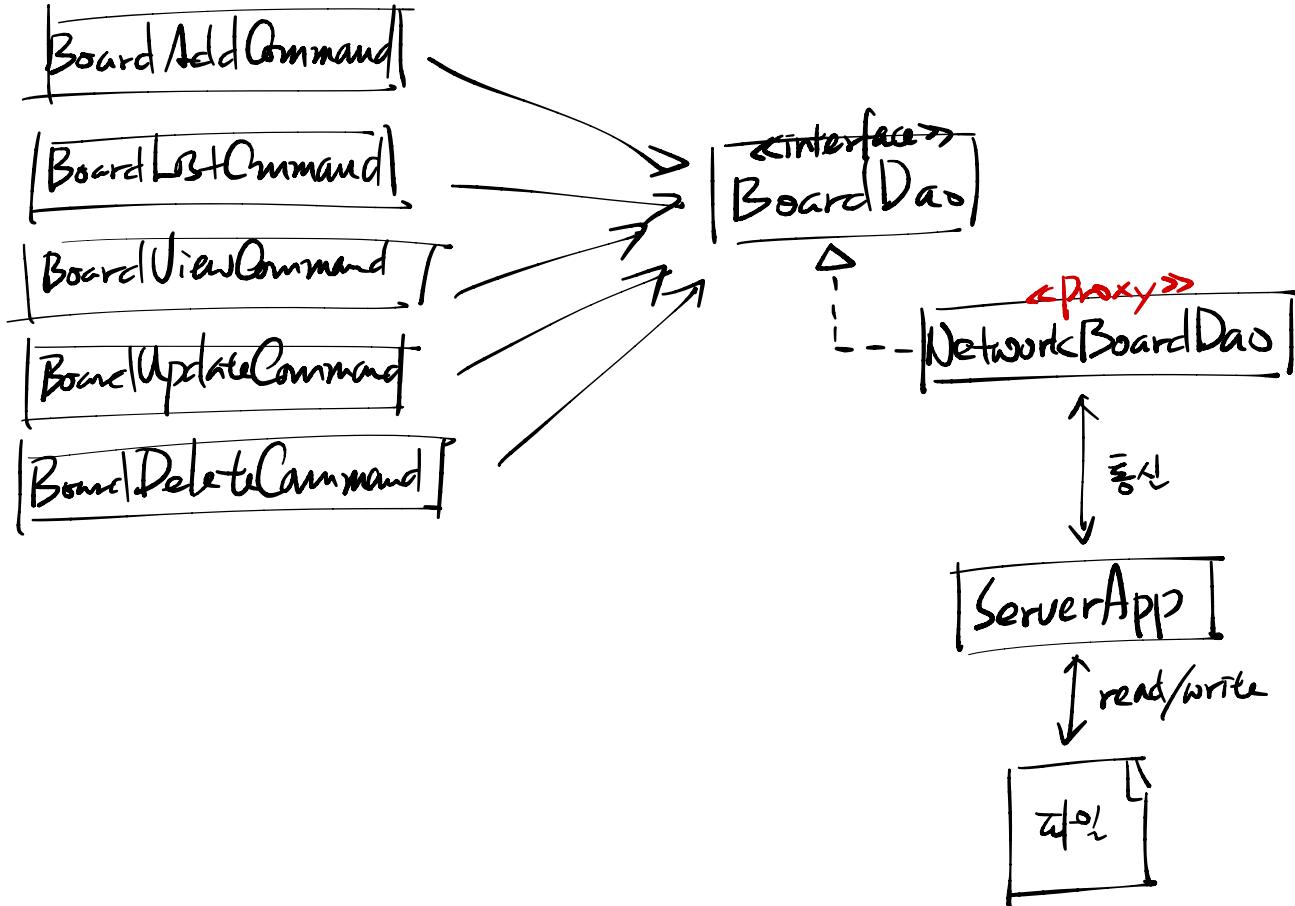


Computer C

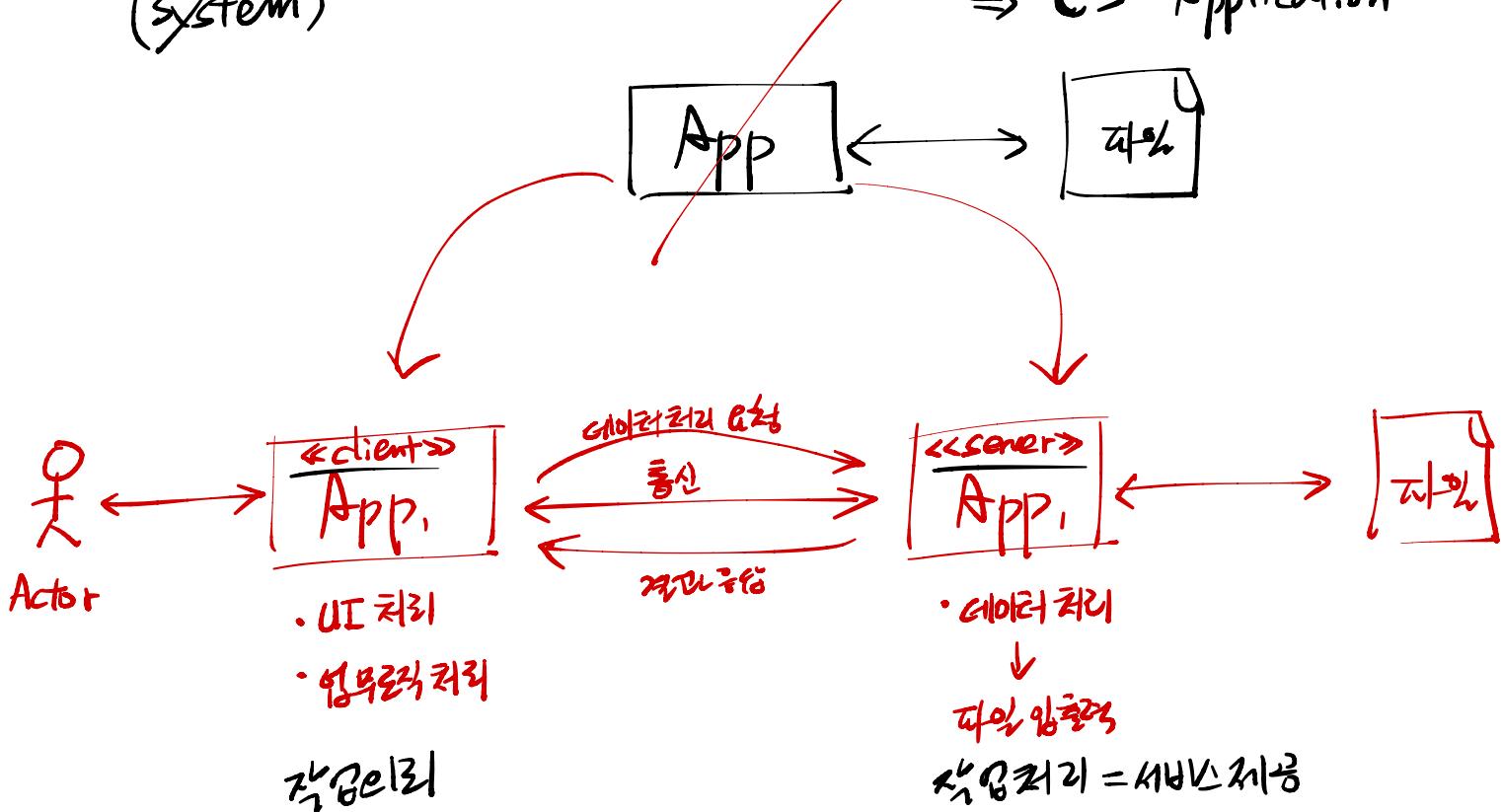


networking

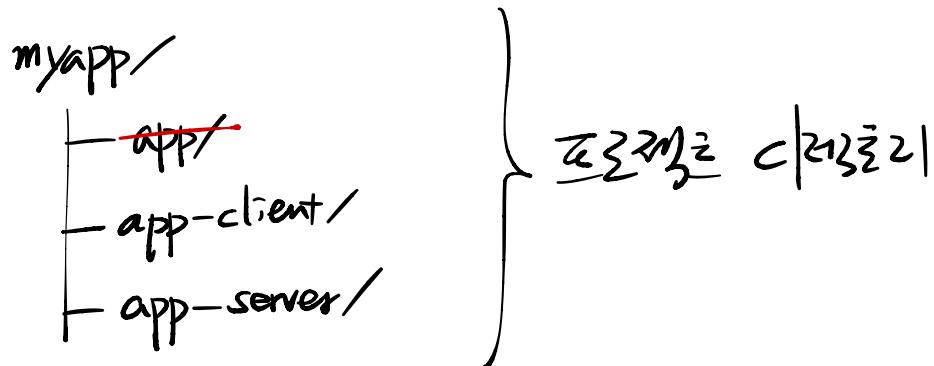
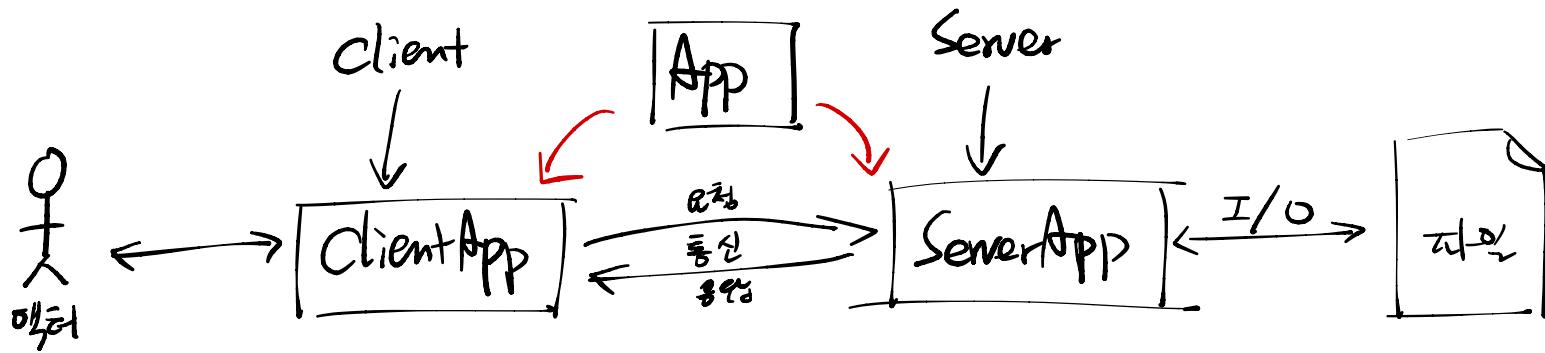
단계 32번의 단계 32번 일정 시스템
(process) (process) 단계 32번
단계 32번의 단계 32번 일정 시스템
(process) (process) 단계 32번



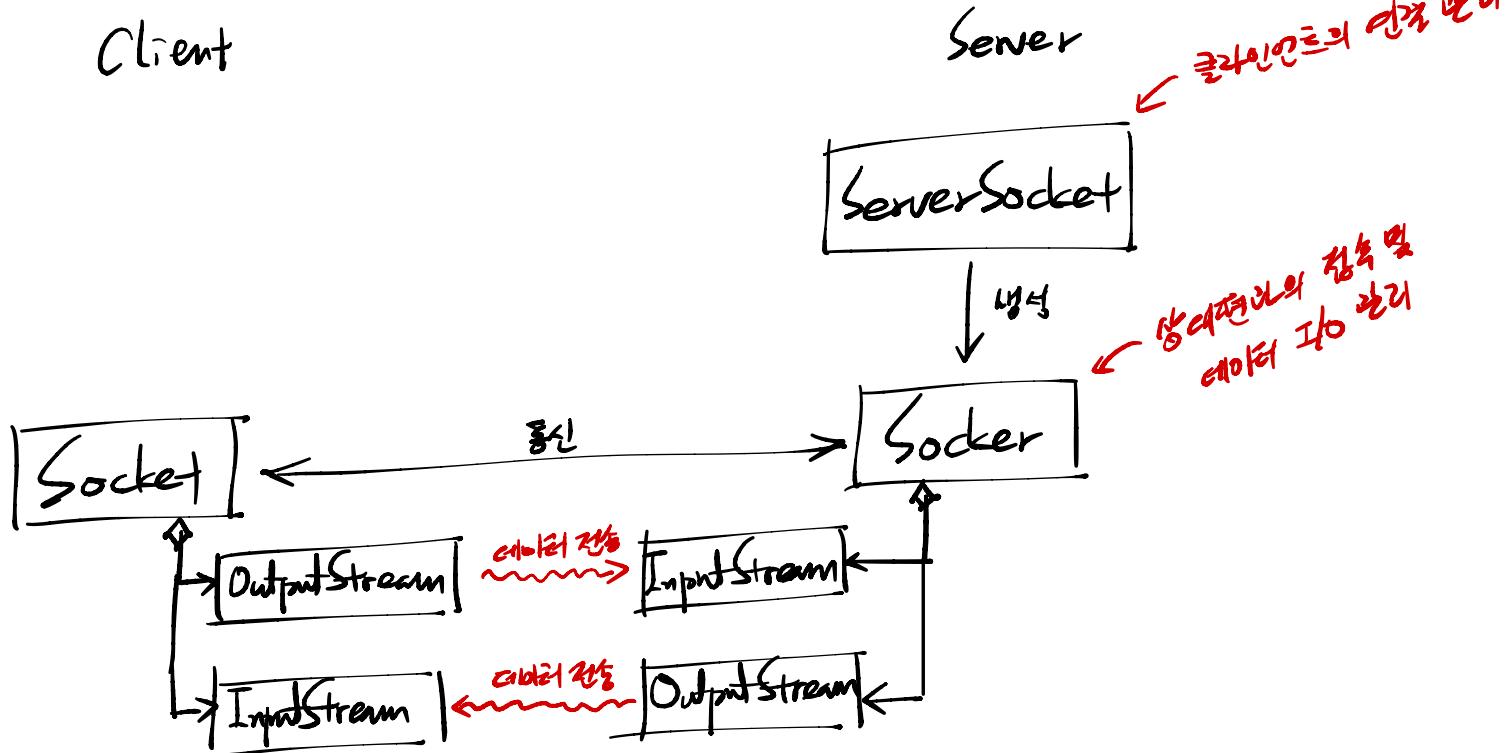
* Software Architecture : Client / Server Architecture
(System) ⇒ CS Application



* CS Architecture 구조

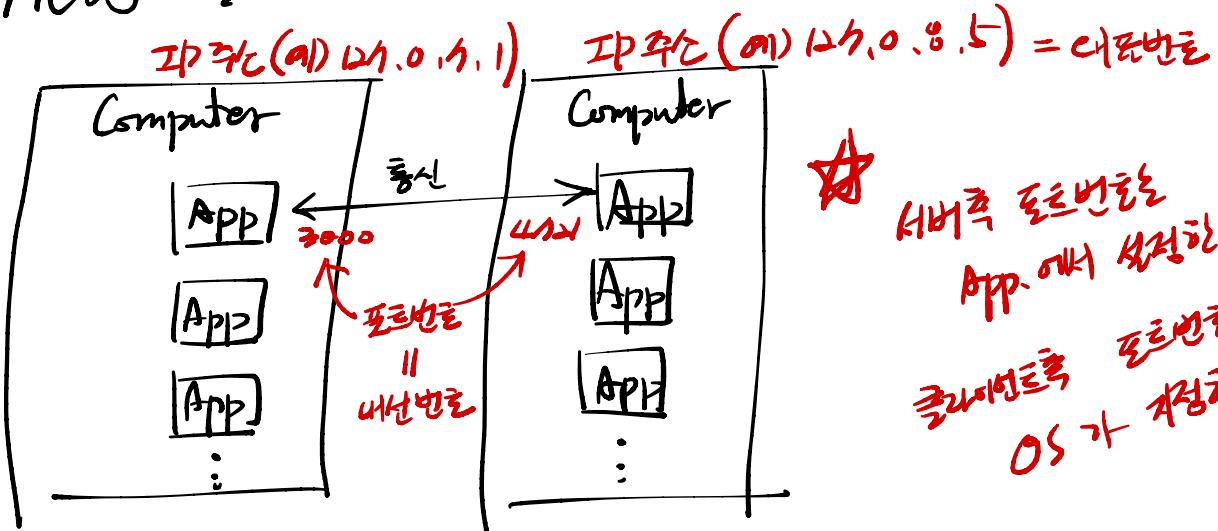


* Networking API API



* 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?
(로컬호스트)

OS가 제공합니다.

* 특별한 IP 주소

127.0.0.1

Local
주소
||
로컬 호스트 주소

||
"localhost"

* Protocol : 데이터 송수신 규칙

client

컬렉션 - "users"

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

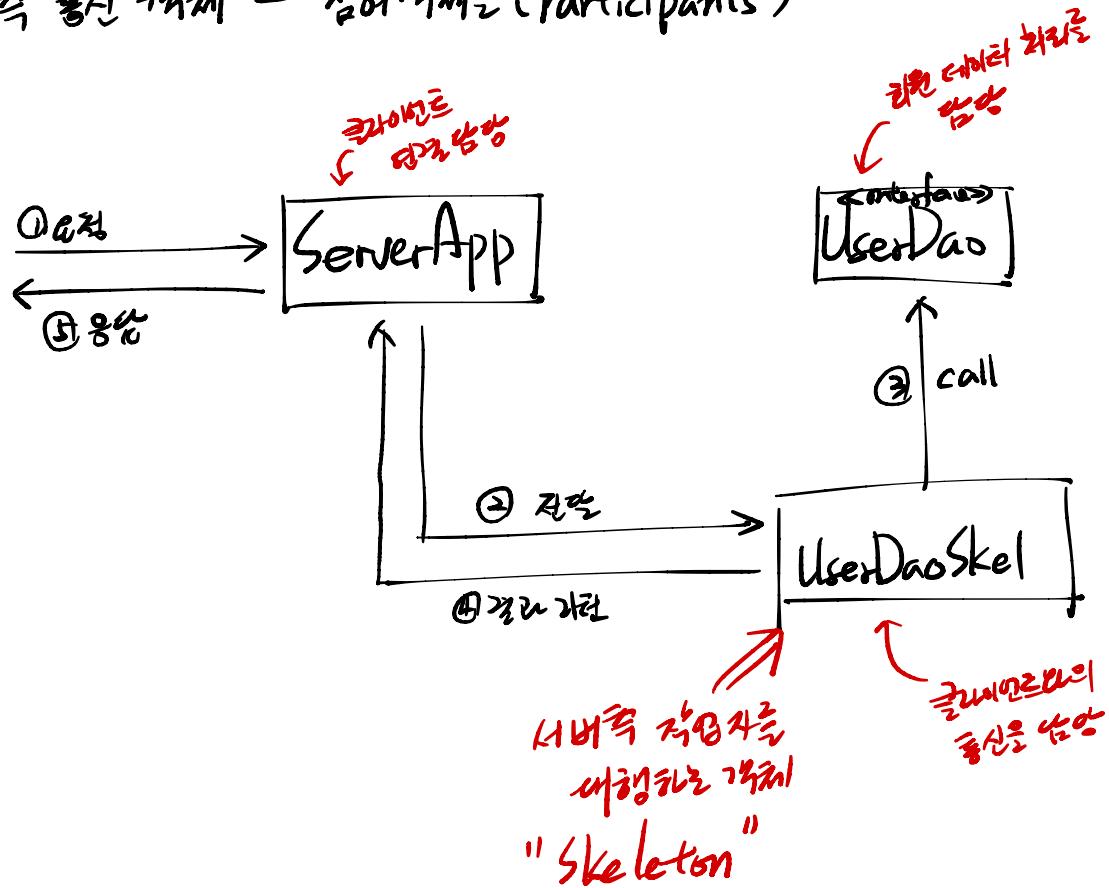
전송 데이터 - *

Server

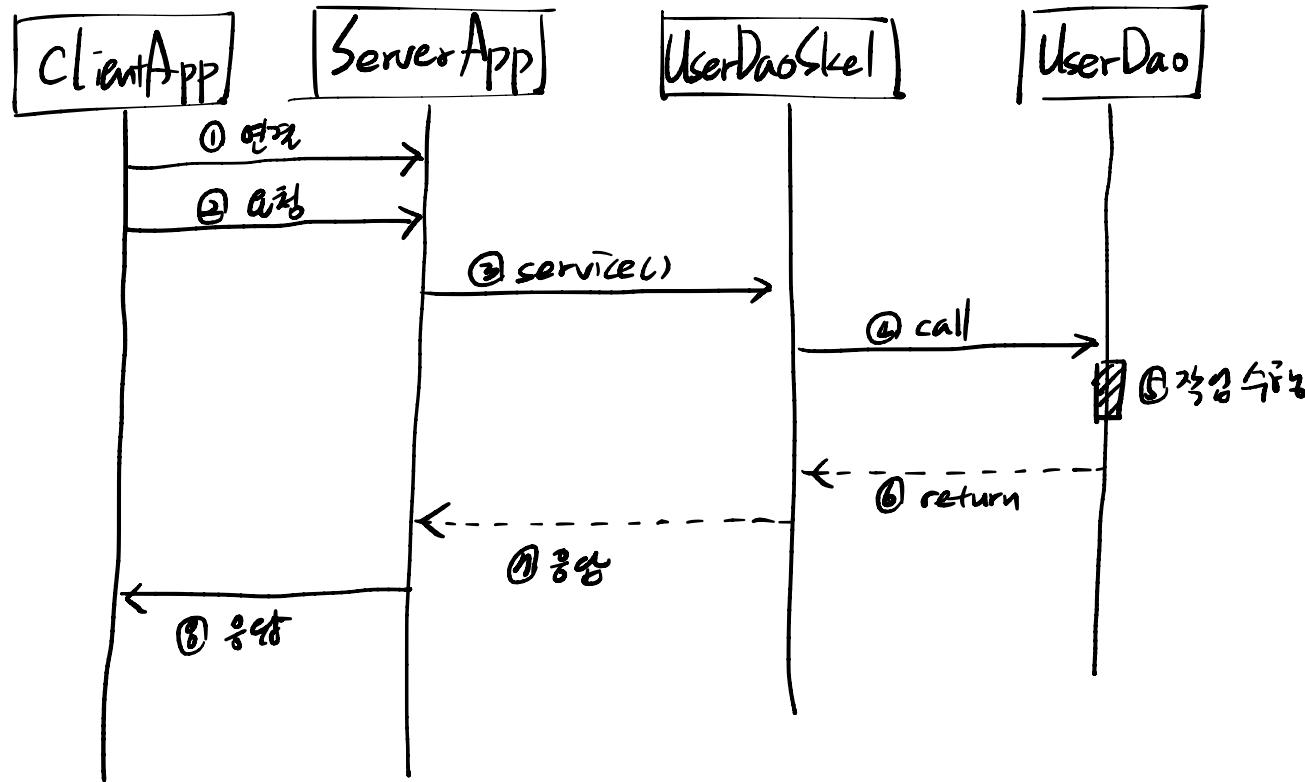
응답 상태 - "success" | "failure"

응답 데이터 - *

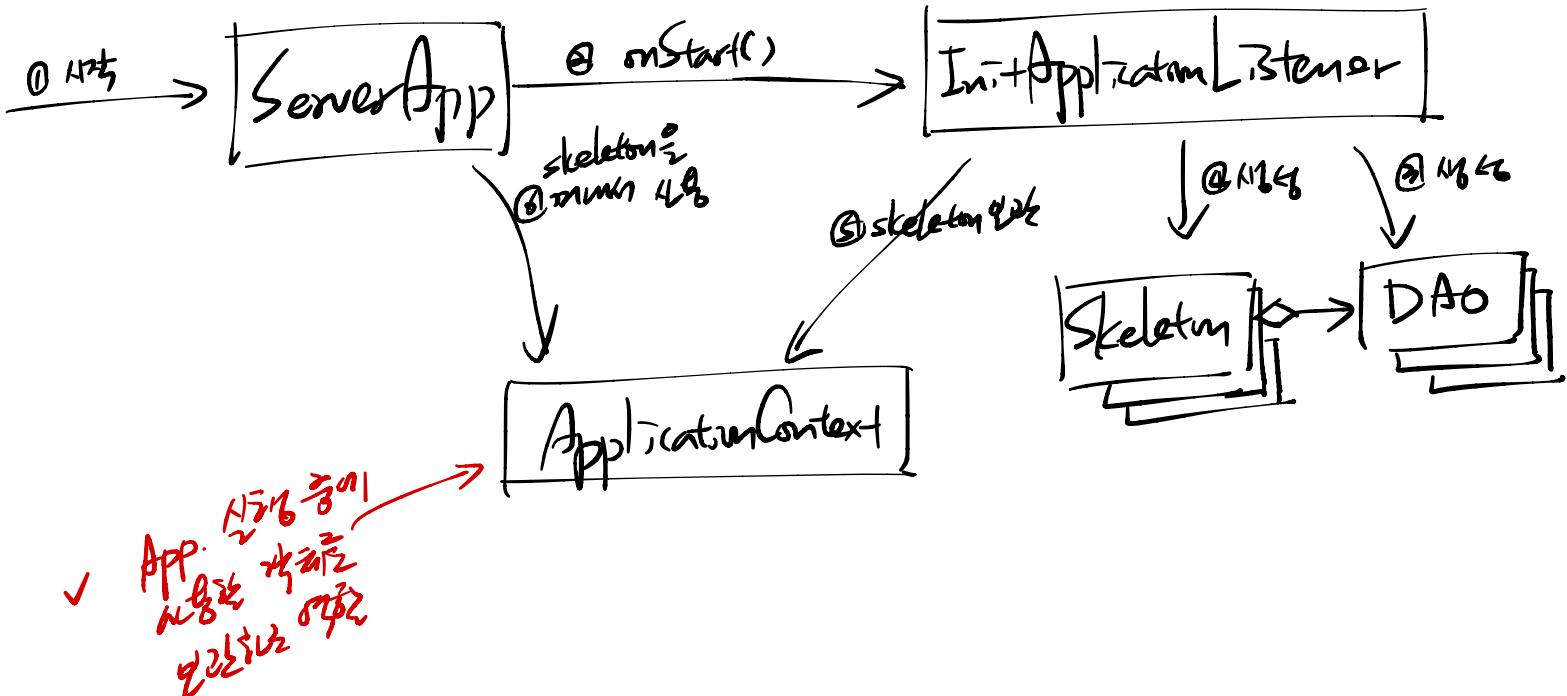
* 서버측 통신 구조 - 참여자들 (Participants)



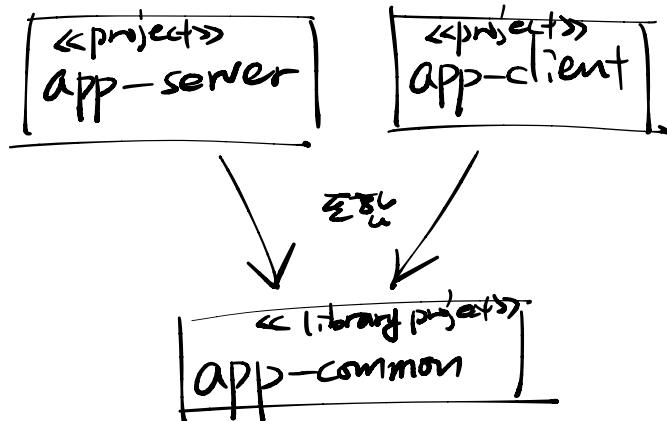
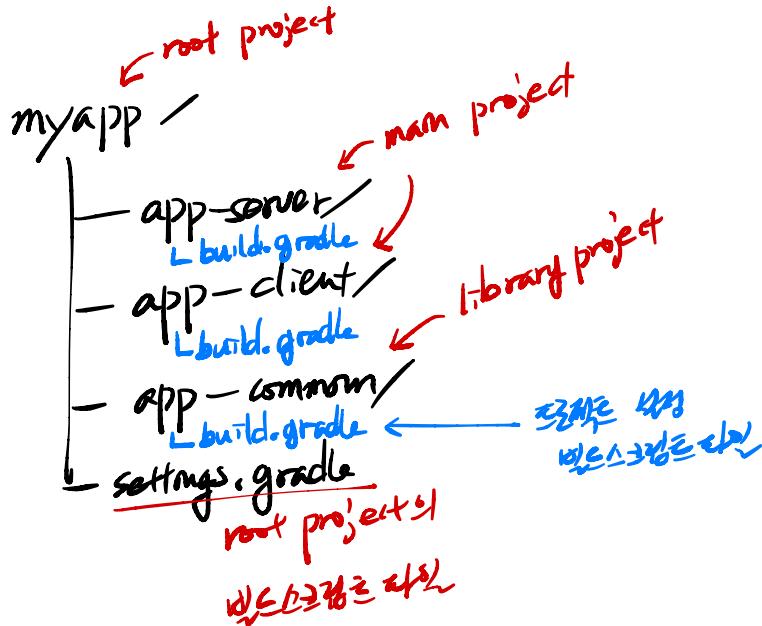
* 서버측 통신 구조 - Sequence Diagram (클라이언트가 서버의 서비스를 호출하는 과정)



* ServerApp in Skeleton

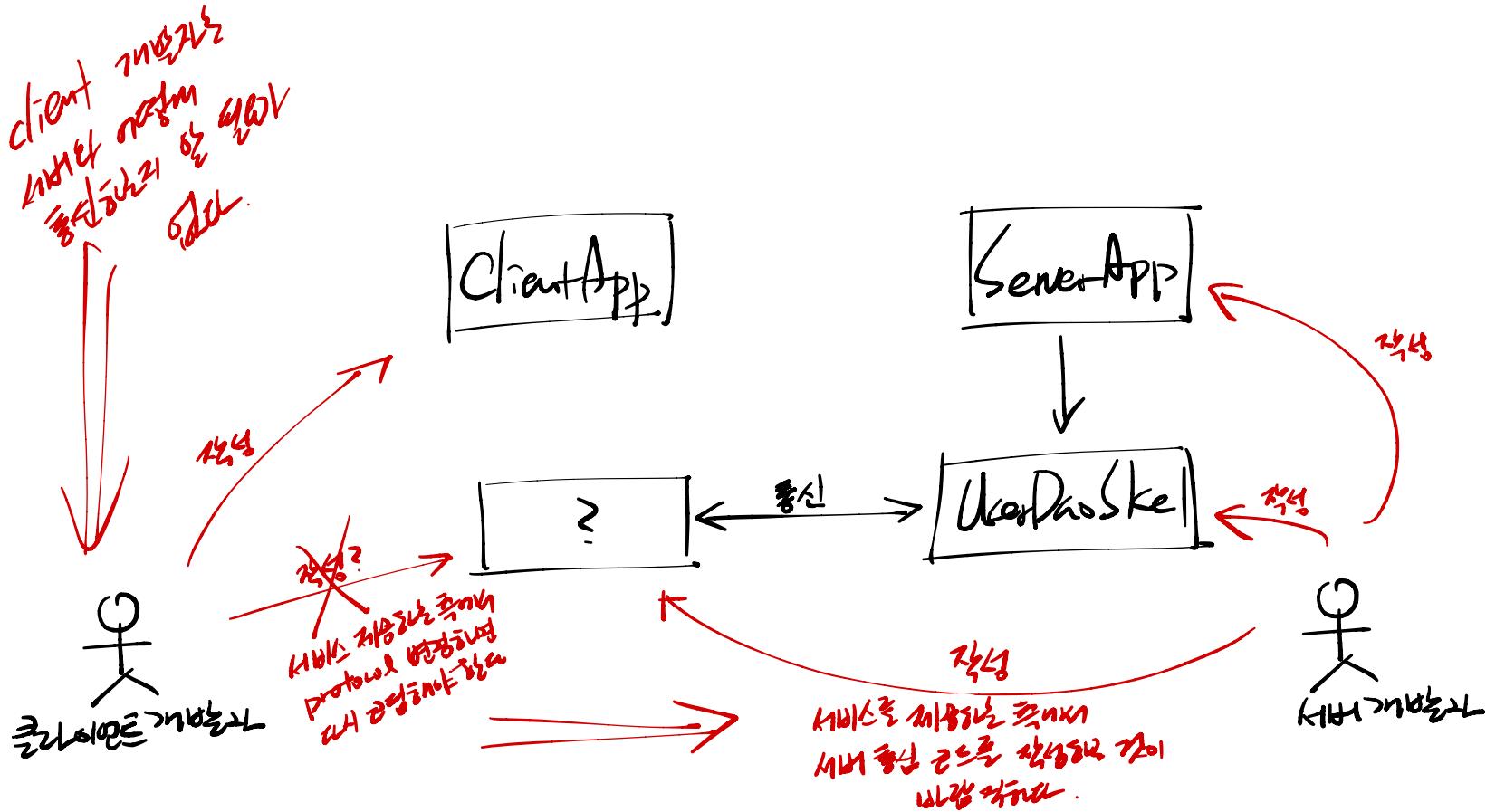


* 2019-2021 အချက် - အကျဉ်းချုပ်ရေး ပုဂ္ဂနိုင် စာမျက်နှာ

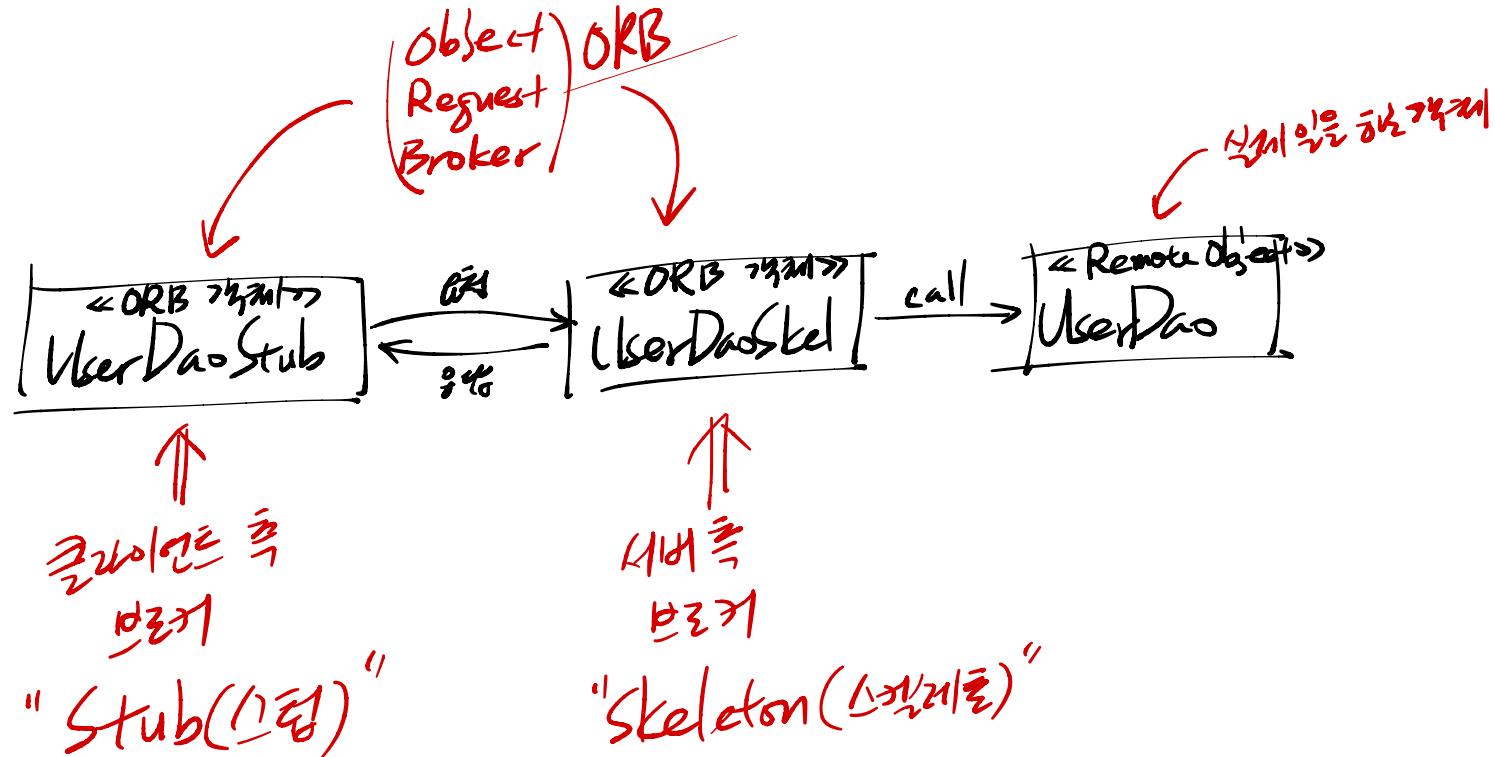


- VO > ၃၃၈
- Net > ၃၃၈

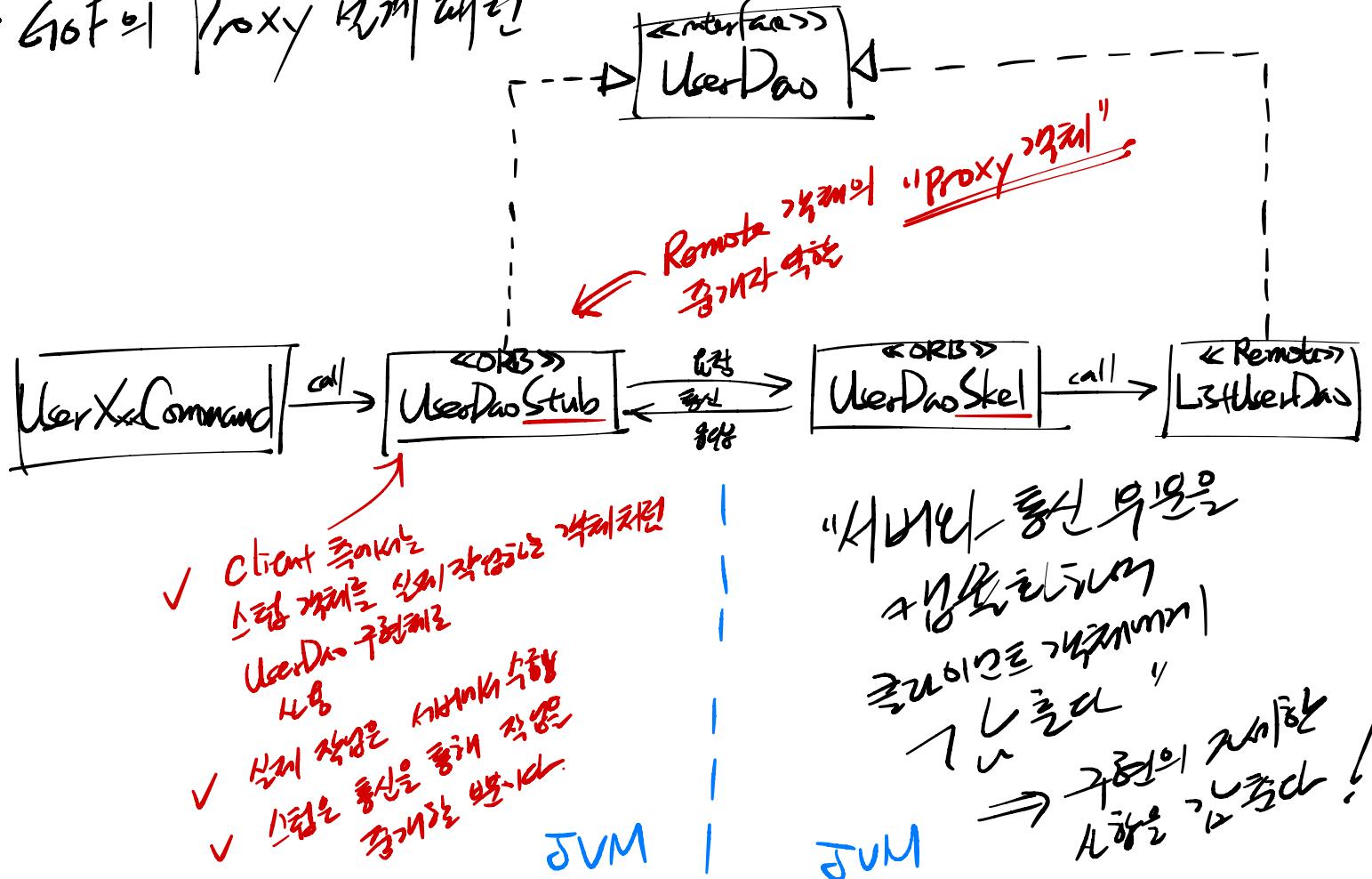
* Server 및 Client



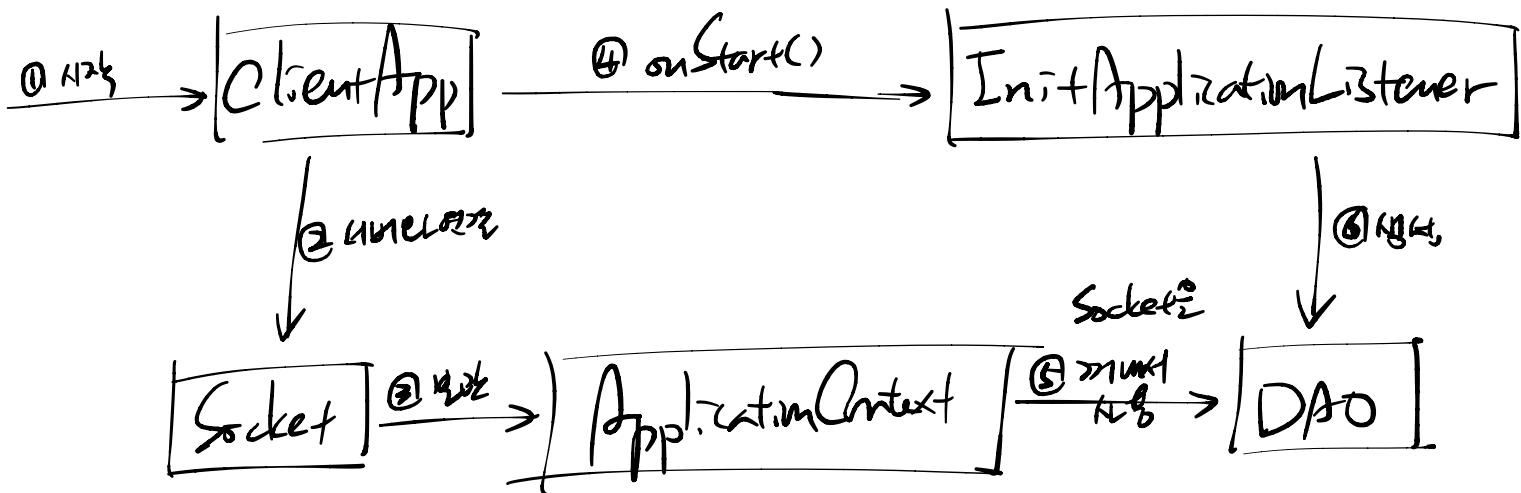
* Skeleton in Stub



* GOF의 Proxy 패턴 구현

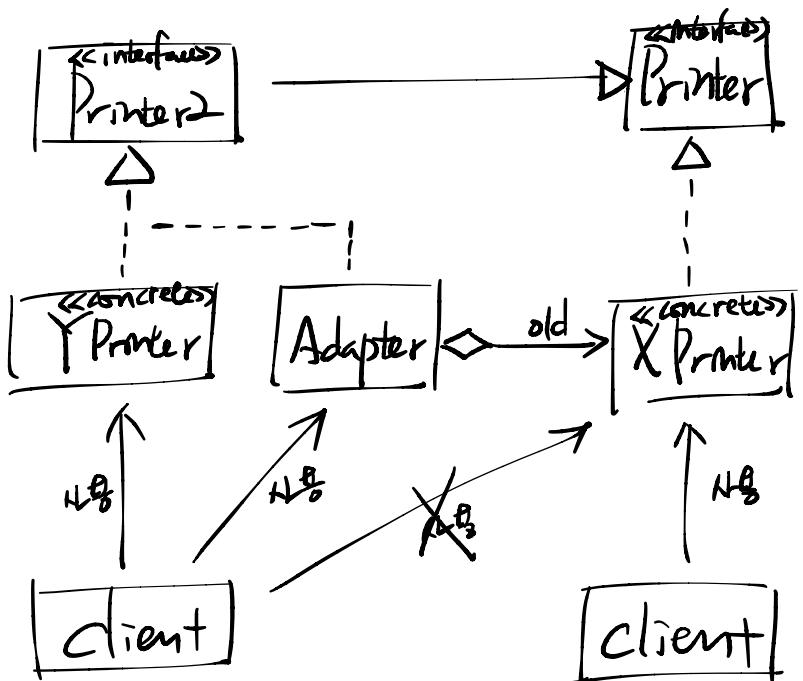


* Client Application & Stub

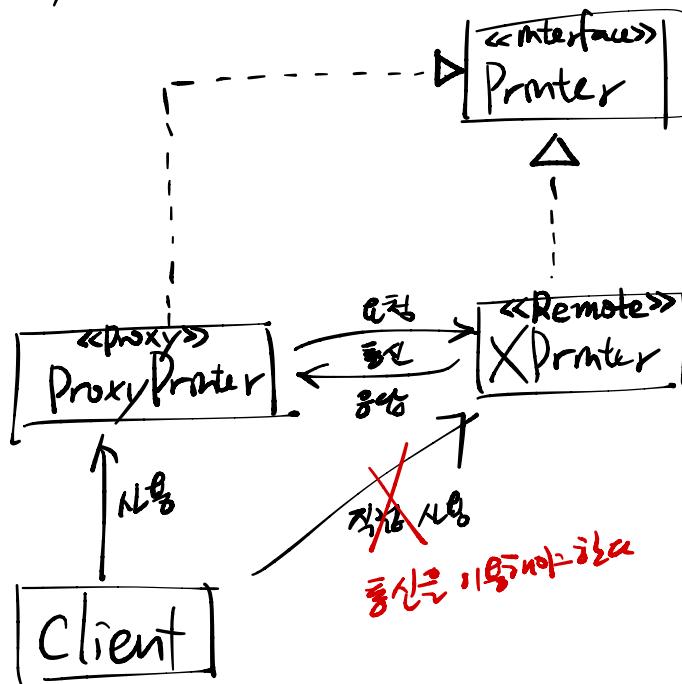


* Adapter 패턴 vs Proxy 패턴

① Adapter

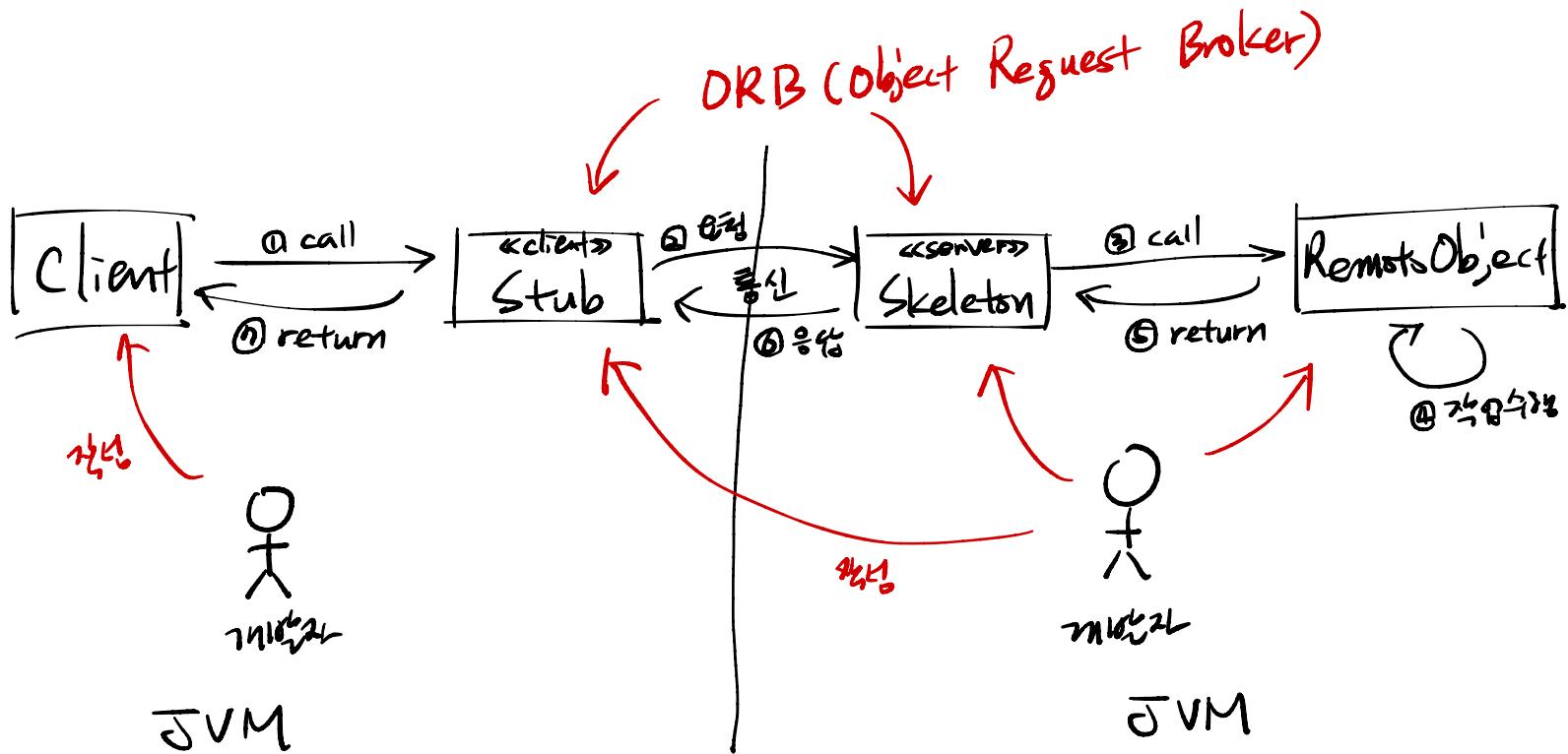


③ Proxy



* Remote Object 개념

↳ 다른 JVM에서 만든 객체
(프록시)

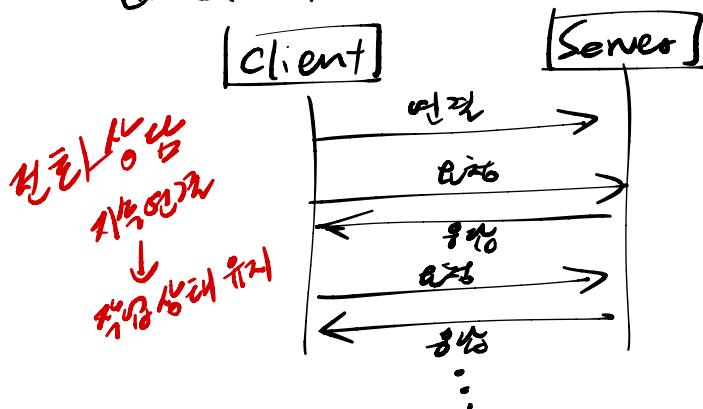


38. Stateful vs Stateless

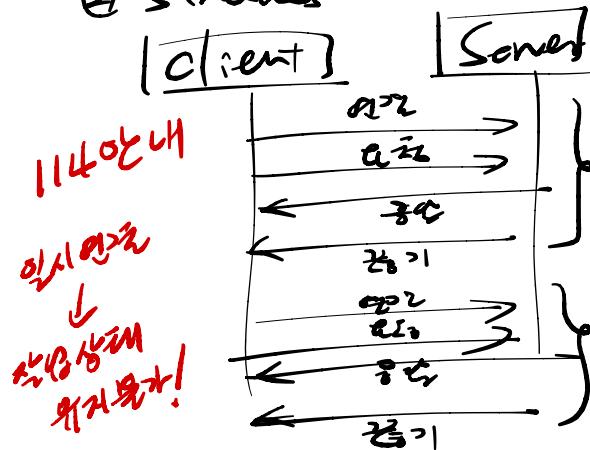
- * Stateful 웹서버 - 한번 연결한 후 여러번 요청/응답 처리
- * Stateless 웹서버 - 한번 연결한 후 한 번 요청/응답 처리



① Statefull

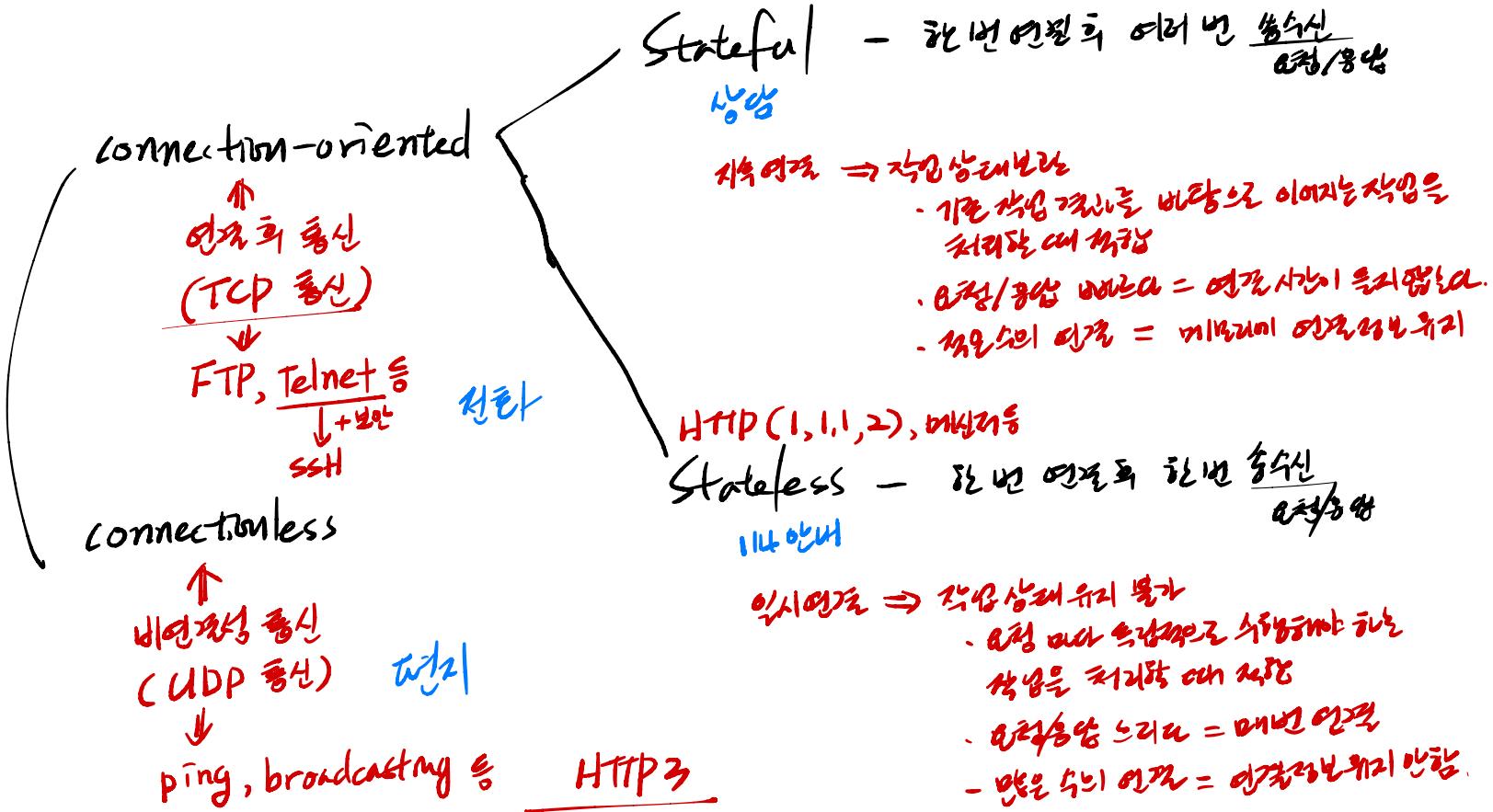


② Stateless

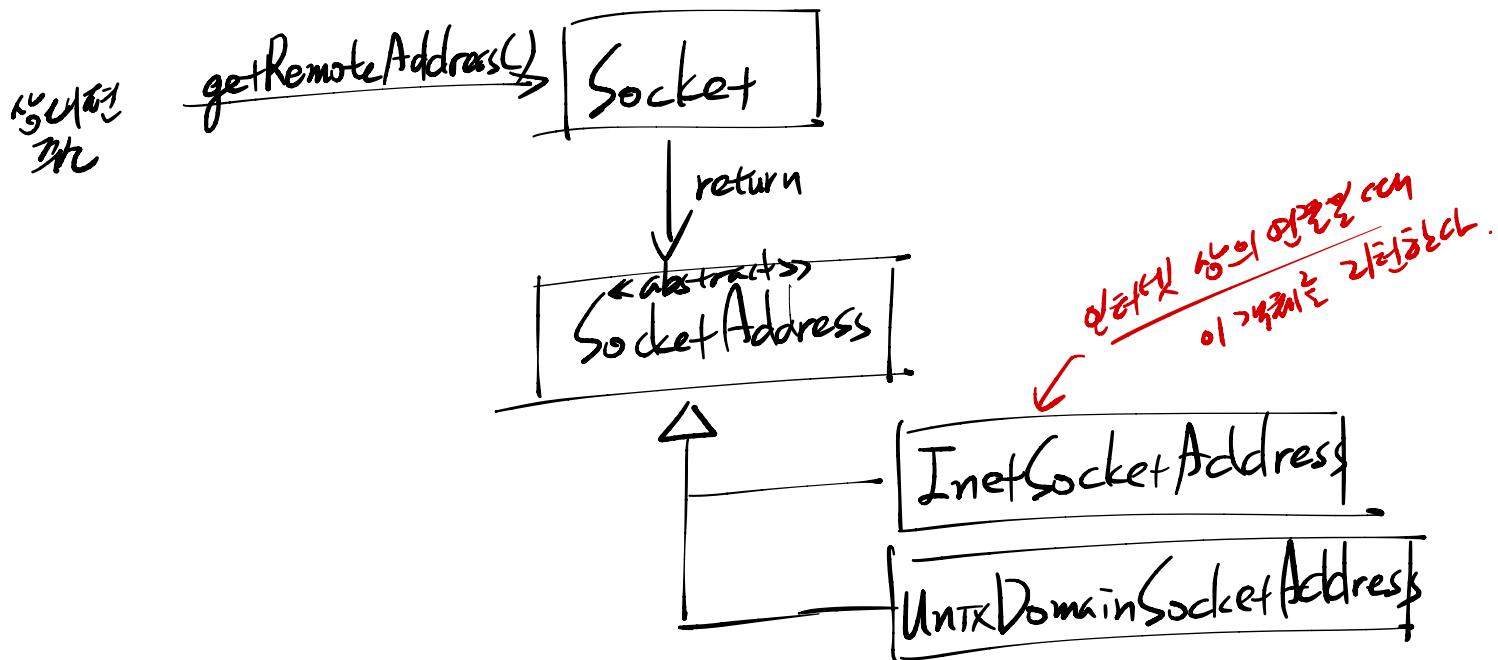


* 통신 18'시'

FTP, Telnet, SSH, 스트리밍, 채팅 등

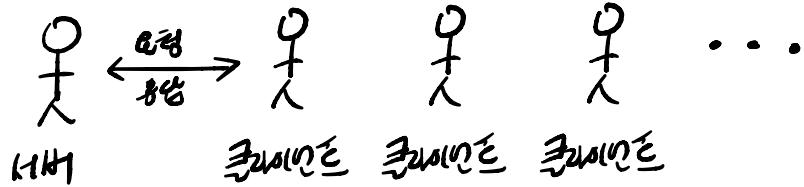


* Socket API



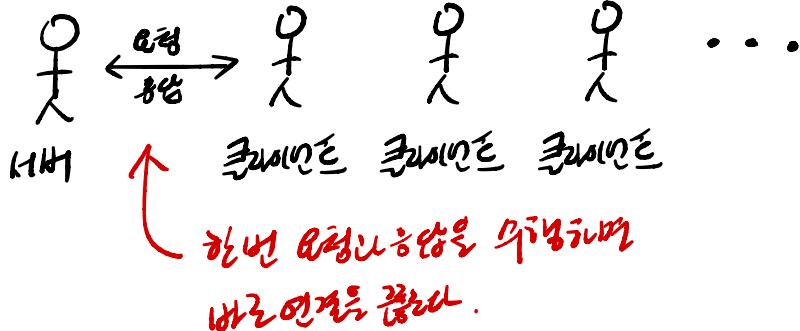
39. 미리그레드 층

① 아종 클러스터드 요청 처리 - Stateful



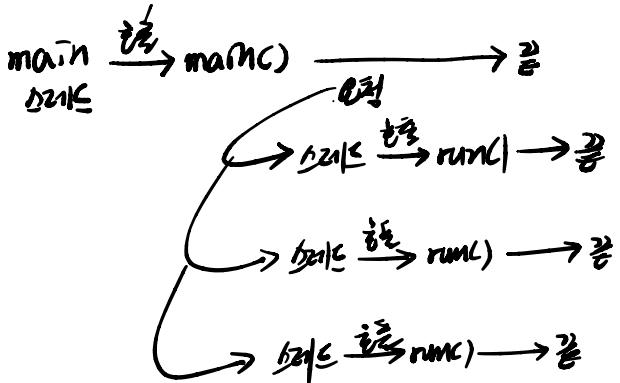
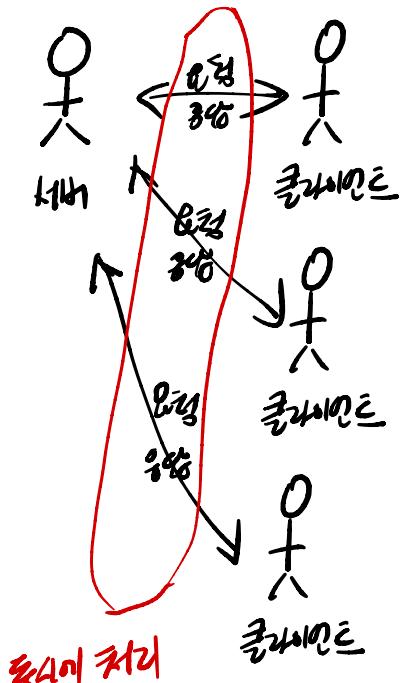
- ✓ 서비스와 연결된 클러스터가
여기서는 공동을 대기자
다른 클러스터는 기다리면 됨.
- ✓ 키/값, 블록체인 처리 된다.

② 아종 클러스터드 요청 처리 - Stateless



- ✓ 한 번 요청 → 한 번 응답/값
- ↓
Stateful 방식, 즉 초기 설정이 필요.
- ✓ 한번 요청한 후에는
다른 인증은 필요 없다.

③ 다중 쓰레드링크 예제 - Multi-threading \Leftarrow stateful & stateless 1회용의 고정적인 문제점 해결



① 쓰레드마다 세 가지 상태가 있다
그 시각화를 어떻게 해야 할까?

클라이언트 측정을 고정적으로 처리!

제거?

내장으로 처리!

클라이언트 측정 처리

main의 실행흐름
원리단계 이후 부정!

* 스레드를 만드는 예제 - main 스레드에서 다른 스레드를 만들 때의 예제

class 씨닝 extends Thread {

void run() {

여기 내용

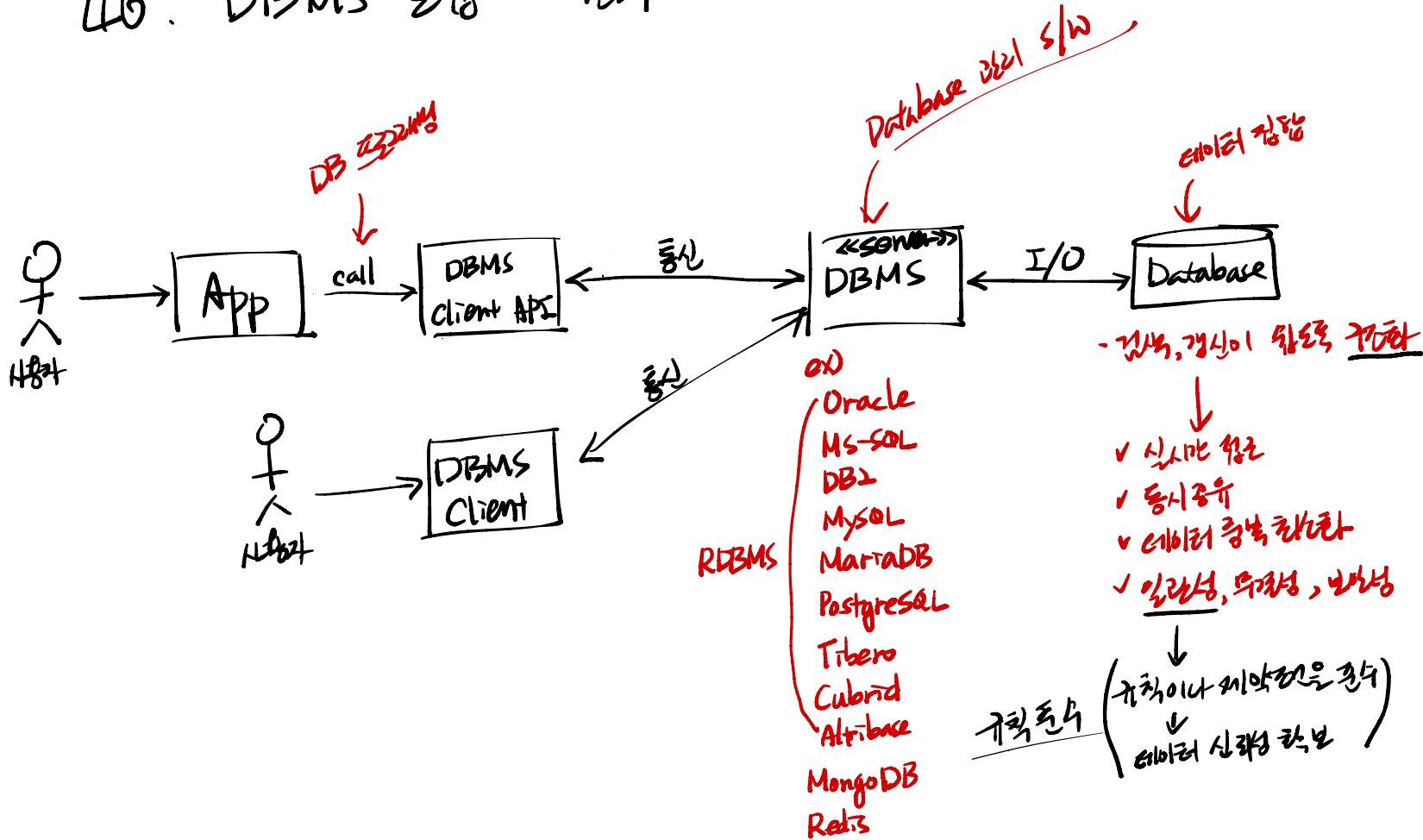
}

}

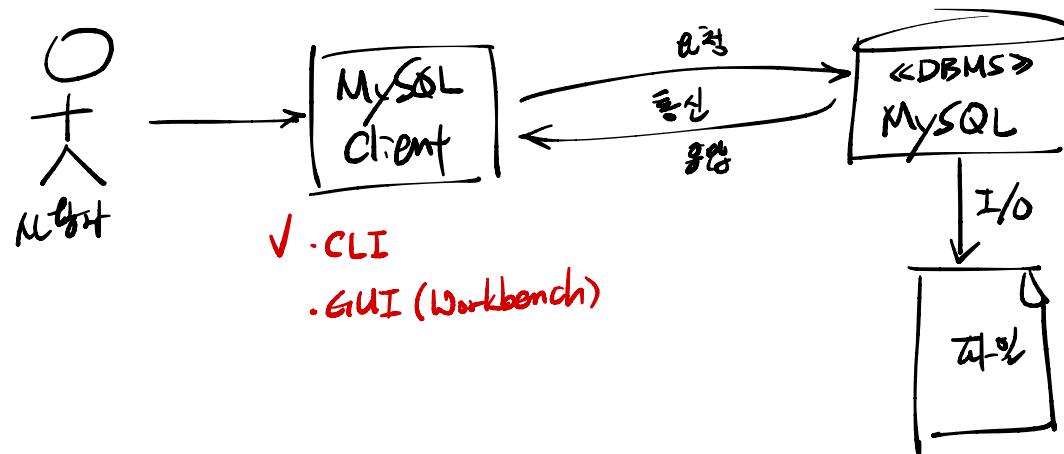
쓰레드.start();

여기서 쓰레드가 시작된다.

40. DBMS 속도 - 톤



40. DBMS 속성 - MySQL 설치



40. DBMS ဆို - DBMS အသေ

① 320J

≠ mysql -u root -P
DBMS 클라이언트 user id 암호입력

\$ mysql -h 127.0.0.1 -u root -p

④ DBMS 사용자 관리

③ 01012181011 484

create database globe11010101 default character set utf8 default collate utf8_general_ci

~~도시락을 사면서~~
~~이제는 막수~~
"드디어 왔어!"

40. DBMS ဆို - DBMS ဘာ

④ 2019년 11월 12일 18:35

(*) 이이이이이이이이스 끝을 주제

show databases

⑥ entwicklung ab k_2^{23}

use [ctrl+shift+b](#)

② 테이블 목록 조회

show tables

⑨ 2019년 8월 1주

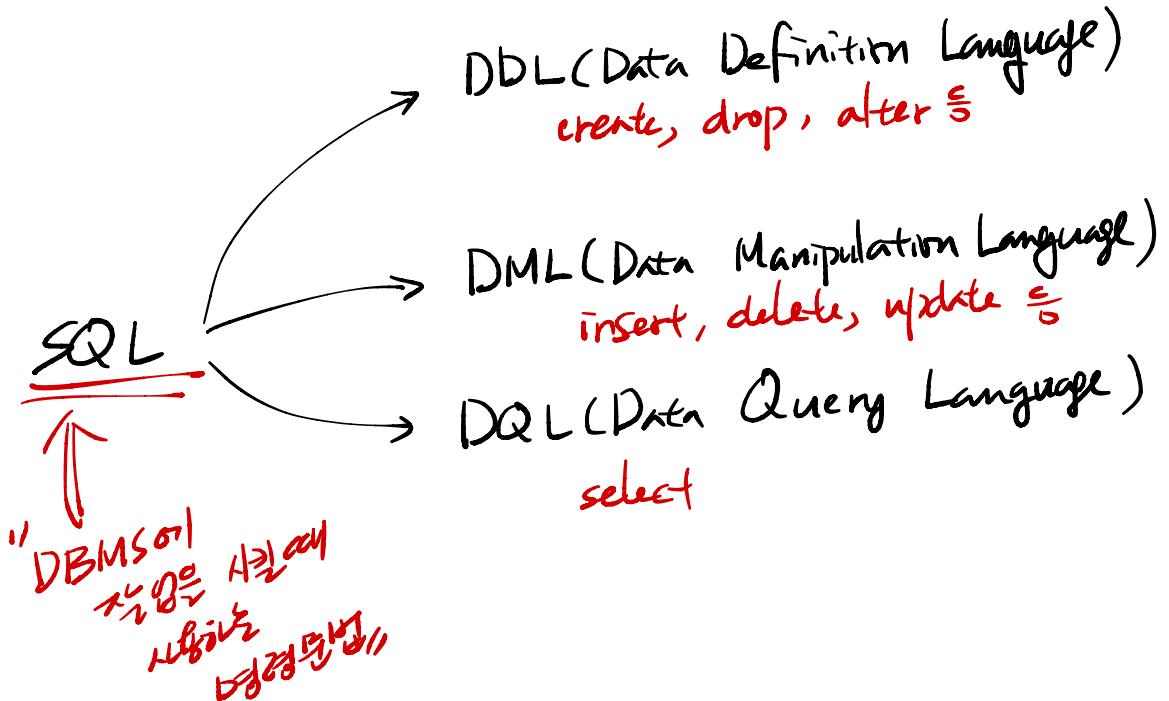
describe 描이브

desc ဆုတေသန

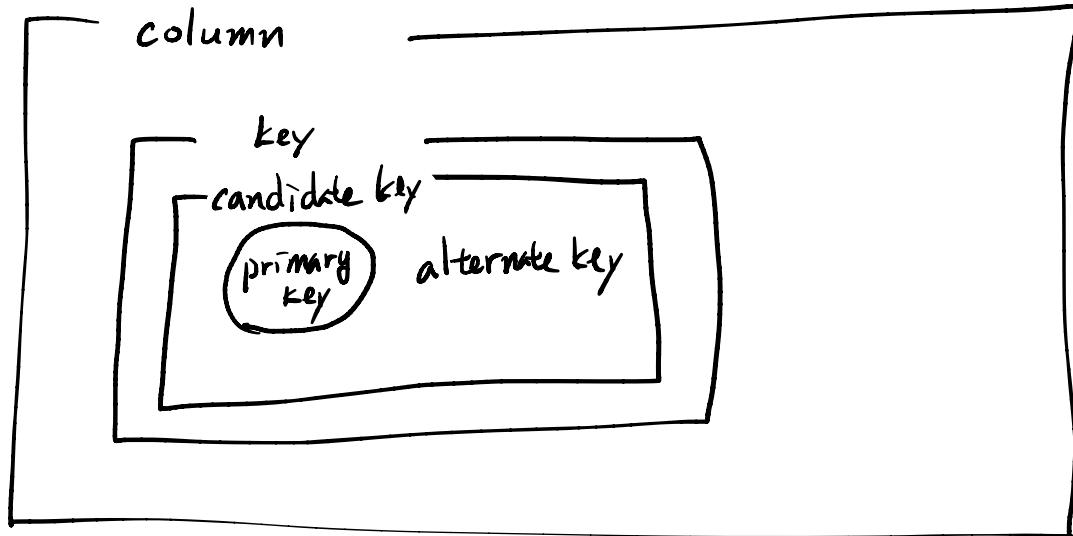
⑨ 亂世の政治

select user, host from mysql.user

110 . DBMS ୱେଳେ - SQL (Structured Query Language)

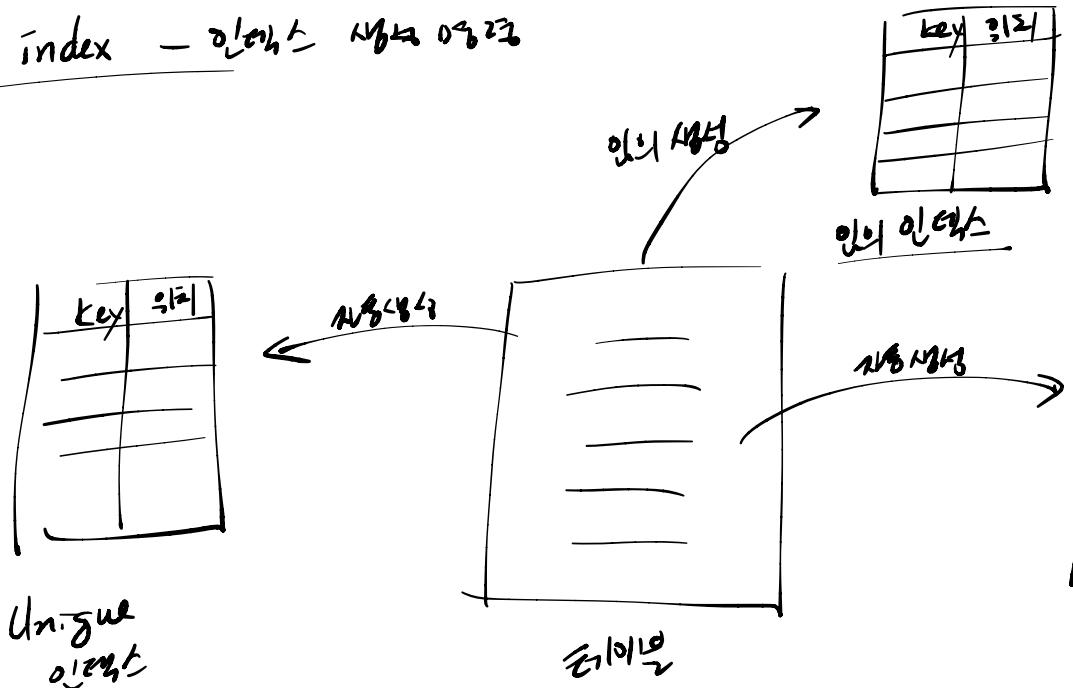


* key - 데이터를 구분할 때 사용하는 키임



(인공키)
artificial key
" "
surrogate key
(서로이기)

* index - depth $\approx 10^3$



Unique
object

A hand-drawn diagram consisting of a vertical rectangle with five horizontal lines inside it. The top line is at the top of the rectangle, and the bottom line is at the bottom. There are four shorter horizontal lines inside the rectangle, spaced evenly apart. This represents a stack of five pages.

key 값	
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10
1	11
1	12
1	13
1	14
1	15
1	16
1	17
1	18
1	19
1	20
1	21
1	22
1	23
1	24
1	25
1	26
1	27
1	28
1	29
1	30
1	31
1	32
1	33
1	34
1	35
1	36
1	37
1	38
1	39
1	40
1	41
1	42
1	43
1	44
1	45
1	46
1	47
1	48
1	49
1	50
1	51
1	52
1	53
1	54
1	55
1	56
1	57
1	58
1	59
1	60
1	61
1	62
1	63
1	64
1	65
1	66
1	67
1	68
1	69
1	70
1	71
1	72
1	73
1	74
1	75
1	76
1	77
1	78
1	79
1	80
1	81
1	82
1	83
1	84
1	85
1	86
1	87
1	88
1	89
1	90
1	91
1	92
1	93
1	94
1	95
1	96
1	97
1	98
1	99
1	100
1	101
1	102
1	103
1	104
1	105
1	106
1	107
1	108
1	109
1	110
1	111
1	112
1	113
1	114
1	115
1	116
1	117
1	118
1	119
1	120
1	121
1	122
1	123
1	124
1	125
1	126
1	127
1	128
1	129
1	130
1	131
1	132
1	133
1	134
1	135
1	136
1	137
1	138
1	139
1	140
1	141
1	142
1	143
1	144
1	145
1	146
1	147
1	148
1	149
1	150
1	151
1	152
1	153
1	154
1	155
1	156
1	157
1	158
1	159
1	160
1	161
1	162
1	163
1	164
1	165
1	166
1	167
1	168
1	169
1	170
1	171
1	172
1	173
1	174
1	175
1	176
1	177
1	178
1	179
1	180
1	181
1	182
1	183
1	184
1	185
1	186
1	187
1	188
1	189
1	190
1	191
1	192
1	193
1	194
1	195
1	196
1	197
1	198
1	199
1	200
1	201
1	202
1	203
1	204
1	205
1	206
1	207
1	208
1	209
1	210
1	211
1	212
1	213
1	214
1	215
1	216
1	217
1	218
1	219
1	220
1	221
1	222
1	223
1	224
1	225
1	226
1	227
1	228
1	229
1	230
1	231
1	232
1	233
1	234
1	235
1	236
1	237
1	238
1	239
1	240
1	241
1	242
1	243
1	244
1	245
1	246
1	247
1	248
1	249
1	250
1	251
1	252
1	253
1	254
1	255
1	256
1	257
1	258
1	259
1	260
1	261
1	262
1	263
1	264
1	265
1	266
1	267
1	268
1	269
1	270
1	271
1	272
1	273
1	274
1	275
1	276
1	277
1	278
1	279
1	280
1	281
1	282
1	283
1	284
1	285
1	286
1	287
1	288
1	289
1	290
1	291
1	292
1	293
1	294
1	295
1	296
1	297
1	298
1	299
1	300
1	301
1	302
1	303
1	304
1	305
1	306
1	307
1	308
1	309
1	310
1	311
1	312
1	313
1	314
1	315
1	316
1	317
1	318
1	319
1	320
1	321
1	322
1	323
1	324
1	325
1	326
1	327
1	328
1	329
1	330
1	331
1	332
1	333
1	334
1	335
1	336
1	337
1	338
1	339
1	340
1	341
1	342
1	343
1	344
1	345
1	346
1	347
1	348
1	349
1	350
1	351
1	352
1	353
1	354
1	355
1	356
1	357
1	358
1	359
1	360
1	361
1	362
1	363
1	364
1	365
1	366
1	367
1	368
1	369
1	370
1	371
1	372
1	373
1	374
1	375
1	376
1	377
1	378
1	379
1	380
1	381
1	382
1	383
1	384
1	385
1	386
1	387
1	388
1	389
1	390
1	391
1	392
1	393
1	394
1	395
1	396
1	397
1	398
1	399
1	400
1	401
1	402
1	403
1	404
1	405
1	406
1	407
1	408
1	409
1	410
1	411
1	412
1	413
1	414
1	415
1	416
1	417
1	418
1	419
1	420
1	421
1	422
1	423
1	424
1	425
1	426
1	427
1	428
1	429
1	430
1	431
1	432
1	433
1	434
1	435
1	436
1	437
1	438
1	439
1	440
1	441
1	442
1	443
1	444
1	445
1	446
1	447
1	448
1	449
1	450
1	451
1	452
1	453
1	454
1	455
1	456
1	457
1	458
1	459
1	460
1	461
1	462
1	463
1	464
1	465
1	466
1	467
1	468
1	469
1	470
1	471
1	472
1	473
1	474
1	475
1	476
1	477
1	478
1	479
1	480
1	481
1	482
1	483
1	484
1	485
1	486
1	487
1	488
1	489
1	490
1	491
1	492
1	493
1	494
1	495
1	496
1	497
1	498
1	499
1	500

<u>key</u>	위치

primary key
الكلمة الرئيسية

- ✓ 철전이 저장한 운송에는 일자리가 (철전) 생길 것이다
 - ✓ 중소기업 저장의 문제에 대해
기술력을 갖는 일부 기업들이 있다. } } → 혼란드 같은

primary key
a) PK
b) ElasticSearch

* projection vs selection

선택
제거
select.m

no	name	class	working	tel
			F	
			N	
			N	
			F	
			N	

"projection" - 테이블에서 원하는
필드만 선택

select no, name, class
from test1;

select no, name, class
from test1
where working='F';

"selection" - 테이블에서 원하는
행만 선택

* view et table V_2^{ex}

