

# Java I/O Stream API

byte stream

character stream

O ↗  
↓ ↘

InputStream

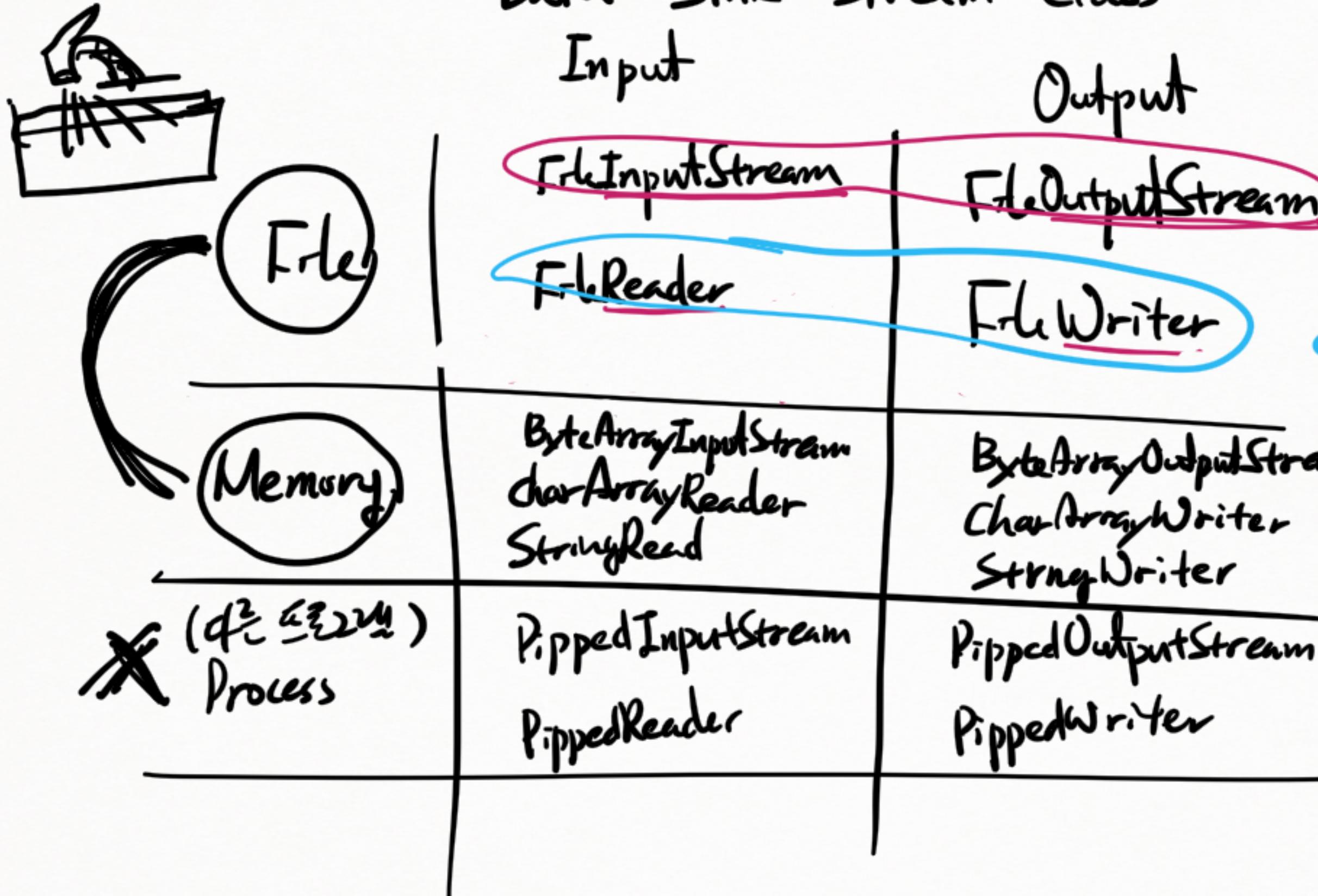
Reader

↗ I ↗  
↓ ↘

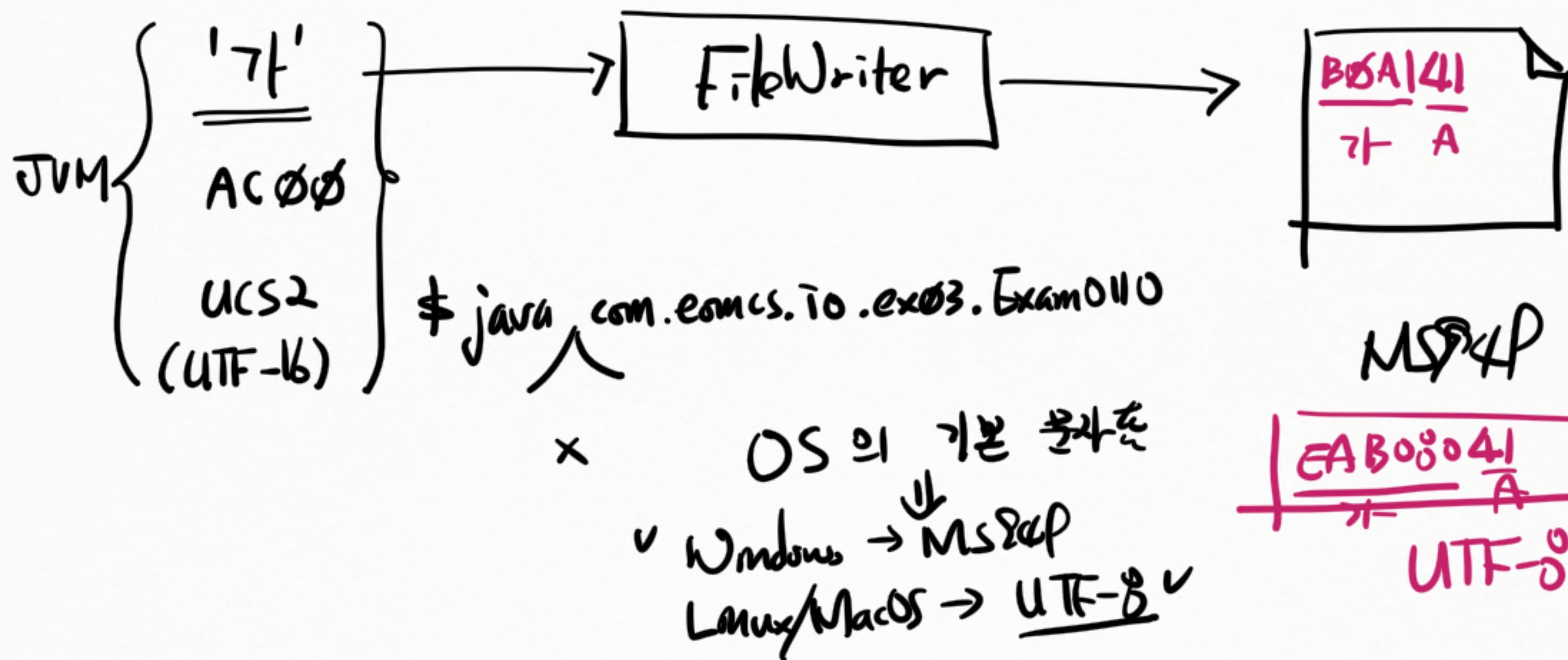
OutputStream

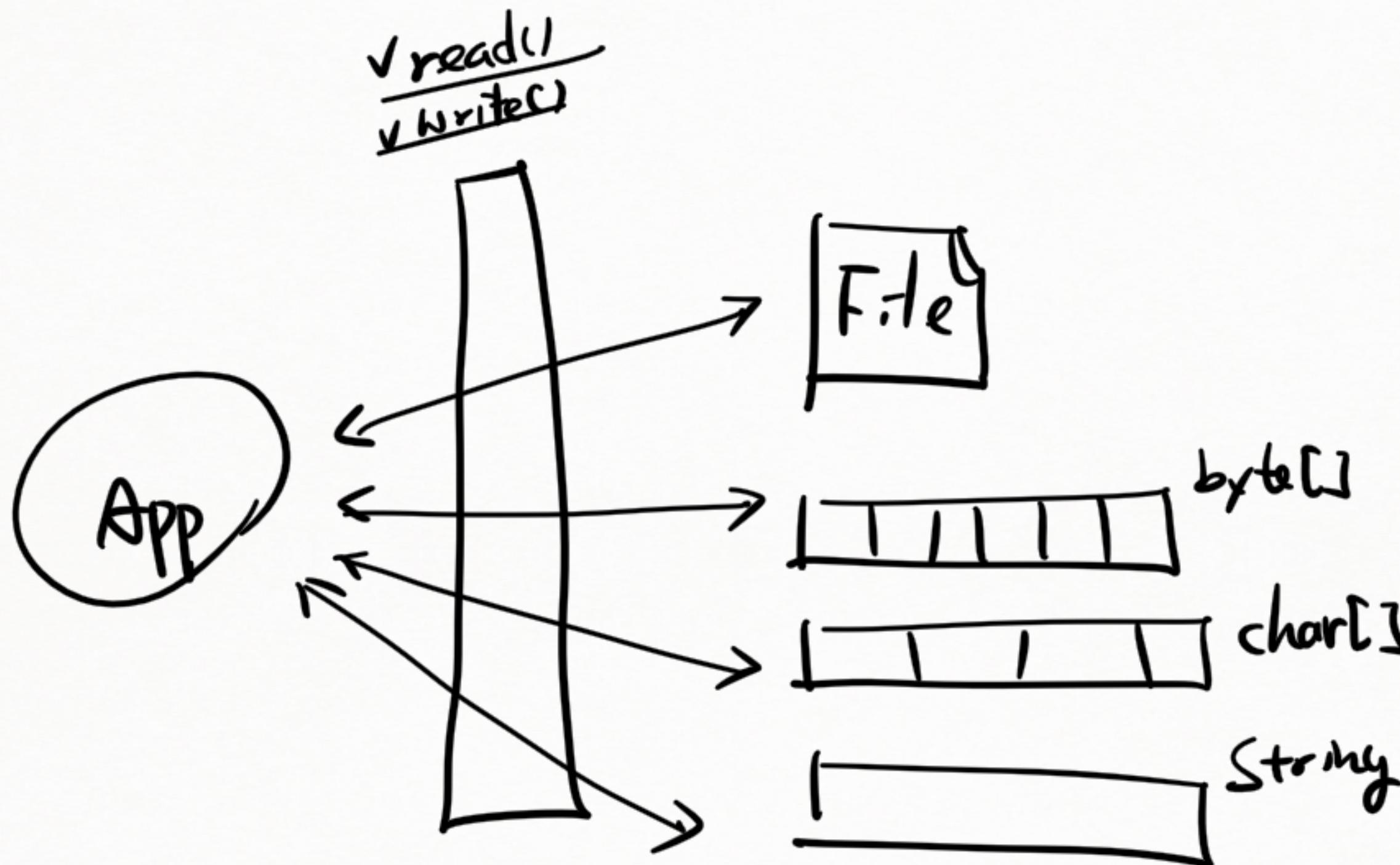
Writer

# Data Sink Stream Class



ex) .grf - jpg - ppt  
.avi - pdf  
ex) .csv .html .java  
.js .css .txt





MS949 : 41 42 B0 A1 B0 A2

UTF-8 : 41 42 EA B0 80 EA B0 81

UTF-16BE : 0041 0042 AC00 AC01

UTF-16LE : 4100 4200 AC00 AC01

JVM  
↓

char : UCS2 (UTF-16BE)

0041    0042    AC00    AC01  
A              B              𠂇              𠂇

new String( byte[] , offset , length , charset )

File/Network

✗ MS949  
✗ EUC-KR  
**UTF-8**



DBMS  
↓  
✗

JVM



UCS2

"

UTF-16BE

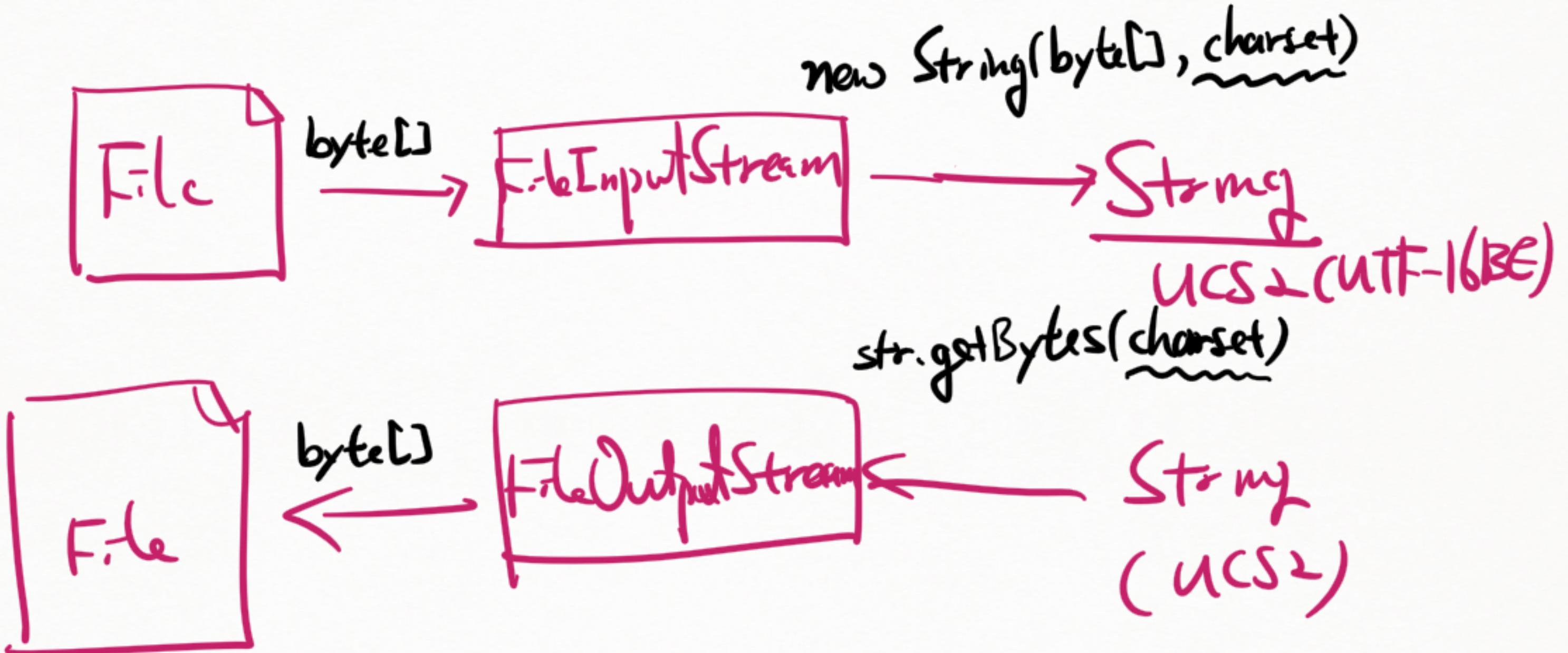
"

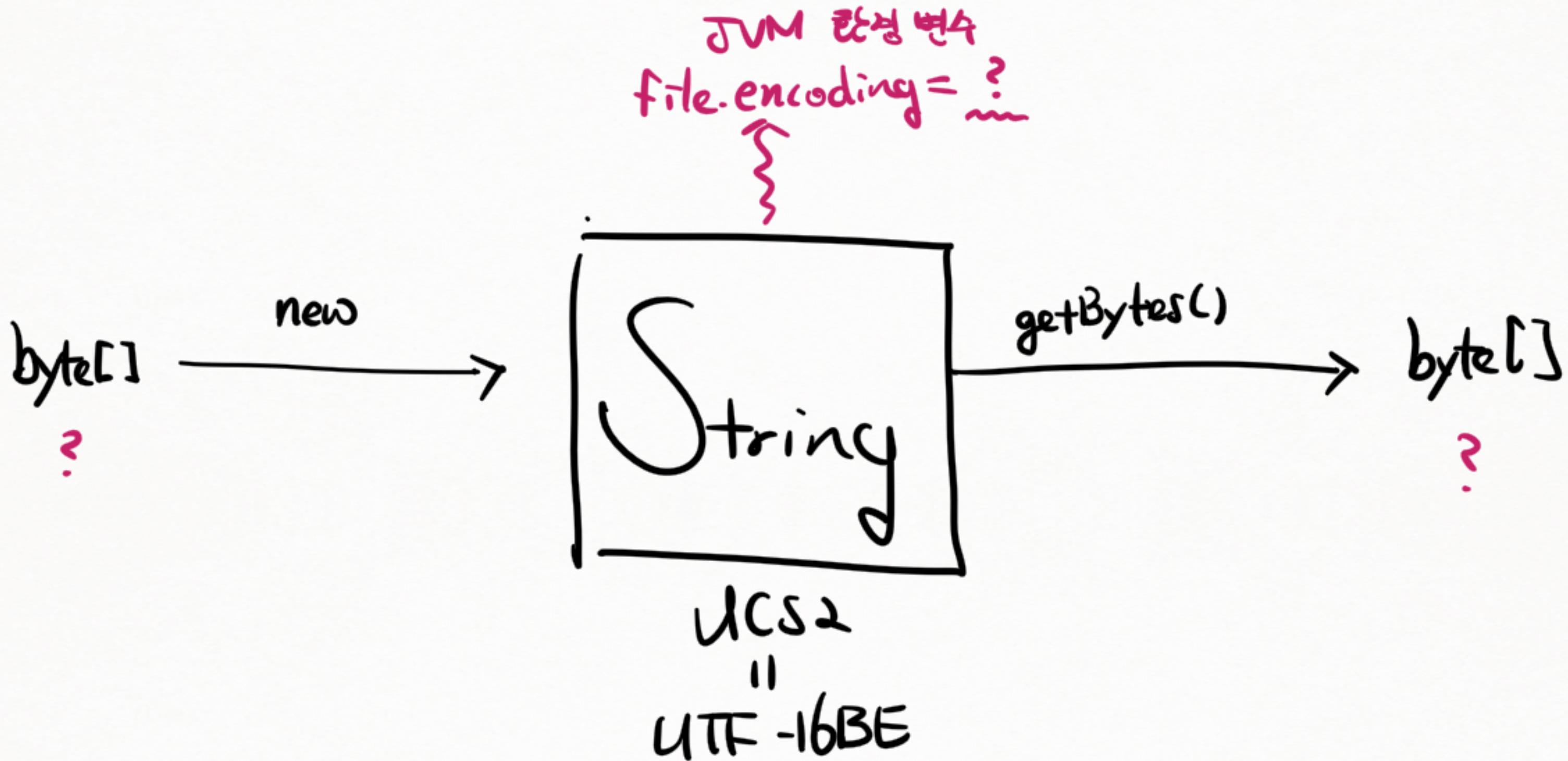
char c;  
~~~ 'A' → 0041  
'가' → Aced

File/Network

MSP430  
✗  
EUC-KR  
**UTF-8**







$$E \Rightarrow \underline{3017} - \underline{3030} \Rightarrow \text{값 } \boxed{?}$$

2×XX

중식

34XX - 30XX 헌식

$$\begin{array}{r} 411X - 30XX - 30XX & 190 \text{ 만원} \\ 9999 + 30XX - 12XX - 30XX & 13 \end{array}$$

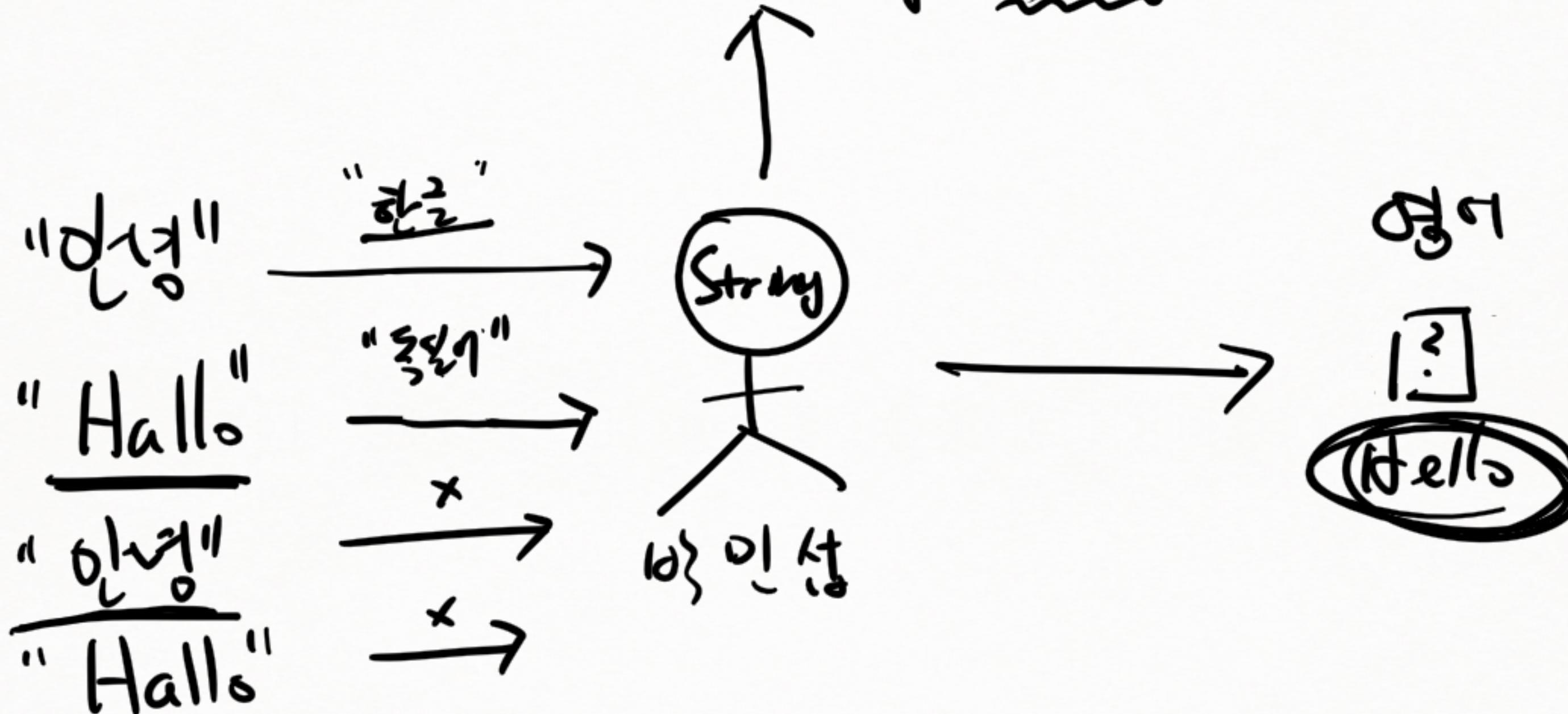
✓ A  $\Rightarrow \begin{array}{r} 2134 \\ 3418 - 3012 \\ \hline 12 \end{array}$

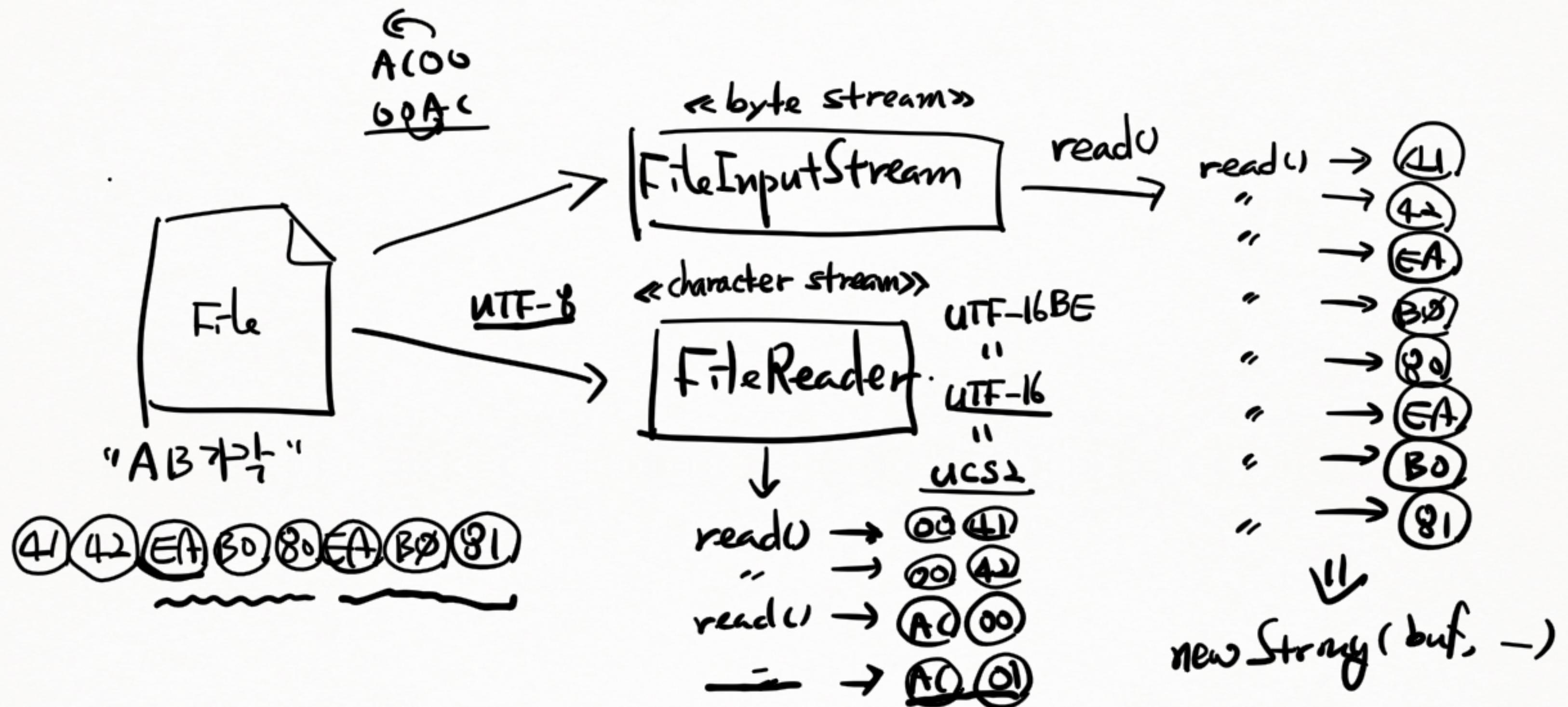
✓ B  $\Rightarrow \begin{array}{r} 3418 - 3011 \\ \hline 17 \end{array}$

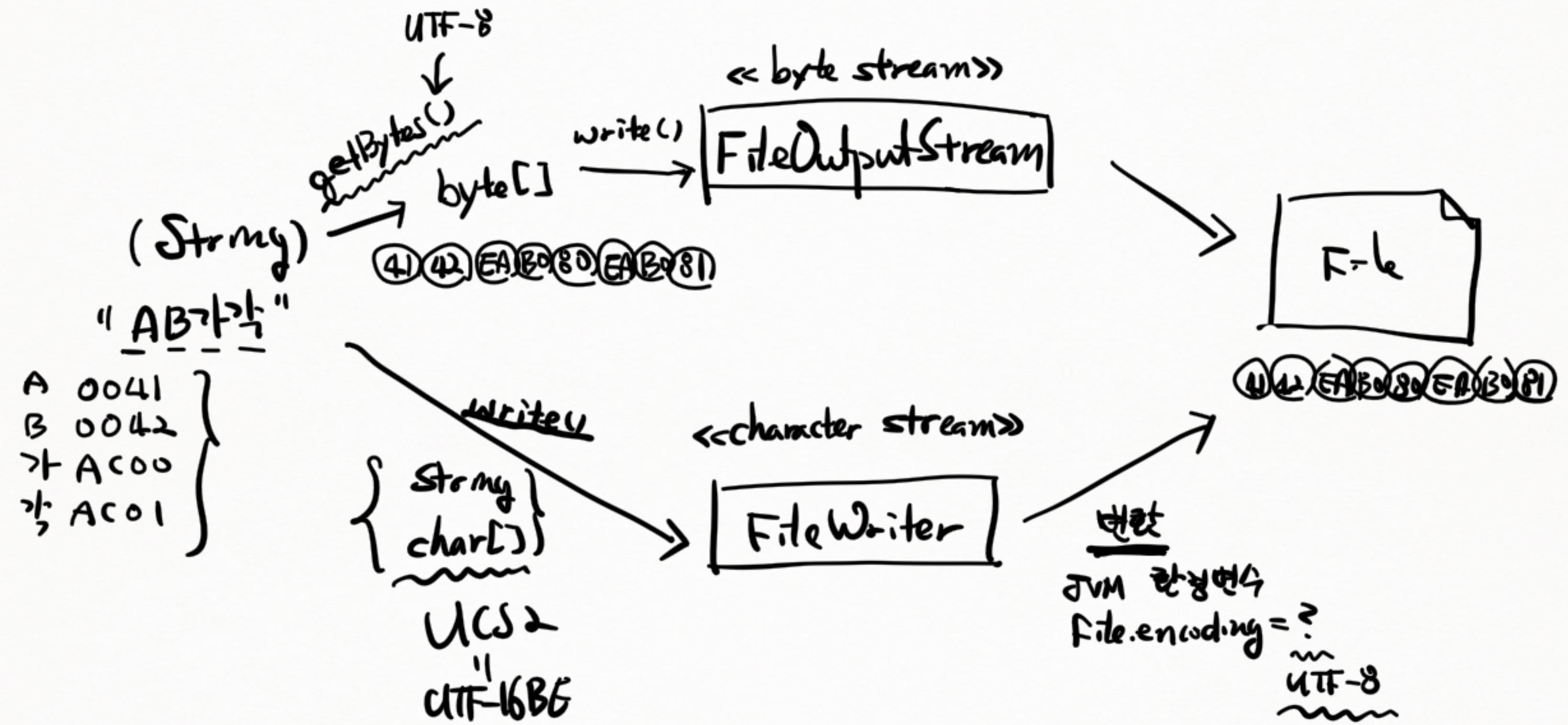
✓ C  $\Rightarrow \begin{array}{r} 4119 - 3017 \\ \hline 12 \end{array}$

D  $\Rightarrow \begin{array}{r} 4119 - \underline{\underline{3017}} \\ \hline 12 \end{array}$

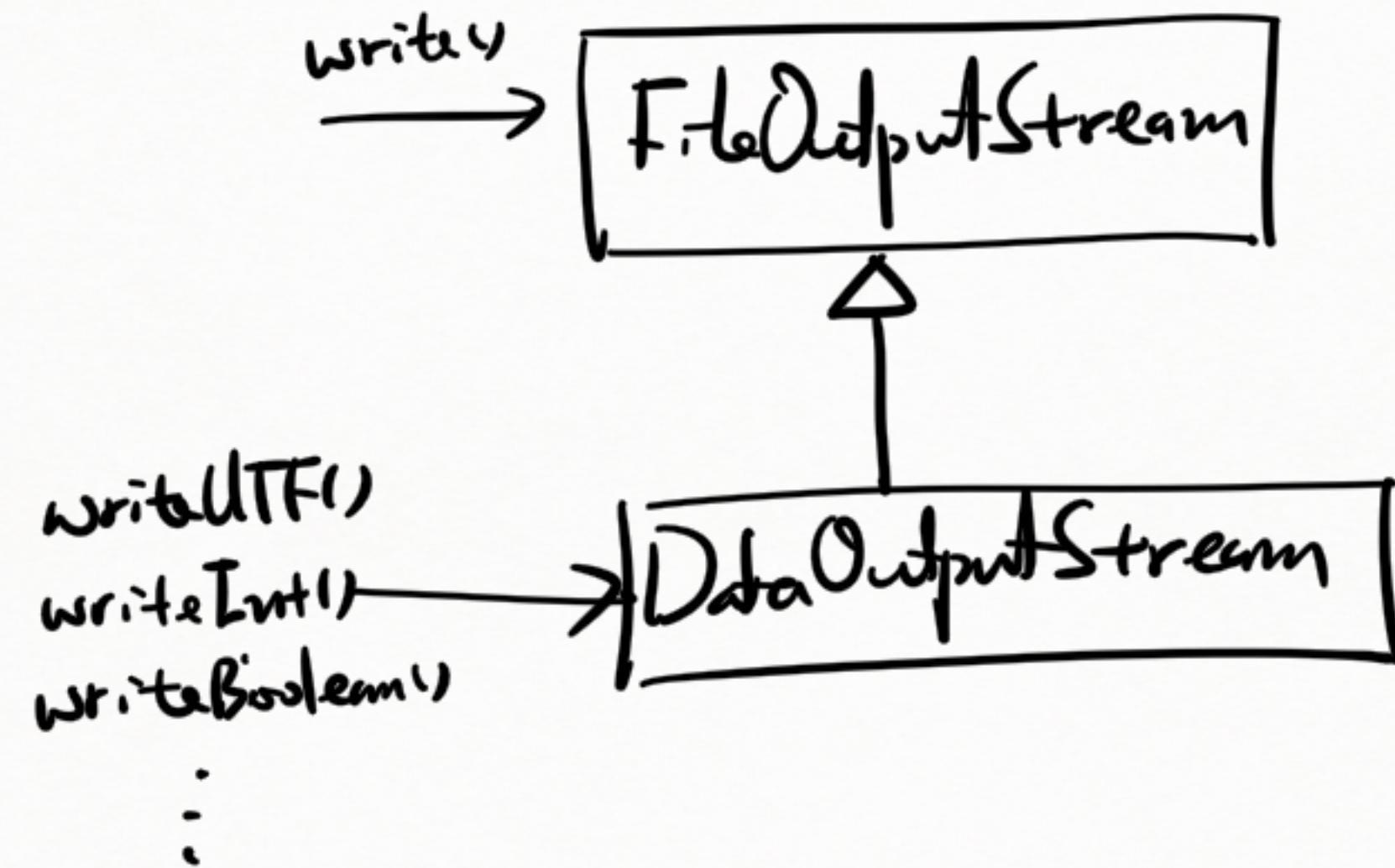
file.encoding = UTF-8

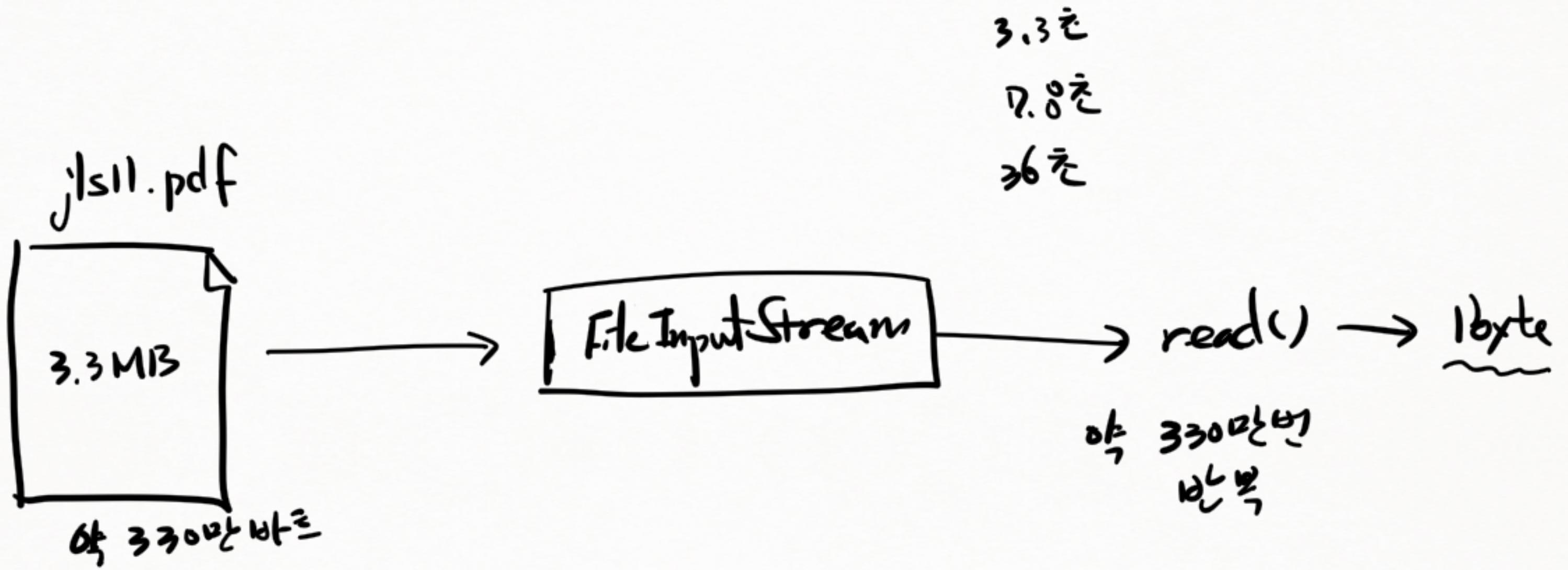


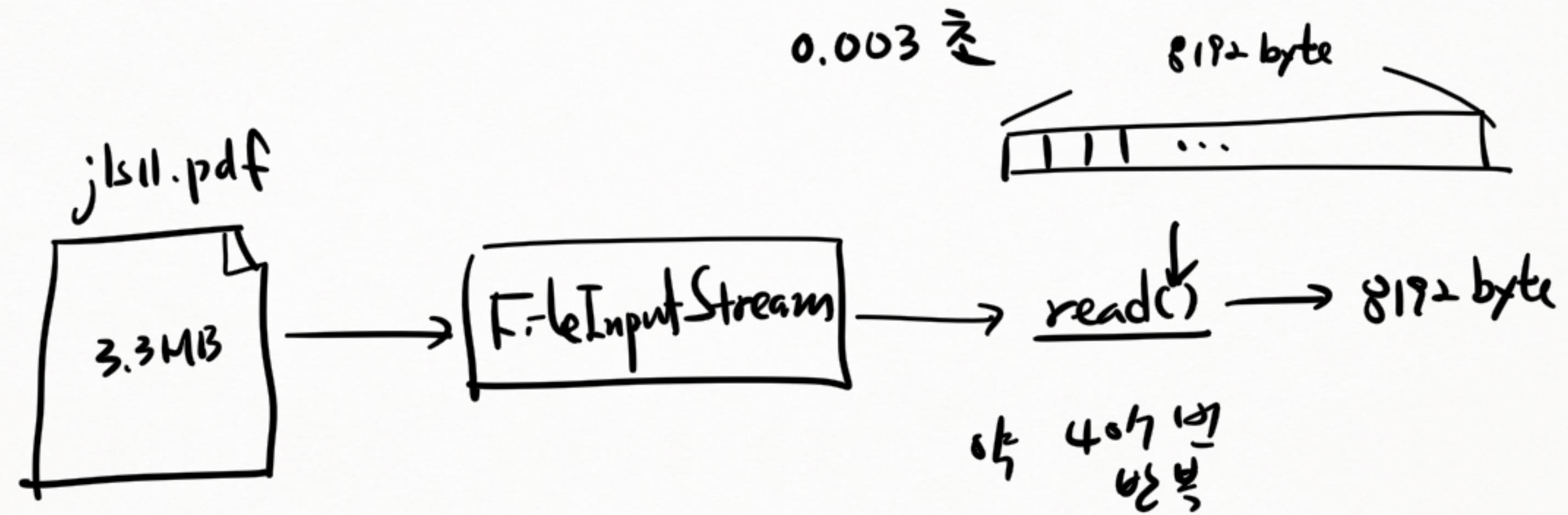


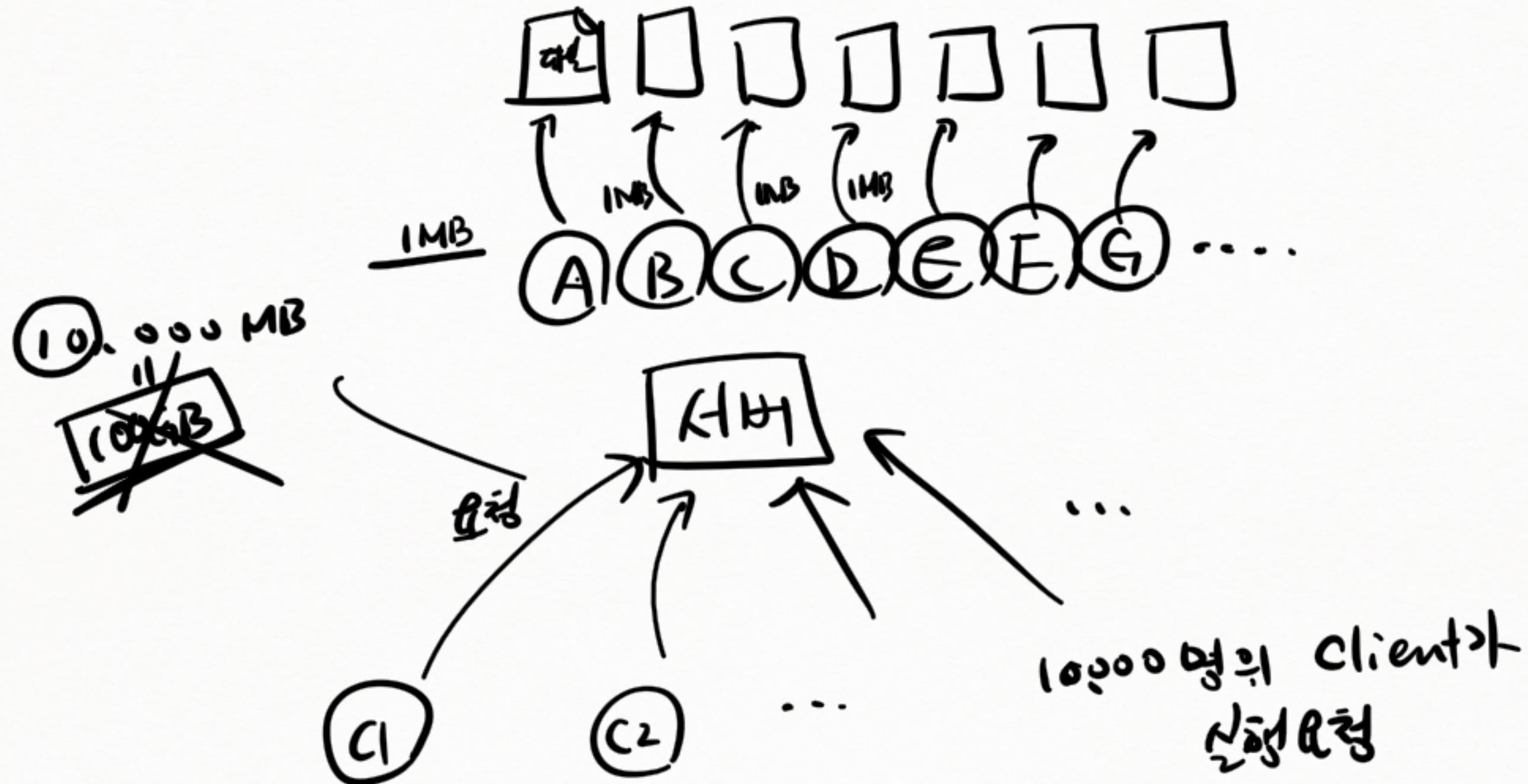


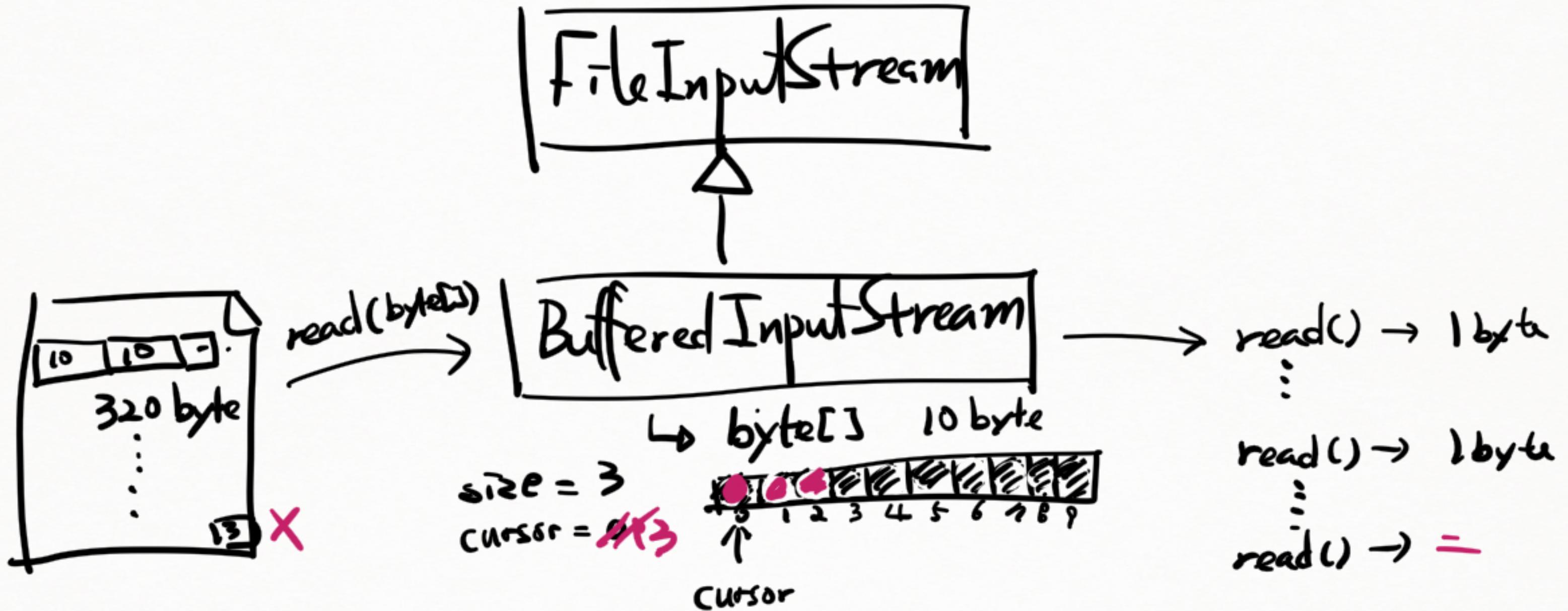


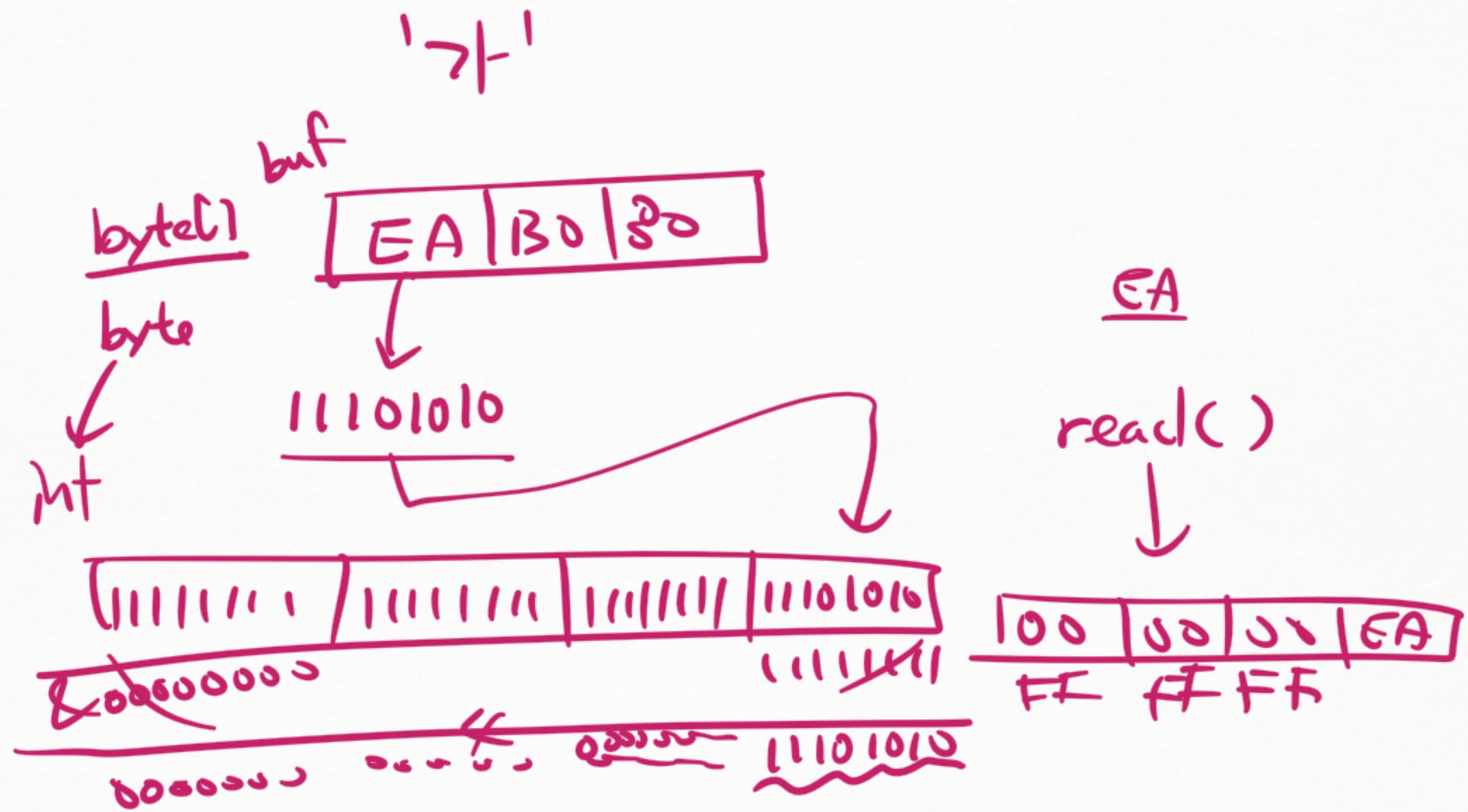


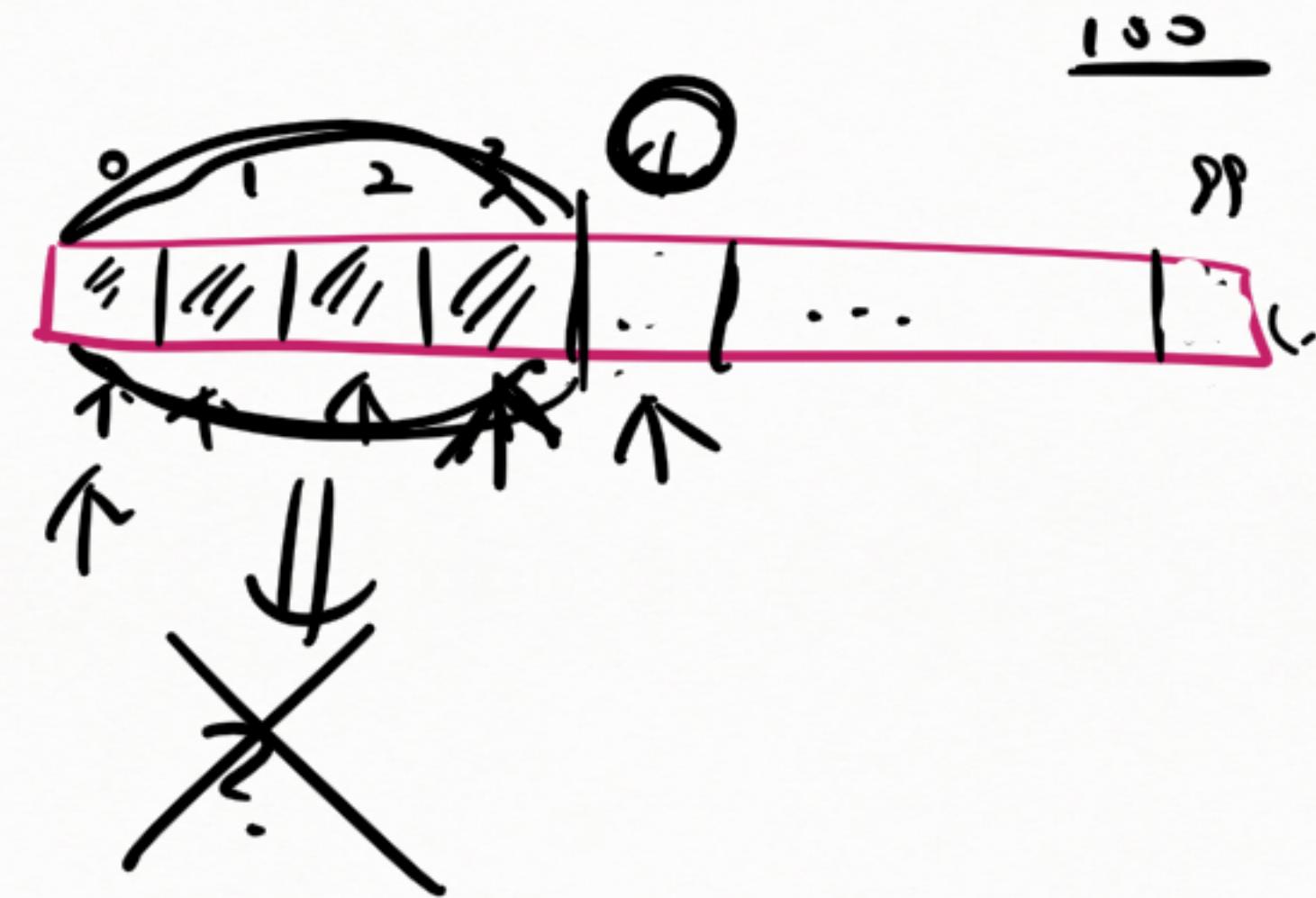


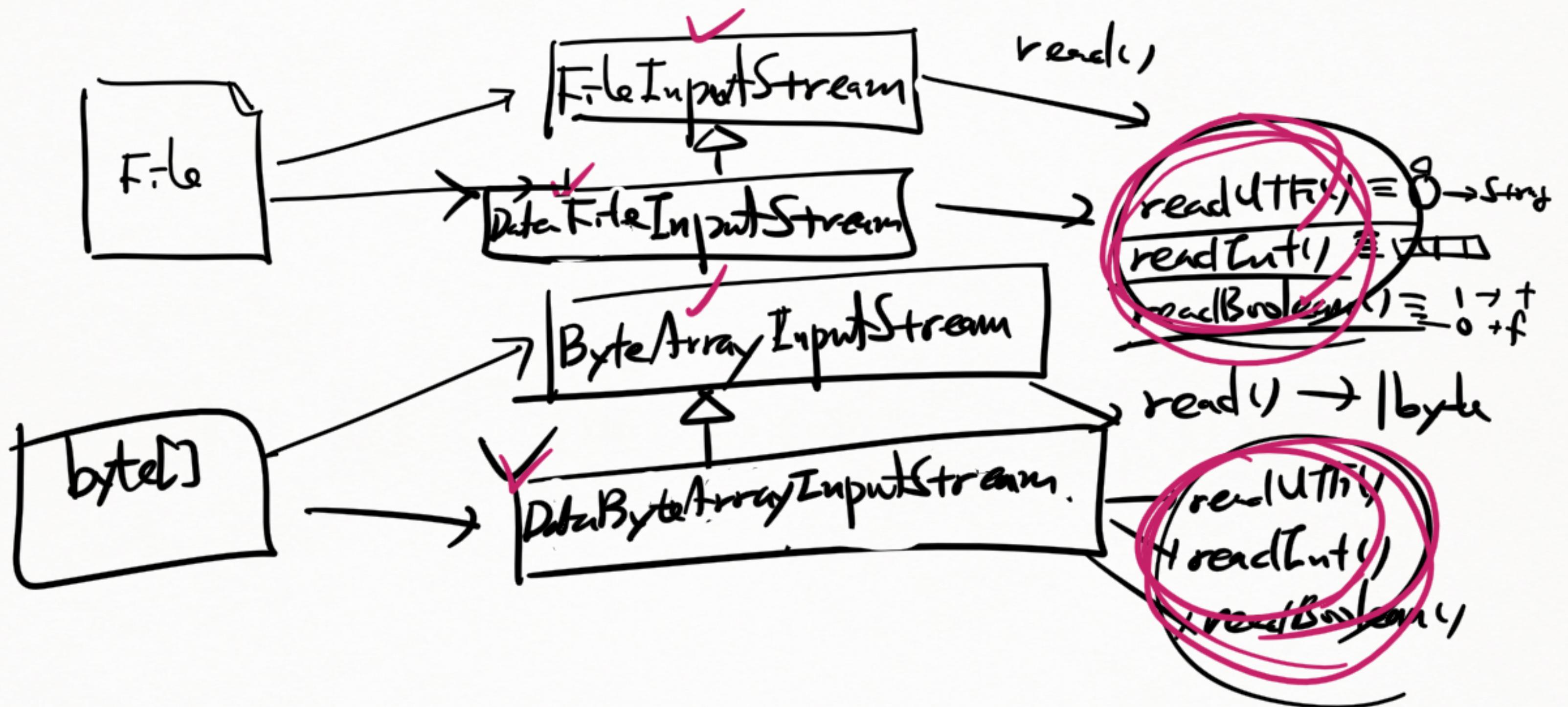


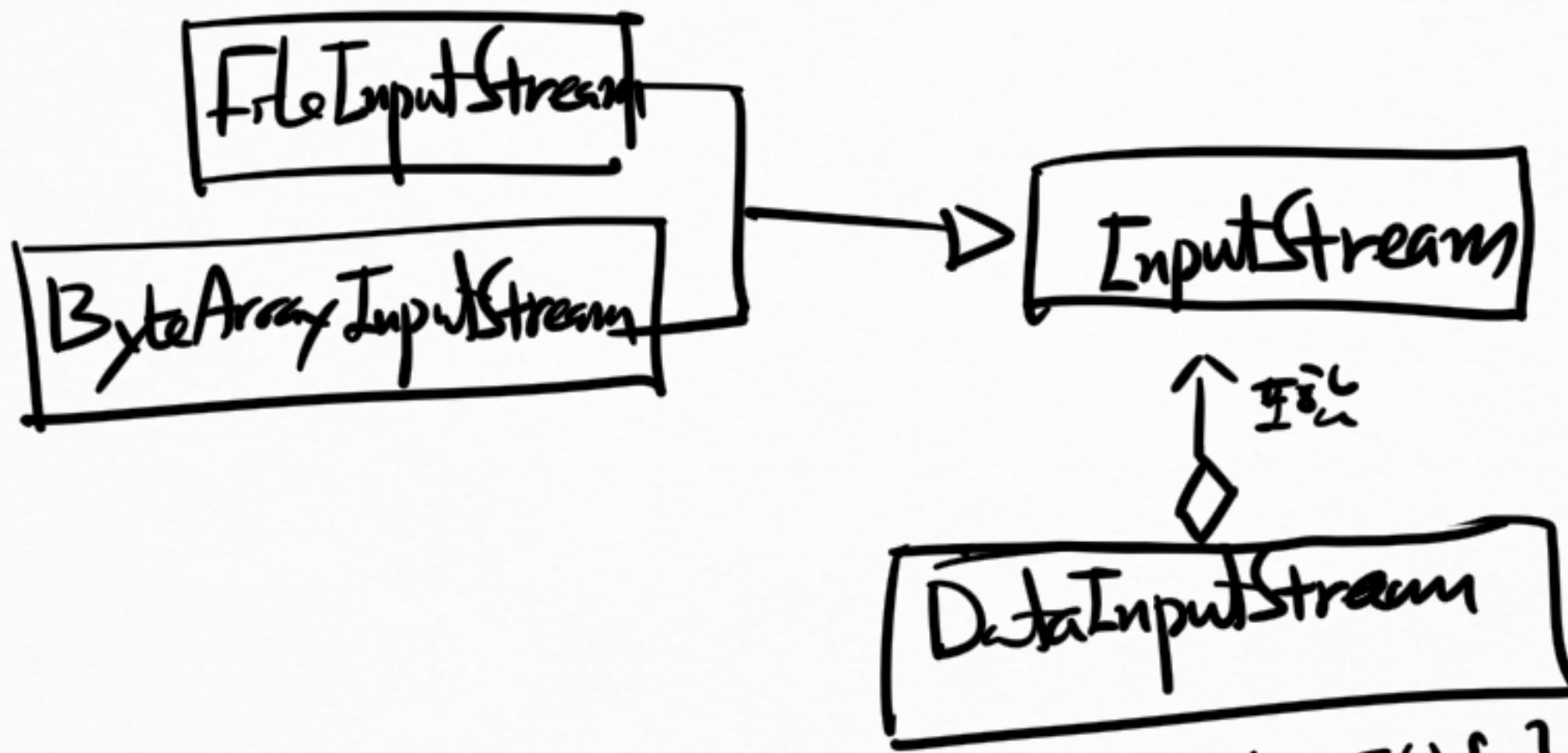






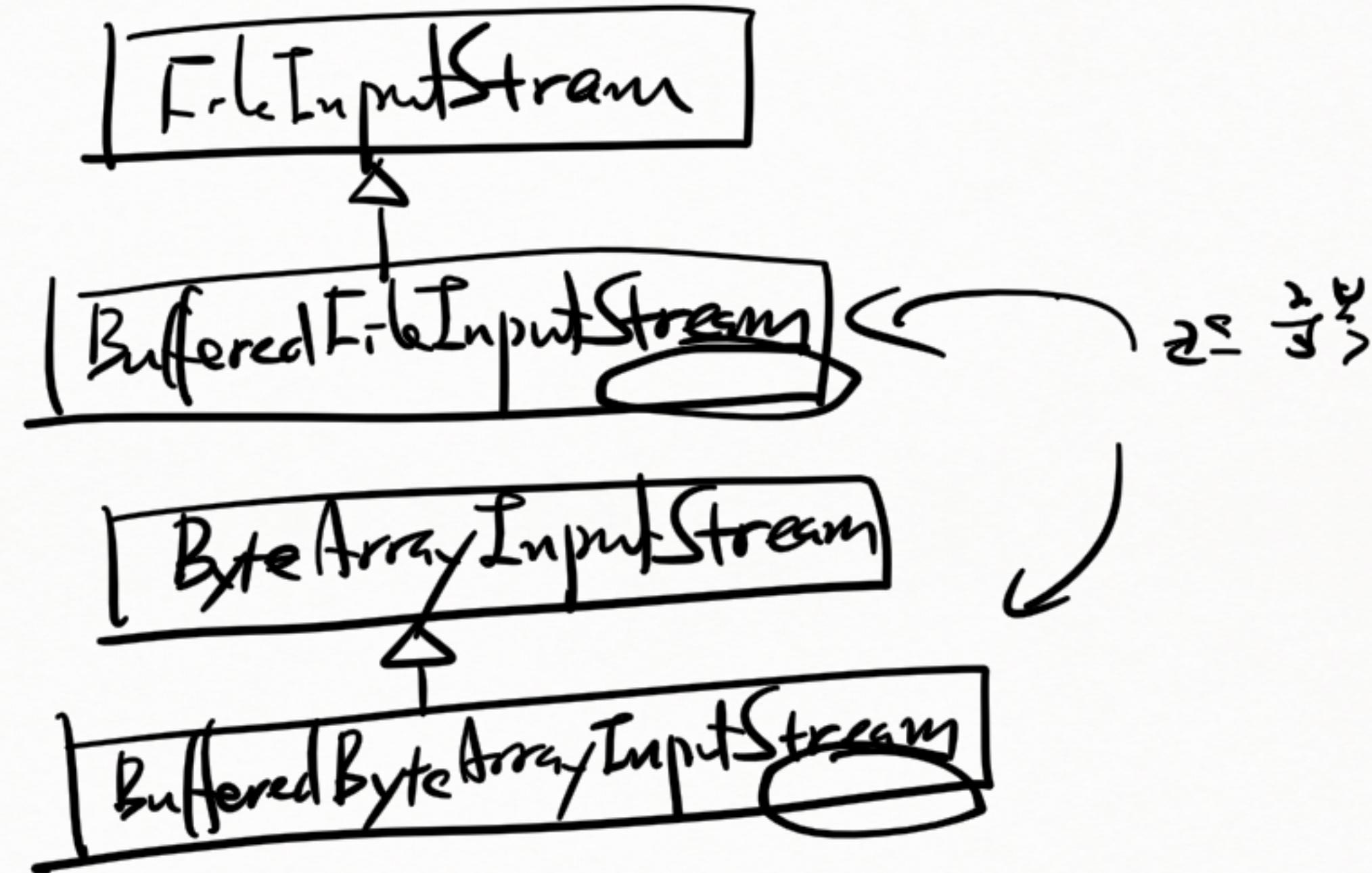


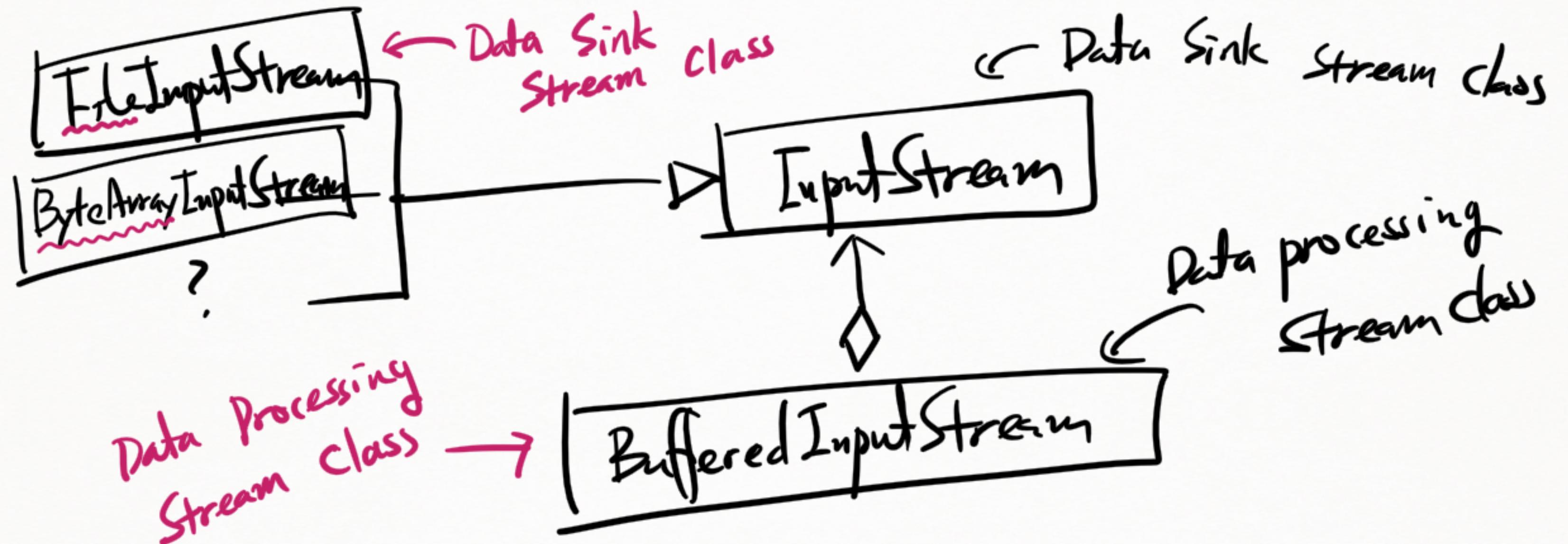




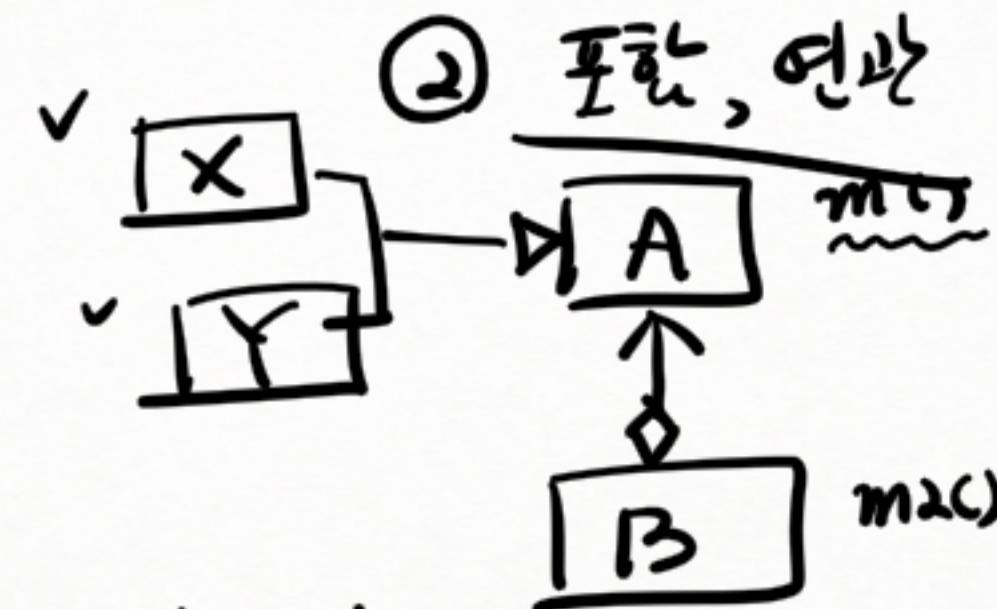
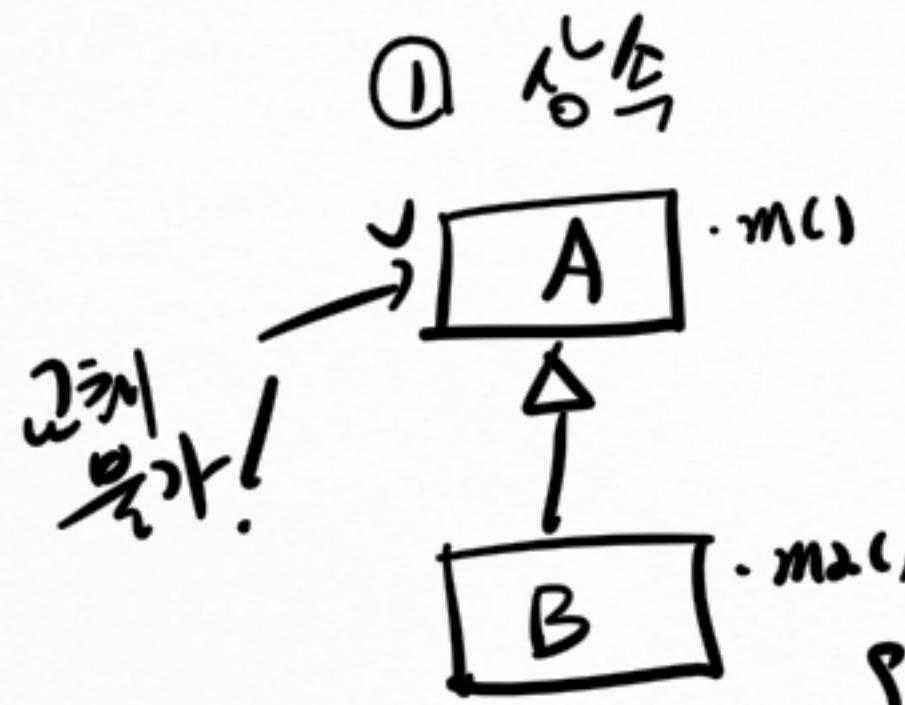
장식구 (Decorator)

- readUTF() {-}
- readInt() {-}
- readLong() {-}
- readBoolean() {-}

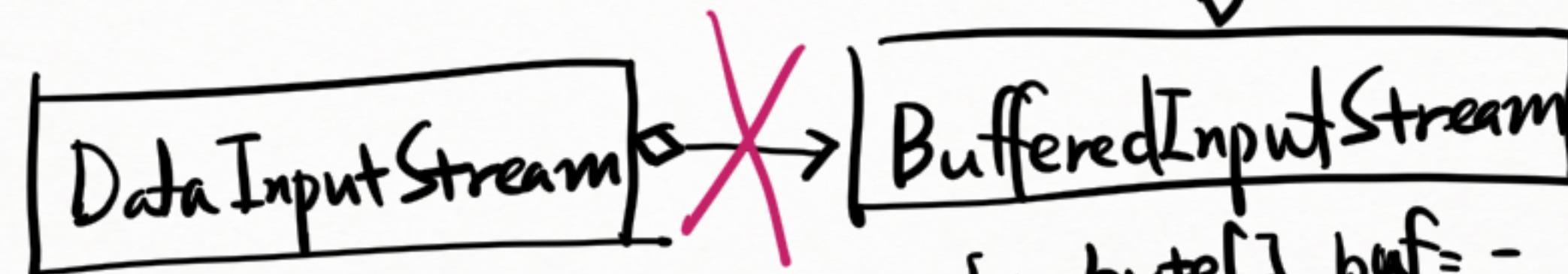
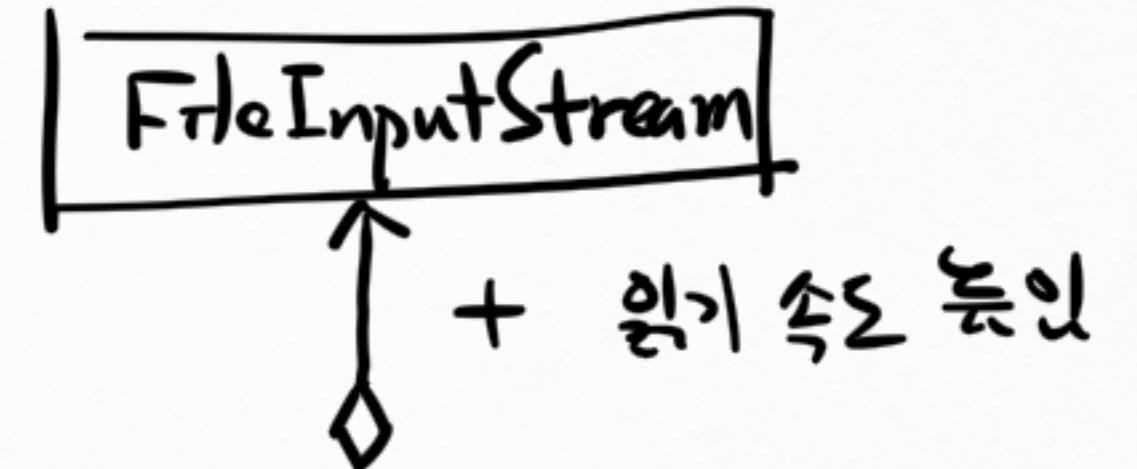




## 기능학장



- }    ✓ 중복 처리화  
      ✓ 재사용  
      ✓ ① 초기화-용이



(  
· 문자열  
· 자바 기본 타입  
· byte[]  
· int  
· long  
· boolean)  
· readUTF()  
· readInt()  
· readLong()  
· readBoolean()

{  
· byte[] buf = -  
· int size  
· int cursor

Data Sink  
Stream class

레고블록

생성자에 다른 InputStream을  
받지 않는데  
"

"완성품 블록"  
예) 인형  
집  
마카  
상

FileInputStream  
ByteArrayInputStream  
...

"Decorator" 패턴

Component

InputStream/OutputStream  
Reader/Writer

FilterInputStream

Decorator

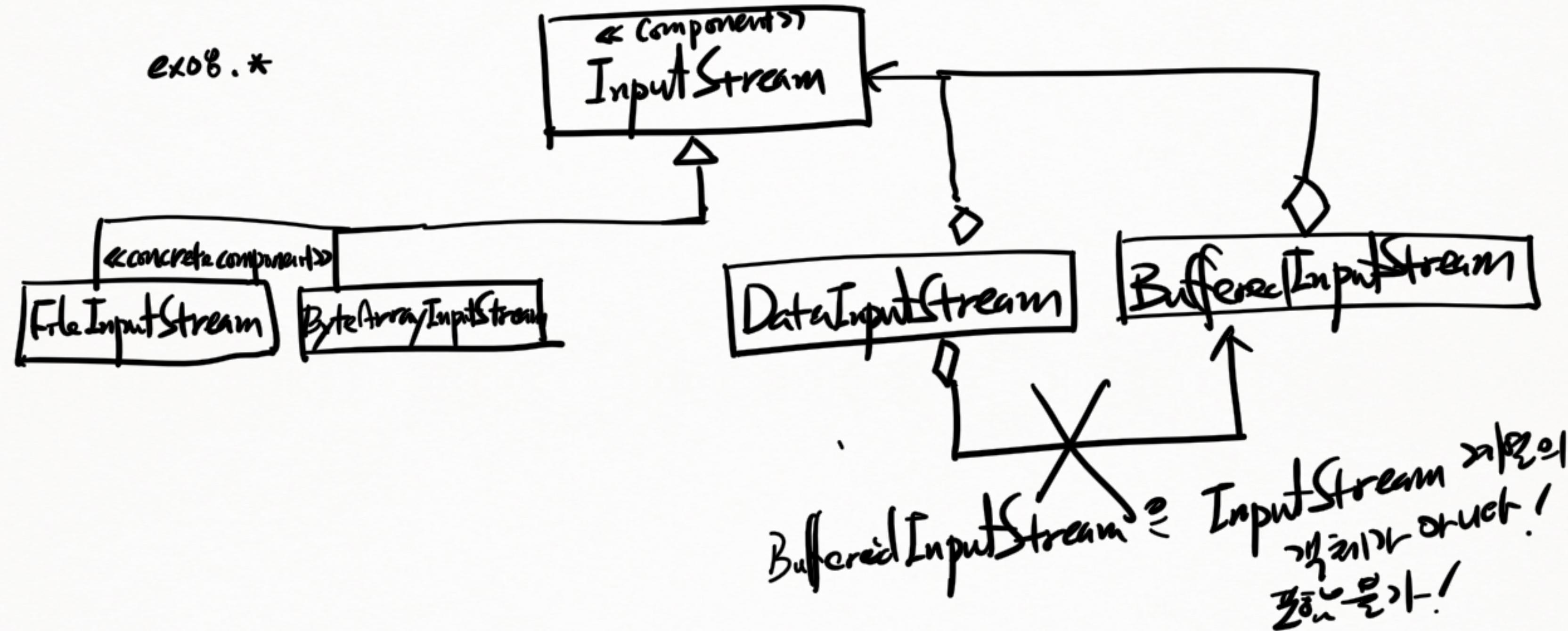
Data processing  
Stream class

중간블록 예) 짧은  
큰화  
나눌 블록

DataInputStream  
BufferedInputStream  
ObjectInputStream

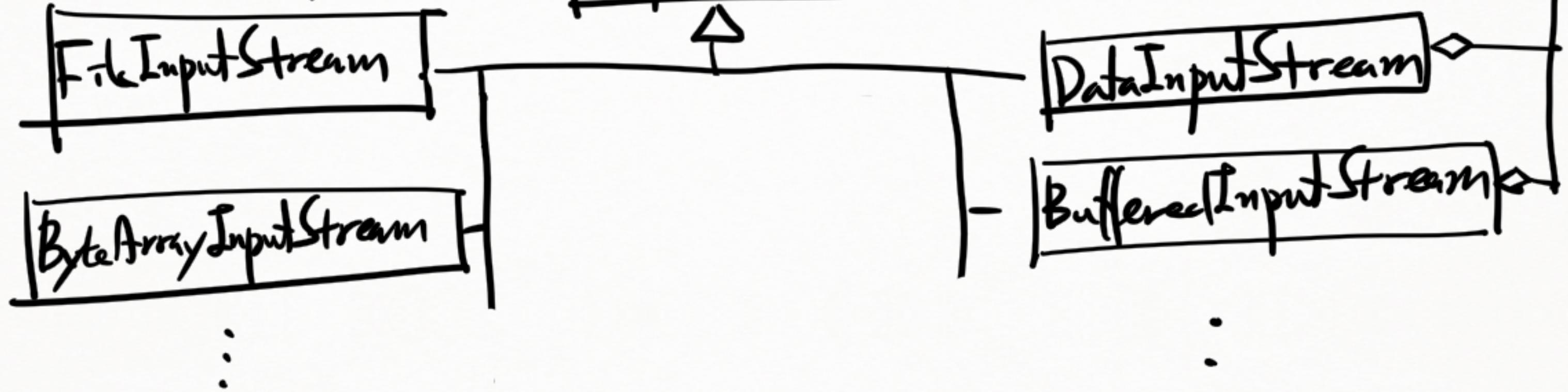
ConcreteDecorator

ex08.\*

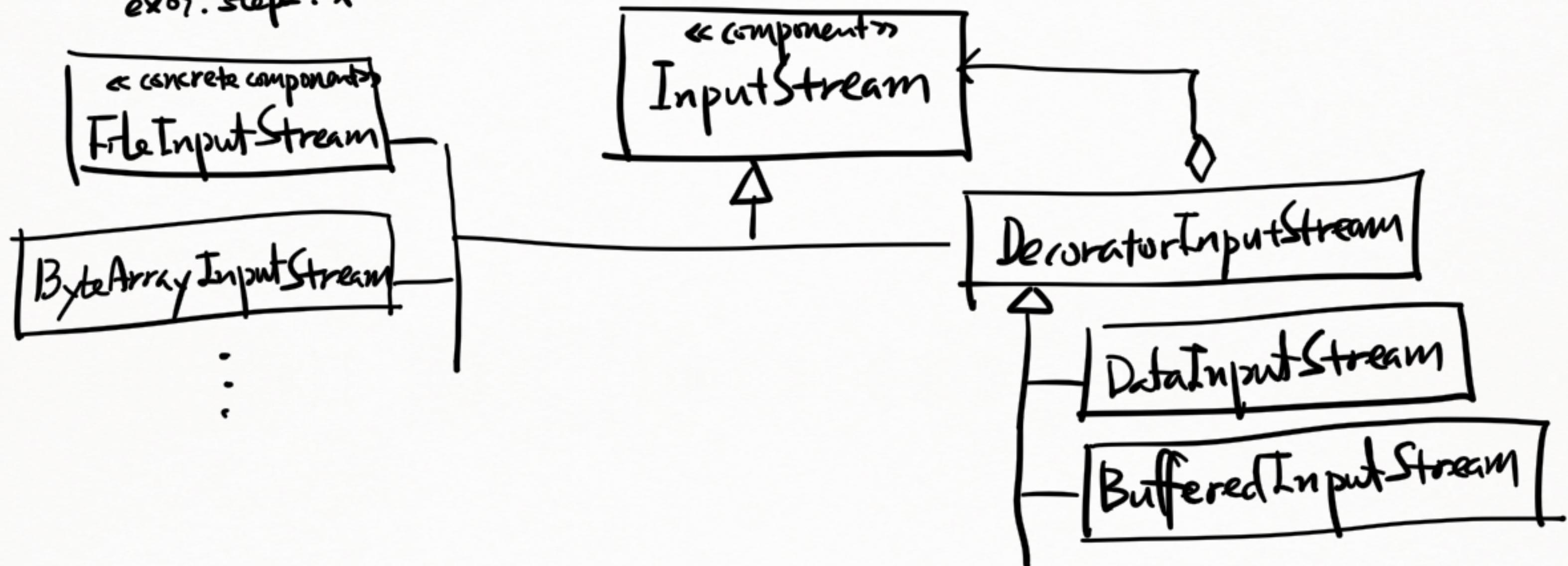


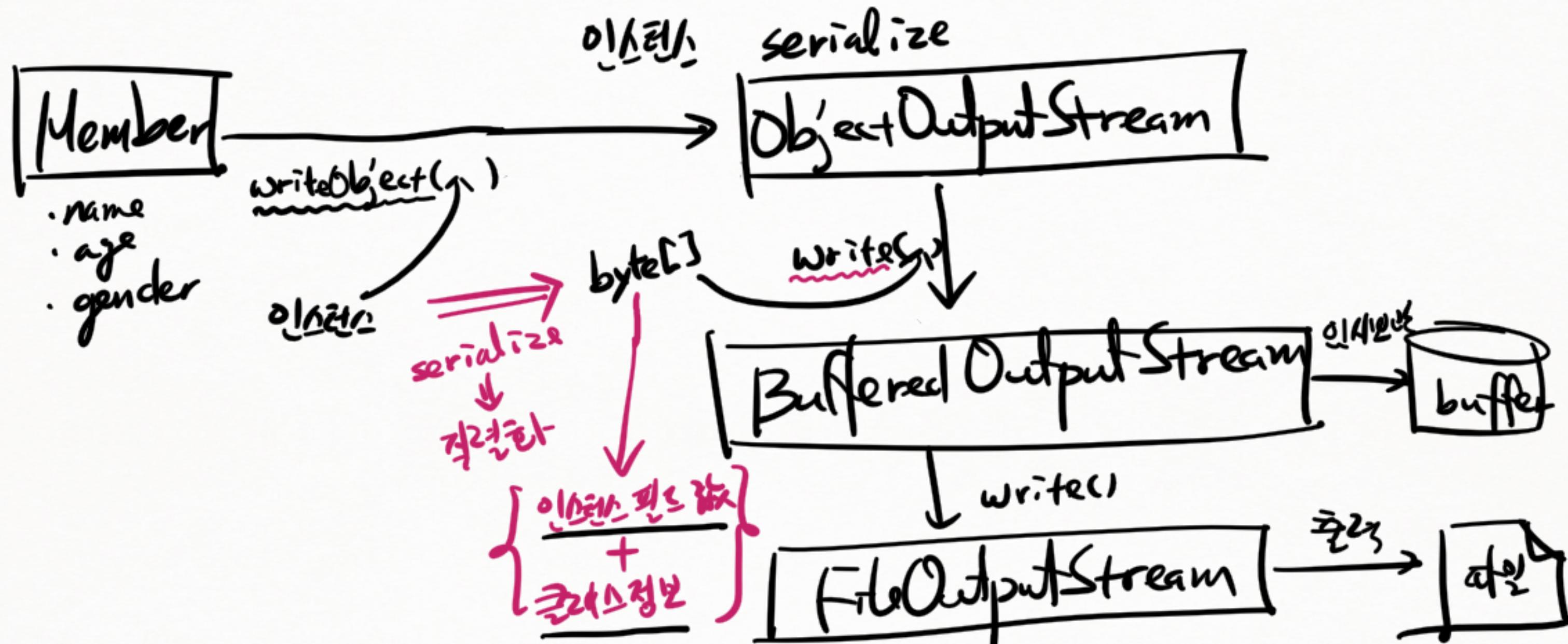
ex of step 1. \*

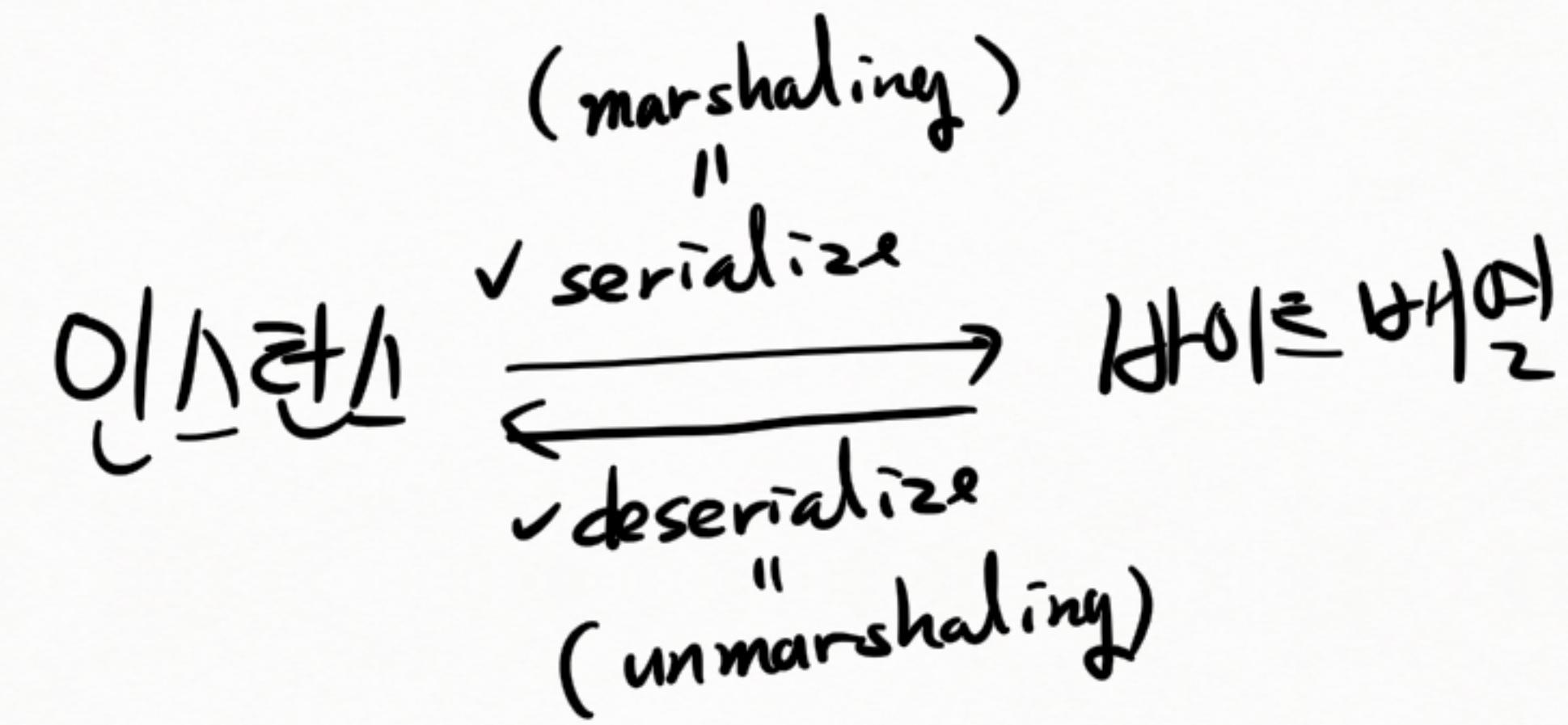
« concrete component »

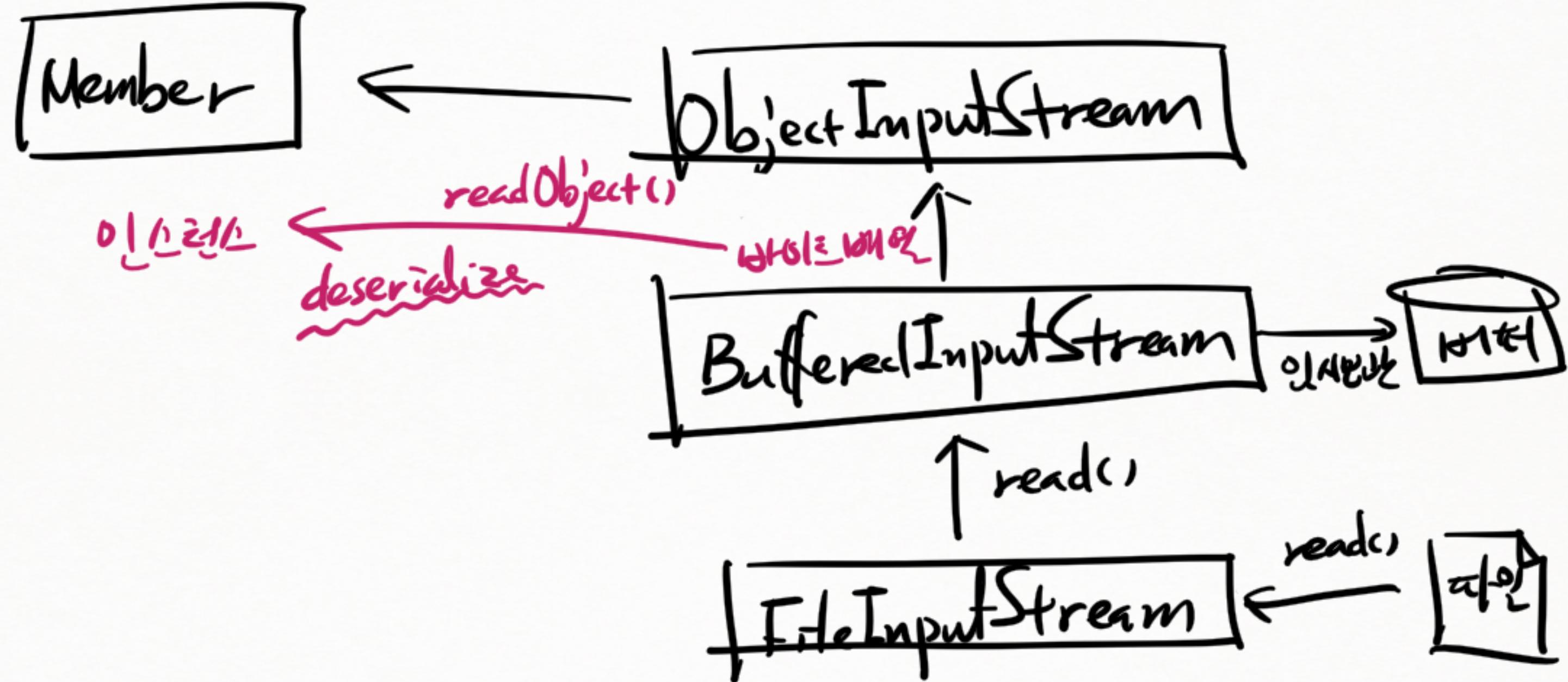


ex09. step2. \*





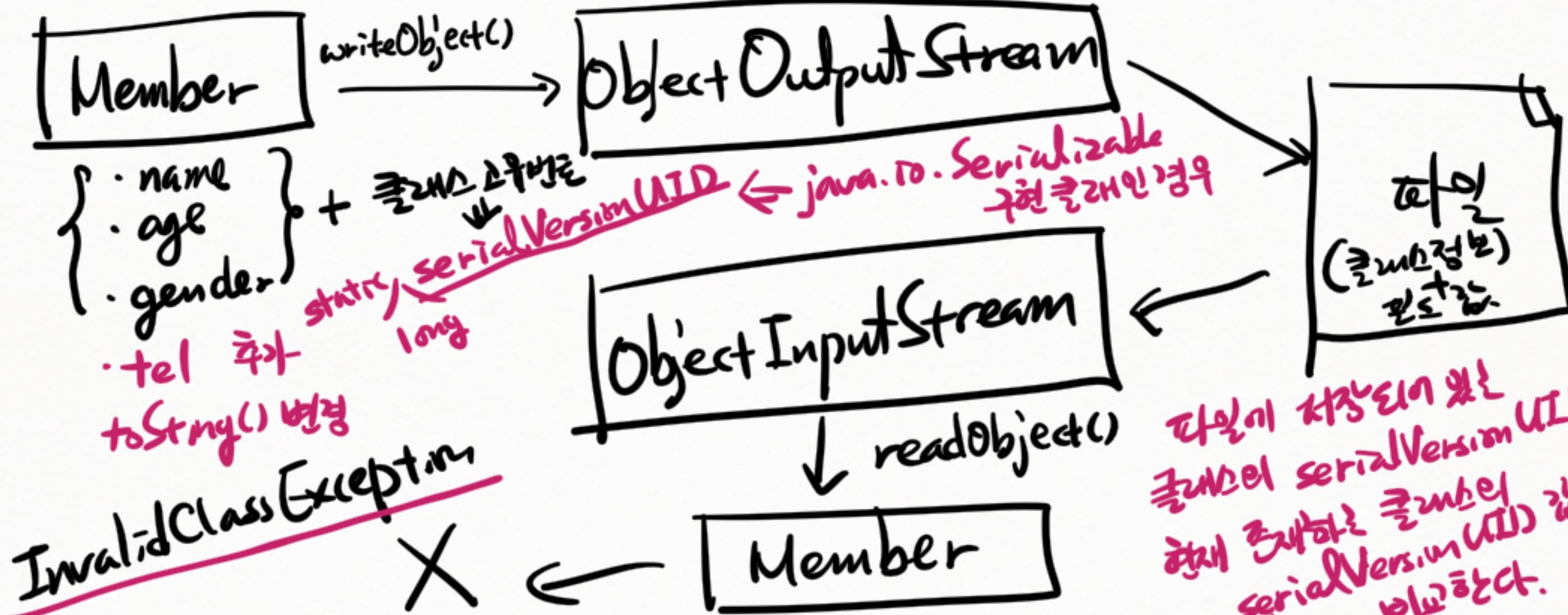




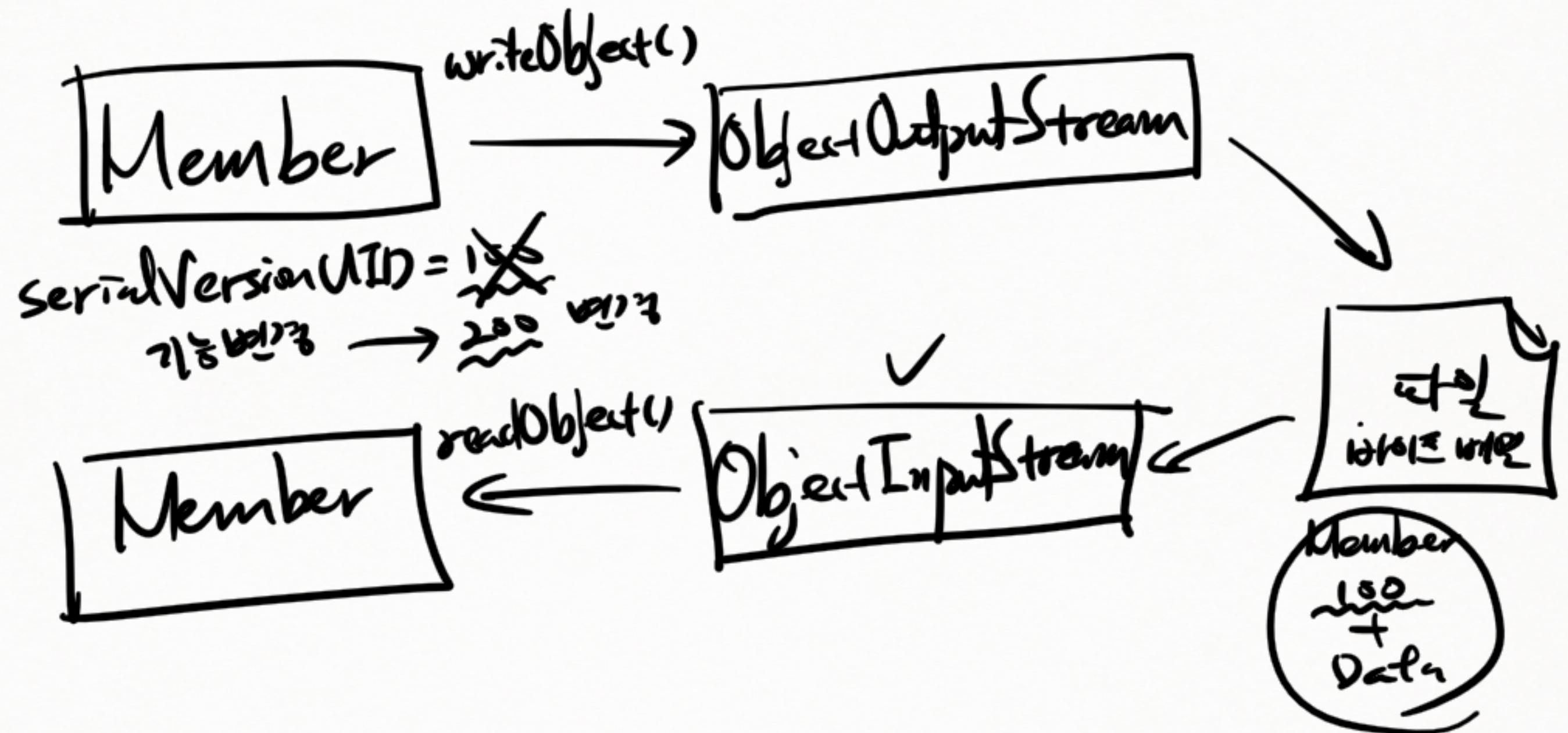
\* serialVersionUID  
값은 고정하지 않고 자동으로 추가된다.  
이유의 경우를 찾는다 → 클래스가 변경되었을  
때마다 값은 바뀐다.

① unique

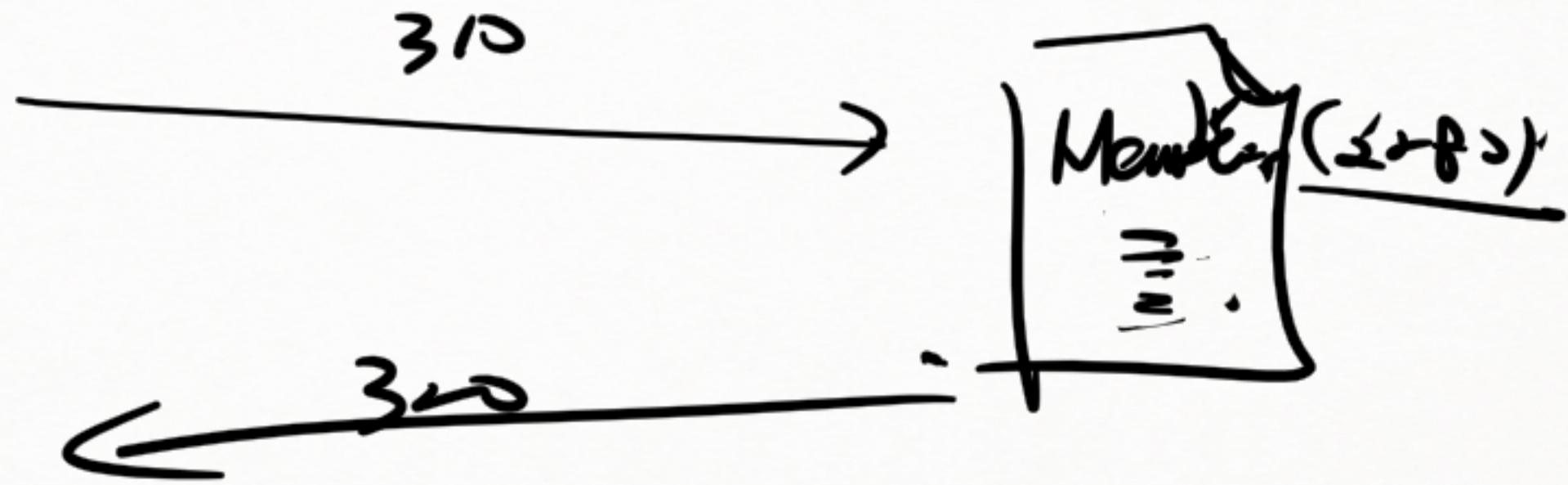
② IDentifiable



파일이 저장되어 있는  
클래스의 `serialVersionUID`와  
현재 클래스를 동일한  
`serialVersionUID` 값이  
같은지 비교한다.



~~1018~~ Member  
~~2283~~  
- name  
- age  
- gender  
- ~~test~~



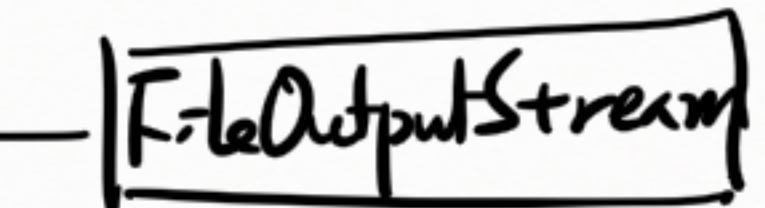
## Data Sink

### File Format

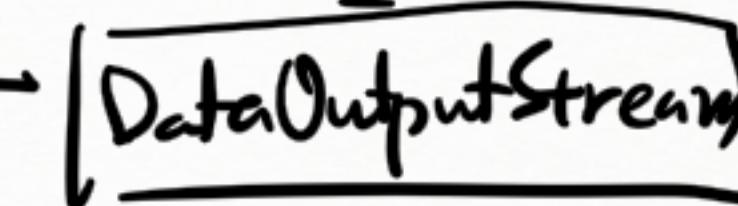
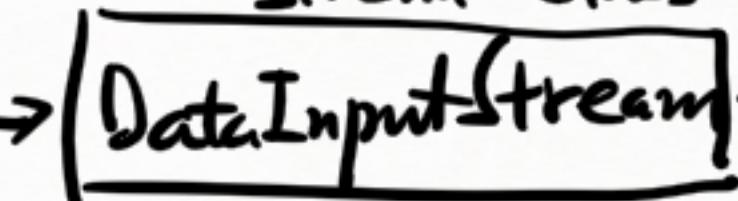
JPEG  
GIF  
WAV  
MP3  
PPT  
:



### Stream Class



### Data processing Stream Class



→ `readInt()`

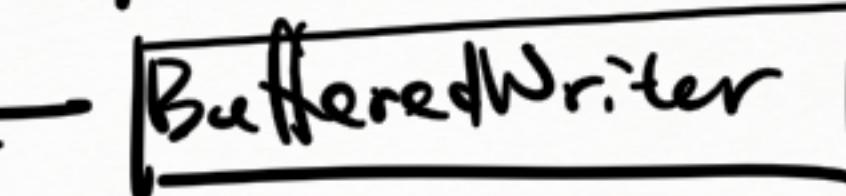
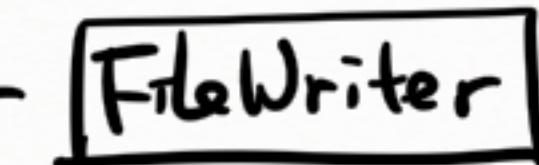
:

⋮

← `write()`

⋮

TXT  
JAVA  
CSS  
HTML  
MD  
CSV  
JSON  
⋮



⋮

- 30-a : FileInputStream / FileOutputStream
  - 30-b : DataInputStream / DataOutputStream
  - 30-c : BufferedInputStream / BufferedOutputStream
  - 30-d : ObjectInputStream / ObjectOutputStream ← 다른 언어와 호환이 안된다
- } file format은 알아야  
알고 싶을 때 있어

30-e : 라이브러리

31-a : FileReader / FileWriter      text 파일 포맷

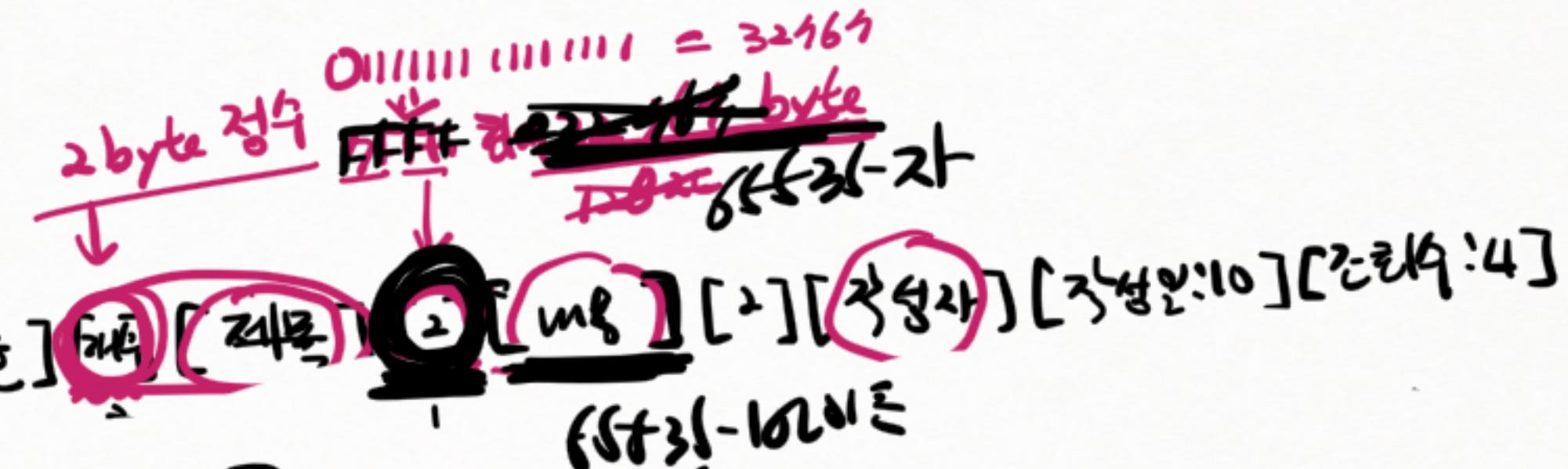
31-b : BufferedReader / BufferedWriter

31-c : 라이브러리 I

31-d : 라이브러리 II

4 bytes

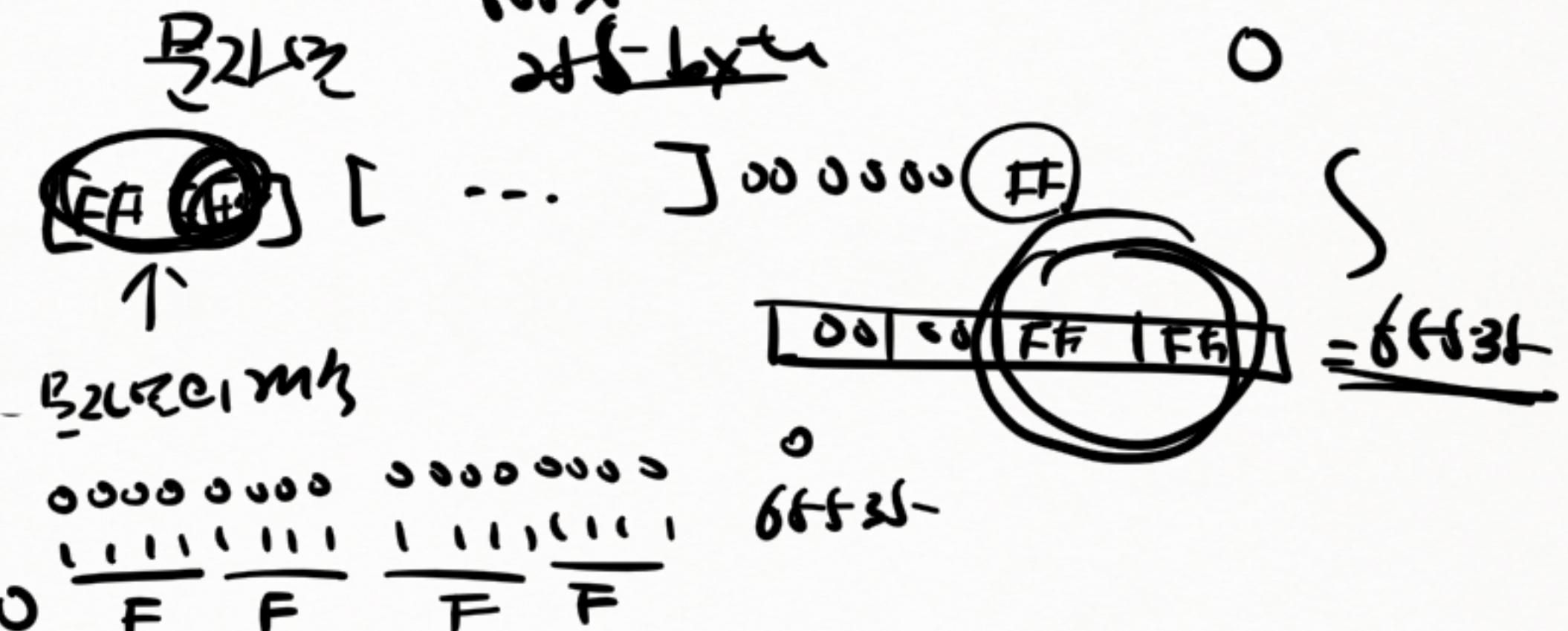
[길이수: 4]



int <sup>len</sup> a = 65535-;

write(a >> 8);  
00 00 FF |

write(a);  
00 00 FF



XML  
HTML

(자료)tag = markup (출판자)

data를 설명 (meta-data)

`<no> data </no>`

시작태그

끝태그

필드

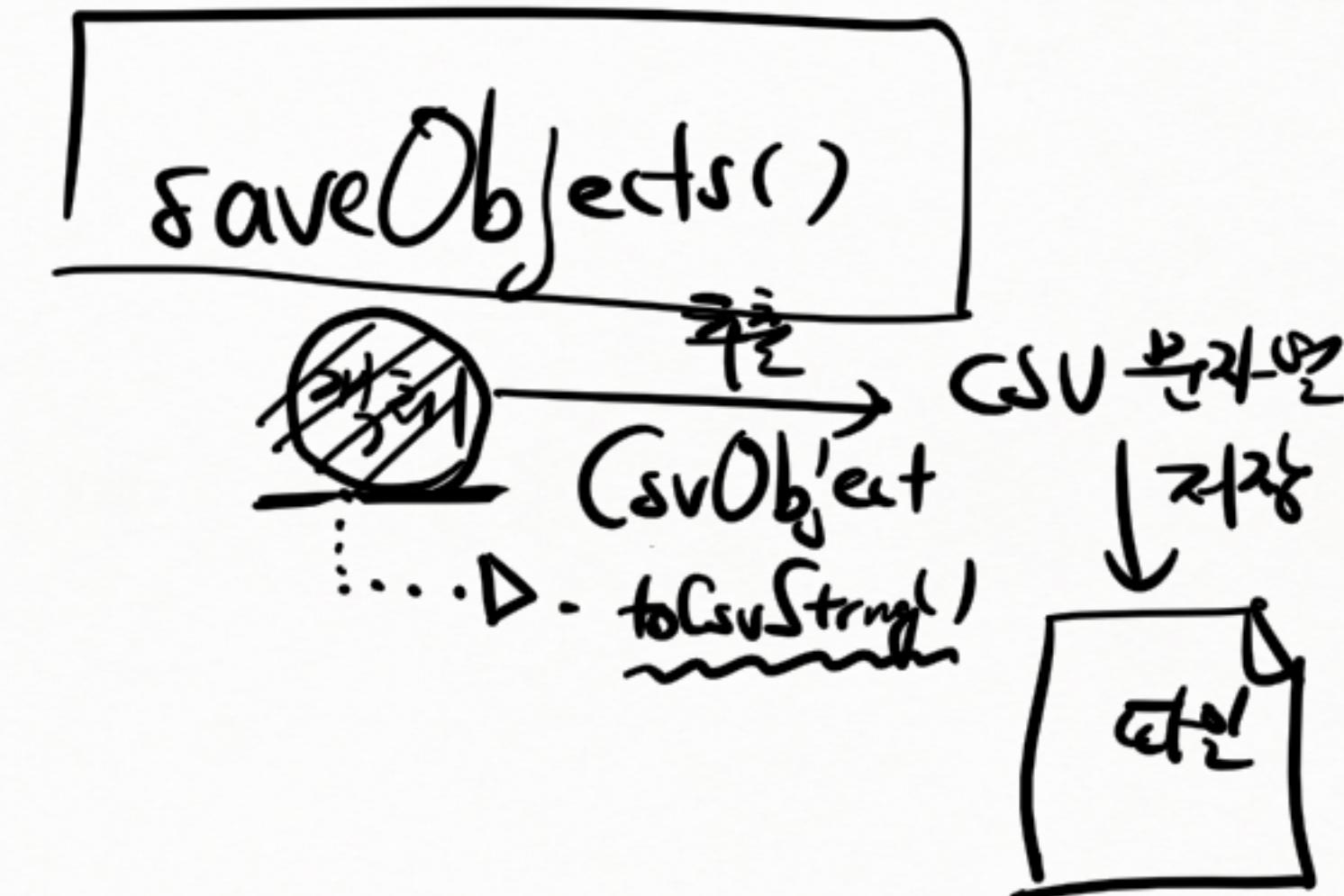
markup (출판자)

element (개별자)

`<h1>`

`<b>`

}  
saveBoards()  
saveMembers()  
saveProjects()  
saveTasks()



## . Data 표현

- Text
- Data 구조

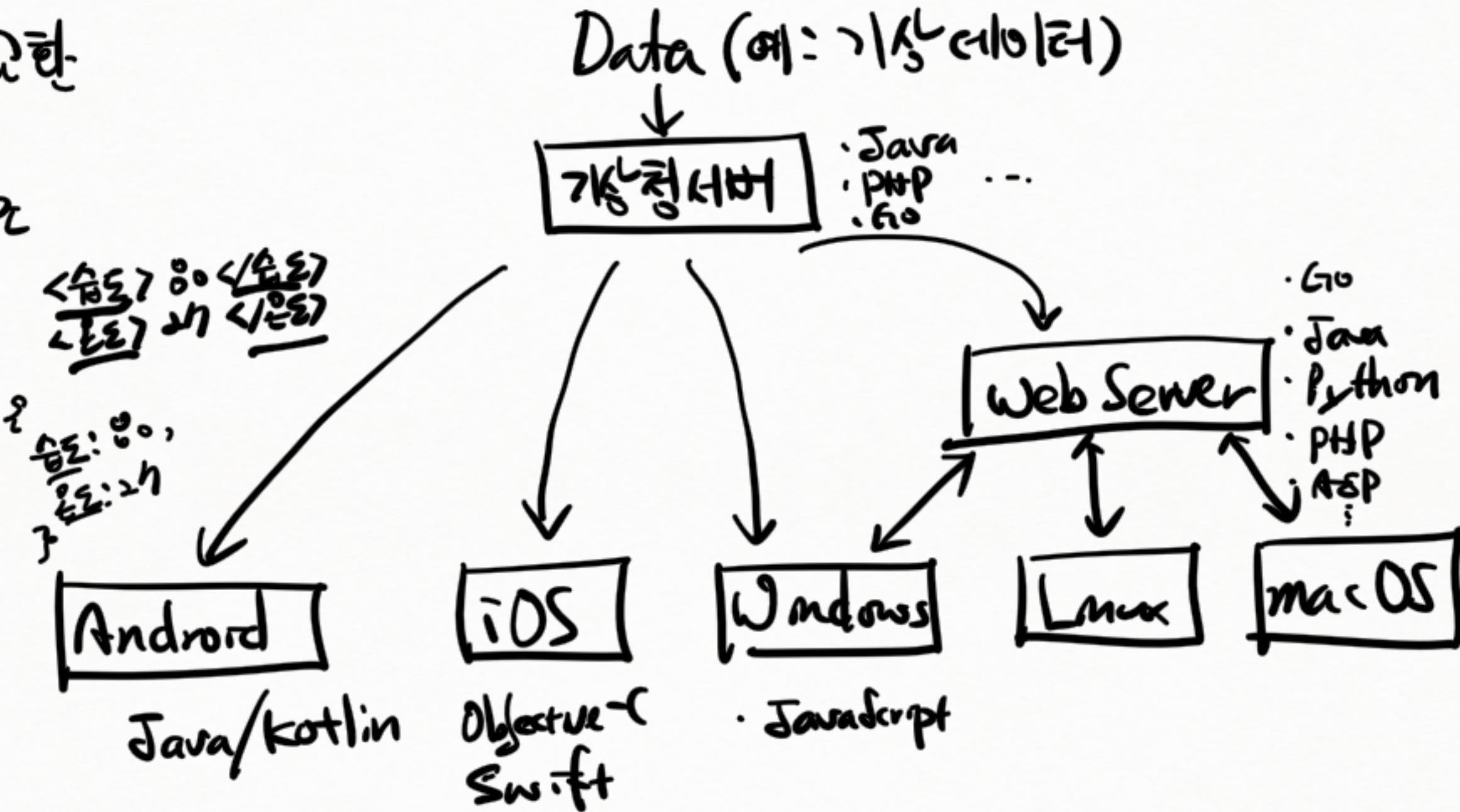
✓ XML

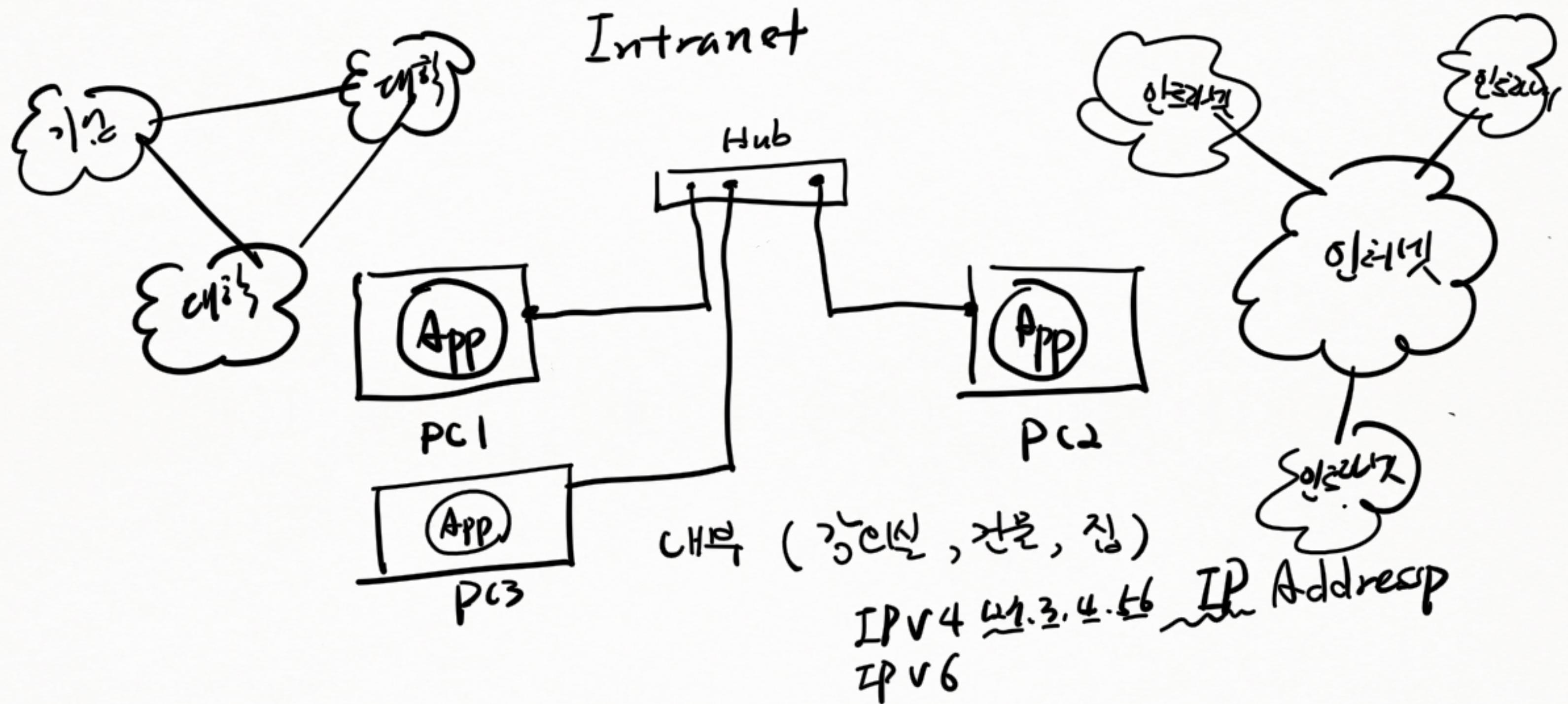
✓ JSON

JavaScript

Object

Notation





0  
0 423

010-1234-4444

0-0.0.0 = 4 byte

255.255.255.255

121.354.119

1 byte 1 byte

221.522.143

22  
200  
101

-128~127  
0~255  
10  
A  
14.12.43.55  
0D.34.2B.31

72001-1221234

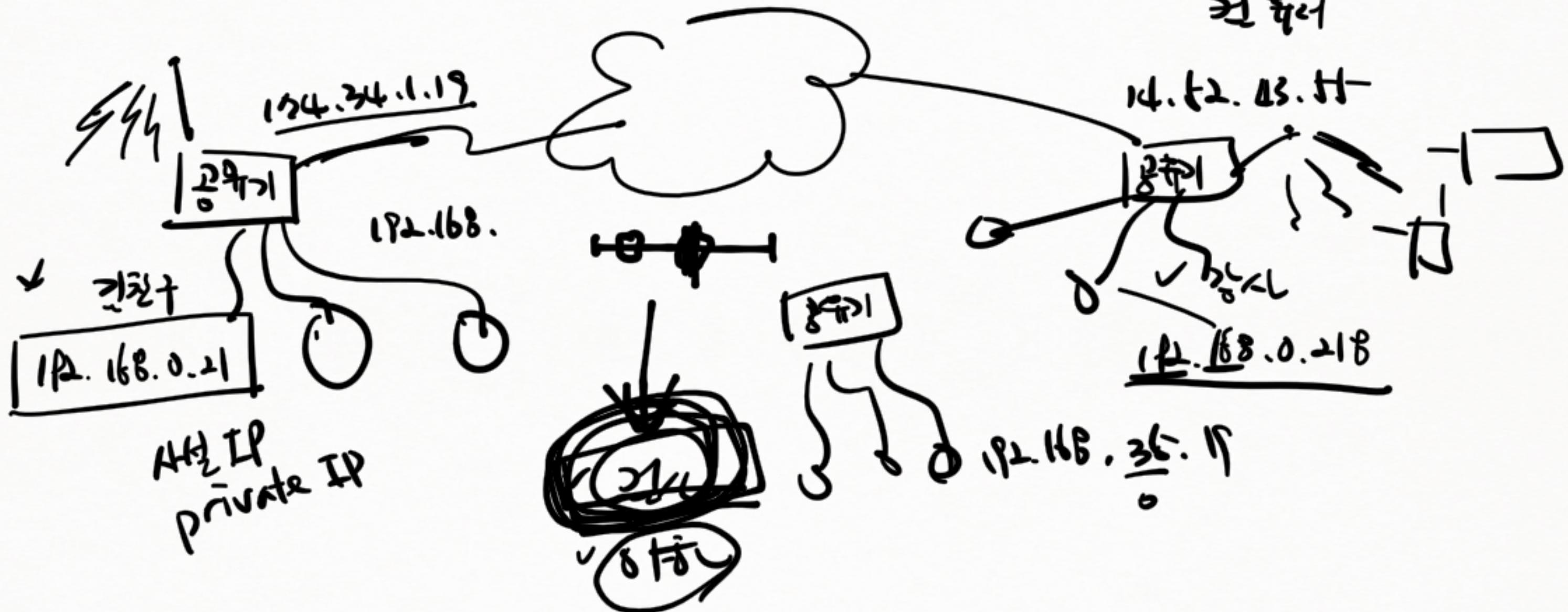
01234...

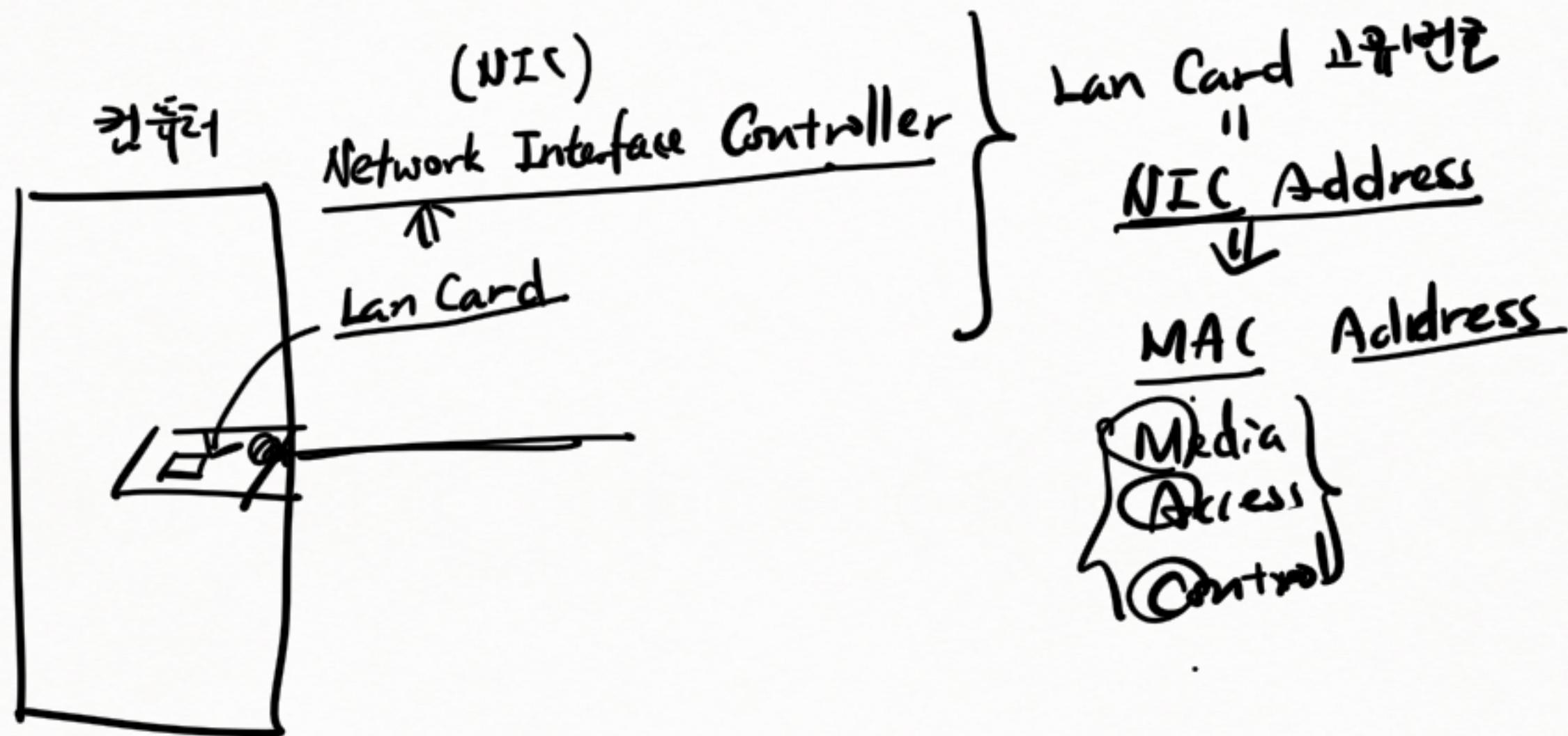
221522943 ... 423 -

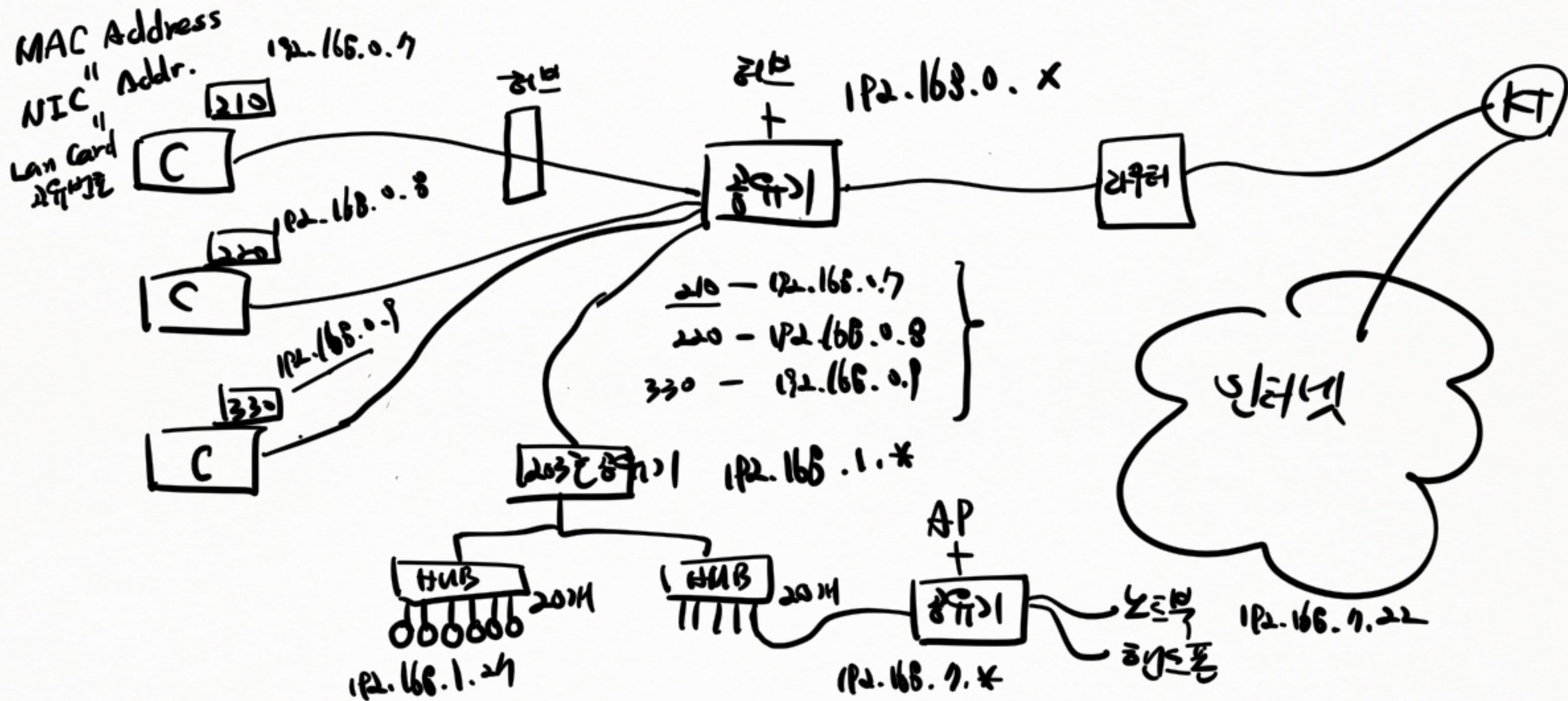
v4

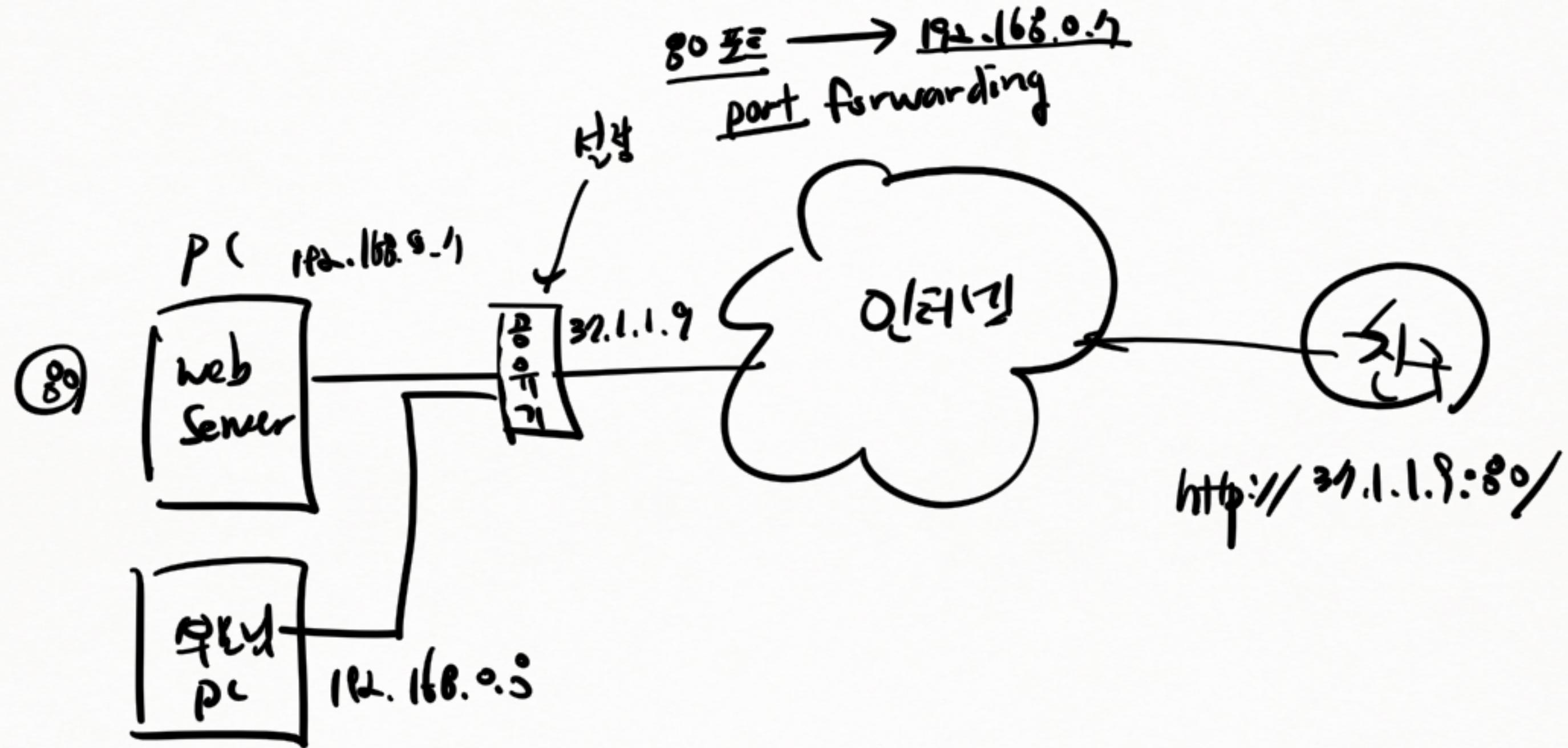
IP Address

$$4 \text{ Byt} = 32 \text{ BCD}$$





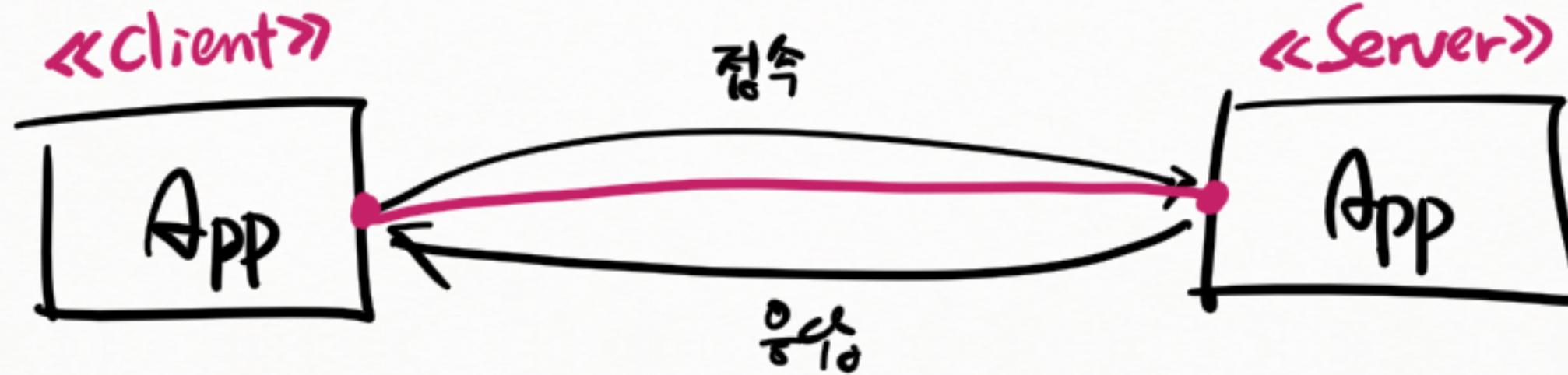


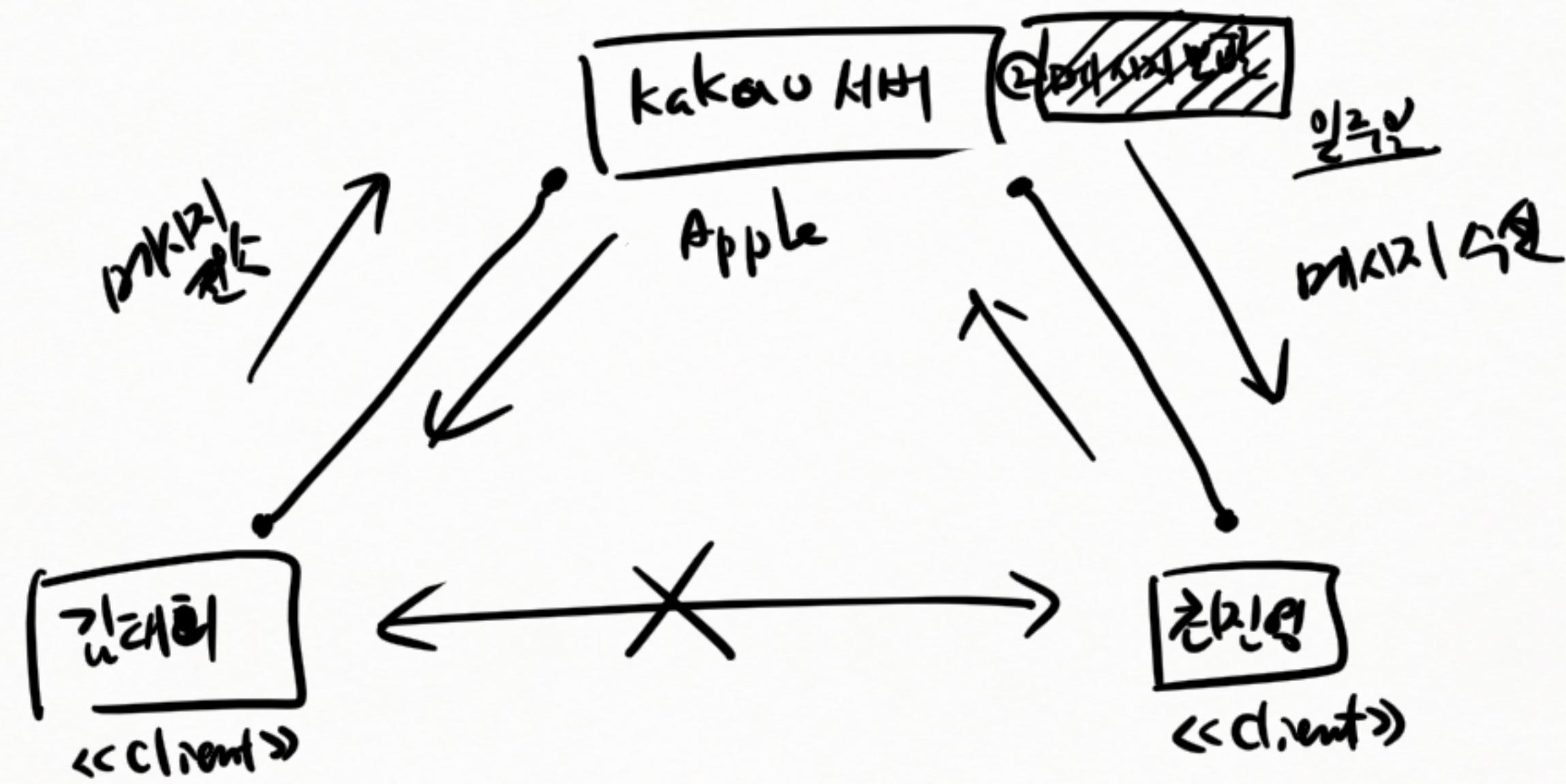


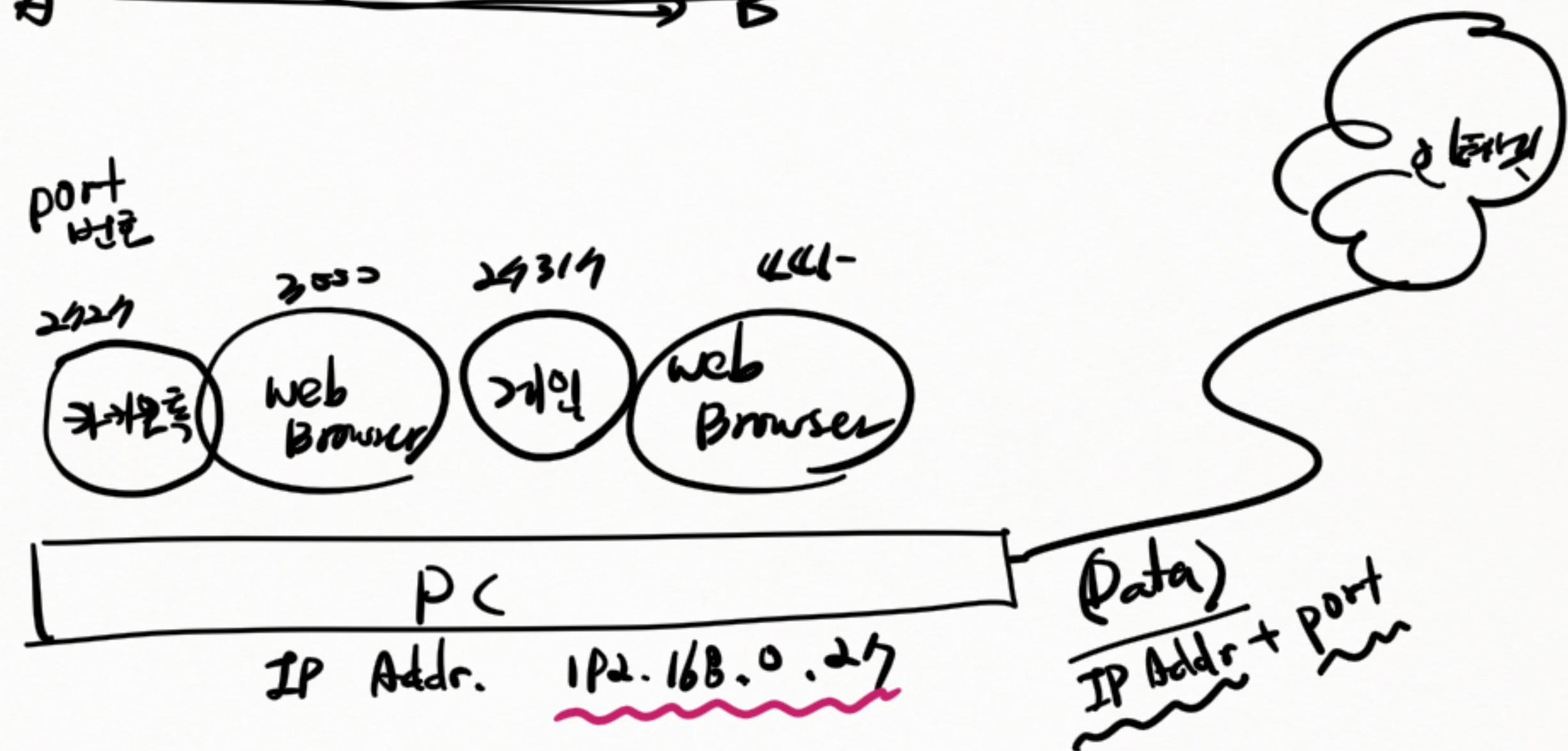
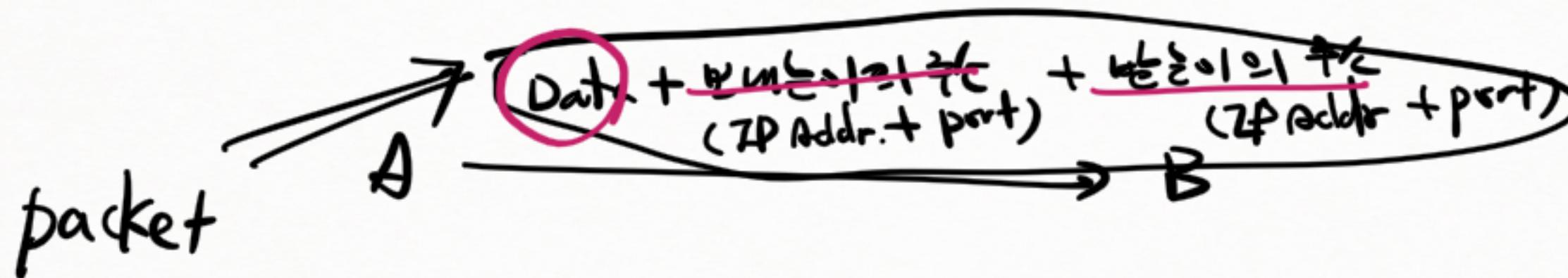
PC

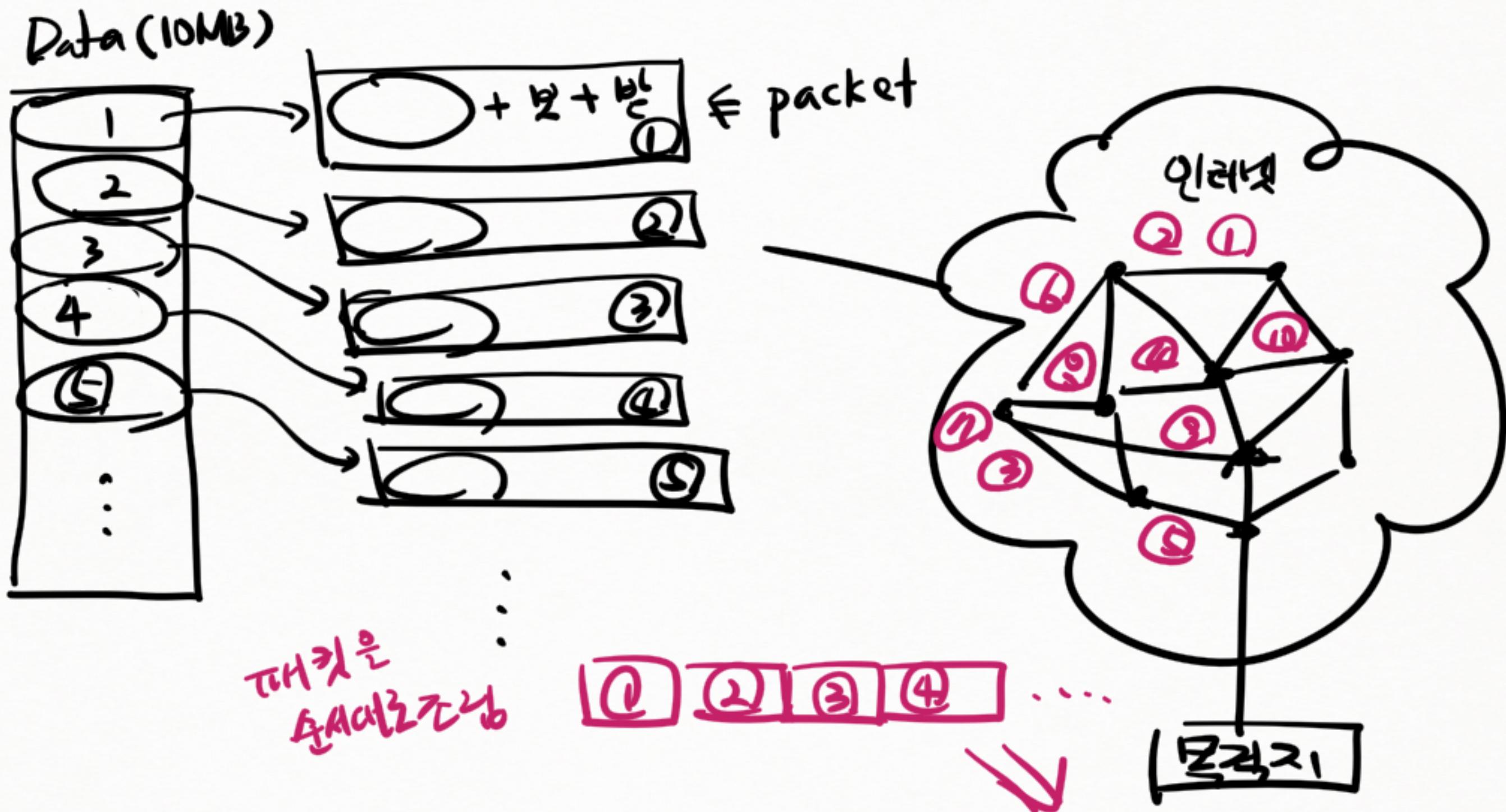
고객 = 사용자

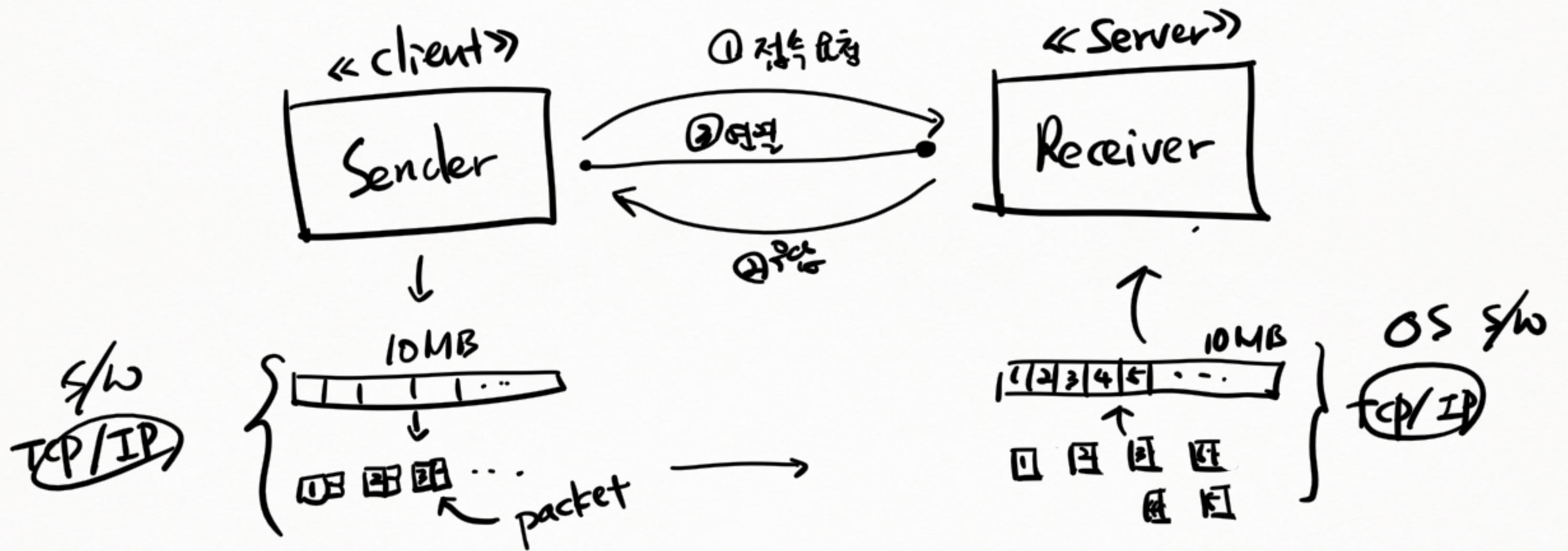
제공자





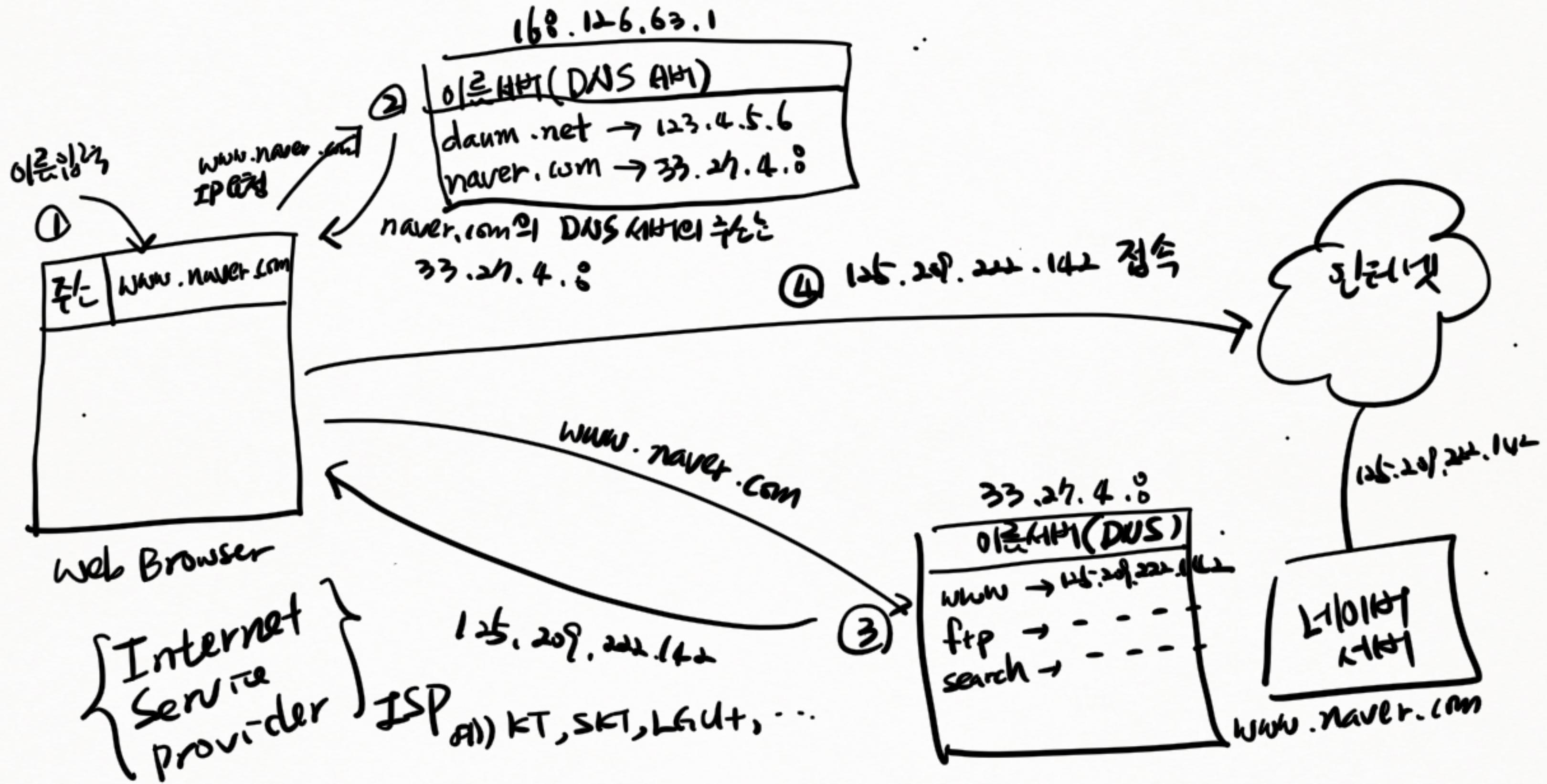






new Socket(IP Address, port)  
          ~~Domain Address~~  
localhost ⇒ localhost

127.0.0.1 ⇒ मेरा कंप्यूटर



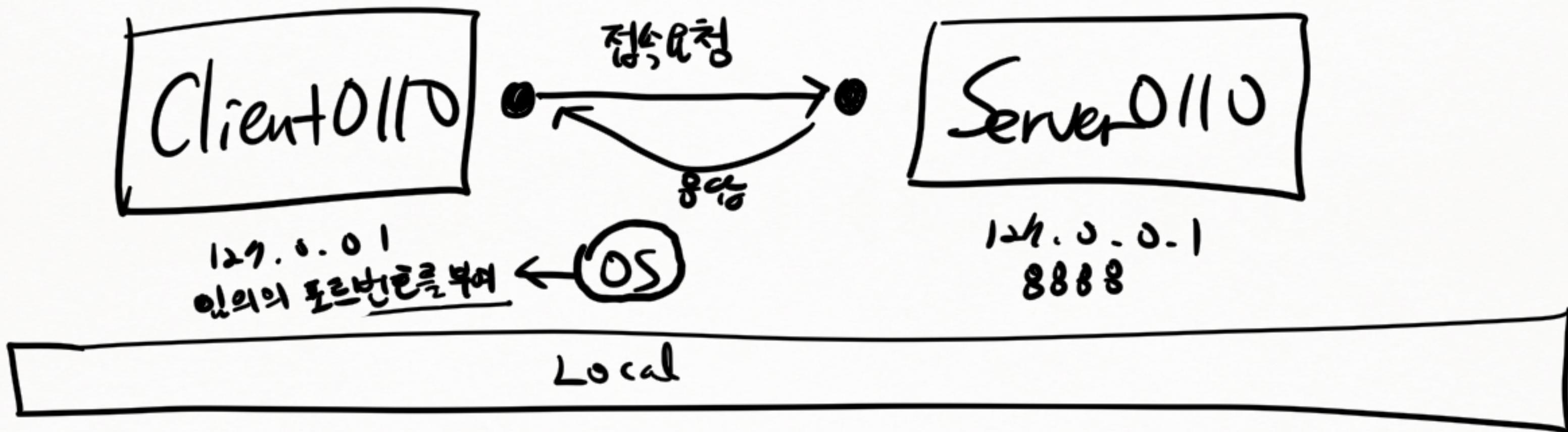
www. naver. com

↑                   ↑

Host name      Domain name

도메인 이름  
서버 이름

`new Socket("localhost", 8883)`



`Client`

↓      ↗ exception

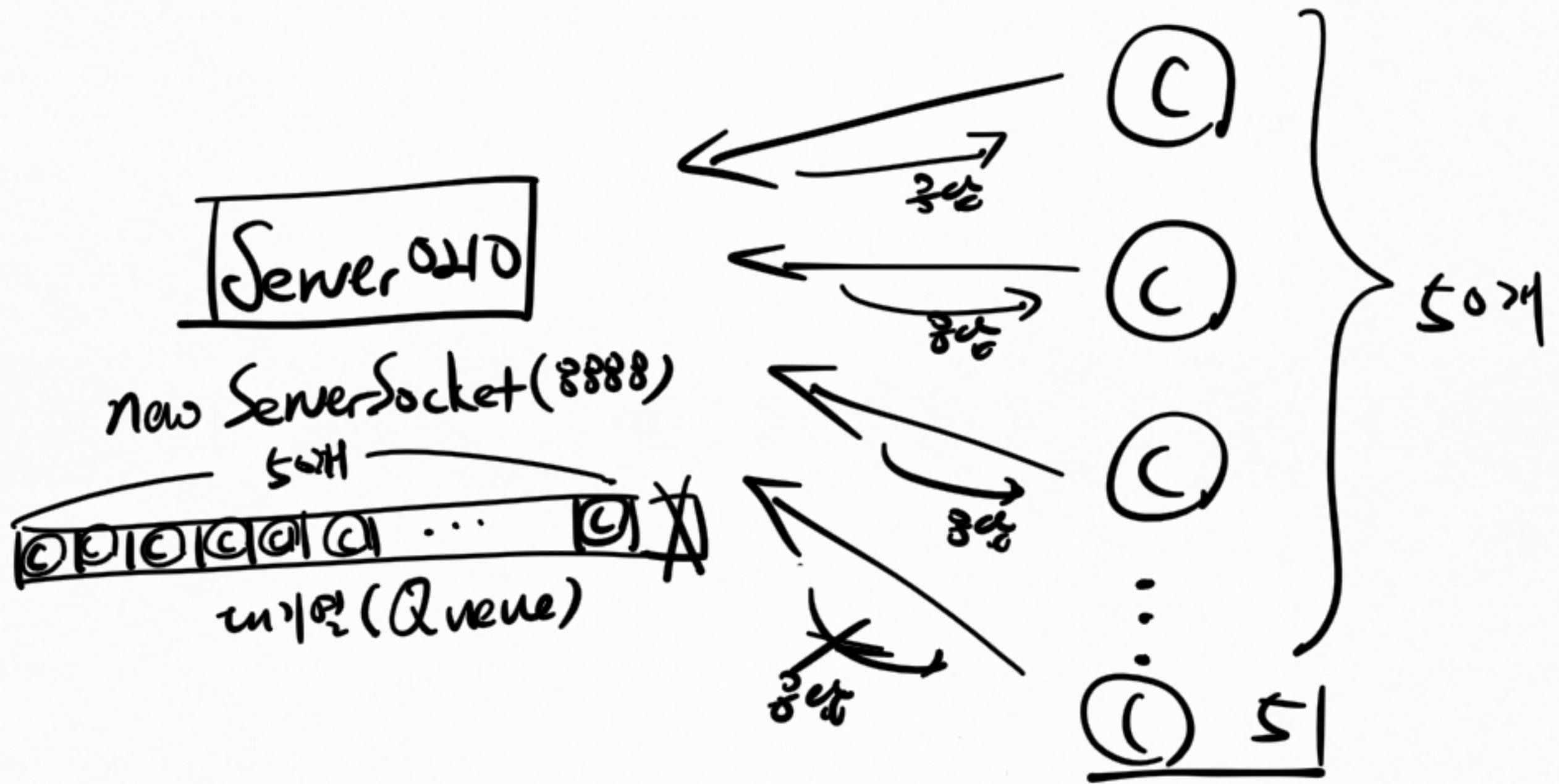
`new Socket(주소, 포트번호)`

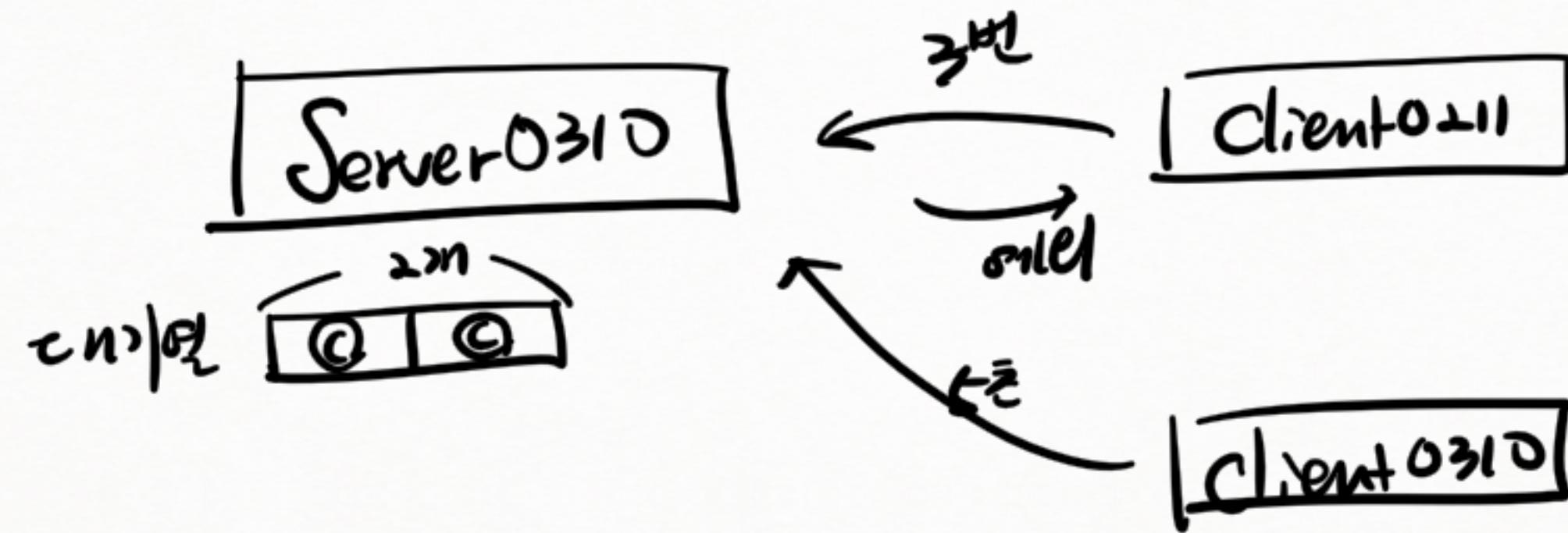
8888  
`Server`

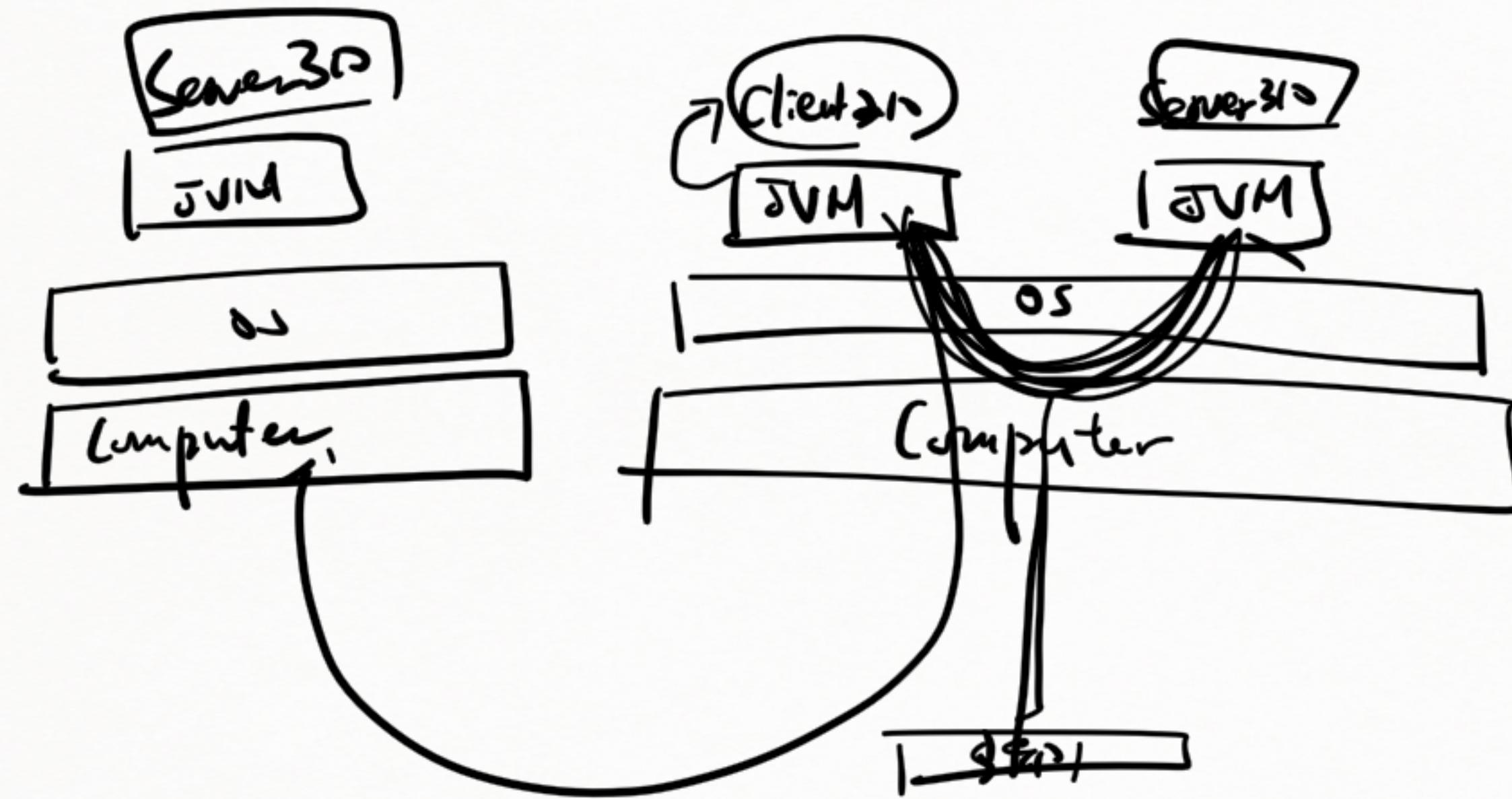
↓      ↗ exception

`new ServerSocket(포트번호)`

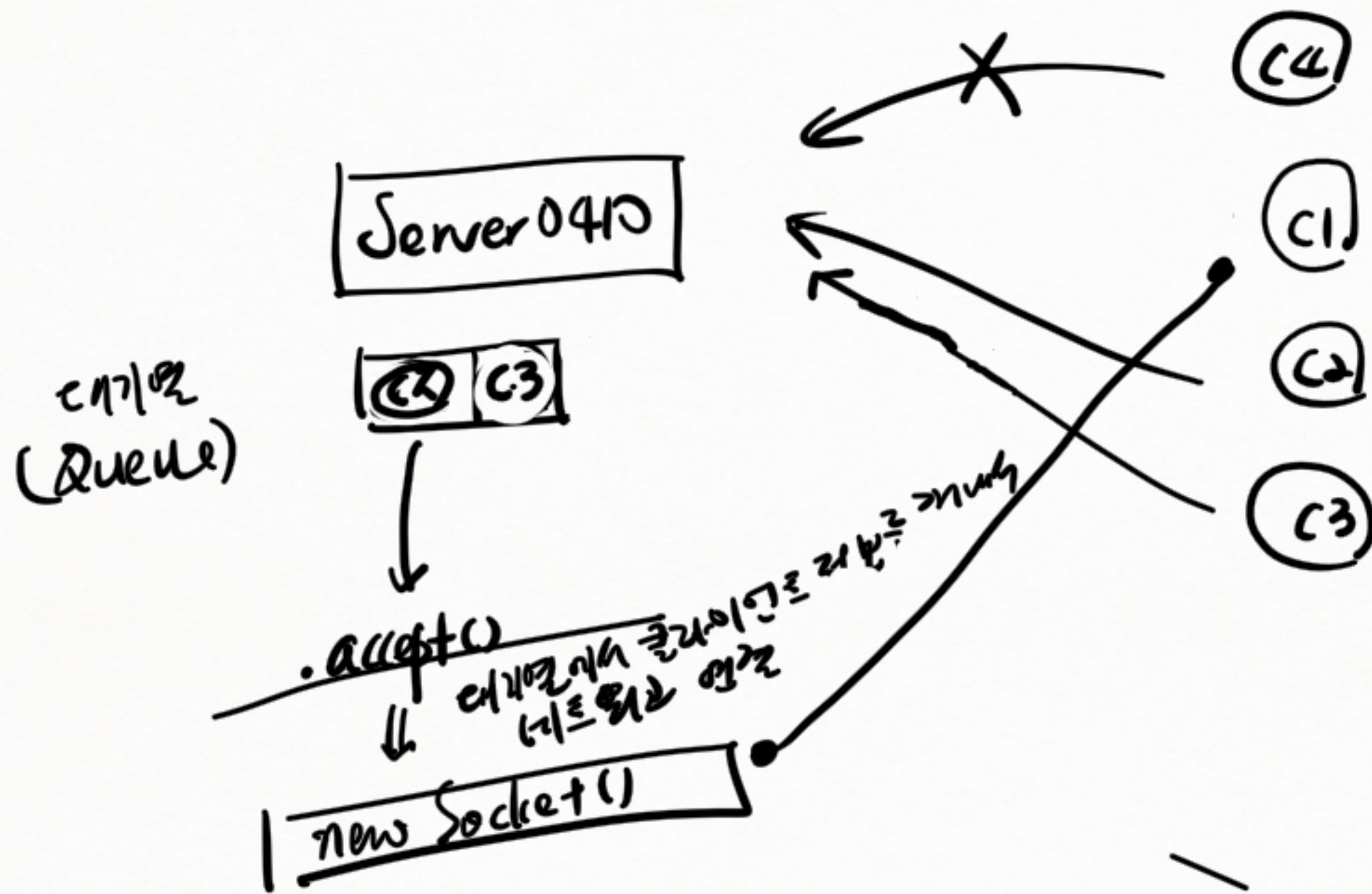


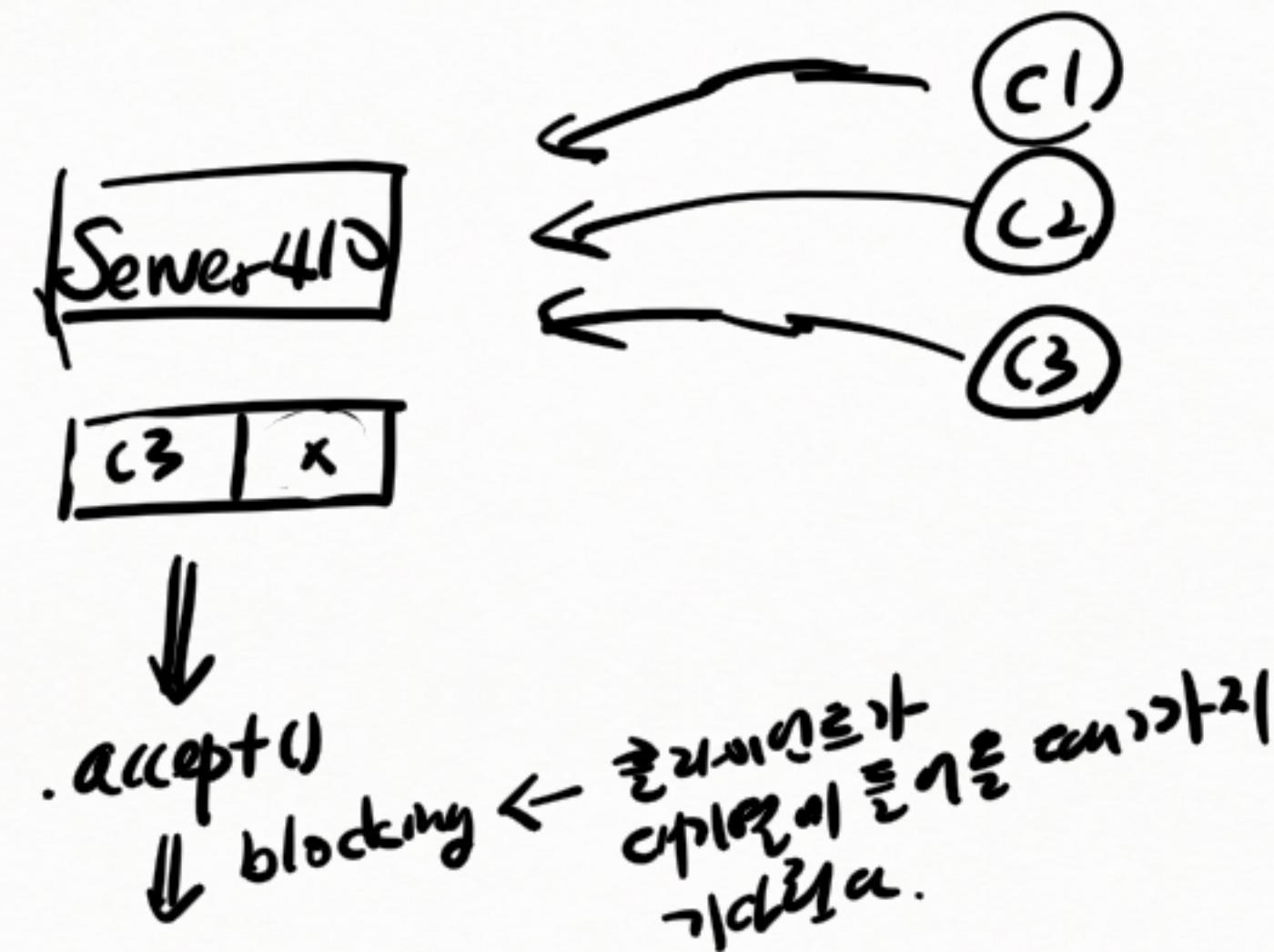


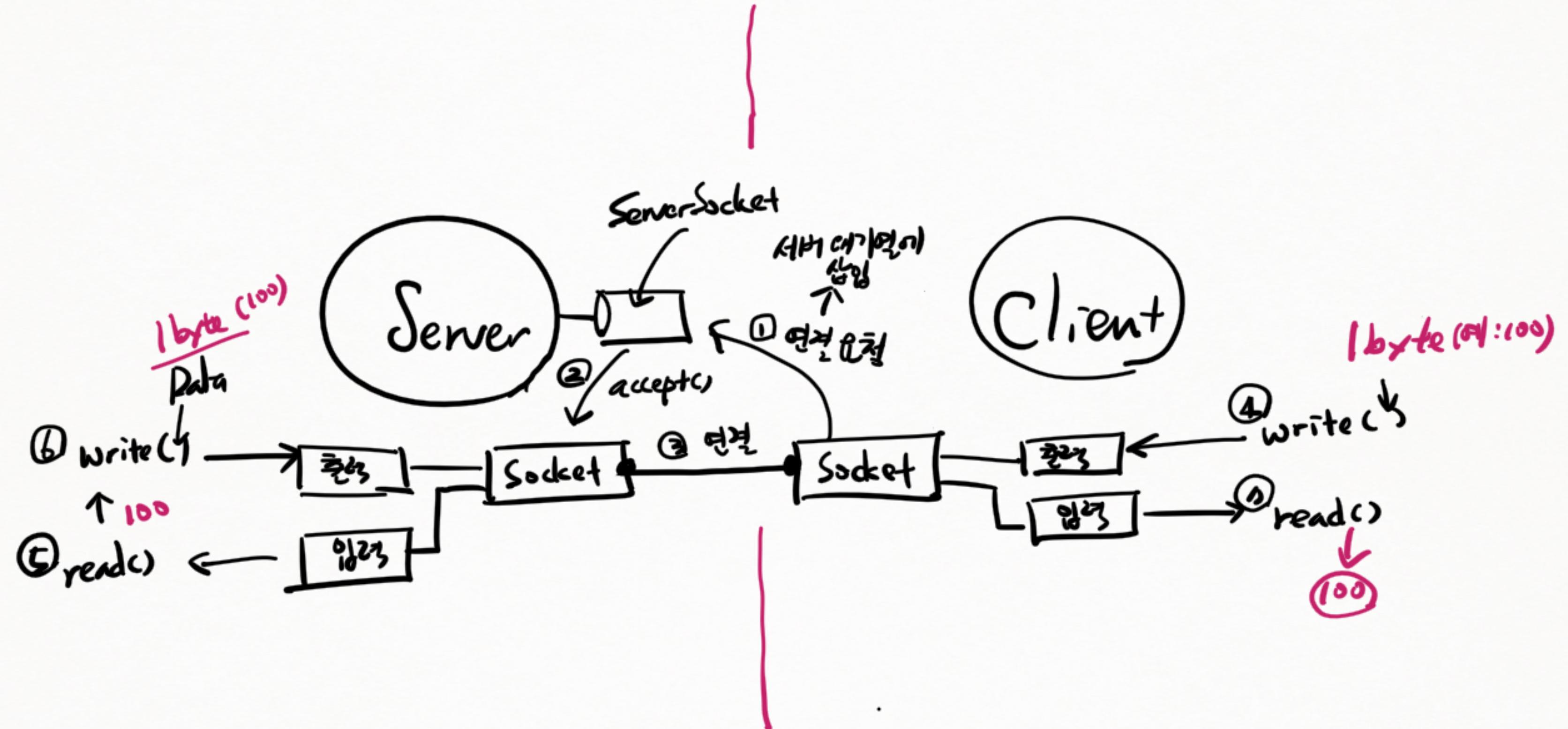


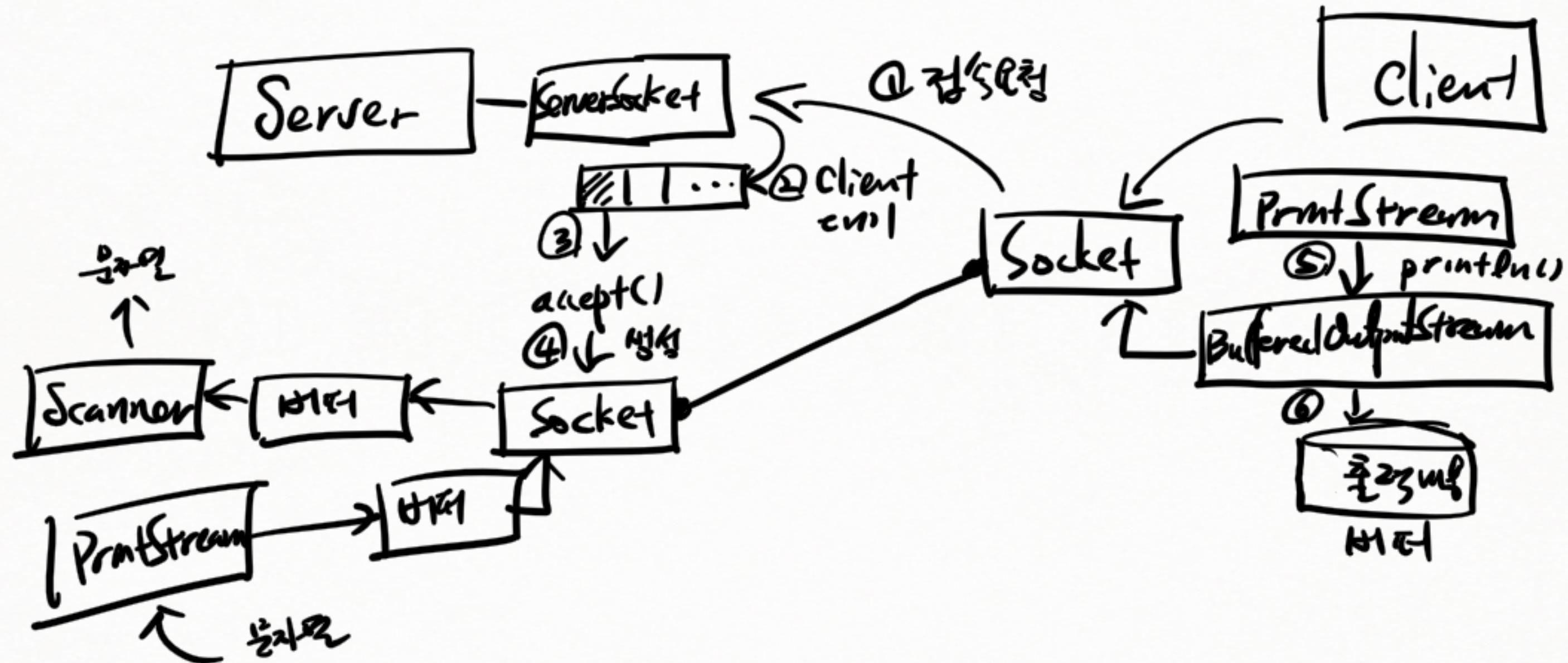


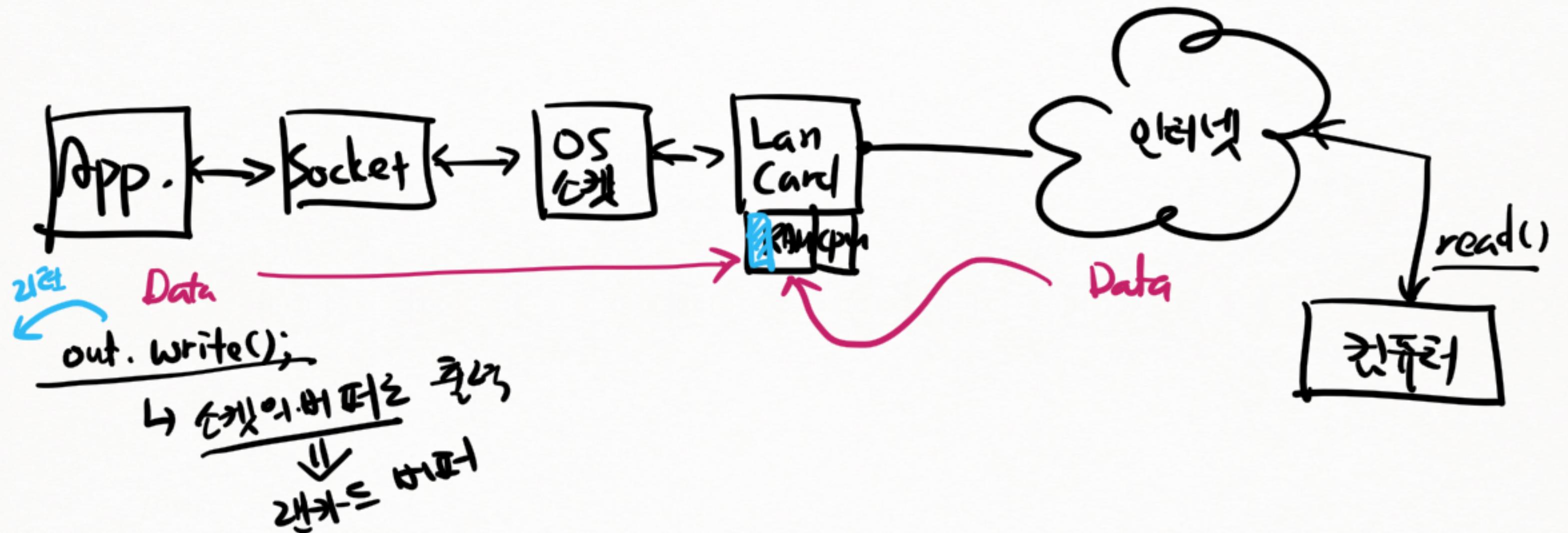
localhost  
127.0.0.1  
2525 TCP

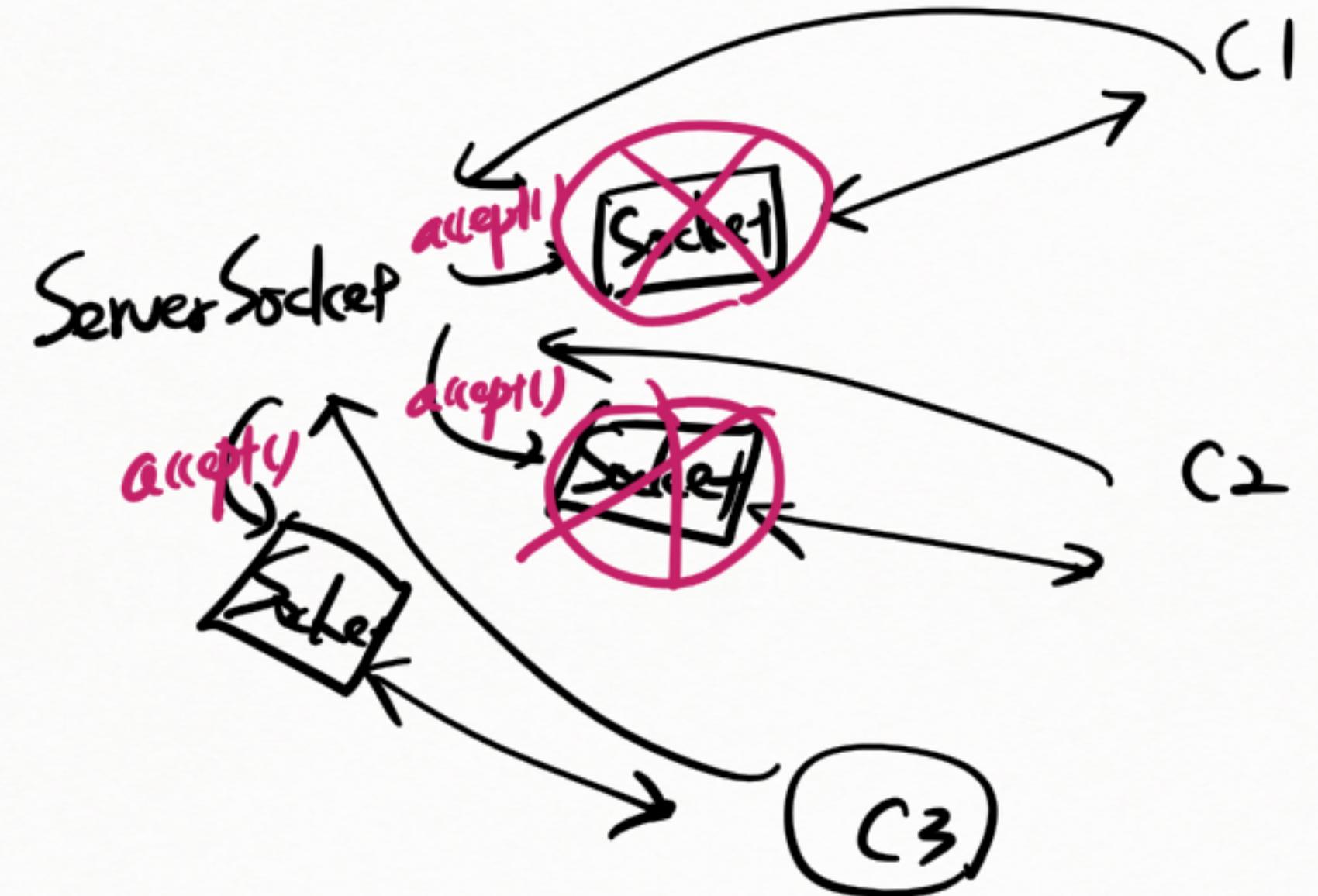


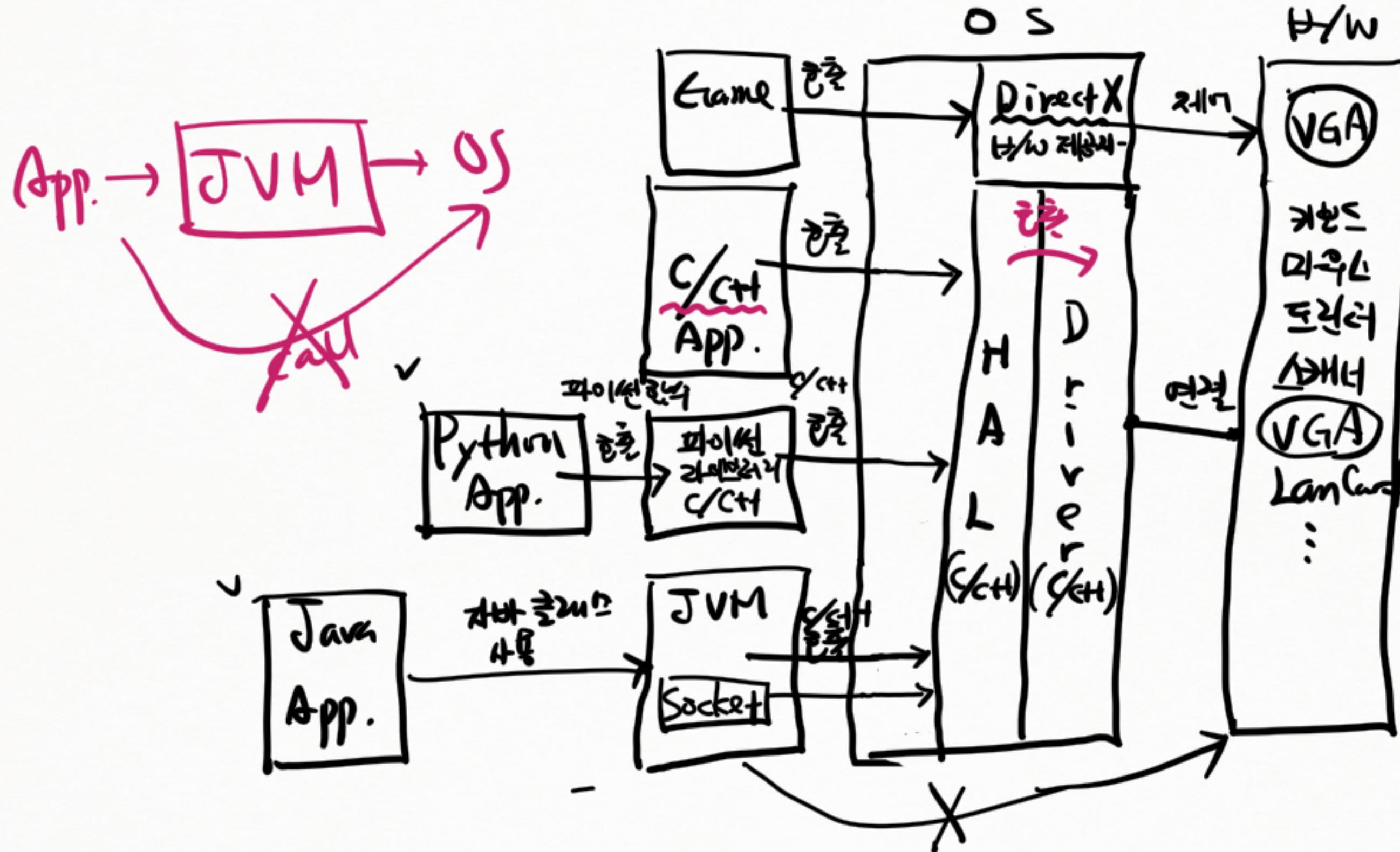


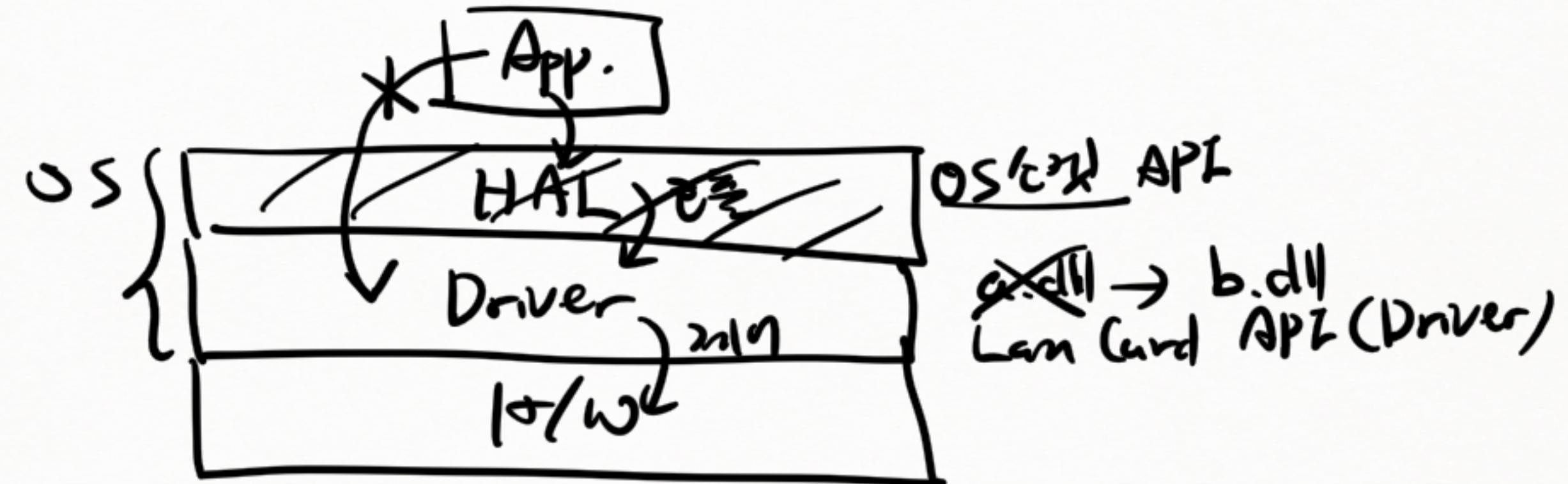


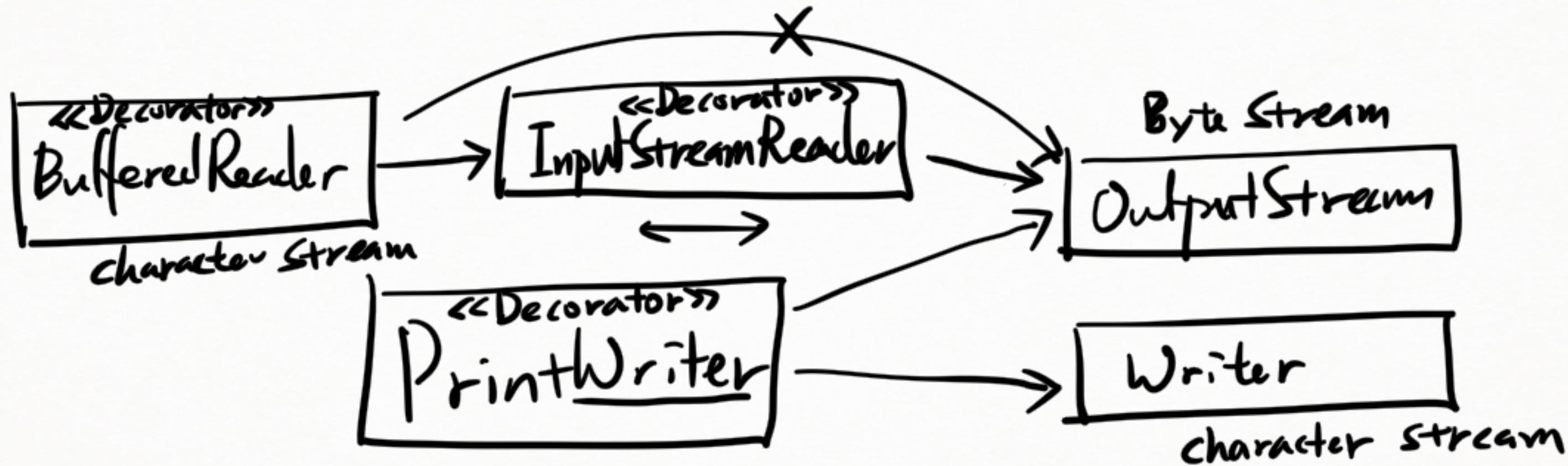








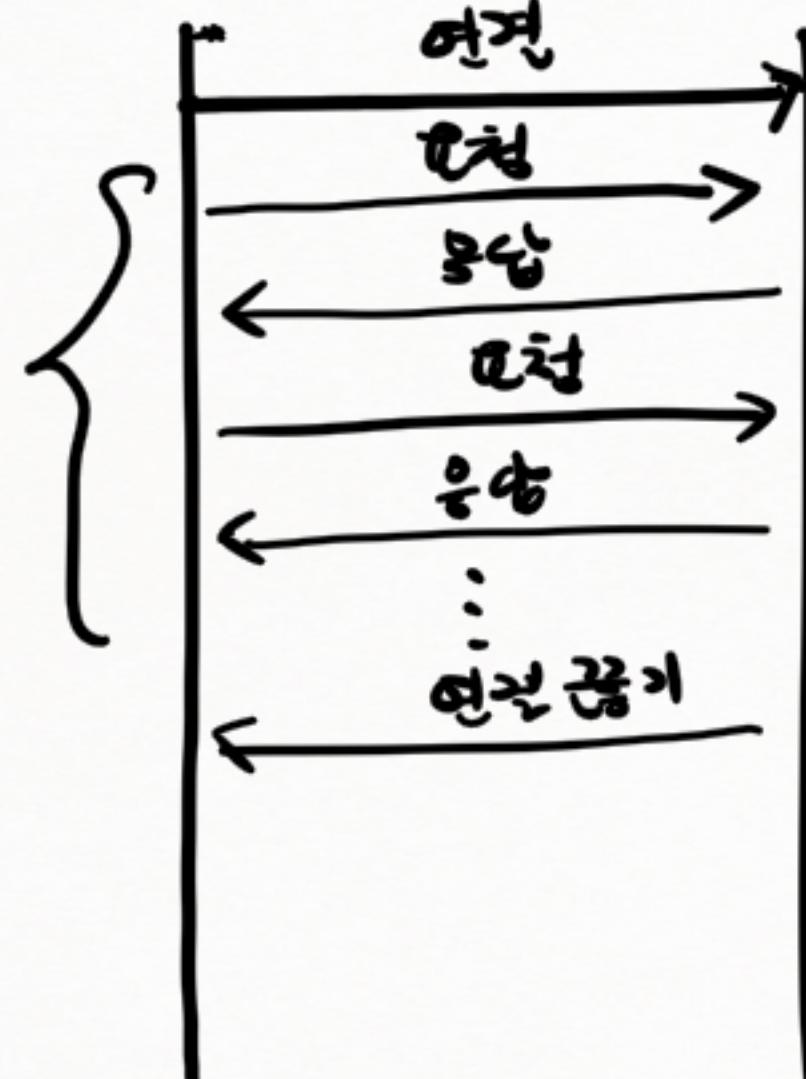




통신방식

## Stateful

Client Server



고개센트  
상담

• ፳፻፲፭

$\sigma_{\text{V}} = \sigma_0 \cdot \sqrt{1 + \frac{1}{2} \sin^2 \alpha}$

client 측 유저별 접근

127

a) FTP, Telnet

TCP

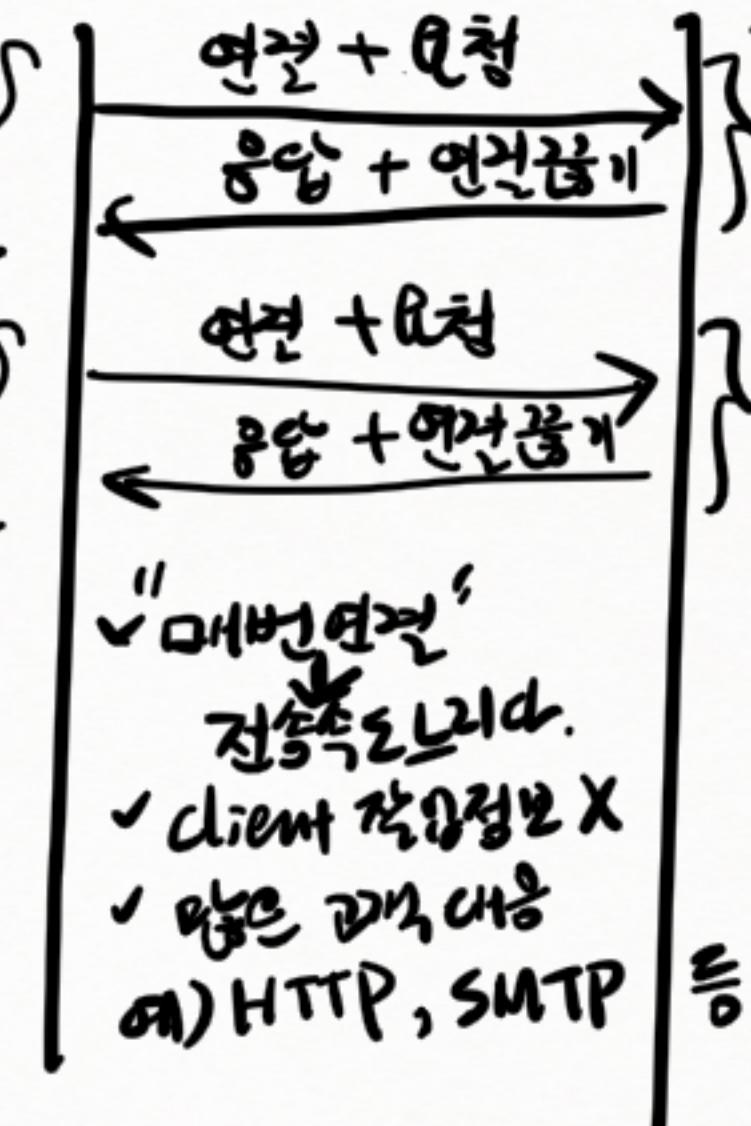
## Connectionful

114  
ott

114  
ok.

## Stateless

ent Server



"매번연결"

전속도록이다

✓ client 깔끔정보 X

יְהוָה יְהוָה יְהוָה

• HTTP, SMTP

CHIEN, JIN

Connectionfull

Client      Server

연결 후 전송

- 유튜브
- 아프리카TV } 채팅
- 오락인게임
- 전화
- 

TCP = Socket

+ IP

연결 방식

Connectionless

Client      Server

전송

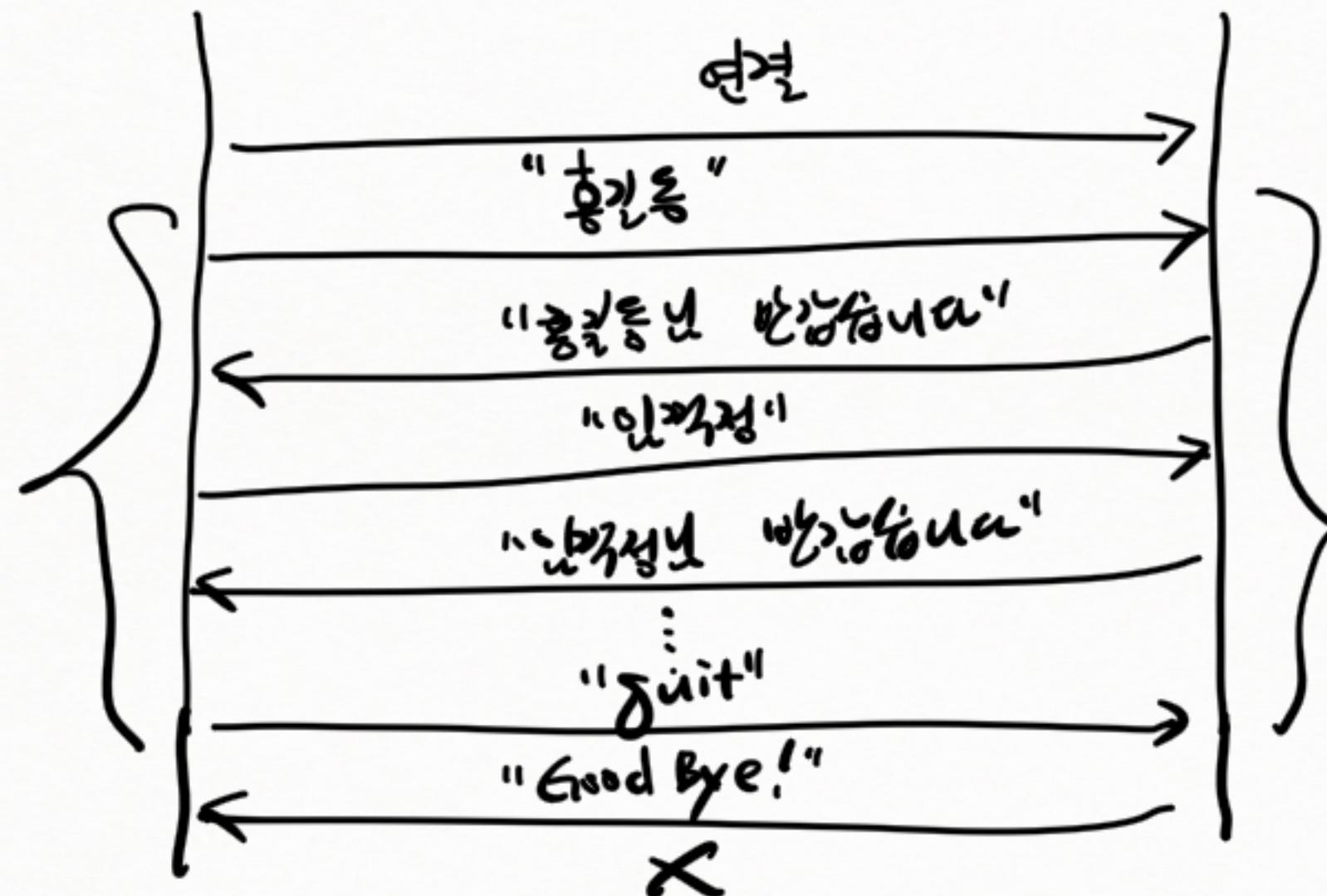
- 편지
- 이메일
- 메신저
- 문자- (메시지)
- TV  
라디오 } 방송

Datagram Socket = UDP

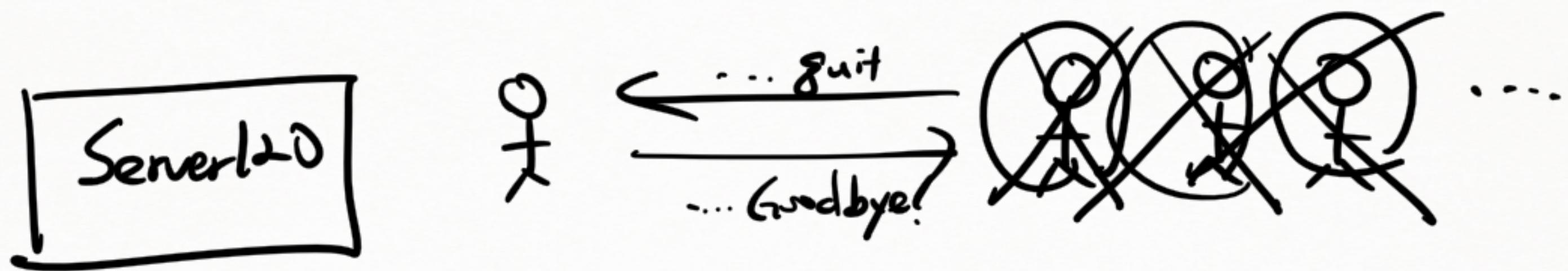
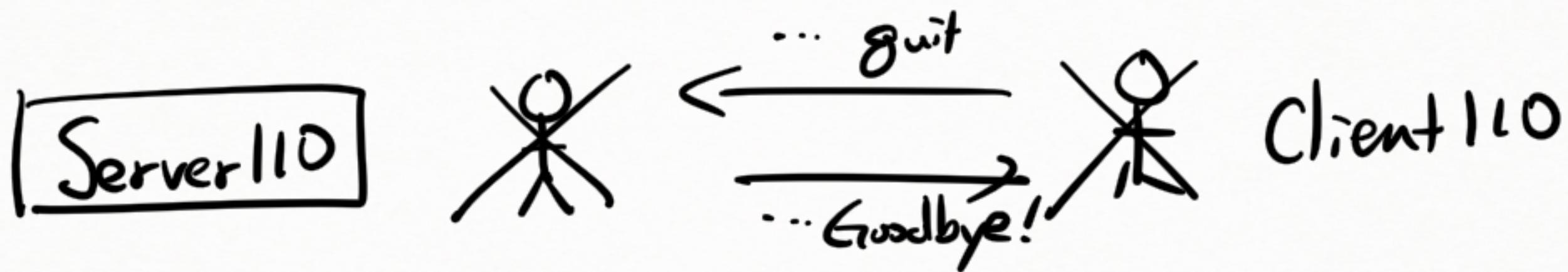
- ✓ Connectionful
  - TCP
    - ↳ Socket
  - ↳ Stateful
  - ↳ Stateless
- ✓ Connectionless
  - UDP
    - ↳ DatagramSocket

Client ↔

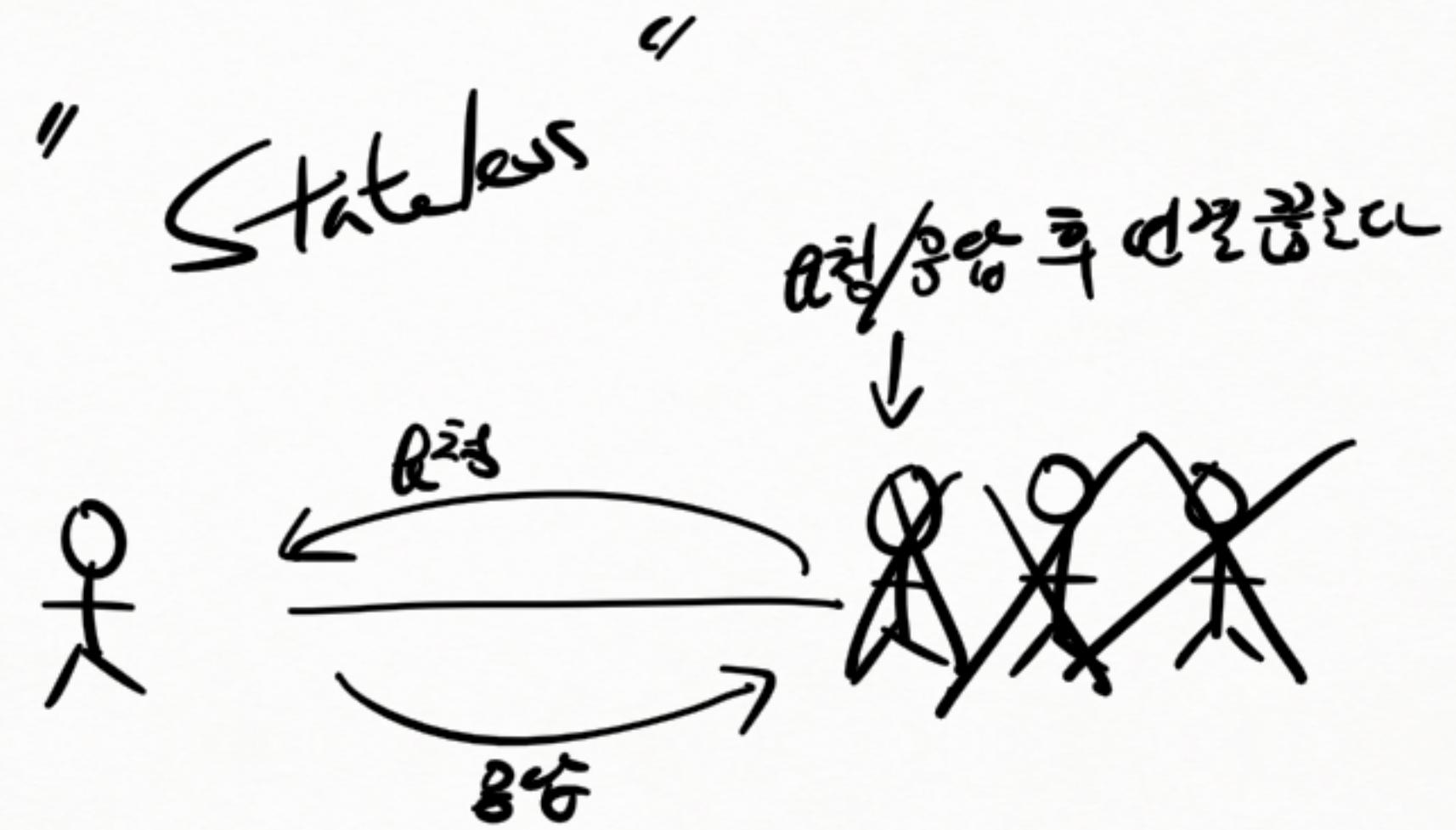
Server ↔



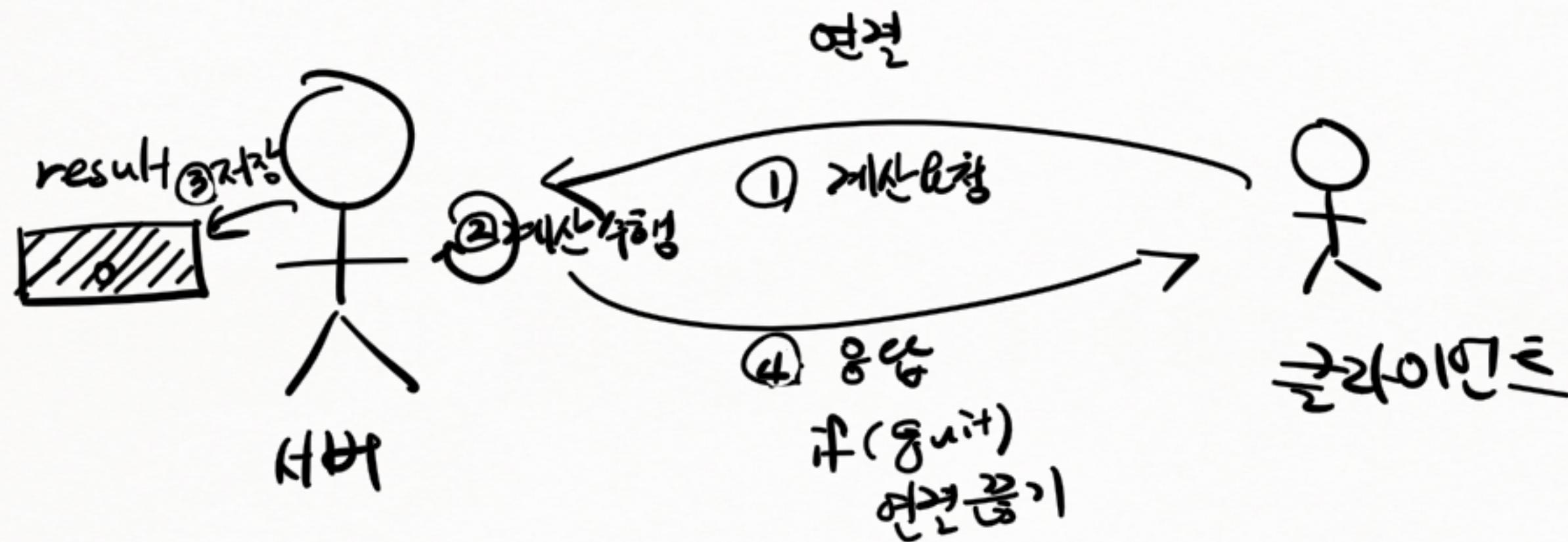
한 번 연결한 후  
여러번 Data 주신  
↓  
"Stateful"  
통신



Server 210



Stateful  
계산기 기능



# Stateless 계산기 서버

고객명단

| clientID | 결과  |
|----------|-----|
| 217      | 82  |
| IP       | 연산자 |
| 201      |     |
| 2222     | 15  |
| 100      |     |
|          |     |

⑤ 결과값을 저장

result  
[80]

새로운 아이디 ID 생성  
(예: 2222)



② 고객의 결과를 반환

① 연결 + 읽기

[long] [UTF] [int]

client ID

연산자



→ 217은

217 - 20

readLong()

readUTF()

readInt()

readLong()

readInt()

④ 결과값은 ID + > 연산결과

[2222 80]

الله  
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

吳

은행원

|       |         |
|-------|---------|
| କୋଡ଼ି | ଟୁମାଣ୍ଡ |
| ୧୧    | ୫୦୦୦    |

은행 전산 시스템

111

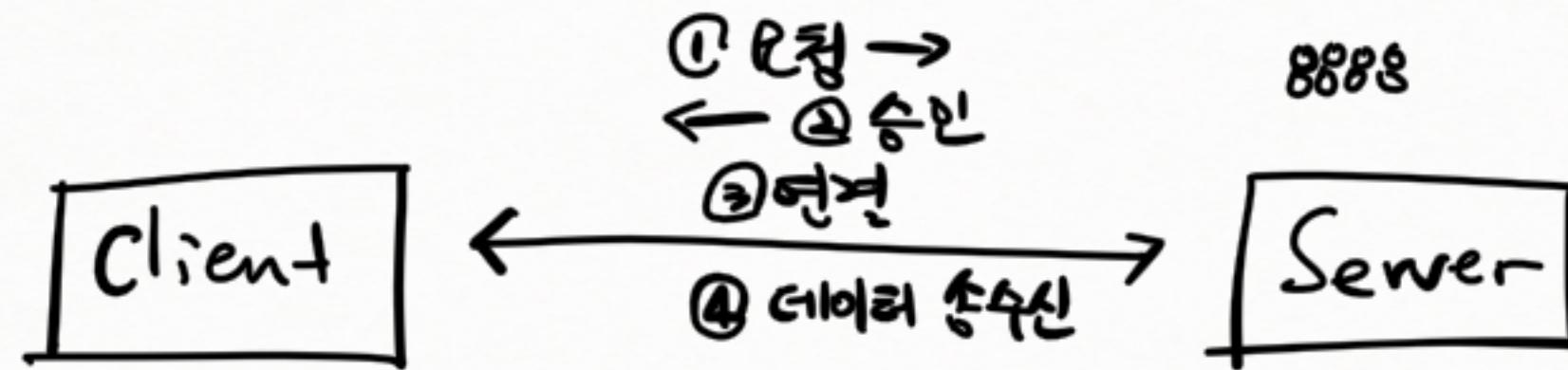
⊕  
300

$$\textcircled{1} \quad \underline{\text{자연}} + \underline{\text{인}}$$

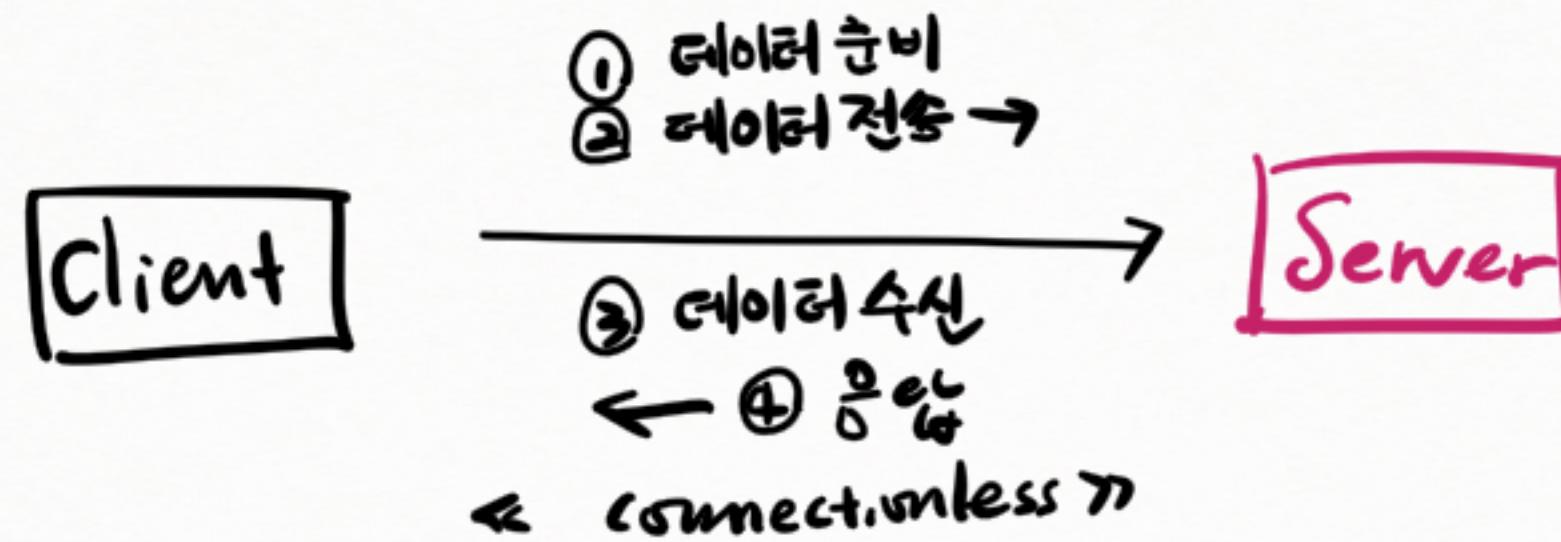
9

۲۷

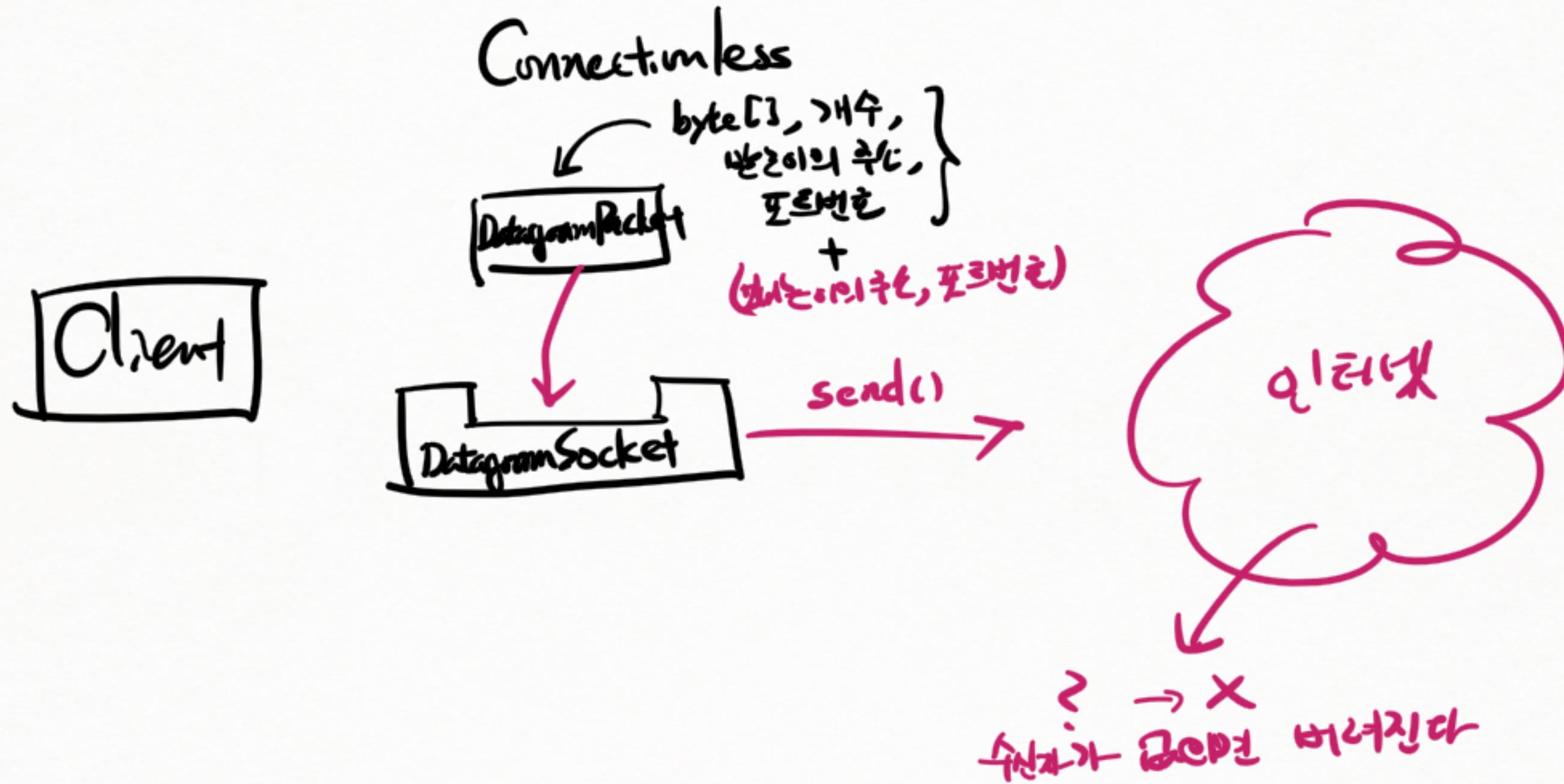
'Stateless' gks!

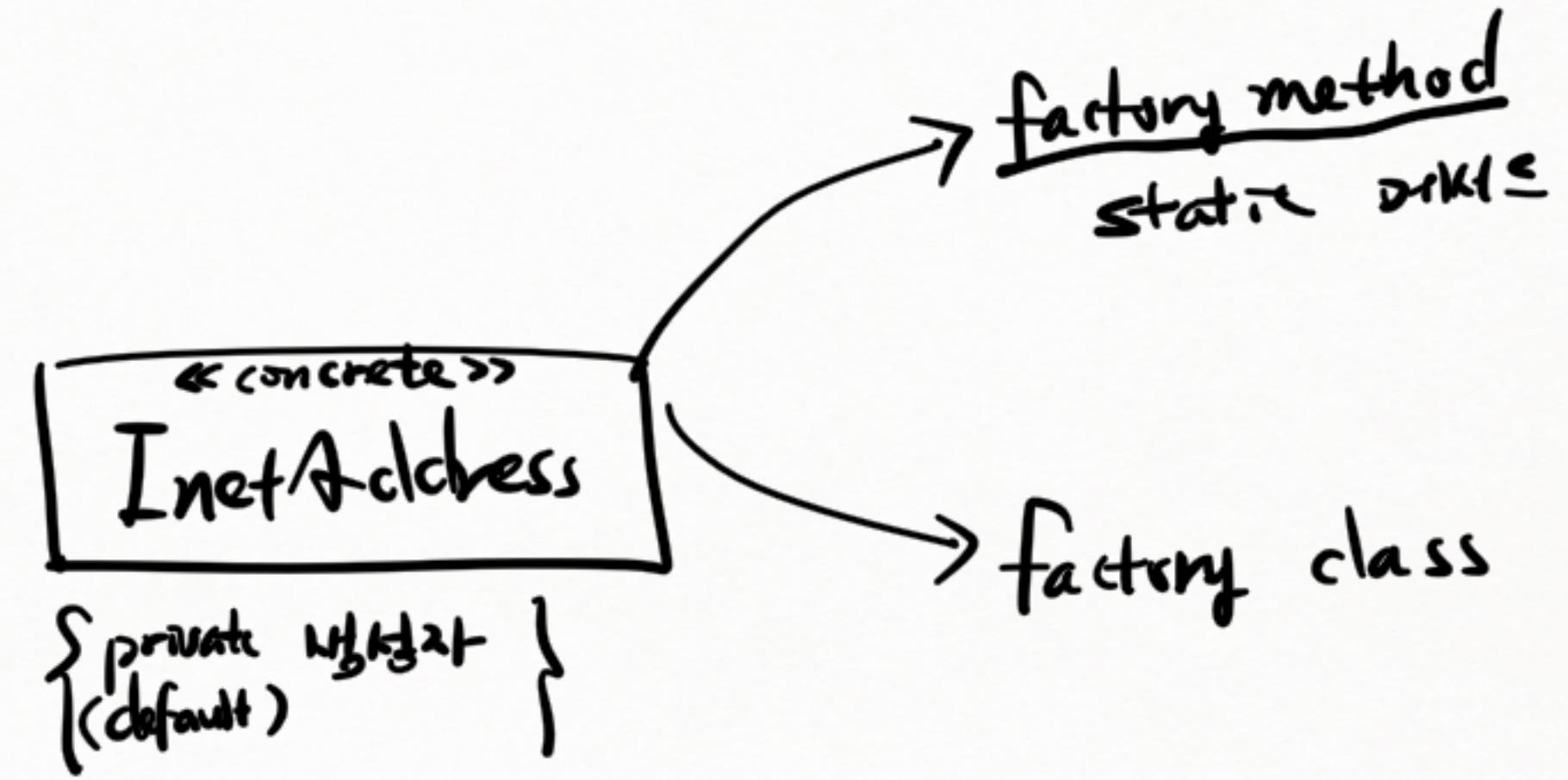


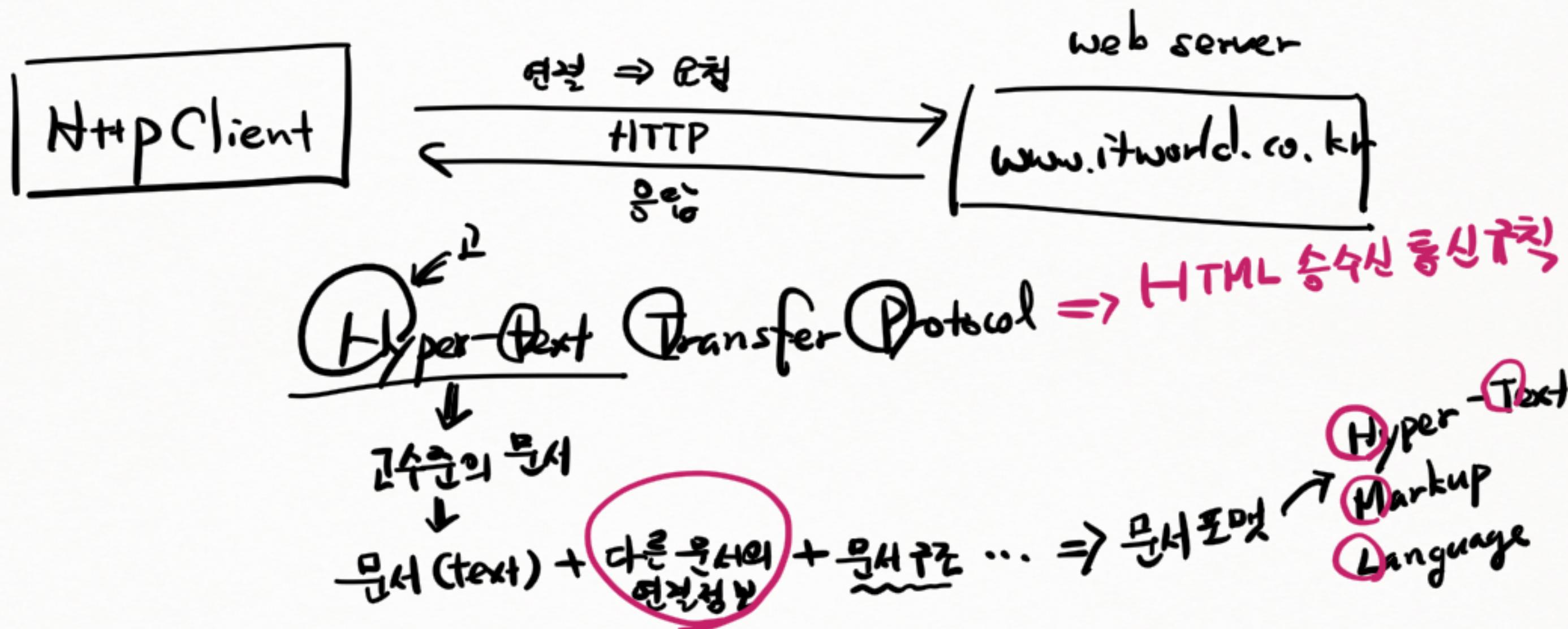
« connection-oriented »



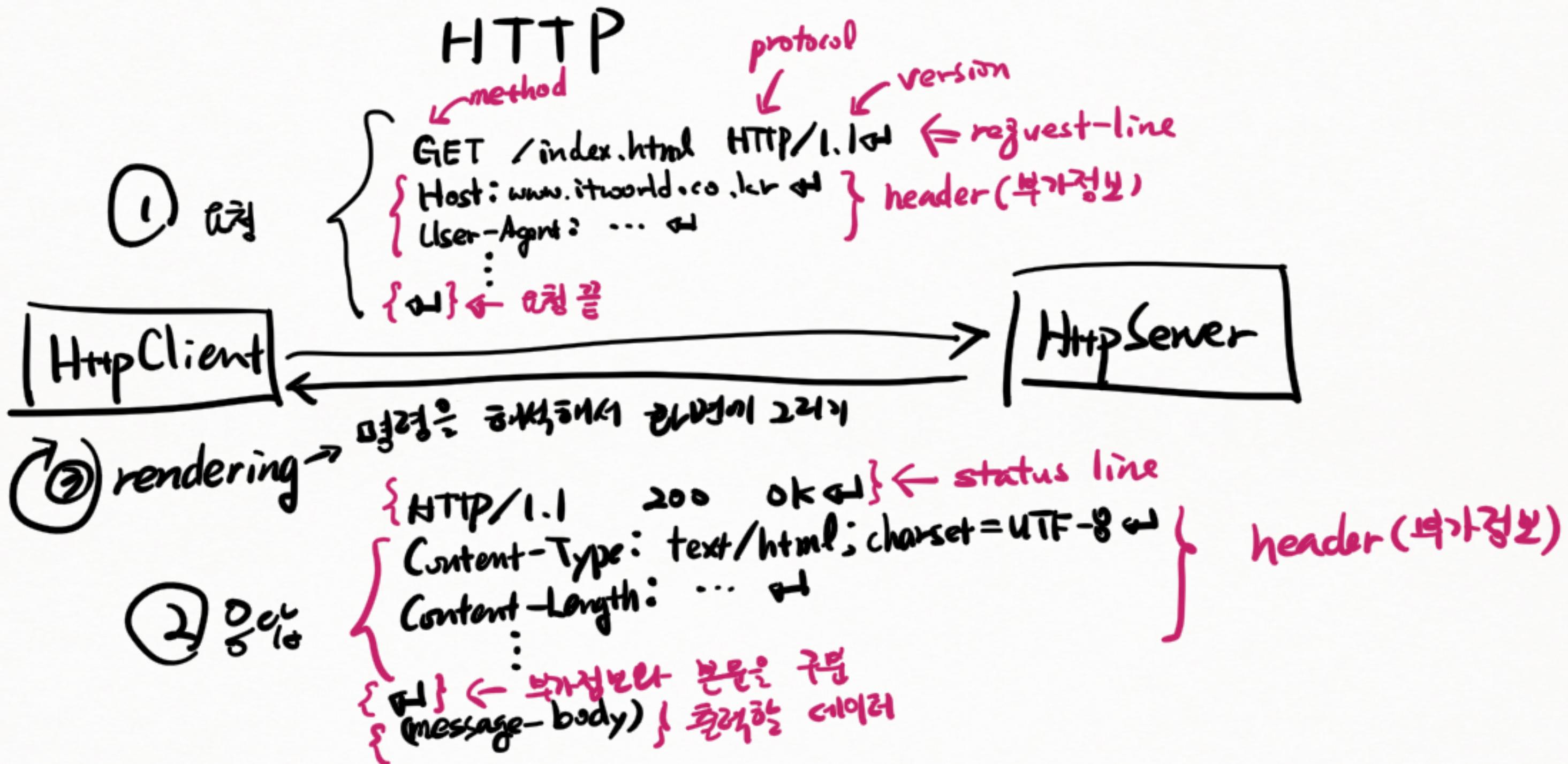
« connectionless »

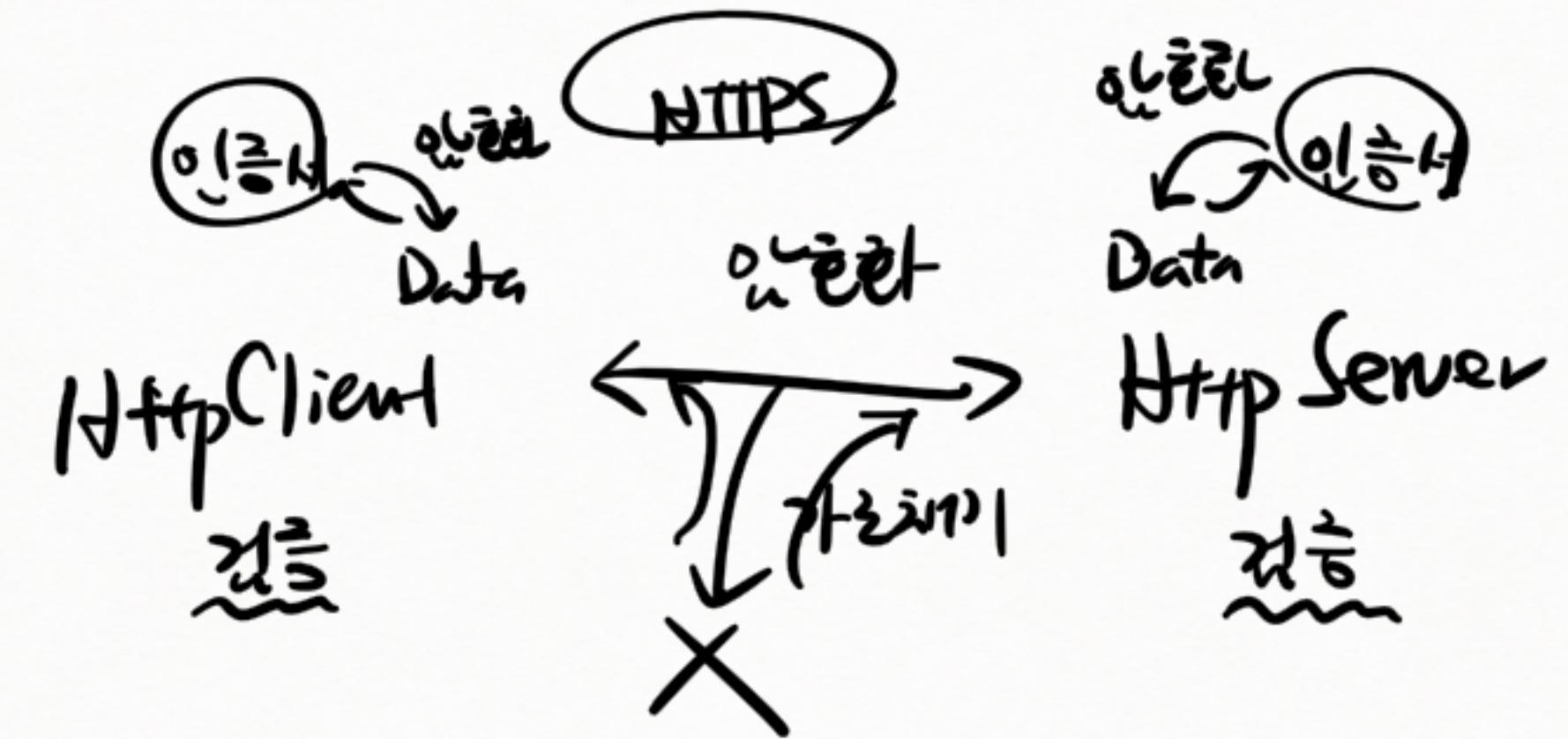






⇒ RFC-2616 명세서







인터넷 상의 자원 위치를 표시하는 방법

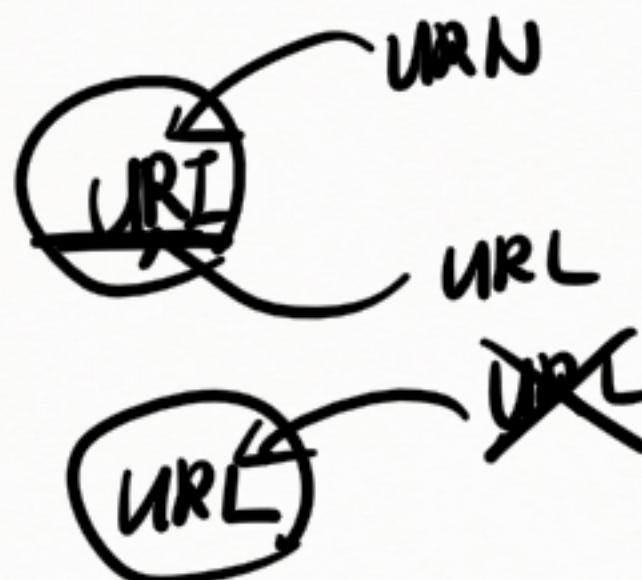


Uniform Resource Identifier (URI)

URL (Uniform Resource Locator)

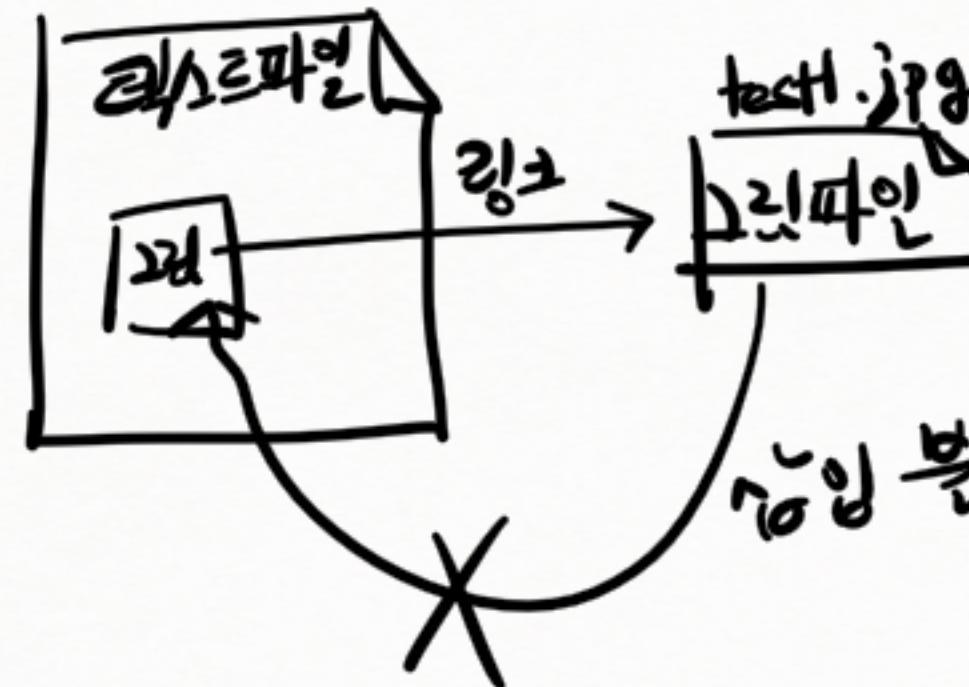
↳ 프로토콜://서버주소:포트/경로  
↳ (http://haha.com/a/b/x.gif)

URN (Uniform Resource Name)



# Base64

test1.html



binary → text  
Why?  
↳ binary data는 텍스트로 데이터를 표현하기 때문.

test2.html

