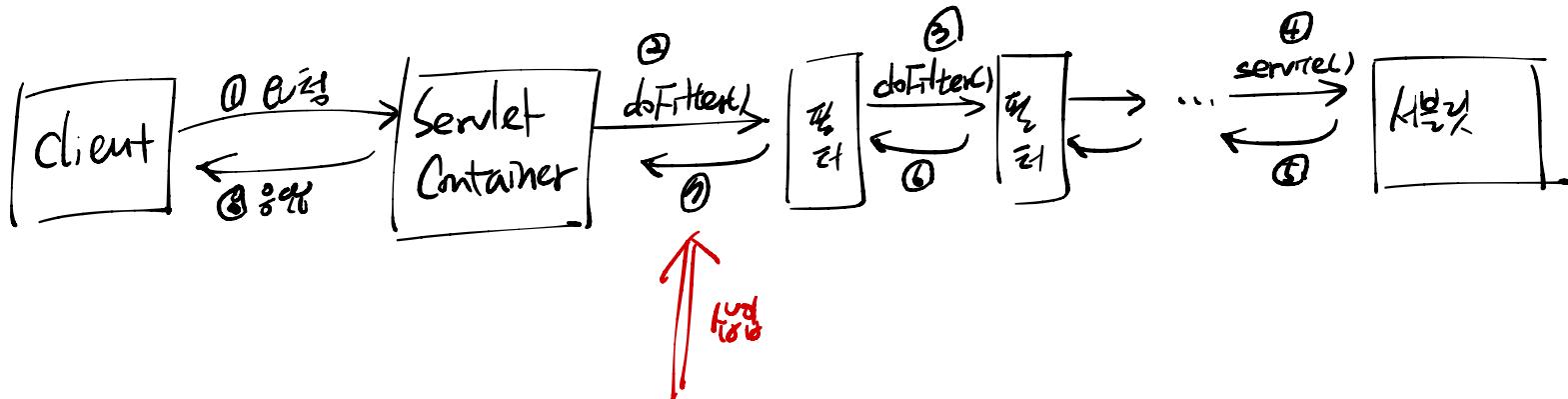


# 11. Spring Security 2회

\* 31C



1. 컨테이너의  
필터는 허용하지  
않는다

⇒ **<<Filter>>**  
**DelegatingFilterProxy**  
**org.springframework.web.filter**

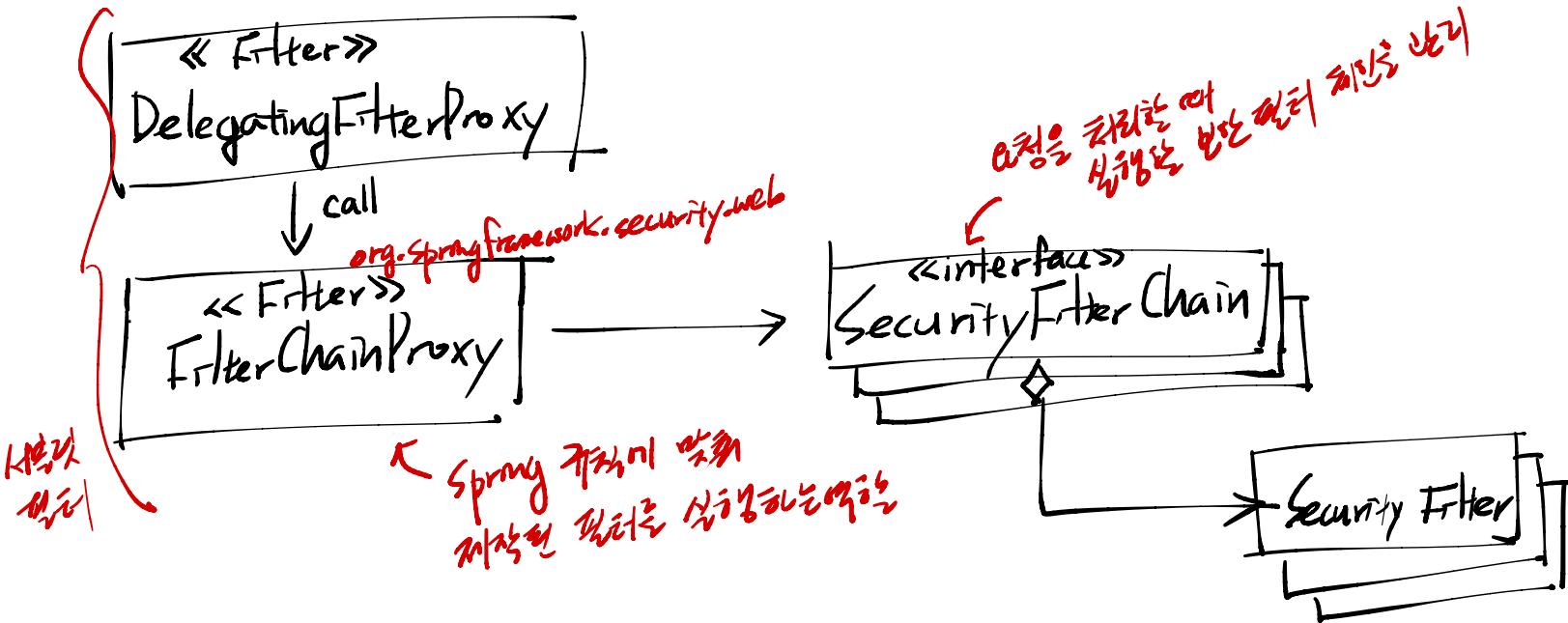
2. 컨테이너의  
필터는 허용

**스프링  
필터체인**

- security
- Batch
- Data
- cloud
- =

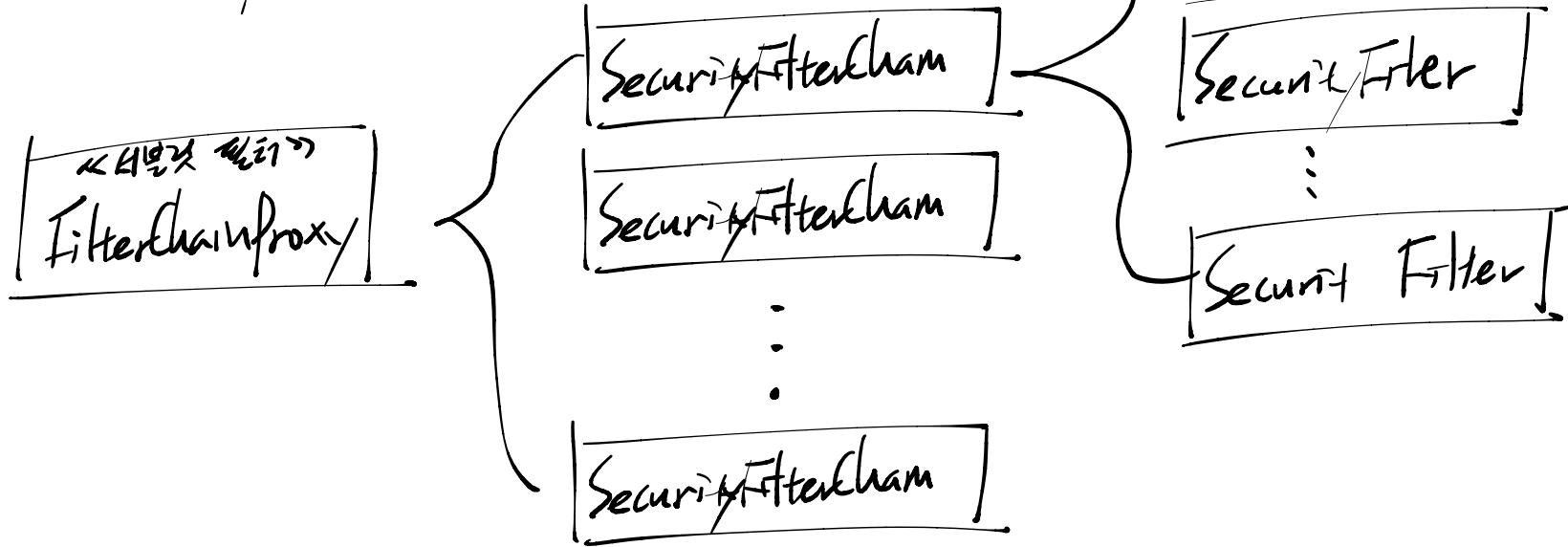
# 11. Spring Security 구조

\* FilterChainProxy ← Spring Security 프로토콜의 핵심 구조.



## 11. Spring Security 介绍

- \* SecurityFilterChain ← 逻辑过滤链



\* HttpSecurity ← Security Filter 12.23 도입

httpSecurity.authorizeHttpRequest( )

요청에 대해 실행 권한을  
가지면 처리도록

수행 권한 설정

권한 설정

권한 설정하는 →  
권한 있는 경우

(authorize) → ?

권한 있는 경우 12.23

\* Nginx Mirroring

authorize.anyRequest().authenticated()

↑  
Mirroring

↑  
SSL passthrough

↑  
SSL offloading by Nginx itself

\* UserDetails et UserDetailsService

↑  
32s! Ab82l284Uz  
wz74xm

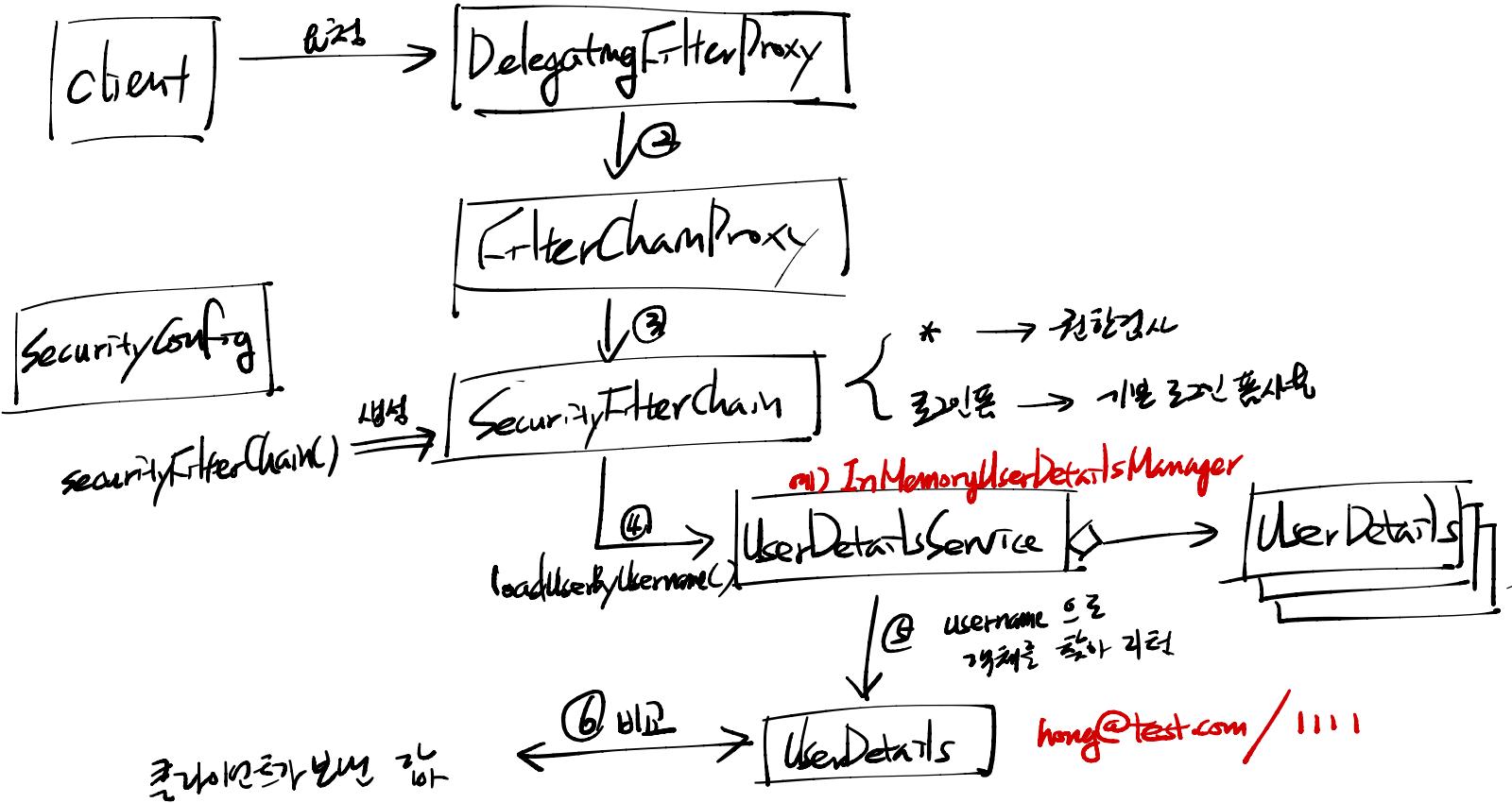
•||.

↑  
ID/pwd z Ab82l284Uz  
76133 74xm

-||.

User → Userservice  
X  
n/a erzeugt Ab82l284Uz

\* 허깅 커스터마이징



\* ↗ ↘ ↗ ↘ ↗

① classic way

→ ② modern way

```
class Car {  
    void setSunRoof();  
    void setNavigation();  
    void setBlackBox();  
    void coloring();  
    void printLabel();  
};
```

```
class Car {  
    void setSunRoof();  
    void setNavigation();  
    void setBlackBox();  
    void coloring();  
    void printLabel();  
};
```

- setSunRoof()
- setNavigation()
- setBlackBox()
- coloring()
- printLabel();

return  
this



Method Chaining

\* သိမ်းသော

