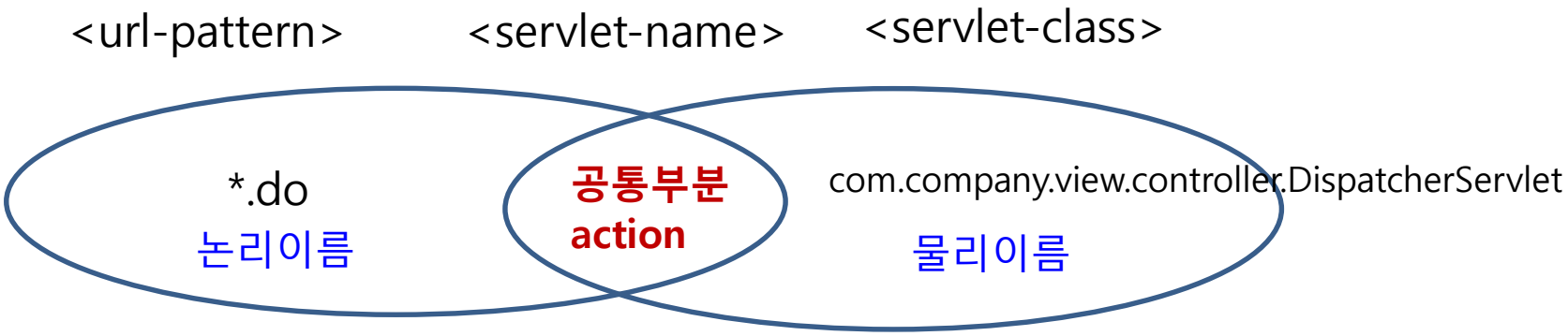
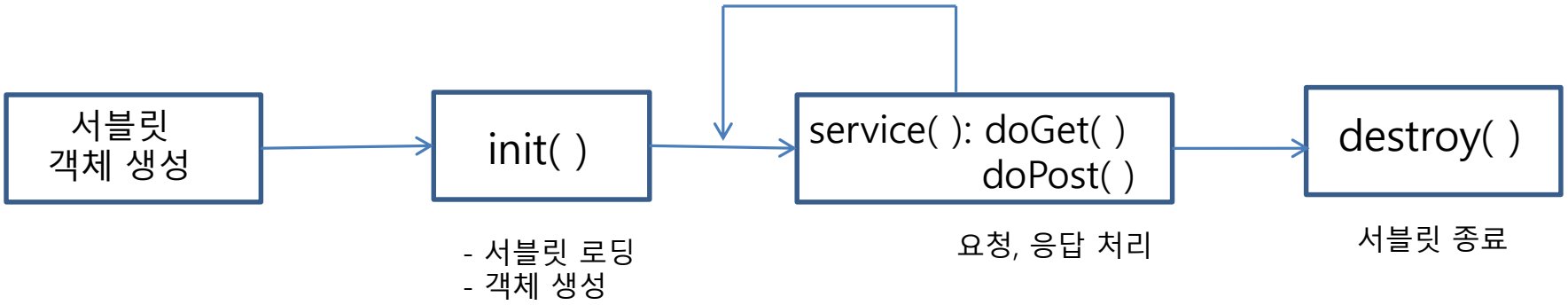
● 회원가입 폼 만들기

회원 기본 정보	
아이디	ID <button>ID중복확인</button>
비밀번호	8~12자리의 영문 대소문자와 숫자, 특수문자로만 입력
비밀번호확인	
이 름	
닉네임	2~10자리의 한글로만 입력
이메일	예)id@domain.com
주민등록번호	- <button>주민번호 정상 유무 체크</button>
핸드폰번호	예)010-1234-5678
성 별	◉남성 ○여성
생년월일	2020 년 1 월 1 일
직 업	직업선택
주 소	우편번호 <button>우편번호 찾기</button> 주소 상세주소(동호수 입력) 참고항목(도로명인 경우: 지역 동이름과 아파트 이름)
<button>회원가입 등록</button> <button>다시쓰기</button>	

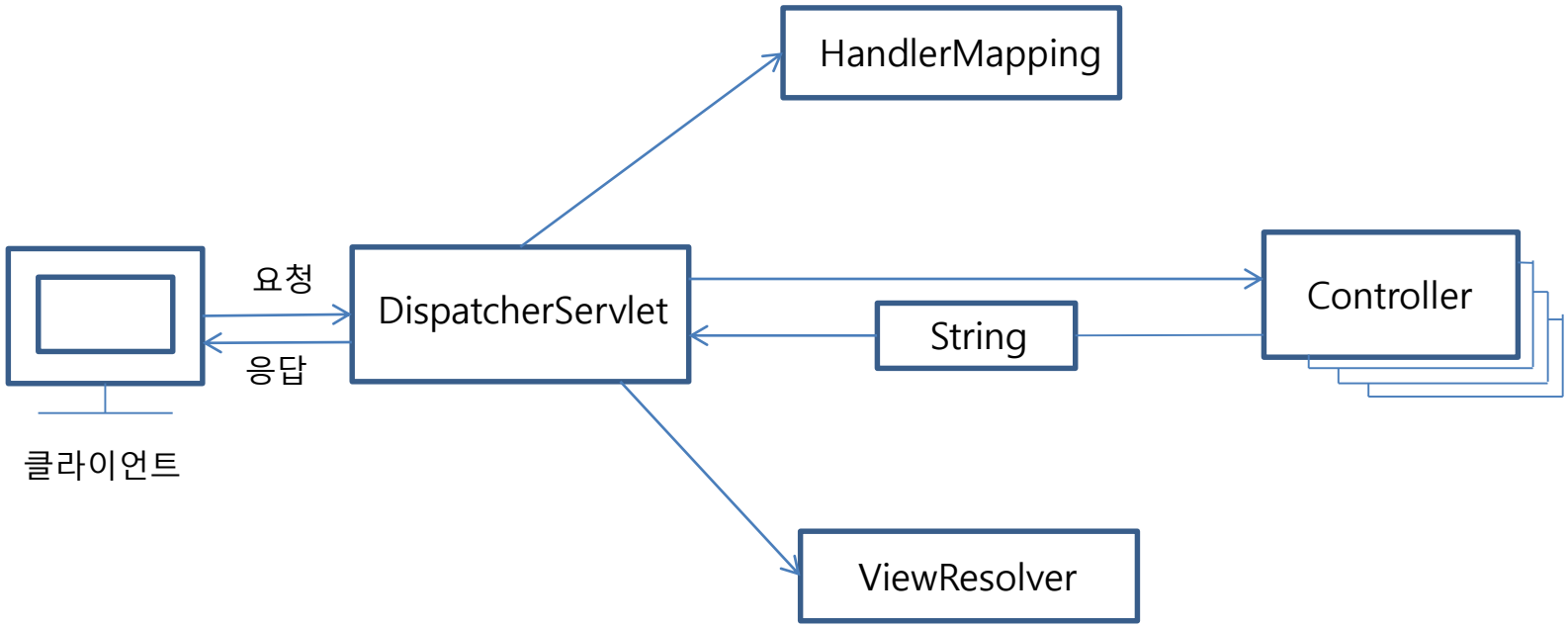
[중요] web.xml 파일의 역할



서블릿의 생명주기


서블릿
객체 생성
init( )
service( ): doGet( )
doPost( )
destroy( )
- 서블릿 로딩
- 객체 생성
요청, 응답 처리
서블릿 종료

MVC 프레임워크 구조의 수행 흐름



왜 이름이 spring일까?

spring이라는 이름을 가진 유래를 보면 **"개발자들의 겨울은 끝났다"**
즉 **봄(spring)이 온다는** 의미로
spring 이라는 이름을 사용하게 되었다고 함.

Whatever happened next, the framework needed a name.
In the book it was referred to as the "Interface21 framework"
(at that point it used com.interface21 package names),
but that was not a name to inspire a community.
Fortunately Yann stepped up with a suggestion: **"Spring"**.
His reasoning was association with nature (having noticed
that I'd trekked to Everest Base Camp in 2000);
and the fact that Spring represented a fresh start after the **"winter"**
of traditional J2EE. We recognized the simplicity and elegance of
this name, and quickly agreed on it.

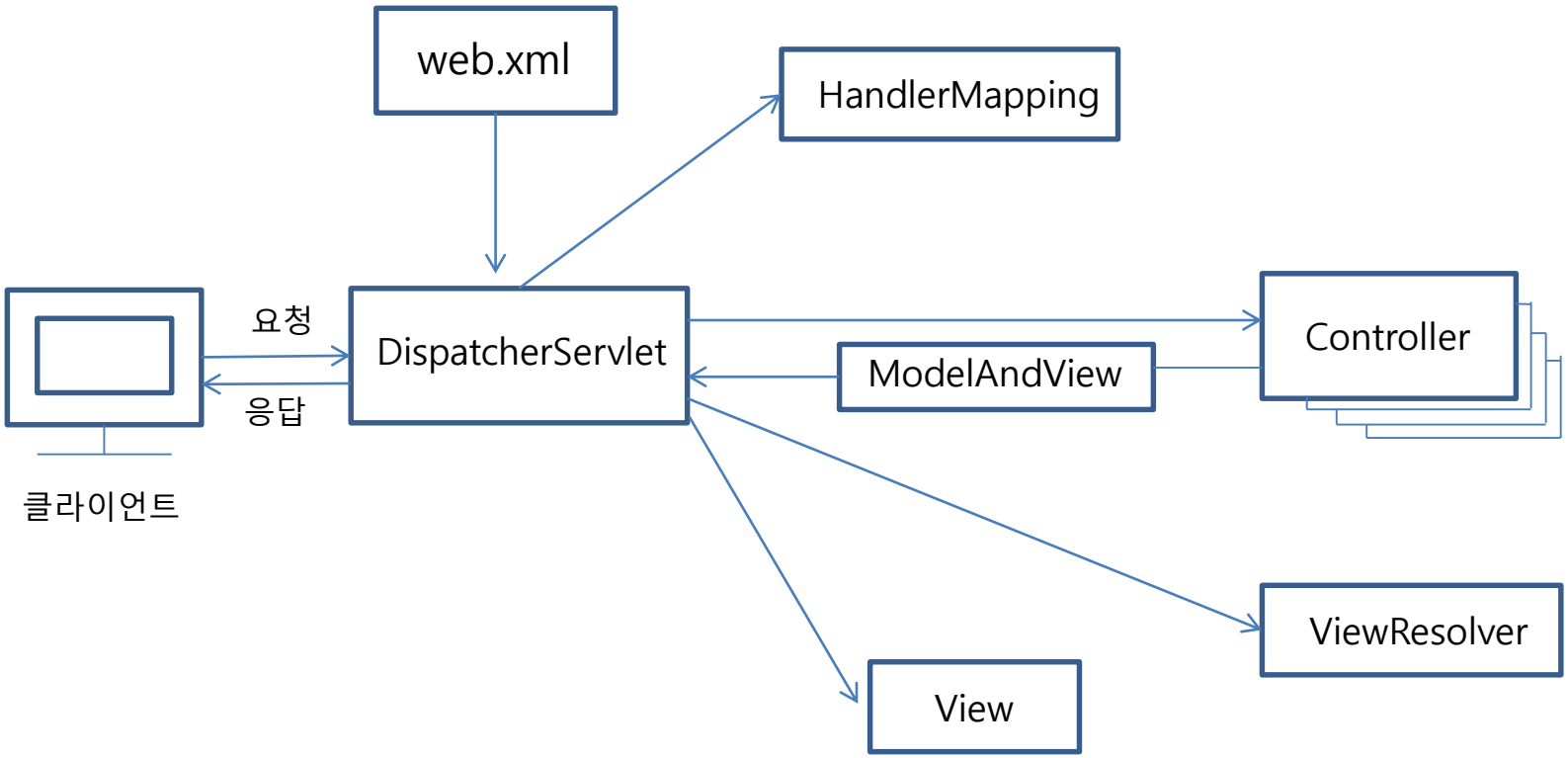
프레임워크를 사용함으로서 얻는 장점

1. 효율적
-> 처음부터 다 만드는 것이 아니라 이미 틀이 잡아져 있는 것을 바탕으로 만들기 때문에 시간과 비용이 절약되며 더 생산적이고 효율적임
2. 퀄리티 및 안정성 향상
-> 어느정도 검증이 이루어진 코드를 사용하는 것이기 대문에 치명적인 오류나 설계상의 문제를 가지고 있을 가능성이 매우 낮음 그렇기 때문에 일정 수준의 퀄리티와 안정성을 보장해줄 수 있음
3. 유지보수에 유리
-> 프레임워크는 보다 코드가 체계적으로 구성되어 있기 때문에 중간에 코드를 작성하는 사람이 바뀐다고 하더라도 상대적으로 코드를 이해하기 쉽고, 유지보수를 함에 있어서 유리함

● AOP 개념 이해

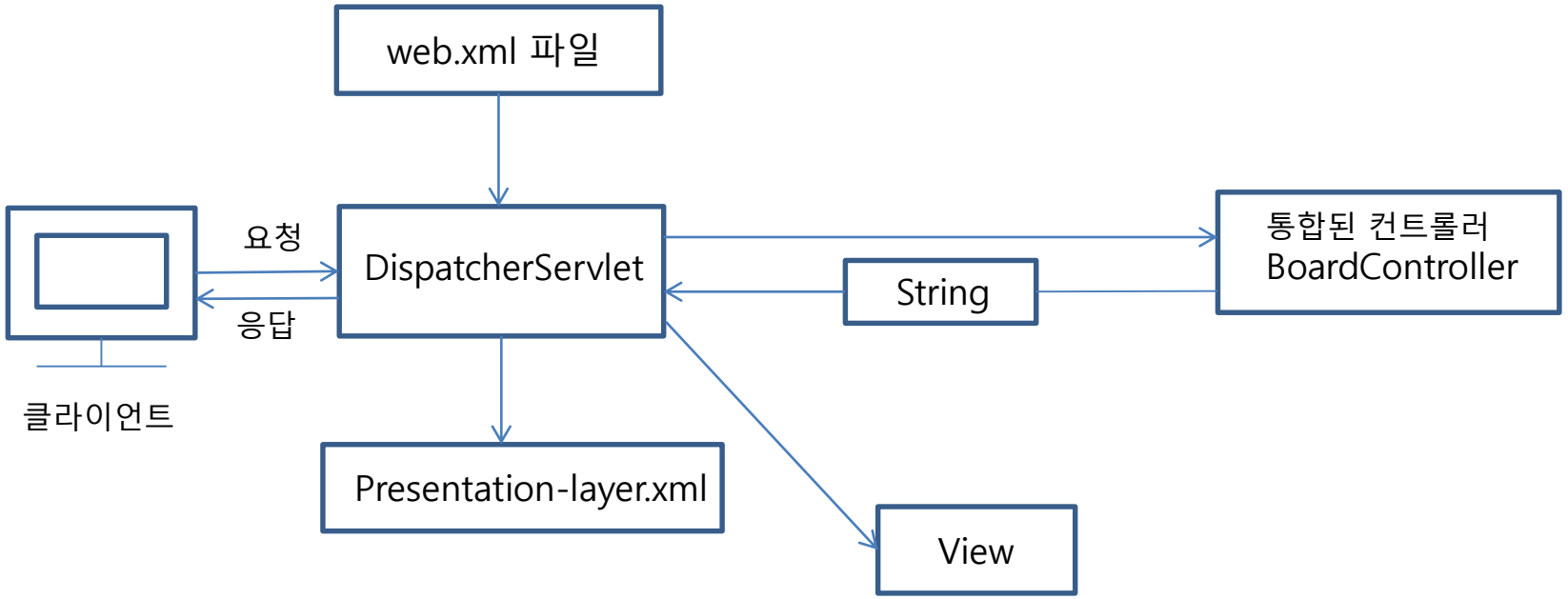
```
public class LogAdvice {
        public void printLog() {
                System.out.println("[공통 로그] 비즈니스 로직 수행 전 동작");
        }
}

<bean id="log" class="com.company.business.common.LogAdvice"></bean>

<!-- AOP 개념을 적용하기 위해서 추가 -->
<bean id="log" class="com.company.business.common.LogAdvice"></bean>

<aop:config>
        <aop:pointcut id="allpointcut" expression="execution(* com.company.business..*Impl.*(..))" />

                                ②
        <aop:aspect ref="log">
                                ④                ①                ③
                <aop:before pointcut-ref="allpointcut" method="printLog()"/>
        </aop:aspect>
</aop:config>
```

스프링 MVC 구조의 수행 흐름

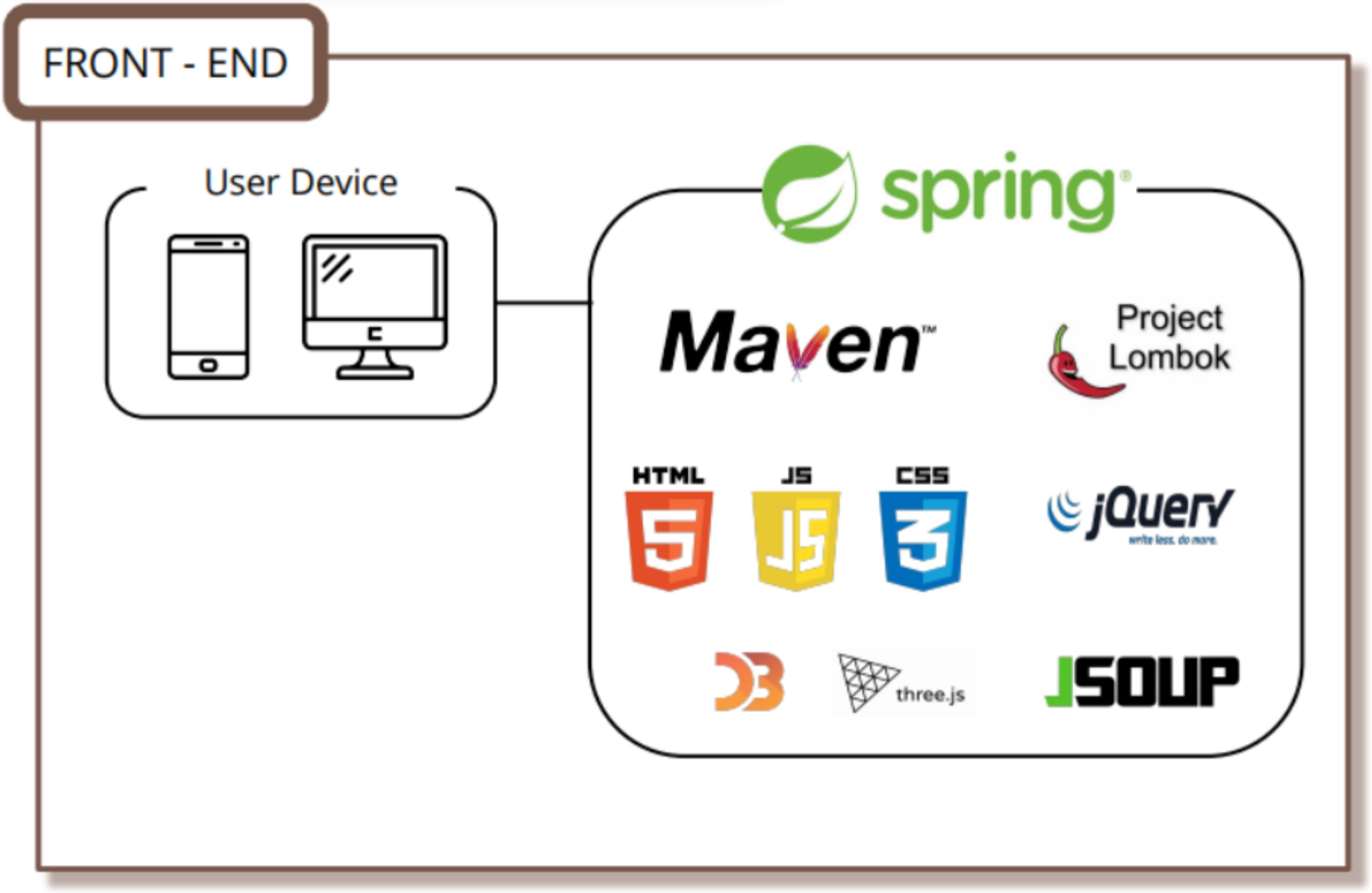


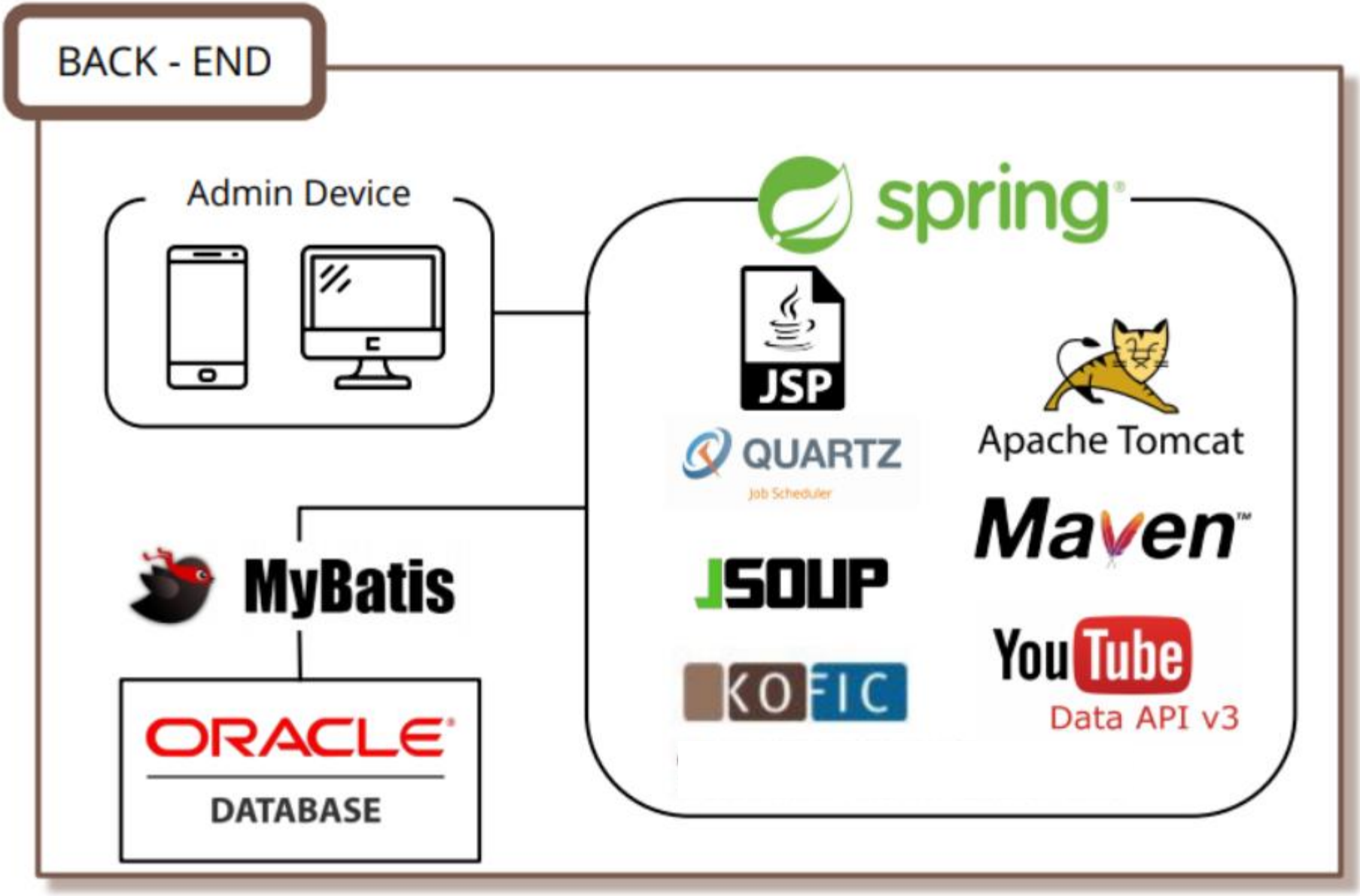
- ModelAndView는 **컴포넌트** 방식으로 ModelAndView 객체를 생성해서 객체 형태로 리턴한다. 값을 넣을 때는 addObject() 메소드를 사용하고, setViewName() 메소드로 보낼곳 View를 세팅한다.

스프링 어노테이션 기반 구조의 수행 흐름



- Model은 **매개변수(Prameter)** 방식으로 해당 메소드에(Model model) 파라미터를 넣어주고 String 형태로 리턴한다. Model은 값을 넣을 때 addArribute() 메소드를 사용한다.


FRONT - END
User Device
spring
Maven
Project
Lombok
HTML
JS
CSS
jQuery
write less, do more.
three.js
JSOUP


BACK - END
Admin Device
spring
JSP
Apache Tomcat
QUARTZ
Job Scheduler
Maven
MyBatis
JSOUP
KOFIC
YouTube
Data API v3
ORACLE
DATABASE