JSP 프로그래밍

(재)대덕인재개발원

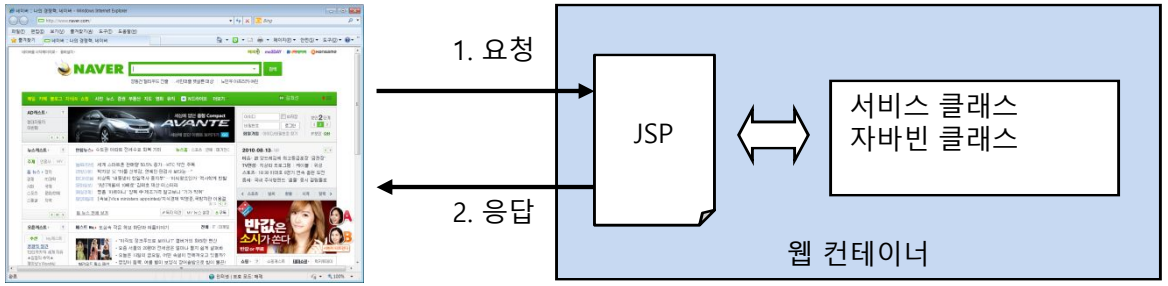
23 MVC 패턴 구현

- 1. 모델 2 구조와 MVC 패턴
- 2. 모델 2 구조를 이용한 MVC 패턴 구현
- 3. 모델 1 구조와 모델 2 구조의 선택

23.1 모델 2 구조와 MVC 패턴

23.1.1 모델 1 구조

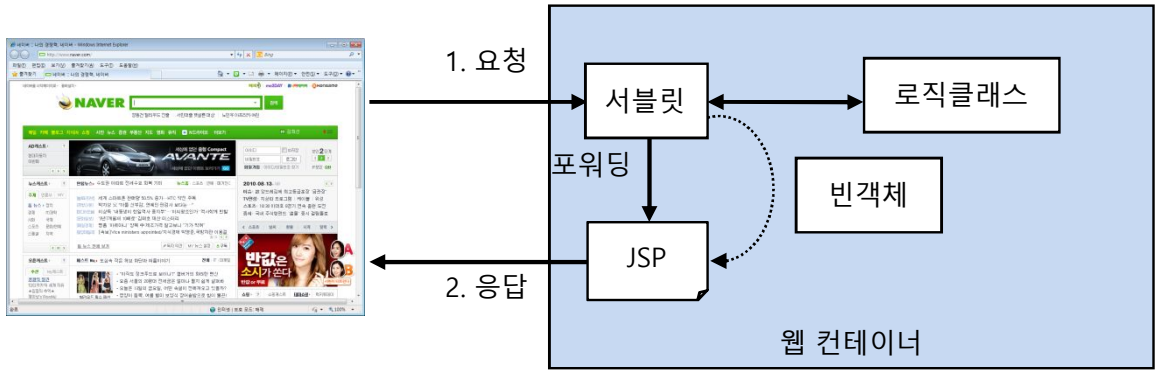
- 모델 I 구조
  - 모델 I 구조는 JSP를 이용한 단순한 모델이다.
  - 웹 브라우저의 요청이 곧바로 JSP에 전달된다.
  - 웹 브라우저의 요청을 받은 JSP는 자바빈이나 서비스 클래스를 사용해서 웹 브라우저가 요청한 작업을 처리하고 그 결과를 클라이언트에 출력해 준다.



[그림] 모델 I 구조의 요청/응답 처리 방식

23.1.2 모델 2 구조

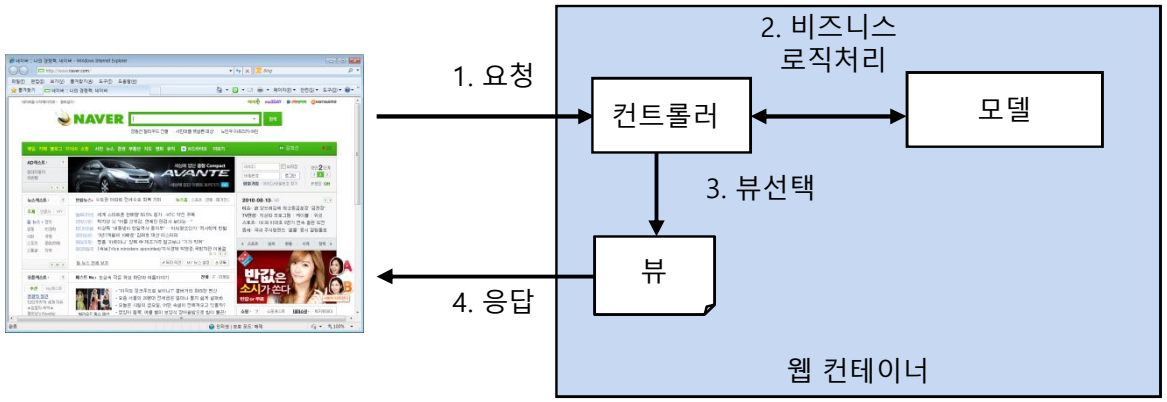
- 모델 2 구조
  - 모델 2 구조는 모델 1 구조와 달리 웹 브라우저의 요청을 하나의 서블릿이 받게 된다.
  - 서블릿은 웹 브라우저의 요청을 알맞게 처리한 후 그 결과를 보여줄 JSP 페이지로 포워딩한다.
  - 포워딩을 통해서 요청 흐름을 받은 JSP 페이지는 결과 화면을 클라이언트에 전송한다. 즉, 서블릿이 비즈니스 로직 부분을 처리하게 되는 것이다.



[그림] 모델 2 구조의 요청/응답 처리 방식

23.1.3 MVC 패턴

- MVC 패턴
  - MVC(Model-View-Controller) 패턴은 크게 모델, 뷰, 컨트롤러의 세 부분으로 구성되면, 각각의 요소는 다음과 같은 역할을 담당한다.
  - - 모델 : 비즈니스 영역의 상태 정보를 처리한다.
  - - 뷰 : 비즈니스 영역에 대한 프리젠테이션 뷰(즉, 사용자가 보게 될 결과 화면)를 담당한다.
  - - 컨트롤러 : 사용자의 입력 및 흐름 제어를 담당한다.



[그림] MVC 패턴의 구조

23.1.4 MVC 패턴과 모델 2 구조의 매핑

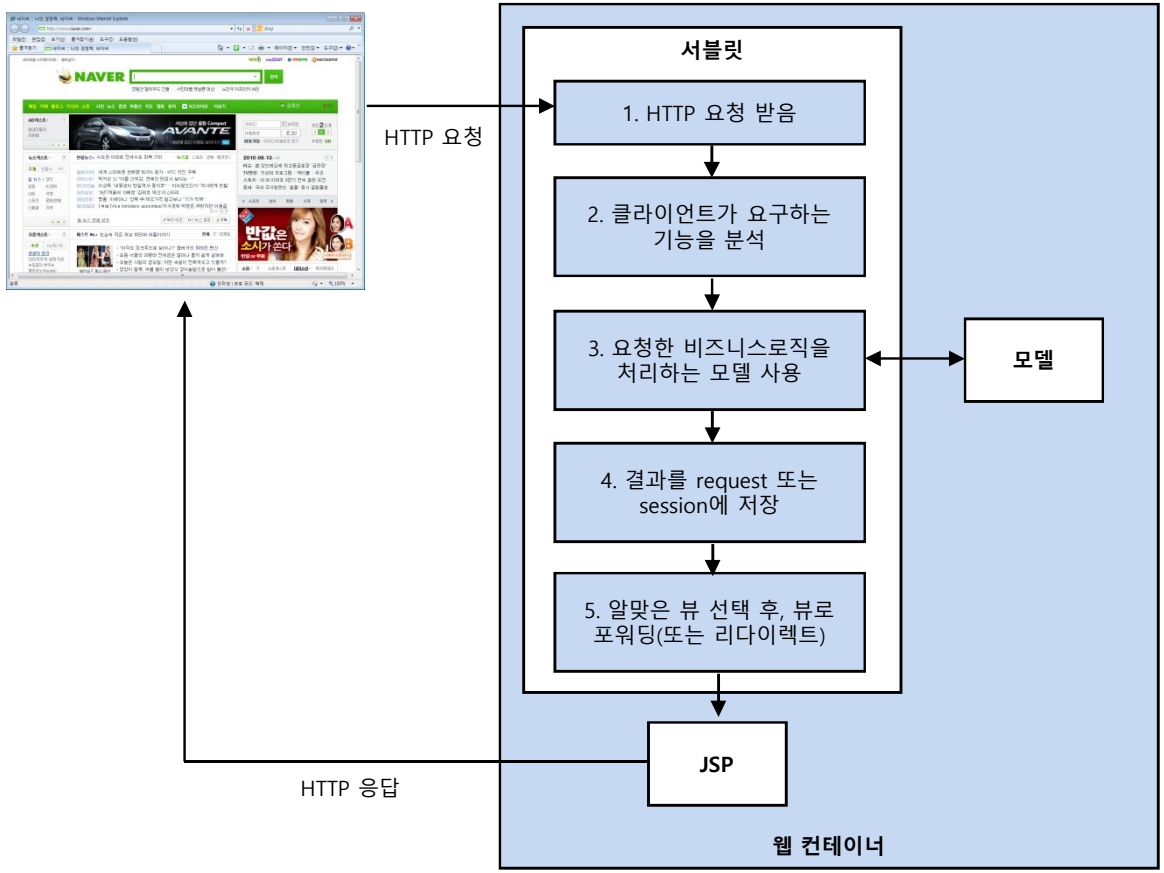
- JSP의 모델 2 구조와 MVC 패턴은 완벽하게 일치한다. 이 둘 사이의 매칭 관계를 살펴보면 다음과 같다.

- - 컨트롤러 = 서블릿
- - 모델 = 비즈니스 로직 처리 클래스, 자바빈
- - 뷰 = JSP
-
- 모델 2 구조에서 웹 브라우저의 요청은 서블릿으로 전달되고, 서블릿은 비즈니스 로직을 수행하는 클래스를 사용하여 웹 브라우저의 요청을 처리하며 뷰의 역할을 하는 JSP 페이지를 통해서 처리 결과를 보여주게 된다.

23.1.5 MVC의 컨트롤러

- MVC의 컨트롤러 : 서블릿
  - 모델 2 구조에서 서블릿은 MVC 패턴의 컨트롤러 역할을 한다.
  - 서블릿은 웹 브라우저의 요청과 웹 어플리케이션의 전체적인 흐름을 제어하게 된다.
  - 그림은 컨트롤러로서의 서블릿이 어떤 순서로 실행되는지를 보여주고 있다.

23.1.4 MVC 패턴과 모델 2 구조의 매핑 p.66



23.1.5 MVC의 컨트롤러 : 서블릿

- 컨트롤러의 역할을 하는 서블릿은 5단계의 과정을 거쳐서 웹 브라우저의 요청을 처리하게 된다.
  - - 과정 1 : 웹 브라우저가 전송할 HTTP 요청을 받는다. 서블릿의 doGet() 메소드나 doPost() 메서드가 호출된다.
  - - 과정 2 : 웹 브라우저가 어떤 기능을 요청했는지 분석한다. 예를 들어, 게시판 목록을 요청했는지, 글쓰기를 요청했는지 알아낸다.
  - - 과정 3 : 모델을 사용하여 요청한 기능을 수행한다.
  - - 과정 4 : 모델로부터 전달받은 결과물을 request나 session의 setAttribute() 메서드를 사용하여 속성에 저장한다. 이렇게 저장된 결과값은 뷰인 JSP에 사용된다.
  - - 과정 5 : 웹 브라우저에 보여줄 JSP를 선택한 후, JSP로 포워딩한다. 경우에 따라서 리다이렉트를 하기도 한다.

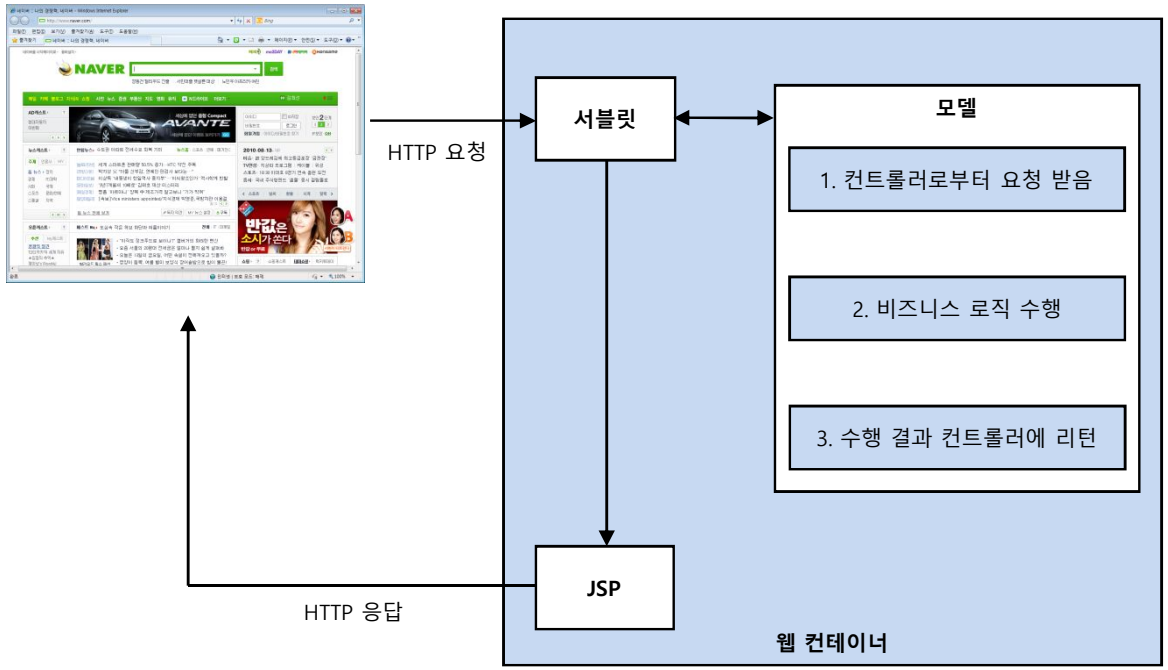
23.1.6 MVC의 뷰 : JSP

- 모델 2 구조에서 JSP는 뷰의 역할을 담당한다.
- 비즈니스 로직과 관련된 코드가 없는 점을 제외하면 일반 JSP 와 거의 동일한 형태를 취한다.
- 차이점이 있다면, 뷰 역할을 하는 JSP는 컨트롤러 부분에서 request 기본 객체나 session 기본 객체에 저장한 데이터를 사용하여 웹 브라우저에 알맞은 결과를 출력해 준다는 점이다.
- 뷰 역할을 하는 JSP는 웹 브라우저가 요청한 결과를 보여주는 프리젠테이션의 역할을 할 뿐만 아니라 웹 브라우저의 요청을 컨트롤러에 전달해 주는 매개체가 되기도 한다.

23.1.7 MVC의 모델

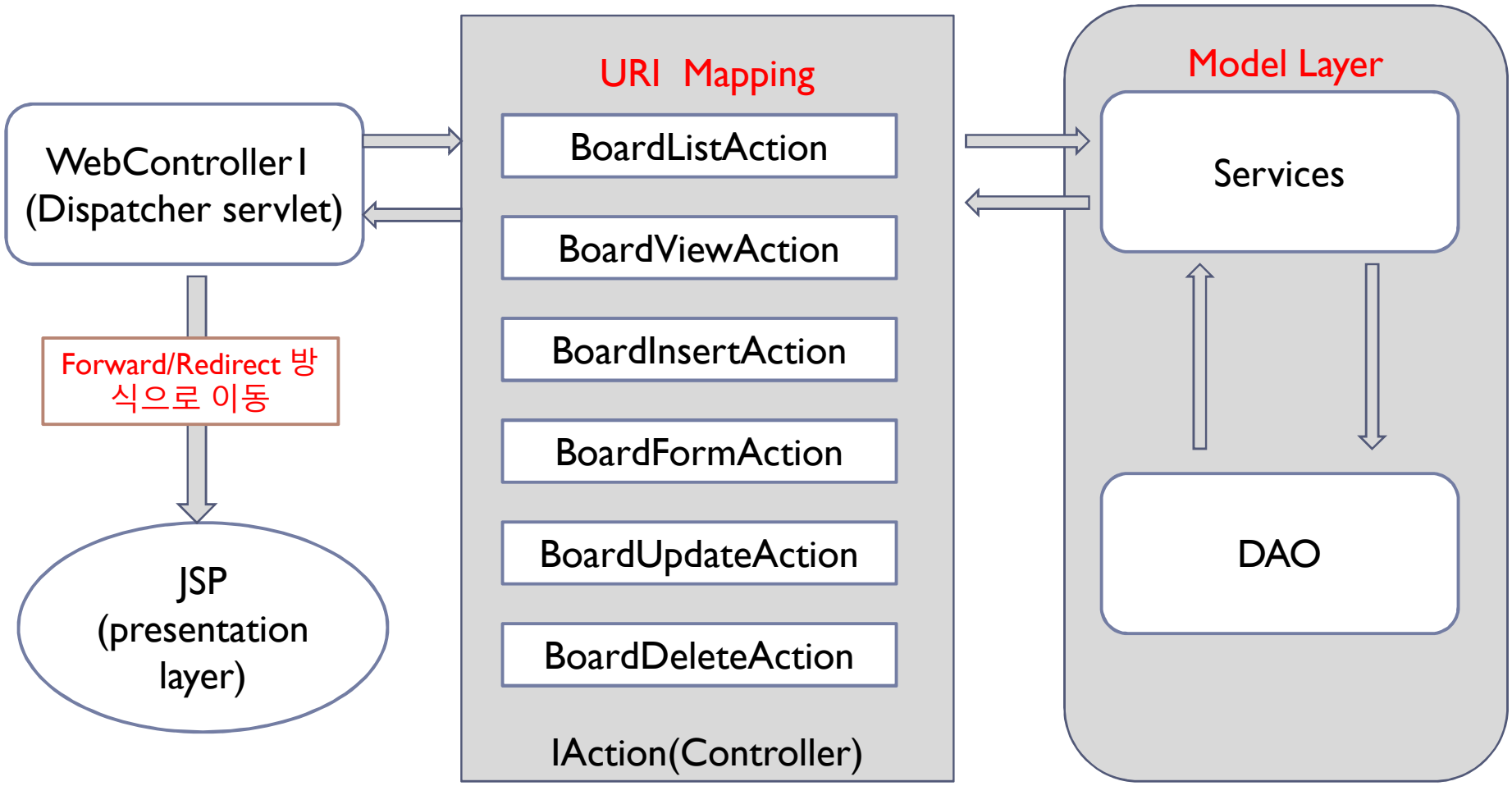
- 컨트롤러는 서블릿을 통해서 구현되고, 뷰는 JSP를 통해서 구현되는 반면에 모델은 명확하게 어떤 것을 통해서 구현된다는 규칙은 없다.
- 비즈니스 로직을 처리해 주면 모델이 될 수 있다.
- 모델이 제공해야 하는 기능은 웹 브라우저의 요청을 처리하는 데 필요한 기능을 제공하는 것이다.

23.1.7 MVC의 모델

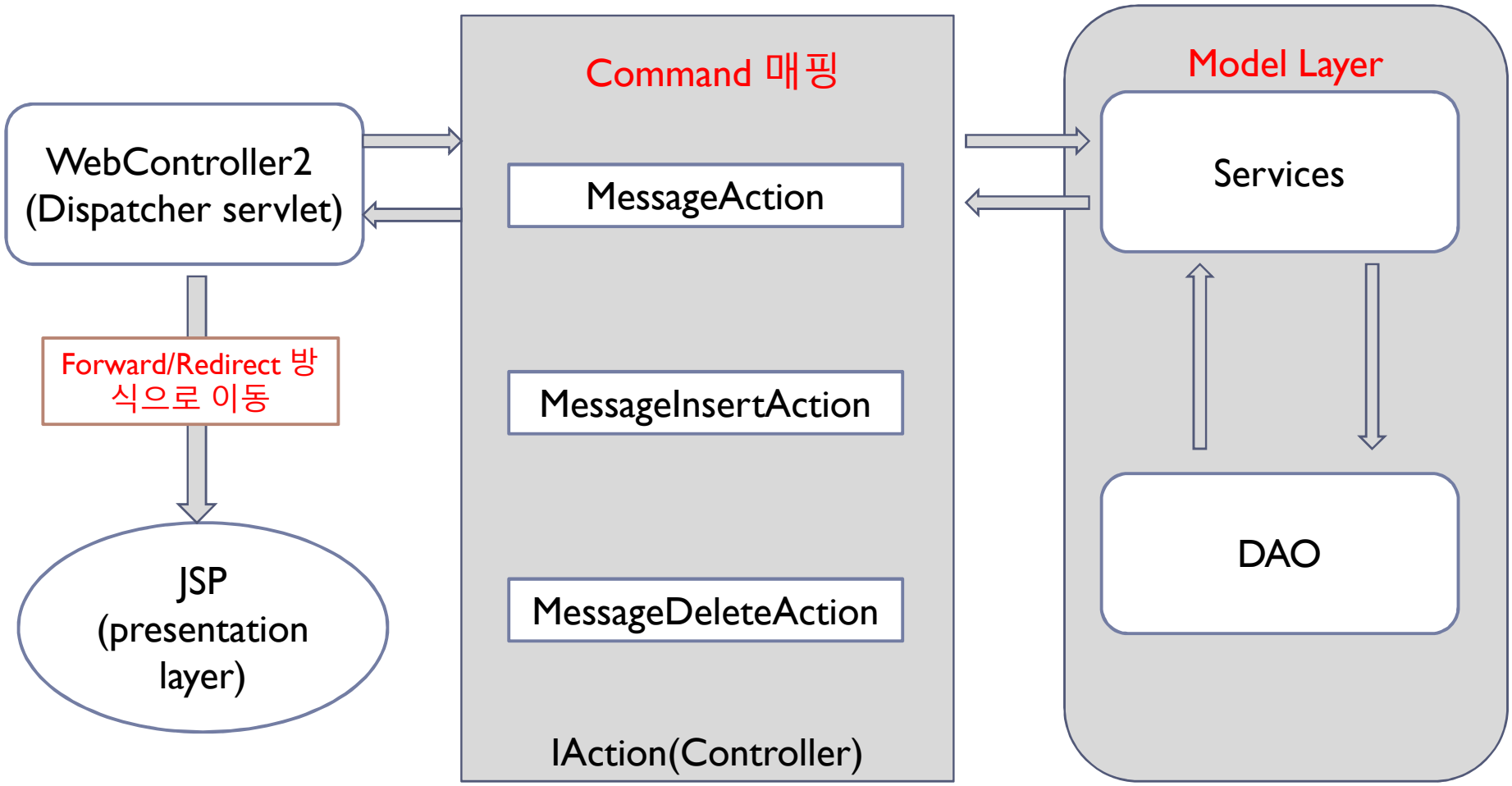


[그림] 컨트롤러 역할을 하는 서블릿과 모델 간의 통신

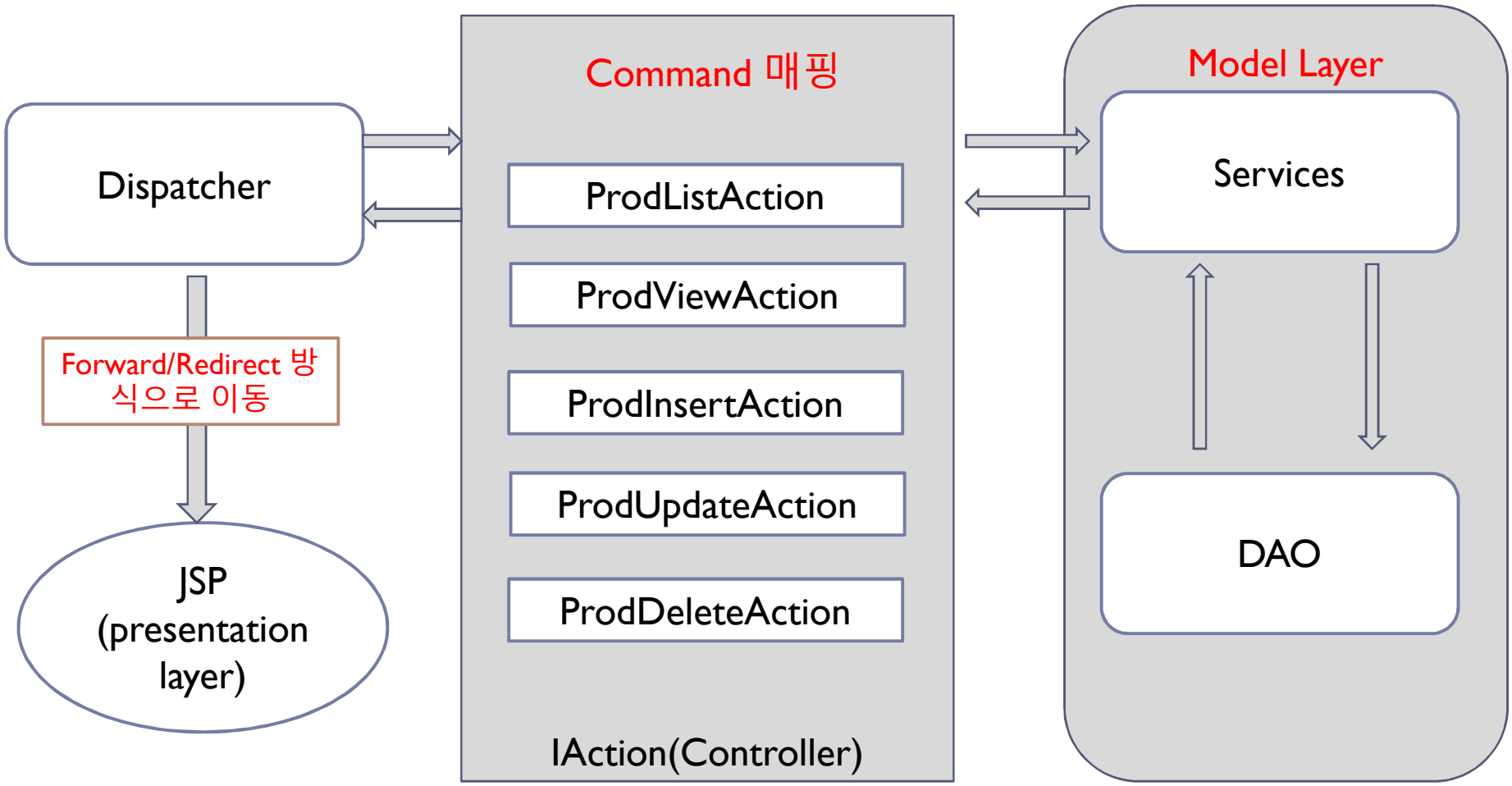
* MVC 모델2 방식의 게시판관리(커맨드패턴) p.67



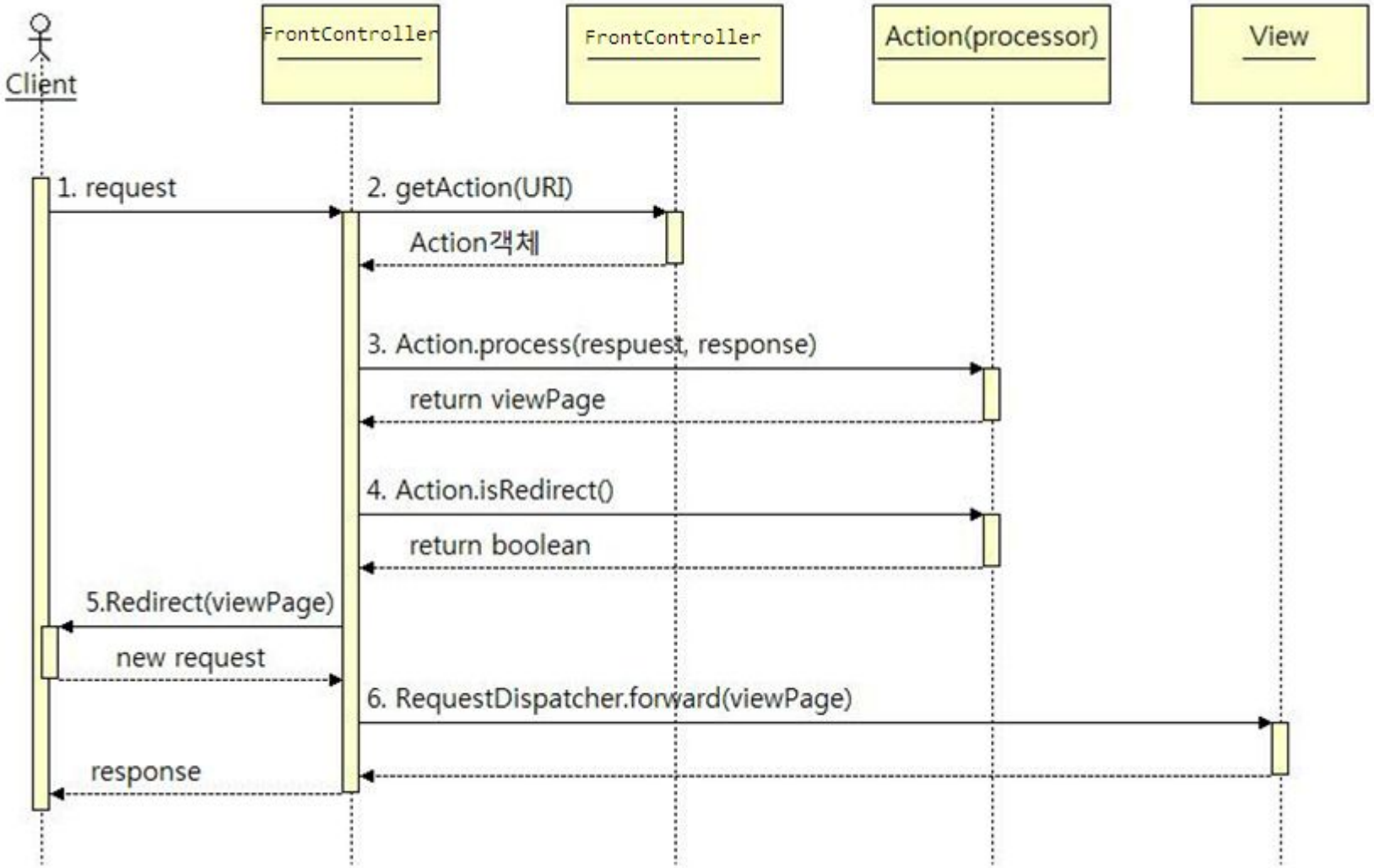
* MVC 모델2 방식의 방명록(커맨드패턴)



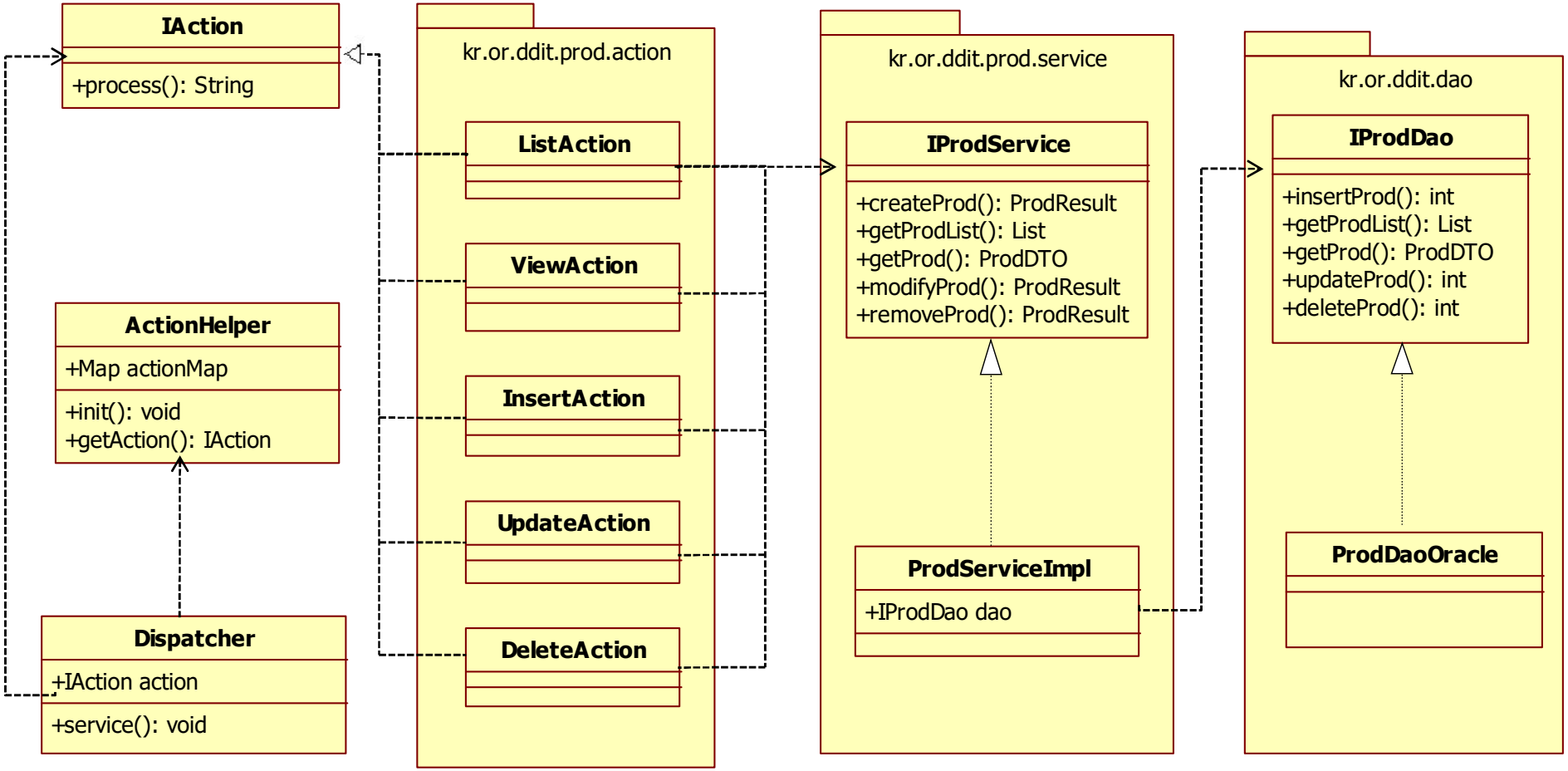
* MVC 모델2 방식의 상품관리(커맨드패턴)



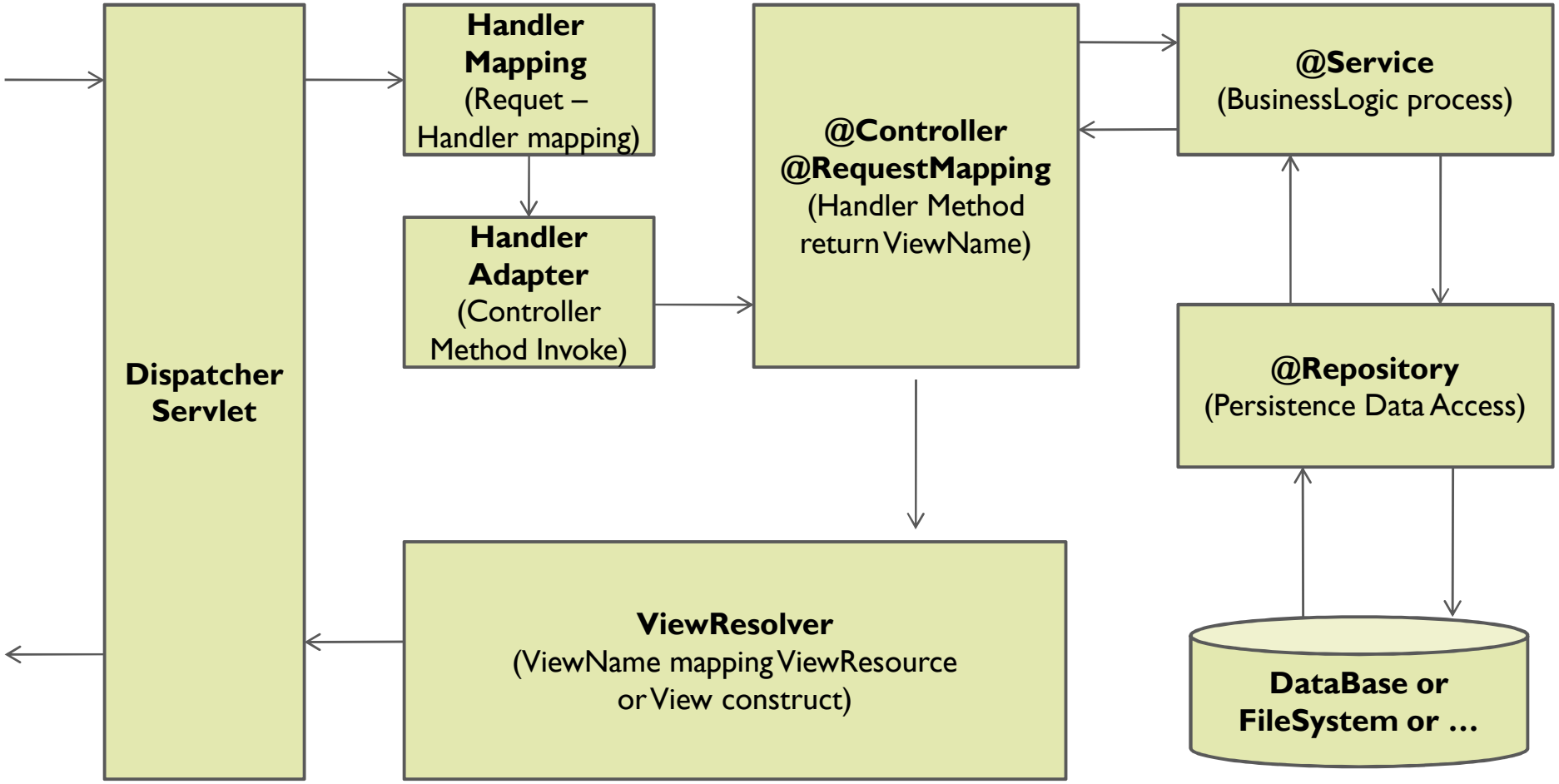
* Request 처리 흐름도(sequence diagram)



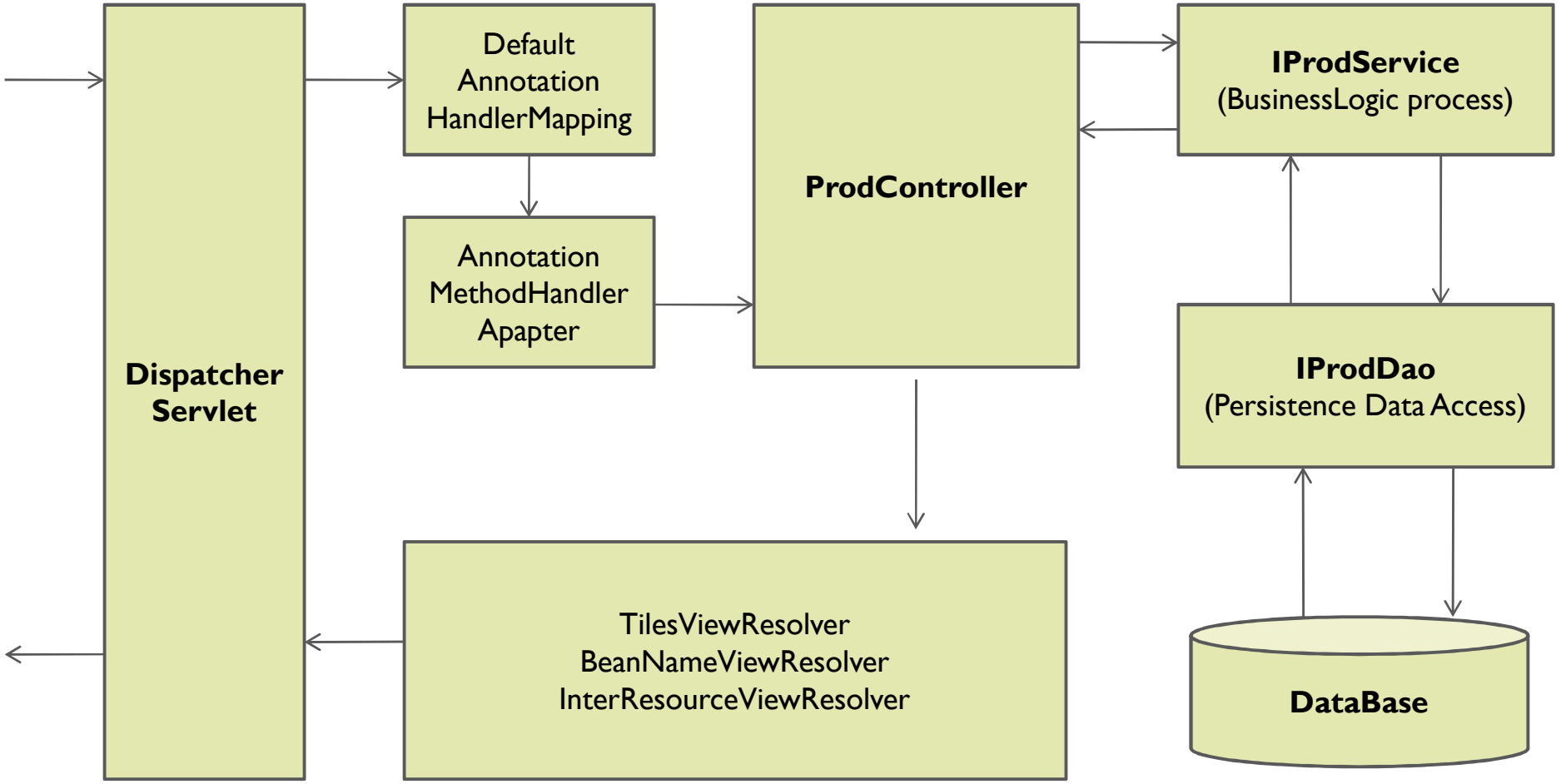
* 의존관계 구조(class diagram)



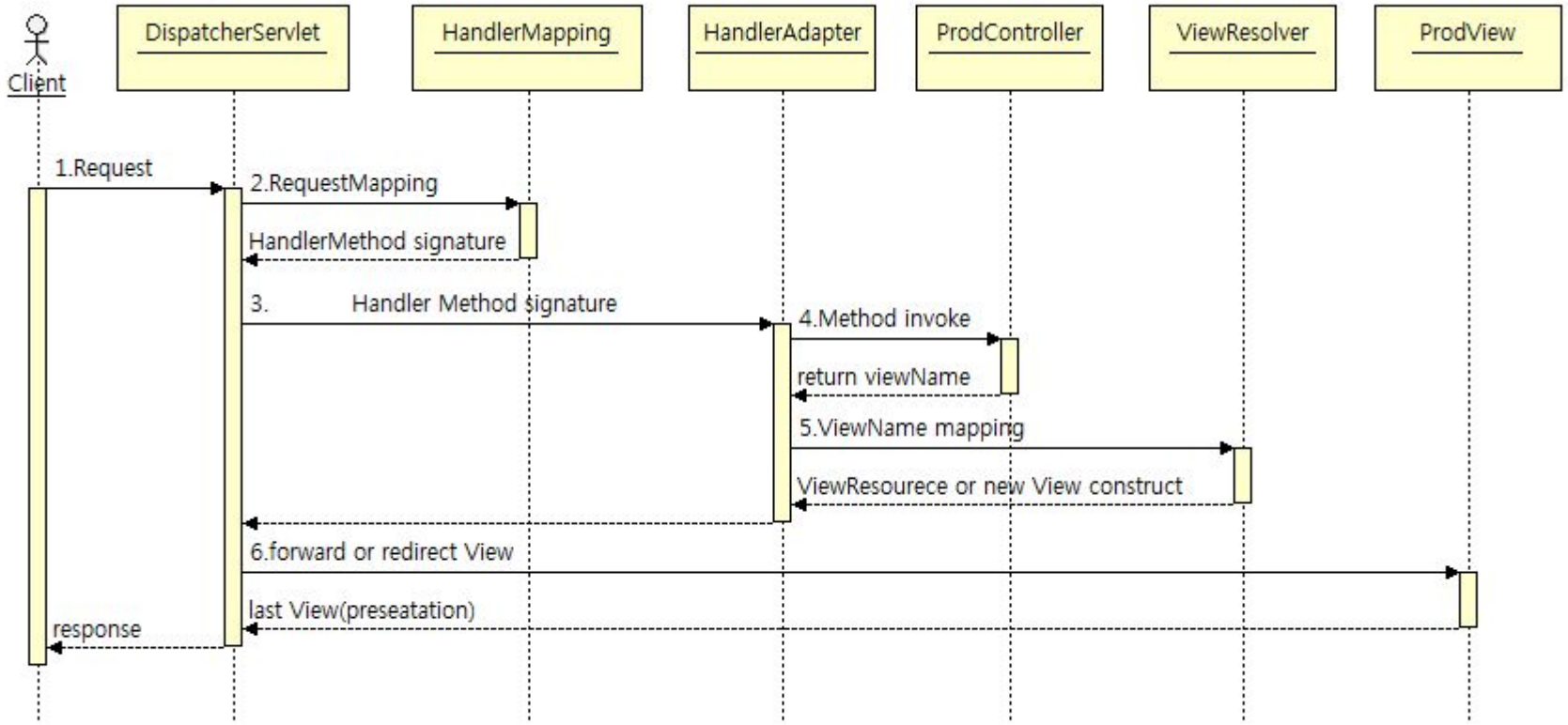
Spring WebMVC Architecture



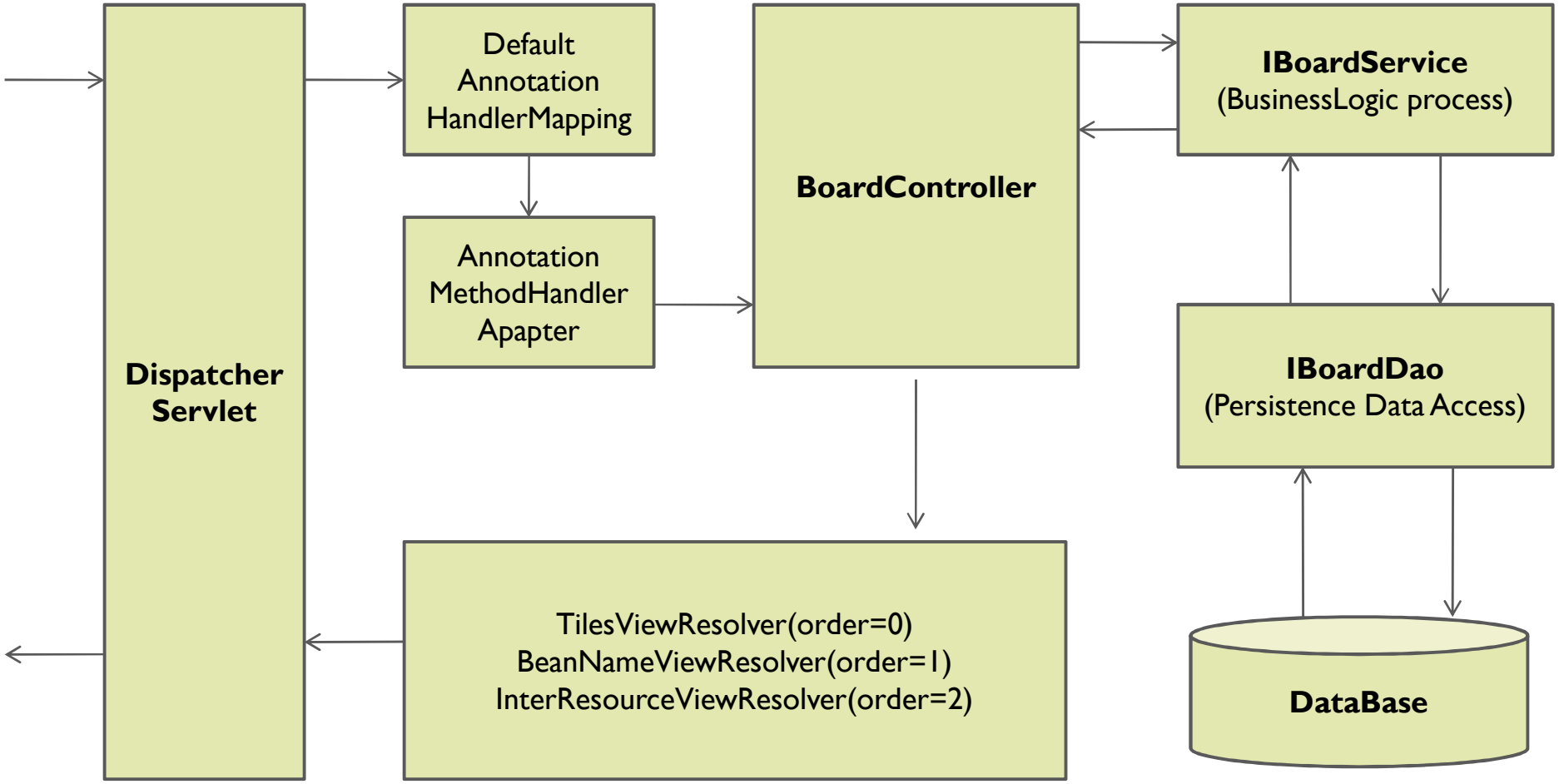
Request 처리 과정
(spring web Applicaion-상품관리)



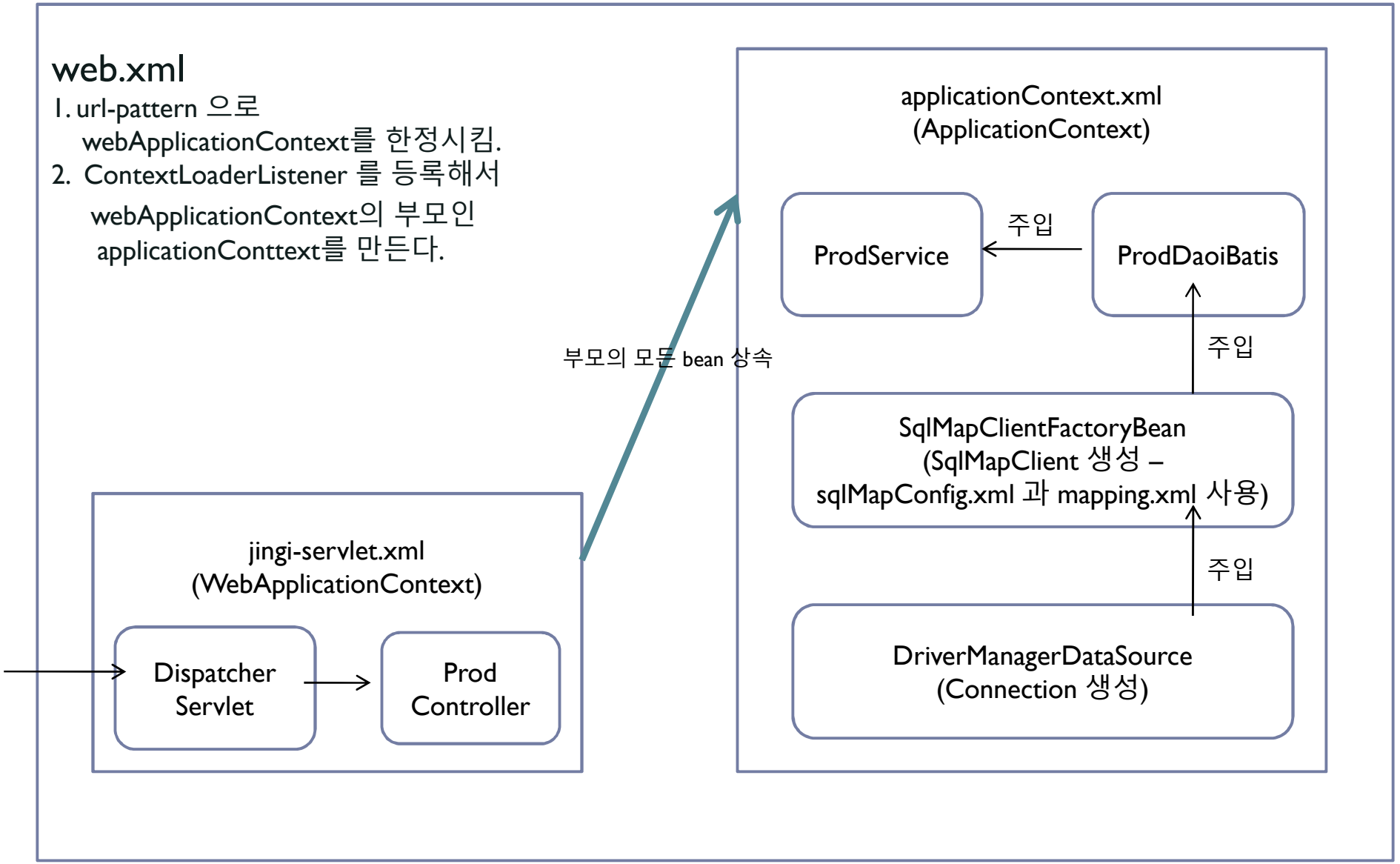
Request 처리 흐름도
(spring web Applicaion-상품관리)



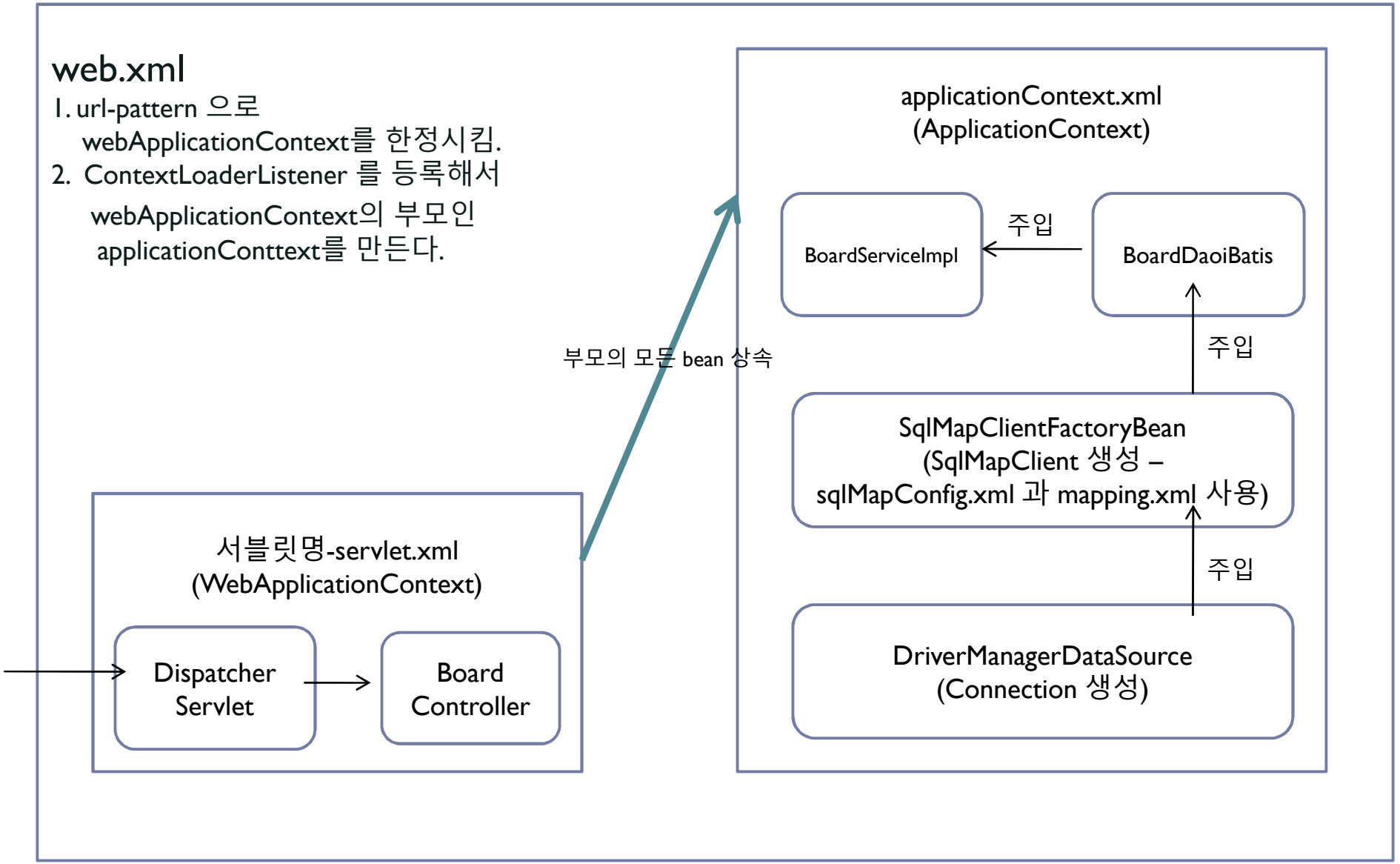
Request 처리 과정
(spring web Applicaion-게시판)



막 그냥 이해를 위한 그림(상품관리)



막 그냥 이해를 위한 그림(게시판)



순서대로 따라하기!!(HelloSpring)

PDF (Spring board tutorial)파일 참고