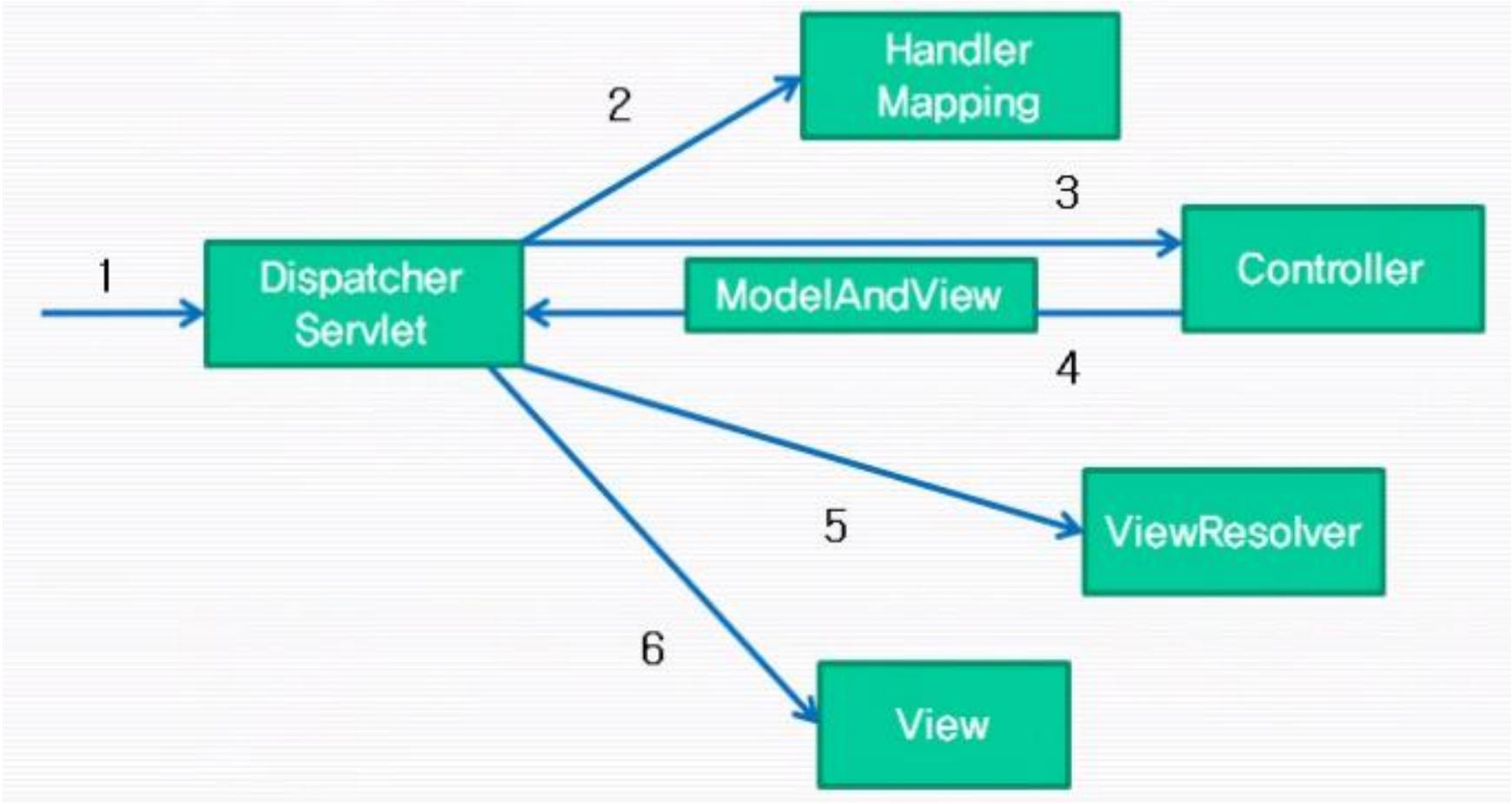
스프링 MVC의 개요

- **스프링 프레임워크에 내장되어 있는 웹 프레임워크**
- **MVC 패턴에 기반하기 때문에 결합도가 낮고 유연한 웹 기반 어플리케이션을 쉽게 작성**
- **낮은 결합도, 종속객체 주입, 확장성이라는 강력하고 유연함**
- **다양한 핸들러 매핑을 제공**
- **다양한 컨트롤러 제공**
  - 한 두개의 Action클래스로 선택의 폭이 한정되어 있는 스트럿츠나 웹워크 등의 MVC웹 프레임워크와 스프링이 구별되는 점이다.

- **요청을 처리하기 위해 컨트롤러를 선택하는 방법과 결과를 보여주기 위해 뷰를 선택하는 방식 간에 낮은 결합도를 유지하고 있다는 점이 무엇보다 중요하다.**
  - 서로 다른 스프링MVC 부분들을 짜맞춰서 각자의 애플리케이션에 가장 적합한 웹 계층을 구축

스프링 MVC의 생명주기



DispatcherServlet

- 스프링MVC의 요체로서 프론트 컨트롤러의 역할을 담당
- 서블릿이름-servlet.xml 이라는 파일을 이용해서 어플리케이션 컨텍스트를 로드
- Application Context를 Application 계층에 따라 xml을 분리하는 것을 권장

서블릿명-security.xml
서블릿명-servlet.xml
서블릿명-service.xml
서블릿명-data.xml

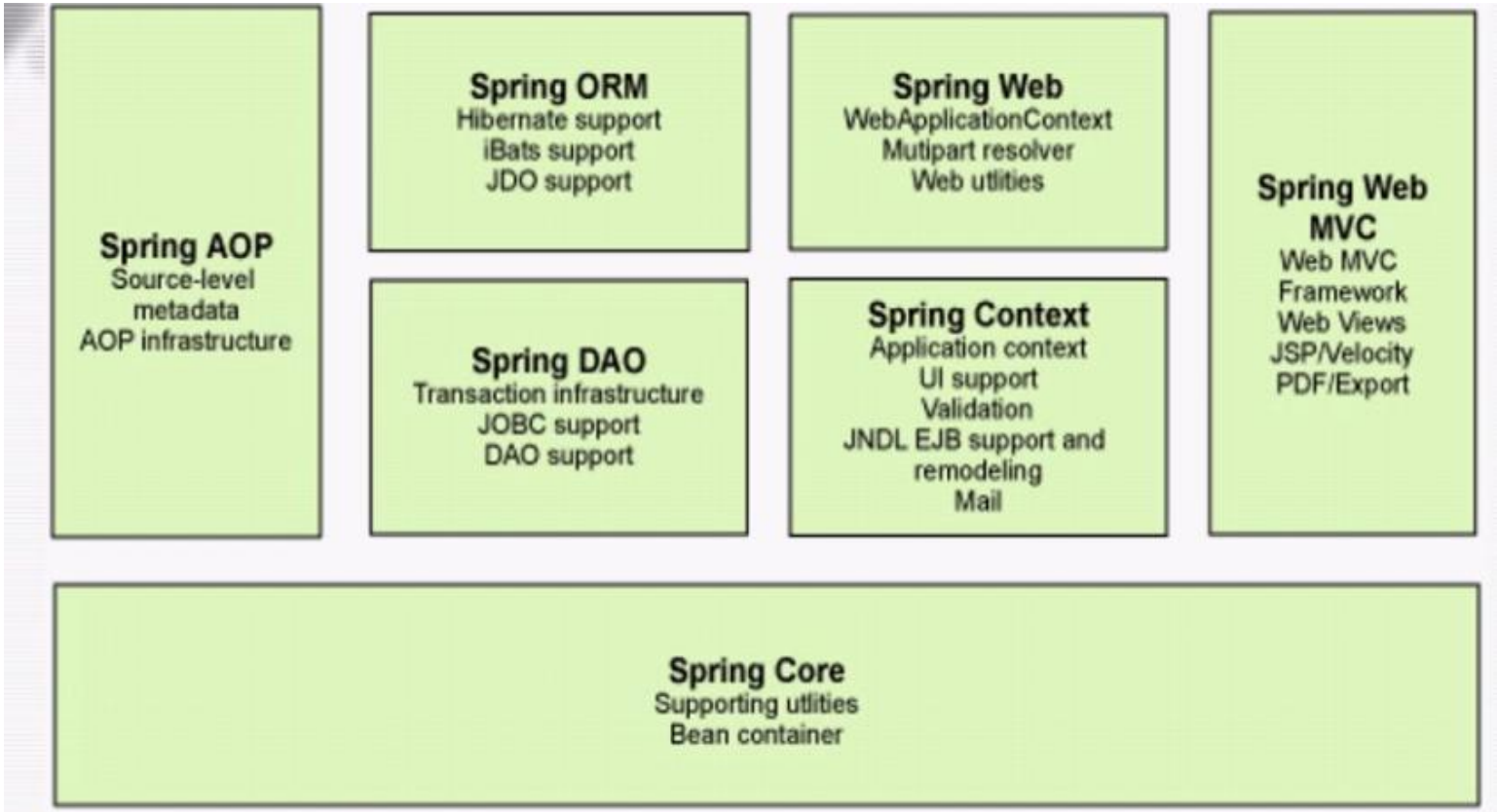
- 컨텍스트 로더 구성
  - 이 파일들이 모두 로드되게 하려면 web.xml에 컨텍스트 로더를 구성해야 한다.
  - ContextLoaderListener 서블릿리스너 이용

```
〈listener〉
    〈listener-class〉org.springframework.web.context.ContextLoaderListener〈/listener-class〉
〈/listener〉
```

- contextConfigLocation파라미터 지정하면 컨텍스트 로더로 하나 이상의 설정파일 지정가능

```
<context-param>
        <param-name> contextConfigLocation</param-name>
        <param-value>
                /WEB-INF/서블릿명-service.xml
                ...
        </param-value>
</context-param>
```

Spring의 구조



Core : DI 라는 Dependency Injection 기능을 제공.
Context : 컨텍스트라는 정보를 제공하는 설정을 관리한다. JNDI, EJB, 국제화, 스케줄링이 여기에 포함
DAO : DB와 관련된 JDBC 코딩 부분을 처리해 주기 위한 JDBC 추상화 레이어를 제공.
ORM : JDO, Hibernate, iBATIS 등 O-R Mapping API를 위한 레이어를 제공.
AOP : 관점지향 프로그래밍을 제공.
Web : 웹 기반의 여러가지 기능을 제공.
Web MVC : Model과 View(Web form) 사이의 구분을 제공하기 위한 관련된 기능을 제공.

Controller 계층 구조