

# CS Study 14주차

Web Questions

김신아



WEB(Spring)에 관한 질문 모음집

## 1. 스프링이 무엇인지 간단히 설명해보세요.

---

- Java SE(Standard Edition)

: 자바 언어의 핵심 기능을 제공한다. 여기서 말하는 핵심 기능은 Basic Type 부터 Networking, security, database access, GUI, XML Parsing까지 모두 해당된다.

- Java EE(Enterprise Edition)

: 자바로 웹개발을 할 수 있게 해주는 표준이며, Java SE를 기반으로 확장된 개념이다.

large-scale, multi-tiered, scalable, reliable and secure network application을 개발하기 위한 API와 런타임환경을 제공한다.

- POJO (Plain Old Java Object)

: 다른 클래스나 인터페이스를 상속/implements받아 매서드가 추가된 클래스가 아닌 일반적으로 우리가 알고 있는 getter, setter 같이 기본적인 기능만 가진 자바 객체

## 1. 스프링이 무엇인지 간단히 설명해보세요.

---



A

스프링은 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크입니다.

자바 SE로 된 자바 객체 POJO를 자바 EE에 의존적이지 않게 연결해주는 역할을 합니다. 스프링의 특징으로는 크기와 부하 측면에서 경량 시킨 것과, IOC 기술로 애플리케이션의 느슨한 결합을 도모시킨 것이 있습니다.

## 2. 스프링이랑 스프링 부트는 차이점이 무엇인가요?

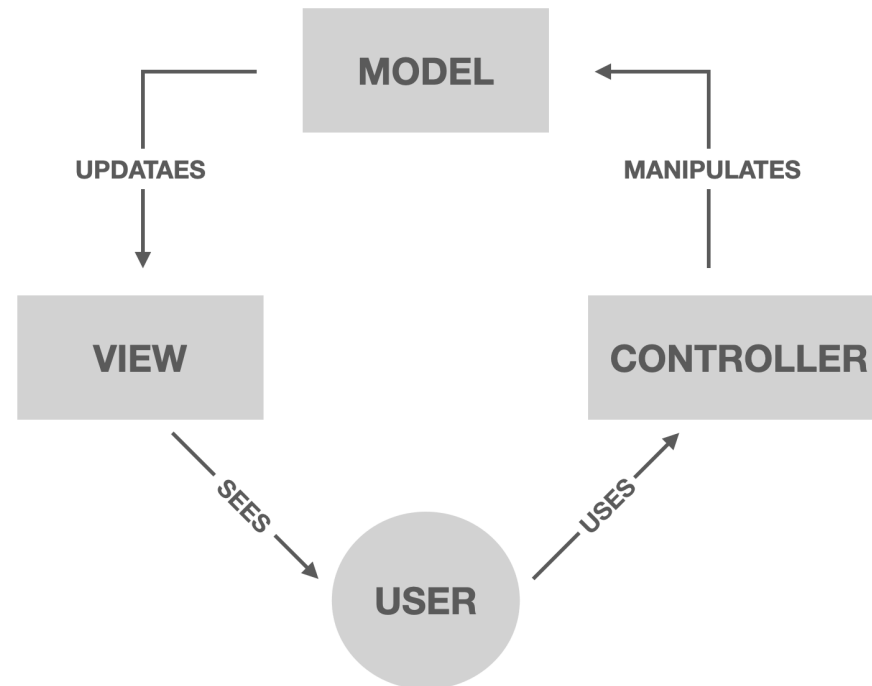
---



A

스프링 부트는 스프링에서 사용하는 프로젝트를 간편하게 셋업할 수 있는 서브 프로젝트입니다. 독립 컨테이너에서 동작할 수 있기 때문에 임베디드 톰캣이 자동으로 실행됩니다. 임베디드 컨테이너에서 애플리케이션을 실행시키기에는 다소 불안전해서 큰 프로젝트는 사용하지 않는 것이 좋습니다.

### 3. MVC 패턴이란?



A

MVC 패턴은 코드의 재사용에 유용하며, 사용자 인터페이스와 응용 프로그램 개발에 소요되는 시간을 줄여주는 효과적인 설계 방식을 말합니다.

구성요소로는 Model, View, Controller가 있습니다.

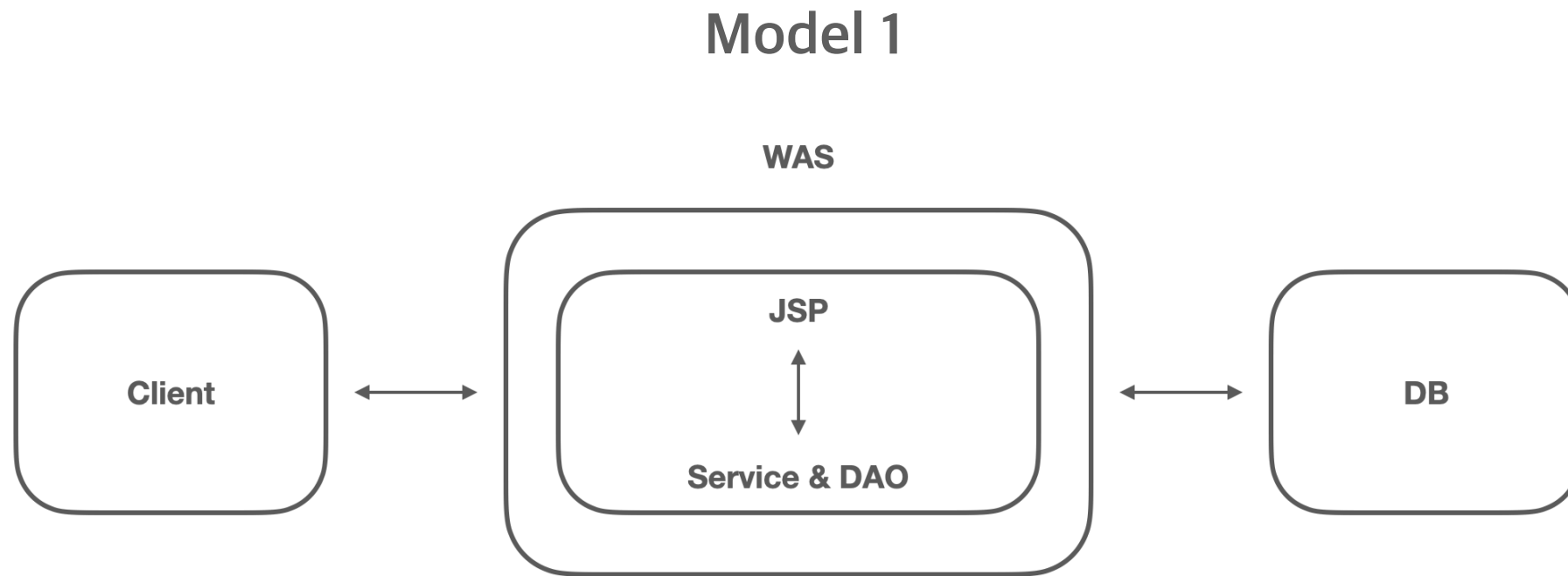
모델은 핵심적인 비즈니스 로직을 담당하여 데이터베이스를 관리하는 부분이고,

뷰는 사용자에게 보여주는 화면,

컨트롤러는 모델과 뷰 사이에서 정보 교환을 할 수 있도록 연결시켜주는 역할을 합니다.

## 4. MVC1이랑 MVC2 패턴 차이에 대해 설명해주세요.

---

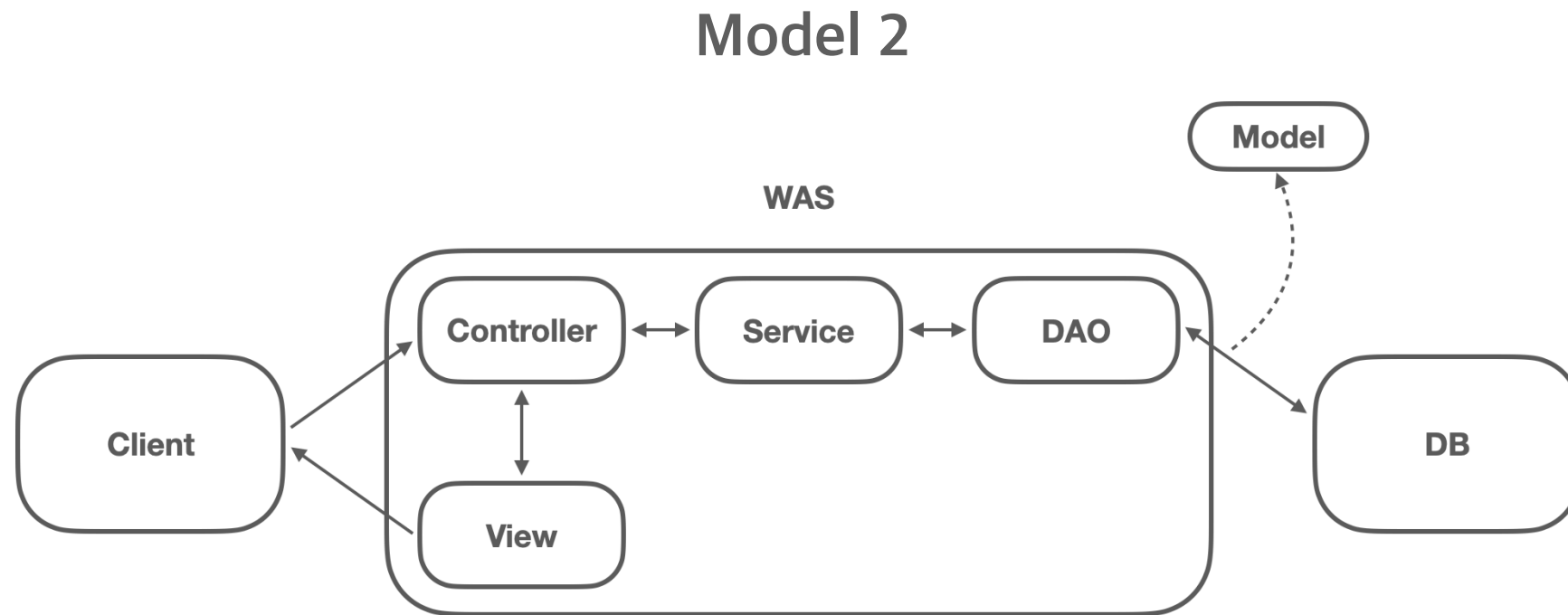


A

모델1은 JSP 페이지 안에서 로직 처리를 위해 자바 코드가 함께 사용됩니다.

요청이 오면, 직접 자바빈이나 클래스를 이용해 작업을 처리하고, 이를 클라이언트에 출력해줍니다. 구조가 단순한 장점이 있지만, JSP내에서 html 코드와 자바 코드가 같이 사용되면서 복잡해지고 유지보수가 어려운 단점이 있습니다.

## 4. MVC1이랑 MVC2 패턴 차이에 대해 설명해주세요.

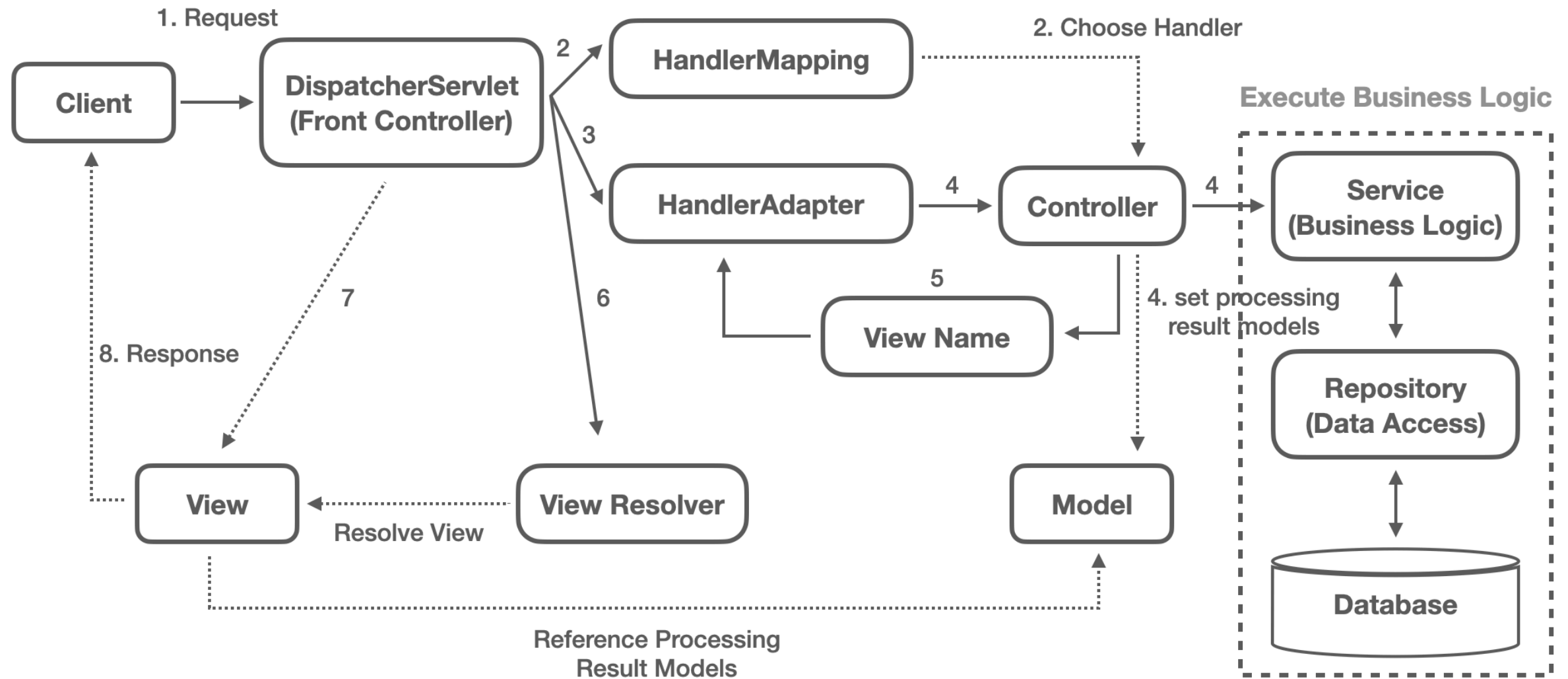


A

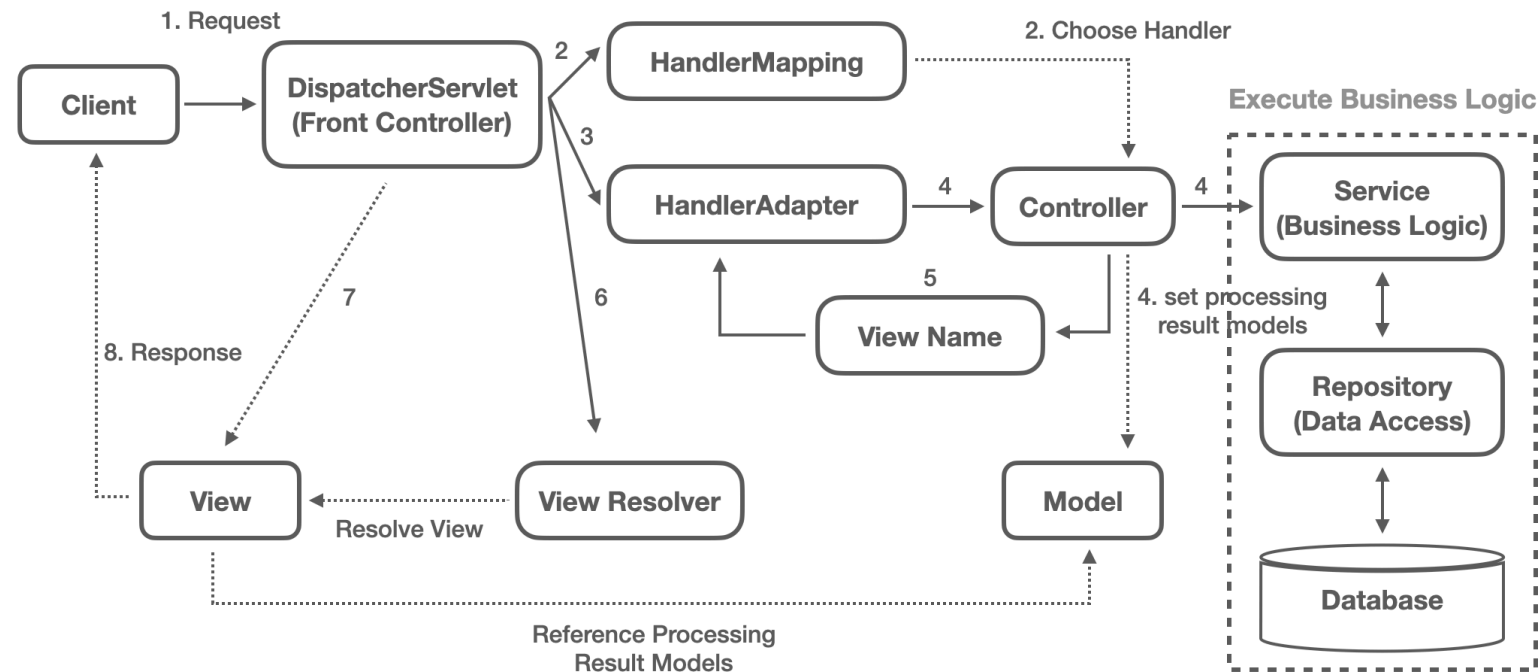
모델2는 모델1과는 다르게 모든 처리를 JSP에서만 담당하는 것이 아니라 서블릿을 만들어 역할 분담을 하는 패턴입니다. 요청 결과를 출력해주는 뷰만 JSP가 담당하고, 흐름을 제어해주고 비즈니스 로직에 해당하는 컨트롤러의 역할을 서블릿이 담당하게 됩니다. 이처럼 역할을 분담하면서 유지보수가 용이해지는 장점이 있지만 습득하기 힘들고 구조가 복잡해지는 단점도 있습니다.



## 5. 스프링 MVC 구조 흐름에 대해 과정대로 설명해보세요.



## 5. 스프링 MVC 구조 흐름에 대해 과정대로 설명해보세요.



A

디스패처 서블릿이 클라이언트로부터 요청을 받으면, 이를 요청할 핸들러 이름을 알기 위해 핸들러 맵핑에게 물어봅니다.

핸들러맵핑은 요청 url을 보고 핸들러 이름을 디스패처 서블릿에게 알려줍니다.

이때 핸들러를 실행하기 전/후에 처리할 것들을 인터셉터로 만들어 줍니다.

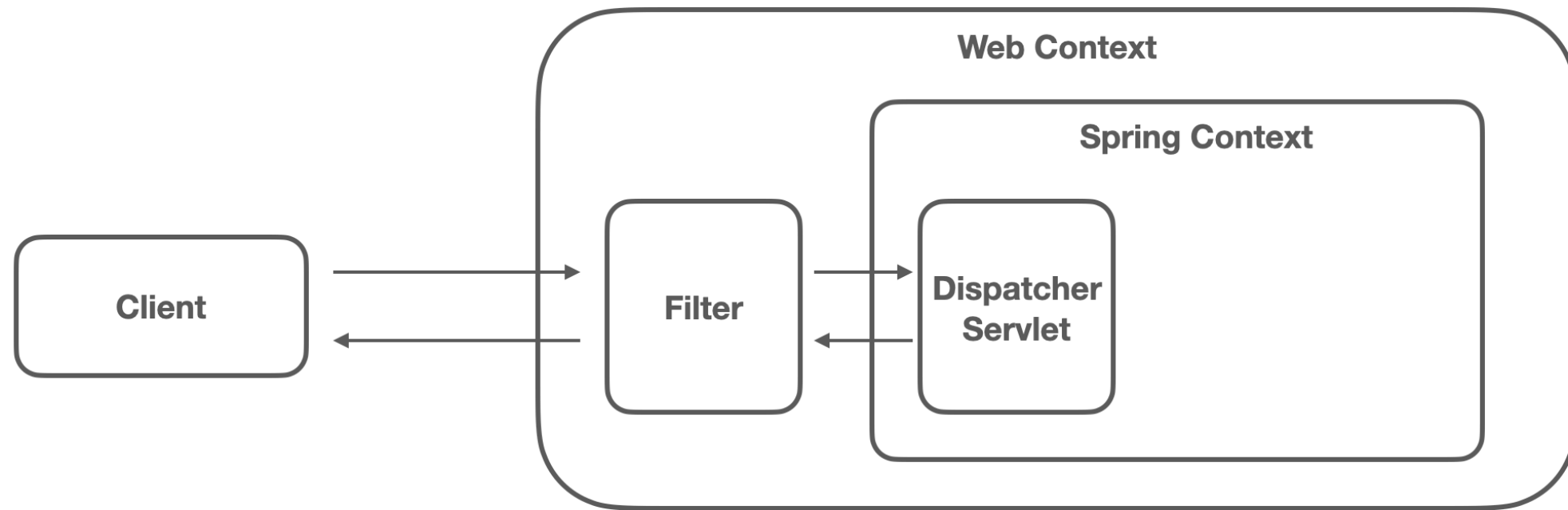
디스패처 서블릿은 해당 핸들러에게 제어권을 넘겨주고, 이 핸들러는 응답에 필요한 서비스를 호출하고 랜더링해야 하는 뷰 이름을 판단하여 디스패처 서블릿에게 전송해줍니다.

디스패처 서블릿은 받은 뷰 이름을 뷰 리졸버에게 전달해 응답에 필요한 뷰를 만들라고 명령합니다.

이때 해당하는 뷰는 디스패처 서블릿에게 받은 모델과 컨트롤러를 활용해 원하는 응답을 생성해서 다시 보내줍니다.

디스패처 서블릿은 뷰로부터 받은 것을 클라이언트에게 응답해줍니다.

## 6. 스프링 필터랑 인터셉터의 차이점은 무엇인가요?

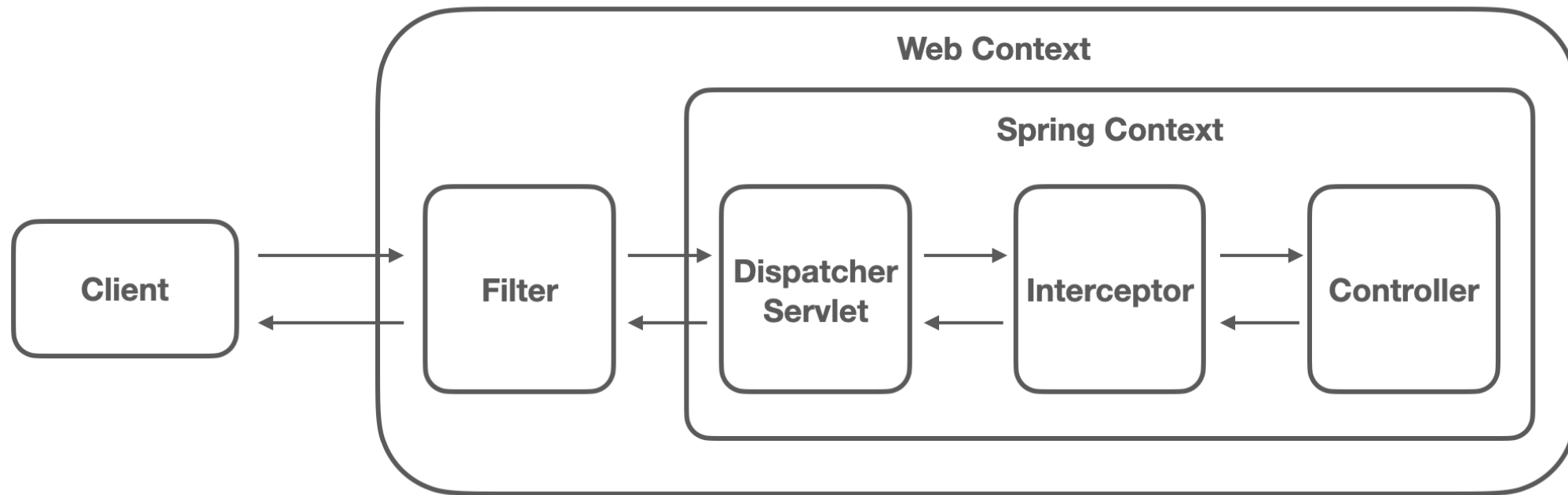


필터는 J2EE 표준 스펙 기능으로 디스패처 서블릿(Dispatcher Servlet)에 요청이 전달되기 전/후에 url 패턴에 맞는 모든 요청에 대해 부가작업을 처리할 수 있는 기능을 제공한다.

디스패처 서블릿은 스프링의 가장 앞단에 존재하는 프론트 컨트롤러이므로, 필터는 스프링 범위 밖에서 처리가 되는 것이다.

즉 스프링 컨테이너가 아닌 톰캣과 같은 웹 컨테이너에 의해 관리가 되는 것이고, 디스패처 서블릿 전/후에 처리하는 것이다.

## 6. 스프링 필터랑 인터셉터의 차이점은 무엇인가요?



인터셉터는 필터와 달리 Spring이 제공하는 기술로써, 디스패처 서블릿이 컨트롤러를 호출하기 전과 후에 요청과 응답을 참조하거나 가공할 수 있는 기능을 제공한다.

디스패처 서블릿은 핸들러 매핑을 통해 적절한 컨트롤러를 찾으려 하는데, 그 결과로 실행 체인을 돌려준다. 그래서 이 실행 체인은 1개 이상의 인터셉터가 등록되어 있다면 순차적으로 인터셉터들을 거쳐 컨트롤러가 실행되도록 하고, 인터셉터가 없다면 바로 컨트롤러를 실행한다.

인터셉터는 스프링 컨테이너 내에서 동작하므로 필터를 거쳐 프론트 컨트롤러인 디스패처 서블릿이 요청을 받은 이후에 동작하게 된다.

## 6. 스프링 필터랑 인터셉터의 차이점은 무엇인가요?

대상	필터(Filter)	인터셉터(Interceptor)
관리되는 컨테이너	웹 컨테이너	스프링 컨테이너
Request/Response 조작 가능 여부	O	X
용도	<ul style="list-style-type: none"><li>• 보안 관련 공통 작업</li><li>• 모든 요청에 대한 로깅 또는 감사</li><li>• 이미지/데이터 압축 및 문자열 인코딩</li></ul>	<ul style="list-style-type: none"><li>• 인증/인가 등과 같은 공통 작업</li><li>• Controller로 넘겨주는 정보의 가공</li><li>• API 호출에 대한 로깅 또는 감사</li></ul>

## 6. 스프링 필터랑 인터셉터의 차이점은 무엇인가요?

---

A

필터와 인터셉터는 실행되는 시점에서 차이가 있습니다.

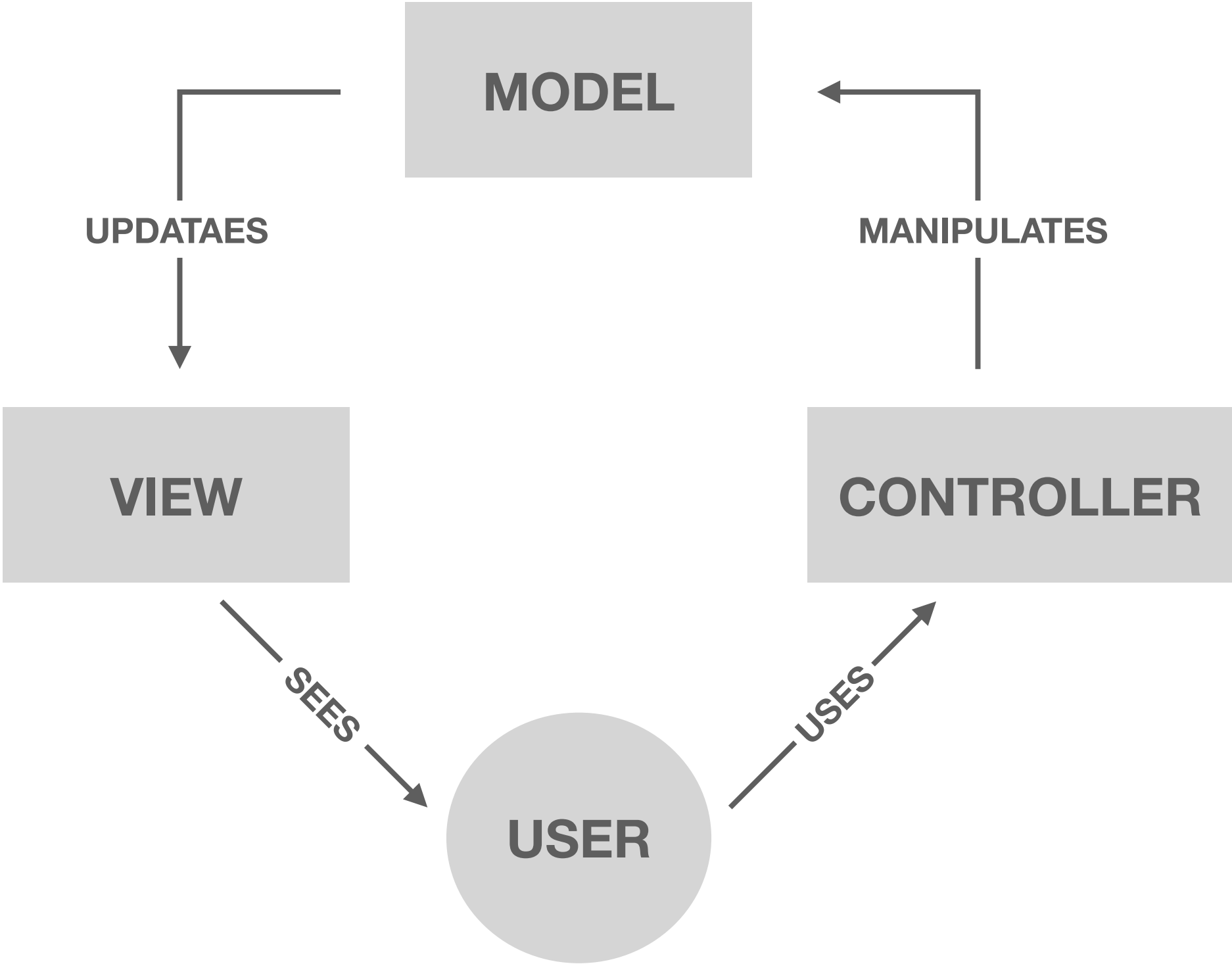
필터는 웹 애플리케이션에 등록을 하고,

인터셉터는 스프링의 context에 등록을 합니다.

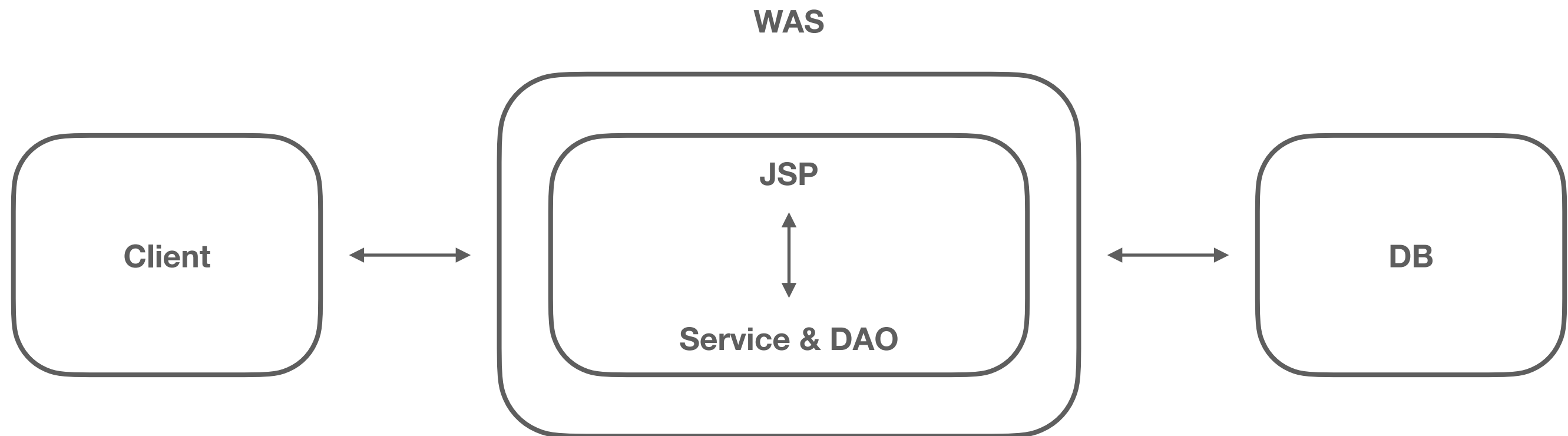
따라서 컨트롤러에 들어가기 전 작업을 처리하기 위해 사용하는 공통점이 있지만, 호출되는 시점에서 차이가 존재합니다.

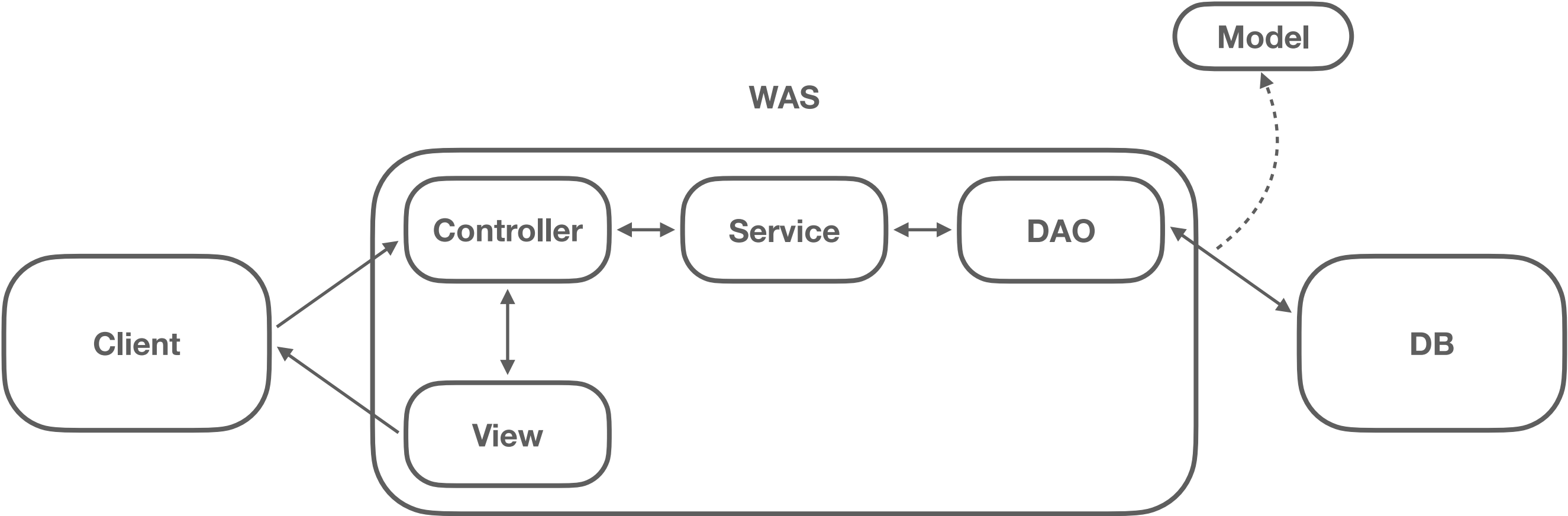
# Thank You

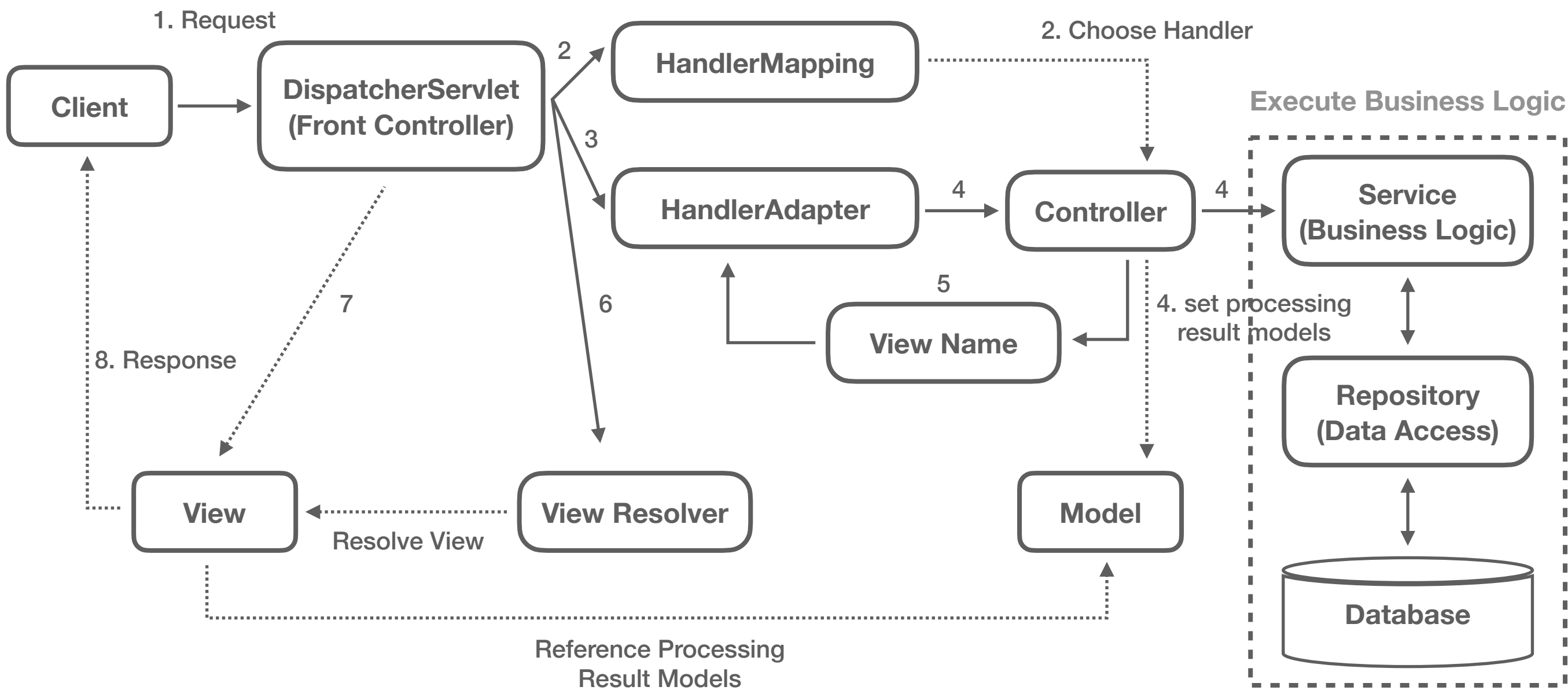
---

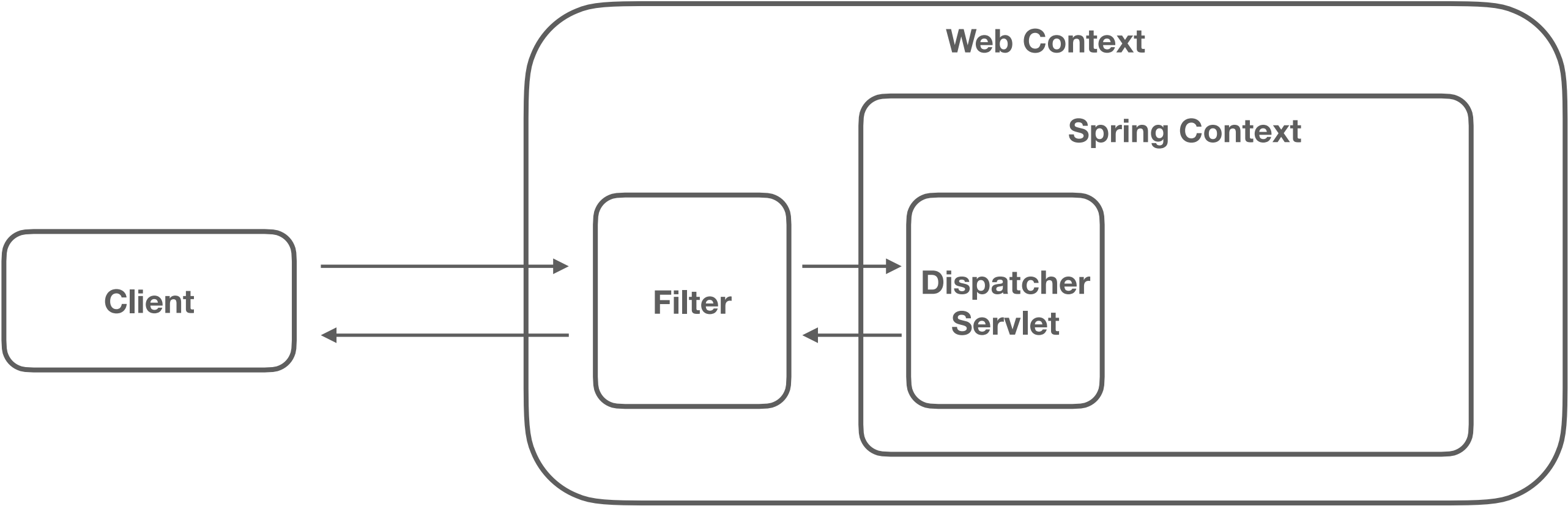


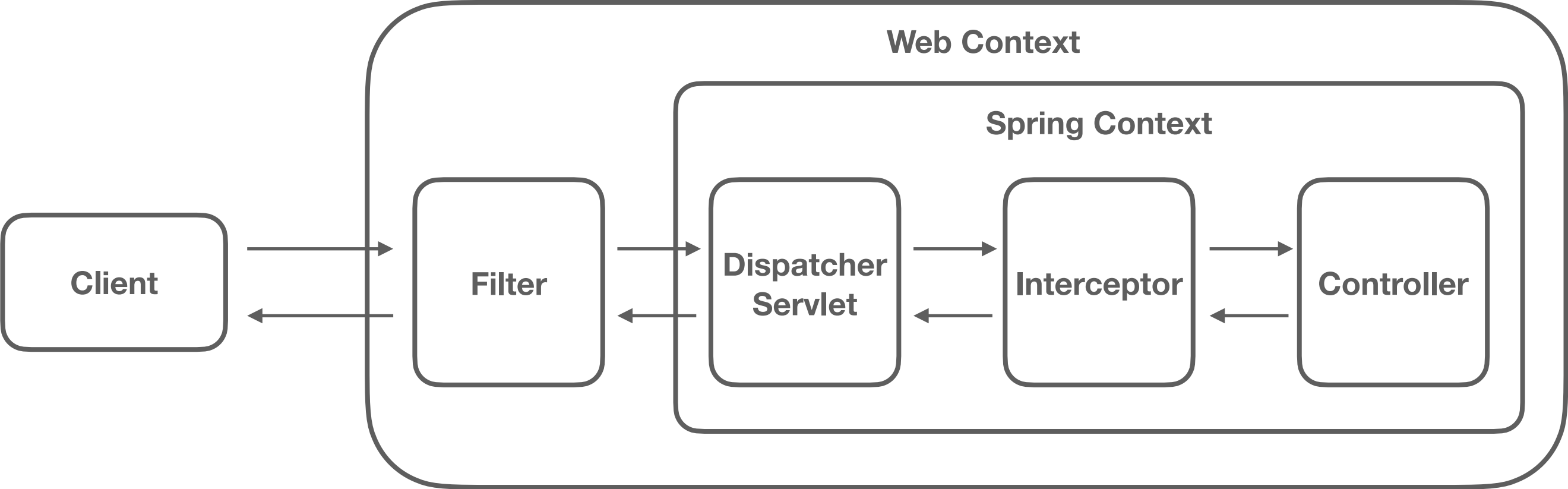












# 별첨

---

<https://gyoogle.dev/blog/interview/%EC%9B%B9.html>

<https://velog.io/@jihoson94/Java-SE-vs-EE>

<https://velog.io/@galaxy/Spring%EC%9D%98-%EA%B8%B0%EB%B3%B8-%ED%8A%B9%EC%A7%95-POJO>

<https://aridom.tistory.com/61>

<https://mangkyu.tistory.com/173>