

MVC Design pattern

Model View Controller Design pattern



MVC

사용자 인터페이스로부터 비즈니스 로직을 분리하여
서로 간의 결합도를 최소화하고 유지보수성을 높이며,
개발자들의 개발 효율을 높일 수 있는 디자인 패턴

Model-View-Controller - wikipedia

MVC

Controller

- 서비스 처리 메소드 호출
- 중계역할

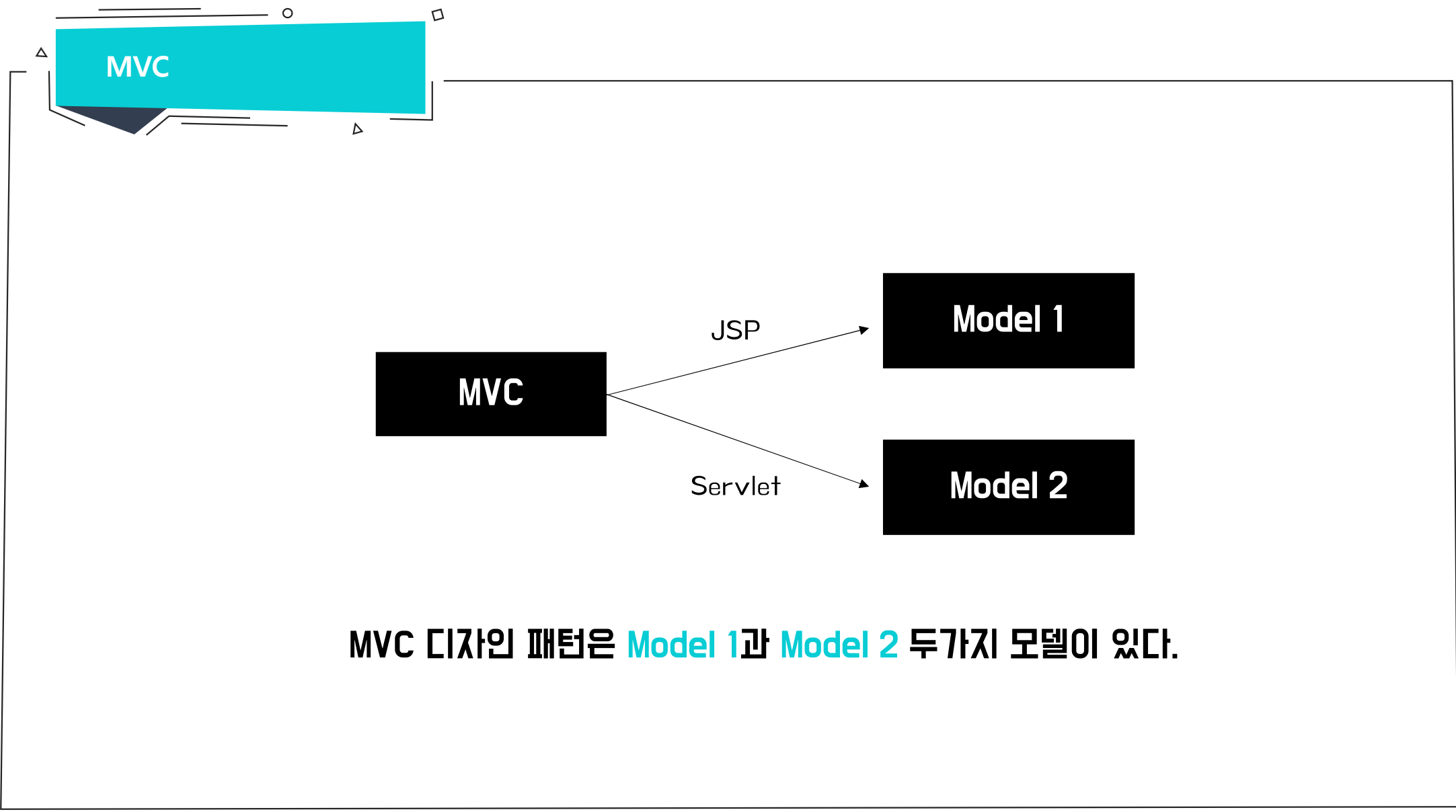
Model

- 동작을 수행하는 코드
- Public으로만 구성
- 비즈니스 로직을 처리
- Service / DAO

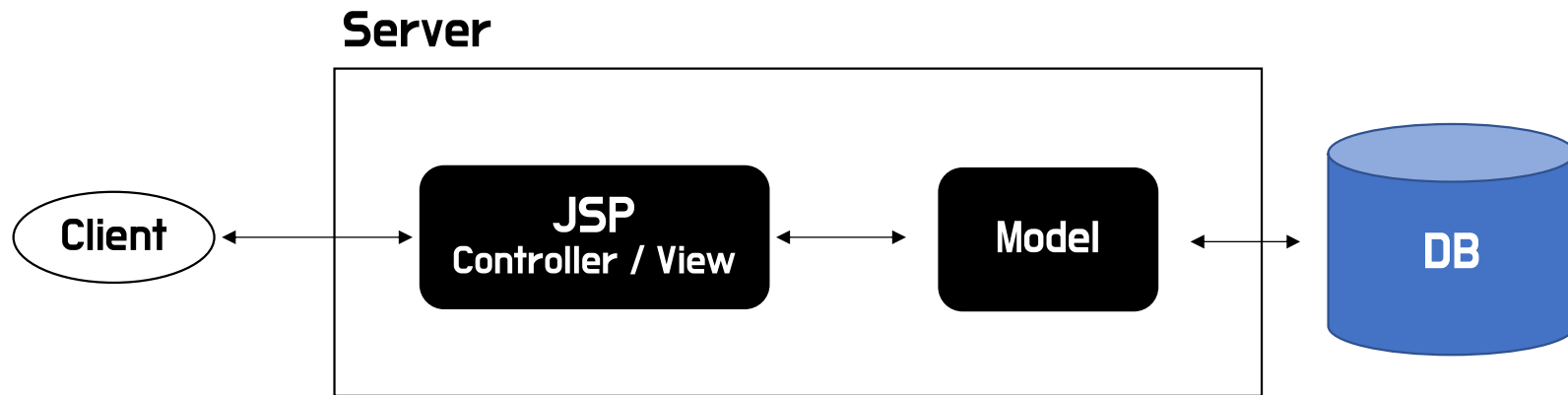
Client

View

- 클라이언트가 보는 화면
- 요청으로 인해 발생한 결과 출력 페이지
- 사용자 인터페이스 요소

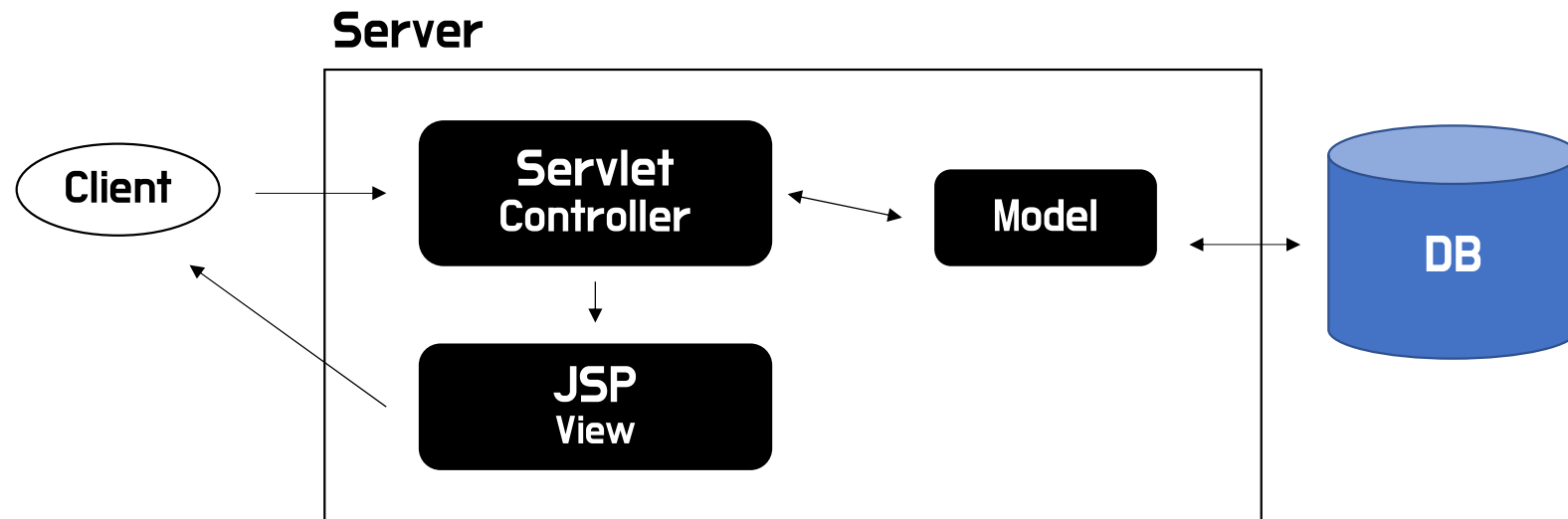


MVC – model 1



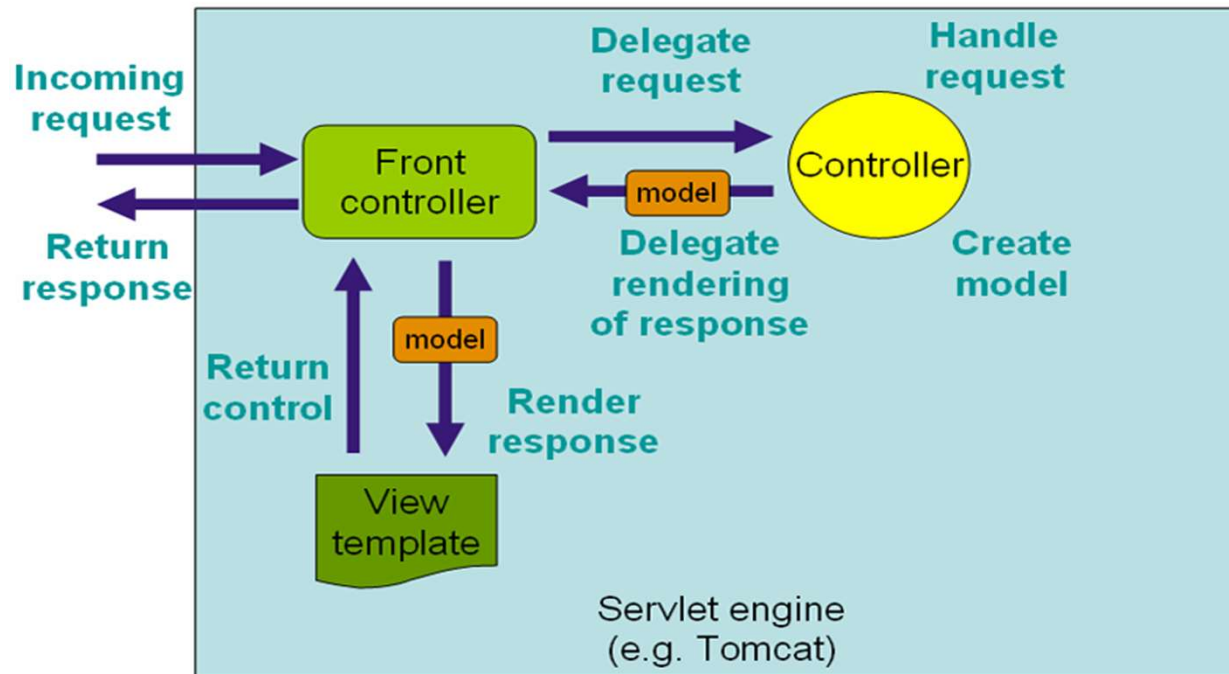
JSP가 요청을 처리하면서 View 역할도 **동시에** 수행
규모가 큰 프로젝트에서는 **유지보수가 힘들**

MVC – model 2



Controller, View의 분리로 유지보수 성능 향상
설계가 까다로워지는 단점 발생

Spring with MVC

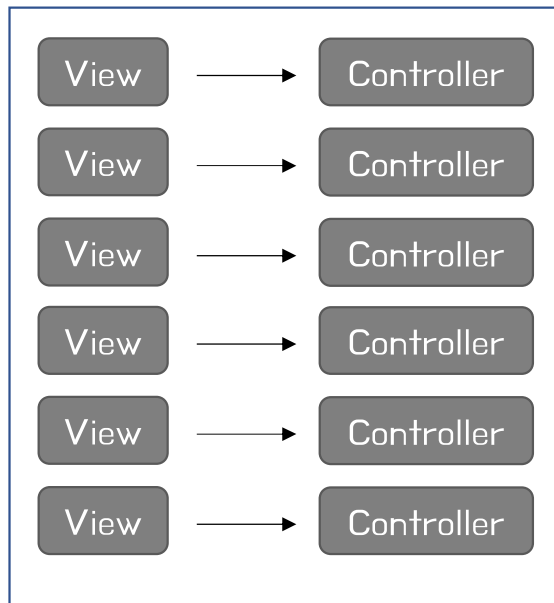


Spring document

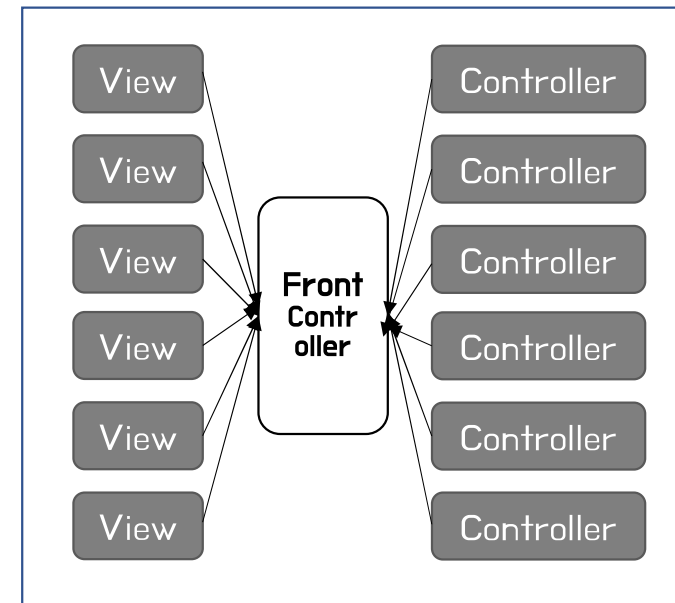
Spring은 Model 2를 선택했다.

Servlet – Front Controller

Before



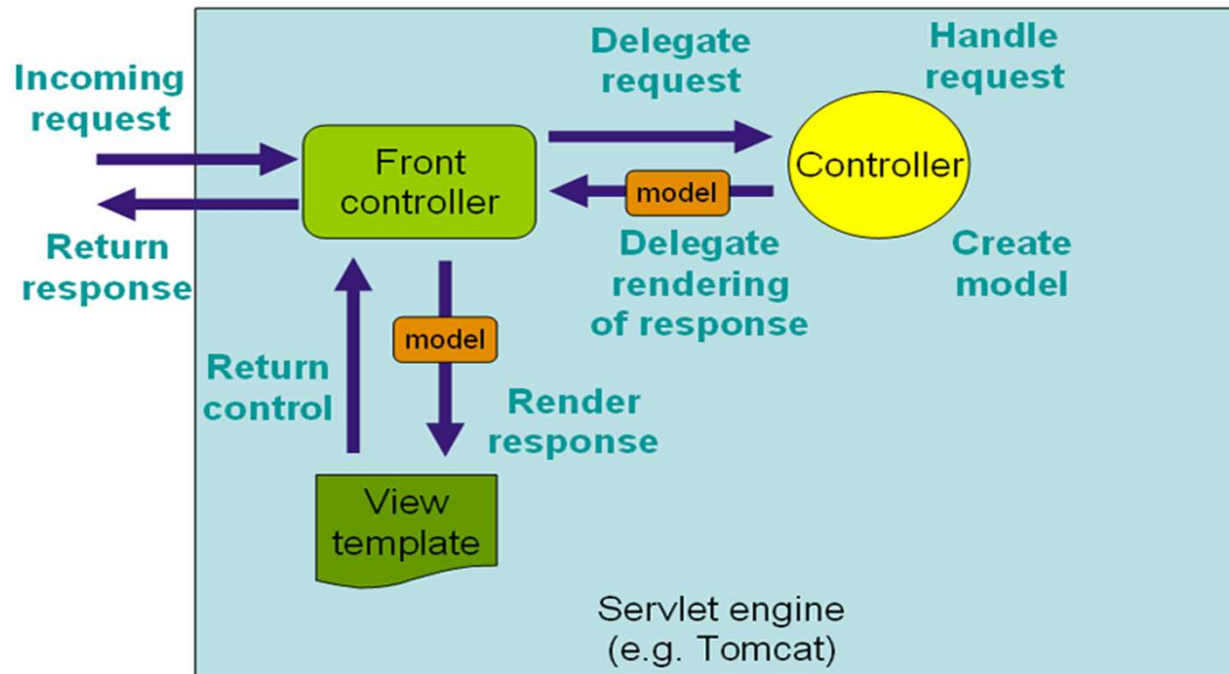
After



Spring document

Front controller Design pattern

3-Tier Architecture



Spring document

Spring은 Model 2를 선택했다.