

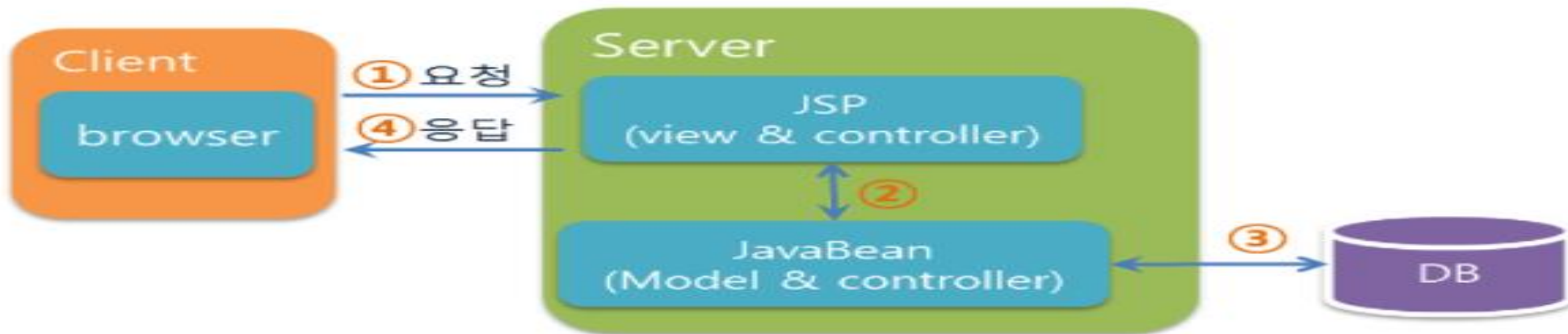
Model2-MVC패턴

안화수

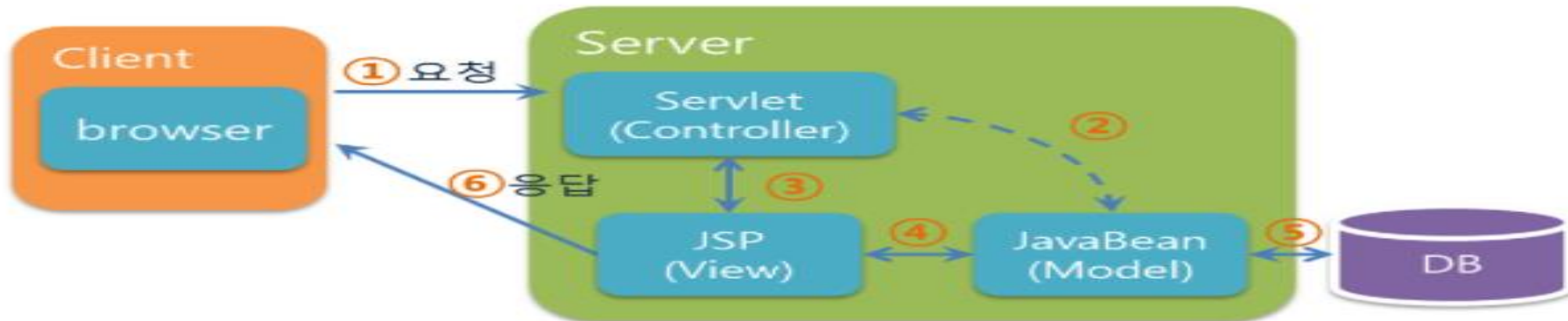
Web Application 설계방식

- ❖ Model 1 설계 방식
- ❖ Model 2 설계 방식

Model 1 and Model 2



<MVC - Model 1>



<MVC - Model 2>

모델1 설계방식

❖ Model 1 개요

- JSP 만 이용하여 개발하는 경우
- JSP + Java Bean을 이용하여 개발하는 경우
- Model2의 Controller 개념이 모호함

❖ Model 1 설계 방식의 장.단점

➤ 장점

- 개발속도가 빠름
- 개발자의 기술적인 숙련도가 낮아도 배우기 쉬워서 빠르게 적용이 가능함

➤ 단점

- JSP 페이지에서 프레젠테이션 로직과 비즈니스 로직이 혼재되어서 복잡함
- 로직의 혼재로 인해서 개발자와 디자이너의 작업 분리가 어려움
- JSP 코드의 복잡도로 인해서 유지보수가 어려워짐

모델2 설계방식

❖ Model 2 개요

- GUI 개발모델인 MVC를 웹 애플리케이션에 적용하여 구현한 방식
- Application의 역할을 Model – View – Controller로 분리시켜서 개발하는 방식

❖ Model 2 설계 방식의 장.단점

➤ 장점

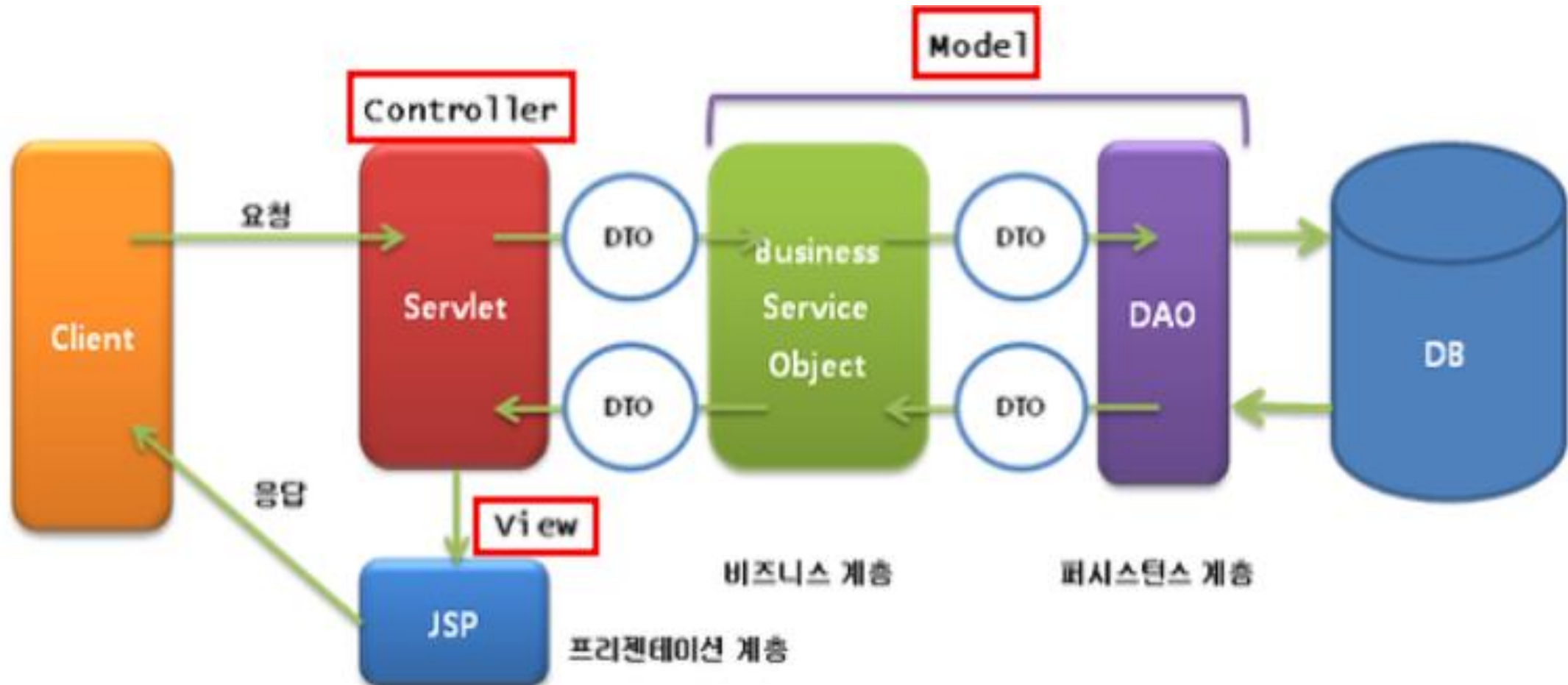
- 비즈니스 로직과 프리젠테이션의 분리로 인해 애플리케이션이 명료해지며 유지보수와 확장이 용이함
- 디자이너와 개발자의 작업을 분리해 줌

➤ 단점

- 개발 초기에 아키텍처 디자인을 위한 시간의 소요로 개발 기간이 늘어남
- MVC 구조에 대한 개발자들의 이해가 필요함

모델2 설계방식

❖ MVC Architecture



모델2 설계방식

❖ MVC 패턴

MVC는 Model / View / Controller 의 약자로 애플리케이션을 세 역할로 나누어서 개발하는 개발 방법론이다.

➤ Model

- 애플리케이션의 데이터 처리를 담당함
- Service 클래스 + DAO 클래스로 구현함

➤ View

- 사용자 인터페이스를 처리함
- JSP 를 이용해서 구현함 : EL(Expression Language, 표현언어) + JSTL(JSP Standard Tag Library)

➤ Controller

- 클라이언트의 요청을 받아 Model과 View사이에서 흐름을 조정한다.
- Java Servlet으로 구현함