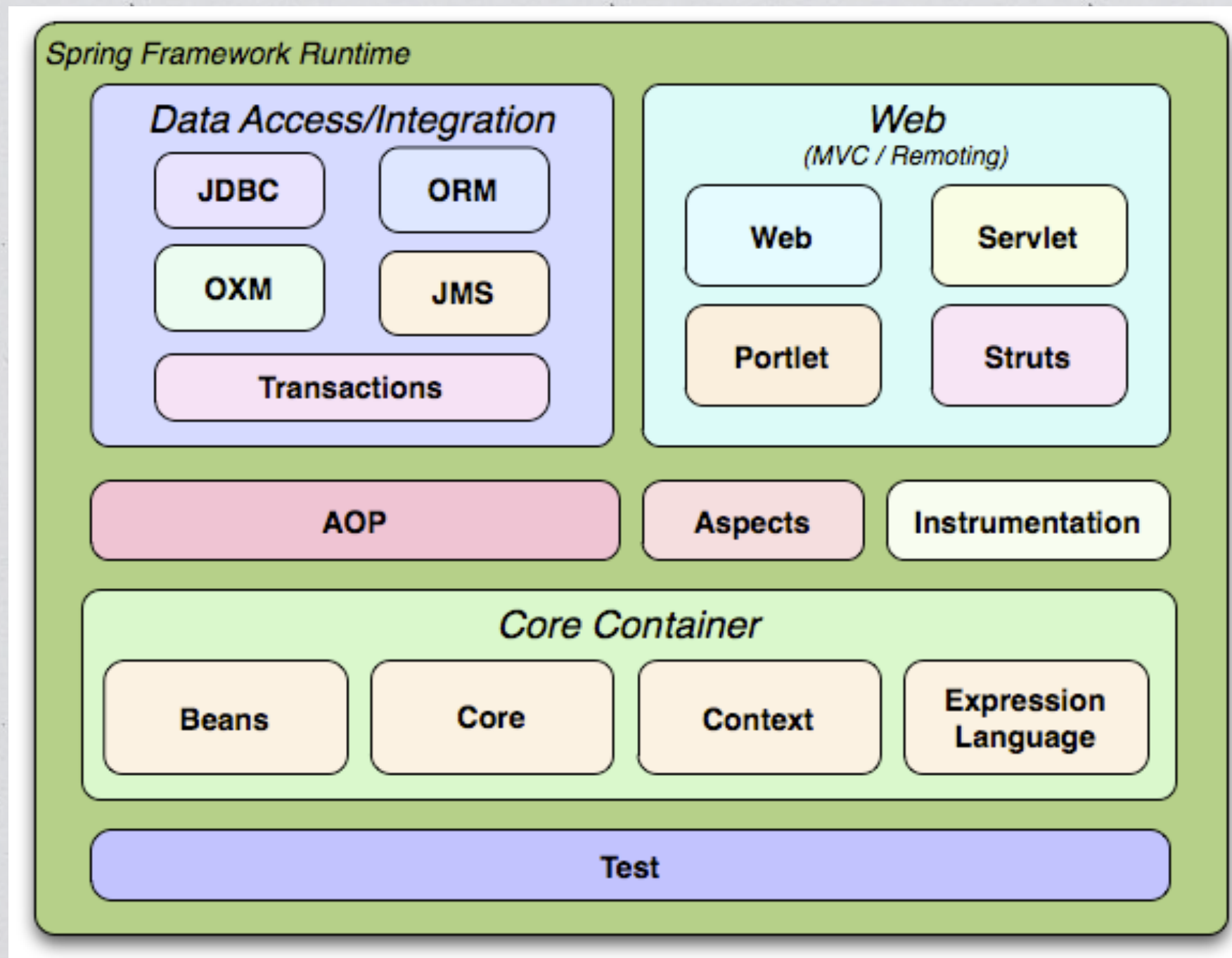

SPRING

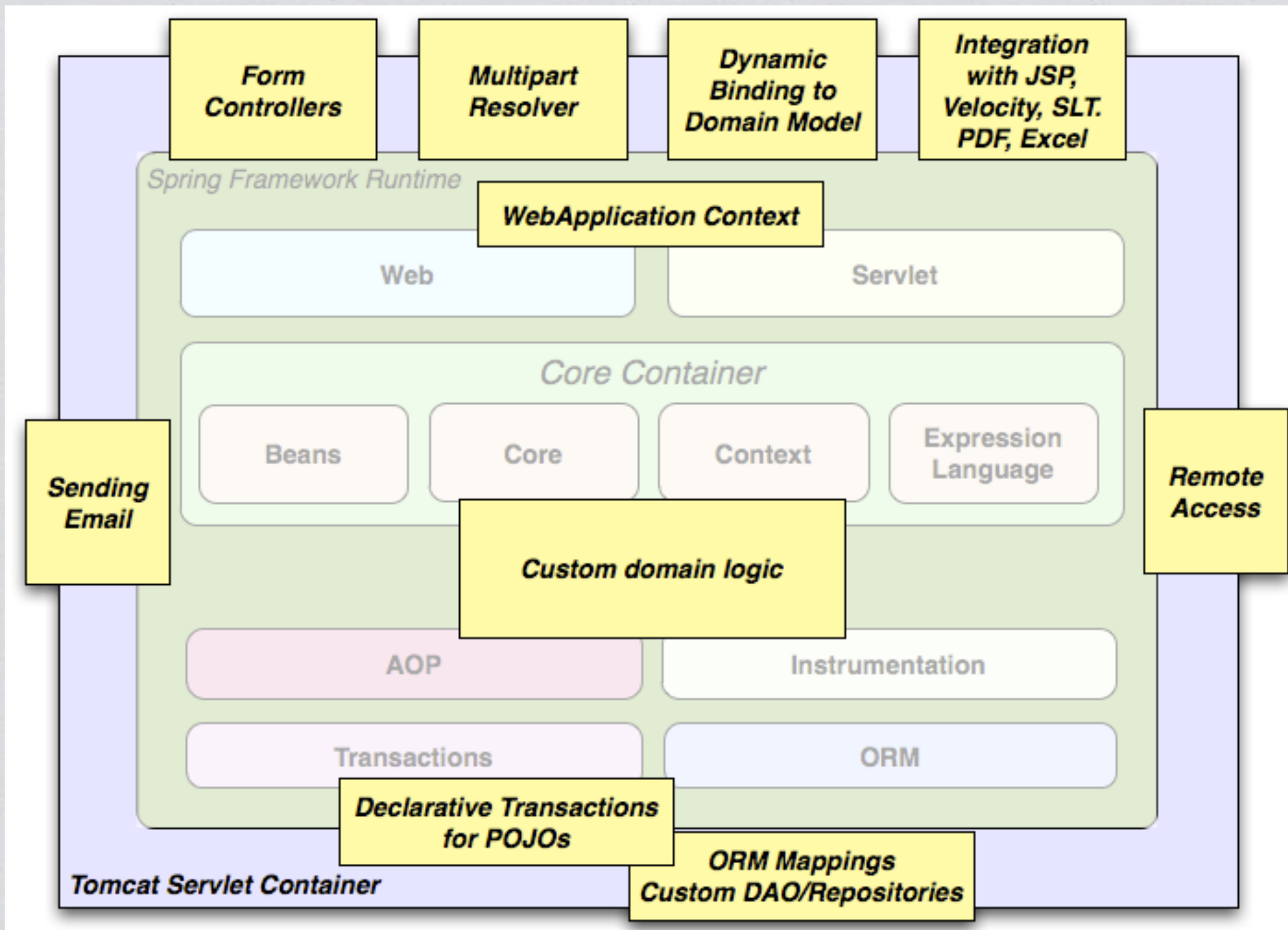
MVC



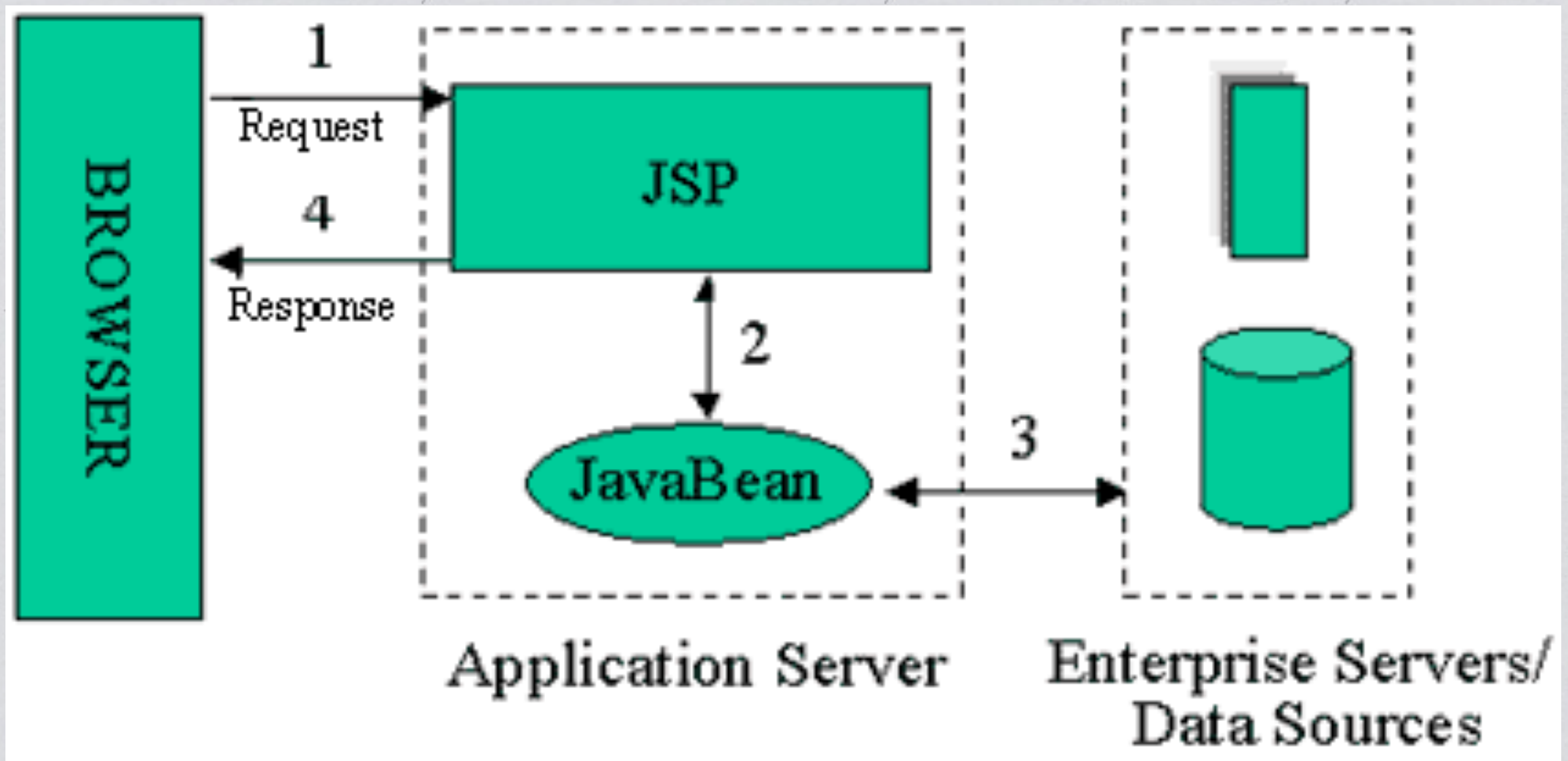
Spring Framework



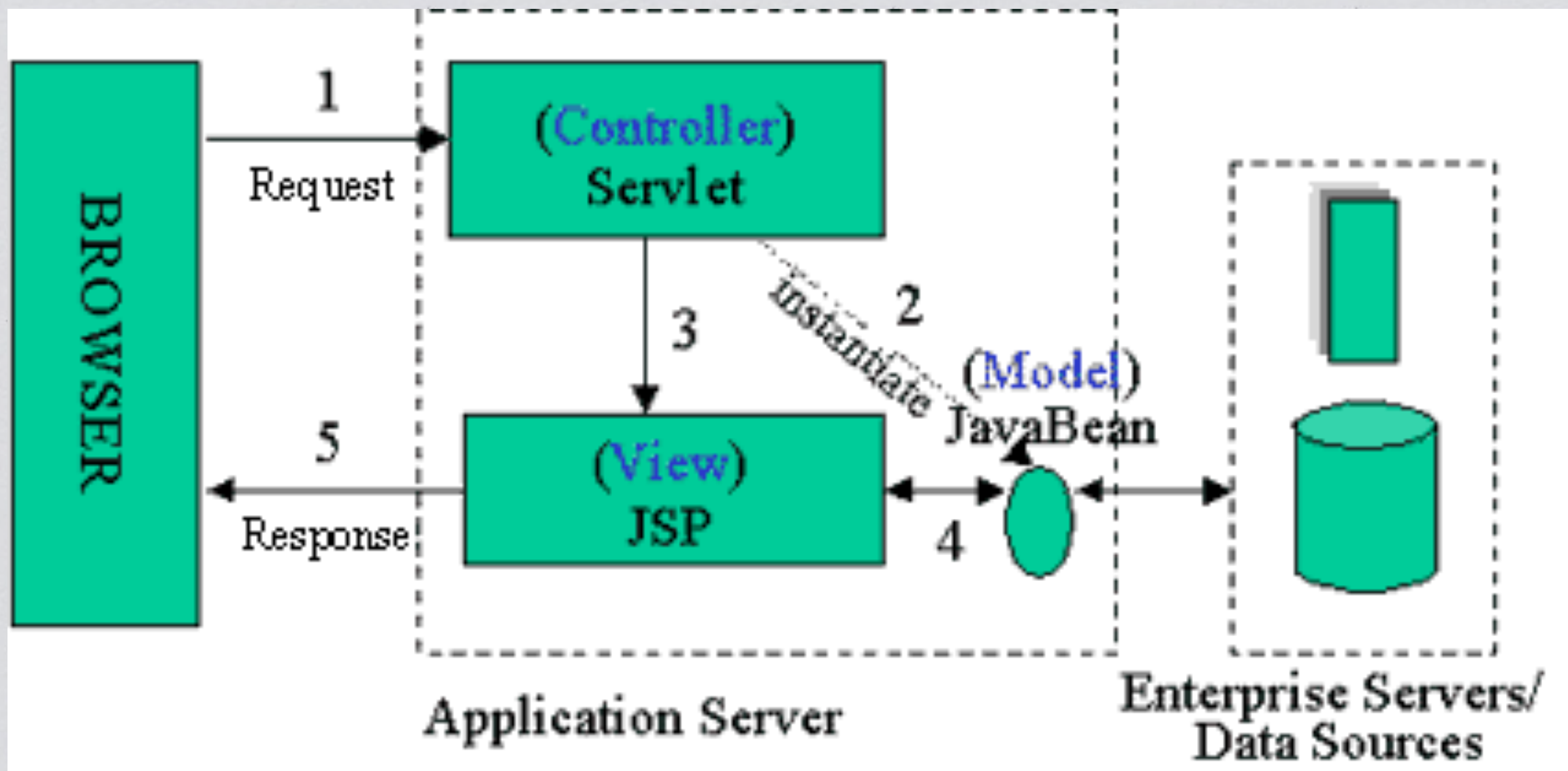
Spring Web Application



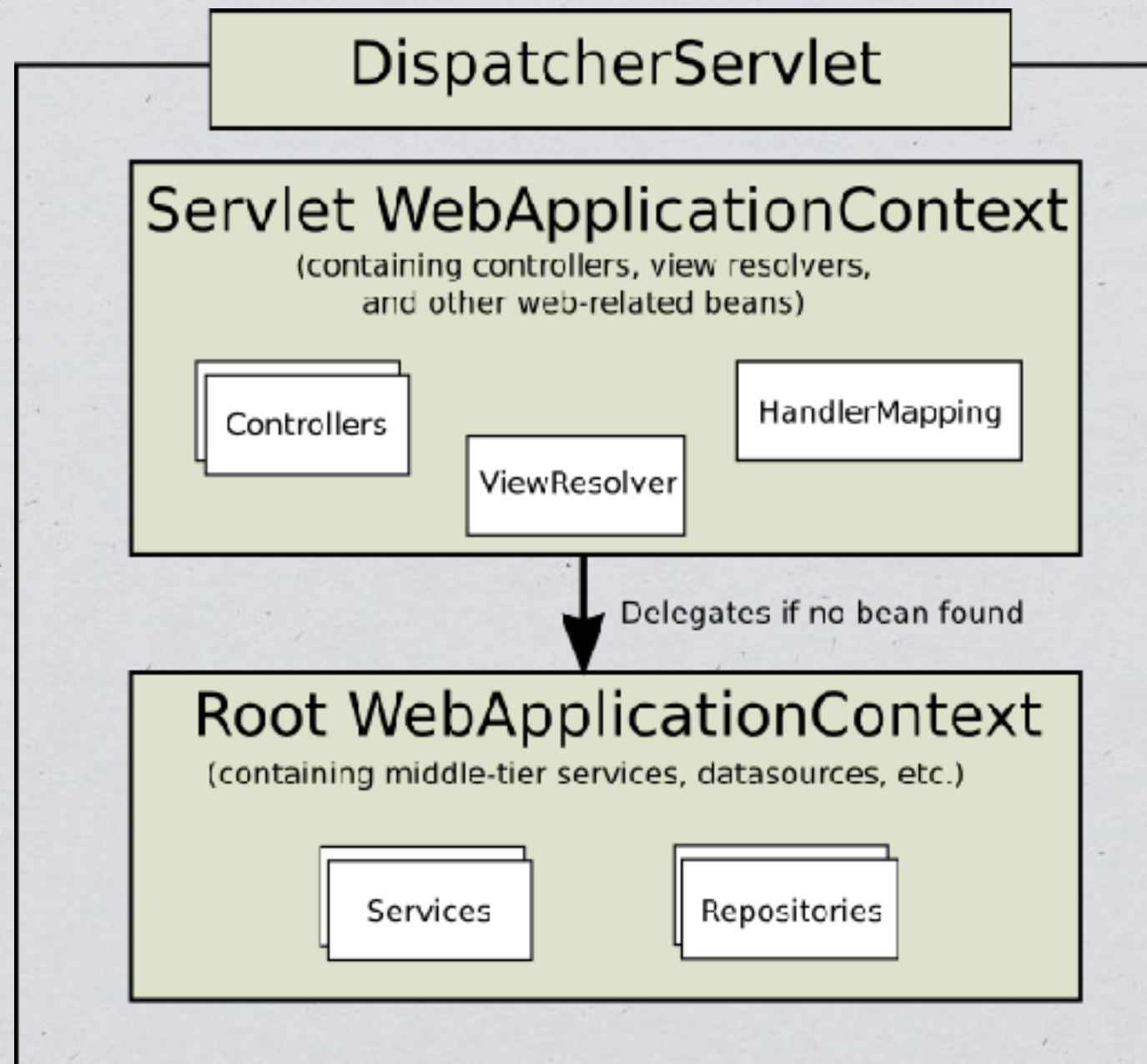
MODEL I



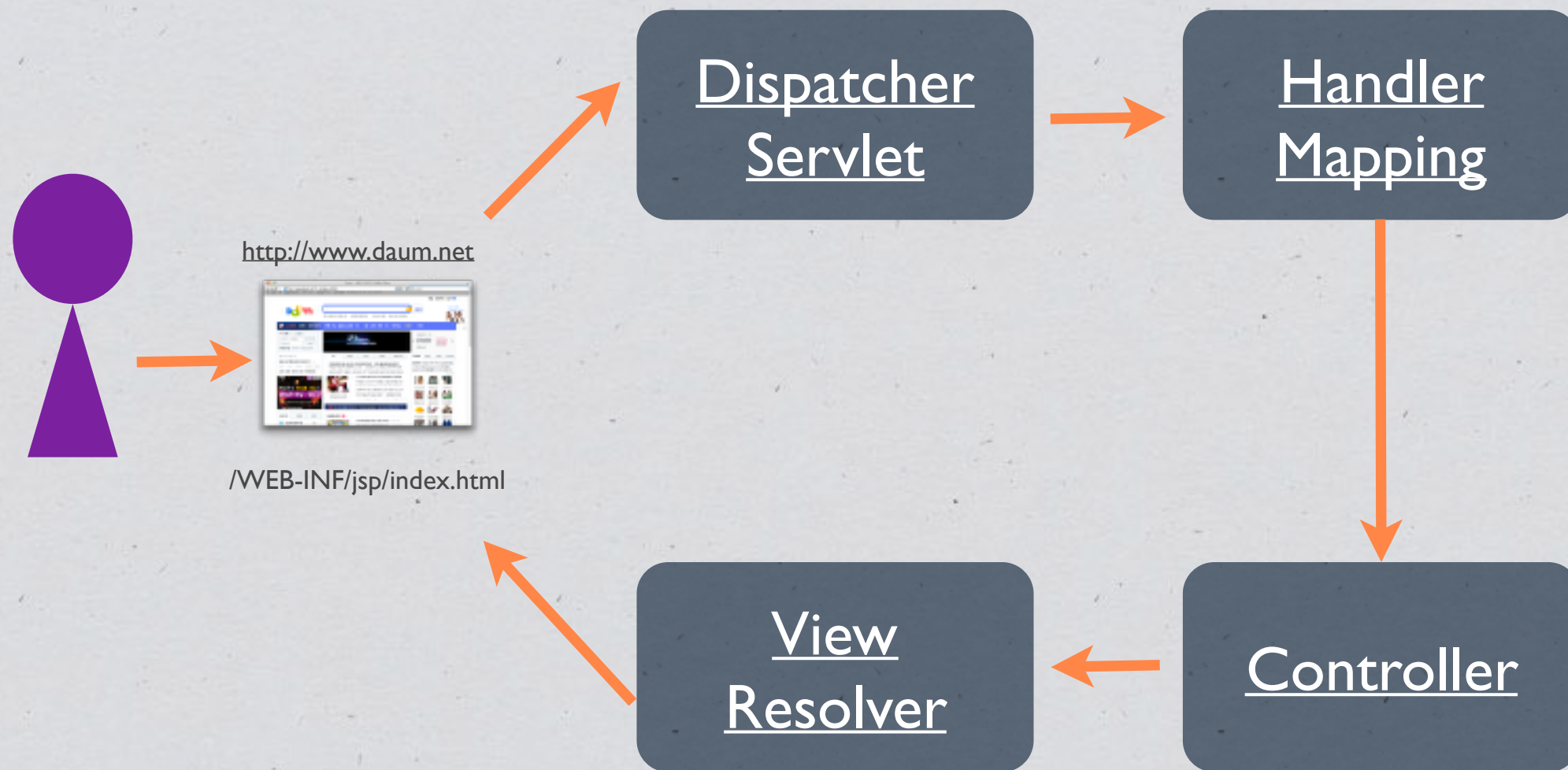
MODEL II



Spring MVC



Spring MVC LifeCycle



Handler Mapping

- * BeanNameUrlHandlerMapping (Default)
- * ~~ControllerBeanNameHandlerMapping~~
- * ~~ControllerClassNameHandlerMapping~~
- * SimpleUrlHandlerMapping
- * RequestMethodHandlerMapping(Default) - 3.1
 - * DefaultAnnotationHandlerMapping (Deprecated)

Handler Adapter

- * SimpleServletHandlerAdapter
- * HttpRequestHandlerAdapter (Default)
- * SimpleControllerHandlerAdapter (Default)
- * RequestMappingHandlerAdapter(Default) - 3.1
 - * AnnotationMethodHandlerAdapter (Deprecated)

View Resolver

- * InternalResourceViewResolver
- * VelocityViewResolver
- * FreeMarkerViewResolver
- * ResourceBundleViewResolver
- * XmlViewResolver
- * ContentNegotiatingViewResolver
- * HandlerExceptionResolver
- * MultipartResolver

View

- * InternalResourceView
- * JstlView
- * RedirectView
- * VelocityView
- * FreeMarkerView
- * MarshallingView
- * AbstractExcelView
- * AbstractPdfView
- * MappingJacksonJsonView
- * AbstractAtomFeedView
- * AbstractRssFeedView
- * XsltView
- * TilesView
- * AbstractJasperReportsView

Interceptor

- * preHandle

- * postHandle

- * afterCompletion

<annotation-driven/>

Spring 3 style type conversion through a `ConversionService` instance in addition to the JavaBeans `PropertyEditors` used for Data Binding.

Support for formatting Number fields using the `@NumberFormat` annotation through the `ConversionService`.

Support for formatting Date, Calendar, Long, and Joda Time fields using the `@DateTimeFormat` annotation.

Support for validating `@Controller` inputs with `@Valid`, if a JSR-303 Provider is present on the classpath.

`HttpMessageConverter` support for `@RequestBody` method parameters and `@ResponseBody` method return values from `@RequestMapping` or `@ExceptionHandler` methods.

This is the complete list of `HttpMessageConverters` set up by `mvc:annotation-driven`:

`ByteArrayHttpMessageConverter` converts byte arrays.

`StringHttpMessageConverter` converts strings.

`ResourceHttpMessageConverter` converts to/from `org.springframework.core.io.Resource` for all media types.

`SourceHttpMessageConverter` converts to/from a `javax.xml.transform.Source`.

`FormHttpMessageConverter` converts form data to/from a `MultiValueMap<String, String>`.

`Jaxb2RootElementHttpMessageConverter` converts Java objects to/from XML — added if JAXB2 is present and

@RequestMapping

* Class Mapping + Method Mapping

```
@RequestMapping(path="/user/add", method=RequestMethod.GET)
```

```
@RequestMapping(path="/user/add", params="type=member")
```

```
@RequestMapping(path="/user/add", headers="content-type=text/*")
```


@Controller (Method Param)

- * HttpServletRequest
- * HttpServletResponse
- * HttpSession
- * @PathVariable

```
@RequestMapping("/member/{membercode}/order/{orderid}")
```

```
public String method(@PathVariable("membercode") String code, @PathVariable("orderid") int orderid)
```

- * @RequestParam

```
public String method(@RequestParam("id") int id)
```

```
public String method(@RequestParam(value="id", required=false, default="1") int id)
```


@Controller (Method Param)

* @CookieValue

```
public String method(@CookieValue("id") String id)
```

* Map, Model, ModelMap

```
public void method(ModelMap model) {
```

```
    User user = new User(1);
```

```
    model.addAttribute(user);
```

```
}
```

* @ModelAttribute

```
public void method(@ModelAttribute User user)
```


@Controller (Method Return)

- * void

- * String

View Name Or Redirect: Or Forward:

- * ModelAndView

- * Map, Model, ModelMap

- * View