

Spring Boot

Web Applications - Spring Web MVC

Marten Deinum

Spring Boot

Web Applications - Spring Web MVC

- Building Web Applications with Spring MVC
- Spring Framework - Spring MVC
- Spring Boot - Auto Configuration
- Spring Boot - Testing
- Spring Boot - Security

Spring Boot

Spring MVC



Pro Spring MVC with WebFlux

Web Development in Spring Framework 5
and Spring Boot 2

—
Second Edition

—
Marten Deinum
Iuliana Cosmina

Apress®

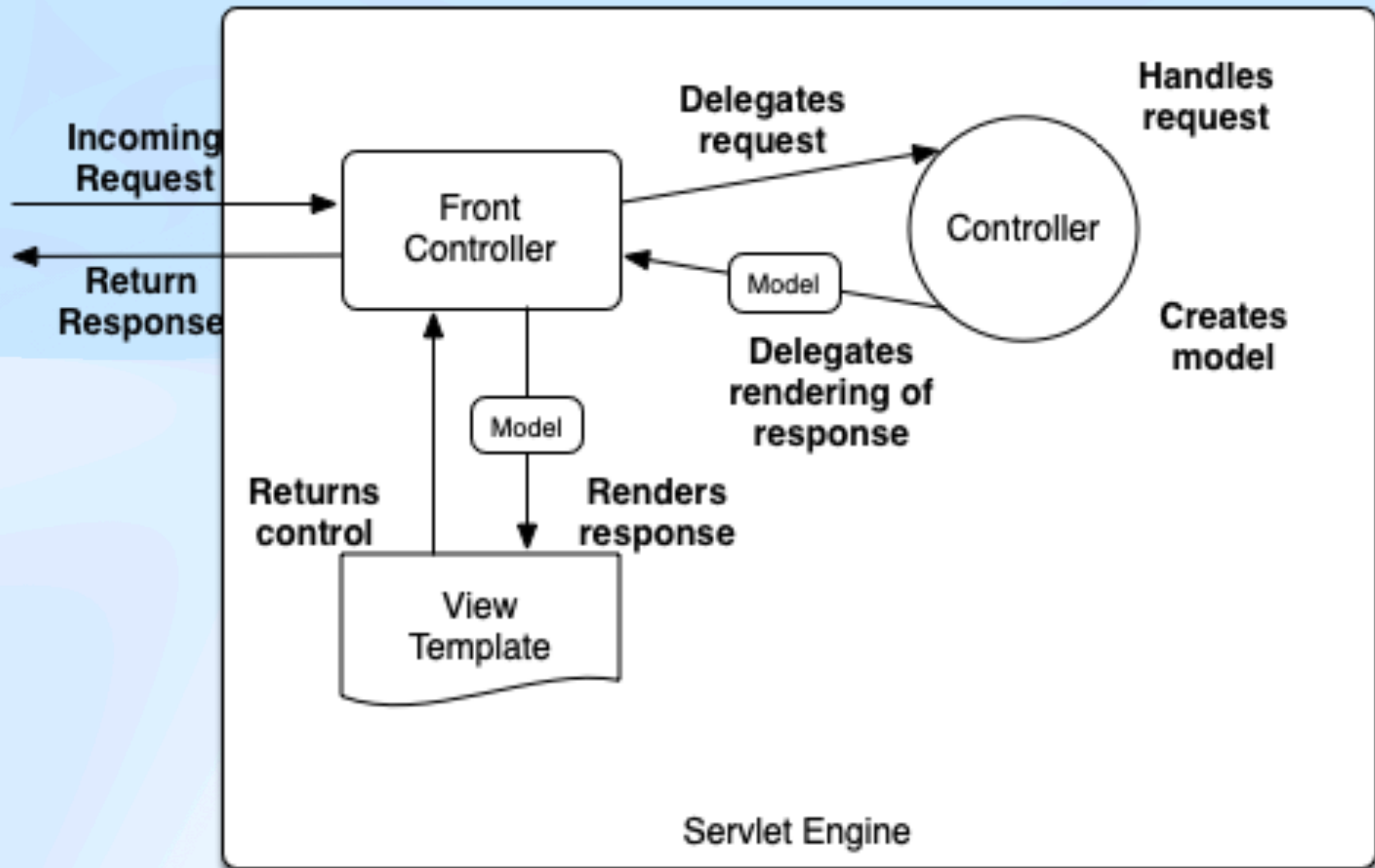
Spring Boot

Web Applications - Spring Web MVC

- Helps you write web applications
 - Build on top of the Servlet API
 - `DispatcherServlet`
- MVC Design pattern (Model 2)
 - Model
 - View
 - Controller
- Explicit concepts (although mostly used internally)

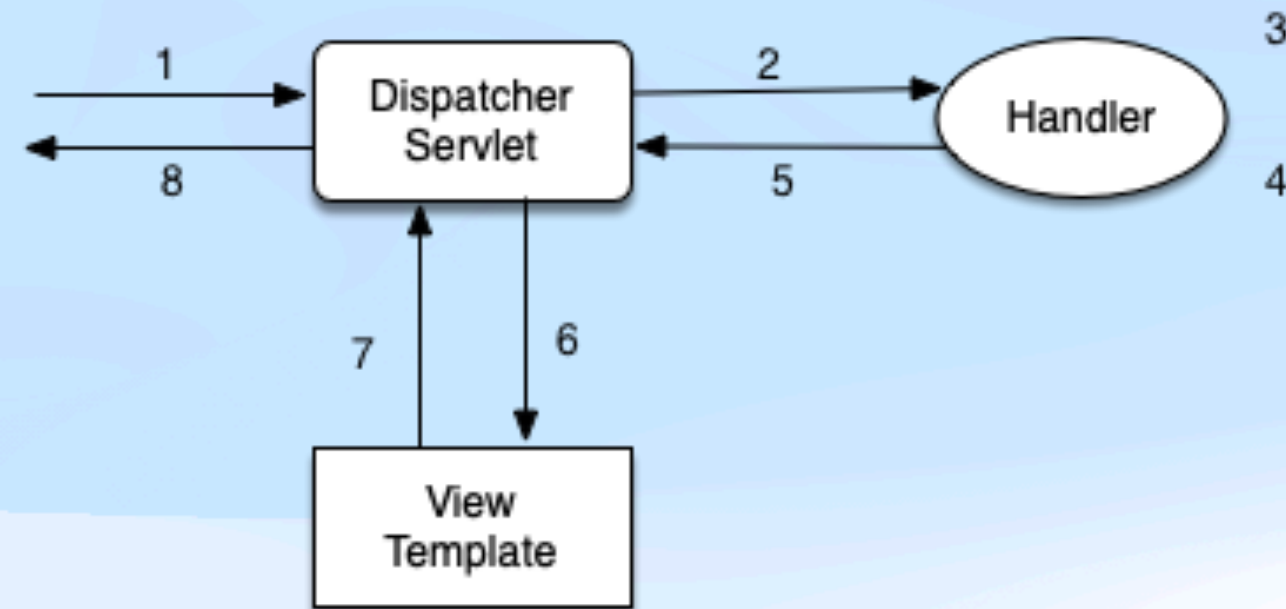
Spring Boot

Web Applications - Spring Web MVC



Spring Boot

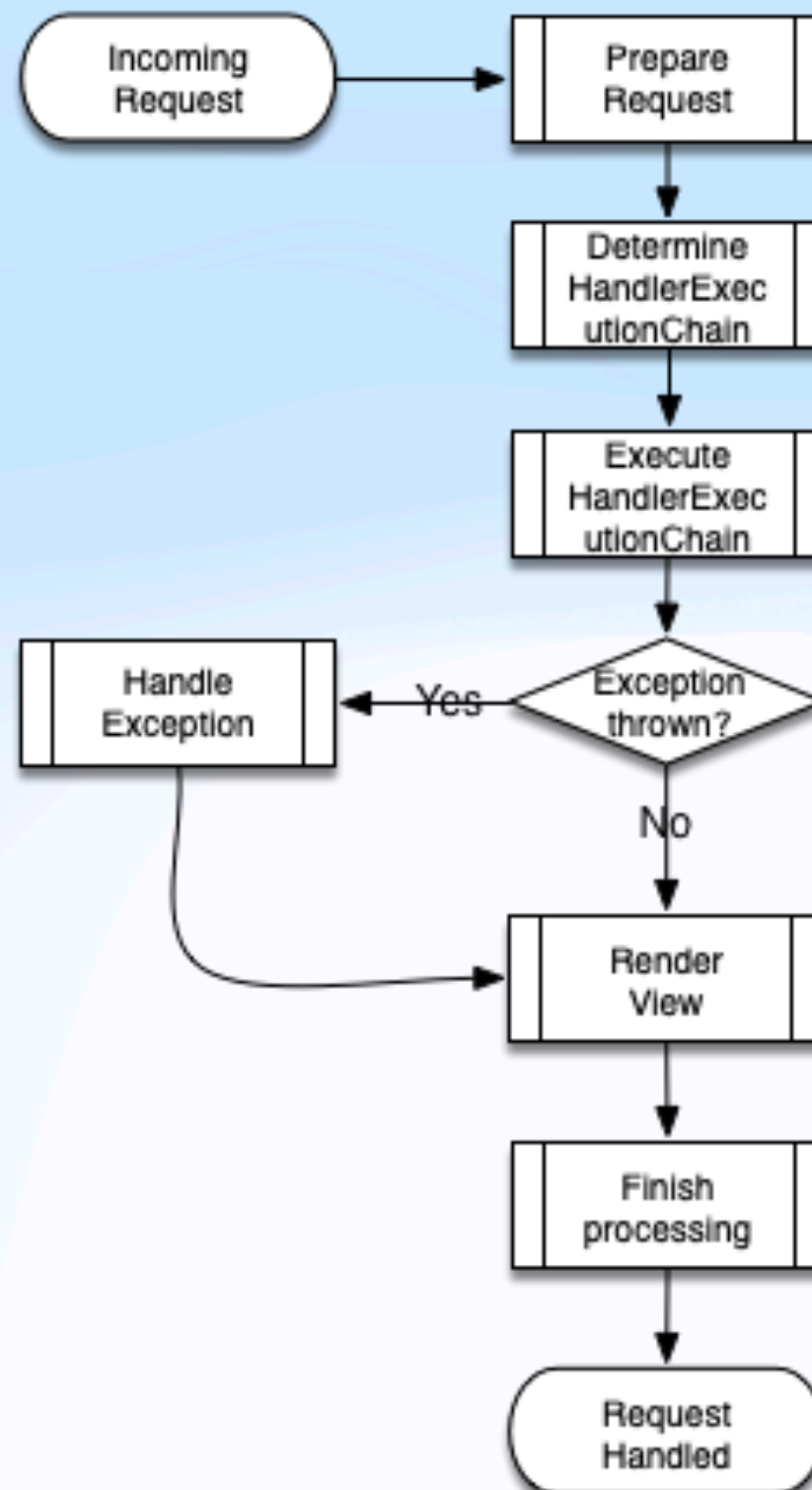
Web Applications - Spring Web MVC



1. The incoming request
2. Dispatching the request to the handler
3. Handling the request
4. Preparing the model and selecting the view
5. Returning `org.springframework.web.servlet.ModelAndView`
6. Rendering the `org.springframework.web.servlet.View` with the model
7. Returning control to the servlet
8. Returning the response to the client

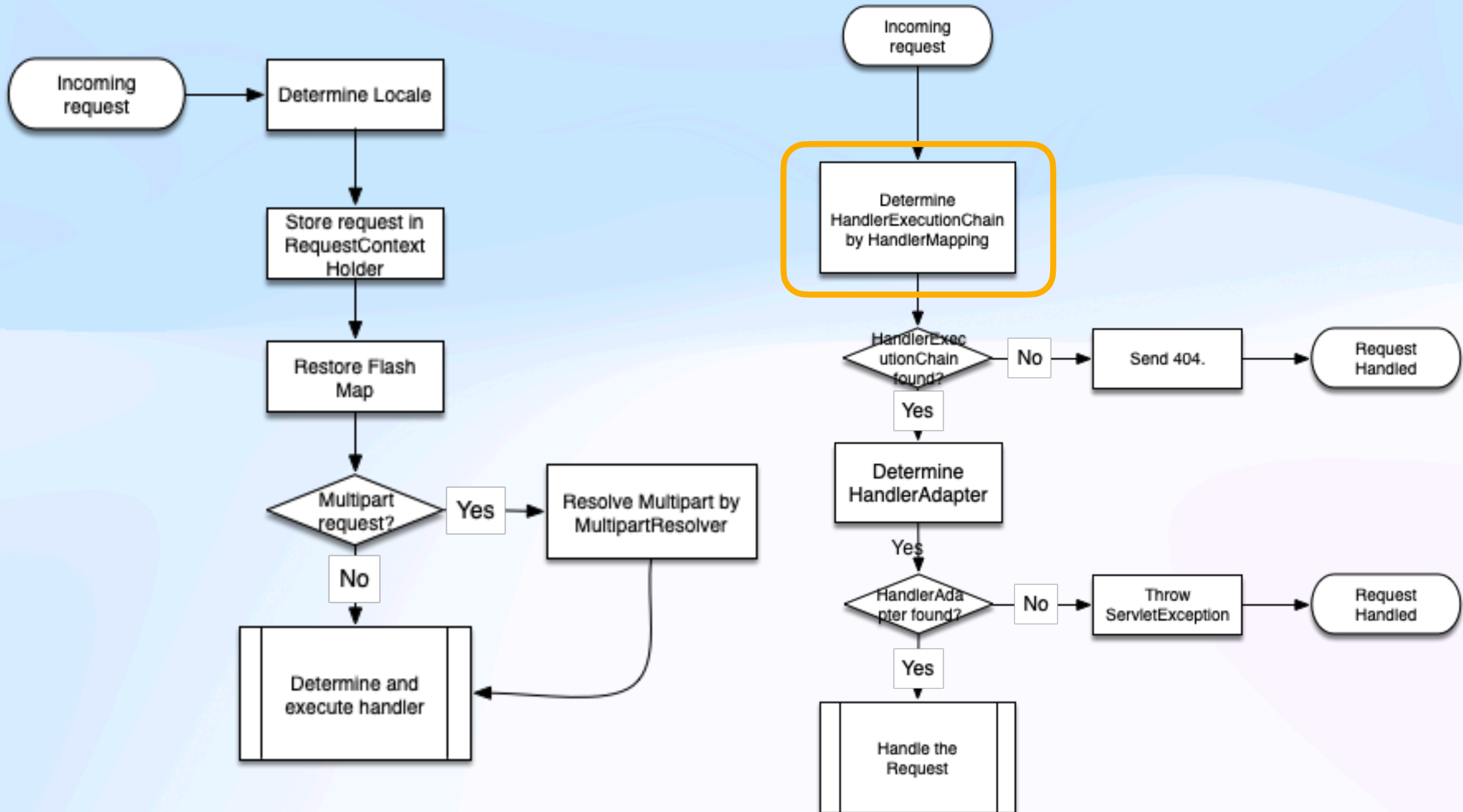
Spring Boot

Web Applications - Spring Web MVC



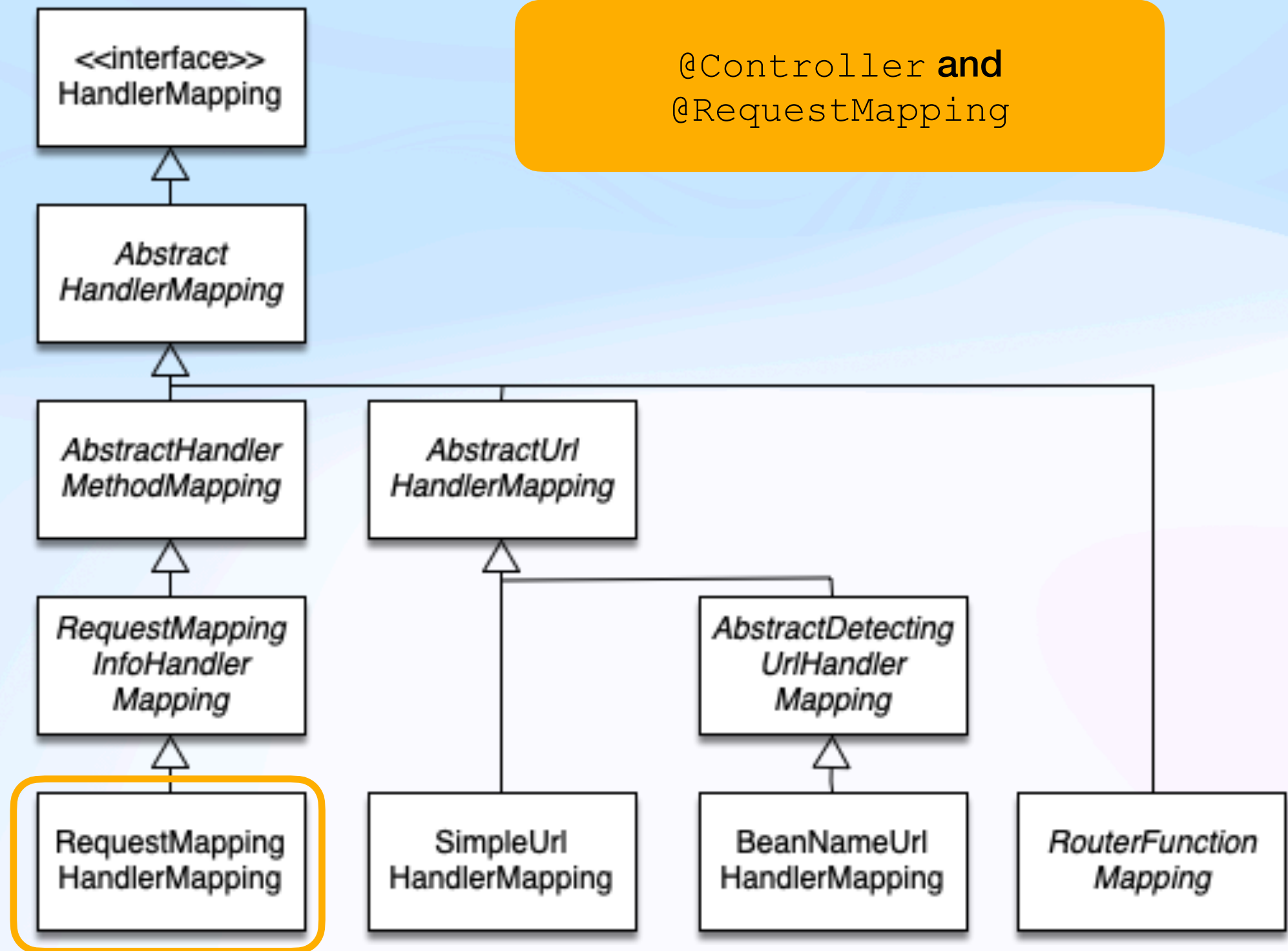
Spring Boot

Web Applications - Spring Web MVC



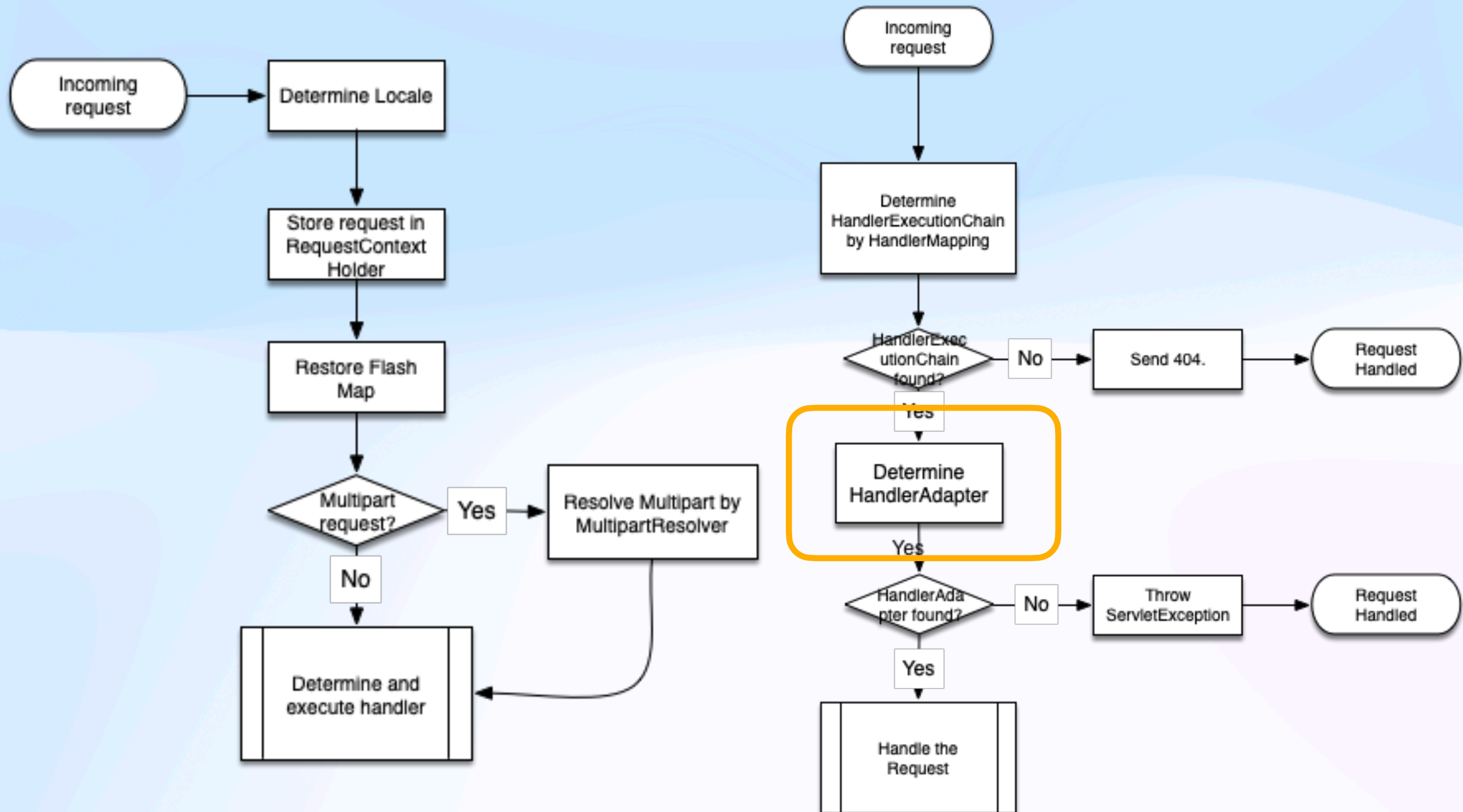
Spring Boot

Web Applications - Spring Web MVC



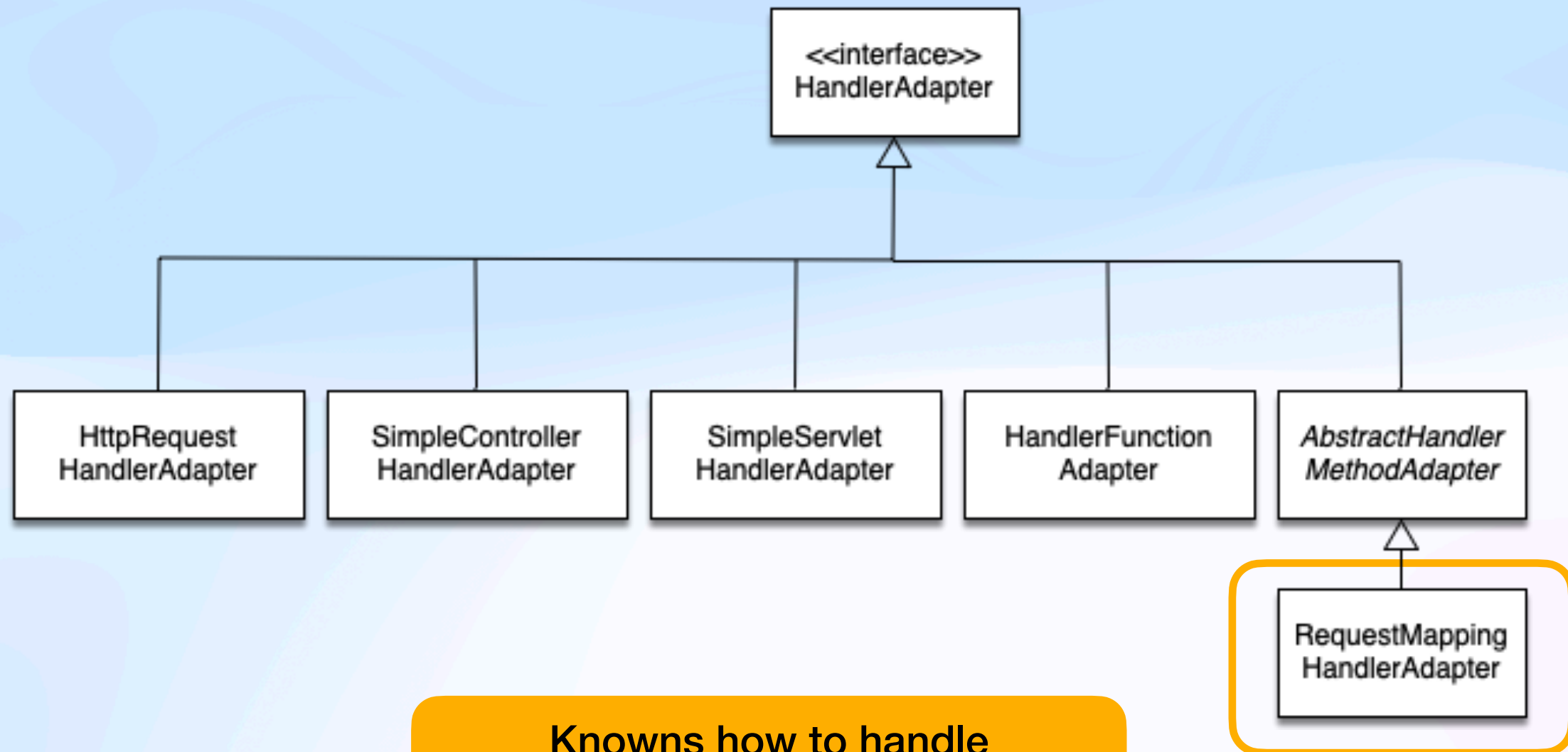
Spring Boot

Web Applications - Spring Web MVC



Spring Boot

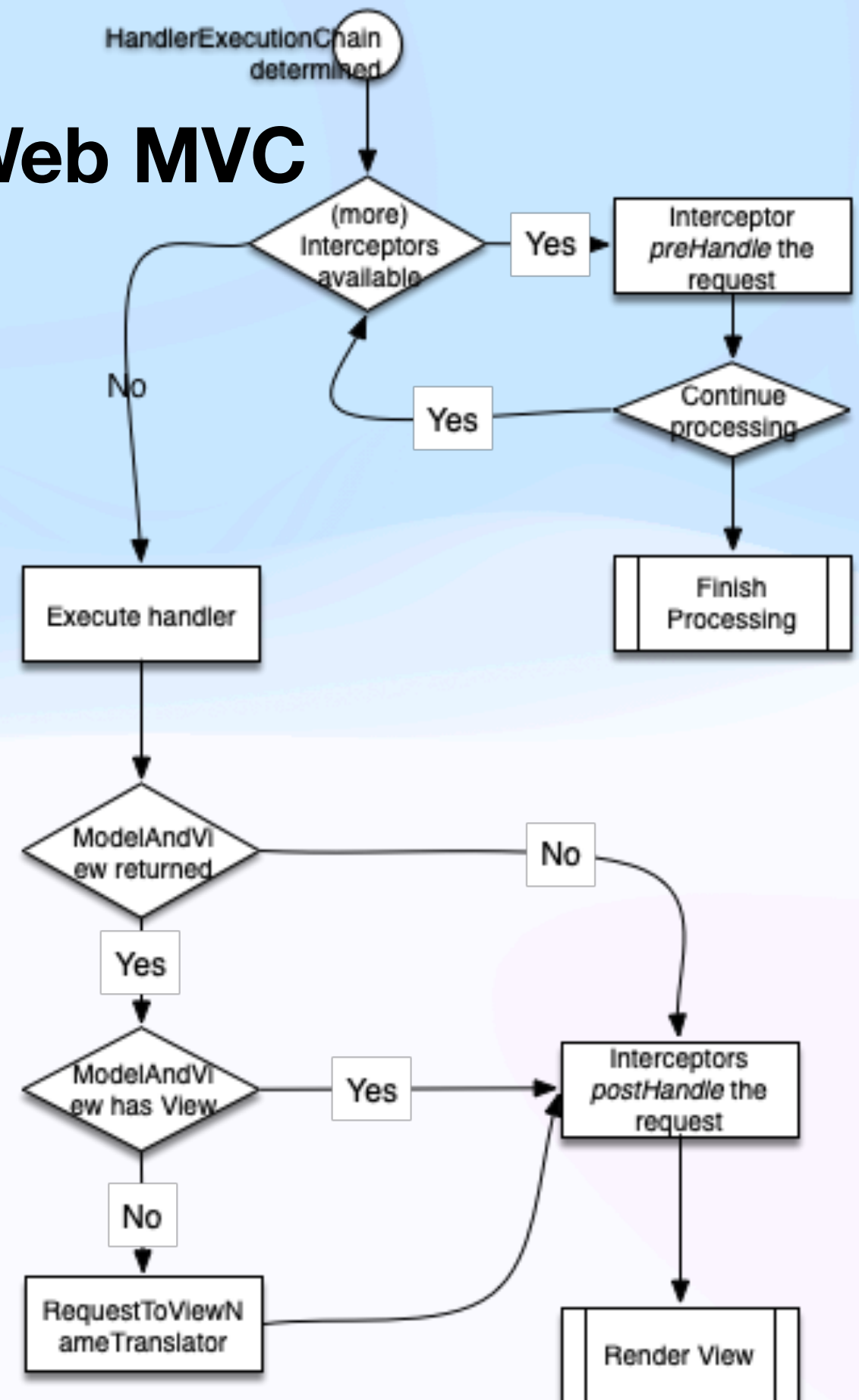
Web Applications - Spring Web MVC



Knows how to handle
`@RequestMapping` annotated
methods

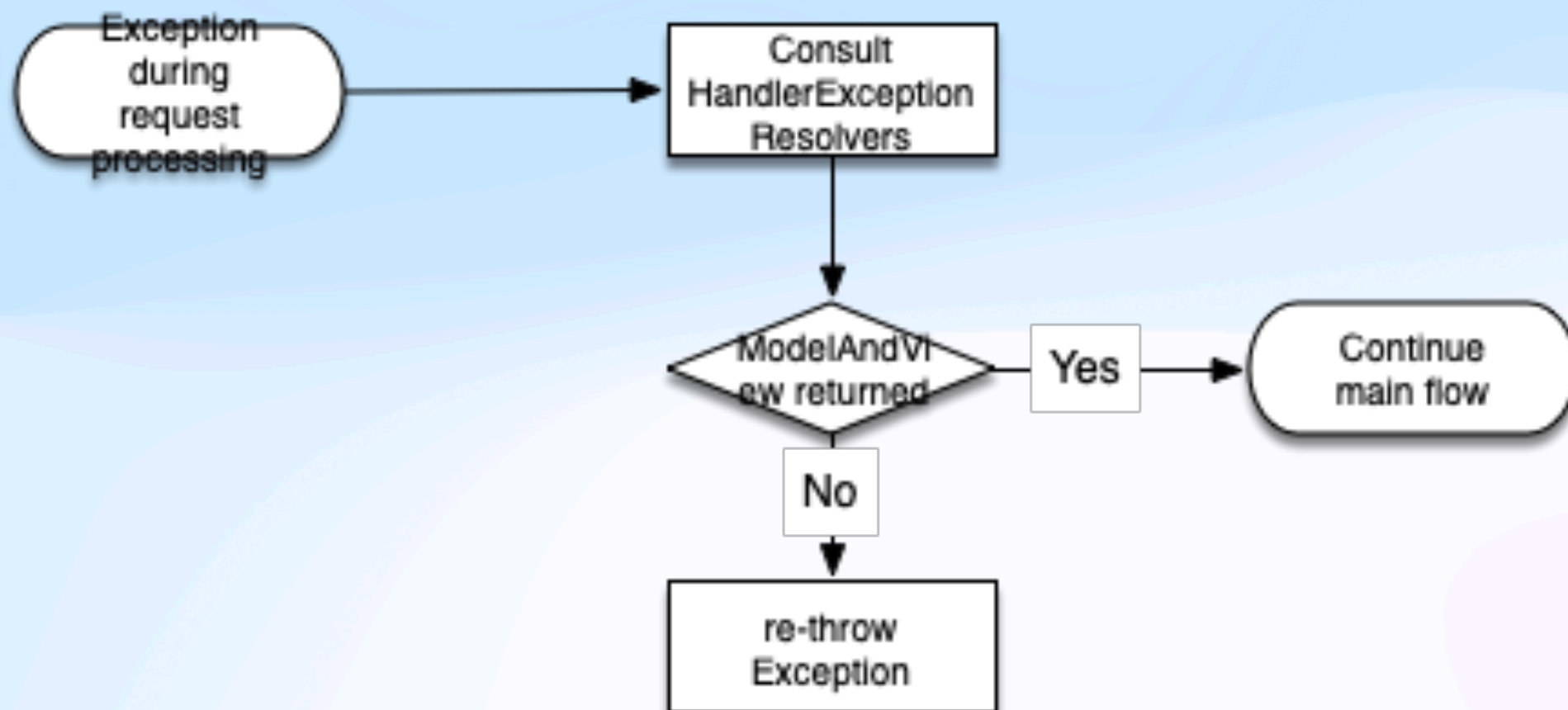
Spring Boot

Web Applications - Spring Web MVC



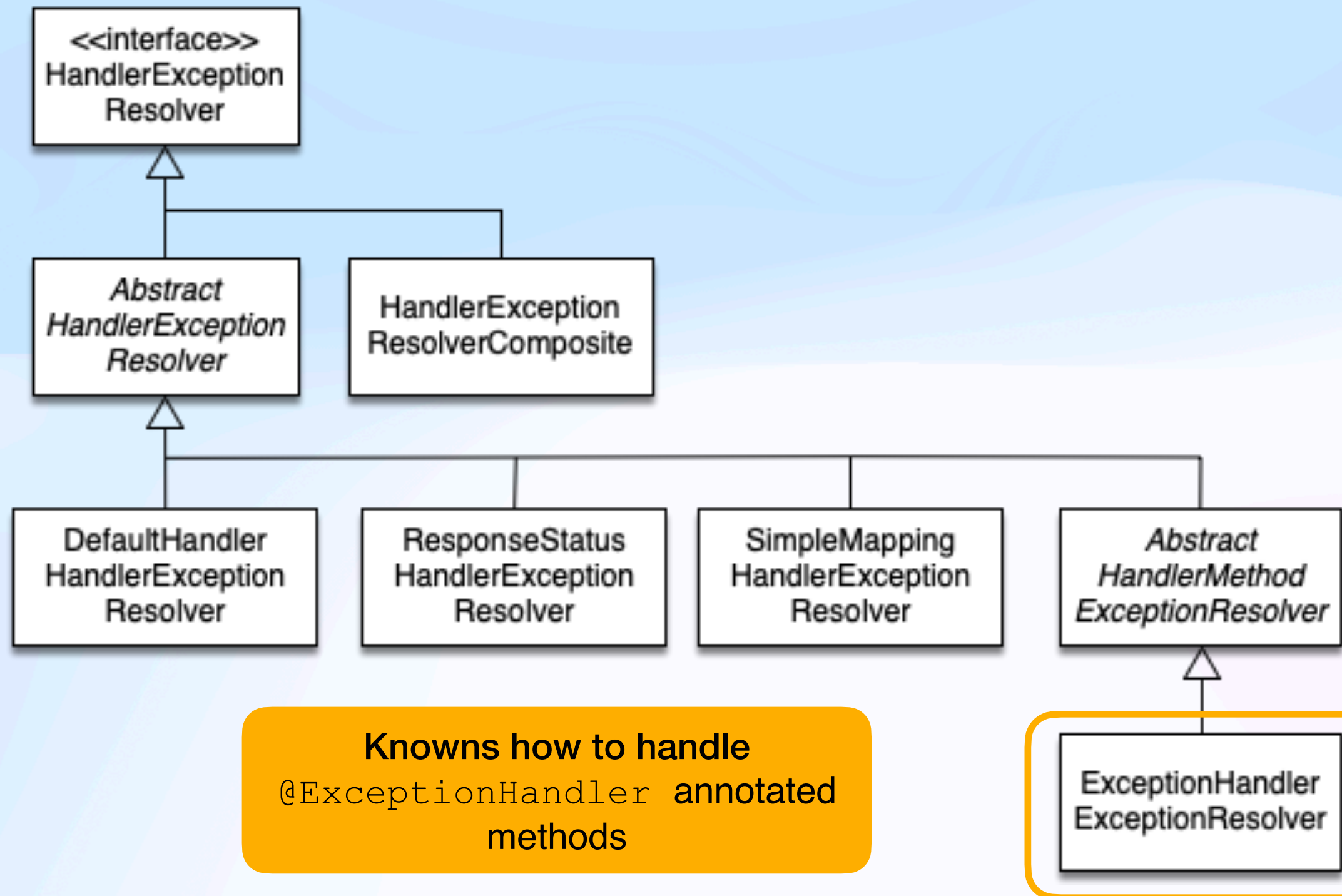
Spring Boot

Web Applications - Spring Web MVC



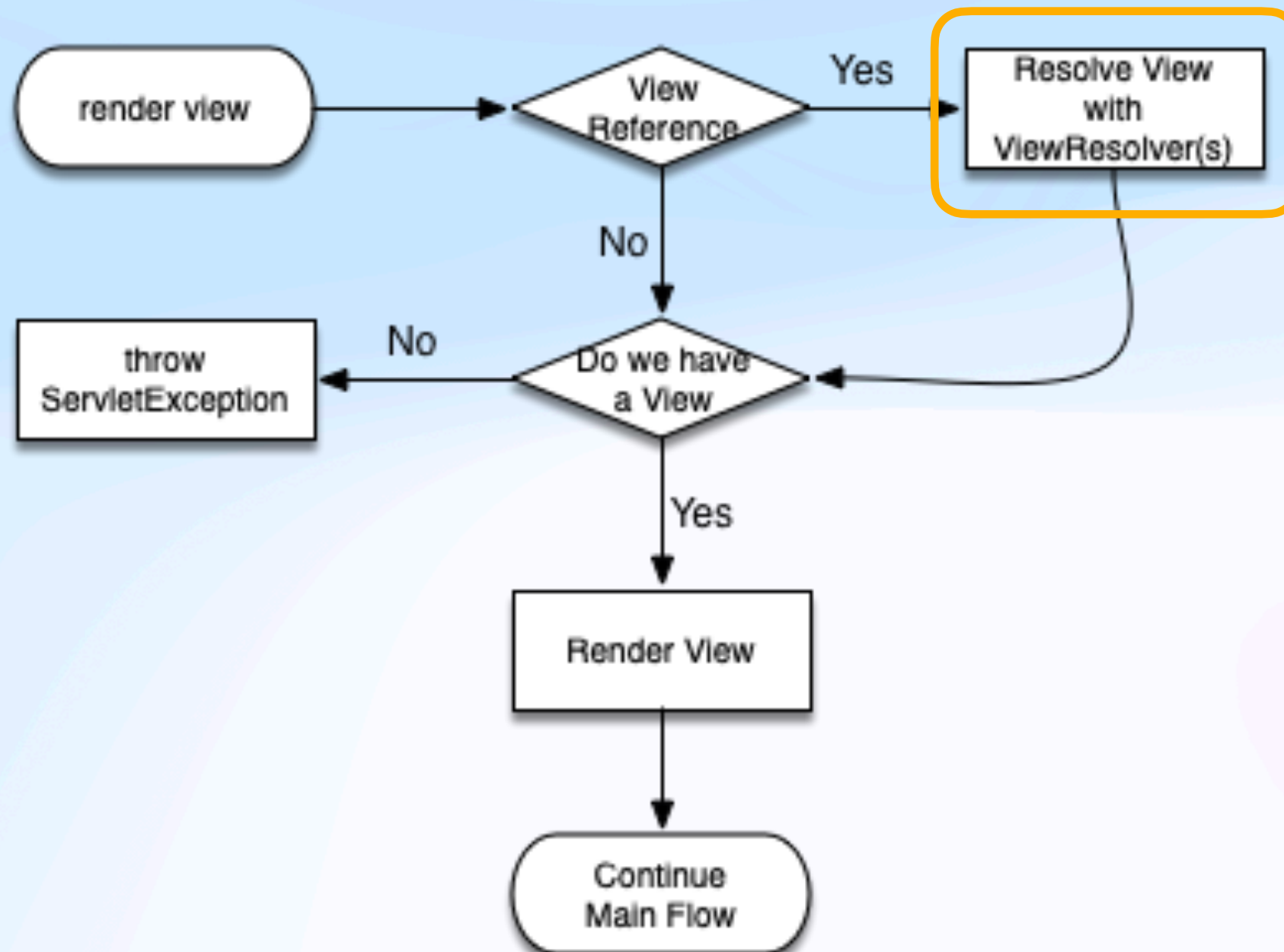
Spring Boot

Web Applications - Spring Web MVC



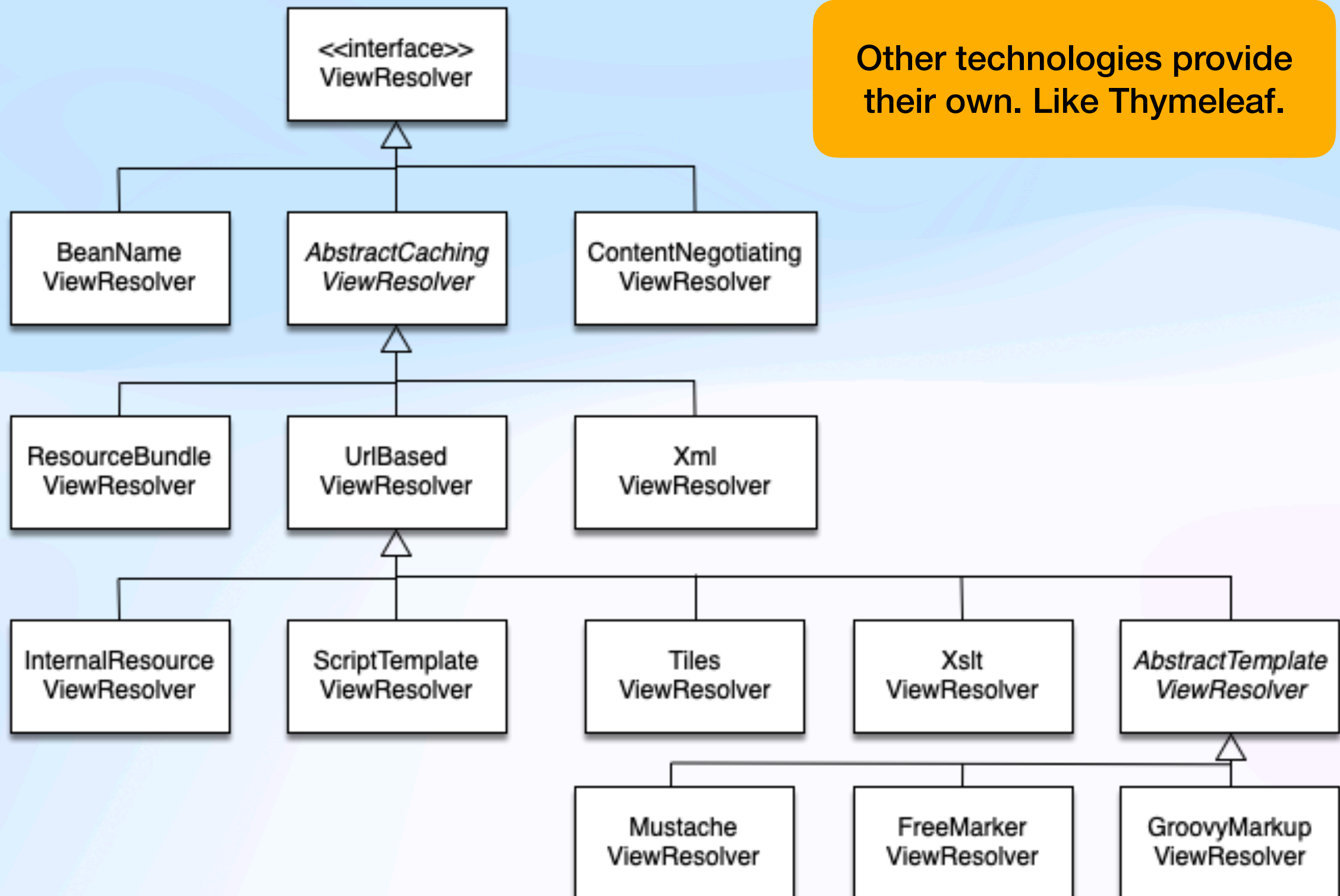
Spring Boot

Web Applications - Spring Web MVC



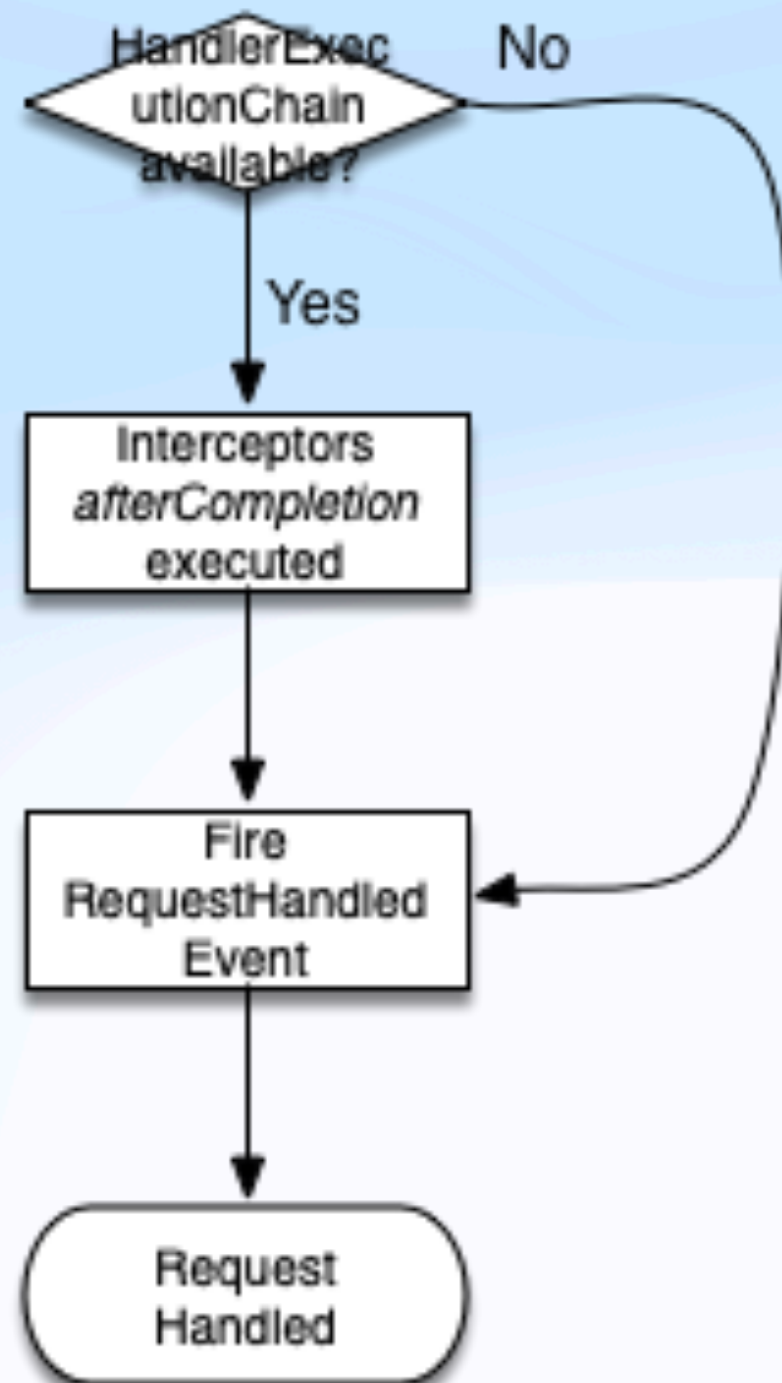
Spring Boot

Web Applications - Spring Web MVC



Spring Boot

Web Applications - Spring Web MVC



Spring Boot

Web Applications - Spring Web MVC

```
@Controller
@RequestMapping("/books")
public class LibraryController {

    . . .

    @GetMapping()
    public String list(Model model) {
    }
```

Used by the
RequestMappingHandlerMapping
to map /books to the list method.

Selected method passed to the
RequestMappingHandlerAdapter
to execute.

Spring Boot

Web Applications - Spring Web MVC

- Requires a `DispatcherServlet` to function
 - There are default settings for the `DispatcherServlet`
 - Better to use `@EnableWebMvc`
 - Allows for more control and configuration
- Annotation model for handlers is most commonly used nowadays
 - But other options exist (functional or historical)

Spring Boot

Spring Boot Auto-Configuration



Pro Spring MVC with WebFlux

Web Development in Spring Framework 5
and Spring Boot 2

—
Second Edition

—
Marten Deinum
Iuliana Cosmina

Apress®

Spring Boot

Spring MVC Auto-Configuration

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-web</artifactId>  
</dependency>
```

- Brings in necessary dependencies
 - Tomcat
 - Spring MVC
 - Jackson (JSON Library)

Spring Boot

Spring MVC Auto-Configuration

- Spring Boot will automatically configure
 - Tomcat (will start the servlet container)
 - Setup Spring MVC (register `@EnableWebMvc` and more)
 - Configure Jackson JSON library (for REST applications)
- When a supported view technology is detected it will configure that automatically as well
 - Thymeleaf, Freemarker, Mustache
- Additional configuration through properties or explicit Java configurations.

Spring Boot

Excercise:
32-spring-boot-web



Spring Boot

Spring Boot Testing



Pro Spring MVC with WebFlux

Web Development in Spring Framework 5
and Spring Boot 2

—
Second Edition

—
Marten Deinum
Iuliana Cosmina

Apress®

Spring Boot

Testing web applications

- To test web specific features there is `@WebMvcTest`
 - Loads a minimal application context with only web related classes.
 - Use `@MockitoBean` to mock dependencies needed by controllers
- Will also integrate Spring Security if detected on the classpath
- Auto configures Spring `Test MockMvc` by default
 - Mocked web environment for testing
 - including assertions for the result

Spring Boot

Testing web applications

- Could also use `@SpringBootTest` for full integration test
- When detected will also configure Selenium WebDriver

Spring Boot

Excercise (cont.):
32-spring-boot-web



Spring Boot Security



Pro Spring MVC with WebFlux

Web Development in Spring Framework 5
and Spring Boot 2

—
Second Edition

—
Marten Deinum
Iuliana Cosmina

Apress®

Spring Boot

Securing Web Applications - Core Concepts

- Principal
 - User that performs an action
- Authentication
 - Establish that a principals' credentials are valid
- Authorization
 - Deciding if a principal is allowed to perform the action
- Authority
 - Permission / credential which enables access (roles, location etc.)
- Secured Resource
 - The actual resource

Spring Boot

Securing Web Applications - Authentication

- Many different authentication mechanisms
 - Basic Auth, Digest, Login Form, Certificates, OAuth 2.0, etc.
- Many different storage options for credentials and authorities
 - JDBC, LDAP, AD, etc.

Spring Boot

Securing Web Applications - Authorization

- Authorization depends on authentication
 - If a user cannot be identified it makes no sense to check authorizations
- Authorization determines if you have the proper authority
- Authorization often based on roles

Spring Boot

Securing Web Applications - Spring Security

- Separation of Concerns
 - Decouple business logic from security concerns
 - Authentication and Authorization are decoupled
- Flexible and Extensible
 - Authentication: Basic, Form, X.509, OAuth, SSO, etc.
 - Storage: LDAP, DB, Files, custom daos, etc.
 - Highly customizable
- Can be used on any Spring Project (not limited to web apps)

Spring Boot

Excercise (cont.):
32-spring-boot-web

