webmvc.jar
(as dependency)

SpringServletContextInitializer
(register DispatcherServlet)

WEB
APPLICATION
(deploy)

web.xml

(metadata)

ServletContext
(object per app)

Tomcat   UNDERTOW   j:

servlet containers

ServletContextListener Approach:
works for only programmatic registration for web application components within
the application itself.

For a third-parties if the ship the implementation of ServletContextListener
aspart of their dependencies, the servlet container would not call the Listener
class

web.xml (project) <web-app metadata-complete="true">

META-INF/services/jakarta.servlet.ServletContainerInitializer
[delcare all the implementation classes of ServletContainerInitializer]

interface ServletContainerInitializer {
  void onStartup(Set<Class<?>> handlers, ServletContext);
}

webmvc.jar
|-META-INF
  |-services
    |-jakarta.servlet.ServletContainerInitializer (file)

@Handlers({SpringRegistration.class})
|-SpringServletContainerInitializer
  |-onStartup(Set<Class<?>> handlers, ServletContext context);

There can be various different initializations/programmatic registrations of
different components might be there within an application
- initialization/registration = Handler class

interface SpringRegistration {
  onStartup(ServletContext);
}