- **Spring MVC (module view controller)** is a complete HTTP oriented MVC framework managed by the Spring Framework and based in Servlets. It would be equivalent to JSF in the JavaEE stack. The most popular elements in it are classes annotated with @Controller, where you implement methods you can access using different HTTP requests. It has an equivalent @RestController to implement REST-based APIs.
- **Spring boot** is a utility for setting up applications quickly, offering an out of the box configuration in order to build Spring-powered applications. As you may know, Spring integrates a wide range of different modules under its umbrella, as *spring-core*, *spring-data*, *spring-web* (which includes Spring MVC, by the way) and so on. With this tool you can tell Spring how many of them to use and you'll get a fast setup for them (you are allowed to change it by yourself later on).

So, Spring MVC is a framework to be used in web applications and Spring Boot is a Spring based *production-ready* project initializer.

Spring Boot and *Spring MVC* are not comparable or mutually exclusive. If you want to do web application development using Spring, you would use *Spring MVC* anyway. Your question then becomes whether to use *Spring Boot* or not.

For developing common Spring applications or starting to learn Spring, I think using *Spring Boot* would be recommended. It considerably eases the job, is production ready and is rapidly being widely adopted.

In simple term it can be stated as:

Spring boot = Spring MVC + Auto Configuration(Don't need to write spring.xml file for configurations) + Server(You can have embedded Tomcat, Netty, Jetty server).