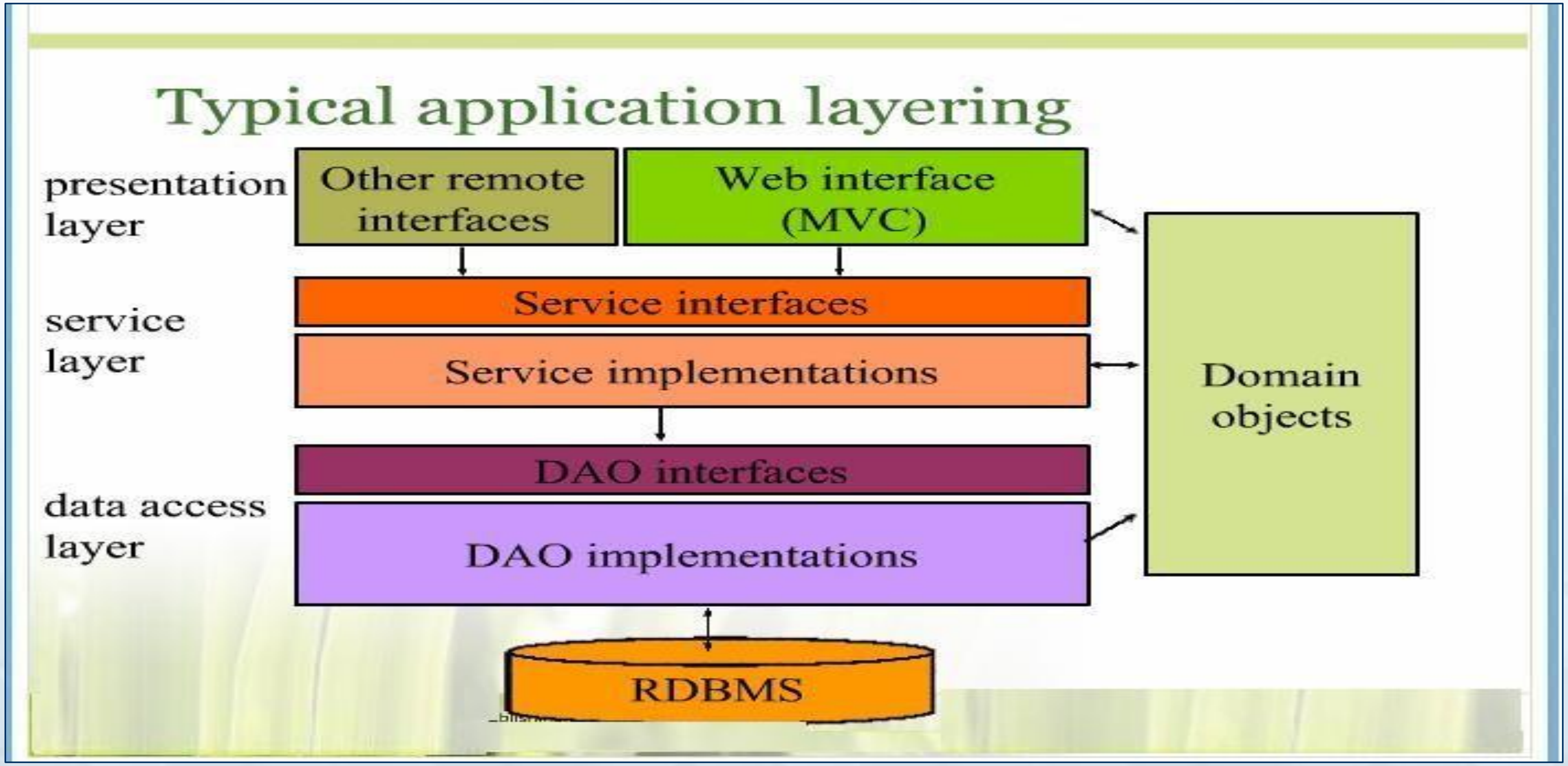
Spring Boot

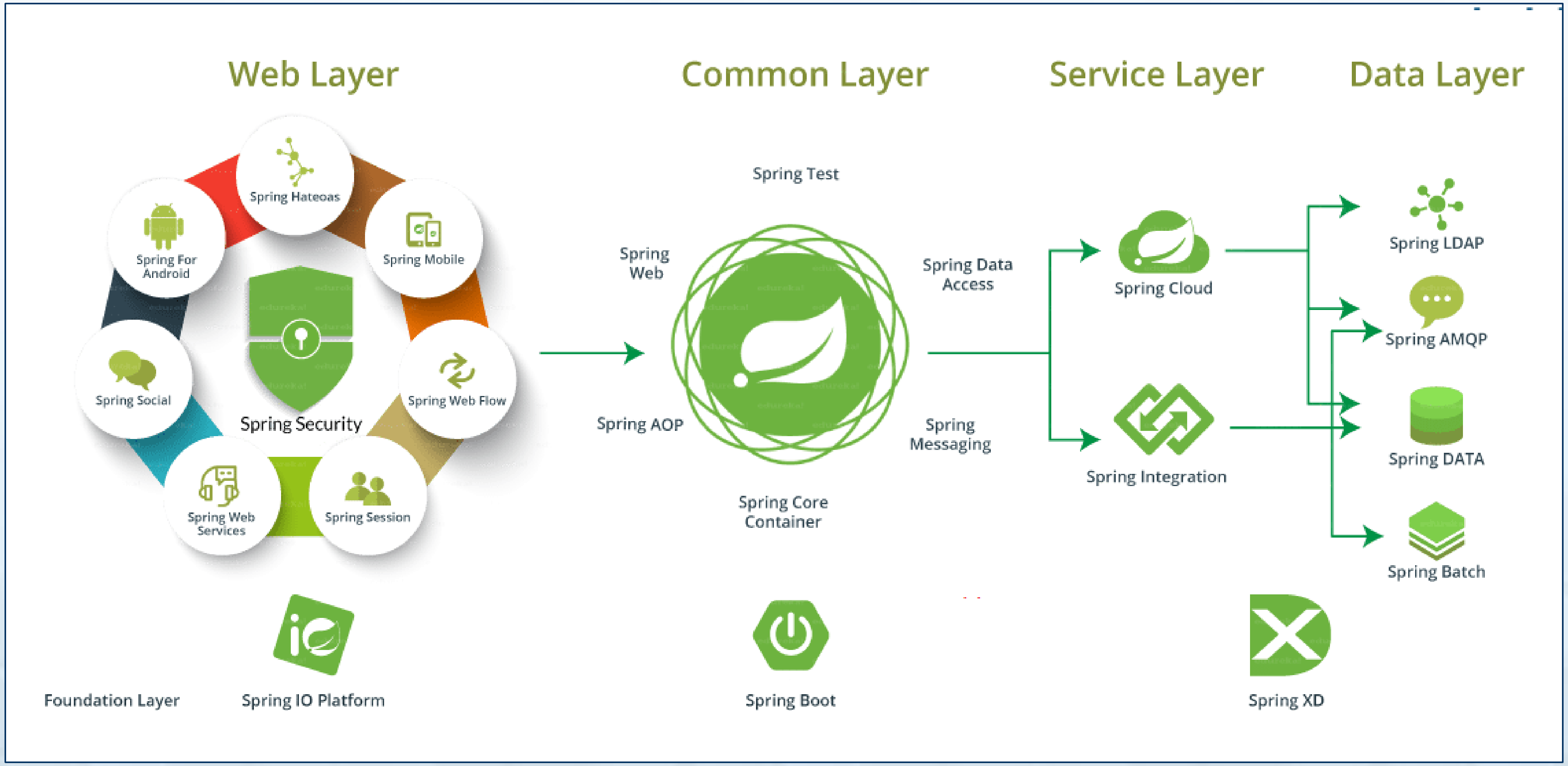
www.hexaware.com | © 2019 Hexaware Technologies limited. All rights reserved.

Session Objective

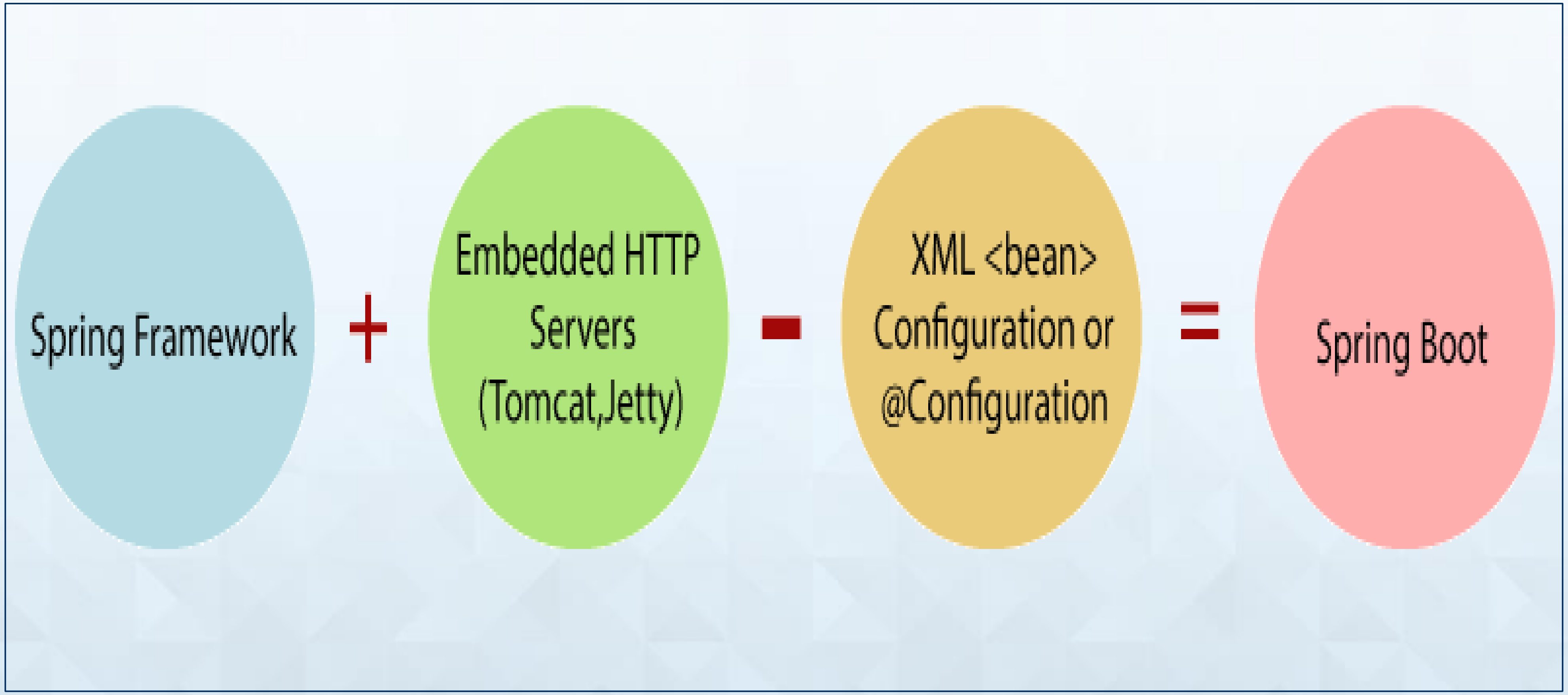
Typical application layer



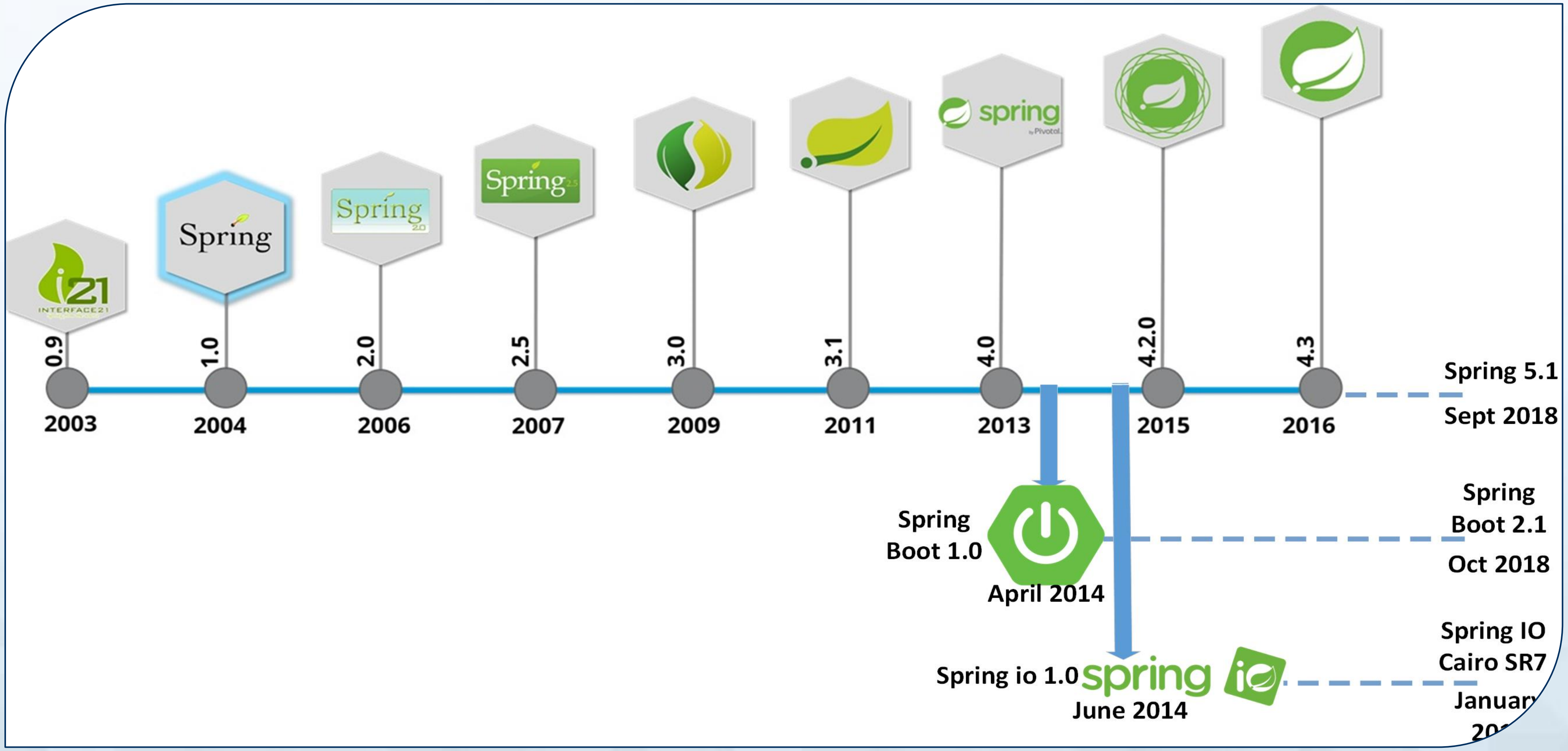
Spring


Web Layer
Common Layer
Service Layer
Data Layer
Spring Hateoas
Spring For Android
Spring Mobile
Spring Social
Spring Security
Spring Web Flow
Spring Web Services
Spring Session
Spring Test
Spring Web
Spring AOP
Spring Data Access
Spring Messaging
Spring Core Container
Spring Cloud
Spring Integration
Spring LDAP
Spring AMQP
Spring DATA
Spring Batch
Foundation Layer
Spring IO Platform
Spring Boot
Spring XD

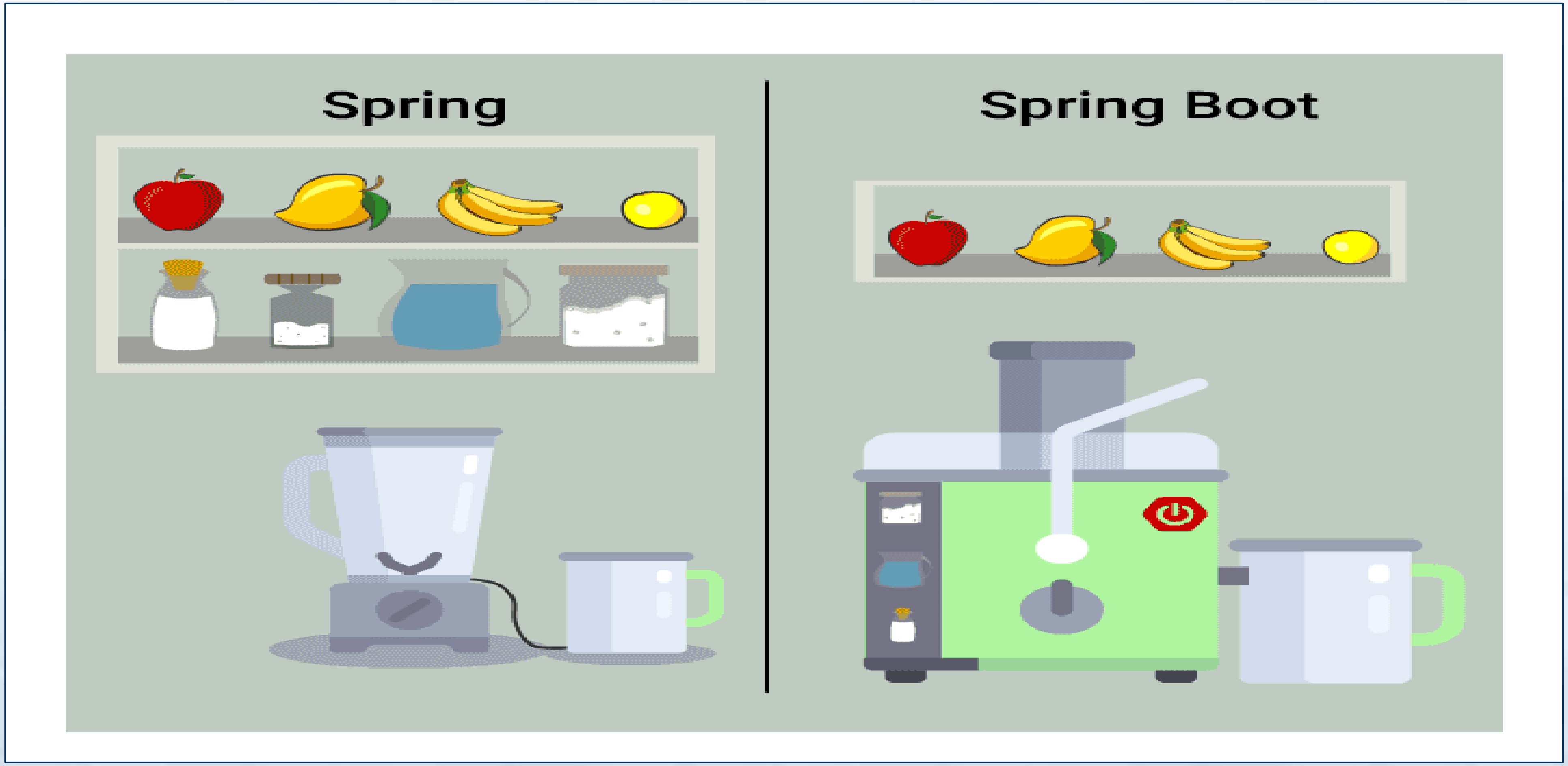
Spring Boot


Spring Framework
+
Embedded HTTP Servers (Tomcat,Jetty)
-
XML <bean> Configuration or @Configuration
=
Spring Boot

Spring Version



Spring Vs Spring Boot



Spring Boot

- Spring Boot is a project/module built on top of Spring Framework.
- Spring Boot provide RAD feature to Spring Framework.
- Spring Frameworks xml Configuration is removed from Spring Boot.
- Spring Boot added his power with embedded server(Tomcat and Jetty)
- Spring Boot contains powerful database transaction management capabilities.
- Spring Boot simplifies integration with other Java frameworks like JPA/Hibernate ORM, Struts, etc.
- Spring Boot reduces the cost and development time of the application.

Spring Boot Starter

- Dependency management is a critical aspects of any complex project.
- Spring Boot starters were built to address this problem.
- Starter POMs are a set of convenient dependency descriptors that you can include in your application.

Spring Boot Starter

We have more than 30 Boot starters available example :

- **The Web Starter**

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

- **The Test Starter**

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

Spring Boot Starter

The Data JPA Starter

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

The Mail Starter:

```
 <dependency>
    <groupId>org.subethamail</groupId>
    <artifactId>subethasmtp</artifactId>
</dependency>
```

Spring Boot Starter Parent

- The spring-boot-starter-parent is a project starter.
- It provides default configurations for our applications. It is used internally by all dependencies.
- All Spring Boot projects use spring-boot-starter-parent as a parent in pom.xml file.

```
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.4.0.RELEASE</version>
</parent>
```

Spring Boot Starter- parent

- Parent Poms allow us to manage the following things for multiple child projects and modules:

- Configuration: It allows us to maintain consistency of Java Version and other related properties.
- Dependency Management: It controls the versions of dependencies to avoid conflict.
- Source encoding
- Default Java Version
- Resource filtering
- It also controls the default plugin configuration.

Spring-Boot initial setup.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
                                        xsi:schemaLocatio
POM/4.0.0 http://maven.apache.org/xsd/maven-4.(
    <parent>
        <groupId>org.springframework.boot</groupId
        <artifactId>spring-boot-starter-parent</artifactId
        <version>1.5.1.RELEASE</version>
    </parent>

    <!-- Specify java version -->
    <properties>
        <java.version>1.8</java.version>
    </properties>
```

Spring-boot-starter-parent dependency inherits from spring-boot-dependencies

Spring-Boot initial setup.

```
    <dependencies>    </dependencies>
    <build>
        <plugins>
            <!-- Package as an executable jar/war -->
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

Spring Application

The Spring Application is a class that provides a convenient way to bootstrap a Spring application. It can be started from the main method. We can call the application by calling a static run() method.

```
public static void main(String[] args)
{
SpringApplication.run(ClassName.class, args);
}
```

Spring Boot Basic Annotations

- @Bean - indicates that a method produces a bean to be managed by Spring.
- @Service - indicates that an annotated class is a service class.
- @Repository - indicates that an annotated class is a repository, which is an abstraction of data access and storage.
- @Configuration - indicates that a class is a configuration class that may contain bean definitions.
- @Controller - marks the class as web controller, capable of handling the requests.

Spring Basic Annotation

@Component is a generic stereotype for a Spring managed component. It turns the class into a Spring bean at the auto-scan time.

@Service: Indicates that an annotated class is a “Service”.

@Repository: Indicates that an annotated class is a “Repository”. This annotation serves as a specialization of @Component and advisable to use with DAO classes.

@Autowired: Autowired annotation is used for automatic injection of beans.

@Qualifier annotation is used in conjunction with Autowired to avoid confusion when we have two of more bean configured for same type.

Spring MVC Annotation

@Controller

This annotation is used to make a class as a web controller, which can handle client requests and send a response back to the client.

@RequestMapping

The value attribute of @RequestMapping annotation is used to specify the URL pattern

@RequestParam

This is another useful Spring MVC annotation that is used to bind HTTP parameters into method arguments of handler methods.

@PathVariable

It enables the controller to handle a request for parameterized URLs.

Spring MVC Annotation

@RequestBody annotations tell the Spring to find a suitable message converter to convert a resource representation coming from a client into an object.

@RestController

This is a convenience annotation for developing a RESTful web service with the Spring MVC framework.

@SprinbBootApplication is a single annotation combines three annotations like @Configuration, @EnableAutoConfiguration, and @ComponentScan it run your application without deploying it into a web server, as it comes with an embedded Tomcat server.

@EnableAutoConfiguration is a Spring boot annotation which enables the auto-configuration feature, which makes Spring guess the configuration based on the JAR presents in the classpath.

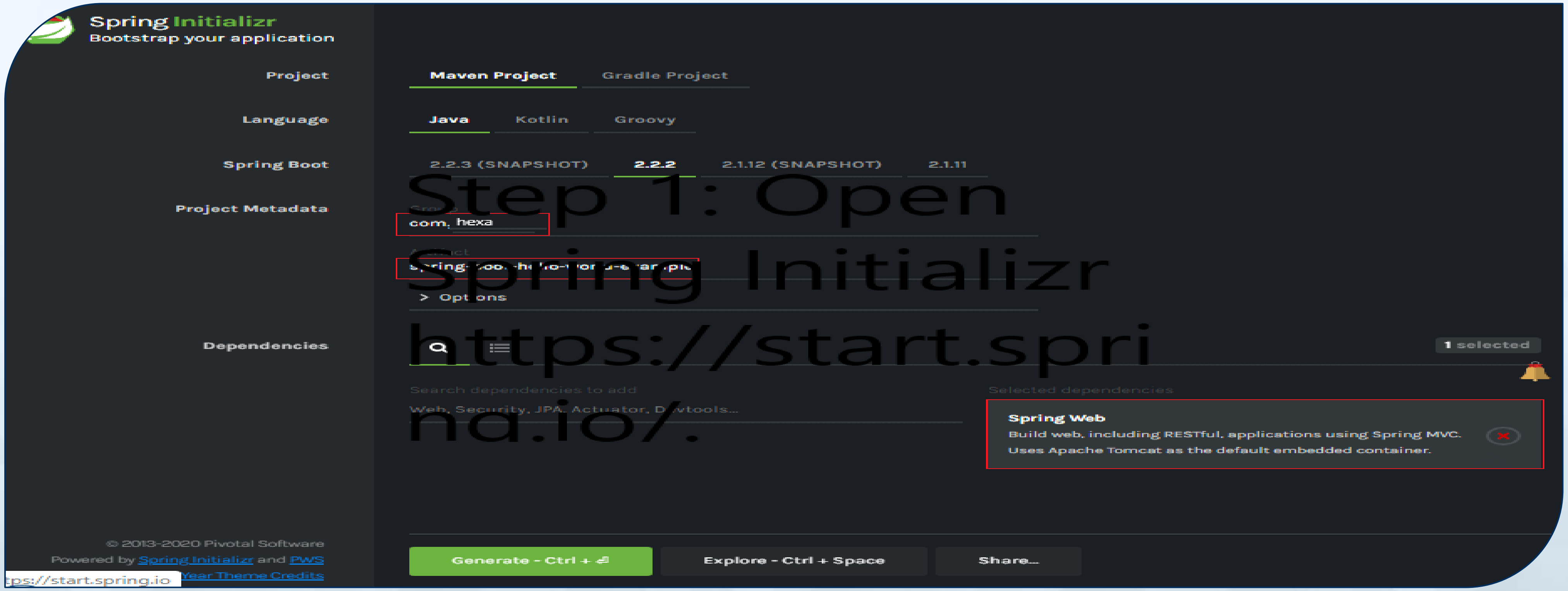
Spring MVC Annotation

@ResponseStatus annotation is used to override the HTTP response code for a response. You can use this annotation for error handling while developing a web application or RESTful web service using Spring.

@ResponseStatus annotation is used to override the HTTP response code for a response. This annotation is for error handling while developing a web application or RESTful web service using Spring.

Spring Boot – Hello world Demo

Step 1: Open Spring Initializer : https://start.spring.io/



Spring Initializr
Bootstrap your application

Project: [✓ Maven Project] Gradle Project

Language: [✓ Java] Kotlin Groovy

Spring Boot: 2.2.3 (SNAPSHOT) [✓ 2.2.2] 2.1.12 (SNAPSHOT) 2.1.11

Project Metadata
Group: com.hexa
Artifact: spring-boot-hello-world-example
> Options

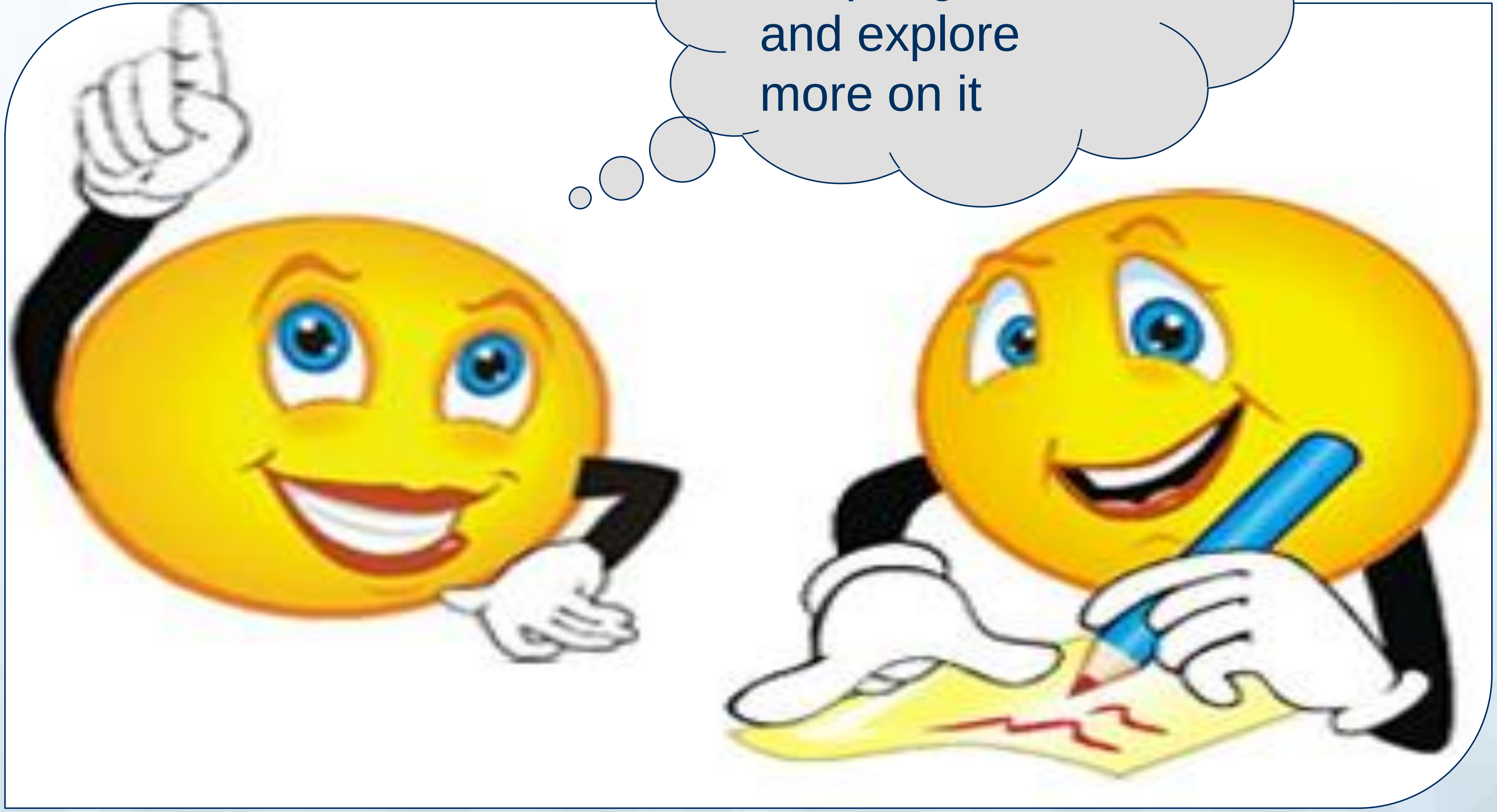
Dependencies — 1 selected
Search dependencies to add: Web, Security, JPA, Actuator, Devtools...
Selected dependencies:
Spring Web
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Generate - Ctrl + ⏎ | Explore - Ctrl + Space | Share...

© 2013-2020 Pivotal Software
Powered by Spring Initializr and PWS
Year Theme Credits

ps://start.spring.io

Step 1: Open Spring Initializr https://start.spring.io/.



Create a project in Spring Boot and explore more on it

HEXAWARE

Innovative Services

Passionate Employees

Delighted Customers

Thank you

www.hexaware.com