

Building Web Applications Using the Spring Framework

- *An application in which we implement all the responsibilities in the same executable artifact.*
- *You can see this as one application that fulfills all the use cases. The responsibilities can sometimes be implemented within different modules to help the application be more comfortable to maintain.*
- *The logic of one can't be separated from the logic of others at run time.*
- *Monolithically architectures offer less flexibility for scaling.*

- *A microservice system has the responsibilities implemented within different executable artifacts.*
- *You can see the system as being formed of multiple applications that execute at the same time and communicate between them when needed via the network.*
- *While this offers more flexibility for scaling, it introduces other difficulties.*
- *We can enumerate here latencies, security concerns, network reliability, distributed persistence, and deployment management.*



- ◆ *Aims to overcome the application problems by enabling the use of simple JavaBeans (POJOs) to implement the business logic.*
- ◆ *Enables the developers to create and test applications easily.*
- ◆ *Aims to minimize the dependency of application code on its framework.*
- ◆ *Enables the use of simplicity and ease of testability in standalone applications.*

Features of Spring





EJB

- ☐ Is a popular non-standard open-source framework developed by Interface21 Inc.
- ☐ Enables you to implement only those features that you require for an application.

- ☐ Is a specification defined by the Java Community Process (JCP) and supported by all the major J2EE vendors.
- ☐ Provides a set of prepackaged features, most of which you may not need.

Benefits of Spring MVC over the Struts framework



- ◆ *Provides a clear separation between the controllers, models, and views.*
- ◆ *Is highly flexible, unlike Struts that does not implement interfaces and forces your Action and Form objects into concrete inheritance.*
- ◆ *Provides controllers for handling user requests.*
- ◆ *supports several view technologies, such as JSP, Velocity, Tiles, and JSF, Thymeleaf, Free Marker.*
- ◆ *Provides controllers that can be easily configured through DI, just like any other application object.*

Benefits of Spring MVC over the Struts framework



- ◆ *Allows the development of application that are easier to test than the Struts MVC Web applications.*
- ◆ *Allows the implementation of validators as application objects, which are not dependent on the Spring API.*
- ◆ *Does not impose dependencies on controllers.*
- ◆ *Provides convenience controller implementations that you can extend, if required, by the application.*
- ◆ *Allows you to integrate easily with the existing technologies.*

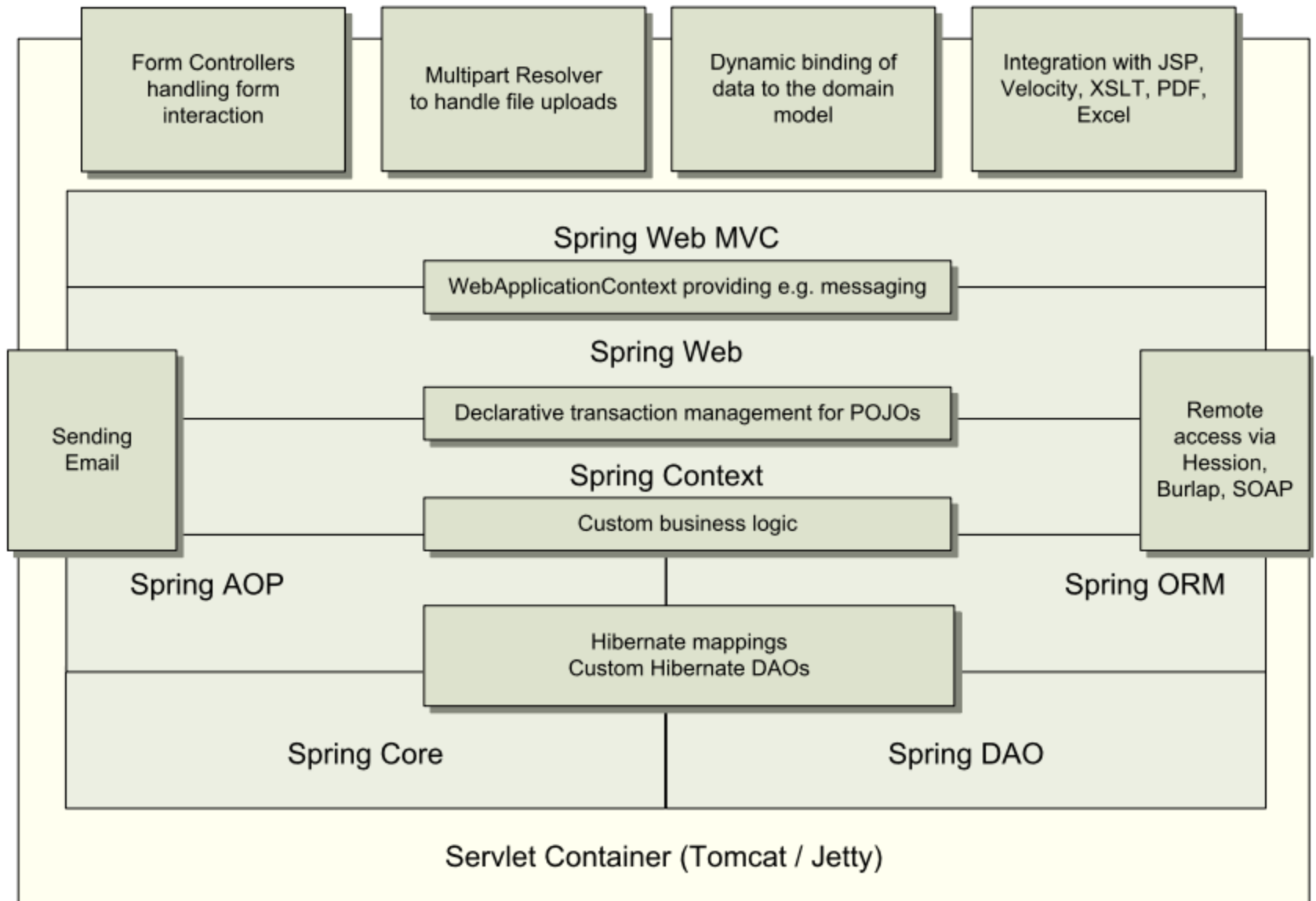
Which one of the following features of Spring eliminates the need for code lookup, allows pluggability and reuse of existing code, and makes application maintenance and testing easy?

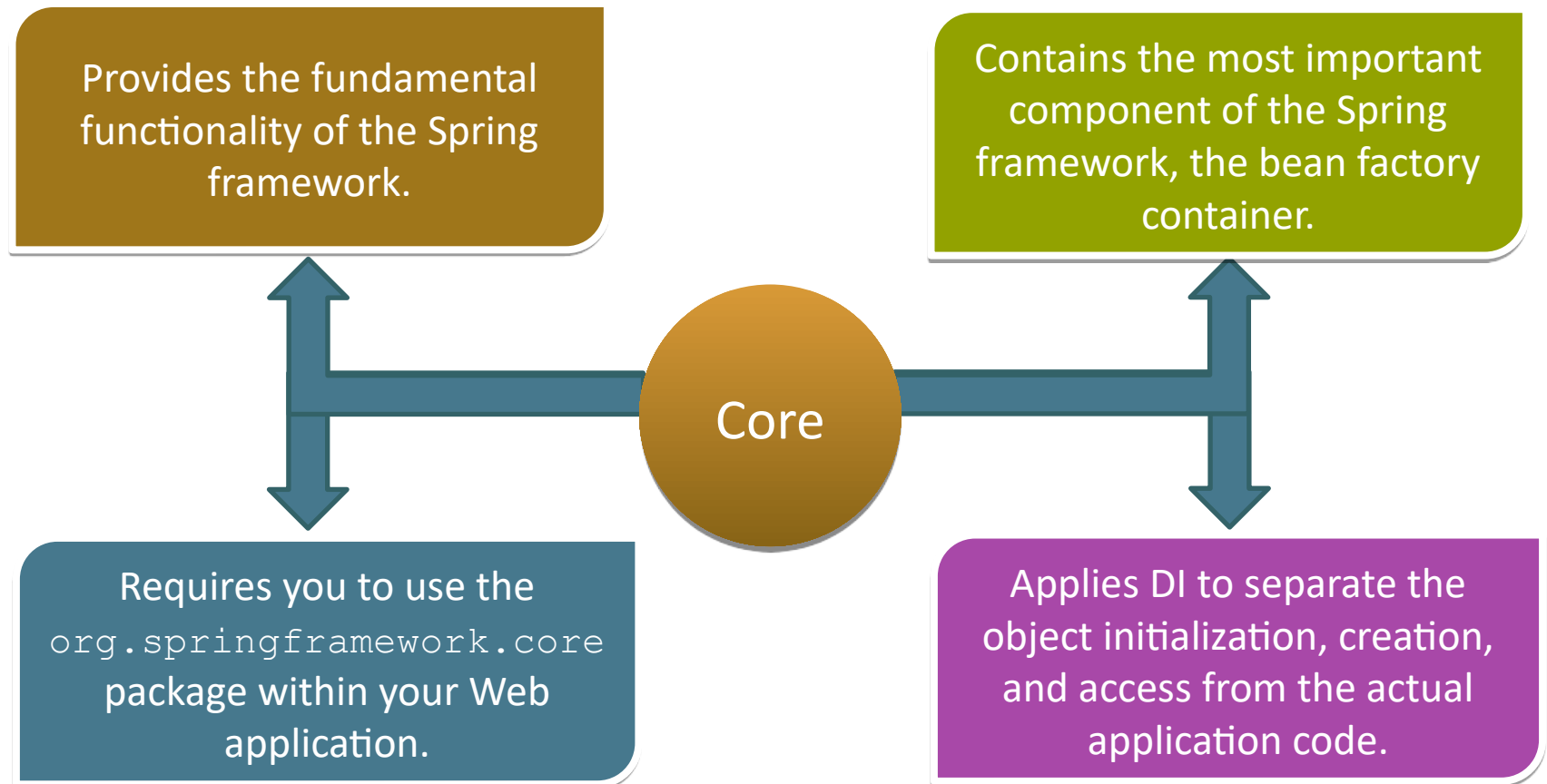
1. DI
2. AOP
3. Container
4. Lightweight

Answer:

1. DI

Introducing the Spring Architecture



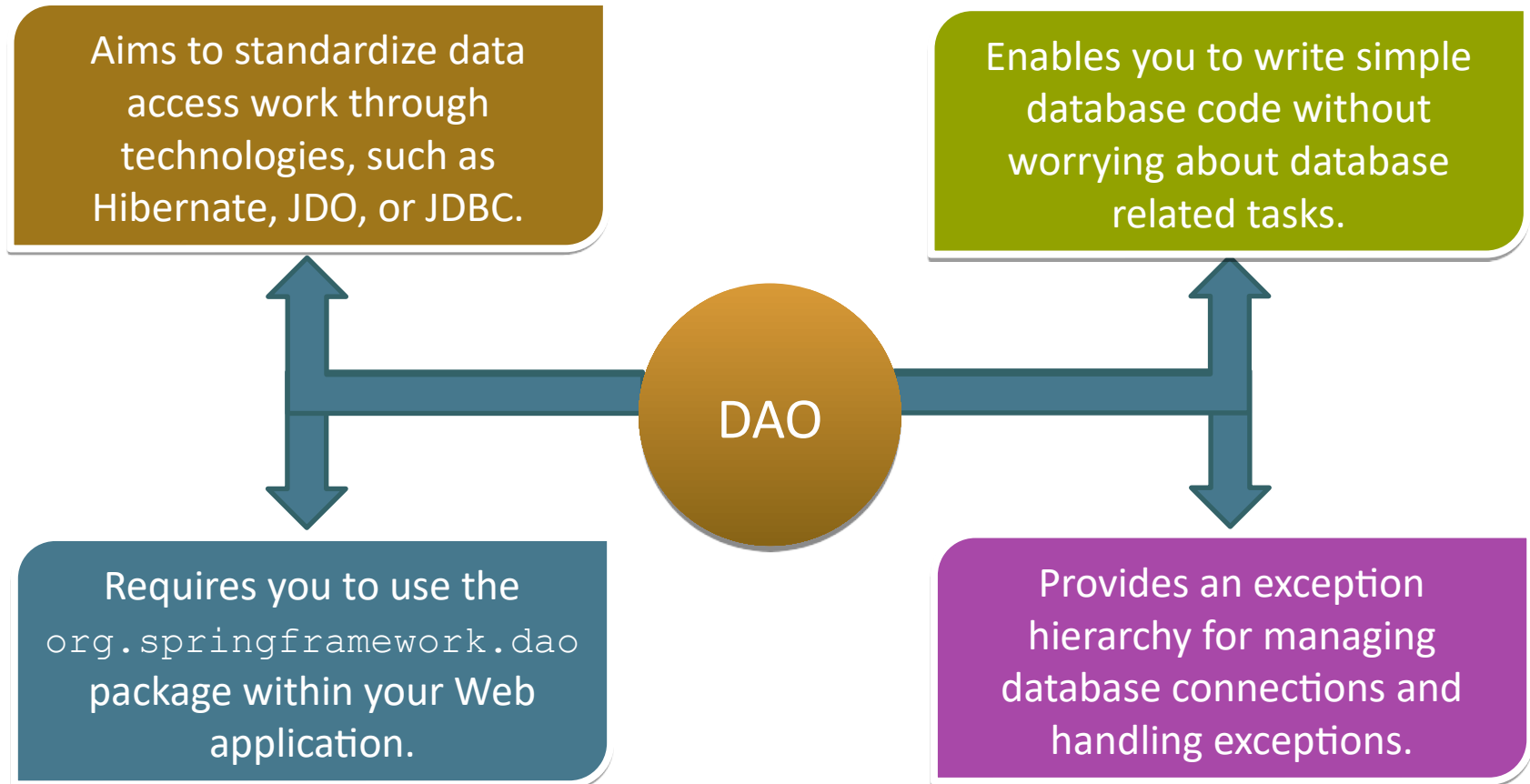


- *Spring Framework does not force you to use everything within it; it is not an all-or-nothing solution.*
- *Existing front-ends built using Thymeleaf, React, Angular, WebWork, Struts, Tapestry, JSF or other UI frameworks can be integrated perfectly well with a Spring-based middle-tier, allowing you to use the transaction features that Spring offers.*
- *The only thing you need to do is wire up your business logic using an **ApplicationContext** and integrate your web layer using a **WebApplicationContext***

Spring Core: *The Core package is the most fundamental part of the framework and provides the IoC and Dependency Injection features. The basic concept here is the BeanFactory, which provides a sophisticated implementation of the factory pattern which removes the need for programmatic singletons and allows you to decouple the configuration and specification of dependencies from your actual program logic.*

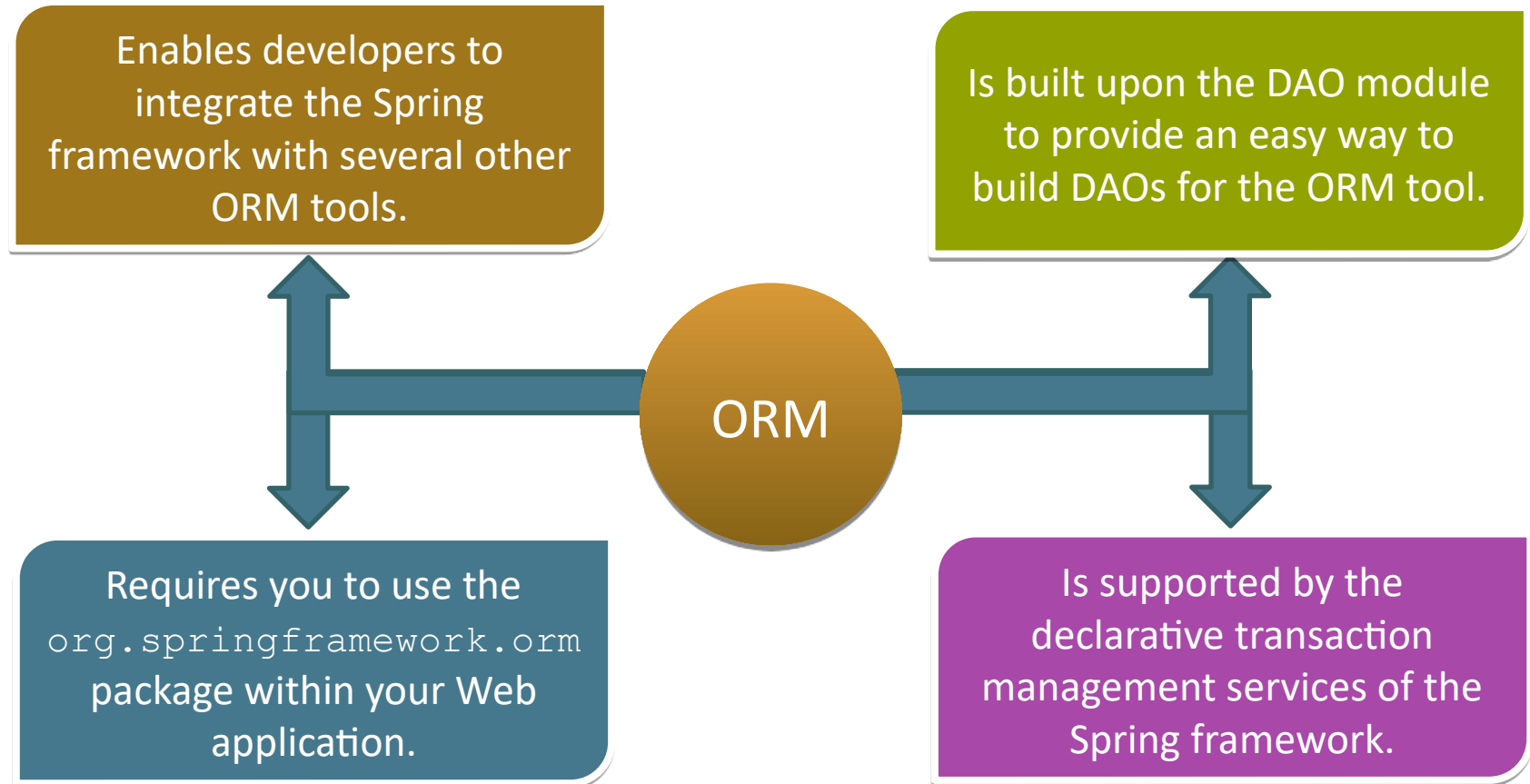
Spring context: *The Context package build on the solid base provided by the Core package: it provides a way to access objects in a framework-style manner in a fashion somewhat reminiscent of a JNDI-registry.*

The context package inherits its features from the beans package and adds support for internationalization (I18N) (using for example resource bundles), event-propagation, resource-loading, and the transparent creation of contexts by, for example, a servlet container.



DAO: *The DAO package provides a JDBC-abstraction layer that removes the need to do tedious JDBC coding and parsing of database-vendor specific error codes. The JDBC package provides a way to do programmatic as well as declarative transaction management, not only for classes implementing special interfaces, but for all your POJOs (plain old Java objects).*

ORM: *The ORM package provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis. Using the ORM package you can use all those O/R-mappers in combination with all the other features Spring offers, such as the simple declarative transaction management feature mentioned previously.*

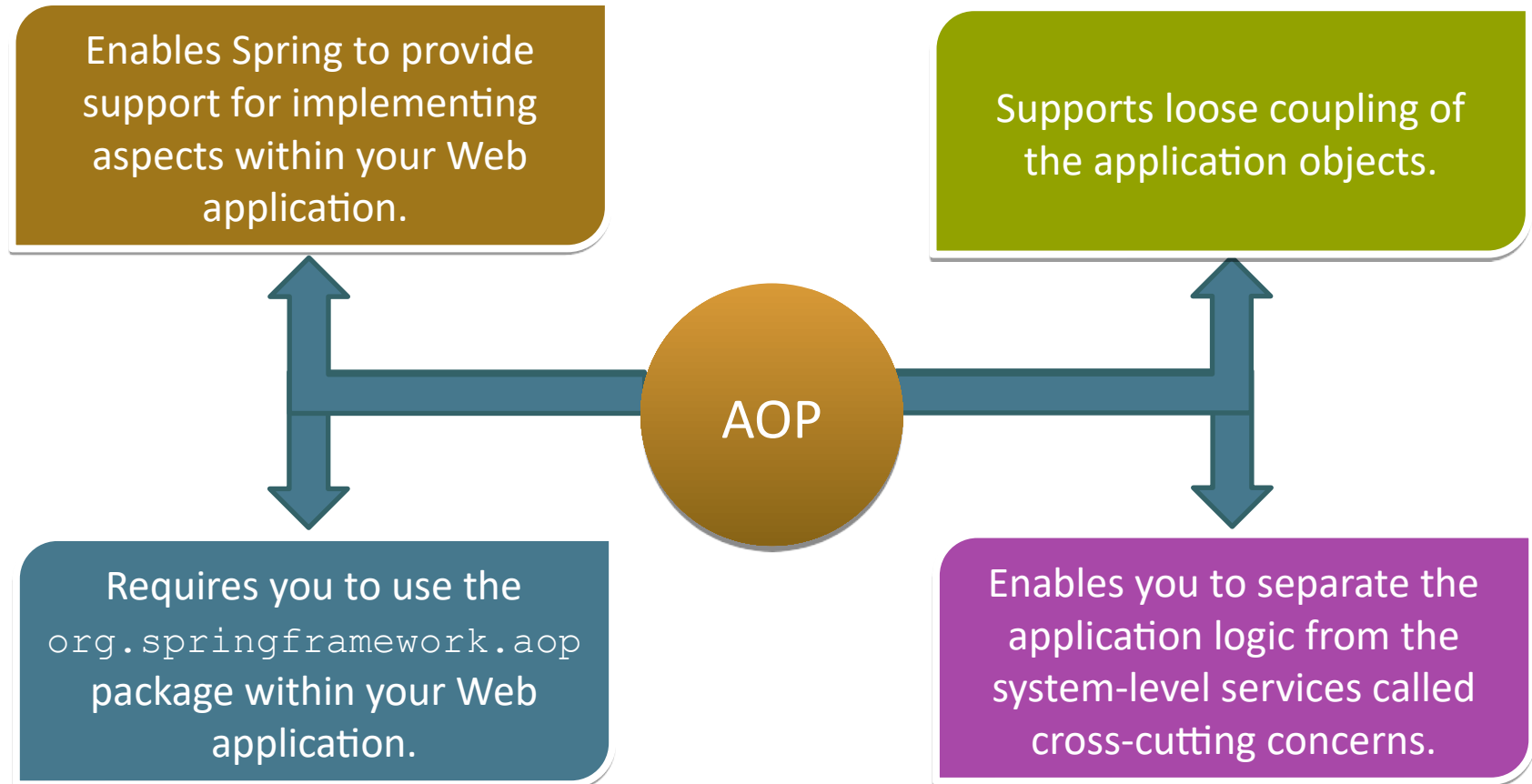


AOP: *Spring's AOP package provides an AOP Alliance-compliant aspect-oriented programming implementation allowing you to define, for example, method-interceptors and pointcuts to cleanly decouple code implementing functionality that should logically speaking be separated.*

Using source-level metadata functionality you can also incorporate all kinds of behavioral information into your code, in a manner similar to that of .NET Attributes.

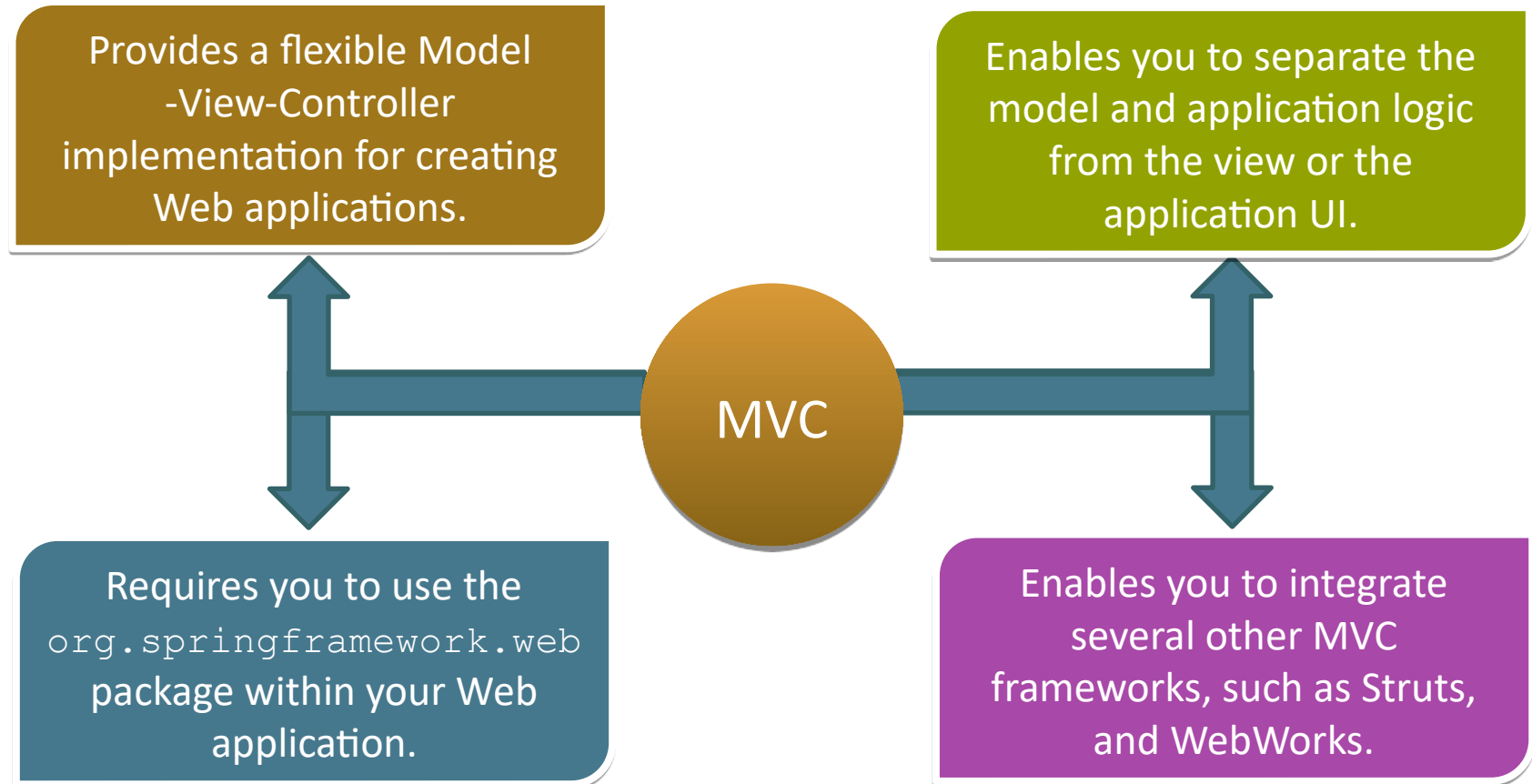
Spring Web: *Spring's Web package provides basic web-oriented integration features, such as multipart file-upload functionality, the initialization of the IoC container using servlet listeners and a web-oriented application context.*

When using Spring together with WebWork or Struts, this is the package to integrate with.

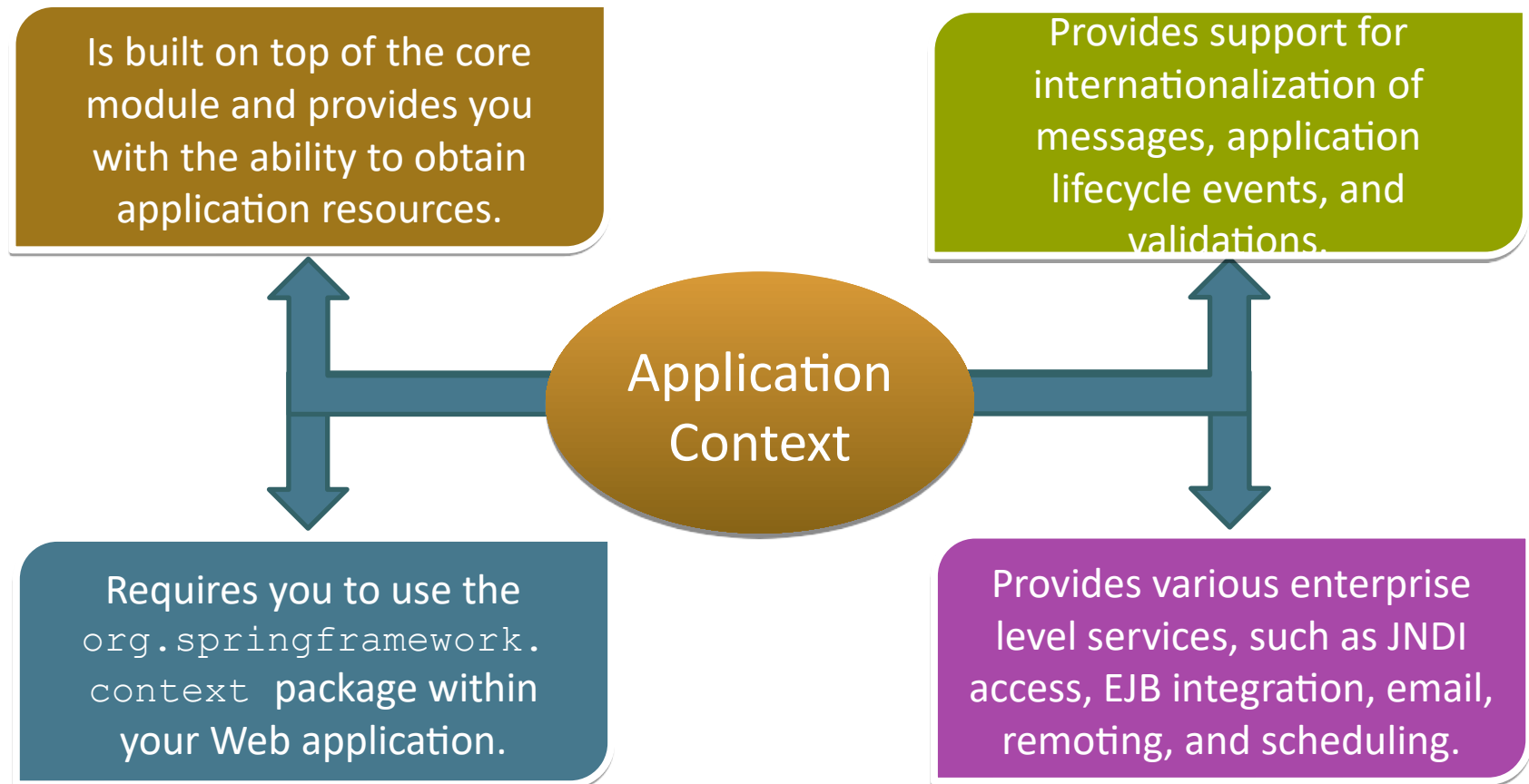


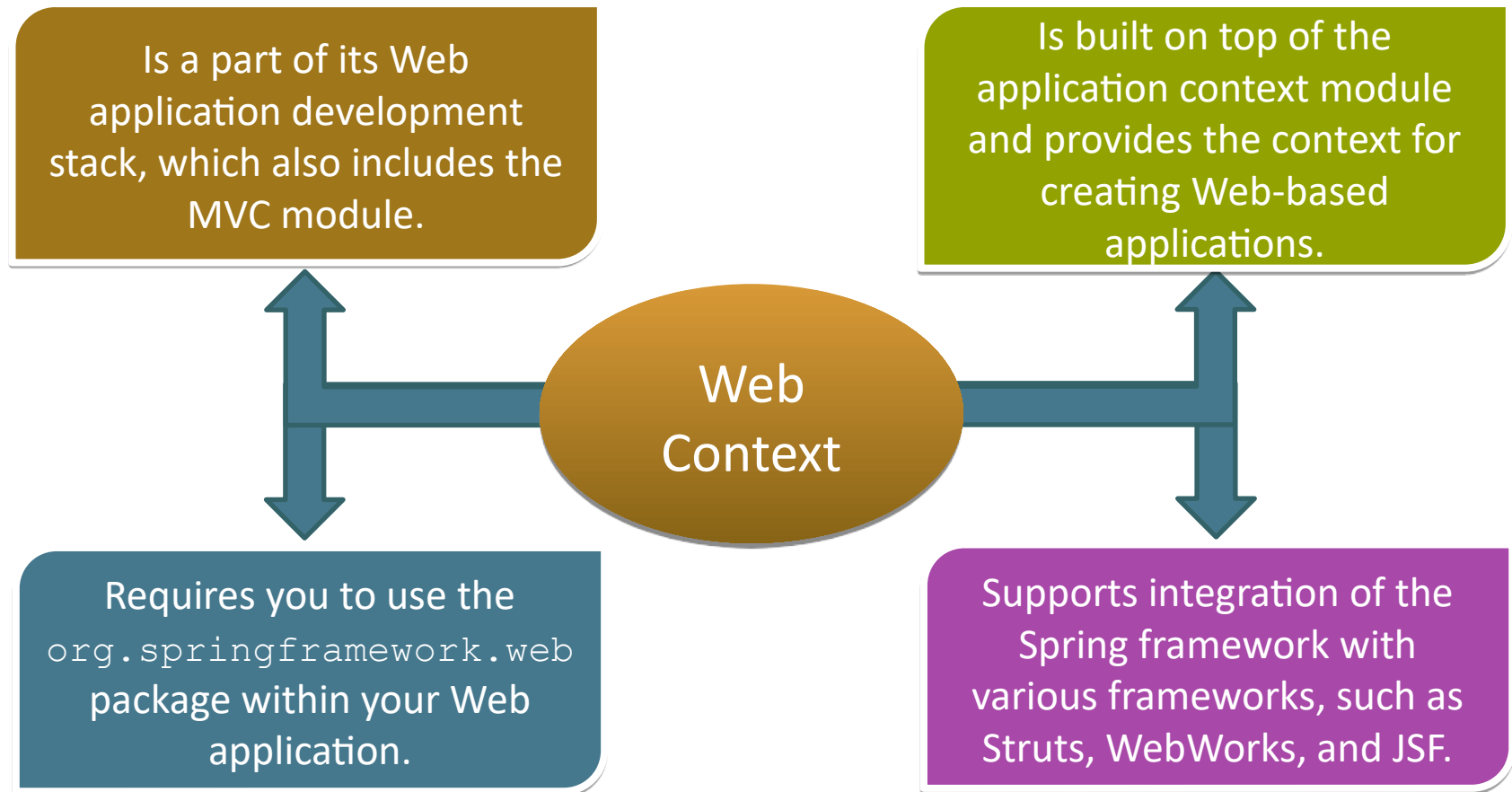
Spring MVC: Spring's MVC package provides a Model-View-Controller (MVC) implementation for web-applications.

Spring's MVC framework is not just any old implementation; it provides a clean separation between domain model code and web forms, and allows you to use all the other features of the Spring Framework.



Application Context





Which one of the following packages do you need to implement the MVC module in your Web application?

1. `org.springframework.web`
2. `org.springframework.context`
3. `org.springframework.core`
4. `org.springframework.dao`

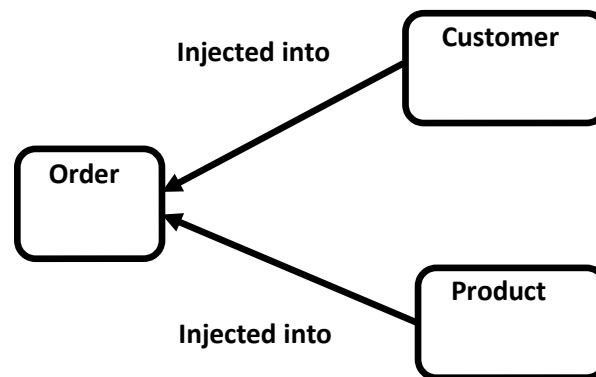
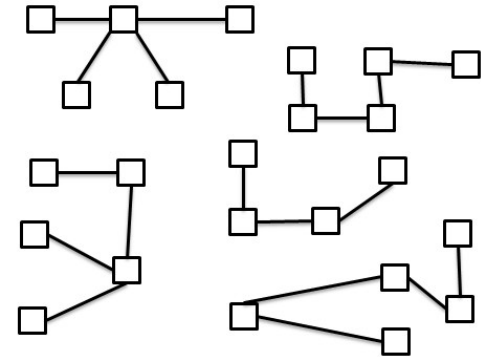
Answer:

1. `org.springframework.web`

Managing Application Objects



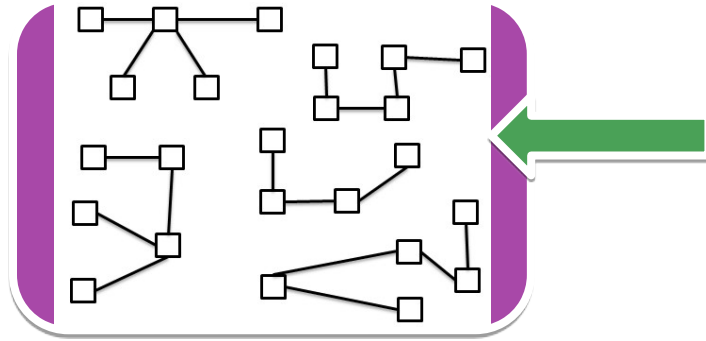
Dependency injection



Managing Application Objects (Contd.)

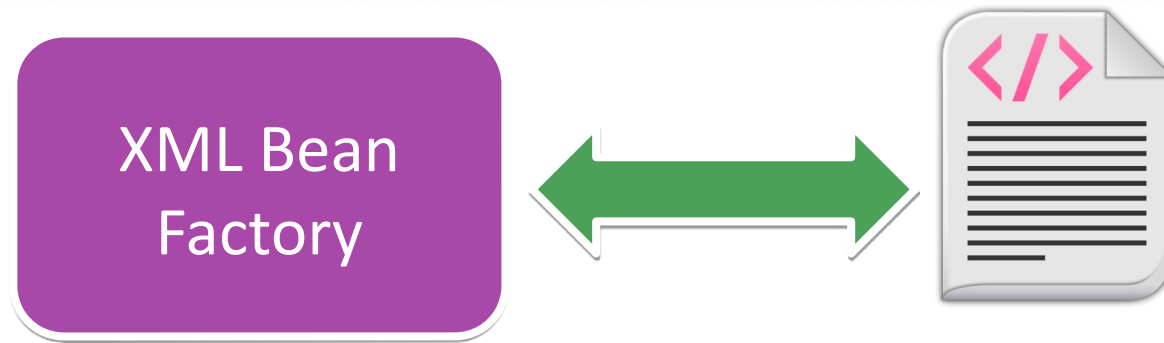


Introducing Bean Factory



- ◆ *The objects created by the bean factory are fully configured, ready to use, and aware of their relationships with the other application objects.*
- ◆ *The bean factory is involved in managing the life cycle of objects that have been created.*
- ◆ *A bean factory is represented by the `org.springframework.beans.factory.BeanFactory` interface.*
- ◆ *The most commonly used bean factory is the XML bean factory represented by the `org.springframework.beans.factory.xml.XmlBeanFactory` interface.*

Introducing Bean Factory (Contd.)



Syntax of a bean configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd
">
    ...
</beans>
```

Introducing Bean Factory (Contd.)

- ◆ *In a bean configuration file, each bean is defined by using the `<bean>` tag within the `<beans>` tag.*
- ◆ *A bean definition contains the basic information about a bean that a container must know, such as the process for creating a bean, details about the bean life cycle, and the various dependencies for that bean.*

Attributes of the `<beans>` tag

id

Used to uniquely identify a bean in the Spring container.

name

Used to specify an alias name for the bean.

class

Used to specify the fully-qualified (package name + class name) name of the bean class.

scope

Used to define the scope of the bean being defined.

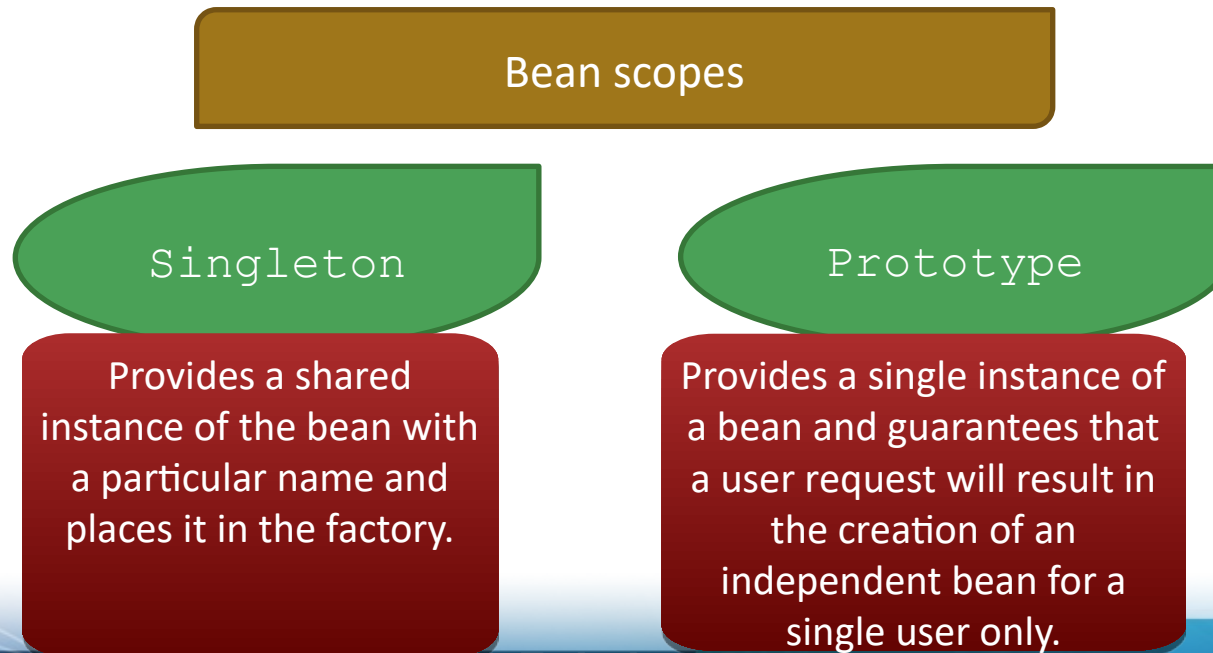
Introducing Bean Factory (Contd.)

Example: Bean declaration

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"
">
<bean id="Beckham" class="Sport.LAGalaxy"/>
</beans>
```

Introducing Bean Factory (Contd.)

- ◆ *To instantiate a bean factory in your Web application, you first need to load the bean configuration file.*
- ◆ *This can be done by using the Resource object, defined by the `org.springframework.core.io.Resource` interface.*
- ◆ *Depending upon the place from where the bean configuration file is to be loaded, the Spring framework provides several implementations of the Resource interface.*



Introducing Bean Factory (Contd.)

Bean
instantiation

- ◆ *You can retrieve an instance of a bean registered under a given name from the bean factory by using the `getBean()` method.*
- ◆ *This method takes the ID of the bean as a string value.*

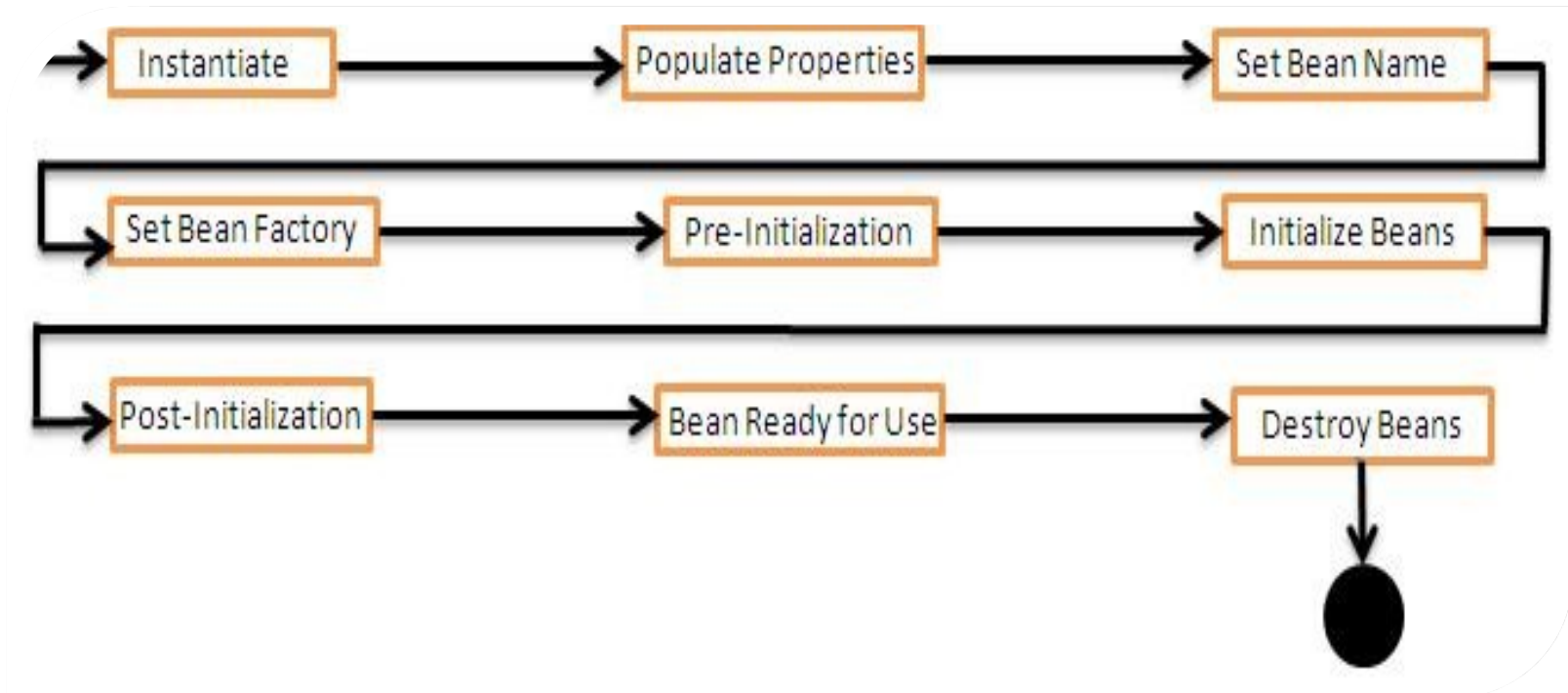
Example:

```
LAGalaxy team = (LAGalaxy)  
factory.getBean("Beckham");  
team.squadSize();
```

- ◆ *When the `getBean()` method is called, the bean factory container instantiates the bean and set its properties through DI.*
- ◆ *After this, the life of the bean begins inside the Spring container.*

Introducing Bean Factory (Contd.)

Life cycle of a bean inside the bean factory



Which one of following attributes is used to uniquely identify a bean in the Spring container?

1. name
2. id
3. scope
4. class

Answer:

2. id

Introducing Application Context



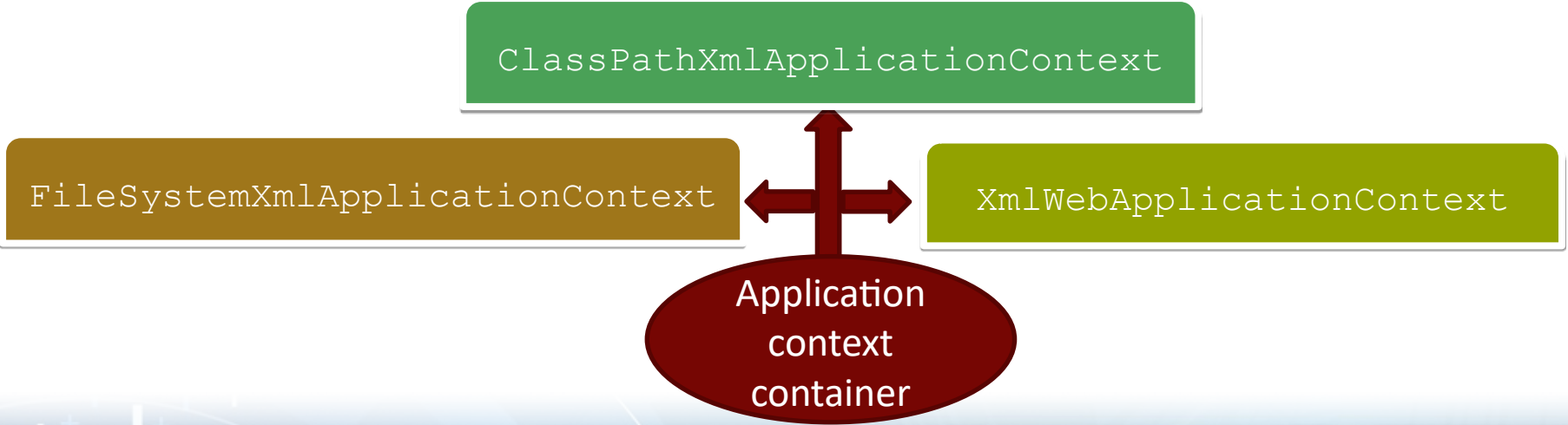
Application
Context

- ◆ *Provides a means for resolving text messages, including support for internationalization of those messages.*
- ◆ *Provides a generic way to load file resources, such as images.*
- ◆ *Publishes events to beans, which are registered as listeners.*

ClassPathXmlApplicationContext

FileSystemXmlApplicationContext

XmlWebApplicationContext



Application
context
container

Introducing Application Context (Contd.)



Loading the
application
context

- ◆ *If you want to use the application context container for managing your application objects, then you have to load a particular `ApplicationContext` implementation.*

Example: `ClassPathXmlApplicationContext`

```
ApplicationContext appContext = new  
ClassPathXmlApplicationContext("Sport/Spring-  
Config.xml");
```

Example: `FileSystemXmlApplicationContext`

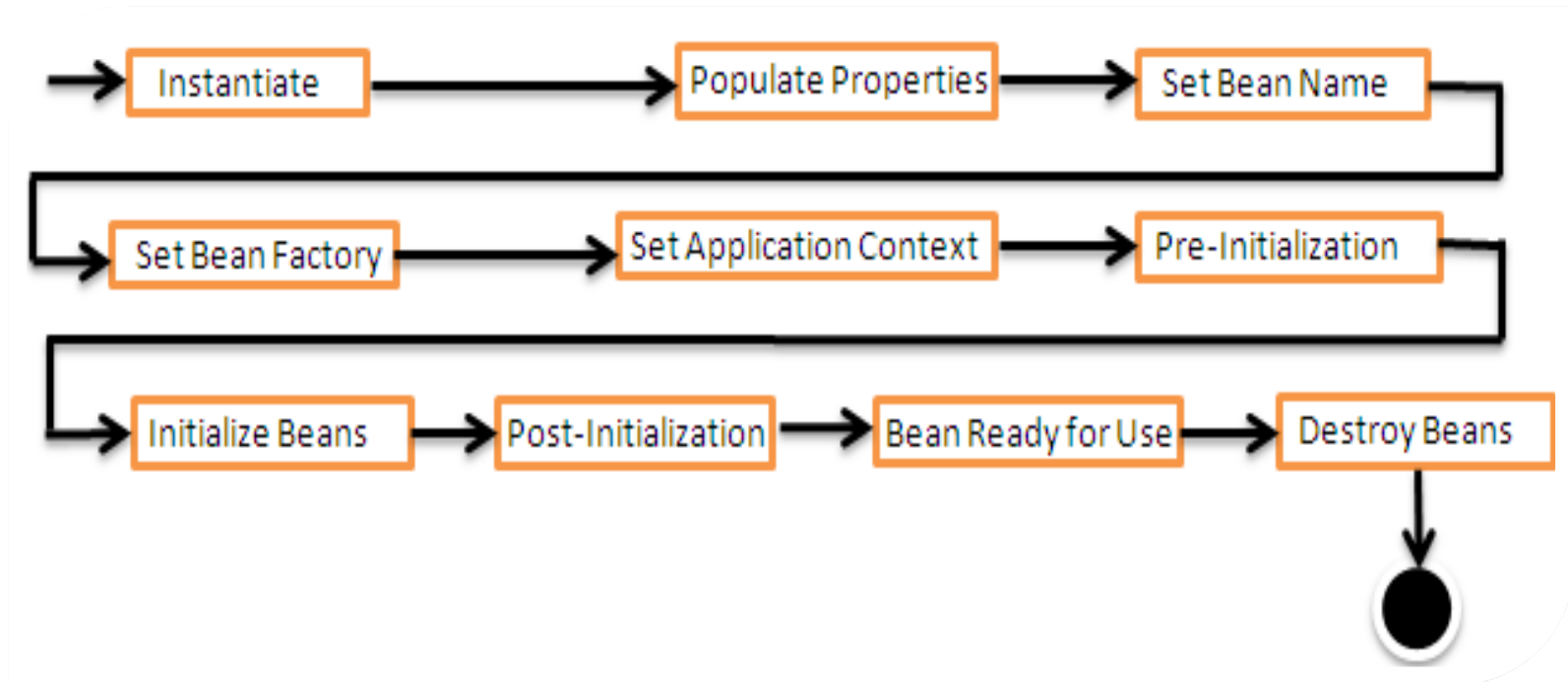
```
ApplicationContext appContext = new  
FileSystemXmlApplicationContext("c:/Sport/Spr  
ing-Config.xml");
```

Example: `XmlWebApplicationContext`

```
XmlWebApplicationContext appContext = new  
XmlWebApplicationContext();
```

Introducing Application Context (Contd.)

Life cycle of a bean inside application context



Which implementation of the application context container is used to load the bean configuration file located at the specified class path?

1. `XmlApplicationContext`
2. `XmlWebApplicationContext`
3. `ClassPathXmlApplicationContext`
4. `FileSystemXmlApplicationContext`

Answer:

3. `ClassPathXmlApplicationContext`