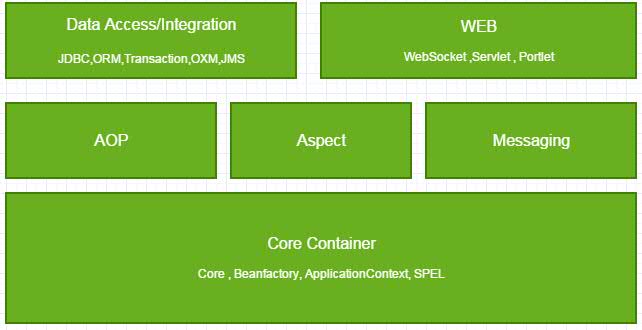
**Spring Architecture**

Spring maintains a Modular architecture. The benefits of Modular architecture are you can take those modules which are applicable for your project. To be precise Spring comes with jars and jars packaged with module basis like (JMS module, ORM module, WEB module, AOP module etc.). so If your project only needs JMS ,JDBC and Spring core you can only add those jars in your classpath . Later if you want to use ORM module just add those ORM specific jars and you are ready to use ORM framework. Very easy isn’t it?

**Spring architecture diagram**

****

**Core Container: Core container consists of following**

**Core Module**: Spring core module provides the basic feature of Spring that is IOC and Dependency injection.

**BeanFactory** : Bean factory is the most basic container. In this container, Spring beans are maintaining their lifecycle. Wiring between them, communicate with each other. In BeanFactory simple POJO transform to Spring Beans. In one word POJO got Spring-ness. BeanFactory maintains factory pattern.

**ApplicationContext** : ApplicationContext builds on top of BeanFactory it is a bit advanced container than BeanFactory. It supports some extra features like event handling, locale handling etc.

**SPel**: Spring expression language used for work on Spring beans.

**AOP**: AOP module supports Aspect Oriented Programing by which developer can segregate cross-cutting concerns like (Logging, transaction) from business logic

**Aspect**: Aspect module provides supports to use AspectJ which is a very powerful Aspect orient programming language.

Messaging: This module uses STOMP protocol. STOMP is the Simple (or Streaming) Text Orientated Messaging Protocol. STOMP provides an interoperable wire format so that STOMP clients can communicate with any STOMP message broker to provide easy and widespread messaging interoperability among many languages, platforms, and brokers.

**Data Access/Integration**: Data Access/Integration consists of following.

**JDBC**: Spring provides a JDBC template which hides boilerplate code. So using this module working on JDBC is very easy.

**ORM**: Spring provides templates for popular ORM framework like Hibernate,ibatis etc.

**OXM**: Spring provides template for popular Object/XML mapping framework like JAXB,XML Beans,Castor etc.

**Transaction**: Spring provides support for Programmatic and declarative transaction. By Spring AOP it provides declarative transaction.

**WEB**: Web layer produce support for the portlet,servlet, and WebSocket programming.

Using Spring MVC we can develop an MVC project.