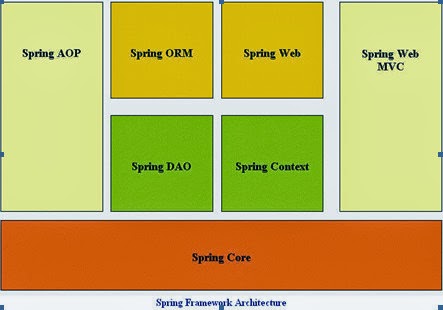
**Spring:**

Spring have 7 main modules:



**Spring core**: the core container provides the fundamental functionality of the Spring framework. In this module primary component is theBeanFactory, an implementation of the Factory pattern. The BeanFactory applies the Inversion of Control (IOC) pattern to separate an application's configuration and dependency specification from the actual application code.

**Spring AOP**: he Spring AOP module allows a software component to be decorated with additional behavior, through its configuration management feature. As a result you can easily AOP-enable any object managed by the Spring framework. The Spring AOP module provides transaction management services for objects in any Spring-based application. With Spring AOP you can incorporate declarative transaction management into your applications without relying on EJB components.

**Spring ORM**: the Spring Framework supports integration with Hibernate, Java Persistence API (JPA) and Java Data Objects (JDO) for resource management, data access object (DAO) implementations, and transaction strategies

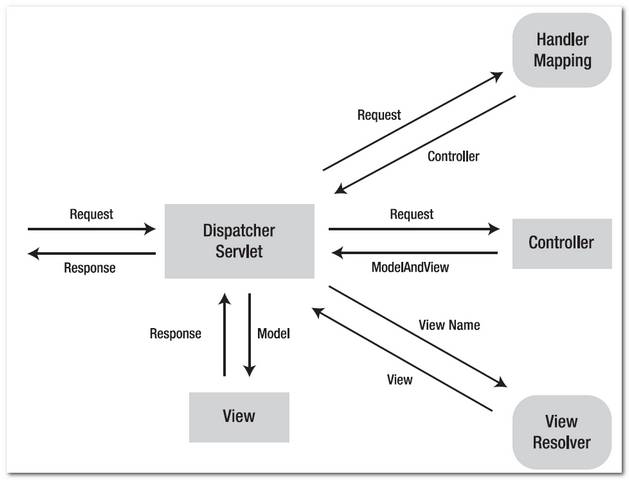
**Spring DAO**: Spring provides integration with OR mapping tools like Hibernate, JDO and iBATIS. Spring transaction management supports each of these ORM frameworks as well as JDBC.

**Spring Web**: The Web context module provides basic web-oriented integration features builds on top of the application context module, providing contexts for Web-based applications. As a result, the Spring framework supports integration with Jakarta Struts. The Web module also eases the tasks of handling multi-part requests and binding request parameters to domain objects.

**Spring Context**: TThe Spring context is a configuration file that provides context information to the Spring framework. The Spring context includes enterprise services such as e-mail, JNDI, EJB, internalization, validation, scheduling and applications lifecycle events. Also included is support for the integration with templating frameworks such as velocity.

**Spring web mvc**: module provide MVC to web application.

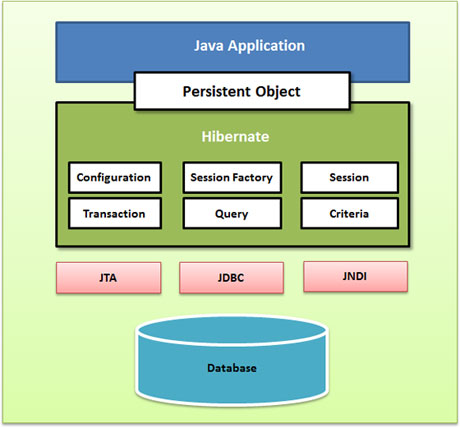
Spring mvc:



1. When have new request from user, Dispatcher servlet will get it.
2. Dispatcher Servlet will send request to Handler Mapping for checking the corresponding Controller.
3. After find the corresponding Controller in Handler Mapping, Dispatcher Servlet will send the request to corresponding Controller.
4. Controller will get request and call appropriate service methods base on @RequestMapping, method: GET, POST,… The method will set the value to the Model based on the definition of business logic. After finish the request, Controller will return the view name to Dispatcher Servlet.
5. Dispatcher Servlet get the view name and based on View Resolver, it will get appropriate View.
6. After get appropriate view, Dispatcher Servlet will transfer data from Model to View and display on user interface.

**Hibernate:**

Hibernate ORM: Hibernate ORM enables developers to more easily write applications whose data outlives the application process. As an Object/Relational Mapping (ORM) framework, Hibernate is concerned with data persistence as it applies to relational databases (via JDBC).



**Configuration:**  use for config hibernate : datasource (connection), class mapping

**Transaction**: use for check the process update, delete, insert, read for go to finish or not, if the process is not finish, it will roll back

**Session**: use to insert, delete, read, update, using associate with criteria and query to retrieve database.

**Session Factory**: is a factory of session and client of ConnectionProvider. It holds second level cache (optional) of data, provides factory method to get the object of Session.

Hibernate search: Hibernate Search transparently indexes your objects and offers fast regular, full-text and geolocation search. Ease of use and easy clustering are core.

Hibernate validator: Hibernate Validator allows to express and validate application constraints. The default metadata source are annotations, with the ability to override and extend through the use of XML. It is not tied to a specific application tier or programming model and is available for both server and client application programming

Hibernate OGM: Hibernate Object/Grid Mapper (OGM) provides Java Persistence (JPA) support for NoSQL solutions. It reuses Hibernate ORM’s engine but persists entities into a NoSQL datastore instead of a relational database.

Note: on my demo: It use native hibernate APIS (hibernate session not entity manager), spring 4