Introduction to the Spring Framework

References

- Spring documentation - http://www.springsource.org/documentation
- Spring API - http://static.springframework.org/spring/docs/2.5.x/api/index.html
- Introduction to the Spring Framework 2.5, by Rod Johnson (Originator of Spring) - http://www.theserverside.com/tt/articles/article.tss?l=IntrotoSpring25
- Spring in Action, 2nd Edition, Manning Publishing, August 2007
- Pro Spring 2.5, Apress Publishing, August 2008
- Maven - http://maven.apache.org/
- Maven - http://books.sonatype.com/maven-book/index.html

What is Spring?

- **A Container**
  - ***Creates objects and makes them available to your application***
- **A Framework**
  - ***Provides an infrastructure of classes that make it easier to accomplish tasks***

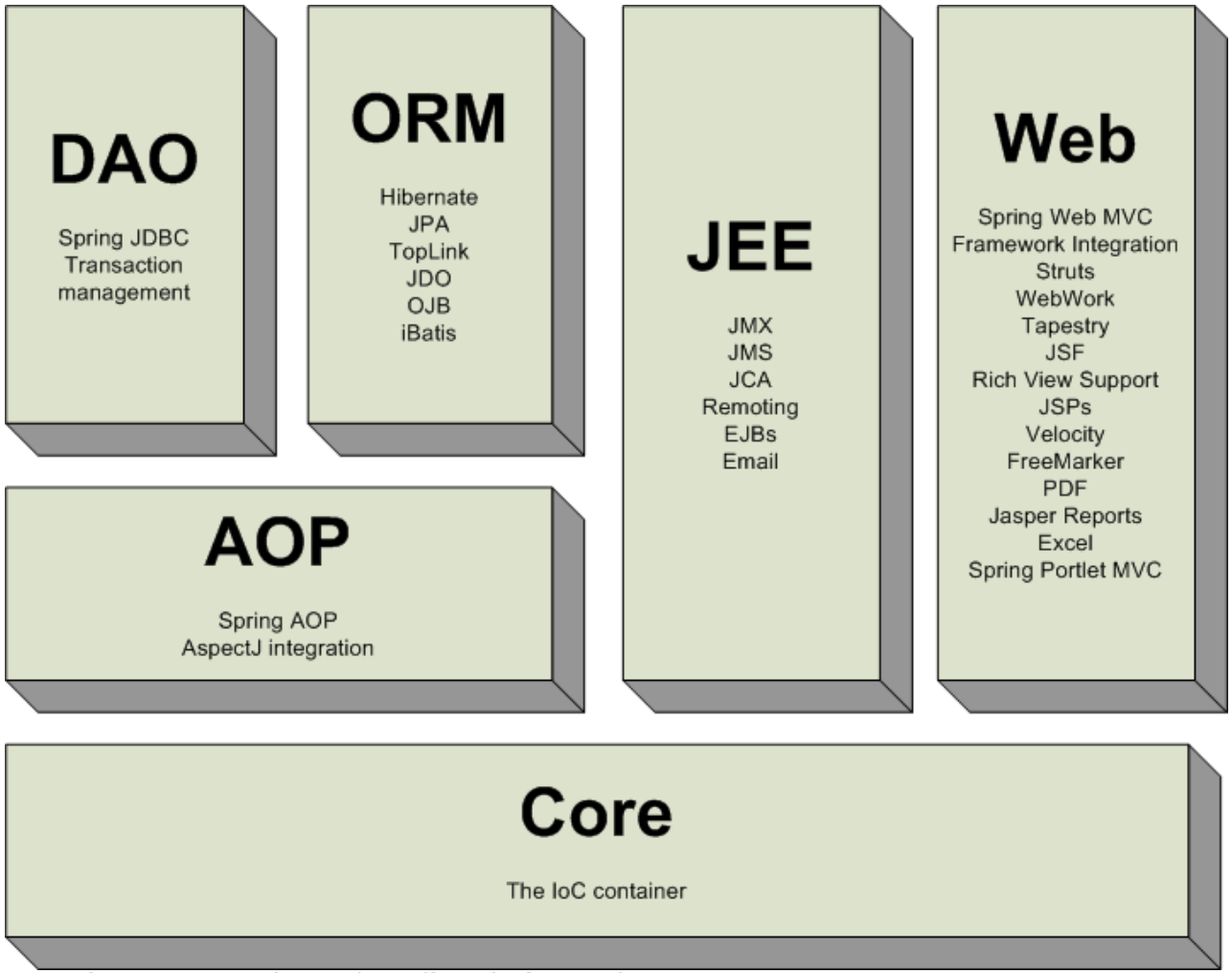
The Spring Framework

- Official 1.0 release in 2004
- Current release (January 2009) is 2.5.6
  - See: http://www.springsource.org/download
- Works with Java 1.4 or 1.5 or 1.6 and J2EE 1.3 and Java EE 5
  - See: http://static.springframework.org/spring/docs/2.5.x/reference/new-in-2.html

What does Spring provide?

- **Lightweight container and framework**
  - *Most of your code will be unaware of the Spring framework*
  - *Use only the parts you of Spring you want*
- **Manages dependencies between your objects**
  - *Encourages use of interfaces*
  - *Lessens "coupling" between objects*
- **Cleaner separation of responsibilities**
  - *Put logic that applies to many objects in one single place*
  - *Separate the class's core responsibility from other duties*
- **Simplifies database integration**
  - *Spring JDBC*
  - *Hibernate*
  - *iBATIS*
  - *Java Persistence*

Spring Modules – Use What You Want



See: http://static.springframework.org/spring/docs/2.5.x/reference/introduction.html#introduction-overview

Example Application

- Contacts – store and retrieve contacts
  - Created using MyEclipse 7
  - Uses Maven 2 to manage libraries
  - Demonstrates basic Spring capabilities
- Contact has a Person, collection of Email, and collection of Phone objects
- Can use an XML file or database as repository
- User interface is via the console
- See READ_ME file under project folder

Dependency Management

- Manage collaboration (dependencies) between Plain Old Java Objects (POJOs)
  - Code to interfaces
  - Use Spring to instantiate specific interface implementations
    - Don’t need
      - InterfaceType anObject = new ClassThatImplementsInterfaceType()
  - Use Spring to provide specific interface implementations to your objects

External Configuration

- Configuration options
  - Properties files
  - XML configuration files
  - Annotations
- XML Configuration
  - Specify the creation of objects
  - Specify the dependencies between objects

Handling *Change*

- Change dependencies without changing code
  - Edit the configuration file
  - Create multiple configuration files

Testing Applications

- Easier to test
  - Use a test configuration file to create test objects and manage their dependencies
    - *Reuse objects across tests*
  - Test service layer objects by creating stub dependent objects
  - See: http://static.springframework.org/spring/docs/2.5.x/reference/testing.html

Separation of Responsibilities

- **Aspect-Oriented Programming (AOP)**
  - **http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html**
- **Aspect-Oriented Programming in Spring**
  - **See: http://static.springframework.org/spring/docs/2.5.x/reference/aop.html**
- **Put a concern (logic) that applies to many different objects in one single place**
  - ***Logging, security, performance testing, transaction management***

Implementing AOP in Spring

- **Advice**
  - **Several different ways to configure Advice objects**
    - ***Implement interfaces and XML configuration***
    - ***Use @AspectJ annotations***
    - ***Integrate with AspectJ***
  - **Advice is commonly applied to a method**
  - **Different ways to apply advice to an object**
    - ***Before advice***
    - ***After returning advice***
    - ***After throwing advice***
    - ***After advice***
    - ***Around advice***

Database Integration

- Spring provides a JDBC framework that removes much of the boiler-plate code
  - See: http://static.springframework.org/spring/docs/2.5.x/reference/jdbc.html
  - Configures the data source
  - Gets the connection
  - Creates the statement
  - Processes the result
  - Creates the business objects
  - Handles exceptions
  - Closes connection

- Spring can integrate with Hibernate, Java Data Objects (JDO), Java Persistence API (JPA), iBATIS, and other Object Relational Management (ORM) technologies

Spring JDBC

- **Spring JDBC framework takes advantage of Java 5**
  - **New classes in Spring 2.0 and 2.5 that use**
    - ***Variable arguments***
    - ***Auto boxing***
    - ***Covariant return types***
    - ***Generics***
  - **Can still use older classes and Java 1.4**

Spring JDBC Capabilities

- Simplified methods for querying and updating tables
  - Query to create a business object - http://static.springframework.org/spring/docs/2.5.x/reference/jdbc.html#jdbc-SimpleJdbcTemplate
  - Query for single value - http://static.springframework.org/spring/docs/2.5.x/reference/jdbc.html#jdbc-statements-querying
  - Insert business objects http://static.springframework.org/spring/docs/2.5.x/reference/jdbc.html#jdbc-simple-jdbc-insert-1

Example - Contacts Application

- Data source is configured and provided to other classes in the configuration file
- See classes ContactSearchService_DB and ContactDataAccess_DB
  - *Extend Springs' SimpleJdbcDaoSupport*
  - *See: http://static.springframework.org/spring/docs/2.5.x/api/org/springframework/jdbc/core/simple/SimpleJdbcDaoSupport.html*

Additional Spring Capabilities

- Transaction Management
  - "A transaction defines a logical unit of work that either completely succeeds or produces no result at all. A distributed transaction is simply a transaction that accesses and updates data on two or more networked resources, and therefore must be coordinated among those resources."
    see: http://archive.devx.com/java/free/articles/dd_jta/jta-1.asp
- Email
- Scheduling
- Web services and remoting
- Spring Web Model View Controller (MVC)
- Integration with Struts 2
  - see:
    http://static.springframework.org/spring/docs/2.5.x/reference/web-integration.html#struts and
    http://struts.apache.org/2.x/docs/spring-plugin.html

Where to Learn More

- Review the Spring documentation online
  - **Spring documentation - http://www.springsource.org/documentation**
  - **Spring API - http://static.springframework.org/spring/docs/2.5.x/api/index.html**
- **Spring support forum**
- **Review on Safari (KU Library)**
  - **Spring in Action, 2nd Edition, Manning Publishing, August 2007**
  - **Pro Spring 2.5, Apress Publishing, August 2008**