The Spring Framework

Simplifying J2EE

Who is Matt Raible?

- Developing websites since 1994 - Developing J2EE webapps since 1999

- Committer on several open source projects: Roller Weblogger, XDoclet, Struts Menu, Display Tag, AppFuse

- J2EE 5.0 and JSF 1.2 Expert Group Member

- Author: Spring Live (SourceBeat) and contributor to Pro JSP (Apress)

Spring Mission Statement

☐ J2EE should be easier to use.

☐ It's best to program to interfaces, rather than classes. Spring reduces the complexity of using interfaces to zero.

☐ JavaBeans offer a great way of configuring applications.

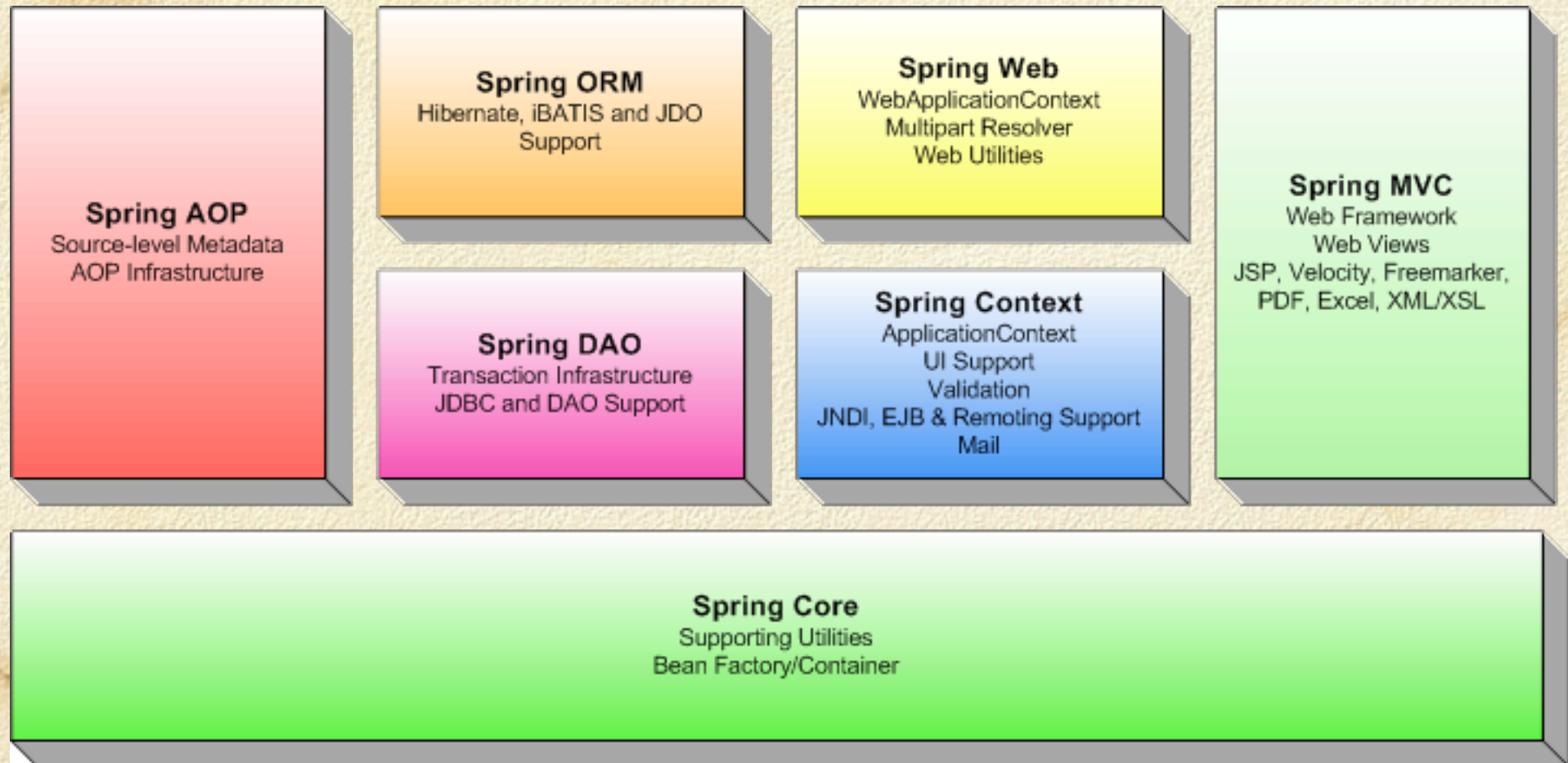☐ OO Design is more important than any implementation technology, such as J2EE.

Spring Mission Statement

- Checked exceptions are overused in Java. A framework shouldn't force you to catch exception you're unlikely to recover from.

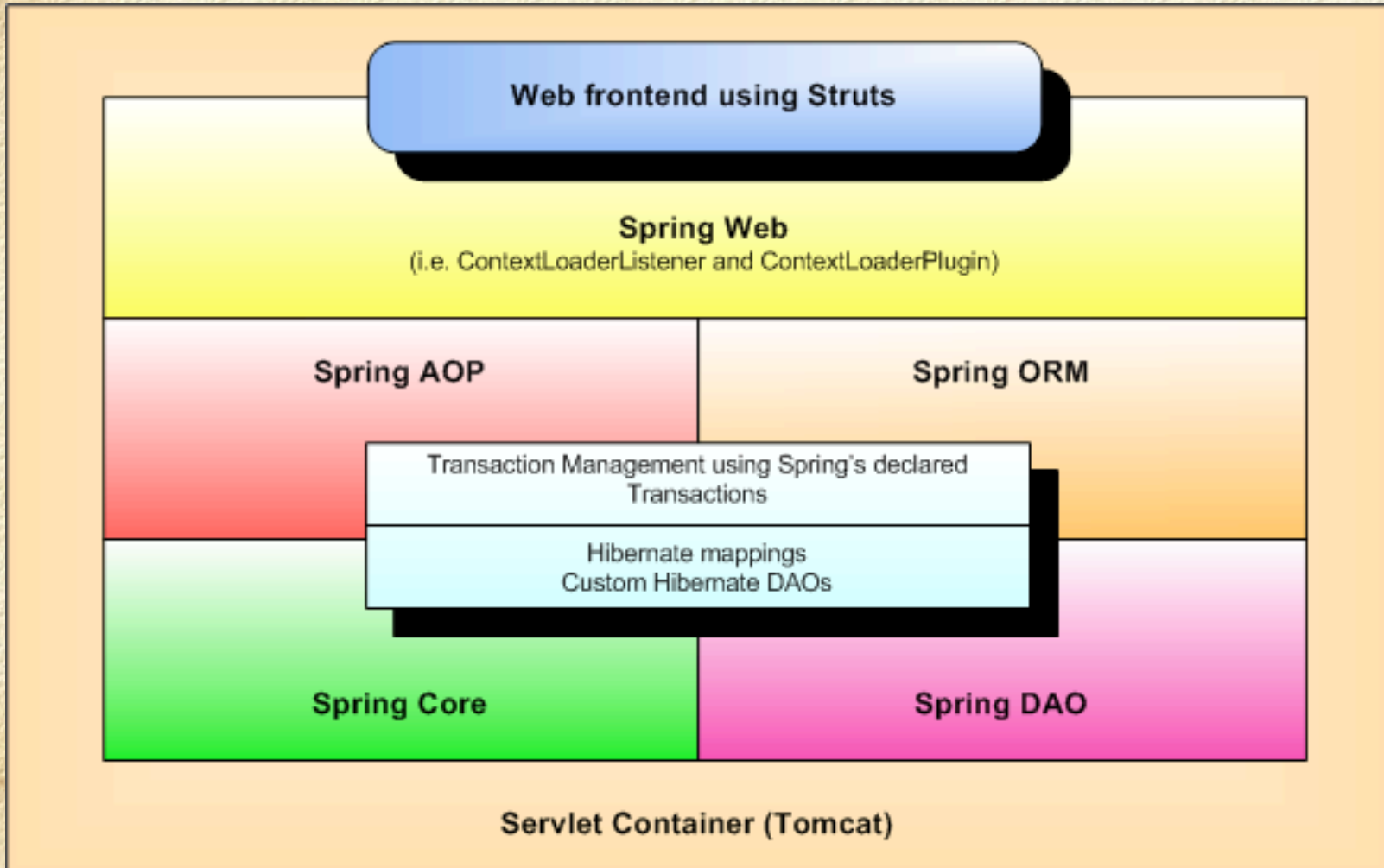- Testability is essential, and a framework such as Spring should help make your code easier to test.

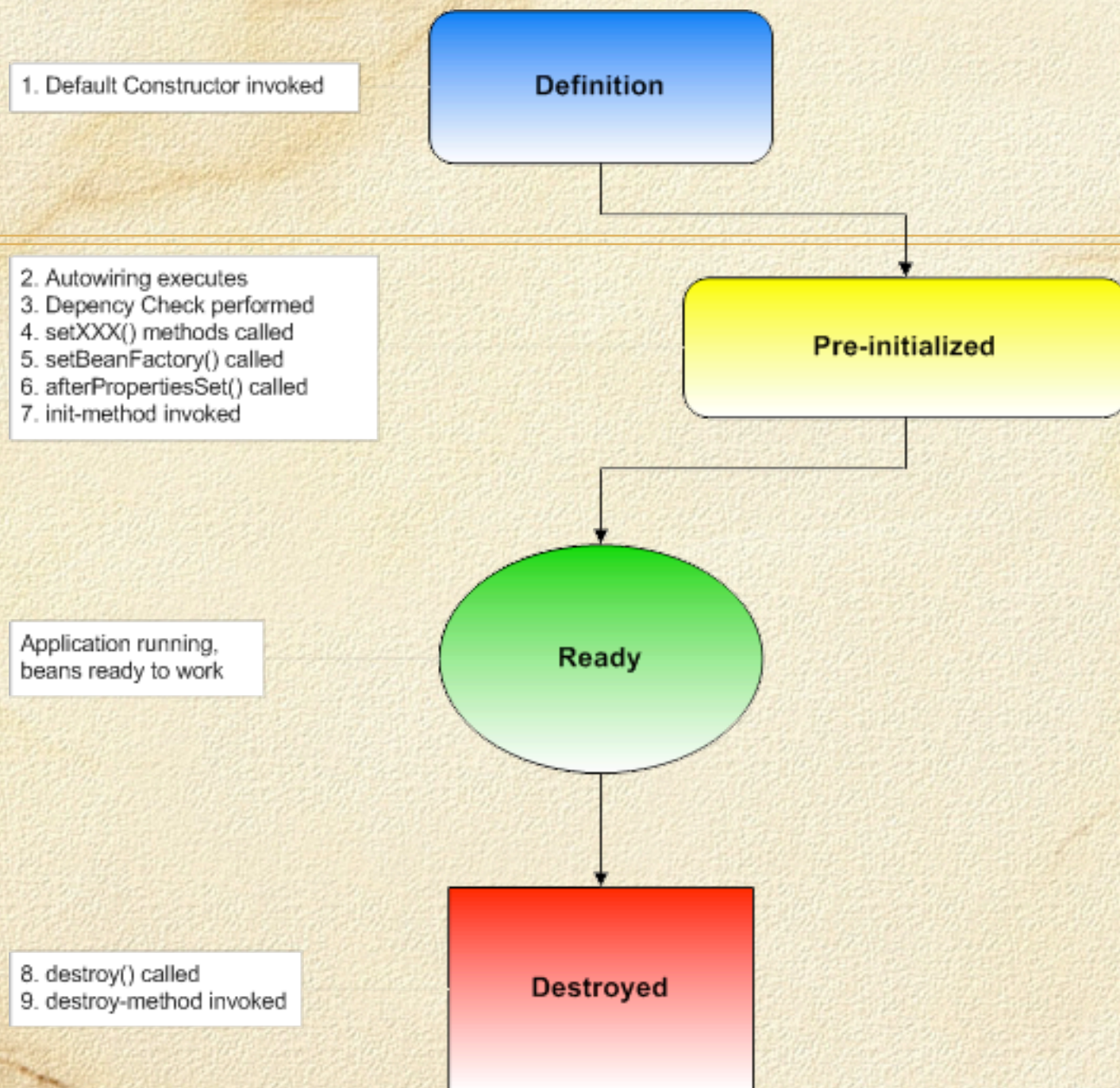# What is it?

- A J2EE Framework designed to make building applications easier.

- Provides a means to manage your business objects and their dependencies.

- Inversion of Control allows classes to be loosely coupled and dependencies written in XML.

What is it?

Spring AOP
Source-level Metadata
AOP Infrastructure

Spring ORM
Hibernate, iBATIS and JDO
Support

Spring Web
WebApplicationContext
Multipart Resolver
Web Utilities

Spring MVC
Web Framework
Web Views
JSP, Velocity, Freemarker,
PDF, Excel, XML/XSL

Spring DAO
Transaction Infrastructure
JDBC and DAO Support

Spring Context
ApplicationContext
UI Support
Validation
JNDI, EJB & Remoting Support
Mail

Spring Core
Supporting Utilities
Bean Factory/Container

Sample Architecture

The BeanFactory

Example: DAO and DataSource

- DAO has a setter or a constructor argument for the DataSource

- DataSource is configured as a dependency of the DAO

- Makes it easy to swap out implementations

- Tests can verify operations succeed on the interface

Spring MVC

- Front-Controller Servlet: DispatcherServlet

- Controllers loaded as beans from servlet's XML context files

- ContextLoaderListener loads other XML context files

- Many different Controller options: Controller, SimpleFormController, Wizard

Examples: Controllers

- The Controller interface is generally used to display data, not for forms.

- SimpleFormController easiest for forms.

- JSP Tags, Velocity, FreeMarker, PDF, Excel

- Validation: Commons Validator or Validation classes

Supports MVC Frameworks

- ContextLoaderListener loads beans into ServletContext

- Struts has ContextLoaderPlugin and ActionSupport classes

- WebWork has SpringObjectFactory

- Tapestry - override BaseEngine and stuff the context into the Global

- JSF - VariableDelegateResolver

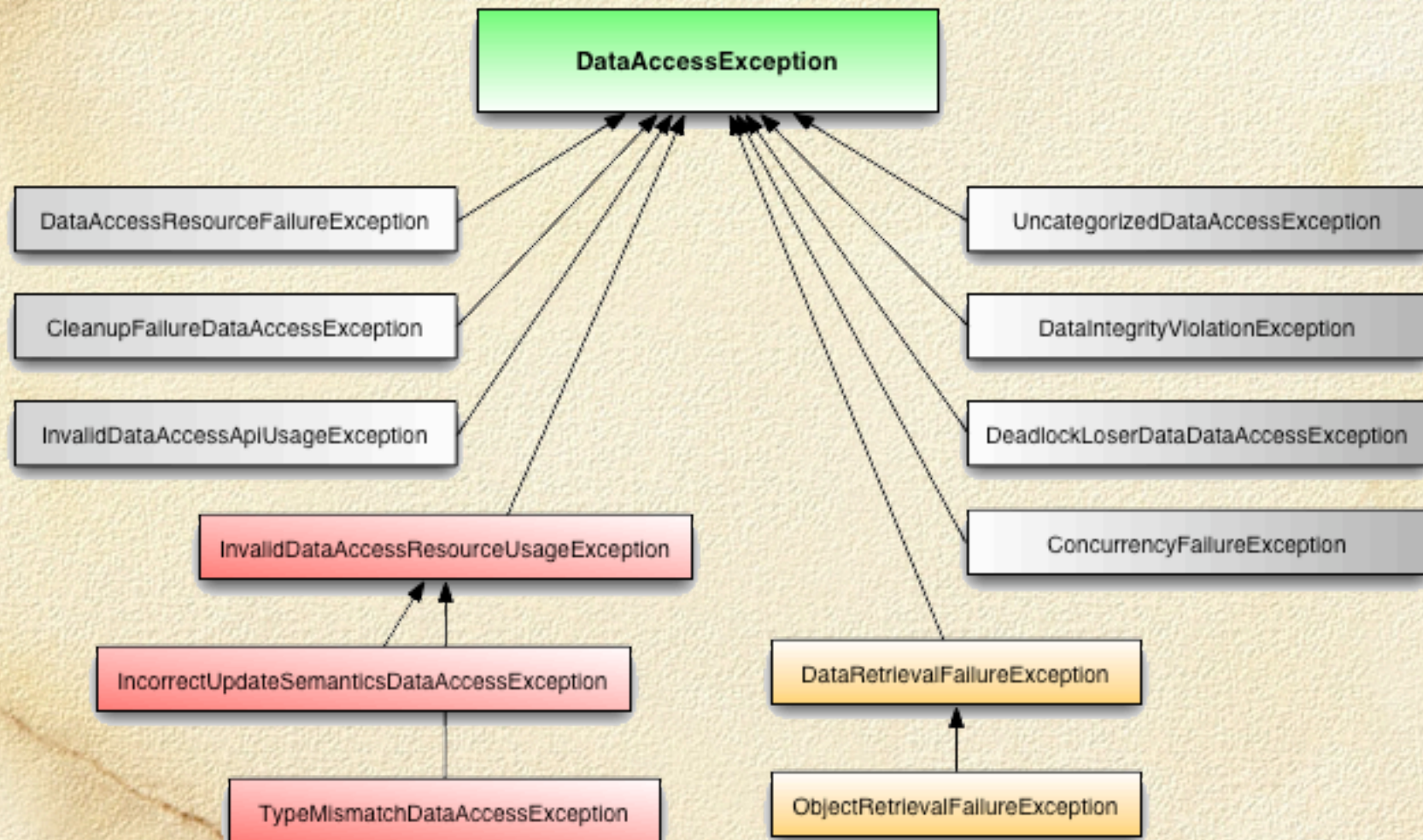# MVC Tips

☐ JSP 2.0 Tag files in JIRA - for syntax simplification

☐ Use SiteMesh - simple yet powerful

☐ AppFuse and Equinox have good integration examples

Data Access

- Spring JDBC Framework

  - No exceptions to catch or resources to close

- Base DAO and Template classes for many frameworks:

  - Hibernate, iBATIS, OJB, JDO - even TopLink

  - Hibernate 3 support in JIRA

DataAccessException

Data Access Examples

☐ UserDAOTest

☐ Hibernate

☐ iBATIS

☐ Spring JDBC

☐ JDO

Transactions

- Much easier to use than UserTransaction in J2EE

- CMT-like capabilities with XML and transparent AOP

- Supports Commons Attributes and JDK 5 Annotations

- Pluggable Transaction Managers, including JTA

Testing Spring Applications

- Easy to mock dependencies with EasyMock and jMock

- Load context in tests and bind dependencies as you would in production

- TIP: Use a static block or Spring classes in Spring Mock. In the org.springframework.test package.

AOP

- AOP - Reduces duplication among classes

- Interceptors make it easy to add before, around and after method advice

- Useful for caching, logging and security

  - EHCache, Performance/Tracing interceptors, Acegi for Security

App Server Support

- Spring should run on any app server.

- Hooks into many app server's Transaction Manager.

- Survey: which app servers are you running it on?

Tools

- Spring IDE Plugin For Eclipse

- BeanDoc

- Screenshots

Spring vs. J2EE

- It's not really a comparison, Spring builds off J2EE and merely makes the APIs easier

- It's still using many of J2EE's and Java's underlying infrastructure

- EJB vs. Spring Managed POJOs

- Opinions and Hesitations?

Questions

- Practical applications of AOP?

- Multithreading with singleton beans - is it safe?

- Tips and Tricks?

- What are the Gotchas?

Questions, cont.

- What environment is the best match for Spring? Where will it not perform well?

- Maintaining inter-class dependencies - tools?

- Is Spring a replacement for Struts?

- Can I use Spring with EJBs and Hibernate for CMT?

Resources

- www.springframework.org and forum.springframework.org

- Interface21 offers commerical support at www.springframework.com

- Spring Live, Spring in Action and Professional Spring Development

- AppFuse (appfuse.dev.java.net) and Equinox (equinox.dev.java.net)

This Presentation

http://equinox.dev.java.net/SpringFramework.pdf