Spring in the Enterprise

Building Scalable Java Applications

Gordon Dickens

Certified Spring Instructor & Mentor
co-Author Spring Roo in Action
Chariot Solutions

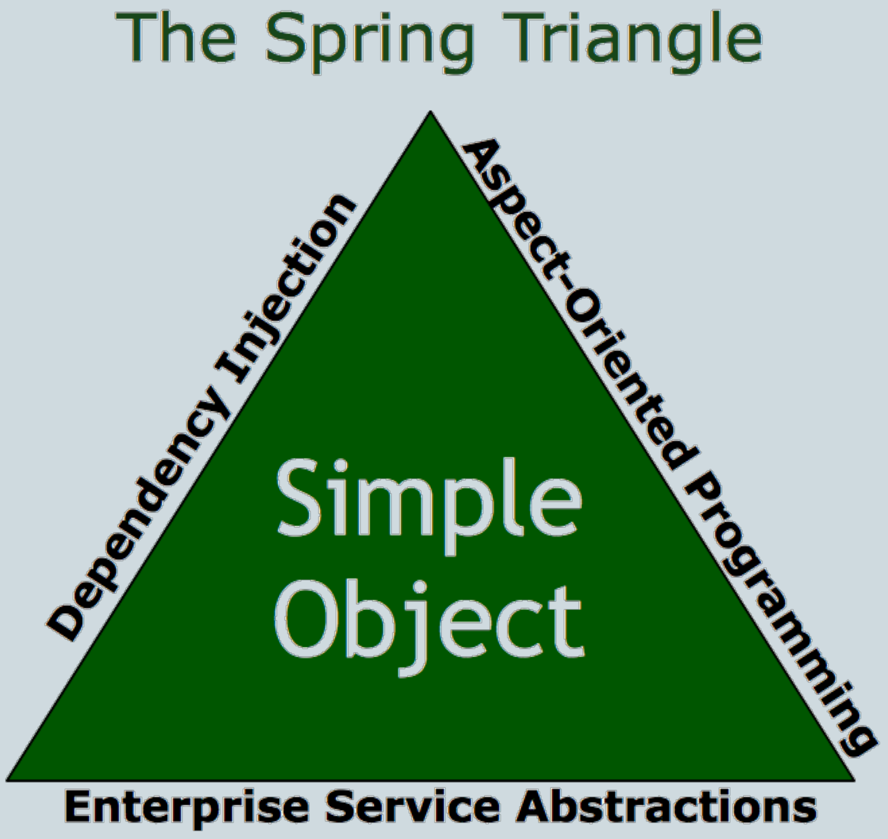
email: gdickens@chariotsolutions.com
Twitter: twitter.com/gdickens
Blog: gordondickens.com
Roo in Action: manning.com/dickens

- Solutions Provider
  - Open Source Project
  - chariotsolutions.com
- Seasoned Application Architects
- Education
  - chariotsolutions.com/education
  - Spring
  - Maven
  - etc
- Techcasts
  - Podcast with open source industry leaders
  - techcast.chariotsolutions.com/

Session Topics

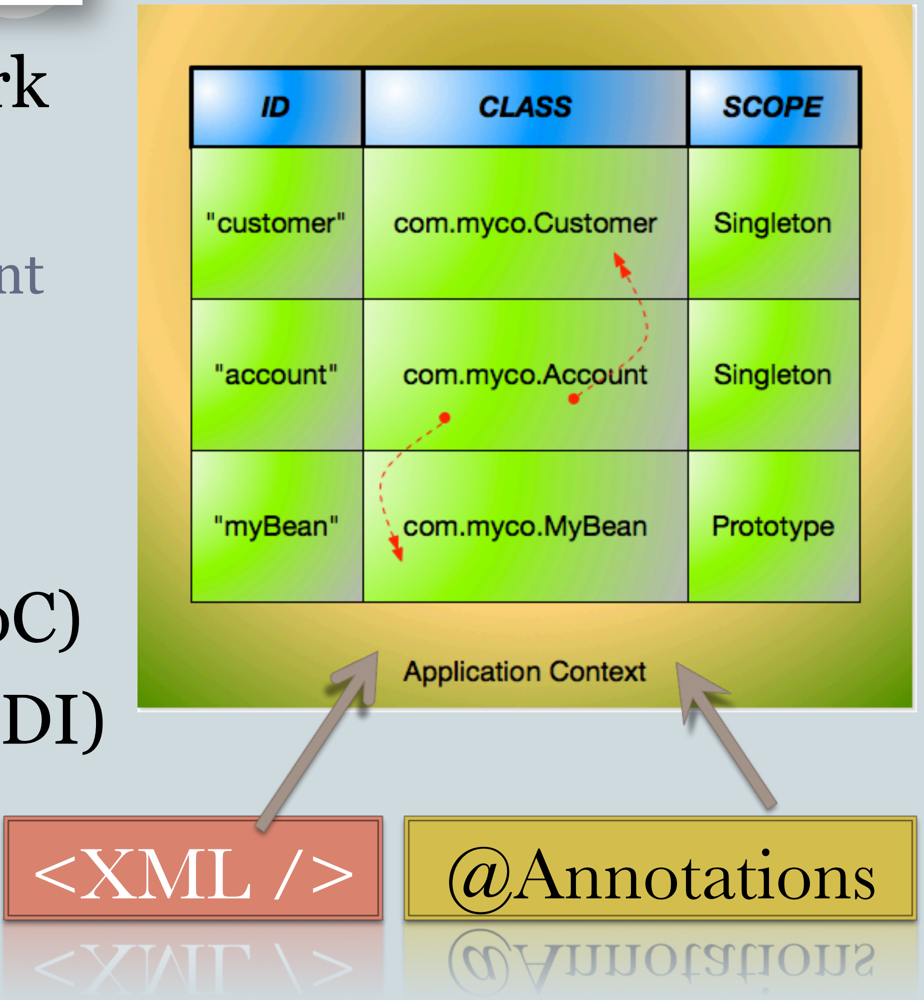
1. What is Spring & Why use it?
2. Architecting in Spring
3. Enterprise Spring
4. Spring & SOA
5. Modular Spring w/ OSGi
6. Spring RAD & Tools

Spring Theory



What is Spring? - The Basics

- Open Source Framework
- Bean Container
  - Bean Lifecycle Management
  - Bean Scope
  - Post Processing Hooks
  - Event Processing
- Inversion of Control (IoC)
- Dependency Injection (DI)
  - Centralized Configuration
  - Annotation Support



ID	CLASS	SCOPE
"customer"	com.myco.Customer	Singleton
"account"	com.myco.Account	Singleton
"myBean"	com.myco.MyBean	Prototype

What is Spring? - Features

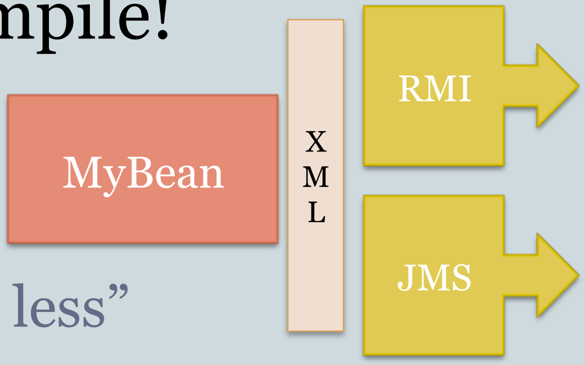
- Core
  - Standard Java Apps
- Spring MVC
  - Powerful controller config
  - Flexible data formatting
  - RESTful
- Rich Web Applications
  - Web Flow
  - BlazeDS (Flex)
  - Spring Faces (JSF)
  - Spring JS
  - Spring JSP/JSTL
- Web Flow
  - Stateful Page Flows
- Enterprise Integration
  - JMS
  - Remoting
  - Spring Integration
  - Spring Batch
  - Web Services

Why Use Spring? – Core Benefits

- POJOs
  - Our Java Beans
  - GOAL: Beans are technology & platform agnostic
- Bean Lifecycle & Dependency Management
  - no more “new myClass()”
- Standard Java Technologies
  - JSR-303, JSR-250, JSR-317, etc.
- Data Management
  - Conversion, Marshaling, Formatting
  - Data from WS, UI, Remoting, Integration, JMS, etc.

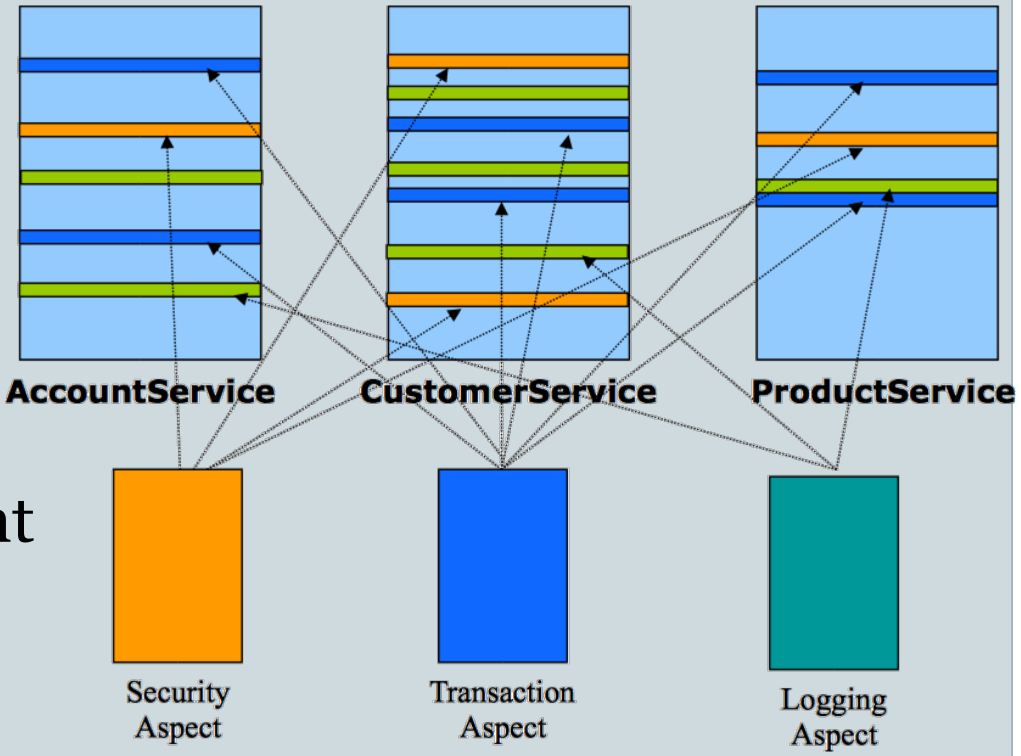
Why Use Spring? – Scalable

- Configuration Flexibility
  - XML, Annotations, JavaConfigX
- Implementation Changes w/o Recompile!
- Plumbing handled for us
  - If it "sucks" in Java, Spring makes it "suck less"
- Aspect Oriented Programming (AOP)



Why use Spring? - AOP

- Spring uses AOP to perform common operations
  - Transactional processing
  - Data Access & Exception Wrapping
  - Security
  - etc.



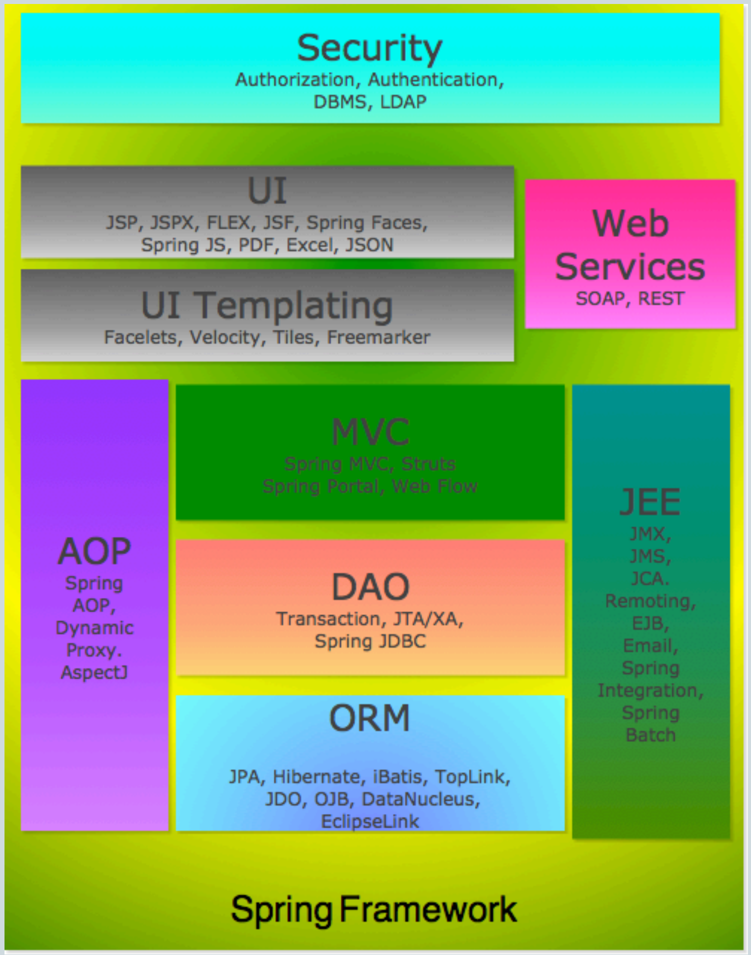
- Write Aspects to Augment POjOs

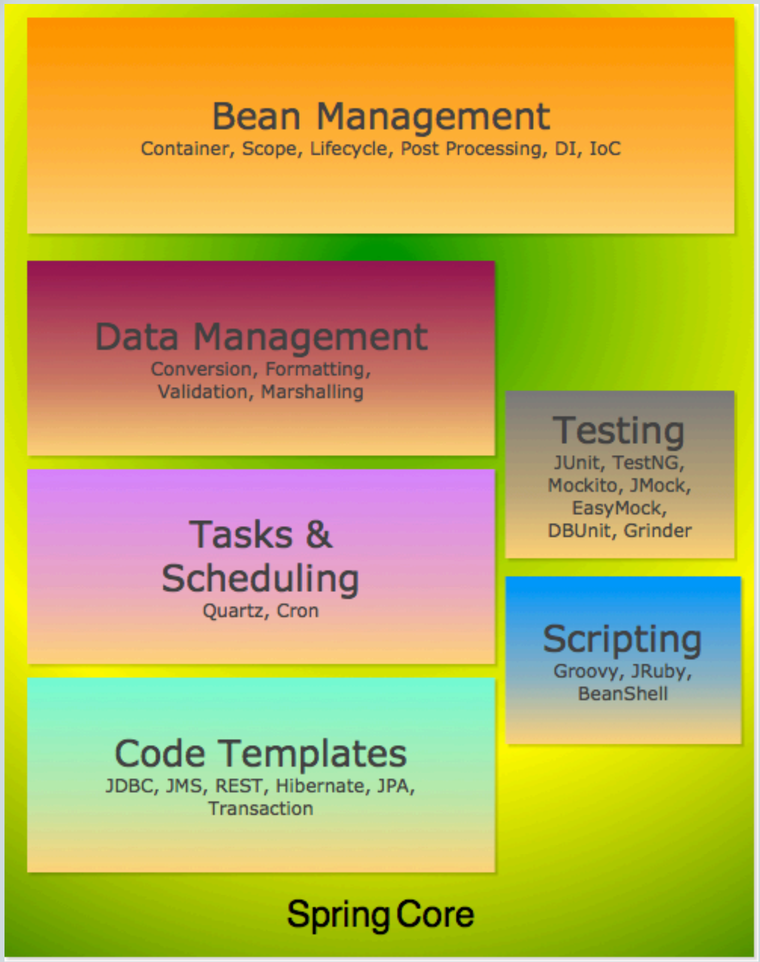
- Spring AOP and AspectJ

Why use Spring? - Enterprise Features

- JEE Server Support
  - GlassFish
  - JBoss
  - Tomcat
  - WebLogic
  - Websphere
- JNDI Access
  - Queues
  - DB Resources
- JTA/XA
- Spring Security
  - Authentication
  - Authorization
- SOAP & RESTful WS
- Spring Integration
- Spring Batch
- Remoting
  - RMI
  - HTTP

Why use Spring? – Comprehensive

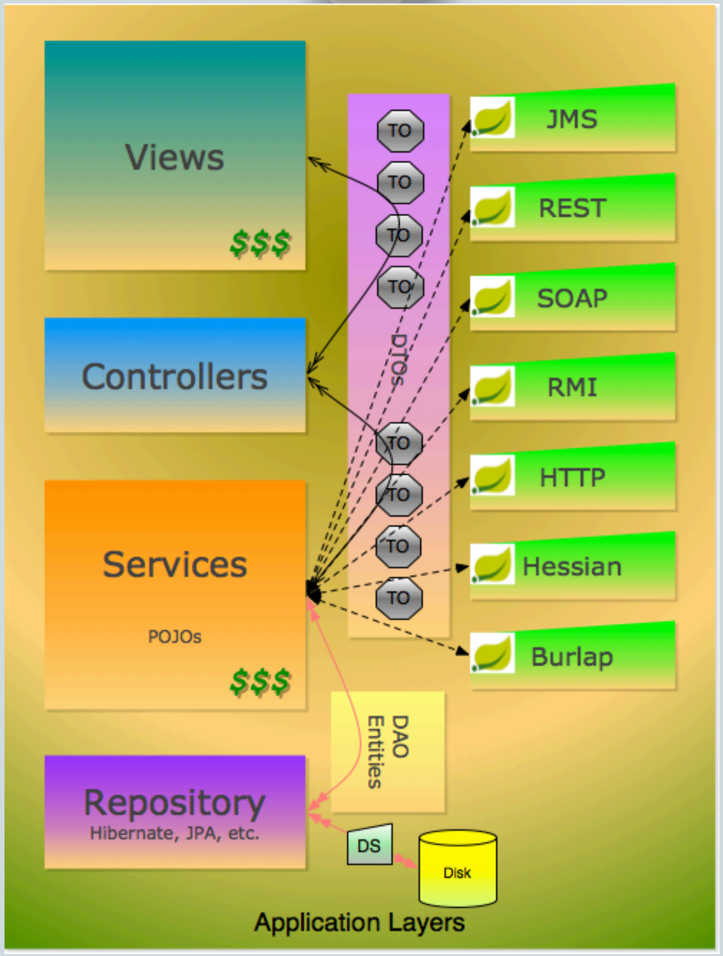

Security
Authorization, Authentication, DBMS, LDAP
UI
JSP, JSPX, FLEX, JSF, Spring Faces, Spring JS, PDF, Excel, JSON
Web Services
SOAP, REST
UI Templating
Facelets, Velocity, Tiles, Freemarker
AOP
Spring AOP, Dynamic Proxy. AspectJ
MVC
Spring MVC, Struts Spring Portal, Web Flow
JEE
JMX, JMS, JCA. Remoting, EJB, Email, Spring Integration, Spring Batch
DAO
Transaction, JTA/XA, Spring JDBC
ORM
JPA, Hibernate, iBatis, TopLink, JDO, OJB, DataNucleus, EclipseLink
Spring Framework


Bean Management
Container, Scope, Lifecycle, Post Processing, DI, IoC
Data Management
Conversion, Formatting, Validation, Marshalling
Testing
JUnit, TestNG, Mockito, JMock, EasyMock, DBUnit, Grinder
Tasks & Scheduling
Quartz, Cron
Scripting
Groovy, JRuby, BeanShell
Code Templates
JDBC, JMS, REST, Hibernate, JPA, Transaction
Spring Core

Session Topics

1. What is Spring & Why use it?
2. Architecting in Spring
3. Enterprise Spring
4. Spring & SOA
5. Modular Spring w/ OSGi
6. Spring RAD & Tools

Architecting in Spring



Architecting Services with POJOs

- Plain Old Java Objects
- Create Services as POJOs
- Expose Services via Configuration
- Do Not:
  - import, extend or inject framework or technology specific classes

Gordo's Rule

- It's Not a POJO if you have plumbing/framework references in the "imports".

Session Topics

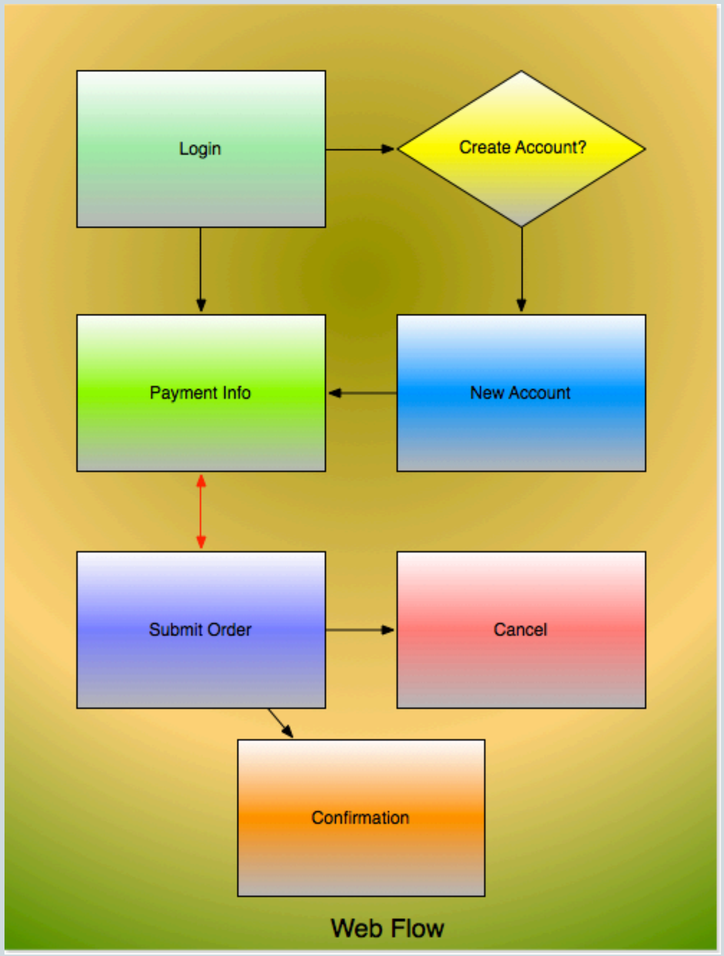
1. What is Spring & Why use it?
2. Architecting in Spring
3. Enterprise Spring
4. Spring & SOA
5. Modular Spring w/ OSGi
6. Spring RAD & Tools

Enterprise Spring Projects

- Spring Web Flow
- Spring Security
- Spring Integration
- Spring Batch

Spring Web Flow

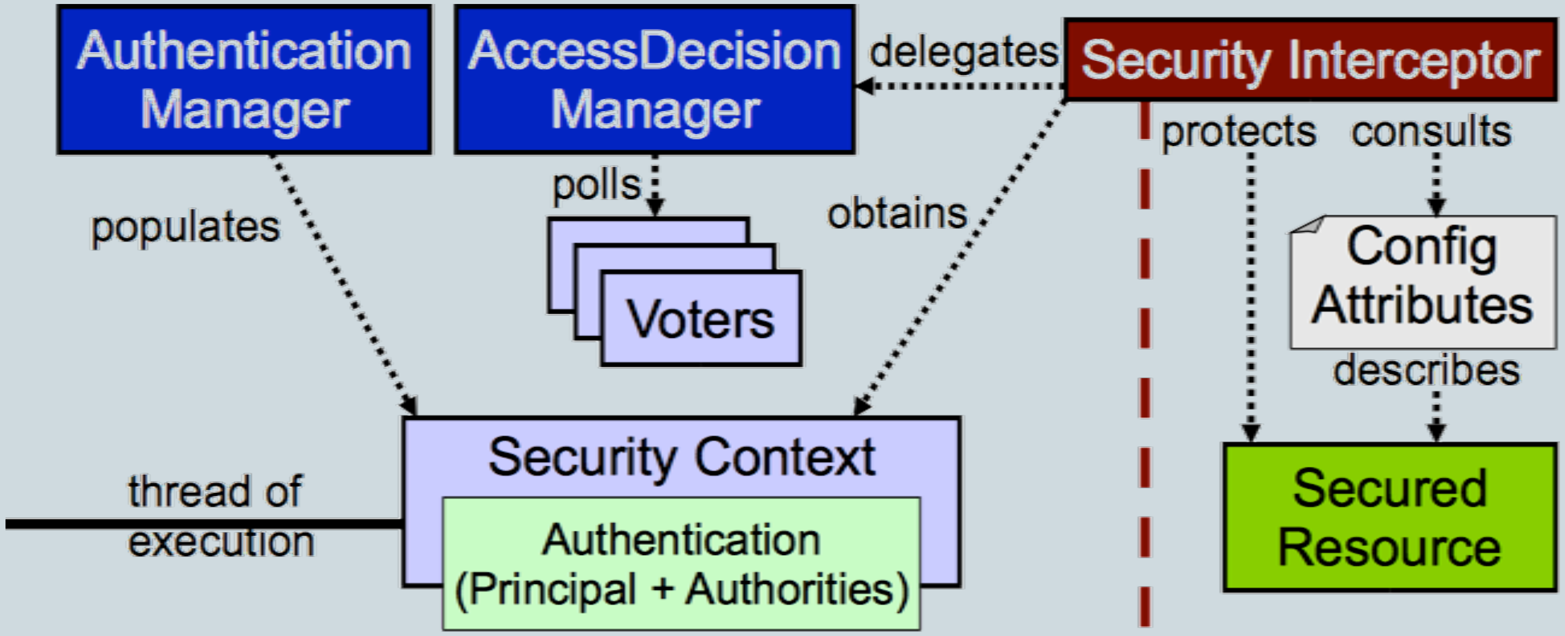
- Stateful Web Page Flows
- Old “wizard” style
- Plugs into Spring MVC Controllers
- Use-Case:
  - Site Registration & Payment



Spring Security

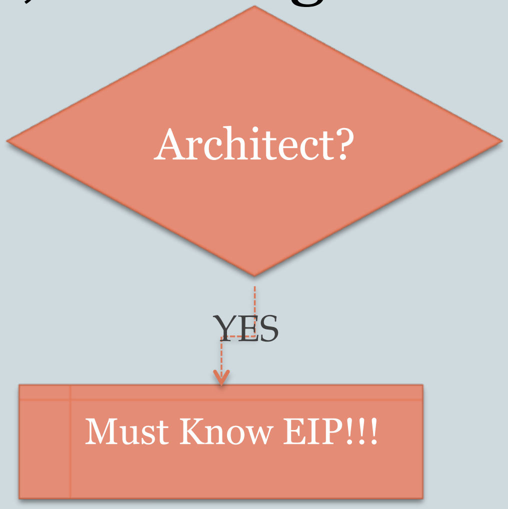
- Separates Authentication from Authorization
- Simple to configure
- Feature Rich
  - Users, Roles, Groups, Voters, ACL
- Authentication
  - DBMS – existing or custom
  - LDAP
- Authorization
  - UI Components
  - Methods
  - URLs

Spring Security Overview



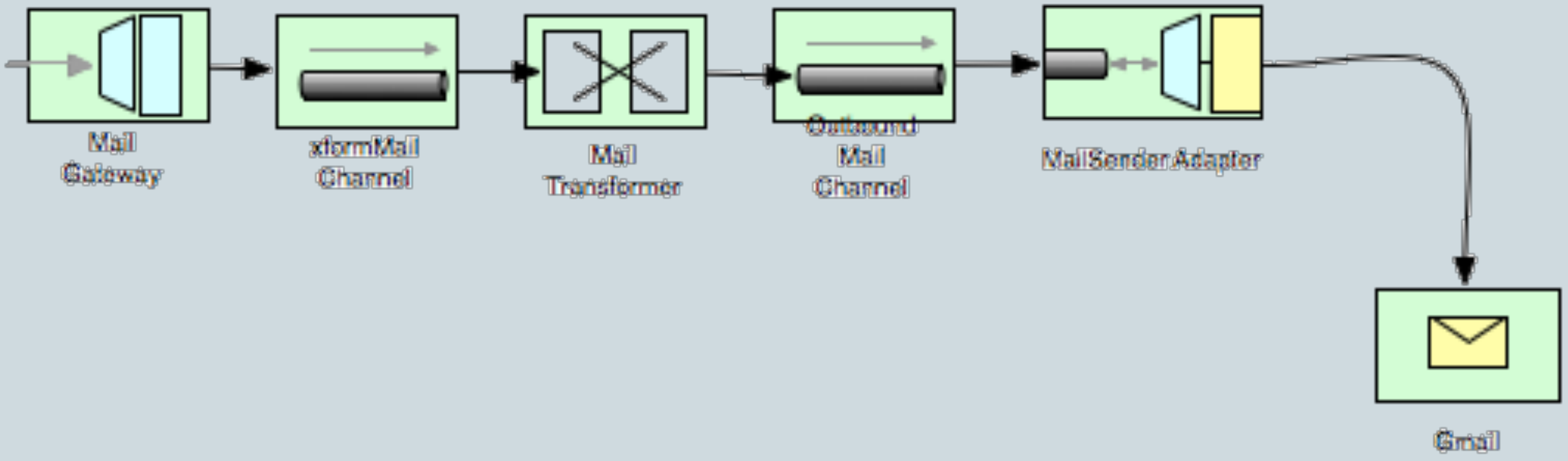
Spring Integration

- Enterprise Messaging
- Message Driven Solution
- Message Routing, Transformation, Filtering
- "Enterprise Integration Patterns"
  - by Hohpe & Wolfe – eaipatterns.com
- Focus on the Payload!



Mail Sender

- Use-Case: Trigger Mail to send to Gmail
- Given `MessageRecipient` Bean
- Advantages?

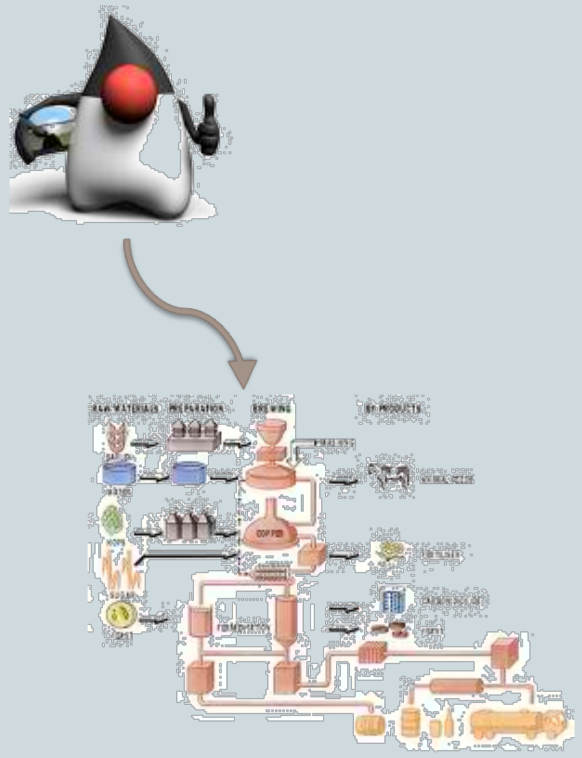


Spring Batch

- Large Dataset Processing

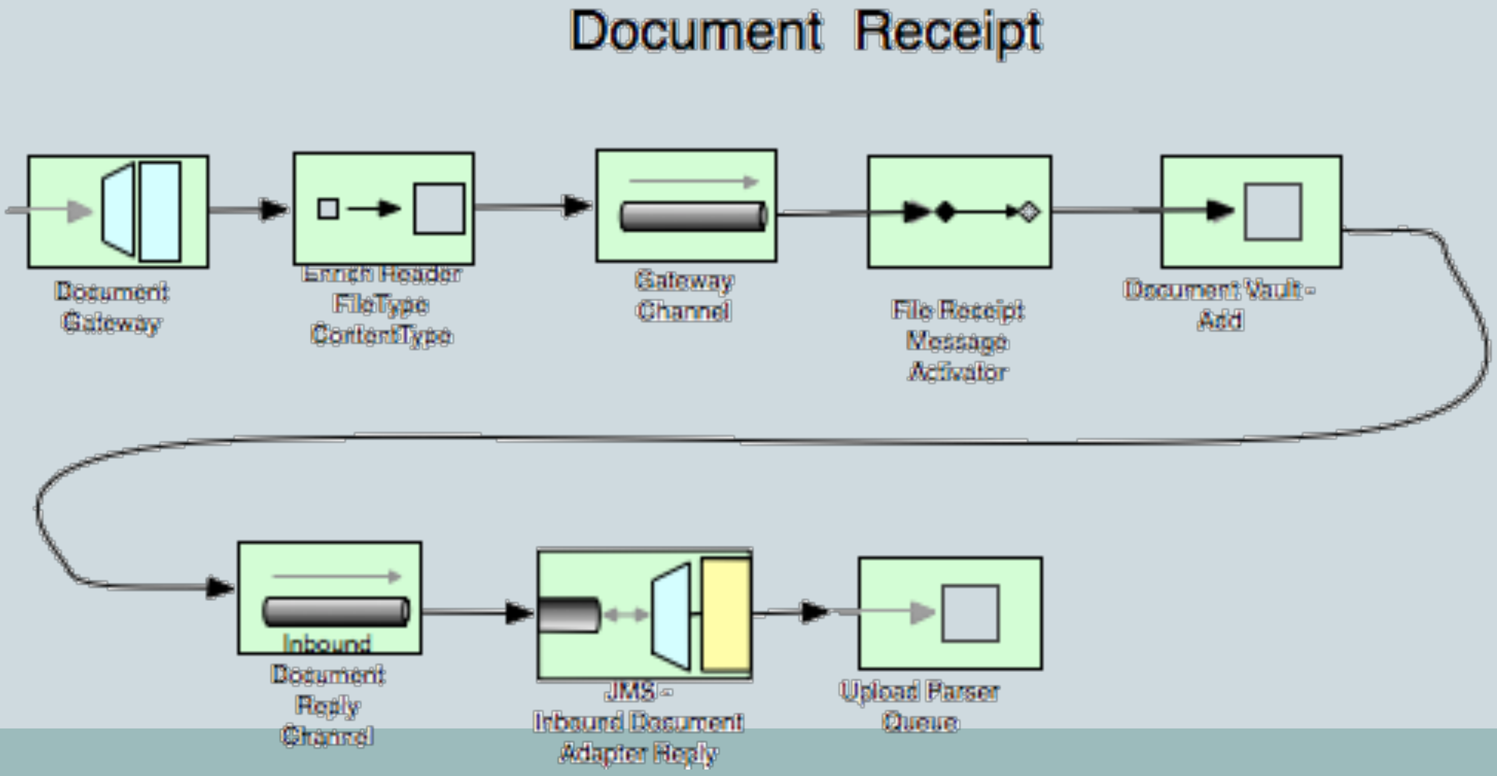
- Offline and Online
- Persistent Job states
- Transaction size configuration
- Job Segment Recovery
- Scheduled or Triggered Jobs
- Web Console

- Use in simple `psv main` app
- Use in enterprise apps



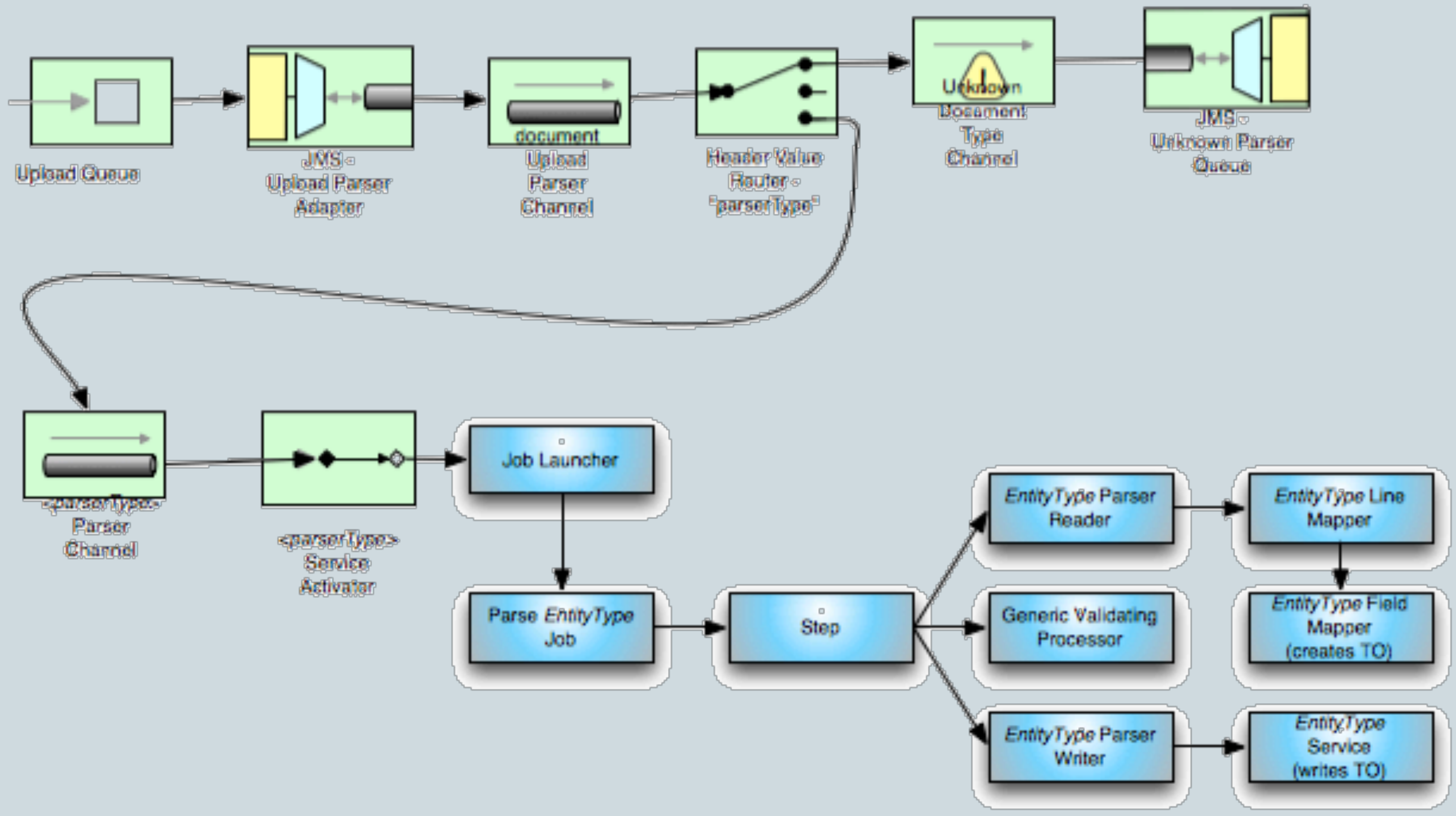
Spring Integration Flows

- Use-Case: File Uploading from UI, parse & persist
  - Large Files
  - File Indexing & Searching


Document Receipt
Document Gateway
Enrich Header FileType ContentType
Gateway Channel
File Receipt Message Activator
Document Vault - Add
Inbound Document Reply Channel
JMS - Inbound Document Adapter Reply
Upload Parser Queue

Spring Integration & Batch

- Receive notification, retrieve file, parse & persist


Upload Queue
JMS Upload Parser Adapter
document Upload Parser Channel
Header Value Router "parserType"
Unknown Document Type Channel
JMS Unknown Parser Queue
<parserType> Parser Channel
<parserType> Service Activator
Job Launcher
Parse EntityType Job
Step
EntityType Parser Reader
Generic Validating Processor
EntityType Parser Writer
EntityType Line Mapper
EntityType Field Mapper (creates TO)
EntityType Service (writes TO)

Session Topics

1. What is Spring & Why use it?
2. Architecting in Spring
3. Enterprise Spring
4. Spring & SOA
5. Modular Spring w/ OSGi
6. Spring RAD & Tools

Spring & SOA

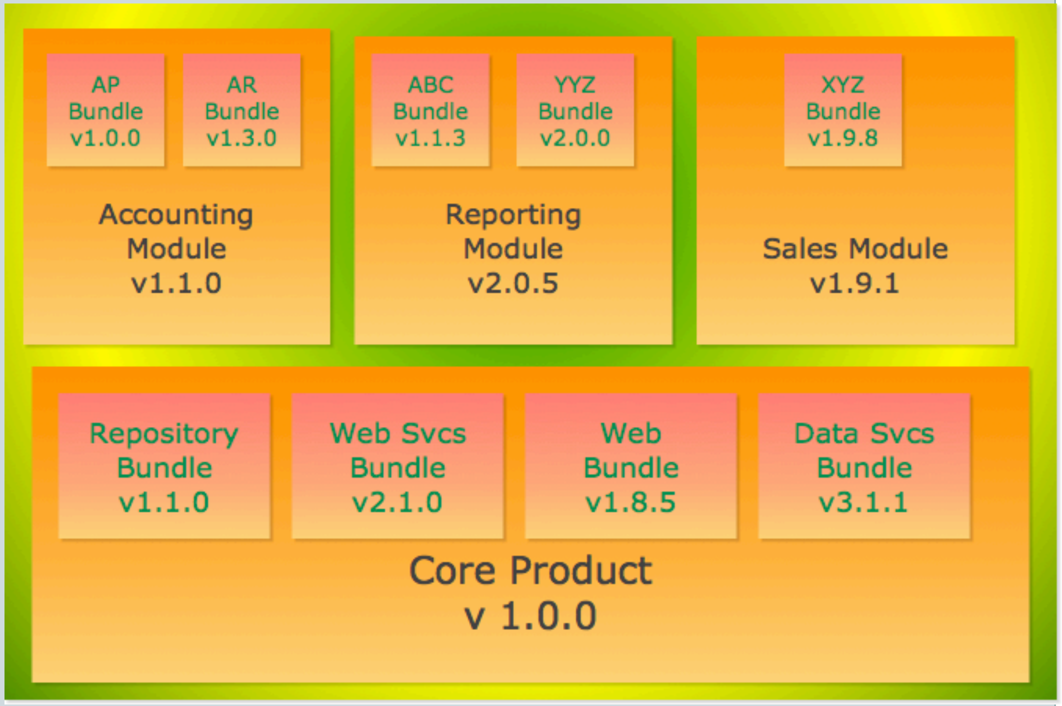
- Spring SOAP Web Services
- Spring RESTful Web Services
- Spring Remoting
  - RMI
  - HTTP
  - Hessian & Burlap
  - Corba IIOP
  - EJB Invocation

Session Topics

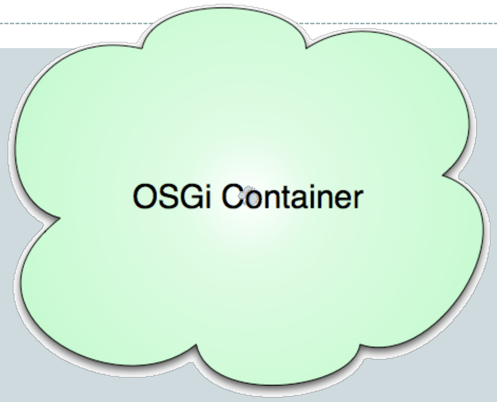
1. What is Spring & Why use it?
2. Architecting in Spring
3. Enterprise Spring
4. Spring & SOA
5. Modular Spring w/ OSGi
6. Spring RAD & Tools

Modular Java with OSGi

- Disciplined paradigm
- Reduces:
  - Classpath hell!
- Controlled QA
- Only deploy modules needed or bought



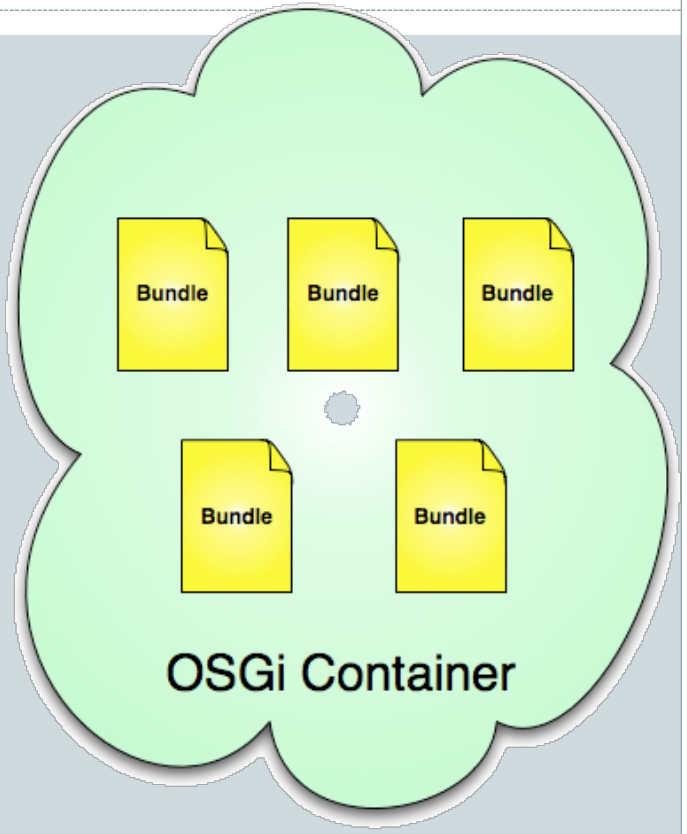
OSGi Container



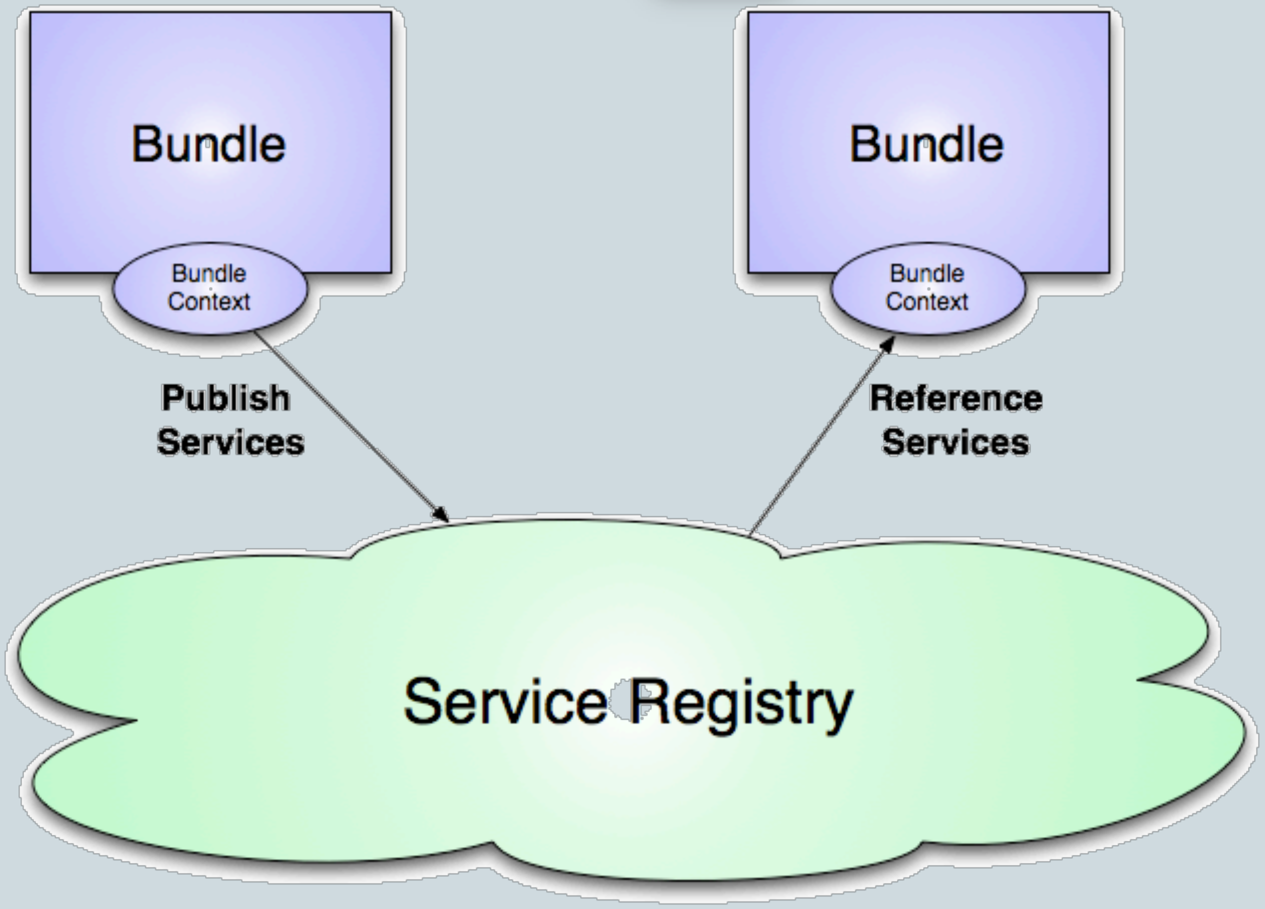
- A dynamic component platform
- Configures, starts & stops components at runtime
- Provides a set of Management APIs and lifecycle events
- Products:
  - Apache Felix
  - Eclipse Equinox
  - Eclipse Virgo
  - Knopflerfish

OSGi Bundles

- Bundle
  - Set of classes & resources deployed as a versioned module
  - Can depend on other bundles by version ranges
- A Java archive
  - Includes special Manifest entries in `META-INF/MANIFEST.MF`
- Deployed to container
- Can be extended with Fragments

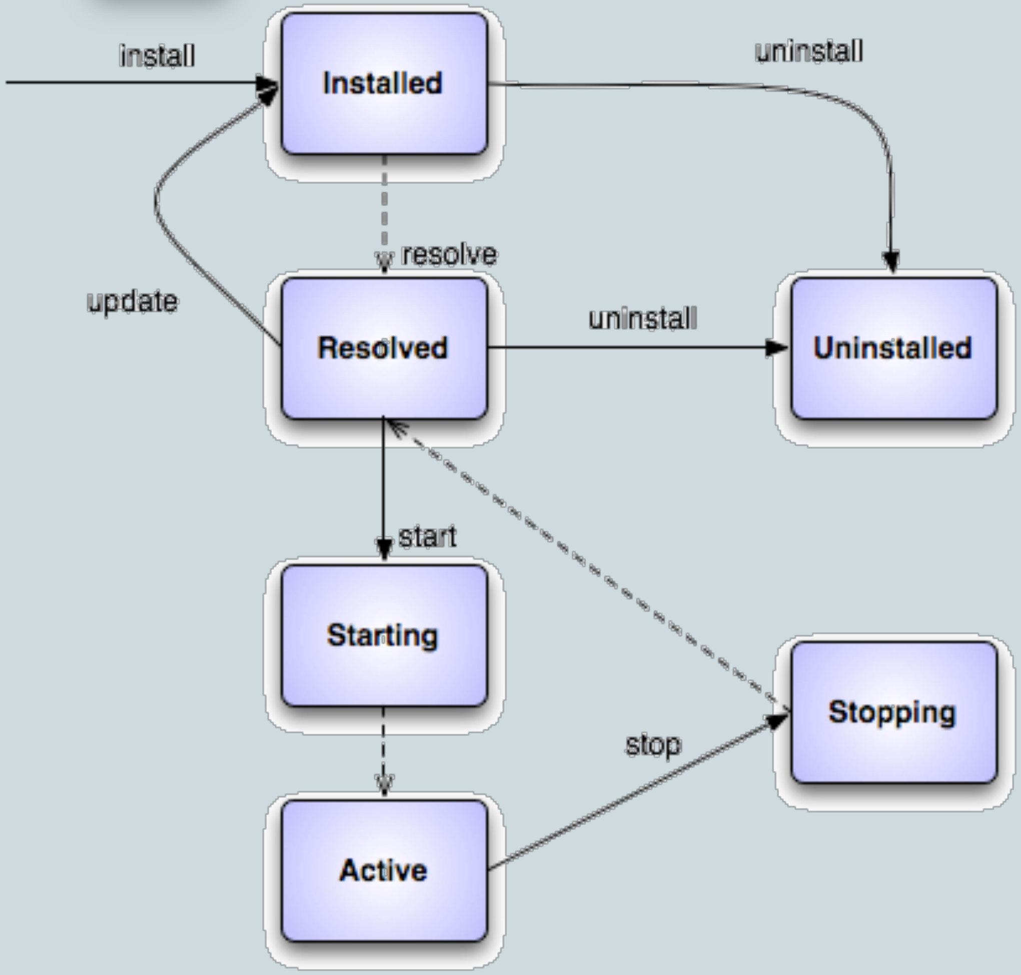


Service Registry



Bundle Lifecycle

- Installed
- Resolved
- Starting
- Active
- Stopping
- Uninstalled



Modular Java - OSGi

- Eclipse Gemini
  - Formerly: Spring Dynamic Modules
  - Spring Blueprint service – Bundle extension for Bean Context
  - Bundle activation
  - Service registration
- Eclipse Virgo
  - Formerly: Spring dm Server
  - Full OSGi server platform
  - Bundle Provisioning
  - Console Management & Logging
  - Modular UI Components
  - Module Scoping

Bundle Manifest

- Contains identifying information
- Lists exported and imported packages
- Can provide a Bundle Activator
- Can list constraints such as JDK version, etc.

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-SymbolicName: maven-osgi-demo-services
Bundle-Name: Maven OSGi Demo – Services
Bundle-Version: 1.0.0.SNAPSHOT
Export-Package: com.chariot.services
Bundle-Activator: com.chariot.services.ServiceActivator
Import-Package:
com.chariot.services,com.chariot.services.impl,com.cha
 riot.services.interfaces,org.osgi.framework;version="1.3"
```

How does Spring help?

- Each Bundle has a Spring Context
- Services Exposed / Imported via OSGi namespace

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:osgi="http://www.springframework.org/schema/osgi"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
      http://www.springframework.org/schema/osgi
      http://www.springframework.org/schema/osgi/spring-osgi-1.0.xsd">


<osgi:service id="exampleBeanService"
      ref="exampleBean"
       interface="com.chariot.demo.bean.ExampleBean"/>

</beans>
```

Eclipse Virgo

- OSGi Server Platform built upon Eclipse Equinox
- Group bundles for deployment into modules
- Module deployment via PAR and PLAN files
- Provisioning
  - Locate bundles in repositories
  - Local or Remote repositories
- Admin Console
  - Deploy & Manage Artifacts
  - Diagnostic dumps
  - Bundle wiring
- Web Server
  - Supports standard WAR files
  - Ships with Tomcat

VIRGO from eclipseRT

www.eclipse.org/virgo/

Session Topics

1. What is Spring & Why use it?
2. Architecting in Spring
3. Enterprise Spring
4. Spring & SOA
5. Modular Spring w/ OSGi
6. Spring RAD & Tools

RAD Tools

- Roo
  - Build and configure Java objects
  - Interactive management
  - Demonstrates Best Practices
  - Database Reverse engineering
- Grails
  - Built on the Spring & Hibernate platform
  - Dynamic coding with Groovy language
  - Completely supports Java libraries
  - Hundreds of plug-ins

SpringSource Tool Suite

- Built on the very popular Eclipse IDE
- Provides Spring specific features
- View components
  - Bean Dependency Relationships
  - Integration Flows
- Supports Roo, Grails & Groovy

Summary

- Comprehensive framework
- Java Standards
- Extensible
- Focus on Problem Domain
- Configure System Domain
- Designed RI for Eclipse OSGi Modular Projects