OSGi Alliance
Community Event

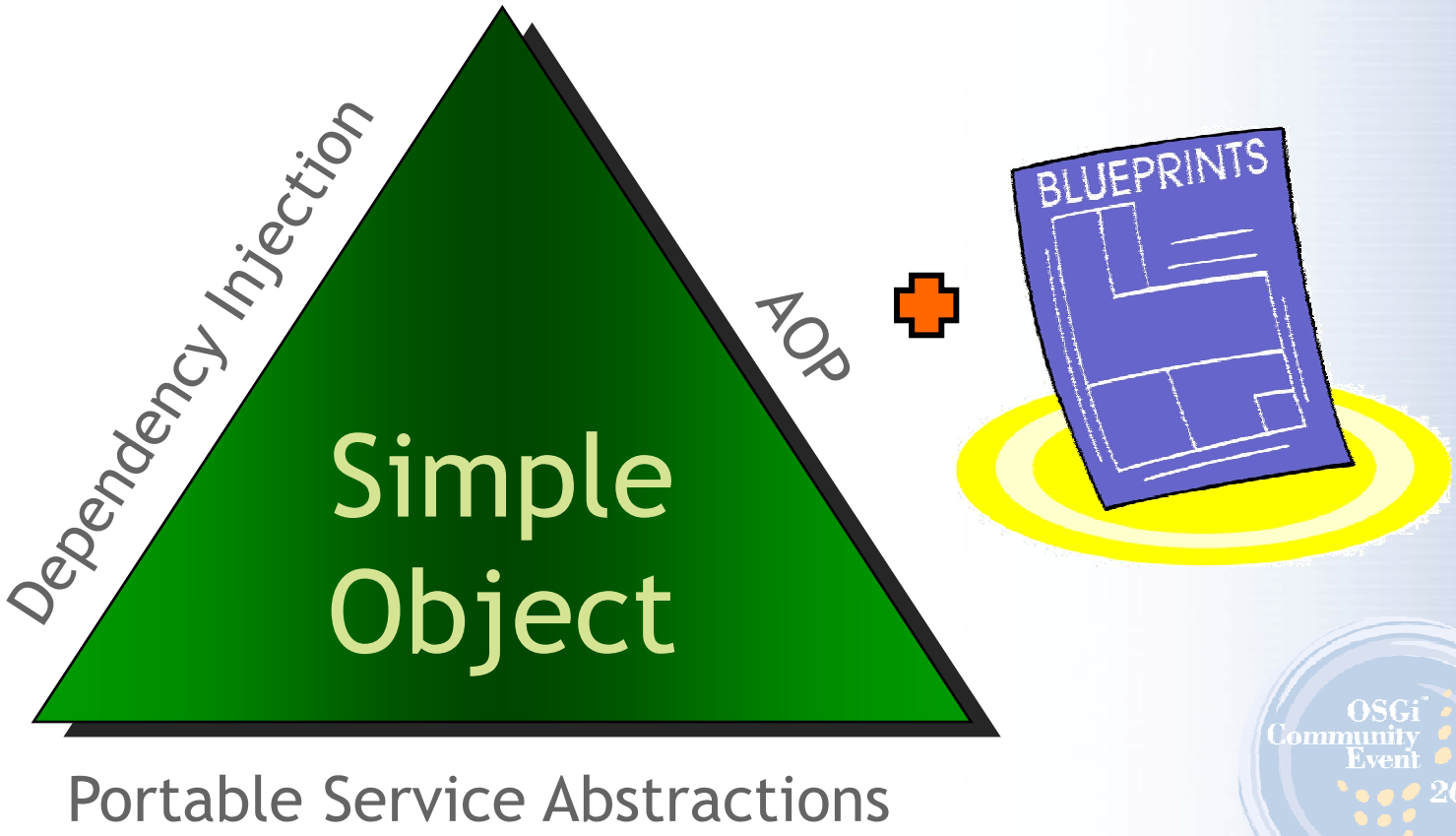
Spring Dynamic Modules by Example

Martin Lippert
Gerd Wütherich

Agenda

- Spring
  - Overview of the Spring framework
  - A short example
- Spring Dynamic Modules
  - Overview of Spring DM
  - A short example
- What‘s new in Spring DM 1.1
  - Web support
  - A short example

Key Spring Concepts



The Heart of Spring

- Lightweight container
  - Full stack, simple object based application development
- Works in any environment
  - web-app, ejb, integration test, standalone

- Provides…
  - a powerful object factory that manages the instantiation, configuration, decoration and assembly of objects

Spring Framework

- Dependency injection
- Aspect Oriented Programming support
- Technology abstractions
  - Integration with persistence technologies (JDBC, Hibernate)
  - Web application support Spring MVC, JSF and Struts
  - Enterprise service abstractions
    - Transactions
    - Messaging
    - …

Spring: An Example

- The Order Service…
  - … can be used to place a stock order for a customer.
  - … is accessible via RMI.
- The Customer Service…
  - … can be used to retrieve customer information.

Intermediate result

- Spring simplifies the development of enterprise applications
- **“De facto standard for JEE development”**

Spring Dynamic Modules for OSGi Service Platforms

- Formerly known as “Spring-OSGi”
- An open source project in the Spring portfolio
  - led by SpringSource
  - committers from BEA and Oracle
- http://www.springframework.org/osgi
- Allows to implement Spring Applications on top of an OSGi framework
- Works with Equinox, Felix, Knopflerfish

Project Objectives

- Bring the benefits of OSGi:
  - modularity
  - versioning
  - lifecycle support
- To Spring-based enterprise application development

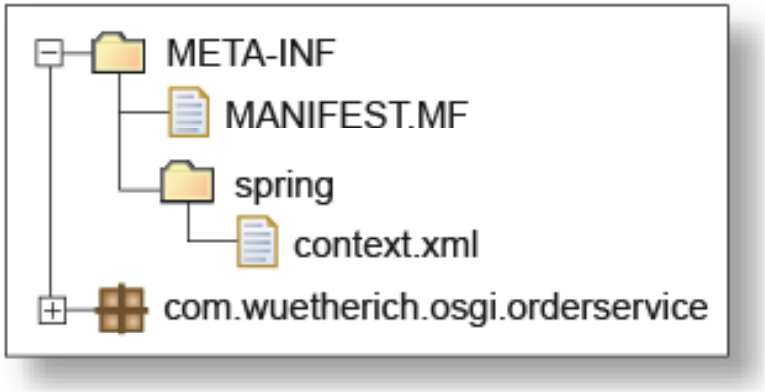
Challenges

- Complexity
  - Bring the simplicity of POJOs to OSGi
- Integrate with OSGi service model
  - Spring beans <==> OSGi services
- Testing
  - Enable testing without OSGi container
  - No dependencies on OSGi APIs
- Allow people to use all the Spring technology abstractions

10

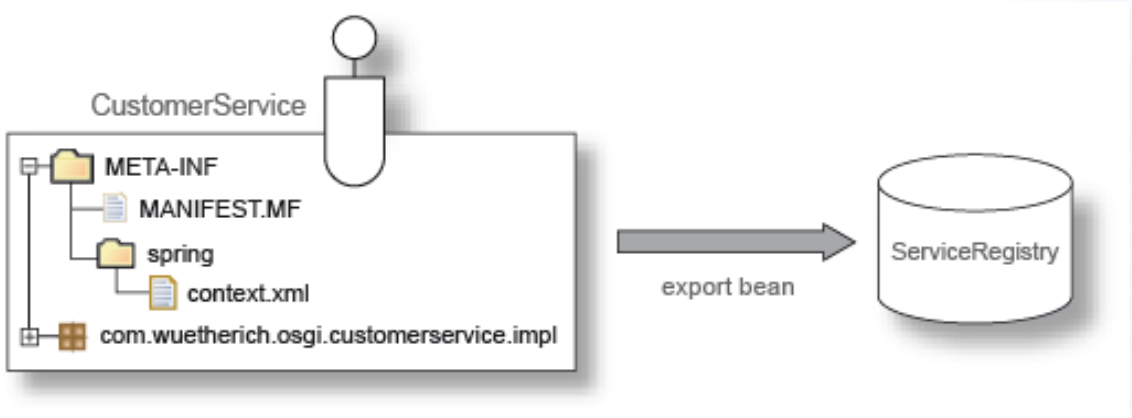
Bundles und Spring

- Application context per bundle:
  - Spring DM creates and destroys the application context when the bundle is started or stopped
  - META-INF/spring/*.xml or Spring-Context header in MANIFEST.MF



11

Beans as OSGi Services



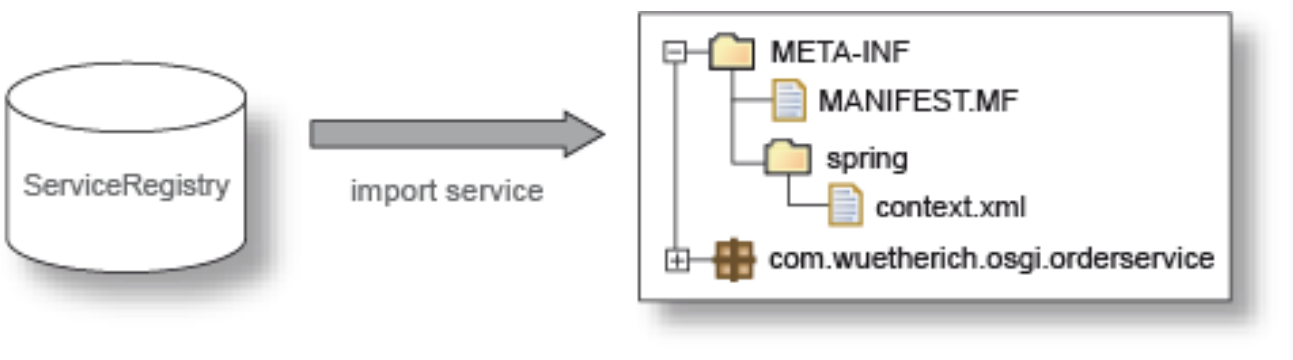
- Spring beans can be exported as OSGi Services
  - (no dependencies on OSGi Service Registry API)

```
<bean name="customerService"
    class="com.wuetherich.osgi.customerservice.impl.CustomerServiceImpl"/>

<osgi:service id="customerServiceOsgi"
        ref="customerService"
        interface="com.wuetherich.osgi.customerservice.CustomerService"/>
```


12

OSGi Services as Beans



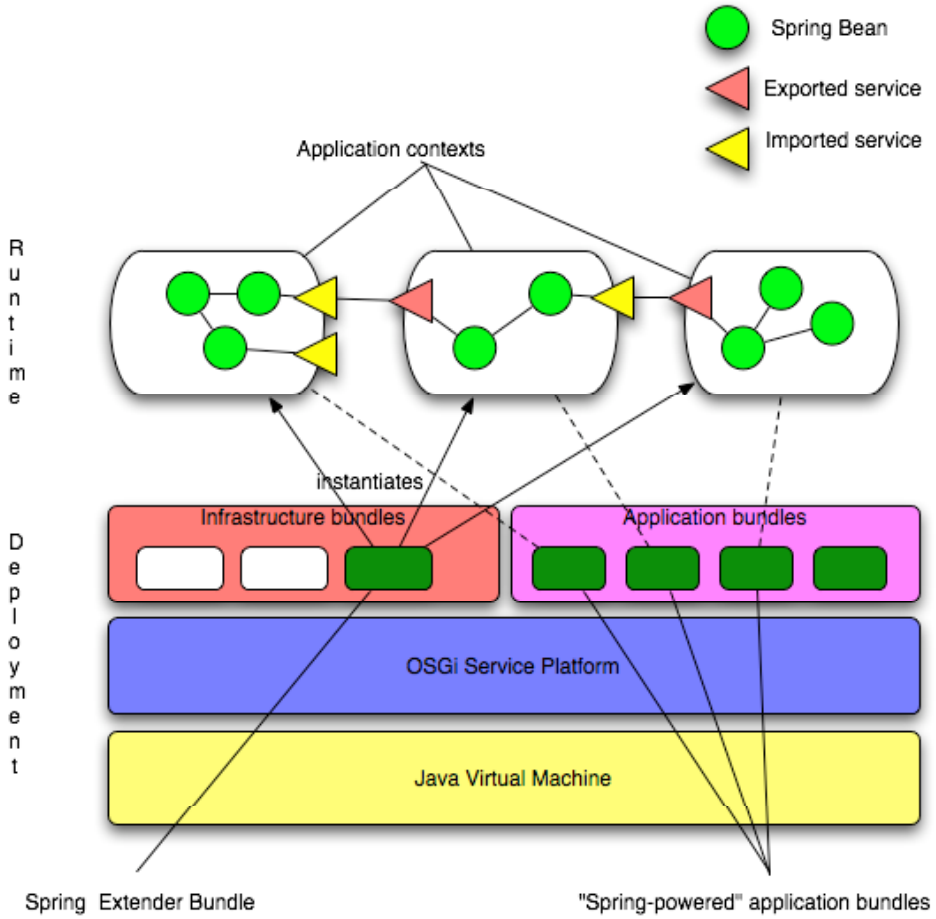
- OSGi Services can be imported and integrated as beans
  - (again no dependencies on OSGi Service Registry API)

```
<osgi:reference id="customerServiceOsgi"
      interface="com.wuetherich.osgi.customerservice.CustomerService"/>

<bean id="orderService"
   class="com.wuetherich.osgi.orderservice.internal.OrderServiceImpl">
  <property name="customerService">
    <ref local="customerServiceOsgi"/>
  </property>
</bean>
```

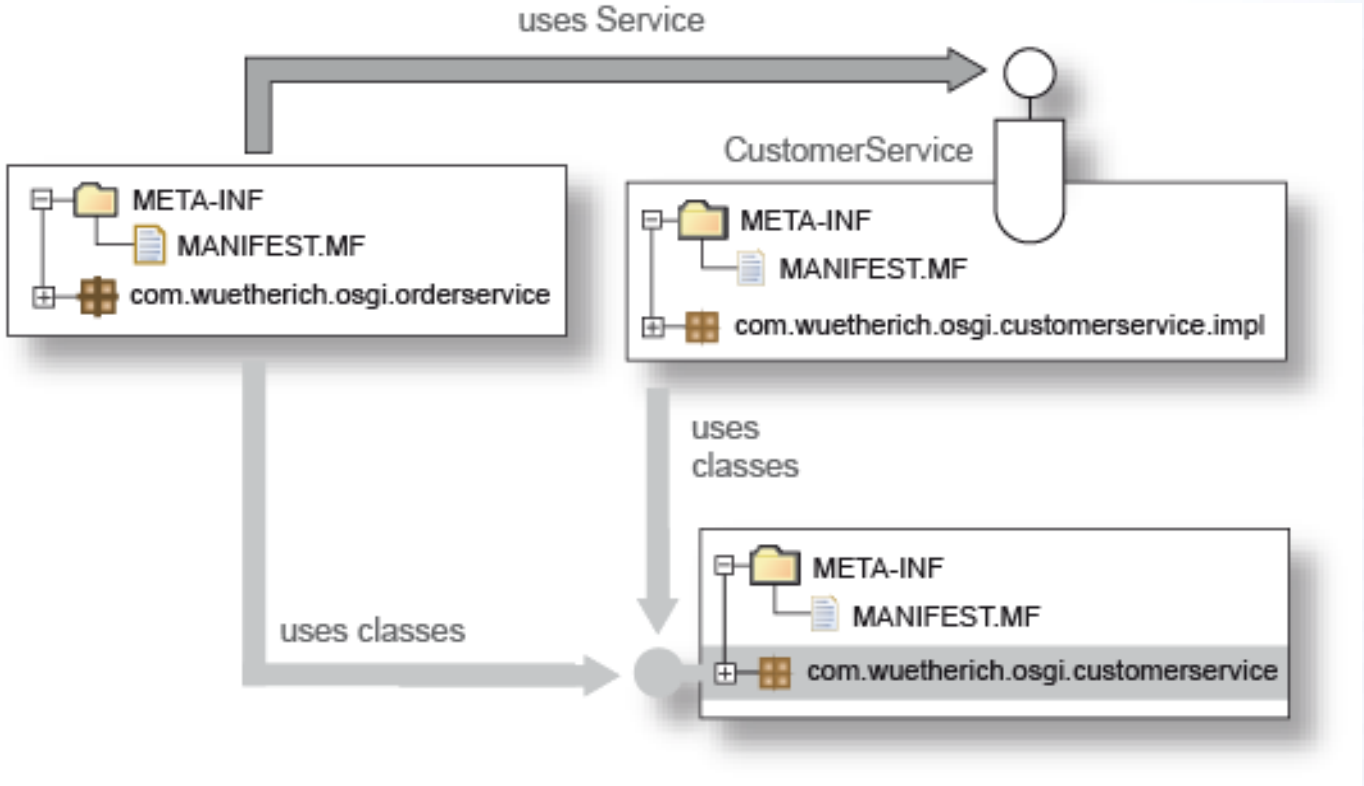

13

The Spring OSGi Modell



14

Spring DM by Example



15

Further Possibilities

- Cardinality
  - Relationship between imported OSGi Services and the representing bean (1..1, 0..1, 1..n, 0..n)
- Service Listener
  - Informs about service changes
- <osgi:property-placeholder>
  - Refer to properties from the ConfigAdmin service
- <osgi:bundle>
  - Refers to the OSGi bundle object
- <osgi:virtual-bundle>
  - Install a virtual OSGi bundle on the fly (testing)

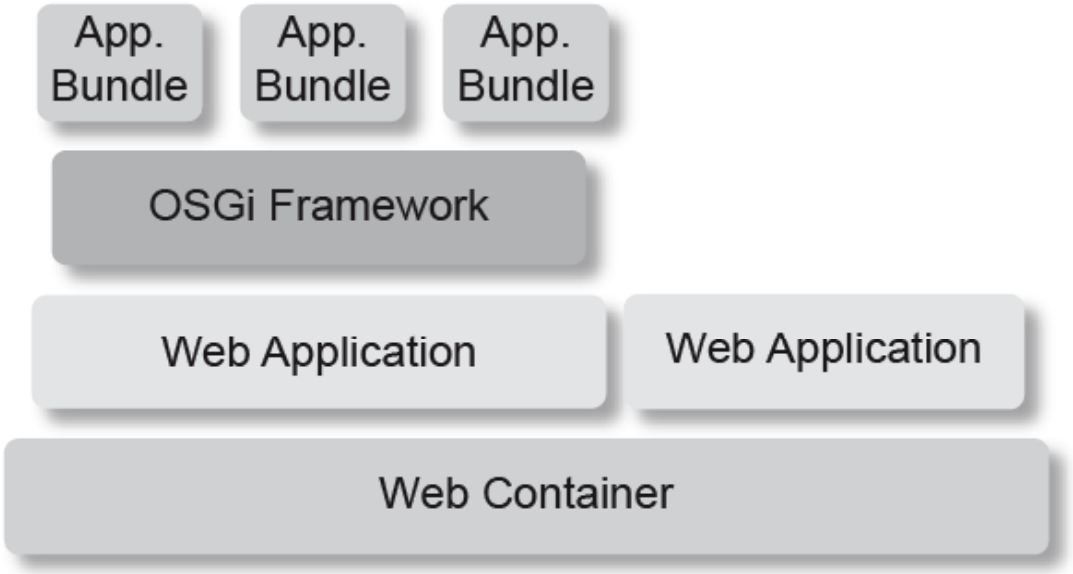
Results

- No OSGi API programming necessary
  - Spring POJO programming model
- Visibilities of spring beans controllable via service exports and imports
  - Dependency injection across bundles
- Dynamic service handling made easier

Spring-DM Web Support

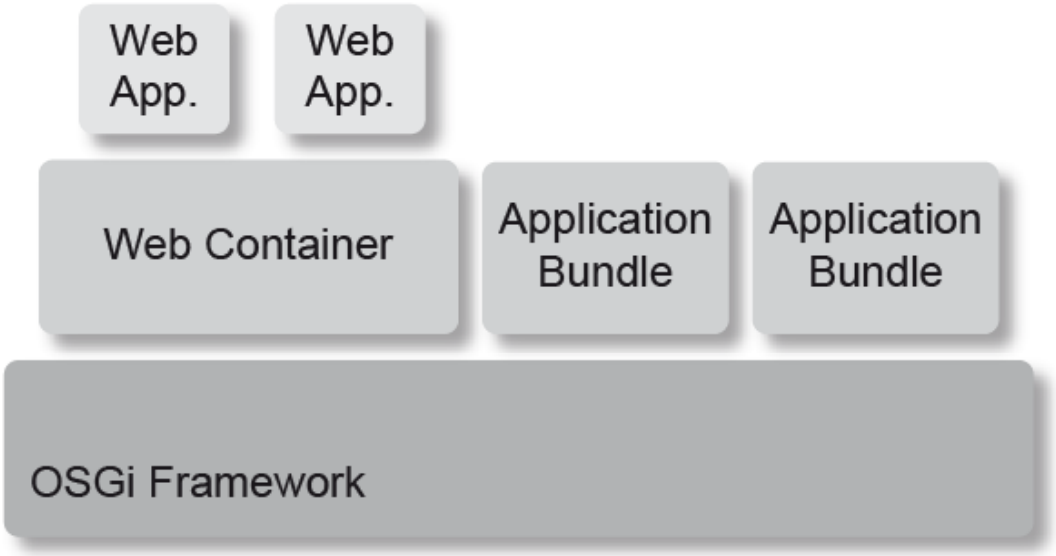
- Focus of Spring DM 1.1
  - Currently under development
  - Integration of existing spring web support with Spring DM
- Web applications with OSGi
  - Different deployment scenarios possible

OSGi-based web apps – part 1



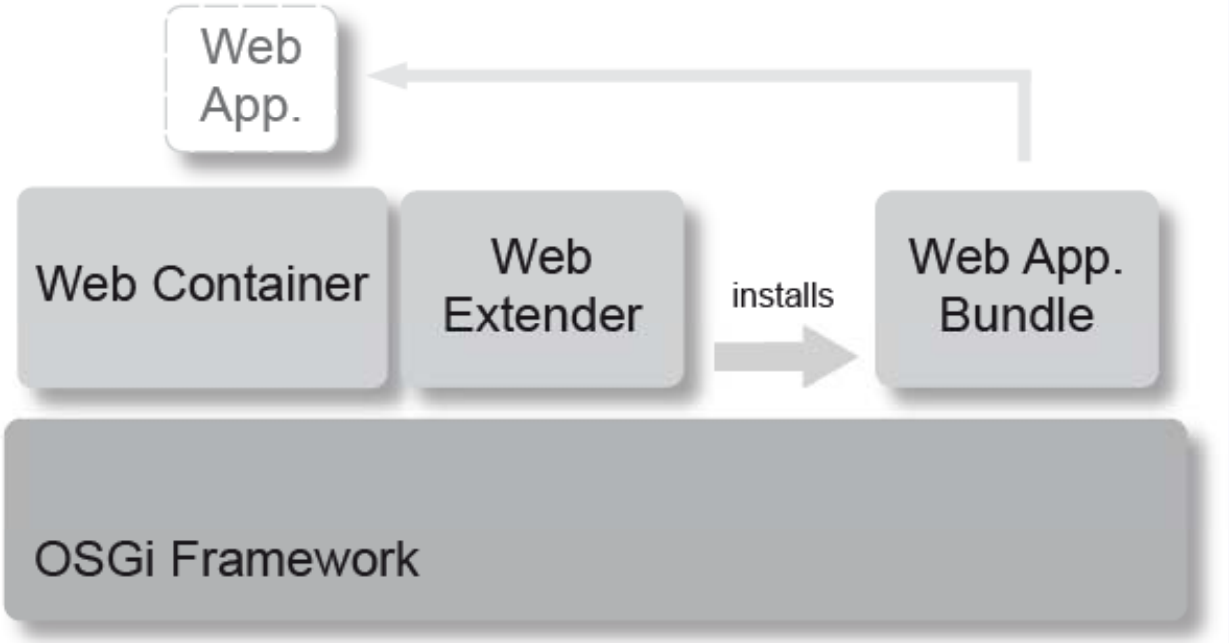
- **Scenario 1: OSGi embedded into web app**

OSGi-based web apps – part 2



- **Scenario 2: Web container embedded into OSGi**

The Spring DM way…



Web applications as Bundles

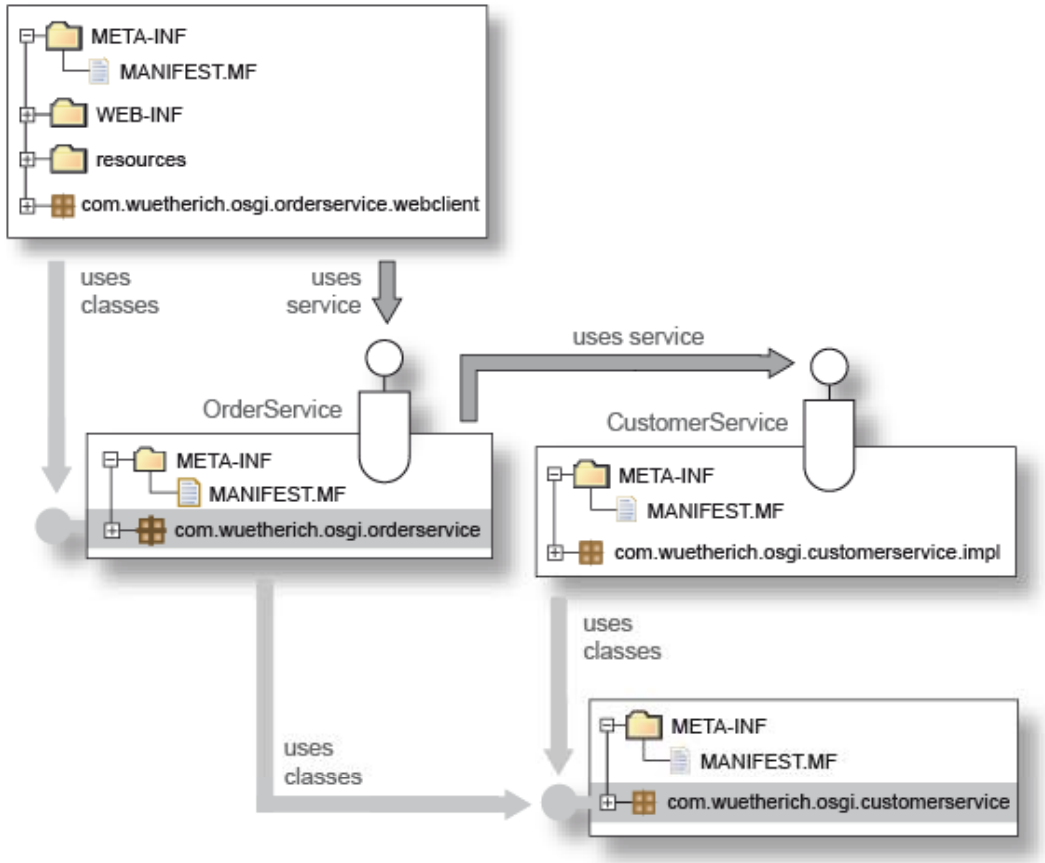
- „Regular“ WAR files
- Additional Bundle-Manifest

- web.xml shows how Spring DM is integrated

```
<context-param>
  <param-name>contextClass</param-name>
  <param-value>org.springframework.osgi.web.context.
    support.OsgiBundleXmlWebApplicationContext</param-value>
</context-param>

<listener>
   <listener-class>
     org.springframework.web.context.ContextLoaderListener
   </listener-class>
</listener>
```

Spring DM Web Support by Example



Even more Spring… ☺

- All the Spring features for clients as well
  - Eclipse-RCP, for example
- Integration with Equinox Extension Registry model possible
  - Dependency injection for extension objects, for example
  - http://www.martinlippert.org/

Outlook

- More information:
  - http://www.springframework.org/osgi
  - http://groups.google.com/group/spring-osgi
  - http://www.springframework.org/osgi/specification
- Spring Dynamic Modules tutorial:
  - http://www.eclipsecon.org/2008/?page=sub/&id=495

Die OSGi Service Platform –
Eine Einführung mit Eclipse Equinox

- Detailed introduction into OSGi
- April 2008, dpunkt.verlag
- ISBN 978-3-89864-457-0
- Already available!!!

- (but German only at the moment…)

Thank you!!!

- … for your attention!!!
- **Q&A**

- martin.lippert@akquinet.de
- gerd@gerd-wuetherich.de

27