

時間	議程
13:00 - 13:30	報到
13:30- 14:00	<p>台灣昇陽 Java 認證及訓練市場趨勢及 Java 證照在台灣戰場上的優勢</p> <p>岳汝誠 昇陽電腦教育訓練服務部 台灣區總經理</p>
14:00 - 14:45	<p>深入淺出 Web 容器 – Tomcat 原始碼分析</p> <p>林信良 昇陽電腦教育訓練服務部 資深技術顧問</p>
14:45 - 15:00	Coffee Break
15:00 - 15:45	<p>jQuery 與 ExtJS 之使用與比較</p> <p>黃登揚 中央大學資訊工程系準博士候選人</p>
15:45 - 16:30	<p>輕鬆開發輕量級、高效能的Java EE應用程式 – 使用 SpringSource dm Server</p> <p>廖峻鋒 昇陽電腦教育訓練服務部 技術顧問</p>
	歡樂抽獎 – iPhone 3G幸運得主就是你！



2009 Java認證日

SpringSource dm Server

輕鬆開發輕量級、模組化的Java EE應用程式



廖峻鋒 (try)

講師/技術顧問

Sun 教育訓練服務部

cfliao@ieee.org

SpringSource dm Server

Spring Dynamic Module

Server



今天講些什麼？

Spring與OSGi

Spring+OSGi=?

Spring DM架構與原理

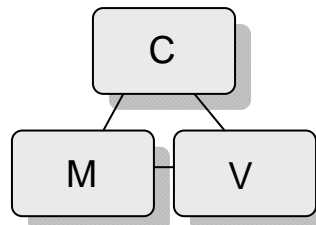
OSGi在Java EE的定位

SpringSource dm Server

早期J2EE 應用程式系統架構

2009 Java 認證日
2009 Java Certification Day

展現層



JSP/ Servlet / JSTL

商務邏輯層



SessionBeans

資料存取層



EntityBeans

Spring Framework 發展背景

2009 Java 認證日
2009 Java Certification Day

- 自2002年起，許多Java企業領域的開放源碼產品在實務上有出色表現：
 - 展現層 - Struts Framework已經幾乎成為標準。
 - 永續層 - Hibernate受到注意，並持續加強功能。
 - 商務層 - 除EJB規格外，並沒有其它的開放原始碼的元件模型被提出。

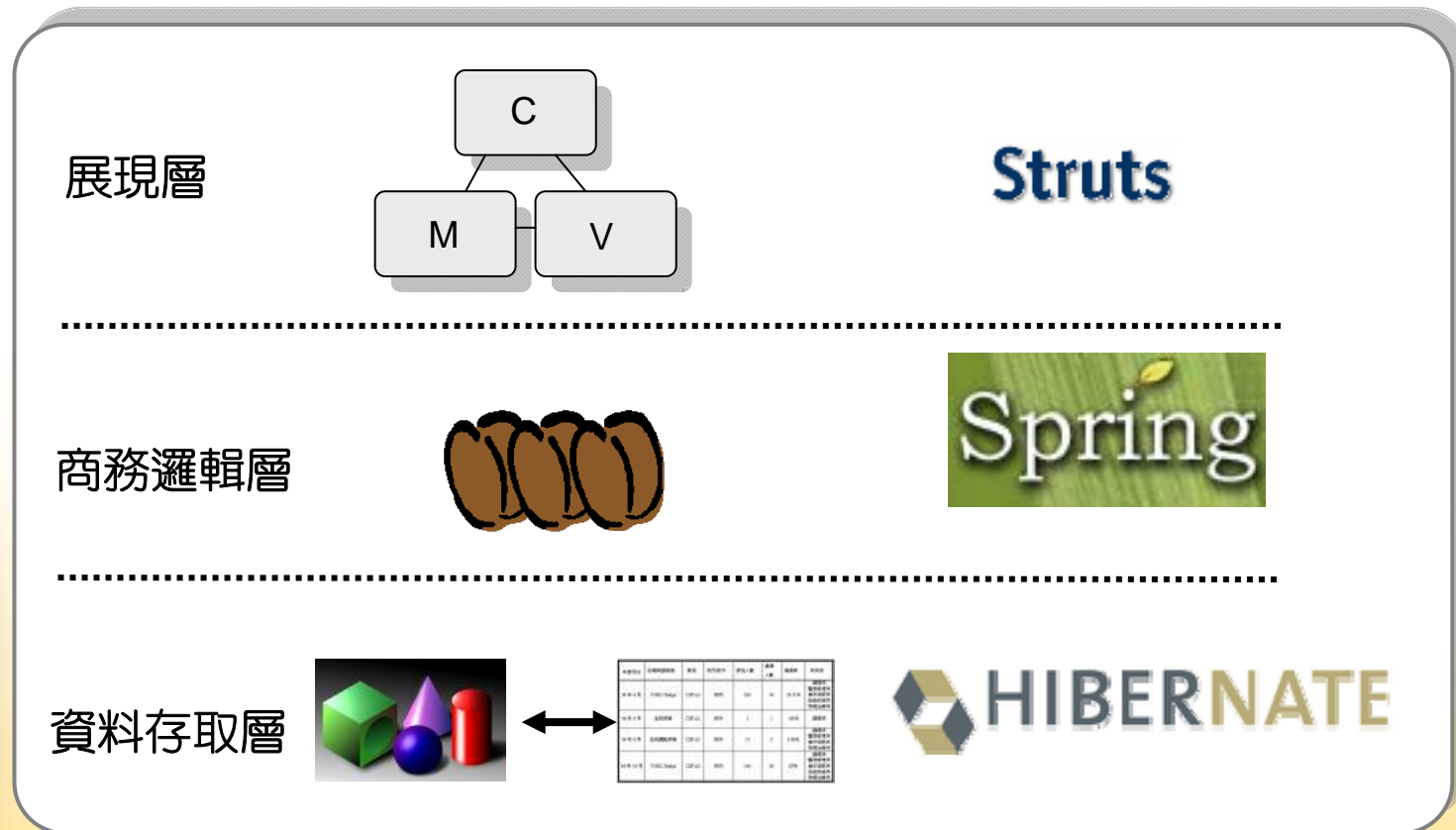
The logo for Struts, featuring the word "Struts" in a bold, blue, sans-serif font with a slight drop shadow.The logo for Hibernate, featuring a stylized green and yellow cube icon to the left of the word "HIBERNATE" in a bold, blue, sans-serif font.The logo for Spring, featuring the word "Spring" in a white, serif font with a small green leaf icon above the letter 'i', all set against a green background with a subtle leaf pattern.

Spring Framework在此時的出現恰銜
接了這一塊空隙

The Sun Microsystems logo, featuring a stylized blue and white sun icon to the left of the word "Sun" in a blue, serif font, with "microsystems" in a smaller, blue, sans-serif font below it.

SSH架構

2009 Java 認證日
2009 Java Certification Day



Dependency

依賴性(Dependency)

2009 Java 認證日
2009 Java Certification Day

- 當某類別繼承另一類別、實作另一介面或者叫用到其它類別，我們就說此類別依賴 (depends-on) 另一個類別。

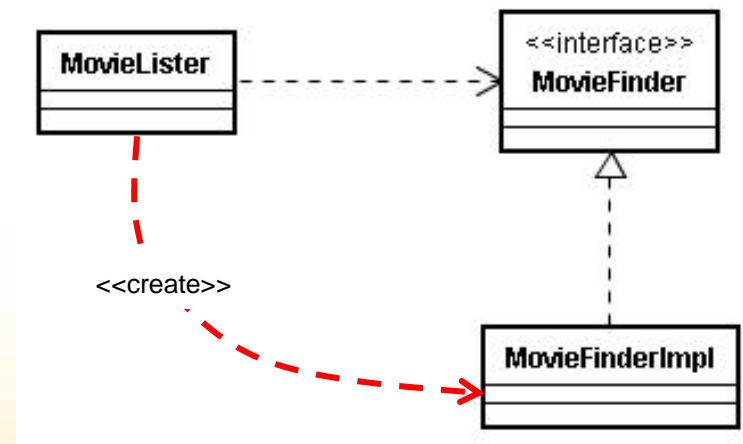
```
public class MySessionBean implements javax.ejb.SessionBean {  
    private javax.ejb.SessionContext ctx;  
  
    public void ejbCreate() {}  
    public void ejbRemove() {}  
    public void ejbActivate() {}  
    public void ejbPassivate() {}  
    public void setSessionContext(SessionContext ctx) {  
        this.ctx = ctx;  
    }  
    ...(略)...  
}
```



利用介面降低依賴性

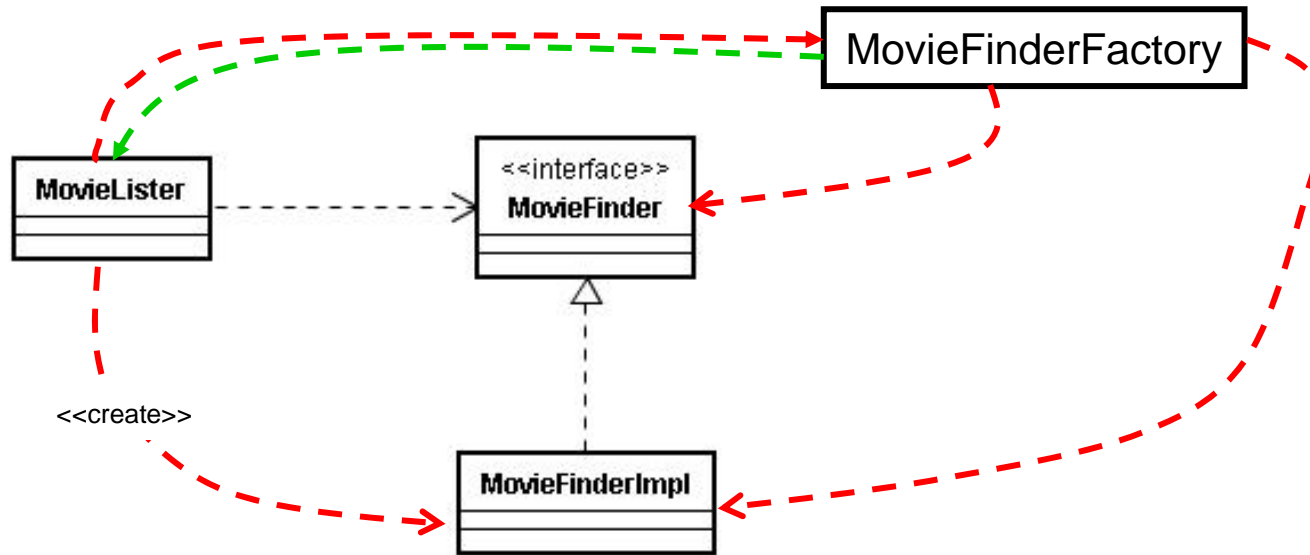
- 將Client與實作類別中間用介面隔開

```
public MovieLister () {  
    movieFinder = new MovieFinderImpl();  
}
```

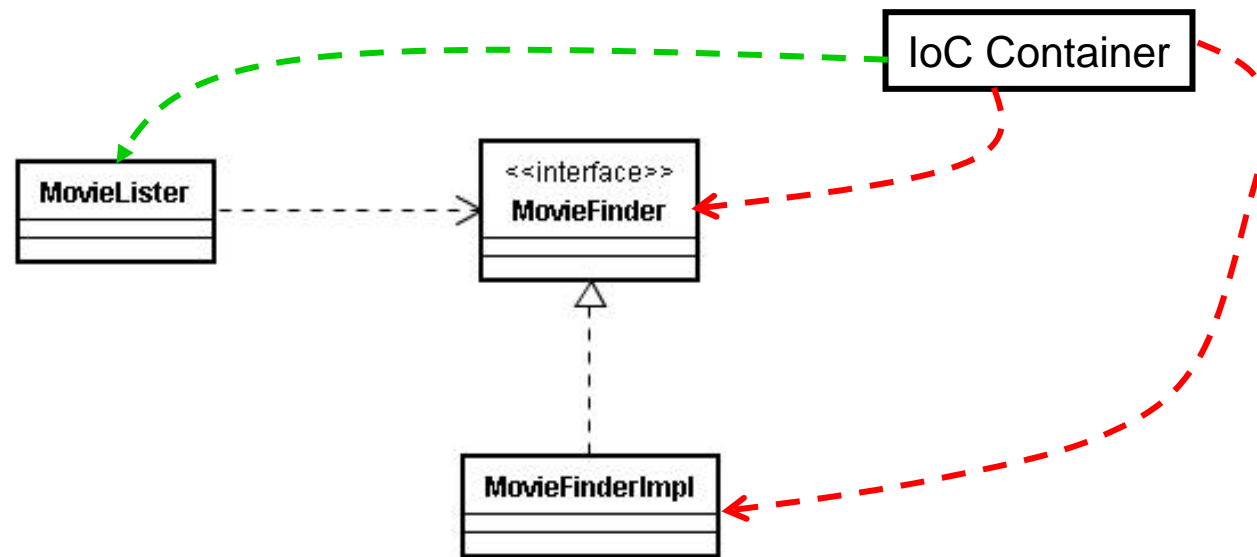


物件生成控制權的倒置

```
public MovieFinder create() {  
    ...  
    return new MovieFinderImpl();  
}
```

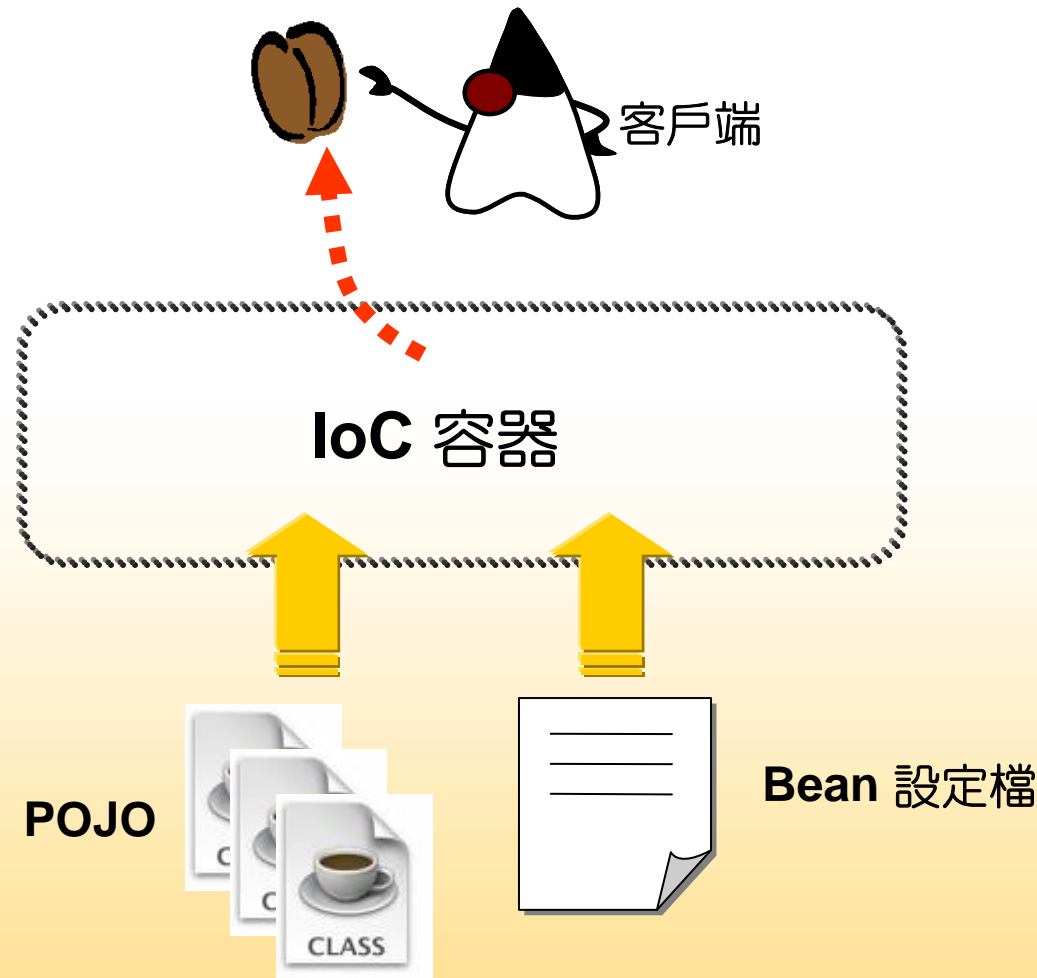


Dependency Injection



Spring DI運作模式

2009 Java 認證日
2009 Java Certification Day



Setter Injection (Spring 2.0)

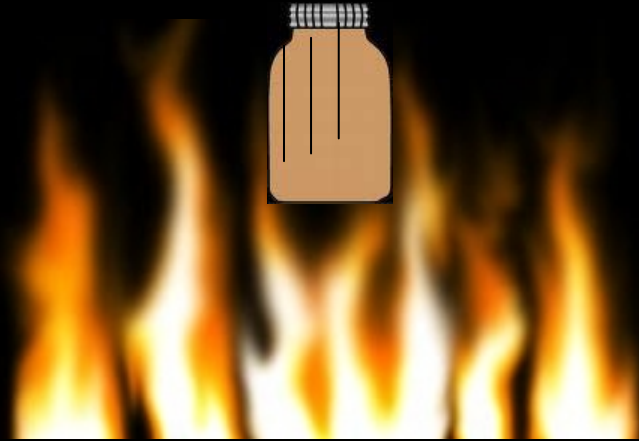
2009 Java 認證日
2009 Java Certification Day

```
class ProductViewer {  
    private Product product;  
    ...  
    public void setProduct(Product product) {  
        this.product = product;  
    }  
}
```

```
<bean id="productViewer" class="ProductViewer">  
    <property name="product" ref="myProduct" />  
</bean>
```

```
<bean id="myProduct" class="Product">  
    <property name="productId" value="10001"/>  
</bean>
```





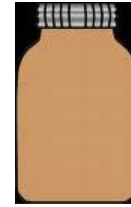
Jar Hell



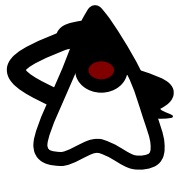
a.jar



b.jar



c.jar



A

A

B

A

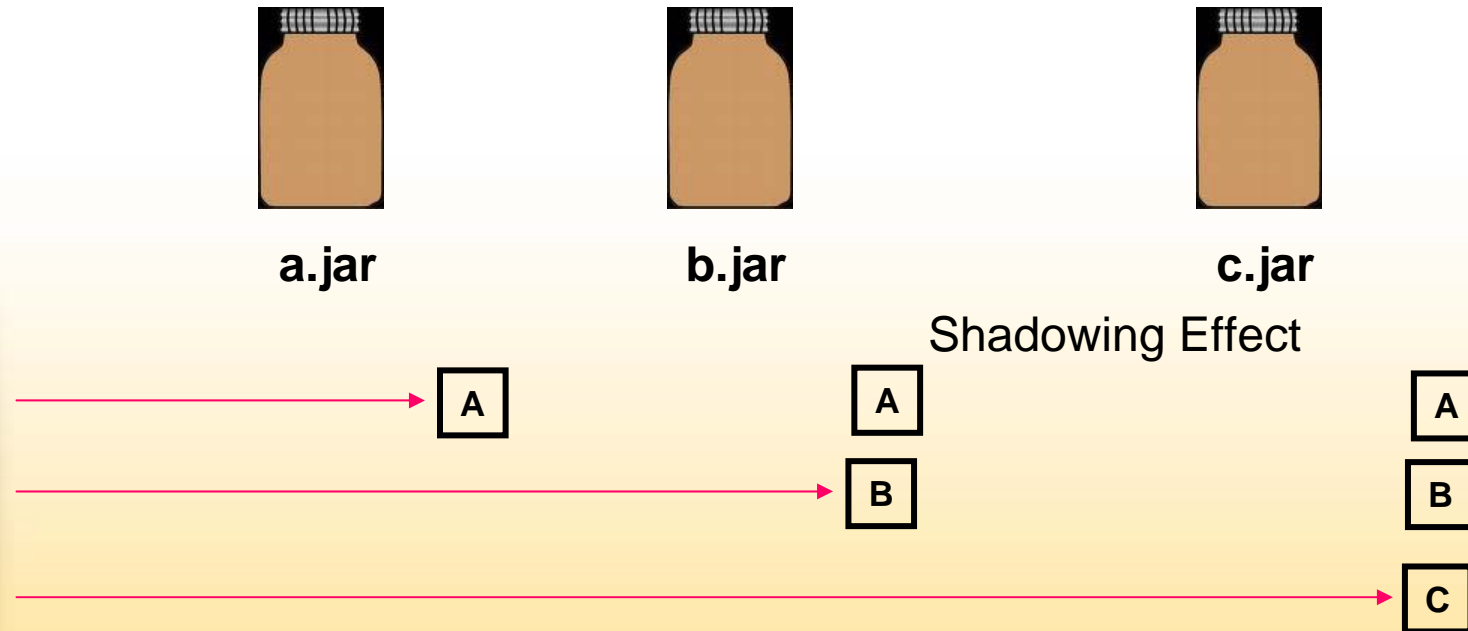
B

C

java -classpath a.jar;b.jar;c.jar

Jar = 模組 !?

- Jar在Run-time沒有意義!

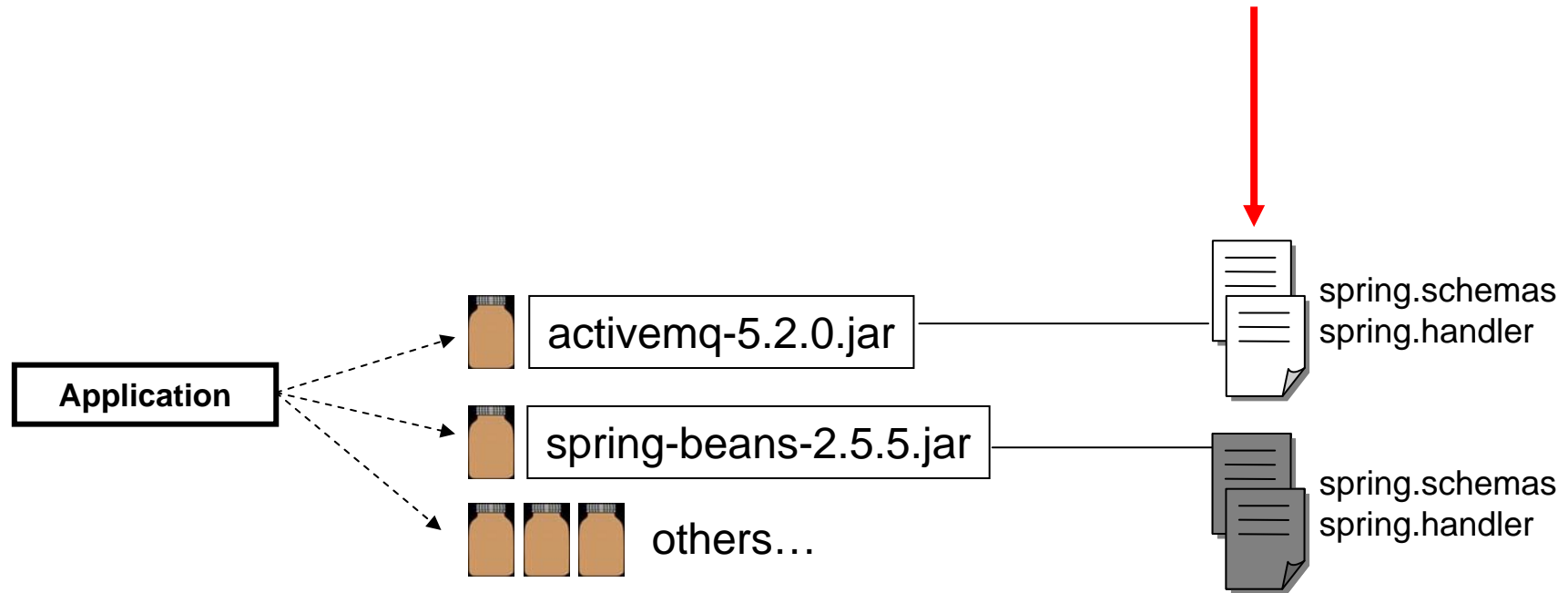


`java -classpath a.jar;b.jar;c.jar`



Spring應用程式 無法離線執行？

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



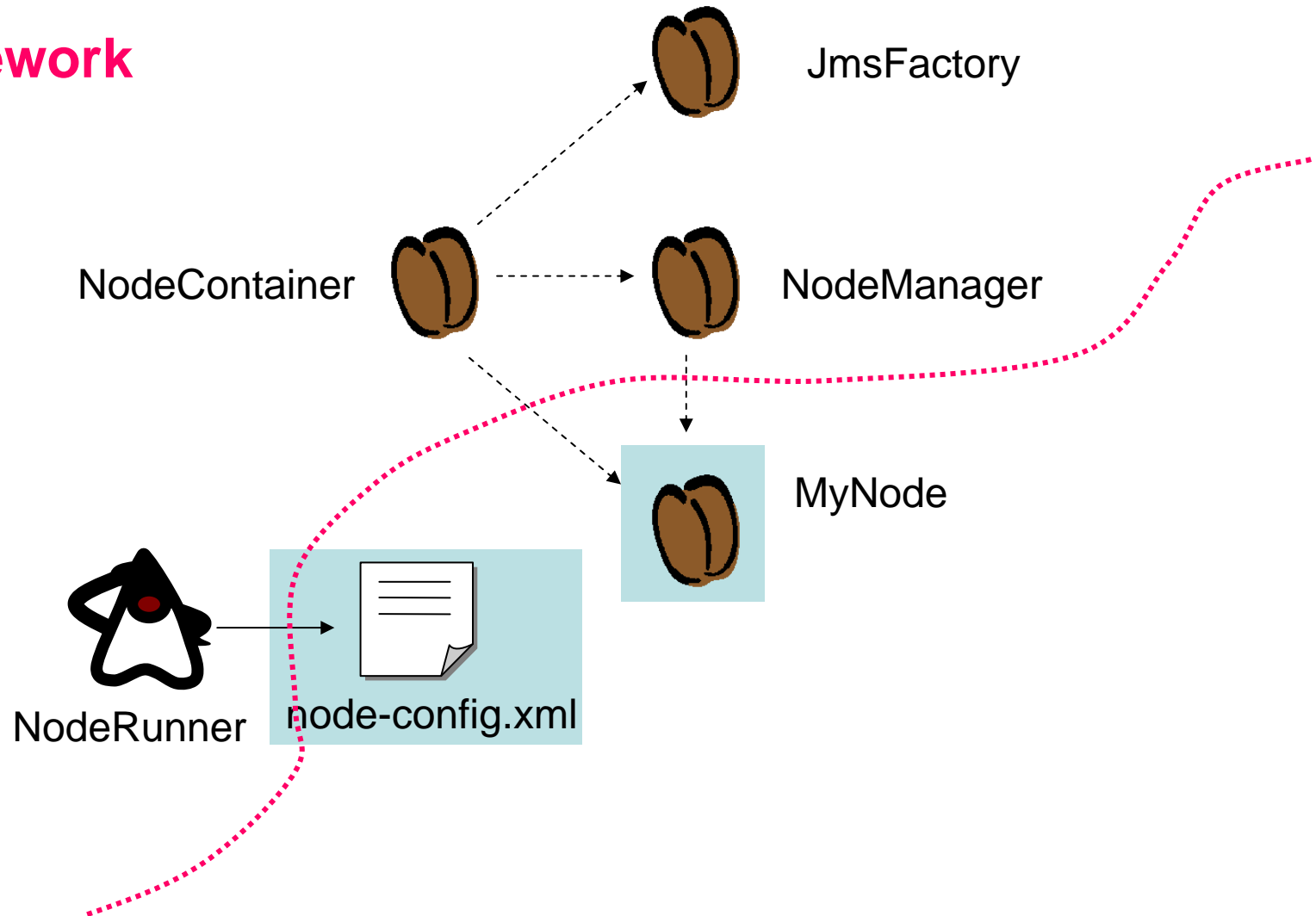
<http://activemq.org/config/1.0=activemq.xsd>
<http://activemq.org/config/1.0/1.0.xsd=activemq.xsd>
<http://activemq.apache.org/schema/core=activemq.xsd>
<http://activemq.apache.org/schema/core/activemq-core.xsd=activemq.xsd>

<http://www.springframework.org/schema/beans/spring-beans-2.0.xsd=org/springframework/beans/factory/xml/spring-beans-2.0.xsd>
<http://www.springframework.org/schema/beans/spring-beans-2.5.xsd=org/springframework/beans/factory/xml/spring-beans-2.5.xsd>

Runtime

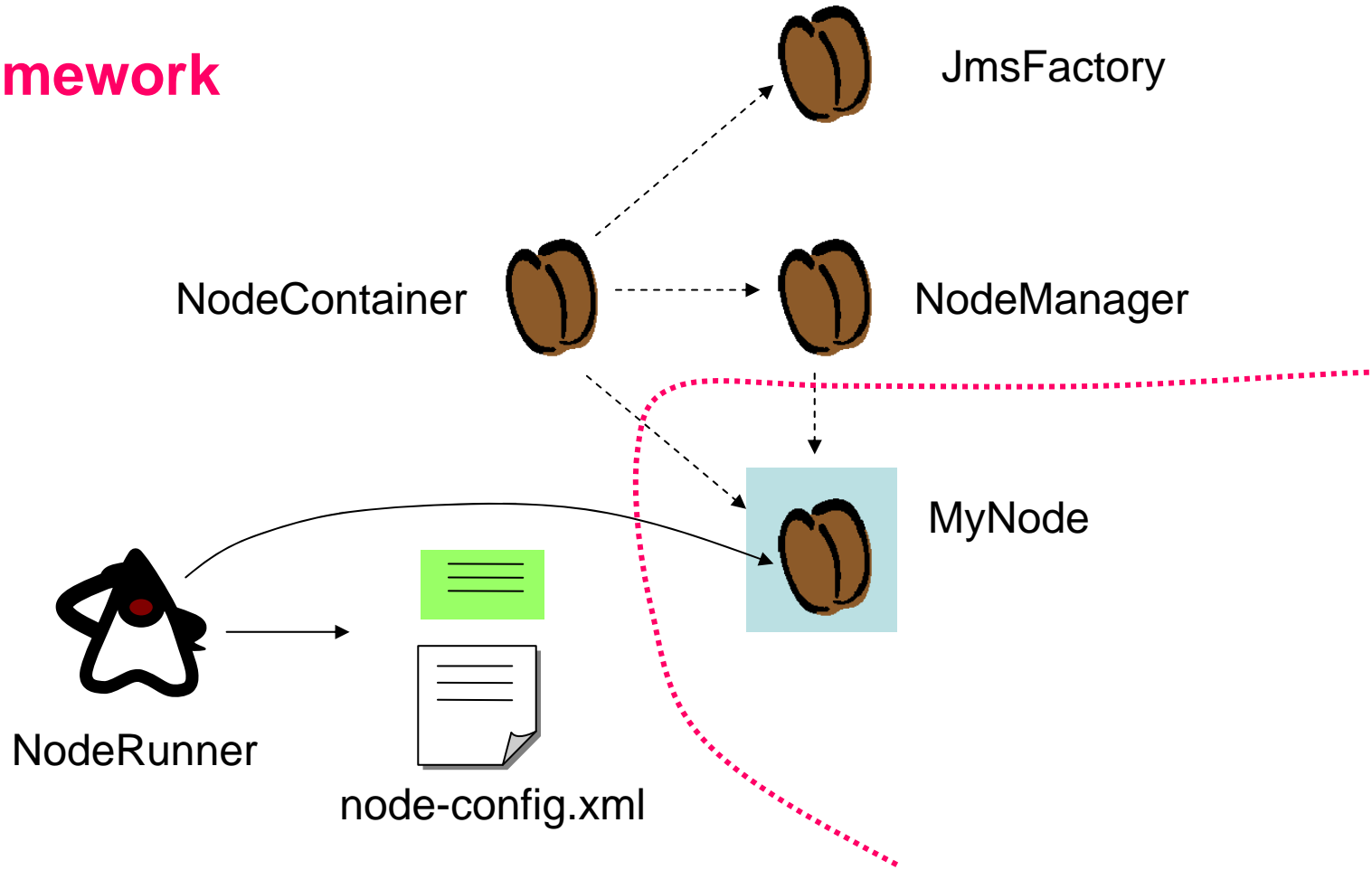
修改Bean定義

Framework



如何在Runtime加入Bean定義？

Framework



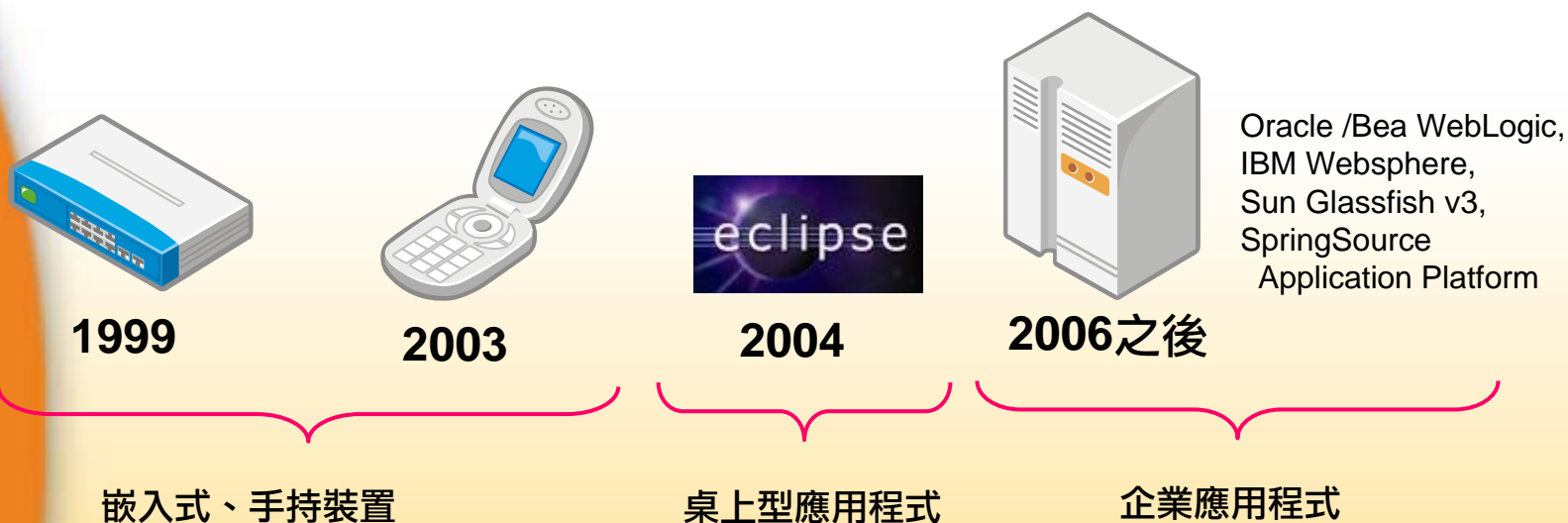
GenericApplicationContext
Autowiring

OSGi



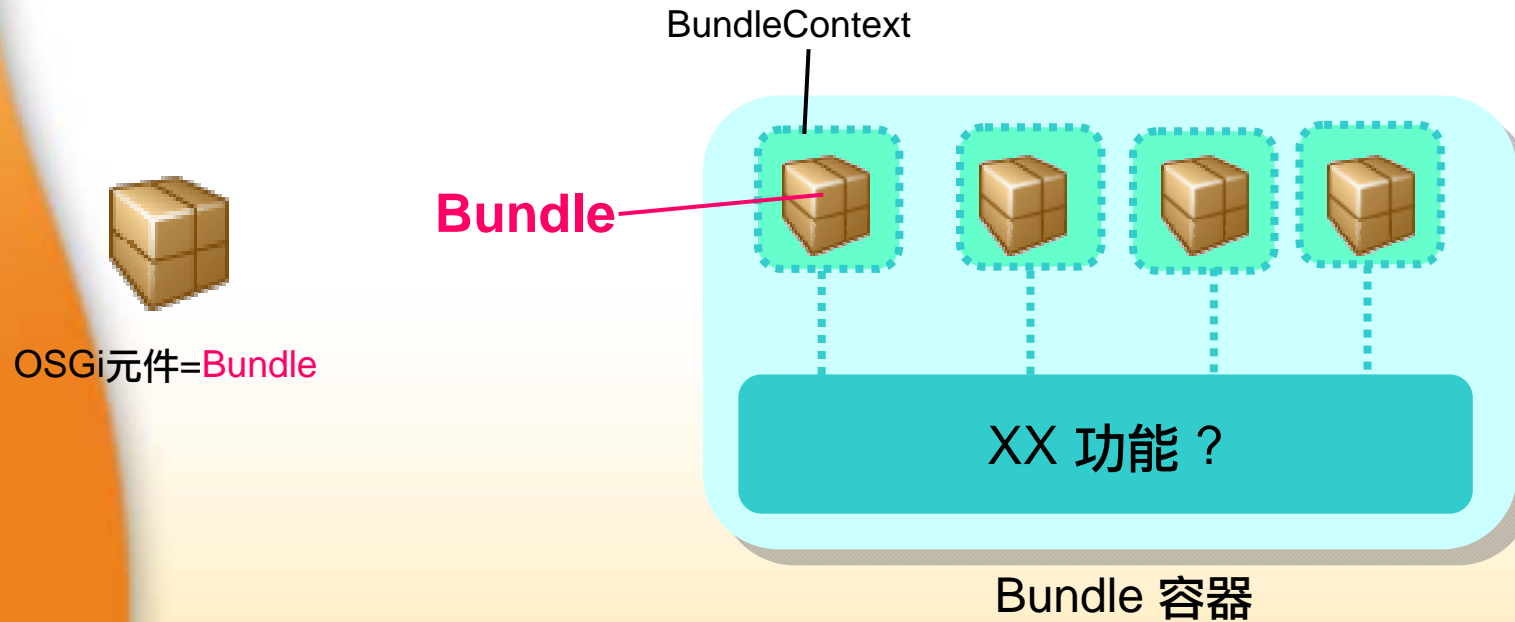
2009 Java 認證日
2009 Java Certification Day

- 由IBM、NEC、Mitsubishi、Motorola、NTT、Nokia、Oracle、Samsung等數十家廠商聯合制定並推動

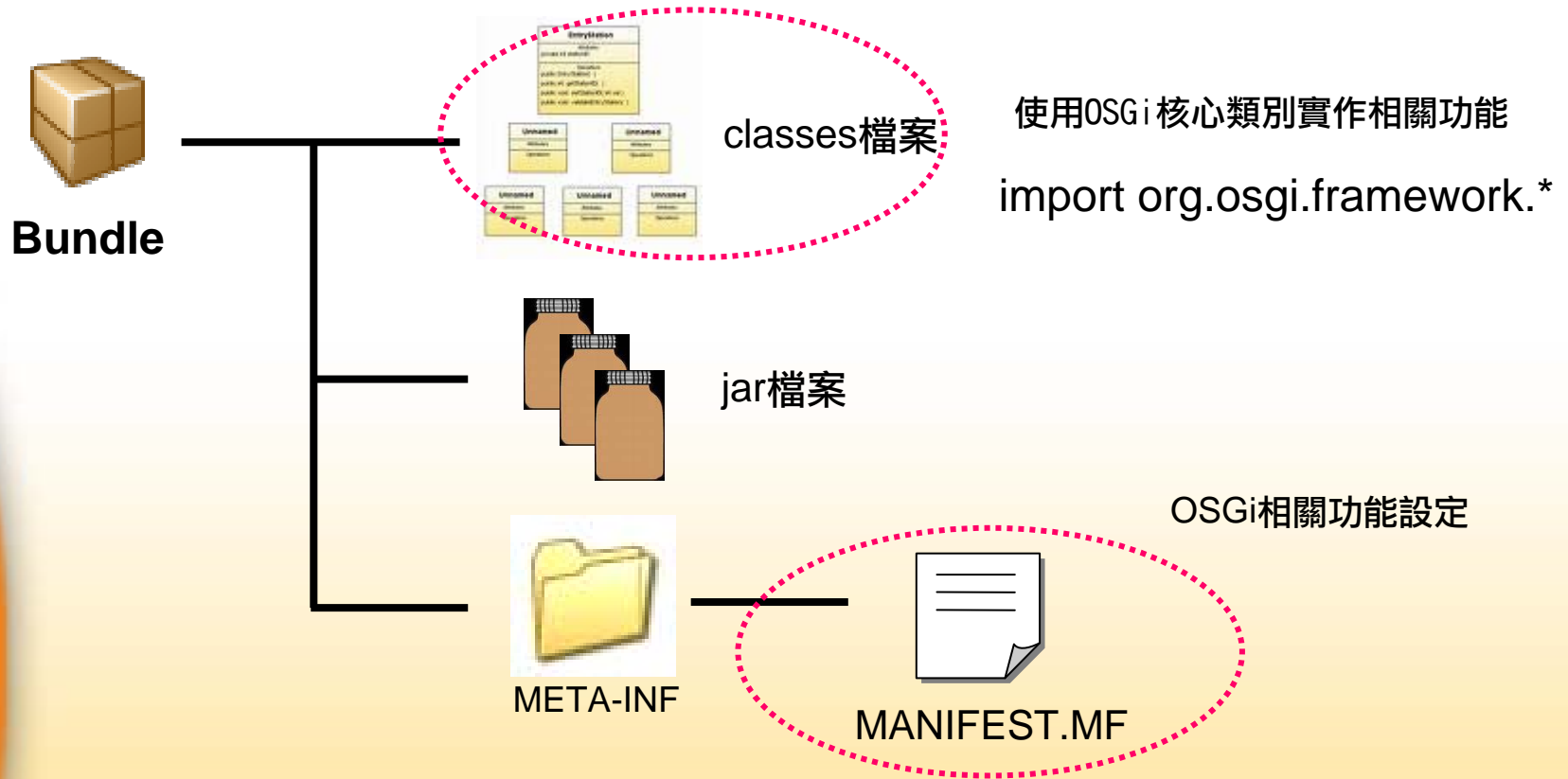


OSGi元件模型

2009 Java 認證日
2009 Java Certification Day



Bundle結構



Demo: OSGi QuickStart

2009 Java 認證日
2009 Java Certification Day

- 啟動與設定OSGi
 - 準備好org.eclipse.osgi_...(依版本不同而不同)...jar
 - java -jar equinox.jar -console
 - 設定config.ini:eclipse.ignoreApp=true
 - 觀察configuration/config.ini
- 基本操作
 - install/uninstall: 安裝/反安裝bundles
 - ss: 列出bundles
 - bundle xx: 詳述bundle資訊



OSGi服務的啟始點

```
import org.osgi.framework.*;
```

```
public class Activator implements BundleActivator
```

```
{
```

```
    public void start(BundleContext context)
```

```
    {
```

```
        System.out.println("Hello Java Certification Days 09 高雄場!");
```

```
    }
```

```
    public void stop(BundleContext context)
```

```
    {
```

```
        System.out.println("Bye!");
```

```
    }
```

```
}
```

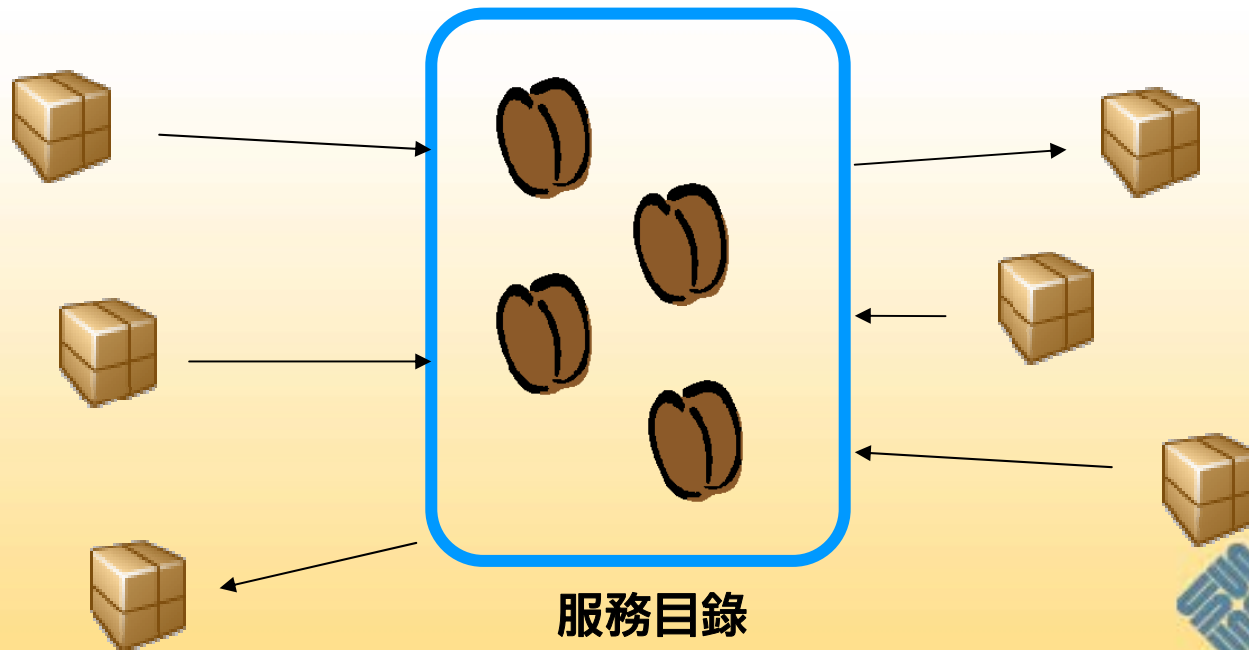
BundleActivator介面提供
bundle啟動與停止時的call
back

← 程式進入點

從BundleContext可取得OSGi相關服務

OSGi服務

- Bundles將可提供服務的POJO註冊到ServiceRegistry中
- Bundle可以發佈(publish)、發現(discovery)並呼叫其它服務。



服務註冊

2009 Java 認證日
2009 Java Certification Day

```
bundleContext.registerService("HelloService", 服務(介面)名稱  
instance, 類別實體  
props); 相關屬性
```

HelloService
Inag=en_US

服務目錄

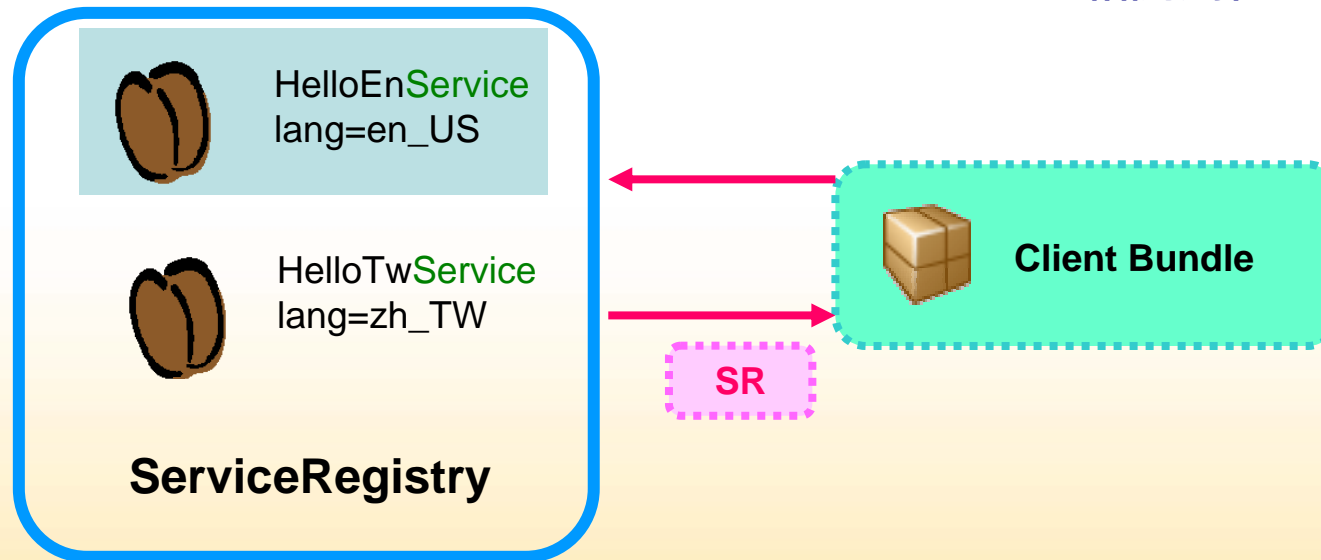


HelloService Bundle

服務查詢

```
ServiceReference sr = bundleContext.getServiceReference("HelloService", 服務名稱  
"(lang=en_US)");
```

相關屬性



取得與使用服務

2009 Java 認證日
2009 Java Certification Day

```
(HelloService) svc = sr.getService();  
System.out.println(svc.hello());
```



Demo: Hello Bundle~

2009 Java 認證日
2009 Java Certification Day

- 使用SpringSource Tools Suite
 - 設定SpringSource dm Server
 - 新增bundle專案
 - 開發專案
 - 將專案加入dm Server
- HelloService
 - Service Interface
 - 中文版 HelloTwService
 - 英文版 HelloEnService
 - Activator (用來註冊服務)
- HelloClient
 - Activator (用來找尋並呼叫服務)





Dependency Injection

Aspect Orientation



Dynamic Module

Service Orientation

Spring + OSGi ?

- OSGi可以為Spring帶來的好處
 - 模組化、Class versioning (jar hell)
 - 動態服務 (可以動態調整dependency)
 - Shared library: Reduce memory footprint
- Spring可以為 OSGi帶來的好處
 - 透過DI簡化Bundle開發流程
 - 透過Spring可更容易開發Java EE程式

Spring + OSGi = Spring DM

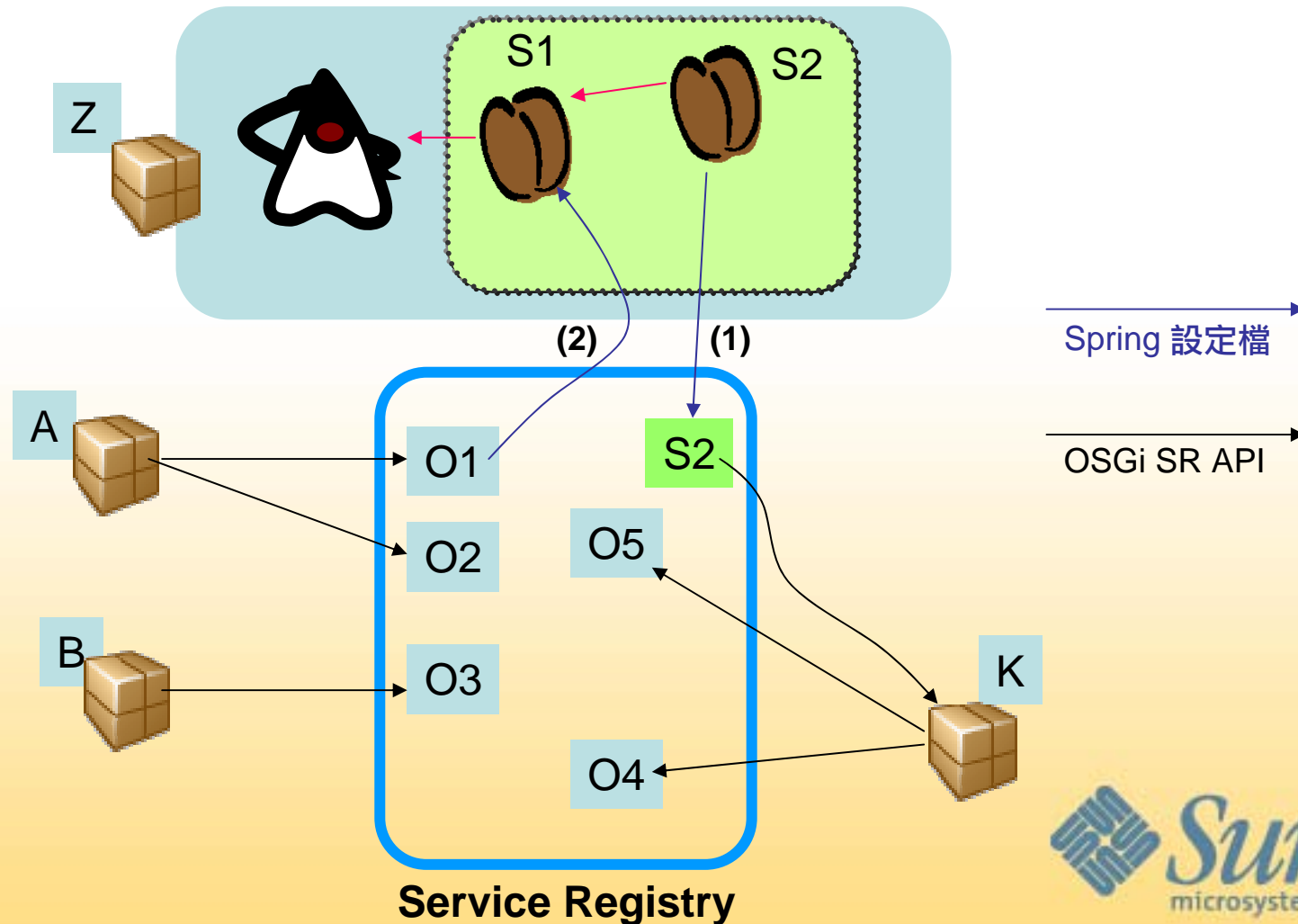
2009 Java 認證日
2009 Java Certification Day

- 研發動機
 - 整合Spring與OSGi的設計理念，發展出兼具二者優點的全新Java應用程式模型
- 發展歷史
 - 2006 開始投入OSGi-Spring整合研究
 - 2008 推出革命性的全模組化應用程式伺服器
SpringSource dm Server



Spring – OSGi整合

- 核心概念: Spring Beans與OSGi Services都是POJO，所以可以將二者透通整合



雙向整合設定範例(1)

將Spring Beans發佈為OSGi Services

```
<bean id="s2Impl" class="com.mycom.impl.s2Impl" />
```

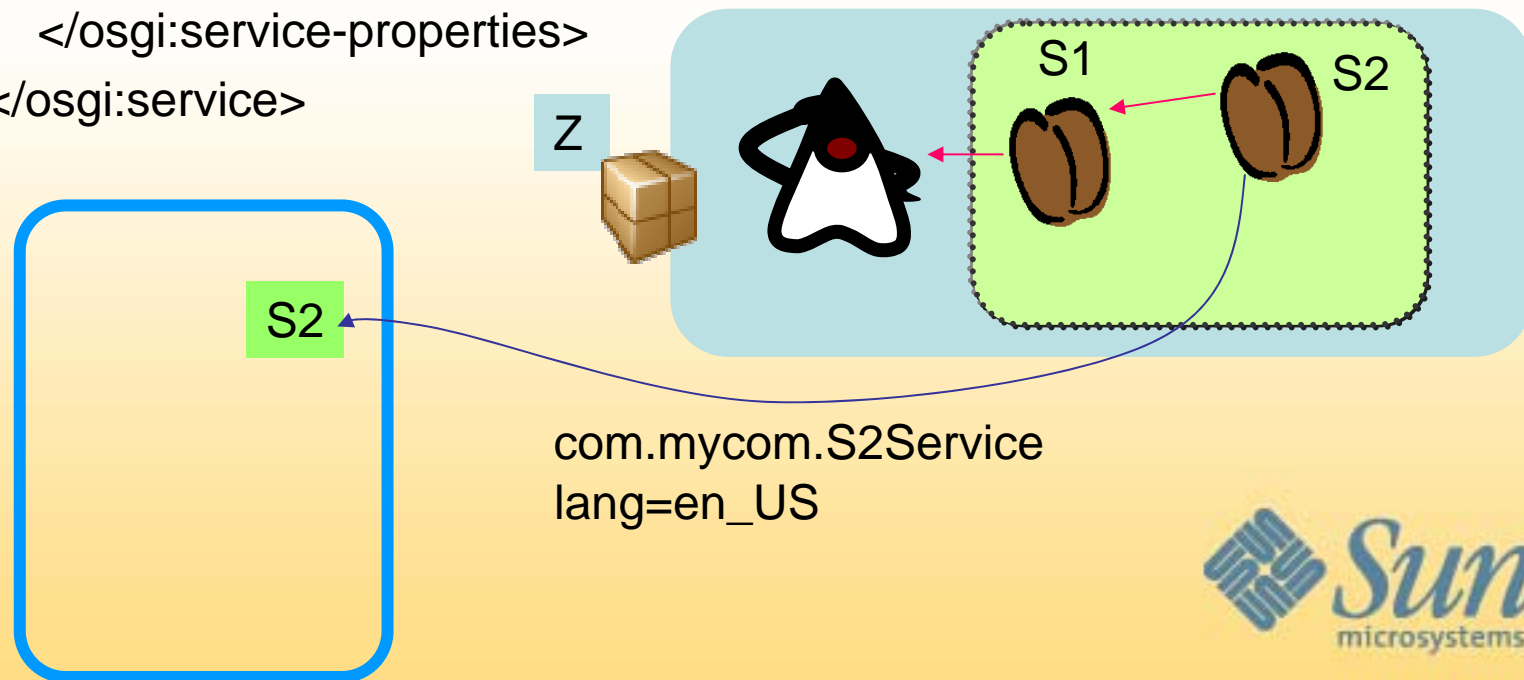
```
<osgi:service interface="com.mycom.S2Service" ref="s2Impl">
```

```
<osgi:service-properties>
```

```
<entry key="locale" value="en_US" />
```

```
</osgi:service-properties>
```

```
</osgi:service>
```

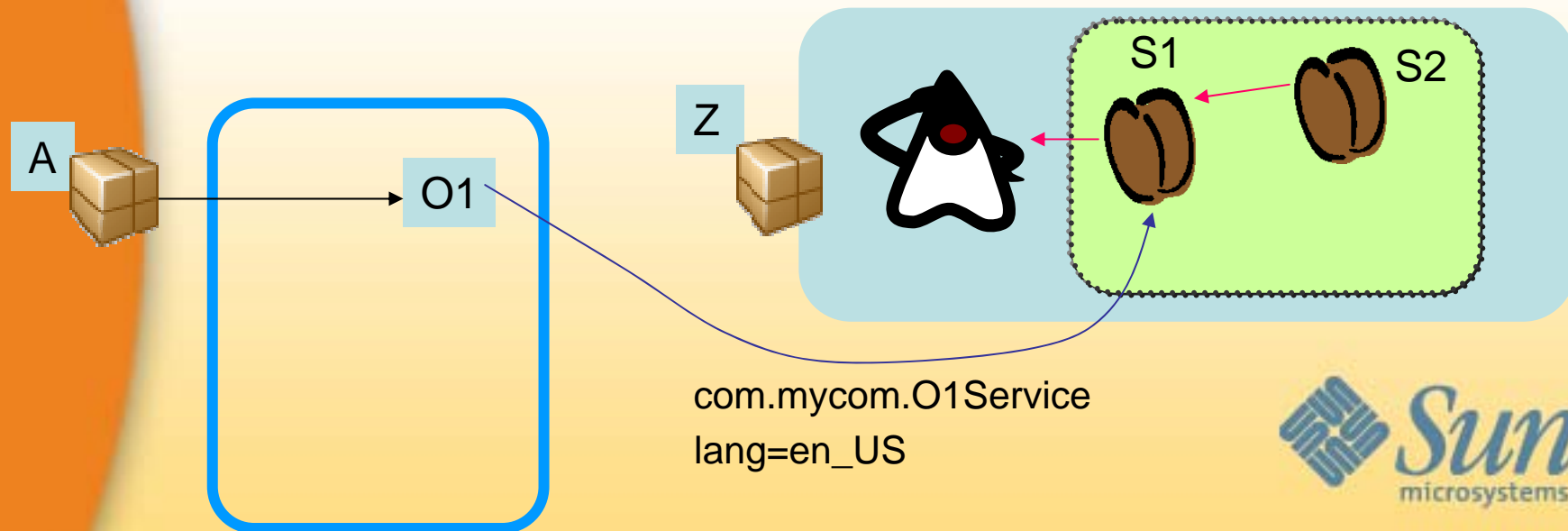


雙向整合設定範例(2)

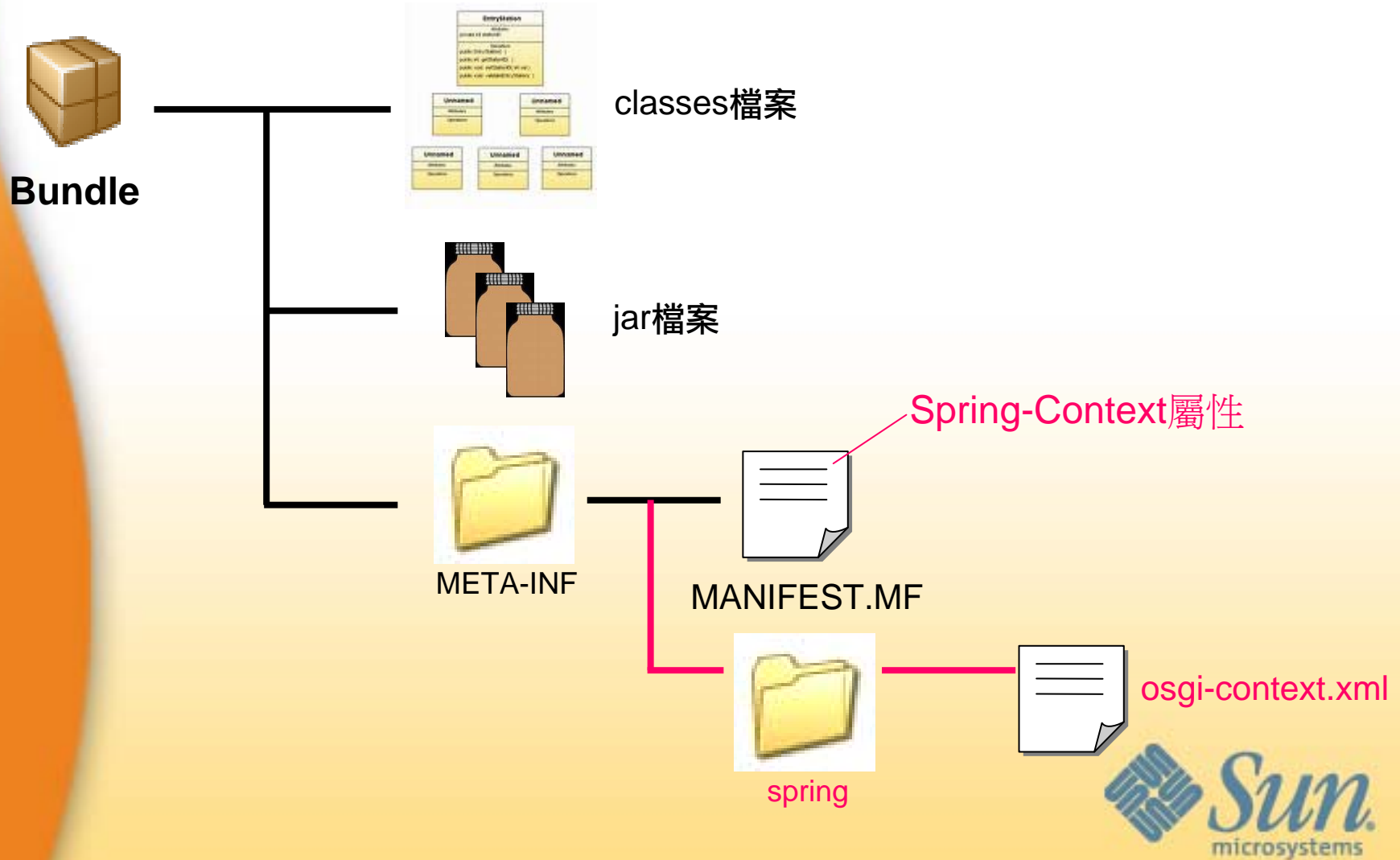
將OSGi Services 注入Spring Beans中

```
<bean id="s1" class="com.mycom.bye.impl.S1ServiceImpl">  
  <property name="o1" ref="o1" />  
</bean>
```

```
<osgi:reference id="o1" interface="com.mycom.O1Service"  
  filter="(lang=en_US)" />
```



如何在Bundle中配置 ApplicationContext



Demo: Hello Spring Bundle!

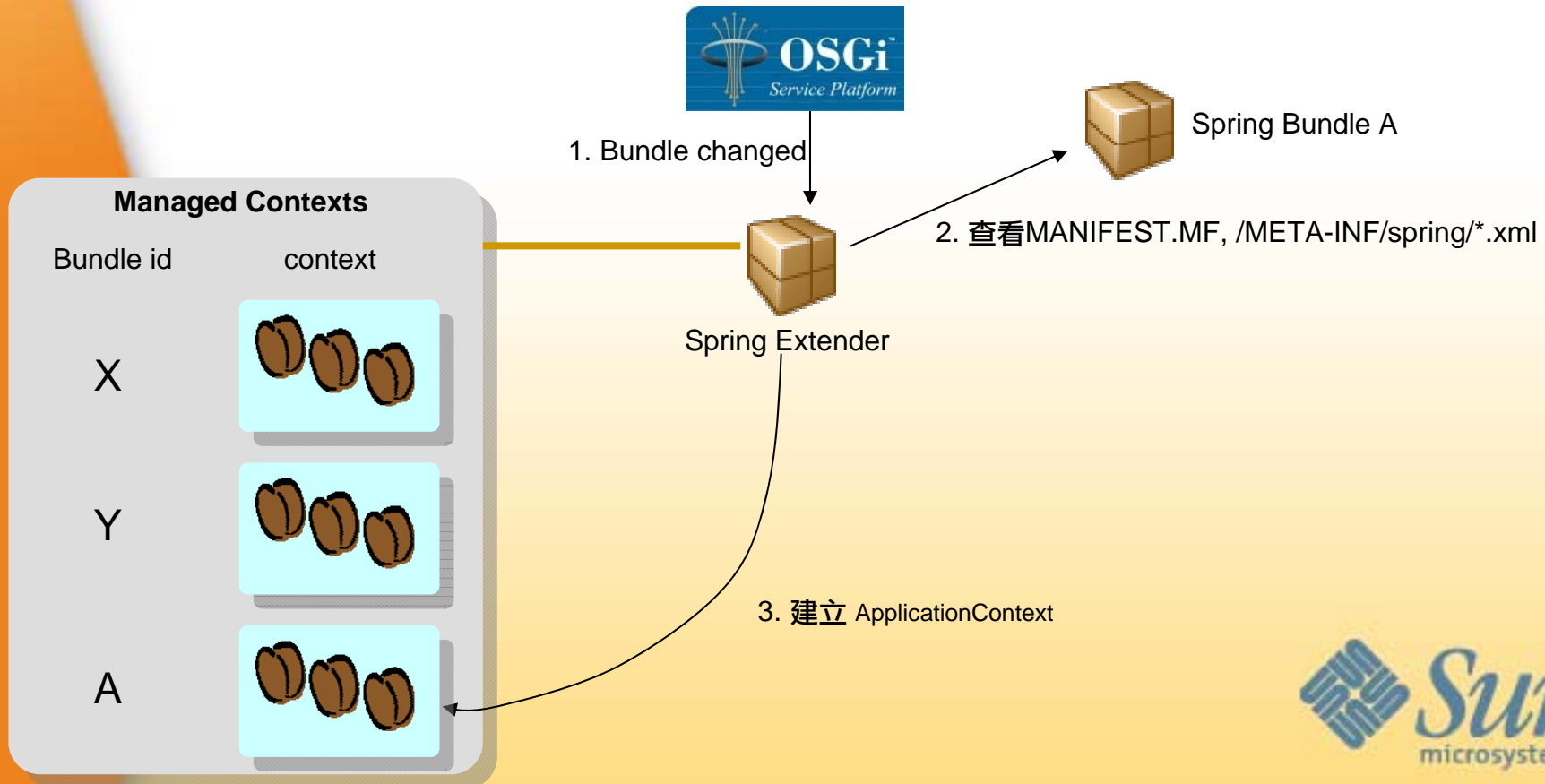
- HelloService
 - Service Interface
 - 中文版 HelloTwService
 - 英文版 HelloEnService
 - /META-INF/spring/osgi-context.xml
- HelloClient
 - HelloClient
 - /META-INF/spring/osgi-context.xml

Extender

Spring DM的運作關鍵: Extender

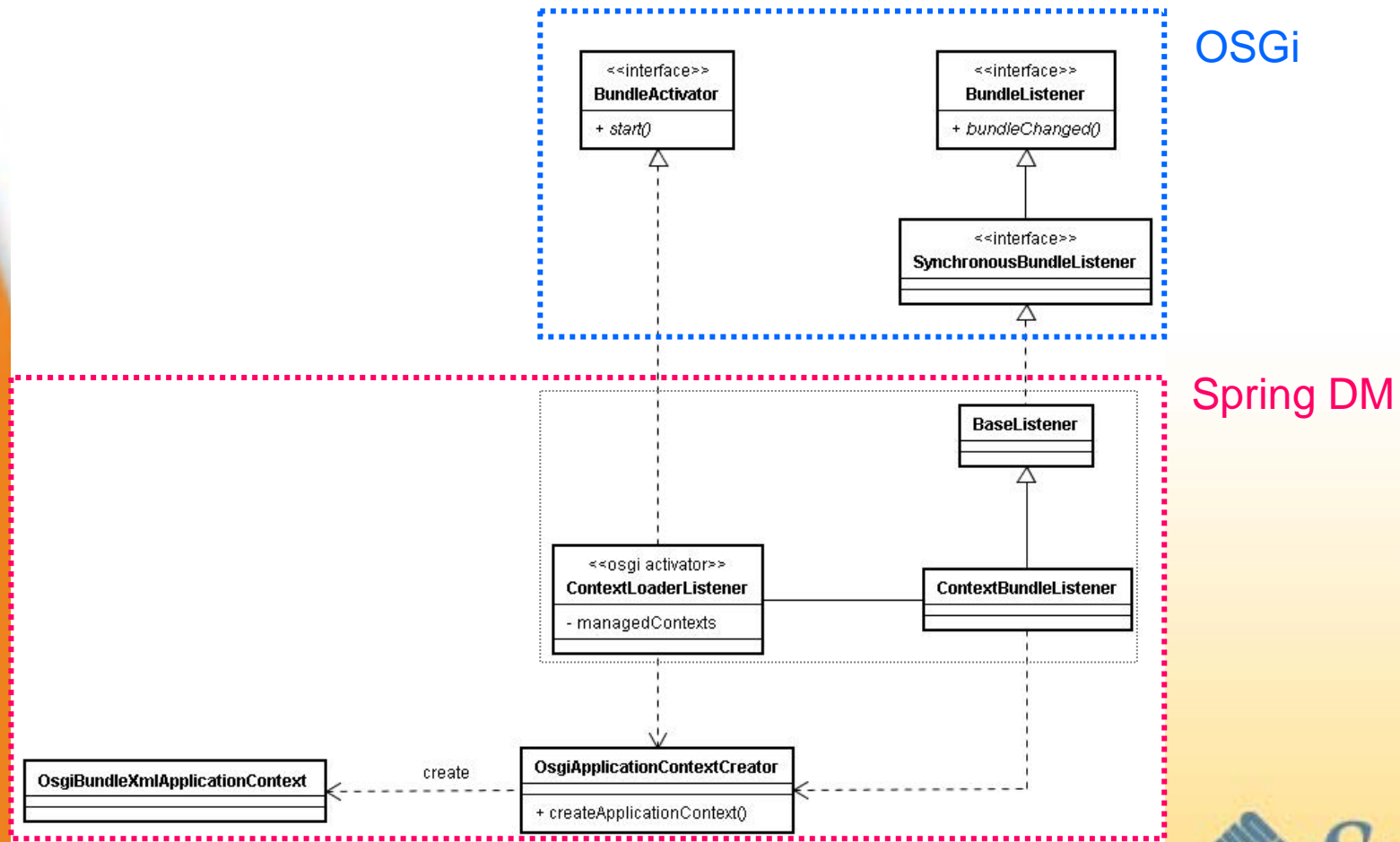
2009 Java 認證日
2009 Java Certification Day

- 掃瞄其它bundles的內容或標頭檔案
- 根據這些檔案，幫這些bundles做一些動作或提供一些服務



Spring DM Extender 原始碼結構

2009 Java 認證日
2009 Java Certification Day



Java EE

OSGi與Java EE

如何結合OSGi與Java EE?

- Idea 1: 在個別Java EE元件中的類別、服務OSGi化
 - 例:在war中嵌入一個OSGi容器
- Idea 2: 將Java EE伺服器OSGi化
 - 例:GlassFish 3及目前大部份的AP Servers

OSGi與Java EE

如何結合OSGi與Java EE?

- Idea 1: 在個別Java EE元件中的類別、服務OSGi化
 - 例:在war中嵌入一個OSGi容器
- Idea 2: 將Java EE伺服器OSGi化
 - 例:GlassFish 3及目前大部份的AP Servers
- Idea 3: 將Java EE元件OSGi化
 - 可能會必須對Java EE元件規格做一定程度的修改

SpringSource dm Server 兼具了上述Idea 2, Idea 3

SpringSource dm Server

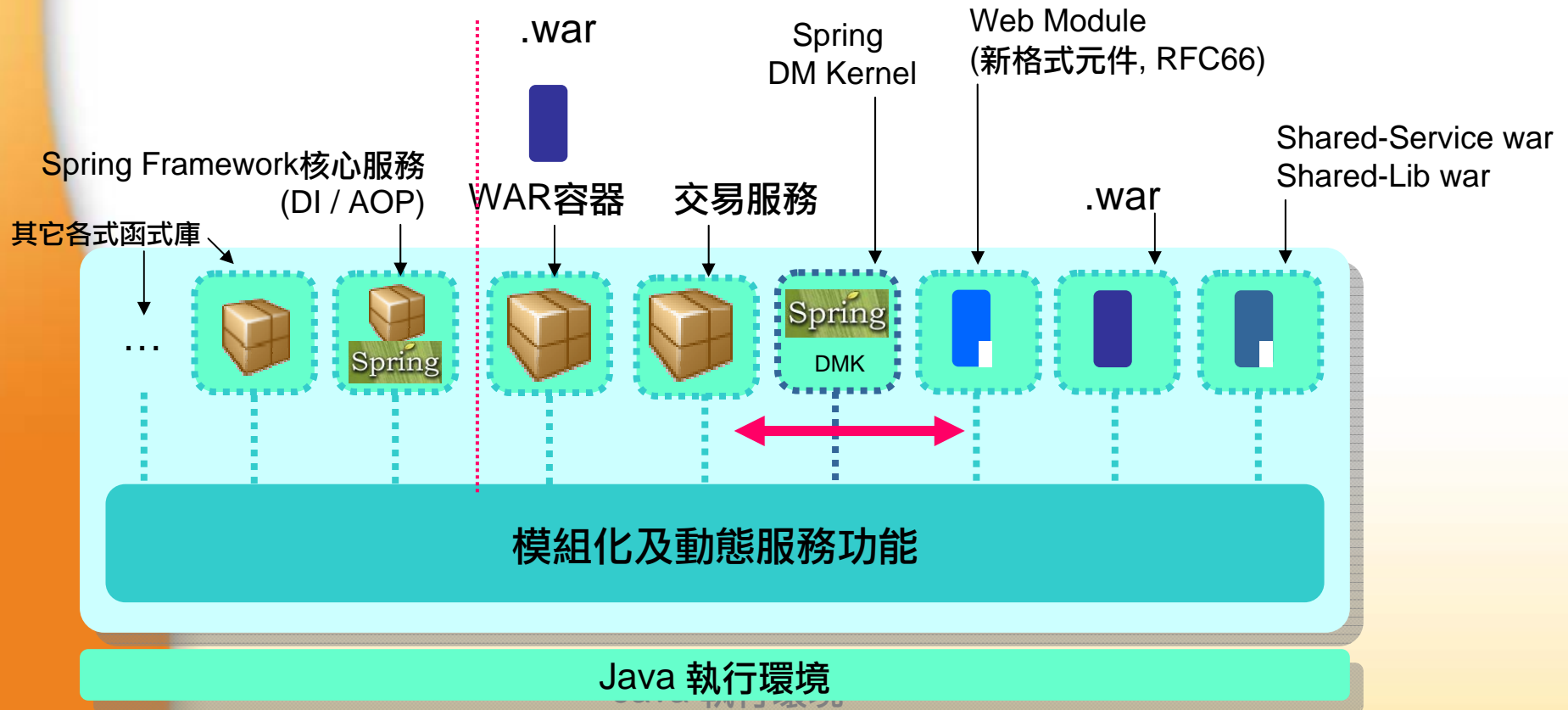
2009 Java 認證日
2009 Java Certification Day

- 革命性的思維
 - 元件容器、元件、服務...全部都是OSGi bundles
 - 為Java EE應用程式開發人員的傳統思維帶來衝擊
 - 定義新的企業元件模型
- 將所有東西都OSGi化: 以OSGi jar做為通用模組格式
 - 以Bundle型式取代傳統Java EE Archives (war, ejb jar)
 - 以PAR型式取代EAR
 - 難度: 相容性、ClassLoader語意及存取整合



將Java EE元件OSGi化

2009 Java 認證日
2009 Java Certification Day



Demo

- 參觀一下SpringSource dm Server!

Personality

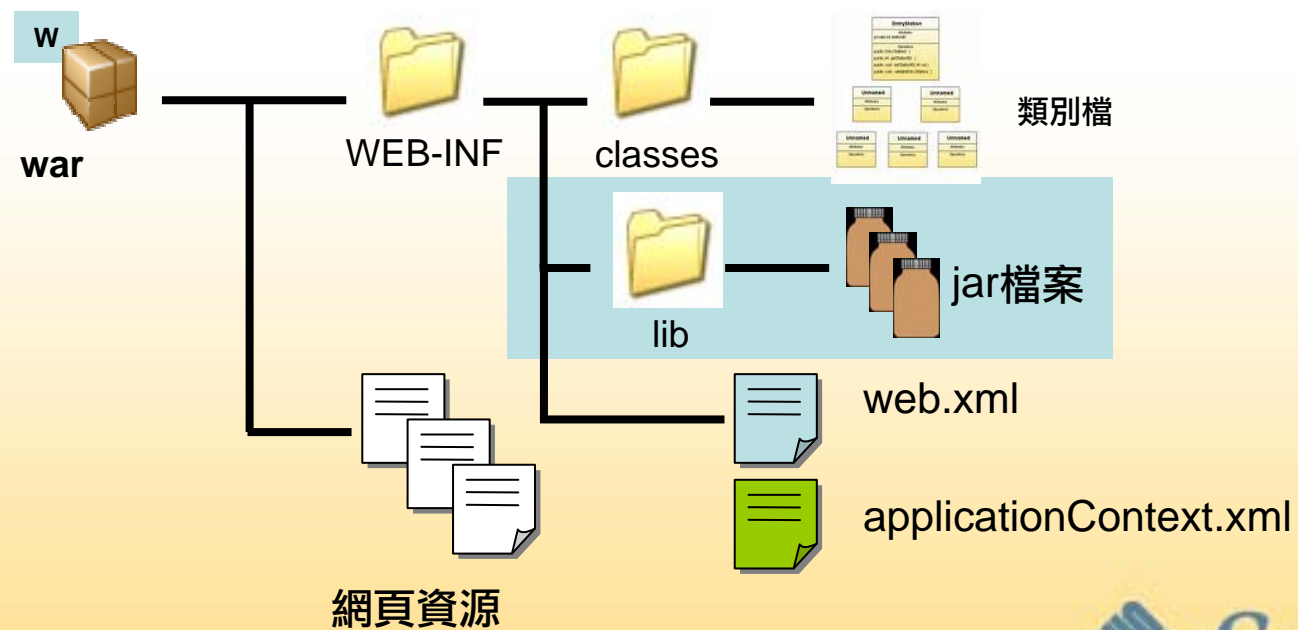
- Personality = Application Type
- SpringSource dm Server中的Personalities
 - Library Bundles
 - Java EE war
 - Shared Library war
 - Shared Service war
 - Web Modules
 - PAR

到Web Module之路

- Standard war
- Shared Library war (Library共享)
- Shared Service war (Service共享，不同Layer, 不同bundle)
- Web Module (完全OSGi化)

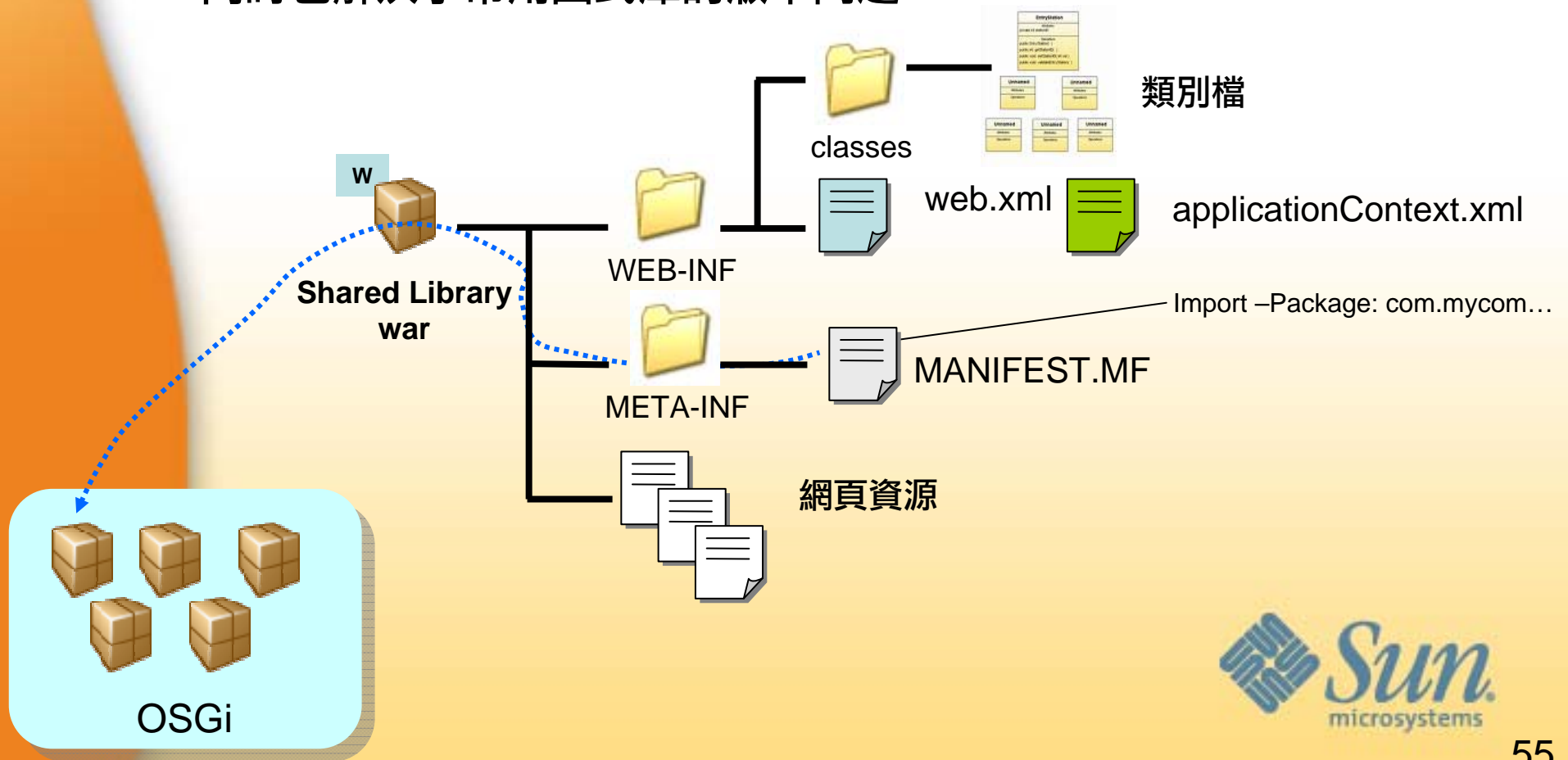
標準的war檔案

- 標準的war檔案可以直接以bundle方式佈署到dm Server上
- dm Server內建的web extender會自動產生bundle相關檔案



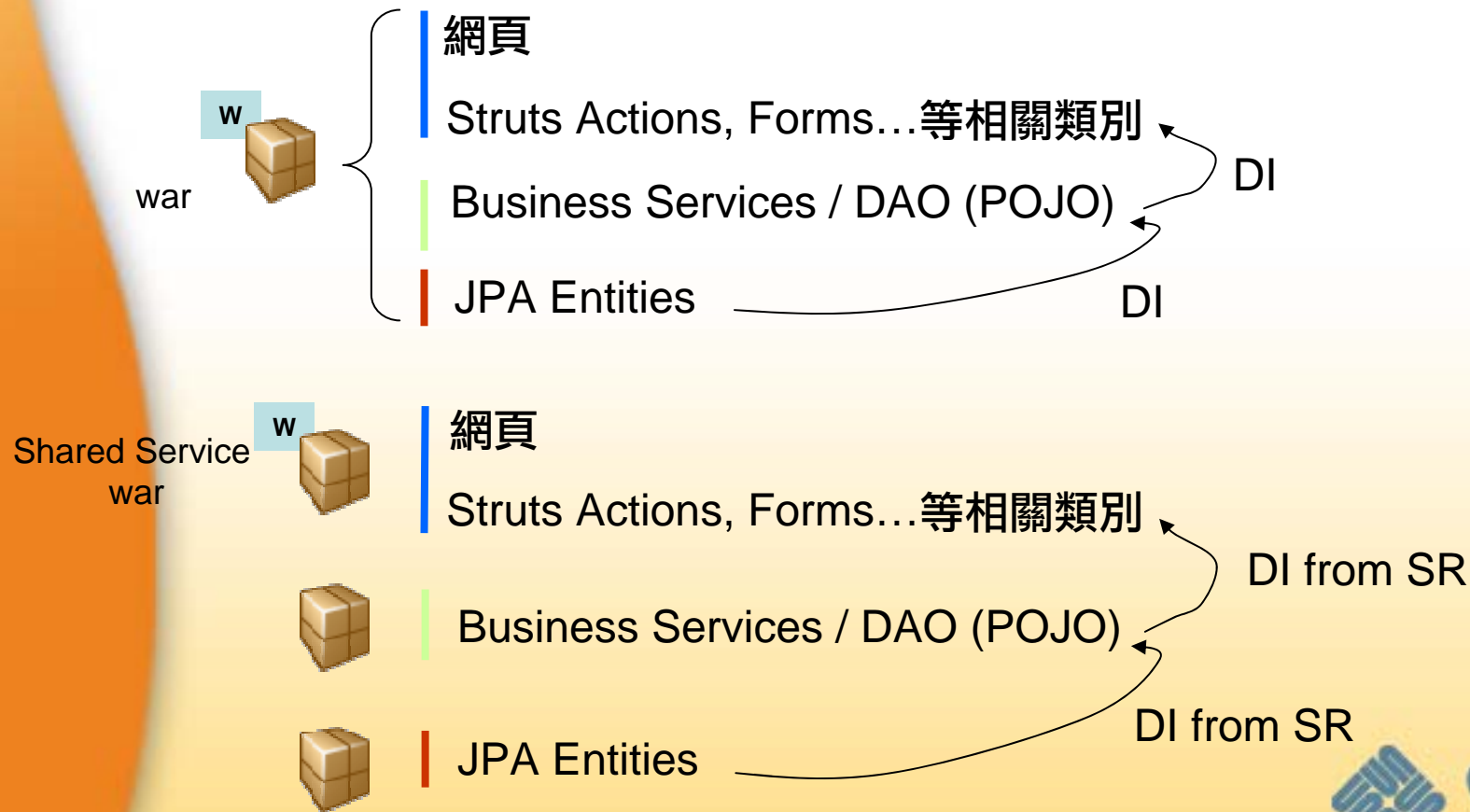
Shared Library war

- WEB-INF/lib中的jar檔以OSGi import方式取代
- 好處: 同一份函式庫的jar檔可以共享，節省AP Server的記憶體空間，同時也解決了常用函式庫的版本問題



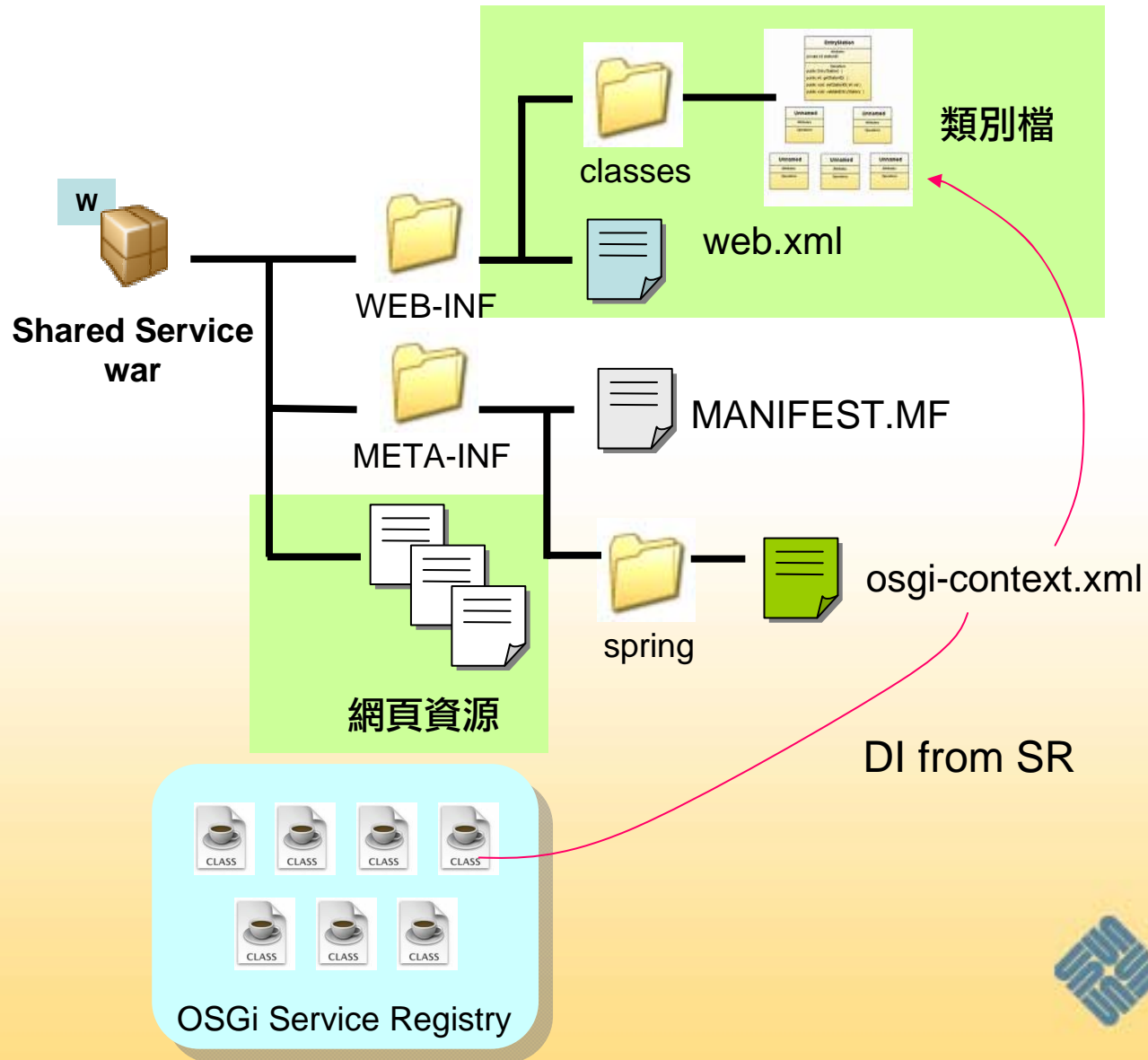
Shared Service war

- 徹底實施多層式架構模組化理念，1 layer 1 bundle，不同 layer 間透過 OSGi Service Registry 結合



Shared Service war

2009 Java 認證日
2009 Java Certification Day



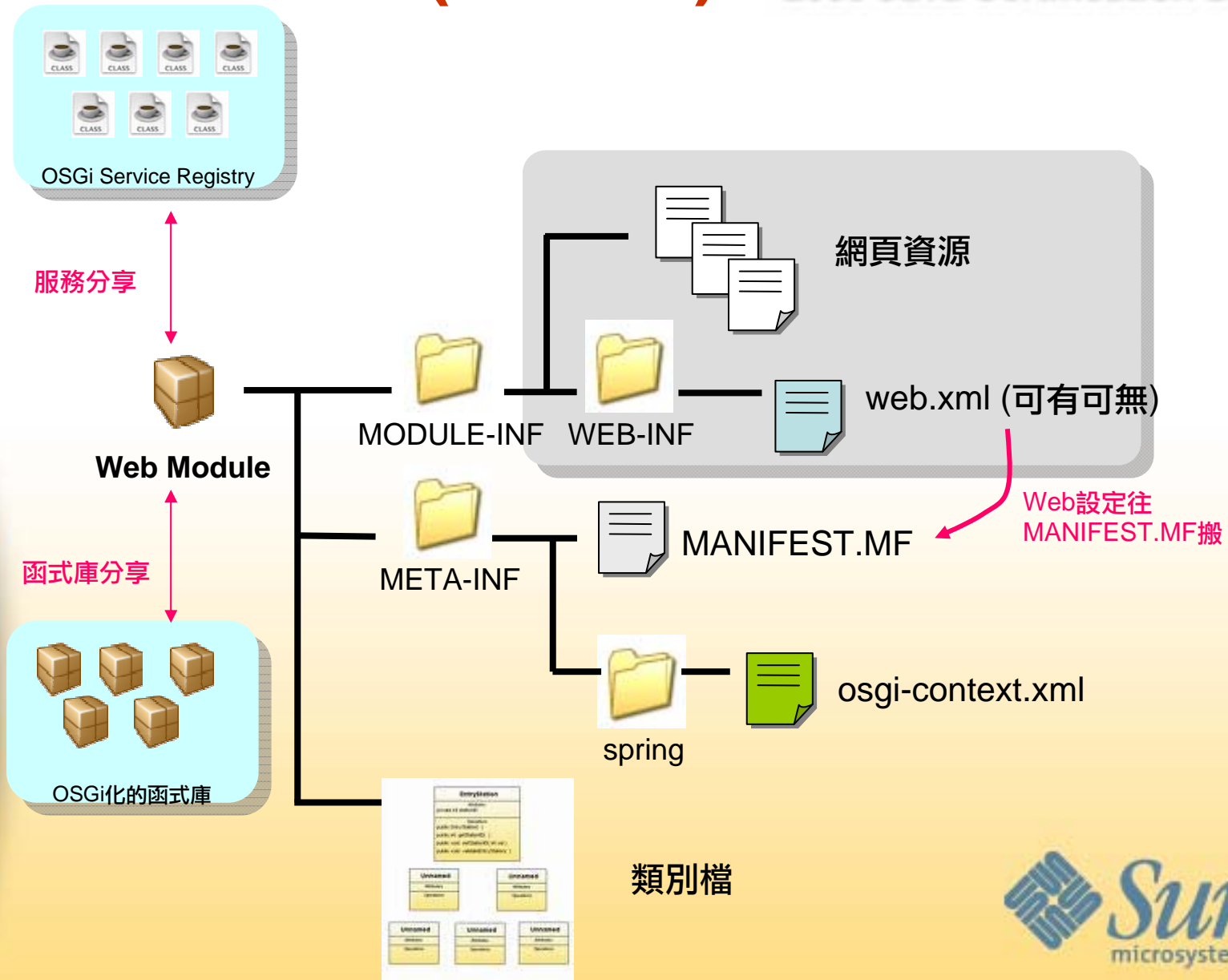
Web Module

- 完全OSGi化的Web 應用程式
 - 其它layers的服務透過Spring DM注入
 - 所用到的函式庫透過Import-*系列headers引入
 - 以OSGi bundle為主的結構，原web相關資源移到MODULE-INF中
 - Java類別向外搬到根目錄
 - 大部份Web設定在MANIFEST.MF做，將web.xml的角色淡化 (可有可無)

```
Manifest-Version: 1.0
Module-Type: Web
Bundle-Version: 1.0.0
Bundle-Name: Web3 Bundle
Bundle-ManifestVersion: 2
Web-ContextPath: www
Bundle-SymbolicName: demo
Import-Package: com.mycom.ifs;version="[1.0.0,1.0.0]",
org.slf4j;version="[1.5.6,1.5.6]"
Application-TraceLevels: *=info,com.mycom.web3.*=debug
```

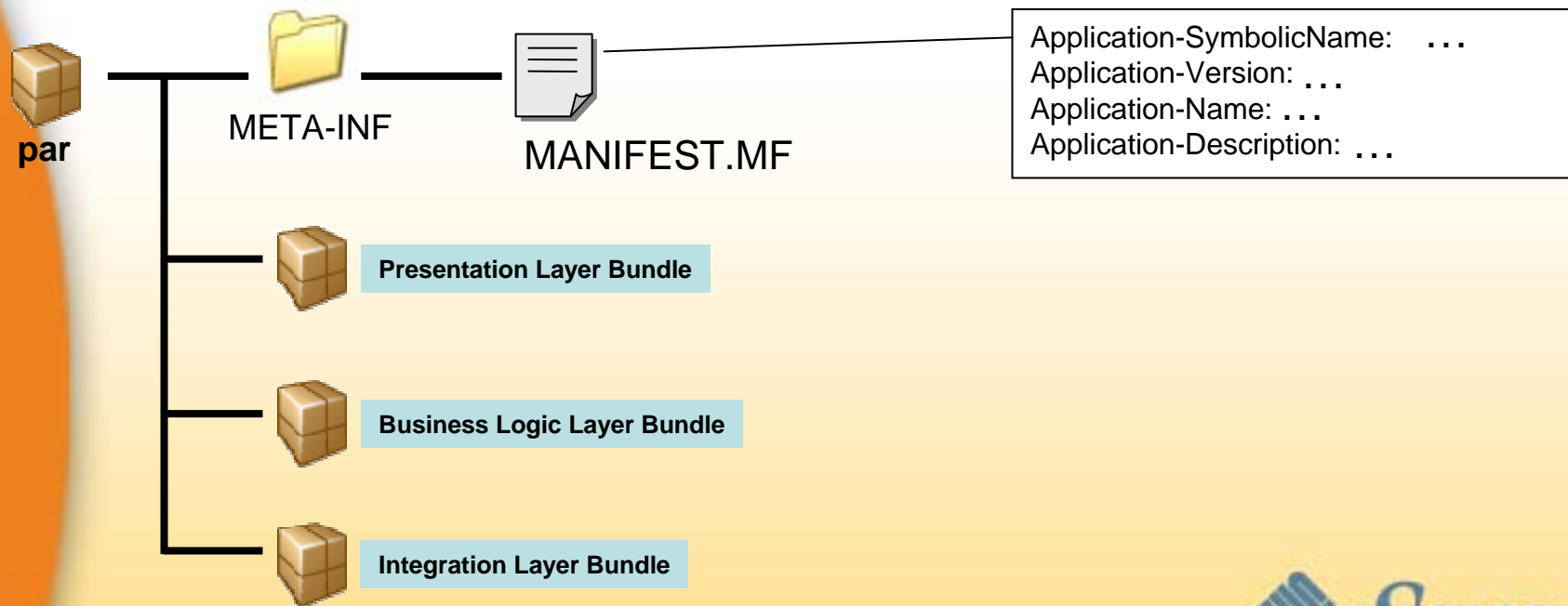


Web Module (RFC66)



PAR (Platform ARchive)

- OSGi化的EAR，為一完整的企業應用程式佈署單位
- 解決無法跨bundle載入類別的問題 (如JPA)
 - OSGi: 除非明確宣告，否則無法載入位於其它bundles的類別
 - JPA: 事先不知道那些entities需要載入
- 打包方式: 將各layers的bundles放在一起，連同MANIFEST.MF一起壓成PAR即可



2009.8.10

SpringSource Chapter Two



3.62億美元現金與5800萬美元的股票



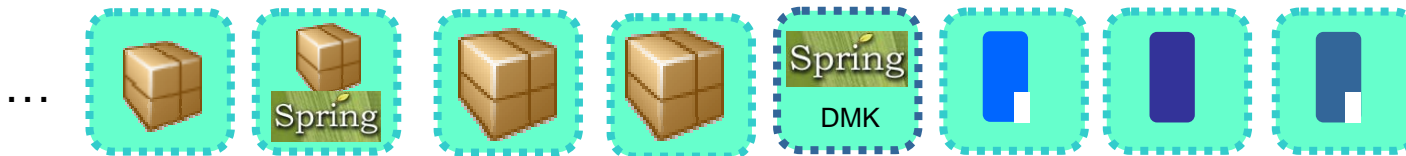
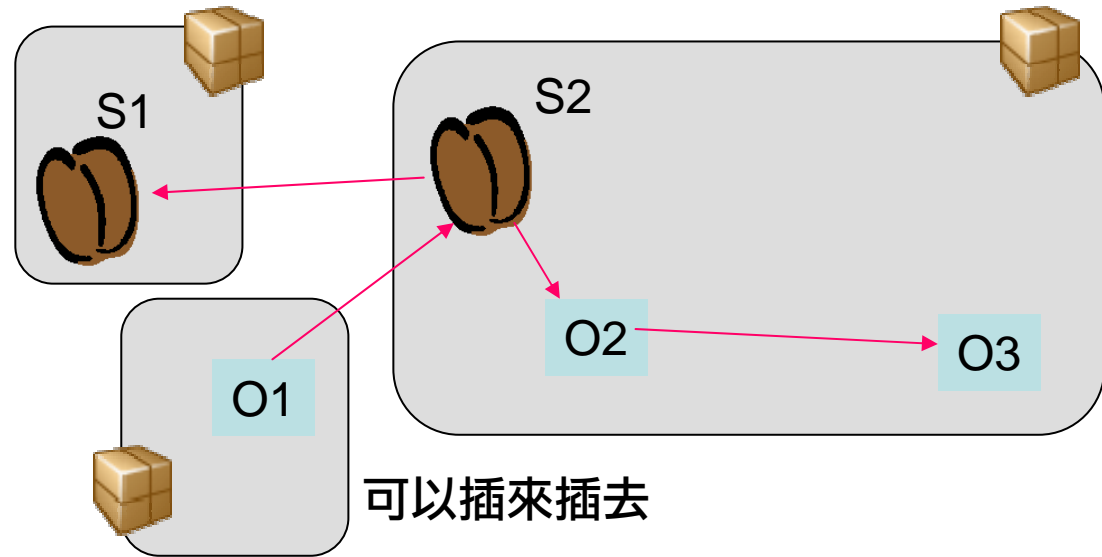
+



= Spring DM



好用



革命性概念



Thank You

廖峻鋒 (try)

<http://www.try.idv.tw> (Personal)

<http://www.attentivehome.org/> (Lab)

cfliao@ieee.org

