

La plate-forme dynamique de service OSGi™

Didier Donsez

Université Joseph Fourier (Grenoble 1)

PolyTech Grenoble LIG/ERODS

`Firstname.Lastname@imag.fr`

`Firstname.Lastname@ieee.org`

<http://membres-liglab.imag.fr/donsez/cours/osgi.pdf>

Sommaire

- ⑦ **Motivations et Rappels**
- ⑦ **Conditionnement et Service**
- ⑦ **Enregistrement et recherche de services**
- ⑦ **Composants**
- ⑦ **Services standards (survol)**
- ⑦ **Acteurs, Concurrences et Perspectives**

Qu'est ce que OSGi™ ?

⑦ Spécification OSGi

- ⑦ définit un canevas de déploiement et d'exécution de services Java
- ⑦ multi-fournisseur, télé-administré
- ⑦ Cible initiale : set top box, modem cable, ou une passerelle résidentielle dédiée.

⑦ OSGi Alliance

- ⑦ Corporation indépendante
- ⑦ Soutenus par les acteurs majeurs des IT, home/building automation, telematics (car automation), ...
- ⑦ de la téléphonie mobiles (Nokia et Motorola)

- ⑦ et Eclipse pour les plugins de son IDE !
- ⑦ et maintenant Apache pour ses serveurs

Qu'est ce que OSGi™ ?

⑦ Histoire

- ⑦ Mars 1999 : Fondation de l'OSGi Alliance
- ⑦ Novembre 1999: SUN transfère le JSR008 du JCP à OSGi
- ⑦ 1.0 : Mai 2000 (189 pages)
- ⑦ 2.0 : Octobre 2001 (288 pages)
- ⑦ 3.0 : Mars 2003 (602 pages)
- ⑦ 4.0: Octobre 2005 (1000 pages)
- ⑦ 4.1: Juin 2007 (optimisation du

OPEN SERVICES GATEWAY INITIATIVE (OSGi) TO DRIVE DEVELOPMENT OF GATEWAY STANDARD FOR HOMES, SOHO AND REMOTE LOCATIONS

Sun's Java™ Technology Accelerates Development of Specification

PALO ALTO, Calif., - November 22, 1999 - Open Services Gateway Initiative (OSGi) and Sun Microsystems, Inc. announce today that Sun has transferred the effort to define an open gateway specification from the Java™ Community Process to the Open Services Gateway Initiative. ...

⑦ Remarque

- ⑦ *Open Services Gateway Initiative est un terme obsolète*

L'ancêtre :

JSR-8 : Open Services Gateway (OSG)

⑦ Java Embedded Server

- ⑦ JavaOne e-Fridge

⑦ Domain : SOHO / ROBO Gateway

⑦ EG

- ⑦ Spec leader : Robert Mines (Sun)
- ⑦ Sun Microsystems, IBM, Nortel, Alcatel, Cable and Wireless, EDF, Enron, Ericsson, Lucent, Motorola, NCI, Phillips, Sybase, Toshiba

⑦ Package names

- ⑦ javax.osg.servicespace
- ⑦ javax.osg.remote
- ⑦ javax.osg.service

⑦ Transferred to the OSGi Alliance

Principales propriétés du canevas OSGi

⑦ Modularisation des applications

⑦ Chargement/**Déchargement** de code dynamique

⑦ Langage Java

⑦ Déploiement dynamique d'applications
sans interruption de la plateforme

⑦ Installation, Lancement, Mise à jour, Arrêt, Retrait

⑦ « *No reboot* »

⑦ Résolution des dépendances **versionnées** de code

⑦ Architecture **orientée service**

⑦ Couplage faible, late-binding

⑦ Reconfiguration dynamique des applications (plugins, services techniques)

⑦ **Visée des systèmes à mémoire restreinte**

⑦ s'accroche à J2ME/CDC

⑦ même si de plus en plus Java Platform 1.5, 6, 7, ...

Rappel sur les chargeurs de classes

⑦ **java.lang.ClassLoader**

- ⑦ Objet (java) chargé de charger en mémoire la définition des classes (.class)

⑦ **Principe de la délégation**

- ⑦ Tout chargeur a un chargeur parent
 - ⑦ **sauf le chargeur primordial**
- ⑦ Tout chargeur vérifie si la classe à charger n'a pas déjà été chargée par un chargeur parent

⑦ **Arbre de délégation basique**

- ⑦ **ClassLoader bootstrap ou primordial**
 - ⑦ **sun.misc.Launcher\$ExtClassLoader** (*extension*)
 - ⑦ **sun.misc.Launcher\$AppClassLoader** (*application ou system*)
- Possibilité de personnaliser les chargeurs

Rappel sur les chargeurs de classes

Pourquoi utiliser les chargeurs de classes

- ⑦ **Classes non présentes dans le CLASSPATH ou le \$JAVA_HOME/lib/ext**
 - ⑦ URLClassLoader, AppletClassLoader, RMIClassLoader...
 - ⑦ ex: WEB-INF/classes et WEB-INF/lib d'une WebApp
 - ⑦ ex: CODEBASE d'une applet, ...
- ⑦ **Emballage particulier**
 - ⑦ JavaEE EAR, OSGi bundle (fichiers JAR imbriqués), java.util.jar.Pack200, Google Android DEX format ...
- ⑦ **Modification du ByteCode à la volée au chargement**
 - ⑦ Instrumentation
 - ⑦ AOP (Aspect Oriented Programming)
 - ⑦ BCEL, ASM, ...
 - ⑦ Protection
- ⑦ **Chargement de ressources associées à la classe**
 - ⑦ properties, images, ...
- ⑦ **Déchargement et Mise à jour du bytecode lors de l'exécution de la VM (runtime)**
 - ⑦ Chargeurs de OSGi

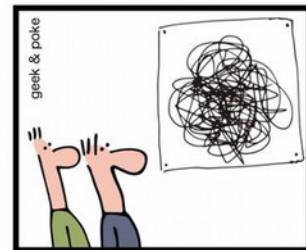
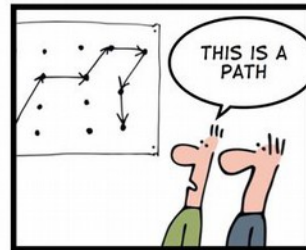
Rappel sur les chargeurs de classes

Pourquoi NE PAS utiliser les chargeurs de classes

- ⑦ **Beaucoup trop complexe pour le commun des mortels (et également pour les autres)**
- ⑦ **Indispensable de comprendre le fonctionnement !**
- ⑦ **Car malheureusement beaucoup bricolent avec !**

Classpath Hell

GRAPH THEORY FOR GEEKS



⑦ <http://geekandpoke.typepad.com/geekandpoke/2011/07/graph-theory-for-geeks.html>

Rappel sur la programmation OO

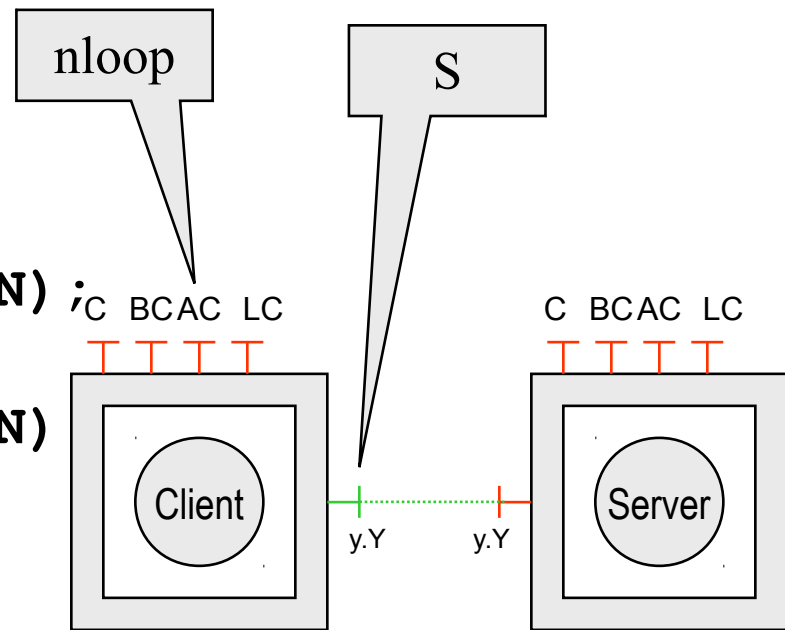
« *programming in the small* »

- ⑦ Un client **C** invoque **N** fois la méthode **execute()** d'un serveur **S**
- ⑦ **S** `s=new S ()`
- ⑦ **C** `c1=new C (s, N) ;`
- ⑦ **C** `c2=new C (s, N) ;`
- ⑦ **Problème: Architecture ? Configuration ?**

Rappel sur la programmation Composant « programming in the large »

⑦ Un client **C** invoque **N** fois la méthode `execute()` d'un serveur **S**

```
⑦ S s=SFactory.create()  
⑦ C c1=CFactory.create();  
⑦ C c2=CFactory.create();  
⑦ c1.setProperty("nloop",N);  
⑦ c1.bind("S",s);  
⑦ c2.setProperty("nloop",N);  
⑦ c2.bind("S",s);  
⑦ s.start()  
⑦ c1.start();  
⑦ c2.start();  
⑦ ...
```



Rappel sur la programmation Composant

```
⑦ ...  
⑦ S s2=SFactory.create()  
⑦ c2.stop();  
⑦ c2.unbind("S");  
⑦ c2.bind("S",s2);  
⑦ s2.start()  
⑦ c2.start();
```

Design Patterns : Factory, IoC, DI, Convention de nommage

⑦ **Problème: Multi-domaines d'administration**

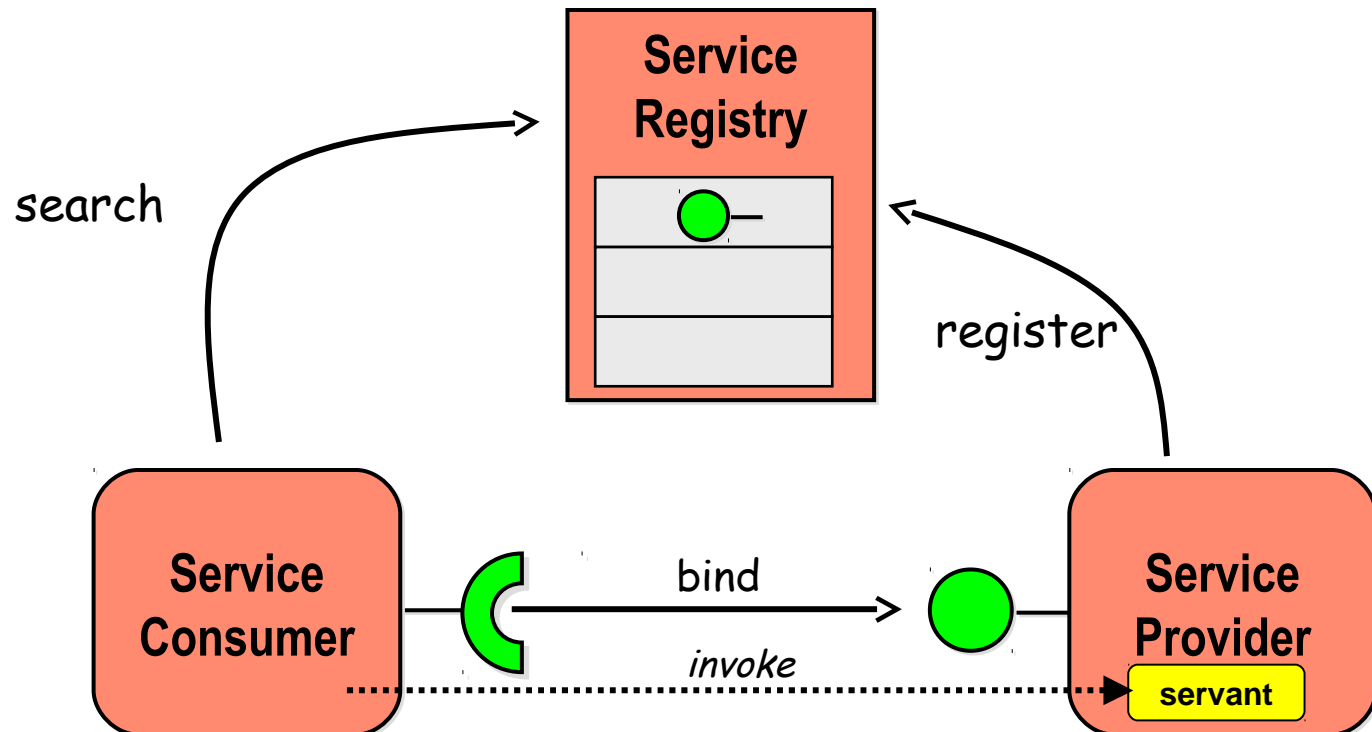
⑦ carte GSM SIM (Javacard) , WS, iTV STB, passerelle domotique, ...

@TODO ADL

Rappel:

Architecture orienté service (SOA) « programming in the VERY large »

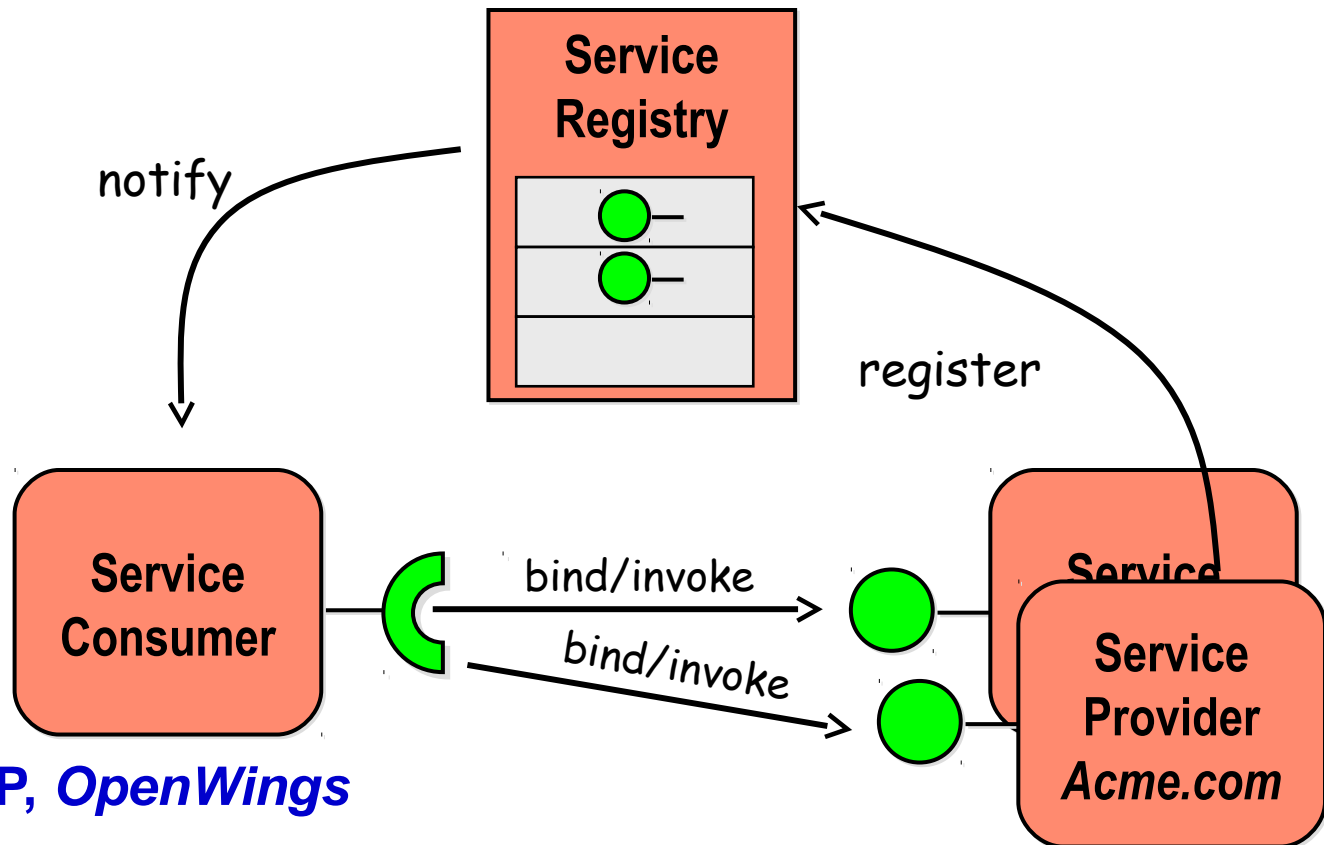
- ⑦ Les services (contrats) ● sont « invariants »



- ⑦ WebServices, CORBA COS Trading, ...

Rappel: SOA Dynamique

⑦ Arrivée dynamique de nouveaux services

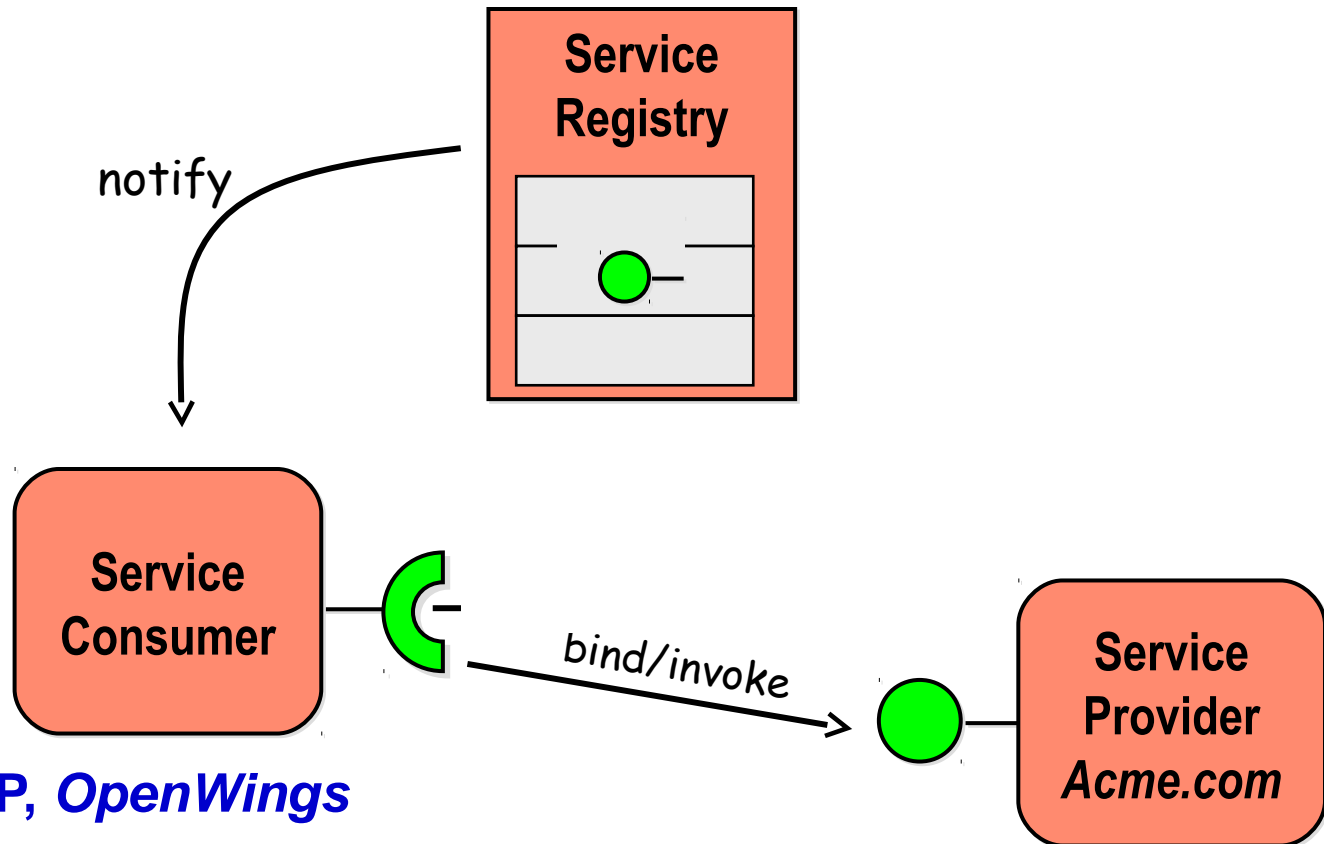


⑦ JINI, UPnP, OpenWings

⑦ OSGi

Rappel: SOA Dynamique

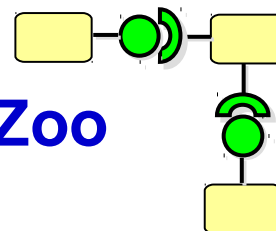
⑦ Retrait dynamique de services utilisés



⑦ JINI, UPnP, *OpenWings*

⑦ OSGi

Dynamic Service Platform Zoo



	Invocation	Removal	Registry Type	Programming Language
JINI	Remote (RMI)	Lease	Distributed (ad-hoc)	Java
OpenWings	Remote (RMI IIOP)	Connector	Distributed (?)	Java
CORBA CosTrading	Remote (IIOP)	No	Distributed (?)	all
UPnP V1	Remote (HTTP/SOAP1.0)	Message Bye	Distributed (ad-hoc)	all
Web Services DPWS	Remote (HTTP/SOAP1.2)	No Message Bye	Centralized (UDDI) WS-Discovering	all
SLP / DNSSD	/	Message Bye	Distributed	all
OSGi	Locale (Référence)	Java Event	Centralized	Java

OSGi

**Modèle d'administration
et Domaines d'application**

Domaines d'application

⑦ Initialement, Systèmes embarqués

- ⑦ Véhicule de transport (*automotive*)
- ⑦ Passerelle résidentiel/domotique/immotique
- ⑦ Contrôle industriel
- ⑦ Téléphonie mobile

⑦ Cependant

- ⑦ Tout concepteur d'application est gagnant à distribuer son application sous forme de plugins conditionnés dans des bundles OSGi

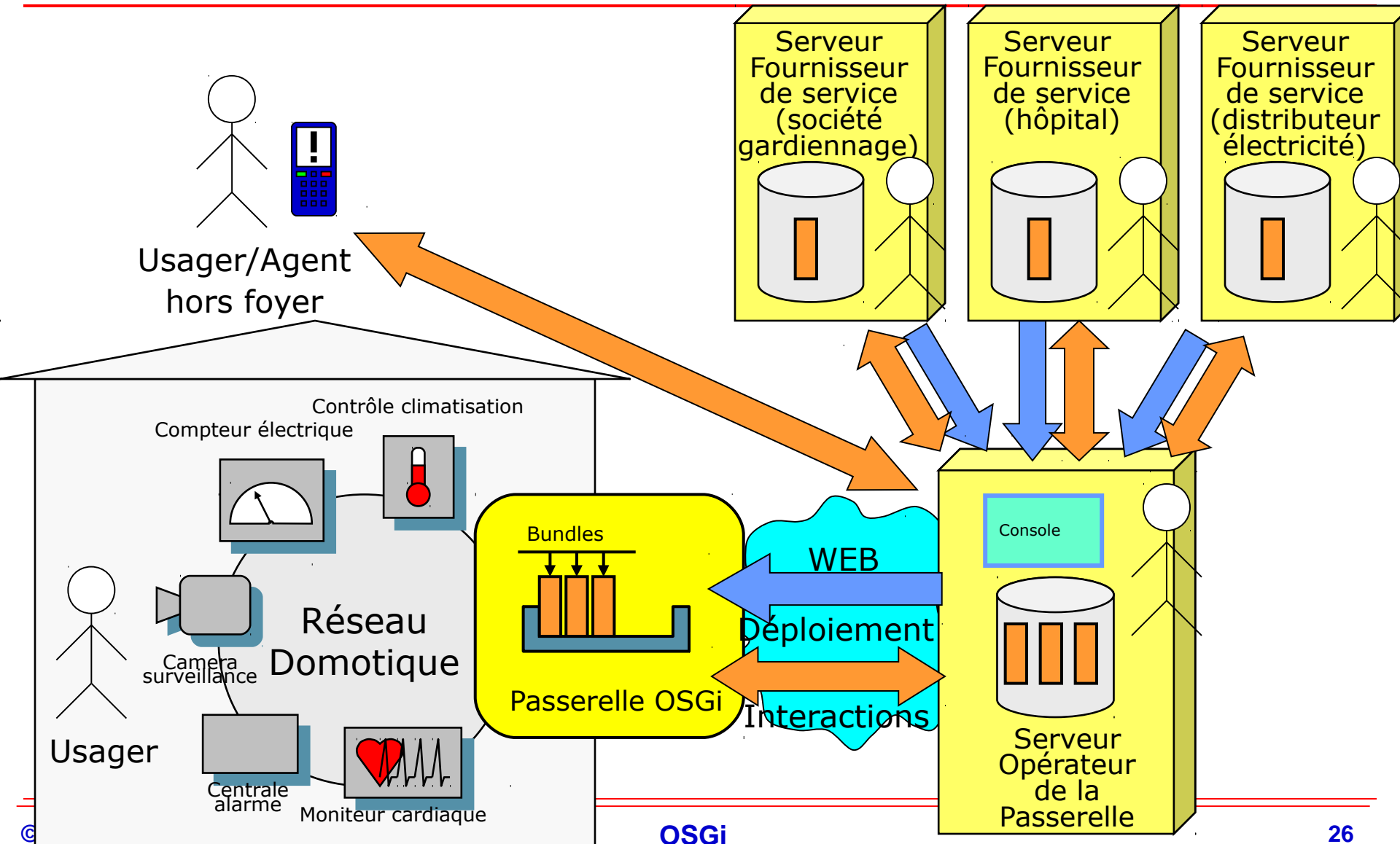


- ⑦ Cela évite l'enfer du CLASSPATH
 - ⑦ CLASSPATH, lib/ext du JRE ou JavaEE, ...

⑦ Maintenant

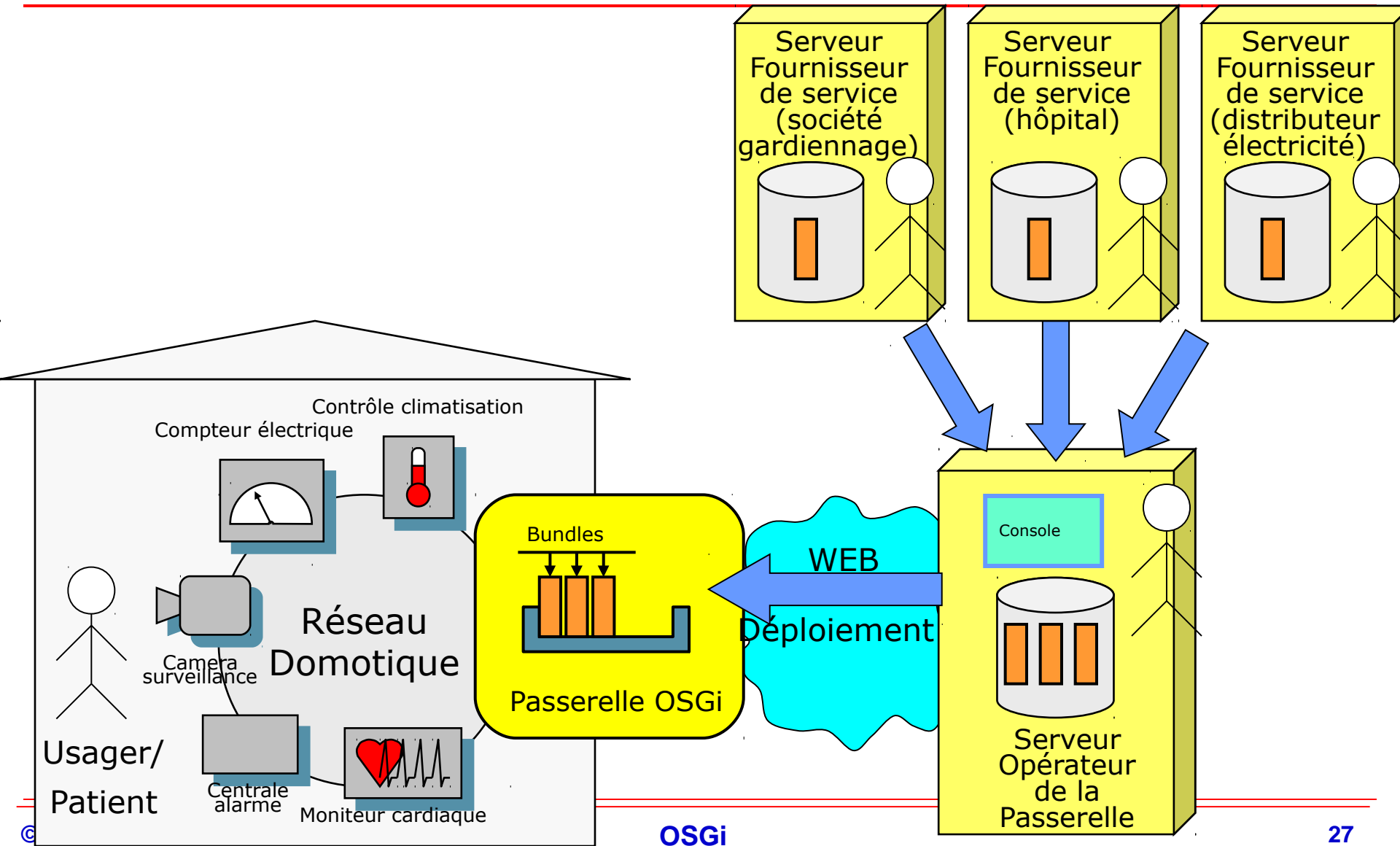
- ⑦ Eclipse RCP, JavaEE, *Harmony JRE pieces*, ...

Architecture générale



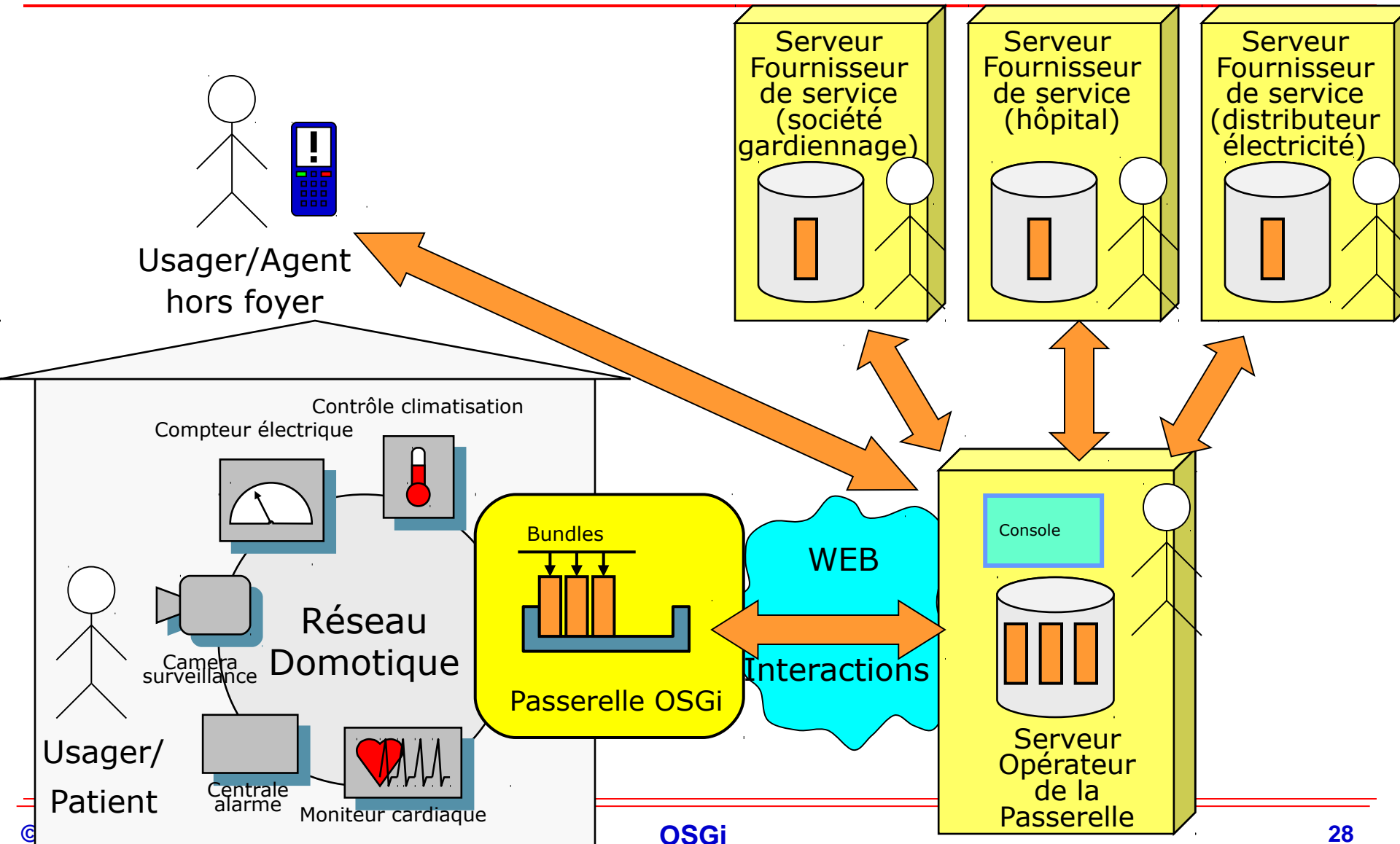
Architecture générale (i)

Déploiement

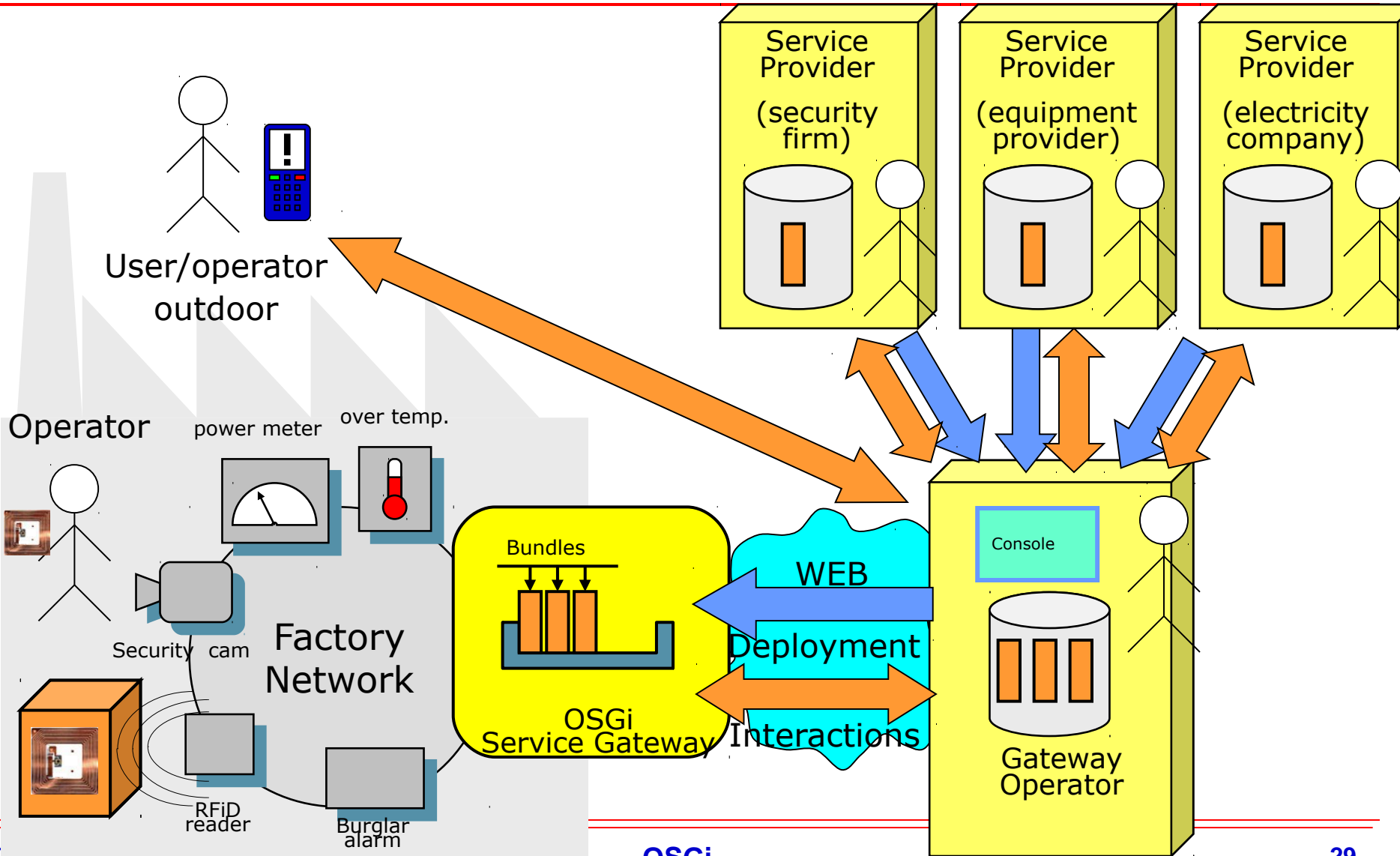


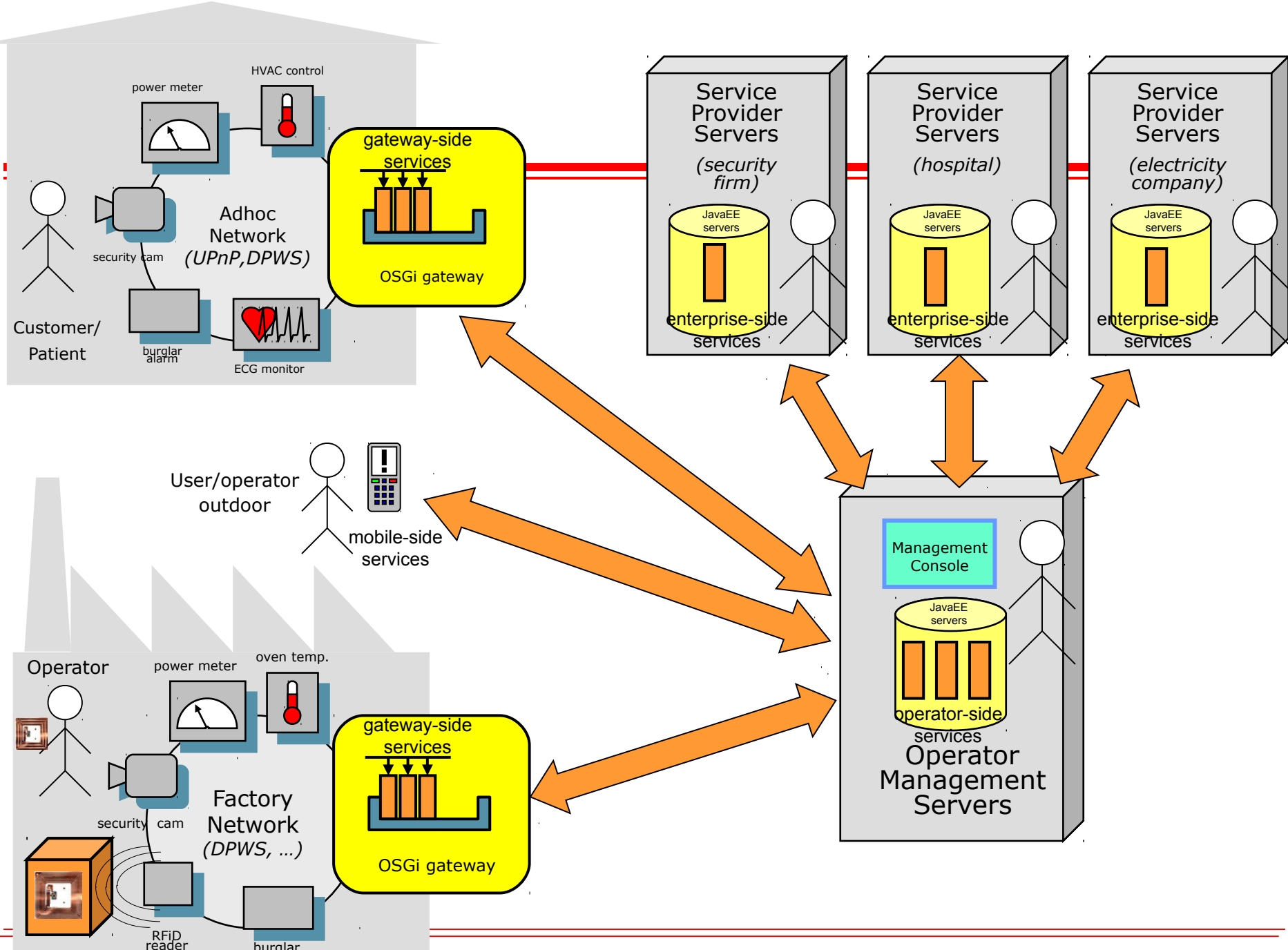
Architecture générale (ii)

Interactions

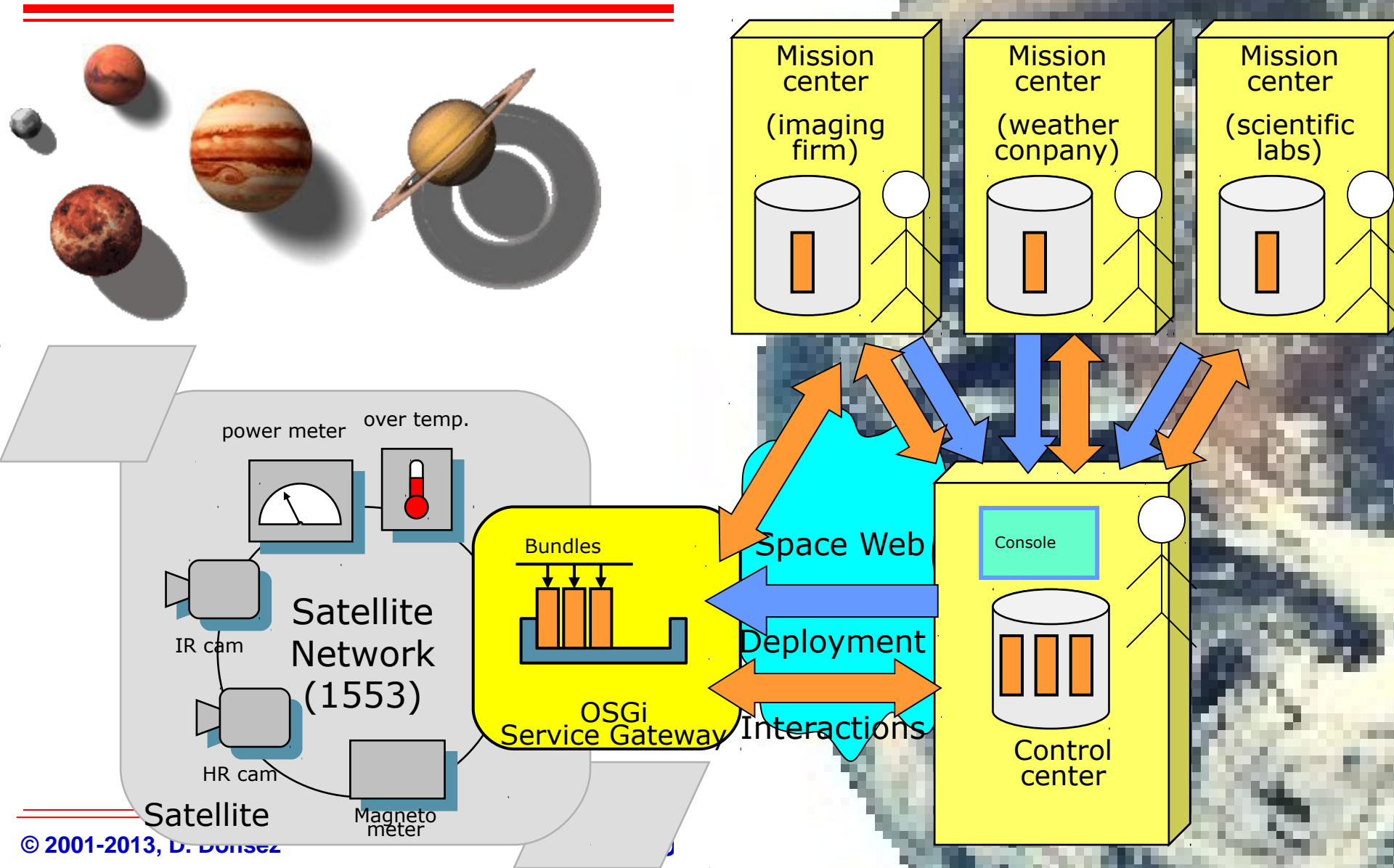


Même architecture générale (iii) Contexte différent

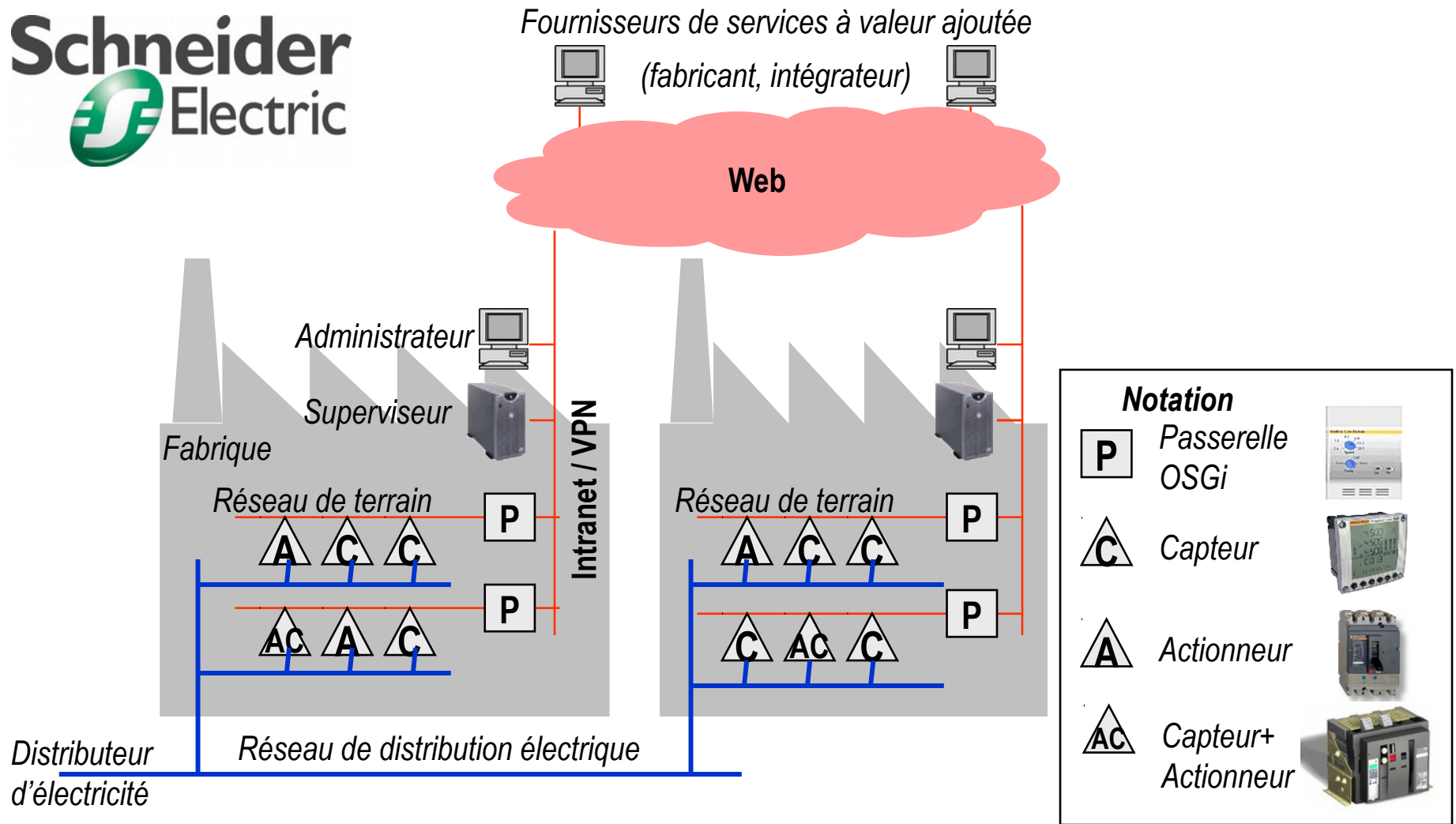








Même architecture générale (iv) Contexte différent



Application à la Distribution Electrique chez Schneider Electric



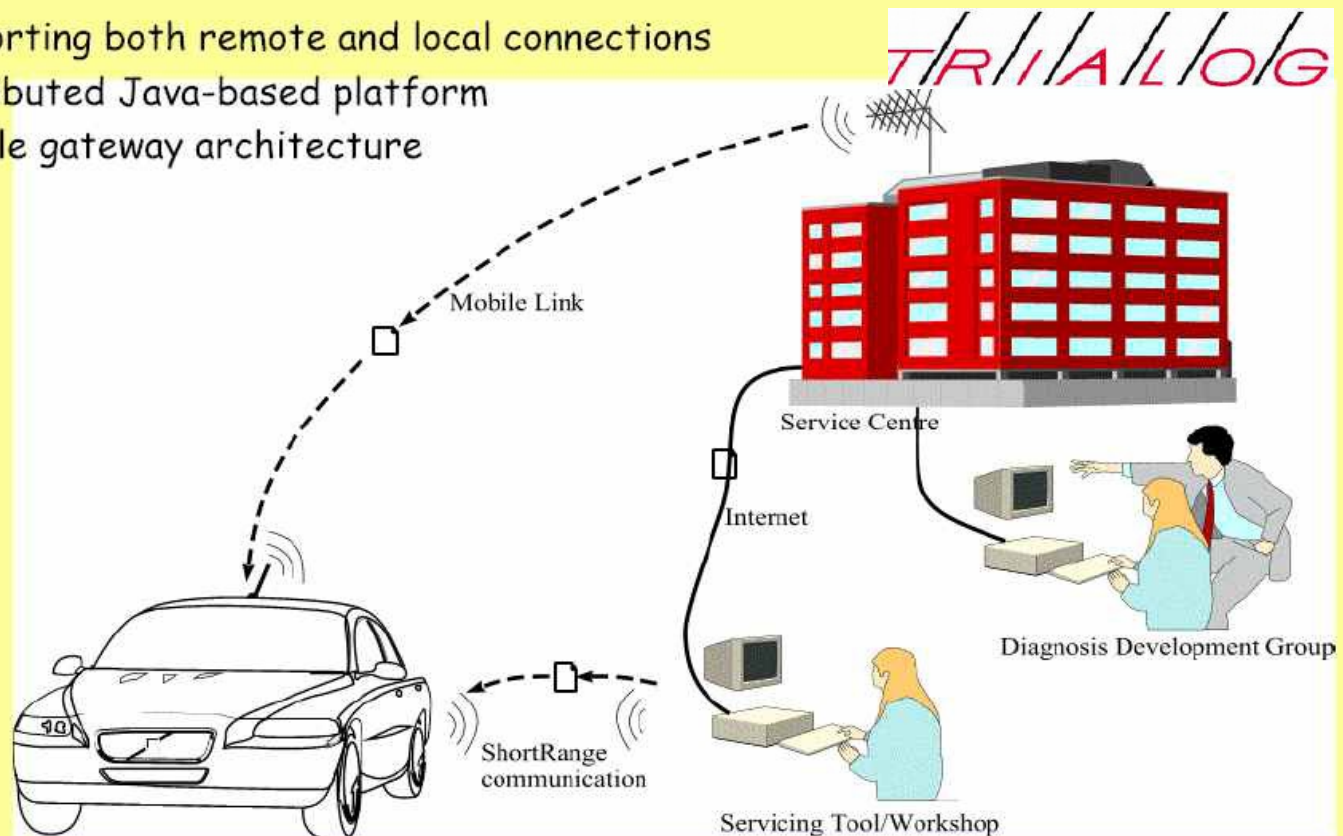
Notation

- P** Passerelle OSGi 
- C** Capteur 
- A** Actionneur 
- AC** Capteur+ Actionneur 

Diagnostic de véhicules à distance

- ⑦ Aujourd'hui la part de l'électronique dans la conception d'un véhicule est de 35%.
 - ⑦ 90% des innovations se font dans le domaine électronique
- ⑦ Cependant 50% des pannes sont provoquées par des défaillances dans ce domaine.

Supporting both remote and local connections
Distributed Java-based platform
Vehicle gateway architecture



Personal gateway

⑦ Human

- ⑦ Cardiac patient, Baby, Prisoner, Soldier on battle field

⑦ Gateway (cell phone, watch, bracelet ...)

⑦ Between

- ⑦ wearable Sensors

- ⑦ ECG, Sphygmomanometer, GPS/GSM, ...

- ⑦ and wearable actuators

- ⑦ PaceMaker, Heart Defibrillator, Tourniquet (garotte),

⑦ and providers

- ⑦ Emergency team, Parents, Prison service, Army , ...

Exemple de Scénario

Le photocopieur en location

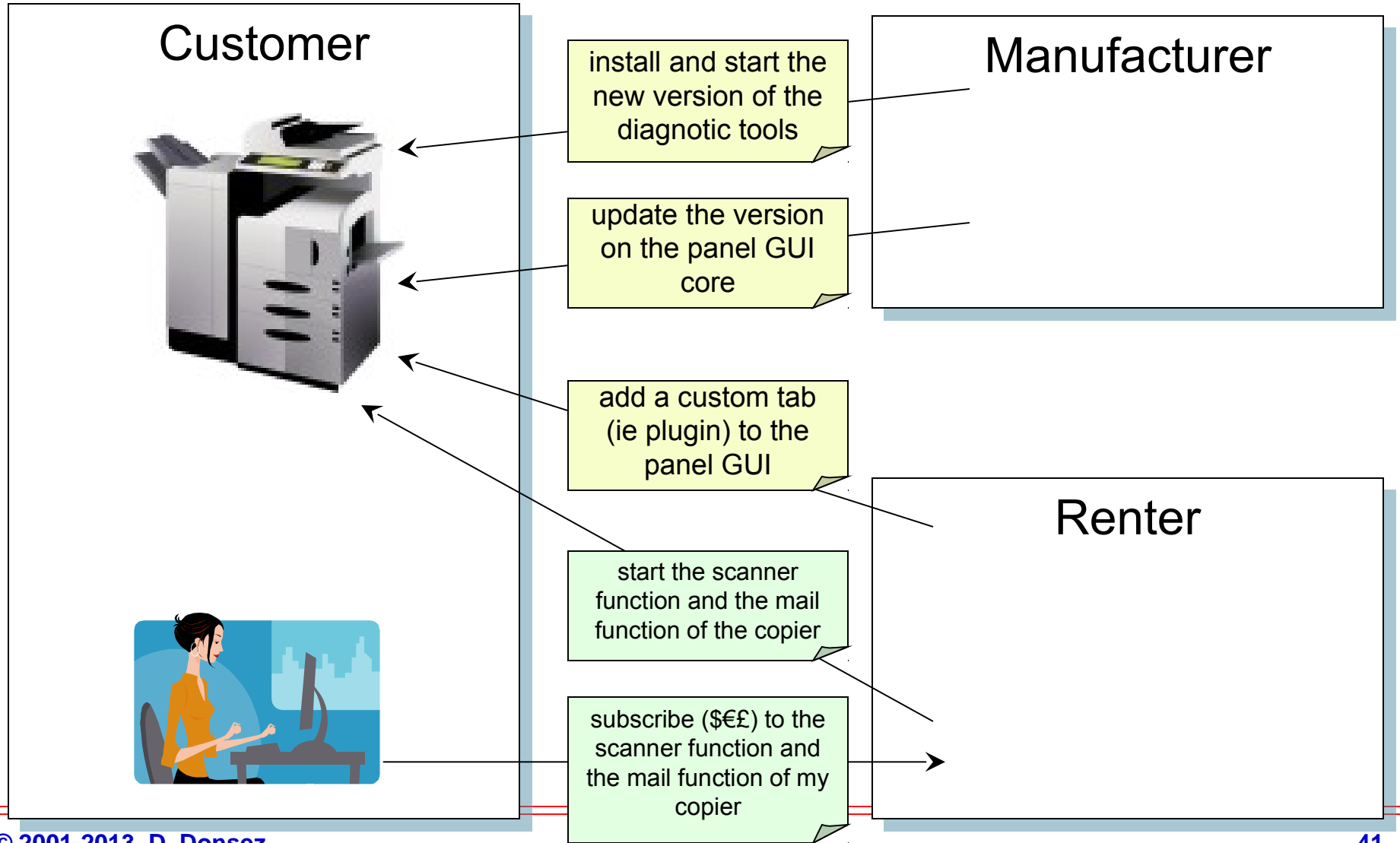


- ⑦ Le photocopieur est installé en location chez le client
- ⑦ Le loueur (est une organisme financier) facture mensuellement à la consommation
 - ⑦ Fixe mensuel + tarif par feuille
- ⑦ Le loueur sous-traite la maintenance simple à une société spécialisée
- ⑦ La société de maintenance réalise un diagnostic à distance avant d'envoyer un agent
- ⑦ L'agent de maintenance interroge sur place le logiciel de diagnostic
- ⑦ Le fabricant peut mettre à jour le logiciel embarqué
 - ⑦ RICOH (26% copier market share) inclut une passerelle OSGi dans ses photocopieurs (en 2006).
 - ⑦ <http://www2.osgi.org/wiki/uploads/Conference/OSGiCommunityBushnaq.pdf>



Exemple de Scénario

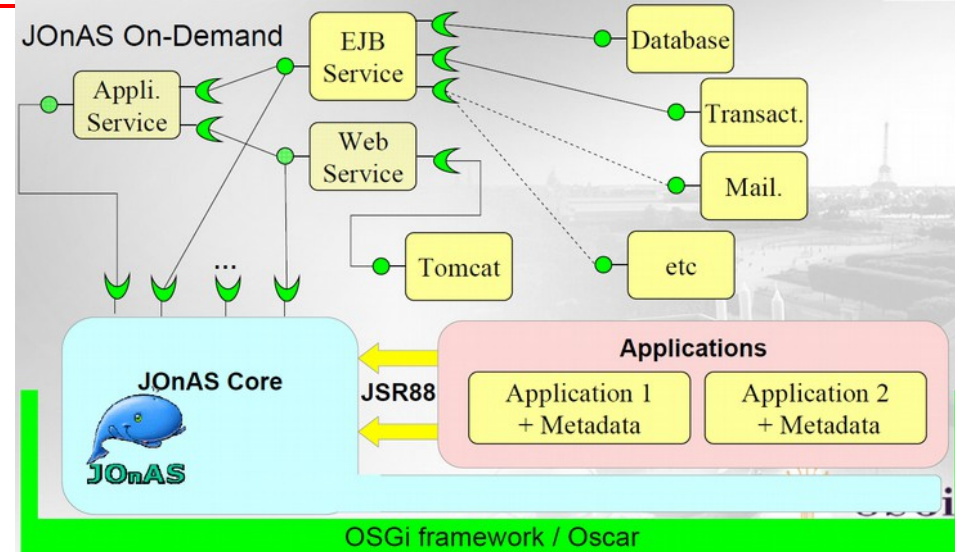
Le photocopieur en location



De plus en plus Server-side OSGi

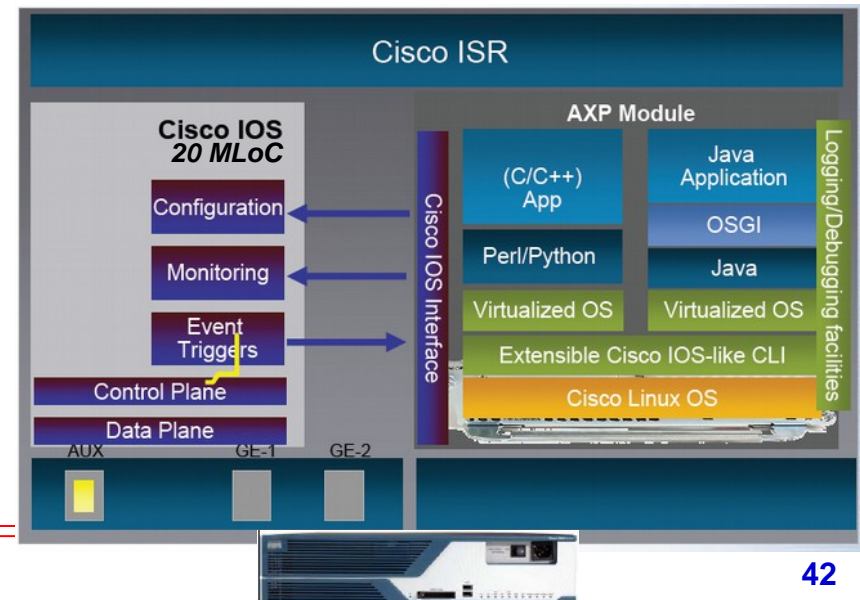
⑦ IT

- ⑦ OW2/Bull JOnAS 5, Oracle Weblogic, IBM Websphere, SUN Glassfish v3, JBoss ...
- ⑦ Apache DS, Sling, ...



⑦ Telco

- ⑦ Cisco' AXP
 - ⑦ Application eXtension Platform
- ⑦ Alcatel-Lucent IMS
- ⑦ ...



Le M2M (Machine-to-Machine)



OSGi

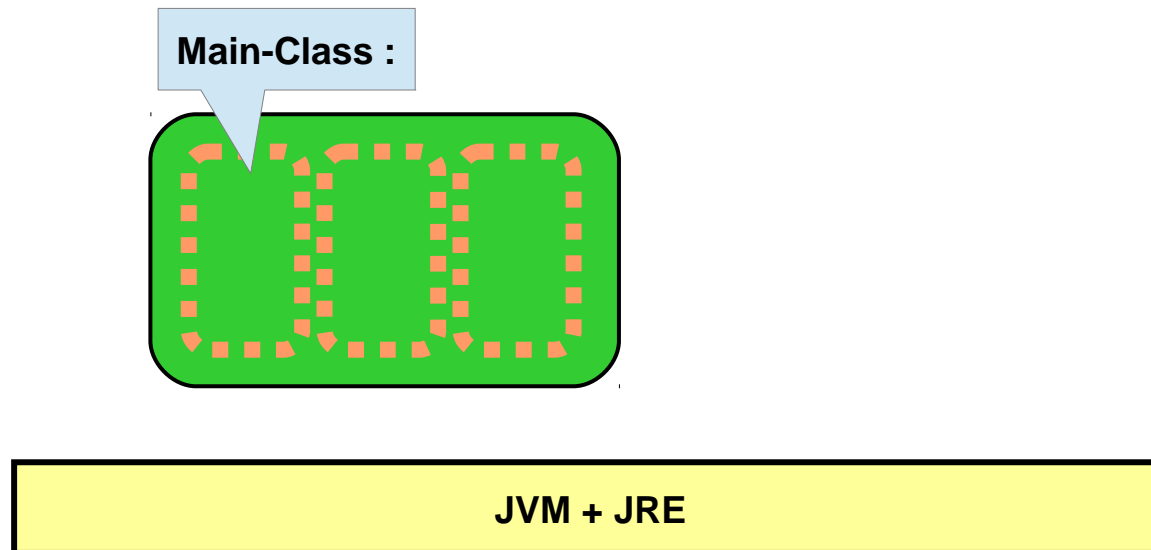
**Conditionnement, Déploiement
et Service**

Rappel :

Une application Java non modulaire

⑦ Ensemble de jarf les

- ⑦ placés statiquement dans le CLASSPATH ou \$JRE_HOME/lib/ext



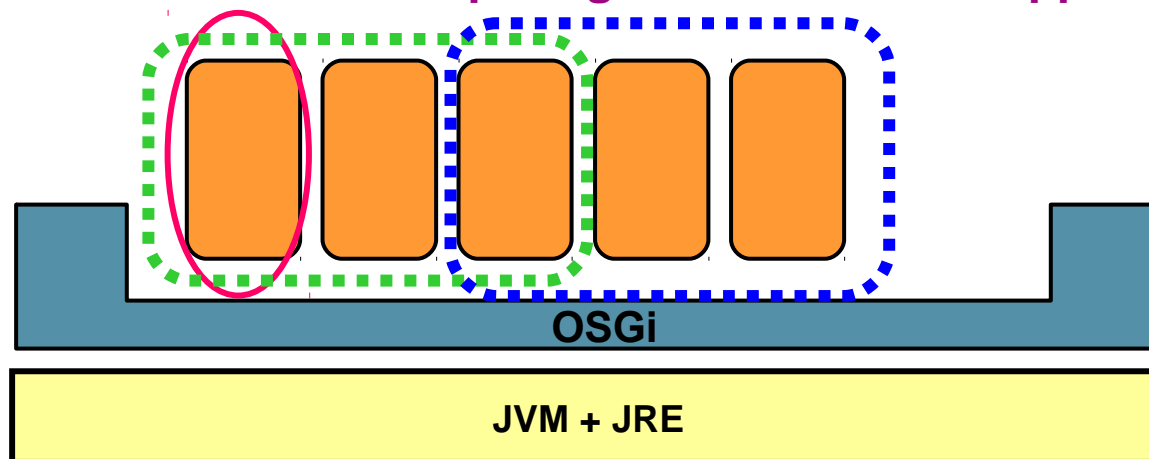
Bundle

⑦ Bundle

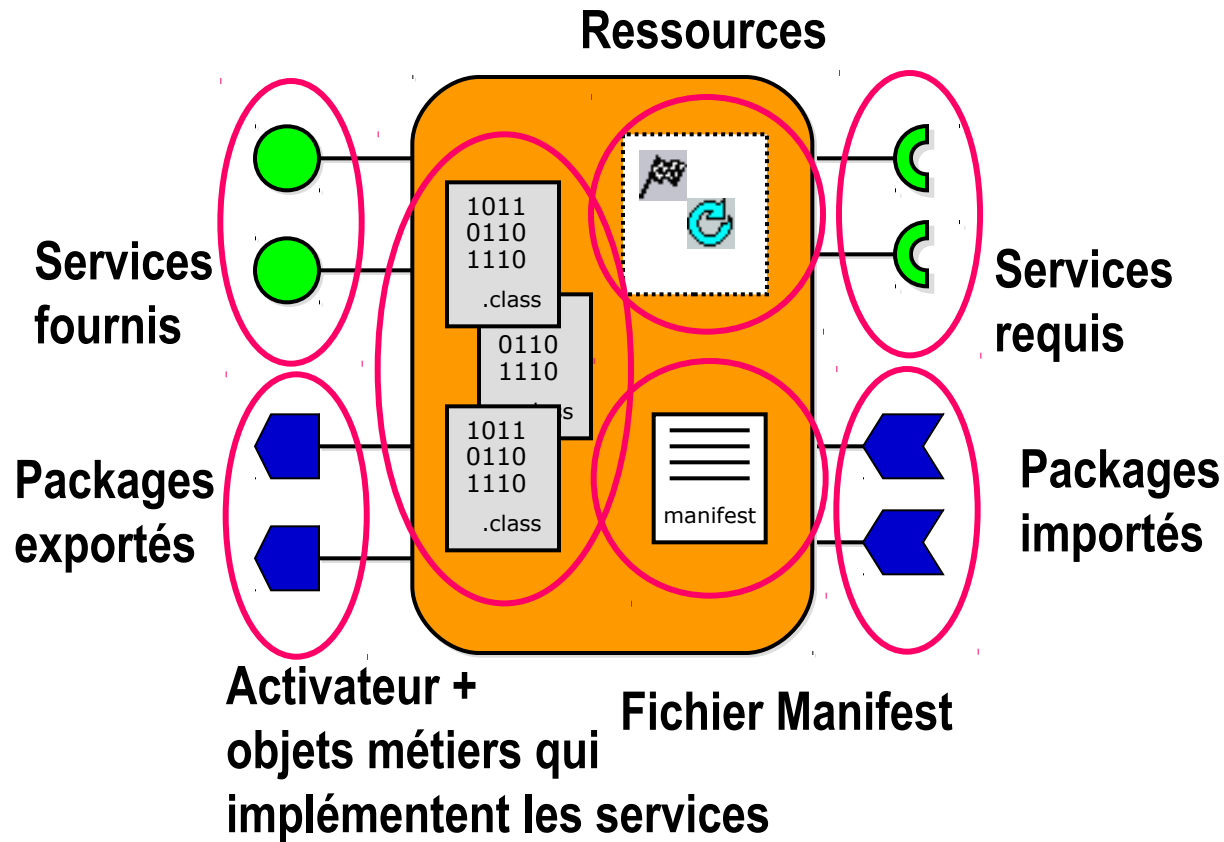
- ⑦ Unité de livraison et de déploiement sous forme de jar file
- ⑦ Unité fonctionnelle (offre des services)

⑦ Application

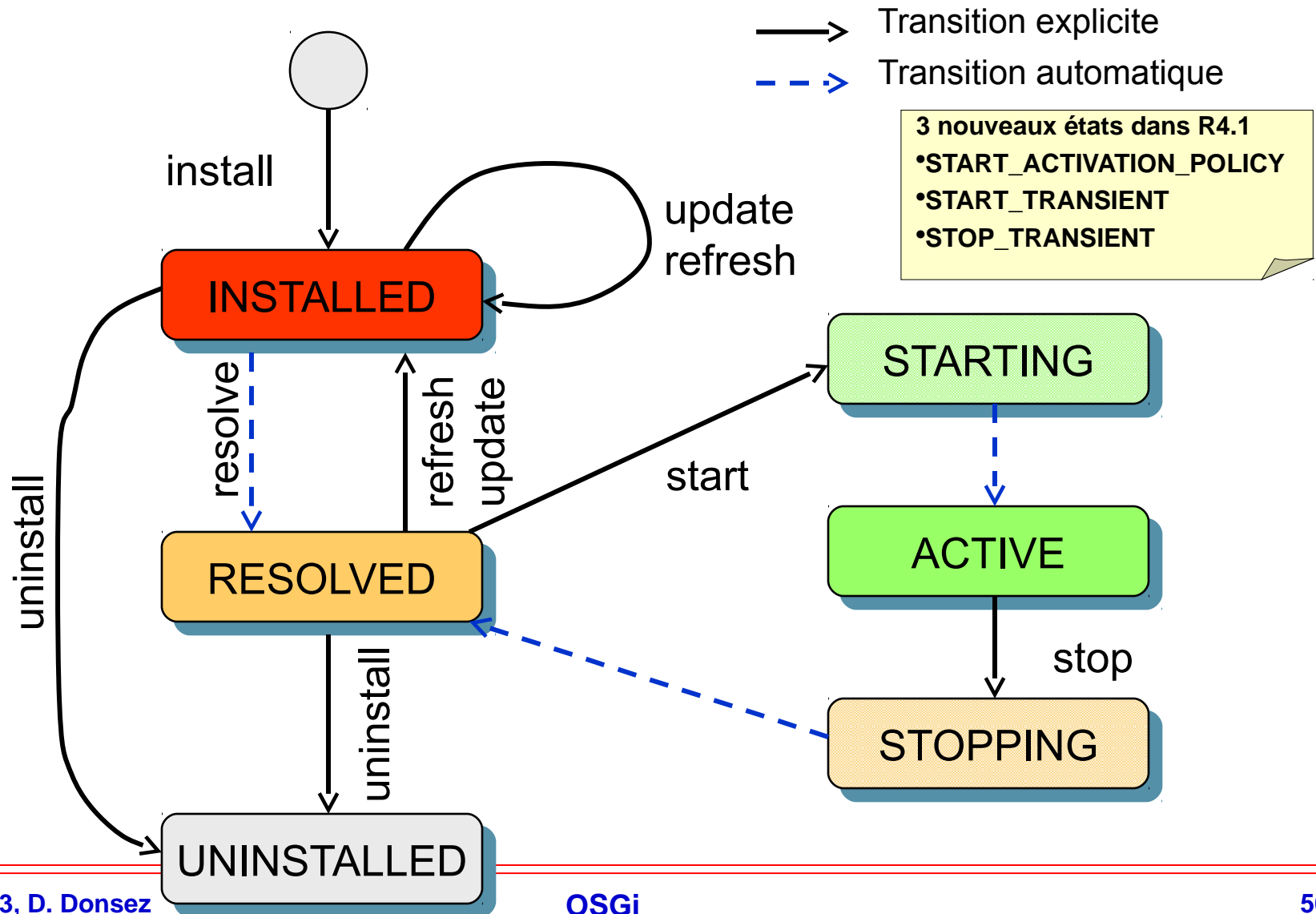
- ⑦ « Ensemble » de bundles
 - ⑦ livrés dynamiquement
 - ⑦ et éventuellement partagés avec d'autres applications



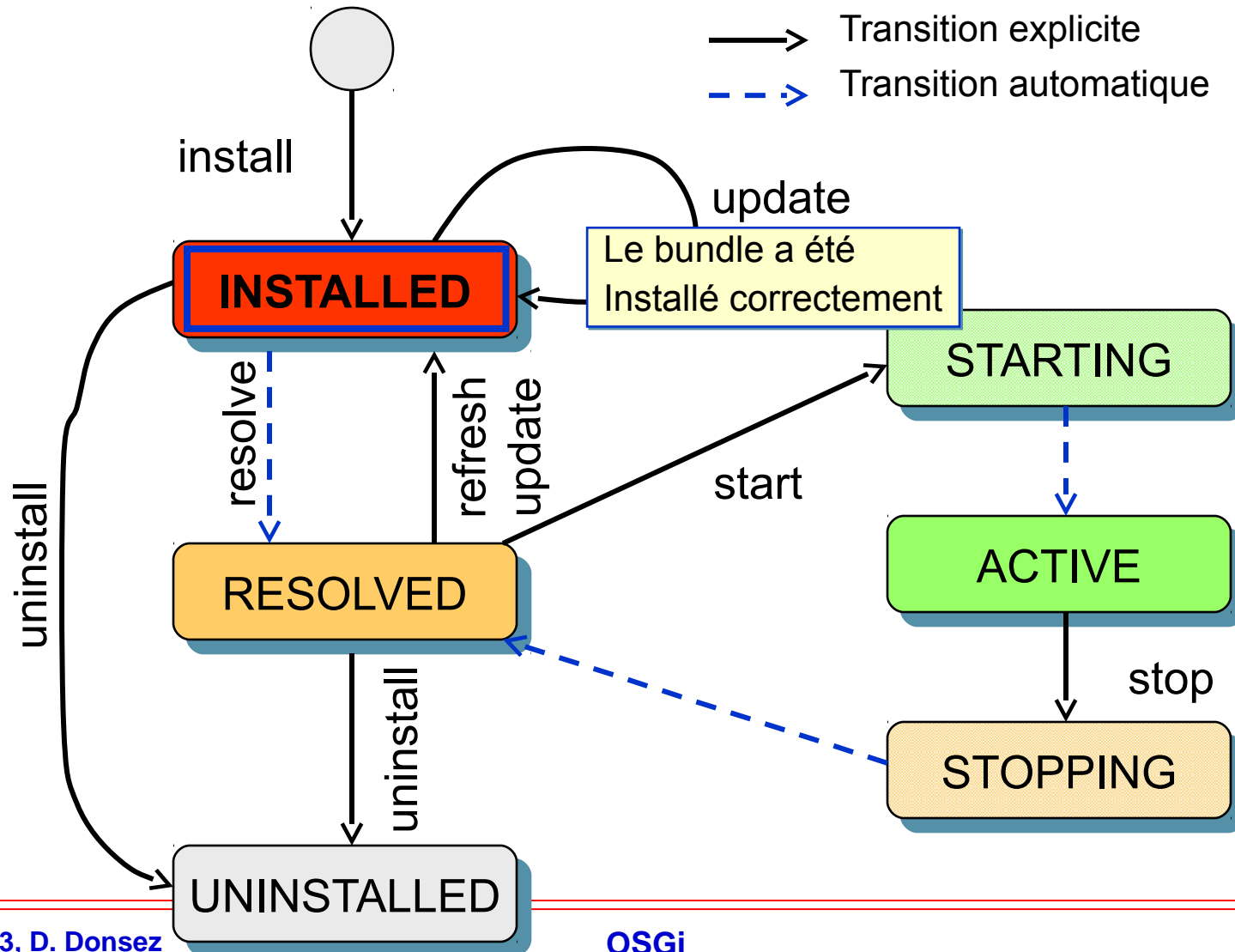
Structure d'un bundle



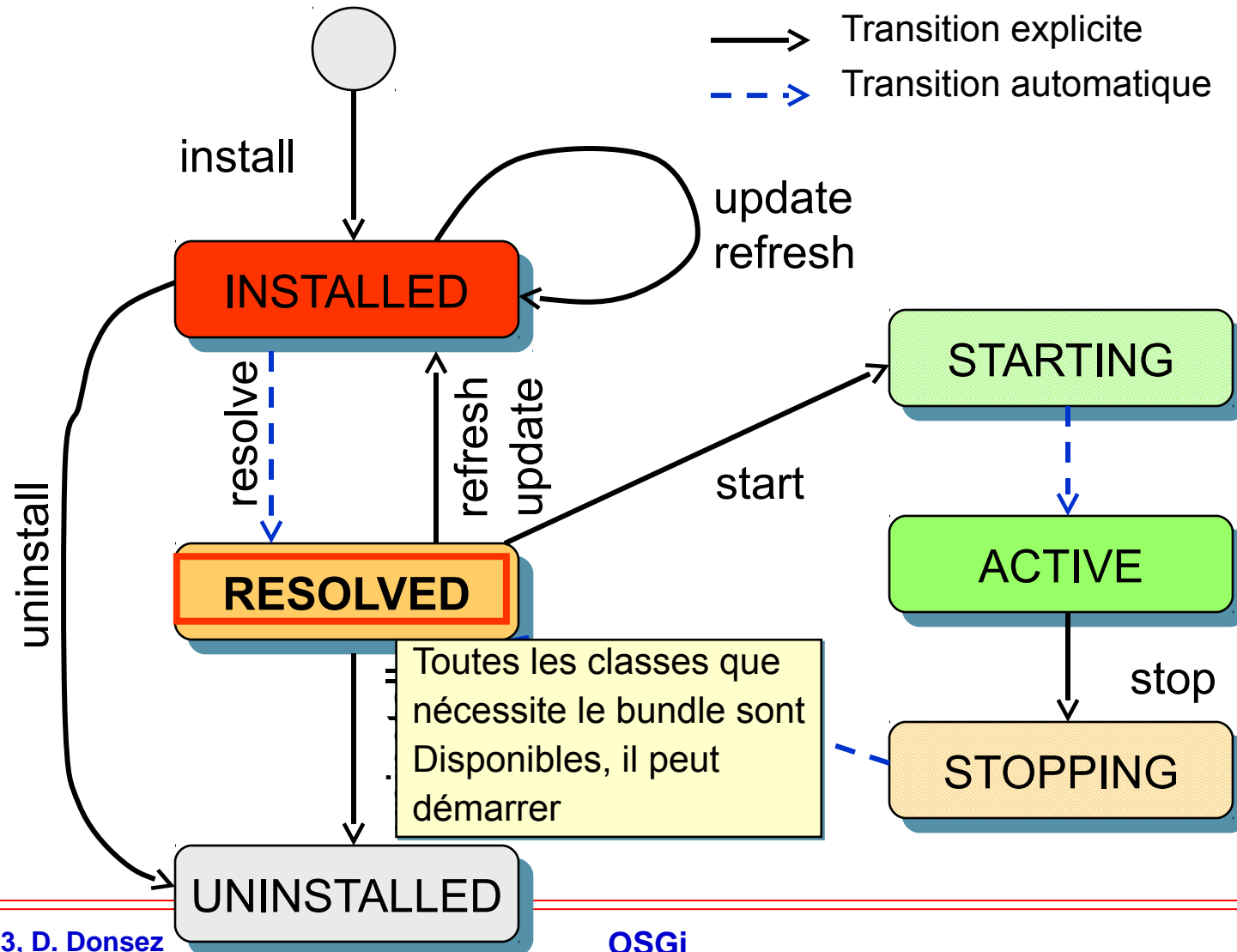
Cycle de vie d'un Bundle (R4)



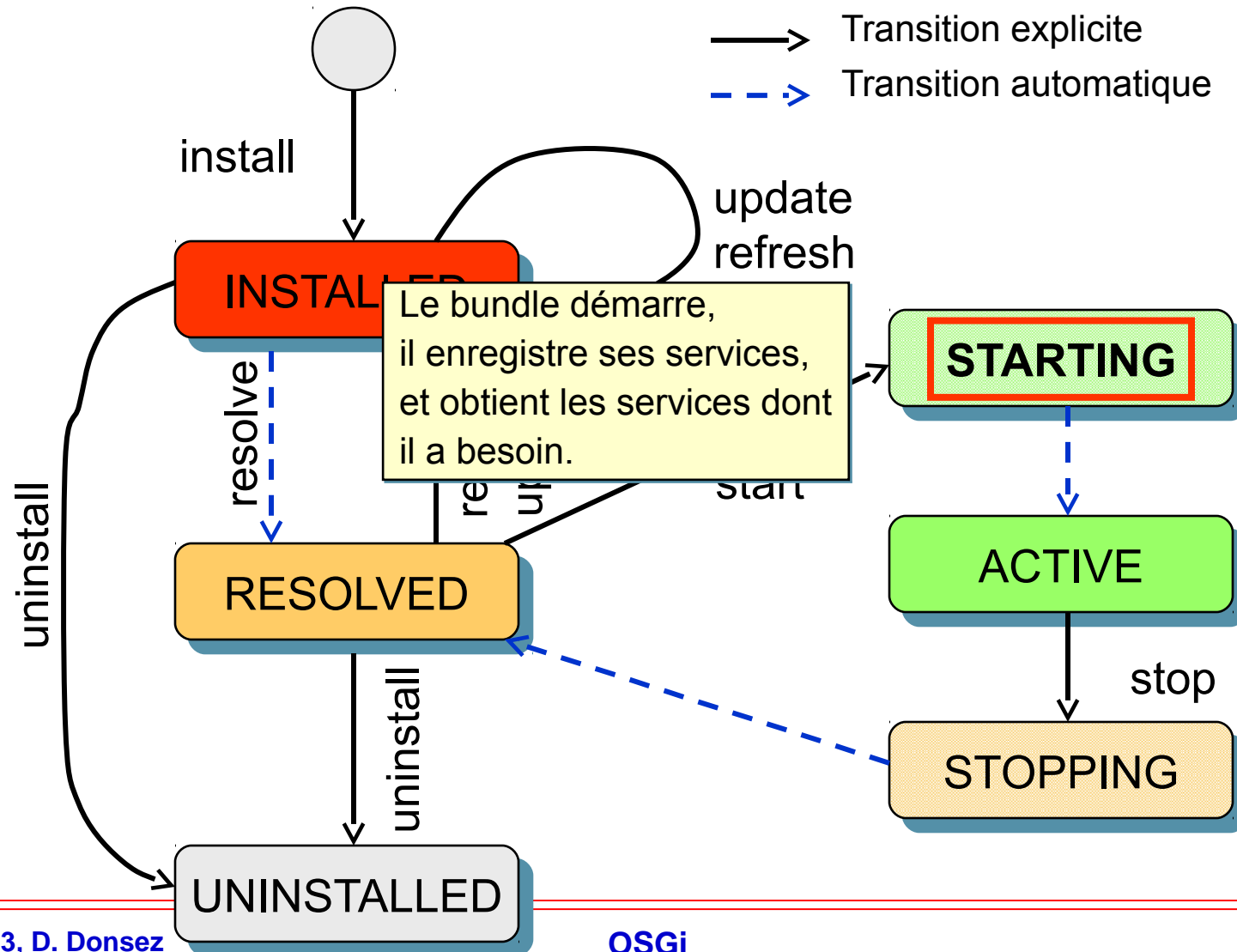
Cycle de vie d'un Bundle



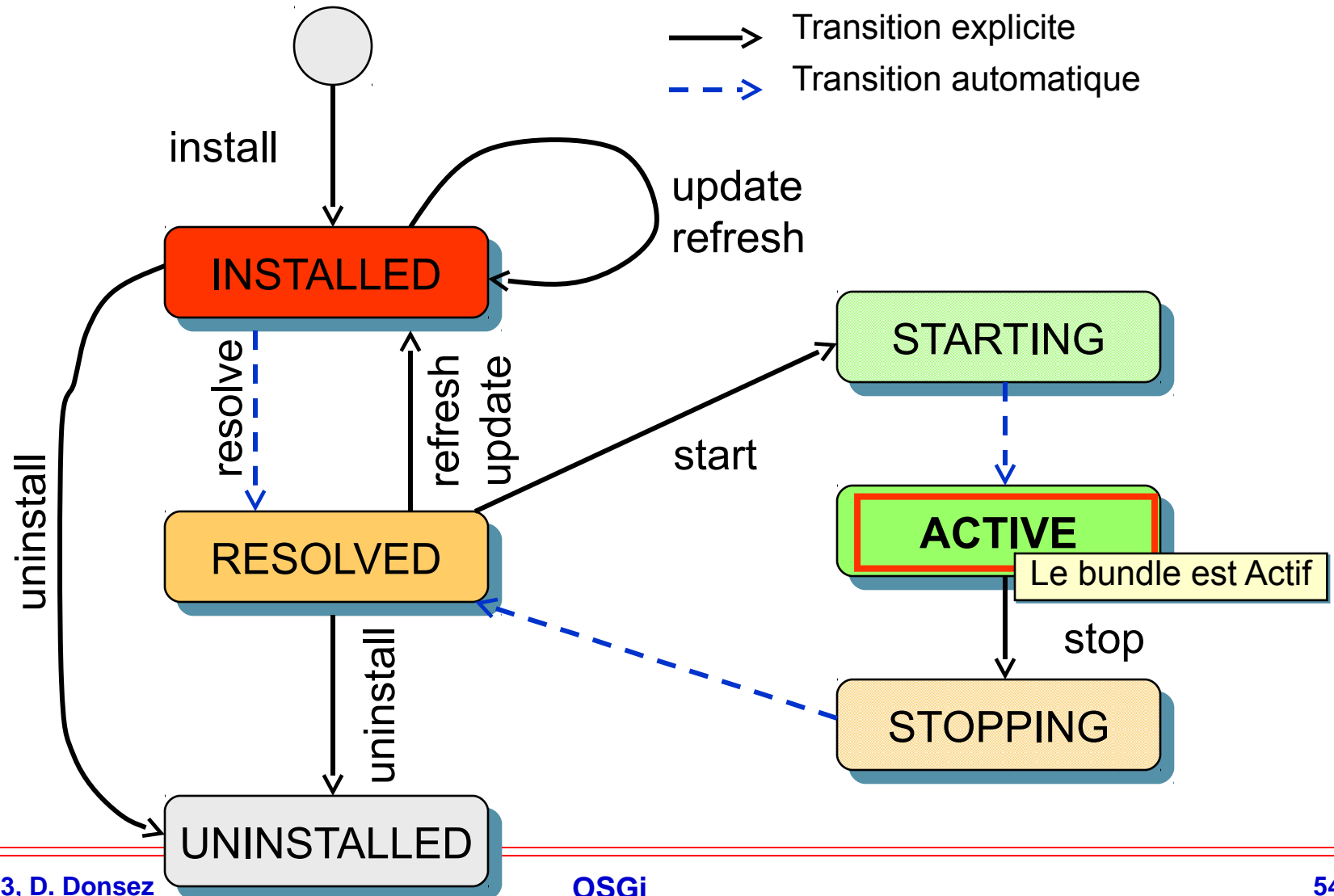
Cycle de vie d'un Bundle



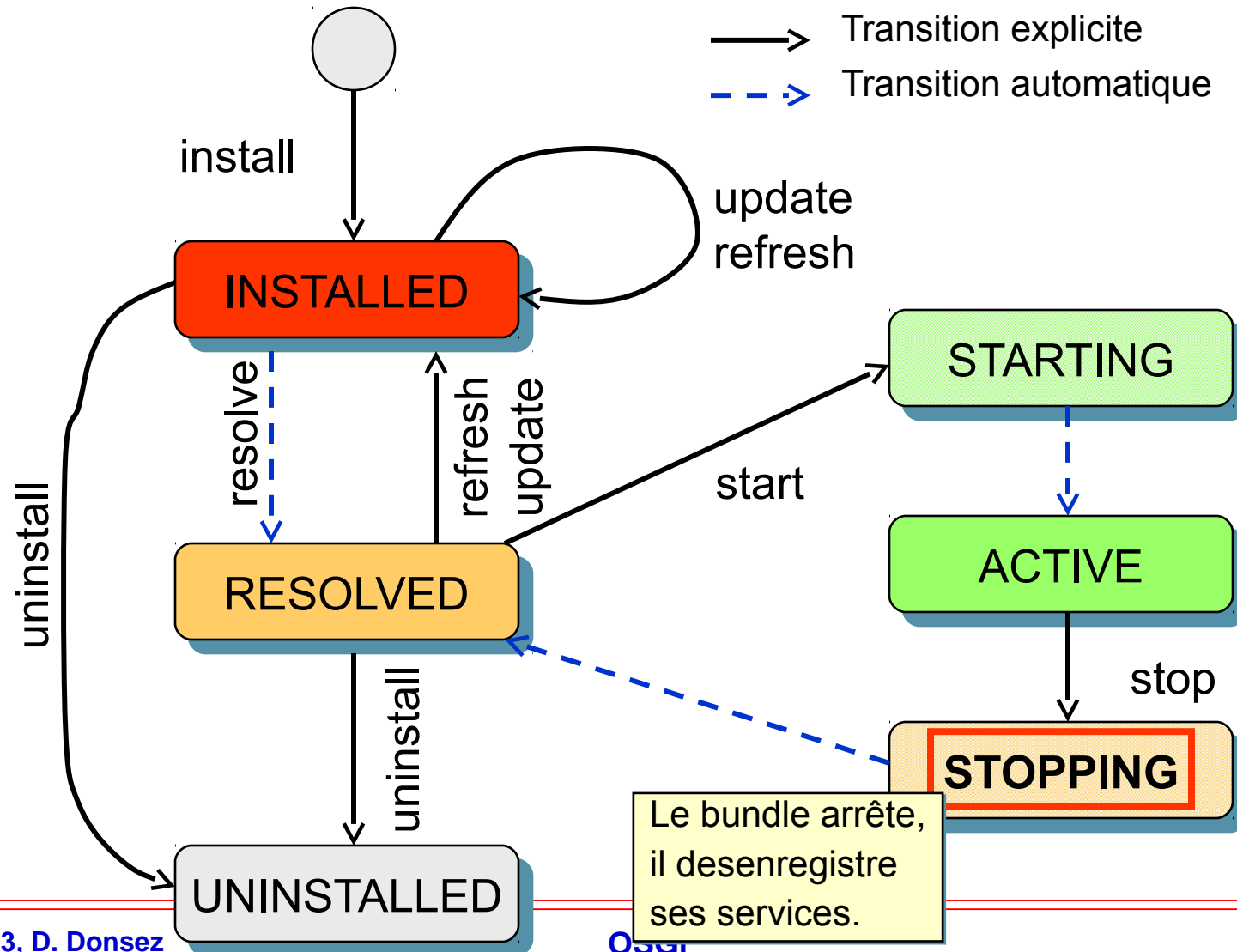
Cycle de vie d'un Bundle



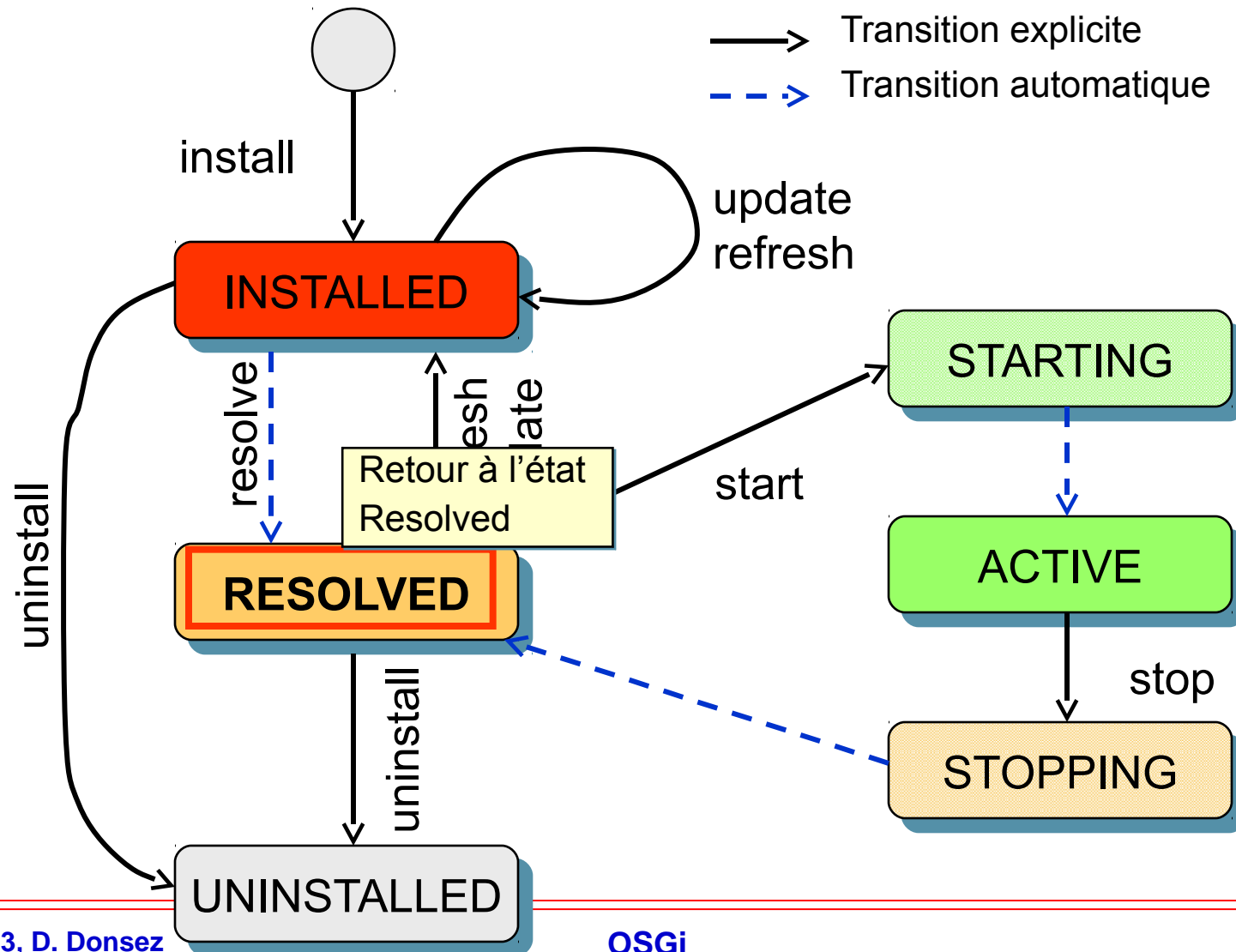
Cycle de vie d'un Bundle



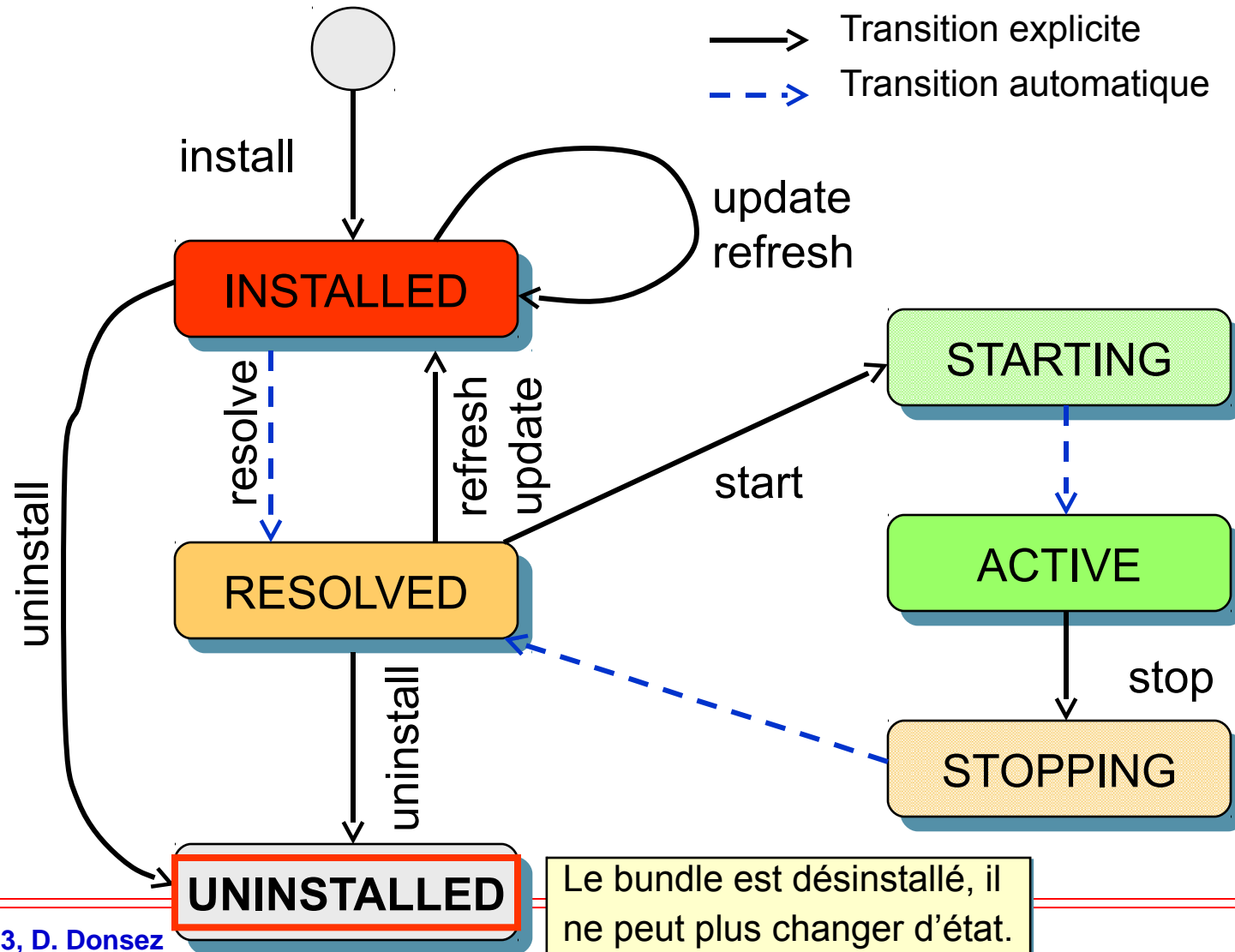
Cycle de vie d'un Bundle



Cycle de vie d'un Bundle



Cycle de vie d'un Bundle



News in R4.1

⑦ By default, persistent start and stop

⑦ Transient start and stop

⑦ `Bundle.start(int transientFlag)` and `Bundle.stop(int transientFlag)`

⑦ Flag sample: do not restart after FW restart (ie do no alter the autostart settin of the bundle)

⑦ By default, « Eager » activation

⑦ The `BundleActivator` is instanciated when `Bundle.start()`

⑦ Lazy Activation

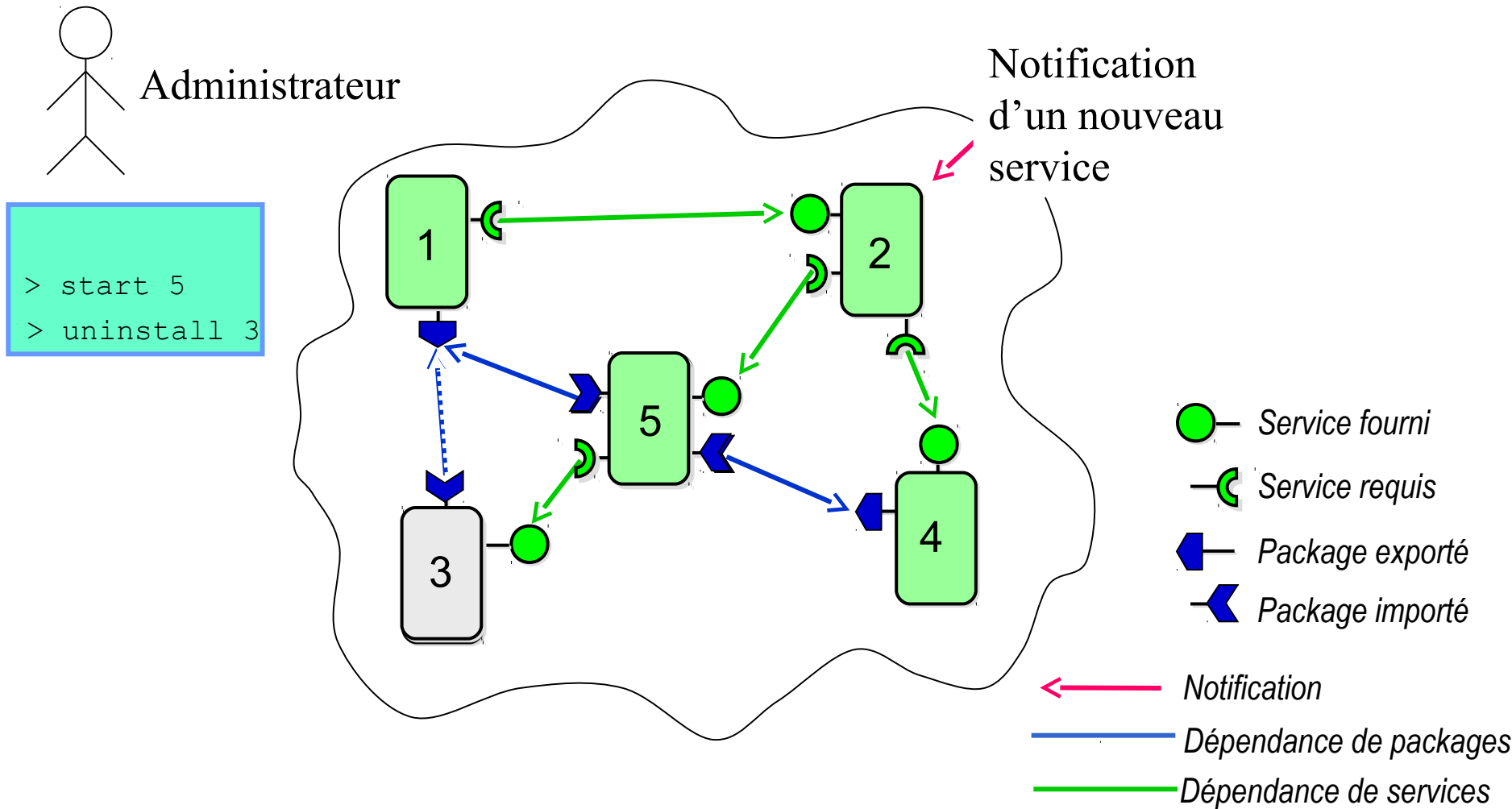
⑦ The bundle is activate when a exported class is required by an other bundles

→ new bundle event (lazy activated)
: useful for extender model

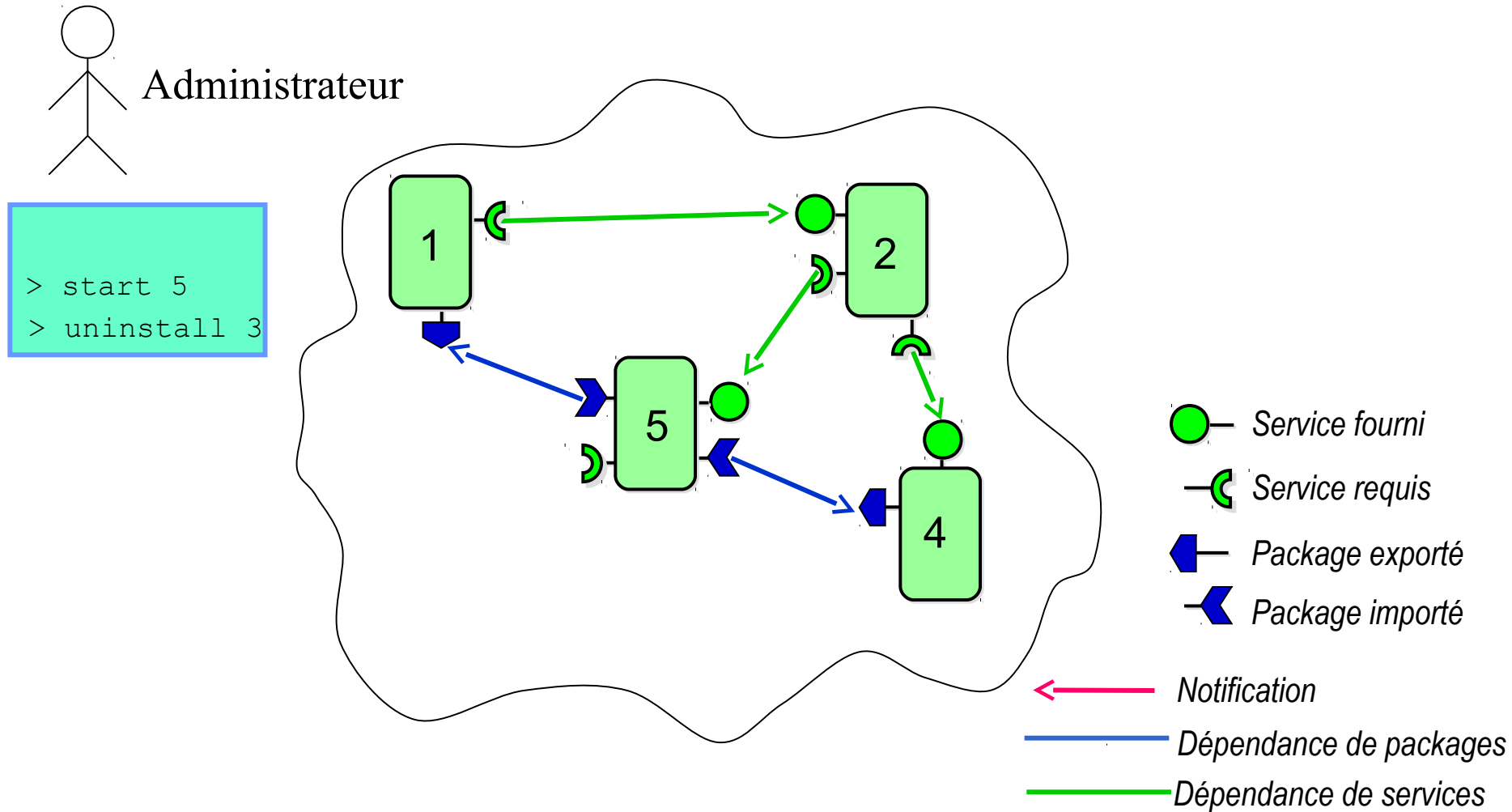
⑦ A lire

⑦ <http://www2.osgi.org/wiki/uploads/Conference/OSGi4.1Overview.pdf>

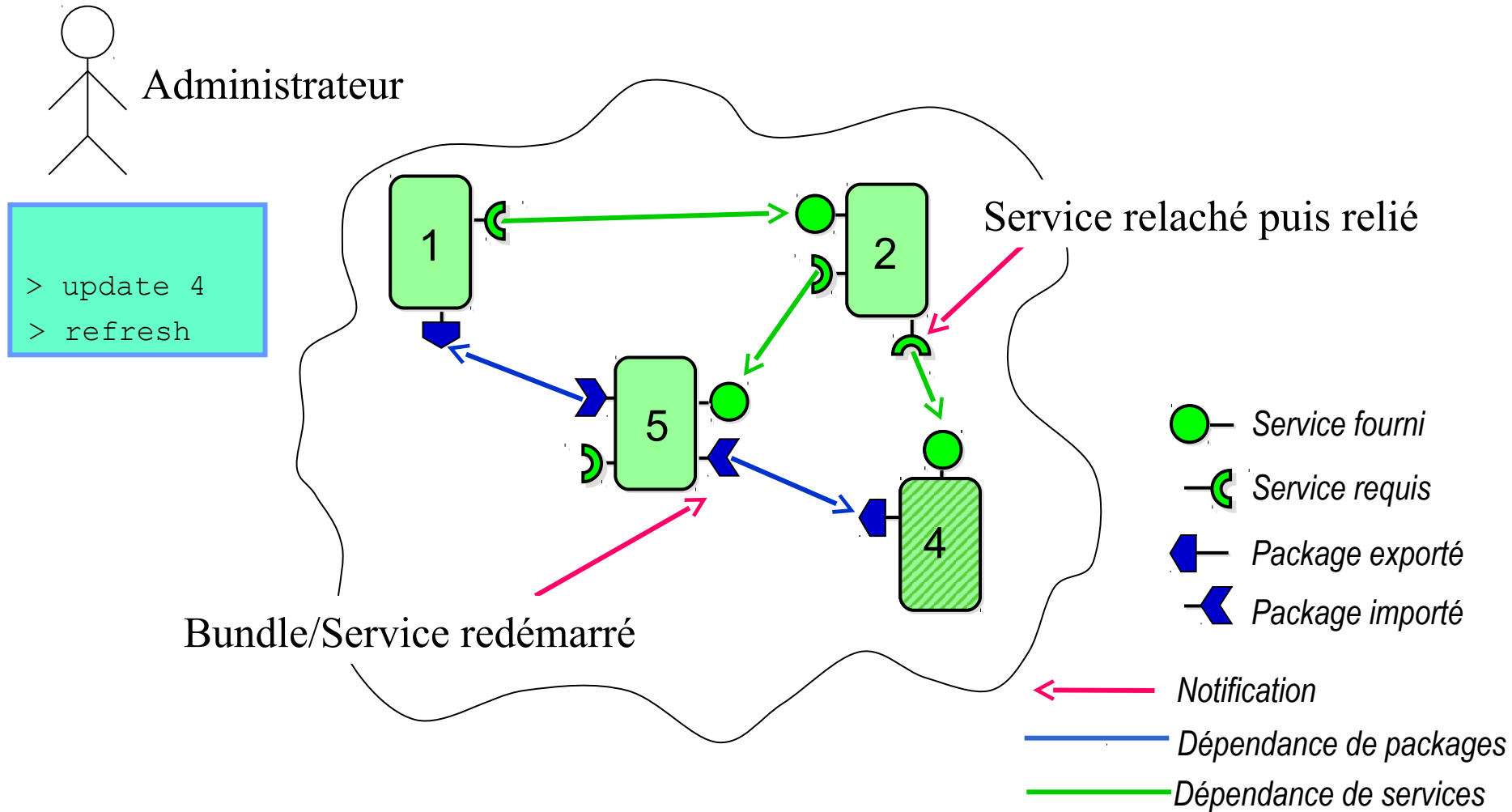
Dépendance & Dynamisme



Dépendance & Dynamisme



Dépendance & Dynamisme



Service

⑦ Une interface (ou plusieurs)

⑦ Des implémentations

- ⑦ multiples implémentations possibles conditionnées dans les bundles.
- ⑦ implémentation normalement non publique.
- ⑦ se trouvent dans des packages différents

⑦ Qualif é par des propriétés.



Classe d'implémentation
`com.hp.printer.laserjet.impl.Driver`

`org.device.print.type=laser`
`location=4th floor`



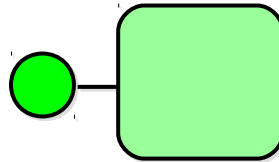
Classe d'implémentation
`com.lexmark.printer.laser.impl.Driver`

`org.device.print.type=laser`
`location=1st floor`

Exemple de service

Interface

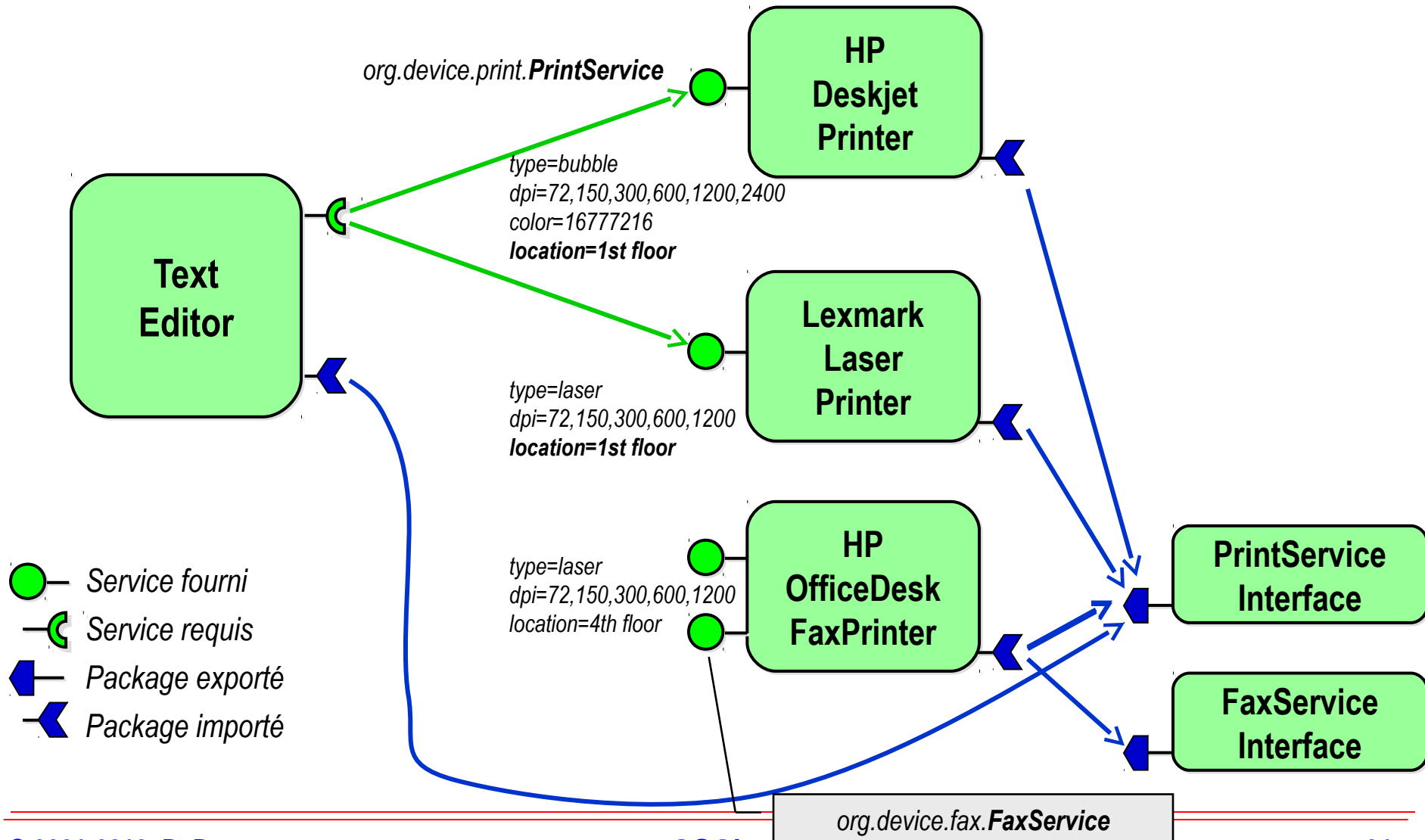
org.device.print.PrintService

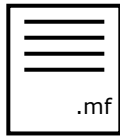


```
package org.device.print;
public interface PrintService {
    public static final String TYPE="org.device.print.type";
    public int print(OutputStream out,
                    String[] printparams)
        throws PrintException;
    public Job[] list()
        throws PrintException;
}
public interface Job { ... }

public class PrintException extends Exception { ... }
```

Exemple d'application





Fichier manifest (i)

⑦ Informations nécessaires au framework

Import-Package		Packages requis (avec/sans la version de spécification)	
Export-Package		Packages fournis (avec/sans la version de spécification)	
Import-Service		Services requis (indicatif, n'est pas utilisé par le FW)	
Export-Service		Services fournis (indicatif, n'est pas utilisé par le FW)	
Bundle-Activator		Nom de la classe Activator	
Bundle-ClassPath		Emplacement des classes et ressources du bundle	
Bundle-NativeCode		Bibliothèques natives à charger en fonction du processeur, du SE, ...	
Bundle-UpdateLocation		URL des mises à jour du bundle	



Fichier manifest (ii)

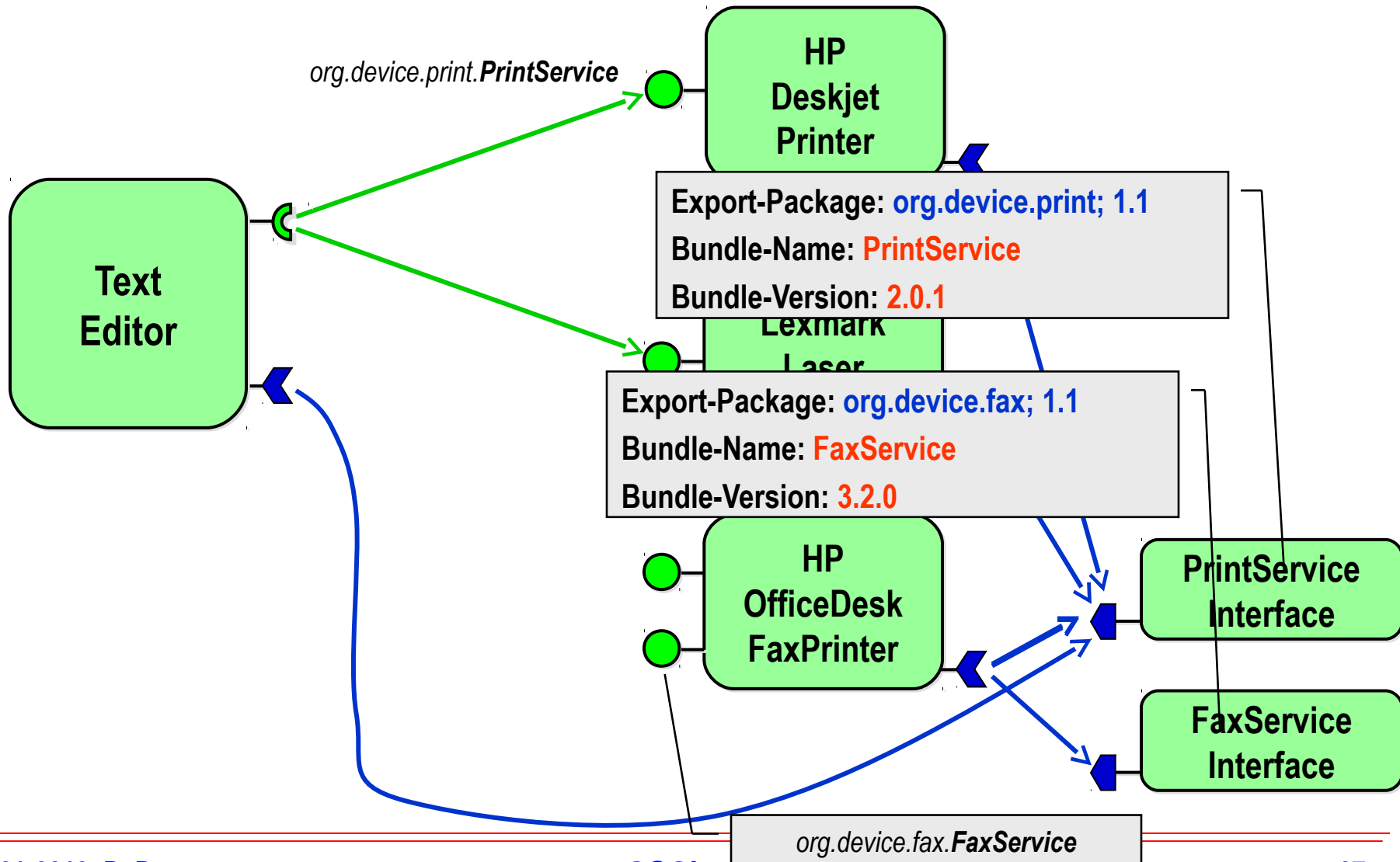


⑦ Informations nécessaires au framework

Bundle-SymbolicName	r4	Nom symbolique du bundle (sert à l'identification)
Bundle-Name		Nom du bundle
Bundle-Description		Description du bundle
Bundle-Version		Version du bundle
Bundle-DocURL		URL de la documentation du bundle
Bundle-ContactAddress		Coordonnée du propriétaire du bundle
Bundle-Category		Catégorie du bundle
Bundle-RequiredExecutionEnvironment	r3	Liste d'environnement qui doivent être présents sur la plateforme (exemple : CDC-1.0/Foundation-1.0, OSGi/Minimum-1.0)
DynamicImport-Package	r3	Liste de package qui pourront être importés en cours d'exécution (com.acme.plugin.*)

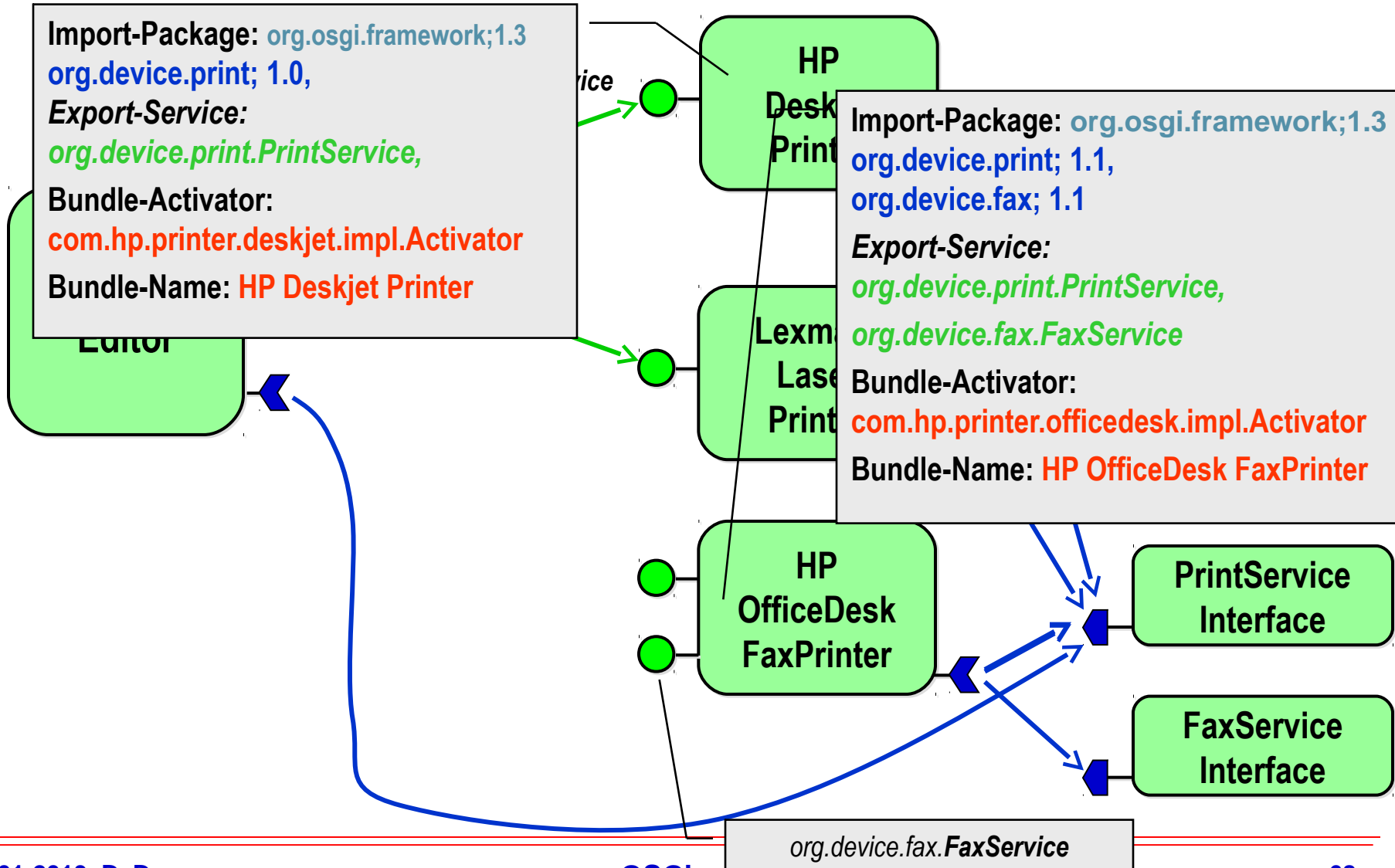


Exemple de manifest (i)



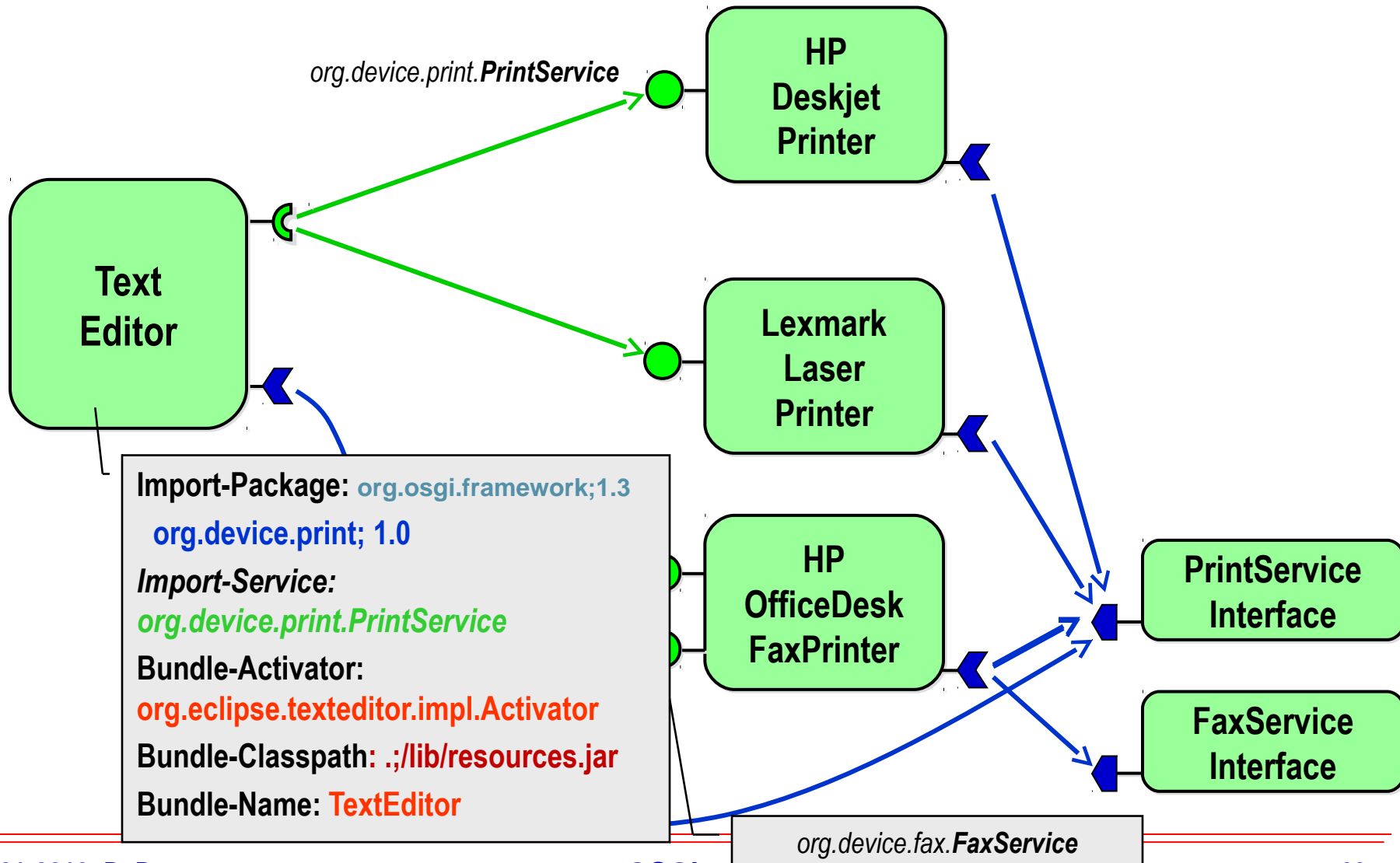


Exemple de manifest (ii)





Exemple de manifest (iii)



Chargement de classes (i)

⑦ 1 ClassLoader par Bundle

⑦ Chargement, Mise à Jour, Déchargement

⑦ Principe de la recherche des classes

⑦ La classe est dans le JRE

⑦ La classe est dans un package ni importé ni exporté

⑦ Utilisation de la classe chargée à partir du BUNDLE-CLASSPATH

⑦ La classe est dans un package importé

⑦ Utilisation de la classe chargée par le CL d'un autre bundle

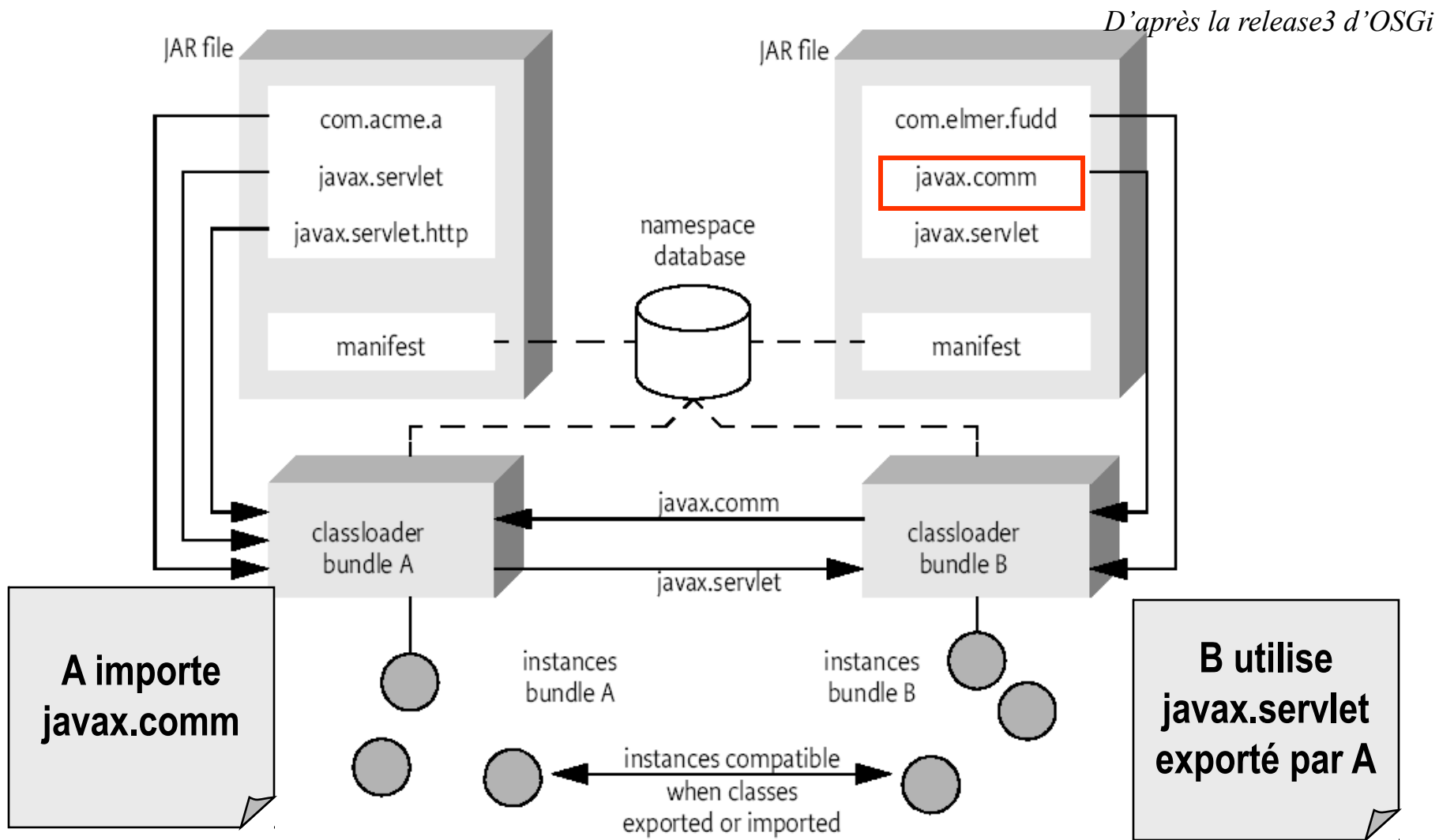
⑦ La classe est dans un package exporté mais déjà exporté par un autre bundle

⑦ Utilisation de la classe chargée par le CL de l'autre bundle

⑦ La classe est dans un package exporté mais non exporté par un autre

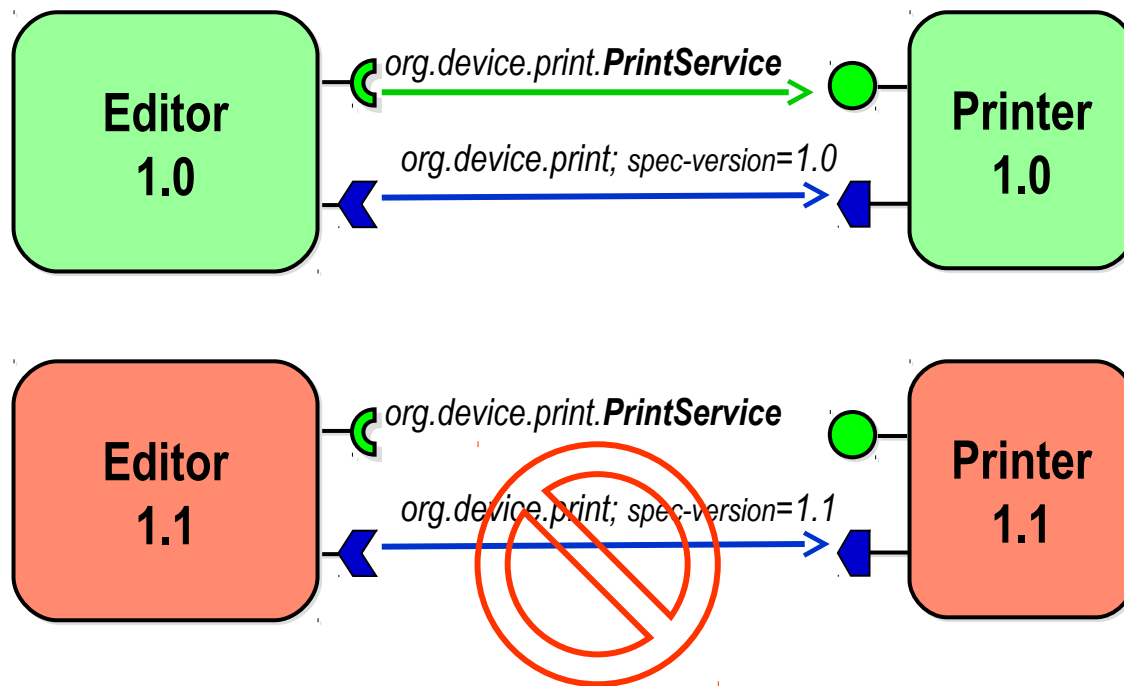
⑦ Utilisation de la classe chargée à partir du BUNDLE-CLASSPATH





Chargement de classes (ii)



Les limites r3

- ⑦ Pas d'activation tant que tous les imports ne sont pas résolus
- ⑦ Un service package actif à la fois
- ⑦ Compatibilité ascendance à assurer *Ad vitam eternam*



-  Service fourni
-  Service requis
-  Package exporté
-  Package importé

Les avancés

r3 R3

- ⑦ Importation dynamique

r4 R4

- ⑦ Bundle fragment
- ⑦ Bundle requis
- ⑦ Bundle extension
- ⑦ Intervalle de version, Politiques sur les versions
- ⑦ Importation et Exportation conditionnelles (attribut et filtre)
- ⑦ Activation simultanée de plusieurs version de packages

⑦ La suite : le JSR 277, JSR 294 ... Jigsaw

- ⑦ Richard S. Hall, “Java modularity, OSGi, and JSRs 277, 291, and 294”, ApacheCon EU 2006



<http://docs.safehaus.org/download/attachments/2995/osgi-apachecon-20060628.pdf>

- ⑦ <http://www.osgi.org/blog/2008/12/project-jigsaw.html>

DynamicImport-Package

- ⑦ **Permet en cours d'exécution d'importer des packages non présents à la résolution**
 - ⑦ `activator.getClass().getClassLoader().loadClass(clazzname)`
 - ⑦ *Surtout pas `Class.forName(clazzname)`*

- ⑦ **L'entrée `DynamicImport-Package` du manifeste liste les packages qui pourront être importés en cours d'exécution**

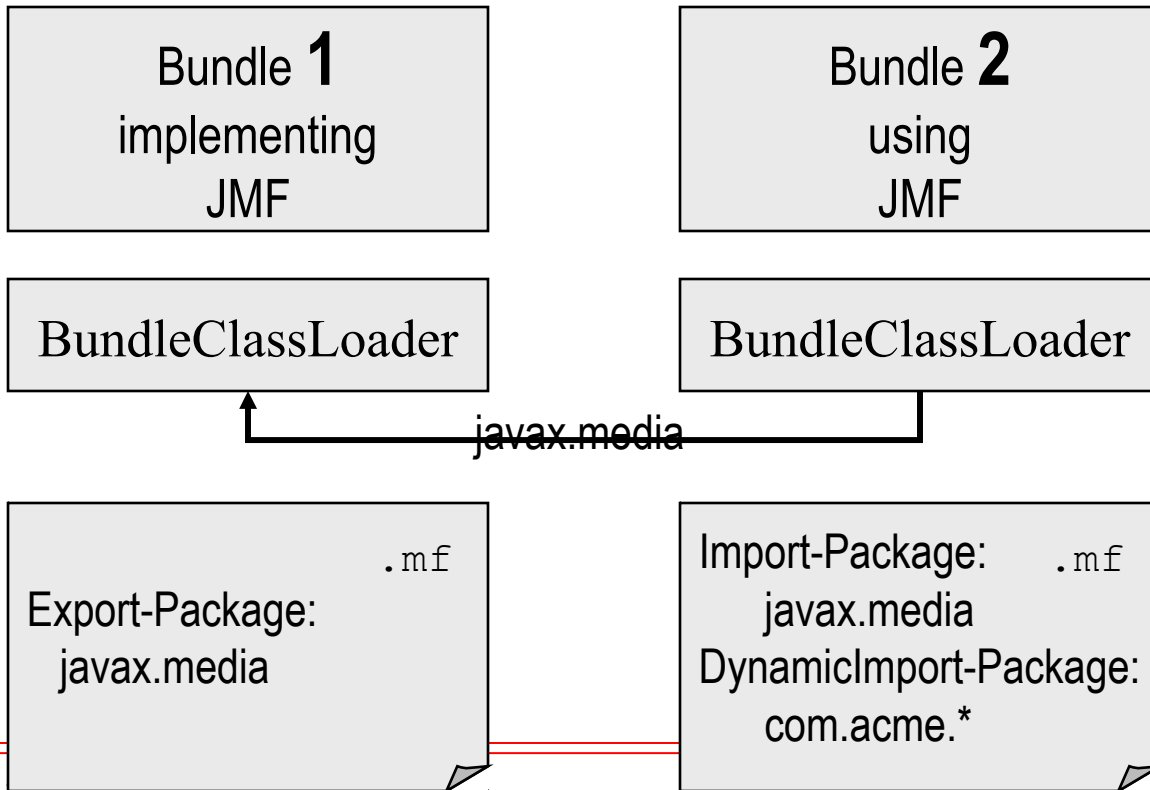
- ⑦ **Usage : framework à plugin ou service provider**
 - ⑦ **Exemple : JMF, JCE, JDBC, ...**

DynamicImport-Package Exemple avec JMF (i)

⑦ Installation de bundle 2

⑦ il passe à l'état **ACTIVE**

`myClassLoader.loadClass("com.acme.mp3.MP3Decoder")`
throws `java.lang.NoClassDefFoundError`

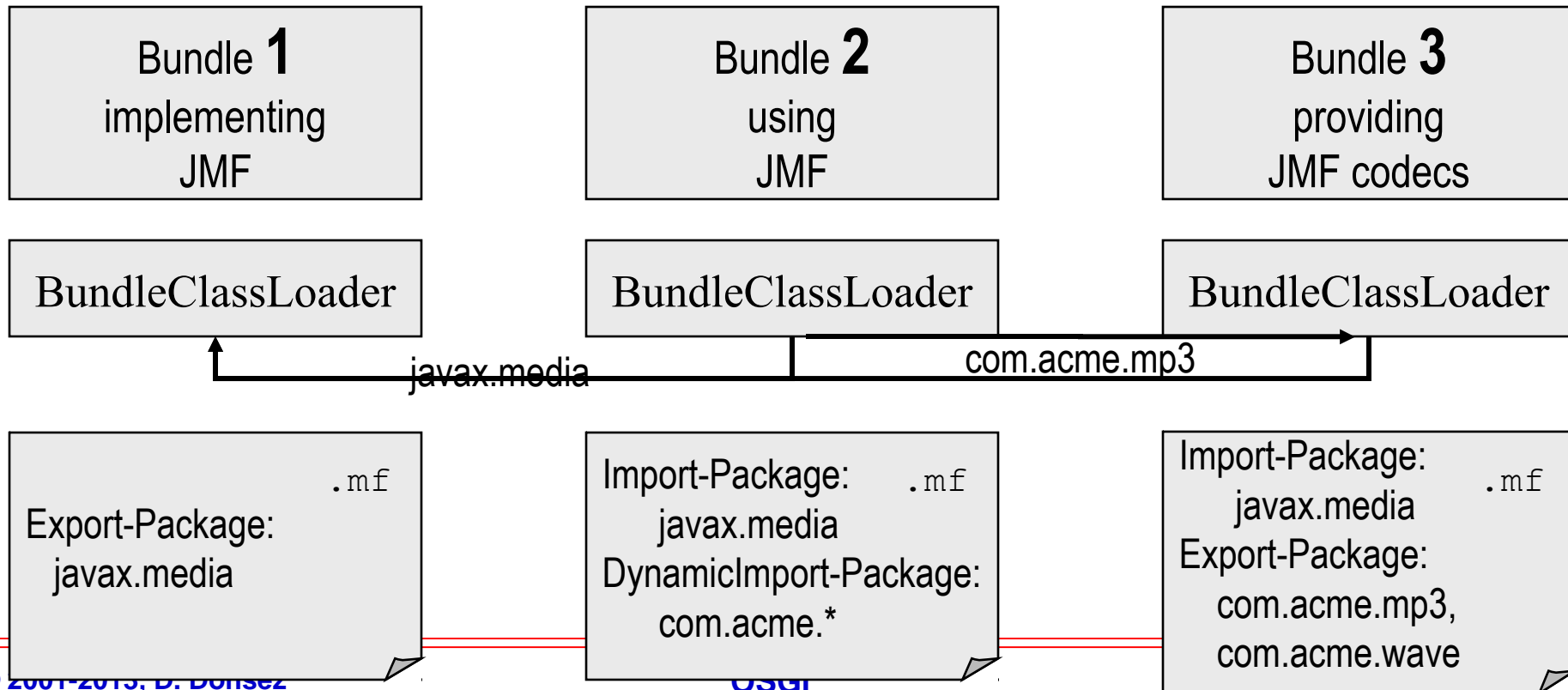


DynamicImport-Package Exemple avec JMF (ii)

R3

⑦ Installation de bundle 3 : il passe à l'état ACTIVE

myClassLoader.loadClass("com.acme.mp3.MP3Decoder")
add a package dependency
then return the class



⑦ Motivation

- ⑦ La compatibilité descendante (*backward compatibility*) est lourde à maintenir (surtout en embarqué).
- ⑦ Les @deprecated disparaissent parfois lors des chargements des majeures de versions

⑦ Idées

- ⑦ Intervalle de version
- ⑦ Import-Package:
javax.servlet; version="[2.0.0,2.4.0)"; resolution="optional"

⑦ Autres

- ⑦ Export-Package:
org.foo.service; version=1.1; vendor="org.foo",
org.foo.service.bar; version=1.1; uses="org.foo.service",
org.foo.service.fzz; include="*Impl"; exclude="*Test"



- ⑦ Richard S. Hall, Java Modularity Support in OSGi R4, ApacheCon (San Diego), December 14th, 2005
 - ⑦ <http://docs.safehaus.org/download/attachments/2995/osgi-apachecon-20051214.pdf>

Accès aux ressources et aux fichiers

⑦ Ressources

⑦ `this.getClass().getResourceAsStream(String path)`

`path="/"` correspond à la racine du JAR (BUNDLE-CLASSPATH)

⑦ Support de persistance

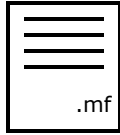
⑦ `BundleContext.getDataFile(String path)`

`path=""` correspond à la racine du cache du bundle

⑦ `FileService`

accès au système de fichiers local (s'il existe !)

et contrôle les permissions du Bundle au passage



Bundle-Classpath

⑦ **Représente (dans le manifeste) les chemins (dans le JAR) de recherche des classes et des ressources**

⑦ **3 cas**

⑦ Bundle-Classpath: . ou Pas de Bundle-Classpath

⑦ **Recherche dans le JAR**

⑦ Bundle-Classpath: .;demo/nested.jar;test/nest.jar

⑦ **Recherche dans le JAR puis dans le JAR inclus**

⑦ Bundle-Classpath: demo/nested.jar

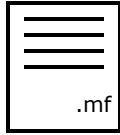
⑦ **Recherche dans le JAR inclus**

⑦ **Aucune classe ou ressource n'est recherchée dans le JAR**

⑦ **Intérêt des JAR inclus**

⑦ **Conservation des signatures, manifestes, ...**

⑦ **Possibilité de *patcher* un sous ensemble des ressources/classes !**



Bibliothèques natives

⑦ Bibliothèques de fonctions natives (C) dépendantes du processeur et de l'OS

⑦ Exemple : Pilotes matériel (javax.comm), Patrimonial (codec), ...

⑦ Bundle-NativeCode dans le MANIFEST

⑦ Spécifie l'emplacement des bibliothèques dépendantes du système et du processeur, à charger dynamiquement (par le ClassLoader)

⑦ Exemple

⑦ **Bundle-NativeCode:** com/mycomp/impl/nativesample/libnat.so;
osname=Solaris; processor=sparc; osversion=5.5,
com/mycomp/impl/nativesample/libnat.so;
osname=SunOS; processor=sparc; osversion=2.5,
com/mycomp/impl/nativesample/nat.dll;
osname=Windows NT; processor=x86; osversion=4.0

⑦ Remarque : Propriétés du framework

⑦ org.osgi.framework.processor, org.osgi.framework.language, org.osgi.framework.os.name,
org.osgi.framework.os.version

La classe d'activation du bundle

⑦ Classe publique

- ⑦ Implémente les 2 méthodes `start()` et `stop()` de `BundleActivator`
- ⑦ qui reçoivent une référence sur un contexte.

⑦ *start(BundleContext ctxt)*

- ⑦ recherche et obtient des services requis auprès du contexte et/ou positionne des listeners sur des événements
- ⑦ enregistre les services fournis auprès du contexte

⑦ *stop(BundleContext ctxt)*

- ⑦ désenregistre les services fournis
- ⑦ relâche les services requis

⑦ **Cependant le FW fait ces opérations si `stop()` est oublié !**



il peut ne pas y avoir de `BundleActivator` dans un bundle

- ⑦ Livraison de classes et ressources
- ⑦ Eclipse extension points
- ⑦ **Extend model**

BundleContext

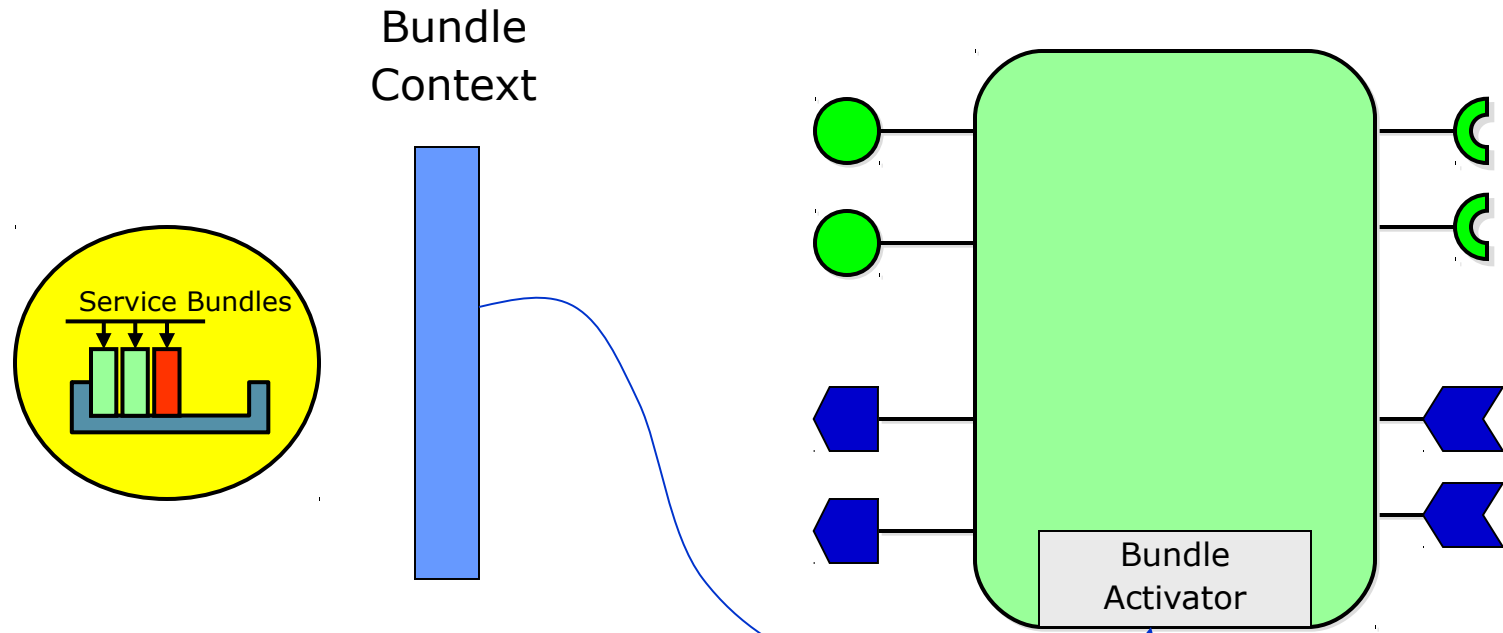
⑦ Interface vers le framework

- ⑦ Passé lors des invocations de start() et stop() de l'Activator

⑦ Permet

- ⑦ L'enregistrement de services
- ⑦ Le courtage de services
- ⑦ L'obtention et la libération des services
- ⑦ La souscription aux évènements du Framework.
- ⑦ L'accès aux ressources du bundle
- ⑦ *L'accès aux propriétés du framework*
- ⑦ *L'installation de nouveaux bundles*
- ⑦ *L'accès à la liste des bundles*

BundleContext et Activator

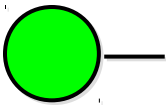


- registerService()
- getServiceReferences()
- getService()
- getDataFile()
- addServiceListener()
- addBundleListener()
- addFrameworkListener()

- start(BundleContext bc)
- stop(BundleContext bc)
- serviceChanged()

Enregistrement de services (Lexmark Laser Printer)

```
package com.lexmark.printer.laser.impl;
public class Activator implements BundleActivator {
    private ServiceRegistration reg=null;
    private PrintService theService=null;
    public void start(BundleContext ctxt) throws BundleException {
        theService=new PrintServiceImpl();
        Properties props=new Properties();
        props.put("type", "laser");
        props.put("dpi", "72,150,300,600,1200");
        props.put("location", "1st floor");
        reg=ctxt.registerService(
            "org.device.print.PrintService", theService, props);
    }
    public void stop(BundleContext ctxt) throws BundleException {
        if(reg != null) reg.unregister();
    }
}
```



Recherche de services (TextEditor)

```
package org.eclipse.texteditor.impl
import org.device.print.PrintService;
class Activator implements BundleActivator {
    public void start(BundleContext ctxt) throws BundleException {
        private PrintService ser;
        // On va voir si quelqu'un offre un PrintService ...
        ServiceReference[] tempRefs
            =ctxt.getServiceReferences
                ("org.device.print.PrintService", "(location=1st floor)");
        if(tempRefs!=null) {
            System.out.println("Found a PrintService! I will use it!!!");
            // On prend le premier offert!
            ser=(PrintService) ctxt.getService(tempRefs[0]);
        }
        ...
    }
    ...
}
```

Recherche (Courtage) de services

- ⑦ **Filtrage par des expressions de condition LDAP (RFC1960) sur les propriétés enregistrées par les services**
- ⑦ **Expressions de filtrage**
 - ⑦ Expressions simples (attribut opérateur valeur)
 - ⑦ Valeurs de type `String`, `Numerique`, `Character`, `Boolean`, `Vector`, `Array`
 - ⑦ Attribut insensible aux majuscules/minuscules
 - ⑦ L'attribut `objectClass` représente le nom du service
 - ⑦ Opérateurs `>=`, `<=`, `=`, `~=` (approximativement égal), `=*` (présent)
 - ⑦ Connecteurs logiques `&`, `|`, `!`

Recherche de services

⑦ Tous les services d'impression

```
refs=bundleContext.getServiceReferences("org.device.print.PrintService", null);  
refs=bundleContext.getServiceReferences(null,  
    "(objectClass=org.device.print.PrintService)");
```

⑦ Certains services d'impression

```
refs=bundleContext.getServiceReferences("org.device.print.PrintService",  
    "&(!(type=laser))(capability=double-sided)(!(dpi<=300))(location=*)" );
```

⑦ Tous les services de org.device

```
refs=bundleContext.getServiceReferences(null,"(objectClass=org.device.*)");
```

⑦ Le service d'impression et de fax au 3ième étage

```
refs=bundleContext.getServiceReferences(null,  
    "&(objectClass=org.device.print.PrintService)(objectClass=org.device.fax.FaxService)"  
    + "(location=4th floor)");
```

Comparaison avec le courtage de JINI

⑦ JINI

- ⑦ Typage fort sur le nom de l'interface et la signature de ces méthodes
- ⑦ Sous-typage des interfaces
- ⑦ Propriétés de courtage

- ⑦ Distribué
- ⑦ Notion de groupe
- ⑦ Bail (lease) d'un enregistrement

⑦ OSGi

- ⑦ Typage sur le nom de l'interface
- ⑦ **Non**
- ⑦ **Oui**

- ⑦ **Centralisé (même JVM)**
- ⑦ **Non**
- ⑦ **Non**

Événements dans le Framework

⑦ FrameworkEvent

⑦ Notifie le démarrage et les erreurs du Framework

⑦ interface `FrameworkListener` méthode `frameworkEvent`

Traitement séquentiel et asynchrone des listeners (par event dispatcher)

⑦ BundleEvent

⑦ Notifie les changements dans le cycle de vie des bundles

⑦ interface `BundleListener` méthode `bundleChanged`

Traitement séquentiel et asynchrone des listeners (par event dispatcher)

⑦ interface `SynchronousBundleListener` méthode `bundleChanged`

Traitement séquentiel et synchrone des listeners (avant le traitement du changement d'état)

⑦ ServiceEvent

⑦ Notifie l'enregistrement ou le retrait de services

⑦ interface `ServiceListener` méthode `serviceChanged`

Traitement séquentiel et synchrone des listeners



Service Registry Hooks (RFC 126 R4.2)

org.osgi.framework.hooks

⑦ Hooks on the service registry operations

⑦ PublishHook

- ⑦ Bundles registering this service will be called during framework service publish (register, modify, and unregister service) operations. This method is called prior to service event delivery when a publishing bundle registers, modifies or unregisters a service and can filter the bundles which receive the event.

⑦ FindHook

- ⑦ Bundles registering this service will be called during framework service find (get service references) operations. This method is called during the service find operation by the finding bundle and can filter the result of the find operation.

⑦ ListenerHook

- ⑦ Bundles registering this service will be called during service listener addition and removal. The hook is notified of the collection of service listeners and what they may be listening for and well as future changes to that collection.

Prendre en compte l'enregistrement et le retrait de service (i)

⑦ Les bundles « requesters » doivent **impérativement** prendre en compte l'enregistrement et le retrait de services « importés »

⑦ Exemple

```
public class PrintListenerActivator implements BundleActivator {  
    PrintServiceListener listener = null;  
    public void start(BundleContext context) {  
        PrintServiceListener listener = new PrintServiceListener(context);  
        context.addServiceListener(listener);  
    }  
    public void stop(BundleContext context) {  
        context.removeServiceListener(listener);  
    }  
}
```

Prendre en compte l'enregistrement et le retrait de service (ii)

⑦ Exemple simpliste et inutile

```
class PrintServiceListener implements ServiceListener {
    public void serviceChanged(ServiceEvent e) {
        ServiceReference ref = e.getServiceReference();
        if((((String)ref.getProperty("objectClass").equals("org.device.print.PrintService"))){
            switch (e.getType()) {
                case ServiceEvent.REGISTERED:
                    println(ref + " has been registered by "+ ref.getBundle().getLocation()); break;
                case ServiceEvent.UNREGISTERING:
                    println(ref + " is being unregistered"); break;
                case ServiceEvent.MODIFIED:
                    println("properties of "+ref+" have been modified:");
                    String[] keys = ref.getPropertyKeys();
                    for (int i=0; i<keys.length; i++) println(keys[i] + "=" + ref.getProperty(keys[i])); break;
            }
        }
        void println(String msg) {System.out.println("events: "+msg); }
    }
}
```

Ajout de
ServiceEvent.MODIFIED_ENDMATCH
en R4.2

Prendre en compte l'enregistrement et le retrait de service (iii)

⑦ Exemple 2 :

```
public class Activator implements BundleActivator {
    final static String filterStr
        ="(&(objectClass=org.device.print.PrintService)(location=4th floor))";
    Map/*<ServiceReference,PrintService>*/ printservices;
    BundleContext context;
    public void start(BundleContext context) throws BundleException {
        this.context=context;
        printservices=new HashMap();
        BindingController ctrl=new BindingController(context,filterStr,printservices);
        ctrl.open();
        context.addServiceListener(ctrl);
    }
}
```

Prendre en compte l'enregistrement et le retrait de service (iv)

```
public class BindingController implements ServiceListener {
    Map/*<ServiceReference,Object>*/ services;
    String filterStr;
    Filter filter;
    BundleContext context;

    public BindingController(BundleContext context, String filterStr, Map services)
    {
        this.context=context;
        this.filterStr=filterStr;
        this.services=services;
        filter=context.createFilter(filterStr);
    }
}
```

Prendre en compte l'enregistrement et le retrait de service (v)

```
...
public void open() {
    // fill the services map
    ServiceReference[] refs=context.getServiceReferences(null,filterStr);
    for(int i=0;i<refs.length;i++){
        Object svc = context.getService(refs[i]);
        if(svc!=null) services.put(refs[i],svc);
    }
}
public void close() {
    // release the references to service
    ...
}
...
```

Prendre en compte l'enregistrement et le retrait de service (vi)

...

```
public void serviceChanged(ServiceEvent e) {  
    ServiceReference servref = e.getServiceReference();  
    Object ref;  
    switch (e.getType()) {  
    case ServiceEvent.REGISTERED:  
        if(filter.match(servref)){  
            println(servref + " (from "+ servref.getBundle().getLocation() + ") is added");  
            services.put(servref,context.getService(servref));  
        };  
        break;  
    }  
}
```

...

Prendre en compte l'enregistrement et le retrait de service (vii)

....

case ServiceEvent.UNREGISTERING:

ref=services.remove(servref);

if(ref!=null) {

println(servref + " is removed");

context.ungetService(servref);

} break;

case ServiceEvent.MODIFIED:

ref=services.get(servref);

if(ref!=null && !filter.match(servref)){

println(servref + " is removed since properties has changed");

services.remove(servref);

context.ungetService(servref);

} break; }}

Prendre en compte l'enregistrement et le retrait de service (viii)

⑦ Le `BindingController` est incomplet

- ⑦ Pas de synchronisation

- ⑦ Les services apparus et disparus entre le `getServiceReferences` et le `addServiceListener`

⑦ Mini conclusion

- ⑦ Vous avez suivi ?

- ⑦ Et maintenant avec 6 services dont 3 **obligatoires**

Prendre en compte l'enregistrement et le retrait de service (viii)

⑦ Mini conclusion

- ⑦ Vous avez suivi ?

⑦ Solutions

- ⑦ La classe utilitaire ServiceTracker (OSGi **R2**)
 - ⑦ Ne gère pas le cycle de vie
- ⑦ Service Component Runtime (OSGi R4)
 - ⑦ ~ ADL pour cycle de vie et liaison
 - ⑦ Repris de ServiceBinder (Cervantes & Hall)

⑦ Motivation

- ⑦ Simplifier l'usage des ServiceListeners

⑦ ServiceTracker

- ⑦ Classe utilitaire pour suivre un type de service (interface + filtre LDAP)

⑦ ServiceTrackerCustomizer

- ⑦ Interface de rappel

⑦ Remarque

- ⑦ peut être fourni par le framework afin d'en optimiser l'implémentation

ServiceTracker : Exemple

⑦ Dans le start(BundleContext)

```
serviceTracker=new ServiceTracker(  
    bundleContext,  
    bundleContext.createFilter(  
        "&(objectClass=org.device.print.PrintService)(type=laser)"),  
    (ServiceTrackerCustomizer)null);  
serviceTracker.open();
```

⑦ Dans les méthodes du service

```
PrintService ps=(PrintService) serviceTracker.getService();  
ps.print(...); ...  
Job[] jobs=ps.list(); ...  
ps=null; // sinon risque de stale reference
```

⑦ Dans le stop(BundleContext)

```
serviceTracker.close();
```

ServiceTrackerCustomizer : Exemple

```
class MyServiceTrackerCustomizer implements ServiceTrackerCustomizer {

    public Object addingService( ServiceReference reference) {
        Object obj = context.getService(reference);
        HttpService servant = (HttpService)obj;
        // Register the Servlet using servant
        ...
        return servant;
    }

    public void removedService( ServiceReference reference, Object object ){
        HttpService servant = (HttpService)obj;
        // Unregister the Servlet using servant
        ...
        context.ungetService(reference);
    }

}
```

⑦ **Modèle simple de composants orienté service**

- ⑦ **Gère le cycle de vie du composant en fonction des dépendances obligatoires de services**

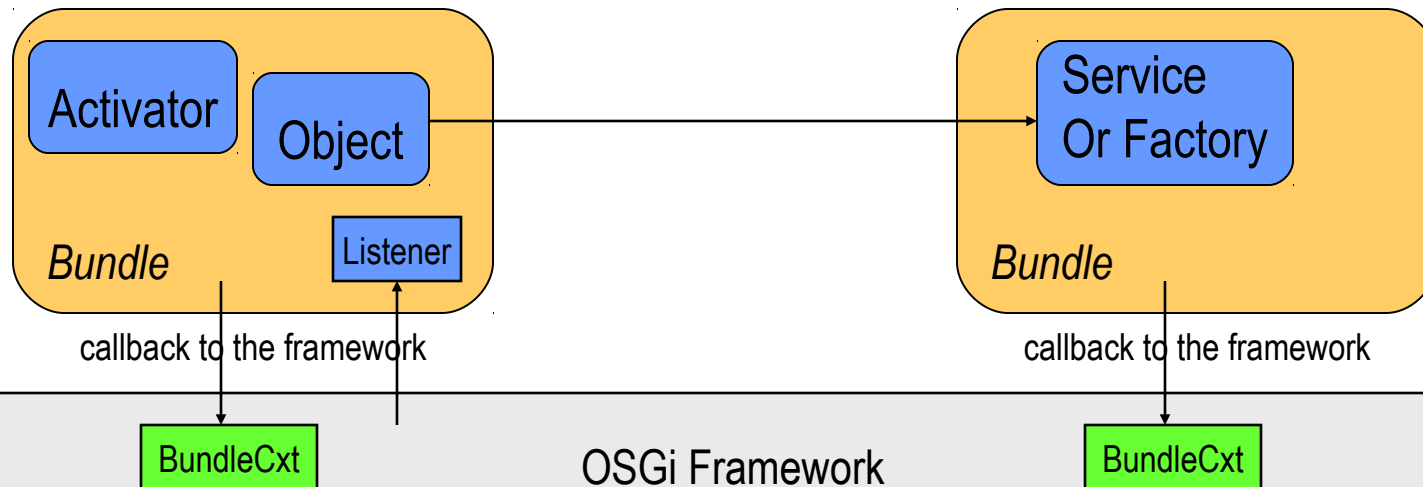
⑦ **Caractéristiques**

- ⑦ **Fabrique de Services**
- ⑦ **Activation retardée (*lazy-activation*)**
- ⑦ **Gestion des liaisons**
 - ⑦ **stratégie événementielle**
 - ⑦ **stratégie de recherche (via le contexte)**
- ⑦ **Entrée du manifeste : Service-Component**
- ⑦ **Descripteur XML**
 - ⑦ **xmlns:scr="http://www.osgi.org/xmlns/scr/v1.0.0"**
- ⑦ **...**

Service Component Runtime

⑦ Programmation SOC Dynamique

- ⑦ Démarrage des instances de services
- ⑦ Listeners pour gérer la liaison dynamique vers les services requis

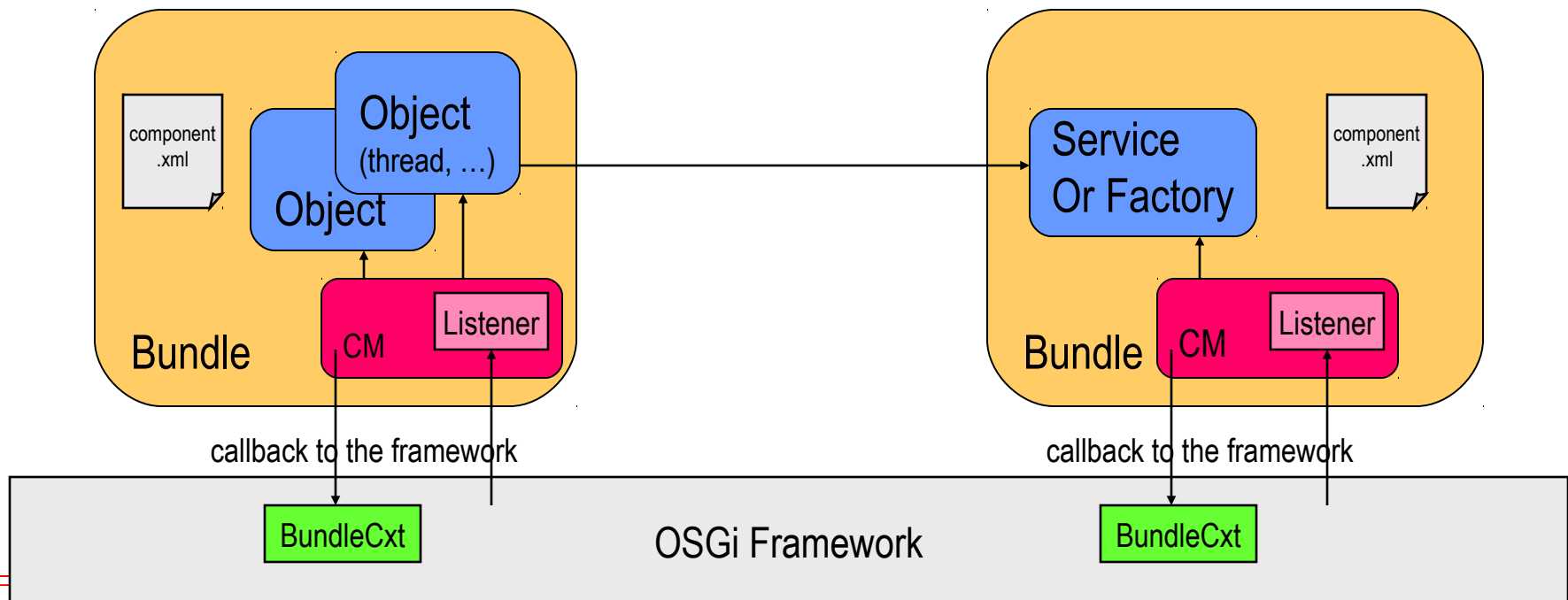


Service Component Runtime

⑦ Gestion descriptive des instances de services et des liaisons

⑦ OSGI-INF/component.xml

CM: SCR Component Manager

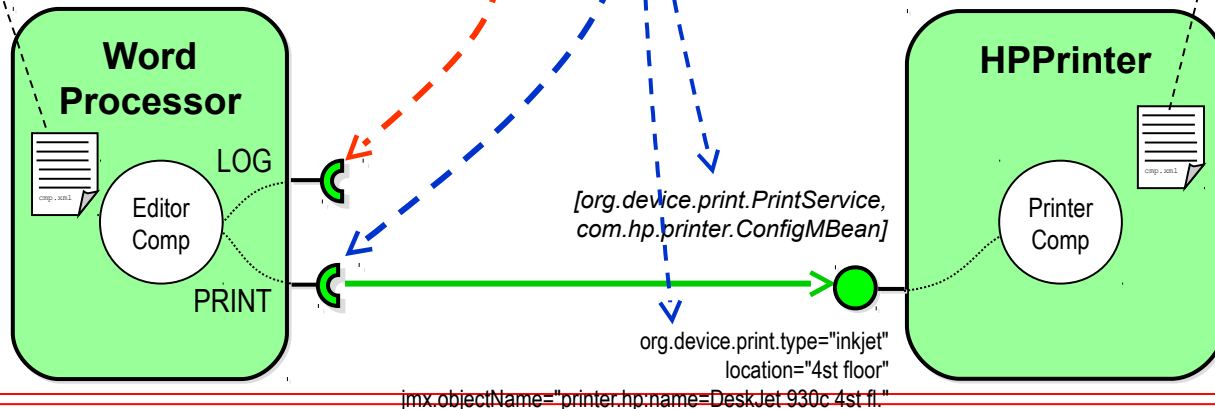


Service Component Runtime

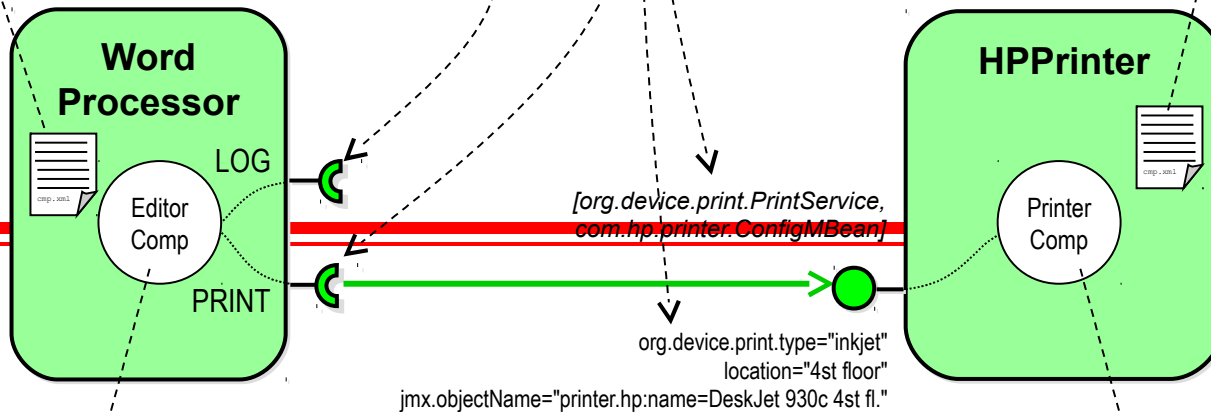
```

<component name="editor">
  <implementation
    class="org.eclipse.texteditor.impl.EditorComp"/>
  <reference name="PRINT"
    interface="org.device.print.PrintService"
    target="(&(location=*)(org.device.print.type=inkjet))"
    cardinality="1..n"
    policy="dynamic"
    bind="bindPrintService"
    unbind="unbindPrintService"
  />
  <reference name="LOG"
    interface="org.osgi.service.log.LogService"
    cardinality="0..1"
    policy="dynamic"
  />
</component>
/OSGI-INF/component.xml

<component name="printer">
  <implementation
    class="com.hp.printer.deskjet.impl.PrinterComp"/>
  <property name="org.device.print.type"
    value="inkjet" type="String"/>
  <property name="location"
    value="4st floor" type="String"/>
  <property name="jmx.objectName"
    value="printer.hp:name=DeskJet 930c 4st fl."
    type="String"/>
  <service>
    <provide interface="org.device.print.PrintService"/>
    <provide interface="com.hp.printer.ConfigMBean"/>
  </service>
</component>
/OSGI-INF/component.xml
  
```



jmx.objectName="printer.hp:name=DeskJet 930c 4st fl."



```

public class EditorComp {
    private List printServices = new ArrayList()

    public void activate(ComponentContext ctxt) {
        LogService log = (LogService)ctxt.locateService("LOG");
        if(log!=null) log.log(LogService.LOG_INFO, "Editor starting");
        log=null; // must release this reference
        // démarre la thread principale du traitement de texte
        ...
    }
    public void deactivate(ComponentContext ctxt) {
        // arrête la thread principale
        ...
        LogService log = (LogService)ctxt.locateService("LOG");
        if(log!=null) log.log(LogService.LOG_INFO, "Editor stopped");
        log=null;
    }
    public void bindPrintService(PrintService ref){
        synchronized (printServices) {
            printServices.add(ref); }
    }
    public void unbindPrintService(PrintService ref){
        synchronized (printServices) {
            printServices.remove(ref); }
    }
    ...
}

```

```

public class PrinterComp
    implements PrintService, ConfigMBean {

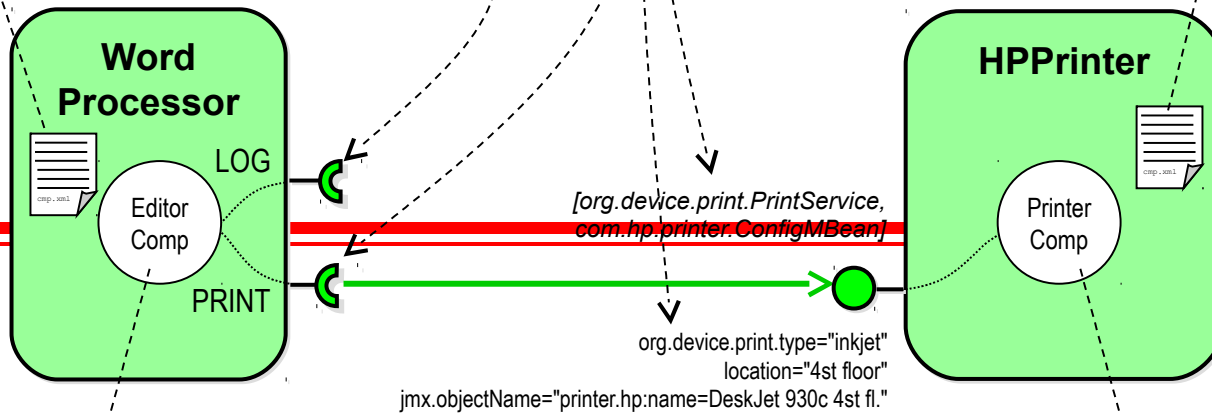
    // méthodes de PrintService
    public Job[] list(){
        ...
    }

    public Job print(InputStream in, Dictionary printParams )
        throws PrintException {
        ...
    }

    // méthodes de ConfigMBean
    public long getPrintedPageCounter(){
        ...
    }

    public void setDefaultTrailer(int trailerId){
        ...
    }
}

```



```

public class EditorComp {
    private List printServices = new ArrayList()

    public void activate(ComponentContext ctx) {
        LogService log = (LogService)ctxt.locateService("LOG");
        if(log!=null) log.log(LogService.LOG_INFO, "Editor starting");
        log=null; // must release this reference
        // démarre la thread principale du traitement de texte
        ...
    }
    public void deactivate(ComponentContext ctx) {
        // arrête la thread principale
        ...
        LogService log = (LogService)ctxt.locateService("LOG");
        if(log!=null) log.log(LogService.LOG_INFO, "Editor stopped");
        log=null;
    }
    public void bindPrintService(PrintService ref){
        synchronized (printServices) {
            printServices.add(ref); }
    }
    public void unbindPrintService(PrintService ref){
        synchronized (printServices) {
            printServices.remove(ref); }
    }
    ...
}

```

```

public class PrinterComp
    implements PrintService, ConfigMBean {

    // méthodes de PrintService
    public Job[] list(){
        ...
    }

    public Job print(InputStream in, Dictionary printParams )
        throws PrintException {
        ...
    }

    // méthodes de ConfigMBean
    public long getPrintedPageCounter(){
        ...
    }

    public void setDefaultTrailer(int trailerId){
        ...
    }
}

```


Apache Felix SCR

⑦ Implementation de Declarative Services

⑦ Metadata: XML, Doclet, Annotation

- ⑦ @Component, @Activate, @Deactivate, @Modified

- ⑦ @Service, @Property, @Reference

⑦ Tools : Ant task, Maven plugin

⑦ A service-oriented component model

⑦ Supporting structural compositions

⑦ Hierarchical

⑦ Built applications are natively dynamic

⑦ Extensible (implemented with an open container)

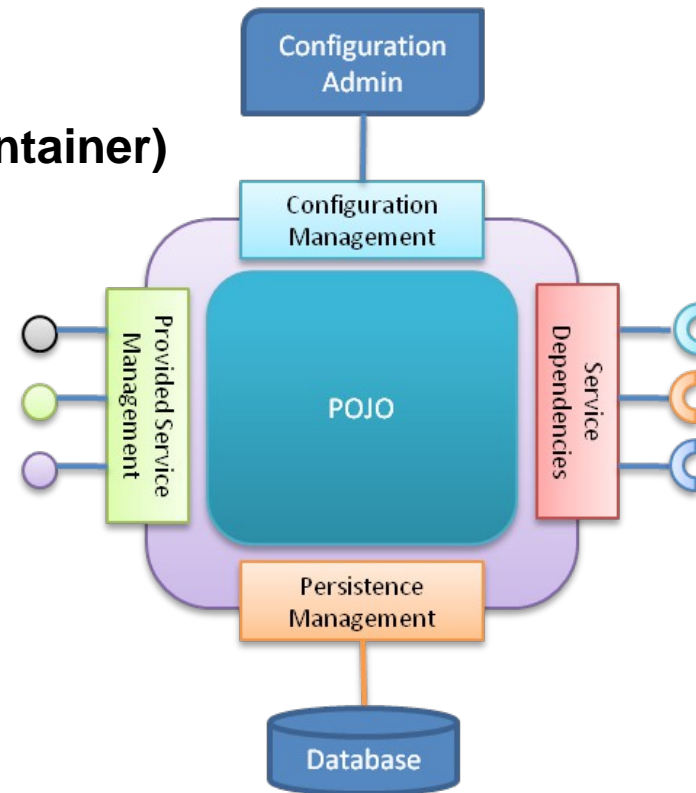
⑦ Key concepts

⑦ Service implementations and instances

⑦ A service specification model

⑦ A service dependency model

⑦ Service context



⑦ Motivations

- ⑦ Exécuter de pur POJOs (Plain Old Java Object)
- ⑦ pour en faire des services ou utilisés des services sans utiliser l'API `org.osgi.framework`

⑦ Principes

- ⑦ Injection de bytecode pour instrumenter les membres (compile time ou deployment time (*manipulator*))
- ⑦ Intercepter *xload/xstore* et *invokex* pour passer le contrôle à des *handlers* (*eux meme des iPOJO*)
- ⑦ Les handlers travaillent en fonction des metadonnées décrites dans un descripteur (`.mf`, `.xml`, `@nnotations 1.5`, *doclet* ...)

iPOJO example

```

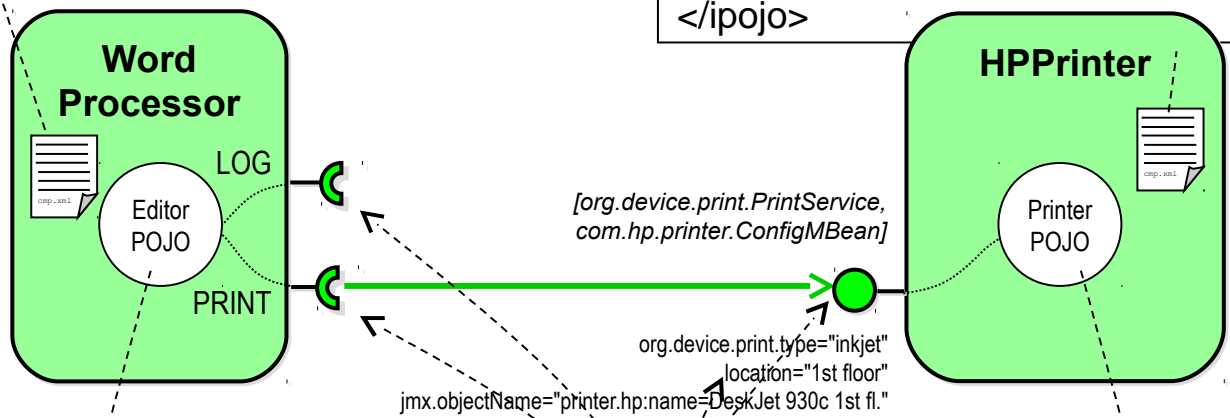
<ipojo>
  <instance component="EditorType"
    name="Editor"/>
</ipojo>
  
```

/metadata.xml

```

<ipojo>
  <instance component="HPPrinterType"
    name="HPPrinter">
    <property name="location" value="1st floor"/>
    <property name="org.device.print.type" value="inkjet"/>
    <property name="jmx.objectName"
      value="printer.hp:name=DeskJet 930c 1st fl."/>
    </instance>
  </ipojo>
  
```

/metadata.xml



```

@Component(name="EditorType", immediate=true)
public class EditorComp {
  @Requires(f Iter="(&(location=*)(org.device.print.type=inkjet))")
  private List<PrintService> printServices;
  @Requires()
  private LogService logService;

  // business methods
  ...
}
  
```

```

@Component(name="HPPrinterType", immediate=true)
@Provides()
public class PrinterComp
  implements PrintService, ConfigMBean {

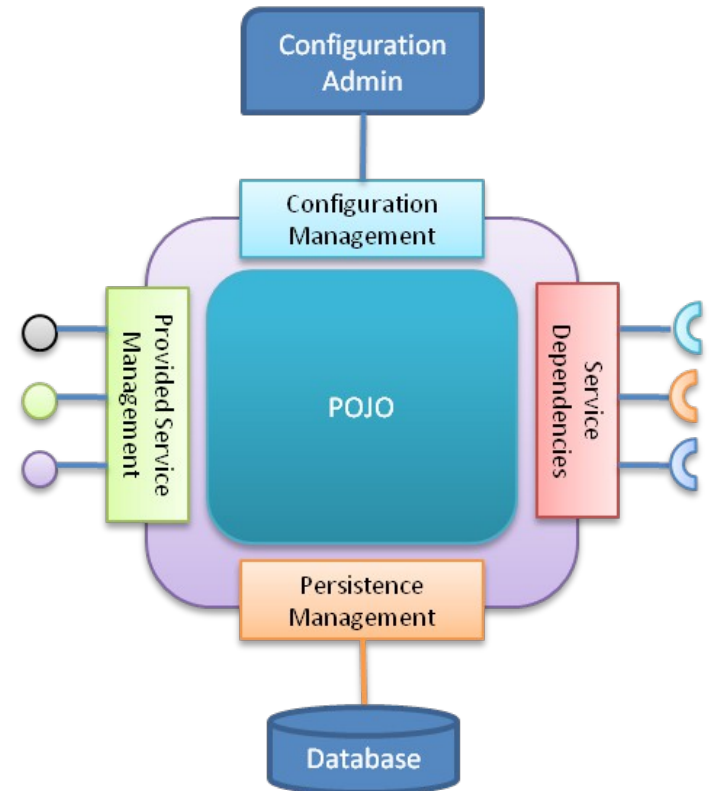
  @ServiceProperty(name="location")
  private String position;

  // méthodes de PrintService
  ...
}
  
```

iPOJO Handlers

- ⑦ Provide
- ⑦ Service Requirement
 - ⑦ Method injection and Field injection
 - ⑦ Nullable object, Default implementation
- ⑦ Temporal Service Requirement
- ⑦ Lifecycle callback
- ⑦ Config
- ⑦ JMX
- ⑦ Event
- ⑦ Extender Pattern
- ⑦ Whiteboard Pattern

- ⑦ Custom ones ...



iPOJO Architecture handler

⑦ Disable architecture

⑦ `<component
 classname="fr.imag.adele.escoffier.hello.impl.HelloServiceImpl"
 architecture="false">`

⑦ Command

- ⑦ `ipojo:factories` - Display iPOJO factories
- ⑦ `ipojo:factory` - Display the information about a specific factory
- ⑦ `ipojo:handlers` - Display iPOJO handlers
- ⑦ `ipojo:instance` - Display the architecture of a specific instance
- ⑦ `ipojo:instances` - Display iPOJO instances

iPOJO Configuration Handler

⑦ Add a CM ManagedService

```
<iPOJO>
<Component className="system.impl.DiskServiceImpl">
  <Provides>
    <Property name="quota" field="m_quota"/>
  </Provides>
  <Properties propagation="true"
    updated="afterReconfiguration"
  />
  <Property name="quota" field="m_quota"/>
  <Property name="threshold" method="updateThresholdArray"/>
  <Property name="disk.name" type="java.lang.String"/>
</Properties>
</Component>
<instance component="system.impl.DiskServiceImpl" name="DiskService">
  <property name="quota" value="100"/>
  <property name="threshold" value="{10, 20, 30}"/>
  <property name="disk.name" value="D"/>
  <property name="managed.service.pid" value="com.acme.system.disk.D"/>
</instance>
</iPOJO>
```

Updates are propagated to the Service Registration

Invoke the method on update

Static property

Instance level

service.pid

```
class DiskServiceImpl implement DiskService {
  int m_quota;
  public void updateThresholdArray(int[] a) { ... }
  public void afterReconfiguration(Dictionary config) { ... }
}
```

iPOJO JMX Handler

⑦ XML

```
<ipojo xmlns:jmx="org.apache.felix.ipojo.handlers.jmx">
  ...
  <jmx:conf g>
    <jmx:property name="quota" field="m_quota"
      rights="r" notification="true"/>
    <jmx:method name="setQuota"/>
  </jmx:conf g>
</ipojo>
```

⑦ @notations (not JSR 255 JMX annotations)

```
@Component
@Conf g(domain="my-domain", usesMOSGi=false)
public class Disk {
  @Property(name="quota", notification=true, rights="r", description="Disk quota")
  private String m_quota;
  @Method(description="set the quota") // Method published in the MBean
  public void setQuota(int q) {
    if(q>0) m_quota = q;
  }
}
```


iPOJO Event Handler Example

⑦ Publication

```
@org.apache.felix.ipojo.handlers.event.Publisher(name="p3", synchronous=true,  
    topics="bar")
```

```
org.apache.felix.ipojo.handlers.event.publisher.Publisher publisher3;
```

```
public void doSomething() {  
    Dictionary d = new Properties();  
    // Fill out the even  
    publisher3.send(d); // Send event  
}
```

⑦ Subscription

```
@Subscriber(name="s1", data_key="data")
```

```
public void receive1(Object o) { // Nothing }
```

```
@Subscriber(name="s2", topics="foo,bar", filter="(data=DIDIER*")
```

```
public void receive2(Event e) { // Nothing }
```

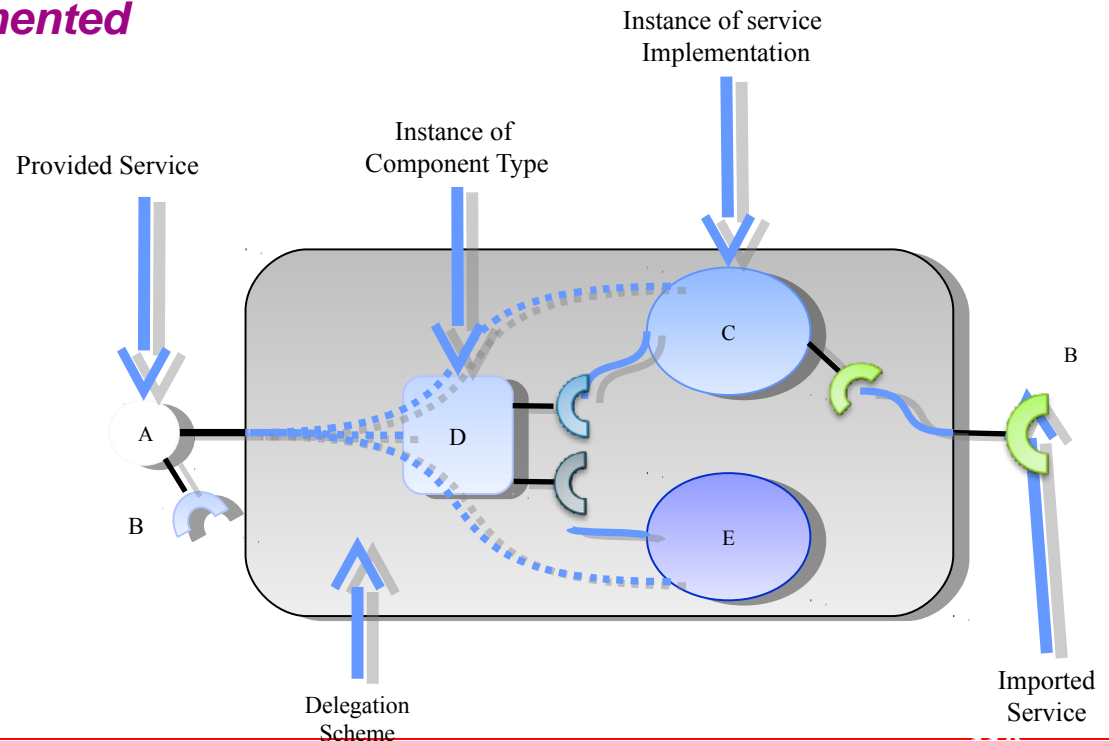
```
@Subscriber(name="s3", topics="foo", data_key="data", data_type="java.lang.String")
```

```
public void receive3(String s) { // Nothing }
```

iPOJO Composite

⑦ Architecture Description Language for

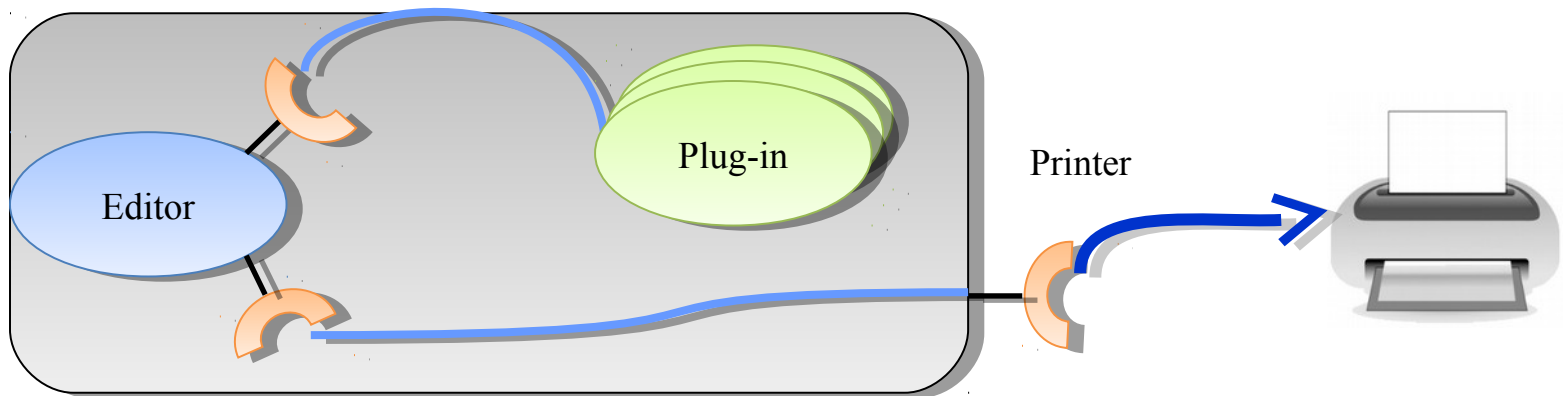
- ⑦ Required Service Specifications
 - ⑦ Instantiated and Imported
- ⑦ Provided Service Specifications
 - ⑦ Exported and Implemented
- ⑦ Component Types



from Clément Escoffier's thesis defense

iPOJO Composite Example

```
<composite name="Editor1">  
<subservice action="instantiate"  
  specification="...Editor"/>  
<subservice action="instantiate"  
  specification="... Plugin" aggregate="true" />  
<subservice action="import"  
  specification="...Printer" optional="true"/>  
</composite>
```



from Clément Escoffier's thesis defense

iPOJO API

⑦ Goal

- ⑦ build factories, components and composites programmatically
 - ⑦ Similar to Dependency Manager

⑦ Example

```
⑦ new PrimitiveComponentType()
⑦ .setBundleContext(context)
⑦ .setClassName(MyComponentImpl.class.getName())
⑦ .addService(new Service()
⑦     .addProperty(new ServiceProperty()
⑦         .setField("myServiceProperty")
⑦         .setName("sample.myProperty"))
⑦     .setCreationStrategy(Service.INSTANCE_STRATEGY))
⑦ .addHandler(new Whiteboard()
⑦     .onArrival("arrival")
⑦     .onDeparture("departure")
⑦     .setFilter("(foo=foo)))
⑦ .createInstance();
```

iPOJO Misc

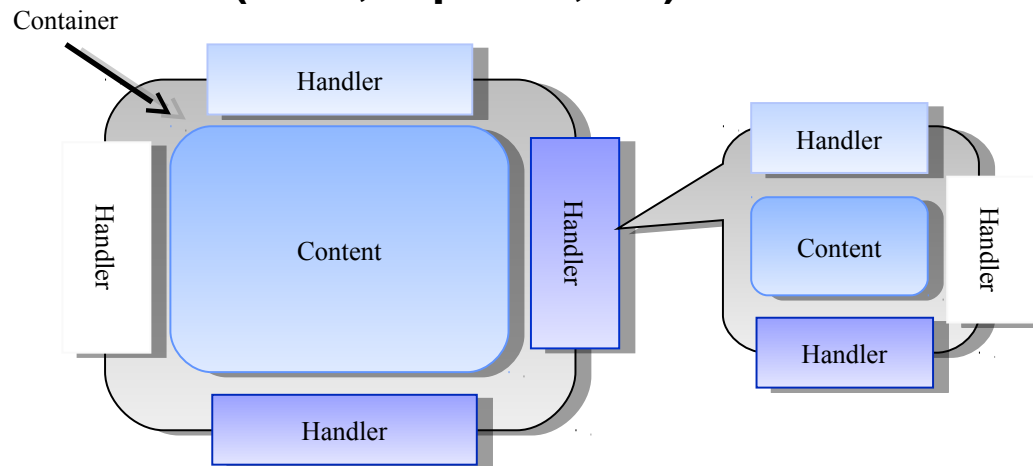
⑦ iPOJO Online Manipulator

⑦

iPOJO implementation

⑦ Main features

- ⑦ **Bytecode manipulation (ASM)**
- ⑦ **Extensible through *Handlers***
 - ⑦ **Handlers are iPOJO instances**
 - ⑦ **Natively support dynamism**
- ⑦ **Heavy use of threads and synchronization constructions**
- ⑦ **on top of OSGi R4.0 (Felix, Equinox, KF) and various JVM 1.4, 1.5+**



from Clément Escoffier's thesis defense

Spring Dynamic Modules (ex Spring OSGi)

- ⑦ **Rappel : Framework POJO (bean)
pour les applications *server-side***
 - ⑦ *Pour les déçus des EJB2*
 - ⑦ *xml ou @notation 1.5*

- ⑦ **Spring Framework + Beans**
 - ⑦ *Conditionnés en bundles*
 - ⑦ *Livrés sur OSGi*
 - ⑦ *+ binding à la Declarative Services (~metadata SCR)*

- ⑦ **Remark**
 - ⑦ *Focus JavaEE ...*

SCA OSGi

⑦ Sujet chaud pour l'EEG

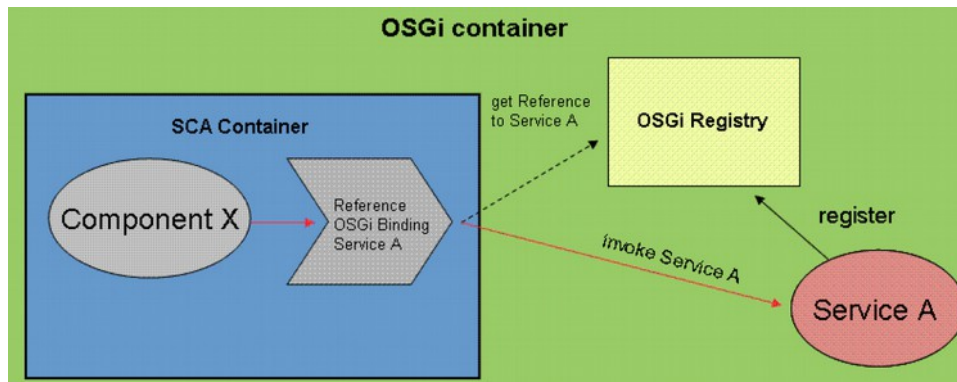
⑦ Service Component Architecture (SCA)

⑦ Hierarchical component model for SO applications

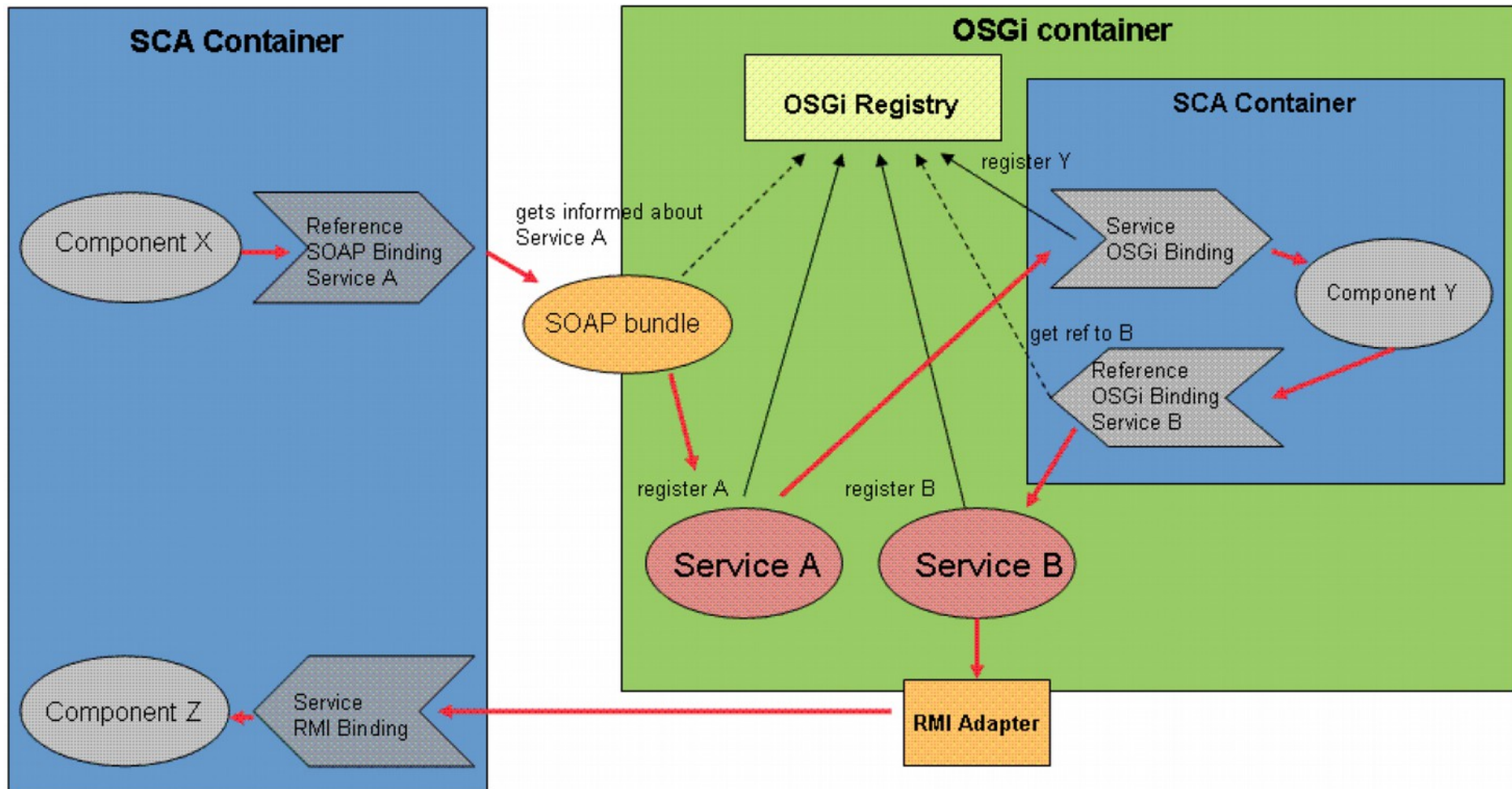
⑦ Java, C#, BPEL, JavaScript, Python ...

⑦ SCA/OSGi

⑦ SCA Container packaged as OSGi bundles then deployed in a OSGi framework

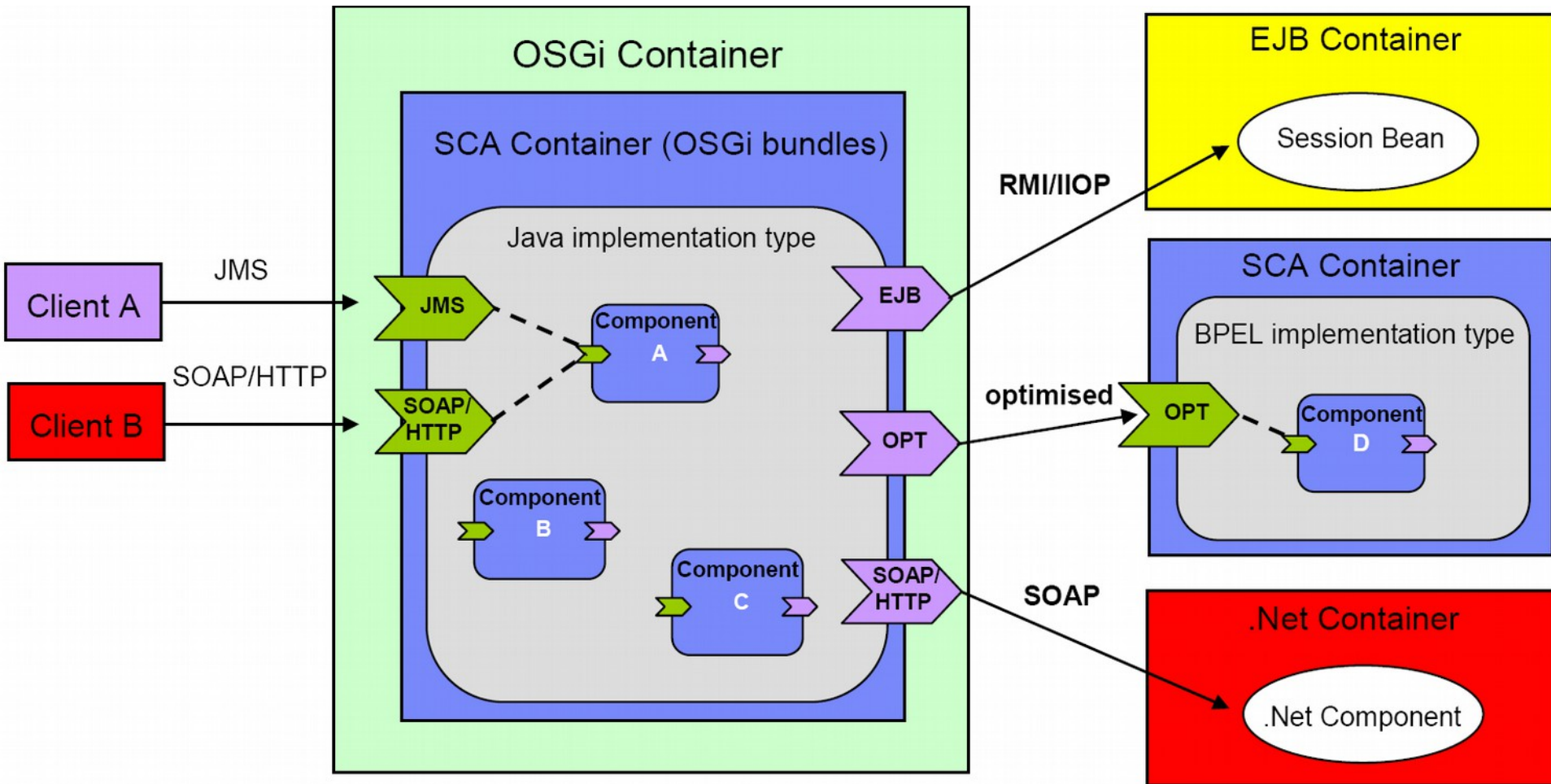


SCA/OSGi Remote Invocation

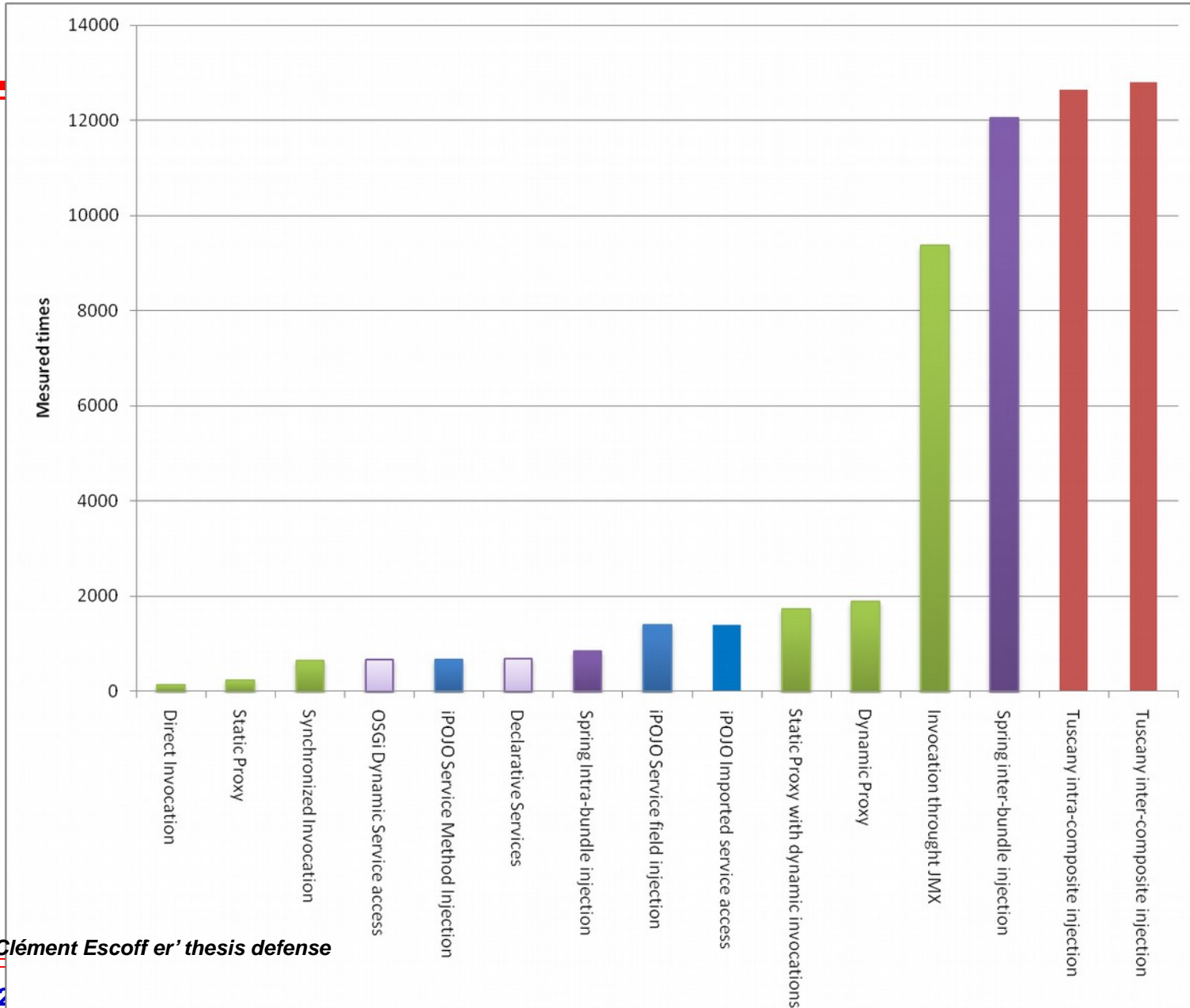


http://www.osoa.org/download/attachments/250/Power_Combination_SCA_Spring_OSGi.pdf?version=3

SCA OSGi



SOCA Benchmark :



from Clément Escoffier's thesis defense

EasyBeans/OSGi

⑦ EasyBeans www.easybeans.org

⑦ Conteneur EJB3.0 – JSR 220 (annotations 5.0) + JSR 181

⑦ *JSR 220* ~ = « *EJB for the dummies* »

⑦ *JSR 181* = *Web Services Metadata for the Java™ Platform*

⑦ Motivations

⑦ Exécuter des POJOs annotés JSR-220 sur OSGi

⑦ Injecter le BundleContext dans les Enterprise Beans (@OSGiRessource)

⑦ *Enregistrement de services / Utilisation de services*

⑦ Fonctionnement

⑦ Conditionner les classes des POJOs dans un bundle

⑦ *l'ejbjar est emballé dans le bundle mais plus de JSR88 !*

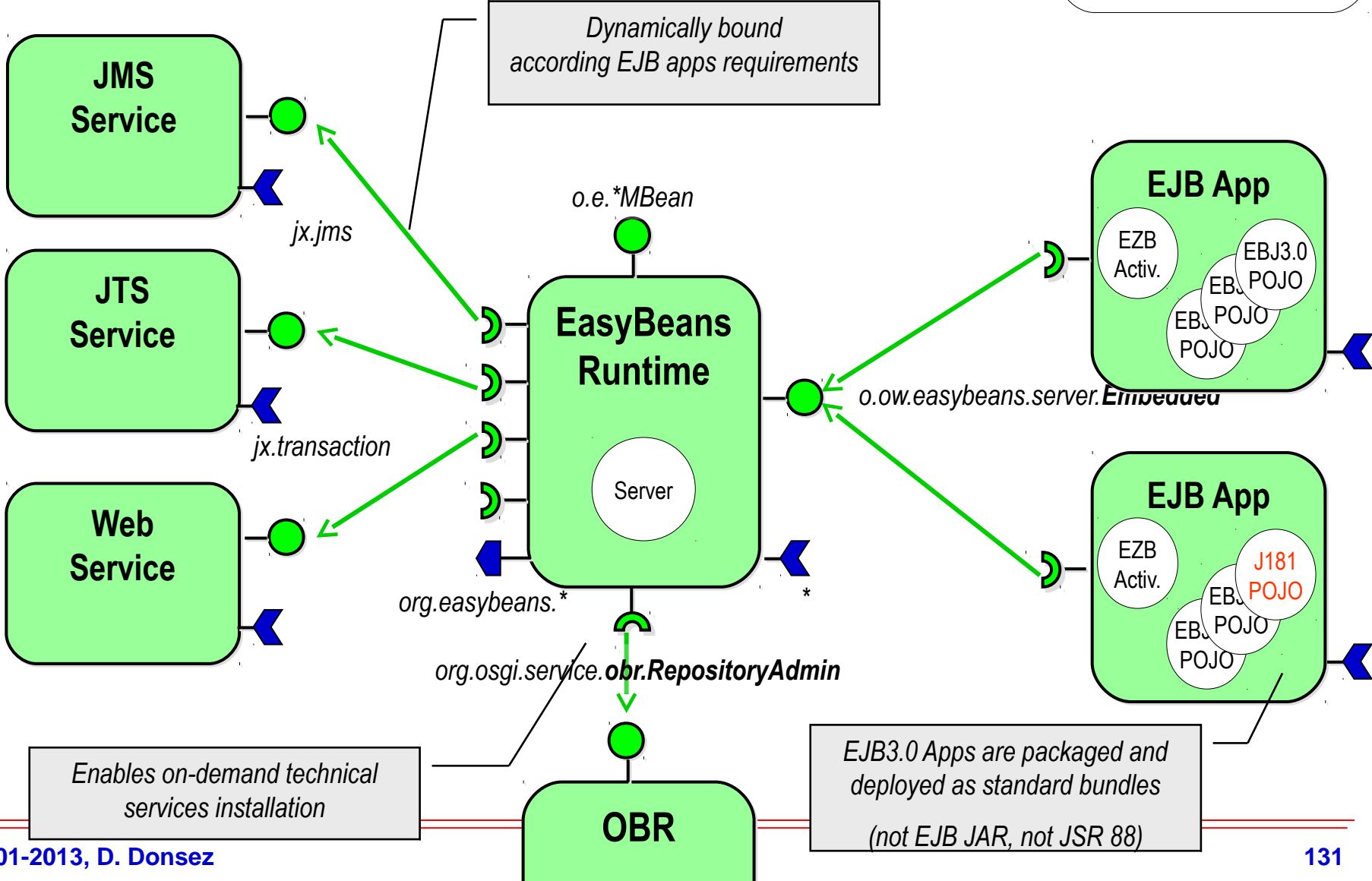
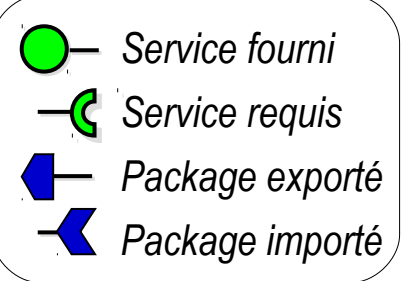
⑦ L'activateur crée un CL qui analyse et injecte les .class annotés et utilise le service du Runtime d'EasyBeans

⑦ Le Runtime d'EasyBeans enregistre les EB Homes et fait l'intermédiaire avec les services techniques requis

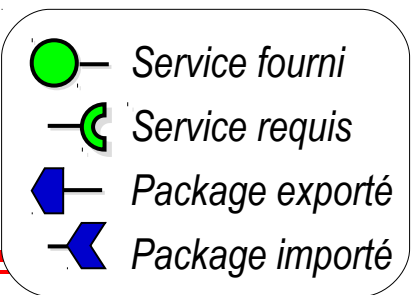


<http://wiki.easybeans.org/xwiki/bin/view/Main/OSGi>

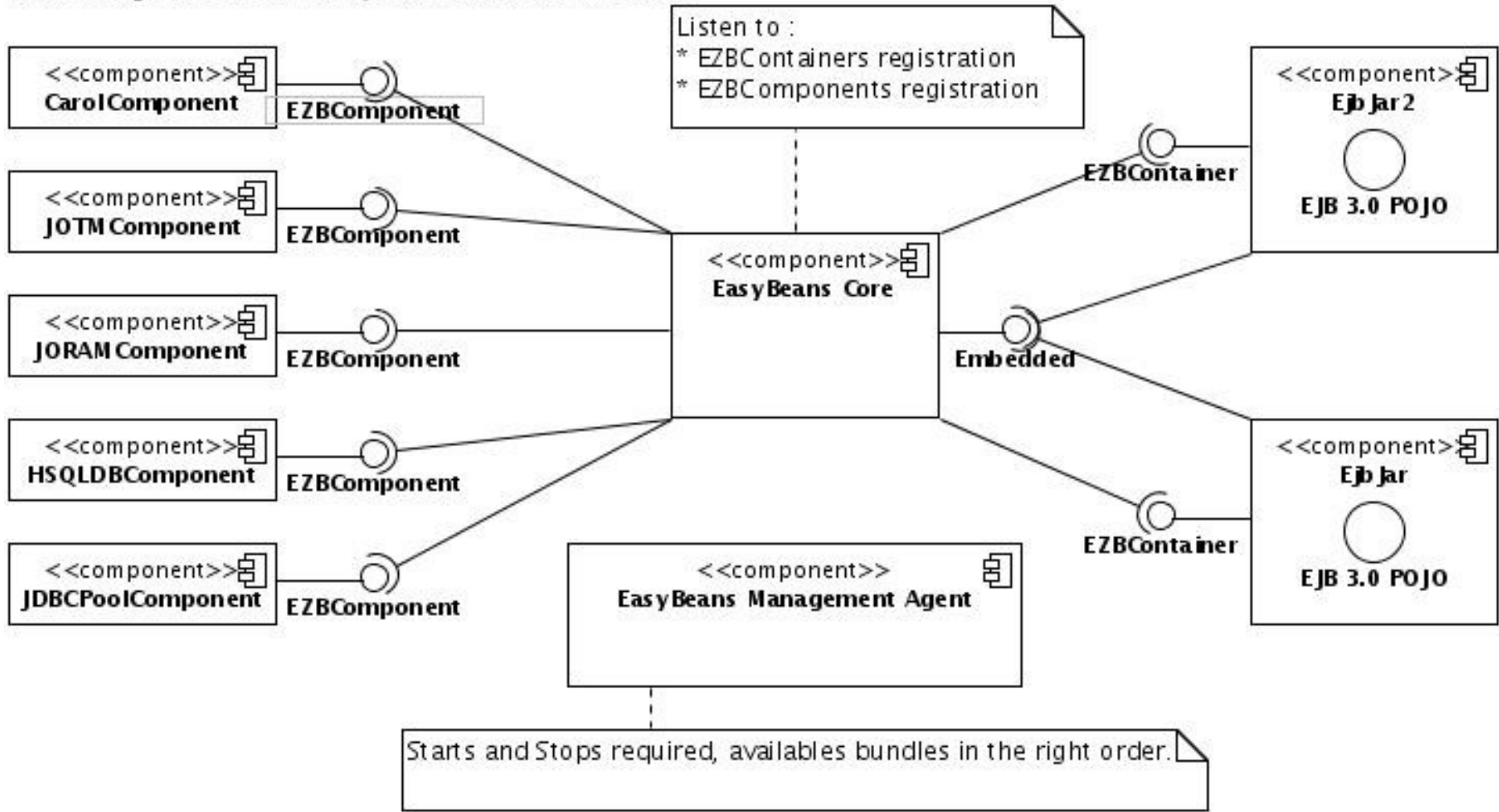
EasyBeans/OSGi Architecture



EasyBeans/OSGi Architecture (état 30/08/2006)



Visual Paradigm for UML Community Edition [not for commercial use]



Guice-OSGi

<http://wiki.ops4j.org/conference/display/ops4j/Guice-OSGi>

⑦ **Google Guice : IoD injector based on Java 5 annotations**

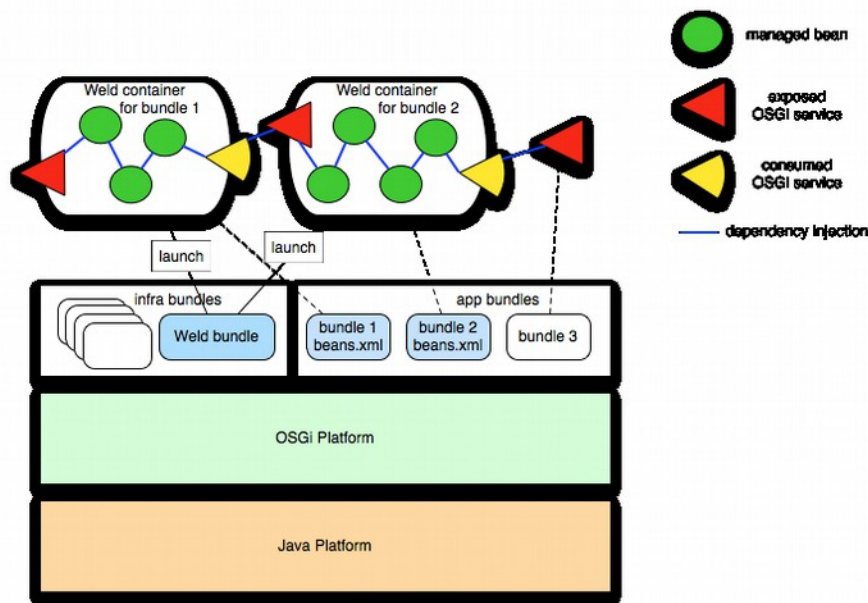
⑦ **Guice-OSGi injects required and provided services in a Java class**

⑦ **Example**

- ⑦ **@Inject @OSGiService MyService unaryProxy;**
- ⑦ **@Inject @OSGiService Iterable<MyService> multipleProxy;**
- ⑦ **@Inject @OSGiService("(code=simple)") /* custom LDAP filter */ MyService filtered;**
- ⑦ **@Inject @OSGiService(interfaces = {MyService.class}) /* custom interface list */ MyOSGiListener listener;**
- ⑦ **@Inject @OSGiServiceRegistration MyService registeredService;**
- ⑦ **@Inject @OSGiServiceRegistration("lang=en,location=uk") /* custom service properties */ MyService configuredService;**
- ⑦ **@Inject @OSGiServiceRegistration(interfaces = {MyBaseService.class}) /* custom interface list */ MyService customizedService;**
 - ⑦ **Registered OSGi services can be controlled using the static methods in the GuiceOSGi utility class (ie. enable/disable/modify).**
- ⑦ **@Inject BundleContext bc;**
 - ⑦ **Inject the bundle context**

Jboss Weld OSGi

- ⑦ **JBoss Weld: CDI runtime**
 - ⑦ JSR 299 RI, aligned with JSR 330 (DI for Java)
- ⑦ **Inject OSGi imported and exported services in JavaSE and JavaEE artifacts (WAR, ...)**



Jboss Weld OSGi Example

@Publish

@ApplicationScoped

```
public class PrinterImpl implements PrinterService {
    @Inject @OSGiService @Filter("&(cat=printer)(subcat=color)")LogService logService;
    @Overrides
    public void print(Document doc) {
        ...
        logService.log("Job #"+job.getId()+" : Printing doc : "+doc.getName());
        ...
    }
    public void onStartup(@Observes BundleContainerInitialized event) {}
    public void onShutdown(@Observes BundleContainerShutdown event) {}
    ...
}
```

ServiceFactory

⑦ Motivation

- ⑦ Retourne une instance par bundle client
 - ⑦ différentes instances pour un même service
 - ⑦ Attention : ~~~ patron de conception de la fabrique (Gamma)
- ⑦ Nécessaire à un singleton manipulant son ServiceRegistration

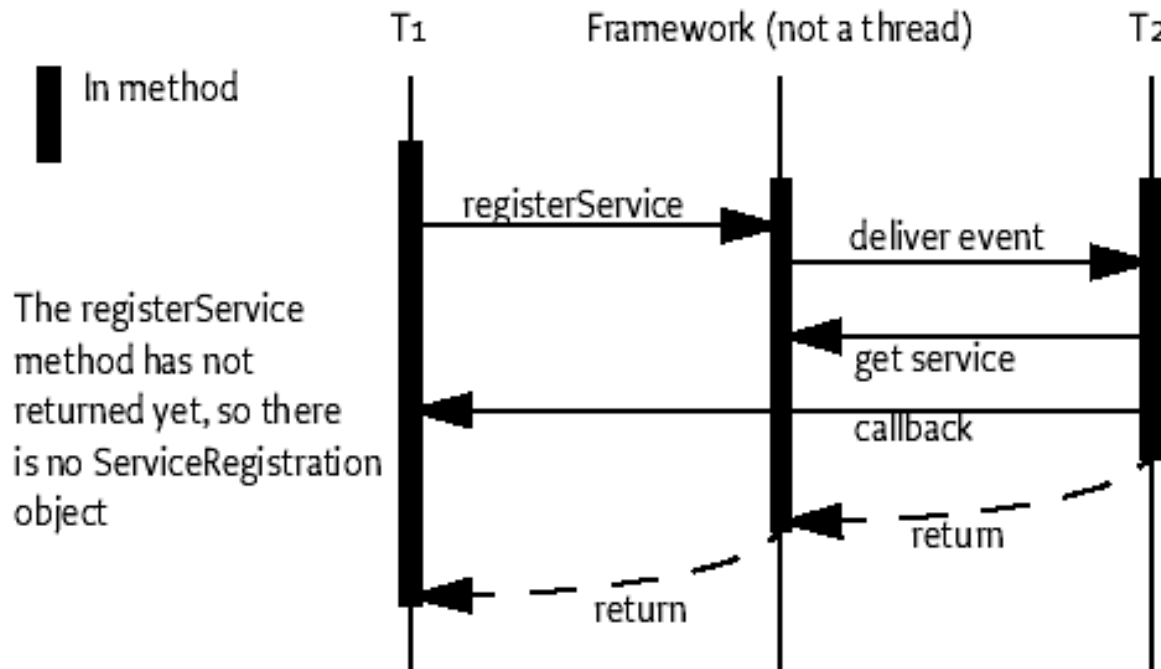
⑦ Utilisation

- ⑦ Gestion de sessions multiples
 - ⑦ Multi-fenêtres
 - ⑦ plusieurs shells
- ⑦ Contrôle d'accès (le bundle réclamant le service est connu)
 - ⑦ Contrôle des permissions
- ⑦ Suivi des références « cachées »
 - ⑦ Patron « Whiteboard » (chapitre suivante)

ServiceFactory

⑦ Interface à implémenter

```
interface ServiceFactory {  
    Object getService(Bundle bundle, ServiceRegistration registration)  
    public void ungetService(Bundle bundle, ServiceRegistration registration, Object service)  
}
```



Sécurité

- ⑦ **Basé sur les permissions du JDK1.2**
 - ⑦ Le SecurityManager vérifie les permissions de chaque bundle
 - ⑦ Exemple : FilePermission, DialPermission, ...
 - ⑦ Accès aux fichiers
 - ⑦ Ceux du cache et aux ressources du bundle
- ⑦ **3 permissions propres à OSGi**
 - ⑦ AdminPermission
 - ⑦ Autorise l'accès aux fonctions d'administration du framework.
 - ⑦ ServicePermission
 - ⑦ Contrôle l'enregistrement et la récupération de services
 - ⑦ PackagePermission
 - ⑦ Contrôle l'import et l'export de packages
- ⑦ **org.osgi.service.PermissionAdmin**
 - ⑦ Service de gestion des permissions des bundles
- ⑦ **Conditional Permission Admin (R4)**

OSGi



Guide de Bonnes pratiques

Modularité

- ⑦ **Séparer les classes « published » (ie contrat) des classes « propriétaires » dans des paquetages des différents**
 - ⑦ **Seul les classes « published » doivent être exportées**
- ⑦ **Conditionner les contrats et les implémentations dans des bundles séparés**
 - ⑦ **Les contrats ne varient peu et sont partagés par plusieurs bundles**
- ⑦ **Import-Package plutôt que Require-Bundle (R4)**
 - ⑦ **substitutabilité avec d'autres fournisseurs de packages**
- ⑦ **Limitez l'usage des fragments**
- ⑦ **Evitez l'usage de DynamicImport-Package**
 - ⑦ **Sauf cas particulier (livraison dynamique de plugin, ...)**
- ⑦ **Définissez le ExecutionEnvironment**
 - ⑦ **Et utilisez le pour la compilation !!!**

Conditionnement

⑦ JAR enfouis

- ⑦ Ne déconditionnez pas les JAR
 - ⑦ Utilisez le Bundle-ClassPath
- ⑦ Conservation des signatures, manifestes, ...
- ⑦ Possibilité de les patcher !
 - ⑦ Bundle-ClassPath: patch.jar,original.jar

Service (i)

⑦ Rappel

1. Code de fourniture d'un service << Code d'usage d'un service

2. 1 service = 1 entrée dans le registre

⑦ Courtage sur plusieurs milliers/millions de services

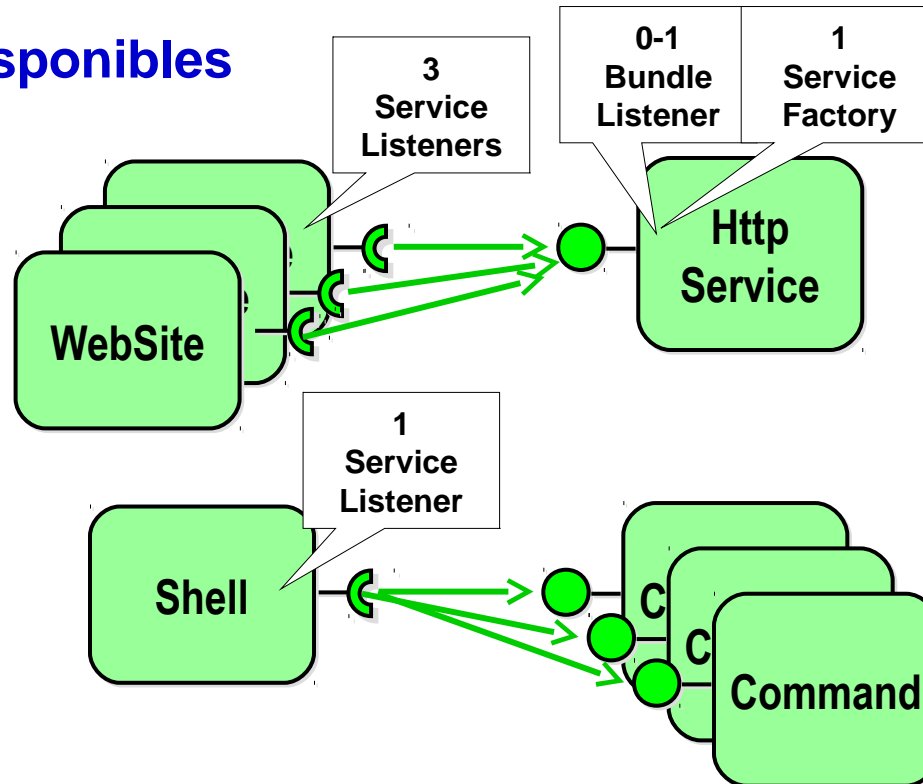
1. Traitement séquentiel (et synchrone) des listeners

⑦ Quid pour plusieurs milliers de listeners par framework

Service (ii)

⑦ 2 patrons disponibles

⑦ *Listener*



Exemple:
LogService,
HttpService,
...

⑦ *Whiteboard*

Exemple:
EventAdmin,
IOConnector,
Felix' Shell,
...

⑦ **Preferer (le plus possible) le patron *Whiteboard* (sauf si (2))**



⑦ Listeners Considered Harmful: The “Whiteboard” Pattern

⑦ http://www.osgi.org/documents/osgi_technology/whiteboard.pdf

Le patron par extension (Extender)

⑦ Limite l'usage des services (listeners)

⑦ Principe

- ⑦ 1 BundleListener scrute les bundles contenant des metadatas particulières (WEB-INF/web.xml, OSGI-INF/component.xml, plugin.xml ...)
- ⑦ et instancie des objets depuis le ClassLoader du bundle scruté lors du démarrage, dépose les objets créés

⑦ Exemples

- ⑦ SCR, iPOJO, Spring DM, Eclipse Extension Point Registry...

Eclipse Extension Point

⑦ Motivation

- ⑦ Legacy : Eclipse 2 Plugin
- ⑦ Fast startup : lazy loading, cached state (no event storm)
- ⑦ Declarative : plugin.xml

⑦ Notion

- ⑦ IExtensionPoint
- ⑦ IExtension
- ⑦ IExtensionRegistry
 - ⑦ Dynamic: IExtensionChangeHandler

⑦ Example

- ⑦ IExtensionRegistry registry = null;
- ⑦ // ...
- ⑦ IExtensionPoint point = registry.getExtensionPoint("xyz");
- ⑦ ExtensionTracker tracker = new ExtensionTracker(registry);
- ⑦ IFilter f lter = tracker.createExtensionPointFilter(point);
- ⑦ tracker.registerHandler(new MyExtensionChangeHandler(), f lter);

⑦ 99.9% of plugin developers ignore dynamic issues

- ⑦ Bad programming → update requires restart

NetBeans Platform

⑦ **TODO**

⑦ **Same**

Service (iii)

⑦ **Granularité**

⑦ ***Objet << Composant << Service << Bundle***

Membres statiques

⑦ **AIE !**

⑦ **Evitez les dans les classes « published »**

⑦ Difficile de tracker le cycle de vie du bundle qui les fournit

⑦ **Il est fréquent de voir**

```
class MyActivator implements BundleActivator {  
    public static BundleContext bundleContext;  
    ...  
}
```

Chargeurs de classe (ClassLoaders)

⑦ Usage

- ⑦ Dynamic bytecode injection
- ⑦ Dynamic aspect-weaving
- ⑦ Dynamic annotation processing
- ⑦ ...

⑦ Principe

- ⑦ Le chargeur de classe doit avoir pour parent le chargeur du bundle
 - ⑦ ie `MyActivator.class.getClassLoader()`
- ⑦ Le chargeur peut demander la récupération des ressources (.class) au bundle
 - ⑦ `Enumeration e = bundle.findEntries("/", "*.class", true);` r4

⑦ Exemples

- ⑦ ProActive, Julia (FROGi), EasyBeans, ...

Cycle de vie

⑦ Activation

- ⑦ 1 sec par bundle

→ 1 minute pour 60 bundles → 5 minutes pour 300 bundles → ...

- ⑦ Evitez des activateurs qui durent longtemps

⑦ Solutions

- ⑦ Rendre la main au canevas le plus vite

- ⑦ Utilisez l'eventing

- ⑦ Ou une thread en arrière plan

- ⑦ Préférez l'activation paresseuse (R4.1 *lazy activation*)

- ⑦ *lazyness is goodness*

Cycle de vie

- ⑦ **Vous développez le start()**
- ⑦ **Pensez aussi au stop()**
 - ⑦ **Proscrire System.exit()**
 - ⑦ **Votre bundle n'est pas le seul sur la JVM**
 - ⑦ **il y en a d'autres qui bossent**
 - ⑦ **Libérez les ressources**
 - ⑦ **Fermez les fichiers, sockets, connections, ...**
 - ⑦ **Arrêtez les *threads* démarrés**
 - ⑦ **Mieux encore rendez les au service de *pool* de *threads***
 - ⑦ **Nullifiez les références vers les servants**
 - ⑦ **Garbage collection (objets et classes)**
- ⑦ **Pensez au *finally* { ... } à la fois dans le start() et le stop() ou activate() et deactivate()**

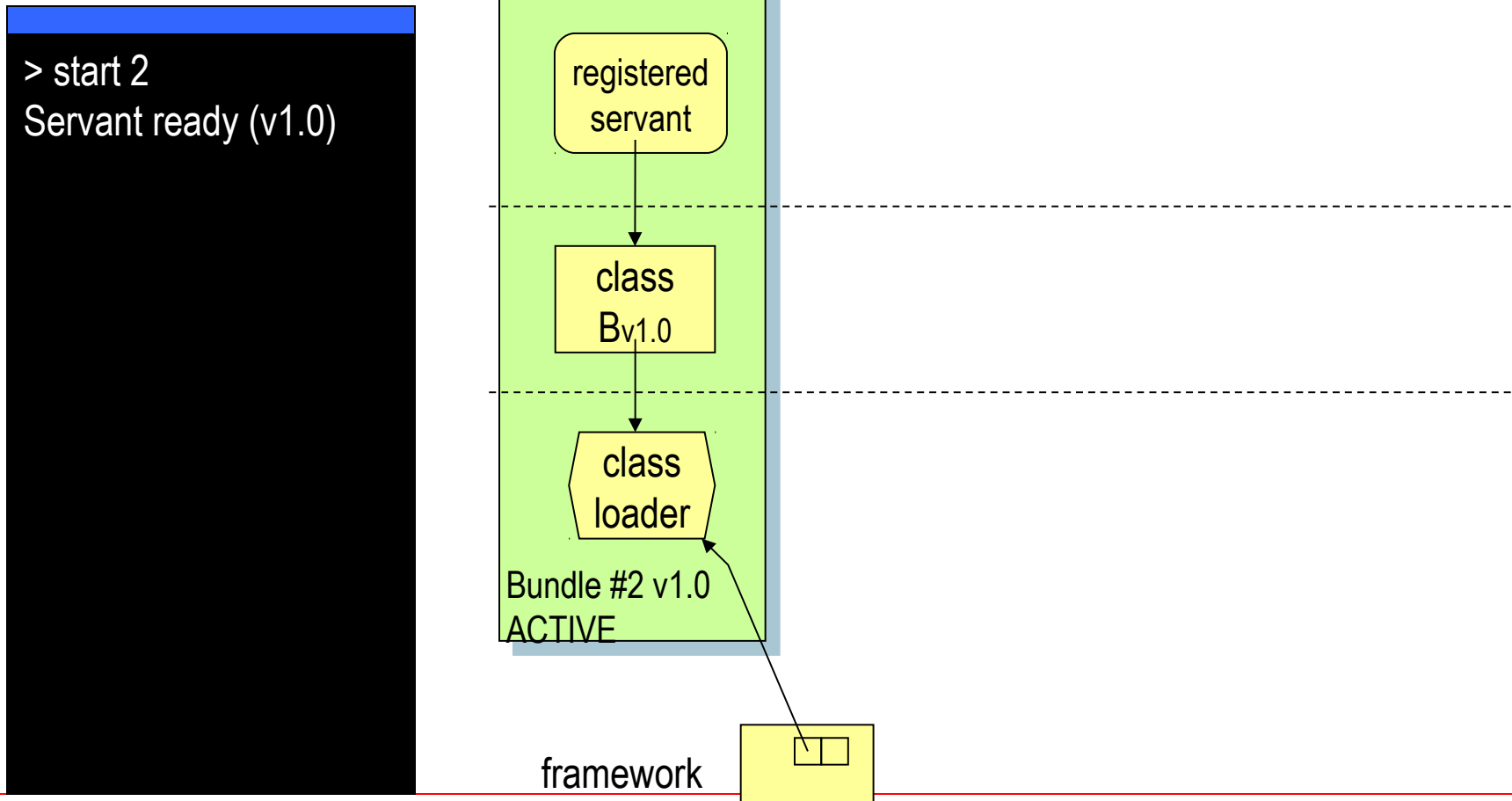
What are Stale References?

“a reference to a Java object that belongs to the class loader of a bundle that is stopped or is associated with a service object that is unregistered”

OSGi R4 Section 5.4

An example of Stale Reference Pathology?

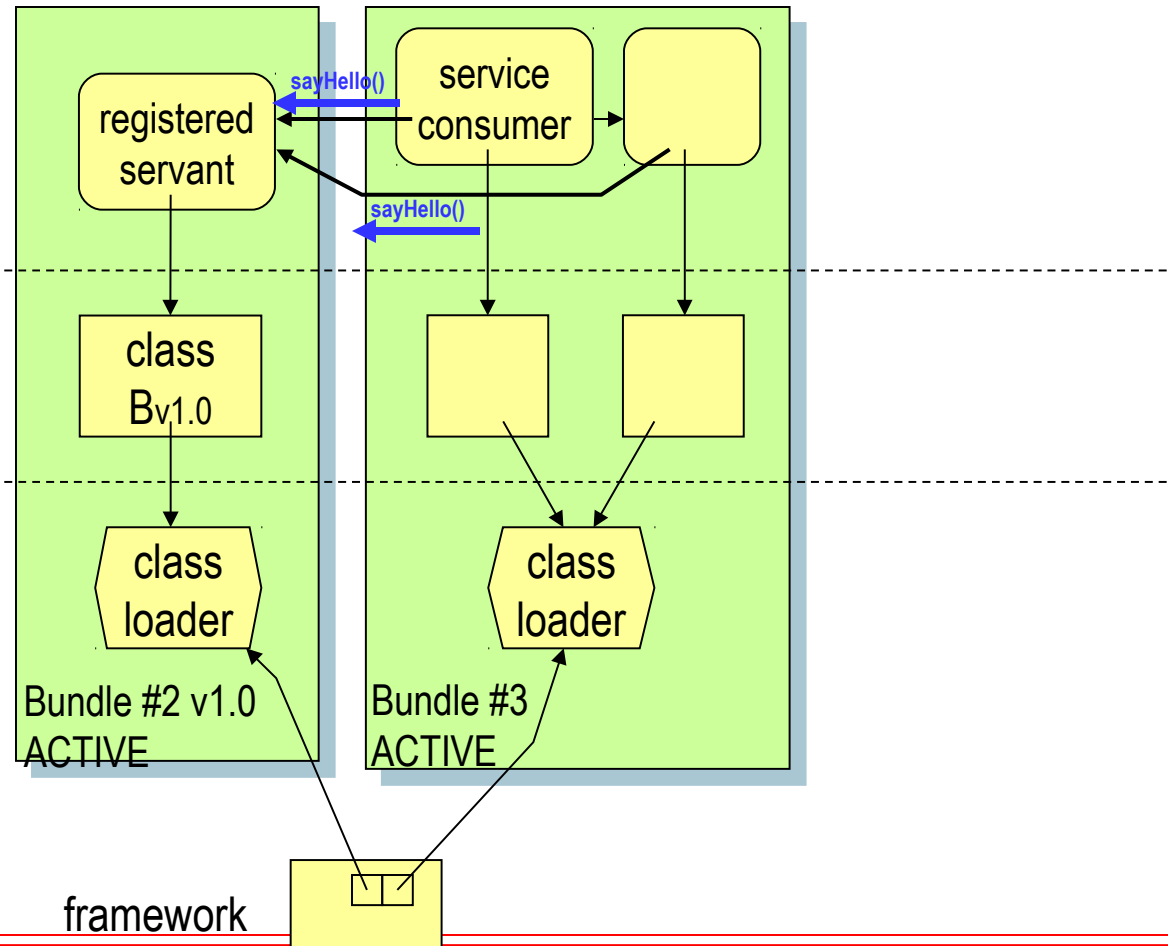
(i) initial



An example of Stale Reference Pathology?

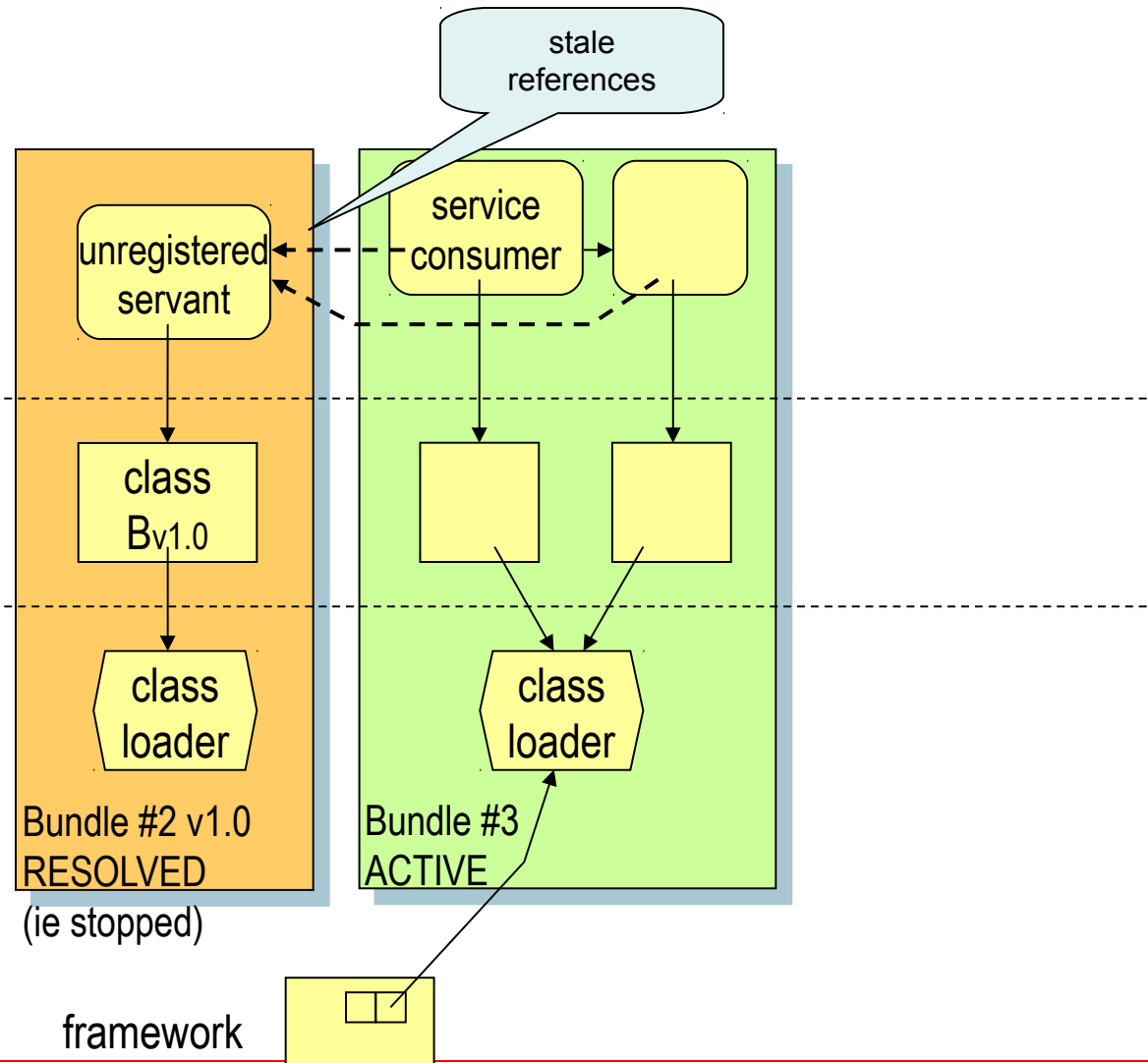
(i) initial

```
> start 2
Servant ready (v1.0)
> start 3
1- Hello World ! (v1.0)
2- Hello World ! (v1.0)
```



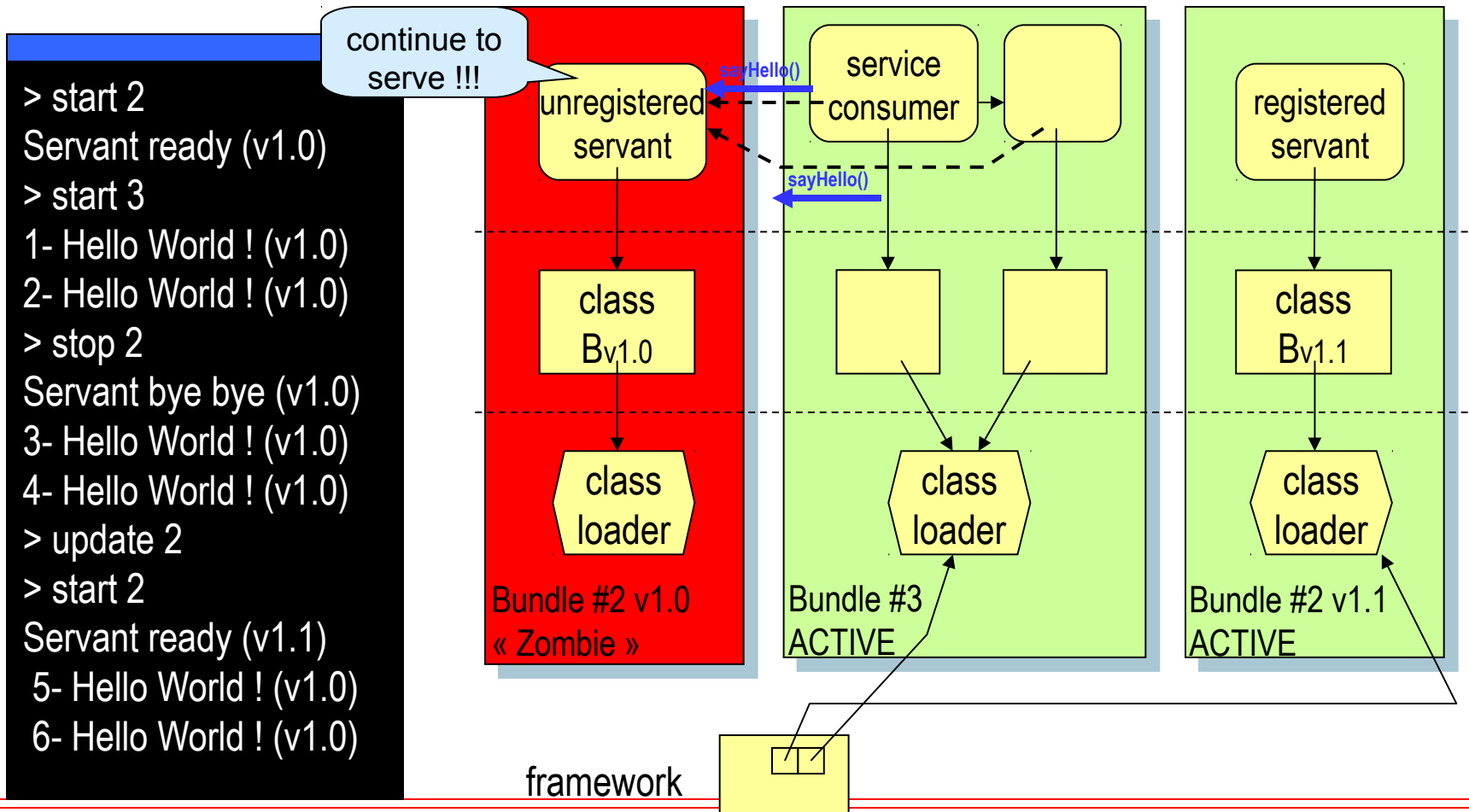
An example of Stale Reference Pathology? (ii) After stop 2

```
> start 2
Servant ready (v1.0)
> start 3
1- Hello World ! (v1.0)
2- Hello World ! (v1.0)
> stop 2
Servant bye bye (v1.0)
```



An example of Stale Reference Pathology?

(iii) After update 2 & start 2



Bad Consequences

⑦ Memory leaks

- ⑦ Retention of the classloader of a stopped or uninstalled bundle
- ⑦ Retention of all `java.lang.Class` loaded by that bundle

⑦ Utilization of invalid services → Inconsistencies!

- ⑦ Service is unregistered but still used (wrong!)
- ⑦ Its context is most likely inconsistent
 - ⑦ e.g. closed connections, old data
- ⑦ Possible exceptions upon service calls
 - ⑦ good because we can see the problem
- ⑦ Silent propagation of incorrect results (worst case!)
 - ⑦ E.g. Returning old cached-data

Other « stale » pathologies

- ⑦ **“Forwarded references”**
 - ⑦ From one bundle to another
- ⑦ **“Stale” threads**
 - ⑦ bundle has stopped but created threads has not
- ⑦ **Unregistered MBeans, RemoteObjects, ...**
- ⑦ **Unreleased resources**
 - ⑦ sockets, file descriptors, locks, ...
- ⑦ **Stale ExtensionPoints**
 - ⑦ Eclipse suggests to restart after updating ! ;-(

How to ensure « stale reference free » applications?

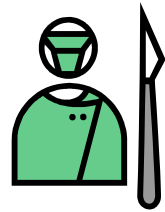
- ⑦ **2 cases of OSGi™ applications**
- ⑦ **From-scratch OSGi™ development**
- ⑦ **Bundlization of Legacy codes**
 - ⑦ Really frequent
 - ⑦ Module with or without Services/Extension Points

- ⑦ **Good OSGi™ programming practices**
 - ⑦ Who trusts their developers ?
- ⑦ **Component Models**
 - ⑦ Necessary but not enough

- ⑦ **Stale references may be there but we can't see them...**

- ⑦ **→ We need Diagnosis**
victim bundles x guilty bundles

The ServiceCoroner tool



- ⑦ A diagnostics tool for detecting stale references in OSGi™ applications
- ⑦ “Inspector” of services death
- ⑦ Runtime diagnosis
- ⑦ Points out victim bundles/services and possible suspects

VisualVM Milestone 3

org.ow2.jonas.commands.admin.ClientAdmin (pid 1448)

Overview Monitor Threads Profiler MBeans JConsole

org.ow2.jonas.commands.admin.ClientAdmin (pid 1448)

JConsole

Service Coroner

ID	Bundle Name	State	Stale References
20	org.ow2.bundles.ow2-util-xmlconfig	STARTED	0
21	org.ow2.bundles.ow2-util-ee-deploy-api	STARTED	0
22	org.ow2.bundles.ow2-util-ee-deploy-impl	STARTED	2
23	org.ow2.bundles.ow2-bundles-externals-commons-collecti...	STARTED	0
24	org.ow2.bundles.ow2-bundles-externals-igr...	STARTED	0

Refresh Try GC Bundle Details

Bundle 22

Service Factories: 0
Service instances: 3
Service references: 4
Stale references: 2

*The coroner is a legal examiner that investigates the causes of unnatural deaths in English speaking countries. Not all coroners have forensic pathology knowledge, but for illustration purposes we have named our tool as ServiceCoroner.

Stale References are not a myth !

Experiment results (May 2008)

I	OSGi-based software	JOnAS (JavaEE server)	SIP Comm. (multiprotocol VoIP and Chat UA)	Newton (SCA container)	Sling (Content Repository)
II	Version	5.0.1	Alpha 3	1.2.3	2.0 incubator snapshot
III	OSGi Impl.	Felix 1.0	Felix 1.0	Equinox 3.3.0	Felix 1.0
IV	Bundles using Component Models	20 iPOJO	6 Service Binder	0	18 Declarative Services
V	Lines of Code	Over 1 500 000	Aprox. 120 000	Aprox. 85 000	Over 125 000
VI	Total Bundles	86	53	90	41
VII	Initial No. of Service Refs.	82	30	142	105
VIII	No. of Bundles w/ Stale Svcs.	4	17	25	2
IX	No. of Stale Services Found	7	19	58	3
X	No. of Stale Threads	2	4	0	0
XI	Stale Services Ratio (IX/VII)	8.5 %	63 %	40.8%	2.8%

[1] Actually the whole Newton implementation is an SCA constructed on top of OSGi, but its bundles did not use an OSGi component model like the other analyzed applications did.

Plus

A lire

- ⑦ **BJ Hargrave & Peter Kriens, « OSGi Best Practices! »,
Session TS 1419, 2007 JavaOne Conference et OSGi Users
Community Event 2007**

⑦ <http://www2.osgi.org/wiki/uploads/Conference/OSGiBestPractices.pdf>

- ⑦ **« Extensions vs Services: Digging Deeper », OSGi Users
Community Event 2007**

⑦ <http://www2.osgi.org/wiki/uploads/Conference/NeilBartlett.pdf>

Autres

⑦ Les outils arrivent ...

- ⑦ PDE, Management (Console)

⑦ Les recettes arrivent ...

- ⑦ Migration vers OSGi en 2 étapes

- ⑦ **Modularisation**

- ⑦ Ecueil : ClassLoader, Thread context switch, Ressources, JNDI, static, default packages (includes for rsc.), ...

- ⑦ **(D)SOA ou Extension Points**

- ⑦ Exemple personnel :

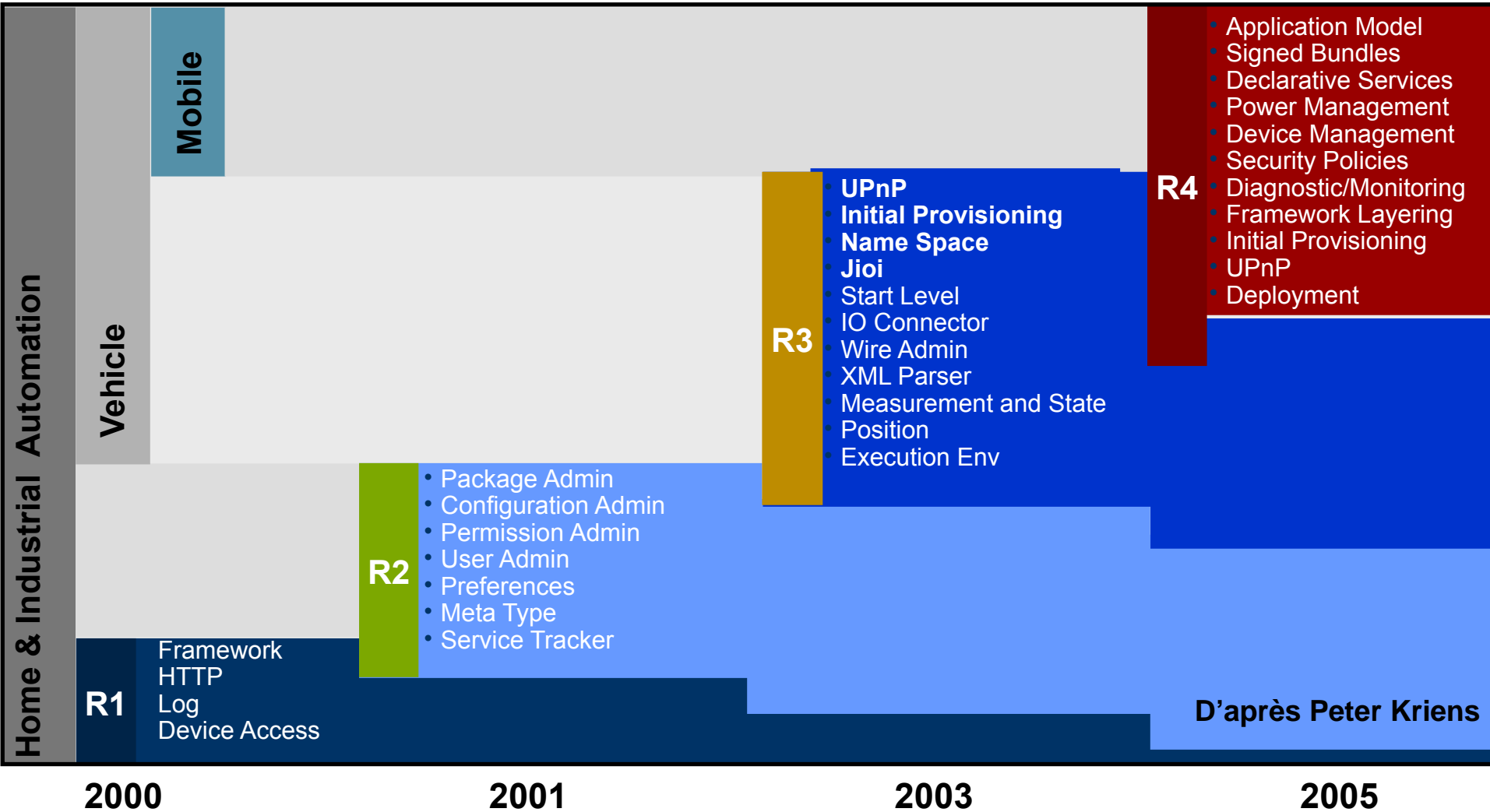
- ⑦ **Tomcat (dans JOnAS), JacORB, ...**

OSGi



Les services standard

Les services OSGi standard



SystemBundle

- ⑦ Représente le framework. Son identifiant est toujours 0.
- ⑦ Son cycle de vie correspond au démarrage et à l'arrêt du Framework

Configuration Admin

⑦ Motivation

⑦ Configuration des bundles

⑦ à partir de configuration externe ou interne (metatype)

⑦ Configuration service.pid + dictionary ({ <key, value> })

⑦ Persistence des configurations et des reconfigurations

⑦ Service

⑦ *ManagedService*

⑦ receive a single configuration dictionary when is registered or when its configuration is modified.

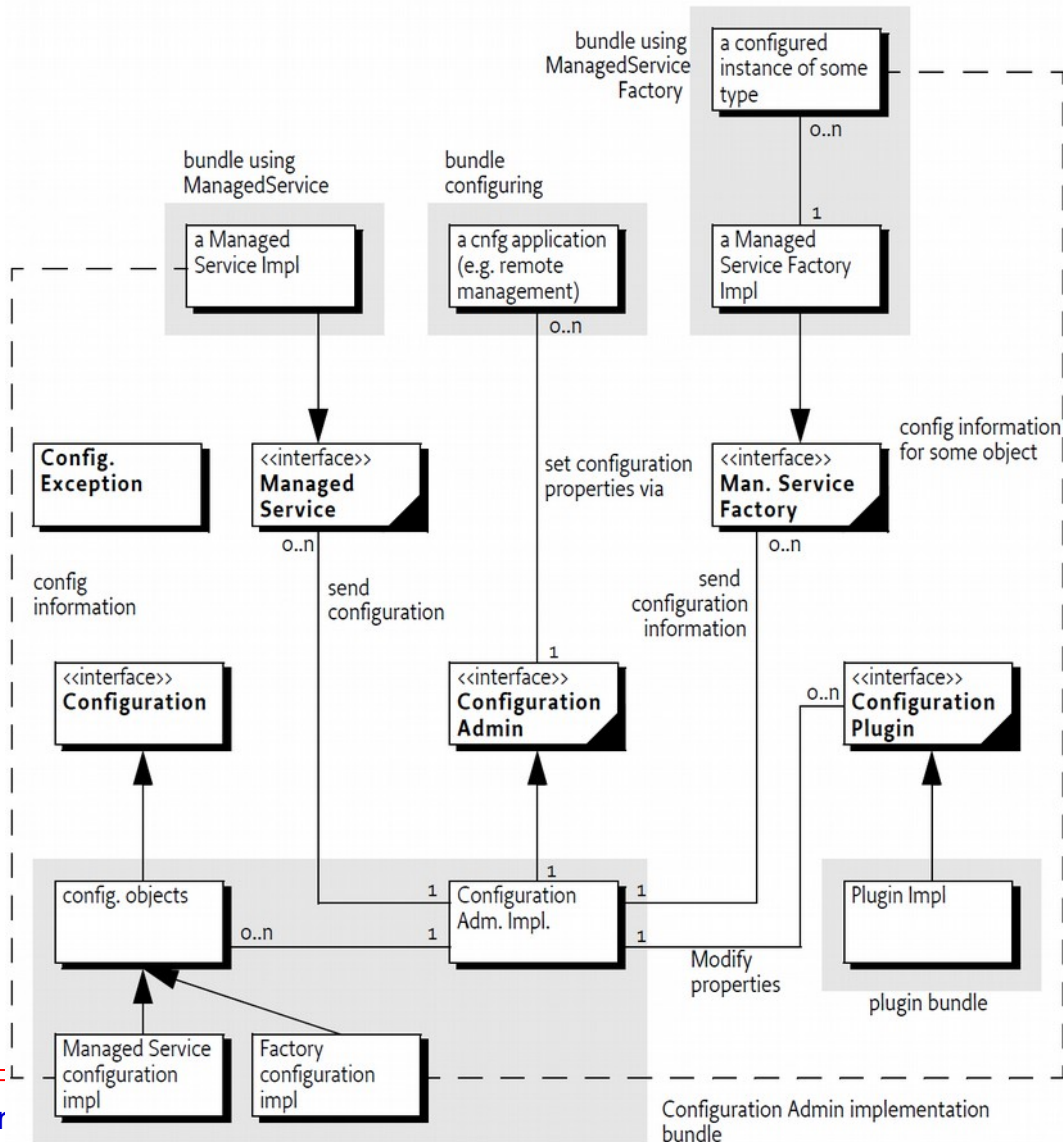
⑦ **Identification de la configuration par la propriété *service.pid***

⑦ *ManagedServiceFactory*

⑦ must receive from zero to n configuration dictionaries when it registers, depending on the current configuration. it is informed of configuration dictionary changes: modifications, creations, and deletions.

⑦ **Identification des configurations par la propriété *service.factoryPid +service.pid***

Configuration Admin API org.osgi.service.cm



Example of ManagedService

```
public class DiskServiceImpl implements ManagedService {
    static final int DEFAULT_QUOTA=100;
    Dictionary properties;
    ServiceRegistration registration;
    int m_quota;
    Disk disk;

    public synchronized void start(BundleContext context) throws Exception {
        properties = new Hashtable();
        properties.put(Constants.SERVICE_PID, "com.acme.system.disk.D");
        properties.put("quota", new Integer(DEFAULT_QUOTA));
        registration = context.registerService(ManagedService.class.getName(),this, properties);
        disk=new Disk();
    }

    public synchronized void updated(Dictionary np) throws ConfigurationException {
        if (np != null) {
            properties = np;
            m_quota = ((Integer) properties.get("quota")).intValue();
            disk.setQuota(m_quota);
            registration.setProperties(properties);
        }
    }
    ...
}
```

Conf gAdmin command in Karaf

- ⑦ **osgi:ls**
- ⑦ **osgi:ls | grep ManagedService**

- ⑦ **conf g:list**
- ⑦ **conf g:edit org.apache.felix.karaf.shell.ssh**
- ⑦ **conf g:proplist**
- ⑦ **conf g:propdel sshRealm**
- ⑦ **conf g:propset sshPort 8102**
- ⑦ **conf g:proplist**
- ⑦ **conf g:update**
- ⑦ **REM the update is persistent**

- ⑦ **osgi:shutdown**
- ⑦ **REM launch Karaf**

- ⑦ **conf g:edit org.apache.felix.karaf.shell.ssh**
- ⑦ **conf g:propset sshRealm karaf**
- ⑦ **conf g:propset sshPort 8101**
- ⑦ **conf g:cancel**

- ⑦ **conf g:edit org.apache.felix.karaf.shell.ssh**
- ⑦ **conf g:propset sshRealm karaf**
- ⑦ **conf g:propset sshPort 8101**
- ⑦ **conf g:update**

- ⑦ **conf g:list**

Conf guration Events

⑦ Conf gurationListener

- ⑦ conf gurationEvent(Conf gurationEvent)

⑦ Conf gurationEvent

- ⑦ getFactoryPid() null for ManagedService
- ⑦ getPid()
- ⑦ getType() : CM_DELETED or CM_UPDATED

⑦ EventAdmin

- ⑦ Topic org/osgi/service/cm/Conf gurationEvent/<event type>

Example

```
public EchoServer implements ManagedService, BundleActivator {
    EchoServerImpl server = EchoServerImpl(-1); // Dummy server
    public void update(Dictionary props ) {
        int port = -1;
        if ( props != null) {
            Object o = props.get("port");
            if ( o != null )
                port = (Integer) o;
        }
        if ( server.getPort() != port ) {
            server.quit();
            server = new EchoServerImpl(port);
        }
    }
}
```

Configuration Admin

Configurable<T>

⑦ Use 1.5 Generics for configuration

⑦ Interface de configuration

```
// PID is com.acme.webconf.WebConf
package com.acme.webconf;
public interface WebConf {
    int port();
    String host();
}
```

⑦ Service

```
@Component
public Webserver implements Configurable<WebConf> {
    public void setup(WebConf conf ) { web( conf.port(), conf.host() ); }
    public void deleted(WebConf conf) { stop(); }
}
```

⑦ Factory

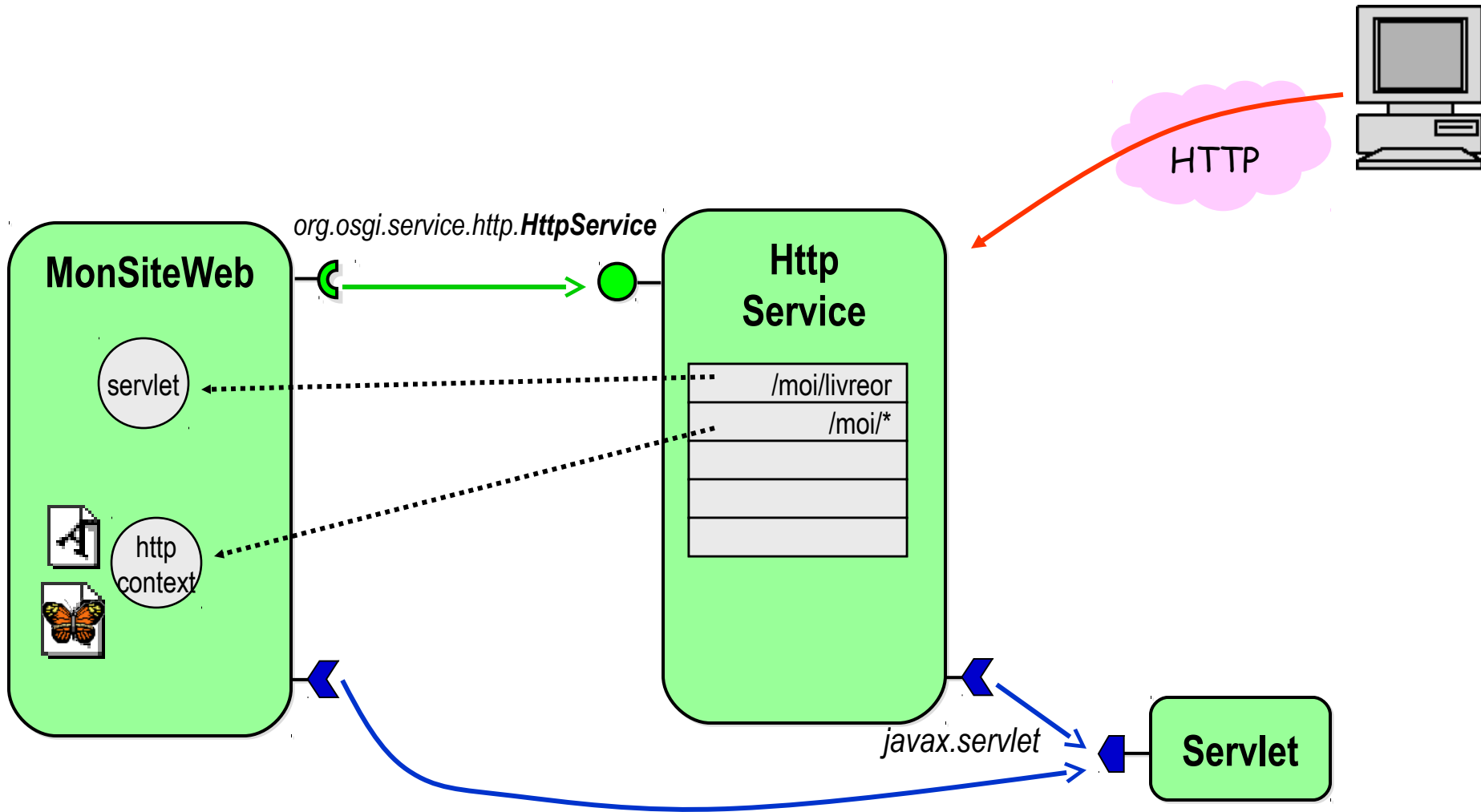
```
public interface ConfigurableFactory<WebConf> {
    Configurable<WebConf> newInstance();
    void noConfigurations();
}
```

HttpService r1

org.service.http.HttpService (i)

- ⑦ **Service permettant à d'autres bundles de publier des servlets et ressources par HTTP**
 - ⑦ Important : Web-based management
- ⑦ **Implémentations**
 - ⑦ embarquent un serveur HTTP compact (Jetty,...)
 - ⑦ Authentification et Autorisation (BasicSchema, SSL)
 - ⑦ Servlets Web-Services : XML-RPC, kSOAP, SOAP/HTTP, RESTFuL, ...
- ⑦ **Extra**
 - ⑦ <jspc> pour éviter d'embarquer un compilateur de JSP
 - ⑦ Convertisseurs WAR to Bundles
 - ⑦ Canevas Web (Cocoon, Wicket, DysoWeb ...)

org.service.http.HttpService (ii) Usage



org.service.http.HttpService (iii)

Exemple d'activateur d'un site Web

```
HttpService https= ...;
```

```
String WEBROOT = "/webroot"; // embedded ressources in BUNDLE-CLASSPATH jarfiles
```

```
String WEBROOT_ALIAS = "/moi";
```

```
String SERVLET_ALIAS = WEBROOT_ALIAS + "/livreor";
```

```
Servlet servlet=new LivreOrServlet(param1,param2);
```

```
https.registerServlet(SERVLET_ALIAS, servlet, null, servlet);
```

```
HttpContext docsContext = new HttpContext() {
```

```
    public String getMimeType(String name) {
```

```
        return (name.endsWith("htm"))?"text/html":null; }
```

```
    public boolean handleSecurity(HttpServletRequest req,HttpServletResponse resp) { return true; }
```

```
    public URL getResource(String name) {
```

```
        URL u = this.getClass().getResource(name);
```

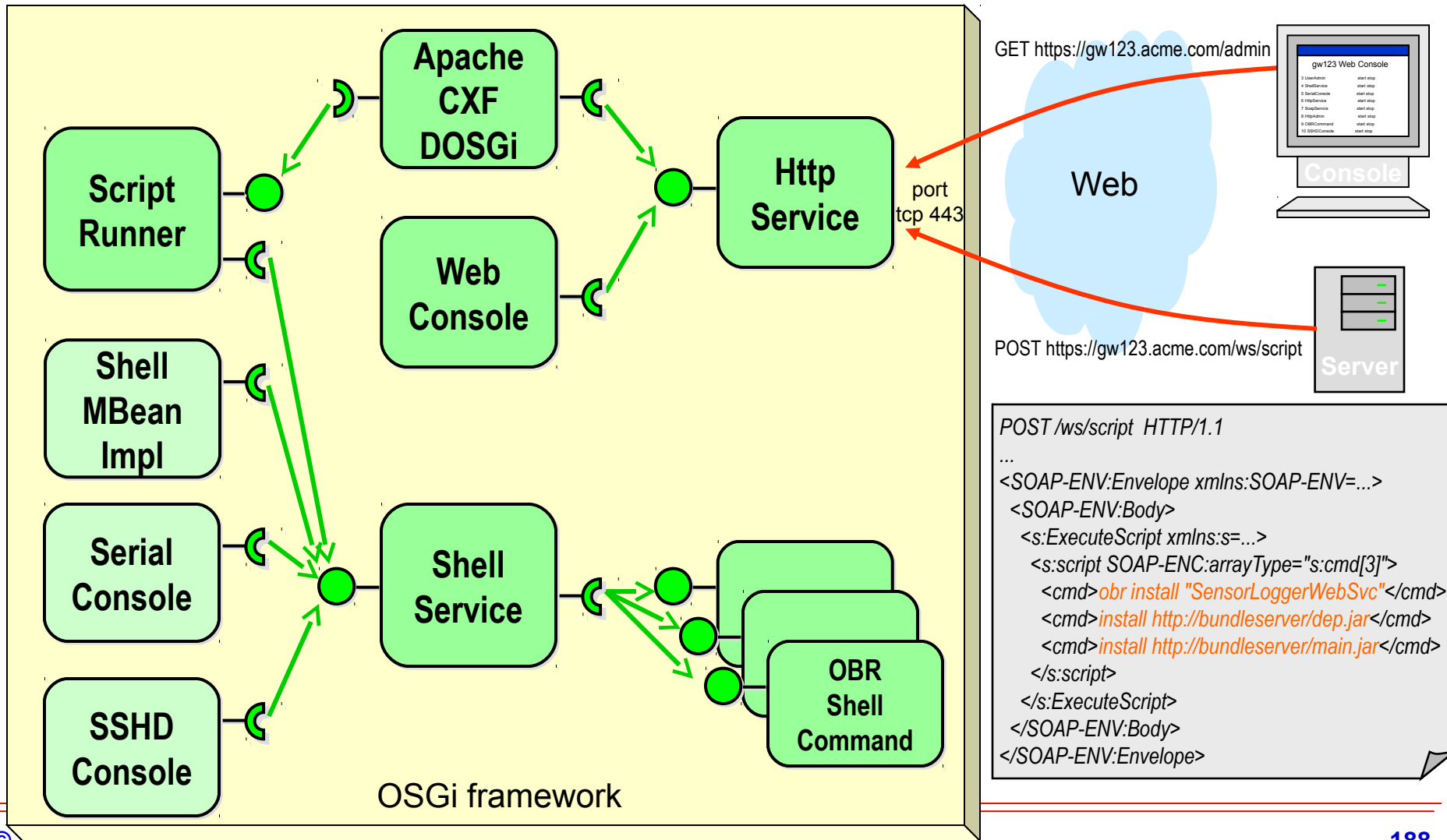
```
        System.out.println(this.getClass().getName());
```

```
        return u;
```

```
    }
```

```
https.registerResources(WEBROOT_ALIAS, WEBROOT, docsContext );
```

Exemple d'architecture pour un shell



org.osgi.service.http Limitations

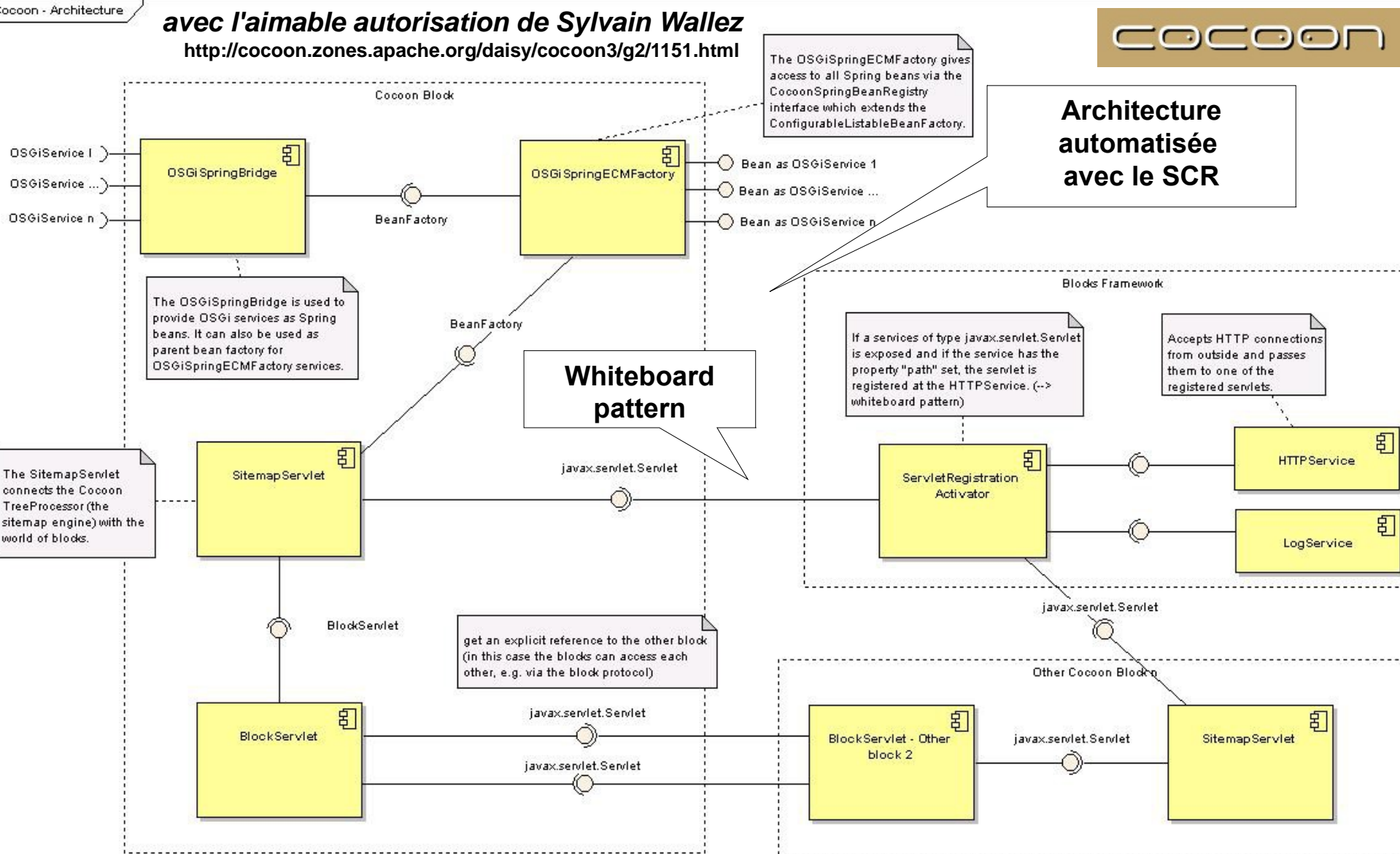
⑦ Motivations

- ⑦ org.osgi.service.http is limited to javax.servlet v2.1, portlet JSR 168
 - ⑦ No filter chain, no listener, basic JSP, no TagLib, ...
- ⑦ Hidden org.osgi.service.http stuff to JEE developers
- ⑦ Keep compatibility with standard webapps (.war)

⑦ Propositions

- ⑦ Cocoon
- ⑦ Wicket
- ⑦ DysoWeb
 - ⑦ Dynamic sub-webapps
 - ⑦ Embed Felix in a War : subwebapps are deployed by Felix FW
 - ⑦ Tested with Tomcat
- ⑦ HttpRegistry
 - ⑦ Server-side equinox <http://www.eclipse.org/equinox/server/>
 - ⑦ Embed Equinox in a War
 - ⑦ Use extender model
 - ⑦ Converter GWT (Google Web Toolkit) module → 2 OSGi bundles
 - ⑦ Tested with Tomcat and Jetty

Un autre exemple: l'architecture de Cocoon 3.0



Un autre exemple: Wicket OSGi

⑦ Wicket

- ⑦ « *Wicket is a **Java web application framework** that takes simplicity, **separation of concerns** and ease of development to a whole new level. Wicket pages can be mocked up, previewed and later revised using standard WYSIWYG HTML design tools. Dynamic content processing and form handling is all handled in Java code using a first-class component model backed by **POJO data beans** that can easily be **persisted** using your favourite technology* » from JavaGeek.org

⑦ Wicket sur OSGi

- ⑦ **TODO**

- ⑦ **Utilise le SCR**



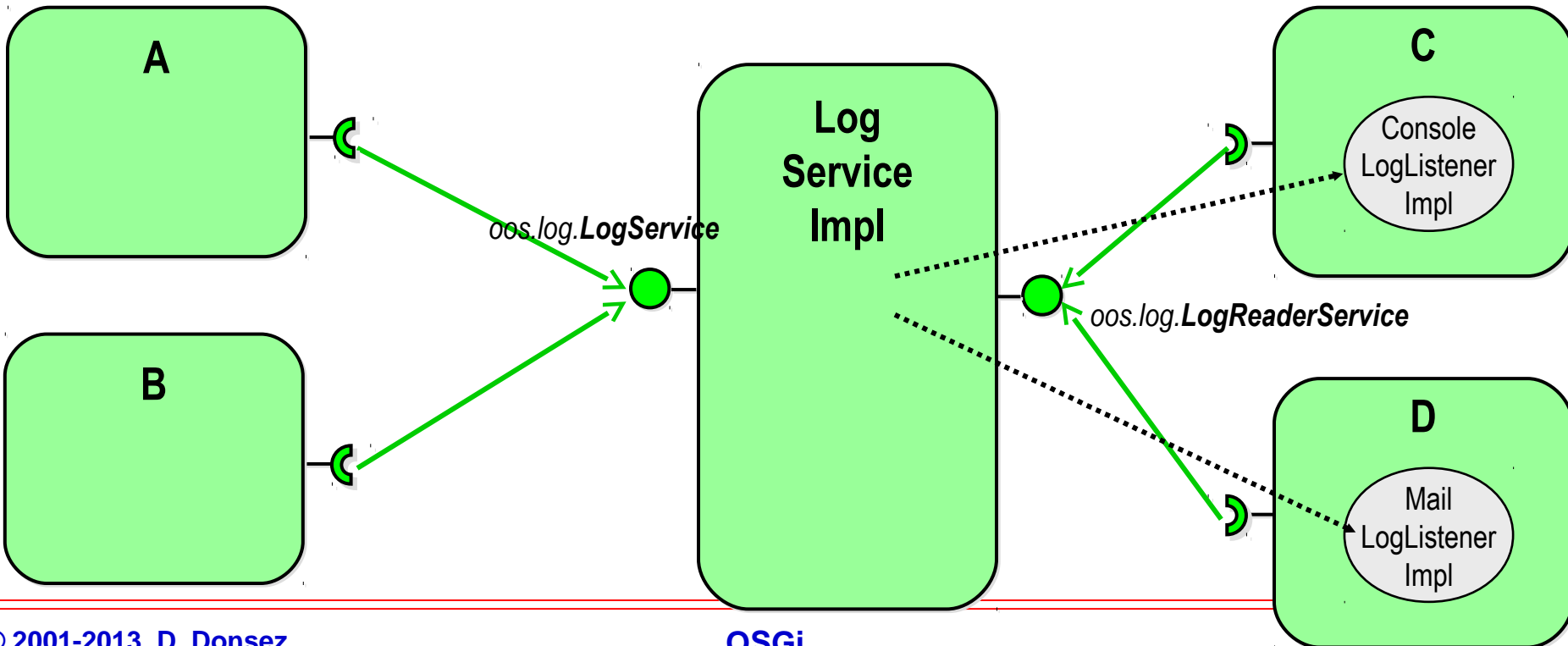
- ⑦ <http://www.wicket-wiki.org.uk/wiki/index.php/OSGi>

LogService

⑦ Motivation

- ⑦ journaliser des traces/événements
- ⑦ se mettre en l'écoute de ces traces/événements
- ⑦ Rappel: `java.util.logging.Logger` est statique

⑦ Architecture



Exemple de LogListener

```
public class ConsoleLogListenerActivator implements BundleActivator {
    LogReaderService logReaderService;
    LogListener loglistener;
    public void start(BundleContext cxt) {
        logReaderService= ...;
        loglistener=new ConsoleLogListenerImpl();
        logReaderService.addLogListener(loglistener);
    }
    public void stop(BundleContext cxt) {
        logReaderService.removeLogListener(loglistener);
    }
}
class ConsoleLogListenerImpl implements org.osgi.service.log.LogListener {
    public final void logged(LogEntry entry) {
        System.out.println("Level:"+entry.getLevel());
        if( entry.getBundle() != null) {
            System.out.println("bundle : "+ entry.getBundle().getBundleId()+" ");
        }
        System.out.println(entry.getMessage());
    }
}
```


Device Manager

⑦ Motivations

- ⑦ Faire apparaître les drivers des périphériques matériels comme des Services OSGi
- ⑦ Charger les drivers grâce aux bundles
- ⑦ Mise à jour des drivers
- ⑦ Un driver fournit plusieurs services plus ou moins raffinés
- ⑦ Plug-and-Play
 - ⑦ Le branchement d'un périphérique provoque l'enregistrement d'un service.
 - ⑦ Le retrait du périphérique provoque le désenregistrement du service

⑦ Plusieurs éléments

- ⑦ DeviceService
- ⑦ Driver
- ⑦ DriverLocator
- ⑦ DeviceManager

Device Manager

⑦ Moteur de raffinement des devices

- ⑦ A l'arrivée d'un nouveau service Device, le Device Manager cherche à enregistrer d'autres Device de plus haut niveau.

⑦ Services utilisés

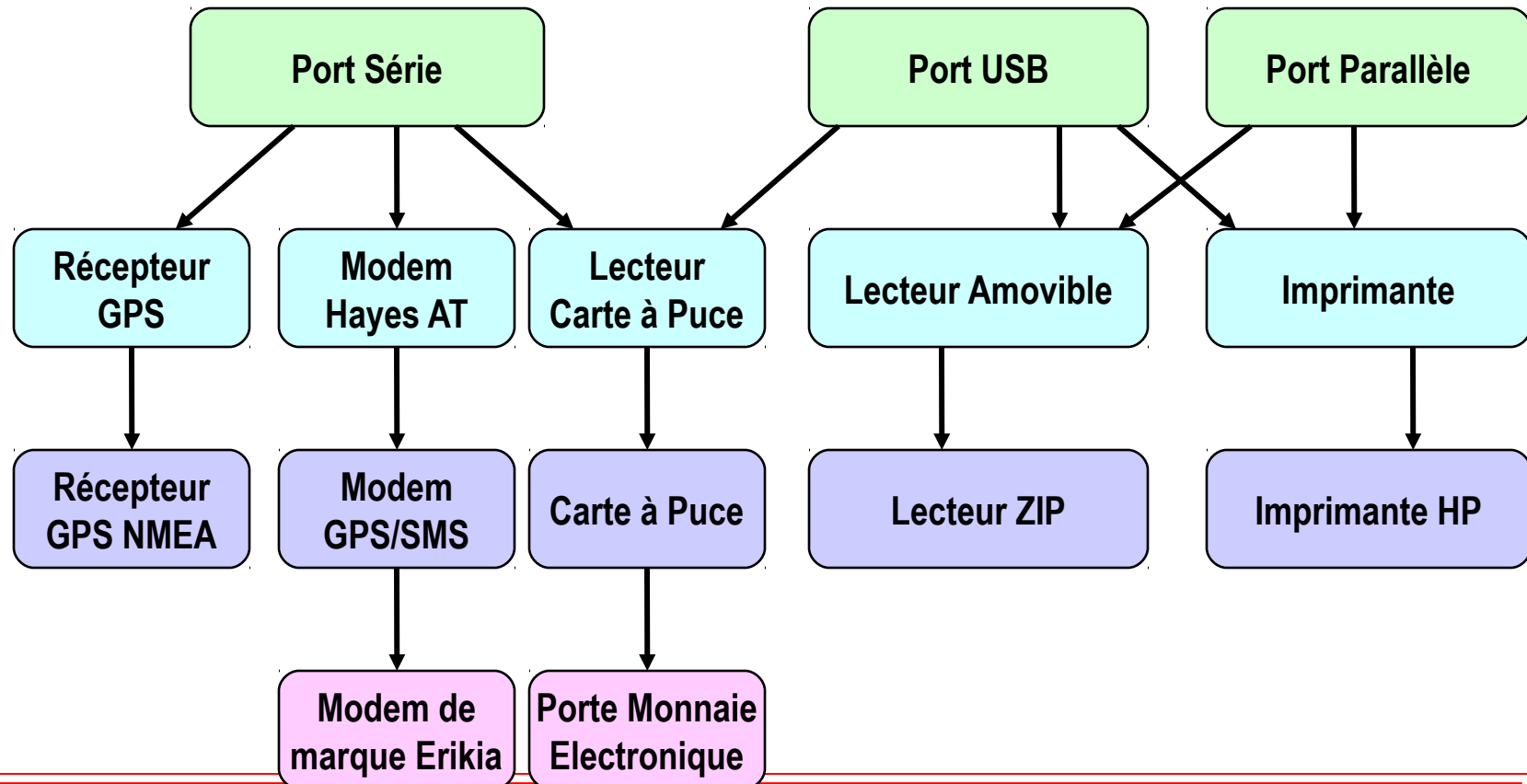
- ⑦ DriverLocator
- ⑦ Driver

Device Manager

⑦ Device

⑦ Gestion d'un périphérique

R
a
f
i
n
e
m
e
n
t



Device Manager

⑦ DriverLocator

⑦ Permet d'associer un Device à Driver

⑦ Device → Id

⑦ Id → URL

⑦ Utiliser par le DeviceManager

⑦ Exemple d'implementation

⑦ Filtre LDAP → Id → URL

⑦ Driver

⑦ Vérifie si l'on peut installer le Device associé à ce Driver

⑦ Retourne une valeur d'évaluation de l'association

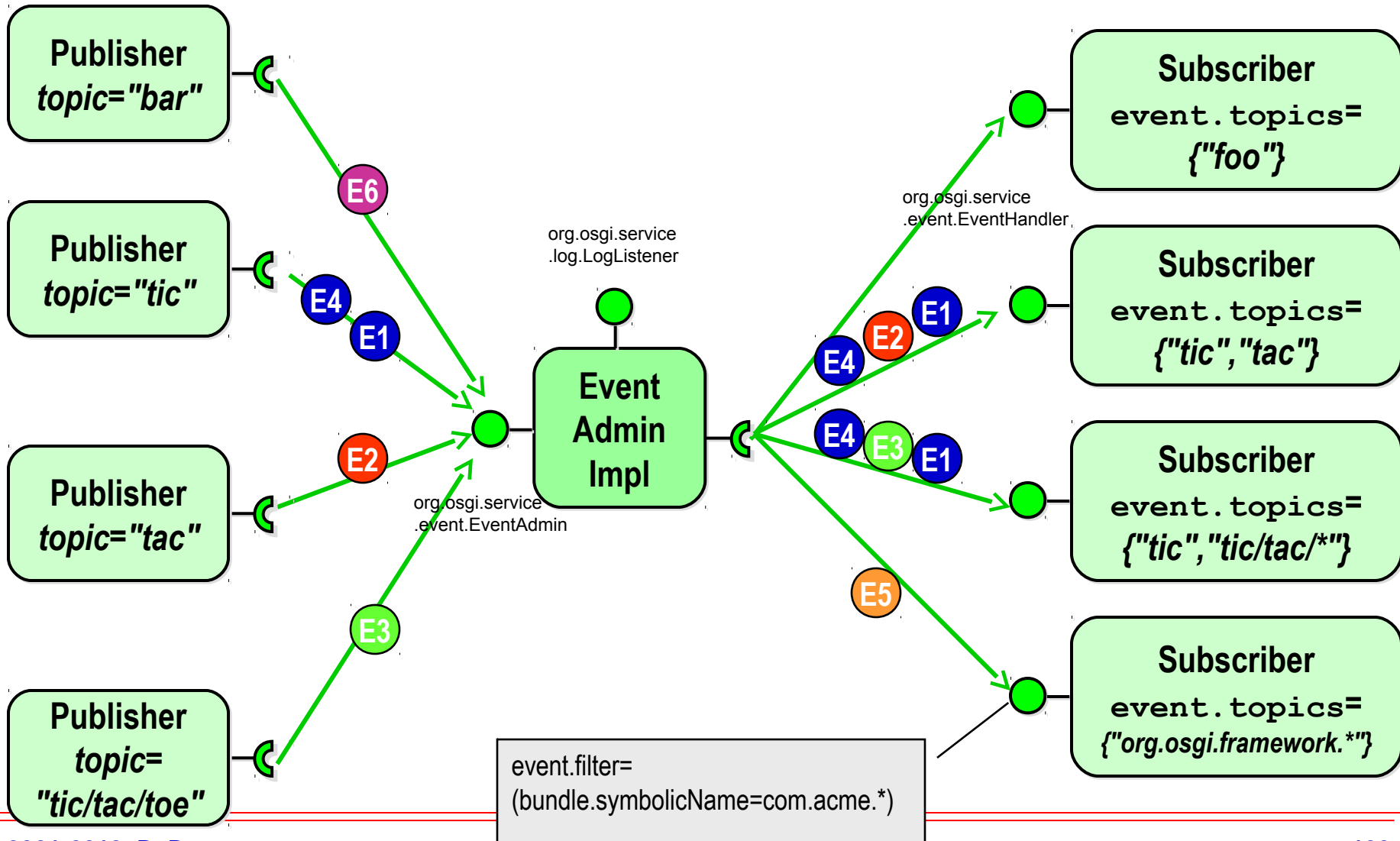
⑦ Non possible si =0

⑦ Implémente une méthode d'attachement.

Event Admin Service (i) r4

- ⑦ Offre un modèle de communication événementiel entre les bundles.
- ⑦ Objet `Event` = *topic* + propriétés.
- ⑦ **Médiateur de Publication-souscription d'événement**
 - ⑦ L'éditeur poste un événement au service `EventAdmin`
 - ⑦ L'`EventAdmin` le diffuse en parallèle à tous les souscripteurs du *topic*.
 - ⑦ Chaque souscripteur enregistre un service `EventHandler`.
 - ⑦ L'éditeur peut être synchronisé (ou non) sur la terminaison des exécutions // de tous les services `EventHandler` concernés.
- ⑦ **Remarque**
 - ⑦ Événements spéciaux liées
 - ⑦ au cycles de vie des Services, bundles et framework
 - ⑦ au `LogService`, `UPnP Base Driver`, ...
 - ⑦ Le service `EventAdmin` gère une *liste noire* des `EventHandler` défectueux ou consommant trop de CPU.

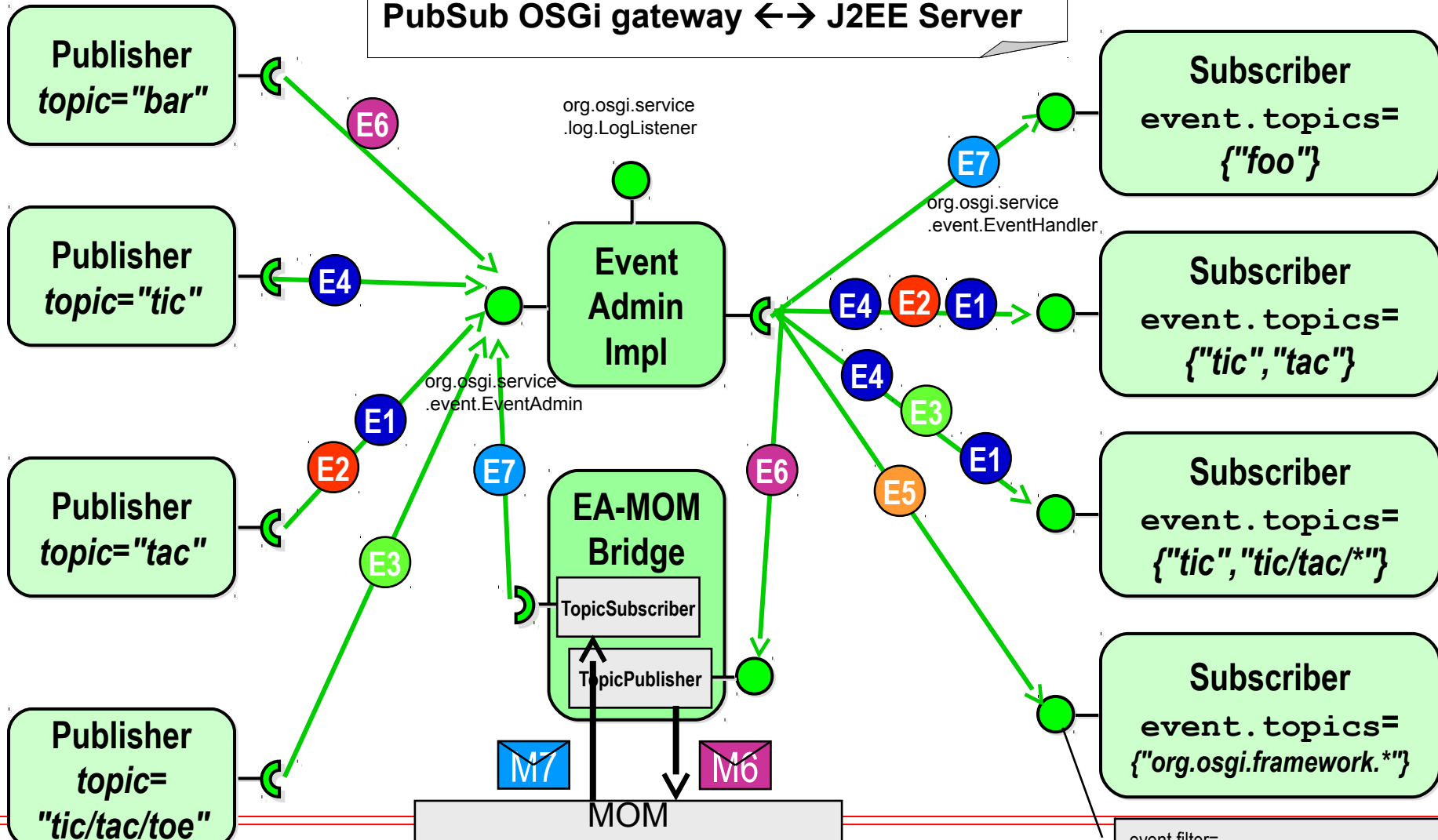
Event Admin Service (ii)



Bridging Event Admin Service and MOM

PubSub inter-gateways

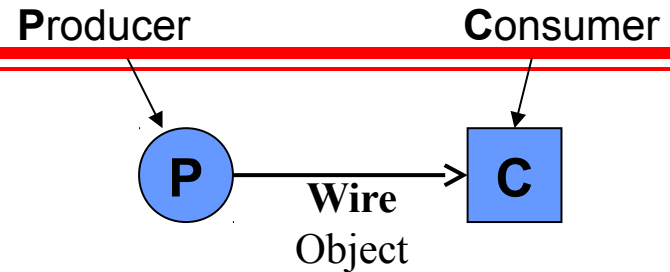
PubSub OSGi gateway ↔ J2EE Server



event.filter=
(bundle.symbolicName=com.acme.*)

Wire Admin Service

R3



⑦ Motivation

⑦ Patron (*design pattern*)

de services producteurs-consommateurs de données

⑦ Données : mesure physique, position, état (discret), ...

⑦ Domaine d'application

⑦ Services basés Capteurs

⑦ Machine-to-Machine (M2M)

⑦ WireAdmin

⑦ Médiateur reliant 0..N producteurs à 0..M consommateurs

⑦ Administrable globalement

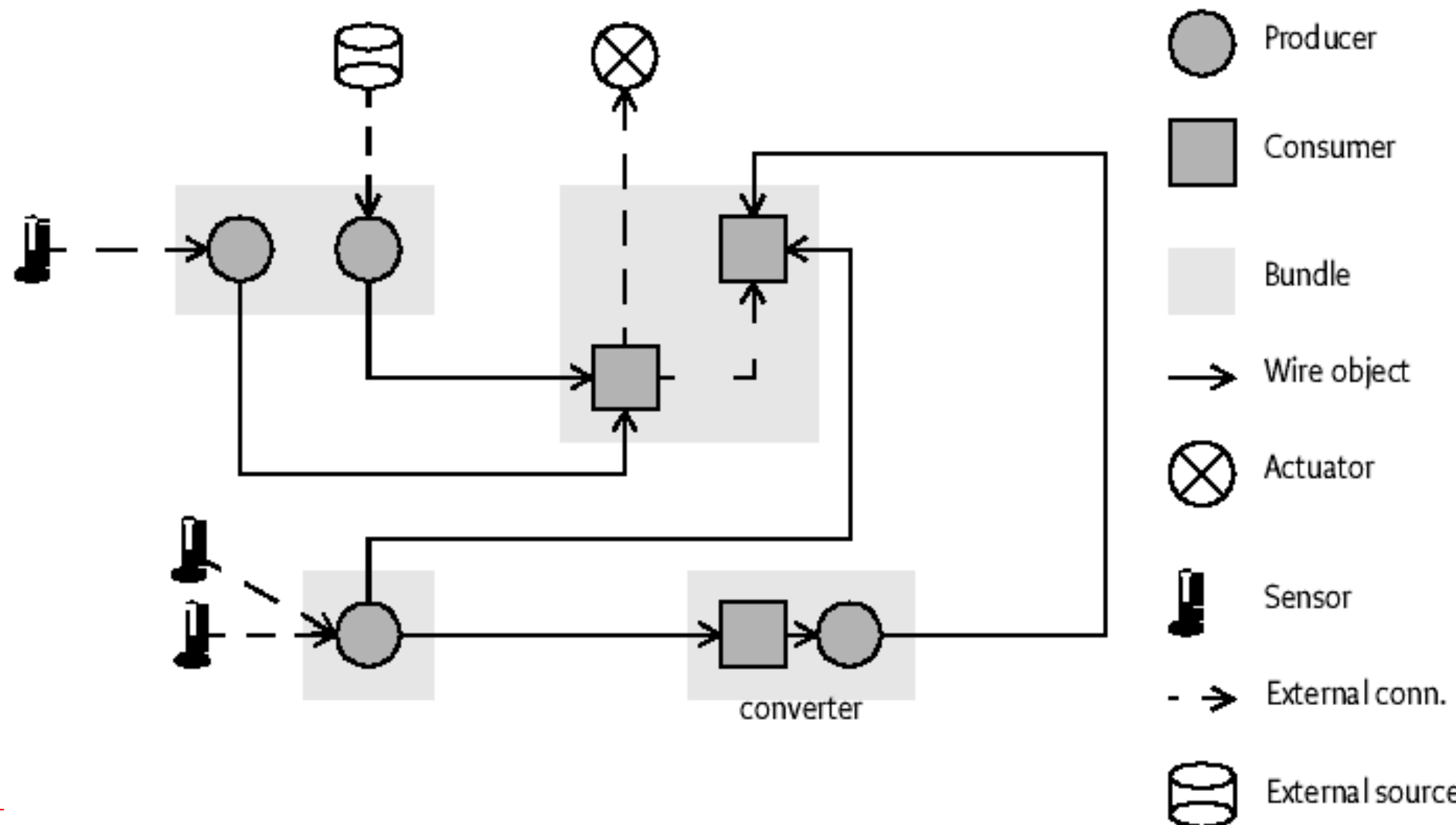
⑦ WireAdmin, WireAdminListener

⑦ Contrôle de la « comptabilité » et adaptation de types de données échangées au travers du Wire

⑦ Flavors

Wire Admin Service

Patron *Producer-Wire-Consumer*

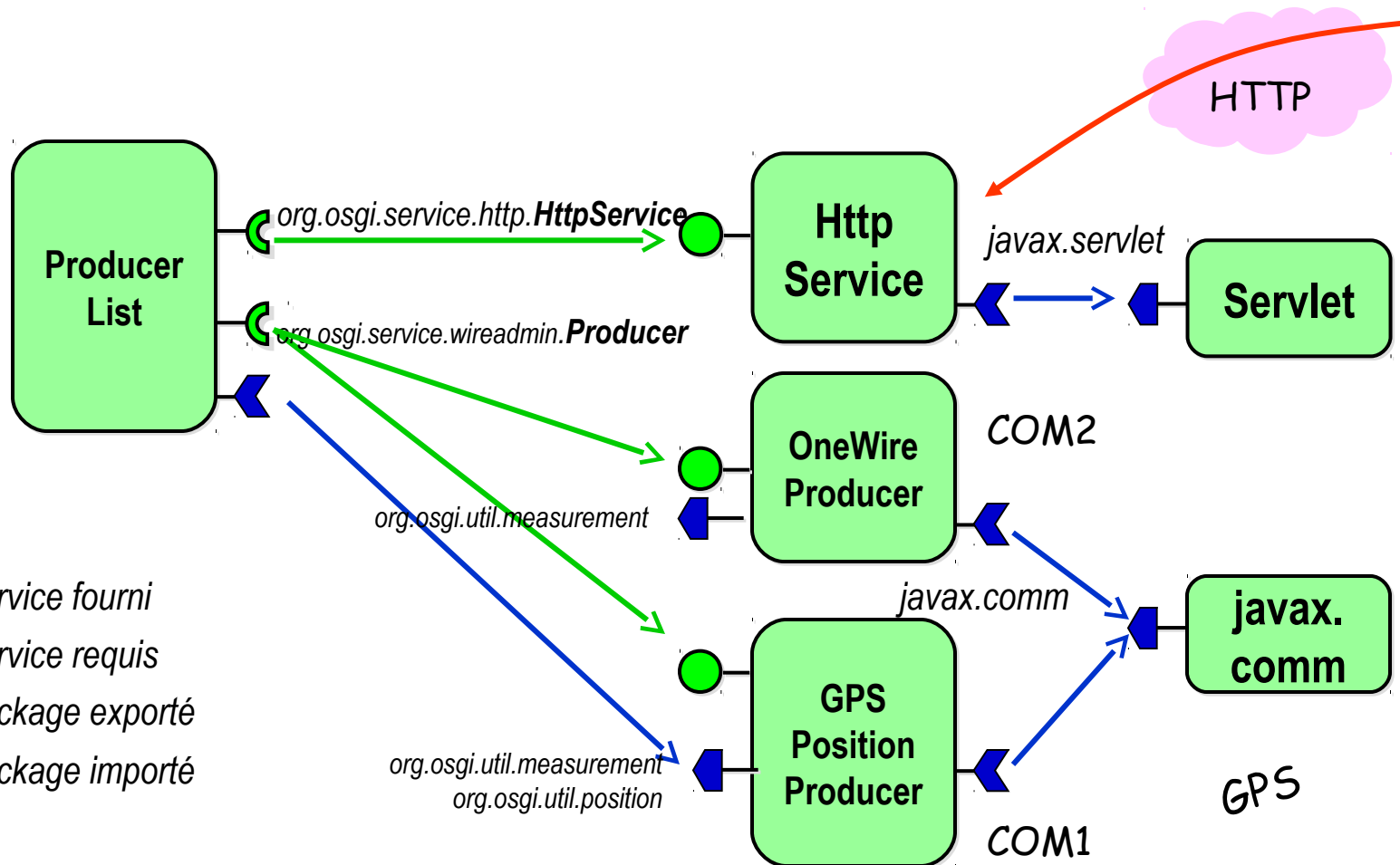


Wire Admin Service

Exemple d'application M2M (i)

⑦ Consultation de mesures via le Web

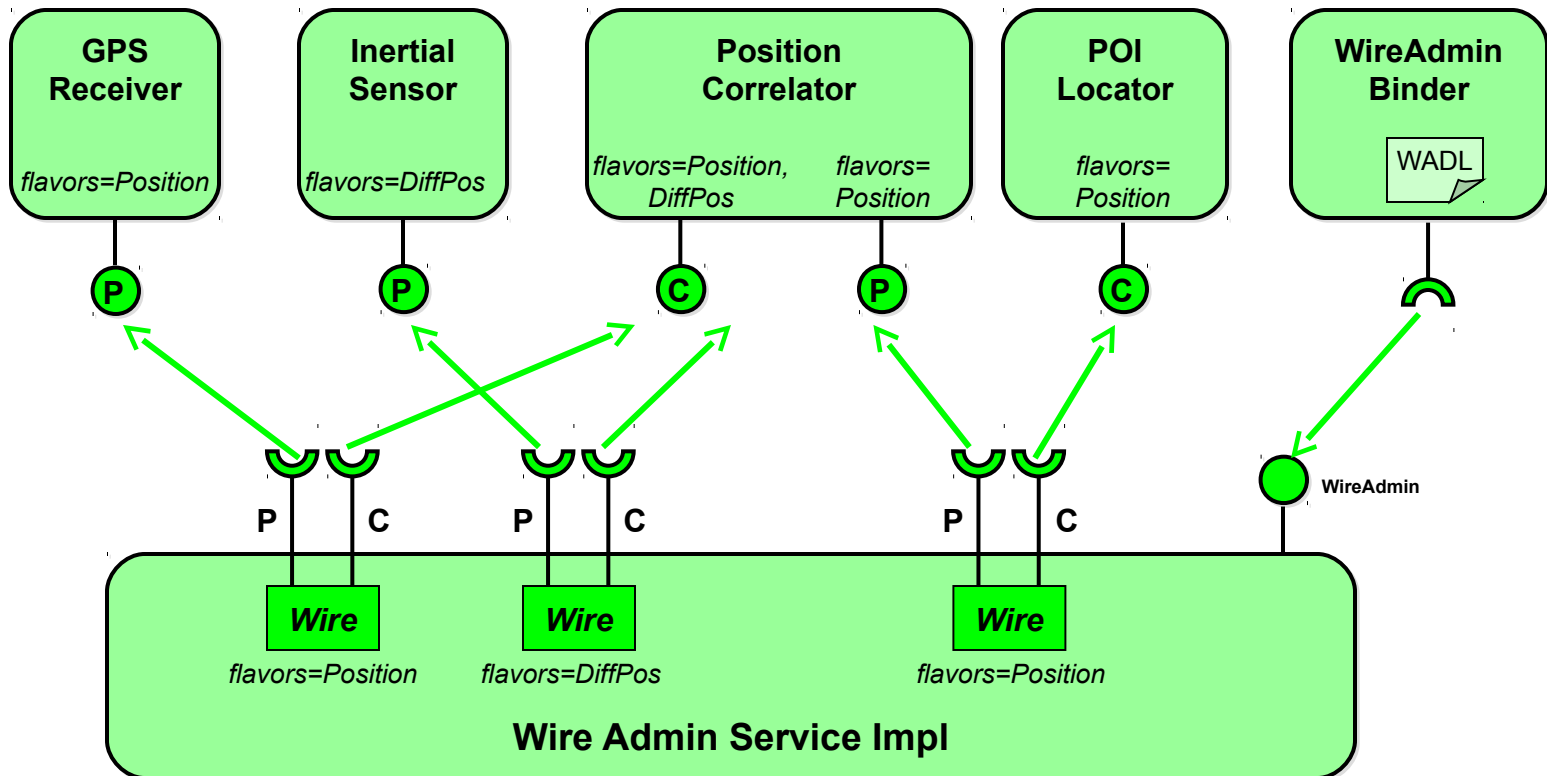
⑦ position GPS + température



Wire Admin Service

Exemple d'application

⑦ Aide à la navigation



UPnP Driver Service

⑦ UPnP (Universal Plug and Play)

- ⑦ Protocoles de découverte d'équipements (SOHO) et d'utilisation leur services

- ⑦ Basé sur SOAP/HTTP (TCP, UDP, UDP Multicast)

- ⑦ Alternative à JINI

- ⑦ Largement répandu et soutenu par les équipementiers SOHO



⑦ Motivations du service UPnP Driver Service

- ⑦ **Spécifie comment des bundles OSGi doivent être développés pour interopérer avec**

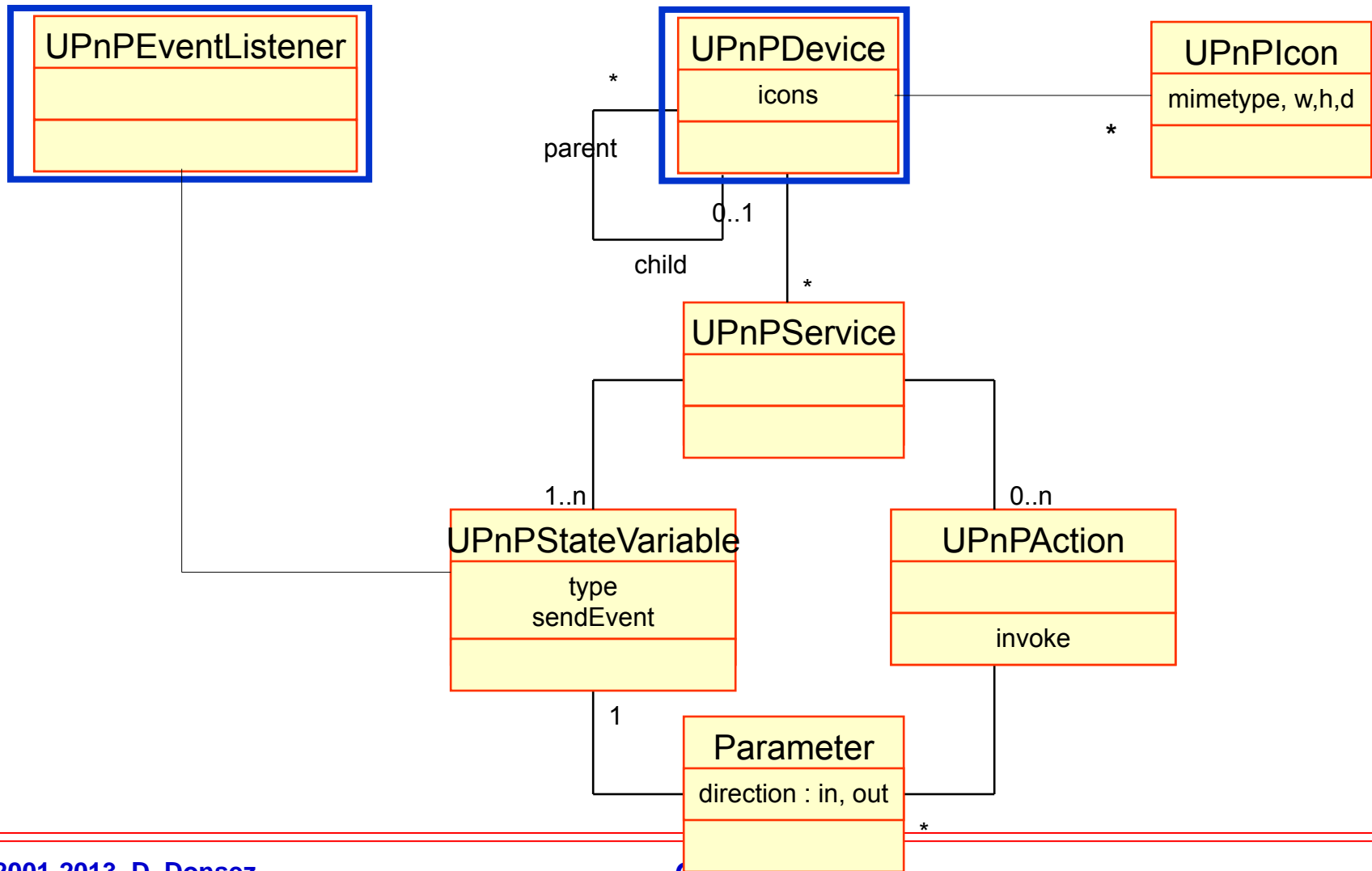
- ⑦ **des équipements UPnP (devices)**

- ⑦ **des points de contrôle UPnP (control points)**

- ⑦ **en respectant la spécification UPnP Device Architecture**

UPnP Driver Service

Les interfaces représentant les équipements UPnP



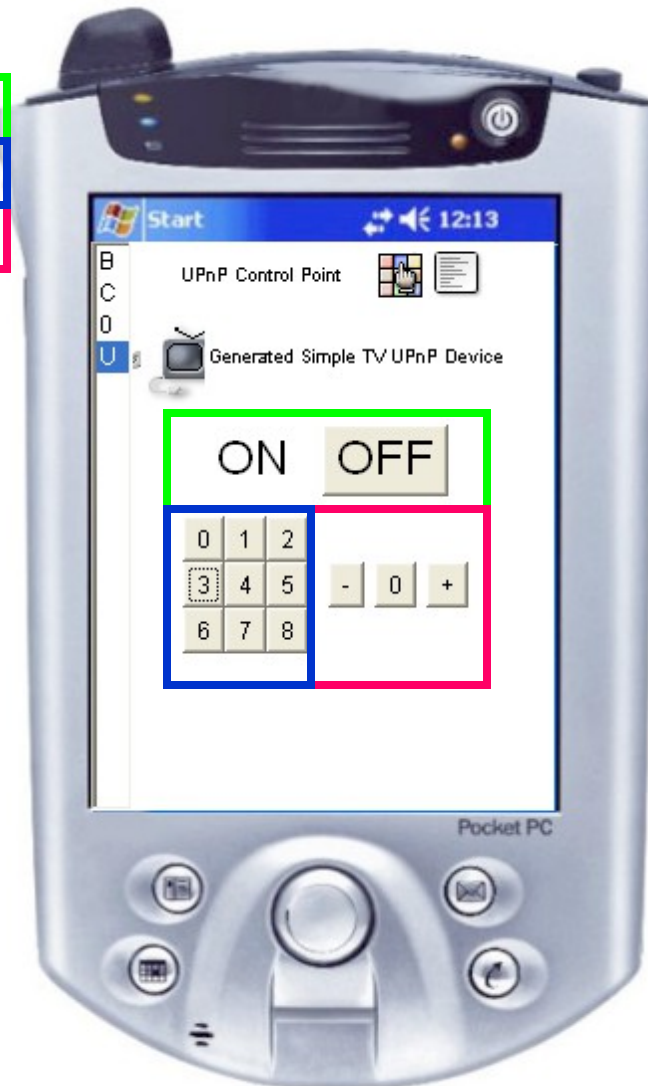
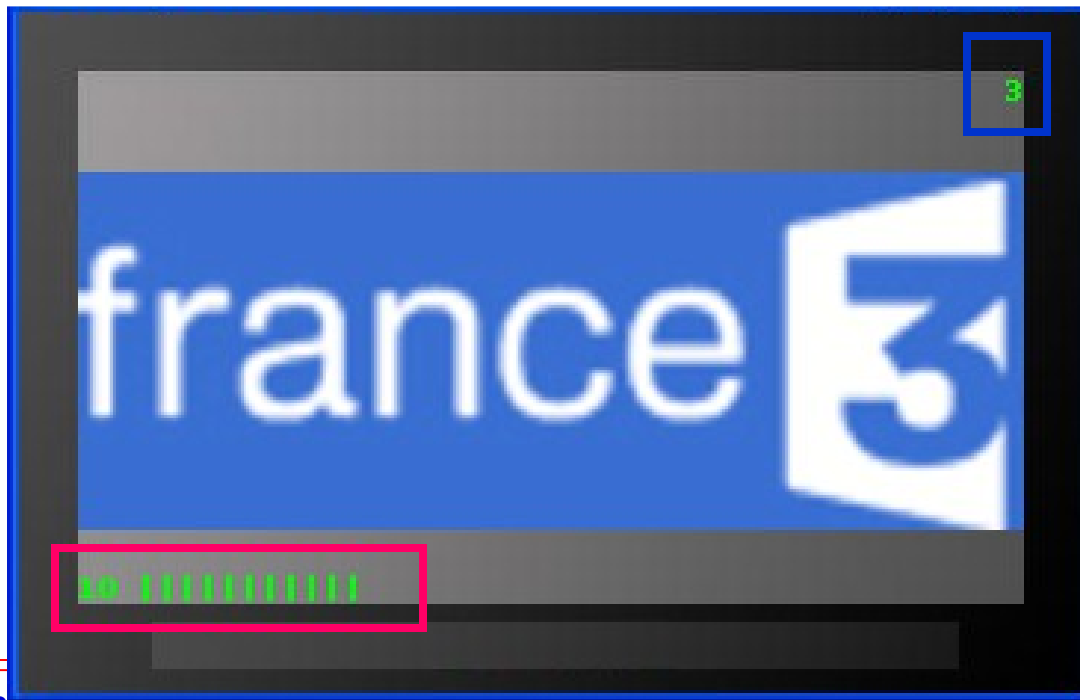
Exemple: un *device* Téléviseur et son point de contrôle

⑦ 3 services

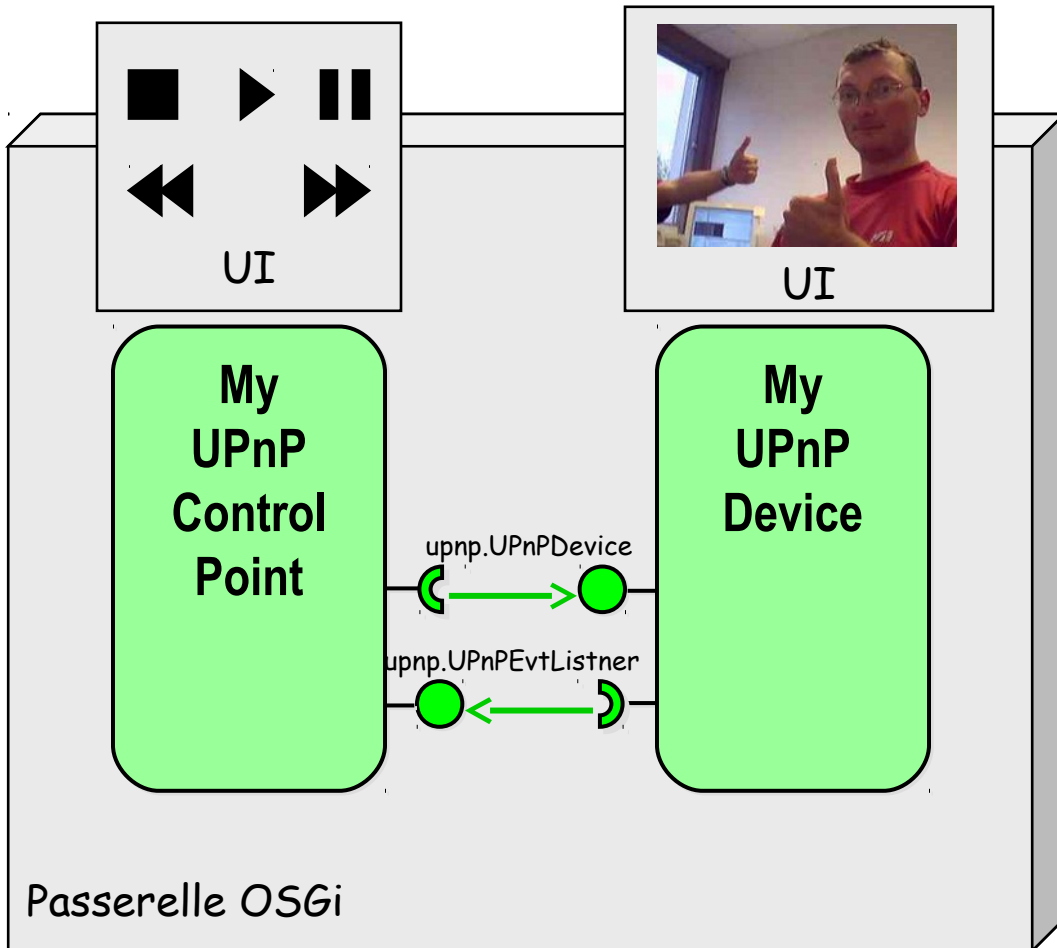
⑦ `urn:schemas-upnp-org:service:SwitchPower:1`

⑦ `urn:schemas-adele-imag-fr:service:ChannelSelector:1`

⑦ `urn:schemas-adele-imag-fr:service:VolumeSelector:1`

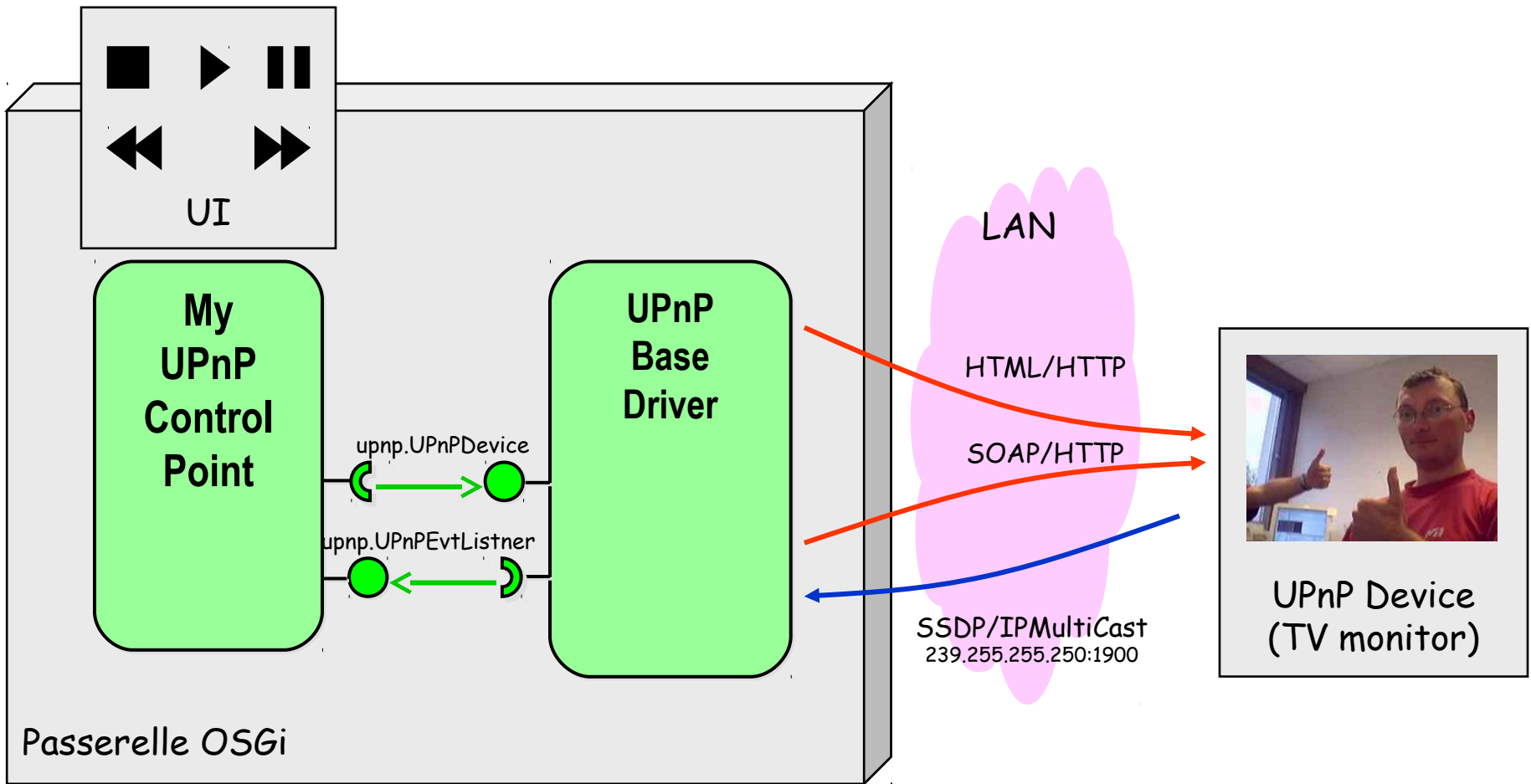


UPnP Device Driver : Mise en œuvre Collocalisé

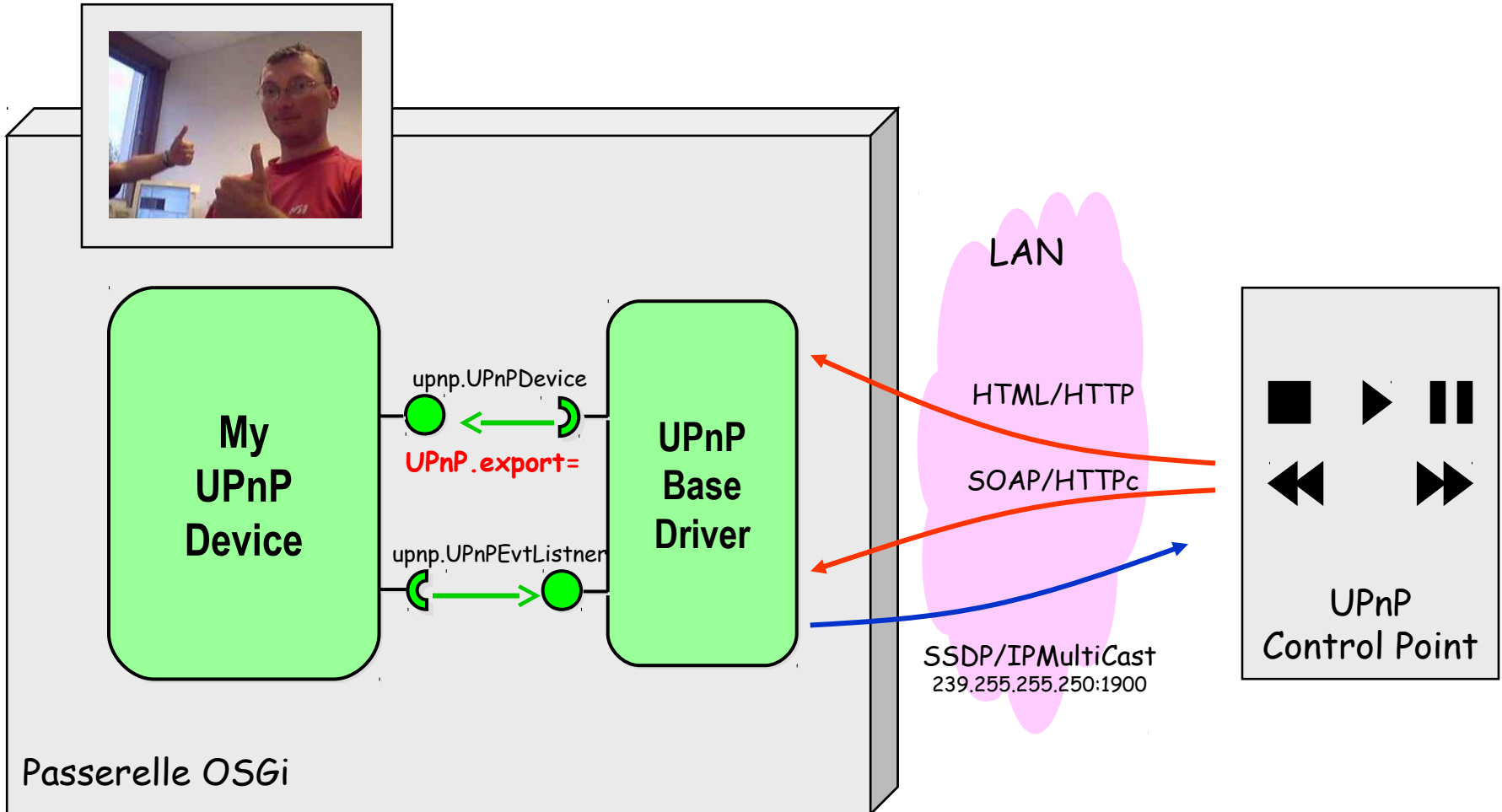


UPnP Device Driver : Mise en œuvre

Point de contrôle

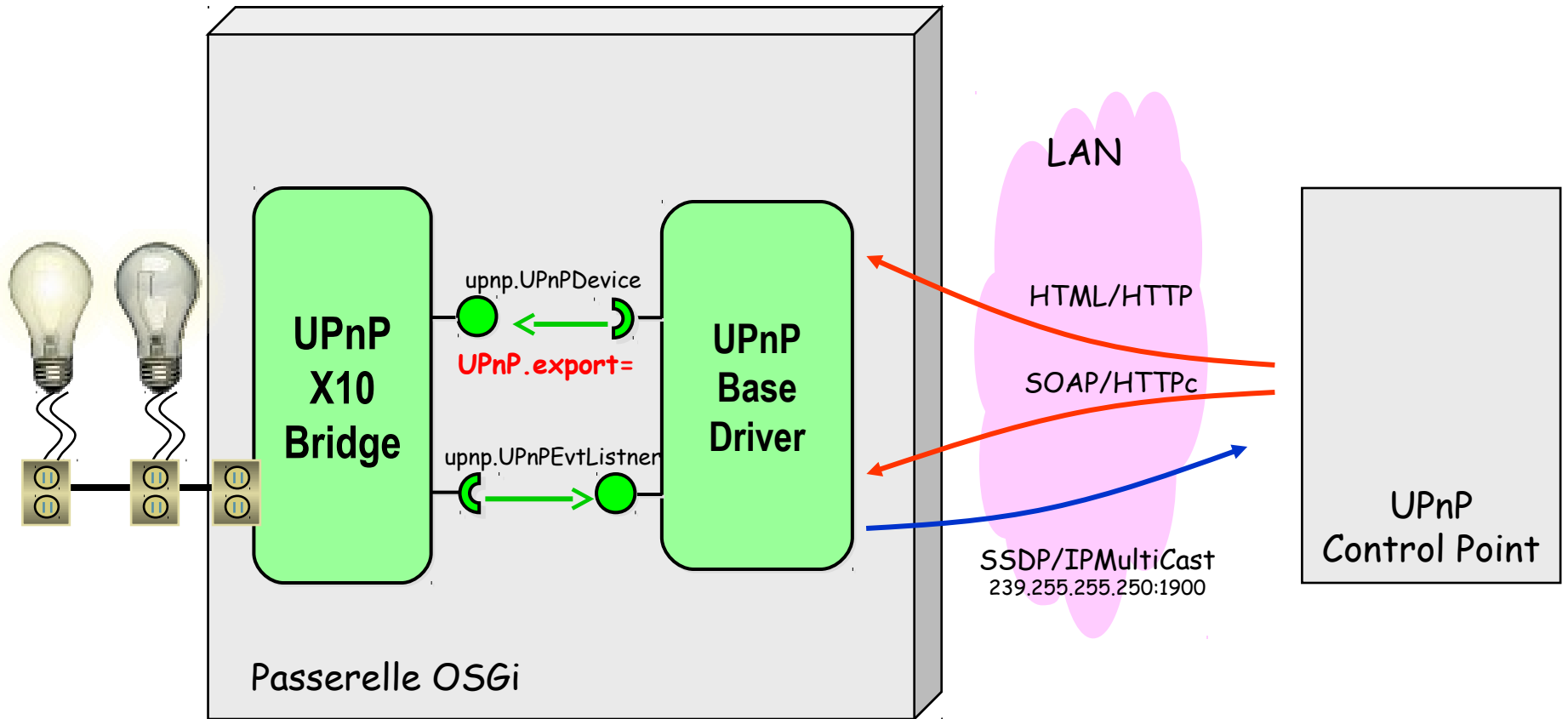


UPnP Device Driver : Mise en œuvre Equipement



UPnP Device Driver : Mise en œuvre

Passerelle micro-monde



Execution Environment Specification

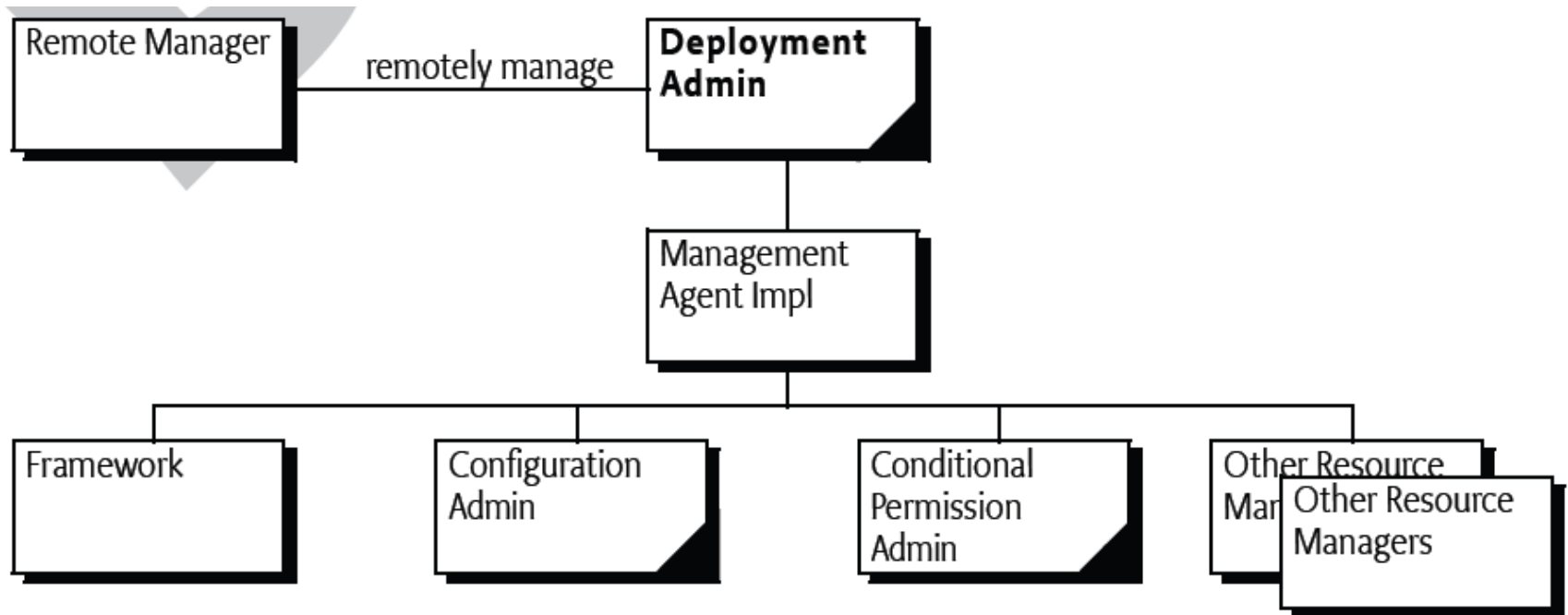
⑦ Définit l'environnement d'exécution (java) requis pour que le bundle puisse s'exécuter

- ⑦ Il doit être inclut dans la liste des environnements du framework d'accueil pour passer dans l'état RESOLVED
- ⑦ **Propriété** `org.osgi.framework.executionenvironment`

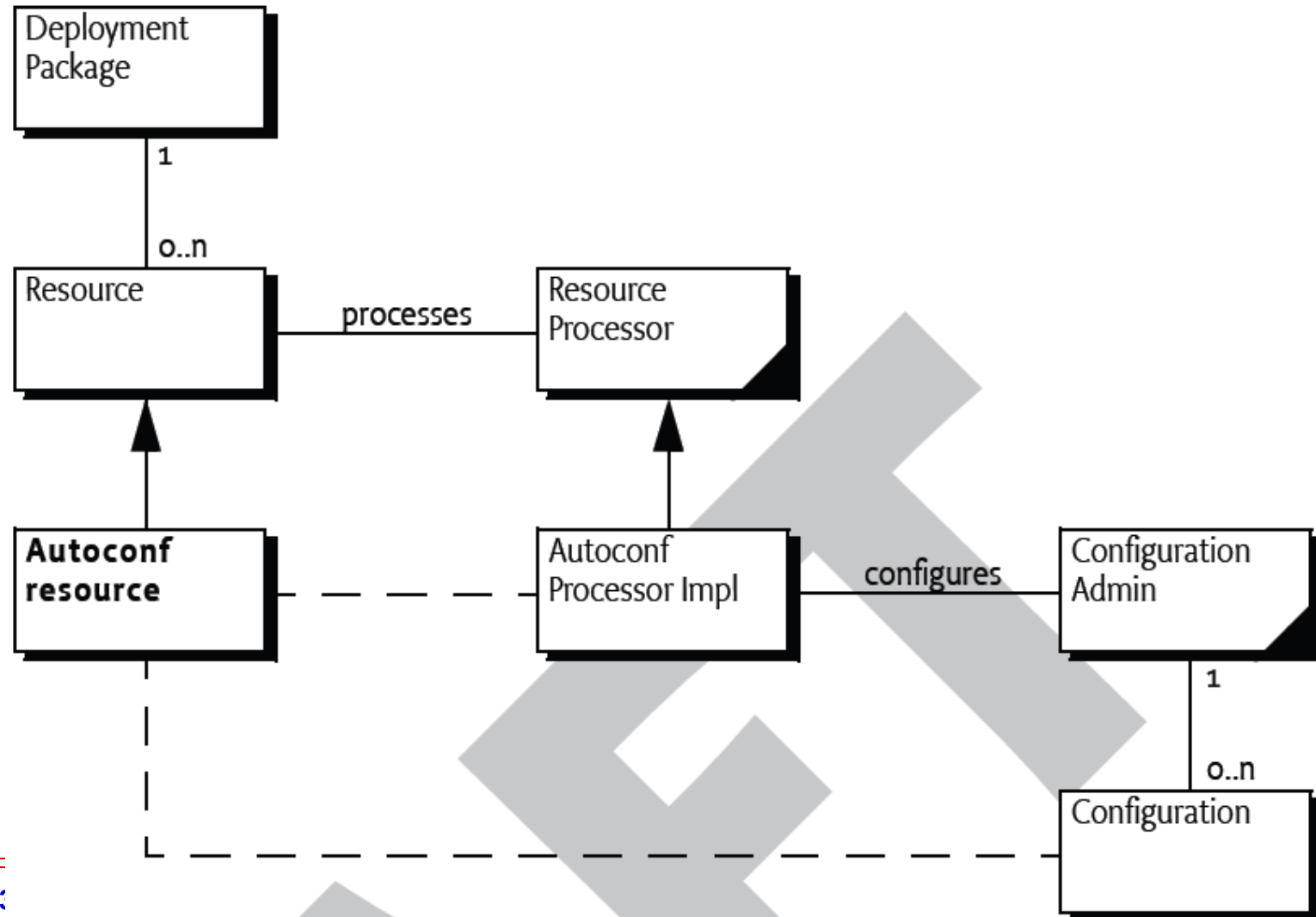
⑦ Entête de Manifest

```
Bundle-RequiredExecutionEnvironment: \  
    CDC-1.0/Foundation-1.0, OSGi/Minimum-1.0
```

114 Deployment Admin Specification

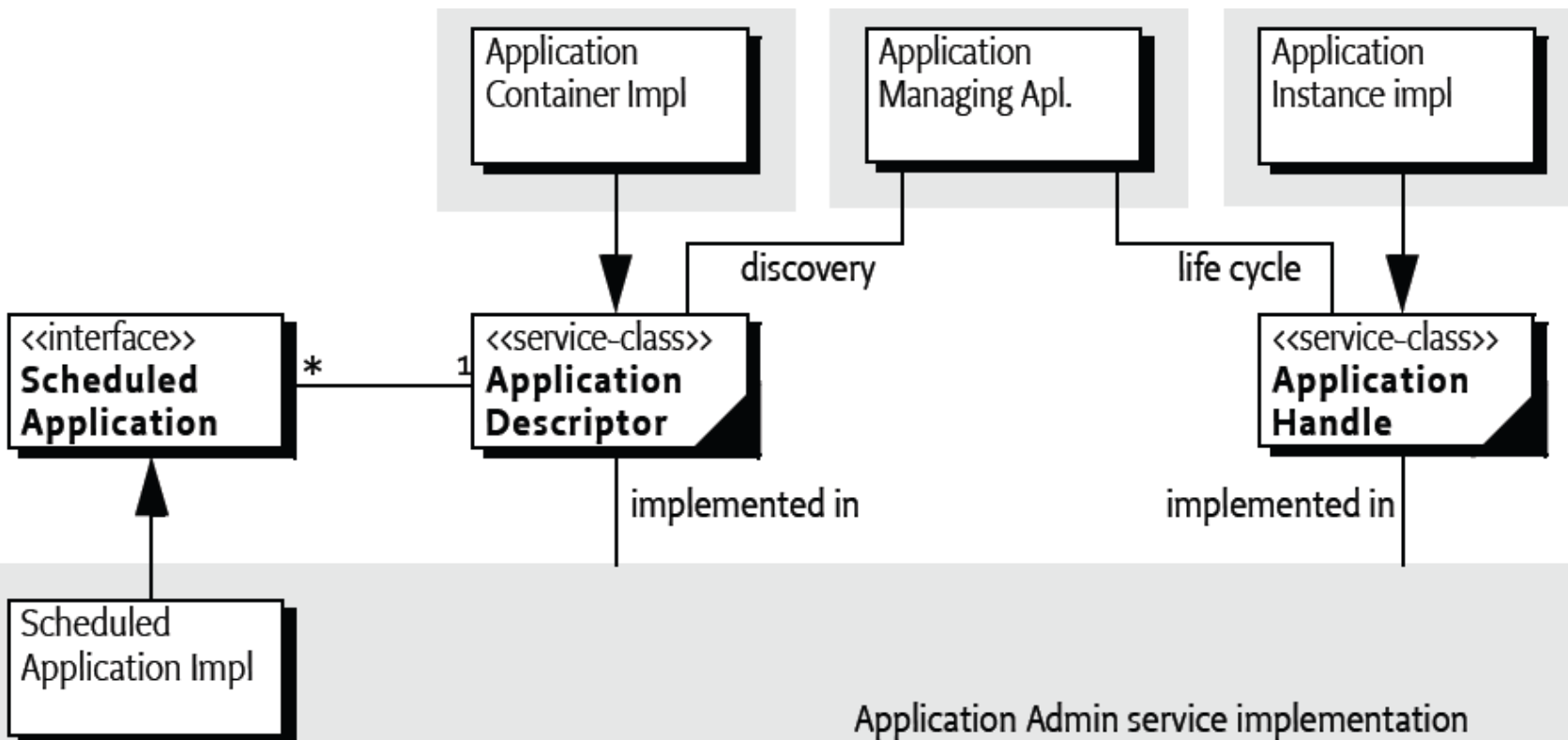


115 Auto Configuration Specification



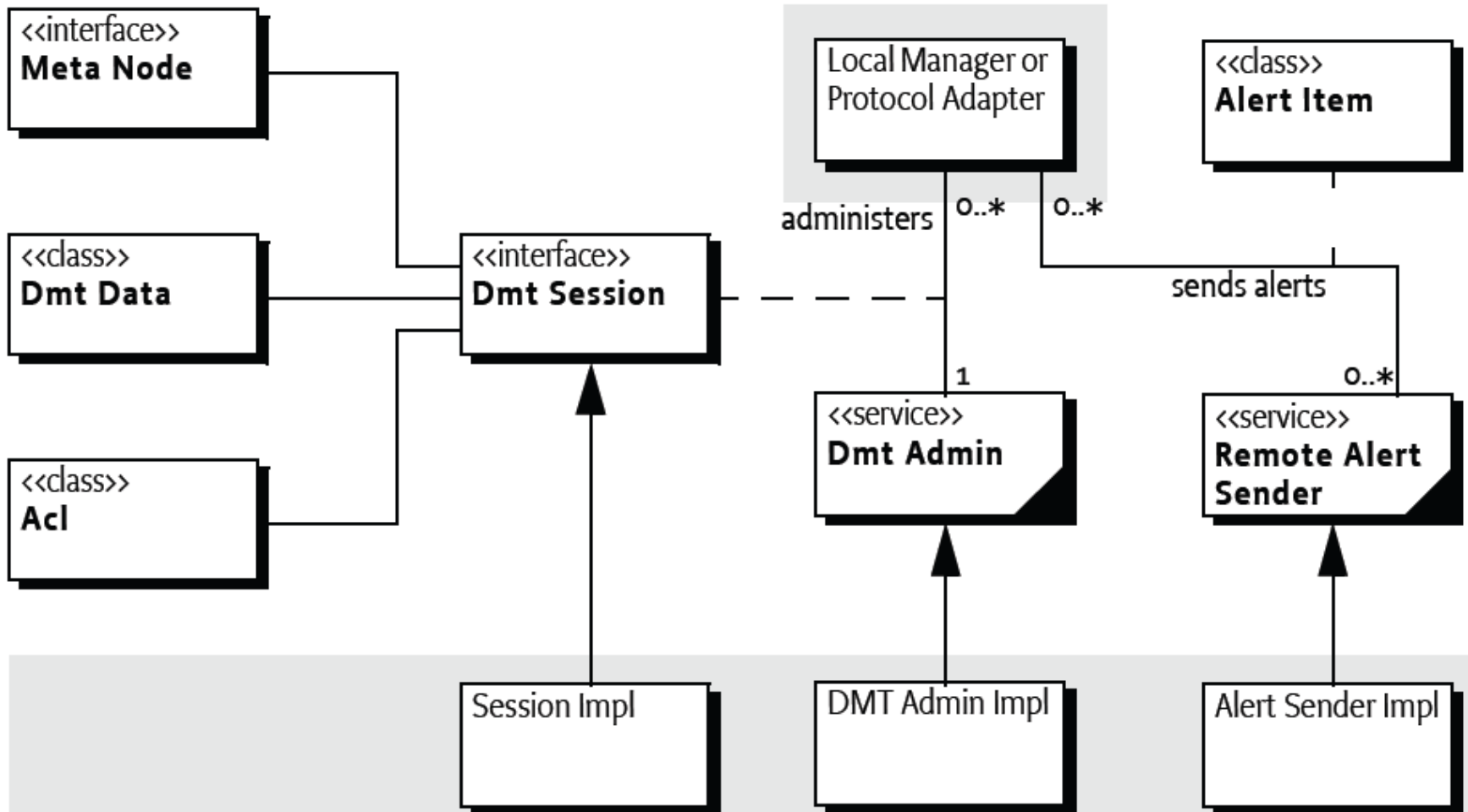
116 Application Admin Service Specification

⑦ *org.osgi.service.application*



117 DMT Admin Service Specification

⑦ *ora.osai.service.dmt*

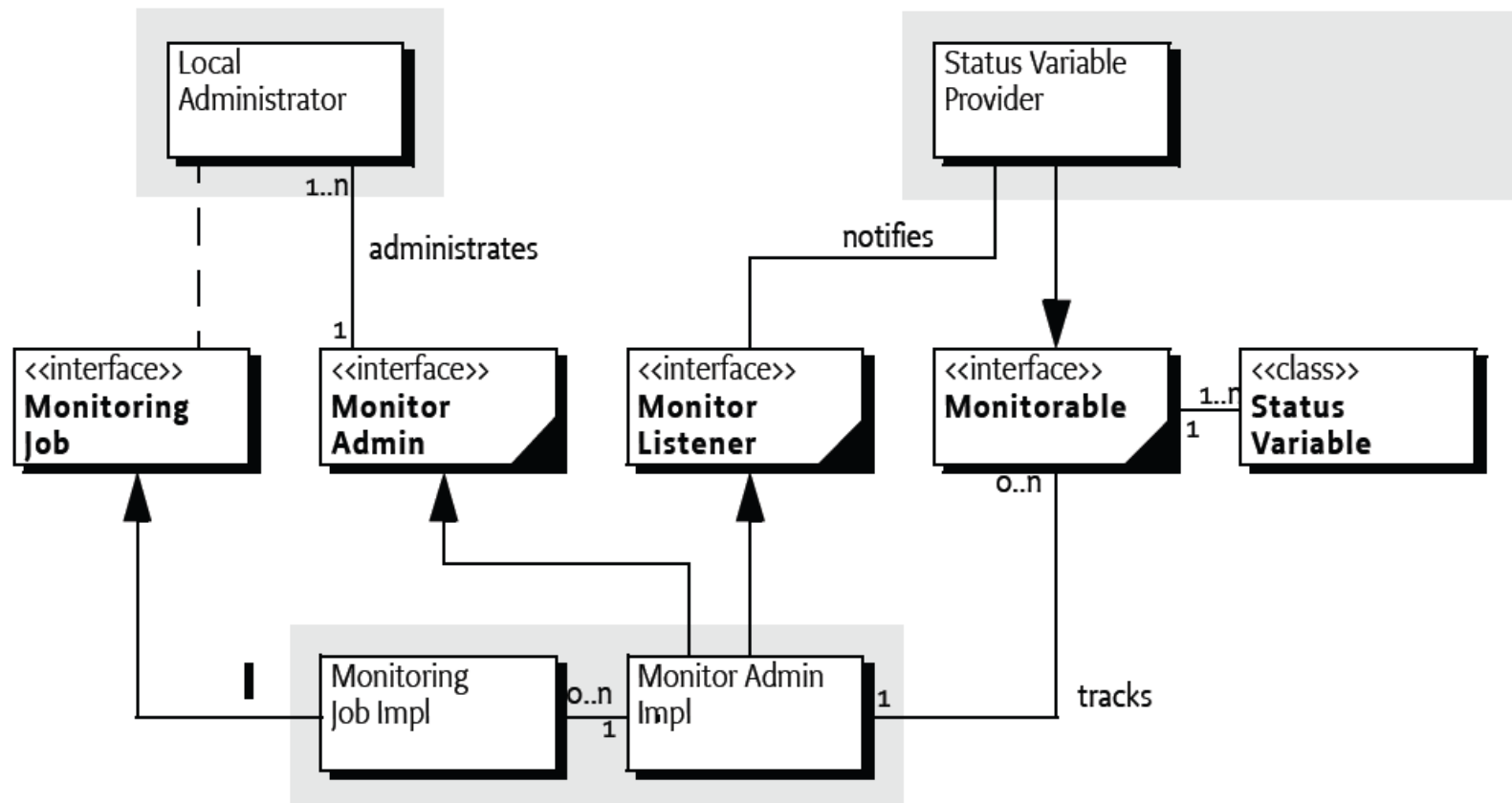


119 Monitor Admin Service Specification

org.osgi.service.monitor

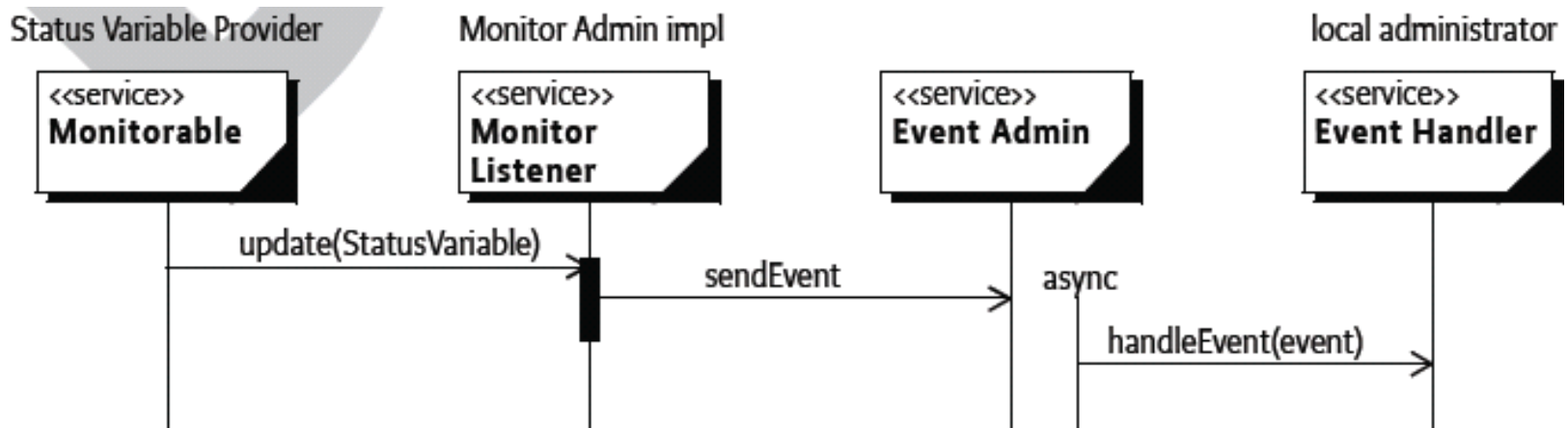
⑦ Motivation

- ⑦ Découvrir des variables d'état et publier/écouter leurs modifications
 - ⑦ mémoire disponible, niveau d'énergie de la batterie, nombre de SMS envoyés ...
- ⑦ S'appuie sur l'Event Admin



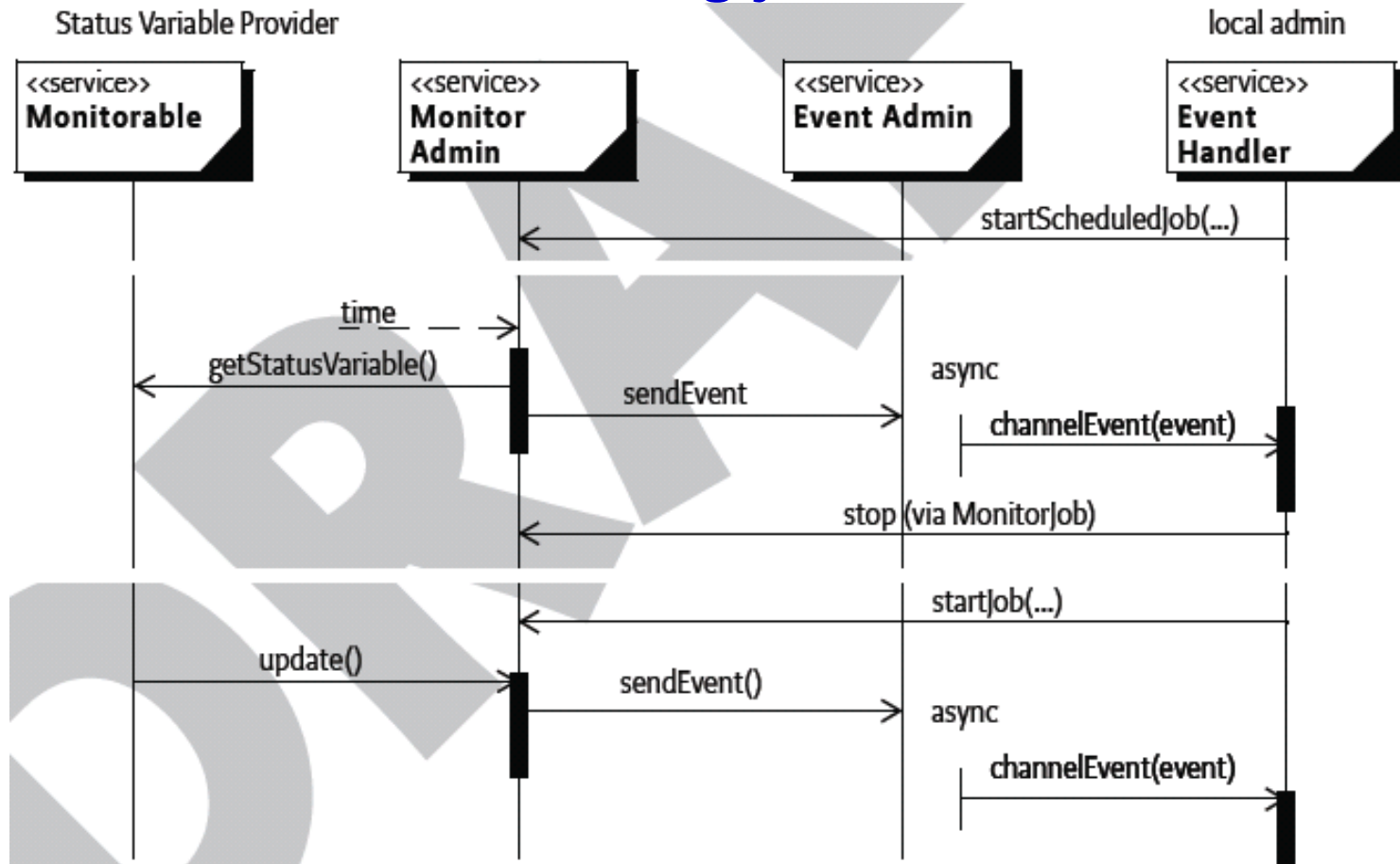
119 Monitor Admin Service Specification

⑦ Notification sur changement



119 Monitor Admin Service Specification

⑦ Time and Event monitoring job



End-to-End Management

⑦ Motivations

- ⑦ Gestion d'un (très grand) parc de plateformes OSGi
- ⑦ Et des équipements qui y sont attachés

⑦ Besoins

- ⑦ Déploiement « transactionnel »
- ⑦ Politiques de déploiement
- ⑦ ...

⑦ E2E Remote Management EG

- ⑦ Expert group a l'Alliance

⑦ Produits

- ⑦ Prosyst Remote Manager
- ⑦ IBM Expeditor Framework
- ⑦ ...

Enterprise-side OSGi (EEG)

⑦ **Eric Newcomer « OSGi could not be a YAJEEC »**

⑦ **Deal with legacy technologies**

⑦ **CORBA, MQ, CICS, Tuxedo, JEE, .NET**

⑦ **Combining with the next technologies**

⑦ **Spring-OSGi, SCA-OSGi**

JavaEE and OSGi

⑦ Motivations

- ⑦ EEG (Enterprise Expert Group)
- ⑦ Modular and Dynamic JavaEE platforms
 - ⑦ Minimize downtime when update
 - ⑦ Incremental delivery and update
 - ⑦ Modular JavaEE Artifact (Webapp = set of WABs)
 - ⑦ Just enough JavaEE platforms
- ⑦ Hybrid JavaEE and OSGi applications
 - ⑦ Persistence and Transaction (JTA) in OSGi bundles
 - ⑦ JavaEE artifacts importing and exporting OSGi services

⑦ JavaEE platforms

- ⑦ JOnAS 5 (the very first one), Oracle (BEA) Weblogic, Glassfish 3, Geronimo, Websphere, Jboss, ...
- ⑦ PeerGreen (hybrid apps), Glassfish Fighterfish (hybrid apps)

Distributed OSGi

⑦ Motivations

- ⑦ Expose/Provides local services to remote consumers
- ⑦ Consumes remote services

⑦ Proposition

- ⑦ Extended Service Binder
- ⑦ R-OSGi
- ⑦ SCA/OSGi
- ⑦ RFC 119

⑦ <http://www.osgi.org/download/osgi-4.2-early-draft3.pdf>.

⑦ RFC 119

- ⑦ <http://cxf.apache.org/distributed-osgi.html>
 - ⑦ provides the Reference Implementation of the Distribution Software (DSW) component of the Distributed OSGi Specification
 - ⑦ using Web Services, leveraging SOAP over HTTP and exposing the Web Service over a WSDL contract.

OSGi and JMX

⑦ **MOSGi**

⑦ **RFC 139**

OSGi ME

- ⑦ **Version d'OSGi ciblant des microcontrolleurs 32 bits ayant peu de mémoire (SRAM + FlashRAM)**
- ⑦ **TODO**

OSGi-Connect

- ⑦ **aka OSGi Lite**
- ⑦ **Version allégée d'OSGi ne conservant que le registre de services**

- ⑦ **Voir <http://code.google.com/p/pojosr>**
 - ⑦ **iPOJO et SCR fonctionnent sur POJOSR**

Conclusion intermédiaire

- + **Gestion des dépendances de package**
- + **Déploiement dynamique de bundles**
- + **Environnement varié:**
embarqué, station de travail, serveur.
- + **Fonctionnalité de base pour le déploiement de composants**

- **Programmation complexe des connexions entre services**
à la charge du développeur
- **Centralisé mais pas mal de travaux sur la distribution**

Q & R

What is OSGi Alliance ?



⑦ Consortium founded in March 1999

⑦ Objective

⑦ Create open specifications for delivering administrated services through a network

⑦ Define

⑦ A common platform (framework)

⑦ Services deployment

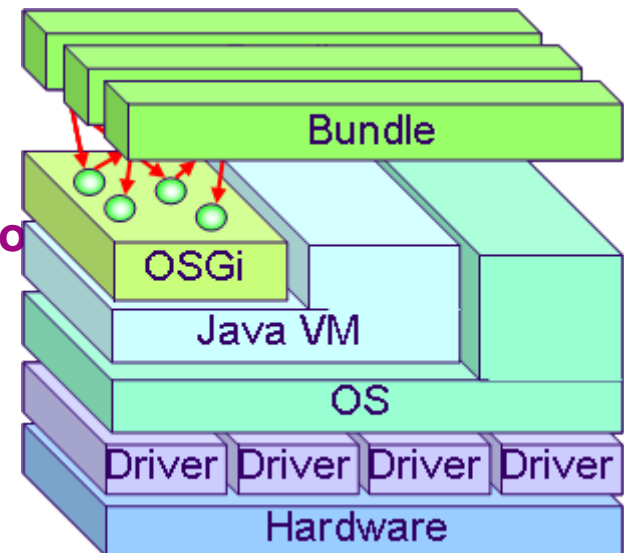
⑦ Services execution and administration

⑦ A set of based services:

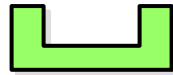
⑦ Log Service, Http Service...

⑦ A device access specification

⑦ A deployment unit, a *bundle*



OSGi Main Concepts



⑦ Framework:

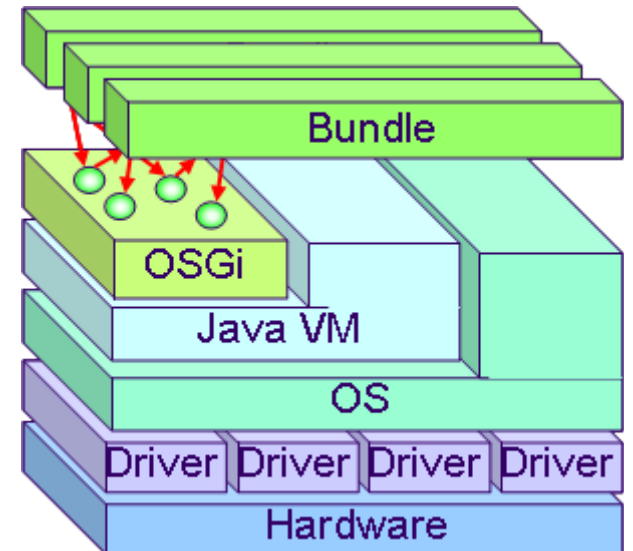
- ⑦ Bundles execution environment
 - ⑦ Oscar (Objectweb) / Felix (Apache), Knopperf sh, Equinox, SMF, ProSyst, Siemens VDO, ...
- ⑦ Event notification

⑦ Bundles:

- ⑦ Services diffusion and deployment unit

⑦ Services:

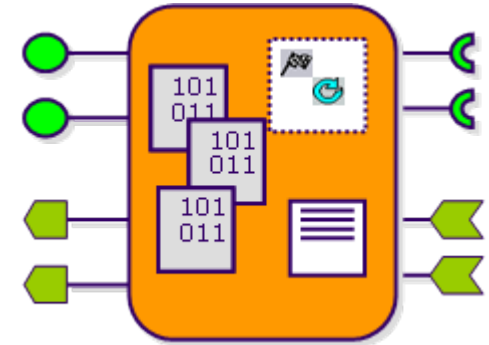
- ⑦ Java Object implementing a well def ne contract



Middleware and Application Packaging

⑦ Modularize the middleware/application

- ⑦ Distribute the different middleware services
- ⑦ Better component visibility
- ⑦ Need of a deployment container
- ⑦ Partial update without restart all



⑦ Implementation

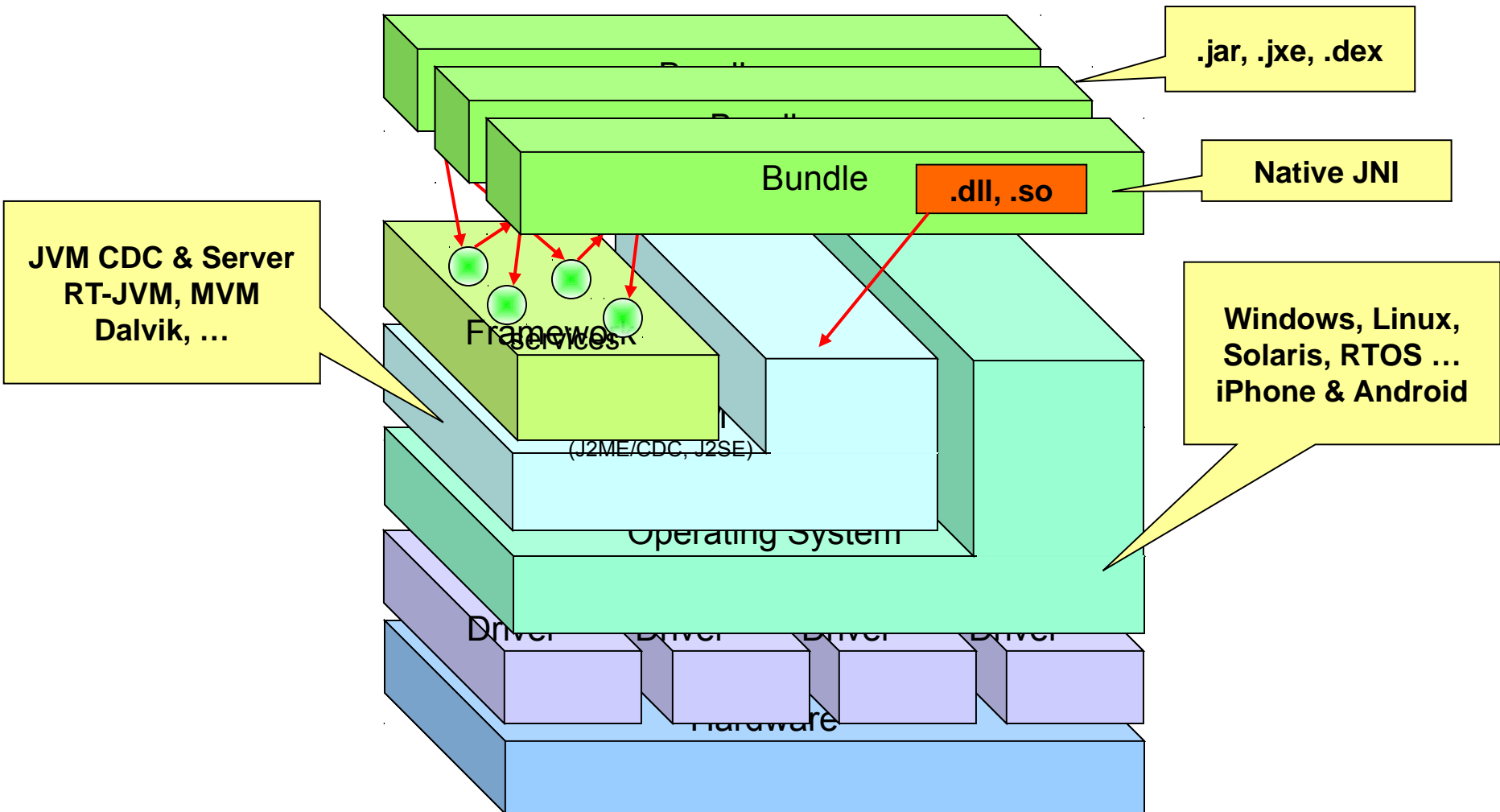
- ⑦ Based on Jar file and Manifest entries
- ⑦ Explicit Package dependencies and Versioning (range)

⑦ Ready for probably next generation standard

- ⑦ JSR 277, Java Module System
- ⑦ Overtake JNLP(JSR-56), J2EE EAR, OSGi R3 bundle
- ⑦ Java Platform 7.0 (2010), Jigsaw

The OSGi « stack »

avec l'aimable autorisation de Peter Kriens





OSGi R4			OSGi ME		NanOSGi
JavaEE	JavaSE	CDC		CLDC	
		J2ME			
RTSJ JVM	HotSpot JVM	CVM		KVM	Bare Metal_ VM
RTOS	OS		RTOS		
μ P 32 64 bits UltraSPARC Itanium i7 i5 i3		μ P 32bits Via Atom ARM11 ARM9		μ P 16 32 bits AVR32 ARM7	
				μ C 8 16 bits AVR8 MSP430	
RAM Usage	100MB	10MB	512KB	32KB	1KB
MW	Energy Consumption				μ W

OSGi

Acteurs, concurrences,
tendances et perspectives

L' OSGi™ Alliance

- ⑦ actuellement 44+ membres
- ⑦ de plusieurs domaines industriels



- ⑦ sur les 4 segments



AUTO



OFFICE



HOME



MOBILE

L'OSGi™ Alliance



Alpine Electronics Europe GmbH , Aplix Corporation , Belgacom , BMW Group , Cablevision Systems , Computer Associates , Deutsche Telekom AG , Echelon Corporation , Electricité de France (EDF) , Ericsson Mobile Platforms AB , Esmertec , Espial Group, Inc. , ETRI Electronics and Telecommunications Research Institute , *France Telecom* , Gatespace Telematics AB , Gemplus , Harman/Becker Automotive Systems GmbH , IBM Corporation , Industrial Technology Research Institute , Insignia Solutions , Intel Corporation , KDDI R&D Laboratories, Inc. , KT Corporation , Mitsubishi Electric Corporation , Motorola, Inc. , NEC Corporation , Nokia Corporation , NTT , Oracle Corporation , Panasonic Technologies, Inc. , Philips Consumer Electronics , ProSyst Software GmbH , Robert Bosch GmbH , Samsung Electronics Co., Ltd. , SavaJe Technologies, Inc. , Sharp Corporation , Siemens AG , Sun Microsystems, Inc. , Telcordia Technologies, Inc. , Telefonica I+D , TeliaSonera , Toshiba Corporation , Vodafone Group Services Limited

Les acteurs incontournables

⑦ IBM

- ⑦ OSGi est au cœur de la stratégie d'IBM
- ⑦ placé sur la partie « edge » du système IT
 - ⑦ poste de travail (RCP)
 - ⑦ serveur enfoui
- ⑦ Remarque:
 - ⑦ Eclipse 3.0 (donc WebSphere Studio) est désormais développé au dessus d'OSGi (Equinox)

⑦ Nokia

- ⑦ Pousse pour
 - « Java (MIDLet) dans toutes les poches » (2002)
 - « Java Server dans toutes les poches » (2005)

Produits

- ⑦ ~~SUN Java Embedded Server (JES)~~ *l'étincelle*
- ⑦ *Echelon LonWorks Bundle Deployment Kit*
- ⑦ *Ericsson - Residential e-services*
- ⑦ *MakeWave (ex Gatespace AB)*
- ⑦ *IBM SMF*
- ⑦ *Insignia*
- ⑦ *Nano Computer System*
- ⑦ *ProSyst Software mBedded Server*
- ⑦ *Wind River*
- ⑦ *Siemens VDO TLA (R3)*
- ⑦ ...

La communauté open-source

⑦ Plusieurs implémentations et communautés

- ⑦ ObjectWeb Oscar
- ⑦ Knopf erf sh
- ⑦ Eclipse Equinox (donation IBM SMF)
- ⑦ Apache Felix (suite d' ObjectWeb Oscar)
- ⑦ JBoss microContainer (<http://labs.jboss.com/jbossmc>)

r3

r4

⑦ Abaissement des barrières de l'OSGi pour le développement open-source.

- ⑦ Dépôts de bundles (org.osgi.service.obr)

Oscar/Felix



⑦ Console texte sur Nokia 770 (JamVM)

▼ X Terminal

```
obr help
packages [<id> ...]
ps [-l | -u]
refresh
services [-u] [-a] [<id> ...]
shutdown
start <id> [<id> <URL> ...]
startlevel [<level>]
stop <id> [<id> ...]
uninstall <id> [<id> ...]
update <id> [<URL>]
version
-> █
```

- Oscar bundle repository.
- list exported packages.
- list installed bundles.
- refresh packages.
- list registered or used services.
- shutdown Oscar.
- start bundle(s).
- get or set framework start level.
- stop bundle(s).
- uninstall bundle(s).
- update bundle.
- display version of Oscar.

ABC ↑ ↵

Merci à Corentin Baron

Apache Felix on Google Android



⑦ Google Android / Dalvik VM

⑦ JVM and JRE for the Google phone

- ⑦ Support Java 5 features (annotations, ...)
- ⑦ Uses a proprietary class format (.dex instead of .class/.jar)

⑦ Felix running on Android

⑦ See

- ⑦ <http://felix.apache.org/site/apache-felix-and-google-android.html>

Knopfer sh OSGi

⑦ La console GUI

The screenshot displays the Knopferfish OSGi desktop GUI. The main window is titled "Knopferfish OSGi desktop (knopferfish)" and features a menu bar (File, Edit, Bundles, View, Help) and a toolbar. The "Start level" is set to "7 HTTP-root-IMPL".

The left pane shows a grid of bundles, including System Bundle, LogService, cm, Console, Declarative-S, Event-Admin, prefs, util-LIB, Crimson-XML, JSDK-API, bundlerepository, Device-Manager, UserAdmin, HTTP-Server, FW-Commanc, LogCommands..., CM-Command..., TTY-Console-I..., Telnet-Consol..., and RemoteFW-A. The "Desktop" bundle is selected.

The right pane displays the details for the selected "#20 Desktop" bundle:

- Location: file:jars/desktop/desktop_all-2.3.7.jar
- State: active
- Symbolic name: org.knopferfish.bundle.desktop
- Last modified: 05/08/09 16:59
- Start level: 6
- Ant-Version: Apache Ant 1.7.1
- Application-Icon: app.png
- Build-Date: Thu July 9 2009, 21:18:27
- Build-From: /home/ekolin/work/kf/svn/knopferfish.org-2.3.2/
- Bundle-APIVendor: Knopferfish
- Bundle-Activator: org.knopferfish.bundle.desktop.swing.Activator
- Bundle-Category: console
- Bundle-Classpath: .
- Bundle-ContactAddress: <http://www.knopferfish.org>
- Bundle-Description: Swing framework desktop
- Bundle-DocURL: <http://www.knopferfish.org/desktop.html>
- Bundle-ManifestVersion: 2
- Bundle-Name: Desktop
- Bundle-SubversionURL: <https://www.knopferfish.org/svn/knopferfish.org/>
- Bundle-SymbolicName: org.knopferfish.bundle.desktop
- Bundle-UUID: org.knopferfish:desktop:2.3.7

The bottom pane shows a console window with the following text:

```
Type 'help' for help or 'alias' for a list of common commands
> [stderr] Error: java.net.ConnectException: Connection timed out: connect
>
```

Knopf erf sh OSGi

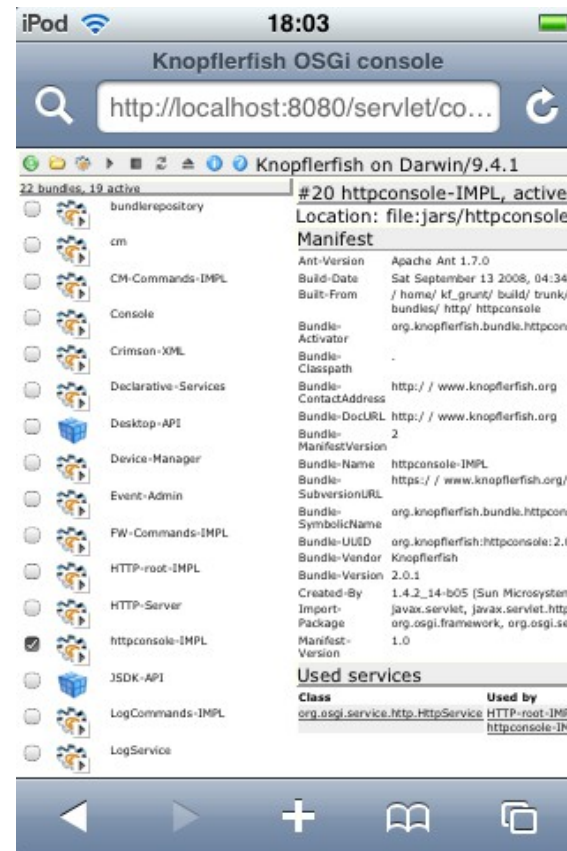
⑦ La console GUI sur le iPod Touch/iPhone 2.1 (JamVM)



iPod 18:01

```
7 2/resolved util-LIB
8 2/active Crimson-XML
9 2/resolved JSDK-API
10 2/active bundlerepository 1
11 3/active Device-Manager
12 3/active UserAdmin 1
13 4/active HTTP-Server
14 5/active FW-Commands-IMPL 1
15 5/active LogCommands-IMPL
16 5/active CM-Commands-IMPL 1
17 5/active TTY-Console-IMPL
18 5/active Telnet-Console-IMPL 1
19 6/resolved Desktop-API
20 6/active httpconsole-IMPL 2
21 7/active HTTP-root-IMPL
>
```

Q W E R T Y U I O P
A S D F G H J K L
↑ Z X C V B N M ↵
.?123 space return



iPod 18:03

Knopflerfish OSGi console

http://localhost:8080/servlet/co...

Knopflerfish on Darwin/9.4.1

22 bundles, 19 active

- bundlerepository
- cm
- CM-Commands-IMPL
- Console
- Crimson-XML
- Declarative-Services
- Desktop-API
- Device-Manager
- Event-Admin
- FW-Commands-IMPL
- HTTP-root-IMPL
- httpconsole-IMPL
- JSDK-API
- LogCommands-IMPL
- LogService

#20 httpconsole-IMPL, active

Location: file:jars/httpconsole/

Manifest

Ant-Version	Apache Ant 1.7.0
Build-Date	Sat September 13 2008, 04:34:1
Build-From	/home/kl_grunt/build/trunk/o
Bundle-Activator	org.knopflerfish.bundle.httpconso
Bundle-Classpath	-
Bundle-ContactAddress	http://www.knopflerfish.org
Bundle-DocURL	http://www.knopflerfish.org
Bundle-ManifestVersion	2
Bundle-Name	httpconsole-IMPL
Bundle-SubversionURL	https://www.knopflerfish.org/s
Bundle-SymbolicName	org.knopflerfish.bundle.httpconso
Bundle-UUID	org.knopflerfish:httpconsole:2.0.1
Bundle-Vendor	Knopflerfish
Bundle-Version	2.0.1
Created-By	1.4.2_14-b05 [Sun Microsystems
Import-Package	javax.servlet, javax.servlet.http,
Manifest-Version	org.osgi.framework, org.osgi.serv

Used services

Class	Used by
org.osgi.service.http.HttpService	HTTP-root-IMPL
	httpconsole-IMPL

Navigation: back, forward, +, book, copy

D'après <http://knopf erf sh.blogspot.com/2008/09/knopf erf sh-osgi-on-ipod-touchiphone.html>

Eclipse/OSGi

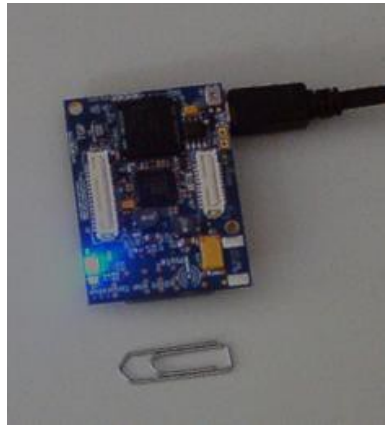
- ⑦ **embarqué dans Eclipse/IDE et Eclipse/RCP**
 - ⑦ Pour le conditionnement et le déploiement des Plugins
 - ⑦ Reconditionné par Prosyst
- ⑦ **La console texte**
 - ⑦ `java -jar %ECLIPSE_HOME%\plugins\org.eclipse.osgi_3.1.0.jar -console`

```
OS/4 Selectionner Invite de commandes - java -jar F:\C\devtools\eclipse\plugins\org.eclip.  
F:\>java -jar F:\C\devtools\eclipse\plugins\org.eclipse.osgi_3.1.0.jar -console  
osgi> help  
---Eclipse Runtime commands.---  
  diag - Displays unsatisfied constraints for the specified bundle(s).  
  active - Displays a list of all bundles currently in the ACTIVE state.  
  getprop { name } - Displays the system properties with the given name, or all of them.  
Valid commands:  
---Controlling the OSGi framework---  
  launch - start the OSGi Framework  
  shutdown - shutdown the OSGi Framework  
  close - shutdown and exit  
  exit - exit immediately (System.exit)  
  gc - perform a garbage collection  
  init - uninstall all bundles
```

Concierge

<http://concierge.sourceforge.net/>

- ⑦ **OSGi R3 implementation optimized for constrained mobile devices (iMote2, BUG (<http://buglabs.net/products>), NSLU, WRT54, ...)**



<http://www.spectrum.ieee.org/video?id=223>

- ⑦ ***Jan S. Rellermeyer, Gustavo Alonso: Concierge: A Service Platform for Resource-Constrained Devices. In: Proceedings of the 2007 ACM EuroSys Conference, Lisbon, Portugal, 2007.***

⑦ Suite de tests pour vérifier la compatibilité

- ⑦ Du framework (Core)
- ⑦ Des services standards (Compendium)

- ⑦ Les tests concernent les fonctionnalités obligatoires et optionnelles (fragments, ...)

⑦ Remarque

- ⑦ Seulement accessible aux membres de l'OSGi Alliance
- ⑦ En principe accessible à quelques plateformes open sources (Felix, Equinox, ...)

JSR 277 : Java™ Module System

- ⑦ *def nes a distribution format and a repository for collections of Java code and related resources.*
- ⑦ *also def nes the discovery, loading, and integrity mechanisms at runtime.*

JSR 294 : Improved Modularity Support in the Java™ Programming Language

⑦ Notion of super-package

⑦ introduce in Java

⑦ in order to define class visibility outside the deployment unit (JSR277)

⑦ Example

```
superpackage A {  
  member package P;  
  member superpackage B;  
}  
superpackage B member A {  
  member superpackage C;  
  export superpackage C;  
}  
superpackage C member B {  
  member package Q;  
  export Q.X;  
}
```

A type P.Y can access type Q.X because Q.X is exported from superpackage C, and C is exported from B, so Q.X is available in A

Jigsaw

⑦ JDK7 is big

⑦ <http://blogs.sun.com/mr/entry/jigsaw>

HK2 « Hundred Kilobytes Kernel »

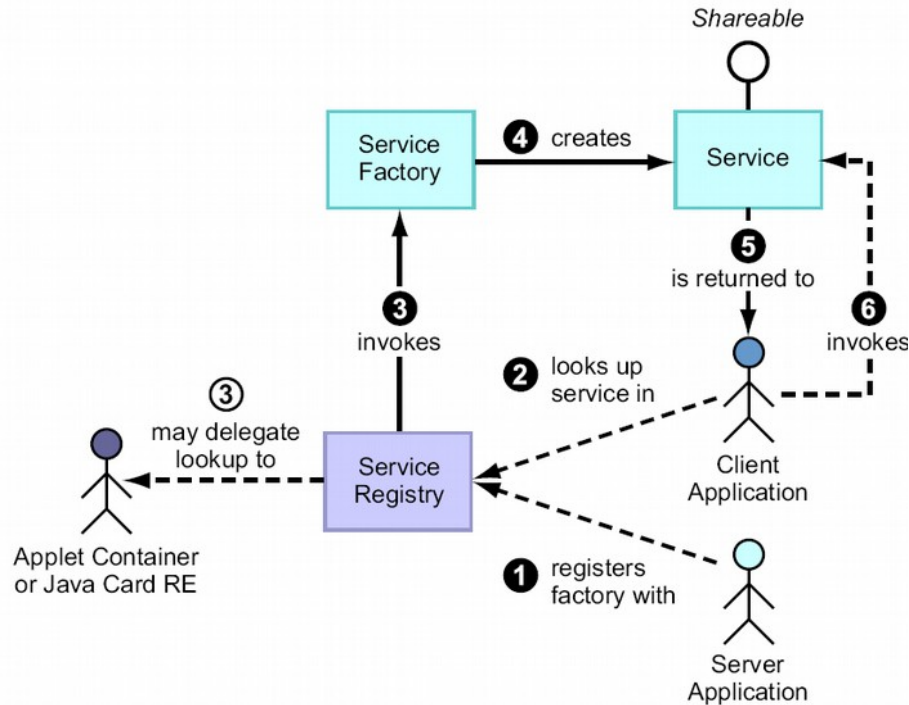
<https://hk2.dev.java.net>

- ⑦ **Small kernel (than OSGi™) to build modular softwares**
 - ⑦ consist of two technologies :
- ⑦ **Modules subsystem**
 - ⑦ offer a better level of isolation between parts of the application.
 - ⑦ path to the implementation of modules (JSR 277 & 294) in Java SE 7.
- ⑦ **Component Model**
 - ⑦ define components which can also be seen as Services.
 - ⑦ These components can be automatically and dynamically discovered by the runtime and can use innovative technologies such as Inversion of Control or injection of dependencies as well as automatic resolution of dependencies.
 - ⑦ Annotations (org.jvnet.hk2.annotations)
 - ⑦ @Contract, @Service, @Inject, @Extract, @Factory, @Scoped, @ContractProvided, ...
- ⑦ **Remark: foundation for the GlassFish V3 application server**
 - ⑦ April 2008 : GlassFish v3 on Apache Felix !!!

JavaCard 3.0 Connected Edition

Shareable Interface Object-based Services (SIO)

⑦ Service-oriented communications between on-card applications



⑦ Limitations

- ⑦ Service classes must inherit of the interface `javacard.framework.Shareable`
- ⑦ Service lookup is only based on service URI (e.g. `sio:///transit/pos/ticketbook`)
- ⑦ `javacardx.facilities.ServiceRegistry` lookup methods return `Shareable` objects
- ⑦ No event (`javacardx.facilities.StandardEvent`) for SIO registration/unregistration

La concurrence : *embedded-Linux*

⑦ Plate-forme

- ⑦ Multi-application
- ⑦ Multi-fournisseurs (users)
- ⑦ Éventuellement Temps Réel



⑦ Solutions

- ⑦ Chargement / Déchargement dynamique de .so
- ⑦ Processus (comme sandbox)

⑦ Avantages

- ⑦ Très très répandu ...

⑦ Inconvénient

- ⑦ Coûts des échanges (IPC,socket) entre les « services »

La concurrence : *JUNGO OpenRG*

<http://www.jungo.com/openrg/index.html>

⑦ **TODO**

- ⑦ ***“OpenRG™ is a complete and integrated gateway software platform for developing network devices in the digital home and small office including triple play residential gateways, home/SOHO routers, home gateways, wireless access points, cable/DSL routers and voice gateways.”***

La concurrence : MicroSoft .NET

⑦ Passerelles industrielles (SCADA) OPC

http://en.wikipedia.org/wiki/OPC_Foundation

- ⑦ COM/DCOM : drivers field bus, domaine SCADA

⑦ .NET

- ⑦ Alternative à Java (et à machine virtuelle)
- ⑦ Mais des approches similaires
 - ⑦ bytecode, JIT, chargeur de classes, ...
- ⑦ et différentes
 - ⑦ Multi-langage, cache de compilation, domaine d'application, ...
- ⑦ Compact.NET alternative à J2ME/CDC
- ⑦ Très récente annonce de .NET Micro Framework (CLR sur *bare metal*)

⑦ Le problème de .NET (1 et 2)

- ⑦ Le déchargement d'une classe requière l'arrêt du domaine d'application (AppDomain).
- ⑦ Donc pas de mise à jour partielle d'une application

⑦ Des compromis (non gratuits) restent possibles

- ⑦ [Escoffier06]
- ⑦ SOF - An OSGI-like modularization framework for C++
 - ⑦ <http://sof.tiddlyspot.com/> http://www.codeproject.com/KB/library/SOF_.aspx



Module and SO C/C++

⑦ Apply OSGi principles to C/C++ development

- ⑦ Modularity and Dynamic Update

⑦ SOF - An OSGI-like modularization framework for C++

- ⑦ <http://sof.tiddlyspot.com/>
http://www.codeproject.com/KB/library/SOF_.aspx

⑦ Celix

- ⑦ a OSGi like implementation in C with a distinct focus on interoperability between Java-OSGi and Celix.
- ⑦ <http://opensource.luminis.net/wiki/display/SITE/Celix>
- ⑦ <http://wiki.apache.org/incubator/CelixProposal>

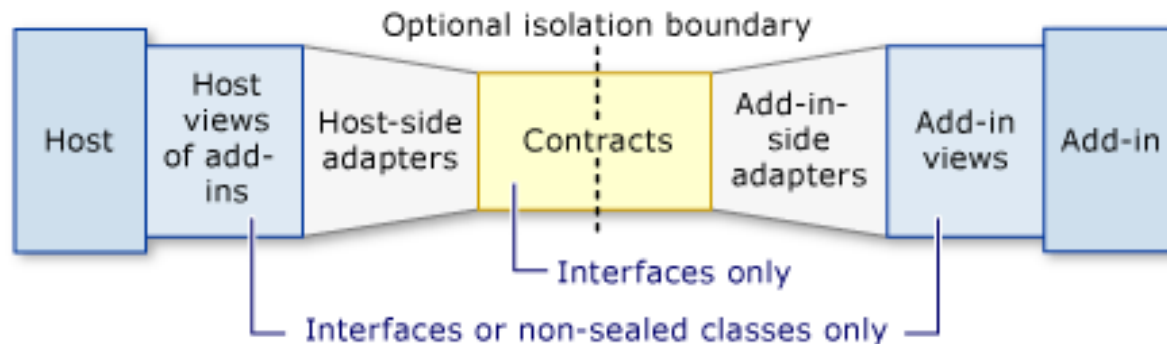
⑦ nOSGi : a posix-compliant native OSGi framework

- ⑦ <http://dl.acm.org/citation.cfm?id=2016555>

La concurrence : MicroSoft .NET

⑦ Managed AddIn Framework (MAF, aka System.AddIn)

⑦ Based on Application Domains



- ⑦ Discovery – **Find addins at runtime**
- ⑦ Activation – **Load an addin at runtime**
- ⑦ Versioning - **Backwards/forward compatibility**
- ⑦ Isolation – **Load addins into separate AppDomains/Processes so they cannot crash they whole app, among other reasons**
- ⑦ Lifetime Management – **MAF will handle addin lifetimes and memory/AppDomain management**
- ⑦ Sandboxing – **Load addins with specific permission sets like “Internet”**
- ⑦ Unloading – **Unload an addin without worrying about tedious AppDomain management**

⑦ ~~Other: Managed Extensibility Framework (MEF) in .NET 4~~

La tendance

- ⑦ **OSGi couvre désormais un spectre étendu de domaine**
 - ⑦ Passerelle résidentiel
 - ⑦ Passerelle véhiculaire
 - ⑦ Passerelle industrielle

 - ⑦ Téléphonie mobile
 - ⑦ Application sur poste de travail (Eclipse RCP)

 - ⑦ Enterprise Side OSGi
 - ⑦ Serveur IT (J2EE, ...), Web Framework, ...
 - ⑦ JOnAS 5, Geronimo, ApacheDS, JAMES, WebSphere, JBoss 5 ...
 - ⑦ ECP

 - ⑦ Enterprise Expert Group à l'OSGi Alliance
 - ⑦ Workshop sur OSGi + J2EE (11/09/2006, San José)
 - ⑦ End-to-End Remote Management Expert Group à l'OSGi Alliance

Java is the leading mobile Application Development Environment

Global installed base millions

1600

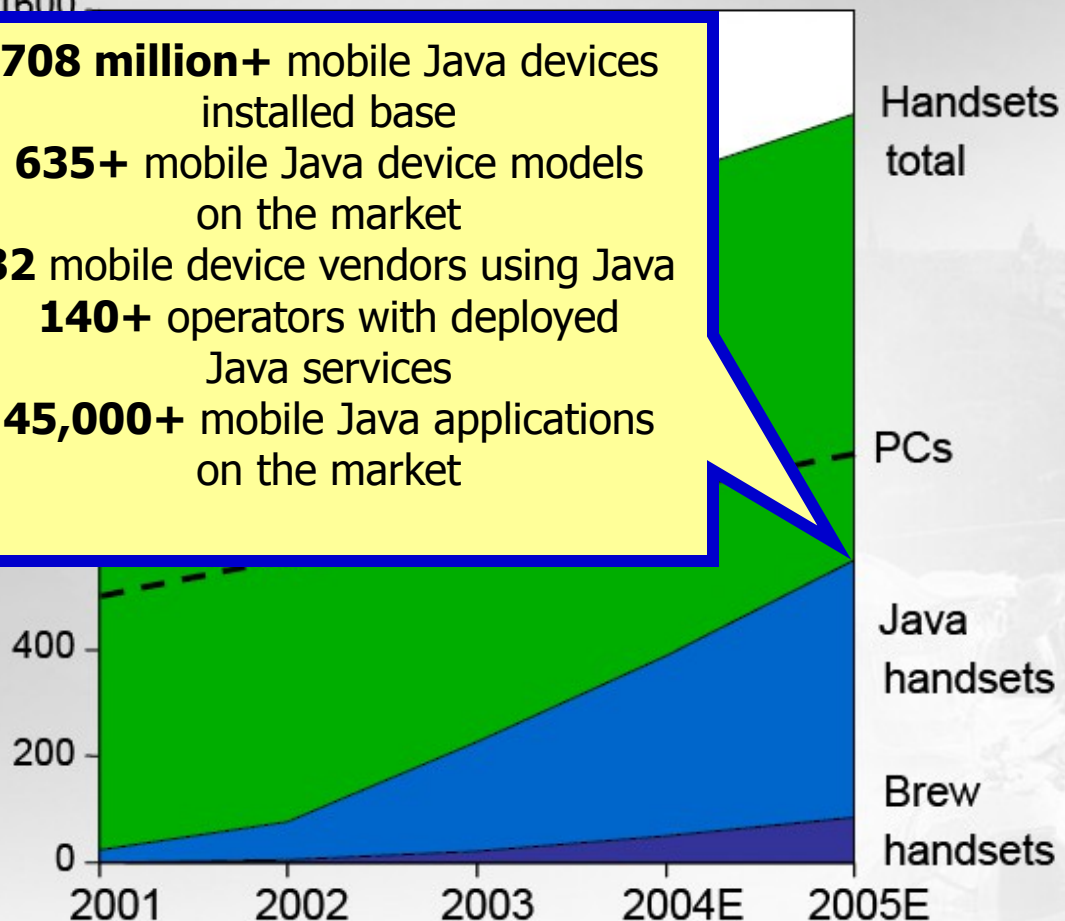
708 million+ mobile Java devices installed base

635+ mobile Java device models on the market

32 mobile device vendors using Java

140+ operators with deployed Java services

45,000+ mobile Java applications on the market



Key message in 2002
JavaOne:

“We will put Java in every pocket”
...done.

Key message in 2005
JavaOne:

“We will put Java server in every pocket”
...working on it...



Les perspectives

⑦ Open-source

- ⑦ Match entre Eclipse et Apache ?

⑦ JSR 277 Java™ Module System

- ⑦ ~ couvert par OSGi R4 Module Layer
- ⑦ Prévu pour Java Platform 7.0

⑦ JSR 291 Dynamic Component Support for Java™ SE

- ⑦ ~ couvert par OSGi R4 SCR
- ⑦ Prévu pour Java Platform 7.0

⑦ JSR 313 Java Platform, Enterprise Edition 6

- ⑦ Extensibilité du serveur JEE

⑦ Quel nouveau rôle pour OSGi™ ?

- ⑦ Définition de Services Standards

⑦ Quand même

« une crainte pour les « penseurs » de l'Alliance »

- ⑦ Faire venir les développeurs de logiciels embarqués à J2ME et à OSGi ...

Pour terminer

Un peu de publicité

⑦ OSGi™ Users' Group France

⑦ Association d'utilisateurs de la technologie OSGi™

⑦ Développeurs, consultants, enseignants, chercheurs, ...

⑦ de FT R&D, EDF R&D, Schneider Electric, Trialog, Siemens VDO, Gemplus, Alcatel, Bull, Thomson, Scalagent ...

⑦ et de l'INRIA, CNRS, ...

⑦ 7 réunions depuis Décembre 2004

⑦ Ecole d'été : 6-7 Septembre 2007
Ile de Bender (près de Vannes)

⑦ Prochaine réunion le 29/1/2008 à Paris

⑦ En savoir plus

⑦ <http://france.osgiusers.org>



Conclusion finale

+ Très fort potentiel

Ne passez pas à côté

Q & R

Bibliographie & Webographie

⑦ Spécification

- ⑦ OSGi Alliance, « OSGi service gateway specification », <http://www.osgi.org>

⑦ Ouvrage

- ⑦ Kirk Chen, Li Gong, « Programming Open Service Gateways with Java Embedded Server Technology », Pub. Addison Wesley, August 2001 ISBN#: 0201711028. 480 pages (un peu ancien)
- ⑦ Niel Barlett, « OSGi in Practice », <http://neilbartlett.name/blog/osgibook/>
- ⑦ Richard S. Hall, Karl Pauls, and Stuart McCulloch, « OSGi in Action », May 2009, 375 pages, ISBN: 1933988916, <http://code.google.com/p/osgi-in-action> & <http://manning.com/hall>
- ⑦ Holly Cummins and Timothy Ward, Enterprise OSGi in Action, March, 2013, 400 pages, ISBN: 9781617290138, <http://www.manning.com/cummins/>

⑦ Blogs

- ⑦ Peter Kriens, *l'évangéliste OSGi*, <http://www.osgi.org/blog/>
- ⑦ Java modularity, JSR 277, JSR 291, JSR 294, OSGi, open source, and software design, <http://underlap.blogspot.com/>

Bibliographie & Webographie

⑦ Framework open source

- ⑦ Oscar, Felix, Equinox, Knopper sh

⑦ Index de bundles

- ⑦ <http://bundles.osgi.org/browse.php>

⑦ Exhibitions

- ⑦ <http://www.osgiworldcongress.com/>

⑦ Complément de cours

- ⑦ Donsez, Hall, Cervantes <http://www-adele.imag.fr/users/Didier.Donsez/cours/osgi.pdf>
- ⑦ Frénot <http://citi.insa-lyon.fr/~sfrenot/cours/OSGi/>
- ⑦ INTech <http://rev.inrialpes.fr/intech/Registration?op=511&meeting=27>

⑦ Plus

- ⑦ <http://france.osgiusers.org/Main/Documentation>

Travaux pratiques

⑦ **Sujet**

- ⑦ <http://www-adele.imag.fr/users/Didier.Donsez/cours/exemplesosgi/tutorialosgi.htm>

⑦ **Programme**

- ⑦ **Installation d'Apache Felix**
- ⑦ **Premières commandes via différentes consoles**
- ⑦ **Déploiement de bundles**
- ⑦ **Développement d'un servent**
- ⑦ **Développement d'un composant SCR**
 - ⑦ **Utilisant Event Admin Service**
 - ⑦ **Utilisant Wire Admin Service**
 - ⑦ **Utilisant Http Service Service**
- ⑦ **Démonstration de l'UPnP Base Driver**
- ⑦ **Démonstration d'administration de passerelle sur NSLU2**
- ⑦ **Démonstration d'administration de passerelle avec JMX**
- ⑦ **Mini-projet (1 bundle utilisant d'autres)**



<http://www.nslu2-linux.org>

Mini-Projet : http.script

- ⑦ **Servlet exécutant une séquence de commandes auprès du service `org.apache.felix.shell.ShellService`**

- ⑦ **Réutiliser le code de**
 - ⑦ `http.webadmin`
 - ⑦ `shell.scriptcmd`

- ⑦ **Utiliser le SCR pour la gestion des liaisons et du cycle de vie**

OSGi



Les Outils

Les outils

- ⑦ **Eclipse PDE**
- ⑦ **Maven Bundle Plugin**
- ⑦ **Bindex**
- ⑦ **BND**
- ⑦ **Console OSGi**
- ⑦ **Unit testing**

Eclipse PDE

- ⑦ **Since Eclipse 3, Plugins are OSGi bundles**
- ⑦ **TODO**

BND (<http://www.aqute.biz/Code/Bnd>)

- ⑦ Tool to create and diagnose OSGi R4 bundles
 - ⑦ Available as Command line tool, Eclipse plugin, Ant Task, Maven Plugin
- ⑦ Key functions
 - ⑦ Compute the classpath, import, export of a bundle
 - ⑦ Show the manifest and JAR contents of a bundle
 - ⑦ Wrap a JAR so that it becomes a bundle
 - ⑦ Create a Bundle from a specification and a class path
 - ⑦ Verify the validity of the manifest entries
- ⑦ Examples
 - ⑦ wrap an existing jar
 - ⑦ `java -jar bnd.jar wrap -output myOSGi.jar myPlain.jar`
 - ⑦ More TODO

BND – Directives

- ⑦ **Export-Package LIST of PATTERN**
 - ⑦ lists the packages that the bundle should export, and thus contain.
- ⑦ **Include-Resource LIST of iclause**
 - ⑦ makes it possible to include arbitrary resources; it contains a list of resource paths. See Include Resource.
- ⑦ **Private-Package LIST of PATTERN**
 - ⑦ lists the packages that the bundle should contain but not export. See Private Package.
- ⑦ **Import-Package LIST of PATTERN**
 - ⑦ lists the packages that are required by the contained packages. See Import Package.
- ⑦ **Conditional-Package LIST of PATTERN experimental**
 - ⑦ Works as private package but will only include the packages when they are imported. When this header is used, bnd will recursively add packages that match the patterns until there are no more additions.
- ⑦ **Bundle-SymbolicName**
 - ⑦ The default is the name of the main bnd file, or if the main bnd file is called bnd.bnd, it will be the name of the directory of the bnd file. An interesting variable is `${project}` that will be set to this default name.
- ⑦ **Bundle-Name**
 - ⑦ If the Bundle-Name is not set, it will default to the Bundle-SymbolicName.
- ⑦ **Bundle-ManifestVersion 2**
 - ⑦ The Bundle-ManifestVersion is always set to 2, there is no way to override this.
- ⑦ **Bundle-Version VERSION**
 - ⑦ The version of the bundle. If no such header is provided, a version of 0 will be set.
- ⑦ **Service-Component LIST of component**

BND - Directives

⑦ Operator @ d'expansion de jarf les

```
<Include-Resource>
```

```
  @hexdump.jar,
```

```
  @libdbus-java.jar,
```

```
  @unix-*.jar,
```

```
  lib/libunix-java.so
```

```
</Include-Resource>
```

```
<Bundle-NativeCode>
```

```
  libunix-java.so;osname=Linux;processor=x86
```

```
</Bundle-NativeCode>
```

BND - Directives

⑦ Macro `f ndpath`

⑦ `${f ndpath;regexp[;replacement]}`

⑦ traverse the files in the built jar file and check each path with the regular expression. If the regexp matches it puts it on a comma separated list. This allows us to construct the **Bundle-Classpath** from the lib directory in the built jar.

```
<Include-Resource>lib=lib</Include-Resource>
```

```
<Bundle-Classpath>${f ndpath;lib/*.jar}</Bundle-Classpath>
```

```
<Bundle-NativeCode>
```

```
libunix-java.so;osname=Linux;processor=x86
```

```
</Bundle-NativeCode>
```


Maven

⑦ Tool for building and managing Java projects

- ⑦ Standard organization of the project directory
- ⑦ Repositories (local and remote, public and private) of artifacts
- ⑦ Project description in pom.xml (Project Object Model)
- ⑦ Set of predefined commands
 - ⑦ mvn clean
 - ⑦ mvn install
 - ⑦ mvn eclipse:eclipse

⑦ Extensibles by plugins

- ⑦ MOJO (in Java, Groovy, ...)
- ⑦ For OSGi, Maven Bundle Plugin (Apache Felix)

Maven Bundle Plugin (Apache Felix)

<http://felix.apache.org/site/maven-bundle-plugin-bnd.html>

- ⑦ **Maven plugin to package a Java (Maven) project as a OSGi R4 bundle**
 - ⑦ Use BND tools (and directives) to calculate import/export packages
- ⑦ **Example of pom.xml**

```
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.felix</groupId>
      <artifactId>maven-bundle-plugin</artifactId>
      <extensions>>true</extensions>
      <configuration>
        <instructions>
          <Import-Package>*</Import-Package>
          <Private-Package>${pom.artifactId}.*</Private-Package>
          <Export-Package>${pom.artifactId};version=1.0.0</Export-Package>
          <Bundle-Activator>${pom.artifactId}.impl.Activator</Bundle-Activator>
          <Export-Service>org.osgi.service.event.EventHandler</Export-Service>
          <Import-Service>org.osgi.service.log.LogService</Import-Service>
        </instructions>
      </configuration>
    </plugin>
  </plugins>
</build>
...
```

Maven Archetypes for OSGi projects

⑦ Maven archetype

- ⑦ Parameterized template to start quickly a Maven-enabled project

⑦ Archetypes for OSGi bundle projects

- ⑦ General purpose

- ⑦ http://gforge.inria.fr/scm/?group_id=526

- ⑦ SpringOSGi

- ⑦ `spring-osgi-bundle-archetype`

- ⑦ ...

Pax Construct

<http://wiki.ops4j.org/conference/display/ops4j/Pax+Construct>

⑦ Templates for kick-start bundle development

⑦ Project / bundles / common techniques

⑦ Maven pax plugin

⑦ Manages project builds file

⑦ **mvn clean install pax:provision**

⑦ + interactive scripts (Windows only)

⑦ Example

pax-create-project

-p com.example.client

-n simple-test-client

-a kxml2

-v 0.1.0-SNAPSHOT

-Dunpack=true

Simple Pax use case

a web based OSGi application

- ⑦ `pax-create-project -g simple.project -a osgi-web-app`
- ⑦ `cd osgi-web-app`
- ⑦ `pax-wrap-jar -g javax.servlet -a servlet-api -v 2.5`
- ⑦ `pax-import-bundle -g org.ops4j.pax.logging -a api -v 0.9.4`
- ⑦ `pax-import-bundle -g org.ops4j.pax.logging -a jcl -v 0.9.4`
- ⑦ `pax-import-bundle -g org.ops4j.pax.logging -a slf4j -v 0.9.4`
- ⑦ `pax-import-bundle -g org.ungoverned.osgi.bundle -a http -v 1.1.2`
- ⑦ `pax-create-bundle -p my.osgi.code.myBundle -n myBundle`
- ⑦ `mvn install pax:provision`

Pax Runner

<http://wiki.ops4j.org/confluence/display/ops4j/Pax+Runner>

- ⑦ Provides URL handlers (mvn,wrap,classpath) to deploy bundles or simple jar files from a Maven 2 repository
- ⑦ Provides scripts to setup initial provisioning configuration files for Felix, Equinox and KF

BIndex (<http://www2.osgi.org/Repository/BIndex>)

⑦ Tool to create RFC-0112 Bundle Repository indexes from bundles manifests.

⑦ Available as Command line tool and Ant Task

⑦ Maven plugin : `org.apache.felix:maven-obr-plugin`

⑦ Incremental index update

⑦ Options

`java -jar bindex.jar`

`[-help]`

`[-r repository.xml (.zip ext is ok)]`

`[-l file:license.html]`

`[-t http://w.com/servlet?s=%s&v=%v %s=symbolic-name %v=version %p=relative path %f=name]`

`[-q (quiet)]`

`<jar file>*`



⑦ the RFC 112

⑦ http://www2.osgi.org/div/rfc-0112_BundleRepository.pdf

⑦ **bushel** <http://code.google.com/p/bushel/>

⑦ *Bushel is a simple utility to convert ad-hoc OSGi bundles into a local Ivy repository.*

Unit testing

⑦ Motivation

- ⑦ Unit testing of individual bundles (w w/o services)
- ⑦ Before packaging and deployment
- ⑦ Running of a OSGi platform (without rebooting)
- ⑦ Integration to builders (Maven Surefire) and continuous integration

⑦ Architectures

- ⑦ Whiteboard pattern
- ⑦ Extended pattern

⑦ Example

- ⑦ JUnit4OSGi
 - ⑦ <http://felix.apache.org/site/apache-felix-ipojo-junit4osgi.html>
- ⑦ Pax Exam
 - ⑦ <http://wiki.ops4j.org/display/paxexam/Pax+Exam>

OSGi and AOT compilation

⑦ Motivations

- ⑦ “Accelerate” and protect OSGi applications
 - ⑦ On-the-fly code injection is harder
 - ⑦ Retrocompilation is harder
- ⑦ Reduce the Runtime size

⑦ Tools

- ⑦ Excelsior JET
 - ⑦ Testing with RCP bundles

OSGi graphical consoles

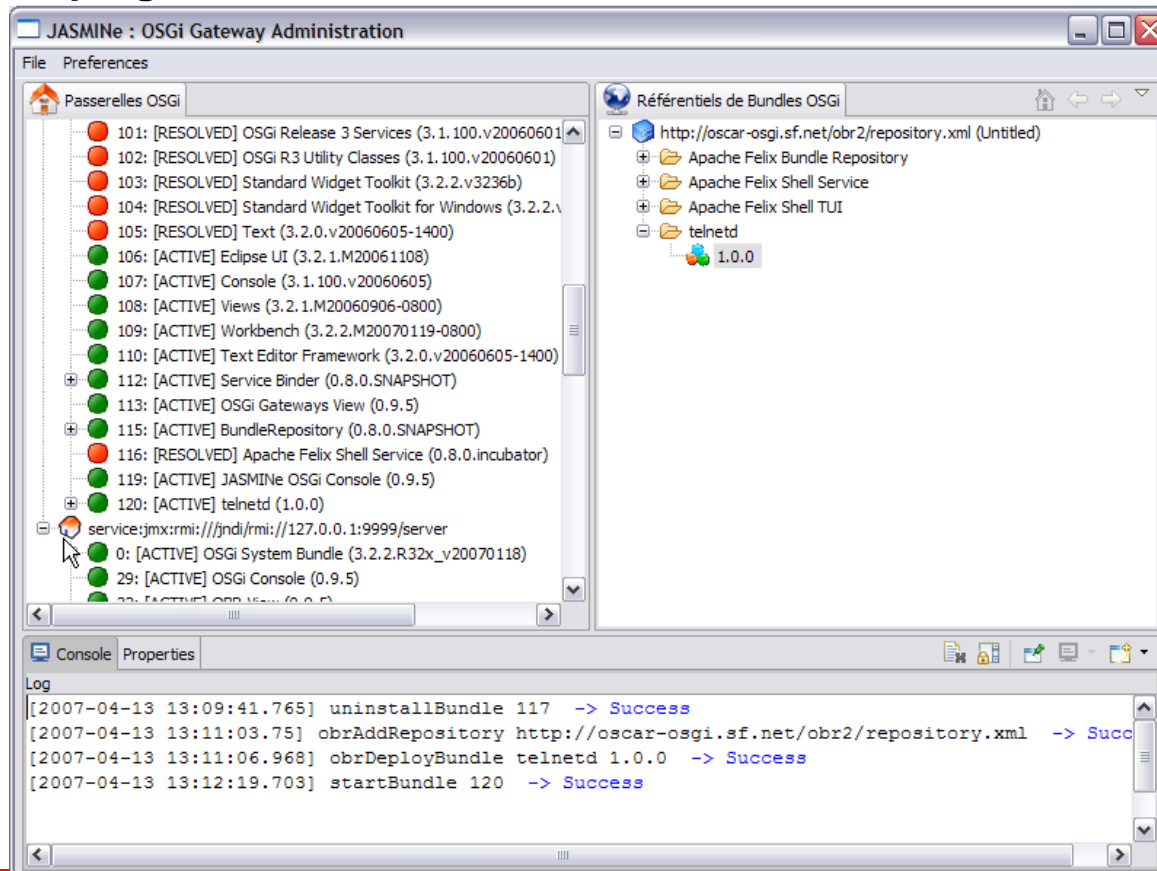
- ⑦ **Felix mOSGi console**
- ⑦ **KF**
- ⑦ **Jasmine OSGi console**
- ⑦ **Prosyst mConsole**
- ⑦ **...**

Jasmine OSGi Console

(<http://jasmine.objectweb.org/doc/console/jasmine.htm>)

⑦ Eclipse RCP & Plugin to manage OSGi platforms

- ⑦ connection to the platform with JMX, WS, UPnP, ...
- ⑦ OBR plugin



Apache Felix mOSGi console

⑦ **TODO**

Prosynt mConsole

The screenshot displays the Prosynt mConsole 1.1 interface. The window title is "mConsole 1.1 - local equinox (MBS)". The menu bar includes "General", "Bundles", and "Help". The toolbar contains various icons for navigation and management. The main area is divided into a "Bundles" panel on the left and a "Property" table on the right.

mConsole main panel (indicated by a blue box on the left)

Bundle tree (indicated by a blue box on the left)

Property Table:

Property	Value
Framework Boot File	Not Defined
Framework Build Version	Not Defined
Framework Console	Not Defined
Framework Security State	Not Defined
Framework Specification Ve...	1.3.0
Framework Storage Directo...	Not Defined
Framework Storage Imple...	Not Defined
Framework Storage Native ...	Not Defined
Framework Vendor	Eclipse
JVM Vendor	Sun Microsystems Inc.
JVM Vendor URL	http://java.sun.com/
JVM Version	1.5.0_06
Language	en
mBS Product Version	Not Defined

Below the table, there are two input fields with "Set" buttons:

- Framework Start Level: 6
- Initial Bundle Start Level: 1

At the bottom, there are tabs for "Framework Properties" and "Runtime Info", and a status bar showing "Ready" and "Server Type: MBS".

Bonus Track



JSR 294: Improved Modularity Support in the Java™ Programming Language

⑦ D'après Rick Hall

⑦ Rational

- ⑦ Java programming language needs better support for hierarchical, modular organization
 - ⑦ Primarily to support information hiding
 - ⑦ Java packages inadequate for this purpose
 - ⑦ Only two levels of visibility: internal to the package or public to everyone

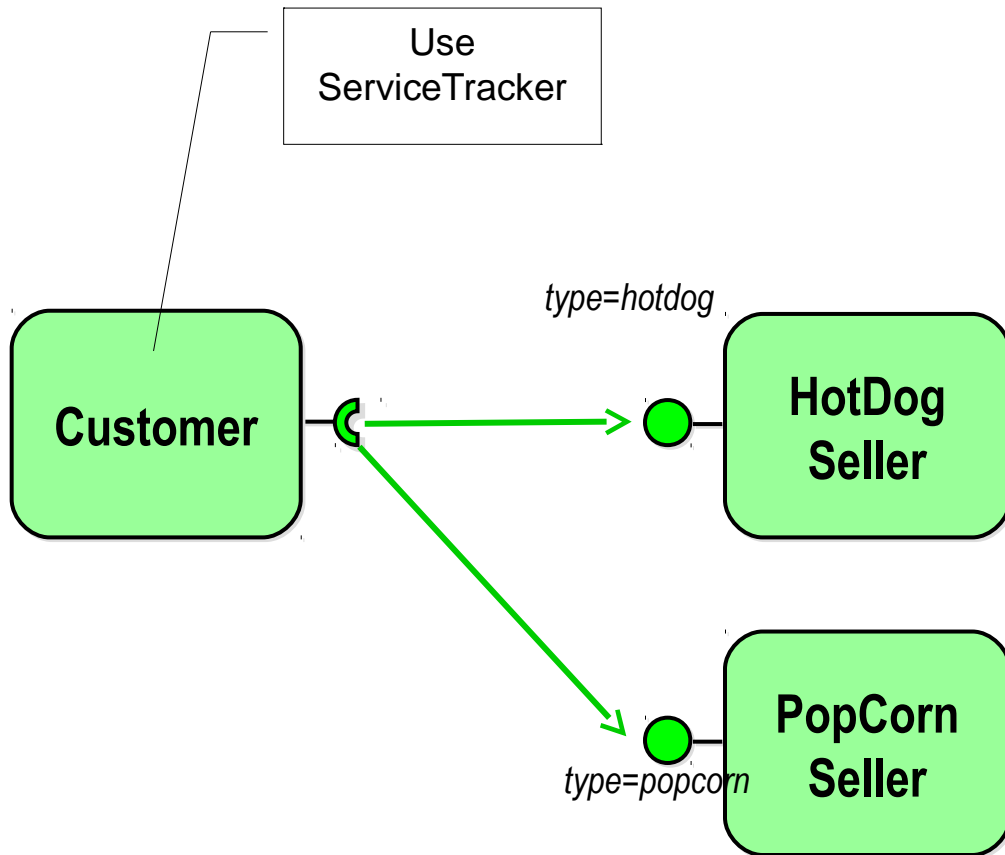
⑦ Module “Files”

```
super package com.foo.moduleA {  
    // Exported classes/interfaces  
    export com.foo.moduleA.api.*;  
    export com.foo.moduleA.ifc.InterfaceC;  
    // Imported modules  
    import org.bar.moduleD;  
    // Module membership  
    com.foo.moduleA.api;  
    com.foo.moduleA.ifc;  
    org.apache.stuff;  
}
```


Application à développer

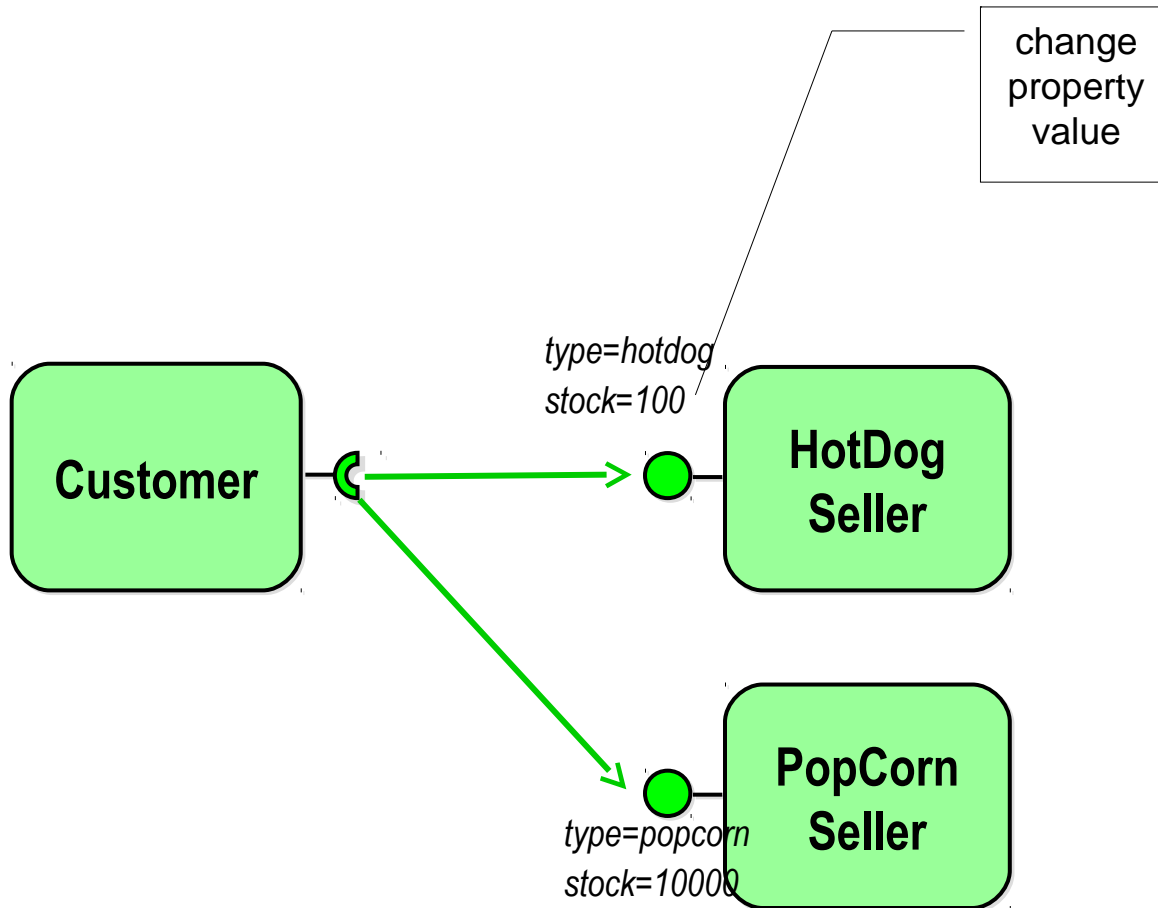
Ecole d'été OSGi 2007

Application 1

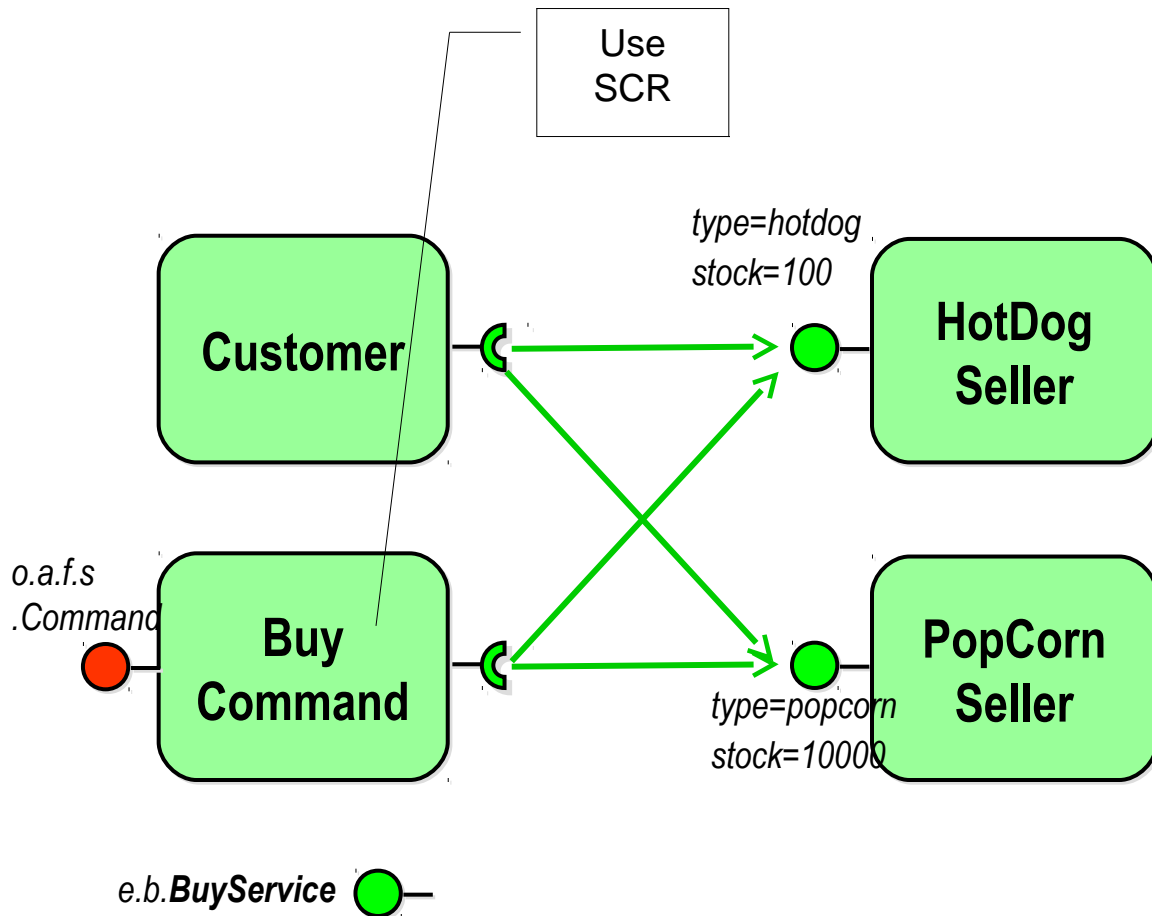


e.b.BuyService

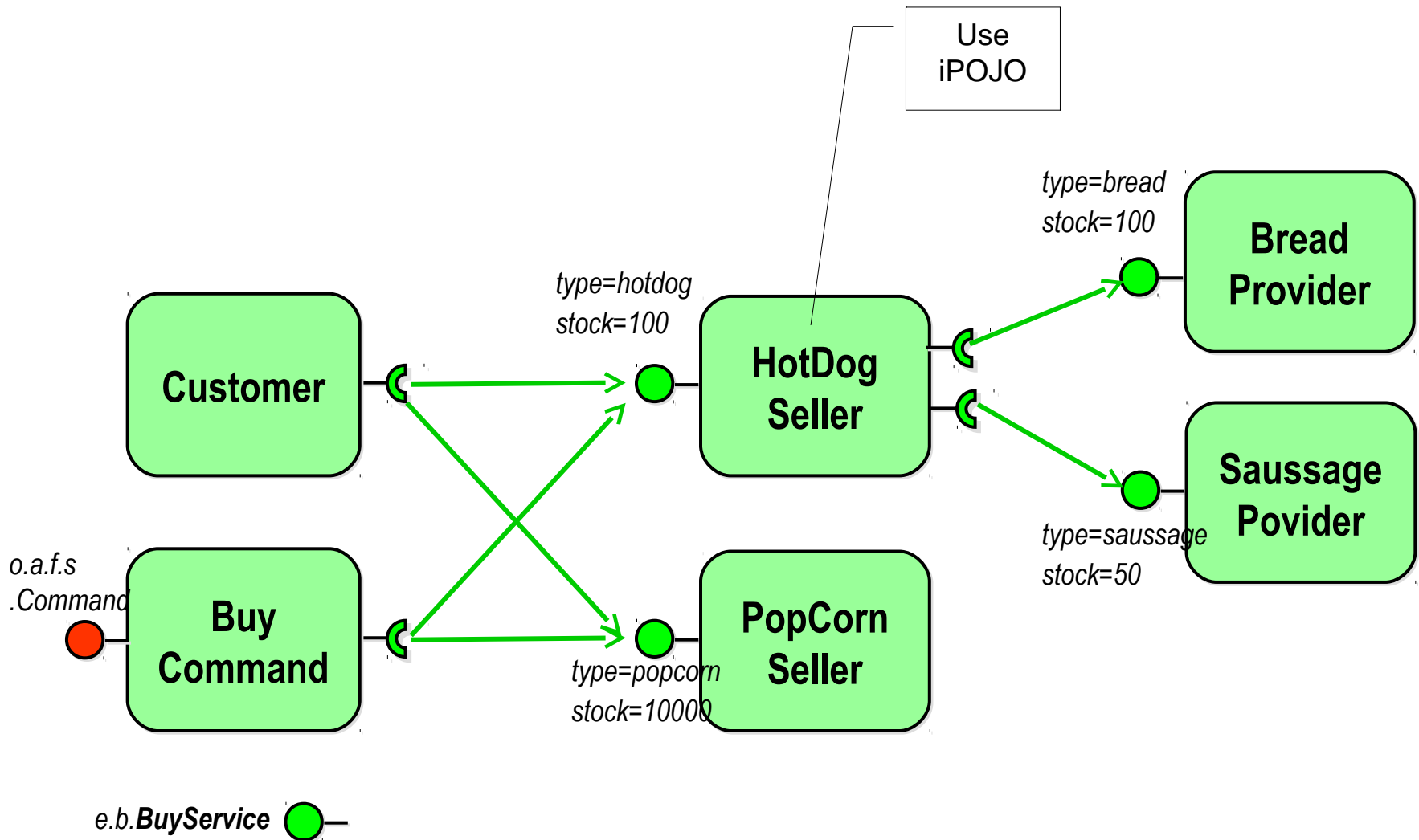
Application 2



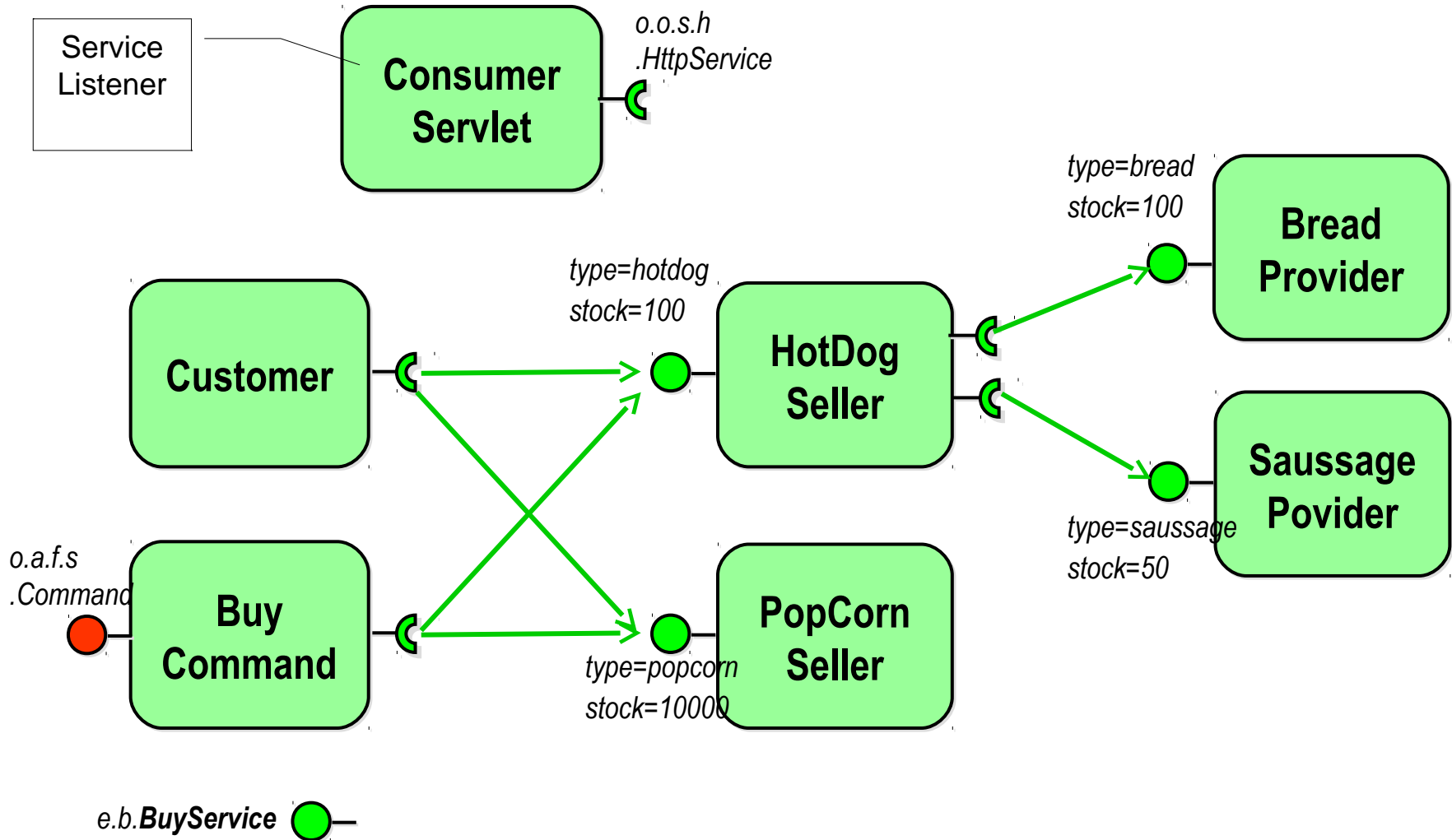
Application 3



Application 4



Application 5



RFC 138 Multiple Frameworks In One JVM

⑦ Motivations

⑦ Limitations

- ⑦ Multiplex singleton factories (URLHandler, XML, ...)
- ⑦ Multiplex System properties
- ⑦ Security Manager for Conditional Permission Admin

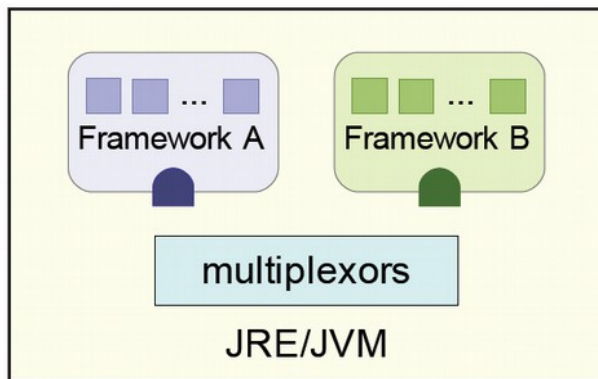
⑦ Solutions

- ⑦ Several FW (several vendors) hosted in the same JVM

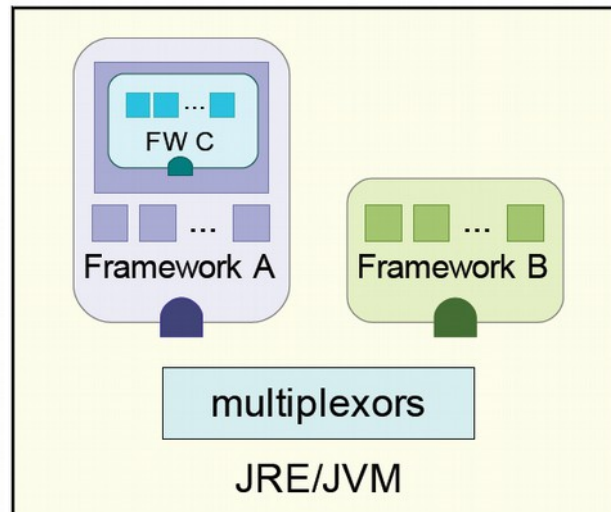
RFC 138 Multiple Frameworks In One JVM

⑦ Configurations

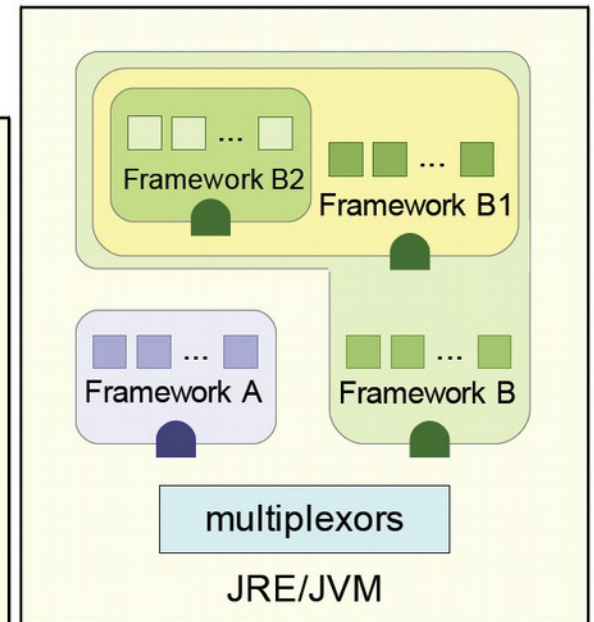
Peer embedded frameworks



Nested frameworks



Parent-Child frameworks



■ — bundle ■ — System bundle

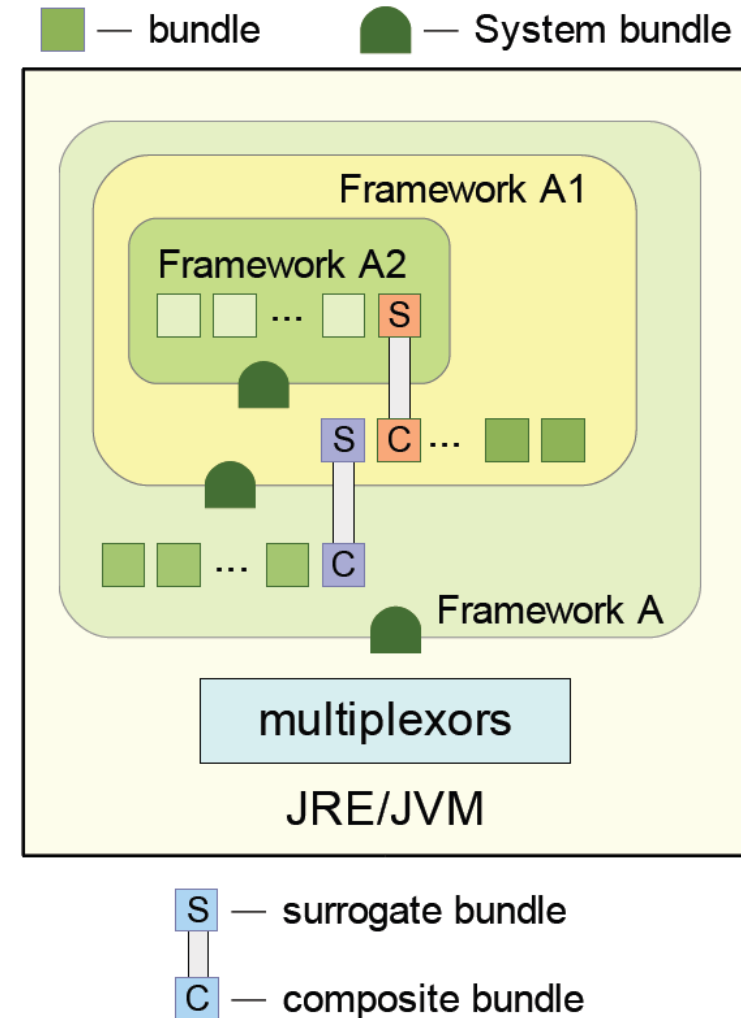
RFC 138

⑦ Surrogate Bundle

⑦ Represents the parent in the child FW

⑦ Composite Bundle

⑦ Represents a child in a parent FW



⑦ Lifecycle

Import / Export

⑦ **Package import**

⑦ **Package export**

Import / Export

- ⑦ **Parent Service import**
 - ⑦ **Filtering** CompositeServiceFilter-Import
- ⑦ **Child Service export (to the parent)**
 - ⑦ **Filtering** CompositeServiceFilter-Export
- ⑦ **service.id (globally unique)**

RFC 139 Distributed OSGi

Distributed OSGi (RFC 119 R4.2 Early 3)

⑦ Motivation

⑦ Pluggable

- ⑦ Transport protocol

- ⑦ Registry

⑦ Reuse

- ⑦ Metadata (SCA)

⑦ Description

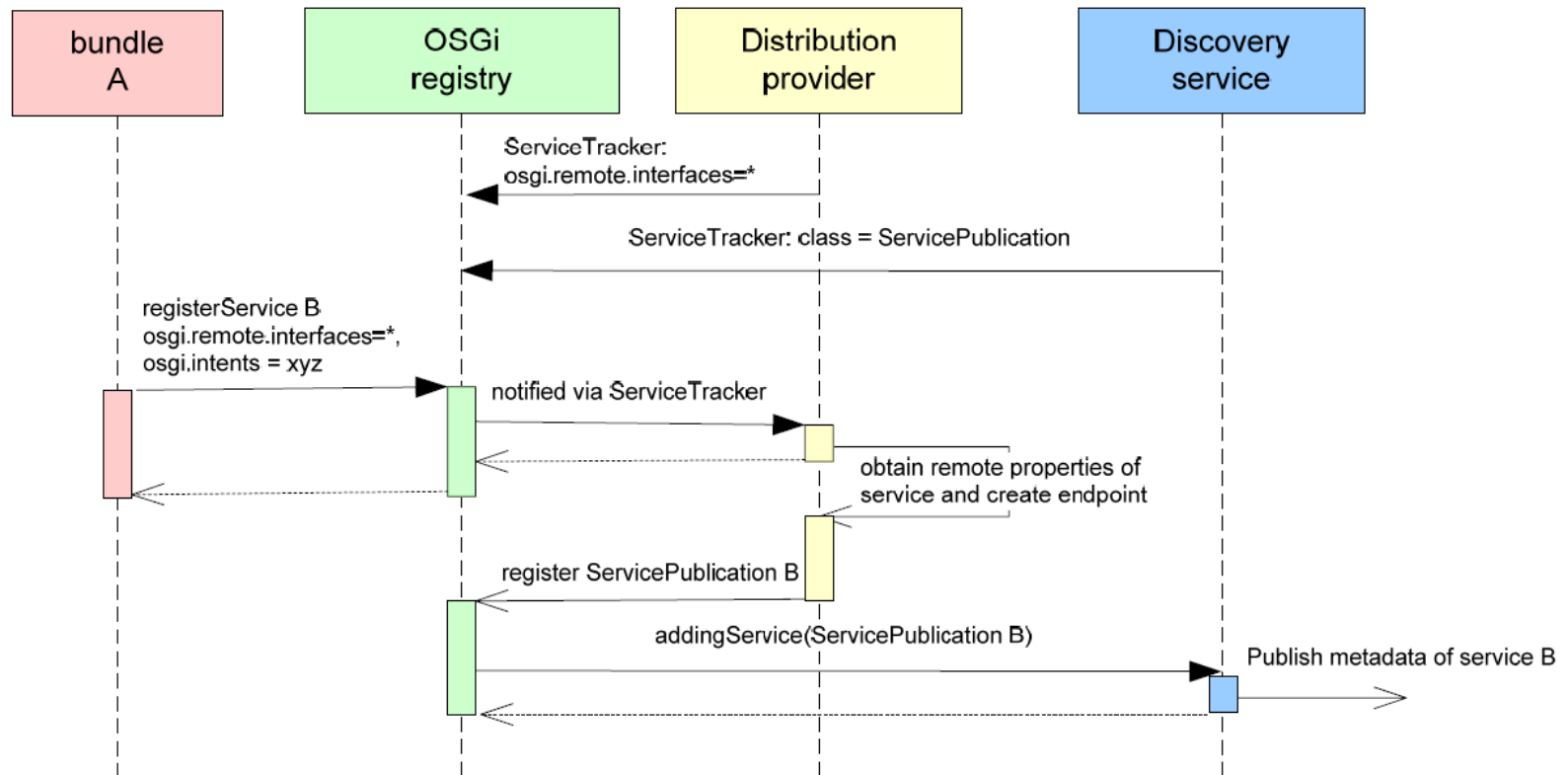
⑦ API

⑦ Extender

- ⑦ [OSGI-INF/remote-service/remote-services.xml](#)

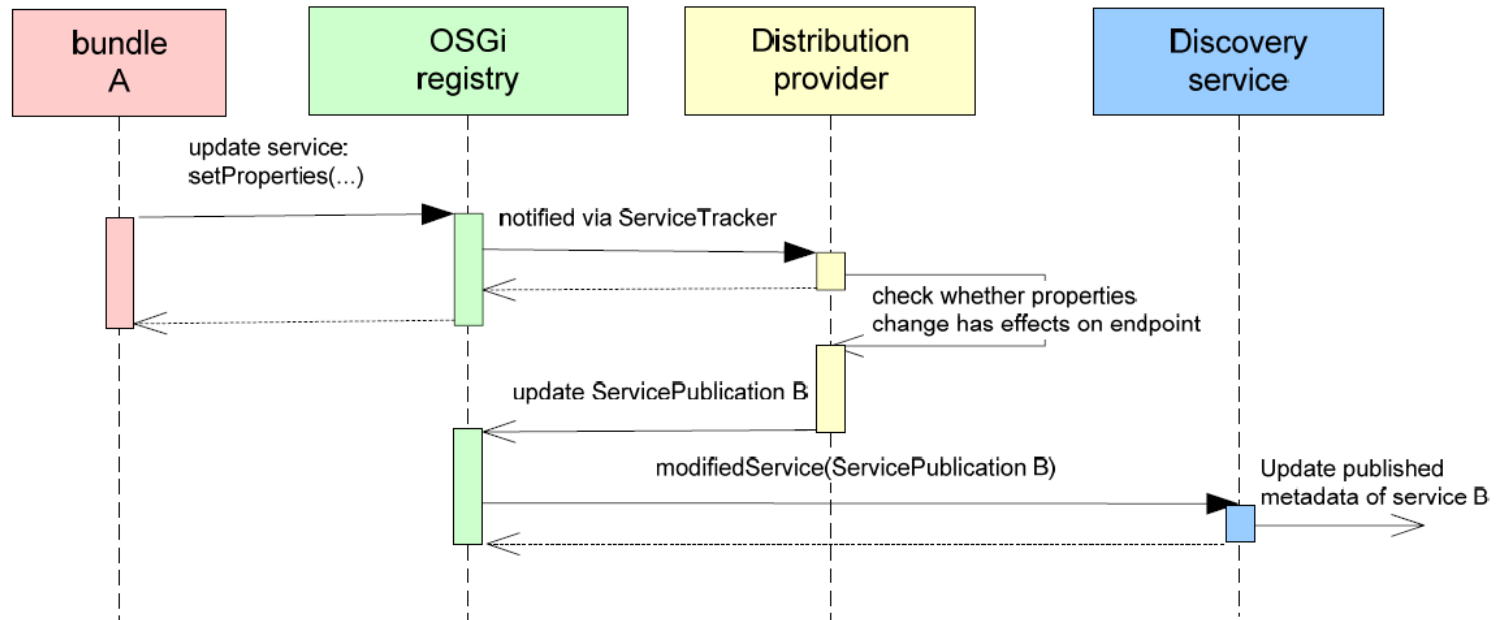
D-OSGi

Server side service registration



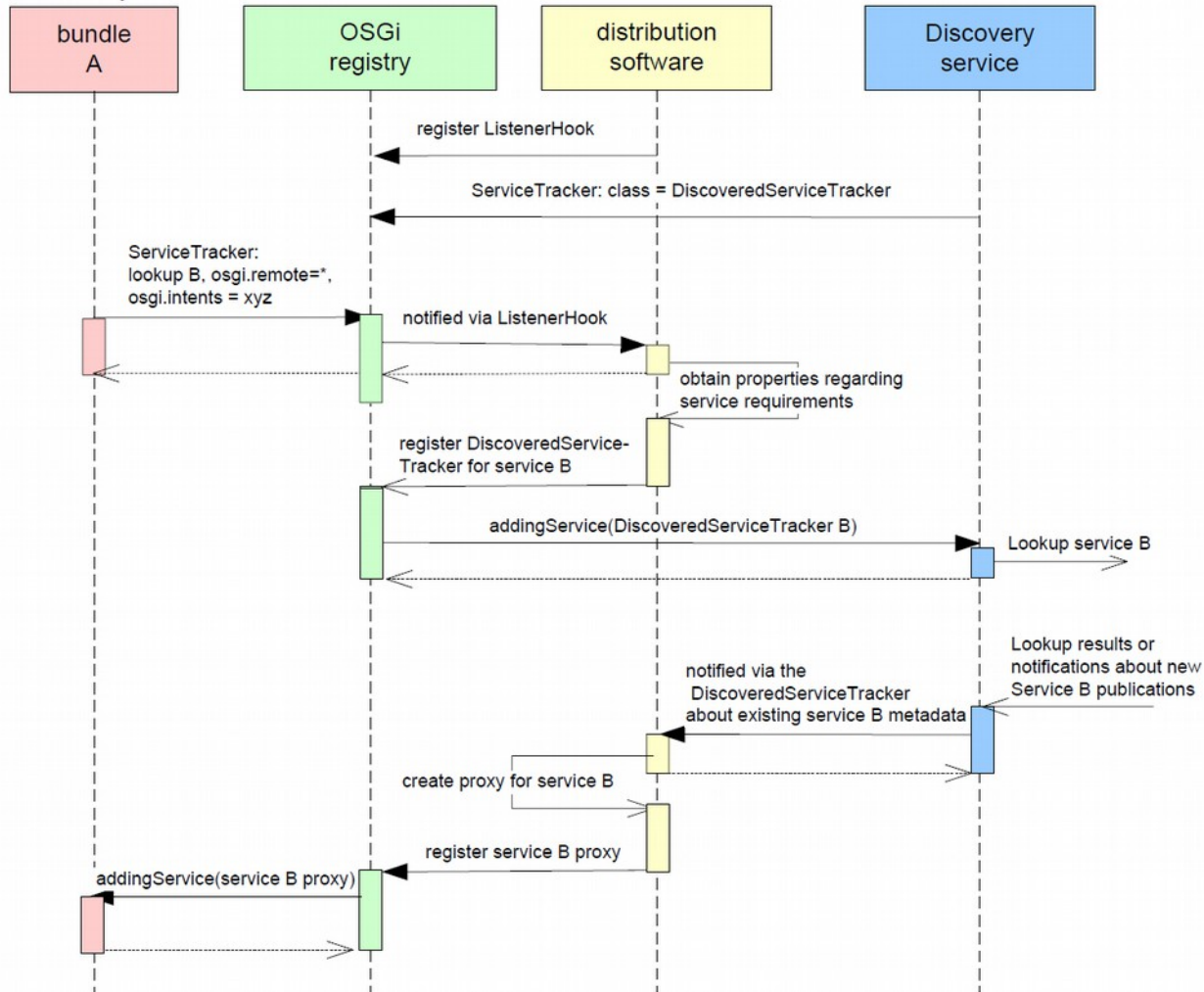
D-OSGi

Update of properties of an exposed service



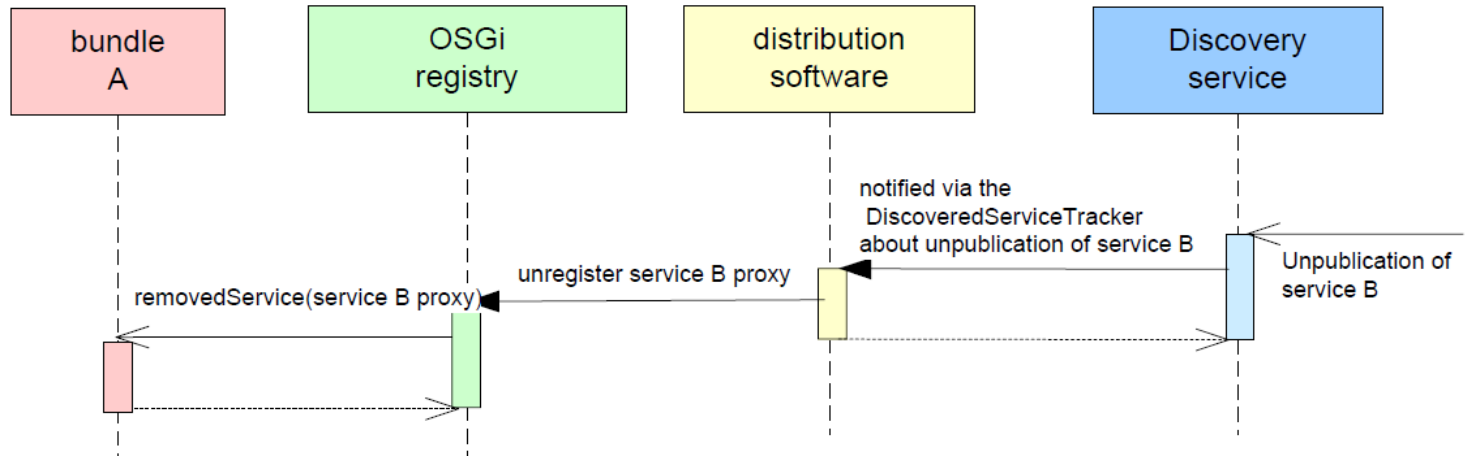
D-OSGi

Client side service lookup



D-OSGi

Service unregistration



D-OSGi implementation

⑦ Apache CXF D-OSGi RI

⑦ <http://cxf.apache.org/distributed-osgi.html>

⑦ provides the Reference Implementation of the Distribution Software (DSW) component of the Distributed OSGi Specification

⑦ using Web Services, leveraging SOAP over HTTP and exposing the Web Service over a WSDL contract.

⑦ No registry

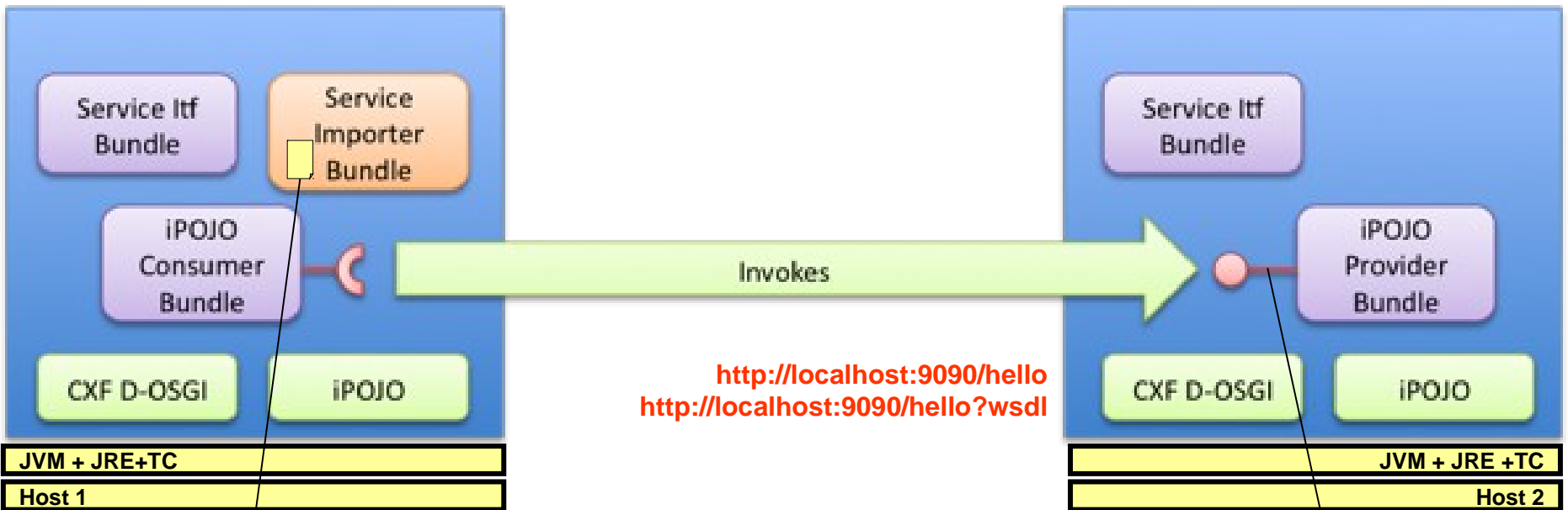
⑦ OW2 Chameleon ROSE

⑦ HTTP/SOAP (IIOP soon)

⑦ Registry: DNS-SD (LDAP soon)

⑦ ...

Apache CXF D-OSGi RI with iPOJO



<http://localhost:9090/hello>
<http://localhost:9090/hello?wsdl>

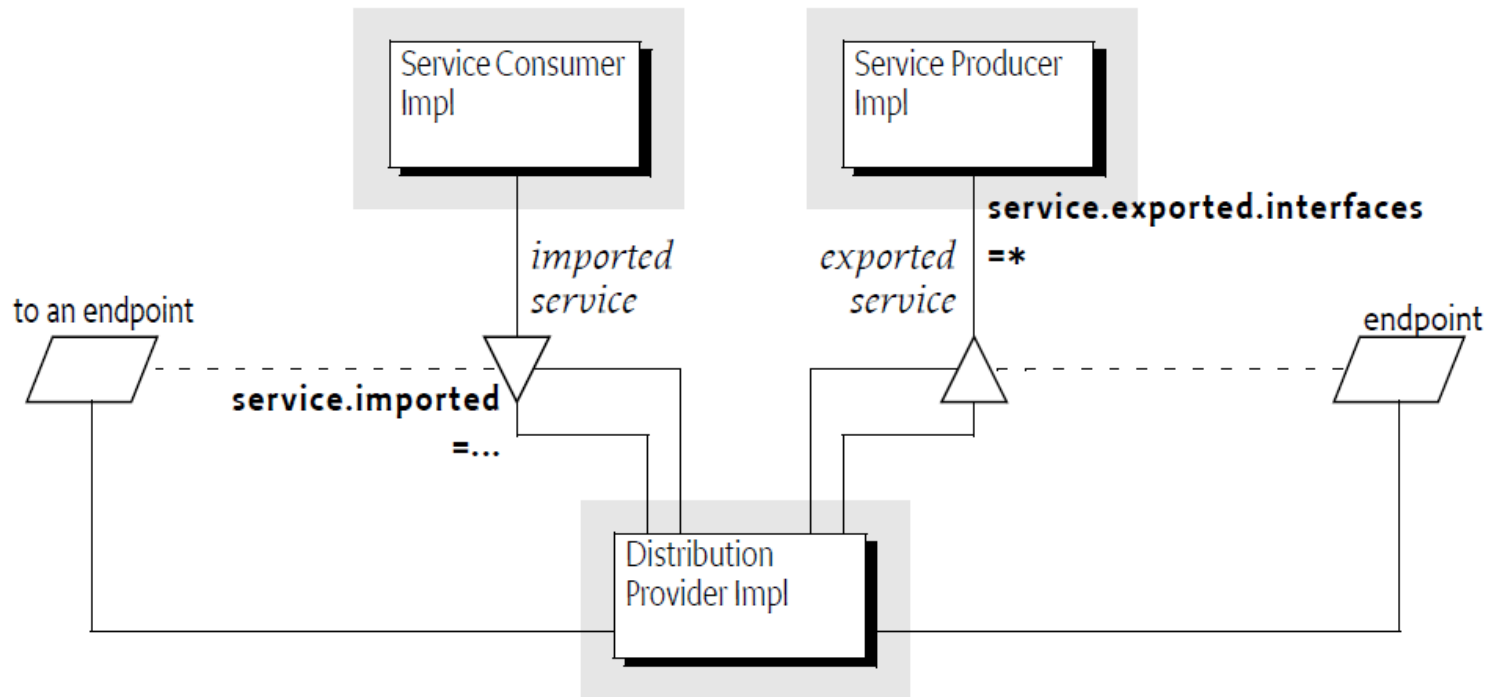
```
<service-descriptions xmlns="http://www.osgi.org/xmlns/sd/v1.0.0">
  <service-description>
    <provide interface="hello.HelloService"/>
    <property name="osgi.remote.interfaces">*</property>
    <property name="osgi.remote.conf.guration.type">pojo</property>
    <property name="osgi.remote.conf.guration.pojo.address">
      http://localhost:9090/hello</property>
    </service-description>
  </service-descriptions>
```

```
<instance component="hello.impl.HelloServiceImpl">
  <property name="osgi.remote.interfaces" value="*" />
  <property name="osgi.remote.conf.guration.type" value="pojo" />
  <property name="osgi.remote.conf.guration.pojo.address"
    value="http://localhost:9090/hello" />
</instance>
```

Remote Service (R4.2 final)

⑦ Simplification of D-OSGi (R4.2 early 3)

- ⑦ Whiteboard pattern with special properties
- ⑦ No service discovery



Eclipse Communication Framework (ECF)

<http://www.eclipse.org/ecf/>

- ⑦ « *ECF is a framework for supporting the development of distributed Eclipse-based tools and applications. It can be used to create other plugins, tools, or full Eclipse RCP applications that require asynchronous point-to-point or publish-and-subscribe messaging.* »
- ⑦ **Architecture TODO**
 - ⑦ **IContainer, IFactory, IPresenceContainer ...**

Eclipse Communication Framework (ECF) Example

⑦ Container creation, ID creation, container adapter, and connection

```
// Create the new container
IContainer client = ContainerFactory.getDefault().createContainer(containerType);
// Create the targetID
ID targetID = IDFactory.getDefault().createID(client.getConnectNamespace(), uri);
// Check for IPresenceContainer....if it is, setup presence UI, if not setup shared object
// container
IPresenceContainer pc = (IPresenceContainer)
    client.getAdapter(IPresenceContainer.class);
if (pc != null) {
    // Setup presence UI
    presenceContainerUI = new PresenceContainerUI(pc);
    presenceContainerUI.setup(client, targetID, username);
} else throw new NullPointerException("IPresenceContainer interface not exposed by
    client with type "+containerType);
// connect
client.connect(targetID, getJoinContext(username, connectData));
...
// dispose
client.dispose();
```

Distributed Shared Memory (DSM)

⑦ **Terracotta (Java)**

⑦ **Oracle Coherence (Java, .NET, C++)**

⑦ <http://www.oracle.com/technology/products/coherence/index.html>

⑦ **3rd Space (Java)**

⑦ <http://code.google.com/p/3rd-space>

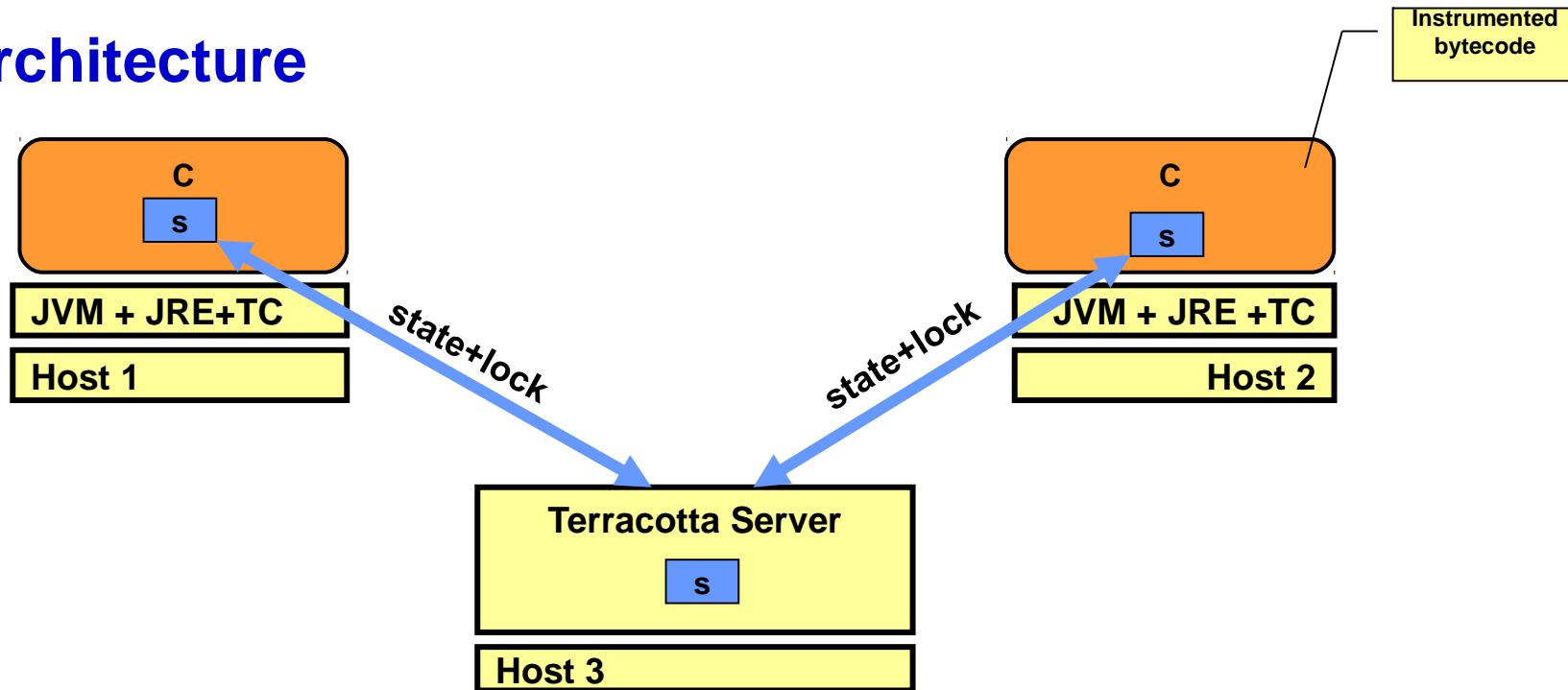
⑦ <http://jist.ece.cornell.edu/docs/040517-thesis.pdf>

⑦ ...

⑦ Distributed Shared Memory for Java

- ⑦ For clustering, cache replication (hibernate), Http session (tomcat), visual apps (dashboard) ...

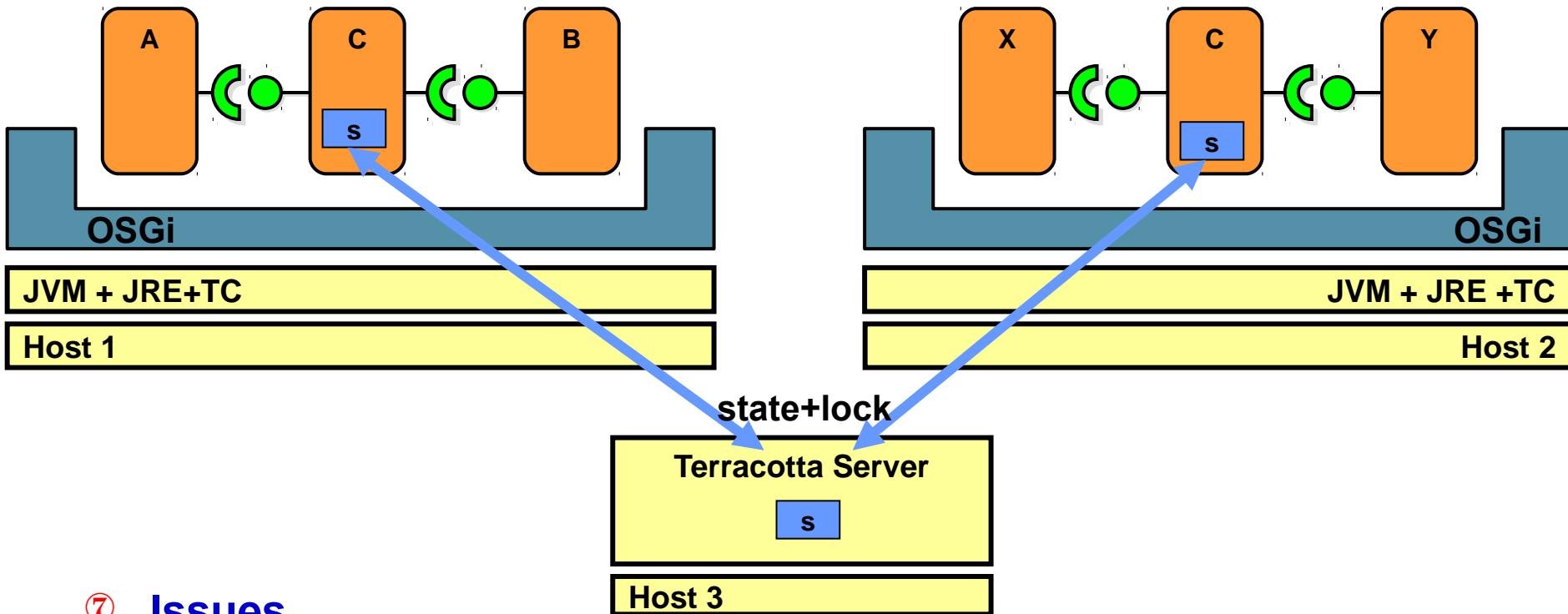
⑦ Architecture



Terracotta + OSGi

⑦ Architecture

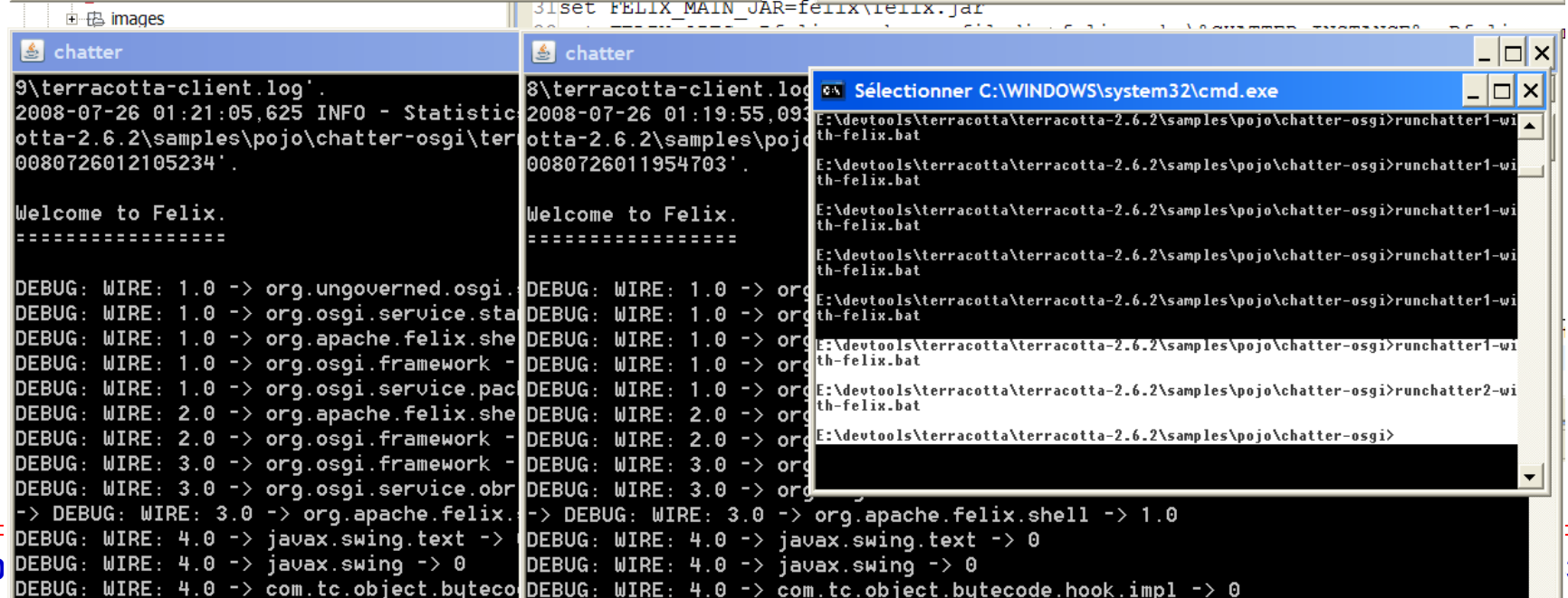
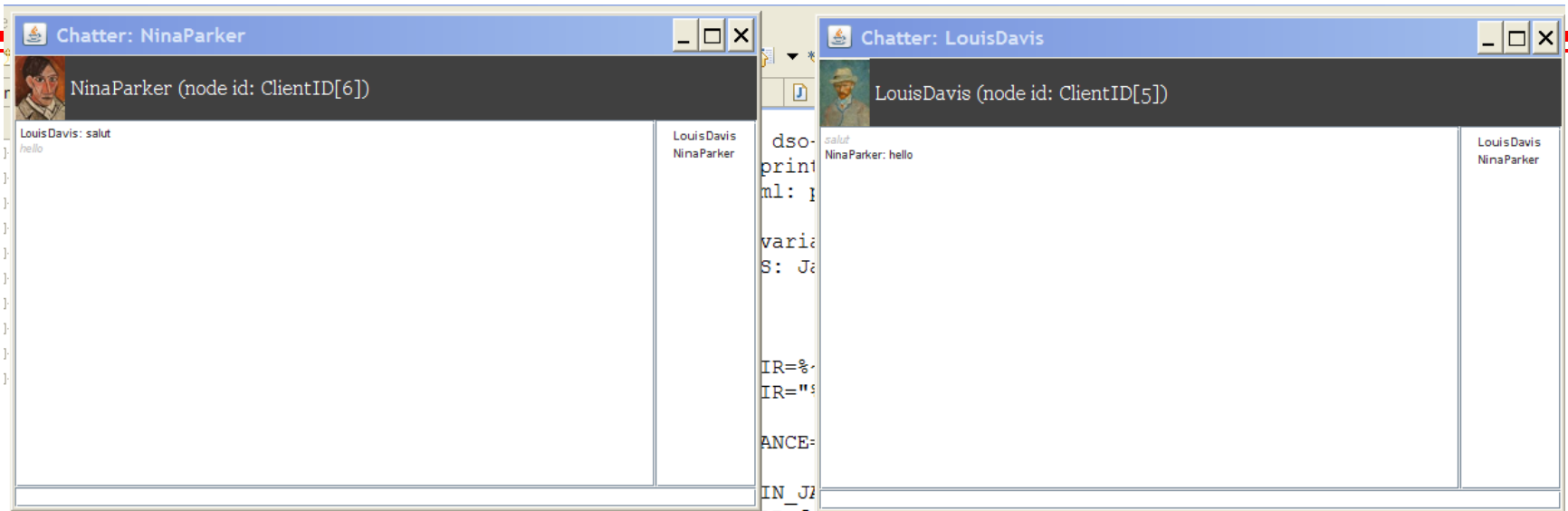
- ⑦ Terracotta Integration Module (TIM) for OSGi
- ⑦ Clustered field (in a bundle C) shared with Terracotta-ed OSGi FWs



⑦ Issues

- ⑦ Dynamic provisioning (tc-conf g.xml), Dynamic updates

Terracotta + OSGi Example (Chatter)



Extra



Provisioning and Deployment



Provisioning and Deployment

⑦ **Notions**

- ⑦ **Initial provisioning**

- ⑦ **Deployment**

 - ⑦ **Resolution**

 - ⑦ **Install**

 - ⑦ **Start**

 - ⑦ **Update**

 - ⑦ **Uninstall**

- ⑦ **Dependencies**

 - ⑦ **Mandatory (requires)**

 - ⑦ **Optional**

 - ⑦ **Package, Service, ...**

- ⑦ **Bundles**

 - ⑦ **Fragment, RequiredBundle**

- ⑦ **Repository**

Provisioning and Deployment

⑦ Repositories

- ⑦ Maven repository
- ⑦ OBR index
- ⑦ Apache Ivy

⑦ Systems

- ⑦ Deployment Admin
- ⑦ Maven repository (Maven URL Handler)
- ⑦ OBR (RFC ???)
- ⑦ *Apache Felix* File Install
- ⑦ *Apache ACE*
- ⑦ OW2 JOnAS Deployment Plan
- ⑦ Ops4j Pax Runner

⑦ Remark

- ⑦ A bundle is not necessary a compilation artifact (embedded jar files)

OSGi Bundle Repository (OBR) RFC 112

⑦ Motivation

- ⑦ Deploy (and start) a bundle and its « dependencies »
- ⑦ Dependencies
 - ⑦ import → export packages, import → export services, ...
- ⑦ Pluggable resolvers
- ⑦ Multiple repositories (index + artifacts' urls)
 - ⑦ <http://www.osgi.org/obr/browse>
 - ⑦ <http://felix.apache.org/obr/releases.xml>
 - ⑦ [.m2/repository/repository.xml](http://felix.apache.org/obr/m2/repository/repository.xml)

OSGi Bundle Repository (OBR) RFC 112

Tools

⑦ Building

- ⑦ Bindex

- ⑦ Maven OBR plugin

⑦ Command (Felix implementation)

- ⑦ Repository management

 - ⑦ `obr add-url [<repository-f le-url> ...]`

 - ⑦ `obr refresh-url [<repository-f le-url> ...]`

 - ⑦ `obr remove-url [<repository-f le-url> ...]`

 - ⑦ `obr list-url`

- ⑦ Search

 - ⑦ `obr list [-v] [<string> ...]`

 - ⑦ `obr info <bundle-name>|<bundle-symbolic-name>|<bundle-id>[;<version>] ...`

- ⑦ Deploy

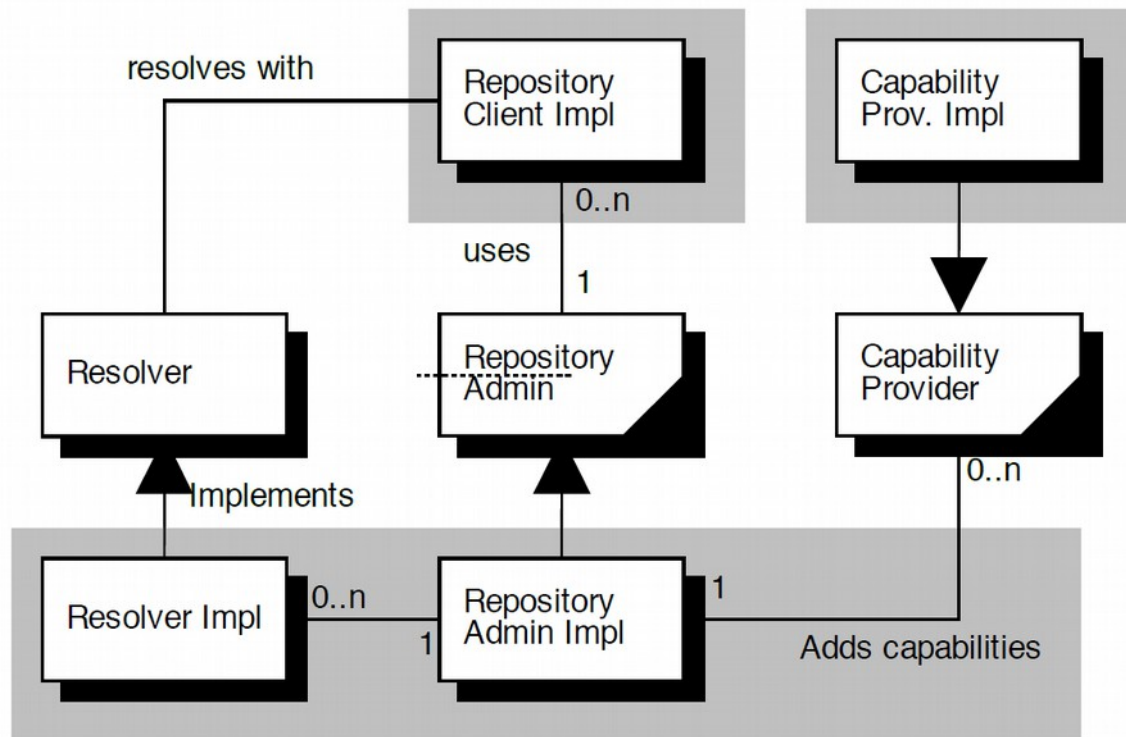
 - ⑦ `obr deploy <bundle-name>|<bundle-symbolic-name>|<bundle-id>[;<version>] ...`

 - ⑦ `obr start <bundle-name>|<bundle-symbolic-name>|<bundle-id>[;<version>] ...`

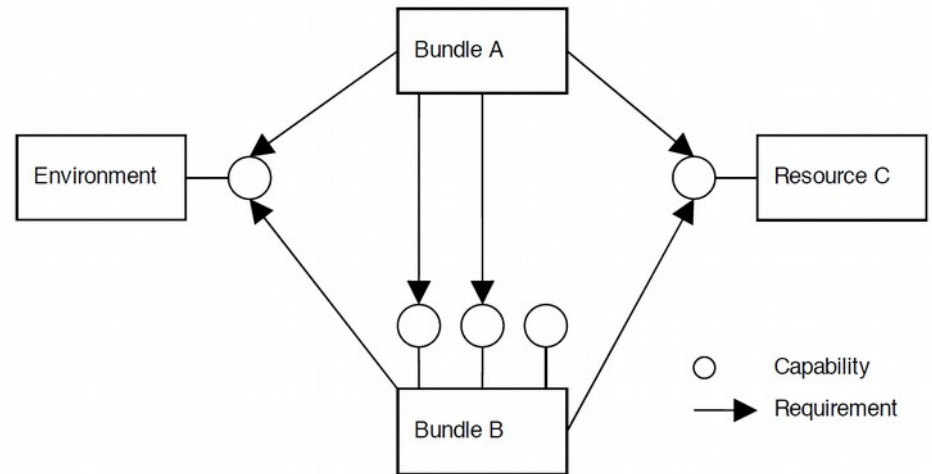
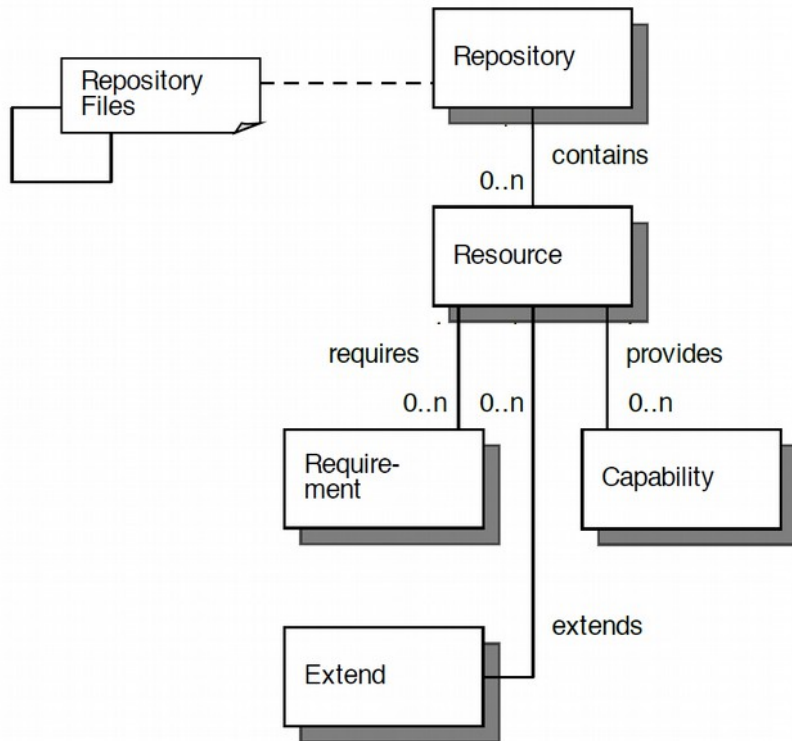
⑦ Web search

- ⑦ <http://www.osgi.org/obr/browse>

OBR



Requirements and Capabilities



Pax URL

<http://wiki.ops4j.org/display/paxurl/Pax+URL>

⑦ Provides URL handlers for bundle deployment

⑦ mvm

⑦ deploy bundles or simple jar files from a Maven 2 repository

- ⑦ mvn:org.ops4j.pax.web.bundles/service/0.2.0
- ⑦ mvn:org.ops4j.pax.web.bundles/service
- ⑦ mvn:org.ops4j.pax.web.bundles/service/LATEST
- ⑦ mvn:http://repository.ops4j.org/maven2!org.ops4j.pax.web.bundles/service/0.2.0

⑦ wrap

⑦ process your legacy jar at runtime and transform it into an OSGi bundle

- ⑦ wrap:file:commons-logging-1.1.jar
- ⑦ wrap:mvn:commons-logging/commons-logging/1.1
- ⑦ wrap:mvn:commons-logging/commons-logging/1.1,file:commons-logging-1.1.bnd
- ⑦ wrap:mvn:commons-logging/commons-logging/1.1\$Bundle-SymbolicName=JCL&BundleVersion=1.1

Pax URL

<http://wiki.ops4j.org/display/paxurl/Pax+URL>

⑦ classpath

⑦ access to resources from thread classpath or all installed bundles or a specific bundle

⑦ `classpath:my-bundle.conf`

⑦ `classpath:my-bundle/my-bundle.conf`

⑦ assembly

⑦ *assembling* a jar on the fly out of files from a directory, another jar file and/or a zip file, in any combination

⑦ `assembly:./my-project/bin`

⑦ `assembly:/Users/pax/Projects/my-project.jar`

⑦ `assembly:mvn:my-project-group/my-project-artifact`

⑦ `assembly:./my-project/bin/**/*.class`

⑦ `assembly:./my-project/bin/**/*.class&META-INF/**`

⑦ `assembly:./my-project/bin/**/*.class&!/*.java`

⑦ `assemblyref:./my-project/assembly.json`

```
{
  "assembly" : {
    "directory" : {
      "path" : "./my-project/bin",
      "include" : "**/*.class",
      "include" : "META-INF/**"
    }
  },
  "manifest" : "./my-project/MANIFEST.MF"
}
```

Pax URL

⑦ Other

- ⑦ War

- ⑦ Dir

- ⑦ Link

- ⑦ Ivy

- ⑦ Obr

 - ⑦ `obr:org.eclipse.equinox.http`

 - ⑦ `obr:org.eclipse.equinox.http/1.0.100.v20070423`

Pax Runner

⑦ Provides scripts to setup initial provisioning configurations for Felix, Equinox and KF

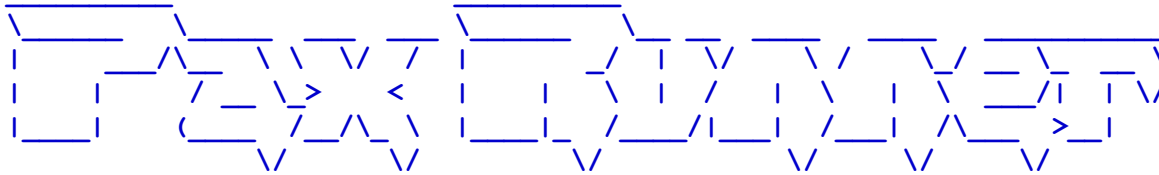
⑦ Scripts

- ⑦ `pax-run --platform=felix`
- ⑦ `pax-run --platform=felix --version=2.0.0`
- ⑦ `pax-run --platform=felix --snapshot`
- ⑦ `pax-run --platform=equinox`
- ⑦ `pax-run --platform=equinox --version=3.2.1`
- ⑦ `pax-run --platform=knopferfsch`
- ⑦ `pax-run --platform=conciierge`

⑦ Eclipse Plugin

Pax Runner

```
E:\devtools\pax-runner-1.2.0\bin>pax-run
```



```
Pax Runner (1.2.0) from OPS4J - http://www.ops4j.org  
-----
```

```
-> Using config [classpath:META-INF/runner.properties]  
-> Using only arguments from command line  
-> Preparing framework [Felix 2.0.0]  
-> Downloading bundles...  
-> Felix 2.0.0 : 389149 bytes @ [ 100kBps ]  
-> Felix Shell (1.4.0) : 60833 bytes @ [ 94kBps ]  
-> Felix TUI Shell (1.4.0) : 12735 bytes @ [ 12735kBps ]  
-> Using execution environment [J2SE-1.6]  
-> Runner has successfully finished his job!
```

```
Welcome to Felix
```

```
=====
```

```
-> ps
```

```
START LEVEL 6
```

ID	State	Level	Name
[0]	[Active] [0]	System Bundle (2.0.0)
[1]	[Active] [1]	Apache Felix Shell Service (1.4.0)
[2]	[Active] [1]	Apache Felix Shell TUI (1.4.0)

```
->
```

Apache Felix File Install

- ⑦ Install and start bundles and .jar+.bnd put in a directory (./load)

- ⑦ Configure/reconfigure ManagedServices
 - ⑦ <PID>.cfg

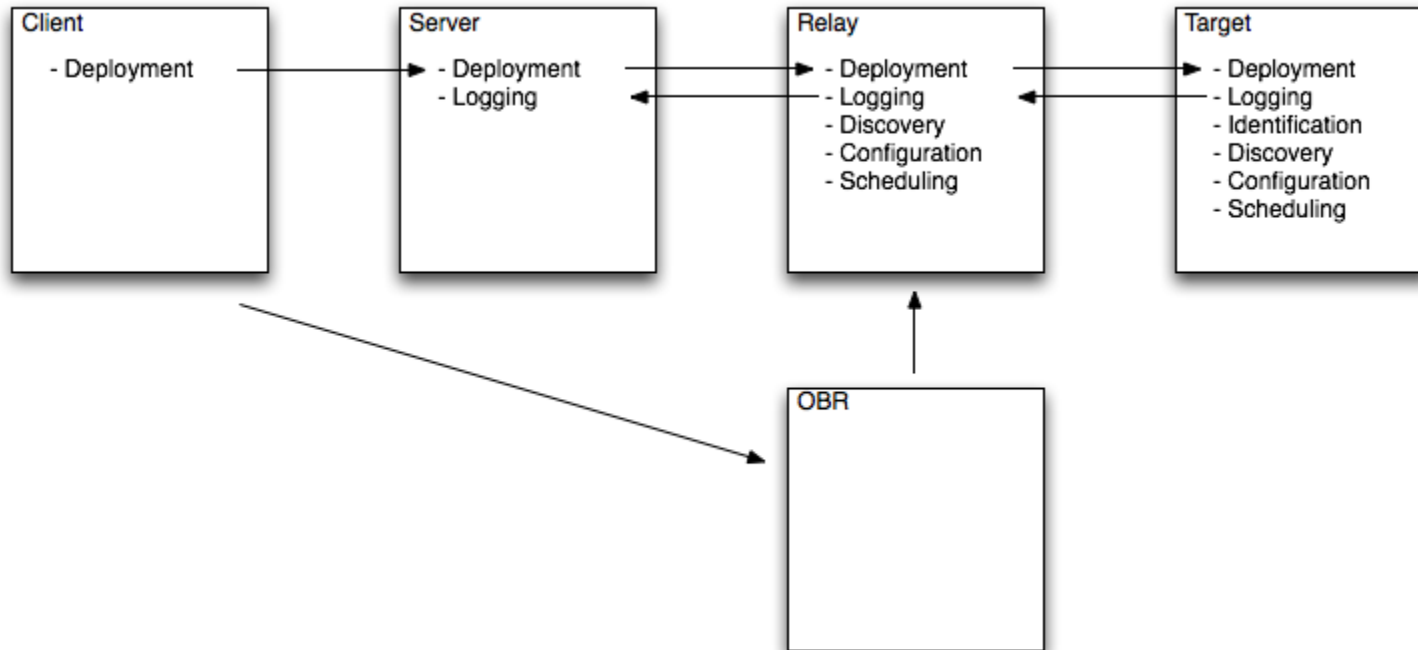
- ⑦ Uninstall then when they are removed of the directory

Apache ACE

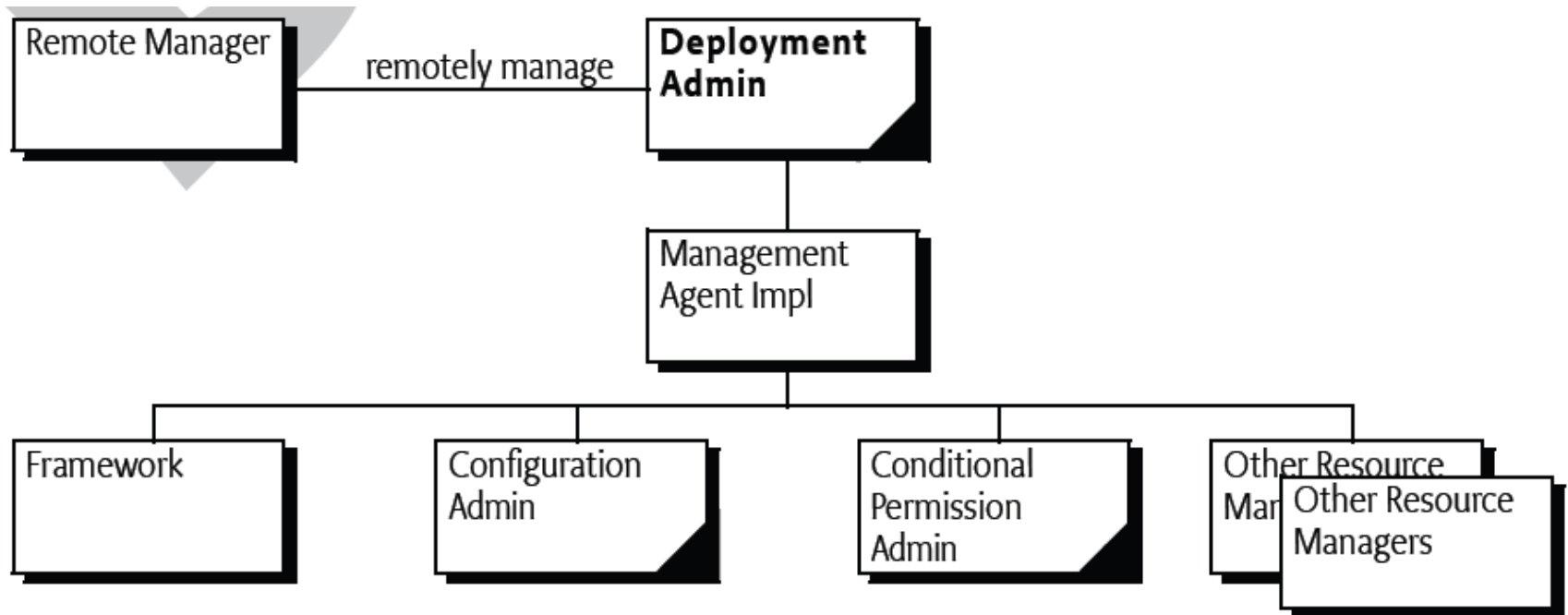
(<http://incubator.apache.org/ace/>)

- ⑦ **Software distribution framework that allows you to centrally manage and distribute software components, configuration data and other artifacts to target systems. It is built using OSGi and can be deployed in different topologies.**
 - ⑦ **As a mechanism to distribute and configure the runtime components (those implementing the enterprise OSGi application programming model).**
 - ⑦ **To distribute and configure enterprise OSGi application components implemented to the enterprise OSGi application programming model.**

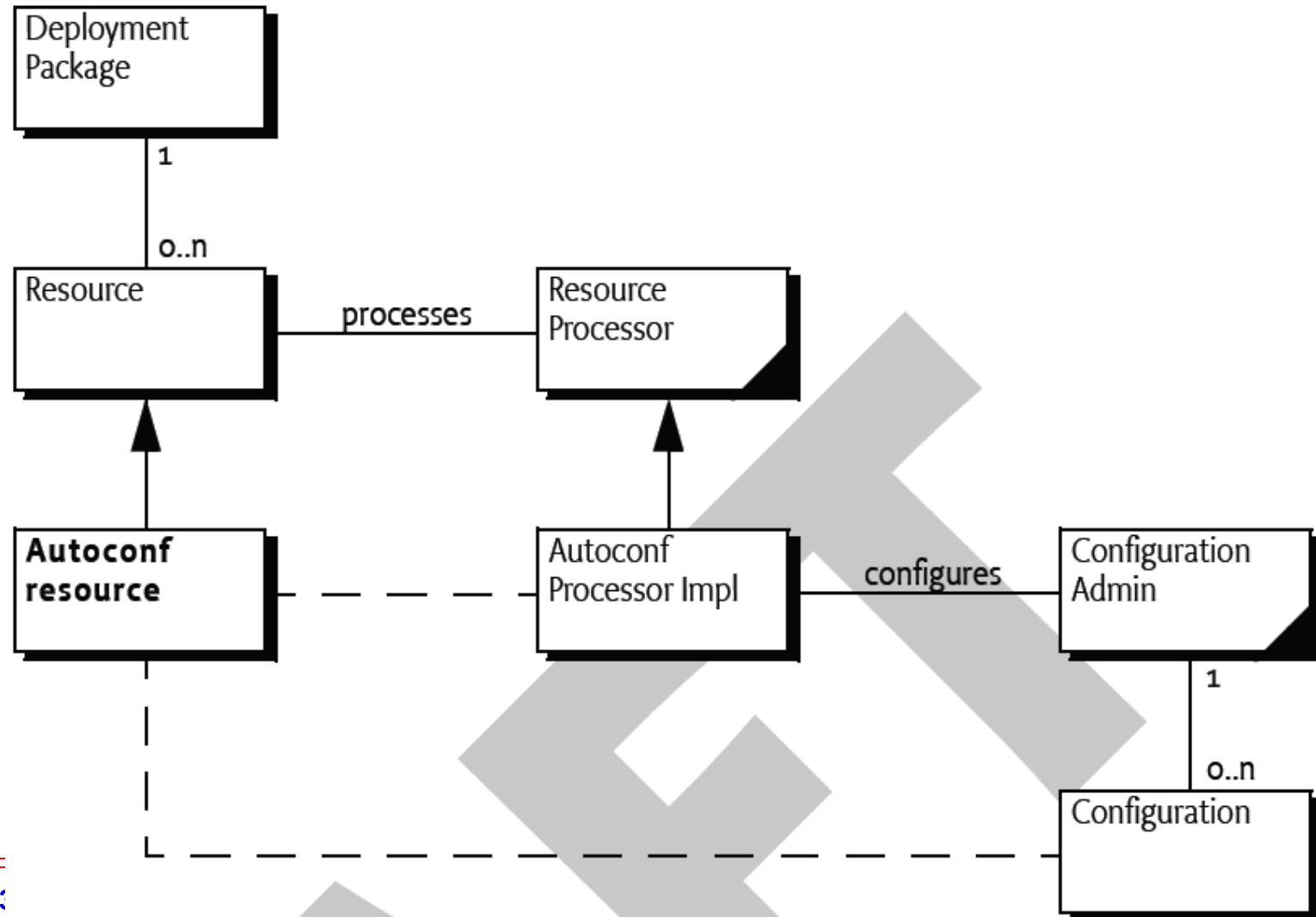
Apache ACE



114 Deployment Admin Specification



115 Auto Configuration Specification



JOnAS 5 Deployer

⑦ Motivation

- ⑦ Unified (dynamic) deployment of JavaEE and OSGi artifacts
 - ⑦ Bundles (JOnAS services are provisionned in bundles)
 - ⑦ EAR, WAR, EjbJar, RAR

⑦ Features

- ⑦ File install
- ⑦ Deployment plans
- ⑦ Update center
 - ⑦ management of grids of JOnAS

JOnAS 5 Deployer

⑦ Deployment plan

- ⑦ Sequence of artifacts to deploy (and start with startlevel)
- ⑦ Could be used for JavaEE server execution profiles
 - ⑦ From μ JOnAS to clustered JOnASs

⑦ Support for multiple repositories

- ⑦ Maven2
- ⑦ OBR
- ⑦ URL

⑦ Architecture

- ⑦ Pluggable repository managers
- ⑦ Pluggable deployers

⑦ Tools

- ⑦ JonasAdmin web interface
- ⑦ `jonas admin -a filename`
- ⑦ Update center

JOnAS 5 Deployer Repository

```
<?xml version="1.0" encoding="UTF-8"?>
<repositories
  xmlns="http://jonas.ow2.org/ns/deployment-plan/repositories/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="repositories-1.0.xsd">

  <repository id="maven-central">
    <type>maven2</type>
    <url>http://repo1.maven.org/maven2/</url>
  </repository>

  <repository id="http-repo">
    <type>url</type>
    <url>http://localhost/</url>
  </repository>

  <repository id="obr-repo">
    <type>obr</type>
    <url>file:///C:/obr/</url>
  </repository>

</repositories>
```

JOnAS 5 Deployment Plan

Deployment-Plan (URL & Maven2)

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment-plan xmlns="http://jonas.ow2.org/ns/deployment-plan/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:url="http://jonas.ow2.org/ns/deployment-plan/url/1.0"
  xmlns:m2="http://jonas.ow2.org/ns/deployment-plan/maven2/1.0"
  xmlns:obr="http://jonas.ow2.org/ns/deployment-plan/obr/1.0">

  <deployment xsi:type="url:url-deploymentType" id="dep1" reloadable="true">
    <url:resource>easybeans-example-statelessbean-1.0.1-SNAPSHOT.jar</url:resource>
  </deployment>

  <deployment xsi:type="m2:maven2-deploymentType" id="dep2">
    <m2:groupId>org.ow2.easybeans</m2:groupId>
    <m2:artifactId>easybeans-example-statefulbean</m2:artifactId>
    <m2:version>1.1.0-SNAPSHOT</m2:version>
  </deployment>

  <deployment xsi:type="m2:maven2-deploymentType" id="dep3">
    <m2:groupId>org.ow2.easybeans</m2:groupId>
    <m2:artifactId>example-server</m2:artifactId>
    <m2:version>1.1.0-SNAPSHOT</m2:version>
    <m2:type>ear</m2:type>
  </deployment>
</deployment-plan>
```

JOnAS 5 Deployer Deployment-Plan

⑦ OBR

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment-plan xmlns="http://jonas.ow2.org/ns/deployment-plan/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:obr="http://jonas.ow2.org/ns/deployment-plan/obr/1.0">

  <deployment xsi:type="obr:obr-deploymentType" id="dep-obr-1">
    <obr:bundle-symbolic-name>org.ow2.easybeans.examples.entitybean</obr:bundle-symbolic-name>
    <obr:bundle-version>=1.1.0-SNAPSHOT</obr:bundle-version>
  </deployment>

  <deployment xsi:type="obr:obr-deploymentType" id="dep-obr-2">
    <obr:f lter>(&(symbolicname=org.ow2.easybeans.examples.statefullbean)
      (version=1.1.0-SNAPSHOT))</obr:f lter>
  </deployment>

</deployment-plan>
```

JOnAS 5 deployer

Update center

⑦ Help tool for update operation

- ⑦ Hot update
- ⑦ At the next restart
- ⑦ Incremental (only for new sessions)

⑦ Process

- ⑦ Cluster topology initialisation
- ⑦ Target server selection
- ⑦ Target service to update
- ⑦ Monitor operation progress

⑦ Features

- ⑦ Resume/Rollback mechanism
- ⑦ Log operations
- ⑦ Planification

Equinox P2

<http://wiki.eclipse.org/Equinox/p2>

- ⑦ component of the Equinox project
- ⑦ provides a provisioning platform for Eclipse-based applications

- ⑦ **HMI**
 - ⑦ SWT
 - ⑦ Console
 - ⑦ Webapp
 - ⑦ http://wiki.eclipse.org/Equinox_p2_Webapp_QuickStart

End-to-End Management

⑦ Motivations

- ⑦ Gestion d'un (très grand) parc de plateformes OSGi
- ⑦ Et des équipements qui y sont attachés

⑦ Besoins

- ⑦ Déploiement « transactionnel »
- ⑦ Politiques de déploiement
- ⑦ ...

⑦ E2E Remote Management EG

- ⑦ Expert group a l'Alliance

⑦ Produits

- ⑦ Prosyst Remote Manager
- ⑦ IBM Expeditor Framework
- ⑦ OW2 JOnAS deployer and update center
- ⑦ ...

JMX and OSGi



OSGi and JMX

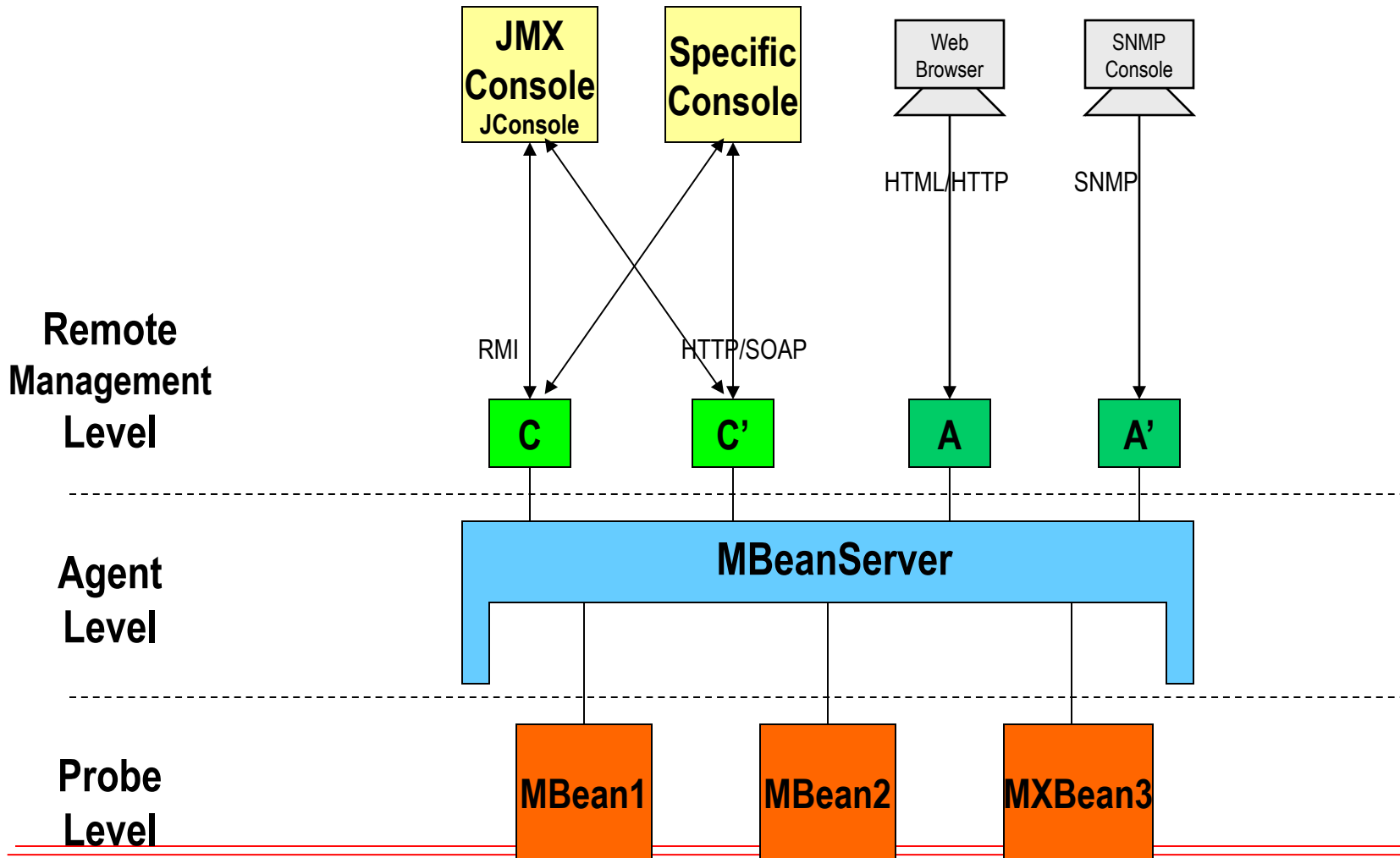
⑦ Motivation

- ⑦ Get and set applications configuration (Pull)
- ⑦ Collect statistics (Pull)
 - ⑦ Performance, Resources usage, Problems
- ⑦ Notify events (Push)
 - ⑦ Alerts, Faults, State changes

⑦ JMX : Java Management Extension

- ⑦ De facto standard for Java application monitoring
 - ⑦ Early adoption in JavaEE
 - ⑦ JMX consoles

JMX 3-Level Architecture



OSGi and JMX

⑦ Motivation

⑦ OSGi Framework monitoring

⑦ start/update/stop bundles, ...

⑦ Re

⑦ Modular and Dynamic MBean (incl. Adaptor and Connector) provisioning

⑦ Reuse of JMX consoles

⑦ Apache Felix MOSGi

⑦ RFC 139

⑦ MBean standardization

JMX and OSGi

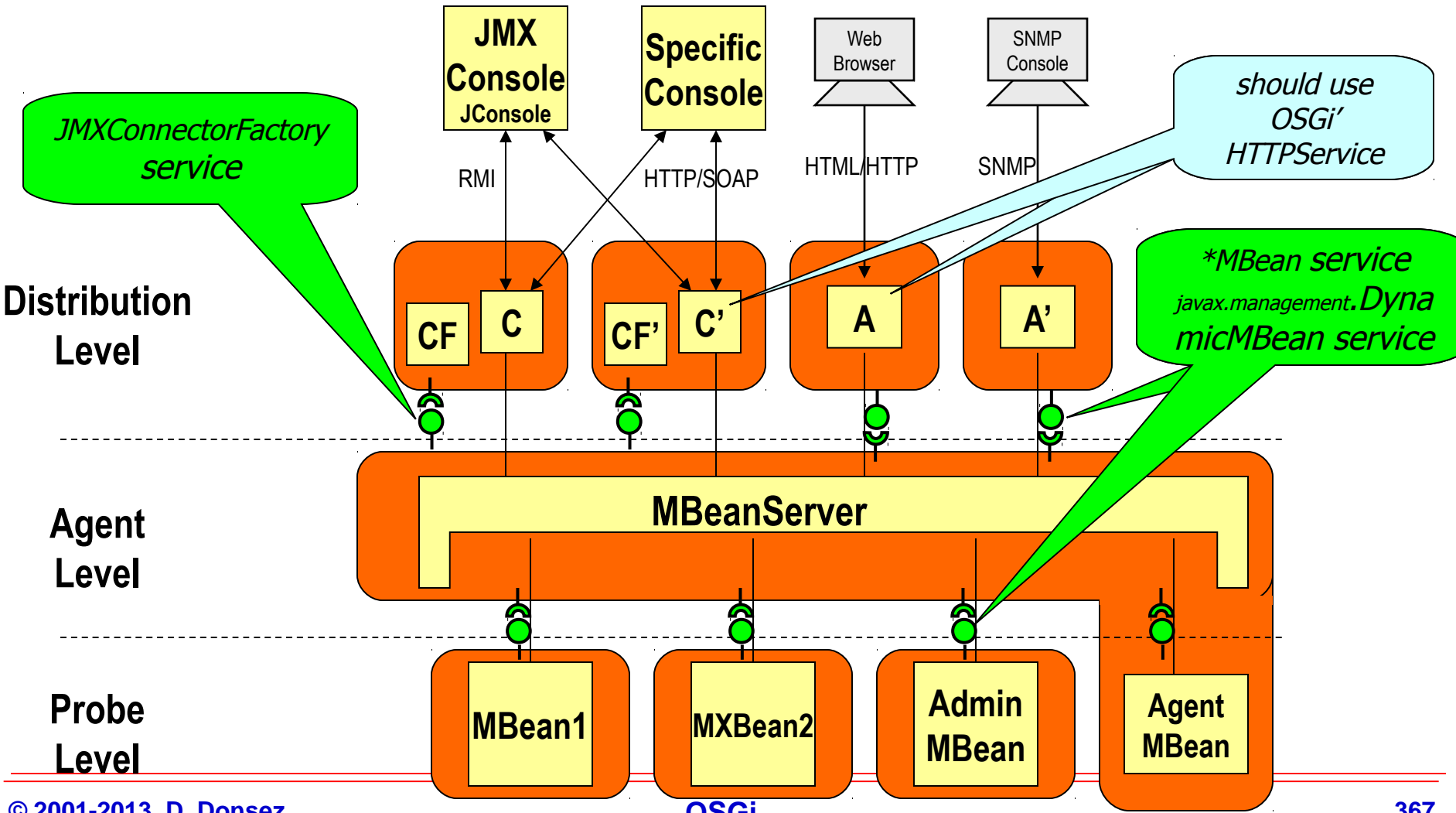
⑦ Principe

- ⑦ Le bundle JMXAgent démarre un MBeanServer
- ⑦ Chaque bundle enregistre un ou plusieurs services MBean
- ⑦ Le bundle JMXAgent récupère tout nouveau service MBean
 - ⑦ `f Iter="(objectClass=*MBean)"`
 - ⑦ Propriété `jmx.objectName`
- ⑦ Le bundle JMXAgent récupère tout service de connection ou d'adaptation

⑦ Voir MOSGi du projet Apache Felix

- ⑦ <http://felix.apache.org/site/mosgi-managed-osgi-framework.html>

JMX and OSGi MOSGi



Http Service Others

⑦ WAR convertor

⑦ Servlet extender

- ⑦ Servlets and ressources

- ⑦ See the Peter Kriens' Blog

 - ⑦ <http://www.aqute.biz/Snippets/Extender>

⑦ Servlet Bridge

- ⑦ Deploy a WAR in a Web container

- ⑦ But register servlets/ressources in the OSGi HttpService

- ⑦ Felix ServletBridge

- ⑦ Equinox ServletBridge

 - ⑦ http://www.eclipse.org/equinox/server/http_in_container.php

Http Service

Felix' Servlet Bridge

⑦ Motivation

- ⑦ use the Http service inside a WAR deployed on a 3rd part applicaton server.

⑦ Setup

- ⑦ Deploy `org.apache.felix.http.proxy` jar file inside the web applicaiton (`WEB-INF/lib`).
- ⑦ In a startup listener (like `ServletContextListener`) set the `BundleContext` as a servlet context attribute (see example).
- ⑦ Def ne `org.apache.felix.http.proxy.ProxyServlet` inside your `web.xml` and register it to serve on all requests `/*` (see example).
- ⑦ Be sure to add `javax.servlet;javax.servlet.http;version=2.5` to OSGi system packages (`(org.osgi.framework.system.packages)`).
- ⑦ Deploy `org.apache.felix.http.bridge` (or `org.apache.felix.http.bundle`) inside the OSGi framework.

Background

Embedding OSGi framework in a servlet container

⑦ Motivation

- ⑦ Run modular webapps in a servlet container
- ⑦ Register regular servlets in the `org.osgi.service.http.HttpService`
- ⑦ Deploy dynamically bundles in the servlet container
- ⑦ Share object between bundles and regular servlets
 - ⑦ JNDI (RFC 142), servlet context, static class member

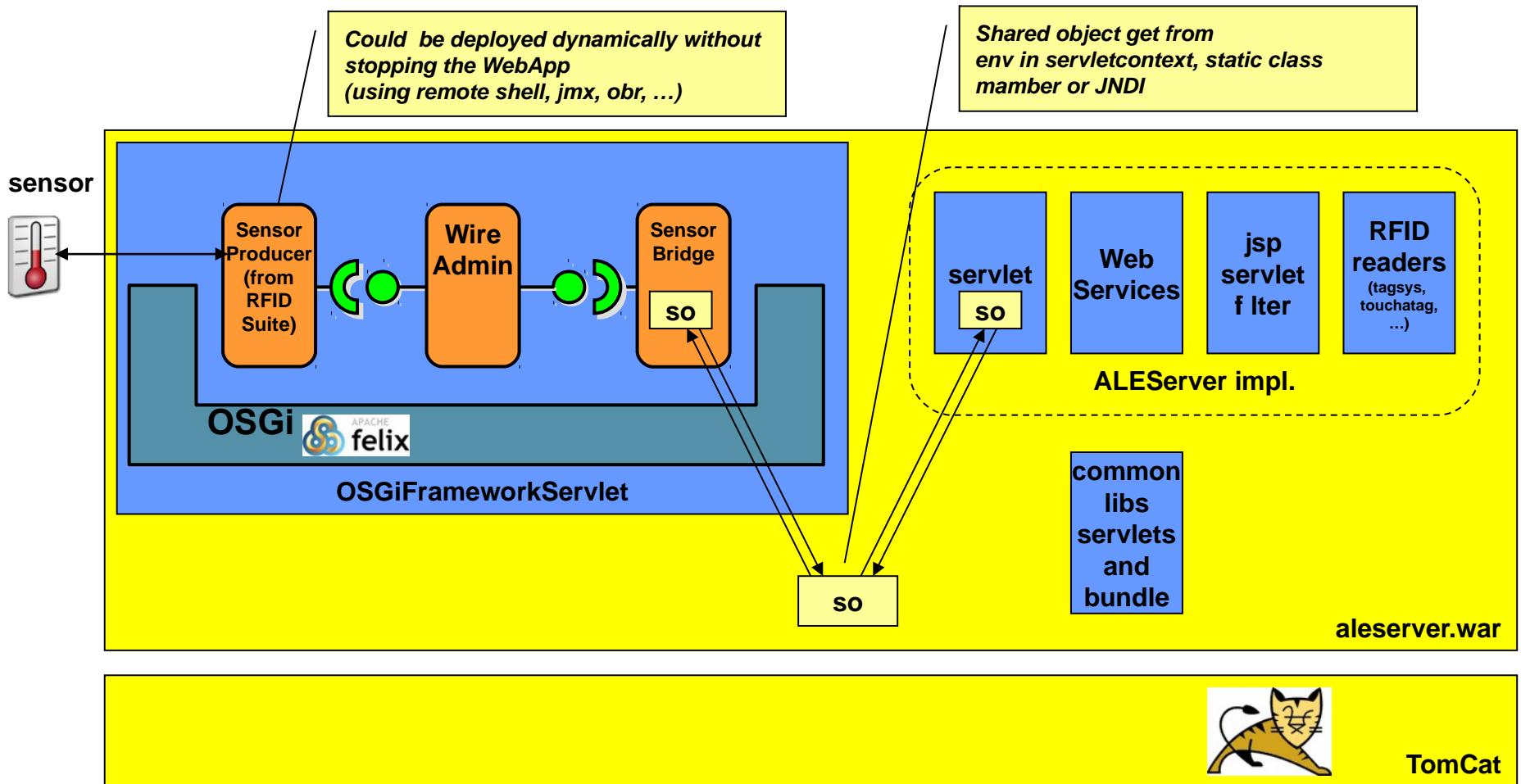
⑦ Equinox

- ⑦ BIRT Servlet
- ⑦ http://www.eclipse.org/equinox/server/http_in_container.php

⑦ Felix

- ⑦ Apache Felix `HttpServletBridge`
- ⑦ OW2 Dysoweb Servlet
 - ⑦ <http://svn.forge.objectweb.org/cgi-bin/viewcvs.cgi/dysoweb/trunk/dysoweb/webapp/>

Embedding OSGi framework in a servlet container for Aspire



OSGi 4.3

(voir présentation OUGF 24 11 2012)

⑦ Generics 1.5

- ⑦ Evolution of the API
- ⑦ `ServiceReference <I>`, `ServiceRegistration<I>`
- ⑦ `ServiceTracker<I,I>`
- ⑦ `BundleTracker<T>` : extender pattern

⑦ Capabilities & Requirements

- ⑦ Old concept but into the API now
- ⑦ Capability : set of name-typed value pairs (some are directives)
- ⑦ Requirement : LDAP filter

OSGi 4.3 Core Services

⑦ **Deprecated**

⑦ **Package Admin Service** → replace by **Bundle Wiring API**

⑦ **Problem with refresh (global)**

⑦ **Start Level Service** → replace by **Start Level API**

⑦ **Hooks**

⑦ **introspection, proactive isolation (instead of OSGi Permission), ...**

⑦ **Service Hook (4.2)**

⑦ **Weaver Hook**

⑦ **Bytecode manipulation**

⑦ **AOP, EE JPA, ...**

⑦ **Static Dynamic-ImportPackage**

⑦ **Resolver Hook**

⑦ **Bundle Hook**