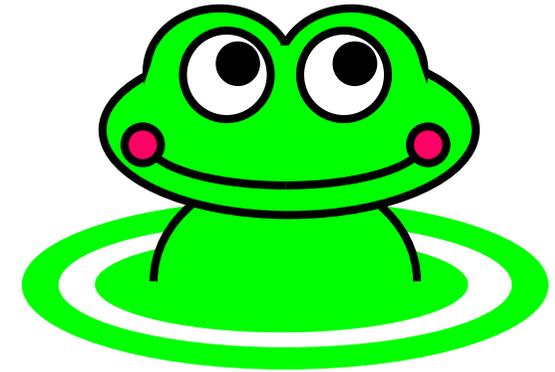




# FROGi

## *Déploiement de Composants Fractal sur OSGi*



**Mikaël Desertot**  
**Didier Donsez**  
**Humberto Cervantes\***

*Université Joseph Fourier (Grenoble 1), IMAG/LSR/ADELE*

*\* Universidad Autónoma Metropolitana-Iztapalapa, Mexico*





# Sommaire

- Motivations
- Rappels : Fractal, OSGi et SOA
- Avantages
- Fractal over OSGi
- Conclusion et perspectives

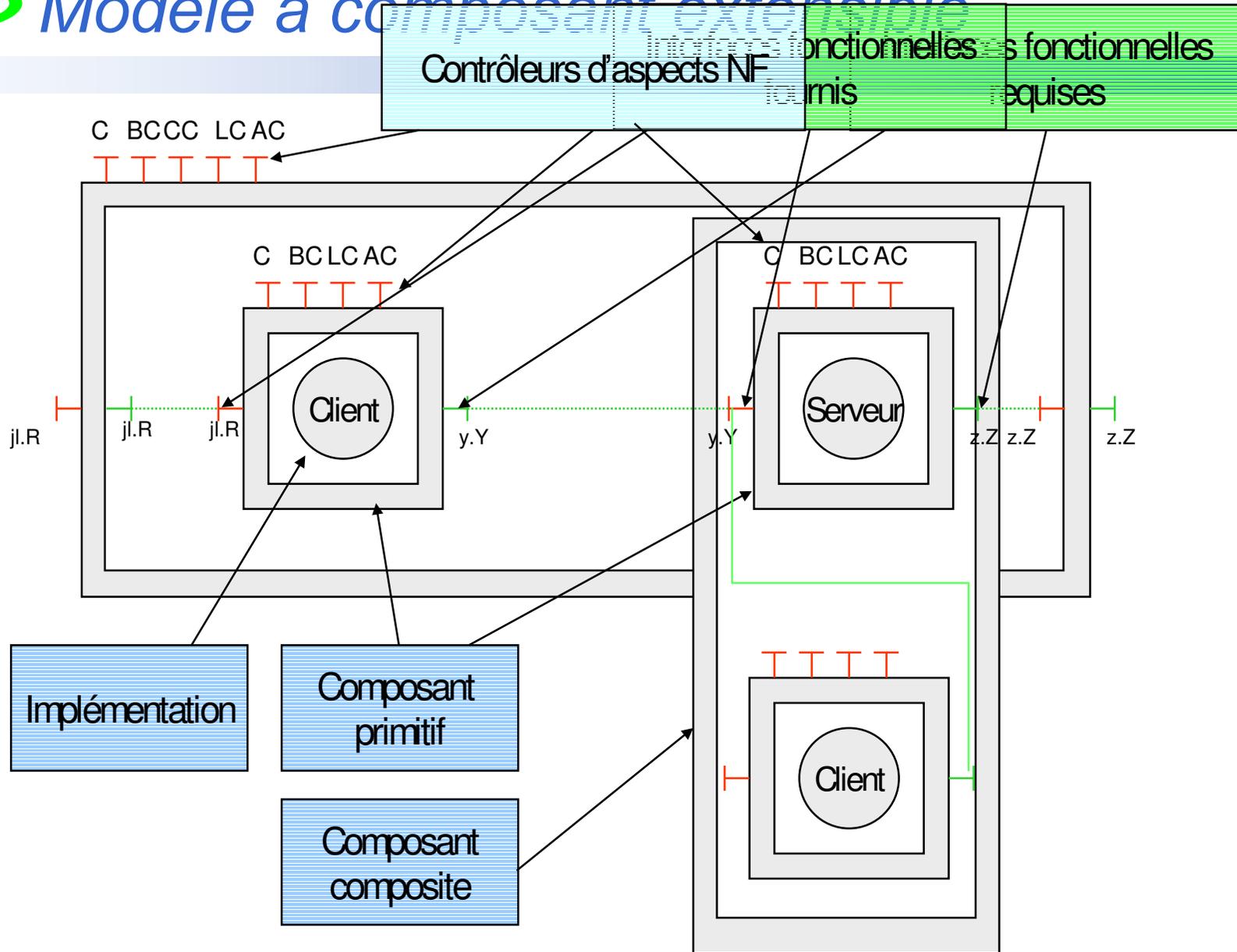


# Motivations

1. Modèle à composants simple dans OSGi
  - Fractal est un bon candidat
    - Outillages matures
    - Nombreux travaux autour de Fractal (ObjectWeb)
2. Packaging et déploiement en émergence dans Fractal/Julia
  - OSGi est un bon candidat
    - Mature
    - SOA dynamique



# Fractal (i) Modèle à composant extensible





# Fractal (ii)

## *Outils*

- Julia : implémentation de référence en Java
- THINK : C pour des ordinateurs nu (sans OS)
- FractalADL : ADL
- Nombreux travaux et nombreux "composants"
  - PERSEUS, JOTM, ...
  
- Mais Packaging et déploiement en émergence
  - [Korna et al 2004]

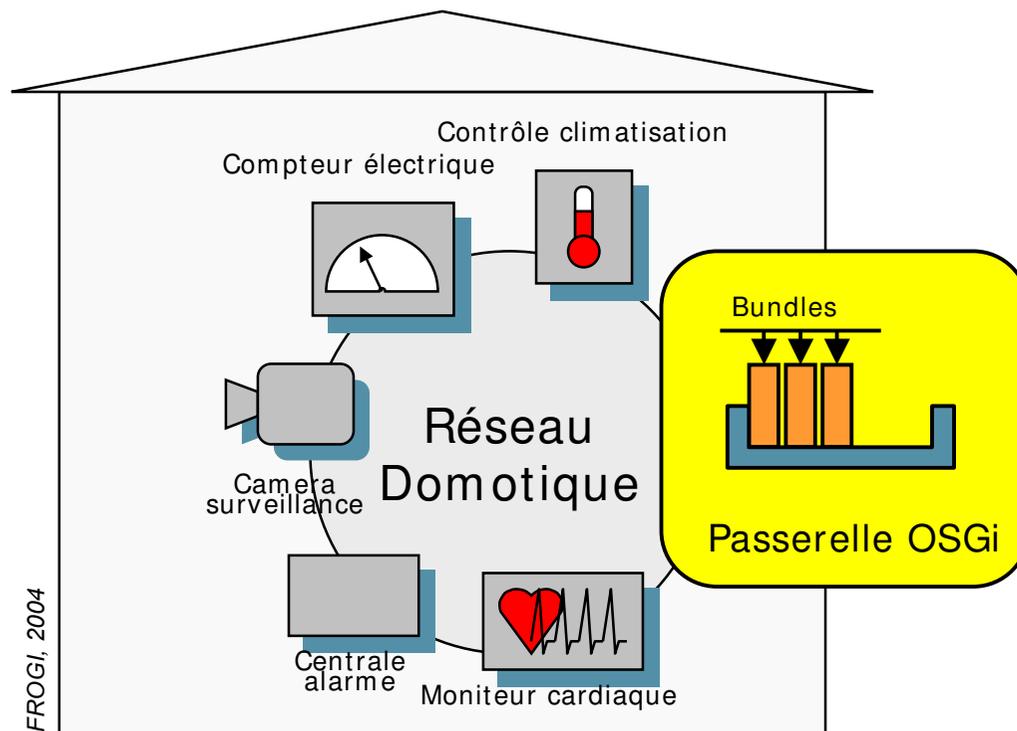


# OSGi (1)

- Consortium industriel
- Plateforme de déploiement de services logiciels
  - Passerelle *Non-Stop* (i.e. sans interruption de service)

## ■ Domaines:

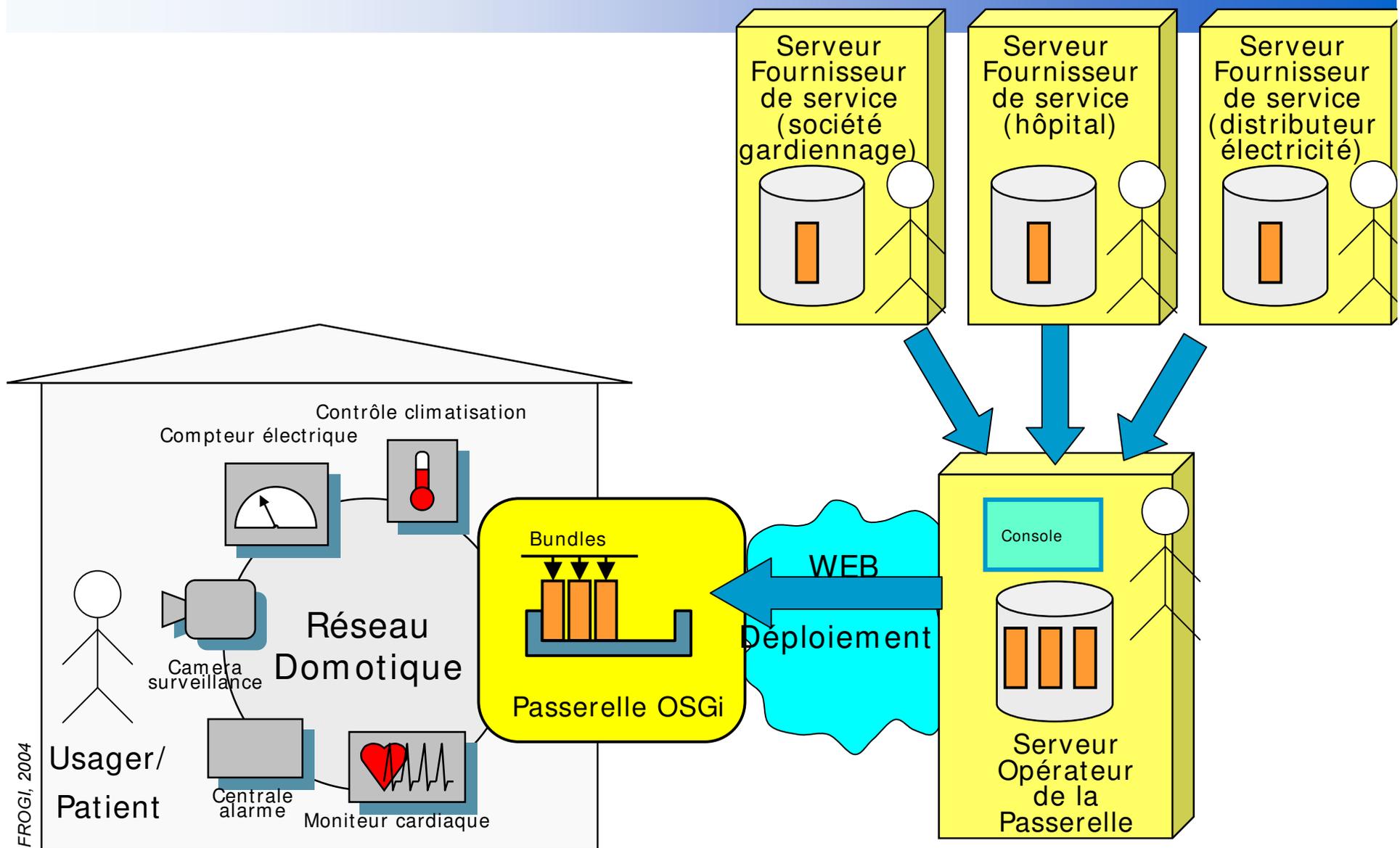
- Domotique/Immotique
- Véhicule
- Industrie
- ...
- Applications à plugin
  - Eclipse 3.0, ...





# OSGi (2)

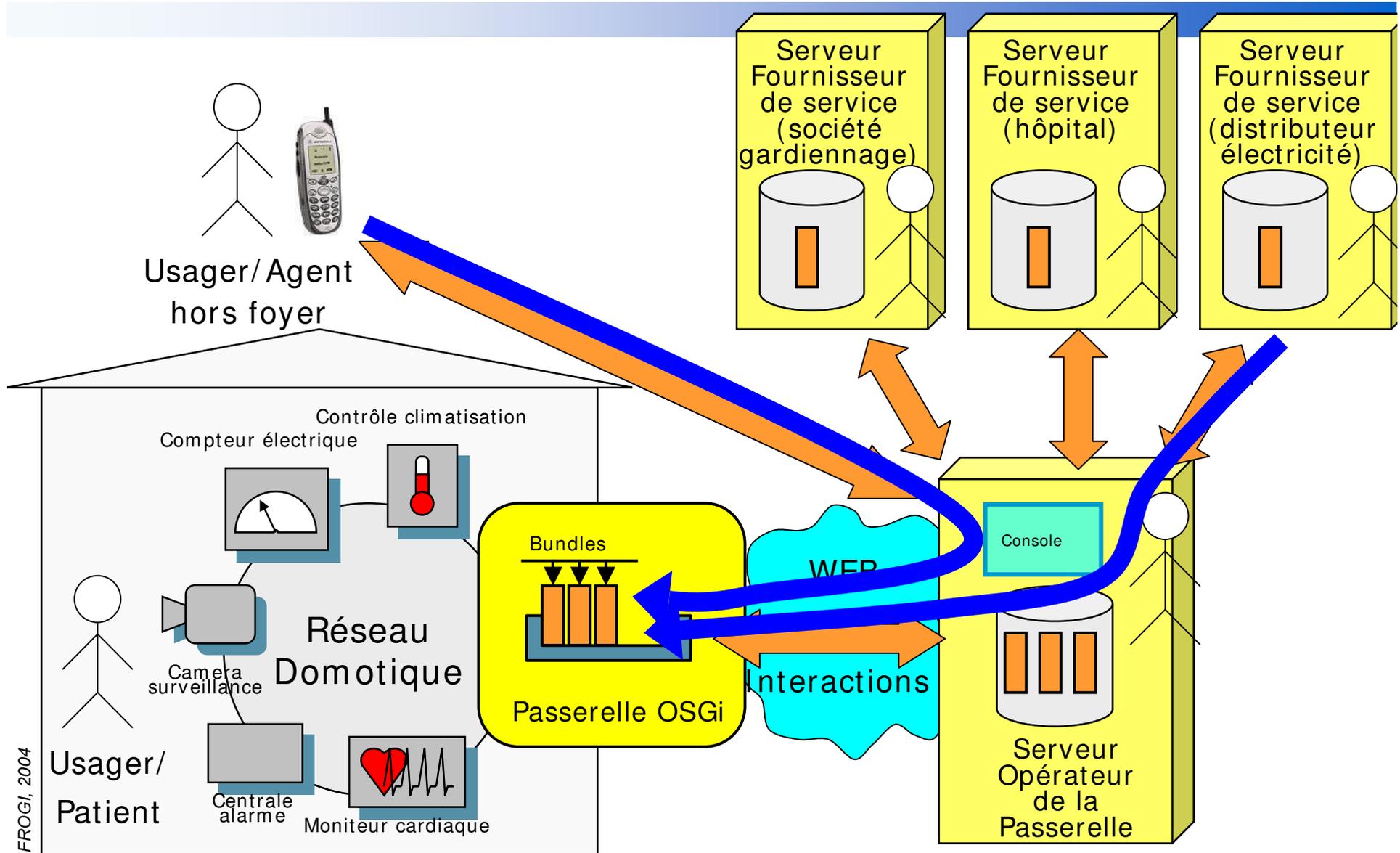
## Déploiement des services de fournisseurs





# OSGi (2)

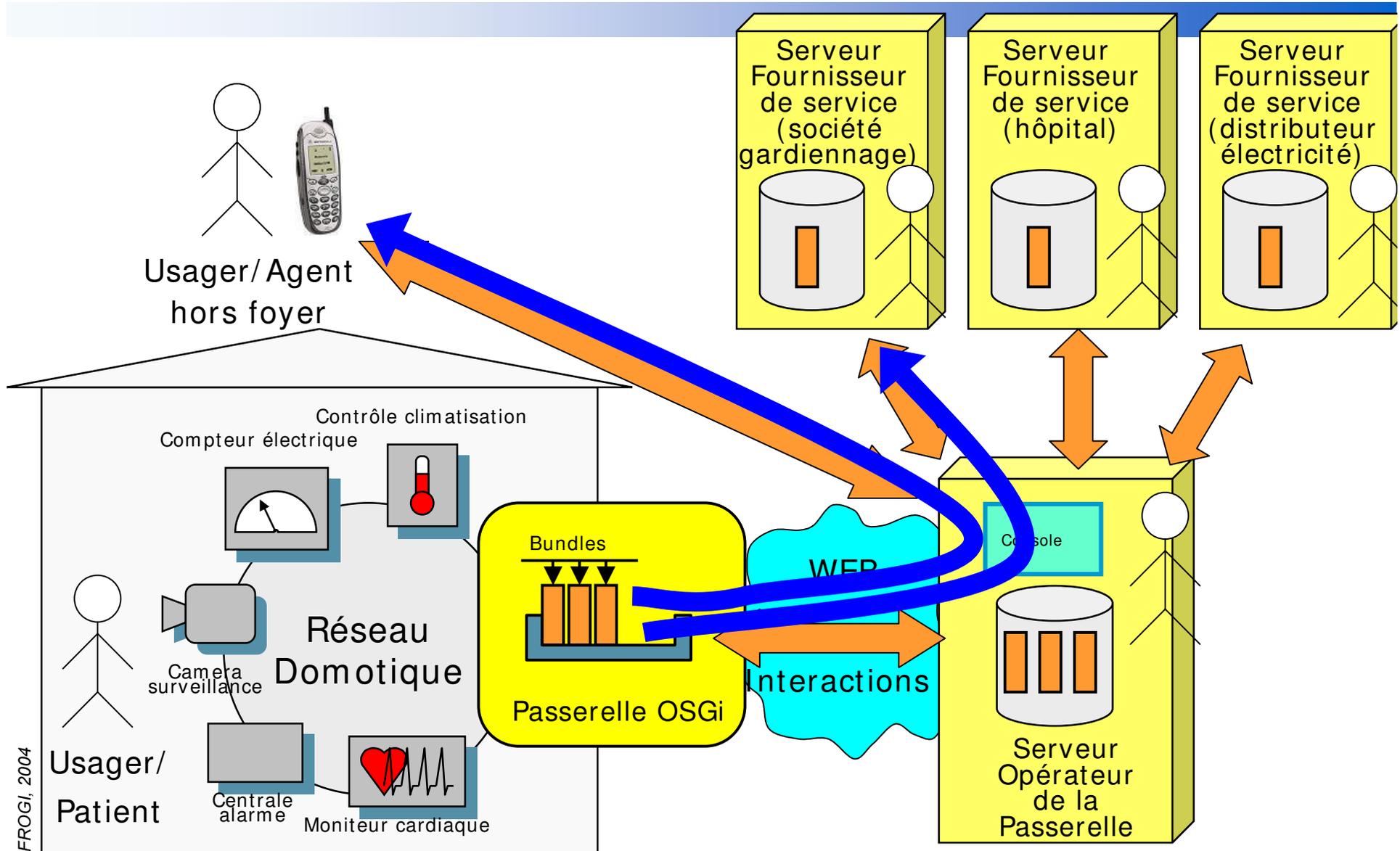
## Interactions services - applications





# OSGi (2)

## Interactions services - applications



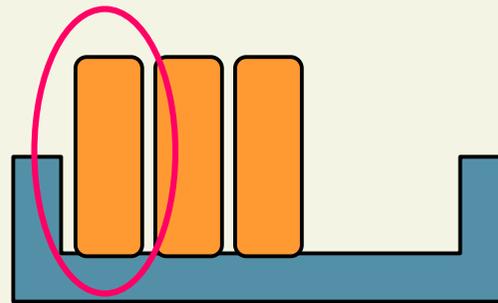


# OSGi (3)

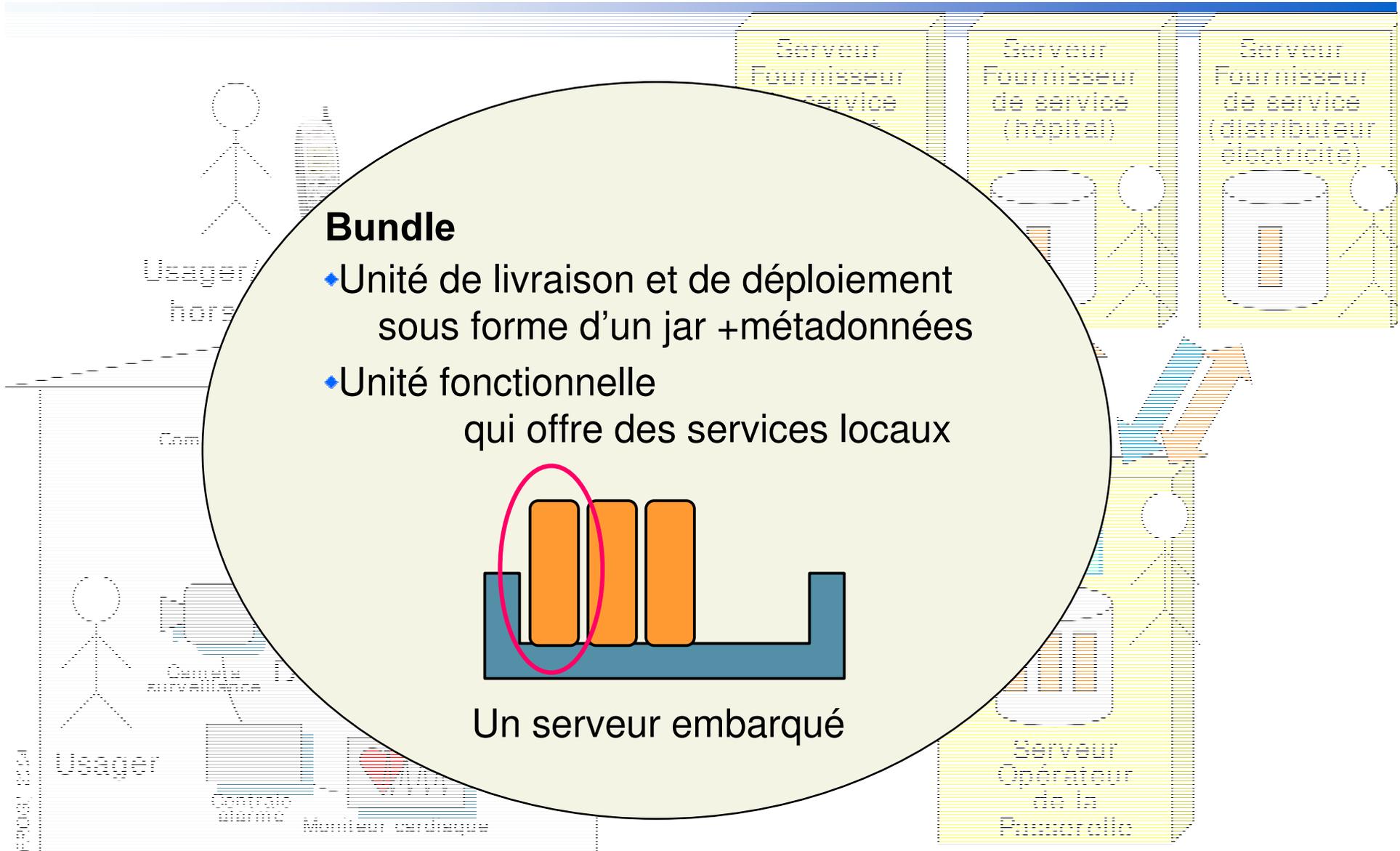
## Bundle et Service

### Bundle

- ◆ Unité de livraison et de déploiement sous forme d'un jar + métadonnées
- ◆ Unité fonctionnelle qui offre des services locaux



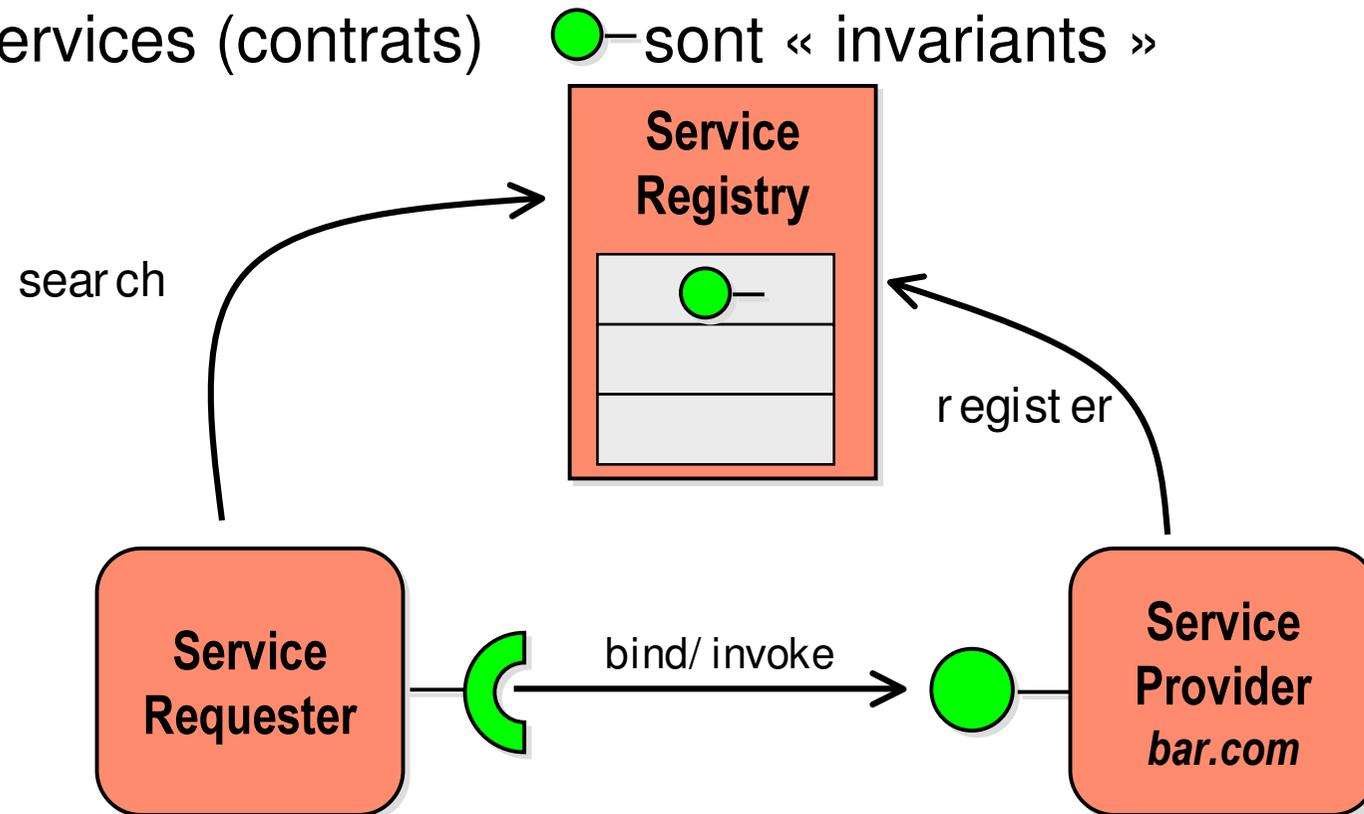
Un serveur embarqué





# SOA (*Service Oriented Architecture*)

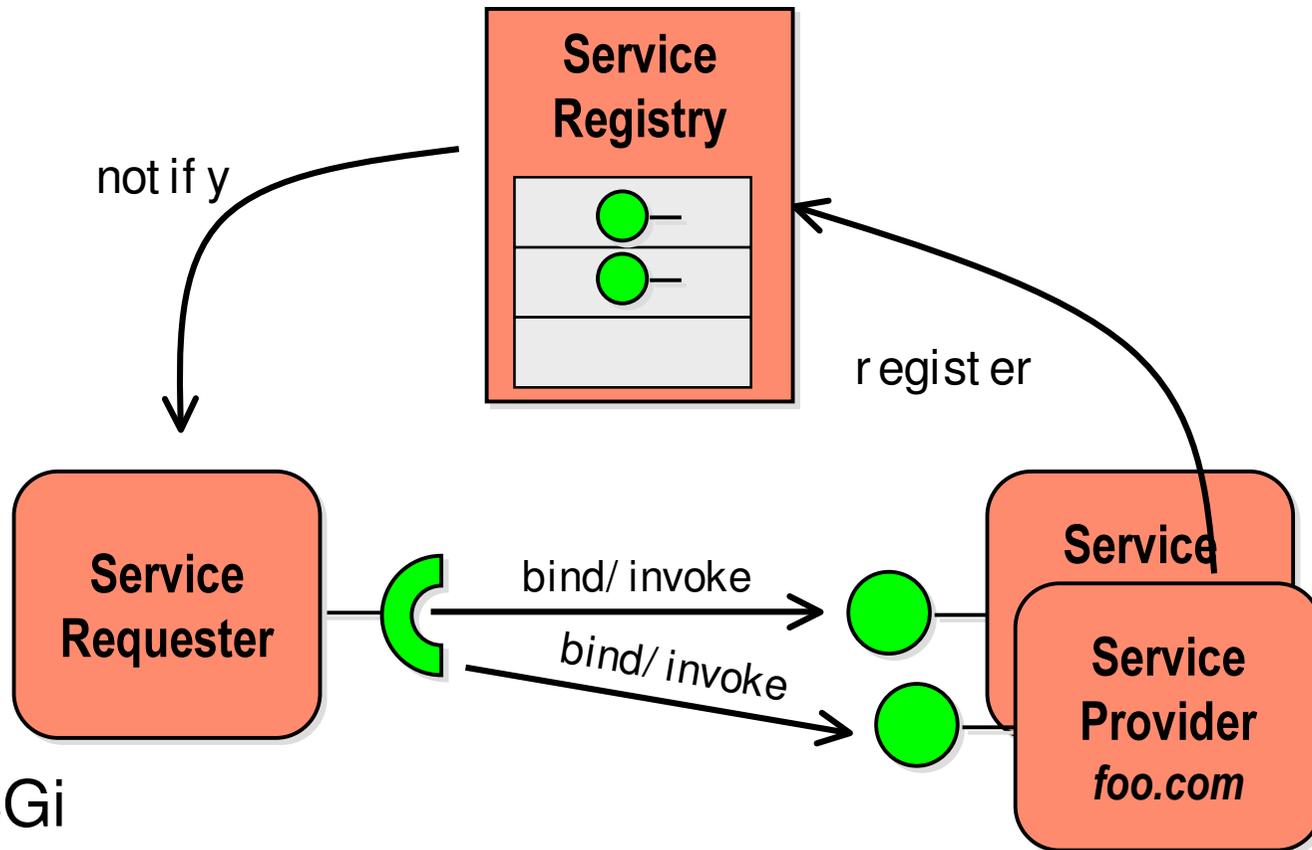
- Les services (contrats) ● sont « invariants »





# SOA Dynamique

## ■ Des services apparaissent ...

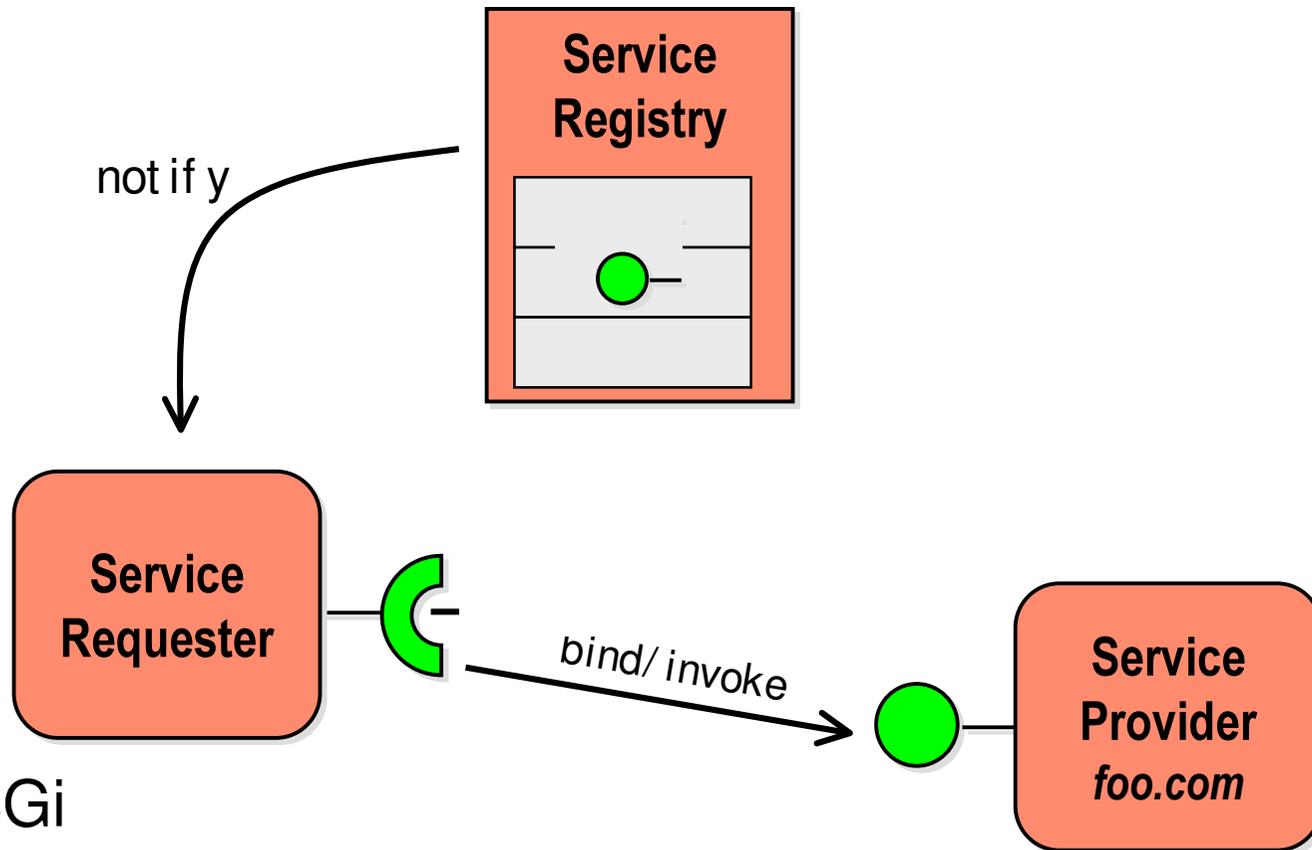


## ■ JINI, OSGi



# SOA Dynamique

- Des services disparaissent ...



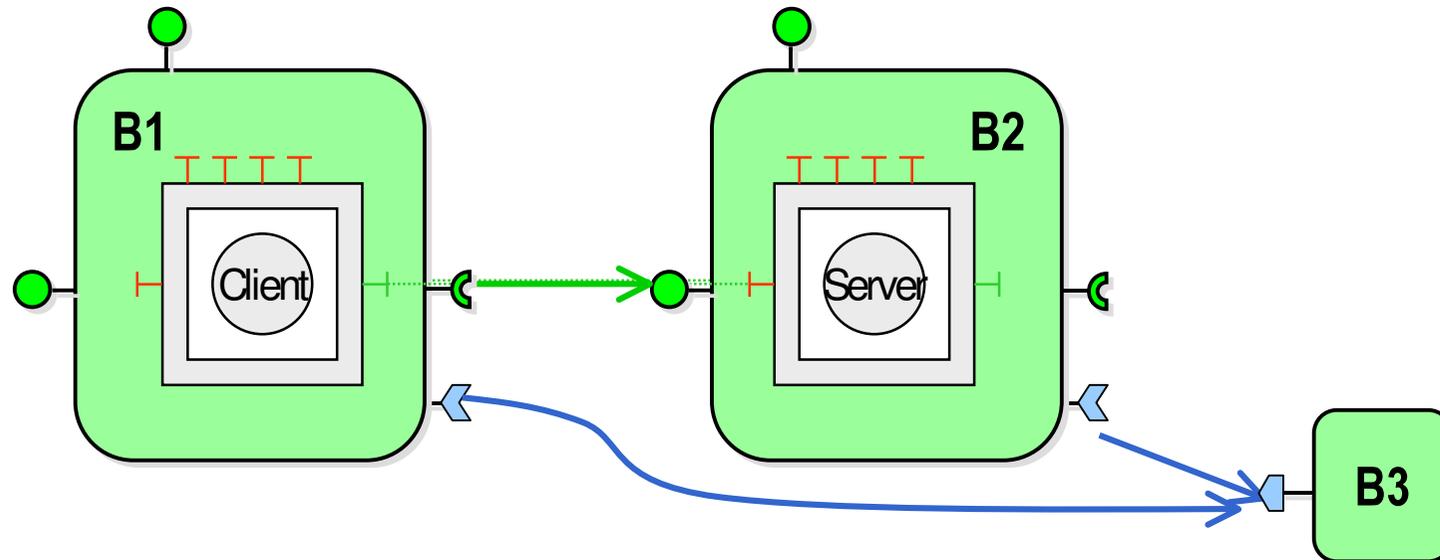
- JINI, OSGi



# Notre proposition

## FROGi : Fractal over OSGi

- Packager et Déployer une application Fractal en plusieurs bundles OSGi
  - Séparation Contrat-Implémentation
  - Mise à jour partielle Dynamique
  - Composants partagés





# Avantages

## ■ Développeur OSGi

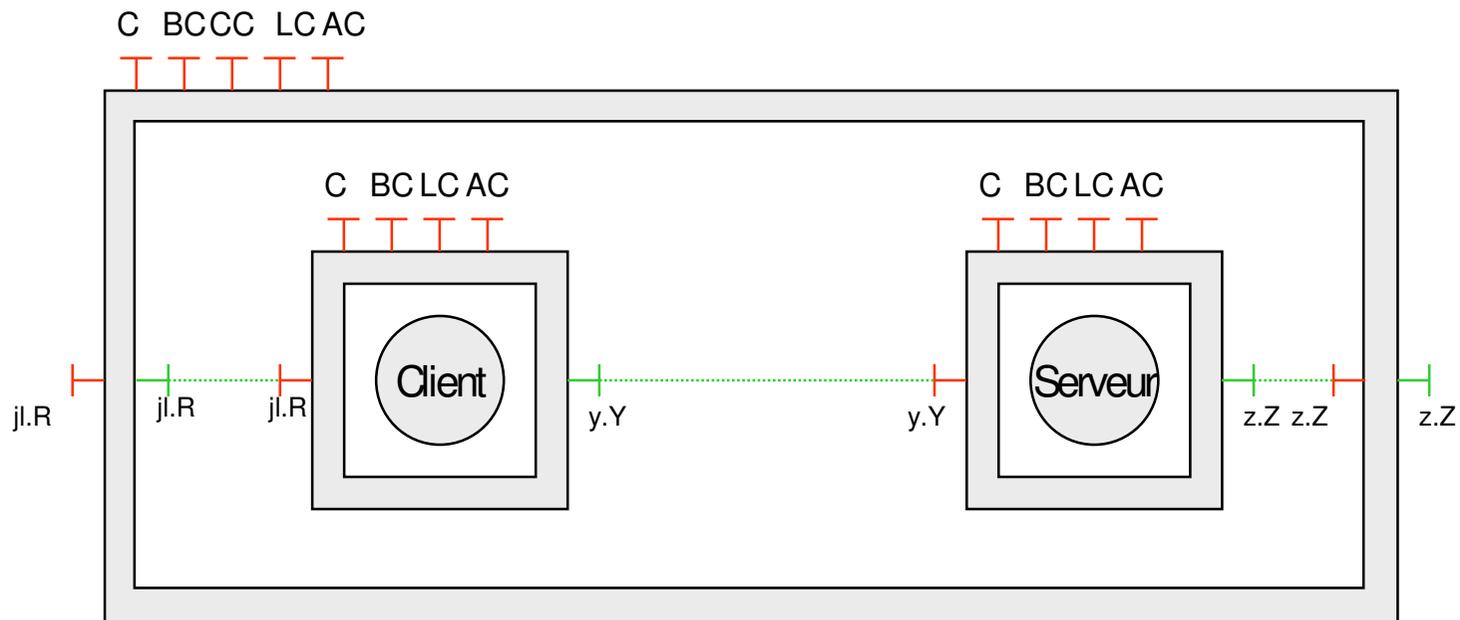
- Programmation orientée composants des services OSGi
  - services standards ou services propriétaires

## ■ Développeur Fractal

- Outils de déploiement OSGi
- Versionning des Contrats et des Implémentations
- Mise à jour partielle
- SOA Dynamique
- Partage de la VM par plusieurs applications simultanées sans interruption de service
- Réutiliser les bundles OSGi standard et patrimoniaux



# Notre application Fractal





# Son Fractal ADL étendu par FROGi

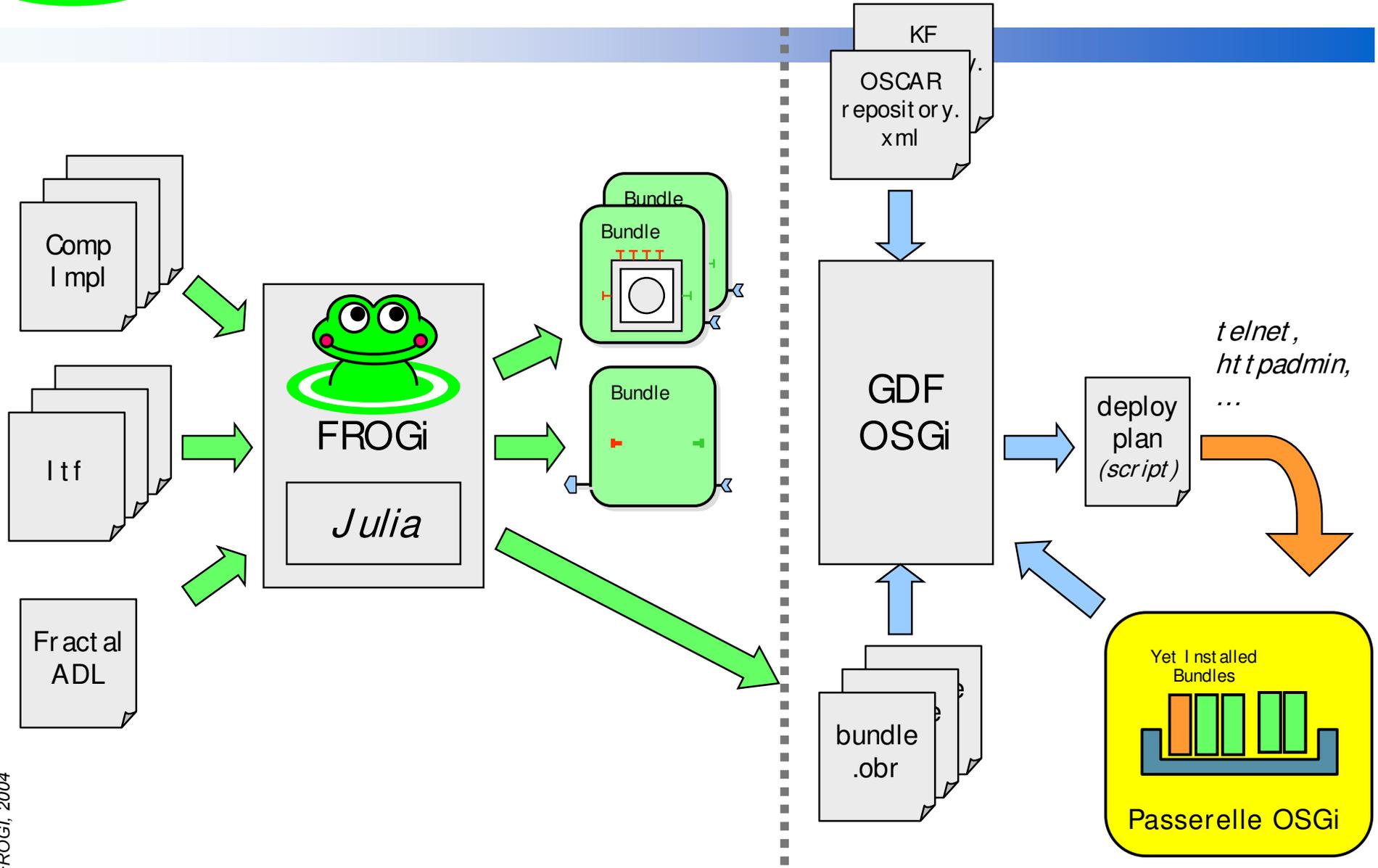
```

<definition name="HelloWorld">
  <bundle name="B0"/>version="0.9.0" />
  <interface name="main" role="server" signature="java.lang.Runnable">
    <property name="cron.pattern" value="* * 3 * * * *" type="java.lang.String"/>
  </interface>
  <component name="client">
    <bundle name="B1"/>version="1.0.0" />
    <interface name="x1" role="server" signature="java.lang.Runnable"/>
    <interface name="cy2" role="client" signature="y.Y" version="1.0.0" bundle="B3"/>
    <content class="ClientImpl"/>
  </component>
  <component name="server">
    <bundle name="B2"/>version="2.0.0" />
    <interface name="y2" role="server" signature="y.Y" version="1.0.0" bundle="B3"/>
    <interface name="cz3" role="client" signature="z.Z" version="2.0.0" bundle="B3"
      cardinality="collection" contingency="optional" policy="dynamic"/>
    <content class="ServerImpl"/>
  </component>
  <binding client="this.x1" server="client.x1"/>
  <binding client="client.cy2" server="server.y2"/>
  <binding client="server.z3" server="this.cz3"/>
  <binding client="this.z3" serverfilter="(&(objectClass=z.Z)(language=fr))"/>
</definition>

```



# La chaîne de génération





# 2 politiques de gestion du cycle de vie et des liaisons

## ■ Politique 1 : *Pilotage par le composite*

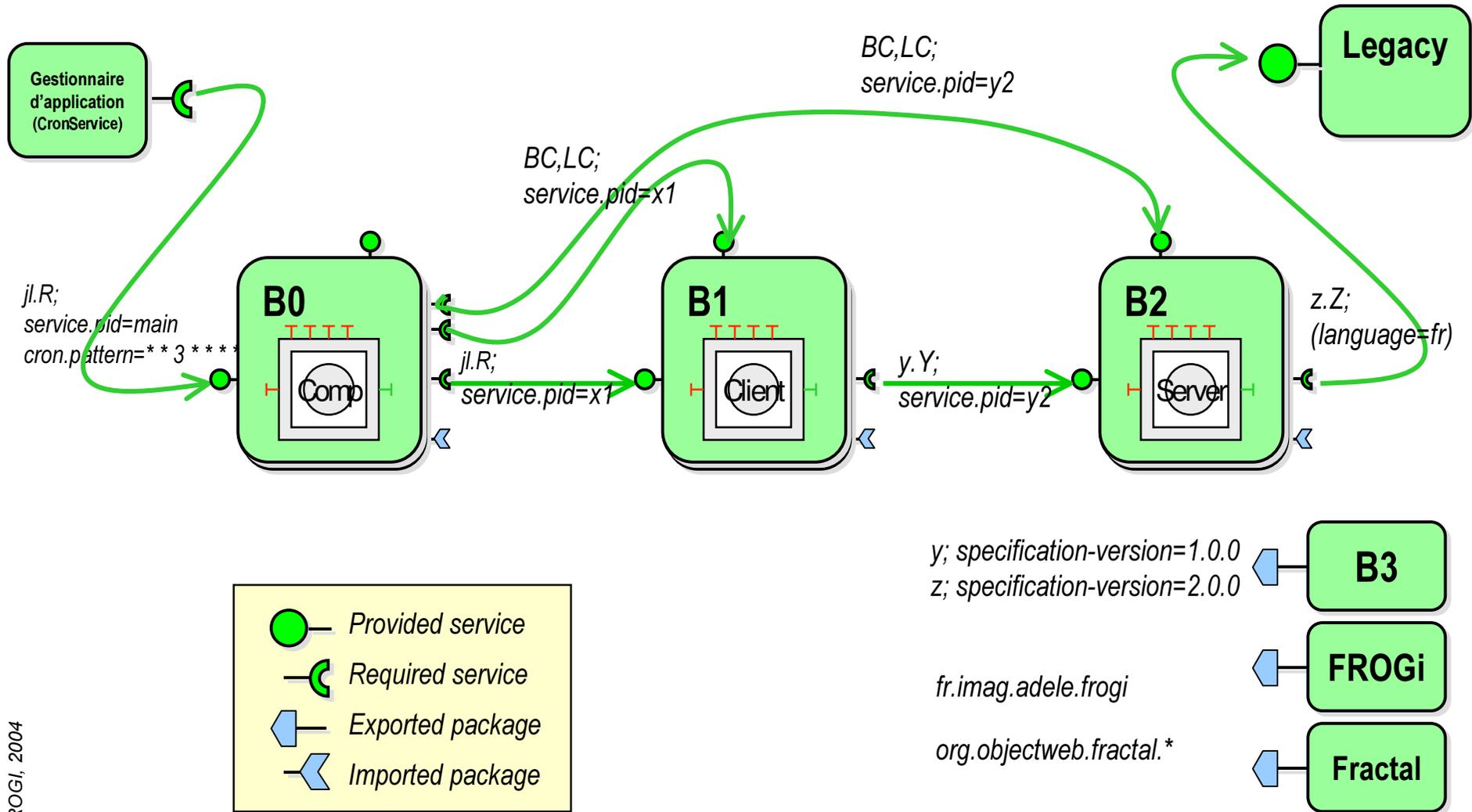
- liaison et cycle de vie dirigé par le composite

## ■ Politique 2 : *Pilotage autonome*

- liaison automatique des composants
- Cycle de vie dépendant de la contingence des services requis
  - Similaire à ServiceBinder

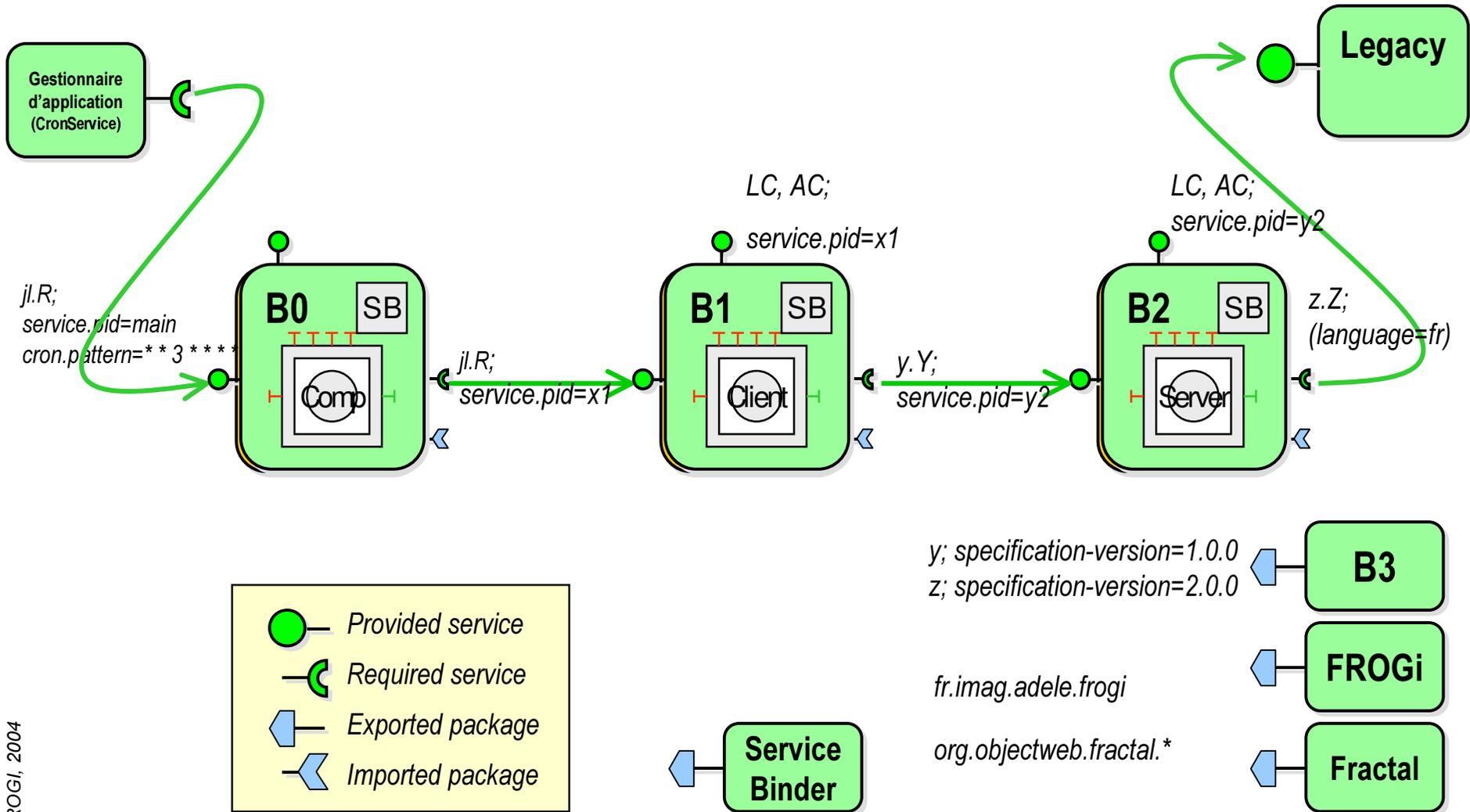


# Politique 1: Pilotage par le composite





# Politique 2: *Pilotage autonome*





# Limitations

- Une seule version (spécification) d'un package java actif à la fois (lié OSGi 3)
  - Vraisemblablement levée avec OSGi 4
- Grain de mise à jour : bundle
  - Au choix/expertise du « packager »
- Fabrication des instances figées par l'ADL
  - Quelques pistes :
    - OSGi ServiceFactory
    - ServiceBinder Factory



# Conclusion & Perspectives

## ■ Conclusion

- Disposer d'un modèle à composant (Fractal) au dessus d'OSGi
- Tirer parti des capacités de déploiement d'OSGi pour déployer des composants Fractal
- Bénéficier des possibilités du SOA d'OSGi pour Fractal

## ■ Perspectives

- Généralisation de l'approche à d'autres modèles
  - AVALON over OSGi
- Interopérabilité multi-modèle via le SOA Dynamique
  - Fractal  $\leftrightarrow$  AVALON  $\leftrightarrow$  EJB

■ Visitez <http://www-adele.imag.fr/frogi>

