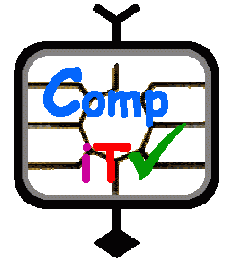




LSR

# OSGiTV

une plate-forme de déploiement  
d'applications de télévision interactive  
basée sur OSGi



Stéphane CHOMAT, Didier DONSEZ

Laboratoire LSR / Fédération IMAG

Equipe ADELE

Stephane.Chomat@imag.fr, Didier.Donsez@imag.fr

# Sommaire

- Contexte : la télévision interactive
- Motivations : applications dynamiques
- Points de Départ
- Service de Déploiement à la Demande
- Architecture proposée du middleware terminal
- Réalisation et Expérimentations
- Conclusion et Perspectives
- Bonus

# Le contexte: la télévision interactive

## ■ Télévision Numérique

- Programmes Vidéo-Audio
- Contrôle d'accès (Abonnement, Pay-per-view, ...)
- Vidéo à la demande : Near VOD, Prefetch VOD, ...

## ■ Accès à des services proposés par l'opérateur TV

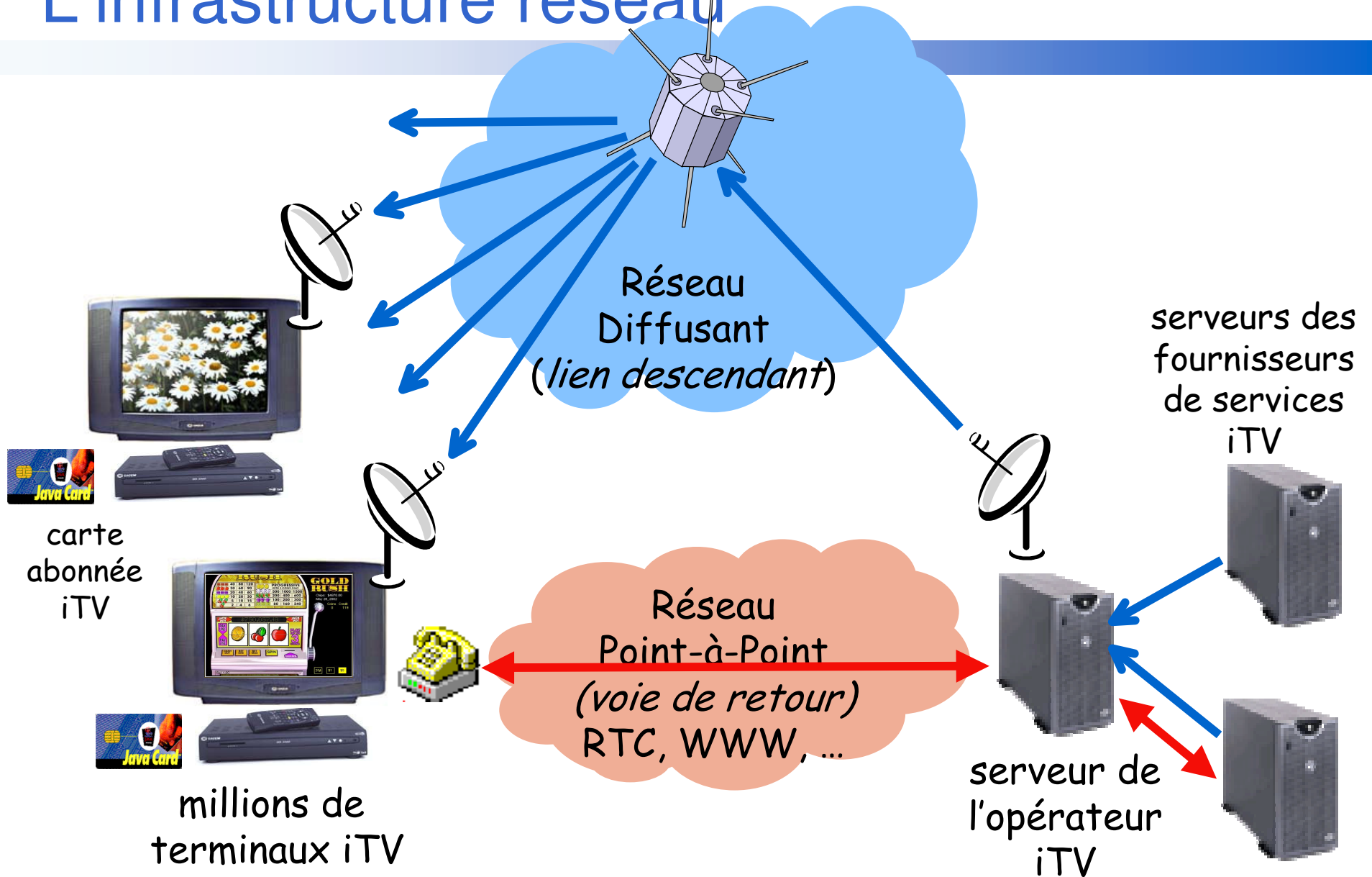
- Avec ou sans liaison avec un programme diffusé
- Exemples:
  - Mail, banque, boutique de shopping (T-Commerce), paris (PMU) , ...

## ■ Remarques

- Commerçants partenaires / fournisseurs de l'opérateur
- **Le parc de terminaux devient de plus en plus hétérogène**

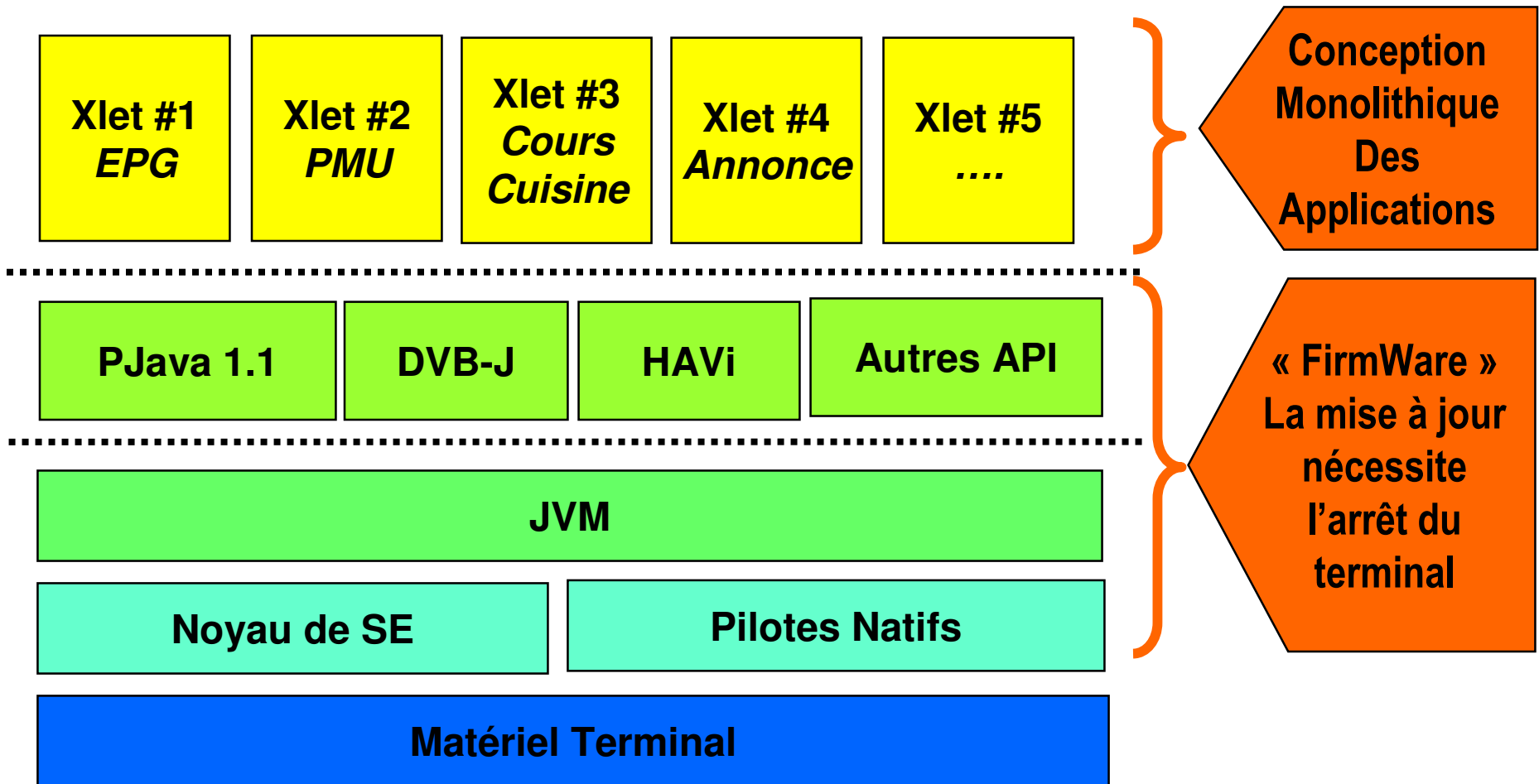
# Le contexte: la télévision interactive

## L'infrastructure réseau



# Le contexte: la télévision interactive

## Architecture DVB-MHP / DVB-J



# Motivations

- Applications modulaires et dynamiques
  - Développement à base de composants dynamiques
    - applicatifs, environnement d'exécution, pilotes périphériques, ...
- Déploiement sur des millions de terminaux
  - Chargement : Fichiers de Code diffusés
  - Installation
    - « Poussée » par l'opérateur (EPG)
    - Sur action de l'utilisateur (Zapping, Insertion de la carte)

# Motivations : Application dynamique

## Exemple

Choix d'un jeu

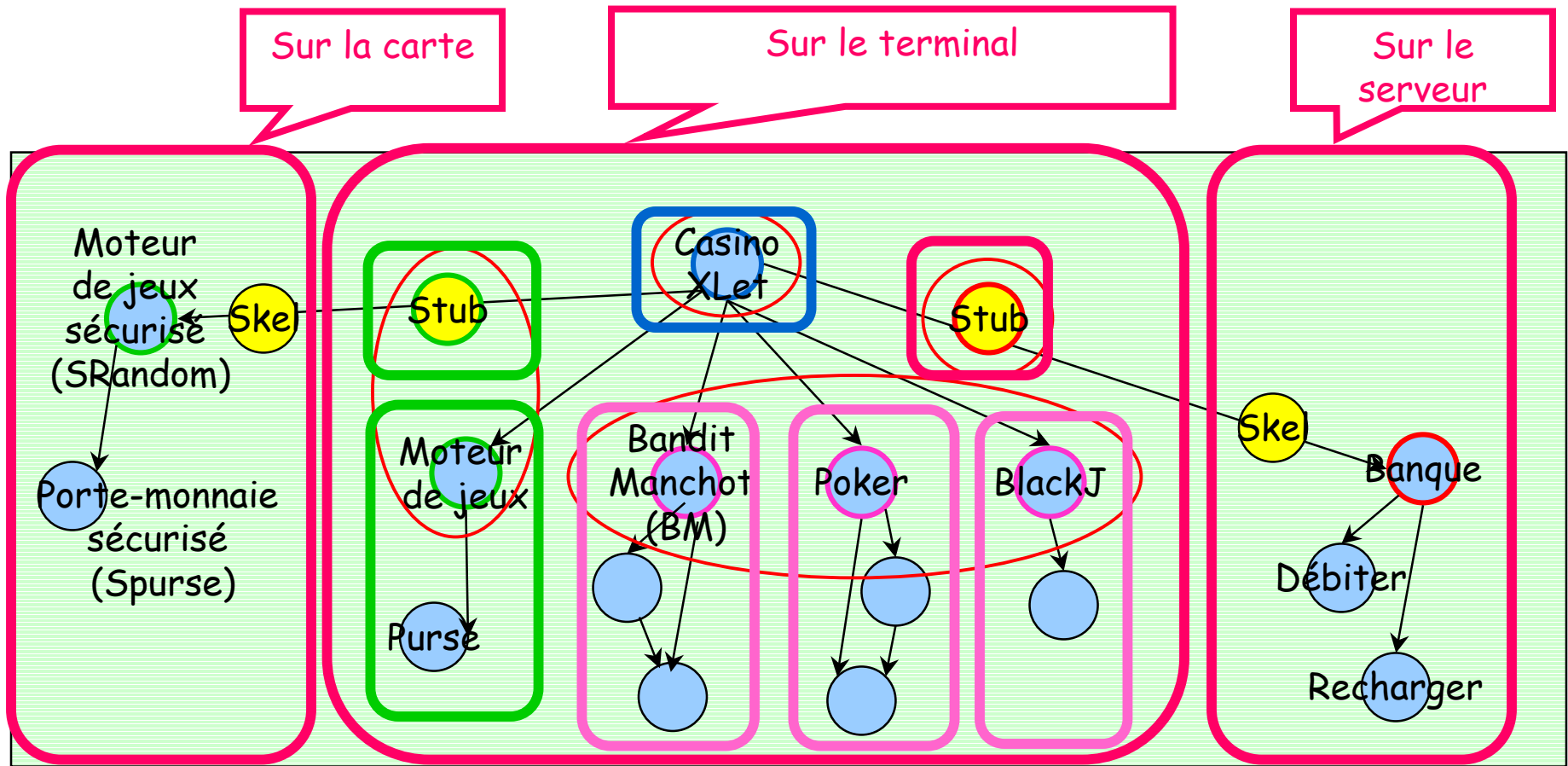
Insertion de la  
carte

Parier et Gagner



# Motivations : Application dynamique

## Conception et distribution de l'application



Services d'application  
de recharge



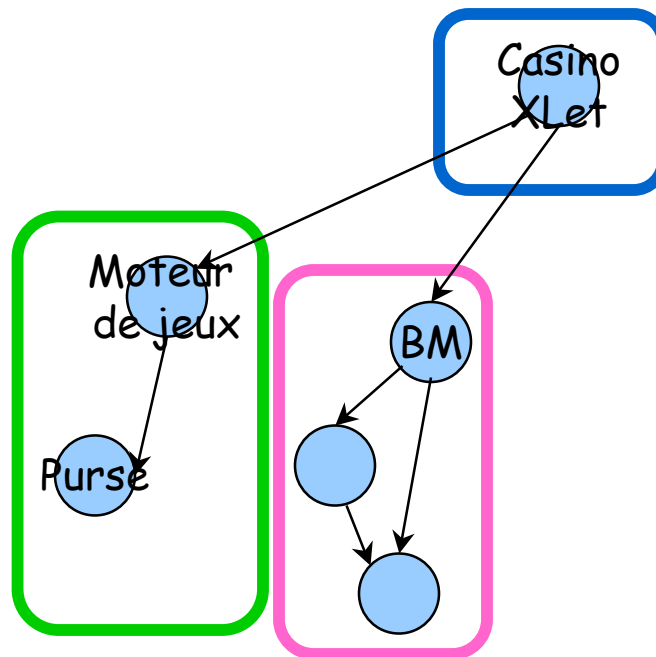


# Motivations : Application dynamique

## Activation de l'application

### ■ Le minimum à activer

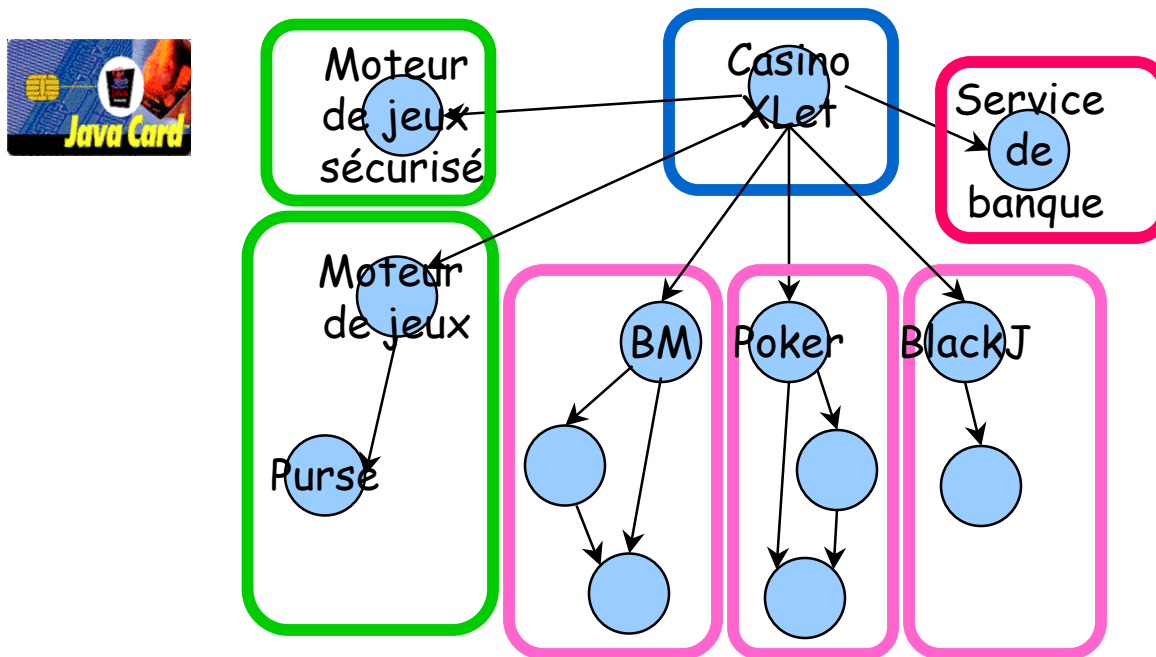
- 1 moteur
- 1 jeu



# Motivations : Application dynamique

## Exécution de l'application

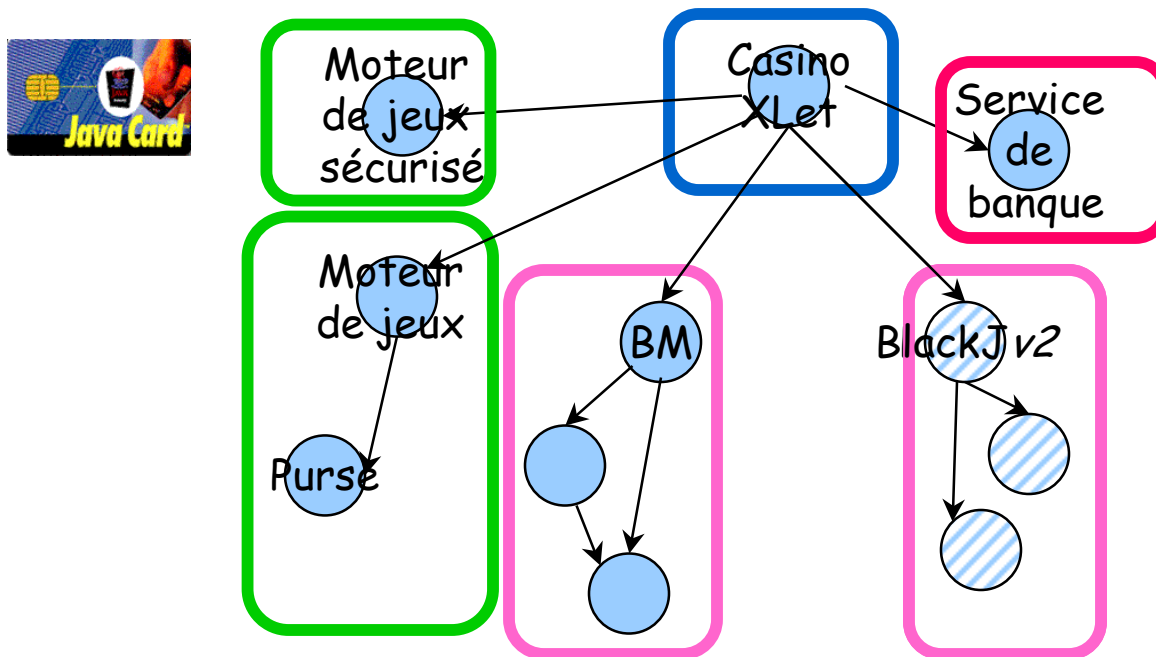
- Apparition de services pendant l'exécution
  - Nouveaux services de jeux
  - Nouveau moteur de jeux à l'insertion de la carte



# Motivations : Application dynamique

## Mise à jour d'un jeu

- Certains services peuvent être remis à jour
  - Adaptation dynamique



# Points de départ

## ■ CESURE

- Installation et connexion des composants (CCM) d'une application pilotées par la carte

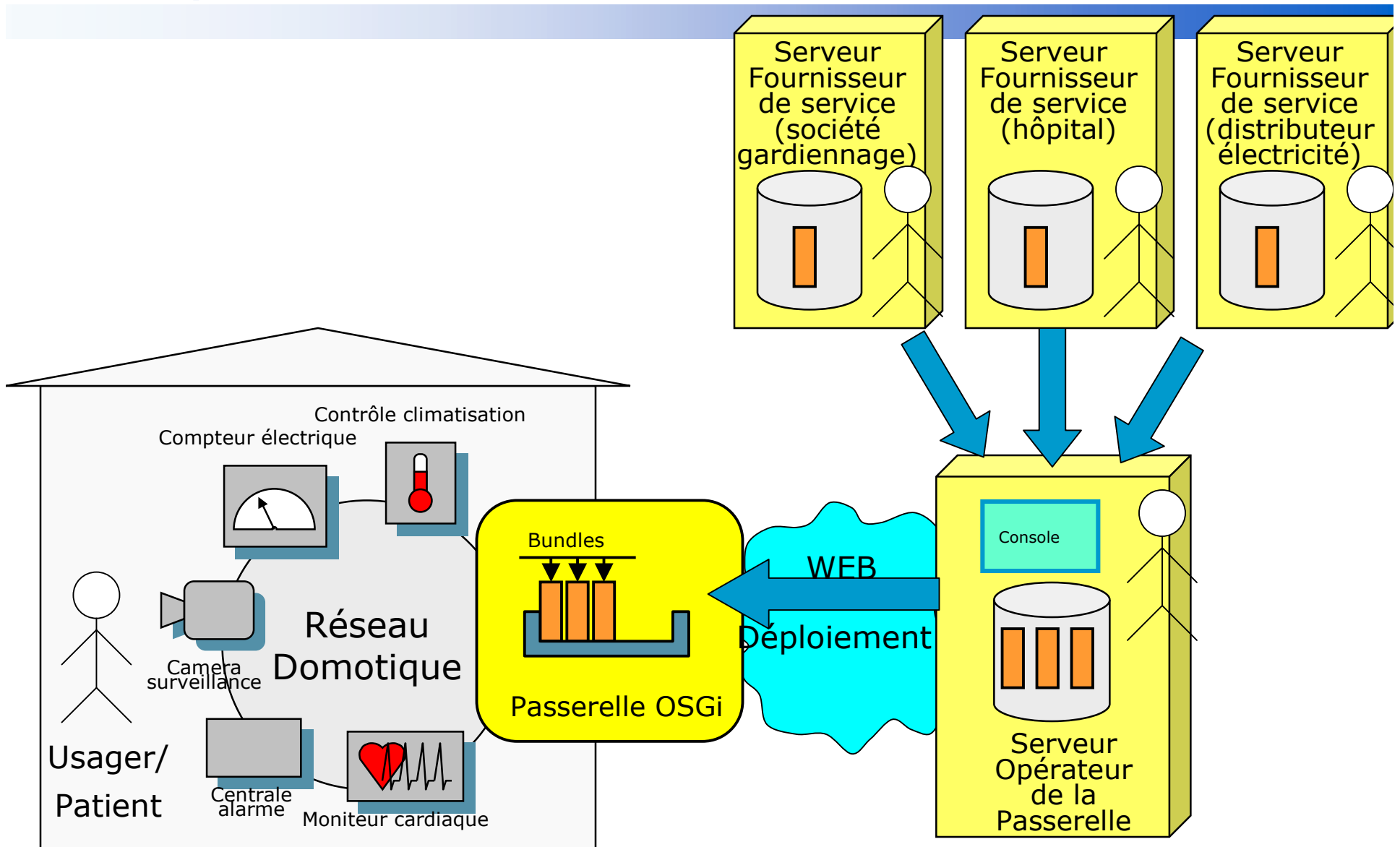
## ■ OSGi

- Déploiement dynamique

## ■ Beanome/ServiceBinder

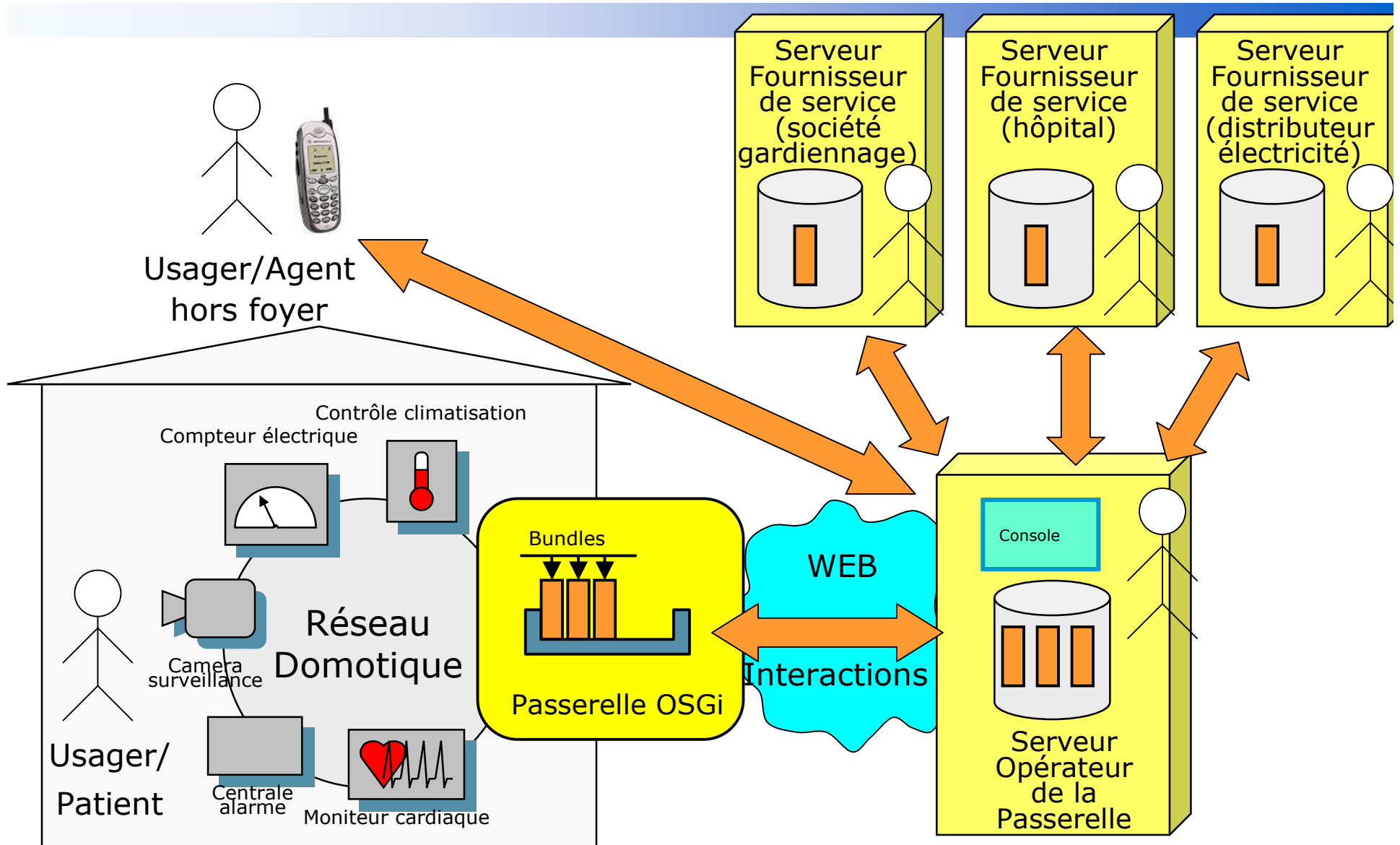
- Modèle de composants orienté service au dessus d'OSGi
- ADL dynamique (liaison et cycle de vie)

# OSGi - Architecture générale (i) Déploiement



# OSGi - Architecture générale (ii)

## Interactions

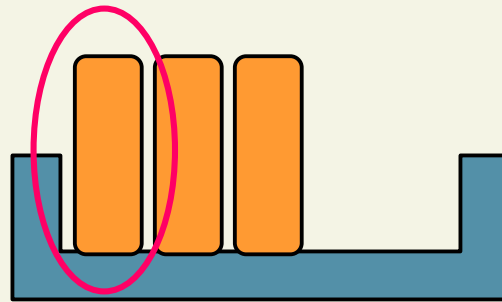


# OSGi

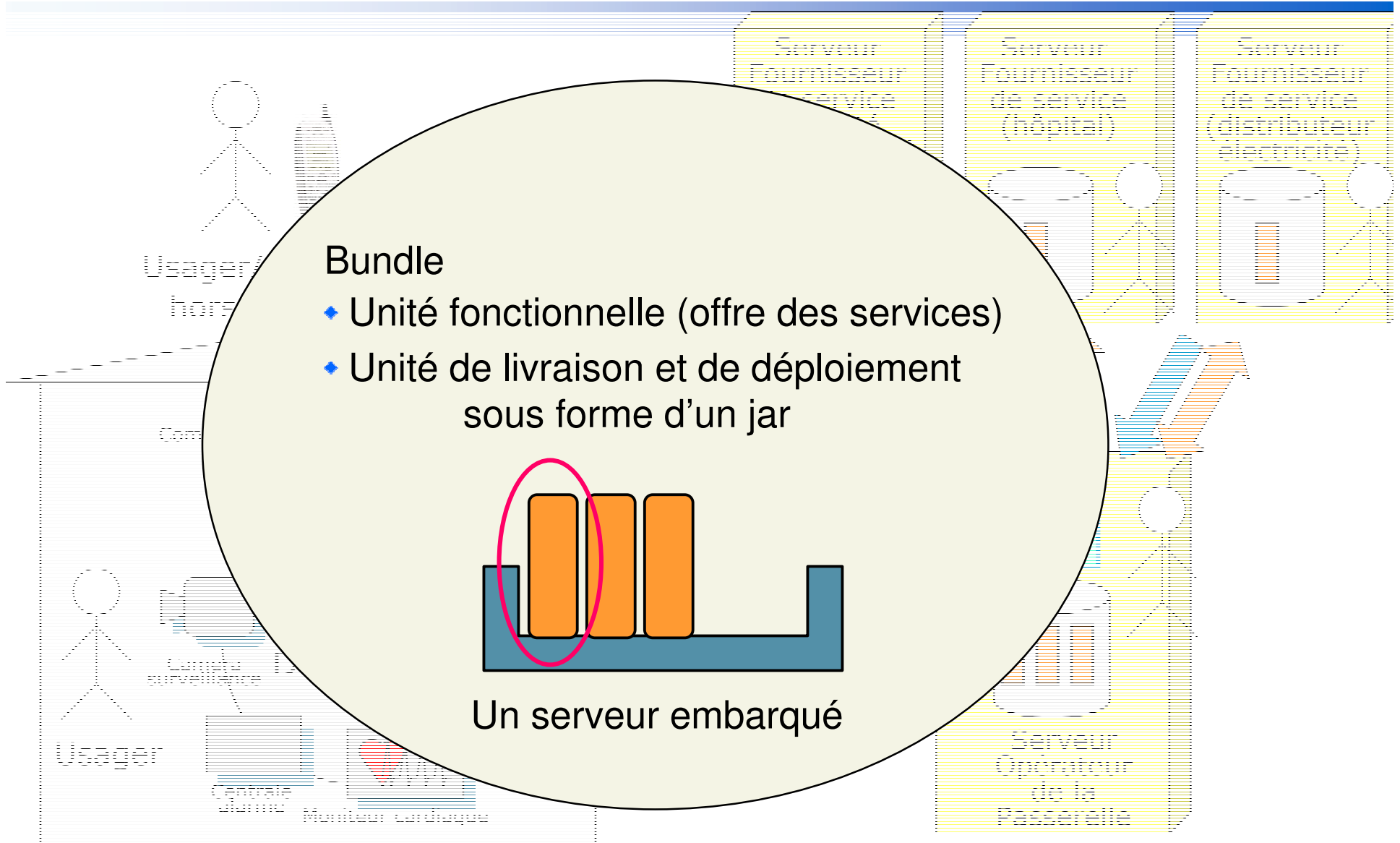
## Passerelle, Bundle et Service

### Bundle

- ◆ Unité fonctionnelle (offre des services)
- ◆ Unité de livraison et de déploiement sous forme d'un jar

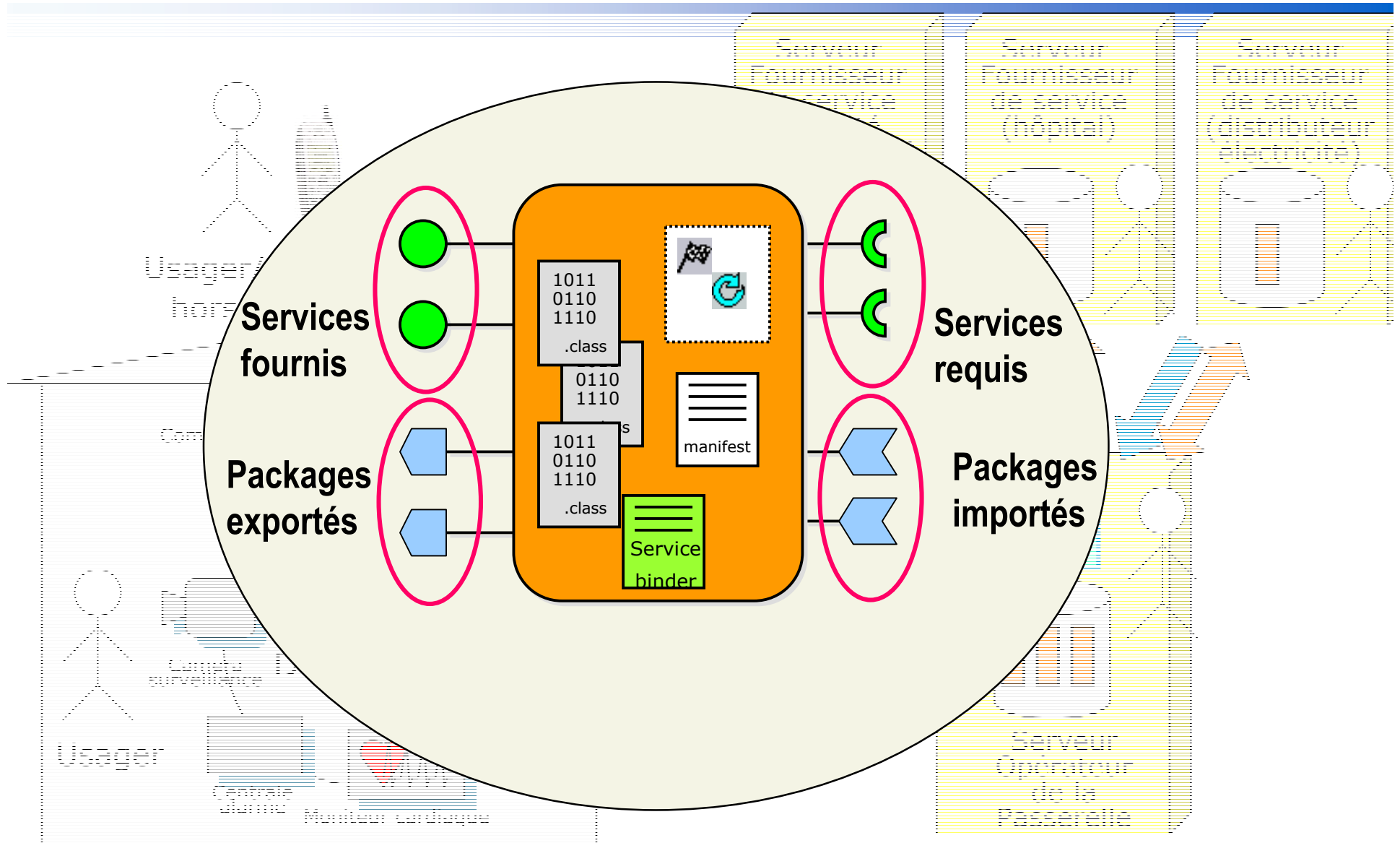


Un serveur embarqué



# OSGi

## Structure d'un bundle





# Pros vs Cons OSGi pour la iTV

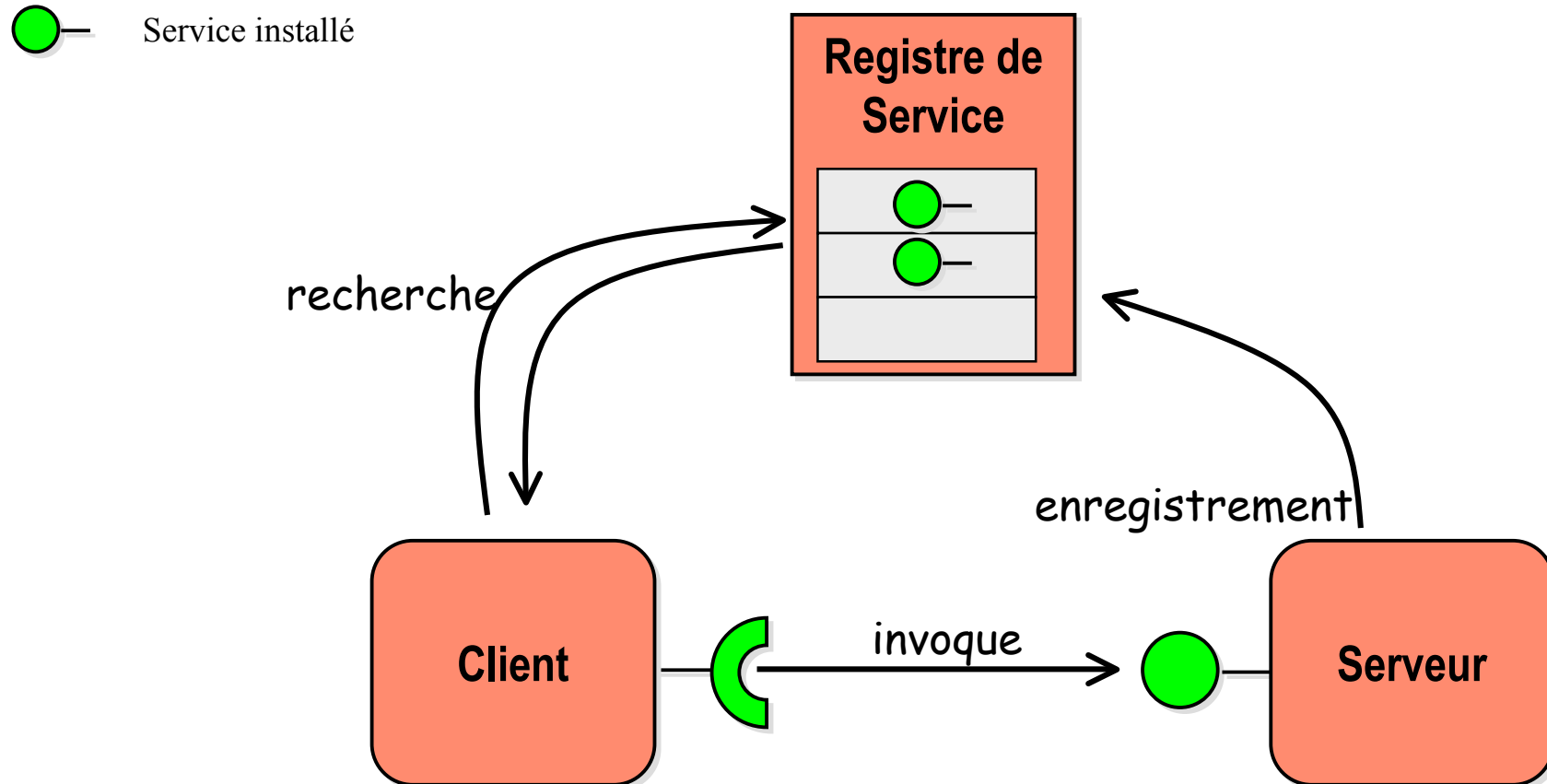
## ■ Pros

- Déploiement dynamique
  - Installation, Mis à jour, Désinstallation
- Applications dynamiques
  - Prise en compte de l'apparition/disparition de services

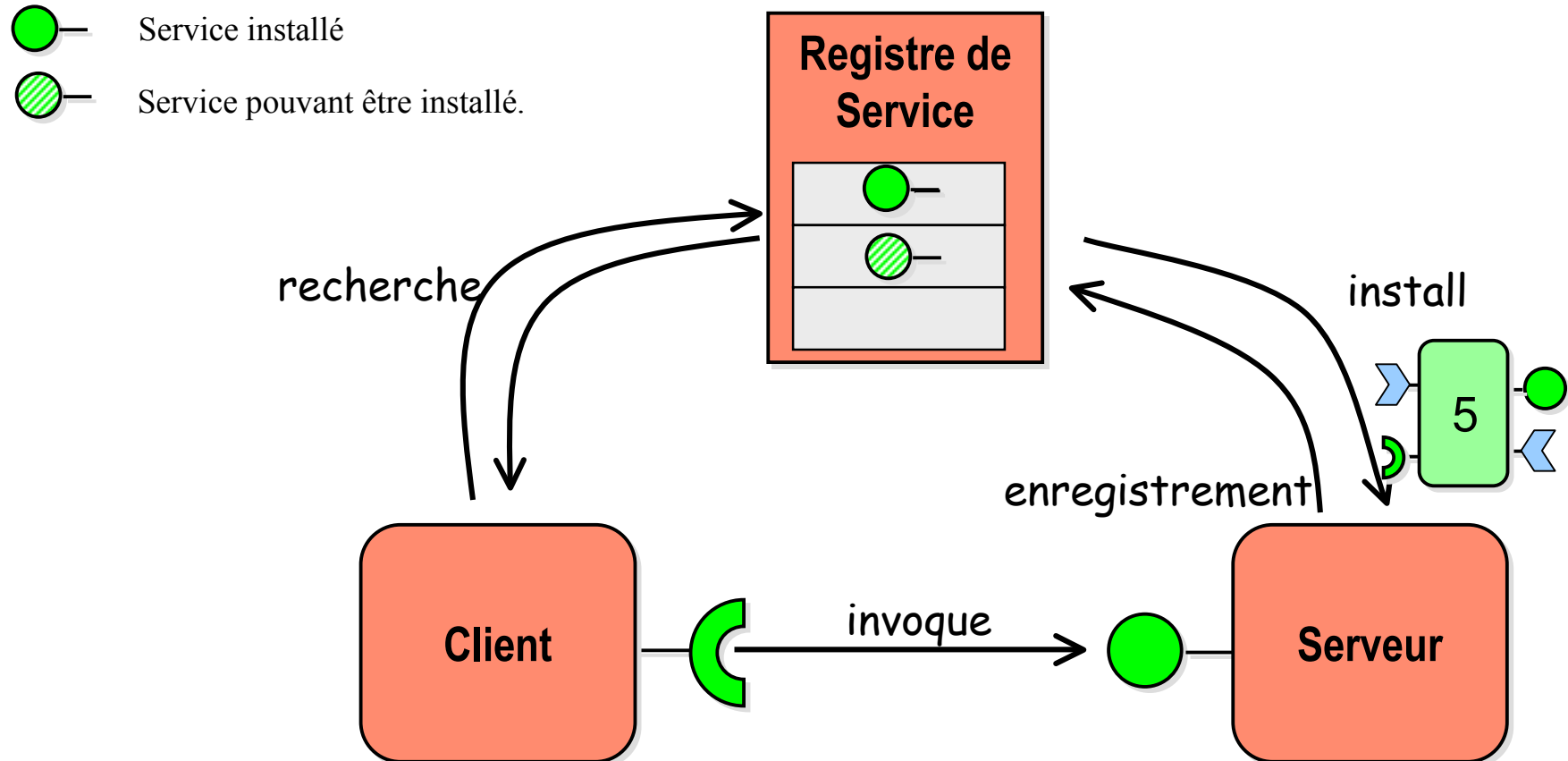
## ■ Cons

- Courtage des seuls services activés localement
- Or plusieurs centaines/milliers services sont potentiellement activables sur un terminal

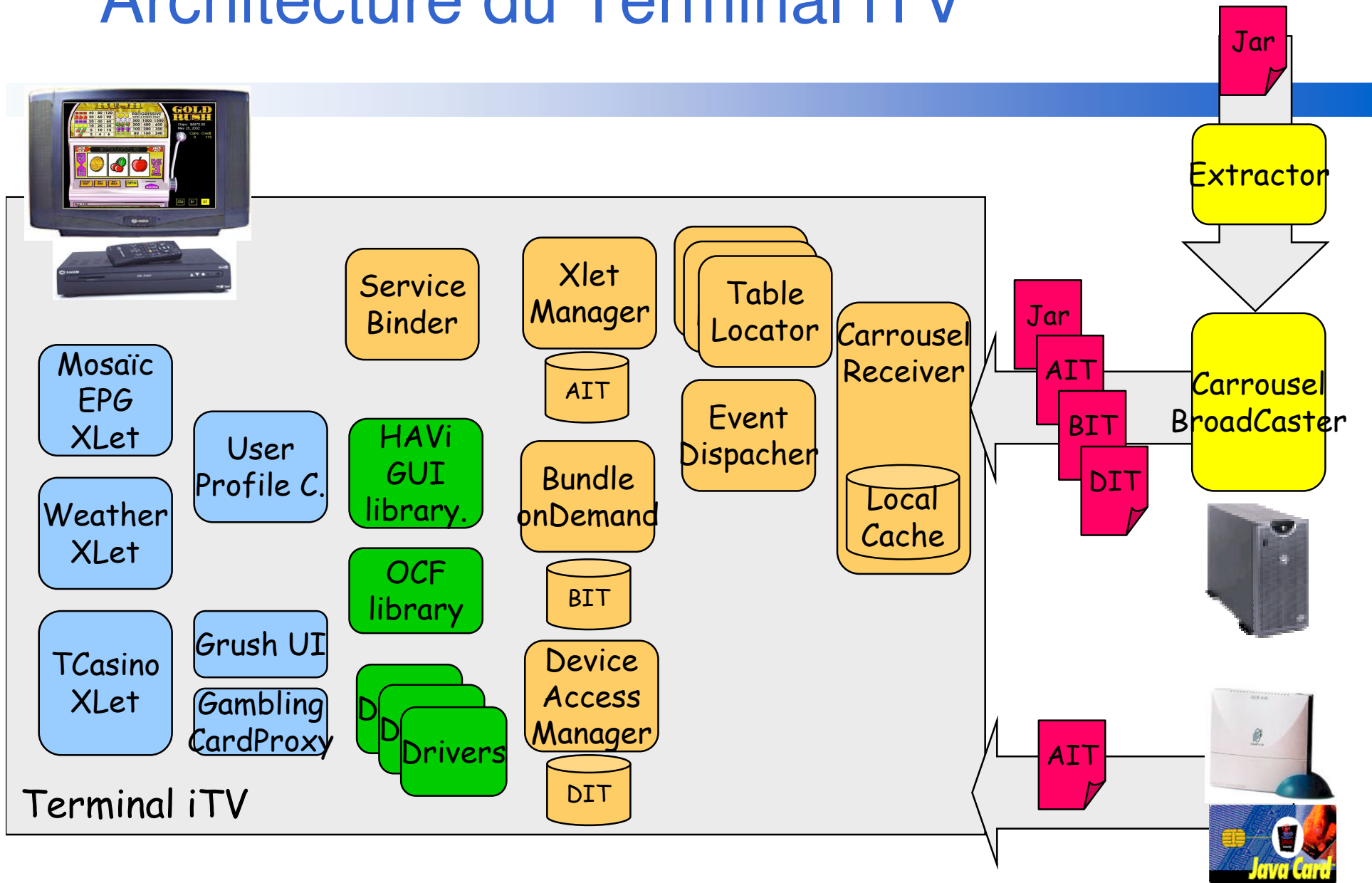
# Installation de service à la demande



# Installation de service à la demande



# Architecture du Terminal iTV



# Réalisation et Expérimentations

## ■ Simulateur réseau opérateur iTV

- Système de fichier IP MultiCast
- coté diffuseur / coté récepteur

## ■ Terminal

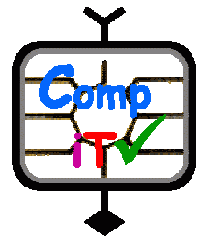
- Pentium II 350Mhz Linux/Win2000, 64MB RAM
- OSCAR 0.9.4 porté sur pJava 1.1
- Plusieurs Xlet de démonstration
  - EPG, TCasino, TTicTacToe, ...
- Drivers Lecteurs + Cartes

## ■ Carte

- Applets JavaCard
  - AITApplet : liste d'applications à insérer dans l'EPG (et à démarrer)
  - TCasinoSecureEngineApplets, ...

# Conclusion

- Déploiement dynamique de services à la demande
  - dans un contexte broadcast/multicast
    - Applicable au téléchargement pull
  - lors de l'insertion d'une carte à puce « abonné »
- Proposition et implémentation d'un middleware iTV
  - Architecture dynamique
- Projet COMPiTV - RNRT 2001
  - Décembre 2001 – Décembre 2003
  - Canal Plus Technologies, Gemplus R&D, LIFL, LAMIH, LSR



# Perspectives

- **Package à la demande**
  - Résolution des dépendances de package
  - Optimisation possible avec la résolution de services
- **Conteneurs extensibles dynamiques**
  - OSGi
  - Composant simple (Services Fonctionnels)
  - + Services non fonctionnels requis par le composant
    - sécurité, trace, ...



Bonus  
OSGiTV





# Démarrage des applications (Xlet)

## ■ Actions

- Démarrer/Arrêter une application (Xlet)
- Mettre/Enlever une Xlet dans la grille de programme

## ■ A partir de

- Flux diffusé (commun à toutes les STB)
- Carte à puce (seulement sur la STB où est insérée la carte)

## ■ Informations diffusées (AIT)

- Configurées par l'opérateur

xletEPG	AUTOSTART
xletWeather	STORE
xletCasino	UNSTORE

# Installation de Service à la Demande

## ■ Informations diffusées (SIT)

- Extraites des Manifest et Metadata.xml

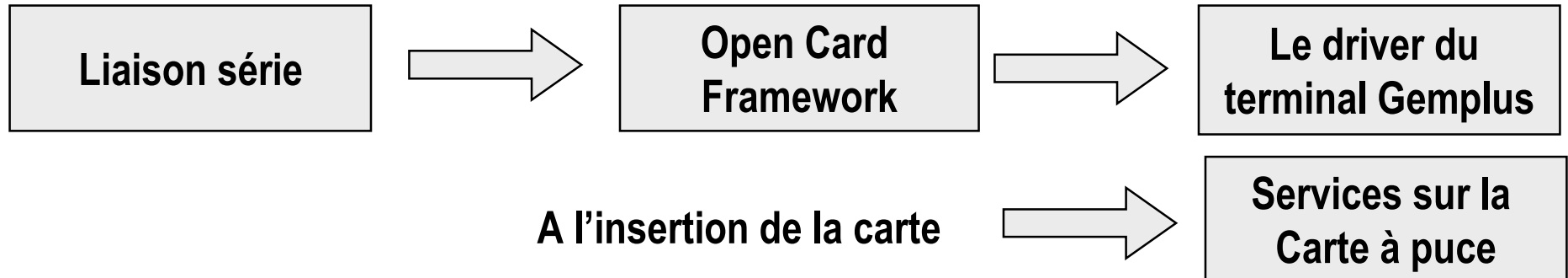
Interface	fr.imag.otrebmuh.bundle.runtime.interfaces.ComponentFactory	
Propriétés	version	1.0.0
	component	compitv.components.ticTacToe
	provides	javax.tv.xlet.Xlet
	xletName	ticTacToe
URL	mcast://224.2.2.2:10000/ticTacToe.jar	

Interface	compitv.tcasino.gamblingcard.xlets.ticTacToe.TicTacToeEngine	
Propriétés	objectClass	compitv.tcasino.gambler.xlets.ticTacToe.TicTacToeEngine
URL	mcast://224.2.2.2:10000/ticTacToeEngine.jar	

# Installation de drivers

## ■ DeviceManager Standard d'OSGi

- Opération standard de raffinement des drivers



## ■ Device Locator sur flux diffusé

## ■ Information diffusée (DIT)


Filtre LDAP	"(deviceCategory=ocf)"
Id	mcast://224.2.2.2:10000/serialDriverGemplusTerminal.jar
URL	mcast://224.2.2.2:10000/serialDriverGemplusTerminal.jar

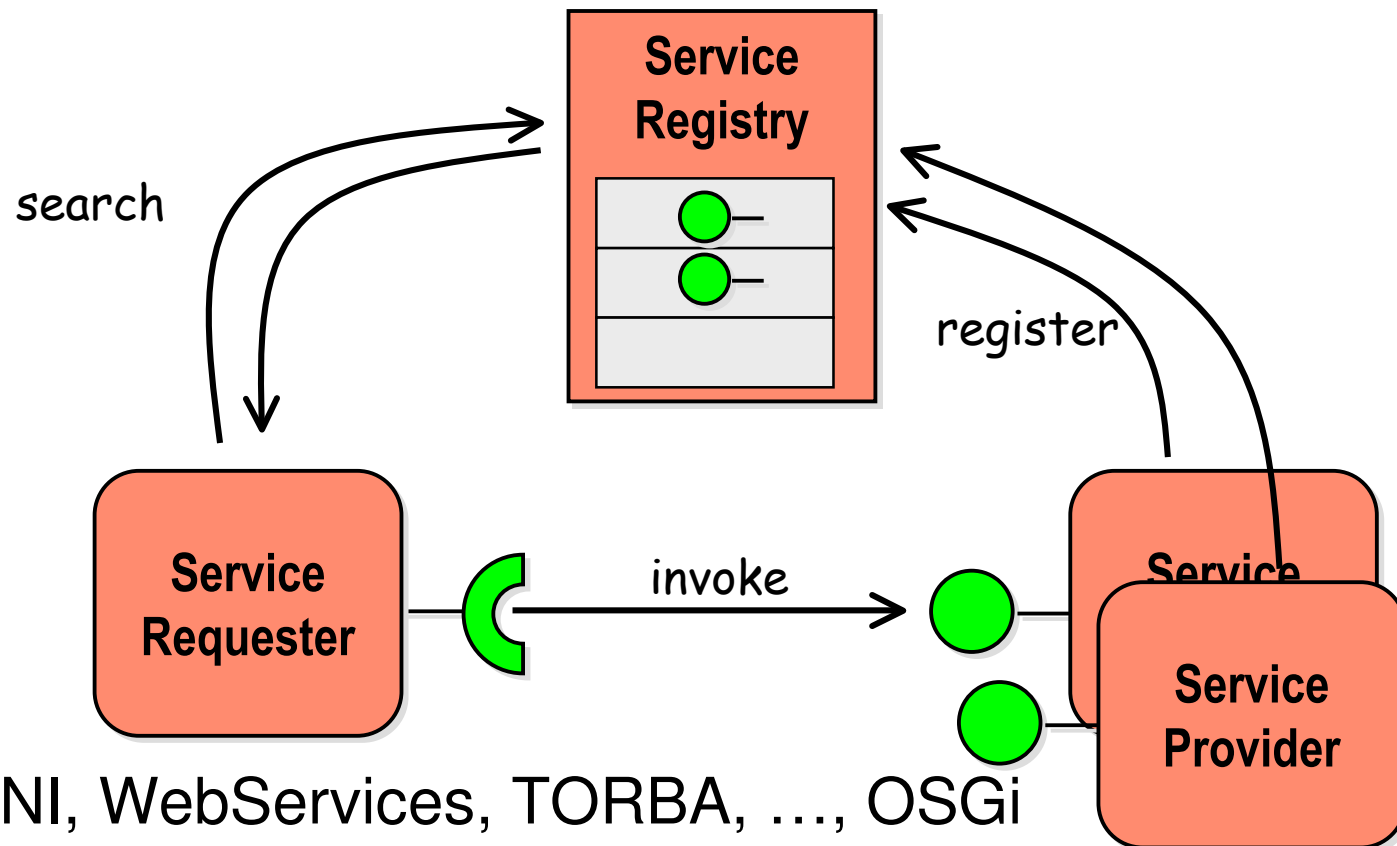


Bonus  
OSGi



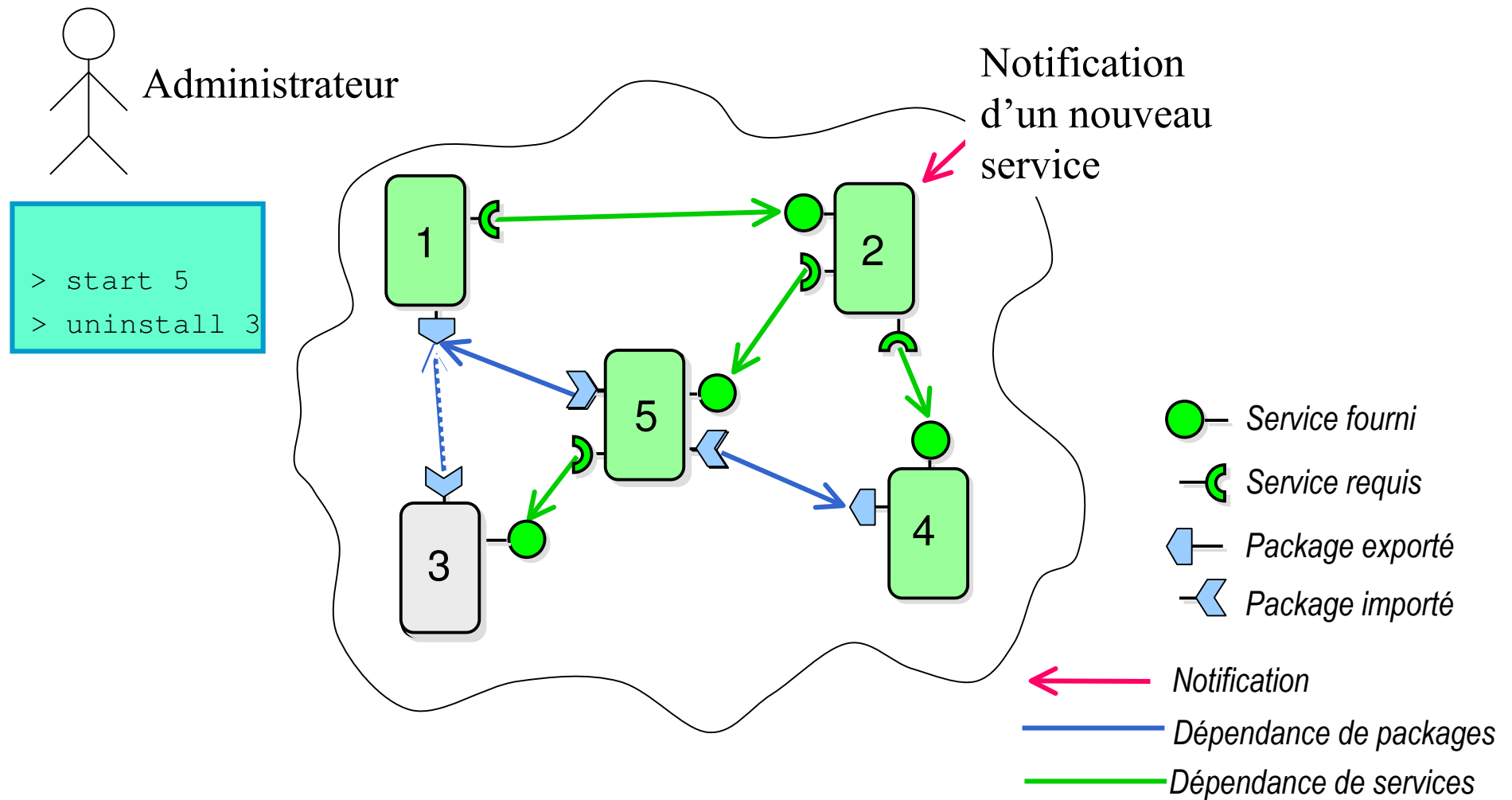
# Rappel sur la programmation orienté service

- Les services  sont « invariants »

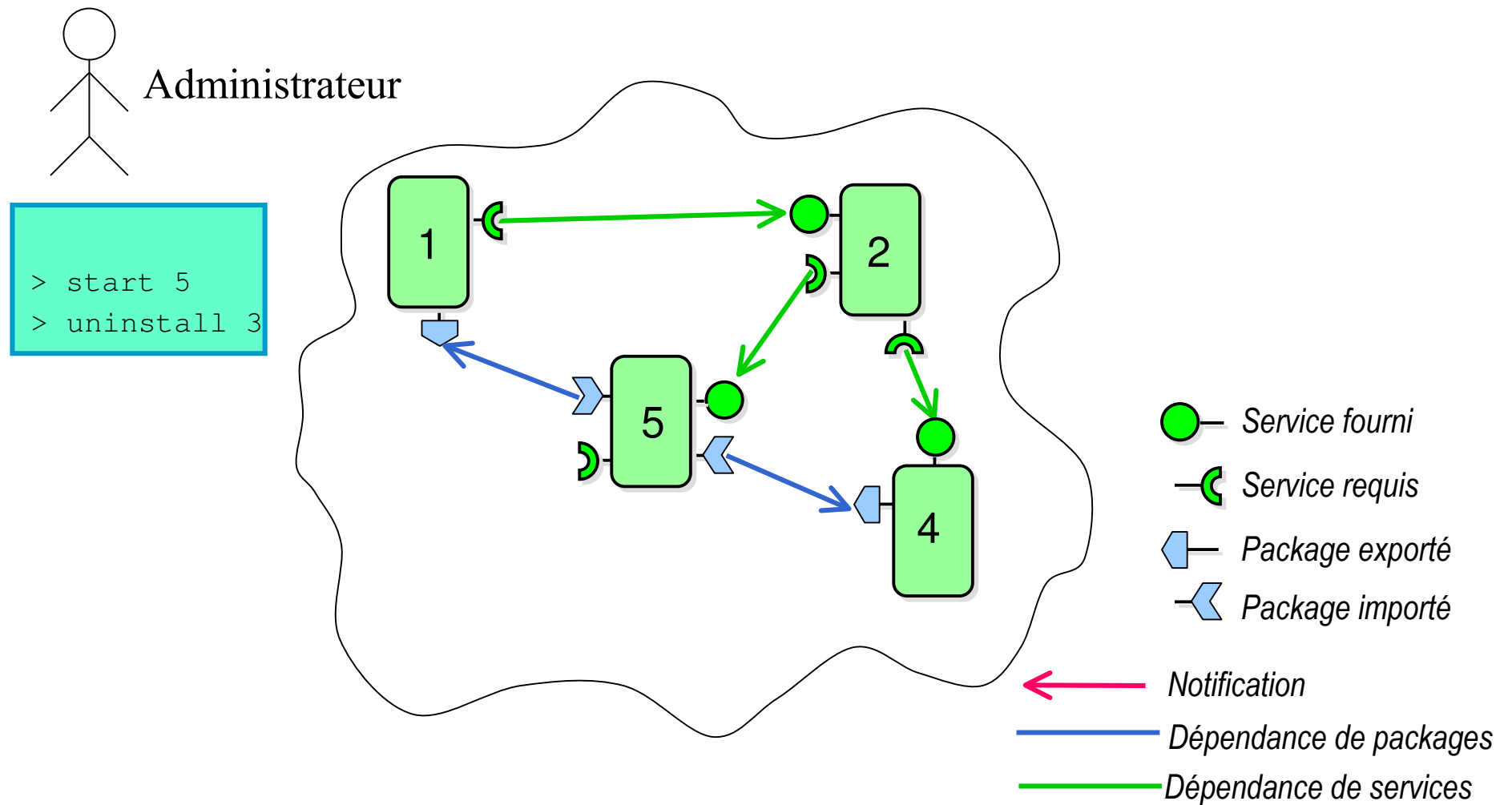


- Ex: JINI, WebServices, TORBA, ..., OSGi
  - Cependant
    - Tout concepteur d'application est gagnant à distribuer son application comme un ensemble de bundles

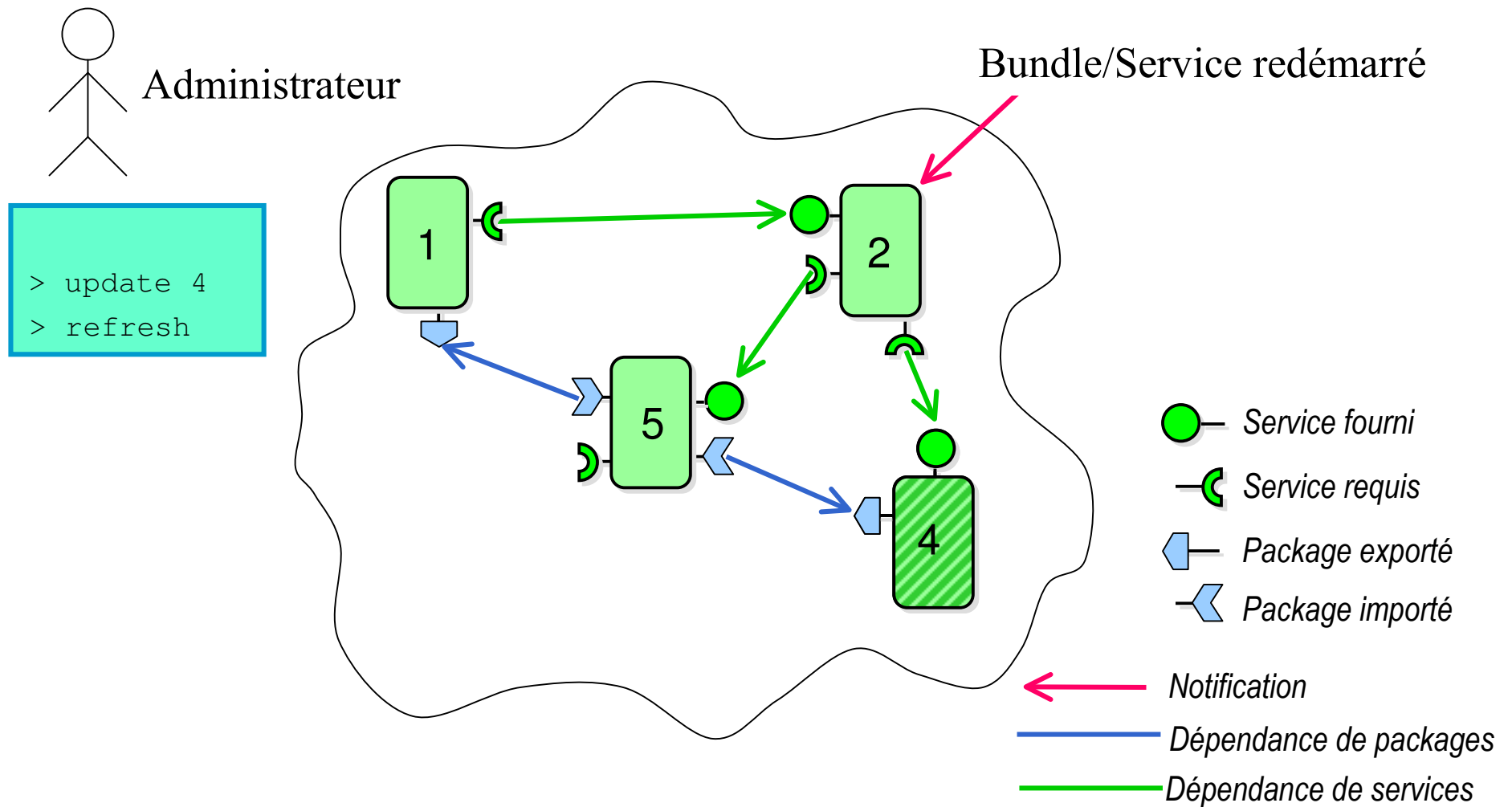
# Dépendance & Dynamisme



# Dépendance & Dynamisme



# Dépendance & Dynamisme





# OSGi

## Déploiement dynamique de services

- **Déploiement dynamique d'applications sans interruption de la passerelle**
  - Installation, Lancement, Mise à jour, Arrêt, Retrait
  - Résolution des dépendances versionnées de code
  - Programmation orientée service
    - courtage local des services actifs
- **Domaines d'application cibles**
  - Systèmes embarqués : systèmes à mémoire restreinte
    - Véhicule de transport (*automotive*)
    - Passerelle résidentiel/domotique/immotique
    - Contrôle industriel
    - ...
  - mais également les applications à plugin ...