

# OSGi

## et le projet IST Amigo

Anne Gérodolle, André Bottaro

05/09/06



recherche & développement



# sommaire

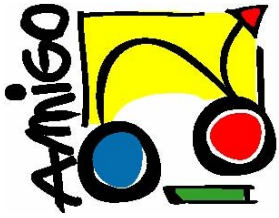
## 1 ■ Contexte : le projet IST Amigo

- Objectifs
- Middleware intéropérable

## 2 ■ OSGi dans Amigo

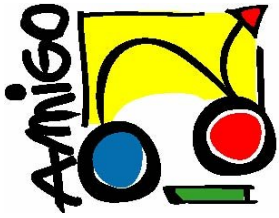
- Objectifs
- Architecture
- Intégration de WS-discovery

## 3 ■ Perspectives et travaux en cours



# Le projet Amigo

# 1



# Amigo



Début : septembre 2004  
42 mois

**The Netherlands**

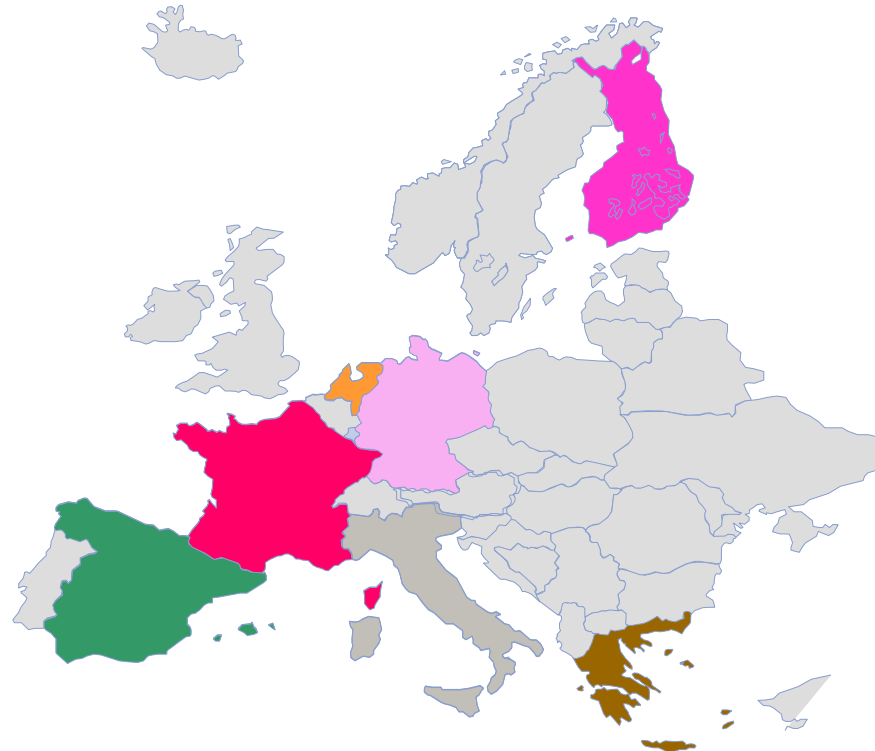
PHILIPS  
Telematica  
Instituut

**France**

INRIA  
france telecom

**Spain**

Ikerlan  
CENTRO DE INVESTIGACIONES  
TECNOLÓGICAS  
Telefonica  
FAGOR



**Germany**

IPSI  
PA-DERBORN  
IMS  
Microsoft

**Finland**

VTT

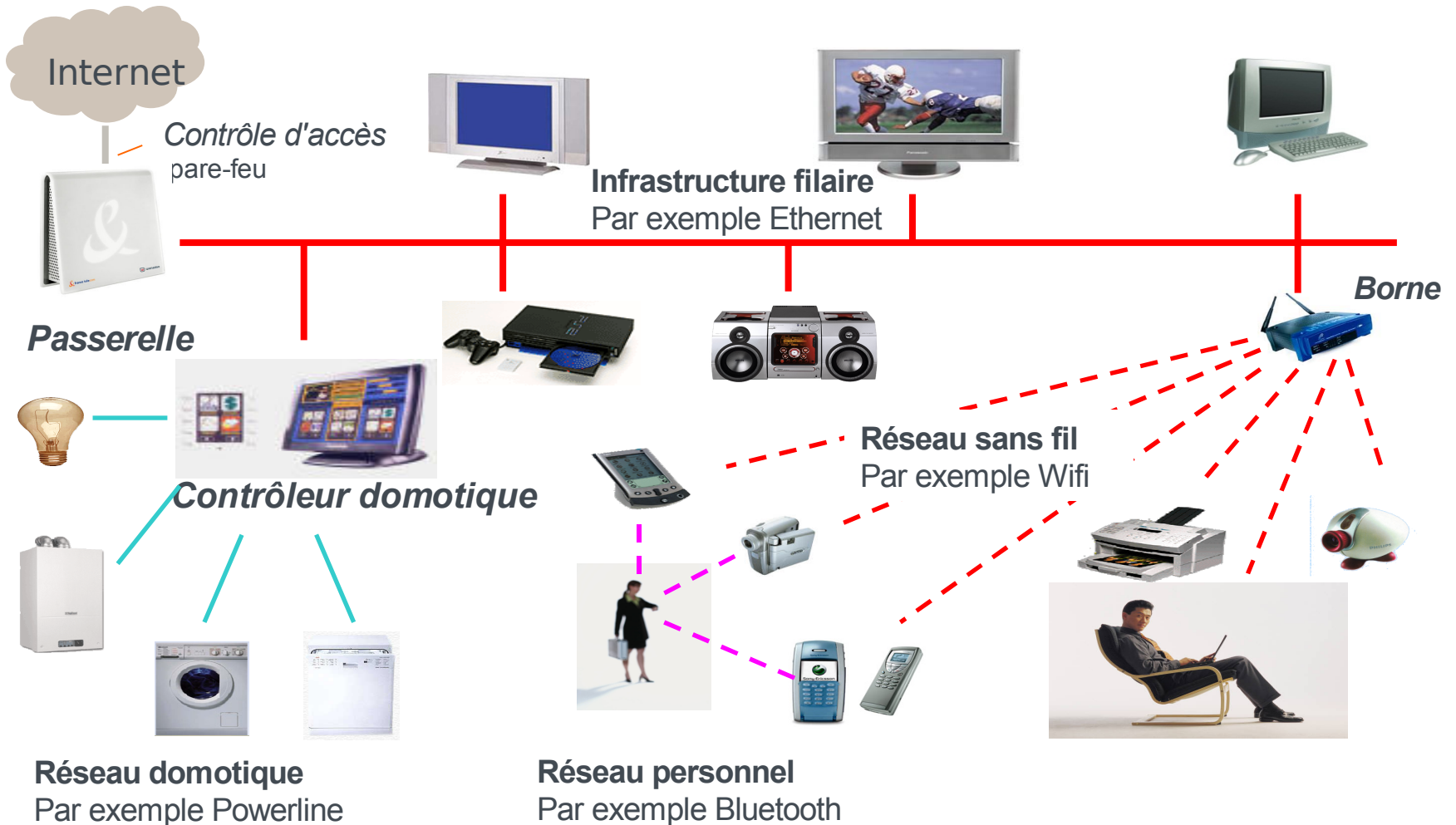
**Italy**

GIUGIARO

**Greece**

KNOWLEDGE  
ETIZEY  
ICCS

# Le réseau domestique de demain



# Pourquoi pas aujourd'hui ?

Deux raisons identifiées:

- Technique : difficultés pour l'installation, la configuration, la programmation de ces réseaux, infrastructures et périphériques.
- Intérêt : quels avantages pour moi si ma machine à laver peut parler à mon téléviseur?
  
- Plusieurs axes de recherche dans Amigo :
  - Proposer des scénarios d'usage
  - Développer des "services intelligents"
  - Disposer d'un **middleware interopérable**

# Middleware interopérable

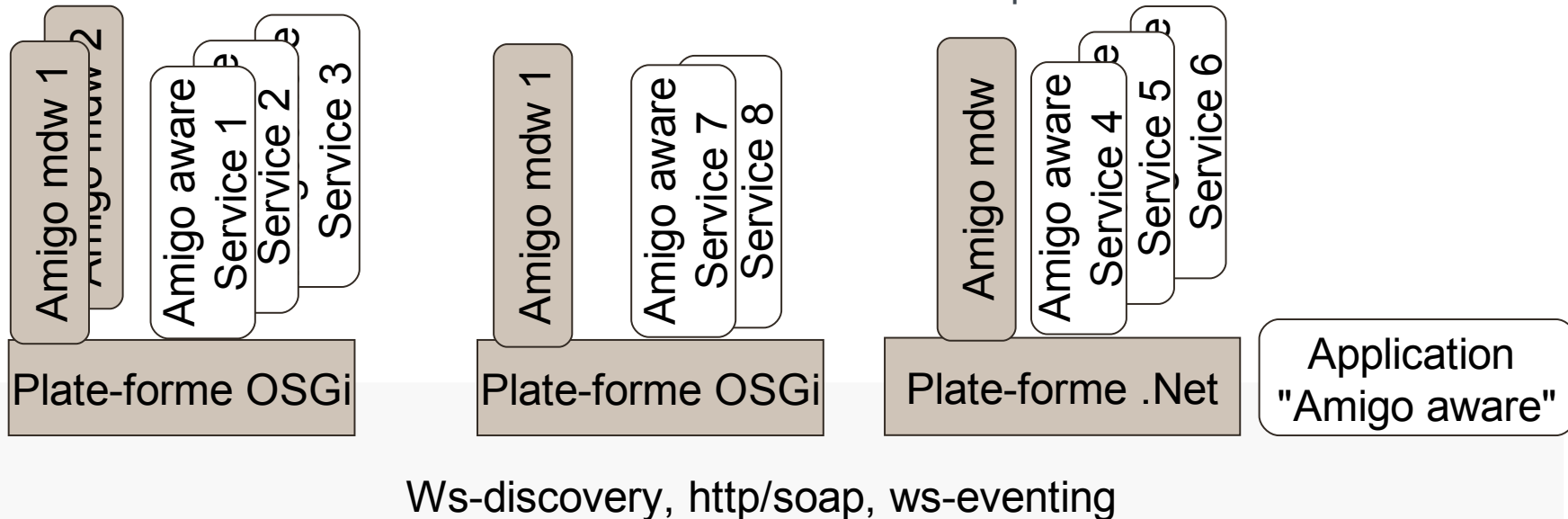
- But: permettre la composition d'éléments de service présents dans le réseau pour offrir à l' "utilisateur" des services de haut niveau
- Faciliter la mise en relation les éléments d'un réseau Amigo
  - Des clients ou des serveurs "legacy" (un media renderer UPnP, une machine à laver programmable,...)
  - Des logiciels "Amigo aware"

c'est à dire la découverte, la communication synchrone, l'envoi asynchrone d'événements

- Des méthodes d'interopérabilité (passerelles) pour intégrer les clients ou serveurs "legacy"
- Choix d'un jeu de protocoles privilégiés pour la communication entre logiciels "Amigo aware"
  - Technos "ws-\*" : ws-discovery, ws-eventing, http/soap

# Infrastructure de déploiement

- Proposition de deux environnements permettant de développer des composants Amigo pour .Net et OSGi
  - Encapsulation des protocoles dans des abstractions facilitant le travail du développeur
  - L'utilisation de ces environnements favorise l'interopérabilité





# OSGi dans Amigo

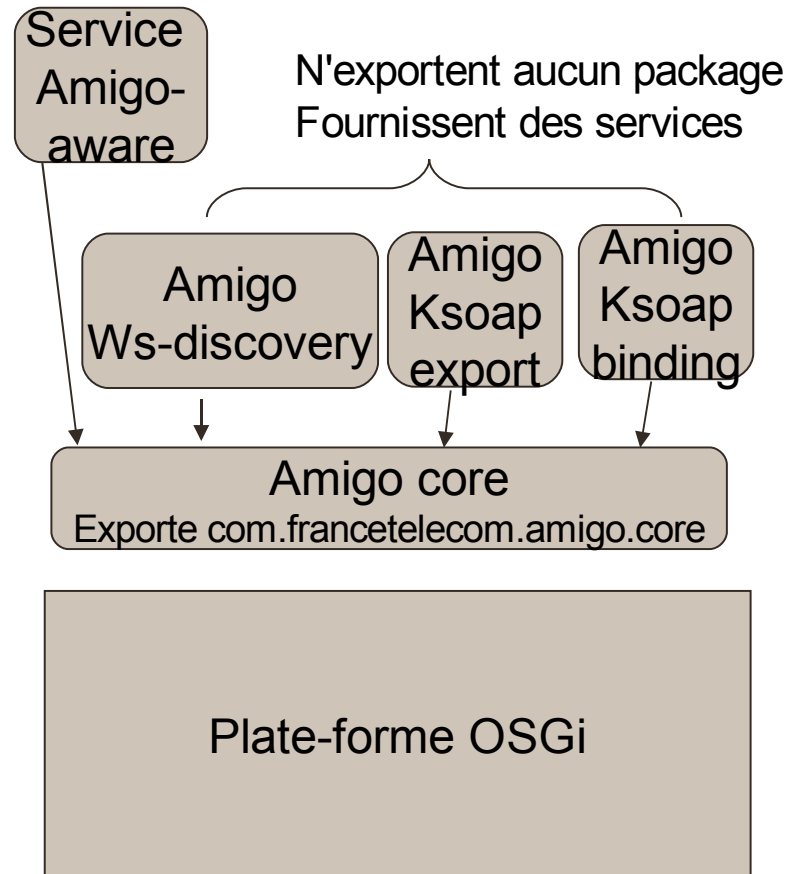
# 2

# Objectifs

- Proposer OSGi comme plate-forme pour le déploiement des services Amigo Java
- Fournir sous forme de bundles OSGi des bibliothèques pour la découverte, l'exposition et l'utilisation de services distants
  - Faciliter le travail du développeur (abstractions de haut niveau indépendantes du protocole sous-jacent)
  - Assurer l'interopérabilité avec les composants développés en utilisant le framework .Net
- Cibles de déploiement: J2SE, J2ME/personal profile

# Architecture

- Un bundle "core" fournit une API indépendante des protocoles
- Des bundles d'adaptation
  - SLP (Service Location Protocol)
  - Ws-discovery (amigo\_wsdiscovery)
  - client ksoap (amigo\_ksoap\_binding)
  - Exposition de service ksoap (amigo\_export\_binding)
- Seul le bundle "core" exporte des packages et est utilisé à la compilation.
- Les protocoles réellement utilisés sont choisis au run-time



# Exemple de code

## ■ Exposition d'un service (serveur)

```
BookShop shop = new BookShopImpl(...);  
AmigoServiceExporter exporter = (AmigoServiceExporter)  
    findOSGiService(AmigoServiceExporter.class);  
AmigoExportedService service = exporter.createService(BookShop.class, shop);  
service.addProperty("ServiceType", "Bookshop");  
service.addProperty("Scope", "urn:amigo");  
AmigoLookup lookup=(AmigoLookup) findOSGiService(AmigoLookup.class);  
lookup.register(service);
```

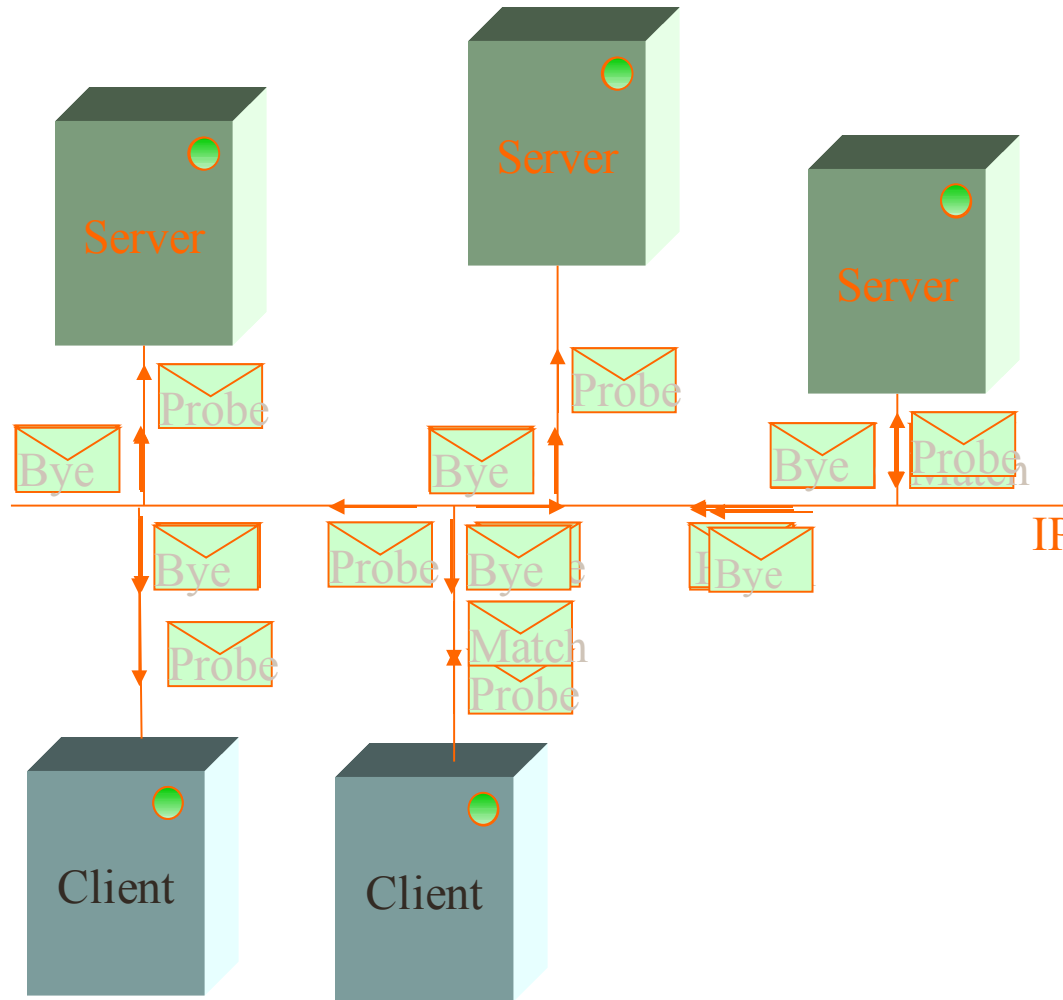
## ■ Recherche d'un service sur le réseau et utilisation (client)

```
AmigoLookup lookup = findOSGiService(AmigoLookup.class);  
AmigoService service = lookup.lookupFirstService("urn:amigo", "Bookshop");  
BookShop stub = service.getSpecificStub(BookShop.class);  
Book[] books = stub.getBooks(...);
```

# WS-discovery

- Protocole de découverte de services sur un réseau local
- Proposition d'un consortium réunissant Microsoft, BEA, Canon, Intel et Webmethods.
- s'appuie sur les standards du W3C (xml, soap, WS-addressing, etc...)
- Retenue dans DPWS (Device Profile for Web services): proposition de Microsoft pour UPnP V2
- Pas d'implémentation Java Open source au 1<sup>er</sup> janvier 2006

# WS-Discovery - Fonctionnement



# Implémentation

- Une API proche du protocole WS-discovery
  - Interface cliente : envoi de message Probe, armement de listeners pour la réception des réponses,...
  - Interface serveur : envoi de Hello/Bye, réception des messages probe et réponse
- Une encapsulation implémentant l'API "Amigo core"  
= un bundle OSGi fournissant un service qui implémente AmigoLookup

# Résultats

- Février 2006
  - Mise en place d'un "Amigo Bundle Repository"
  - Fourniture des bundles `amigo_core`, et des bundles d'adaptation `amigo_slp`, `amigo_ksoap_export`, `amigo_ksoap_binding`
- Mai 2006
  - Remplacement de `amigo_slp` par `amigo_wsdiscovery`
  - Enrichissement de l'API `amigo_core`
    - ajout d'une méthode plus proche des abstractions de ws-discovery
    - API pour la gestion des souscriptions à des sources d'événements asynchrone
  - Implémentation de ws-eventing
- API masquant la complexité des protocoles
- Intéropérable avec l'implémentation C# / .Net
- Fonctionne sur J2ME/personal profile
- Limitation : ne permet pas d'utiliser toute la richesse des protocoles
  
- Mise en open source (LGPL) en cours

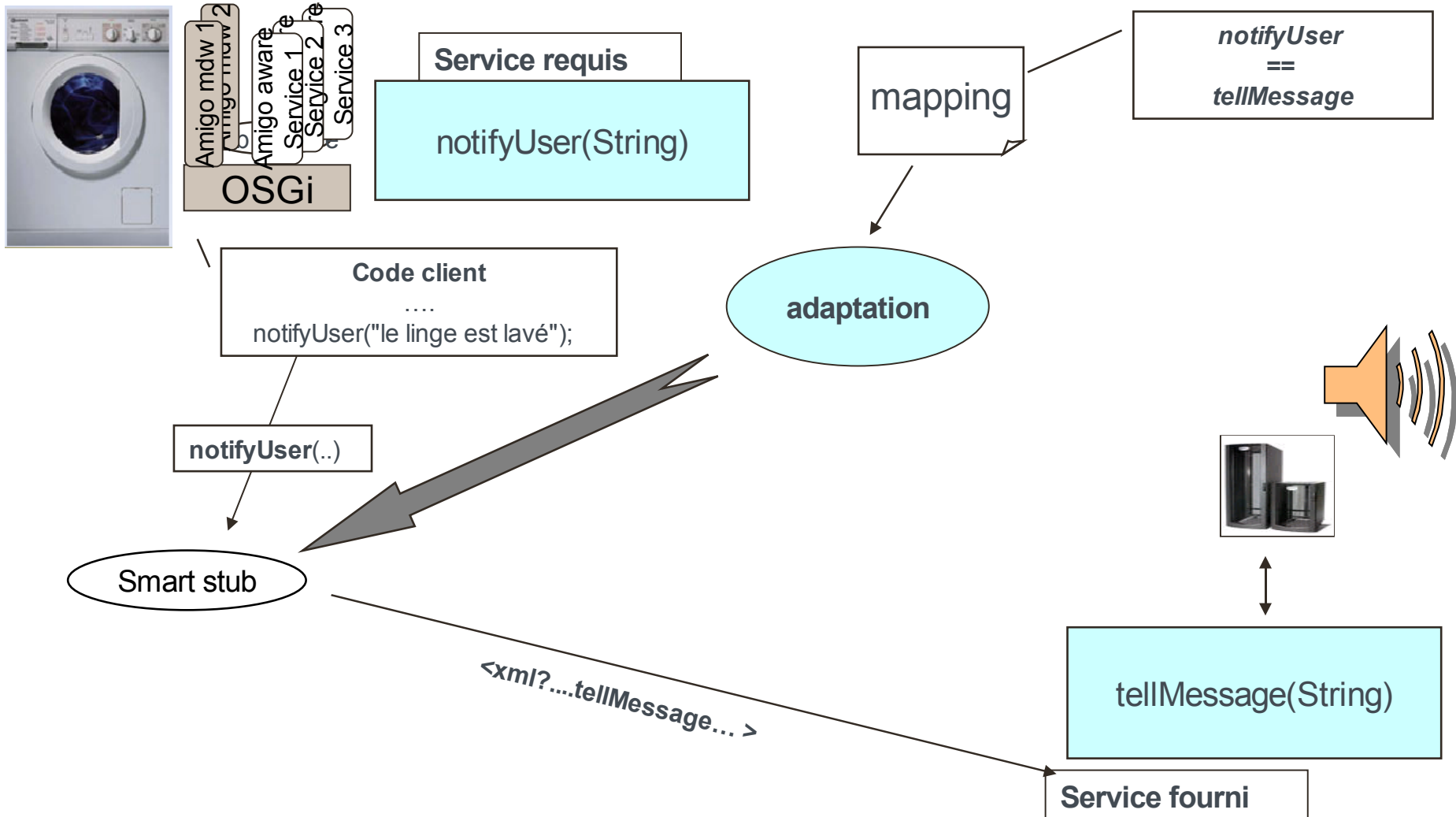


# 3 Perspectives et travaux en cours

# Amigo Service Binder

- Aujourd'hui : appel explicite à l'API `amigo_core`
  - le programmeur gère le cycle de vie de son composant et des liaisons avec les services distants utilisés
  - Possibilité d'utiliser le service binder (OSGi declarative services) pour la découverte des services OSGi locaux (ex : lookup, service exporter)
- Amigo Service Binder : sur le modèle du Service Binder
  - Le programmeur fournit
    - Description des services distants requis et exposés dans un fichier de méta-données
    - Callbacks à appeler si un service distant apparaît / disparaît
  - L' "Amigo Service Binder" gère
    - l'appel au service de découverte distribué
    - La liaison (fabrication de stubs, exposition du service sur le réseau)

# Adaptation sémantique



# Autres travaux relatifs à OSGi dans Amigo

- Déploiement sémantique
  - Frédéric Le Mouel et Noha Ibrahim, INRIA ARES