



**ARMINES**  
STRUCTURE DE  
RECHERCHE CONTRACTUELLE



## *UbiquiTalk*

***Une infrastructure pour l'informatique  
UBIQUitaire en smallTALK***

<http://csl.ensm-douai.fr/UbiquiTalk>

Noury Bouraqadi & Michaël Piel

*25 novembre 2006, Paris*

Soutenu par le CPER TAC 2004-2006 de la région  
Nord-Pas de Calais et par le fond européen FEDER.



Contexte:

# *Informatique Ubiquitaire*

## ■ **Miniaturisation des ordinateurs (nanotechnologies)**



624 MHz
128 Mo RAM
7,7 x 13,5 x 1,5 cm
200 g
...



55 MHz
8 Mo RAM
3,5 x 1,9 x 1,9 cm
18 g
...

## ■ **"ordinateurs" partout et souvent invisible**

- *Voitures (~20 micro), lave-linge, frigots, vêtements, ...*

## ■ **Réseau partout et souvent sans fil**

- *GPRS, UMTS, Wifi, WiMax, Bluetooth...*

Contexte:

# *Informatique Ubiquitaire*

- **Plusieurs machines par personne (téléphone, PDA, ...)**
  - *Access aux services (logiciels + données) depuis différentes machines*
  - *Utilisation simultanée de plusieurs machines*
    - *Connexion, "Synchronization" = Replication de données, ...*
- **Mobilité à la fois des personnes et des machines**
  - *Seules quelques machines se déplacent avec leur propriétaires*
    - *Souvent petites => peu de ressources (memoire, batterie...)*
  - *Access distant aux services (logiciels + données)*
    - *Echange de données, travail collaboratif, ...*
  - *Environment variable*
    - *Type et caractéristiques du réseau, périphériques, ...*





# *Finalité d'UbiquiTalk*

- ***Simplifier la production de logiciels distribués***
  - *Canevas ("Framework") pour le développement*
  - *Middleware pour le déploiement automatique*
  
- ***Minimiser les tâches d'administration***
  - *"Zero networking configuration"*

# *Infrastructure pour l'info. ubiquitaire*



---

- **Infrastructure =**

- *Middleware P2P*
- *Canevas ("Framework")*
  - *Objets applicatifs*
  - *Interface Utilisateur (UI = "User Interface")*

- **Informatique Ubiquitaire**

- *Interactions distantes non-anticipées*
- *Equipements hétérogènes*



# *Deux Concepts Fondamentaux*

---

- **Hôte ("Host"):**

- *N'importe quelle machine avec des capacités de calcul et une interface réseau*

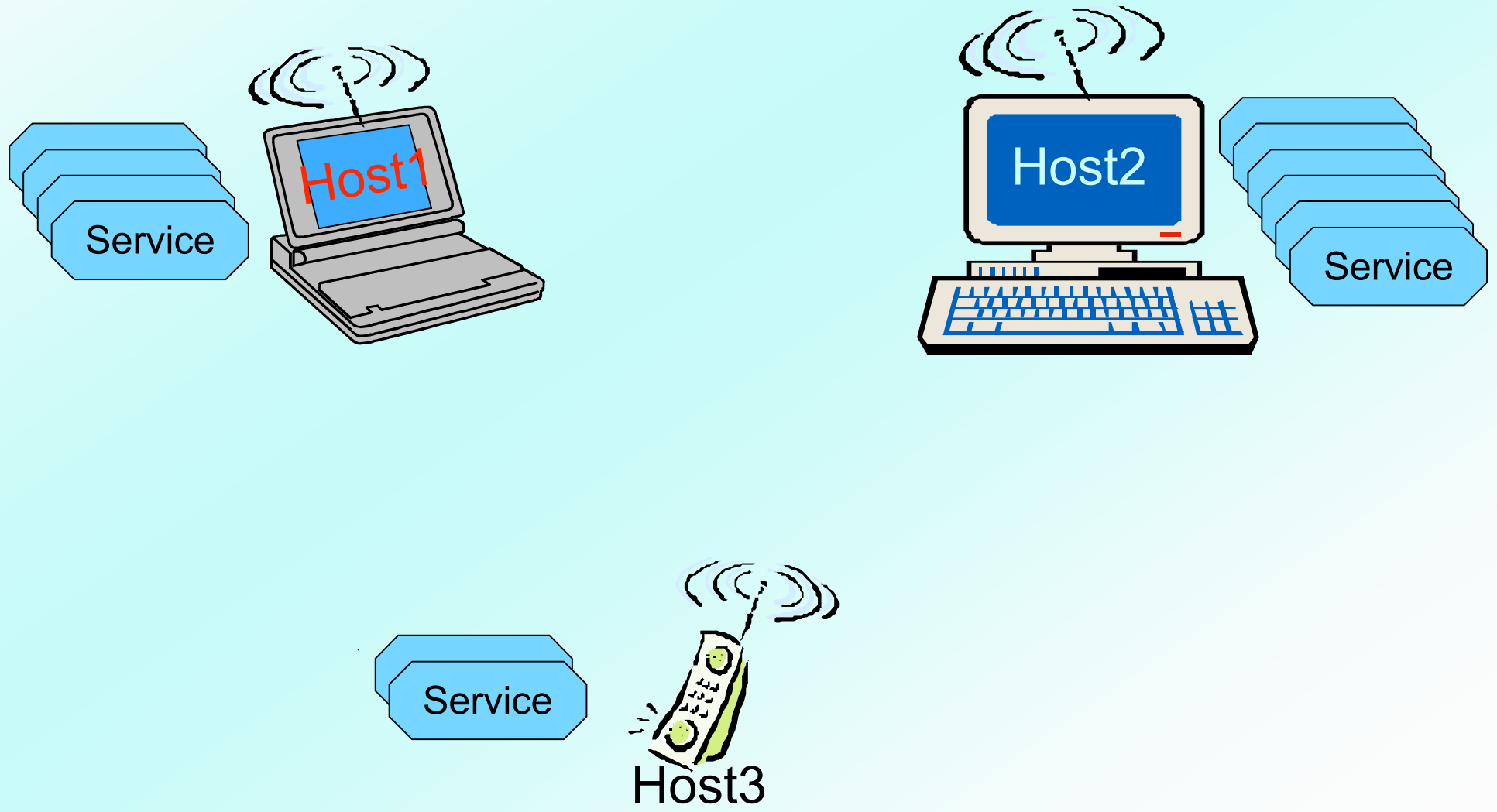
- **Service: objet d'un hôte manipulable à distance**

- *Service Applicatif : réalise un tâche "métier"*
- *Service Middleware : fournit un support pour le fonctionnement du Hôte (i.e. les services applicatif)*
  - *Découverte d'Hôtes, Annuaire de services, ...*

# Vue globale

## *Machines hétérogènes en réseau*

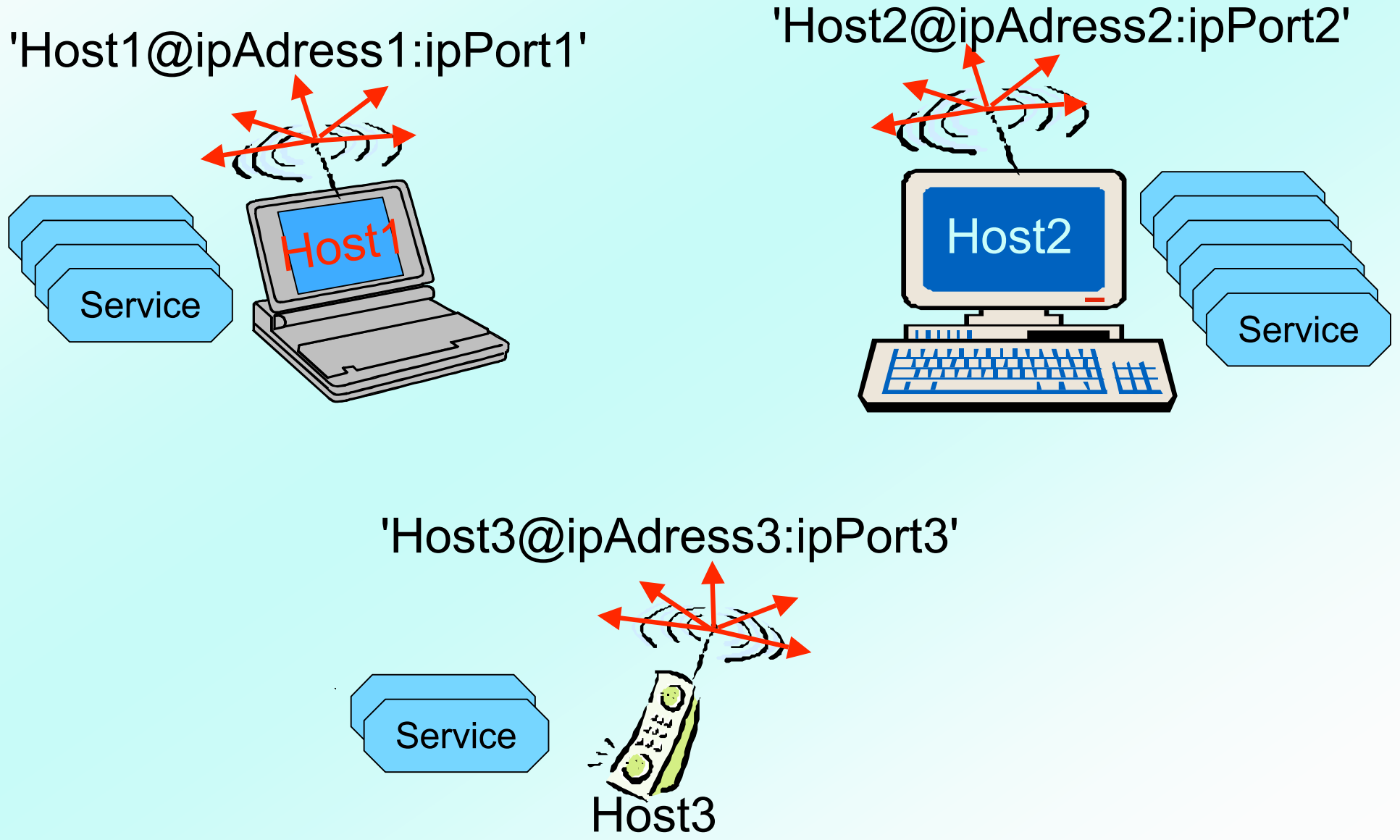
Noury Bouraqadi - Dépt. I. A.



# Vue globale

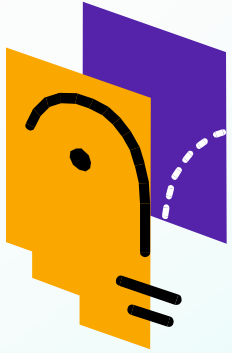
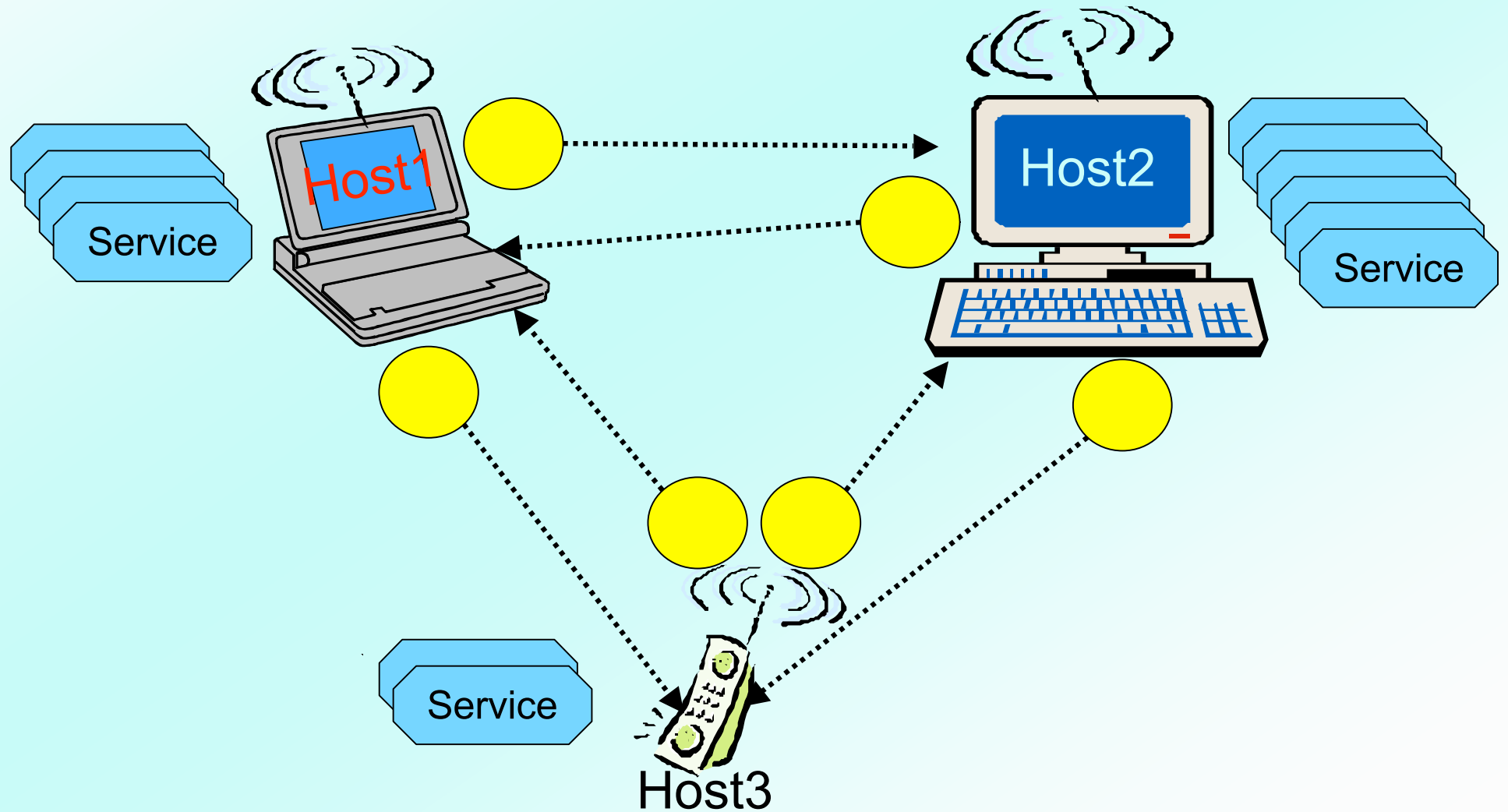
## *Boucle de notification de présence*

Noury Bouraqadi - Dépt. I. A.



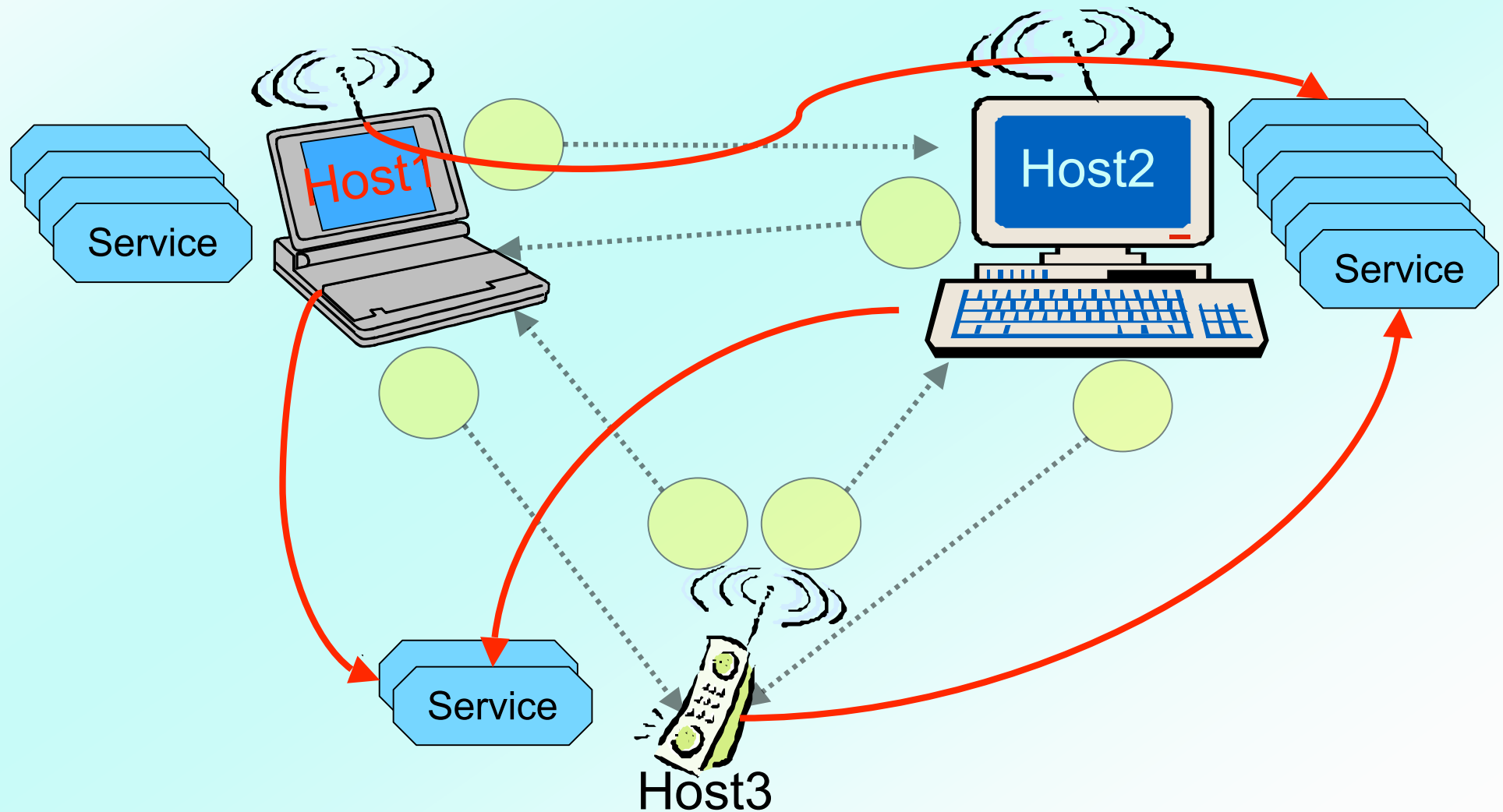
# Vue globale

## *Détection => Proxies sur les autres hôtes*



# Vue globale

*Interaction => Messages distants*

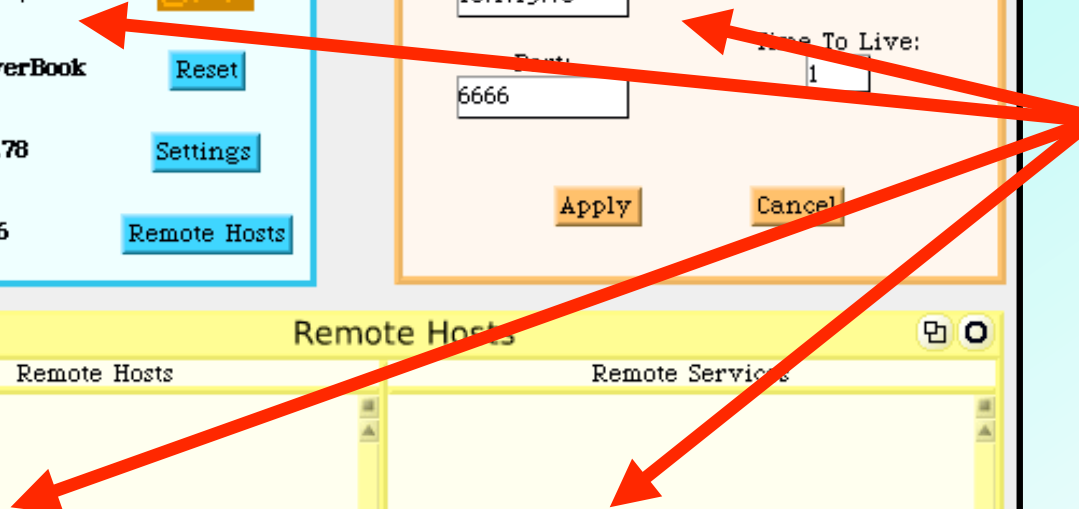
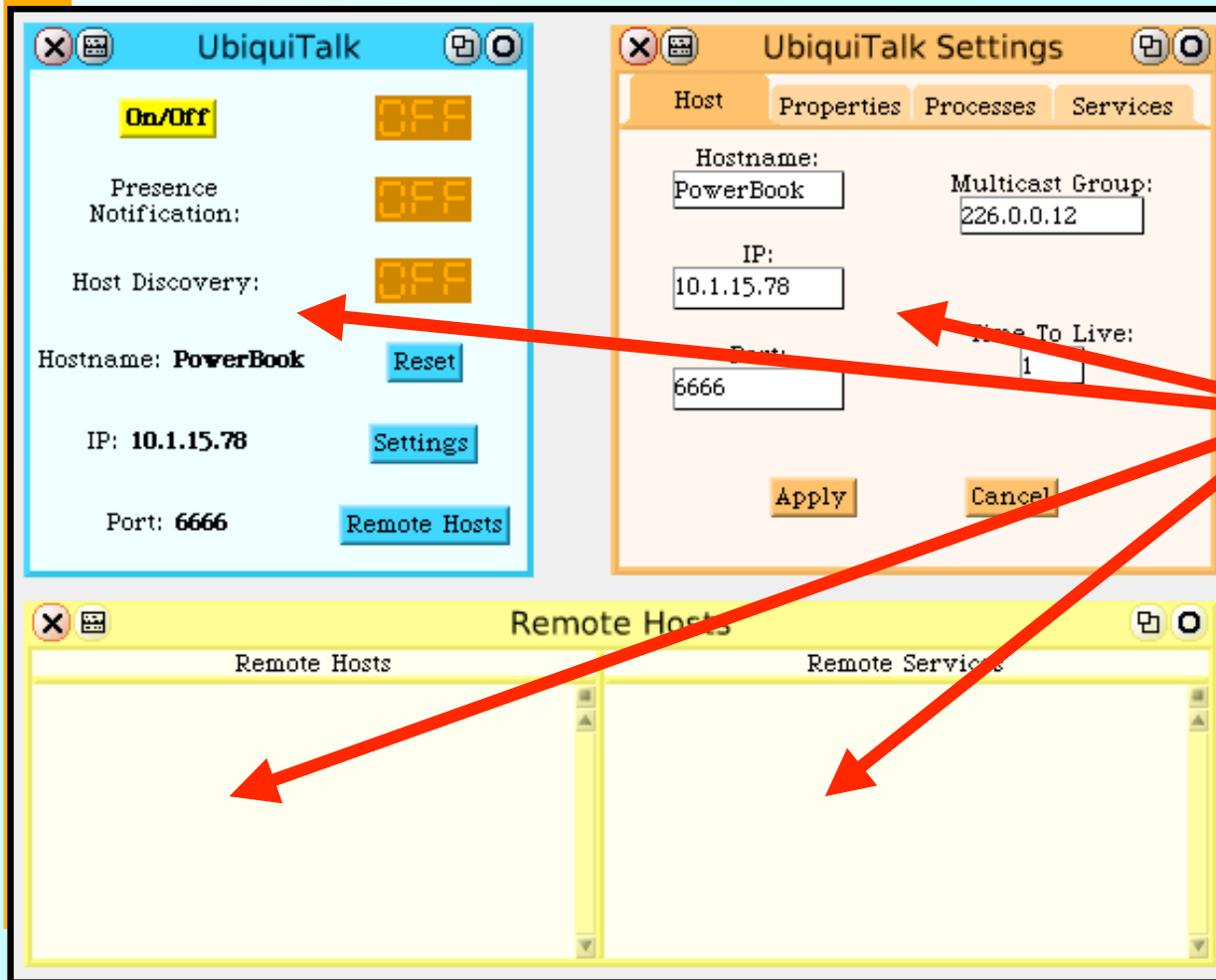


# Deux interfaces graphiques selon la cible

Noury Bouraqadi - Dépt. I. A.

Pour Desktops/Laptops

Pour PDAs





# *Exemples d'applications*

---

- ***Copy/Past distribué***
- ***FTP***
- ***Chat Conférence***
- ***Impression***
- ***Impression géo-localisée***
- ***Administration à distance***
- ***Contrôle distant***

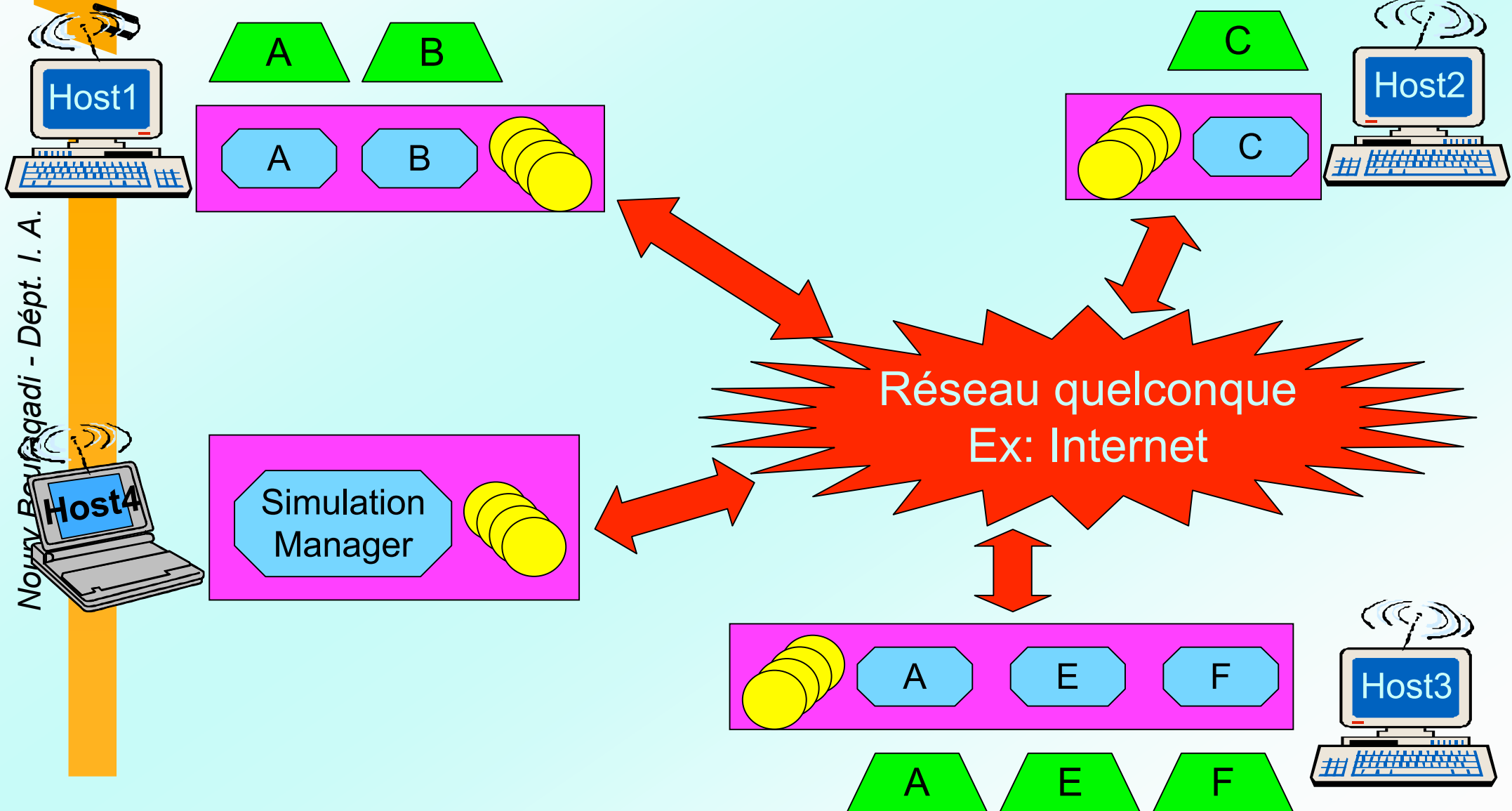
# Application au Calcul Scientifique

## *Problème*

- **Projet en cours**
- **Application à grande échelle avec de multiples utilisateurs**
  - *Consortium de chimistes (industriels et académiques)*
- **Ensemble ouvert de logiciels de simulation**
  - *Développé depuis 15 ans*
  - *Logiciels de simulation "Standalone"*
    - *Pas d'interaction prévue*
  - *Chaque partenaire dispose d'un sous-ensemble différent de logiciels*
- **Objectifs :**
  - *Réaliser des simulations croisées*
  - *Piloter les simulations à distance*

# Application au Calcul Scientifique

## *Solution à base d'UbiquiTalk*



# Application aux systèmes embarqués

## *Sauvetage Robotisé (Robotic Rescue)*

- **Flotte de robots qui coopèrent dans un environnement hostile**
  - *Catastrophe naturelle/industrielle, Accident, Site contaminé ...*
- **Robots pas cher**
  - *Très contraints en ressources*



### Partenaires

- IRD/Géodes, Bondy & Hanoi
- CNRS/MICA, Hanoi
- AUF/IFI/MSI, Hanoi
- INRIA/LORIA/MAIA, Nancy
- ITC, Phnom Penh
- CNRS/INRIA/LRI/TAO, Orsay
- CNRS/GREYC/MAD, Caen
- ARMINES/CSL, Douai

# *Etapes du cycle de vie d'un service*

- 1. *Implanter le service sur l'hôte fournisseur***
  - a) *Implanter les classes nécessaires au client*
  - b) *Implanter les classes nécessaires au service*
  - c) *Implanter la classe du point d'accès au service*
    - *Hérite de la classe abstraite UTService*
    - *Implante les méthodes d'utilisation du service*
- 2. *Déployer le service sur l'hôte fournisseur seulement***
  - *Ajouter le service à la liste des services disponibles*
  - *Activer & Configurer le service (si nécessaire)*
- 3. *Lancer l'hôte fournisseur et l'hôte utilisateur***
  - *Sans ordre particulier, sauf si c'est sur la même machine*
- 4. *Utiliser le service***



## *Démo d'UbiquiTalk*

<http://csl.ensm-douai.fr/UbiquiTalk>

# *Exemple "Hello World"*

## 1a- Client du service

Object subclass: #HelloClient  
instanceVariableNames: "  
classVariableNames: "  
poolDictionaries: "  
category: 'Exemple Cours'

■ ***Méthode de classe :***

*executer: helloService*

*helloService afficherBonjour*

# *Exemple "Hello World"*

## 1b- Classes nécessaires au service

---

- ***Utilisation de l'objet Transcript déjà disponible***
  - *Pas d'implantation nécessaire*

# Exemple "Hello World"

## 1c- Point d'entrée au service (1/2)

UTService subclass: #HelloService  
 instanceVariableNames: "  
 classVariableNames: "  
 poolDictionaries: "  
 category: 'Exemple Cours'

- **Traitements réalisés par le service (nom quelconque)**

afficherBonjour

Transcript cr; show: 'Hello World'

- **Informations sur le service**

defaultName

^'Hello'

defaultDescription

^'Affiche "Hello World" sur le Transcript distant'

# *Exemple "Hello World"*

## 1c- Point d'entrée au service (2/2)

- **Gestion du client**

hasClientCodeFor: aHost

^true

fileOutClientCodeOn: stream for: aHost

HelloClient fileOutOn: stream

clientStartupReceiverNameFor: aHost

^UTRemoteGlobalObject named: #HelloClient

clientStartupSelectorFor: aHost

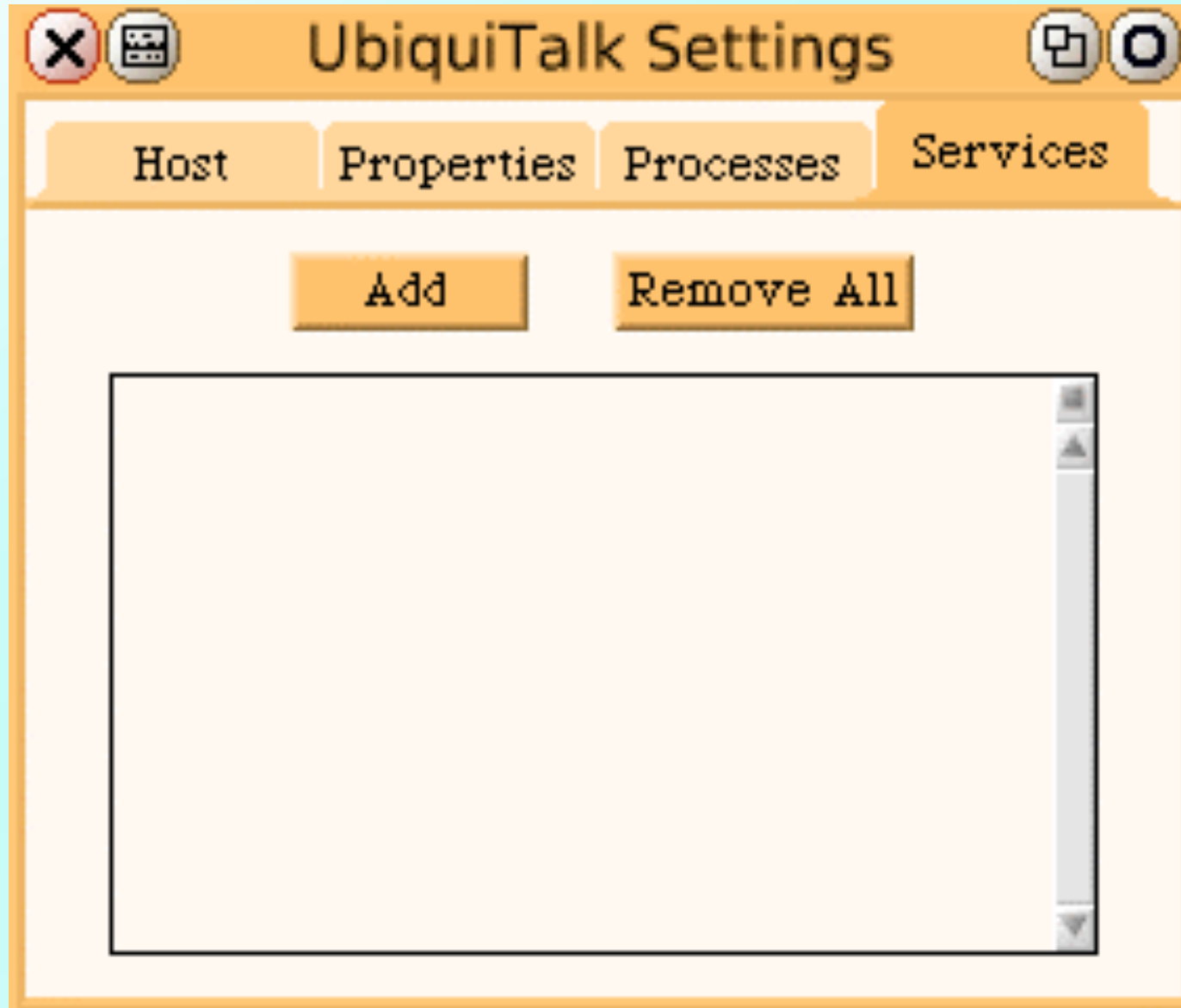
^#executer:

clientStartupArgumentsFor: aHost

^{self}

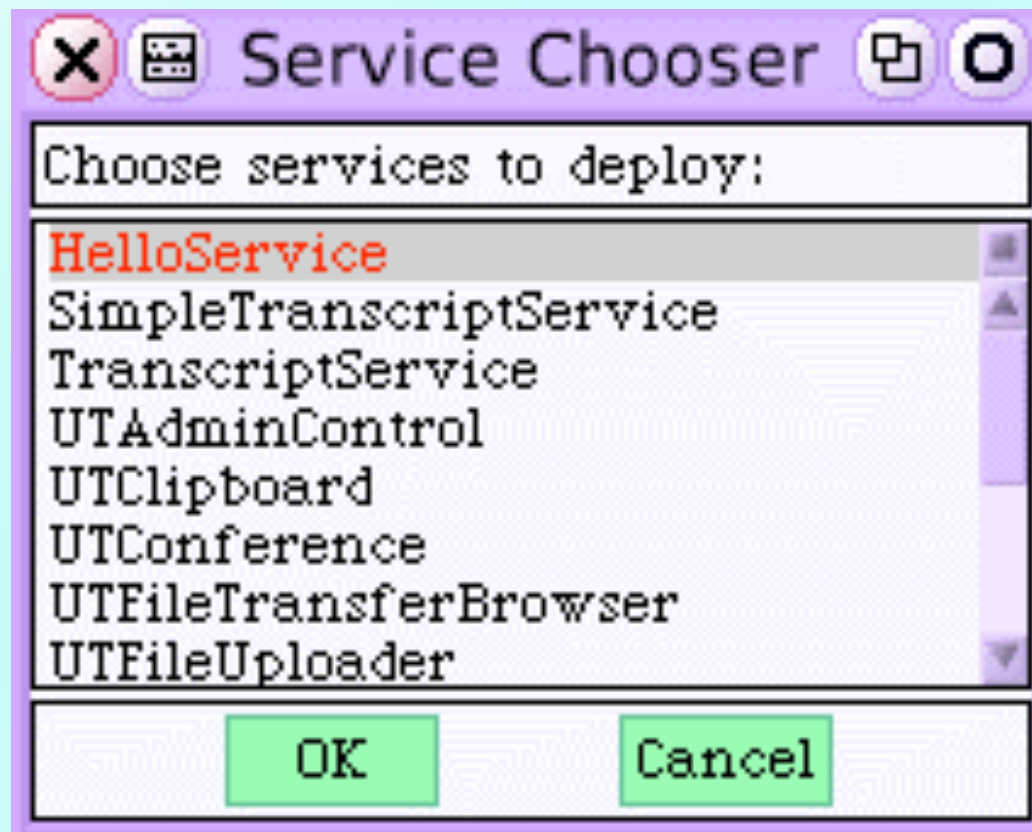
# *Exemple "Hello World"*

## 2-Déployer le service (1/3)



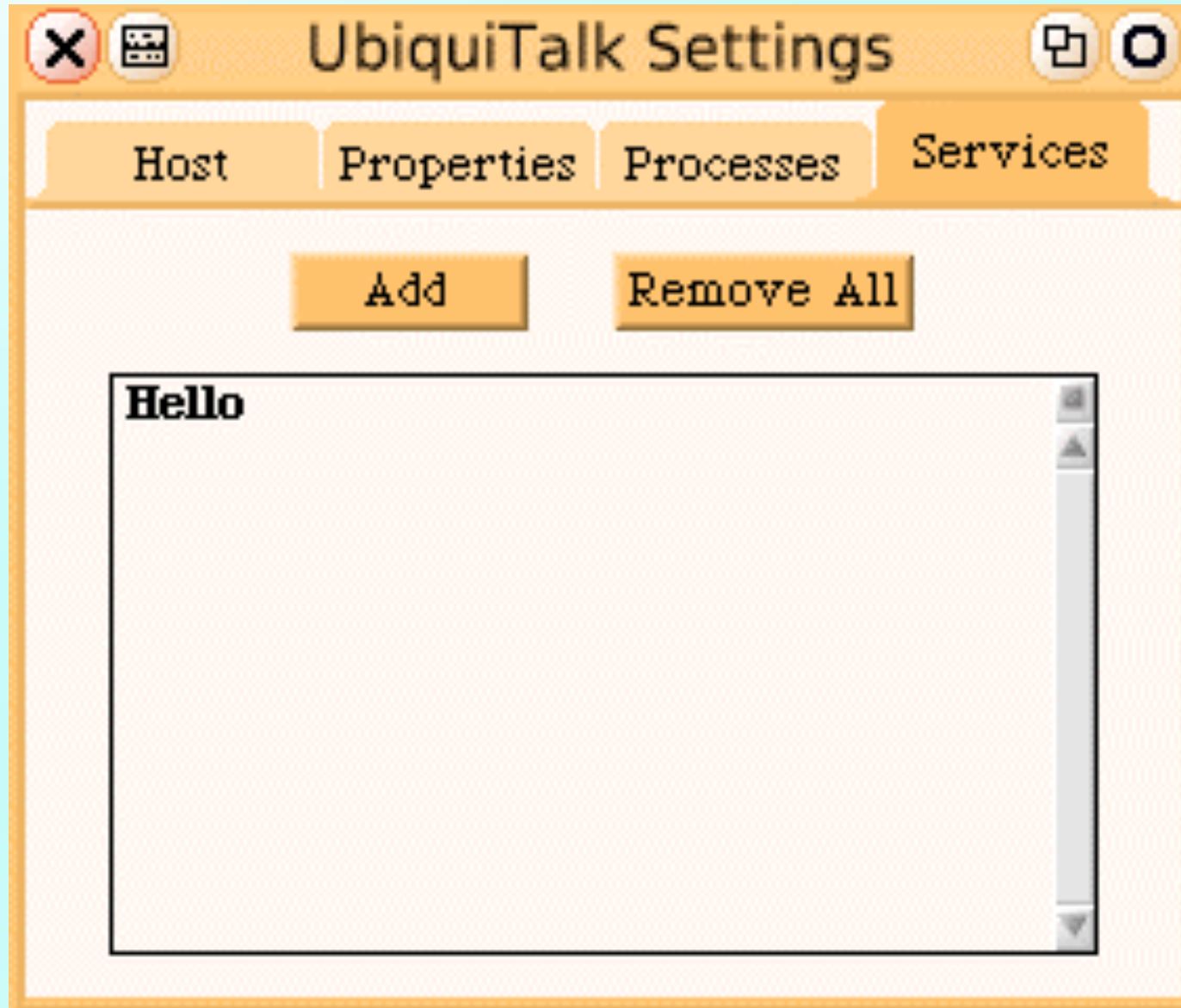
# *Exemple "Hello World"*

## 2-Déployer le service (2/3)



# *Exemple "Hello World"*

## 2-Déployer le service (3/3)



# Exemple "Hello World"

## 3- Lancer le fournisseur et l'utilisateur

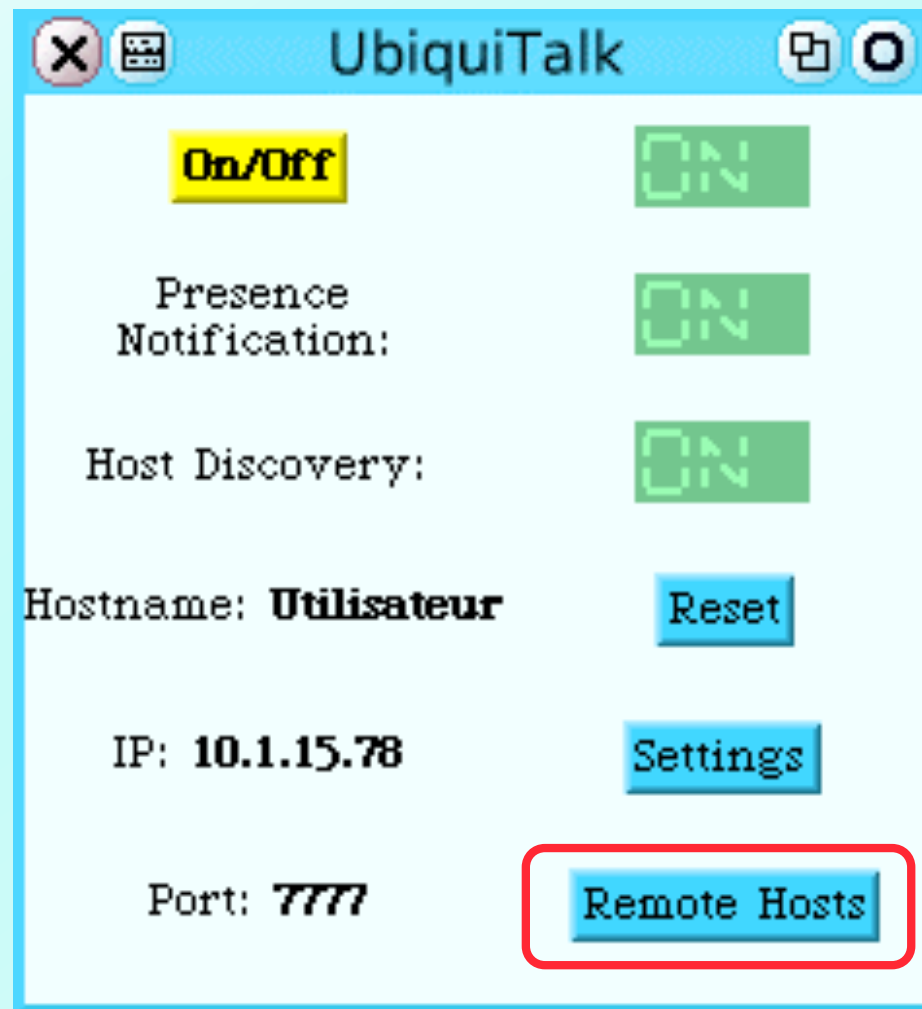
The image displays two side-by-side screenshots of the UbiquiTalk configuration interface. The left window, labeled '1', is for the 'Utilisateur' (User) and shows the following settings: 'On/Off' (highlighted in yellow), 'Presence Notification: ON', 'Host Discovery: ON', 'Hostname: Utilisateur', 'IP: 10.1.15.78', and 'Port: 7777'. The right window, labeled '2', is for the 'Fournisseur' (Provider) and shows: 'On/Off' (highlighted in yellow), 'Presence Notification: ON', 'Host Discovery: ON', 'Hostname: Fournisseur', 'IP: 10.1.15.78', and 'Port: 6666'. Both windows have 'Reset', 'Settings', and 'Remote Hosts' buttons at the bottom.

Ordre de lancement + port

importants si c'est sur la même machine

# Exemple "Hello World"

## 4- Utiliser le service (1/2)



# Exemple "Hello World"

## 4- Utiliser le service (2/2)

