Applications Mobiles

Christophe Gransart
INRETS
christophe.gransart@inrets.fr

Plan

- Introduction sur les objets mobiles
- Les principaux axes de recherche
- Client/serveur mobile
- Systèmes de découverte de l'environnement
- Exemples de middleware appliqués aux OMC
  - CORBA et .net
  - Composants CORBA
  - JMS illustré avec iBus

Introduction

Objets mobiles ?

Objets mobiles

- Objets physiques muni de
  - capacité de calcul
  - capacité de communication
- Avec des contraintes
  - très hétérogènes
  - aux faibles capacités de stockage
  - avec des processeurs et des systèmes limités
  - utilisant des communications sans fil
  - et très mobiles

Mettre le monde en réseau

- Aujourd'hui
  - Internet connecte tous les ordinateurs
- Demain
  - Les terminaux seront de plus en plus omniprésents
  - Informatiques embarquées
- Après demain ?
  - Chaque objet pourra être connecté
  - Informatique diffuse

Déconnexions fréquentes

- Blanc entre cellules (>1*ms* pour la plupart des réseaux cellulaires)
- Déconnexions liés aux problèmes d'énergie
  - changement de batterie
  - Batterie en charge
  - Déconnexion volontaire pour économiser l'énergie
  - Mode veille
- Déconnexions liés à l'environnement
  - Tunnel, environnement hostile
  - Absence de routage (Roam-off disconnections)

Bande passante limitée

- Nettement plus faible que pour les réseaux fixes
- Fort taux d'erreur (bit error rates : BER)
- Nombre de personnes dans une cellule non contrôlée (congestion)
- Presque impossible de garantir une qualité de service (Quality of Service : QoS)
- Bande passante asymétrique
  - Minitel 75 / 1200
- Bande passante limitée par la puissance nécessaire
  - Batterie (durée, puissance, …)

Limitations imposées par la mobilité

- **Limitations liées aux applications**
  - Les modèles de programmation ne sont pas adaptés aux caractéristiques des applications
    - Lien composants ou objets / services techniques
    - Bibliothèques peu sophistiquées
      - Exemple : J2ME pour les interfaces

Limitations imposées par la mobilité

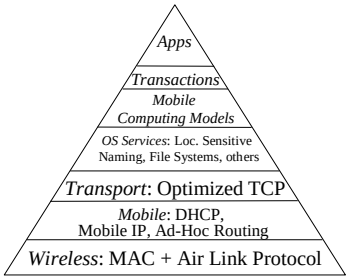
- **Limitations imposées par le système**
  - *Réseau* : les protocoles de transport actuels sont inefficaces dans le cadre de réseaux hétérogènes (partie fixe et partie sans fils)
  - *Session et présentation* : inapproprié pour les environnements nomades
    - fonctionne essentiellement par session, le client devant rester connecté
  - *Systèmes d'exploitation* : revoir complètement l'architecture
    - gestion des caches
  - *Client/serveur* :
    - redéfinir le concept pour prendre en compte les modifications de l'environnement

Limitations liés au terminal mobile

- Durée de vie de la batterie faible
  - max ~ 5 heures
- Sujet à la perte ou à la destruction (ou au vol)
  - Peu sûr
- Fortement indisponible
  - En fonctionnement normal : éteint pour préserver la batterie
- Capacité limitée
  - Écran, mémoire, entrée, processeur, espace disque, …
- Pas encore complètement intégré sur le réseau
  - Dépend d'une station

Les principaux axes de recherche

Pyramide de recherche



13

Problèmes liés au réseau

- Mobile IP
- Wireless Transport
- Ad-Hoc Networks
- Location Management
- Wireless Network Benchmarking
- Ad-Hoc Network Simulation
- Wireless Link Simulation

14

Wireless and Mobile Computing Models

- Mobility-aware Client/Server using Proxies
- Disconnected Operations
- Application-aware Adaptations
- Mobile Agents and Objects
- Thin Client/Server
- Mobile Caching and Replication
- Client/Air Computing (PUSH technology)

15

Mobile file and Database Systems

- Wireless File System Access
- Disconnected File Systems
- Mobile Access to C/S or Distributed Databases
- Ad-Hoc Database Systems
- Checkpointing
- Database recovery
- Mobile Database Design

16

Mobile Transaction and Workflow

- ACID Relaxation
- Mobile Transaction Models
- Optimistic Data Replication
- Semantic-based Conflict Resolution
- Consensus in Mobile Environment

17

Wireless and Mobile Applications and Services

- Application Design for Wireless networks
- Application Design for Mobility
- Wireless WWW Access
- Mobile Groupware
- Location-sensitive Yellow Service

18

Performance and QoS

- QoS Measures in Wireless and Mobile Environments
- QoS Guarantees
- Simulators and Emulators of Wireless Links
- Simulators of Mobile and Ad-hoc Networks
- Wireless Networking Benchmarking

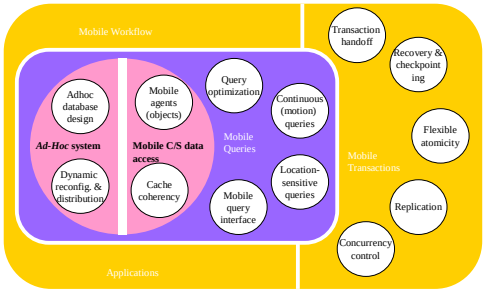
19

Emerging Standards

- The BlueTooth Standard
- The Wireless Application Protocol (WAP)
- The W3C effort in CompactHTML
- The Network Computer Reference Specification
- Telecom Standards: UMTS
- IEEE Wi-Fi

20

Ongoing or Needed research

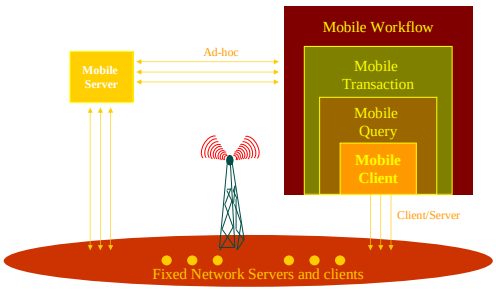


21

Client/serveur mobile

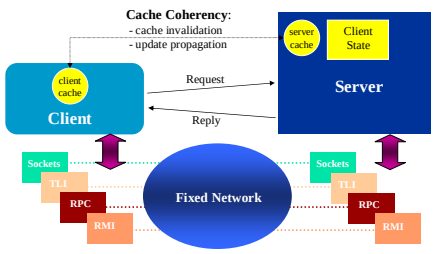
22

Hierarchy of Computing Models



23

Client/Server Computing



24

Client/Server Design

- Stateless/stateful client/server design
- Caching and cache invalidation
  - server invalidates client cache *and/or*
  - client requests server to validate its cache.
  - file system caching: writes => update propagation
- Connectionless/connection-oriented design
- TCP/IP & Interfaces
- Other issues: multi-threading &deadlocks

25

Fixed Network C/S Assumptions

- Client Connectivity
  - client is always connected with availability comparable to the server's. Server can always invalidate the client cache
- Server Availability & Reliability
  - server is highly available. Reliable if stateless (but state info is exchanged in every C/S interaction), or if implements recovery procedures (may require client availability)
- Network
  - fast*, reliable*, BER < 10^{-6}, bounded delay variance

26

Taxonomy of C/S Adaptations

- **System-transparent, application-transparent**
  - The conventional, "*unaware*" client/server model
- **System-aware, application-transparent**
  - the client/proxy/server model
  - the disconnected operation model
- **System-transparent, application-aware**
  - dynamic client/server model
  - the mobile agent (object) model
- **System-aware, application-aware**

27

The *Unaware* Client/Server Model

- Full client on mobile host and full server on fixed network (SLIP/PPP C/S)
- Client and Server are not mobility-aware
- Client caching does not work as the client can be disconnected when the server invalidates the cache
- Not reliable and of unpredictable performance
- *Requires special cache invalidation algorithms to enable caching despite long client disconnections*

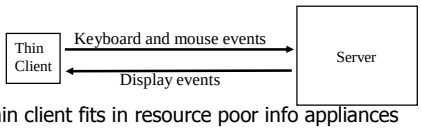
28

The Client/Proxy/Server Model

- Adding mobility-awareness between the client and the server. Client and server are not mobility-aware.
- Proxy functions as a client to the fixed network server, and as a mobility-aware server to the mobile client
- Proxy may be placed in the mobile host (Coda's Venus), or the fixed network, or both (WebExpress)
- Application- and user-dependent
- One advantage: enables *thin client* design for resource-poor mobile computers

29

Thin Client/Server Model



- Thin client fits in resource poor info appliances
- Bounded communication
- Requires at least weak connection
- CITRIX WinFrame and ICA thin client
- InfoPad

30

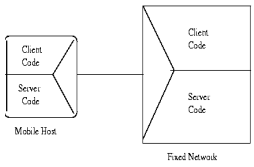
The Disconnected Operation Model

- **Approach I:**
  - Provide full client and a thin version of the server on the mobile platform. In addition, needed data is replicated into the mobile platform. Upon reconnection, updated replicas are synchronized with the home server. Conflict resolution strategies are needed (Coda/Venus & Oracle Lite)
- **Approach II:**
  - Provide a full client and a mobility agent that intercepts requests to the unreachable server, emulates the server, buffers the requests, and transmit them upon reconnection (Oracle Mobile Agents)

31

The Dynamic Client/Server Model

- Servers (or their thin versions) dynamically relocate between mobile and fixed hosts. Proxies created and relocated dynamically
- A spectrum of design and adaptation possibilities
- Dynamic availability/performance tuning



32

Dynamic Client/Server Model

- **Mobile objects:**
  - applications programmed with dynamic object relocation policies for adaptation (Rover's RDOs)
- **Collaborative Groups:**
  - disconnected mobile clients turns into a group of collaborating, mobile servers and clients connected via an ad-hoc net. (Bayou architecture)
- **Virtual Mobility of Servers:**
  - servers relocate in the fixed network, near the mobile host, transparently, as the latter moves.

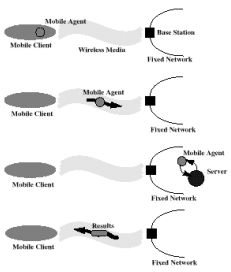
33

The Mobile Agent Model

- Mobile agent programmed with platform limitations and user profile receives a request; moves into the fixed network near the requested service
- Mobile agent acts as a client to the server, or invokes a client to the server
- Based on the nature of the results, *experienced* communication delays, and programmed knowledge, the mobile agent performs transformations and filtering.
- Mobile agent returns back to mobile platform, when the client is connected.

34

Mobile Agents in the Mobile Environment



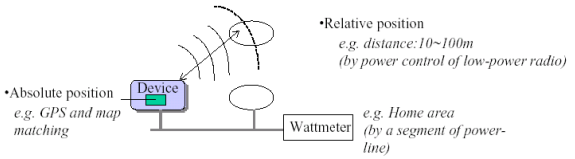
35

Systèmes de découverte de l'environnement

36

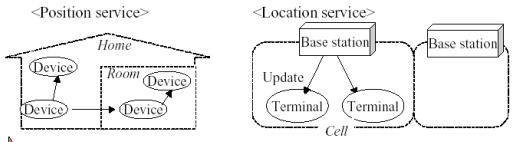
Service d'information sur l'environnement

- Position service
  - Absolute position service
  - Neighborhood service around each device
    - e.g. room/floor/building, line/process/factory, etc.
    - e.g. by using low-power radio, power-line
    - All devices can not know their positions by themselves



Service d'information sur l'environnement

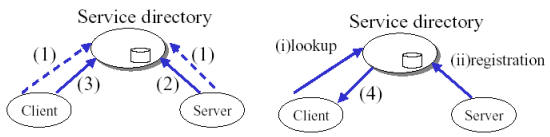
- Relationship to other activity in OMG
  - Location service ("Wireless CORBA Whitepaper") Telecom TF
    - Location of terminals in cellular structure
    - e.g. longitude, latitude, etc.



Collaborative development of precise(local) position and relative area service
- All devices are base station and terminal -

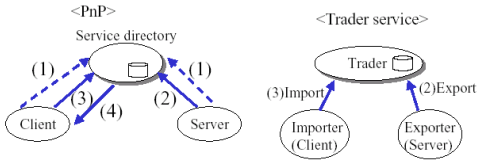
Plug and Play autonome

- Functions of existing PnP
  - (1) Discovery of service directory
  - (2) Service registration
  - (3) Service lookup
  - (4) Service notification



Plug and Play autonome

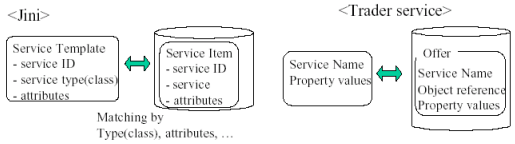
- Relationship to existing OMG specifications
  - Trader service: service export and import



Extend to Active Trader
- Discovery in unknown environment when clients/servers join and/or leaves
- Notification of service availability

Plug and Play autonome

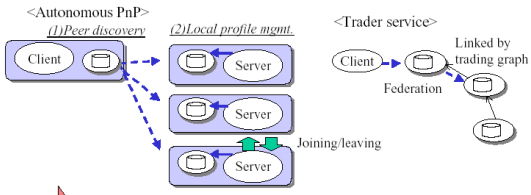
- Service matching in (3)service lookup
  - e.g.) Jini: service ID, type, etc. and/or a combination of them
- Relationship to existing OMG specifications
  - Trader service: matching by service type



Extend to support flexible matching

Plug and Play autonome

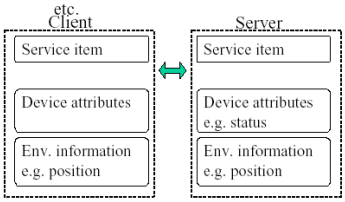
- Functions for autonomous PnP
  - (1) Peer-to-peer discovery
    - From multiple service directory joining/leaving the system
  - (2) Local registration
    - to self-managed service directory (profile)



Extend to Decentralized Trader with (1),(2)

Plug and Play autonome

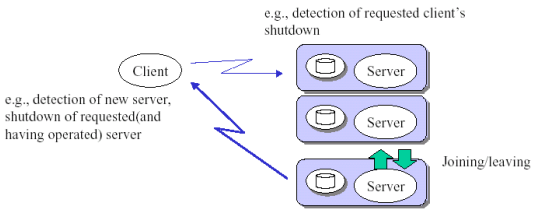
- Functions for autonomous PnP (cont'd)
  - Service matching in (3)service lookup
    - Matching with dynamically changing attributes
      - Device attributes: working or idle device, etc.
      - Environment information: nearest one, those in a room, etc.



43

Plug and Play autonome

- Functions for autonomous PnP (cont'd)
  - (4) Continuous notification of service and client status
    - service availability, out of use, and request stop, etc.



Develop continuous status notification service e.g., Heartbeat and timeout detection

44

Exemples

- Jini
- Salutation
- Havi
- Upnp
- OSGI
- Trader CORBA

- Bref, beaucoup de systèmes hétérogènes …
  - Chacun a son système de communication
  - Chacun a sa propre représentation des données

45

Exemples de middleware appliqués aux OMC

CORBA et .net
Composants CORBA
JMS illustré avec iBus

46

CORBA sur PDA

Illustration au travers d'une application de diffusion d'informations

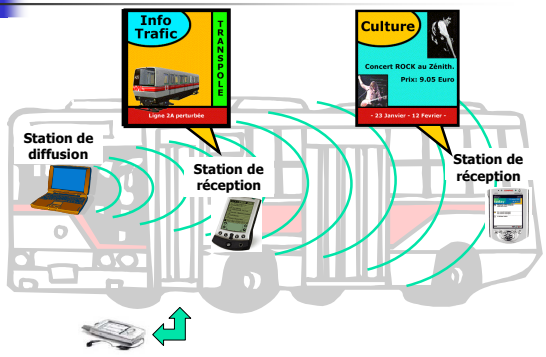
47

Système de diffusion d'informations

- Système de diffusion d'informations
  - Ensemble de PDA communicants à l'aide d'un réseau sans fil
- Apporter public des informations contextualisées
  - déroulement du trajet (prochain arrêt, correspondance, information multimodale, tarif, …)
  - environnement traversé (lieux touristiques, expositions, …)
  - actualités, divertissements

48

Contenu du prototype

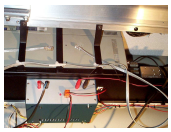


49

Architecture matérielle

50

Architecture matérielle



PC industriel
Récepteur GPS
Réseau 802.11b
PDA + 802.11b
Écrans plats

51

Architecture logicielle

Application CORBA multicast

Base de données
Station de diffusion
Acquisition GPS
PDA de réception
Réseau sans fil (IEEE 802.11b – Wi-Fi)

52

Architecture logicielle du PDA

User application	User application
ORB + Plug-in Multicast	Common Language Runtime
JVM J9 + library	.net + library
Operating System	Operating System
802.11b Network	802.11b Network
Hardware	Hardware
Client Java/CORBA	Client C#/.net

53

OpenCCM sur PDA

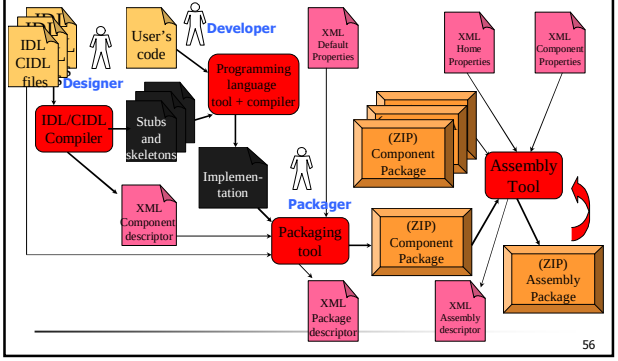
54

Le modèle de composants CORBA

- The CORBA Component Model (CCM) is the Next Best Thing ;-)
  - 1st open standard for Distributed Component Computing
  - Composants « métiers » hétérogènes répartis
  - Multi-languages, multi-OSs, multi-ORBs, multi-providers, ...
- 5 sous-modèles (artefacts + techniques)
  - abstrait : *caractériser - concepteurs*
    - extension OMG IDL3 pour exprimer ports
  - programmation : *produire - développeurs*
    - *Component Implementation Definition Language (CIDL)*
  - packaging : *diffuser* et *assembler - architectes*
    - *Open Software Description (XML)*
  - déploiement : *instancier - administrateurs*
    - API OMG IDL2
  - exécution : *exploiter - fournisseurs*
    - containers « à la » EJB

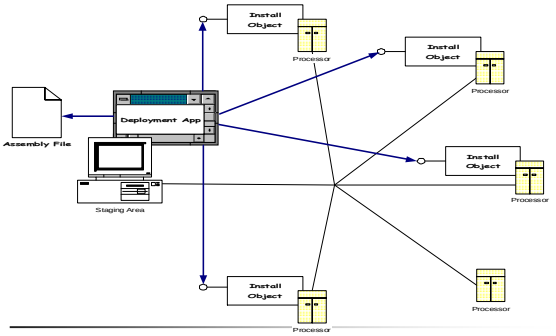
55

Le processus logiciel du CCM : de l'analyse au packaging



56

Le modèle de déploiement : le déploiement des composants



57

L'état actuel de la plate-forme OpenCCM

- Une chaîne de production ouverte
  - un référentiel OMG IDL3
  - un compilateur OMG IDL3
  - un générateur équivalent OMG IDL2
  - un générateur de squelettes Java étendus
- Un environnement de déploiement flexible
  - un serveur Java générique d'accueil des composants
  - des API pour contrôler le déploiement
    - téléchargement à distance dans les serveurs
  - déploiement piloté via des scripts OMG IDLscript

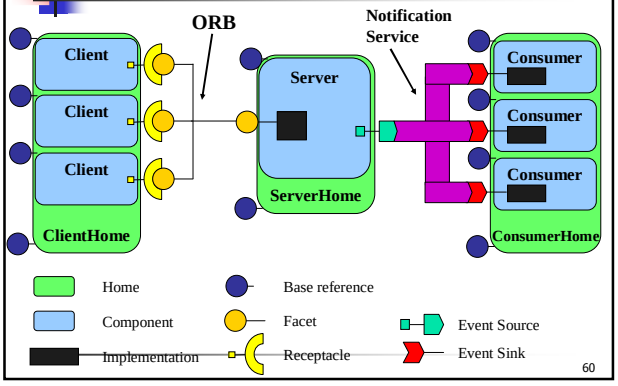
58

Quelques informations techniques

- Entièrement écrit en Java
  - portabilité, maintenance et support
  - Linux, Solaris, Windows, WindowsCE
  - ORBacus 4, OpenORB 1.0, VisiBroker 4
- Compilateur OMG IDL3
  - parser JavaCC : orienté objet et proche de la BNF
  - couche adaptateur : simplifie règles BNF et masque API référentiel
- Référentiel OMG IDL3
  - API propriétaire en OMG IDL2
  - 1ères validations du futur OMG IR3 en cours de révision
- Générateurs
  - pas de nouveaux « mapping » langages de programmation !
    - OMG IDL3 -> OMG IDL2
  - squelettes étendus implantent la connectique
  - réutilisation générateurs souches/squelettes GIOP/IIOP

59

Une application distribuée avec CCM



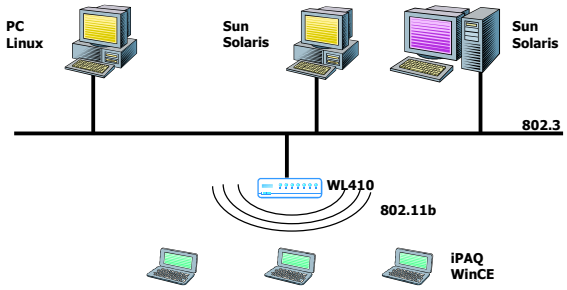
60

Notre plate-forme de test

- Matériel
  - Compaq iPAQ 3660 (StrongArm 206MHz, 64 Mo RAM)
  - Slot PCMCIA
  - Carte 802.11b format PCMCIA (WL110)
  - Pont 802.3/802.11b (WL410)
  - PC, Sun
- Logiciel
  - JVM IBM J9 (sur PDA)
  - JVM JDK 1.3 (sur PC, Sun)
  - ORBacus 4.1 + Naming + Trader + OpenCCM runtime

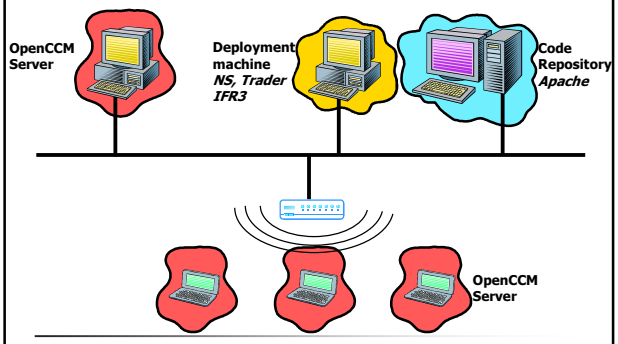
61

Architecture de test



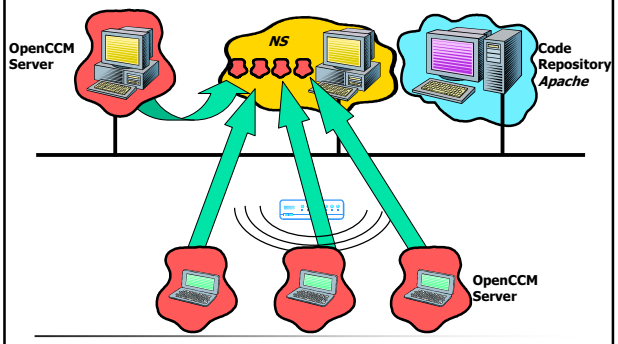
62

Architecture de test



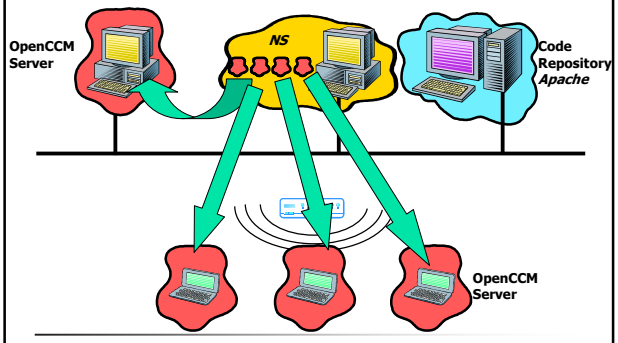
63

Enregistrement des serveurs



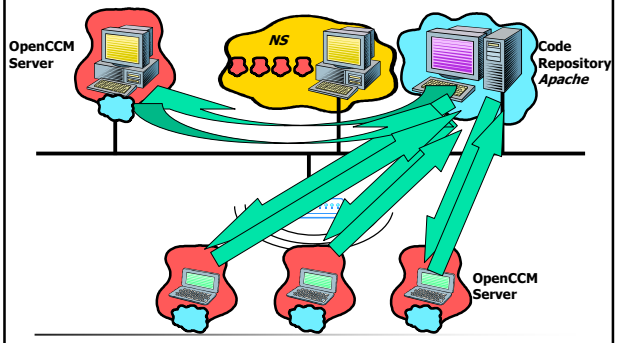
64

Ordre de déploiement



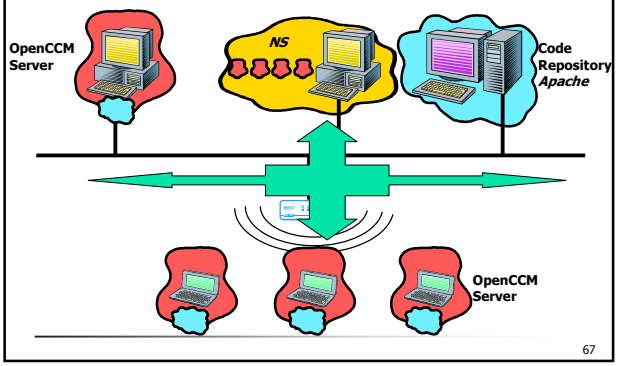
65

Chargement de code



66

Interconnexion et exécution



Premières évaluations

- Les composants CCM déployés sur les PDA sont les mêmes que ceux sur PC ou Sun
  - à l'interface graphique près (support AWT sur J9)

- Que 6 Mo à l'exécution sur les PDAs ...
  - JVM + ORB + NS + Trader + OpenCCM

Les problèmes à résoudre

- Connectivité avec l'infrastructure
  - mobilité limitée si déploiement entre réseau filaire/non filaire
- Changement de la philosophie d'utilisation des composants
  - « classique » un ensemble de serveurs qui accueillent des composants
  - PDA = objet personnel
  - laisser le choix d'installation des composants à l'utilisateur
    - fenêtre de confirmation, certification des composants, ...

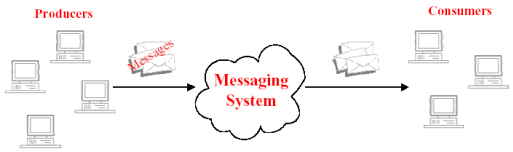
Système de messagerie asynchrone

Qu'est-ce que JMS?

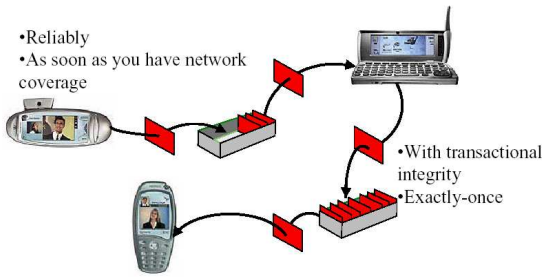
- JMS : "Java Message Service«
- Sun's Definition: *JMS is an API for accessing enterprise messaging systems from Java programs.*
- JMS is for *messaging systems* what JDBC is for *database systems*: A standardized API to access them.

What are Messaging Systems?

L'équivalent du mail pour les applications



E-mail for applications



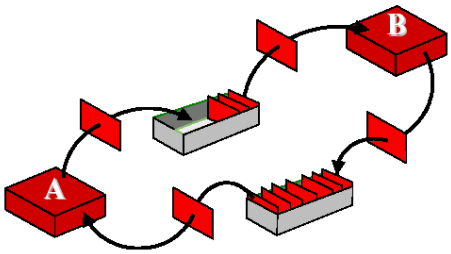
73

Pré-requis

- Device Support
  - "Always-on" capability, "connected" stand-by mode (like a phone being able to receive SMS any time)
  - Support for packet-oriented bearers
- Network
  - Packet-oriented bearer (GPRS, UMTS), Roaming support for worldwide operation
- Middleware
  - Message-oriented, store-and-forward, scalable

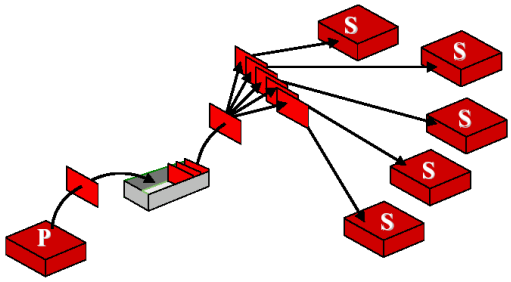
74

JMS point-to-point



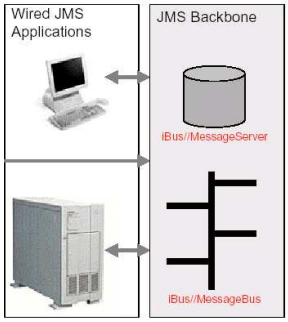
75

JMS Publish-Subscribe



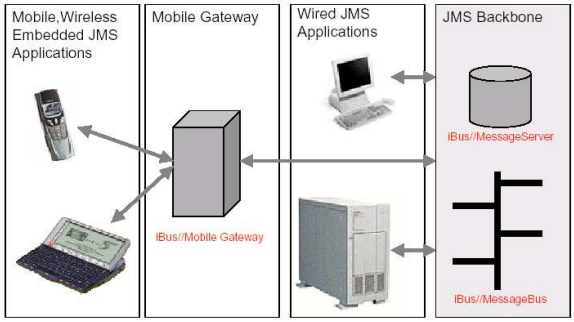
76

Exemple : iBus (Softwired)



77

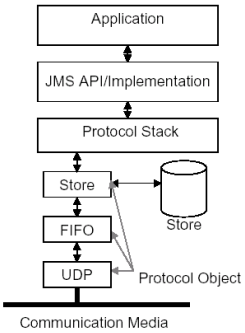
Exemple : iBus (Softwired)



78

Architecture côté client

- **Utilisation de l'API standard JMS**
- **Pile de protocole configurable**
- **Stockage en cas d'interruption du lien réseau**
- **Définition du protocole et de la QoS sous forme d'objets**



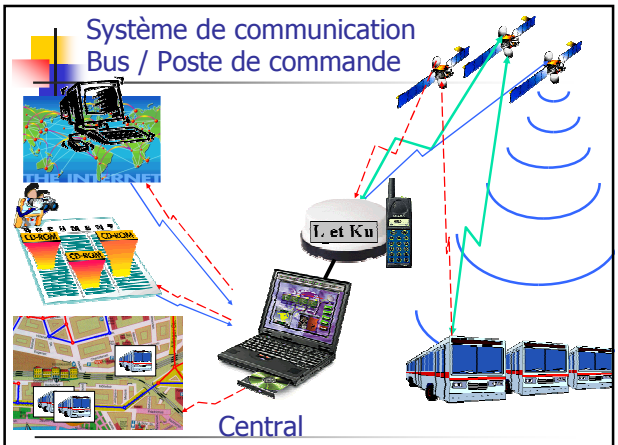
79

JMS en environnement mobile

Illustration sur une application dans les transports en commun

80

Système de communication Bus / Poste de commande



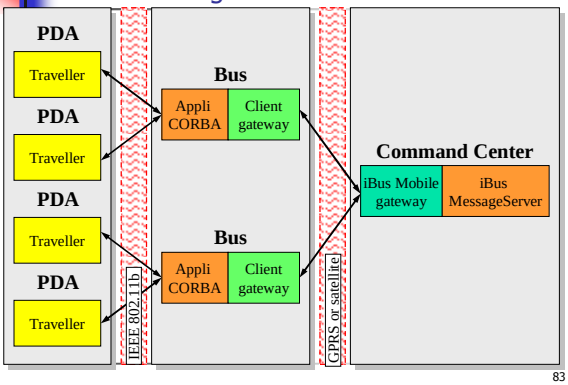
81

Système de communication Bus / Poste de commande

- Communication basée sur un réseau GPRS ou satellite (bande L et Ku)
  - Pb de connectivité (déconnexions intempestives)
  - Pb de couverture satellite en milieu urbain
- Etude d'un système de message asynchrone
  - basé sur le standard JMS (Java Messaging Service)
  - interopérabilité JMS / CORBA

82

Architecture générale



83

Retour d'expérience

- Ca marche en labo ! (réseau Wi-Fi)
  - Gestion des connexions/déconnexions correctement
- Sur GPRS
  - Débit de 1,5 Kbit/s
  - Problème de pile de protocoles dans la passerelle filaire/sans fil
- Avec satellite bi-directionnel (stack IP)
  - Débit de 30 Kbit/s
  - A tester en situation de mobilité …

84

Conclusion

Conclusion

- Les besoins/recherches sont à deux niveaux
  - Infrastructure de découverte
    - Quand découvrir, que découvrir (-> ontologies)
    - Heartbeat / positionnement
  - Infrastructure de communication
    - Communication P2P et anonyme
    - Synchrone / asynchrone
    - Connecté / déconnecté
- Encore beaucoup de travail en perspective …