

# Programmation d'applications pour terminaux mobiles

## Chapitre 1 - Introduction

### Master 2 SIMI

Sébastien Choplin  
IUP MIAGe Amiens

2005

# Introduction

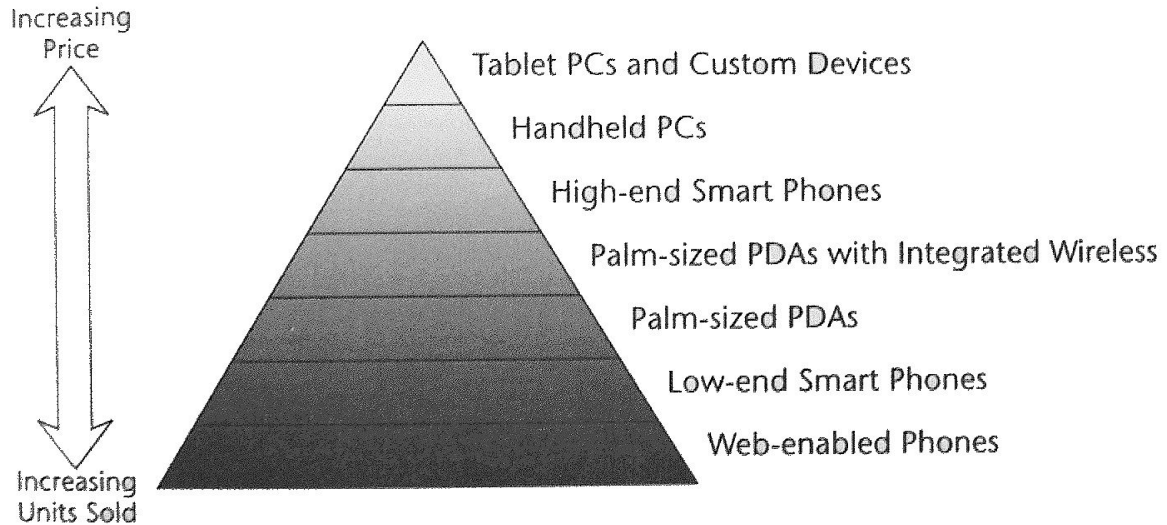
But du cours : découvrir la programmation sur/pour terminaux mobiles (PDAs, téléphones).

Ressource: *Mobile and Wireless Design Essentials* -  
Martyn Mallick - Edition WILEY

# Plan

- Chapitre 1 - Introduction
- Chapitre 2 - Développement "Smart Client"
- Chapitre 3 - Persistance des données
- Chapitre 4 - Développement "Thin Client"

# Classification des terminaux



**Figure 2.2** Device classifications.

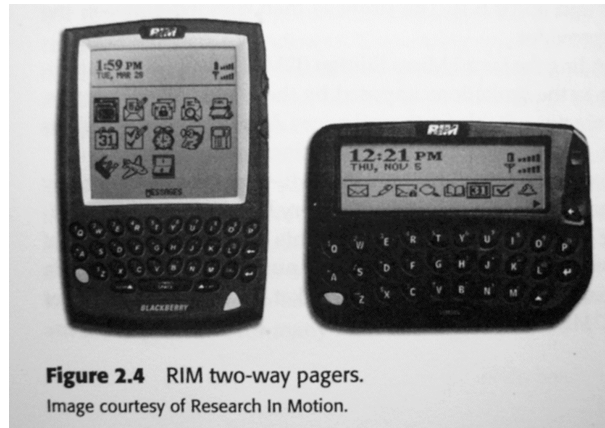
## Classification des terminaux: Téléphones “Web-enabled”

- WAP,
- IHM pauvre,
- haute disponibilité,
- capacité de calcul limitée  
⇒ durée de batterie importante



## Classification des terminaux: Agendas (Pagers)

- suite logicielle non évolutive
- système propriétaire
- “vrai” clavier
- ⇒ convergence vers PDA



**Figure 2.4** RIM two-way pagers.  
Image courtesy of Research In Motion.

## Classification des terminaux: Smart Phones

- idem Web-enabled
- exécution d'applications locales (J2ME)

## Classification des terminaux: PDAs

PDA=Personnal Digital Assistant

- entre le téléphone et le tablet-PC
- démarrage instantané
- écran tactile VGA (320x240)
- de + en + performant (>500MHz)
- consommateur d'énergie (surtout si wireless)
- convergence avec les téléphones (High-end Smart Phones)



**Figure 2.5** Palm i705 with Integrated Wireless.  
Image courtesy of Palm Inc.

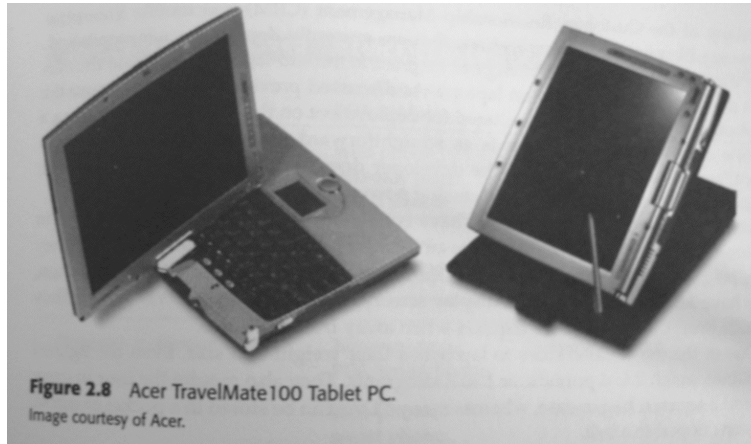
## Classification des terminaux: Handheld PCs



Figure 2.7 Samsung NEXiO Handheld PC.

- + gros mais + performant que les PDAs
- parfois avec un clavier
- démarrage instantané
- rattrapé par les PDAs

## Classification des terminaux: Tablets PCs



- mélange de portable et PDAs
- écran tactile
- utilisation spécifique justifiant le surcoût

## Classification des terminaux: Portables

- capacité de calcul importante
- durée de batterie faible ( $\sim \leq 4H$ )
- encombrement, poids et démarrage

## Classification des terminaux: conclusion

- Convergence vers les PDAs,
- problématiques pour les applications “mobiles” sur TabletPc et Portables similaires
- ...

⇒ nous nous concentrerons sur les PDAs et Smart Phones.

## Architectures et OS

Périphériques hétérogènes  $\Rightarrow$  architectures hétérogènes  $\Rightarrow$  OS hétérogène

Arch. : ARM, MIPS, PowerPC, SHx, x86, X-scale, ...

OS: Windows CE, Palm OS, Symbian OS, Linux, Java ...

Convergence des OS propriétaires vers ceux ci-dessus.

# Windows CE

**Table 7.1** Comparison of Windows CE Versions

<b>Version/ Features</b>	<b>Year of Release</b>	<b>Processors Supported</b>	<b>Internet Browser</b>	<b>Wireless Capabilities</b>	<b>Multimedia Support</b>
CE 2.12	1998	ARM, MIPS, PowerPC, SHx, x86	Based on Internet Explorer 4.0	IrDA	No (audio only)
CE 3.0	2000	ARM, MIPS, PowerPC, SHx, x86	Based on Internet Explorer 4.0	IrDA	Yes
CE .NET	2002	X-Scale, ARM,	Based on Internet Explorer 5.5	Bluetooth, 802.11x, Media Sense, OBEX	Yes

Aujourd'hui: Windows Pocket PC 2003 2nd edition

## Caractéristiques:

- Multi-tâche,
- temps-réel,
- Suite bureautique (PocketWord, PocketExcel, PocketOutlook, PocketIE, ...) ⇒ facilité pour les usagers de Windows
- API de dev. réduction de l'API Windows ⇒ facilité de dev.

Remarque: PocketPC  $\neq$  Windows CE

## Palm OS

Leader du marché pendant de nombreuses années (75% du marché en 2000).

### Caractéristiques

- à peu près les mêmes que WPocketPc (sauf suite bureautique), avec un peu de retard..
- bcp d'applications existantes (car + ancien)

## Symbian OS

OS pour les téléphones (aussi PDA), créé par Nokia, Ericsson, Motorola, Matshushita, Psion. (Basé sur EPOC de Psion)

Objectifs: intégrer toutes les fonctionnalités réseau dans les téléphones et PDAs

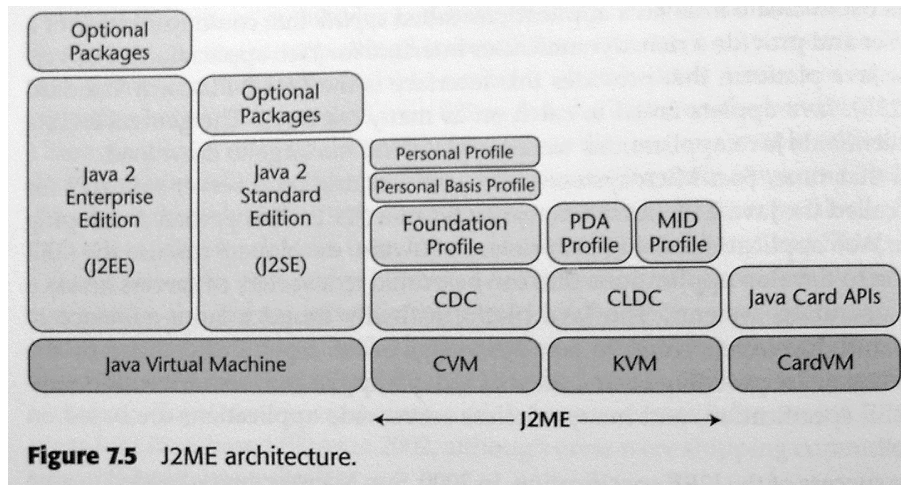
## Linux

Au départ plus du “hacking”

à partir de 2002: Sharp Zaurus, Yopi

## Java/J2ME

Java était prévu pour les périphériques embarqués...  
En 2000, retour de Java avec J2ME, une version 'light'



Même principe de portabilité que J2SE et J2EE.

## Java/J2ME et OS

Tous les OS précédent supporte JAVA avec une JVM

OS uniquement Java: Research In Motion (C++ puis Java)

## Particularités de J2ME

Notions de “configuration” et “profile”

configuration: JVM, bibliothèques de base, classes standards, et APIs.

Il existe 2 configurations:

- CLDC (Connected Limited Device Configuration)
- CDC (Connected Device Configuration)

profile: au-dessus d’une configuration, fourni une API complète pour un type de périphérique

## J2ME: configuration CLDC

Machine virtuelle: KVM

Prévu pour les terminaux ayant peu de ressources (CPU 16 ou 32-bits, <512KB de mémoire par application).

Profile: MIDP(Mobile Information Device Profile) et PDA.

## J2ME: configuration CDC

Machine virtuelle: CVM

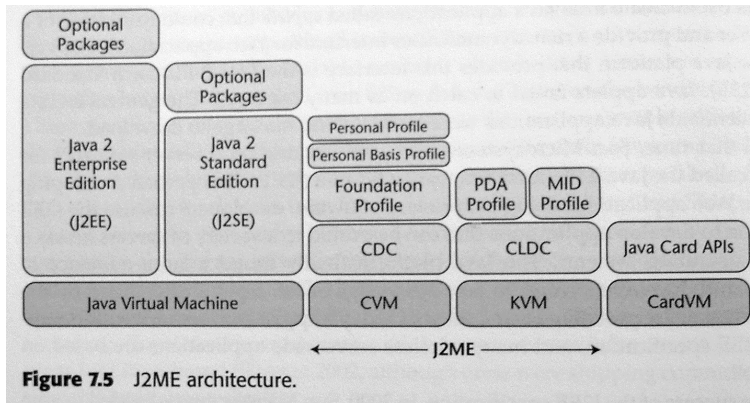
Prévues pour

- CPU  $\geq$  32bits
- 2Mo de mémoire par application

profile: "Foundation Profile" < "Personnal Base Profile"  
< "Personnal Profile"

"Personnal Profile"  $\sim$  J2SE

# J2ME: point



Points forts:

- portabilité
- réutilisation du code Java

Points faibles:

- performances

## Marché en 2004

[http://www4.gartner.com/press\\_releases/asset\\_113913\\_11.html](http://www4.gartner.com/press_releases/asset_113913_11.html)

Estimation des ventes du 3eme trimestre 2004 d'OS pour PDAs :

Company	3Q04 Shipments	3Q04 Market Share	3Q03 Shipments	3Q03 Market Share	Growth
Windows CE	1,375,866	48.1 %	1,037,874	41.2 %	2.6 %
Palm OS	850,821	29.8 %	1,181,202	46.9 %	28.0 %
Research In Motion	565,000	19.8 %	123,775	4.9 %	56.5 %
Linux	24,500	0.9 %	48,550	1.9 %	49.5 %
Others	42,620	1.5 %	125,745	5.0 %	66.1 %
Total	2,858,807	100.0 %	2,517,146	100.0 %	3.6 %

## “Thin” client (client léger)

Architecture de type web (WAP):

- navigateur léger sur le terminal (microbrowser)
- serveur web capable de diffuser du contenu adapté

Avantages:

- déploiement facile
- modèle du web,
- interface familière aux utilisateurs
- intégration dans le SI existant facile
- données toujours à jour

Inconvénients:

- Connectivité nécessaire
- Interface limitée
- Performance en cas de congestion
- Difficile de tester sur tous les navigateurs
- Disponibilité en cas de panne du serveur
- Sécurité
- Coût (dépend de l'environnement)

## “Thin” client: exemples

- Transaction boursières
- Services d’information
- Jeux
- *m*–commerce
- ...

## “Smart” client

Smart: le terminal est en mesure d'exécuter une application particulière de manière autonome et a des fonctionnalités avancées

⇒ nécessité d'un mécanisme de stockage des données et de synchronisation

Avantages:

- données toujours dispo.
- interface riche
- performances
- calculs répartis
- sécurité
- coût (pas de connexion permanente)

Inconvénients:

- intégration + compliquée
- déploiement
- virus
- développement + compliqué
- hétérogénéité des terminaux

## “Smart” client: exemples

- VRP
- Médecin avec dossier patient
- collecte de données
- ...

## Conclusion

Le choix du type d'application est primordial, il doit être le résultat d'une analyse poussée du SI...





