

MultiMédia Mobile

Rémi Cozot

cozot@irisa.fr

Plan du Cours

1. Introduction : le marché du multimédia sur mobile
 - Offre multimédia sur mobiles
 - évolution du marché multimédia mobile
 - Les acteurs : Studio, éditeur, agrégateur, opérateur, constructeur, plate-forme, logiciel
 - Chaîne de valeur et relation entre acteurs : Accès à la distribution, au client final et facturation
2. L'environnement technique
 - Architecture d'un téléphone
 - Embarqué et/ou téléchargement
 - Les solutions technologiques : JAVA, DOJA, Brew, In-fusio, Symbian, iphone, android, ...
3. JAVA
 - Sun et « JAVA mobile »
 - CDLC 1.1
 - MIDP 2.0
 - JSRs
 - JSRs 3D : JSR 184, JSR 239
 - JSR Multimédia : JSR 135, JSR 235
 - JSR de communication : SMS, MMS
 - Autres JSRs : Géo localisation, Personnalisation
4. Contraintes à prendre en compte
 - Diversité des matériels
 - Gestion de la diversité

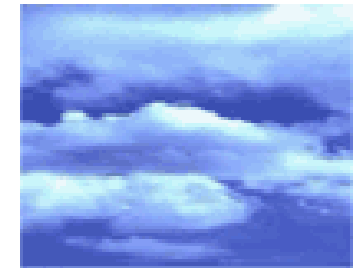
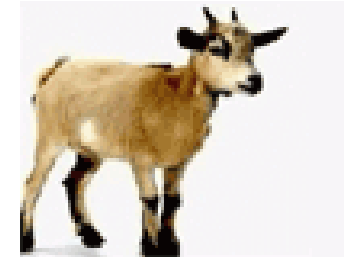
Introduction

Le marché du multimédia sur mobile

Partie 1

Offre multimédia sur mobiles

- Produits
 - Images
 - Logos, logos animés
 - Images
 - Son & Musique
 - Bruitages
 - Sonneries
 - Musique
 - Vidéo
 - Film à la demande
 - Télévision



Offre multimédia sur mobiles

Télévision

- Télévision sur 3G
 - Technologie
 - Fonctionnement identique à télévision sur ADSL
 - Offre
 - Télévision en direct
 - Vidéo à la demande
 - Caractéristique
 - L'opérateur à la main
 - Modèle économique
 - Abonnement
 - Paiement à l'acte
- Télévision Numérique Terrestre Mobile
 - Technologie
 - Fonctionnement identique à télévision numérique terrestre
 - Offre
 - Télévision en direct
 - Caractéristique
 - L'opérateur « perd » la main
 - Caractéristique du téléphone
 - Modèle économique
 - ?



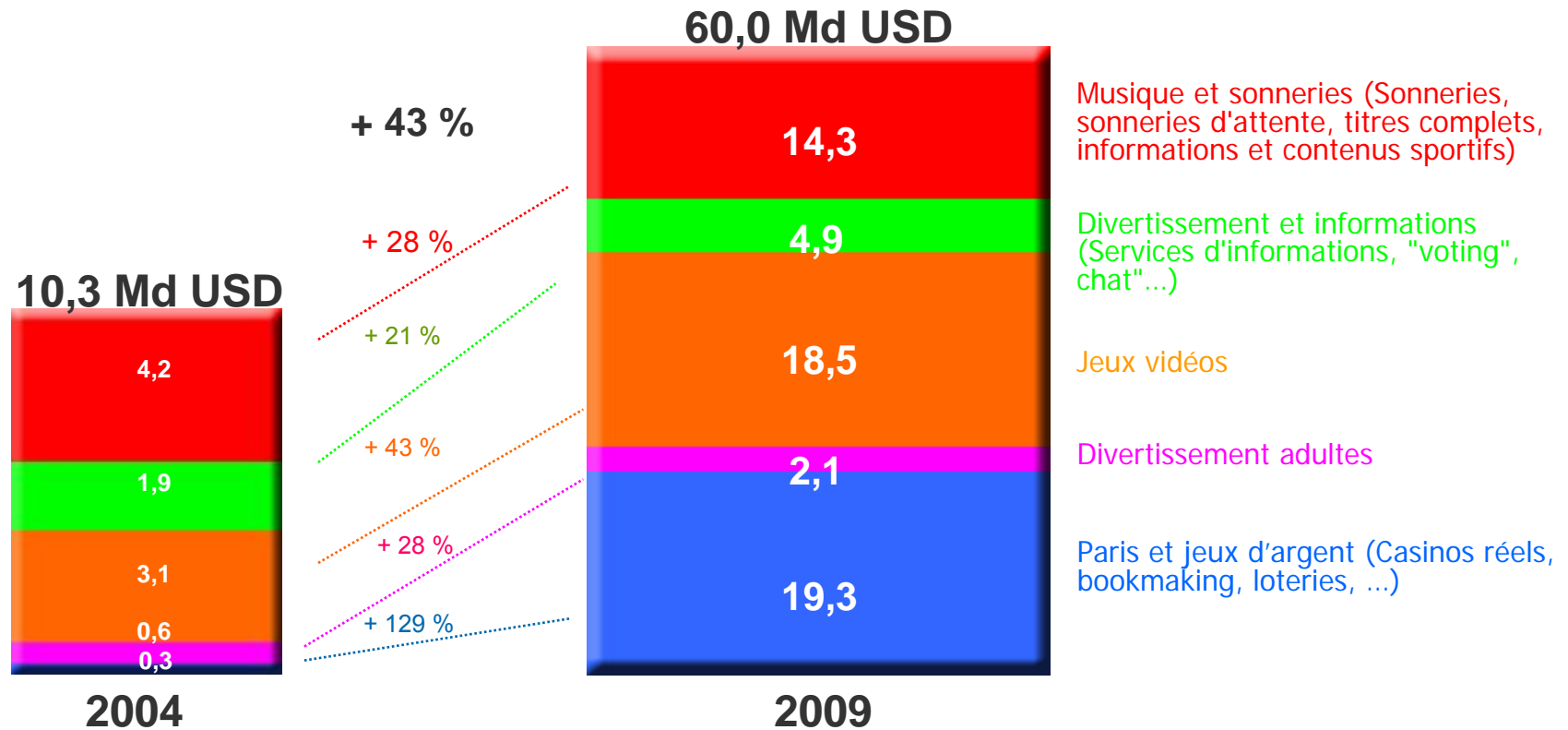
Offre multimédia sur mobiles

Télévision

7

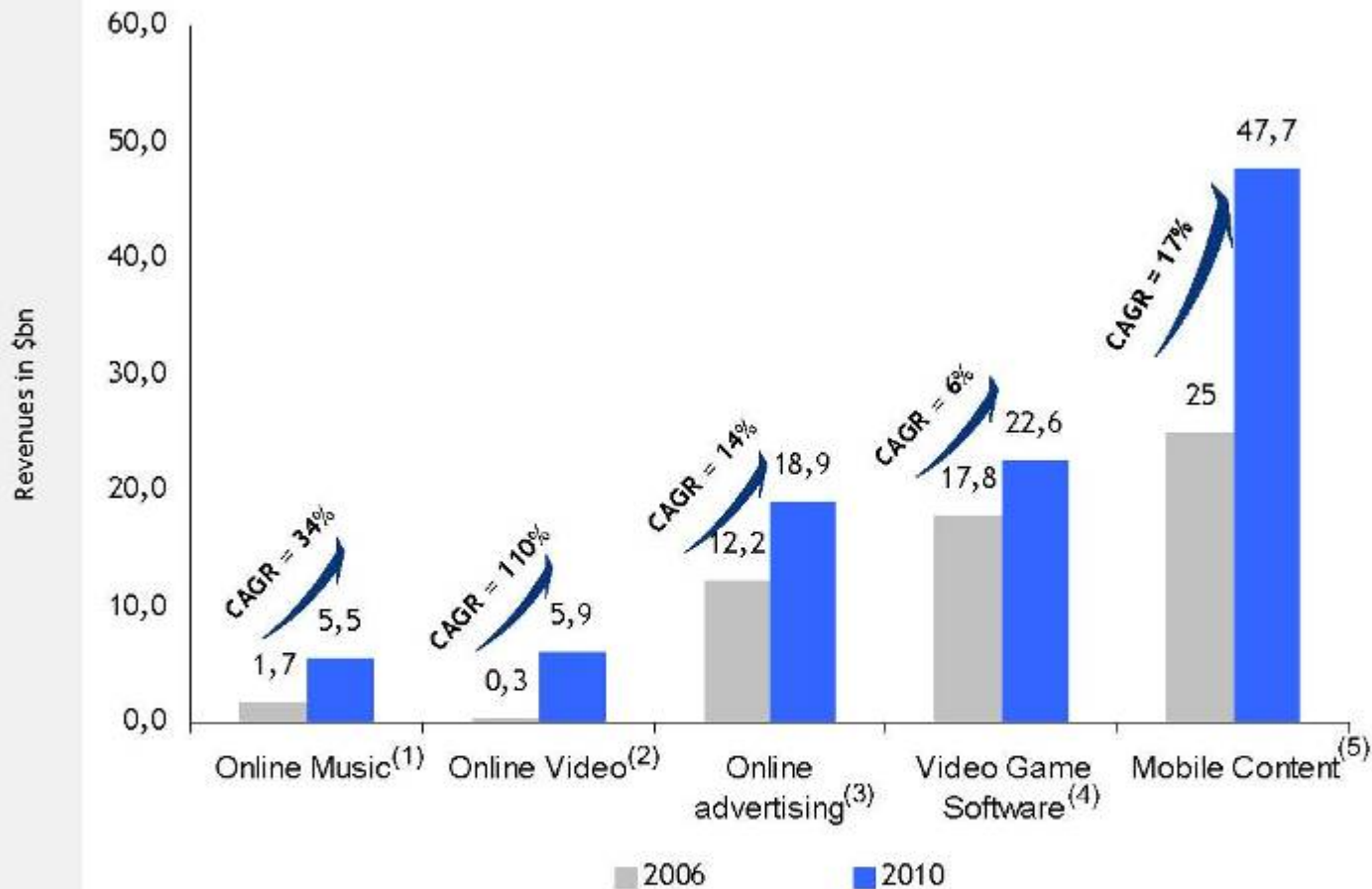


Évolution du marché multimédia mobile



Source : Juniper Research

Évolution du marché mobile multimédia



Source: (1) IDC Worldwide Paid Online Music Service Revenue, includes Pay-per-download, subscription and premium internet radio
(2) Strategy Analytics, includes downloads, rentals and subscriptions of TV shows, movies and other pre-recorded video
(3) Jupiter Research
(4) IDC Worldwide Videogame Software Revenue
(5) IDC Worldwide Mobile Content Revenue

Acteurs

- Cinq types d'acteurs
 - Matériel, les fabricants des matériels
 - Opérateurs de téléphonie
 - Agrégateurs
 - Studios indépendants, éditeurs
 - Éditeurs de Middleware

Acteurs Matériel

Cellon

infineon

GRUPO VITELCOM

wavecom



NOKIA
Connecting People



MOTOROLA



Sony Ericsson

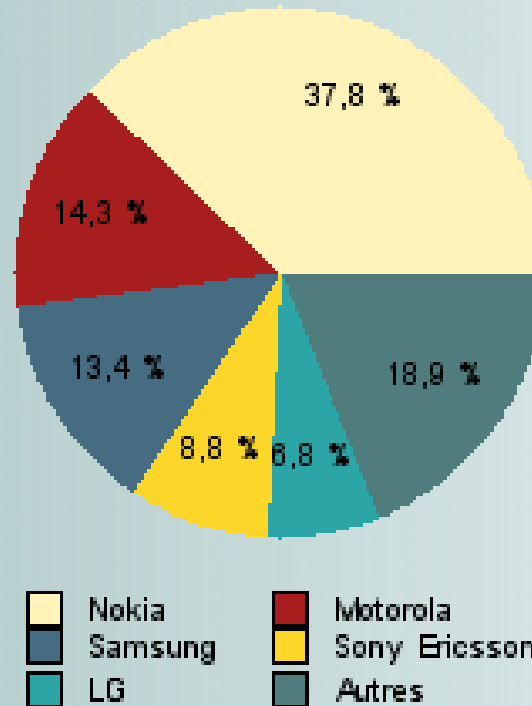
- Constructeur de plates-formes
 - Téléphone en « marque blanche »
 - Infineon, Wavecom, Vitelcom, Cellon, etc.
 - Peu de « marques » conçoivent le matériel, fabriquent leurs téléphones
- Constructeur de téléphones, Marque
 - Téléphones pour le grand public
 - Nokia, Motorola, Philips, Alcatel, etc.
 - Gamme de téléphones
 - Segmentation marketing
 - Jeunes, adultes jeunes, adultes, seniors
 - Hommes, Femmes
 - Actifs, Ménagères, etc

Acteurs Matériel

- Constructeur
 - Concurrence forte
 - Importance du multimédia



Part de marché mondiale des fabricants de mobile en 2007



Source : Gartner, février 2008

Acteurs

Opérateurs

- Offre de téléphonie mobile auprès du client final
 - Orange, SFR, Bouygues
- Offre de services
 - Portail Internet mobile
- Facturation
 - Abonnement
 - Communication voix et données
 - Services
- Propose des téléphones
 - Boutiques d'opérateurs
 - Revendeurs des offres



Acteurs

Agrégateurs

- Agrègent des contenus pour en faire une offre globale
 - Vendue en marque propre
 - Vendue en marque blanche auprès de groupes média
- Forte présence dans la publicité – presse en particulier
- Plate-forme technique de distribution
 - Traitement des commandes
 - Audiotel, SMS
 - Facturation
 - Accord avec les opérateurs et/ou les autorités de régulation
 - Envoi
 - SMS, Push WAP



MediaPlazza
WorldWide Mobile Entertainment



Acteurs

Studio, éditeur

- Création des contenus
 - Principalement jeux vidéos
 - Conception
 - Développement
 - Communication
 - Publicité



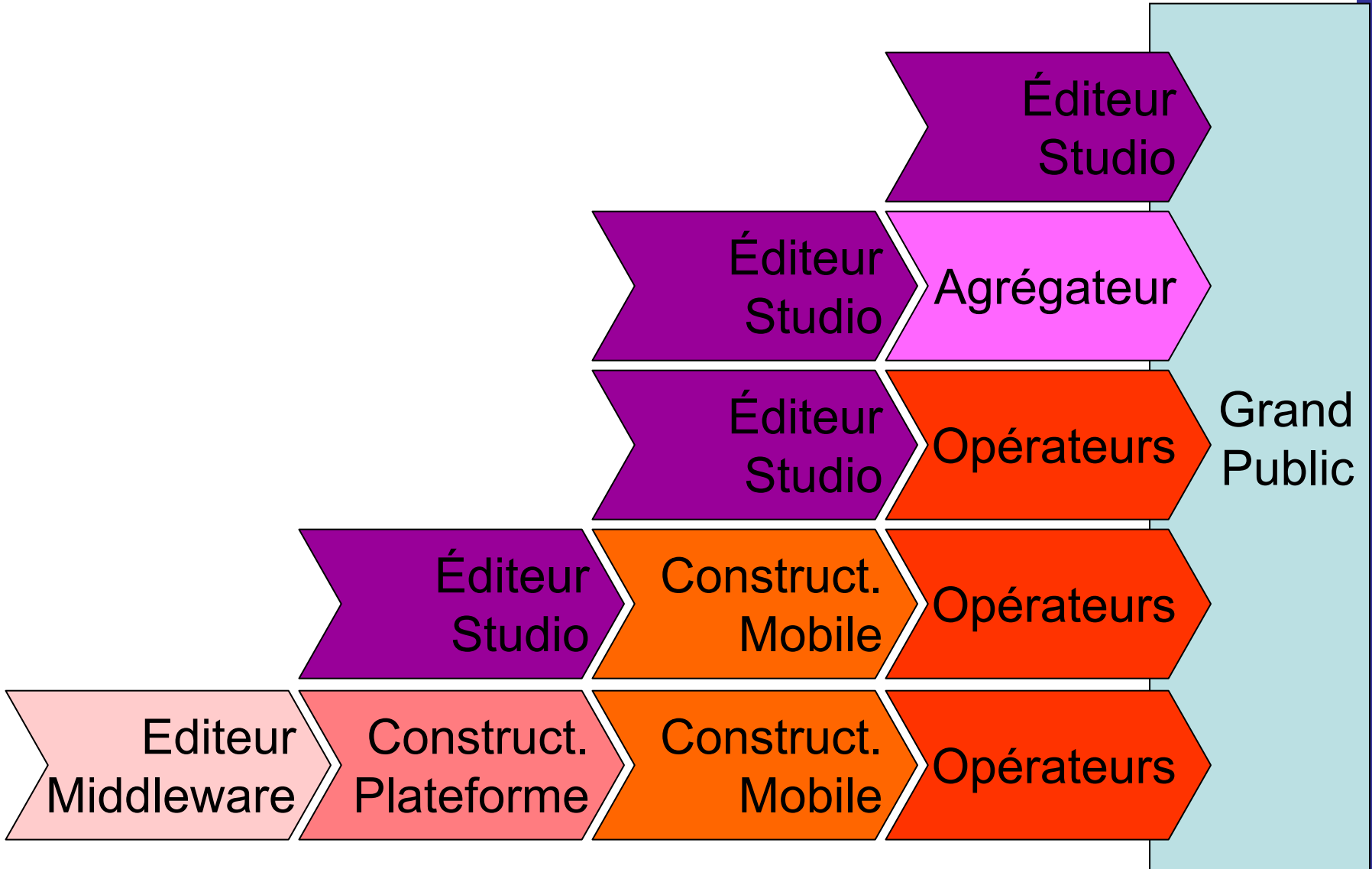
Acteurs

Middleware

- Créations des logiciels applicatifs résidents dans le téléphone
 - Logiciel système
 - Gestion radio
 - Interfaces utilisateurs
 - Carnet d'adresses
 - Agenda
 - Navigateur internet
 - OpenWave
 - Opera software
 - Machine virtuelle Java
 - Sun, Esmertec, Aplix
 - Lecteur de vidéo
 - Real

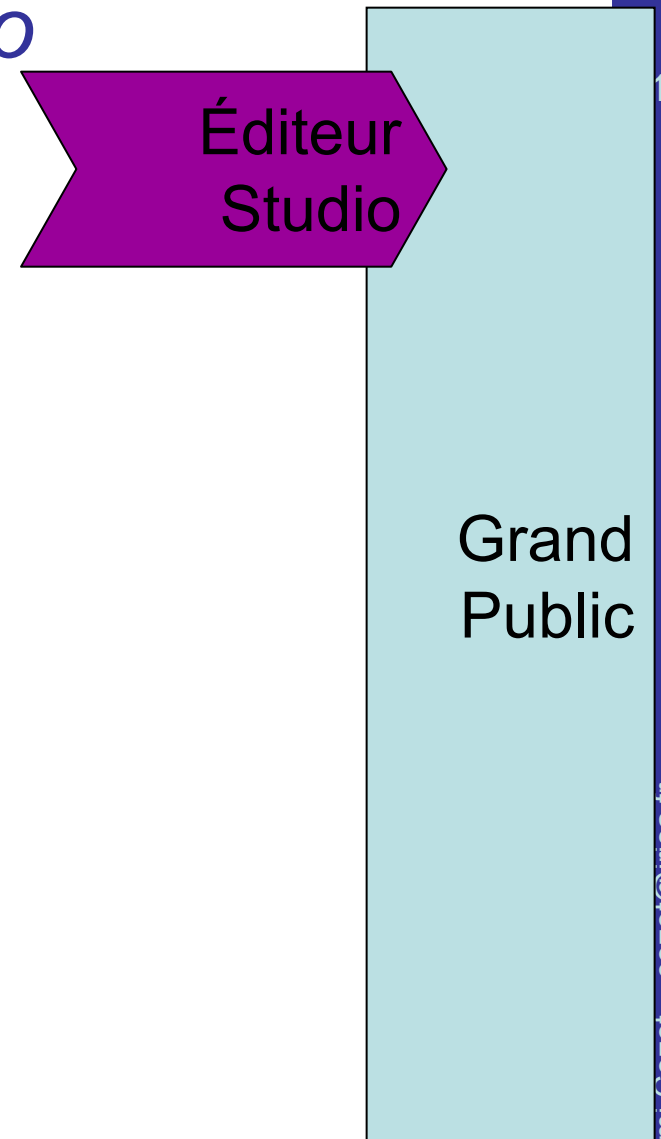


Chaîne de valeur



Chaîne de valeur

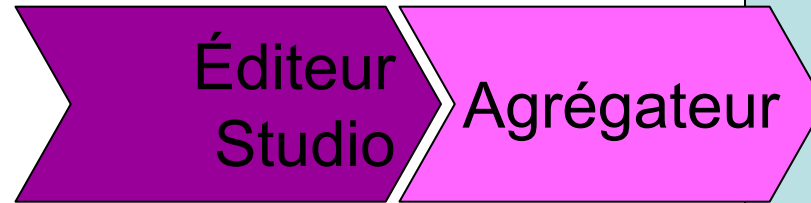
Pour un Studio



- Vente directe
 - Plus
 - Faire connaître la marque
 - Marge plus importante
 - Moins
 - Besoin d'une offre importante
 - Coût de la communication
 - Avoir sa chaîne de distribution « provisioning »

Chaîne de valeur

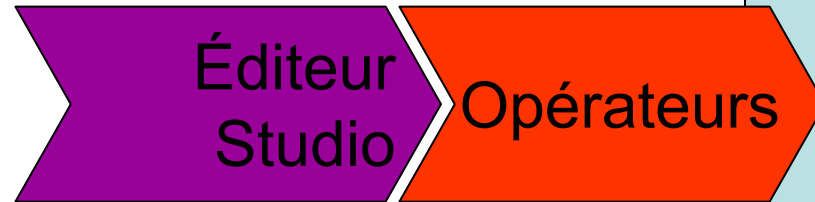
Pour un Studio



- Vente à des agrégateurs
 - Plus
 - Vente facile
 - Moins
 - Marge faible
 - Pas de contrôle sur la communication

Chaîne de valeur

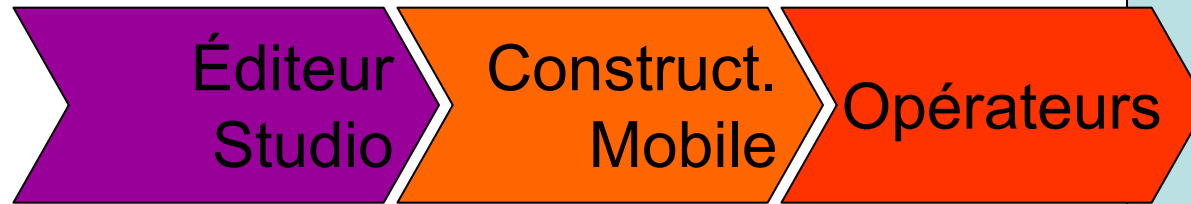
Pour un Studio



- Vente aux opérateurs
 - Opérateur
 - Distributeur ou Éditeur (politique éditoriale)
 - Plus
 - Un seul intermédiaire
 - Plate-forme de paiement
 - Moins
 - Marge faible
 - Pas de contrôle sur la communication

Chaîne de valeur

Pour un Studio



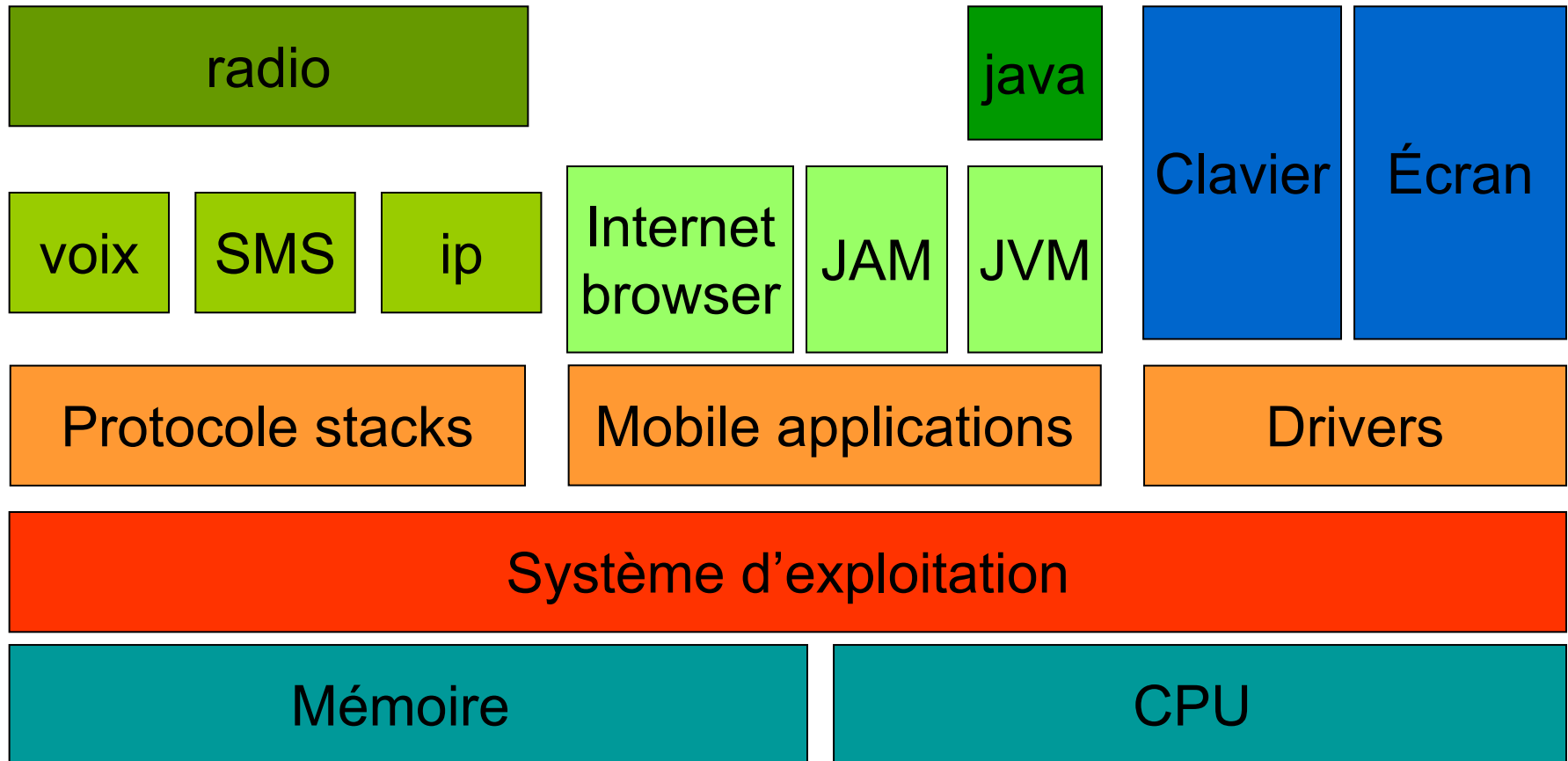
- Vente aux constructeurs
 - Plus
 - Offre en fonction de la cible du téléphone
 - Pas de communication
 - Facturation lié à la production de téléphones
 - Moins
 - Très concurrentiel
 - Vente « industrielle »

L'environnement technique

Partie 2

Architecture d'un téléphone

23



Embarqué ou téléchargement

- Problématique des logiciels sur mobiles
 - Pré-installé
 - configuration fixe initiale(embarqué)
 - A installer
 - Installation par câble
 - Téléchargement (OTA Over The Air)
- Discussion
 - % population ayant un mobile > % population ayant un ordinateur >> **téléchargement**
 - Usage simple (contrairement au PC) >> **installation simple**
 - Téléphone outil fiable (contrairement au PC) >> **sécurité des applications**

Embarqué ou téléchargement

- Synthèse : choix des opérateurs
 - Monde professionnel
 - Installation par câble
 - Monde Grand public
 - Téléchargement
- Contre exemple ?
 - iphone

Solutions technologiques

- Brew (Qualcomm) – **B**inary **R**untime **E**nvironment for **W**ireless
 - Présence : US
 - SDK : C++, accès système (screen buffer par exemple)
 - Téléchargement : oui
 - Faiblesse : sécurité de fonctionnement (crash !)
- Symbian OS (Symbian Ltd)
 - Présence : monde sur SmartPhone
 - SDK : C++, accès système
 - Téléchargement : non
 - Faiblesse : sécurité
- Windows Mobile (Microsoft)
 - Présence : monde sur SmartPhone
 - SDK : C++, accès système
 - Téléchargement : ?
 - Faiblesse : sécurité
- iPhone OS (Apple)
 - Présence : monde sur iPhone
 - SDK : C++, accès système
 - Téléchargement : oui
 - Faiblesse : distribution iTunes uniquement



Solutions technologiques

- LiMo
 - Présence : monde sur SmartPhone (Motorola et ?)
 - SDK : C++, accès système
 - Téléchargement : non mais ?
 - Faiblesse : sécurité
- Androids (Google)
 - Présence : HTC G1 T-Mobile
 - SDK : JAVA like
 - Téléchargement : oui
 - Faiblesse : encore jeune, arrive trop tard ?
- Java (sun)
 - Présence : monde sur presque tous les téléphones, sauf iPhone
 - SDK : Java
 - Téléchargement : oui
 - Faiblesse : lenteur relative
 - Force : choisi par l'OMA
- Doja (NTTDocomo)
 - Présence : monde chez les opérateur iMode
 - SDK : Java
 - Téléchargement : oui
 - Faiblesse : lenteur relative



ANDROID



Solutions techniques

- Pourquoi Java retenu par l'OMA
 - Téléchargement (Over The Air)
 - Modèle de sécurité strict
 - Déconnexion de la partie Java du reste du téléphone
 - Grand nombre de développeurs
 - Pas de « sunStore »
 - Liberté du modèle économique

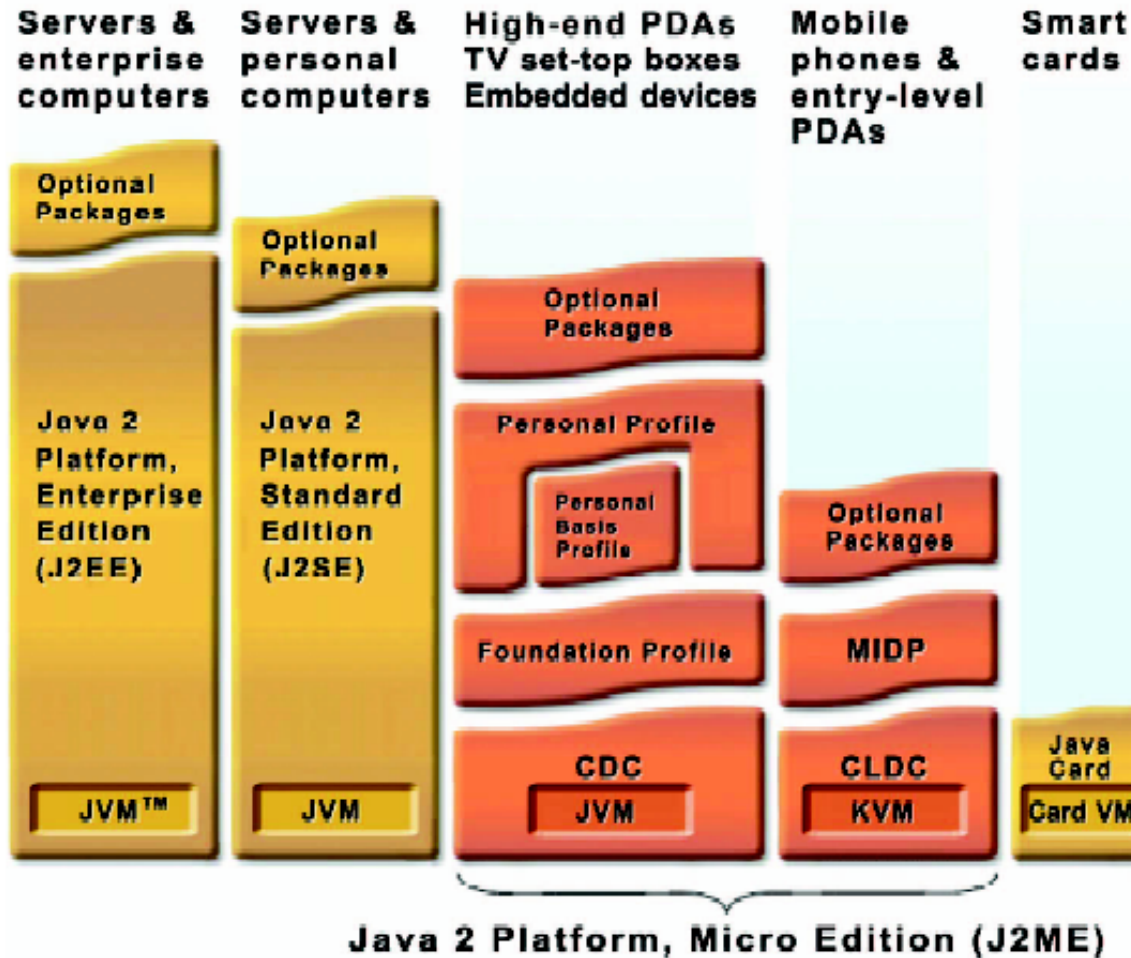
Java

Partie 3



JAVA

et les Mobiles



JAVA

Les raisons du choix

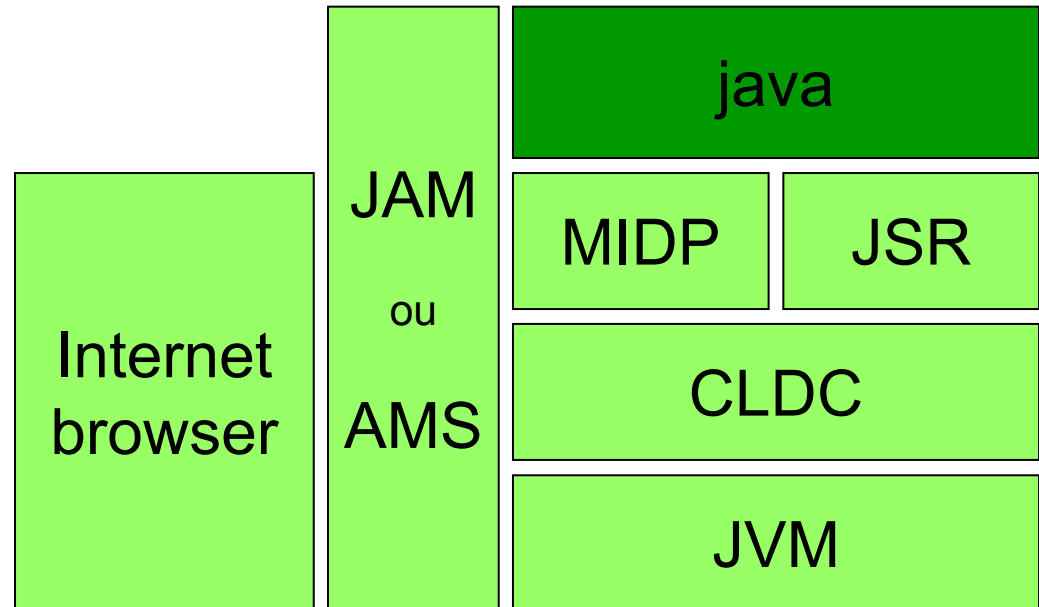


- Solution éprouvée
 - JAVA présent partout
 - Du serveur à la carte à puce
 - Large communauté de développeurs
- Solution portable
 - Indépendance vis à vis
 - Système d'exploitation
 - Processeur
- Solution normalisée
 - Spécification publique
 - Éditeurs tiers possibles
 - Certification SUN
- Adapté à la problématique mobile
 - Léger et scalable
 - Téléchargement par Internet
 - Sûreté de fonctionnement

JAVA Architecture

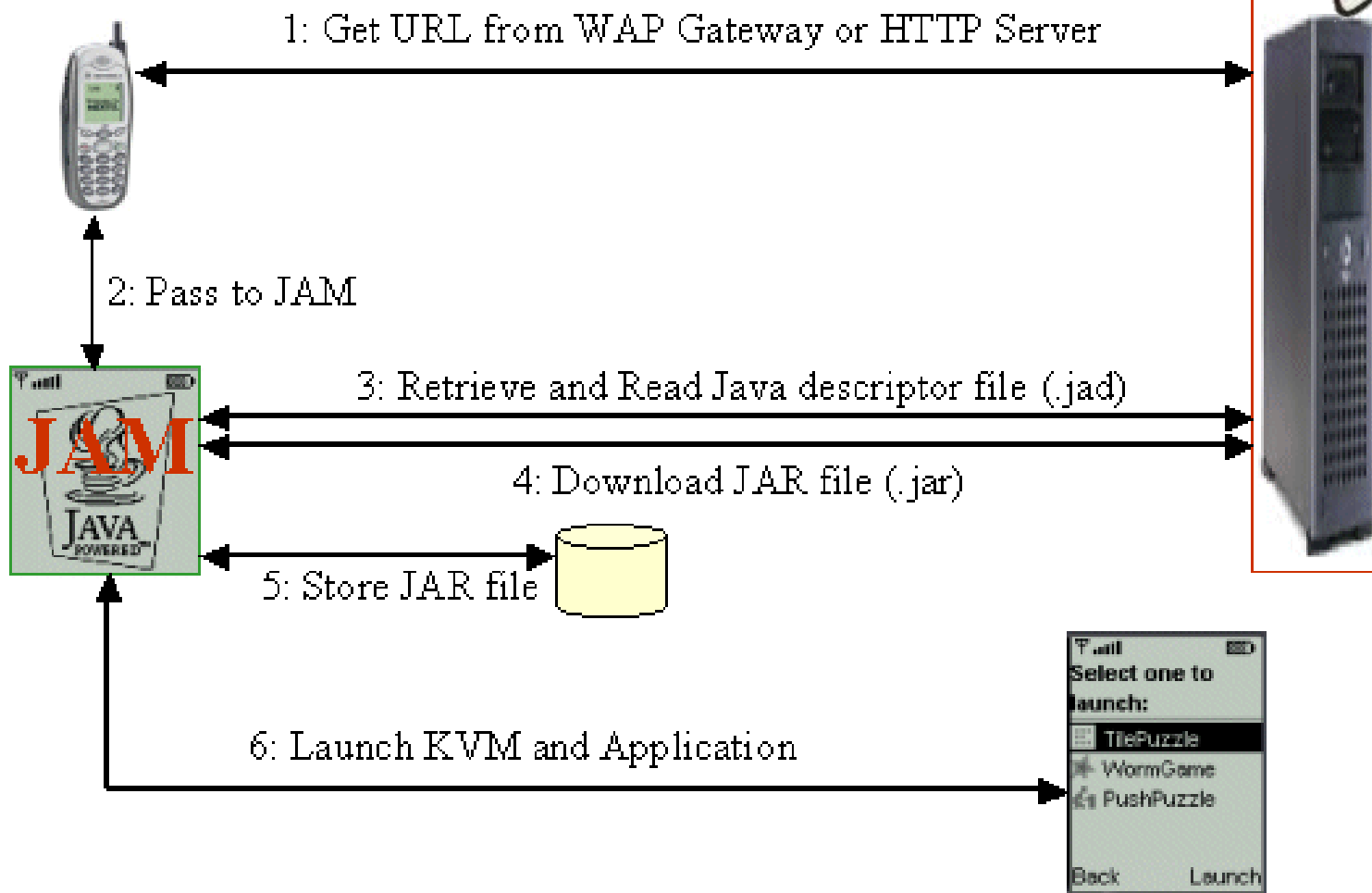


- JAM : Java Application Manager
- AMS : Application Management system
- JVM : Java Virtual Machine
- CLDC : Connected Limited Device Configuration
- MIDP : Mobile Information Device Profile
- JSR : Java Specification Request



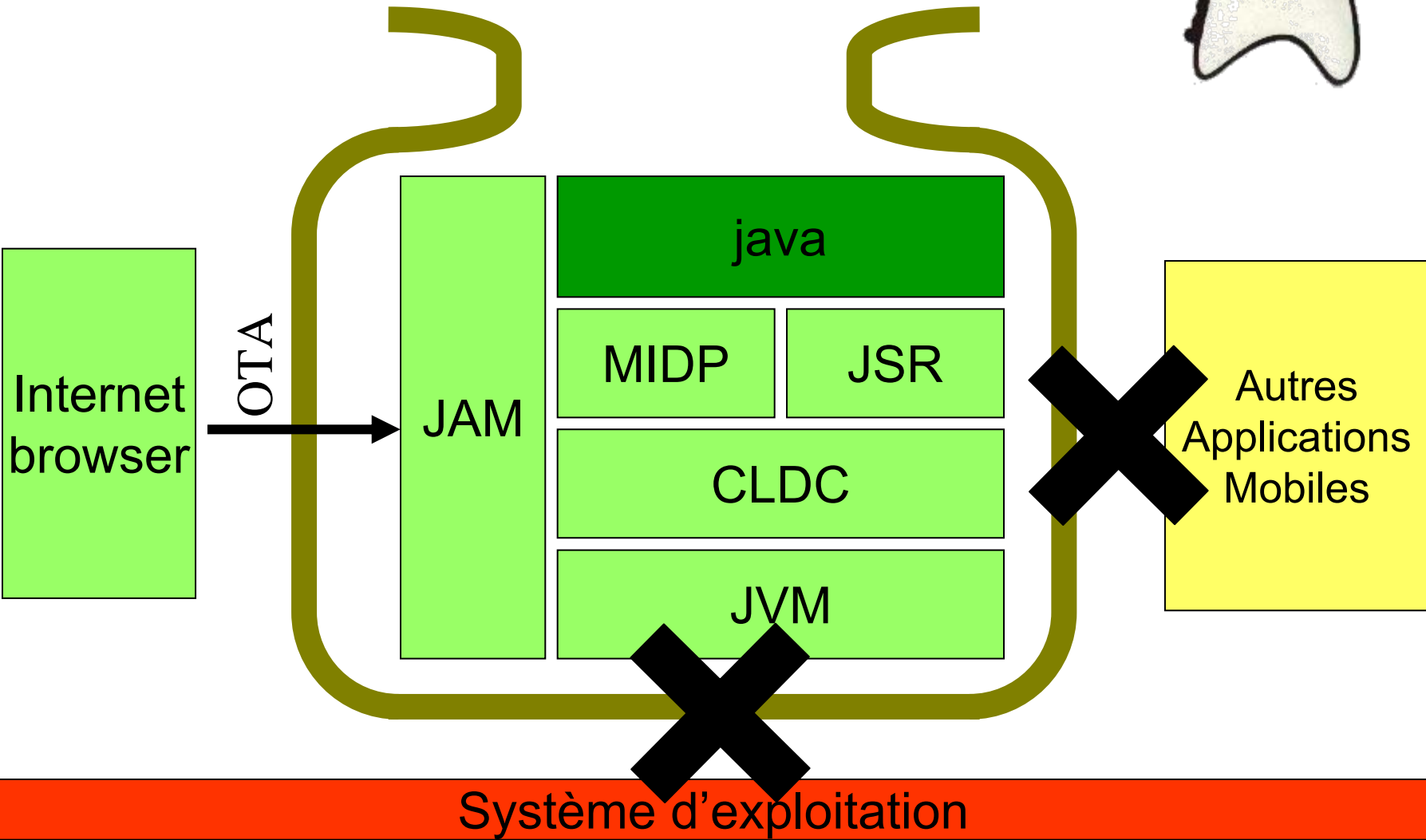
JAVA

téléchargement « OTA »



JAVA

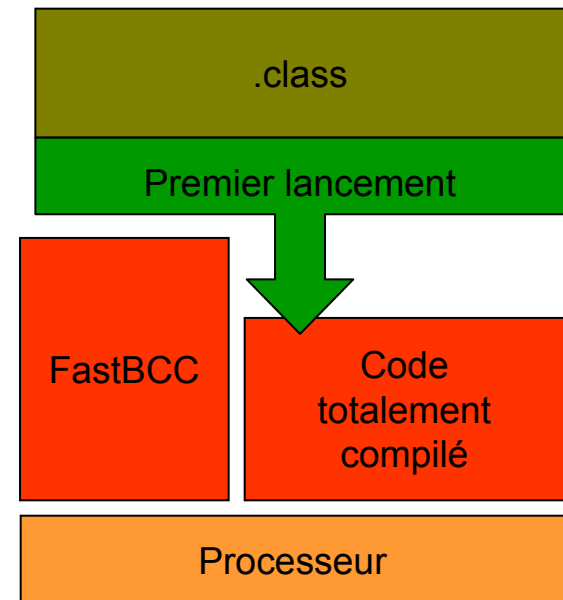
Modèle de sécurité



JAVA

Java2 Micro Edition et JVM

- JVM – J2ME
 - Contrairement à J2SE, ouverture aux éditeurs tiers pour la fourniture de JVM
 - Forte concurrence entre les éditeurs
 - Marché important
- Trois grandes technologies
 - Interprétation totale
 - KVM – Sun (gratuite)
 - Compilation partielle : Just In Time
 - Le code le plus utilisé est compilé
 - HotSpot – Sun
 - JBlend – Aplix
 - FastDAC – Esmertec
 - Compilation totale
 - FastBCC – Esmertec





JAVA

Configuration CLDC 1.1

- Configuration
 - ensemble commun de fonctionnalités pour toute une gamme d'appareils
 - Classes de bases
 - Téléphones mobiles >>CLDC
- CLDC
 - Connected Limited Device Configuration



JAVA

Configuration CLDC 1.1

- Packages

- java.io classes pour les entrées/sorties sous la forme de data streams
- java.lang classes des types **fondamentaux**
- java.lang.ref
- java.util types pour les **collection**, date et gestion du temps
- javax.microedition.io classes pour la gestion du réseaux
Generic Connection Framework



JAVA

Configuration CLDC 1.1

- Java.lang
 - Boolean
 - Byte
 - Character
 - Class
 - **Double** Nouveau depuis CLDC 1.1
 - **Float** Nouveau depuis CLDC 1.1
 - Integer
 - Long
 - Math
 - Object
 - Runtime
 - Short
 - String
 - StringBuffer
 - System
 - Thread
 - Throwable



JAVA

Configuration CLDC 1.1

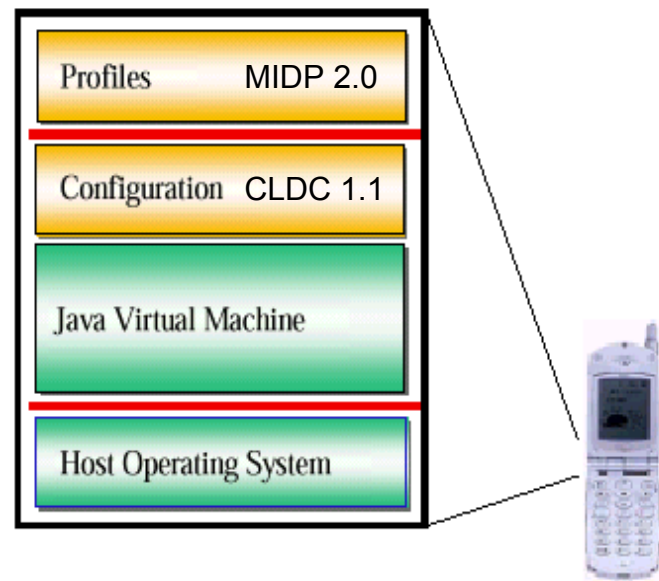
- Pas d'extension native
 - Java Native Interface en J2EE ou J2SE
 - Pas possibilité d'étendre ou d'optimiser les fonctionnalités



JAVA

Profile MIDP

- MIDP - **M**obile **I**nformation **D**evice **P**rofile
 - Prévu pour des environnements
 - Ayant peu de mémoire (128kO)
 - Graphique
 - Gestion de l'écran
 - Connecté sur un réseau Internet
 - Application
 - MIDlet



JAVA

Profile MIDP

- Les grandes fonctionnalités
 - Gestion de l'application
 - javax.microedition.midlet
 - Interface utilisateur
 - javax.microedition.lcdui
 - javax.microedition.lcdui.game
 - Audio
 - javax.microedition.media
 - javax.microedition.media.control
 - Réseau
 - Connexions sécurisés
 - javax.microedition.pki
 - Core
 - java.io
 - java.lang
 - java.util
 - Stockage persistant
 - javax.microedition.rms

- Gestion du « plein écran »
- Multicouches (« layers »)

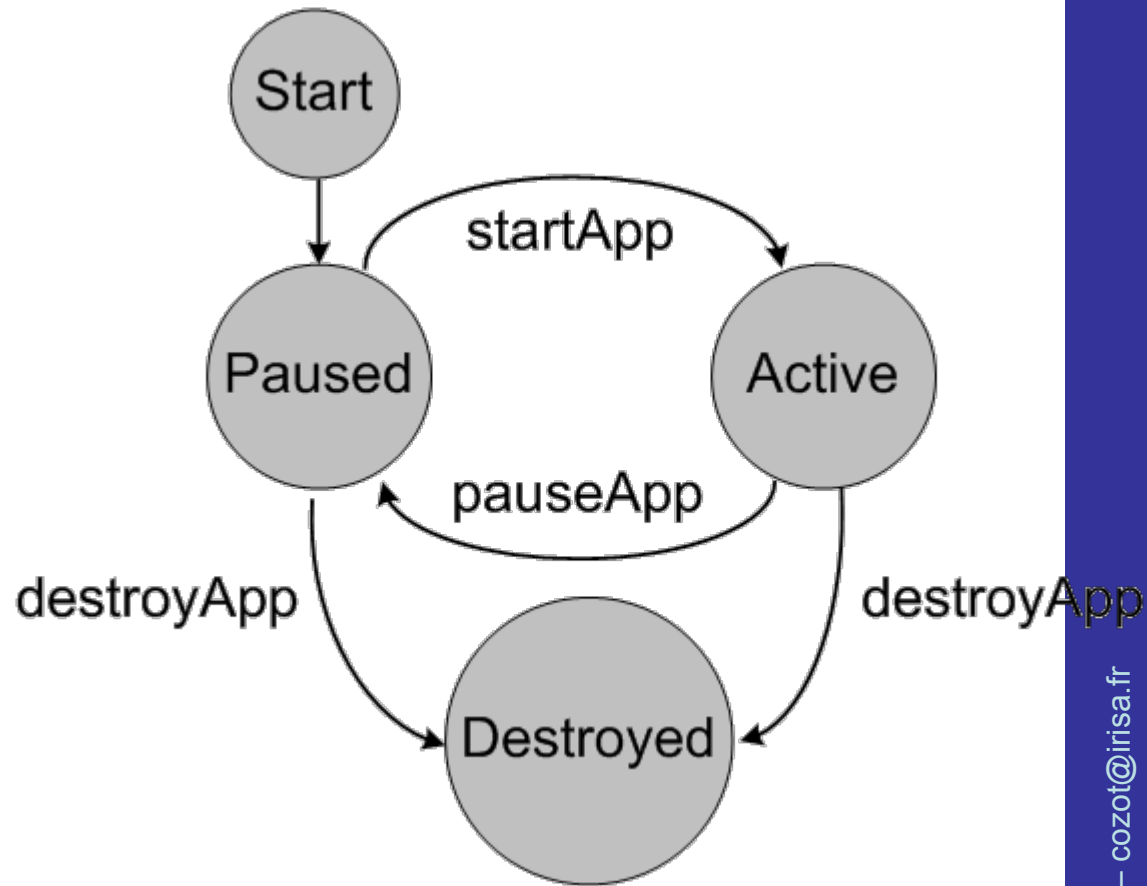
- Gestion du son
- Gestion d'un « player » son

- Gestion du réseau (http)
- Gestion des certificats (https)

JAVA

Profile MIDP

- javax.microedition.midlet
 - Gestion de l'état de la MIDlet
 - Gérer par l'AMS
 - Interface MIDlet
 - pauseApp
 - startApp
 - destroyApp
 - notifyDestroyed
 - notifyPaused
 - resumeRequest
 - getAppProperty



JAVA

Profile MIDP

- **protected abstract void *destroyApp()***
 - L'AMS signale à la MIDlet de finir.
 - La MIDlet rentre dans l'état **destroyed**
- **void *notifyDestroyed()***
 - La MIDlet demande d'entrer dans l'état **destroyed**
- **void *notifyPaused()***
 - La MIDlet demande d'être dans l'état **paused**
- **protected abstract void *pauseApp()***
 - L'AMS signale à la MIDlet de se mettre en pause
 - La MIDlet entre dans l'état **paused**
- **void *resumeRequest()***
 - La MIDlet demande de ré-entrer dans l'état **active**
- **protected abstract void *startApp()***
 - L'AMS signale à la MIDlet qu'elle est dans l'état **active**

JAVA

Profile MIDP

- **Premier exemple de MIDlet**

```
import javax.microedition.midlet.* ;  
import javax.microedition.lcdui.* ;
```

```
public class SimpleMIDlet extends MIDlet {  
    private Display display;
```

```
    public SimpleMIDlet() {display = Display.getDisplay(this);} 
```

```
    public void startApp() {  
        TextBox t = new TextBox("MIDlet", "Bonjour!", 256, 0);  
        display.setCurrent(t);  
    }
```

```
    public void pauseApp() {}
```

```
    public void destroyApp(boolean unconditional) {}  
}
```



JAVA

Profile MIDP

- javax.microedition.lcdui
 - Classe pour la création d'IHM
 - Alert
 - DateField
 - Font
 - Form
 - Gauge
 - Image sans manipulation des pixels en MIDP 1.0
 - Item
 - List
 - TextBox
 - Etc.

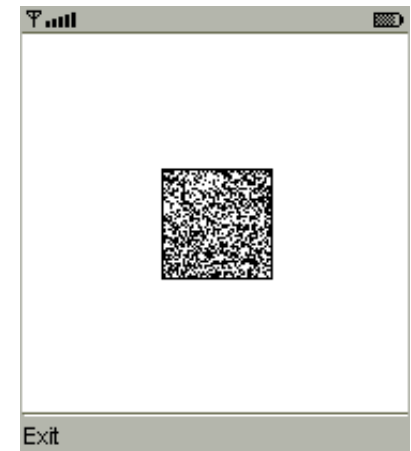


JAVA

Profile MIDP

- `javax.microedition.LcdUI`
 - Image dans MIDP 2.0
 - Accès aux pixels de l'image en écriture

```
public void drawRGB (  
    int[] rgbData, int offset, int scanlength,  
    int x, int y, int width, int height,  
    boolean processAlpha)
```

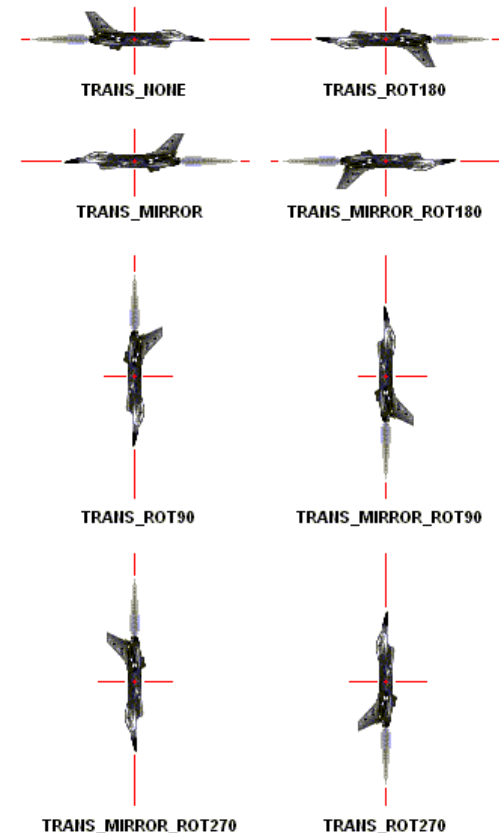




JAVA

Profile MIDP

- `javax.microedition.lcdui.game`
 - Classes graphiques pour les jeux
 - FullCanvas
 - Layer
 - TiledLayer
 - Sprite





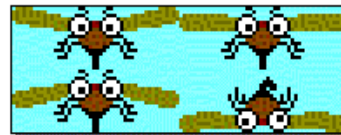
JAVA

Profile MIDP

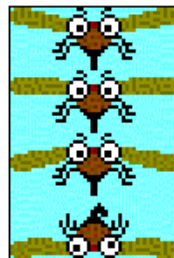
- `javax.microedition.lcdui.game`
 - Classes graphiques pour les jeux
 - Sprite



OR



OR



Frames



frameWidth

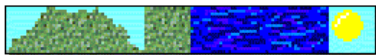




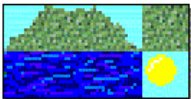
JAVA

Profile MIDP

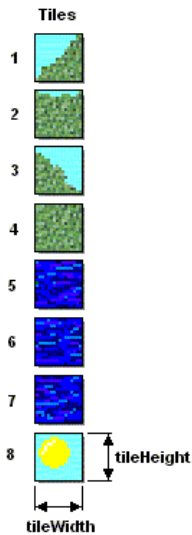
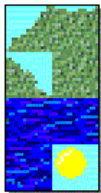
- javax.microedition.lcdui.game
 - Classes graphiques pour les jeux
 - TiledLayer



OR

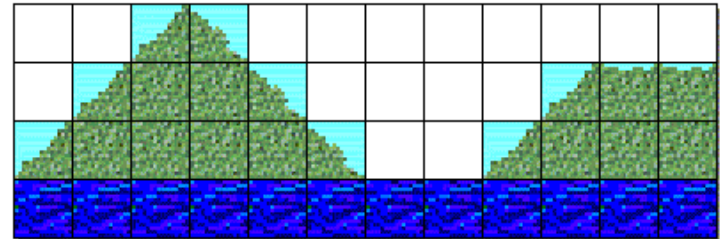


OR



Cells

0	0	1	3	0	0	0	0	0	0	0	0
0	1	4	4	3	0	0	0	0	1	2	2
1	4	4	4	4	3	0	0	1	4	4	4
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1





JAVA

Profile MIDP

- Squelette de MIDlet d'animation (1/3)

```
// classe principale
```

```
public MainClass {
```

```
    // surface de dessin
```

```
    private GameCanvas canvas = new MyGameCanvas();
```

```
    public MainClass() {
```

```
        // démarre le thread associé au canvas
```

```
        canvas.start();
```

```
    }
```

```
    // le reste du code
```

```
}
```



JAVA

Profile MIDP

- Squelette de MIDlet d'animation (2/3)

// la classe de dessin – GameCanvas et Threads

```
public MyGameCanvas extends GameCanvas implements Runnable {
```

```
    // constructeur
```

```
    public MyGameCanvas() { ... }
```

```
    // méthodes héritées de Runnable
```

```
public void start() {
```

```
    // Initialisations
```

```
    // lancement du thread
```

```
    Thread runner = new Thread(this);
```

```
    runner.start();
```

```
}
```



JAVA

Profile MIDP

- Squelette de MIDlet d'animation (3/3)

// la classe de dessin – GameCanvas et Threads

```
private void run() {  
    while(keeprunning == true) {  
        // keeprunning est une variable d'instance (volatile)  
        // logique du jeux  
            verifyGameState();  
            checkUserInput();  
            // mise à jour de l'affichage  
            updateGameScreen(getGraphics());  
            // contrôle du « frame rate »  
            Thread.sleep(milliseconds);  
    }  
}
```

JAVA

Profile MIDP

- Quelques détails sur GameCanvas
 - Double buffer
 - Animation douce
 - Buffer affiché et buffer de travail
 - `flushGraphics()` le buffer de travail est affiché
 - Mémorisation des actions sur les touches
 - `getKeyState()` récupère les actions depuis le dernier appel

```
int keyState = getKeyStates();  
if ((keyState & LEFT_PRESSED) != 0) {  
    System.out.println("Left Key Pressed");  
}
```

JAVA

Profile MIDP

- Quelques détails sur TiledTlayer

```
private TiledLayer create() {
    Image image = Image.createImage("/tiles.png"); }
    TiledLayer tiledLayer = new TiledLayer(10, 10, image, 16, 16);
    int[] map = {
        1, 1, 1, 1, 11, 0, 0, 0, 0, 0,
        ....
        0, 0, 0, 0, 0, 7, 6, 0, 0, 0 };
    for (int i = 0; i < map.length; i++) {
        int column = i % 10; int row = (i - column) / 10;
        tiledLayer.setCell(column, row, map[i]);
    }
    return tiledLayer;
}
```



JAVA

Profile MIDP

- `javax.microedition.rms`
 - Accès en écriture et lecture à la mémoire persistante de la MIDlet
 - « RecordStore »
 - Les données des RecordStores persistent après l'arrêt de la MIDlet et/ou du téléphone

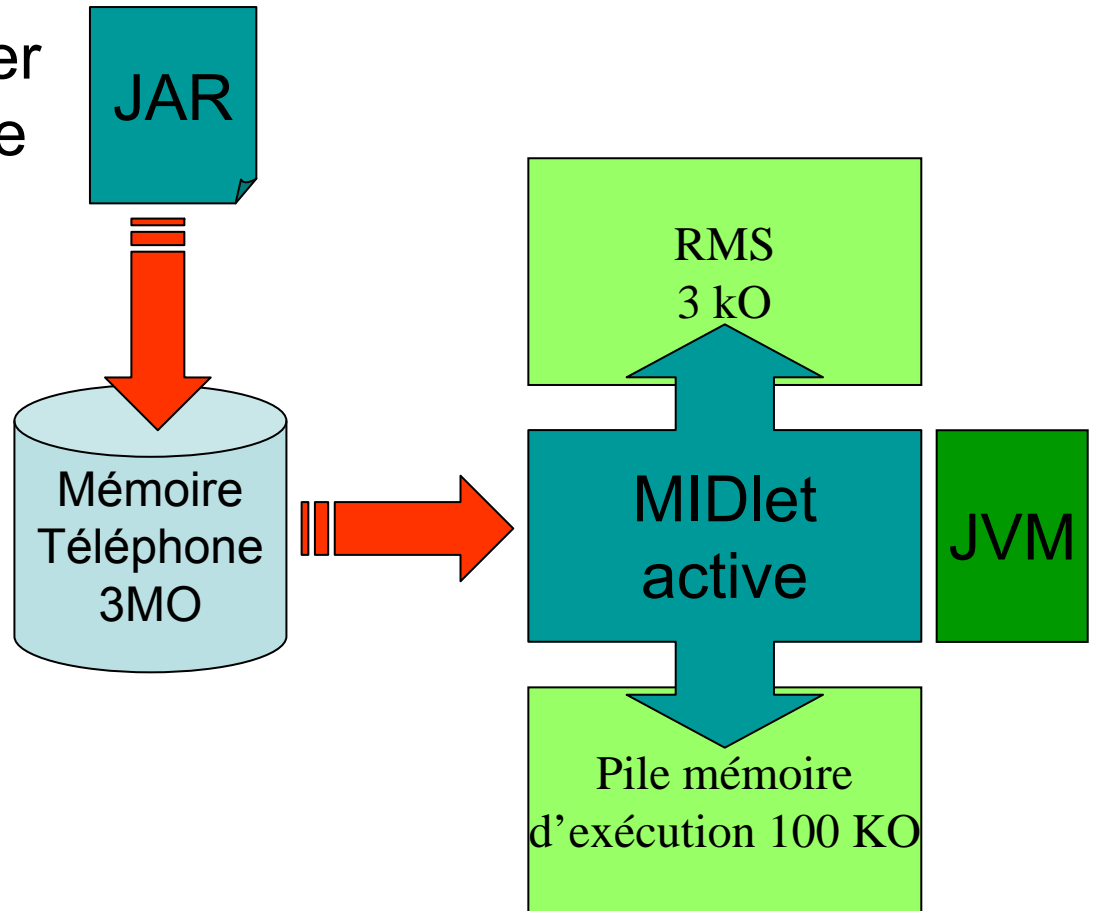
JAVA : Sauvegarde persistante de données

- Les fonctionnalités de RecordStore
 - openRecordStore
 - addRecord
 - deleteRecord
 - getRecord
 - setRecord
 - getRecordSize
 - getNumRecords
 - getNextRecordID

JAVA

Profile MIDP

Taille du fichier
Download size
256 kO



JAVA

Profile MIDP

- `javax.microedition.io`
 - HTTP
 - Interaction serveur WEB
 - HTTPs
 - Connexion sécurisée
 - TCP
 - UDP
 - Plus adapté aux jeux

JAVA

Profile MIDP

- Les fichiers « .class » et les ressources sont regroupés dans une Midlet suite
 - Fichier « .jar »
 - « .class », bytecode
 - Les fichiers ressources
 - Images
 - niveaux
 - Ficher « .JAD »
 - Java Application Description



JAVA

Profile MIDP

- JAD
 - MIDlet-Name
 - MIDlet-Version
 - MIDlet-Vendor
 - MIDlet-Jar-URL
 - MIDlet-Jar-Size
 - MicroEdition-Profile
 - MicroEdition-Configuration

JAVA

Java Specification Request

- JSR – JAVA Specification Request
 - Extension officielle de JAVA
 - Certaines concernent MIDP
 - Les constructeurs peuvent les inclure
- Remarque
 - Ce sont les opérateurs qui les sélectionnent

JAVA

Java Specification Request

- JSR 184 – 3D for mobile
- JSR 239 – Open GL ES binding
- JSR 120 – messaging API
- JSR 205 – messaging API V2
- JSR 135 – multimedia API
- JSR 234 – multimedia API V2
- JSR 179 – localisation API

JAVA

Java Specification Request

- JSR 120 – SMS

```
try {  
    String addr = "sms://+358401234567";  
    MessageConnection conn = (MessageConnection)  
        Connector.open(addr);  
  
    TextMessage msg =  
        (TextMessage)conn.newMessage(  
            MessageConnection.TEXT_MESSAGE);  
    msg.setPayloadText("Hello World!");  
    conn.send(msg);  
}  
catch (Exception e) {...}
```

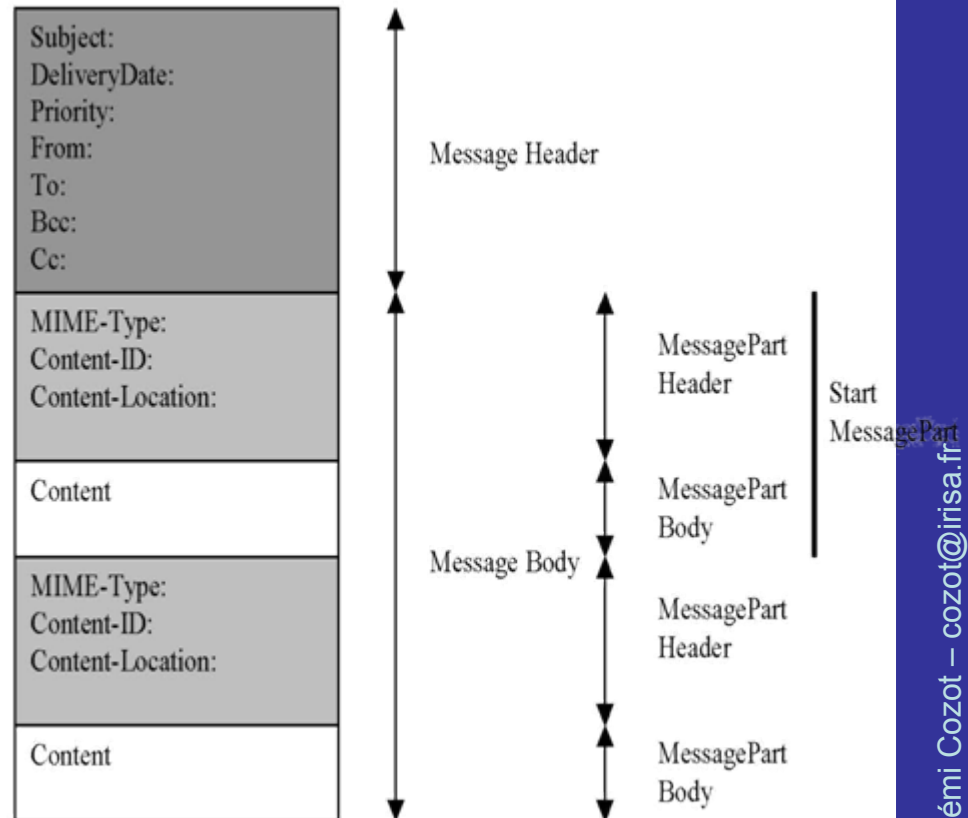
JAVA

Java Specification Request

- JSR 205 – Wireless Messaging 2

- Extension de la JSR 120

- Messages en plusieurs parties
- Messages binaires
 - Dont MMS



JAVA

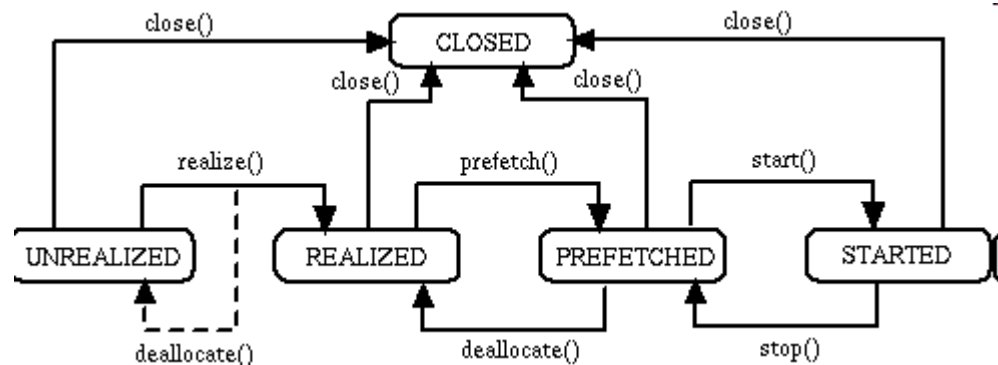
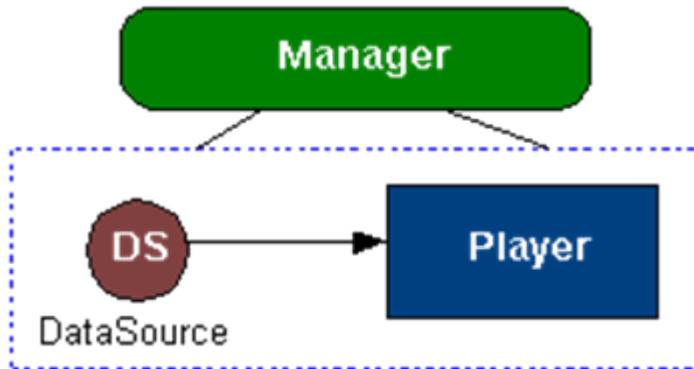
Java Specification Request

```
try {
    String addr = "sms://5432";
    MessageConnection conn = (MessageConnection) Connector.open(addr);
    Message msg = null;
    while (someExitCondition) {
        // attente de la reception d'un message
        msg = conn.receive();
        // message reçu
        if (msg instanceof TextMessage) {
            TextMessage tmsg = (TextMessage)msg;
            String receivedText = tmsg.getPayloadText();
            // réponse avec l'entête "Received:"
            tmsg.setPayloadText("Received:" + receivedText);
            conn.send(tmsg);
        }
        else { // message reçu est binaire ... }
    }
}
catch (Exception e) { ...}
```

JAVA

Java Specification Request

- JSR 135 – Multimédia
 - javax.microedition.media
 - class **Manager** – Classe centrale
 - javax.microedition.media.control
 - javax.microedition.media.protocol



JAVA

Java Specification Request

- Exemple : vidéo dans le .jar

```
try {  
    InputStream is =  
        getClass().getResourceAsStream("audio.wav");  
    Player p = Manager.createPlayer(is, "audio/X-wav");  
    p.start();  
}  
catch (IOException ioe) { }  
catch (MediaException me) { }
```

JAVA

Java Specification Request

- Gestion de la caméra

```
Player p; VideoControl vc;
// initialisation de la caméra
try { p = Manager.createPlayer("capture://video"); p.realize();
    // récupération du contrôle de la caméra et affichage
    vc = (VideoControl)p.getControl("VideoControl");
    if (vc != null) {
        Form form = new Form("video");
        form.append((Item)vc.initDisplayMode(vc.USE_GUI_PRIMITIVE, null));
        Display.getDisplay(midlet).setCurrent(form);
    }
    p.start(); }
catch (IOException ioe) { }
catch (MediaException me) { }
// capture d'une image
try { byte[] pngImage = vc.getSnapshot(null);
    // traitement image ...
}
catch (MediaException me) { }
```

JAVA

Java Specification Request

- JSR 234 – multimédia avancé
 - Plus de formats supportés
 - Son
 - 3D
 - Mixage
 - Effets Sonores
 - Doppler
 - Réverbération
 - etc

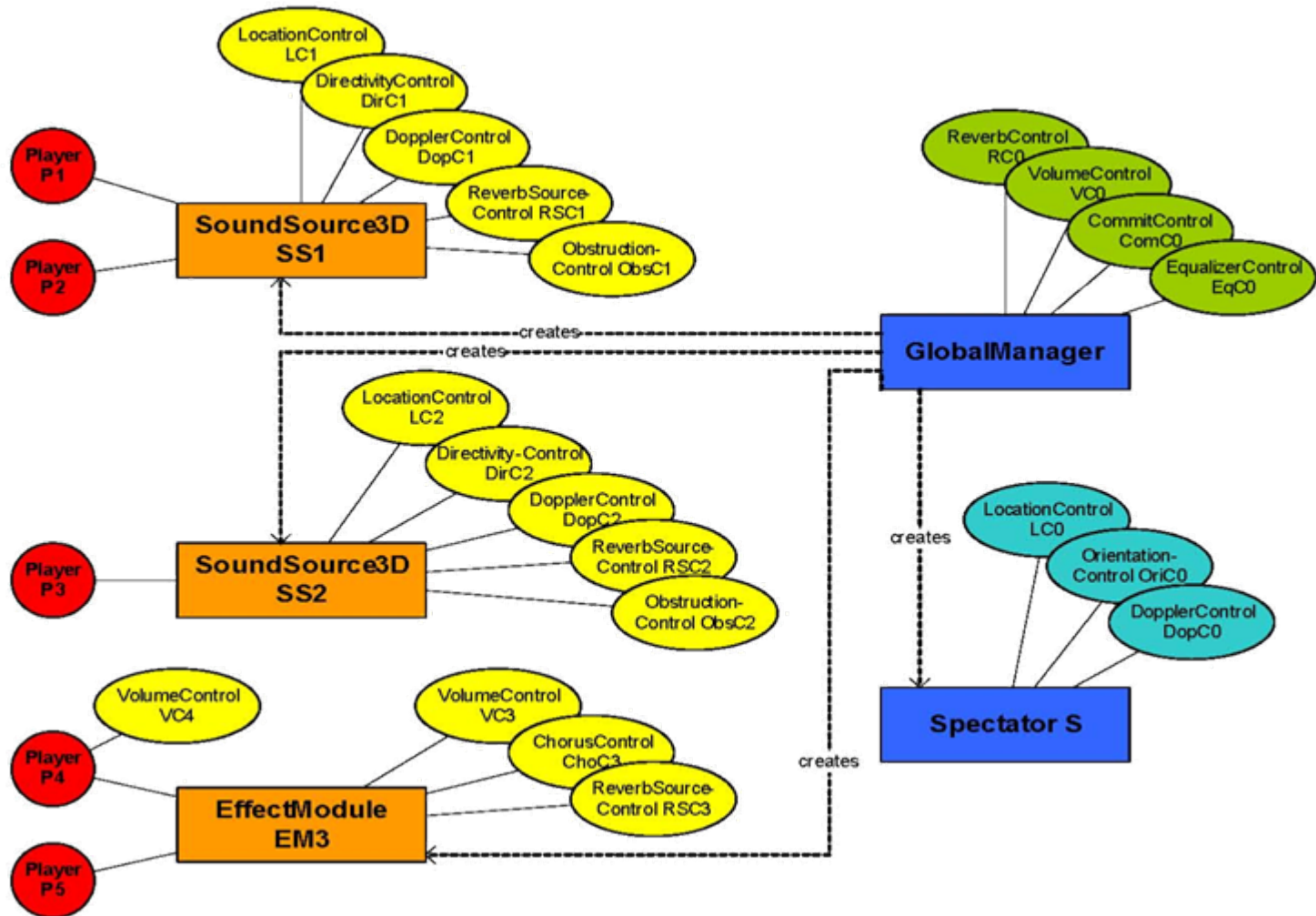
JAVA

Java Specification Request

- JSR 234 – multimédia avancé
 - Contrôle de la caméra
 - Exposition
 - Flash
 - Mise au point – Focus
 - Focale – zoom
 - Post traitement des images
 - Tonalité
 - Balance de blanc
 - Multiplan – overlay
 - Radio

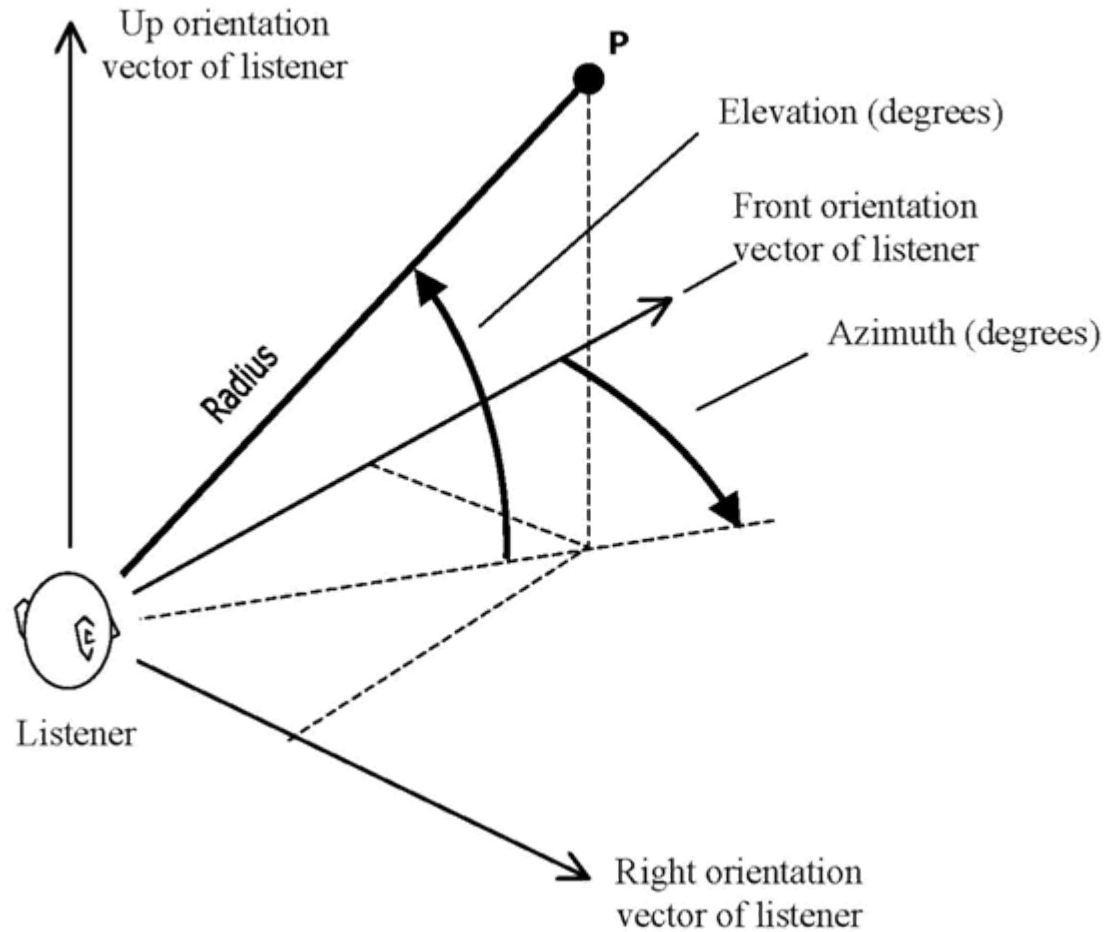
JAVA

Java Specification Request



JAVA

Java Specification Request



JAVA

Java Specification Request

- JSR 179 – géolocalisation
- Exemple

```
try {  
    // Objet Criteria spécifie les paramètres de la localisation  
    Criteria cr = new Criteria();  
    // précision horizontale : 500 mètres  
    cr.setHorizontalAccuracy(500);  
    LocationProvider lp = LocationProvider.getInstance(cr);  
    // récupération de la localisation avec "time out" 60 secondes  
    Location l = lp.getLocation(60);  
    Coordinates c = l.getQualifiedCoordinates();  
    if (c != null) { // utilisation des coordonnées }  
} catch (LocationException e) { // incapable de localiser }
```

JAVA

Java Specification Request

- JSR 184 – JAVA 3D
 - Interface haut niveau
 - Graphe de scène, squelette, etc



JAVA

Java Specification Request

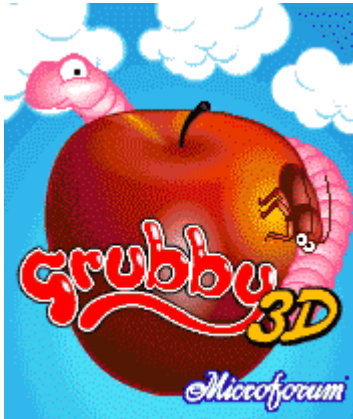
- JSR 184
 - Exemples



GRAND PRIX

TM&©2005 SCEI

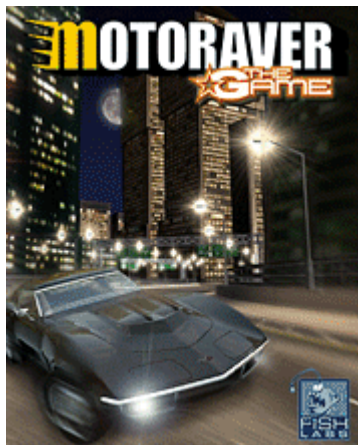
AS YOU COMPLETE TRACKS IN THIS M



JAVA

Java Specification Request

- JSR 239
 - Interface JAVA pour OpenGL ES



JAVA

Java Specification Request

Les JSR offrent des vraies plus
en terme de fonctionnalités

MAIS

Elles fragmentent encore plus le
marché



Contraintes à prendre en compte

Partie 4

Diversité des matériels

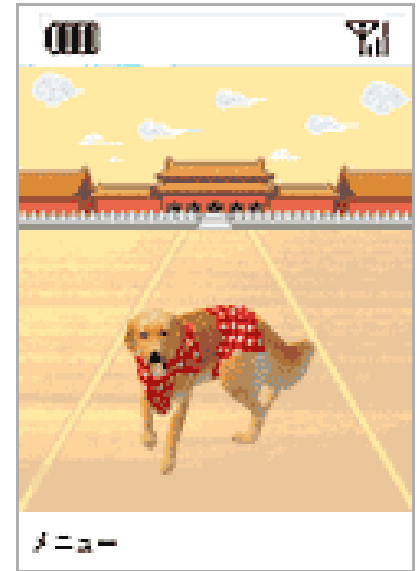


Diversité des matériels



Diversité des matériels

- Évolution du matériel
 - Plus de puissance
 - Coprocesseurs arithmétiques
 - Plus de mémoire
- Arrivée des processeurs 3D
 - Accélération 2D
 - Accélération 3D



Diversité des matériels

- Ergonomie
 - Touches
 - Disposition
 - Réaction lente
 - « Unique »
 - PAD
 - Quatre directions
 - Manque de précision

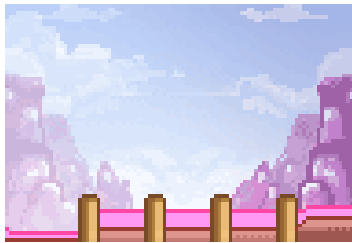


Diversité des matériels

- Ergonomie ... suite
 - Écran
 - Faible résolution
 - 128x128 devient le minimum
 - Rendu des couleurs
 - Nombre de couleurs
 - Qualité des couleurs
 - Rémanence

Gestion de la diversité

- Portages
 - Bien prévoir toutes les versions
 - Tailles d'écrans



Gestion de la diversité

- Portages ... suite
 - Performance
 - Mémoire
 - Langue
 - Extensions supportés



Gestion de la diversité

- Projet et portage
 - Finalement quasiment une version par téléphone



Merci !