

**WAP** *Wireless Application Protocol*  
**WML** *Wireless Markup Language*



**Didier DONSEZ**

Université Joseph Fourier

IMA – IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`

`Didier.Donsez@ieee.org`

# Au sommaire

---

- Motivations
- L'architecture WAP
- WML
- WMLScript
- WAP 2.0
- Divers

# Motivations

- Services en ligne
- Adapter HTTP et HTML aux « handsets » nomades
  - Affichage limité, contrôle limité (i.e. clavier, pointage, ...)
  - faible mémoire, Débit limité et coût de communication
  - *Cible : Téléphones cellulaires, PalmTops, ...*



# Motivations

## ■ WAP *Wireless Application Protocol*

- stack réseau pour un protocole HTTP « like » et « light »
  - Utilise les couches réseaux des réseaux cellulaires
  - à la place d 'IP, UDP, TCP

## ■ WML *Wireless Markup Language*

- langage hypertextuel adapté aux handsets nomades
  - basé sur XML/DTD : il existe un format plus compact (bytecodé)

# Exemple d 'Applications WAP/WML

- Bourse
- Banque
- Administration
- Actualité
- Météo
- Info Trafic
- Horaires/Retards/Grèves transport en commun
- Mobile Commerce,
- Mobile Payment
- Ticket
- Jeux
- Géo-Localisation
  - Alternative au GPS
- Cartographie
- Email, VoiceMail
- Mobile PIM
  - (Personal Info Management)
  - Agenda
  - Carnet d'Adresse...

# Motivations du WAP

- architecture en couche extensible et évolutive
- supporter par les différents réseaux mobiles
- économie des ressources du terminal/réseau
  - mémoire, CPU, consommation électrique, bande passante
- sécurisation des applications et des communications
- modèle programmable pour les services téléphoniques
- créer des interfaces homme-machine flexibles
- réutiliser sur des technologies largement répandues
  - XML, ECMAScript, VCard, VCal, ...

# WAP

## Wireless Application Protocol

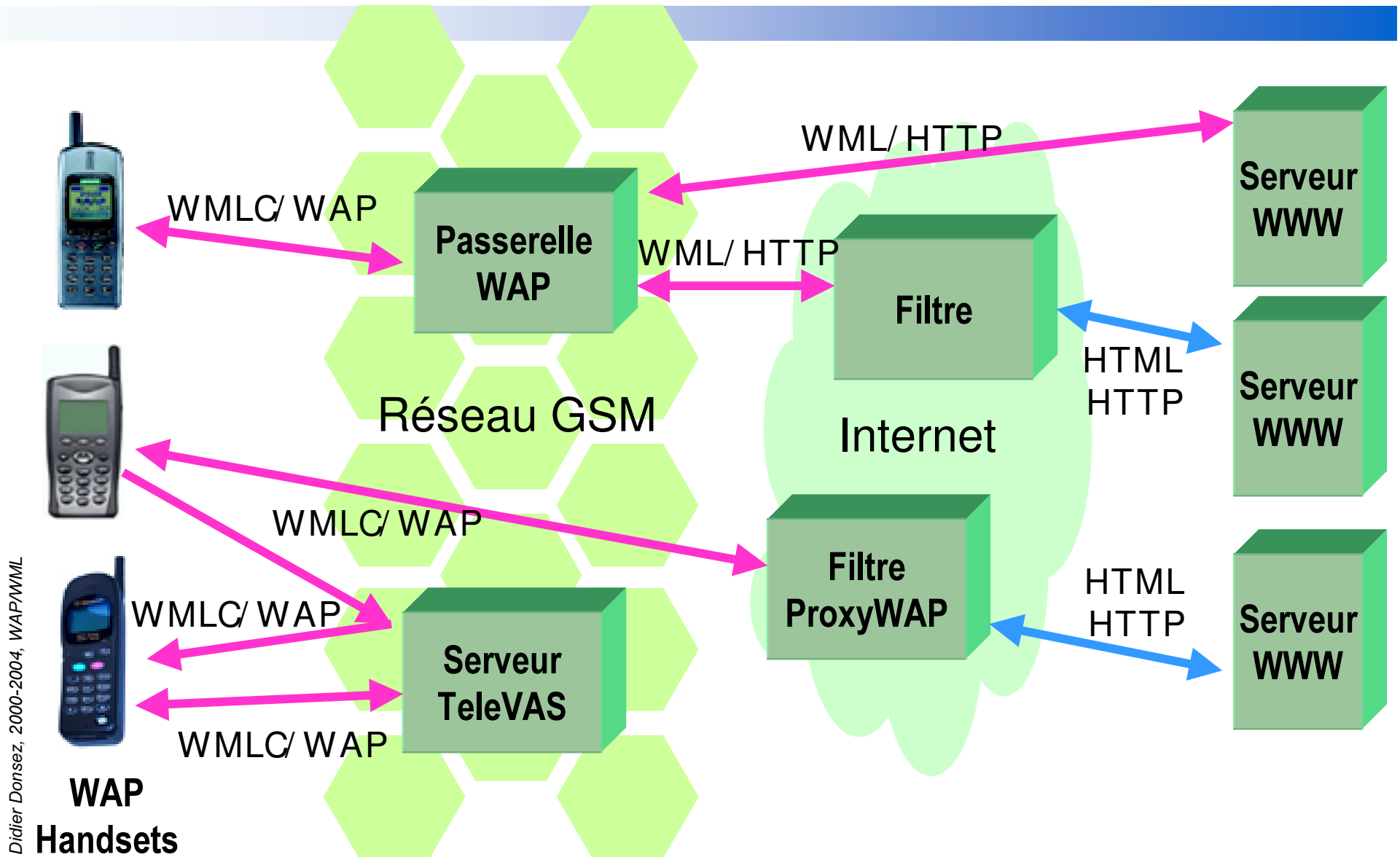
### ■ Architecture WAP

- Client WAP (WAE User Agent)
  - Supporte WML, WMLScript (*en ByteCode*)
  - Supporte d'autres formats éventuellement (*vCard, vCal, Multipart, ...*)
- WAP Proxy
- WAP Server

### ■ 2 modèles d'interaction

- Pull
- Push

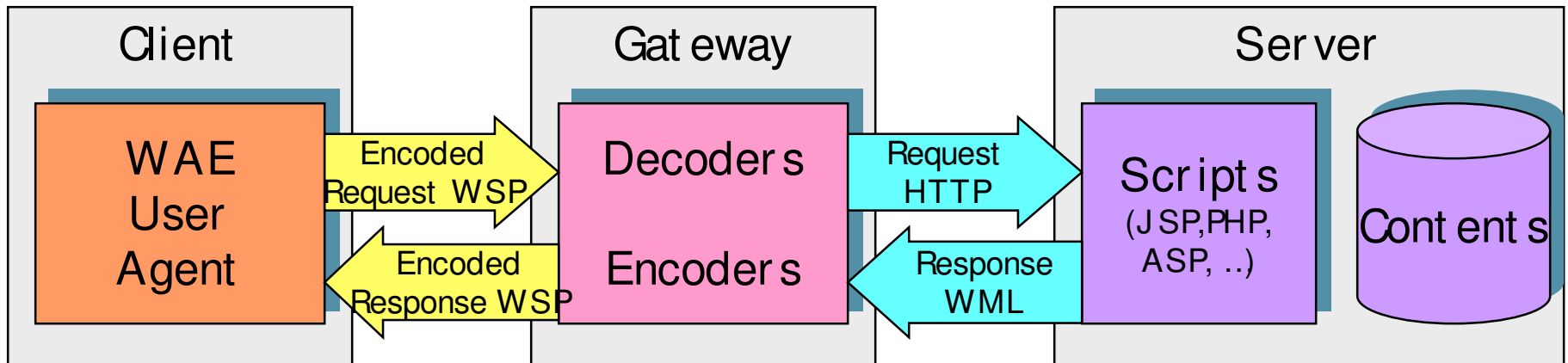
# Architecture Réseau



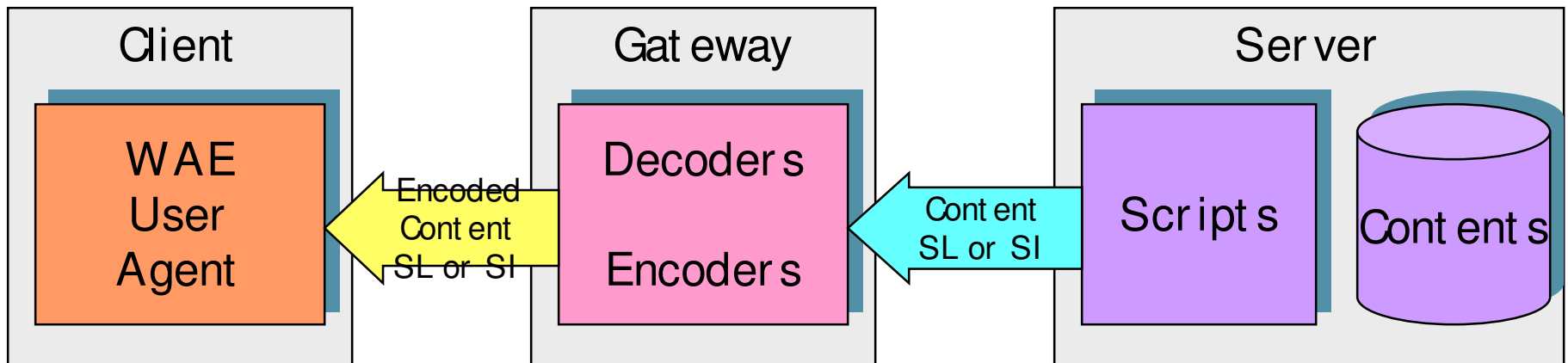


# Modèles d'interaction

## Pull

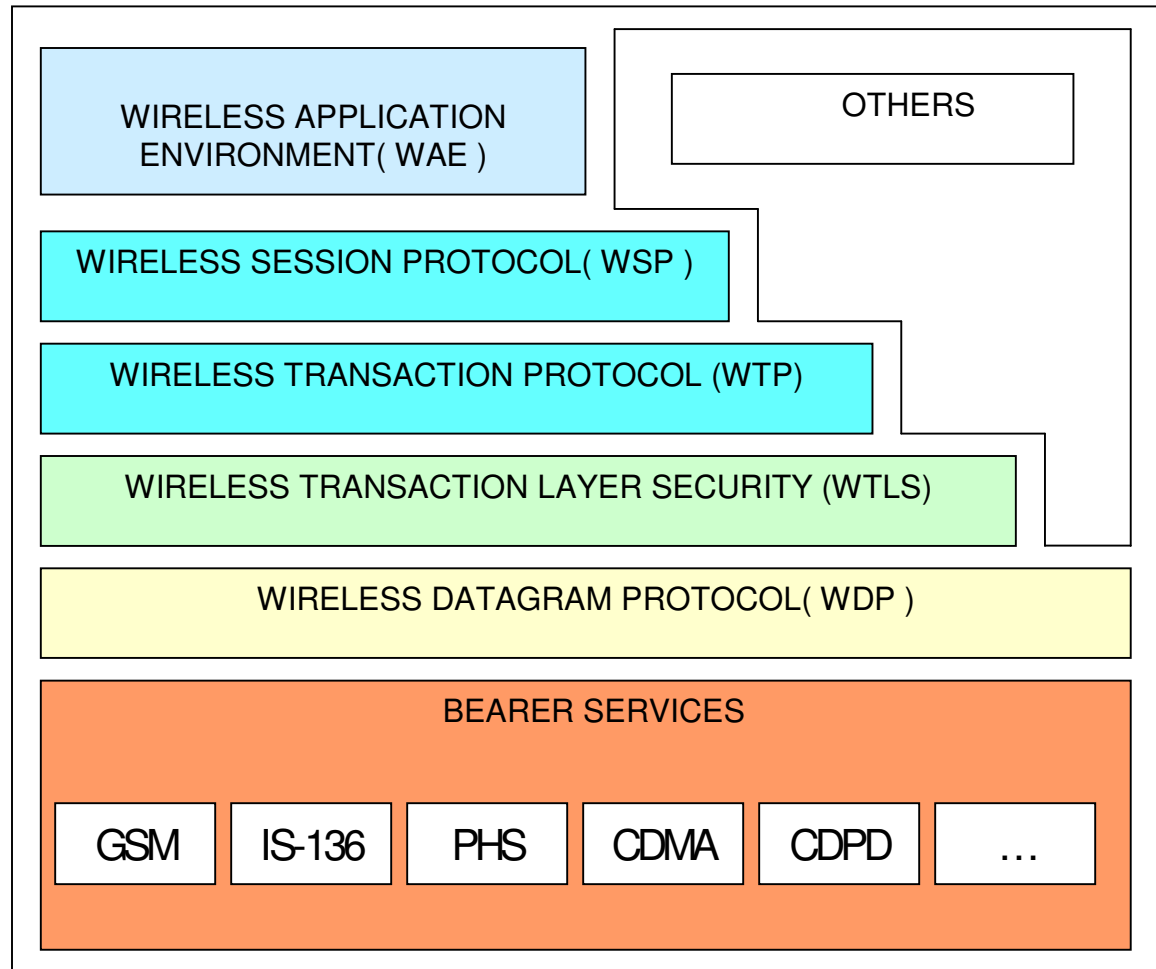
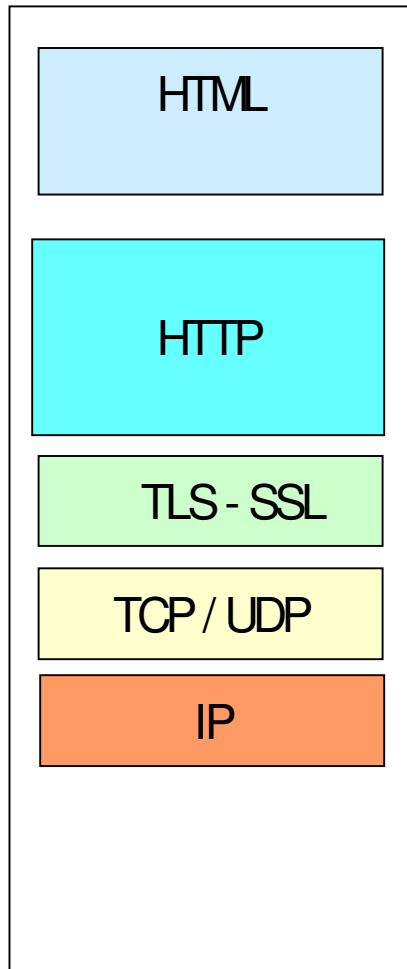


## Push



# WAP

## Les Couches



# Les supports porteurs – Bearers

## ■ Différents services des supports

- GSM, PHS, CDMA, FLEX, iDEN, CDCP, IS6136
- GPRS, UMTS, W-CDMA (WAP 2.0)

☞ différents niveaux de qualité de service

# WDP Wireless Datagram Protocol

- Opère au dessus des services des porteurs de données
- Supporte de nombreux réseaux.
- offre un service aux couches supérieures du WAP et communique de façon transparente au dessus de l'un des services des supports.

# WTLS Wireless Transport Layer Security

## ■ But

- protocole de transport WAP optimisé pour l'usage des canaux de communication de bande passante faible.

## ■ Fonctionnalités

- Intégrité des données
- Privacy : Préserve la vie privée du client
- Authentification (PKI)
- Protection contre les accès non autorisés
- sécuriser de terminaux à terminaux
- utilisation des « credentials » des cartes à puce

## ■ Activation/Déactivation de WTLS par les applications

- en fonction des besoin de sécurité
- et des caractéristiques du réseau
  - ex: dans le GSM où la sécurité se fait dans une couche plus basse

## ■ Basé sur le protocole TLS ( Transport Layer Security )

# WTP Wireless Transaction Protocol

- au dessus du service datagramme  
fonctionne comme un protocole de transaction orientée.
- 3 classes de service de transaction
  - requêtes simples non fiables (classe 0)
  - requêtes simples fiables (classe 1)
  - transaction de requête/réponse fiables (classe 2)
- En option
  - fiabilité utilisateur-utilisateur
    - protocole de confirmation de messages reçus
  - out of band data on acknowledgement
    - Concaténation PDU et acquittement retardé pour limiter le nombre de messages envoyés
  - Transaction asynchrone

# WSP Wireless Session Protocol

## ■ 2 types de service de session

- mode orienté connexion
  - opère au dessus du protocole de la couche transaction WTP
- mode non connecté
  - il opère au dessus du service de datagramme sécurisé ou non sécurisé (WDP)

## ■ Fonctionnalités suivantes

- Mode connexion
  - Entête plus compact d'HTTP/1.1  
Négociation du contenu, ...
  - Etat de session longue durée
  - Suspension de session et aussi reprise avec migration de session
- PUSH

# Optimisation des échanges réseaux

## HTTP/TCP/IP

1. → TCP SYN
2. ← TCP SYN, ACK of SYN
3. → ACK of SYN, **Data Request**
4. ← ACK of Data
5. → **Reply**
6. ← ACK of Reply
7. → **Data Request**
8. ← ACK of Data
9. → **Reply**
10. ← ACK of Reply
11. → **Data Request**
12. ← ACK of Data
13. → **Reply**
14. ← ACK of Reply
15. → TCP FIN
16. ← TCP FIN, ACK of FIN
17. → ACK of FIN

## WSP/WTP/UDP

1. → **Data Request**
2. ← ACK, **Reply**
3. → ACK, **Data Request**
4. ← ACK, **Reply**
5. → ACK, **Data Request**
6. ← ACK, **Reply**
7. → ACK

*Bold packets contain payload  
Non-bold items are overhead*

### Typical Handset Session — 3 Requests, 3 Responses

#### HTTP/TCP/IP

**17 packets**

**65% Overhead\***

#### WSP/WTP/UDP

**7 packets**

**14% Overhead\***

*\*does not account for DNS, SSL, Authentication or Cookies*



# WAE Wireless Application Environment

## ■ Environnement du User Agent

- WML Wireless Markup Language
  - documents hypertextuels pour mobile
- WMLScript
  - langage léger de script ( ressemble au Javascript )
- Autres Formats de Données
  - image, agenda téléphonique (vCard), calendrier (vCal), ...

## ■ Générateur de contenu

## ■ Encodage Standard du Contenu

- Bytecode WML et WMLScript

## ■ WTA, WTAI Wireless Telephony Application

- interfaces programmables des service téléphonique

# Mise en Œuvre de WDP sur le GSM



- SMS
- USSD
- Circuit commuté (circuit switched)
- GPRS (Commutation de Paquets)

# WML *Wireless Markup Language*

## ■ Document WML

- **DTD** : [http://www.wapforum.org/DTD/wml\\_1.1.xml](http://www.wapforum.org/DTD/wml_1.1.xml)
- « jeu de cartes » desk of cards
- Éléments dans les cartes
  - Tables, images (wbmp), ancres (navigation entre cards)
- Fonctions WMLScript (appelé par les commandes, timer ...)

## ■ 2 formes

- textuelle (text/vnd.wap.wml )
- bytecode (application/vnd.wap. wmlc)
  - compact (stockage ,transfert)
  - simplifie l'automate du client (CPU)

# Un premier exemple

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Weather Forecast">
    <p>
      <table columns="3" align="LCC">
        <tr><td>Date</td><td>F&apos;cast</td><td>T &#xB0;C</td></tr>
        <tr><td>M 6/7</td><td></td>
          <td>25&#xB0;C</td></tr>
        <tr><td>T 6/8</td><td>
          </td><td>27&#xB0;C</td></tr>
        <tr><td>W 6/9</td><td></td>
          <td>24&#xB0;C</td></tr>
        <tr><td>T 6/10</td><td></td>
          <td>28&#xB0;C</td></tr>
        <tr><td>F 6/11</td><td></td>
          <td>29&#xB0;C</td></tr>
      </table>
    </p>
  </card>
</wml>

```



# Le document WML

- Analogie au jeu (desk) de plusieurs cartes (card)
  - Chaque carte constitue une page d'affichage sur le UA
  - L'utilisateur peut naviguer d'une carte à l'autre (sans téléchargement)

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<card id="card1" title="Carte 1"><p>Carte 1</p></card >
```

```
<card id="card2" title="Carte 2"><p>Carte 2</p></card >
```

```
<card id="card3" title="Carte 3"><p>Carte 3</p></card>
```

```
</wml>
```

# L'élément <wml>

## ■ Représente un document

## ■ Contient

- Un entête <head>
  - Contient des méta-informations <meta> et des règles d'accès <access>
- Un patron <template> pour les cartes
  - Factorise des commandes <do>  
ou les traitements des événements <onevent>  
entre toutes les cartes
- Une suite de <card>

# L'élément <card>

## ■ Contient

- Des paragraphes <p> à afficher
- Des commandes <do> à réaliser sur actions des entrées
- Des traitements sur événement <onevent>
- Un déclenchement de temporisateur <timer>

## ■ Attributs

- newcontext
  - =true, les variables sont relâchées, l'historique est supprimé
- title
- ordered
- onenterforward, onenterbackward, ontimer

# Mise en page d'une Carte

## ■ Élément paragraphe <p>

- Attribut align=left|center|right
- Attribut mode=wrap|nowrap

<card id="card1" title="Carte 1">

<p>paragraphe a gauche par default</p>

<p align="center"> paragraphe centre</p>

<p align="right"> paragraphe a droite</p>

<p mode="nowrap"> paragraphe sans passage a la ligne </p>

</card >

## ■ Élément de rupture de ligne <br/>

<p>rupture<br/>de ligne</p>

## ■ Propriétés des caractères

<i>votre texte en italique</i><b>votre texte en gras</b>

<small>votre texte en petit</small><big>votre texte en grand</big>

<u>votre texte souligné</u>



# Les entités WML

■ "	&quot;	■ à	&#xE0;
■ &	&amp;	■ é	&#xE2;
■ '	&apos;	■ è	&#xE9;
■ <	&lt;	■ ê	&#xEA;
■ >	&gt;	■ ù	&#xFF9;
■ [Espace]	&nbsp;	■ û	&#xFB;
■ -	&shy;	■ ç	&#xE7;
		■ ©	&#xA9;
		■ ®	&#xAE;

# Table

## ■ Exemple de table 3x3

```
<table columns="3" title="matrix" align="RCL">  
  <tr> <td>1</td><td>2</td><td>3</td> </tr>  
  <tr> <td>4</td><td>5</td><td>6</td> </tr>  
  <tr> <td>7</td><td>8</td><td>9</td> </tr>  
</table>
```

## ■ Attributs

- **columns** est obligatoire : simplifie l'interpréteur
- **title** (optionnel) peut être affiché par le navigateur
- **align** (optionnel) décrit l'alignement de chaque colonne
  - L pour Left, R pour Right, C pour Center

# Les Images

## ■ WML supporte l'inclusion d'images

```

```

```
<a href="#card2"></a>
```

Le UA peut supporter plusieurs formats (PNG, BMP)  
cependant le format WBMP (Wireless BitMaP) est optimisé  
pour le Wireless (convertisseur JPEG,GIF,BMP vers WBMP)

## ■ Pictogrammes

- Diffère entre les UA en WAP 1.0
  - l'application n'est pas portable
- Désormais normalisé dans WAP2.0

# Variable

## ■ Déclaration <setvar>

```
<setvar name="location" value="http://wap.mycomp.com#sommaire">
<do label="Sommaire"><go href="$(location)"/></do>
```

## ■ Substitution de variable et échappement

```
<setvar name="var" value="Wap, do wap">
<setvar name="escvar" value="Wap%2C+do+wap">
```

- This is a \$var. sans conversion
- This is a \$(var). sans conversion
- This is a \$(var:n). sans conversion
- This is a \$(var:noesc). sans conversion
- This is an escaped \$(var:e). Wap%2C+do+wap
- This is an escaped \$(var:escape). Wap%2C+do+wap
- This is an unescaped \$(escvar:unesc). Wap, do wap
- Remarque : l'échappement sert au paramétrage des URL

# Les taches (task)

## ■ action à réaliser par le navigateur

- en réponse à un événement `<onevent>`
- à une interaction avec l'utilisateur `<do>`

## ■ `<go>`

- Navigue vers une nouvelle carte

## ■ `<prev>`

- Retourne à la carte précédente dans l'historique
- Après avoir effectué l'affectation de variables

## ■ `<refresh>`

- Effectue l'affectation des variables `<setvar>`
- Réafficher la carte courante (avec les nouvelles substitutions de variables)

## ■ `<noop>`

- Ne fait rien (« no operation »)

# La tâche <go>

- Navigue vers une nouvelle carte

## ■ Attributs

- href
  - l'URL cible
- method
  - la méthode utilisée pour l'envoi des paramètres : GET (par défaut) et POST
- sendreferer
  - si true, le navigateur inclut dans les champs de la réponse
- accept-charset
  - encodage des paramètres de la requête POST

## ■ Élément <postfield>

- spécifie les paramètres de la requête (GET ou POST)

# Exemple de tache <go>

## ■ avec <setvar>

```
<go href="$(page).wml">  
  <setvar name="page" value="bad"/>  
</go>
```

- si page="login", la carte courante devient login.wml et page==bad

## ■ avec <postfield>

```
<go href="login.cgi?l=$(login)&p=$(password)" method="get"/>
```

```
<go href="login.cgi?l=$(login)" method="get">  
  <postfield name="p" value="$(password)"/>  
</go>
```

```
<go href="login.cgi" method="get">  
  <postfield name="l" value="$(login)"/>  
  <postfield name="p" value="$(password)"/>  
</go>
```

# La tache <prev>

## ■ <prev>

- Retourne à la carte précédente dans l'historique
- Après avoir effectuer l'affectation de variables

## ■ Exemple

```
<prev/>
```

```
<prev>
```

```
<setvar name="password" value=""/>
```

```
<prev/>
```

```
<do type="prev"><prev/></do><noop/>
```

- Généralement associé à l'action du bouton de retour arrière BACK



# La tache <refresh>

- Effectue l'affectation des variables <setvar>
- Puis réafficher la carte courante  
(avec les nouvelles substitutions de variables)

## ■ Exemple

```
<onevent type="onenterbackward">  
  <refresh>  
    <setvar name="login" value=""/>  
    <setvar name="password" value=""/>  
  </refresh>  
</onevent>
```

# Les événements <onevent>

- Déclarés en début de la carte (ou du template)
- Réalisent des tâches <go>, <refresh>, ...

## ■ Événements de <card> et <template>

- onenterforward
  - Déclenché lors de l'entrée dans la carte (<go>, <anchor>, bookmark, ...)
- onenterbackward
  - Déclenché lors du retour (historique) dans la carte (<prev>)
- ontimer
  - Déclenché sur expiration du timer de la carte

## ■ Événements de <card> et <template>

- onpick
  - Déclenché sur sélection d'une <option>

# Exemple d' <onevent>

```

<card title="Reset on reentry">
  <onevent type="onenterforward">
    <refresh>
      <setvar name="login" value=""/>
      <setvar name="password" value=""/>
    </refresh>
  </onevent>
  <onevent type="onenterbackward">
    <refresh>
      <setvar name="password" value=""/>
    </refresh>
  </onevent>
  <p>Login:  <input name="login"/></p>
  <p>Password: <input name="password"/></p>
  <do type="accept" label="Submit">
    <go href="login.cgi" method="post">
      <postfield name="l" value="$(login)"/>
      <postfield name="p" value="$(password)"/>
    </go>
  </do>
</card>

```

# Les interactions avec l'utilisateur

## ■ Les ancres

- équivalent aux ancres HTML
- **<a>**
- **<anchor>**

## ■ Les commandes

- équivalent aux boutons des formulaire HTML
- **<do>**

## ■ Les champs de saisie

- équivalent aux champs de formulaire HTML
- **<input>** : entrée textuelle
- **<select>+<option>** : liste de choix (unique ou multiple)

# Les ancres <a> et <anchor>

## ■ Notion d'hyperlien

## ■ Ancre vers une autre Card du document courant

```
<a href="#card2">Vers page 2</a>
```

```
<a title="page2" href="#card2">Vers page 2</a>
```

## ■ Ancre vers un autre document

```
<a href="prod.wml">Nos Produits</a>
```

```
<a href="prod.wml#cardnew">Nos Nouveaut &#xE2;s</a>
```

```
<a title="catalog" href="http://wap.mycomp.com/catalog">Nos produits</a>
```

## ■ La forme <anchor>

```
<anchor title="next"><go href="page10.wml"/>Next Page</anchor>
```

est équivalent à

```
<a title="next" href="page10.wml">Next Page</a>
```

# Les ancrs <a> et <anchor>

- Déclenchement de l'exécution de fonction WMLScript  
`<a href="func.wmls#Catalog(1,10)">Autres</a>`  
`<a href="http://wap.mycomp.com/func.wmls#Catalog(1,10)">Autres</a>`
- Appel au composeur de Email  
`<a href="mailto:webmaster@mycomp.com">Envoyer un email</a>`
- Composition automatique d'un numéro de téléphone  
`<a href="wtai://wp/mc;+33601234567">Composer le numéro</a>`

# Les commandes <do>

- Correspond à des actions de l'utilisateur
  - (sur son clavier, écran tactile, ...)
- Désignation par un type `<do type="...">`
  - **accept**, **prev**, **help**, **reset**, **options**, **delete**
  - **unknown** correspond au traitement d'un type inconnu de l'UA
  - **X-??** correspond à un type expérimental
  - **vnd.??** correspond à un type propriétaire/vendeur (de l'UA)
  - Associé par l'UA à une touche, bouton graphique (dépend de l'UA)
- Exemples
  - Navigation dans l'historique  
`<do type="prev" label="Retour"><prev/></do>`
  - Navigation vers une autre carte  
`<do label="Sommaire"><go href="http://wap.mycomp.com#sommaire"/></do>`
  - Mise à zéro d'un formulaire  
`<do type="refresh" label="Recommencer"><refresh/></do>`

# Champs de saisie

## Entrée textuelle <input>

### ■ Attribut

- key=nmtoken nom de la variable
- default=vdata, value=vdata, type=(text|password), format=cdata
- emptyok=boolean obligatoirement saisie par l'utilisateur
- size=number, maxlength=number, tabindex=number, title=vdata

### ■ Exemple

```
<card title="Login">
  <p>Username: <input name="user" format="*x"/></p>
  <p>Password: <input name="pwd" type="password"/></p>
  <do type="accept" title="Log In">
    <go href="login?u=$(user:e)&p=$(pwd:e)"/>
  </do>
</card>
```



# Champs de saisie

## Liste `<select>` de choix `<option>`

### ■ Attribut de select

- `key=nmtoken`, `default=vdata`
- `ikkey=nmtoken` (0 si aucune option n'est sélectionnée)
- `idefault=vdata`, `mutiple=boolean`, `tabindex=number`, `title=vdata`

### ■ Attribut d'`<option>`

- `value=vdata`, `title=vdata`, `onpick=url`

### ■ Exemple

```
<card title="Pizza Toppings" id="toppings">
```

```
<p>Choose your toppings:
```

```
<select name="toppings" multiple="true">
```

```
<option value="p">Pepperoni</option>
```

```
<option value="b">Spicy Beef</option>
```

```
<option value="a">Anchovies</option>
```

```
<option value="c">Chillies</option>
```

```
<option value="o">Olives</option>
```

```
<option value="m">Mushrooms</option>
```

```
</select></p>
```

```
<do type="accept" label="Order"><go href="order?toppings=$(toppings:e)"/></do>
```

```
</card>
```

# Champs de saisie

## Liste <select> de choix <option>

### ■ Exemple des options à choix multiples

```

<card title="Set Pizzas" id="pizzas">
  <p>Choose a Pizza:
  <select name="toppings">
    <option value="">Plain</option>
    <option value="p;m;o">Pepperoni Supreme</option>
    <option value="p;b;h;a;m;o;c">Full House</option>
    <option value="c;o;m">Vegetarian</option>
  </select></p>
  <do type="accept" label="Order">
    <go href="order?toppings=$(toppings:e)"/>
  </do>
  <do type="accept" label="Customize"><go href="#toppings"/></do>
</card>

```

# Champs de saisie

## Groupage <optgroup> des choix

- Groupe des options lors de leur affichage par le navigateur
  - Quand la liste est longue, le navigateur permet alors les afficher par groupe
- Exemple

```
<card title="Pizza Toppings"> <p>Choose your toppings:  
  <select name="toppings" multiple="true">  
    <optgroup title="Meat & Fish">  
      <option value="p">Pepperoni</option>  
      <option value="b">Spicy Beef</option>  
      <option value="a">Anchovies</option>  
    </optgroup>  
    <optgroup title="Vegetable">  
      <option value="c">Chillies</option>  
      <option value="o">Olives</option>  
      <option value="m">Mushrooms</option>  
    </optgroup>  
  </select></p>  
  <do type="accept" label="Order"><go href="order?toppings=$(toppings:e)"/></do>  
</card>
```

# Champs de saisie

```

<card>
  <do type="accept">
    <go href="/submit?f=$(fname)&l=$(lname)&s=$(sex)&a=$(age)"/>
  </do>
  <fieldset="name">
    First name: <input type="text" key="fname" maxlength="32"/><br/>
    Last name: <input type="text" key="lname" maxlength="32"/><br/>
  </fieldset>
  <fieldset title="info">
    <select key="sex">
      <option value="f">Female</option>
      <option value="m">Male</option>
    </select>
    <br/>
    Age: <input type="text" key="age" format="*N"/>
  </fieldset>
</card>

```

A : majuscule  
 a : minuscule  
 N: nombre  
 M: caractère quelconque  
 \*A: 0 à N majuscules  
 8N: 8 nombres

# Temporisateur <timer>

## ■ Temporisateur associé à une carte

- La valeur de la temporisation est exprimée en 1/10 de secondes

## ■ Exemple

- redirection vers une page sur écheance d'un timer

```
<card ontimer="http://wap.mycomp.com/index"
      newcontext="true" title="Chargement en cours ...">
  <timer value="30"/>
</card>
```

```
<card newcontext="true" title="Chargement en cours ...">
  <onevent type="ontimer"><go href="http://wap.mycomp.com/index"/></onevent>
  <timer value="30"/>
</card>
```

## ■ Exercice :

- comment émuler une image animée ?

# Exemple de temporisateur

## L'image animée

```

<wml> <card title="Animate Three Images">
  <onevent type="onenterforward">
    <refresh>
      <setvar name="img0" value="first.wbmp"/><setvar name="time0" value="5"/>
      <setvar name="img1" value="second.wbmp"/><setvar name="time1" value="10"/>
      <setvar name="img2" value="third.wbmp"/><setvar name="time2" value="10"/>
    </refresh>
  </onevent>
  <onevent type="ontimer">
    <refresh>
      <setvar name="imgtmp" value="$(img0)"/><setvar name="timetmp" value="$(timetmp)"/>
      <setvar name="img0" value="$(img1)"/><setvar name="time0" value="$(time1)"/>
      <setvar name="img1" value="$(img2)"/><setvar name="time1" value="$(time2)"/>
      <setvar name="img2" value="$(imgtmp)"/><setvar name="time2" value="$(timetmp)"/>
    </refresh>
  </onevent>
  <timer value="$(time0)"/>
  <p></p>
</card></wml>

```

# Exemple de temporisateur

```
<wml>
  <card title="First card" id="one" ontimer="#three">
    <timer name="timeout" value="600"/>

    <p>The timer is running...</p>
    <do type="" label="Check"><go href="#two"/></do>
  </card>

  <card title="Second card" id="two">
    <p>Time remaining:
      $(timeout) tenths of a second</p>
    <do type="prev" label="Back"><prev/></do>
  </card>

  <card title="Third card" id="three">
    <p>Timer expired!</p>
  </card>
</wml>
```

# Template (patron)

- Mise en commun d'événements et des commandes (first, fourth)
- Peuvent être masqués (second) ou remplacé (third)

## ■ Exemple

```
<wml>
  <template>
    <do type="options" name="do1" label="default"><prev/></do>
  </template>
  <card id="first">Premier</p></card>
  < card id="second"><p>Second</p>
    <do type="options" name="do1"><noop/></do>
  </card>
  <card id="third"><p>Troisieme</p>
    <do type="options" name="do1" label="option"><go href="/option"/></do>
  </card>
  <card id="fourth"><p>Quatrieme</p></card>
</wml>
```

Le bouton est masqué

Le bouton est remplacé



# WML

## *Exemple de Formulaire*

```
<wml>
  <card name="homepage" title="Send Email">
    <p>To:</p>
    <p><input name="email"/><br/> </p>
    <p> Sender's Email Address:</p>
    <p><input name="fr"/><br/> </p>
    <p>Subject:</p>
    <p><input name="sub"/><br/></p>
    <p>Message:</p>
    <p><input name="msg"/><br/></p>
    <do label="Send Email" name="Send Email" type="ACCEPT">
      <go href="wap2.cgi?query=email3&str=,,&"/></do>
    <do label="Main Menu" name="Main Menu" type="ACCEPT">
      <go href="http://vcom.edge.com.sg/wap/wap2.cgi"/></do>
    <do label="Back" type="prev"><prev/></do></card>
</wml>
```

# WML

## *Exemple de Banque Nomade*

```
<wml>
  <card name="homepage" title="Banque Nomade">
    <p>Selectionner un service:</p>
    <p><a href="service?id=cpt">1. Compte Courante</a><br/> </p>
    <p><a href="service?id=peL">2. P.E.L.</a><br/> </p>
    <p><a href="service?id=peA">3. P.E.A.</a><br/> </p>
    <p><a href="service?id=trade">4. La bourse</a><br/> </p>
    <do label="Main Menu" name="Main Menu" type="ACCEPT">
      <go href="http://wap.mobibank.com/"></do>
    <do label="Back" type="prev"><prev/></do>
  </card>
</wml>
```

# WMLScript

## ■ Langage de script (côté client)

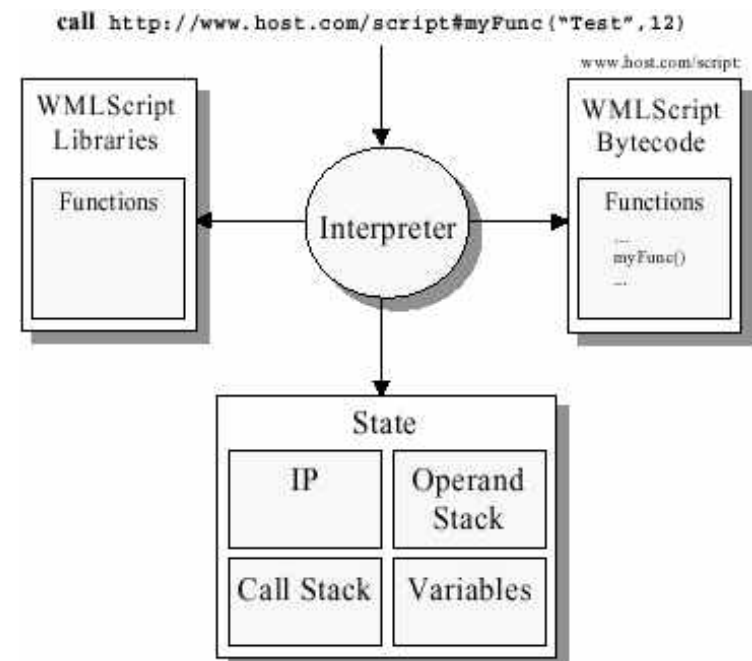
- Syntaxe ECMAScript
- format textuel (.wmls / text/vnd.wap.wmlscript)
- format bytecode (.wmlsc / application/vnd.wap.wmlscript)

## ■ Motivation

- augmente l'interactivité (temps de réponse)
- diminuer les interactions avec le serveur

## ■ L'interpréteur de l'UA

- Interprète du bytecode
  - Du fichier
  - De bibliothèques
    - Standard
    - Dépendante de l'UA
    - Dépendante du Bearer



# Types de données

## ■ Types de Données

- **boolean (deux valeurs true et false)**
- **integer**
- **float**
- **string**
  
- **invalid (indique que l'expression est invalide)**

## ■ Remarque

- pas de tableau

# Opérateurs

## ■ Arithmétiques

- `+`, `-`, `*`, `/`, `div`, `%`, `++`, `--`, `+=`, `-=`, `div=`, ...

## ■ Bitwise

- `>>`, `<<`, `>>>`, `<<<`, `|`, `&`

## ■ Comparateurs

- `>`, `<`, `==`, `!=`, `<=`, `>=`

## ■ Conditionnels

- `?:`, `||`, `&&`

# Conversion de type et validité

## ■ Conversion de type

<code>2&lt;"10"</code>	<code>"2"&lt;"10"</code>	boolean	false
<code>"FOO"&lt;"foo"</code>		boolean	true
<code>"x"+"y"</code>		string	"xy"
<code>1+2</code>		integer	3
<code>1+2e0</code>		float	3.0
<code>"2"-9e0"</code>		float	-7.0
<code>2&lt;"10"</code>	<code>"2"&lt;"10"</code>	boolean	false
<code>-"1e999"</code>		invalid	
<code>(17+1/0)</code>		invalid	

## ■ Opérateurs de type

- `typeof` retourne 5 valeurs possibles
- integer 0, float 1, string 2, boolean 3, invalide 4

## ■ Validité

- `if (typeof result != 4) { ... }`
- `if (isvalid result) { ... }`

# Variables

## ■ Variables Locales

- Déclaration mais typage dynamique  
`var x=10; x="bonjour";`
- Durée de vie : la fonction
- Portée : la fonction (pas de niveau de blocs)

```
function fn(x) {  
  if(x){ var y=10; }  
  return y;  
}
```

## ■ Variables Globales

- Fonctions de bibliothèque `WMLBrowser.getVar` et `WMLBrowser.setVar`

```
function check(s) {  
  var secret=WMLBrowser.getVar("secret");  
  if(s==secret) {WMLBrowser.setVar("secret",s);  
}
```

# Structures de Contrôle

**bloc {}, if else, while, for, break, continue, return**



# Fonctions

- Définition de Fonction

```
function check(par1, par2) { return par1==par2; };
```

- Appels de fonction

```
if(check(enterpassword,"toto")){ ... }
```

- Appels de fonction d'un autre fichier

```
if(script.wmls#check(enterpassword,"toto")){ ... }
```

- Utilisation de NameSpace

```
use url ScriptLib "http://www.script.com/lib/script.wmls";  
function test(par1, par2) { return ScriptLib#check(par1,par2); };
```

- Appels de fonction de bibliothèque

```
var msg="hello"; var len=String.length(msg); msg=msg+ ' world';
```

- Appels de fonction en URL

```
<go href="http://wap.mycomp.com/func.wmls#Catalog(1,10)">
```

# Pragmas

- équivalent à <meta> et <access>

## ■ use access

- autorise l'accès si l'URL du référant correspond
- à l'expression du chemin et/ou à l'expression du domaine  
**use access domain "mycomp.com" path "/private"**

## ■ use meta

- ajoute des méta-informations arbitraires
- *use meta name value scheme*
  - ajouté par le serveur  
**use meta name "last-modified" "2001-04-20";**
- *use http equiv name value scheme*
  - spécifie un champ de l'entête HTTP ou WSP  
**use http equiv "cache-control" "no-cache";**
- *use user agent name value scheme*
  - passé à l'interpréteur WMLScript  
**use meta name "x-debugging" "on";**

# Exemple : Quizz

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="quizz" title="Quizz...">
<p>Que veut dire WAP ? <br/>
<select name="reponse" multiple="false" >
<option value="1">Waou Api Po</option>
<option value="2">Wireless Application Protocol</option>
<option value="3">What Apended Pentagone</option>
<option value="4">Wil Affer Patereted</option>
</select></p>
<do type="accept" label="Verifier..."><go href="quizz.wmls#verification('$reponse')"/></do>
</card>
<card id="bonnereponse" title="Resultats">
<p>Bonne réponse ! Bravo, vous etes un connaisseur.<br/></p>
</card>
<card id="mauvaisereponse" title="Resultats"><p>Vous avez faux.<br/>Recommencez</p>
</card></wml>

```

# Exemple :

## Convertisseur Franc-Euro (WML)

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="euoconvert" title="EuroConvert" newcontext="true"><p>
    Somme: <input format="*N" name="somme" title="Somme:"/>
    Convertir des: <select name="des" value="FRA" title="Des:">
        <option value="FRA">Francs</option>
        <option value="EUR">Euros</option>
    </select>
    En: <select name="en" value="EUR" title="En:">
        <option value="EUR">Euros</option>
        <option value="FRA">Francs</option>
    </select>
    <br/> = <u>$(conversion)</u>
<do type="accept" label="Calculer...">
    <go href="euroconvert.wmls#convert('conversion','$(des)','$(en)',$(somme))"/>
</do>
</p>
</card></wml>

```

# Exemple :

## Convertisseur Franc-Euro (WMLScript)

```
extern function convert(varName,des,en,somme) {
    var multiplier = 0.0;
    var returnString = "Erreur !";
    var result;
    if (des == "FRA") {
        var FRA_EUR = 6.55;
        if (en == "FRA") multiplier = 1.0;
        else if (en == "EUR") multiplier = FRA_EUR;
    } else if (des == "EUR") {
        var EUR_FRA = 0.1526;
        if (en == "EUR") multiplier = 1.0;
        else if (en == "FRA") multiplier = EUR_FRA;
    }
    if (multiplier != 0.0) {
        result = somme / multiplier;
        returnString = String.toString(result);
        returnString = String.format("%.2f", returnString);
    }
    WMLBrowser.setVar(varName,returnString);
    WMLBrowser.refresh();
}
```

# Exemple : Password

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="Mot-de-passe" newcontext="true">
    <do type="options" label="Verifier">
      <go href="password.wmls#verif('${URL:noesc}')"/>
    </do>
    <p align="center">Entrez le mot-de-passe:<input type="text" name="URL"/><br/></p>
  </card>
</wml>
```

```
// password.wmls
extern function verif(password){
  if (String.find(password, ".wml") < 0){ password=password+".wml"; }
  WMLBrowser.go(password);
}
```

# Bibliothèque standard de fonctions

## ■ Lang

- abs, min, max, parseInt, parseFloat, random, seed, exit, abort, ...

## ■ Float

- int, floor, ceil, pow, ...

## ■ String

- length, isEmpty, replace, elementAt, trim, format, ...

## ■ URL

- isValid, getHost, getScheme, getParameters, resolve, escape, unescape, ...

## ■ WMLBrowser

- getVar, setVar, go, prev, newContext, getCurrentCard, refresh

## ■ Dialog

- prompt, confirm, alert

# Bibliothèque WTAI de fonctions

## ■ Wireless Telephony Application Interface

- Permet de piloter les services de telephonie ou du réseau

## ■ Une commune

- WTACallCont
  - setup, accept, release, sendDTMF
- WTANetText
  - send, read, delete, getFieldValue
- WTAPhoneBook
  - write, read, delete, getFieldValue
- WTAMisc
  - indication, endcontext

## ■ Les autres spécifiques aux Bearers

- WTAI GSM, ...

## ■ Invocation des fonctions avec des URI spéciales

- `WTACallCont.setup(";+336012345678",1)` `wtai://cc/sc;"+336012345678";1`
- `WTAGSM.transfer("1","+336012345678")` `wtai://gsm/ct;1;"+336012345678"`



# Configuration des serveurs et Types MIME

## ■ Serveurs (MS IIS, Apache, ...)

- Documents statiques (WML, WMLScript, ...)
- Documents dynamiques (Servlet, JSP, ASP, PHP, ...)

## ■ Types MIME associés

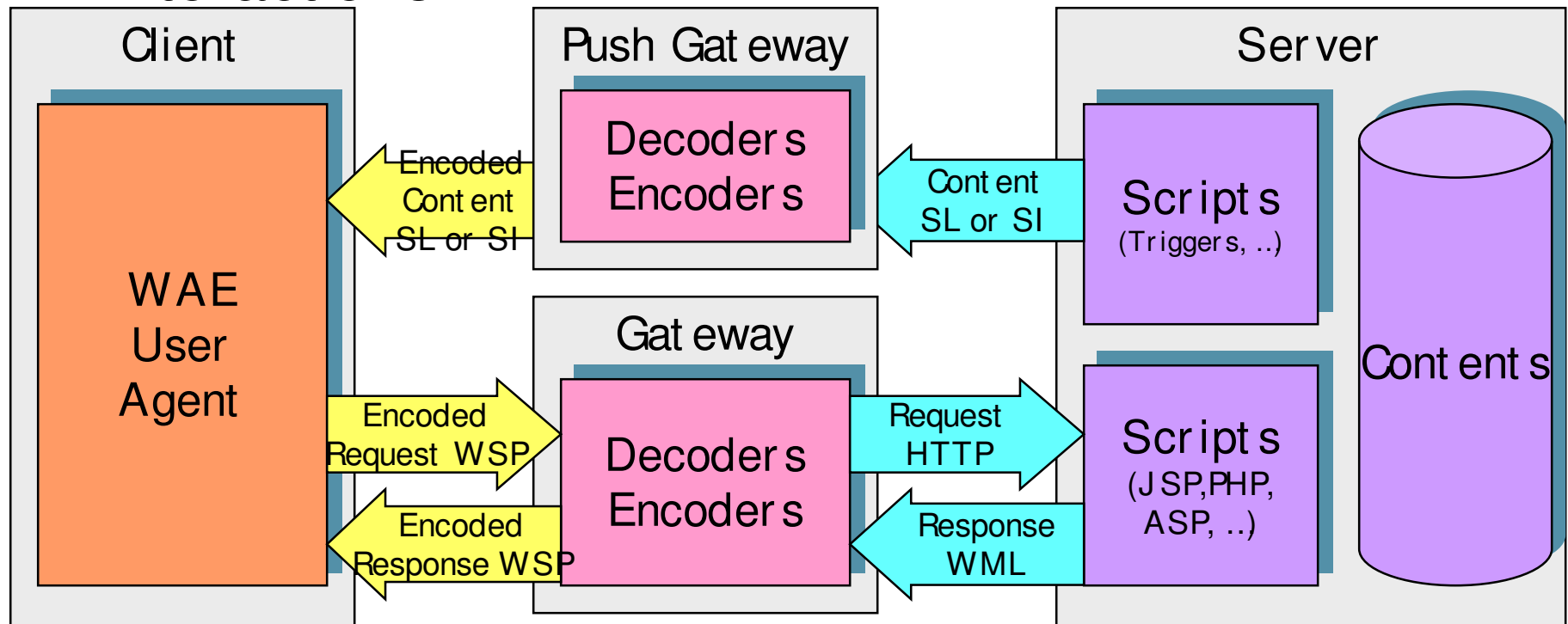
- Fichier WML (extension wml)  
`text/vnd.wap.wml`
- Fichier compilé (extension wmlc)  
`application/vnd.wap.wmlc`
- Fichier script (extension wmls)  
`text/vnd.wap.wmlscript`
- Fichier script compilé (extension wmlsc)  
`application/vnd.wap.wmlscriptc`
- Image WBMP (extension wbmp)  
`image/vnd.wap.wbmp`

# Le Push

## ■ Notification asynchrone d'événement au UA

- Alertes, Mises à jour, ...
- Nouveaux Email, Changement cours boursier, enchères, ...

## ■ Interactions



# Service Indication (SI)

## ■ Envoi d'une alerte

- Le message reçu est affiché à l'utilisateur
- L'utilisateur peut choisir de charger la page WML référencée dans le message

## ■ Type MIME

- `text/vnd.wap.si` et `application/vnd.wap.sic`

## ■ Exemple

```
<?xml version="1.0"?>
```

```
<!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"  
  "http://www.wapforum.org/DTD/si.dtd">
```

```
<si>
```

```
<indication href= "http://stockquote.com/welcome.wml"  
  si-id= "cust-msg-num52" created= "2000-03-31T15.28.19Z"  
  si-expires= "2000-03-31T23.59.59Z" action= "signal-medium">
```

```
View stock quotes now?
```

```
</indication>
```

```
<info><item class="MoreInfo">Your last update was Tuesday at 3 PM.</item></info>
```

```
</si>
```

# Service Loading (SL)

## ■ Notification d'une page à charger (« pull »)

- action= "execute-low" action= "execute-high"
  - Le message contient une référence que l'UA charge et affiche dans un nouveau contexte
- action= "cache"
  - la page est mise dans le cache

## ■ Type MIME

- text/vnd.wap.sl et application/vnd.wap.slc

## ■ Exemple

```
<?xml version="1.0"?>
<!DOCTYPE sl PUBLIC "-//WAPFORUM//DTD SL 1.0//EN"
    "http://www.wapforum.org/DTD/sl.dtd">
<sl href= "http://www.stockquote.com/myport.wml" action= "execute-low">
</sl>
```

# La Gestion des Caches

## ■ Principe

- Les clients, les passerelles cachent les documents reçus ou en transit

## ■ Avantages

- consultation offline des documents
- économie bande passante

## ■ Inconvénients

- incohérence dans les applications transactionnelles
  - Commande si en stock, Réservation ferme de places (sans surbooking), ...

## ■ Solution

- `<meta http-equiv="Cache-control" content="no-cache"/>`
- `use meta http equiv "Cache-control" "no-cache"`

# L'Équipe.fr

## ■ Stage DESS TNSI 99-00 (Valenciennes)

- Les objectifs : afficher des brèves sur le sport
- Les informations recueillies : des fichiers XML
- Les solutions logicielles utilisées :
  - Java 1.2.2 (servlets)
  - Microsoft IIS



# Et la carte à puce ...

## ■ Carte GSM/SIM (Subscriber Identification Module)

- SIMToolKit
- applications embarqués sur le handset GSM et exécutées par le module SIM
- la GUI du handset est pilotée par le module SIM

## ■ Cartes WIM et SWIM

- L'application WML/WMLScript dialogue avec la carte WIM
  - Authentification, Paiement électronique
  - Carnet d'adresse, Bookmark
- WAP 1.2

# Les nouveautés de WAP 2.0 (i)

- Compatibilité WAP 1.1
- Compatibilité de la pile IP
  - IP, TCP, TLS, HTTP
- Support de nouveaux porteurs
  - GPRS, W-CDMA, UMTS, ...
- Fonctionnement Push étendu
  - Temps réel, durée de vie des messages, ...
- WAE
  - XHTML, CSS Mobile Profile, SMIL
- Pictogrammes normalisés
  - Prédéfinies dans la ROM des mobiles.



# Les nouveautés de WAP 2.0 (ii)

## ■ User Agent Profile (UAProf)

- Basé sur Composite Capabilities / Preference Profiles (CC/PP) du W3C
  - Permet aux serveurs d'obtenir des informations sur le client et l'utilisateur pour préparer la réponse
  - Les informations privées ne sont communiquées qu'après acceptation de l'utilisateur

## ■ Multimedia Messaging Service (MMS)

- Solution de messagerie étendue (contenu multimédia : image, SMIL, ...) transmis par SMS ou Email (MIME). S'appuie sur le Push pour la notification et sur UAProf pour l'adaptation optimale des contenus.

# Les nouveautés de WAP 2.0 (iii)

## ■ External Functionality Interface (EFI)

- donne accès à des plugins ou à des périphériques externes (e.g. smart cards, GPS devices, health care devices and digital cameras, ...)

## ■ Persistent Storage Interface

- specifies a standard set of storage services for organizing, accessing, storing and retrieving data on the wireless device or other connected memory device.

## ■ Data Synchronization

- Échange de messages SyncML au dessus de WSP et HTTP/1.1 pour la synchronisation de données entre mobiles et stations

## ■ Provisioning

- Gestion standard des mobiles par l'opérateur.

# XHTML



- Appliquer l'extensibilité de XML à HTML4.01
  - ajout de nouveaux éléments, ...

## ■ 3 DTD définies par le W3C

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
    "DTD/xhtml1-frameset.dtd">
```

## ■ 1 DTD définie par le Wap Forum

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"  
    "http://www.wapforum.org/DTD/xhtml-mobile10.dtd" >
```

# XHTML

## Les différences avec HTML4

### ■ Document correct

- pas de chevauchement

**<p>les éléments<em>s 'emboitent</em></p>**

au lieu de

**<p>les balises<em>se chevauchent pas</p></em>**

- éléments vides

**<br/><hr/>** au lieu de **<br><hr>** qui est INCORRECT

- valeurs attributs

**<tab col="3">** au lieu de **<tab col=3>** qui est INCORRECT

### ■ Style et Script

**<script>**

**<![CDATA[**

**... unescaped script content ...**

**]]>**

**</script>**

# XHTML - Exemple

- Document XML utilisant des éléments du domaine de noms XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- initially, the default namespace is "books" -->
  <book xmlns='urn:loc.gov:books'
        xmlns:isbn='urn:ISBN:0-395-36341-6' xml:lang="en" lang="en">
    <title>Cheaper by the Dozen</title>
    <isbn:number>1568491379</isbn:number>
    <notes>
      <!-- make HTML the default namespace for a hypertext commentary -->
      <p xmlns="http://www.w3.org/1999/xhtml">
        This is also available <a href="http://www.w3.org/">online</a>.
      </p>
    </notes>
  </book>
```

# XHTML - Exemple

## ■ Document XHTML utilisant la recommandation MathML

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head><title>A Math Example</title></head>
<body>
  <p>The following is MathML markup:</p>
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply> <log/>
      <logbase>
        <cn> 3 </cn>
      </logbase>
      <ci> x </ci>
    </apply>
  </math>
</body></html>
```

# Acteurs

- WAP Forum
  - <http://www.wapforum.org>
- Open Mobile Alliance
  - <http://www.openmobilealliance.org>
- Nokia, Ericsson/Sony, Motorola, Alcatel, ...

# Outils

## ■ Client

- Motorola, Nokia, Alcatel, Ericsson, Sagem, Mutsubishi ...
- Emulateur pour PalmOS, WinCE, EPOC (Symbian)
- Emulateur sous IP et Windows
  - Nokia, Ericsson, UP (phone.com), ...

## ■ Serveur et Gateway

- Nokia, Ericsson, ...

## ■ Portail

- Oracle PortalToGo, ...

## ■ Scripts

- Servlet, JSP, ASP, XSP (voir [xml.apache.org](http://xml.apache.org))
- XML, XSLT

## ■ Editeurs

- WML, WMLC, Convertisseurs WBMP



# Outils: Nokia Mobile Internet Toolkit

<http://www.forum.nokia.com/>

The screenshot displays the Nokia Mobile Internet Toolkit (NMIT 4.0) interface. The main window shows the configuration for an MMS message. The message type is set to 'm-retrieve-conf'. The multipart headers table is as follows:

Send/Save	Header Name	Header Value
<input checked="" type="checkbox"/>	X-Mms-Message-Type	m-retrieve-conf
<input checked="" type="checkbox"/>	X-Mms-Transaction-ID	1234
<input checked="" type="checkbox"/>	X-Mms-MMS-Version	1.0

The Part Properties section shows the following headers for the selected part:

Send/Save	Header Name	Header Value
<input checked="" type="checkbox"/>	Content-Type	text/plain
<input checked="" type="checkbox"/>	Content-Location	goalinfo.txt

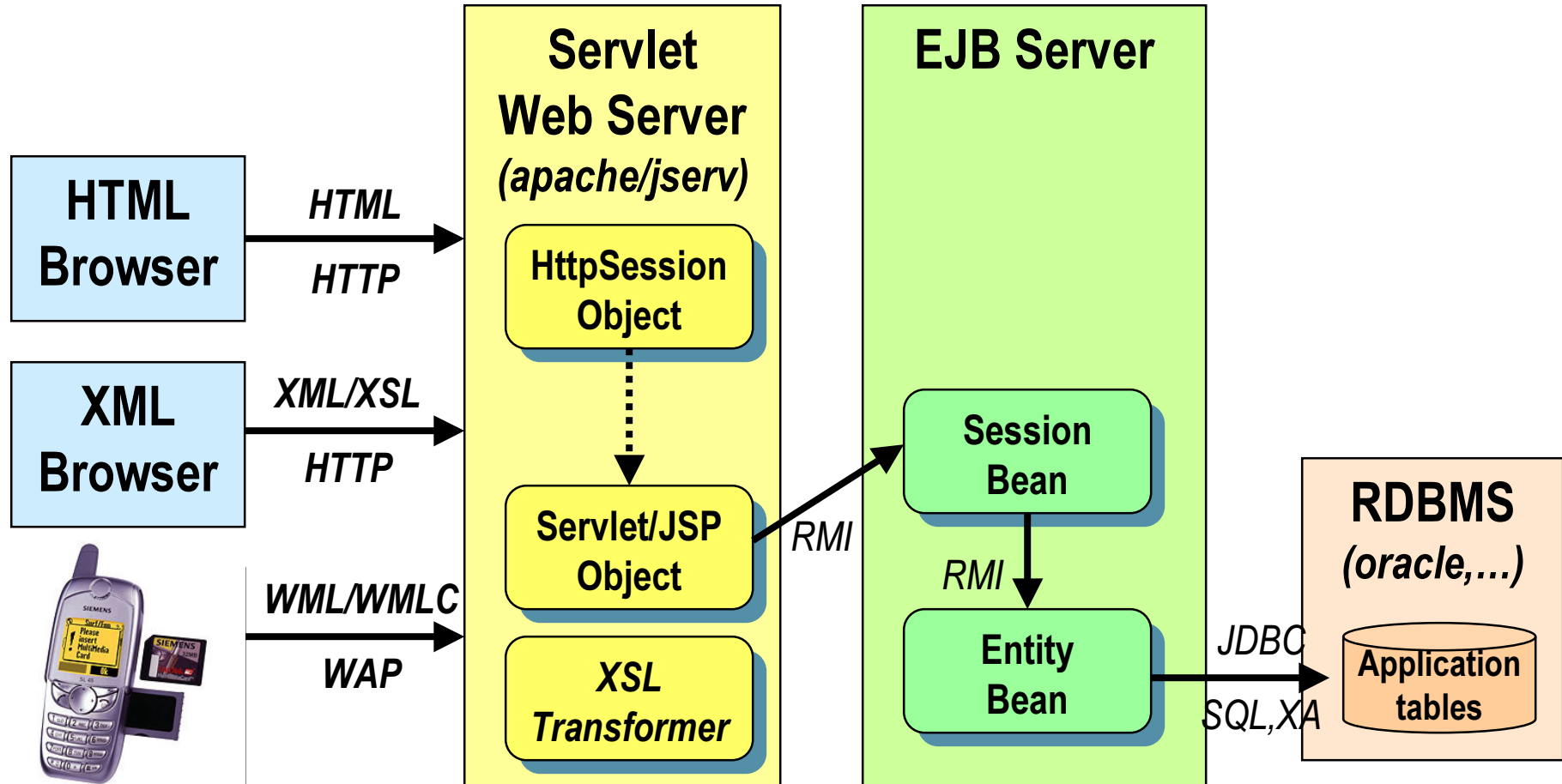
The message content is: "Ronaldo - 49th Minute  
6th Goal of World Cup". The total size is 3296 bytes (approx).

On the right, the Nokia Mobile Browser (NMB 4.0) displays a simulated mobile phone screen showing the "AlloCiné" website. The page content includes:

**AlloCiné**  
--Les films--  
[Chercher](#)  
[Nouveautés](#)  
[Comédies](#)  
[Drames](#)  
[Action Avent.](#)  
[Policier](#)  
[Thriller](#)  
[Fantast.](#)  
[Accueil](#)

Navigation buttons: Options, Retour.

# J2EE et Wireless



# Transformation XSL-T

## ■ XSL-T

- XSL-T permet la transformation d'une application XML vers une autre application XML
- La feuille de style (stylesheet) XSL-T décrit les règles de transformation

## ■ Application à WML

- WML et xHTML sont des applications de XML
- Une servlet/JSP/ASP peut répondre au client en fonction de sa capacité
  - 1) extraction des données du système d'information
  - 2) formatage dans un format pivot (ex: rowset.dtd)
  - 3) transformation avec une feuille XSL-T (rowset2wml.xsl ou rowset2html.xsl)
  - 4) application du bytecoder

## ■ Outils

- MS, Apache/Xalan, Apache/Coccon, Oracle/XSQLServlet, ...

# Concurrence

## ■ HDML (*HandHeld Markup Language*)

- Client : UP (phone.com)

## ■ Tiny HTML

- Client : Avant-Go (Push de site sur PDA)

## ■ cHTML

- Sous ensemble de HTML
- iMode, opérateur cellulaire NTT DoCoMo
  - lancé en 1999, 20 Millions d'usagers japonais en février 2001



## ■ XHTML+CSS

- HTML bien formé et valide
- Désormais dans WAP 2.0

# Concurrency : J2ME CLDC/MIDP

## ■ J2ME CLDC/MIDP (Mobile Information Device Profile)

- Two configurations :
  - Connected device config CDC : > 512 KB
  - Connected limited device config CDLC : < 512KB  
requires KVM (Kernel Virtual Machine)
- MIDlet : same principles than J2SE Applets
  - J2ME Application running in a KVM  
in a mobile phone or bidirectional pager



## ■ Pour en savoir plus

- <http://www-adele.imag.fr/~donsez/cours/j2me.pdf>

# Divers

## ■ L'UA héberge d'autres applications

- Carnet d'adresse
  - vCard
- Agenda (PIM)
  - vCalendar
- ...

## ■ Synchronisation d'application sur WAP

- Entre un serveur et un client nomade
- Applications
  - Carnet d'Adresse, Agenda, Catalogue de produits, ...

# vCal



## ■ vCalendar, the Electronic Calendaring and Scheduling Format

- Internet Mail Consortium <http://www.imc.org/pdi/vcal-10.doc>
- .vcs            text/x-vcal

## ■ Autres

- RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar)
- RFC 2446, iCalendar Transport-Independent Interoperability Protocol (iTIP): Scheduling Events, BusyTime, To-dos and Journal Entries
- RFC 2447, iCalendar Message-based Interoperability Protocol (iMIP)

# vCard



## ■ Electronic Business Card

- Internet Mail Consorsium <http://www.imc.org/pdi/vcard-21.doc>
- .vcf      text/x-vcard

### Exemple

```
begin:vcard
n:DONSEZ;Didier
tel;work:+33 3.27.51.12.34
url:http://www.univ-valenciennes.fr/LAMIH/donsez
org:UVHC;ISTV;CNRS,UMR 8530-LAMIH/ROI/SID
adr;;;Le Mont Houy;Valenciennes;;59313;France
version:2.1
email;internet:donsez@univ-valenciennes.fr
title:Maitre de conférences
fn:Didier Donsez
end:vcard
```



# Synchronisation sur WSP

## ■ Message SyncML (<http://www.syncml.org>)

- comportant des éléments décrivant une synchronisation
  - Add, Alert, Atomic, Copy, Delete, Exec, Get, Map, Replace, Search, Sequence, Sync
- pour des applications embarquées sur le handheld
- 2 représentations
  - texte WBXML `application/vnd.syncml+xml`
  - bytecode `application/vnd.syncml-wbxml`

## ■ 2 méthodes

- POST
- PUSH

# Bibliographie

## ■ Voir les cours sur HTTP, HTML, XML, JavaScript

- <http://www-adele.imag.fr/~donsez/cours>

## ■ Livres

- The WAP Forum, « The Official WAP », Ed Wiley, 1999, ISBN 0-471-32755-1
- Steve Mann, “Programming Applications with the Wireless Application Protocol, the complete developer’s guide”, 2000, Ed Wiley, 0-471-32754-9
- « Construire vos applications WAP », Ed Eyrolles, 2000 (en français)
- Professional WAP, Ed Wrox, 2000
- Martin Frost, "Learning WML & WMLScript", Ed O'Reilly, 2000, 1-56592-947-0
- SCN Education B.V., « Mobile Networking with WAP », 2000, ISBN 3-528-03149-2
- Harvey M. Deitel, Paul J. Deitel, Tem R. Nieto, Kate Steinbuhler, "Wireless Internet and Mobile Business How to Program, 1/e", 2002, 1000 pp., Prentice Hall, ISBN 0-13-009288-6,  
<http://vig.prenhall.com/catalog/academic/product/1,4096,0130092886,00.html>

# Sites

## ■ Constructeurs et Développement

- <http://www.wapforum.org>
- <http://www.openmobilealliance.org>
- <http://www.3gpp.org>
- <http://www.w3.org/Mobile>
  
- <http://www.nokia.com>
- <http://www.ericsson.com>