

WAP

Wireless Application Protocol

Alain DESEINE

Sommaire

<i>Sommaire</i>	2
<i>Table des figures, tableaux, exemples</i>	4
<i>Avant propos</i>	6
<i>Introduction</i>	7
1 Architecture de WAP	8
1.1 Le modèle du World Wide Web	8
1.2 Le modèle WAP	9
1.3 Exemple de réseau WAP	10
1.4 Sécurité	11
2 Les composantes de l'architecture WAP	12
2.1 Wireless Application Environment (WAE)	12
2.2 Wireless Session Protocol (WSP)	13
2.3 Wireless Transaction Protocol	13
2.4 Wireless Transport Layer Security (WTLS)	14
2.5 Wireless Datagram Protocol (WDP)	14
2.6 Les couches support réseau	15
2.7 Autres services et applications	15
2.8 Exemples de configurations	15
3 Wireless Application Environment	17
3.1 Fonctions générales de WAE	17
3.1.1 Interface de la couche session	17
3.1.2 Authentification	17
3.1.3 URLs	17
3.1.4 Caractéristiques des agents utilisateurs	18
3.1.5 Wireless Markup Language (WML)	20
3.1.6 WMLScript	21
3.1.7 Agents utilisateurs WAE	21
3.1.7.1 Les agents utilisateur WTA	21
3.1.7.2 Les agents utilisateurs WML	21
3.1.8 Types de média WAE	22
3.1.8.1 Format WML encodé	22
3.1.8.2 Format WMLScript encodé	22
3.1.8.3 Format carte de visite électronique	22
3.1.8.4 Format calendrier électronique	22
3.1.8.5 Format image	22
3.1.8.6 Messages multipart	23
4 Wireless Markup Language (WML)	24
4.1 La syntaxe de WML	24
4.1.1 Les « entités »	24

4.1.2	Les éléments	24
4.1.3	Les attributs	25
4.1.4	Les commentaires	25
4.2	Les variables	25
4.2.1	Erreurs	25
4.3	Exemple de document WML	26
5	WMLScript	30
6	Wireless Telephony Application	31
6.1	Architecture	31
6.1.1	L'agent utilisateur WTA	32
6.1.2	Le serveur WTA	32
6.2	Sécurité	32
6.2.1	Contrôle d'accès	33
6.2.2	Permission d'accès utilisateur	33
7	Wireless Telephony Application Interface	34
7.1	Les bibliothèques WTAI	34
7.2	Gestion des événements	34
7.3	Format d'une URI WTAI	34
7.4	Exemple d'utilisation de fonction WTAI depuis un deck WML	35
8	Les acteurs du marché	37
8.1	Les opérateurs WAP	37
8.2	Les fabricants de périphériques WAP	37
8.3	Les éditeurs de logiciels WAP	38
8.4	Les fournisseurs d'infrastructure WAP	38
8.5	Les membres du forum WAP	39
9	Les outils et plates-formes du marché	41
Annexe A	Abréviations	42
Annexe B	Glossaire	44
Annexe C	Exemples WML	48
Index		49

Table des figures, tableaux, exemples

<i>Figure 1 – Le modèle World Wide Web</i>	8
<i>Figure 2 – Le modèle WAP</i>	9
<i>Figure 3 – Exemple de réseau WAP</i>	11
<i>Figure 4 – Architecture WAP</i>	12
<i>Figure 5 – Exemples de piles WAP</i>	16
<i>Tableau 1 – Caractéristiques des agents utilisateurs</i>	20
<i>Exemple 1 – Exemple de fichier WML</i>	26
<i>Ecran 1 – Résultat de l'affichage de la première card du deck</i>	27
<i>Ecran 2 – Résultat de l'affichage de la deuxième card du deck</i>	27
<i>Exemple 2 – Exemple de fichier WML</i>	28
<i>Ecran 3 – Affichage de l'écran de saisie</i>	29
<i>Ecran 4 – L'écran de saisie sur un autre périphérique</i>	29
<i>Figure 6 – Exemple d'architecture WTA</i>	31
<i>Figure 7 – Contrôle d'accès et numéro de ports WDP</i>	33
<i>Tableau 2 – Liste des librairies WTAI</i>	35
<i>Exemple 3 – Exemple d'appel de fonction WTAI</i>	36

Avant propos

Ce document présente une vue globale du protocole WAP (Wireless Application Protocol) ainsi que les outils aujourd'hui disponibles autour de WAP. Après un tour d'horizon rapide des différents protocoles composant WAP, nous nous arrêterons un petit peu plus longtemps sur le protocole WML (Wireless Markup Language) afin d'étudier ses possibilités et ses contraintes actuelles. Nous présenterons également un état de l'art des solutions existantes actuellement en terme de fourniture de services vers des périphériques mobiles.

WAP est aujourd'hui un ensemble de protocoles émergents qui fournissent différents niveaux de services permettant à un périphérique mobile de s'interconnecter sur les réseaux TCP/IP en général, et sur Internet plus particulièrement.

A cet égard, certains outils aujourd'hui sur le marché ne se sont pas encore imposés comme standard du marché, nous essayerons donc dans la mesure du possible de présenter les outils qui nous semblent être susceptibles de le devenir, tout en présentant les autres alternatives du marché.

Toutefois, en tout état de cause, les protocoles utilisés par ces différents outils sont identiques et les outils devraient donc présenter un minimum de compatibilité entre eux. Malheureusement, à l'instar de ce qui a pu se passer au niveau des navigateurs Web, certaines disparités peuvent apparaître dans l'interprétation des différentes normes, induisant ainsi des différences gênantes pour les fournisseurs de contenus.

Introduction

L'architecture WAP définit un ensemble de protocoles réseaux et applicatifs permettant une interconnexion des réseaux pour périphériques mobiles (téléphones mobiles, pagers, PDAs, etc.) à des réseaux TCP/IP comme Internet. Le rôle des différents protocoles WAP est de créer une couche d'abstraction permettant d'utiliser ces mêmes protocoles sur des réseaux et des matériels foncièrement différents. Ainsi, WAP est implémenté sur des réseaux tels que GSM, PHS, IS-136, CDMA, CDPD, PDC-P, IDEN, FLEX, etc.

L'architecture WAP permet aux fournisseurs de contenus de bénéficier de protocoles simples à mettre en œuvre comme WML et WMLScript, tout en réutilisant des protocoles internet existants comme HTTP par exemple. Cette architecture permet donc d'une part une économie d'échelle, ainsi qu'une centralisation des contenus à destination de différents médias (navigateur Web, téléphone mobile, PDA, etc.).

De part sa structure, l'architecture WAP permet aux périphériques mobiles d'accéder à des réseaux de type TCP/IP comme Internet et/ou des réseaux privés d'entreprise par exemple, et ce quelque soit la norme de transmission de bas niveau utilisée par le périphérique.

WAP est situé à la frontière de deux mondes technologiques qui évoluent très rapidement, le monde TCP/IP et le monde des réseaux sans fil. Le challenge de WAP, est de mettre à disposition des utilisateurs de périphériques mobiles, l'immense masse d'information aujourd'hui disponible sur le Web, et de proposer des services mixtes consultables aussi bien sur le Web, que sur un périphérique mobile.

1 Architecture de WAP

1.1 Le modèle du World Wide Web

L'architecture du World Wide Web fournit un modèle de programmation simple et puissant. Les applications et le contenu sont présentés dans des formats de données standards et sont « lues » par des applications appelées « navigateurs ». Un navigateur est une application réseau de type client/serveur, c'est-à-dire qu'il envoie des requêtes pour obtenir des objets auprès d'un serveur sur le réseau, et que celui-ci lui répond en envoyant les objets demandés en utilisant des formats de données standards.

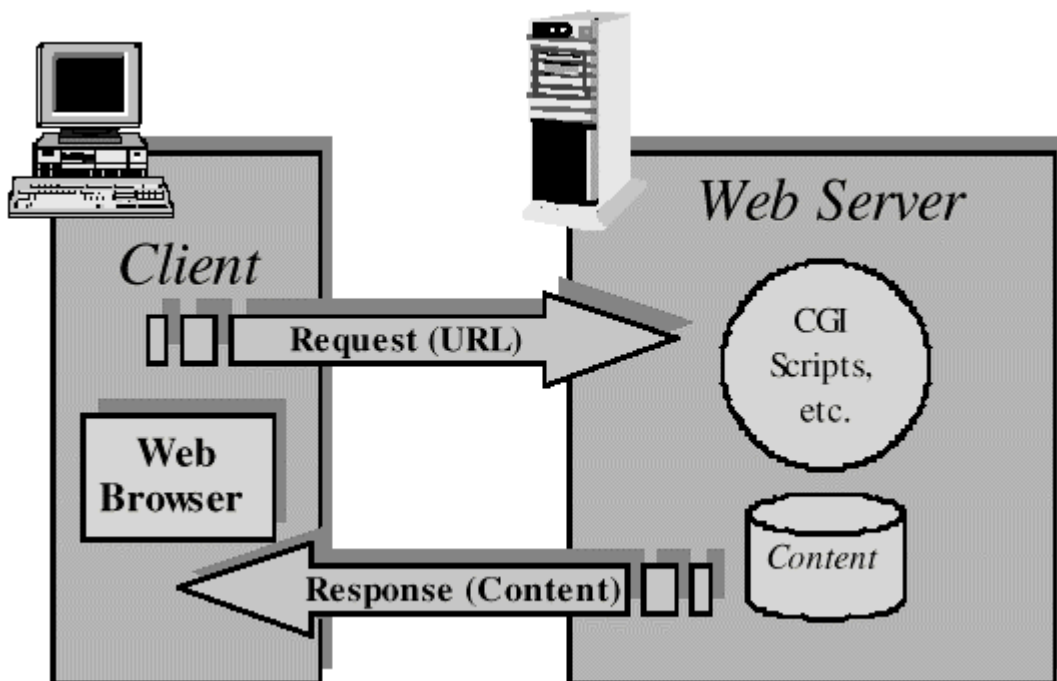


Figure 1 – Le modèle World Wide Web

Le World Wide Web fournit un grand nombre de mécanismes standards :

- Modèle de nommage standard. Tous les serveurs et tous les documents sont accessibles au travers d'une adresse appelée URL. Les URLs sont définis par les documents RFC2045 et RFC1808.
- Définition du contenu. Tout objet sur le World Wide Web se voit attribuer un « type » permettant aux navigateurs d'interpréter correctement leur contenu. Les types MIME sont définis dans les documents RFC2045 et RFC2048.
- Formats de contenu standard. Tous les navigateurs supportent un certain nombre de formats de contenu standard, comme HTML, JavaScript, etc.

- Protocole standard. Ces protocoles permettent à n'importe quels navigateurs de communiquer avec n'importe quels serveurs. Le protocole le plus utilisé est HTTP qui est défini par le RFC 2068.

Cette architecture permet aux utilisateurs d'atteindre facilement un très grand nombre d'applications et de services fournissant du contenu. Par ailleurs, elle permet également aux développeurs d'application de créer très facilement des applications et des services fournissant du contenu à un très grand nombre d'utilisateurs.

Les protocoles du World Wide Web définissent trois types de serveurs :

- Les serveurs originels, sont les serveurs sur lesquels se trouve l'objet demandé, ou sur lesquels celui-ci est créé.
- Les serveurs proxys, sont des serveurs intermédiaires qui agissent comme un serveur pour le client d'une part, et comme client du serveur originel d'autre part. Ils sont utilisés principalement dans deux cas, comme serveur cache afin de fluidifier la navigation, ou comme intermédiaire sécuritaire dans des problématiques d'accès sécurisé à des réseaux.
- Les serveurs passerelles (Gateway), sont des serveurs qui agissent comme des intermédiaires vis-à-vis d'un autre réseau. A la différence d'un proxy, un client communiquant avec une passerelle croit communiquer avec le serveur originel.

1.2 Le modèle WAP

Le modèle d'architecture WAP est similaire à celui du World Wide Web. Ceci apporte un grand nombre d'avantages aux développeurs d'applications, en leur permettant de réutiliser leurs outils et leurs modèles de programmation sur une architecture connue et stable. Des extensions, comme les langages WML et WMLScript, permettent de prendre en charge les caractéristiques spécifiques aux environnements des réseaux mobiles. Les standards existants ont été autant que possibles utilisés, ou utilisés comme point de départ pour les technologies WAP.

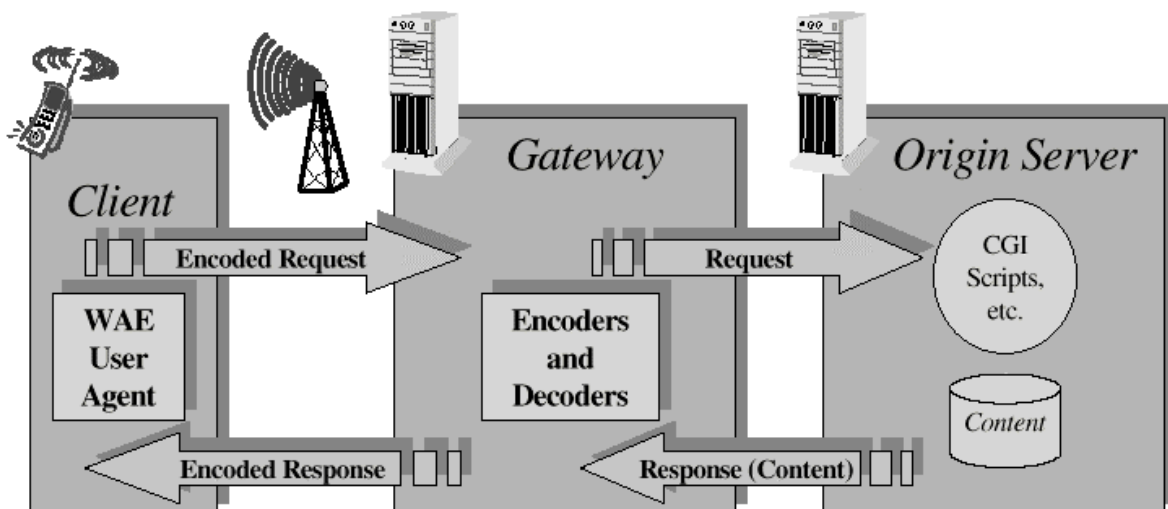


Figure 2 – Le modèle WAP

Les applications et les contenus WAP sont écrits dans des formats similaires aux formats en usage sur le World Wide Web. Ces contenus sont transportés en utilisant des protocoles de communication standards du World Wide Web. Un micro navigateur dans le terminal mobile gère l'interface utilisateur et ses analogies vis-à-vis d'un navigateur Web.

L'architecture WAP définit un certain nombre de composants standards qui permettent la communication entre les terminaux mobiles et les serveurs réseaux :

- Un modèle de nommage standard utilisant les URLs du World Wide Web est utilisé pour localiser les contenus et/ou les applications WAP. Des URIs sont utilisés pour identifier les ressources locales à un périphérique, comme par exemple des fonctions de contrôle de la numérotation.
- Un type spécifique a été attribué à tous les types de contenus WAP. Ceci permet aux navigateurs WAP de gérer correctement les contenus reçus en se basant sur leurs types.
- Un jeu de protocole de communication standard permet aux périphériques WAP de communiquer avec un serveur Web du réseau.

Les protocoles et les types de contenus WAP ont été optimisés pour les marchés de masse et pour les périphériques mobiles « mains libres ». WAP met en œuvre une passerelle permettant d'interconnecter le monde du réseau mobile avec celui du World Wide Web. Cette passerelle comprend généralement les fonctionnalités suivantes :

- Une passerelle de protocoles traduit les requêtes en provenance de la pile de protocole WAP (WSP, WTP, WTLS et WDP) vers la pile de protocoles World Wide Web (HTTP et TCP/IP).
- Un encodeur/décodeur de contenus traduit le contenu WAP en un format codé compact afin de réduire la taille des datas au travers du réseau mobile.

Cette infrastructure assure les utilisateurs de terminaux mobiles de pouvoir utiliser une grande variété d'applications et de contenus. D'autre part elle assure également l'auteur d'applications ou de contenus de pouvoir distribuer ces applications et ces contenus vers un large panel de terminaux. Ceci permet d'héberger les contenus WAP sur des serveurs Web standards en utilisant des technologies éprouvées comme les scripts CGI.

1.3 Exemple de réseau WAP

L'architecture suivante n'est qu'un exemple de ce que pourrait être un réseau WAP.

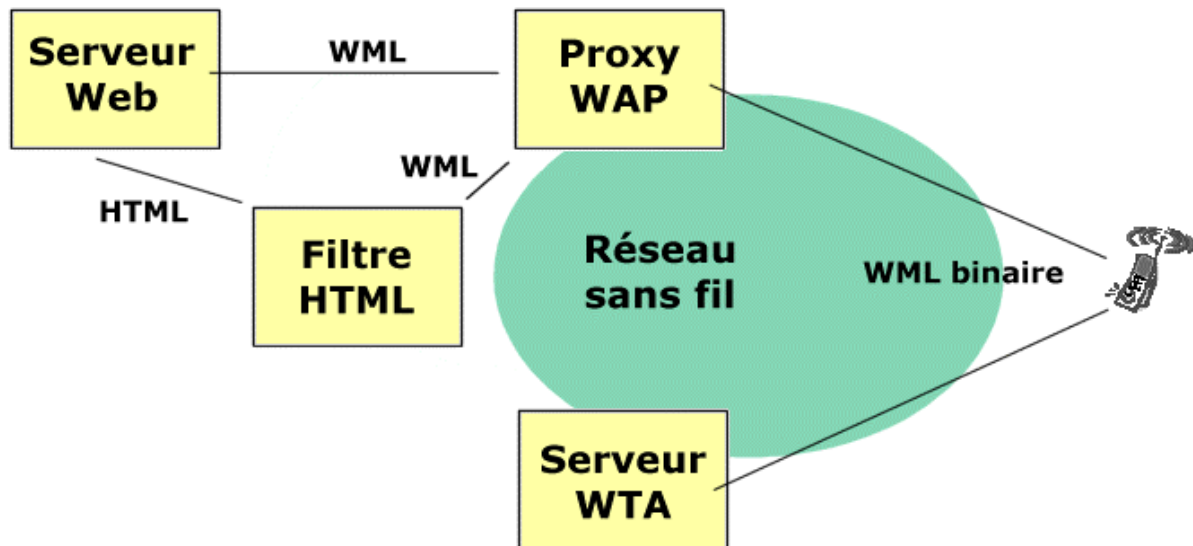


Figure 3 – Exemple de réseau WAP

Dans cet exemple le client WAP peut communiquer avec deux serveurs dans le réseau mobile. Le proxy WAP traduit les requêtes WAP en requêtes HTTP permettant ainsi au client WAP de soumettre des requêtes au serveur Web. Le proxy se charge également de coder la réponse du serveur Web dans un format binaire compact compréhensible par le client.

Si le serveur Web fournit du contenu WAP (des fichiers WML par exemple), le serveur proxy le récupère directement. Par contre si le serveur Web fournit un contenu World Wide Web (documents HTML par exemple), un filtre est utilisé pour traduire le contenu World Wide Web en contenu WAP. Par exemple le filtre transformerait un document HTML en document WML.

Le serveur WTA (Wireless Telephony Application) est un exemple de serveur originel ou de passerelle répondant directement aux requêtes émanant du client WAP. Le serveur WTA est utilisé pour fournir un accès WAP aux fonctionnalités de l'infrastructure de télécommunication du fournisseur de réseau mobile.

1.4 Sécurité

L'architecture WAP fournit une couche sécuritaire flexible qui permet de fournir une connexion sécurisée entre un client WAP et un serveur.

WAP peut fournir un niveau de sécurité entre les deux points terminaux. Si un navigateur et un serveur originel désirent mettre en place une liaison sécurisée de bout en bout, ils doivent communiquer directement en utilisant les protocoles sécuritaires WAP.

2 Les composantes de l'architecture WAP

L'architecture WAP fournit un ensemble de protocoles constituant un modèle à couches. Chaque couche de cette architecture est accessible par les couches supérieures et inférieures, aussi bien que par d'autres services et applications.

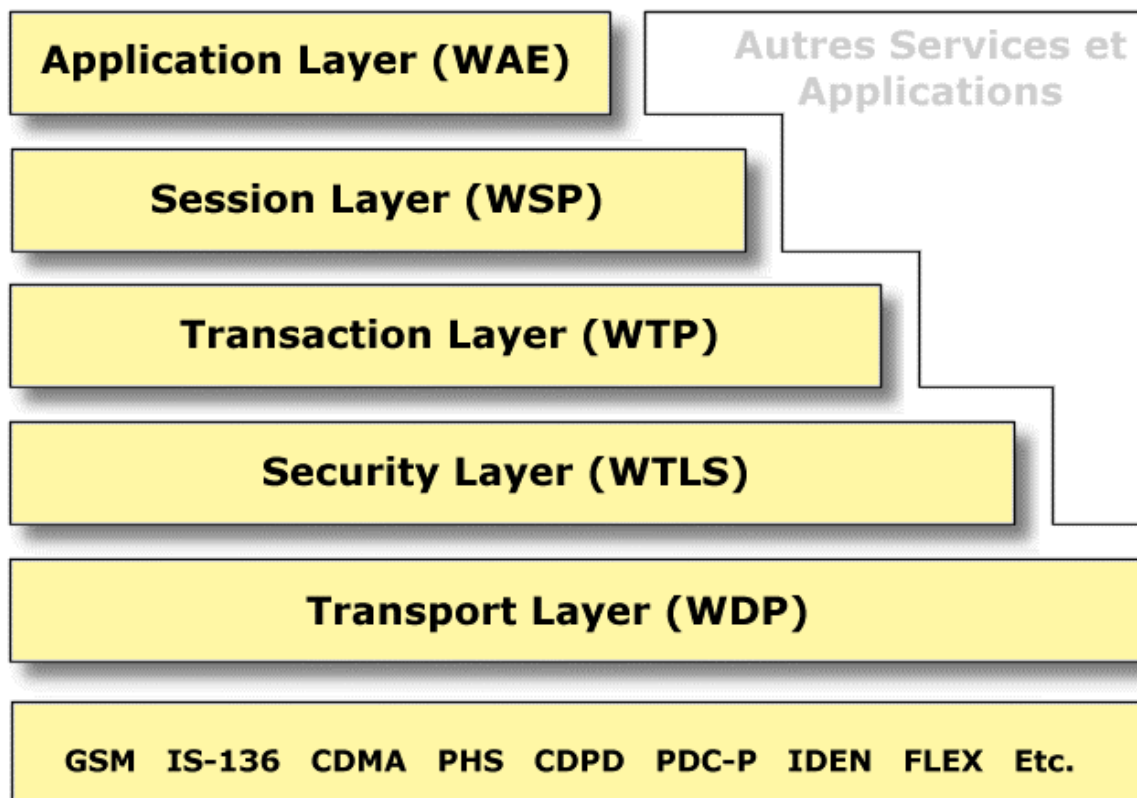


Figure 4 – Architecture WAP

Ce modèle à couches autorise d'autres services et applications à utiliser les fonctionnalités de la pile WAP au travers d'un jeu d'interface bien défini. Ces applications externes peuvent accéder aux couches session, transaction, sécurité et transport directement. Les chapitres suivants fournissent une description des différents éléments composant cette pile de protocoles.

2.1 Wireless Application Environment (WAE)

WAE est un environnement applicatif générique basé sur une combinaison des technologies World Wide Web, et des technologies de téléphonie mobile. L'objectif premier de cet environnement est d'établir un environnement inter opérable permettant aux opérateurs et aux fournisseurs de services de construire des services et des applications pouvant atteindre une large gamme de différentes plates-formes mobiles, d'une manière efficace et pratique. WAE comprend un environnement de micro navigateur répondant aux fonctionnalités suivantes :

- Gestion du WML, un langage similaire au HTML mais optimisé pour une utilisation sur des terminaux mobiles portables.
- Gestion du WMLScript, un langage de script simplifié similaire à JavaScript.
- Support des protocoles WTA (Wireless Telephony Application) et WTAI (Wireless Telephony Application Interface). Ces protocoles fournissent des services ainsi qu'une interface de programmation vers le périphérique mobile.
- Un jeu de formats de contenus incluant images, annuaires téléphoniques, calendriers, etc.

Une description détaillée de cette couche sera effectuée ultérieurement.

2.2 Wireless Session Protocol (WSP)

WSP fournit à la couche application de WAP (WAE) une interface logique pour deux services orientés session. Le premier est un service orienté connexion qui opère au dessus de la couche transaction (WTP). Le second est un service non connecté qui opère sur un service datagramme (WDP) sécurisé ou non.

WSP consiste en des services spécialisés pour des applications de navigation comportant les fonctionnalités suivantes (WSP/B) :

- Support du protocole HTTP 1.1.
- Gestion des sessions utilisateur.
- Suspension et destruction des sessions avec possibilité de migration de la session.
- Possibilité de fournir des données de type push.
- Négociation des caractéristiques du protocole.

Les protocoles dans la famille WSP sont optimisés pour les réseaux de faible bande passante avec des temps de latences relativement longs. WSP/B a été conçu pour permettre à un proxy WAP d'inter connecter un client WSP/B à un serveur HTTP.

2.3 Wireless Transaction Protocol

WTP opère au dessus d'un service datagramme, il fournit les fonctionnalités nécessaires à l'implémentation d'un protocole orienté transaction pour des périphériques légers, comme des téléphones mobiles. WTP peut fonctionner sur des réseaux de types datagramme qu'ils soient sécurisés ou non. WSP propose les fonctionnalités suivantes :

- Trois classes de services orientées transaction :
 - Services de requêtes unidirectionnelles non fiables.
 - Services de requêtes unidirectionnelles fiables.
 - Services de requêtes bidirectionnelles fiables.
- Fiabilité optionnelle d'utilisateur à utilisateur. Les utilisateurs de WTP déclenchent la confirmation pour chaque message reçu.
- Transactions asynchrones.

2.4 Wireless Transport Layer Security (WTLS)

WTLS est un protocole sécuritaire basé sur le standard industriel TLS (Transport Layer Security), plus communément connu sous le nom de SSL (Secure Sockets Layer). WTLS est destiné à être utilisé avec les protocoles de transport WAP, et a été optimisé pour être sur des canaux de communication à faible bande passante. WTLS propose les fonctionnalités suivantes :

- Intégrité des données : WTLS permet de s'assurer que les données qui circulent entre un serveur et un client sont authentiques et non corrompues.
- Confidentialité : WTLS permet d'assurer entre un serveur et un client une communication privée ne pouvant être déchiffrée par une tierce partie qui aurait intercepté le flux des données transmises.
- Authentification : WTLS contient des mécanismes permettant de s'assurer que le serveur applicatif, et le terminal client sont bien ceux qu'ils prétendent être.
- Protection contre les attaques par déni de service : WTLS possède des fonctionnalités permettant de détecter et de rejeter les données répétitives et/ou non authentifiées. Ceci permet de rendre les attaques par déni de service les plus courantes plus difficiles à réaliser, tout en protégeant les couches de protocoles supérieures.

WTLS a été conçu non seulement pour être utilisé dans des communications sécurisées entre terminaux et des serveurs, mais il peut également être utilisé dans le cadre de l'authentification des cartes de paiements pour le commerce électronique.

Les applications doivent être capables d'utiliser ou non les fonctionnalités de WTLS, en fonction de leur besoin du moment, et des caractéristiques du réseau utilisé (en effet, dans certains cas, certains réseaux peuvent fournir une couche sécuritaire de plus bas niveau, rendant ainsi WTLS inutile.).

2.5 Wireless Datagram Protocol (WDP)

La couche transport de WAP est connue sous le nom de Wireless Datagram Protocol (WDP). La couche WDP se situe juste au dessus de la couche de service réseau. WDP agit comme une couche de transport généraliste, en offrant des services logiques aux couches de protocoles supérieures, et en communiquant de manière transparente avec les différentes couches support disponibles.

WDP offrant une interface commune aux couches de protocoles supérieures, les couches Sécurité, Session, et Application sont à mêmes de fonctionner en totale indépendance du type de réseau utilisé. Ainsi, ce modèle, ainsi que toutes les applications qui pourront être développées seront disponibles dans le cas où un nouveau type de réseau apparaîtrait. Ceci est possible en adaptant la couche transport aux fonctionnalités et aux caractéristiques du réseau servant au transport. En conservant la cohérence et les fonctionnalités de base de la couche transport, on accède à une interopérabilité globale par l'utilisation de passerelles, permettant à des périphériques utilisant des supports réseaux différents, d'accéder aux mêmes services, avec le même niveau de qualité de service en terme de fonctionnalité.

2.6 Les couches support réseau

Les protocoles WAP sont conçus pour être utilisés au dessus d'une large variété de couches support réseau différentes, comme les « short message », « circuit-switched data », et « packet data ». Ces supports offrent différents niveaux de qualité de service, en terme de délais, taux d'erreurs, et de débit. Les protocoles WAP sont conçus pour compenser ces disparités de qualité de services.

WTP décrit la liste des supports réseaux supportés ainsi que les techniques permettant aux protocoles WAP de fonctionner avec chacun de ces supports réseaux. La liste de ces supports réseaux est susceptible d'évoluer dans le temps au rythme des nouveaux supports réseaux et de l'évolution du marché des périphériques mobiles.

2.7 Autres services et applications

Le modèle d'architecture à couches de WAP permet à d'autres services, et applications, d'utiliser les fonctionnalités de la pile WAP au travers d'un jeu d'interface bien défini. Ces applications externes peuvent accéder directement aux couches session, transaction, sécurité, et transport. Ceci permet d'utiliser la pile WAP pour des applications ou des services qui ne sont pas encore actuellement définis mais jugés très intéressants pour le marché des périphériques mobiles. Par exemple les applications comme le courrier électronique, la gestion d'agenda, le répertoire téléphonique, le bloc notes, le commerce électronique, ou des services comme les pages jaunes ou les pages blanches peuvent être développés pour utiliser les protocoles WAP.

2.8 Exemples de configurations

Le schéma de la figure 5 illustre différents exemples de piles de protocoles utilisant les technologies WAP. Ces exemples ne sont donnés qu'à titre indicatif et ne constituent en aucun cas une référence de conformité ou d'interopérabilité.

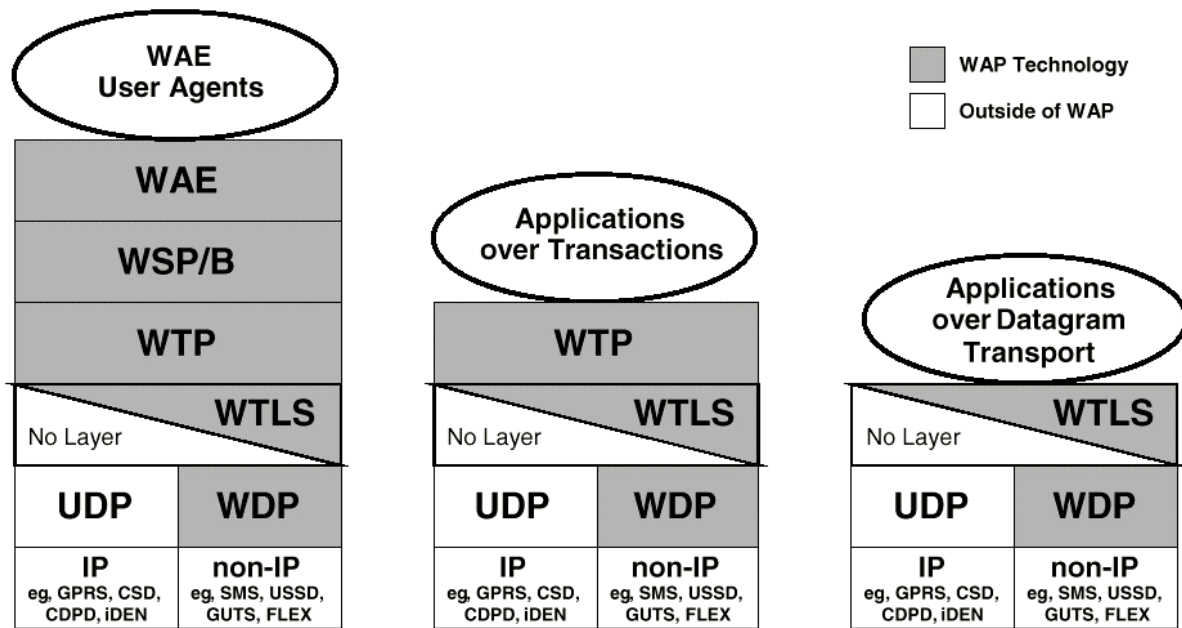


Figure 5 - Exemples de piles WAP

La pile de gauche représente un exemple typique d'application WAP, un navigateur WAE fonctionne au dessus de l'ensemble complet des protocoles WAP. La pile du milieu serait utilisée par des applications ou des services qui nécessiteraient des services transactionnels avec ou sans couches de sécurité. La couche de droite serait utilisée pour des applications ou des services qui nécessiteraient uniquement un transport de datagramme avec ou sans sécurité.

3 Wireless Application Environment

Nous allons nous attacher plus particulièrement dans ce chapitre à la couche WAE. En effet, c'est au travers de cette couche que les fournisseurs d'applications, de services, et/ou de contenus vont concevoir leurs applicatifs.

Nous allons étudier les différentes composantes de WAE dont, Wireless Markup Language (WML), Wireless Markup Scripting Language (WMLScript), les agents utilisateurs WAE, et les types de médias WAE.

3.1 Fonctions générales de WAE

3.1.1 Interface de la couche session

WML et les agents utilisateurs WTA communiquent en utilisant WSP (Wireless Session Protocol), sur une ou plusieurs sessions WSP par agent utilisateur. Cette communication réseau s'effectue sous la forme d'une entête et d'un contenu WSP / HTTP 1.1. La session WSP est créée et contrôlée par l'entité d'administration de la session (SME – Session Management Entity). SME n'est pas défini par l'architecture WAP et son implémentation est spécifique.

3.1.2 Authentification

Les agents utilisateurs WAE doivent implémenter l'authentification basique spécifiée dans le protocole HTTP 1.1 [RFC2068].

3.1.3 URLs

Les agents utilisateurs WAP doivent respecter les règles de nommage standard des URLs. Une URL de type «http:» désigne une ressource située sur un serveur HTTP originel [RFC2396]. Cette règle de nommage n'implique l'utilisation d'aucun protocole de communication particulier entre un téléphone et une passerelle réseau. Le serveur originel désigné par l'URL peut être accédé par une passerelle (ou un proxy) WSP – vers – HTTP. Par contre l'URL peut désigner un serveur du réseau qui combine à la fois le rôle de serveur HTTP et celui de passerelle WSP. Dans ce dernier cas l'accès à la ressource s'effectue directement au travers du protocole WSP.

Des URLs non standards sont définis pour désigner des accès spécifiques client/terminal, au sein de la spécification WTAI. Etant donné que ces règles de nomenclatures sont spécifiques à un agent utilisateur WAE, elles ne sont pas incluses dans cette section.

3.1.4 Caractéristiques des agents utilisateurs

Note importante :

Les informations présentées dans cette section sont encore en cours de réflexion et de développement. Le contenu de ce chapitre est sujet à modifications. Il est donc possible que ces modifications ou ces ajouts ne soient pas compatibles avec les informations présentées ici.

Dans le but d'optimiser le modèle client WAE/serveur un certain nombre de caractéristiques sont envoyées par l'agent utilisateur vers le serveur originel WAP. Ces caractéristiques permettent au serveur originel de ne pas envoyer un contenu inapproprié à l'agent utilisateur. Ceci permet également au serveur et à la passerelle de personnaliser la réponse pour un agent utilisateur particulier.

La couche WSP fournit un modèle de transfert des données. Les entêtes de contenu WSP/HTTP 1.1 sont utilisées pour mettre en place une négociation de contenu, et définir un jeu de caractères ainsi qu'un langage pour la communication entre l'agent utilisateur et le serveur originel.

Les serveurs originels ou les passerelles WAP peuvent être amenés à modifier les entêtes de réponse en fonction des caractéristiques de l'agent utilisateur. Pour chacun des types de médias WAP inclus dans l'entête WSP/STTP `Accept`, l'agent utilisateur doit inclure un paramètre appelé `uaprof`, spécifiant l'URI d'un profil spécifiant les caractéristiques de l'agent utilisateur.

Exemple d'entête `Accept`

```
Accept:      application/x-wap.wmlc;uaprof=http://www.vendeur.com/telephone1,
            application/x-wap.wmlscriptc;uaprof=http://www.vendeur.com/téléphone1,
            text/x-wcard,
            text/x-vcal
```

Pour les types de médias WAP, le paramètre `uaprof` combiné avec les entêtes WSP/HTTP `Accept`, `Accept-Language`, et `Accept-Charset` permettent de décrire complètement toutes les caractéristiques de contenu adaptables à l'agent utilisateur. La combinaison de l'ensemble de ces trois paramètres est appelée « entête caractéristique ».

Les passerelles qui reçoivent une requête utilisant « les entêtes caractéristiques » doivent conserver ces entêtes pour toute requête adressée au serveur originel pour le compte de l'agent utilisateur, elles doivent également insérer le paramètre `uaprof` sur tous les types de médias. Par exemple, si l'entête `Accept` d'une requête d'un agent utilisateur spécifie qu'il accepte le type `application/x-wap.wmlc`, et que le document du serveur originel demandé à la passerelle est de type `text/x-wap.wml`, alors la passerelle doit copier le paramètre `uaprof` de cet agent utilisateur dans l'entête `Accept` qu'elle envoie au serveur originel.

Certaines passerelles peuvent mettre en cache les documents reçus des serveurs originels. Si une requête d'un agent utilisateur concerne un document mis en cache par la passerelle et que la requête contient les « entêtes caractéristiques », la passerelle ne doit pas fournir le document mis en cache sauf si au moins une des trois conditions suivantes est vraie :

- Les «entêtes caractéristiques » spécifiées dans la requête sont identiques à celles utilisées par la passerelle quand celle ci à initialement récupérée le document.
- Les «entêtes caractéristiques » spécifiées dans la requête sont identiques à celles utilisées par la passerelle quand celle ci à initialement récupérée le document, sauf si le paramètre `uaprof`, et le profil spécifié par l'URI du paramètre `uaprof` sont différents mais sémantiquement équivalents à ceux utilisés lorsque la passerelle à initialement récupérée le document.
- La passerelle est capable de garantir au travers d'autres mécanismes (l'analyse des métadonnées HTTP, par exemple) qu'une nouvelle requête sur le serveur originel en utilisant l'entête `Accept` de l'agent utilisateur, résulterait dans l'envoi par le serveur originel du même contenu que celui mis en cache par la passerelle.

Le tableau ci-dessous récapitule l'utilisation des entêtes WSP/HTTP

ID	Caractéristique de l'agent utilisateur	Type de données	Description	Méthode de transmission
1	Character set / encoding	Jeu de caractères IANA	Jeu de caractères et d'encodage supporté par le client	Accept-Charset WSP/HTTP header [RFC2068]
2	Language	Code de Pays IANA	Langage supporté par le client	Accept-Language WSP/HTTP header [RFC2068]
3	Media Type	Type de média IANA	Type de format et d'encodage supporté par le client	Accept WSP/HTTP header [RFC2068]
4	WML version	Numéro de version	Réfère au langage WML supporté. La version actuelle est 1.0	Défini avec les caractéristiques de l'agent utilisateur (voir <code>uaprof</code>)
5	WMLScript version WMLScript support Support de la virgule flottante	Numéro de version	Réfère au langage WMLScript supporté. La version actuelle est 1.0 Si une version est définie, cela signifie que l'agent utilisateur supporte le langage de script. Dans le cas contraire le serveur originel doit réagir comme si le client ne supportait pas, ou ne voulait pas utiliser le langage de script. Ajout au numéro de version permettant d'indiquer le support de la virgule flottante. La version courante est 1.1F	Défini avec les caractéristiques de l'agent utilisateur (voir <code>uaprof</code>)
6	Standard Libraries Supported	Numéro de version	Se rapporte aux bibliothèques de fonctions supportées par WMLScript.	Défini avec les caractéristiques de l'agent utilisateur (voir <code>uaprof</code>)
7	WTA version Event tables used	Numéro de version	Se rapporte au numéro de version de WTA supporté par le client. La version courante est 1.0 Ajout au numéro de version. Permet d'indiquer la version	Défini avec les caractéristiques de l'agent utilisateur (voir <code>uaprof</code>)

ID	Caractéristique de l'agent utilisateur	Type de données	Description	Méthode de transmission
			WTA supportant les tables d'évènements. La version courante est 1.0C	
8	WTAI Basic version	Numéro de version	Se rapporte au numéro de version de la librairie WTAI utilisée. La version courante est 1.0	Défini avec les caractéristiques de l'agent utilisateur (voir uaprof)
9	WTAI Public Version	Numéro de version	Se rapporte au numéro de version de la librairie publique WTAI utilisée. La version courante est 1.0	Défini avec les caractéristiques de l'agent utilisateur (voir uaprof)
10	List WTAI Net-Spec Versions	Liste de numéros de version	Se rapporte au « WTAI net-spec lib » supportée. Les valeurs possibles sont : 1.0GSM, 1.0IS-136, 1.0IS-95, et 1.0PDC.	Défini avec les caractéristiques de l'agent utilisateur (voir uaprof)

Tableau 1 – Caractéristiques des agents utilisateurs

3.1.5 Wireless Markup Language (WML)

WML définit un langage à base de marqueurs basé sur la norme XML et destiné à être utilisé pour générer du contenu ainsi que des interfaces utilisateurs à destinations de périphériques à faible bande passante comme des téléphones cellulaires ou des pagers.

Cela implique les contraintes suivantes :

- Zone d'affichage réduite et possibilité de saisie utilisateur réduite.
- Faible bande passante pour la connexion au réseau de l'agent utilisateur.
- Ressource mémoire et CPU limité sur l'agent utilisateur.

WML inclut les fonctionnalités suivantes :

- Présentation et mise en page de textes. WML inclut le support du texte et des images ainsi qu'une gamme de commande de formatage et de mise en page, comme gras, italique, etc.
- Organisation sous la forme de cartes. Toutes les informations WML sont organisées sous la forme de collection de cartes. Une carte correspond à une ou plusieurs interactions utilisateur (un écran texte, un choix, ou un champ de saisie). L'utilisateur peut naviguer au travers de sa collection de cartes en visualisant le contenu de la carte, en entrant les informations requises, en validant un choix, ou en se déplaçant vers une autre carte. Les collections de cartes sont organisées en jeu (deck). Un jeu WML est similaire à une page HTML, et il est identifié par une URL [RFC2396]. Le jeu (deck) est l'unité de transmission de contenu WML.
- Navigation et chaînage inter-carte. WML inclut les mécanismes permettant de naviguer entre des cartes et/ou des jeux (decks). WML permet également de gérer des événements pouvant survenir dans le périphérique, qui peuvent être utilisés pour naviguer, ou pour exécuter des scripts. WML supporte aussi les ancres à l'instar de ceux utilisés en HTML.

- Gestion des paramètres. Tous les jeux WML peuvent être paramétrés en utilisant des variables. Ces variables sont substituées au moment de l'exécution.

3.1.6 WMLScript

WMLScript est un langage procédural pouvant être utilisé sur le client pour apporter une certaine logique de contrôle. WMLScript est basé sur la norme ECMAScript [ECMA262], mais a été modifié afin de mieux supporter les communications à faible bande passante en direction de clients légers. WMLScript peut être utilisé en association avec le langage WML afin d'apporter de l'intelligence aux agents utilisateurs. Toutefois il peut également être utilisé seul, de manière autonome.

WMLScript a été débarrassé des fonctionnalités avancées de la norme ECMAScript. De plus, il est transmis au client sous une forme encodée permettant ainsi à la fois un gain de bande passante et un gain de temps CPU sur le client.

3.1.7 Agents utilisateurs WAE

Les agents utilisateurs WML sont des composants fondamentaux de WAE. Toutefois, WAE n'est pas limité à l'utilisation de WML. WAE permet l'intégration d'agents utilisateurs spécifiques avec des architectures et des environnements très variés. En particulier, un agent utilisateur WTA (Wireless Telephony Application), et un agent utilisateur WTAI (Wireless Telephony Interface) ont été normalisés comme une partie intégrante de WAE pour l'environnement de la téléphonie mobile. Les fonctions WTAI permettent aux programmeurs d'accéder et d'interagir directement avec les fonctionnalités de téléphonie du téléphone mobile (contrôle d'une communication par exemple).

3.1.7.1 *Les agents utilisateur WTA*

Les agents utilisateurs WTA ne sont pas complètement normalisés dans le cadre de la spécification WAP. Un certain nombre de recommandations sont spécifiées dans la norme WTA. Une spécification de WTAI (Wireless Telephony Application Interface), qui correspond à l'API téléphonique WAP, est définie dans la norme WTAI.

3.1.7.2 *Les agents utilisateurs WML*

Les agents utilisateurs WML ne sont pas complètement normalisés dans le cadre de la spécification WAP. Un certain nombre de recommandations sont fournies dans la spécification du langage WML, et dans la spécification du langage de script WMLScript.

3.1.8 Types de média WAE

WAE adopte un certain nombre de formats de contenus qui facilitent les échanges de données. Les plus importants sont le WML encodé et le WMLScript encodé. L'encodage des contenus WML et des scripts WMLScript permet une transmission plus efficace, et minimise les efforts nécessaires aux clients pour les exécuter.

WAE adopte également une classe de médias additionnels permettant de faciliter l'échange d'objets entre le client et le serveur, ou bien encore entre deux clients. Ceux ci sont actuellement limités aux cartes de visite électroniques et aux objets calendriers électroniques. Ces objets peuvent être échangés en utilisant des datagrammes WDP, ou au travers d'une session WSP. Dans le cas d'un échange utilisant des datagrammes un jeu de ports bien défini a été réservé afin de conserver une interopérabilité entre différentes implémentations.

3.1.8.1 *Format WML encodé*

Le format WML encodé correspond à une version compacte du code WML. Les tags WML sont tokenisés afin de rendre le code WML plus compact, et donc plus facilement transportable sur des réseaux à faible bande passante. Cette tokénisation permet également d'implémenter sur le client un interpréteur plus compact, en accord avec la faible puissance CPU et la faible quantité de RAM présente sur le client. La transformation du WML en WML encodé est généralement prise en charge par la passerelle (ou le proxy) WAP. Toutefois il est possible à un serveur originel de générer directement du code WML encodé, le serveur originel jouant ainsi également le rôle de passerelle WAP.

3.1.8.2 *Format WMLScript encodé*

Le principe d'encodage du WMLScript est sensiblement identique à celui du WML. Son utilisation est soumise aux mêmes restrictions et aux mêmes règles que WML.

3.1.8.3 *Format carte de visite électronique*

Le format vCard est défini par le consortium Versit, et est actuellement administré par l'IMC.

3.1.8.4 *Format calendrier électronique*

Le format vCalendar est défini par le consortium Versit, et est actuellement administré par l'IMC.

3.1.8.5 *Format image*

WAE fournit un environnement visuel qui a été conçu pour répondre à certaines exigences comme le support de multiples profondeurs de pixels, le support de tables de couleur, l'encodage compact, l'utilisation de faibles capacités CPU et RAM pour le décodage.

WAE répond à ces exigences en :

- Supportant les types standards WSP/HTTP pour les formats d'images habituels comme par exemple, `image/png`
- Introduisant un format bitmap optimisé appelé Wireless BitMaP (WBMP). Le type de média correspondant étant `image/x-wap.wbmp`.

3.1.8.6 Messages multipart

WAE inclut une spécification d'encodage multipart. WSP traduit les entités multipart MIME [RFC2045] dans un format binaire compact, optimisé pour les environnement à faible bande passante.

4 Wireless Markup Language (WML)

WML est un langage de marqueurs basés sur la norme XML, permettant de décrire des interfaces utilisateur, des contenus, et des interactions pour des périphériques à faible bande passante comme des téléphones portables, ou des pagers. Lors de sa conception, les contraintes dues à la faible bande passante ont été omniprésentes.

- La taille des écrans est petite et de faible résolution. Un téléphone portable ne possède que quelques lignes n'excédant pas 8 à 12 caractères
- Les périphériques offrent des possibilités de saisies utilisateurs très limitées. Un téléphone ne permet que de saisir aisément des chiffres, plus difficilement des lettres, et éventuellement quelques autres touches de fonctions spécialisées.
- Les périphériques possèdent de très faible capacité de traitement CPU, et possèdent généralement une quantité de mémoire RAM très faible.
- Les réseaux mobiles interconnectant ces périphériques ont une bande passante très faible d'environ 300 bps, à 10 kbps, avec un temps de latence compris entre 5 et 10 secondes.

WML assume complètement l'héritage de l'architecture du World Wide Web en général, et de HTML en particulier. A ce titre, il utilise des URLs [RFC2396] pour localiser les documents et/ou les applications. Il utilise également le protocole HTTP 1.1 [RFC2068], comme protocole de transport. Les URLs sont utilisées dans WML aussi bien pour naviguer, que pour désigner une ressource externe (script, image, etc.).

4.1 La syntaxe de WML

4.1.1 Les « entités »

Un document WML peut contenir des entités. Les entités sont en fait des substitutions à des caractères, ou des chaînes de caractères. Par exemple le caractère «& » est représenté en WML par l'entité `&`. Toutes les entités commencent par le caractère «& », et finissent par le caractère «; ».

4.1.2 Les éléments

Ils permettent d'effectuer le marquage du document afin de structurer l'information qu'il contient. Ils contiennent un tag de début, un contenu optionnel, et un tag de fin lui aussi optionnel. Ainsi un élément peut être :

```
<tag> contenu </tag>
```

ou

```
<tag/>
```

Les éléments sont sensibles aux majuscules / minuscules.

4.1.3 Les attributs

Les attributs permettent d'apporter un degré de précision supplémentaire aux éléments, en apportant des informations supplémentaires. Ils sont toujours situés dans le tag de début d'un élément.

```
<tag attr="abcd"/>
```

Tout comme les éléments, les attributs sont sensibles aux majuscules / minuscules.

4.1.4 Les commentaires

Les commentaires suivent la norme XML

```
<!--Un commentaire -->
```

Les commentaires sont autorisés pour simplifier le développement et la maintenance des documents WML, mais ils sont totalement ignorés par les agents utilisateurs. Attention toutefois à ne pas en abuser, car dans le cas du WML, et de la faible bande passante disponible, ils viennent grever la taille maximum autorisée pour un deck WML.

4.2 Les variables

Les cards et les decks WML peuvent être paramétrés à l'aide de variables. La syntaxe suivante est utilisée pour substituer une variable en WML.

```
$identifiant  
$(identifiant)  
$(identifiant:conversion)
```

Les parenthèses sont requises dans le cas où un espace ne vient pas délimiter le nom de l'identifiant. La substitution de variable a le plus haut degré de priorité en WML, ce qui implique que tout caractère suivant le caractère non échappé "\$" sera considéré comme étant un identifiant de variable. La succession de deux signes dollar (\$\$) représente le symbole \$.

4.2.1 Erreurs

WML étant basé sur la norme XML, le concept de document bien formé (well formed) lui est applicable. Ainsi, tout document WML violant la définition XML d'un document bien formé, sera considéré comme étant un document en erreur.

4.3 Exemple de document WML

Cet exemple d'application WML, est consultable en ligne à l'adresse suivante :

<http://www.agoraweb.org/wap/index.wml>

Le but de ce document n'étant pas de présenter de manière exhaustive le langage WML, nous allons maintenant regarder d'un peu plus près un document WML afin de comprendre son fonctionnement, et ainsi d'entre apercevoir les possibilités offertes par ce langage.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<!-- Main menu for Agora Web -->

<wml>
  <card title="Agora Web">
    <p align="center">
      Bienvenue sur <br/>Agora Web<br/>
      <select>
        <option onpick="#apps">Recherches</option>
        <option onpick="/cgi-local/info_wap.cgi?step=1">Infos
pratiques</option>
      </select>
    </p>
  </card>

  <card id="apps">
    <p>
      Rechercher <br/>
      <select>
        <option onpick="/wap/s.wml">Une commune</option>
        <option onpick="/wap/sl.wml">Un loueur automobile</option>
      </select>
    </p>
  </card>

</wml>
```

Exemple 1 – Exemple de fichier WML

Dans l'exemple ci-dessus, le deck WML contient deux cards. La première est automatiquement affichée par le périphérique lorsqu'il reçoit le deck. Elle propose deux choix à l'utilisateur sous la forme d'une liste de choix que l'utilisateur peut sélectionner. Ce choix est généré par l'élément WML "SELECT".



Ecran 1 – Résultat de l'affichage de la première card du deck

Le premier choix renvoi vers la seconde card du deck, alors que le second choix pointe vers une URL externe qui se trouve en fait être un script CGI tout a fait banal, à ceci près qu'il génère en sortie du WML, au lieu de générer du HTML comme c'est habituellement le cas sur le Web.

Lorsque le premier choix est sélectionné, la deuxième card est affichée, proposant à nouveau un choix à l'utilisateur.



Ecran 2 – Résultat de l'affichage de la deuxième card du deck

Dans chacun des choix proposés à l'utilisateur, un lien est réalisé vers un autre document WML, qui sera chargé lorsque l'utilisateur exprimera un choix en sélectionnant une des deux propositions.

Admettons, que l'utilisateur choisisse le choix 1, dans ce cas le fichier WML suivant est demandé par le périphérique mobile.

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
           "http://www.wapforum.org/DTD/wml_1.1.xml">
<!-- Recherche de communes -->

<wml>
  <card id="i1">
    <do type="accept">
      <go href="http://www.agoraweb.org/cgi-local/s_wap.cgi">
        <postfield name="searchvalue" value="$chx"/>
        <postfield name="searchtype" value="communes"/>
      </go>
    </do>

    <p>
      Saisir une ville ou un code postal <br/>
      <input name="chx" type="text" format="*m"/>
    </p>
  </card>

  <card id="i2">
    <do type="options" label="Clear">
      <refresh>
        <setvar name="$pswd" value=""/>
      </refresh>
    </do>
    <p>
      nous recherchons $chx <br/>
      merci de bien vouloir patientez ...
    </p>
  </card>
</wml>

```

Exemple 2 – Exemple de fichier WML

Ce deck WML contient à nouveau deux card. La première qui est chargée lors de la réception du deck par le périphérique demande à l'utilisateur de saisir une information.



Ecran 3 – Affichage de l'écran de saisie

L'utilisateur peut alors saisir un code postal ou le nom d'une commune et valider sa saisie. La validation de l'utilisateur demandera une URL externe correspondant à un script CGI, en passant dans la chaîne de requête deux informations sous la forme de champ HTTP. Ces informations étant en fait le paramètre "searchtype" qui est utilisé en interne par le script CGI, et la saisie utilisateur qui est passé en paramètre par le biais de la variable \$chx

Quand à la seconde card, et bien elle n'est tout simplement jamais affichée car rien ne la relie à la première.

Affiché par un périphérique différent, un écran WML peut être très différent, comme le montre la photo ci-dessous.



Ecran 4 – L'écran de saisie sur un autre périphérique

Cet exemple montre bien combien il est difficile de s'adapter à des périphériques relativement sobre en terme de possibilité d'affichage.

Les scripts CGI ne seront pas présentés ici car leur explication nécessiterais de trop nombreuses explications ayant trait plus aux script CGI et au langage PERL, qu'à la syntaxe du langage WML.

5 WMLScript

6 Wireless Telephony Application

Cette spécification définit l'architecture WTA, ainsi que les agents utilisateurs WTA. L'architecture WTA enrichit l'architecture WAE en proposant :

- Une interface entre WML et WMLScript, et l'ensemble des fonctions spécifiques de téléphonie du client. Cette interface est appelée Wireless Telephony Application Interface (WTAI).
- Une gestion des événements réseau. Cela signifie que des événements ayant pour origines le réseau mobile peuvent être détectés par l'agent utilisateur WTA, et que des actions peuvent être définies en réponse à ces événements.
- Un modèle de sécurité obligatoire.
- Un modèle pour gérer l'état et le contexte d'un agent utilisateur WTA.

6.1 Architecture

WTA est une architecture applicative pour les services de téléphonie. L'agent utilisateur de WTA est sensiblement similaire à l'agent utilisateur WML avec en plus des possibilités d'interfaçage entre les services du réseau mobile et le périphérique de téléphonie mobile, comme par exemple initier ou recevoir des appels téléphoniques. Le schéma ci-dessous décrit une des configurations possibles pour l'architecture WTA, en définissant les composants contenus sur le client.

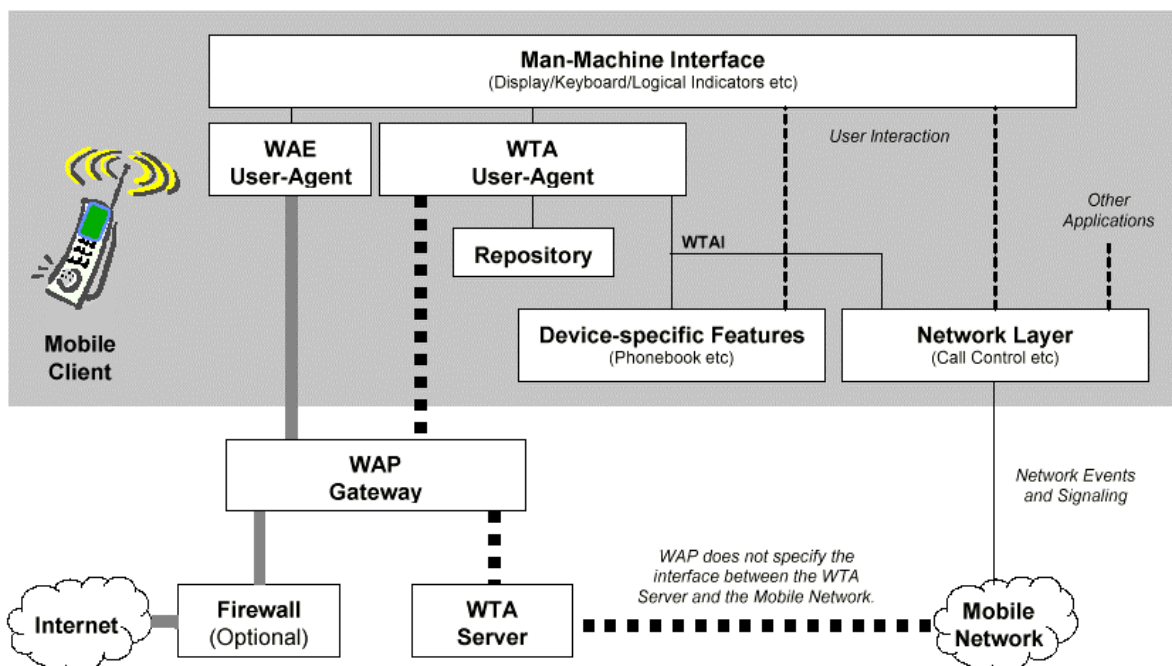


Figure 6 – Exemple d'architecture WTA

Comme indiqué ci-dessus l'architecture WTA repose sur un agent utilisateur dédié, sur le client. Le serveur WTA n'est pas spécifié dans la norme WAP, mais des lignes directrices sont données à titre indicatif.

6.1.1 L'agent utilisateur WTA

Le schéma précédent illustre les interactions qui existent entre l'agent utilisateur WTA, le repository, WTAI, et les autres entités d'un périphérique mobile compatible WTA.

L'agent utilisateur WAE récupère ses données exclusivement au travers de la passerelle WAP. L'agent utilisateur WTA peut de plus récupérer ses contenus depuis le repository. WTAI permet à l'agent utilisateur WTA d'interagir avec les fonctionnalités fournies par le réseau mobile (initier les appels téléphoniques par exemple) et de manipuler les fonctionnalités spécifiques du périphérique (manipulation du carnet d'adresse par exemple).

6.1.2 Le serveur WTA

Le serveur WTA peut être vu comme un serveur Web délivrant un contenu demandé par un client. A l'instar d'un navigateur Web, un agent utilisateur WTA utilise des URLs pour référencer des contenus sur le serveur WTA.

Une URL peut également référencer une application sur un serveur Web (script CGI par exemple) qui serait exécutée. Ce genre d'applications peuvent servir à gérer un très grand nombre de tâches comme par exemple générer du contenu dynamique, ou interagir avec des périphériques externes.

Un serveur WTA peut également utiliser ce concept. En référençant des applications sur un serveur WTA, il est possible de créer des services qui utilisent des URLs pour interagir avec le réseau mobile et d'autres entités (un système de messagerie vocale par exemple). Le concept de référencement d'application sur un serveur WTA fournit un modèle simple mais puissant pour intégrer des services à valeur ajoutée qui s'exécuteront localement sur le client WAP.

6.2 Sécurité

C'est à chaque opérateur de réseau mobile de fournir à ses clients un niveau de sécurité acceptable. La manière dont cette sécurité dans le domaine de l'opérateur de réseau mobile est mise en œuvre, n'est pas définie par l'architecture WAP. Le fournisseur de services de téléphonie mobile peut choisir d'implémenter lui-même tous les services WTA, ou bien de déléguer l'administration de ses services WTA à une ou des tierces parties.

6.2.1 Contrôle d'accès

WDP fournit un moyen d'implémenter séparément un service WTA d'un service WAE en utilisant des ports prédéfinis.

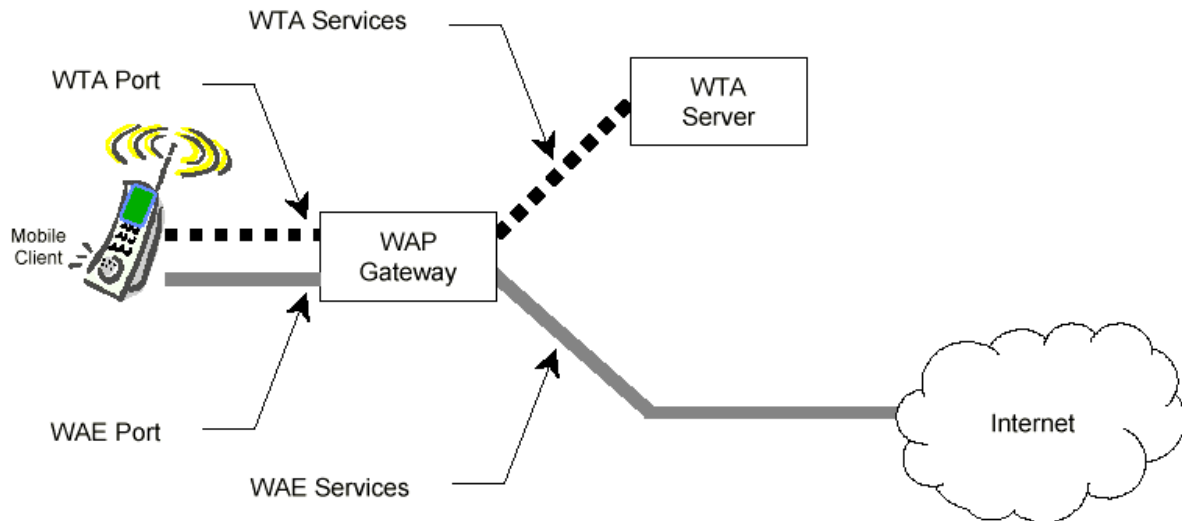


Figure 7 – Contrôle d'accès et numéro de ports WDP

Une session WTA établit par un agent utilisateur WTA doit utiliser un des ports sécurisés dédiés de la passerelle. Un agent utilisateur WTA ne doit en aucun cas récupérer des contenus WTA en dehors d'une session WTA. Tout contenu WTA reçu en dehors de la session WTA doit être écarté.

6.2.2 Permission d'accès utilisateur

Des permissions utilisateur doivent être établies pour tout appel à des fonctions WTAI.

7 Wireless Telephony Application Interface

Les fonctionnalités fournies par WTAI permettent de créer des applications de téléphonie. Un exemple typique étant de générer un appel téléphonique depuis le mobile en utilisant des fonctions WTAI appelées depuis un deck WML ou depuis un script WMLScript.

L'agent utilisateur WTA utilise des bibliothèques de fonction WTAI pour accéder aux services du réseau, et il est capable de recevoir des événements WTA en provenance du réseau ainsi que de recevoir du contenu de type « push », comme des deck WML ou des événements WTA.

7.1 Les bibliothèques WTAI

Les fonctionnalités WTAI sont segmentées dans une série de bibliothèques de fonction. Ces bibliothèques de fonction sont accessibles depuis WML au travers d'URLs ou depuis un script WMLScript au travers d'une fonction.

Ces bibliothèques sont réparties de la manière suivante :

Network Common WTA	Cette bibliothèque contient les fonctionnalités les plus communes qui sont disponibles sur tous les réseaux. Elles sont accessibles uniquement depuis l'agent utilisateur WTA
Network Specific WTA	Ces fonctionnalités sont disponibles uniquement sur certains types de réseaux. Les fonctionnalités spécifiques à un opérateur font également partie de cette catégorie
Public WTA	Fonctionnalités simples accessibles à des applications de tierce partie exécutées au travers de l'agent utilisateur WAE

7.2 Gestion des événements

La gestion des événements WTA est un moyen de réagir aux changements d'état du serveur WTA ou du réseau mobile. L'agent utilisateur WTA peut être paramétré pour réagir à un événement WTA. Les événements WTA doivent être liés à une URL indiquant ainsi le contenu à charger, et/ou à exécuter en réponse à un événement WTA.

7.3 Format d'une URI WTAI

L'accès aux fonctions WTAI se faisant au travers d'une URL, un format a été mis au point pour désigner ces fonctions.

```
wtai://<library>/<function> (; <parameter>)* [! <result> (; <result> )* ]
```

Avec «library» indiquant la librairie WTAI à laquelle appartient la fonction. Le tableau ci-dessous donne les valeurs possibles de library

Librairie	Nom	Description
Public WTAI	wp	Fonction publique WTAI
Voice Call Control	vc	Librairie de contrôle des appels
Network Text	nt	Librairie d'envoi et de réception de texte sur le réseau
Phonebook	pb	Librairie permettant de manipuler le phonebook du périphérique
Call Logs	cl	Permet d'accéder aux logs du périphérique
Miscellaneous	ms	Gestion des fonctions diverses

Tableau 2 – Liste des librairies WTAI

Ainsi, l'URI suivante permet d'initier un appel téléphonique vers le numéro 0802802802.

```
wtai://wp/mc; 0802802802
```

Le même appel de fonction en WMLScript serait :

```
WTAPublic.makeCall("0802802802")
```

7.4 Exemple d'utilisation de fonction WTAI depuis un deck WML

L'exemple suivant utilise WML pour proposer à l'utilisateur un menu lui permettant d'appeler directement un fournisseur de nourriture en validant simplement un choix. Ce deck WML pourrait très bien être envoyé à l'agent utilisateur soit par un serveur originel WAE, soit par un serveur WTA.

```
<WML>
  <CARD>
    <DO TYPE="ACCEPT" TASK="GO" URL="eFood"/>
    Bonjour !
  </CARD>

  <CARD NAME="eFood">
    <DO TYPE="ACCEPT" TASK="GO" URL="wtai://wp/mc;$FoodNum"/>
    Choisissez un type de nourriture :
    <SELECT KEY="FoodNum">
      <OPTION VALUE="0802802801">Pizza</OPTION>
      <OPTION VALUE="0802802802">Chinois</OPTION>
      <OPTION VALUE="0802802803">Sandwichs</OPTION>
      <OPTION VALUE="0802802804">Hamburger</OPTION>
    </SELECT>
  </CARD>
```

</WML>

Exemple 3 – Exemple d'appel de fonction WTAI

8 Les acteurs du marchés

Nous allons dans ce chapitre essayer de balayer de manière la plus exhaustive possible l'ensemble des acteurs du monde WAP. Nous irons des fournisseurs d'infrastructure aux fabricants de périphériques en passant par les différents acteurs de ce marché. Toutefois le monde WAP étant en perpétuelle évolution, une remise à jour régulière des informations présentées ici est fortement conseillée.

8.1 Les opérateurs WAP

Les opérateurs WAP sont les fournisseurs de réseau qui permettent aux utilisateurs de périphériques mobiles de s'interconnecter.

- Bouygues Télécom
- SFR – CEGETEL
- France Télécom
- AT&T
- DDI
- BELL SOUTH
- Hong-Kong Télécom
- CELLNET
- SWISSCOM
- SONERA
- IDO
- SPRINT PCS
- TELIA
- VODAFONE
- TIM
- TELENORD
- TELSTRA
- T MOBIL
- ETC.

8.2 Les fabricants de périphériques WAP

Les fabricants de périphériques WAP sont les constructeurs qui intègrent aujourd'hui dans au moins un de leur produit des composantes de l'architecture WAP.

- ALCATEL
- DELARUE
- MOTOROLA
- BOSCH
- ERICSSON

-
- ACER
 - GEM PLUS
 - NOKIA
 - MITSUBISHI
 - INTEL
 - PHILIPS
 - PANASONIC
 - SAMSUNG
 - UNIDEN
 - TELITAL
 - QUALCOMM
 - SCHLUMBERGER
 - Etc.

8.3 Les éditeurs de logiciels WAP

Les éditeurs de logiciels WAP sont des sociétés qui fournissent des logiciels implémentant une ou plusieurs couches de l'architecture WAP.

- APION
- FUJITSU
- GOWORKS
- IBM
- DOLPHIN
- CTC
- PUMA
- STARFISH
- COMVERSE
- SOFTLINE
- TEGIC COMMUNICATION
- SYMBIAN
- SPYGLASS
- UNWIRED PLANET
- SEND IT
- RSA DATA SECURITY
- Etc.

8.4 Les fournisseurs d'infrastructure WAP

Les fournisseurs d'infrastructure sont des entreprises qui ont mis en place une infrastructure mettant en œuvre les protocoles WAP.

- ALCATEL
- MOTOROLA
- IBM

-
- ERICSSON
 - CERTICOM
 - CTC
 - NORTEL NETWORKS
 - COMVERSE
 - CMG
 - LOGICA
 - NOKIA
 - NEC
 - SCHLUMBERGER
 - SIEMENS
 - QUALCOM
 - SEMA GROUP
 - UNISYS
 - TECNOMEN
 - Etc.

8.5 Les membres du forum WAP

- ALCATEL
- APION
- AT&T
- BELL ATLANTIC MOBILE
- BEESOUTH
- BOSCH
- ACER
- CCL
- SFR CEGETEL
- CELINET
- CERTICOM
- CMG
- BOUYGUES TELECOM
- COMVERSE
- CTC
- DDI
- ERICSSON
- IBM
- IDO
- France TELECOM
- FUJITSU
- NEC
- PANASONIC
- MITSUBISHI
- MOTOROLA
- NOKIA

-
- INTEL
 - NORTEL NETWORKS
 - OMNITEL
 - ROGERS
 - PHILIPS
 - PUMA
 - SOFTLINE
 - SONERA
 - SONY
 - STARFISH
 - SPRINT PCS
 - SEND IT
 - SCHLUMBERGER
 - SIEMENS
 - SPYGLASS
 - TEGIC
 - TIM
 - TELENOR
 - SYMBIAN
 - SWISSCOM
 - TECHNOMEN
 - TELENOR UNISYS
 - UNIDEN
 - VODAFONE
 - VTT
 - T MOBIL
 - TELSTRA
 - TELITAL
 - DELARUE
 - GEMPLUS
 - DOLPHIN
 - Etc.

9 Les outils et plates-formes du marché

Un certain nombre de société ont développées des produits implémentant tout ou partie de la pile de protocoles WAP. Il faut discerné à mon sens trois type de logiciel WAP :

- Les agents utilisateurs
- Les passerelles et les proxy
- Les outils d'authoring, de débuggage , etc.

Les navigateurs sont implémentés d'une part par les constructeurs de périphériques (NOKIA, ERICSSON, etc.) WAP, et d'autre part par des éditeurs de logiciel WAP indépendants (phone.com par exemple). Les passerelles elles aussi sont développées soit par des sociétés indépendantes (phone.com), soit par des constructeurs de périphériques et/ou d'infrastructure (NOKIA, ERICSSON, etc.). Bien qu'encore peu nombreux, les outils d'authoring et de débuggage sont aujourd'hui livrés sous la forme de toolkit librement téléchargeable par à peu près tous les acteurs de ce marché (phone.com, NOKIA, etc.).

Aujourd'hui, une marque comme NOKIA, livre certains de ses téléphones portables avec son navigateur "maison", et d'autres avec le navigateur de phone.com auprès de qui NOKIA à acheté une licence.

De même, les opérateurs de réseau semblerais aujourd'hui implémenter plusieurs plate-forme WAP, au moins à des fins d'évaluation. Ainsi la passerelle WAP de phone.com est aujourd'hui déployée chez SFR-CEGETEL, et en cours d'évaluation chez France Télécom et Bouygues Télécom.

L'intérêt pour les fournisseur de contenus WAP est qu'il n'ont pas besoin, a priori, d'implémenter de telle passerelles au sein de leur infrastructure, puisque c'est au opérateurs de réseau de les mettre en place à la frontière de leur réseau mobile, et du réseau Internet. Cependant, bien que toutes les passerelles WAP du marché répondent quasi scrupuleusement aux contraintes des protocoles WAP, des disparités en terme de richesse fonctionnelle tendent à apparaître. En effet, entre deux passerelle WAP, l'une va peut être proposer des services que l'autre ne proposera pas, comme par exemple des mécanismes plus ou moins évolués de gestion de "cookies". Le problème se pose alors au fournisseur de contenu en terme d'interopérabilité de ces applicatifs en fonction des support réseau accédant à son application.

Annexe A Abréviations

API	Application Programming Interface
BNF	Backus-Naur-Form
CGI	Common Gateway Interface
DCS	Digital Communications System
DTD	Document Type Definition
ECMA	European Computer Manufacturer Association
GC	Garbage Collection
GSM	Global System for Mobile Communication
HDML	Handheld Markup Language
HTML	HyperText Markup Language
HTTP	HyperText Transfert Protocol [RFC2068]
IANA	Internet Assigned Numbers Authority
IMC	Internet Mail Consortium
IN	Intelligent Network
IS-136	TDMA Cellular/PCS – Radio Interface – Mobile Station - -Base Station Compatibility Standard
LSB	Least Significant Bit
MMI	Man Machine Interface
MMI	Man-Machine Interface
MSB	Most Significant Bit
MSISDN	Mobil Station International Subscriber Device Number
OS	Operating System
PCS	Personal Communication System
PDA	Personal Digital Assistant
PDC	Personal Digital Cellular
PICS	Protocol Implementation Conformance Statement
RFC	Request For Comments
SGML	Standardised Generalised Markup Language
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
TLS	Transport Layer Security
UCS	Universal Multiple-Octet Coded Character Set
UI	User Interface
URI	Uniform Resource Identifier [RFC2396]
URL	Uniform Resource Locator [RFC2396]
URN	Uniform Resource Name
UTF	UCS Transformation Format
W3C	World Wide Web Consortium
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WBMP	Wireless BitMaP
WDP	Wireless Datagram Protocol
WML	Wireless Markup Language
WSP	Wireless Session Protocol
WTA	Wireless Telephony Application

WTAI	Wireless Telephony Application Interface
WTLS	Wireless Transport Layer Security
WTP	Wireless Transaction Protocol
WWW	World Wide Web
XML	Extensible Markup Language

Annexe B Glossaire

American standard code for information interchange (ASCII)

ASCII est un standard développé par l'ANSI (American National Standards Institute) visant à définir des valeurs compréhensibles par un ordinateur pour les caractères alphabétiques. De nombreuses déclinaisons du code ASCII sont devenues des standards nationaux ou internationaux notamment la famille de standard international ISO-8859, définissant les extensions appropriées à certains groupes de langage. Le membre le plus important de ce groupe étant ISO-8859-1 plus connu sous le nom de ISO-Latin-1 qui fournit les jeux de caractères pour l'Europe de l'ouest.

Attribut

Composante syntaxique d'un élément WML (ou de manière plus générale de tout langage dérivé de SGML, et /ou XML) utilisé le plus souvent comme caractéristique qualifiante de ce même élément.

Auteur

Un auteur est une personne ou un programme qui écrit ou génère du WML, WMLScript, ou tout autre type de contenu.

Bande passante

La bande passante est le débit en bits par seconde que tolère un média. Pour une communication vocale ou analogique, la bande passante est mesurée par la différence entre la fréquence de transmission la plus haute et la fréquence de transmission la plus basse. Ces fréquences sont exprimées en hertz. Pour des communications numériques la bande passante est mesurée en bits par seconde.

Octets

Un octet est une séquence consécutive de huit bits traités comme un seul bloc.

Bytecode

Informations codées dont le contenu est habituellement des instructions de bas niveaux pour des sous ensemble HardWare ou des machines virtuelles.

Card

Unité WML d'interface utilisateur et de navigation. Une carte peut contenir des informations présentées à l'utilisateur et/ou des instructions pour des saisies utilisateur.

Client

Un client désigne une application ou un dispositif qui initie une demande de connexions à un serveur.

Common Gateway Interface (CGI)

CGI définit une interface entre des requêtes utilisateur et des langages de programmation permettant de traiter des problématiques de transmission de données utilisateur.

Concaténation

Concaténer signifie mettre bout à bout deux chaînes de caractères pour en créer une troisième par exemple la concaténation des chaînes «foo » et «bar » donne la chaîne «foobar ».

Contenu

Informations stockées ou générées par un serveur WEB. Le contenu est généralement affiché ou interprété par un agent utilisateur (navigateur) en réponse à une requête utilisateur.

Deck

Collection de cartes WML.

Document type définition (DTD)

Un DTD définit la manière dont les éléments peuvent être imbriqués. Un DTD définit les règles suivantes :

- le nom et le contenu de tous les éléments autorisés dans le cadre de ce DTD.
- combien de fois un élément peut apparaître,
- l'ordre dans lequel les éléments doivent apparaître,
- si oui ou non le tag de fin peut être omis,
- le contenu autorisé pour chacun des éléments,
- les attributs ainsi que leurs valeurs par défaut,
- le nom des symboles de référence qui peuvent être utilisés.

Element

Un élément spécifie le marquage et la structure des informations contenues dans un « WML deck ». Un élément peut contenir un tag de début, un contenu, et un tag de fin.

Extensible Markup Language (XML)

XML est un standard du W3C (World Wide Web Consortium) permettant de définir des applications basées sur un langage de marqueurs. WML est une application de XML. XML est un sous ensemble restreint et enrichie de SGML (Standardized Generalised Markup Language).

Hypertext Transfer Protocol (HTTP)

HTTP est un protocole définissant la manière dont sont véhiculées les informations du Web sur TCP/IP. C'est le protocole de transfert utilisé entre les navigateurs Web et les serveurs Web.

JavaScript

JavaScript est un langage dérivé de la norme ECMAScript utilisé principalement pour intégrer des traitements dans des pages HTML.

Serveur

Désigne une application ou un dispositif qui attend des requêtes de connexion en provenance d'un ou plusieurs clients, et qui en réponse fournit éventuellement de l'information.

Standardized Generalised Markup Language

SGML est un langage générique permettant de définir des langages à base de marqueurs pour des domaines spécifiques. SGML est définie par la norme ISO8879.

Uniform Resource Locator (URL)

Une URL est une adresse référent à un document situé sur le domaine Internet. Une URL est constituée de trois éléments :

- le protocole, ou le langage de communication à utiliser,
- le nom de domaine (ou l'adresse IP) identifiant le serveur,
- le chemin et le nom du document à atteindre sur le serveur.

Wireless Application Environment (WAE)

WAE définit un environnement applicatif basé fondamentalement sur la philosophie et les technologies du World Wide Web. WAE spécifie un environnement permettant aux opérateurs et aux fournisseurs de services de construire des applications et des services pouvant toucher un grand nombre de plates-formes différentes. WAE est un sous-ensemble de WAP.

Wireless Application Protocol (WAP)

WAP définit une architecture applicative ainsi que des protocoles réseaux permettant à des dispositifs mobiles tels que des téléphones, des pagers, des PDAs, etc. d'accéder à des réseaux TCP/IP.

Wireless Markup Language (WML)

WML est un langage dérivé de XML destiné à fournir l'interface utilisateur ainsi que le contenu à des périphériques à faibles bandes passantes comme des téléphones mobiles, des pagers, ou des PDAs.

Wireless Markup Language Script (WMLScript)

WMLScript est un langage utilisé pour apporter de la programmation du côté du périphérique mobile. WMLScript est un sous-ensemble, adapté, de la norme ECMAScript.

Wireless Session Protocol (WSP)

WSP fournit la couche applicative de haut niveau de WAP avec une interface logique pour deux services orientés session. Le premier opère une connexion au travers d'un protocole de transaction, et le second, sans connexion opère quant à lui au travers d'un service de transport de Datagramme qu'il soit sécurisé ou non.

Annexe C Exemples WML

Index

C

CGI _____ 10, 27, 29, 32, 43, 46

E

ECMAScript _____ 21, 47, 48

H

HTML _____ 8, 11, 13, 21, 24, 27, 43, 47

HTTP _____ 7, 9, 10, 11, 13, 17, 18, 19, 23, 24, 29, 43, 47

J

JavaScript _____ 8, 13, 47

P

passerelle _____ 9, 10, 11, 17, 18, 19, 22, 32, 33, 41

passerelle WAP _____ 22, 32, 41

proxy _____ 9, 11, 13, 17, 22, 41

proxy WAP _____ 11, 13

R

réseau _____ 2, 4, 8, 9, 10, 11, 14, 15, 17, 20, 31, 32, 34, 35, 37, 41

S

sécurité _____ 11, 12, 15, 16, 31, 32

support réseau _____ 2, 15, 41

W

WAE _____ 2, 12, 13, 16, 17, 18, 21, 22, 23, 31, 32, 33, 34, 35, 43, 48

WDP _____ 2, 4, 10, 13, 14, 15, 22, 33, 44

WML _____ 2, 3, 4, 6, 7, 9, 11, 13, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 34, 35, 36, 44, 45, 46, 47, 48, 49

WMLScript _____ 2, 3, 7, 9, 13, 17, 19, 20, 21, 22, 23, 30, 31, 34, 35, 45, 48

WSP _____ 2, 10, 13, 14, 17, 18, 19, 22, 23, 44, 48

WTA _____ 2, 3, 4, 11, 13, 17, 20, 21, 22, 31, 32, 33, 34, 35, 44

WTAI _____ 3, 4, 13, 17, 20, 21, 22, 31, 32, 33, 34, 35, 36, 44

X

XML _____ 20, 24, 25, 26, 44, 45, 47, 48