	<b>Référence :</b> KIT-V3.5	<b>Version :</b> 2.18	<b>Page :</b> 1/63
	<b>Destinataires :</b> Editeurs Internet Plus	<b>Date :</b> 02/09/2009	
	<b>Auteur :</b> Fabien MOREAU	<b>Revue</b>	<b>Visa</b>
	<b>Sujet :</b> KIT v3.5		
<b>Type de document :</b> Descriptif technique général <b>Titre du document :</b> Documentation du Kit v3.5 (Abonnement) w-HA pour Internet+			

Version	Date	Auteur	Action	Motifs
2.00	04/10/2007	FM	Ajout	Ajouts de nouvelles fonctionnalités liées à la nouvelle version du Kit v3.5 (1.2)
2.01	09/10/2007	FM	Correction	Corrections suite à relecture + précisions sur les logs
2.02	24/10/2007	FM	Correction	Corrections suite à remarques de FT R&D
2.03	19/11/2007	FM	Ajout	Ajout sur la méthode d'acquiescement de la servlet /responder lors d'une notification de résiliation (§ 0)
2.04	26/11/2007	FM	Précision	Précisions sur le format (calcul hmac) de l'acquiescement de la servlet /responder lors d'une notification de résiliation (§ 0)
2.05	15/04/2008	LM	Ajout	Ajout sur le test du Responder Modification des exemples d'URL Modification de la structure et l'exemple du fichier « web.xml » de la servlet « responder » Modification de la structure et l'exemple du fichier « merchants.xml » de la servlet « responder »
2.06	22/09/2008	LM	Correction	Modification de la méthode pour afficher les écrans w-HA (choix de l'opérateur, authentification, panneau de paiement) en page courante
2.07	15/12/2008	AD	Correction	Resiliation online (§ 2.3.2.1) choix du FAI : route
2.08	16/04/2009	AD	Correction	Acquiescement (§3.2.4.6) suppression du « ; » à la fin du message : c=ack
2.09	28/05/2009	AD	Correction	Exemple page courante abonnement
2.10	29/05/2009	AD	Suppression	suppression Test responder et Contrôle d'accès pull
2.11	03/07/2009	AD	Correction	Accès MSCA + procédure création compte
2.12	11/08/2009	AD	Ajout	Résiliation et remboursements serveur à serveur
2.13	18/08/2009	AD	Homogénéisation	Homogénéisation doc acte & abo
2.14	28/08/2009	AD	Ajout	paramétrage des url dans le kit (§ ANNEXE III)
2.15	02/09/2009	AD	Correction	url appelée par le kit (urlRedirect) (§ 3.2.4.5.)
2.16	19/10/2009	RC	Correction	mise à jour des nœuds w-HA + remboursements / résiliations
2.17	22/03/2010	AD	Modification	Schéma notification de résiliation
2.18	07/06/2010	AT	Ajout	Ajout détails résiliation effective (§ 3.2.4.1)



## Sommaire

<b>Sommaire</b> .....	<b>2</b>
<b>Introduction</b> .....	<b>5</b>
<b>Objet du document</b> .....	<b>5</b>
<b>1 - Principes généraux du système w-HA</b> .....	<b>6</b>
1.1 - Vocabulaire .....	6
1.2 - Schémas de principe (aspect commercial et marketing) .....	7
1.2.1 - Processus d'abonnement (temps réel) vu du client .....	7
1.2.2 - Flux financiers (différé) .....	8
1.3 - Cinématique d'achat via le système de paiement w-HA (aspect technique) .....	9
1.4 - Inscription, Authentification et Paiement de l'Internaute .....	9
1.4.1 - Inscription .....	9
1.4.2 - Authentification .....	10
1.4.3 - Paiement .....	10
1.5 - Les 3 grandes phases de l'abonnement .....	10
1.5.1 - La Souscription (début de l'abonnement) .....	10
1.5.2 - Les Contrôles d'Accès ou Consommation (vie de l'abonnement) .....	10
1.5.3 - La Résiliation ou Expiration (fin de l'abonnement) .....	10
<b>2 - Contrôle d'accès réalisé par l'éditeur</b> .....	<b>11</b>
2.1 - Souscription (début de l'abonnement) / Contrôle accès EdS: .....	11
2.1.1 - Etape #1 : Demande d'autorisation de souscription en utilisant la servlet : /pos_bundle?action=authorizeOffer .....	11
2.1.2 - Etape #2 : 1 <sup>ère</sup> Mise à Jour de la base de données EdS .....	12
2.1.3 - Etape #3 : Confirmation de souscription, en utilisant la servlet : /pos_request?action=offerConfirm ..	13
2.1.4 - Etape #4 : 2 <sup>nde</sup> Mise à Jour de la base de données EdS .....	13
2.2 Contrôle d'accès / consommation (vie de l'abonnement) .....	14
2.2.1 - L'internaute n'a souscrit aucune Offre contenant ce Produit .....	14
2.2.2 - L'internaute a déjà souscrit une (ou plusieurs) Offre(s) contenant ce Produit .....	14
2.3 - Rbst / Résil. / Expiration (fin de l'abonnement) .....	15
2.3.1 - Remboursement d'une transaction liée à un Abonnement .....	15
2.3.2 - Résiliation d'un Abonnement .....	16
2.3.2.1 - Résiliation d'un Abonnement par l'Internaute (online) .....	16
2.3.2.2 - Résiliation d'un Abonnement par le Service Clients Orange Internet .....	17
2.3.3 - Mode « push » : notifications de résiliation .....	17
<b>3 - Détail du fonctionnement des différentes fonctionnalités / servlets</b> .....	<b>18</b>
3.1 - Servlets avec redirection de l'internaute vers la plate-forme w-HA .....	18
3.1.1 - Demande Souscription / « pos_bundle?action=authorizeOffer » .....	18
3.1.1.1 - Appel de la servlet .....	18
3.1.1.2 - Exemple d'appel de la servlet .....	18
3.1.1.3 - Paramètres d'appel de la servlet .....	19
3.1.1.4 - Fonctionnement interne de la servlet .....	19
3.1.1.5 - Exemple d'URL générée par la servlet .....	20
3.1.1.6 - Réponse de w-HA à la servlet .....	20
3.1.1.7 - Prise en compte de la souscription (fulfillmentUrl de l'Offre) .....	24
3.1.2 - Rechargement des fichiers de configuration locaux / « pos_bundle?action=reload » .....	25
3.2 - Servlets pour les requêtes de serveur à serveur .....	25
3.2.1 - Confirmation « différée » / pos_request?action=offerConfirm .....	26
3.2.1.1 - Appel de la servlet .....	26
3.2.1.2 - Exemple d'appel de la servlet .....	26
3.2.1.3 - Paramètres d'appel de la servlet .....	27
3.2.1.4 - Fonctionnement interne de la servlet .....	27
3.2.1.5 - Exemple d'URL générée par la servlet .....	27
3.2.1.6 - Réponse de la plate-forme w-HA à une requête de confirmation différée .....	28
3.2.1.7 - Exemple d'utilisation de la servlet : formulaire web .....	28
3.2.2 - Résiliation d'un Abonnement / « pos_request?action=cancelOffer » .....	29
3.2.2.1 - Appel de la servlet .....	29
3.2.2.2 - Exemple d'appel de la servlet .....	29
3.2.2.3 - Paramètres d'appel de la servlet .....	29
3.2.2.4 - Fonctionnement interne de la servlet .....	30
3.2.2.5 - Exemple d'URL générée par la servlet .....	30
3.2.2.6 - Réponse de w-HA à la servlet .....	31
3.2.2.7 - Exemple d'utilisation de la servlet : formulaire web .....	31



3.2.3 - Remboursement d'une transaction / « pos_request ?action=refund » .....	33
3.2.3.1 - Appel de la servlet .....	33
3.2.3.2 - Exemple d'appel de la servlet .....	33
3.2.3.3 - Paramètres d'appel de la servlet .....	33
3.2.3.4 - Fonctionnement interne de la servlet .....	34
3.2.3.5 - Exemple d'URL générée par la servlet .....	34
3.2.3.5 - Réponse de w-HA à la servlet .....	35
3.2.3.7 - Exemple d'utilisation de la servlet : formulaire web .....	35
3.2.4 - Notification de résiliation / « responder ?c=NMPOC_NEW » .....	37
3.2.4.1 - Appel de la servlet .....	38
3.2.4.2 - Exemple d'appel de la servlet .....	40
3.2.4.3 - Fonctionnement interne de la servlet .....	41
3.2.4.4 - Parsing du fichier « log_notification_XXX.txt » .....	41
3.2.4.5 - Redirection des notifications vers une URL .....	42
3.2.4.6 - Réponse de la servlet à w-HA : acquittement .....	42
3.2.4.7 - En cas de non réponse de la servlet .....	43
<b>4 - Ce que le marchand doit mettre en œuvre pour utiliser le Kit v3.5 .....</b>	<b>44</b>
4.1 - Création des offres via le MSCA .....	44
4.2 - Paramétrage des fichiers de configuration « *.xml » .....	44
4.3 - Confirmation de souscription .....	45
4.4 - Notifications de résiliation : mode « push » .....	45
4.5 - Réalisation de pages Web de présentation des offres .....	45
4.6 - Affichage des écrans w-HA en page courante .....	46
4.7 - Protection du contenu (utilisation du hmac) .....	48
<b>5 – ANNEXE I : SDK .....</b>	<b>50</b>
5.1 - Développement d'un « confirm » de souscription via le SDK .....	50
5.1.1 - Implémentation .....	50
5.1.2 - Exemple de code .....	51
5.2 - Développement d'un « responder » spécifique via le SDK .....	51
5.2.1 - Implémentation .....	52
5.2.2 - Exemple de code : .....	52
<b>6 - ANNEXE II : Pré-requis techniques pour le fonctionnement du Kit v3.5 .....</b>	<b>53</b>
6.1 - Plate-forme d'hébergement .....	54
6.1.1 - Système d'exploitation .....	54
6.1.2 - Machine Virtuelle Java .....	55
6.1.3 - Module de gestion du protocole SSL : « JSSE » .....	55
6.1.4 - Moteur de Servlet .....	55
6.2 - Considérations Réseau .....	56
6.2.1 - Pendant l'installation de l'application w-HA .....	56
6.2.2 - En production .....	56
6.3 - Autres pré-requis pour l'intégration .....	56
6.3.1 - Utilitaire de décompression .....	56
6.3.2 - Redémarrage ("re-boot") du Serveur Web .....	56
6.3.3 - Présence de l'administrateur système .....	56
<b>7 - ANNEXE III : Composants du Kit v3.5 .....</b>	<b>57</b>
7.1 - Servlets « pos_bundle » et « pos_request » .....	59
7.1.1 - Fichier de configuration « web.xml » .....	59
7.1.1.1 - Structure du fichier « web.xml » .....	59
7.1.1.2 - Exemple de fichier « web.xml » .....	60
7.1.2 - Fichier de configuration « productbundle_XXX.xml » .....	62
7.1.2.1 - Structure du fichier « productbundle_XXX.xml » .....	62
7.1.2.2 - Exemple de fichier « productbundle_XXX.xml » .....	62
7.1.3 - Fichier de configuration « merchants.xml » .....	63
7.1.3.1 - Structure du fichier « merchants.xml » .....	63
7.1.3.2 - Exemple de fichier « merchants.xml » .....	63
7.1.4 - Fichier de log « log_request_XXX.txt » .....	64
7.1.4.1 - Structure du fichier « log_request_XXX.txt » .....	64
7.1.4.2 - Exemple de fichier « log_request_XXX.txt » .....	64
7.1.5 - Fichier de log « log_response_XXX.txt » .....	65
7.1.5.1 - Structure du fichier « log_response_XXX.txt » .....	65
7.1.5.2 - Exemple de fichier « log_response_XXX.txt » .....	65
7.2 - Servlet « responder » .....	66
7.2.1 - Fichier de configuration « web.xml » .....	66
7.2.1.1 - Structure du fichier « web.xml » .....	66
7.2.1.2 - Exemple de fichier « web.xml » .....	66



7.2.2 - Fichier de configuration « merchants.xml » .....	67
7.2.2.1 - Structure du fichier « merchants.xml » .....	67
7.2.2.2 - Exemple de fichier « merchants.xml » .....	67
7.2.3 - Fichier de log "log_notification_XXX.txt".....	68
7.2.3.1 - Structure du fichier de log « log_notification_XXX.txt » .....	68
7.2.3.2 - Exemple de fichier de log « log_notification_XXX.txt » :.....	68



## Introduction

Afin de répondre à la demande du marché, la solution Internet+ ([www.internetplus.fr](http://www.internetplus.fr)) évolue pour gérer non seulement les **paiements à l'acte**, mais également les **abonnements récurrents**.

Cette évolution s'accompagne de quelques modifications au niveau de la plate-forme w-HA : mise en œuvre d'une nouvelle application (v3.5), mise à disposition des éditeurs d'une nouvelle interface Web permettant de créer les offres et les produits (**MSCA – Merchant Self Care Application**).

La **base produit** de l'éditeur, c.à.d. l'ensemble des informations (identifiant, prix, description, modalités d'achat, ...) relatives à ses offres et produits est alors présente au niveau de la **plate-forme w-HA**, et non plus au niveau de l'application w-HA (Kit v3.5) installée sur son (ses) serveur(s).

Il reste cependant un **catalogue produit**, limité à un identifiant et à une URL de livraison (ffUrl), **au niveau du Kit v3.5**, permettant de faire le lien avec la base produit.

**La nouvelle application w-HA (Kit 3.5)** installée sur le(s) serveur(s) de l'éditeur présente, en outre, de nombreux avantages, parmi lesquels :

- **multi-types de paiement : acte et abonnement**
- **multi-marchands/boutiques : un seul Kit pour plusieurs marchands ou boutiques**

## Objet du document

L'objet du document est de décrire le fonctionnement du Kit v3.5 en relation avec la plate-forme w-HA, dans le cadre de l'offre Internet+, pour gérer les achats de type acte et abonnement.

Il est principalement destiné :

- à l'**administrateur système**, en charge de l'installation du Kit v3.5
- au **responsable du développement**, en charge de l'intégration de la solution de paiement au sein de l'application de l'éditeur.

Un autre document intitulé « Manuel d'utilisation de l'interface de gestion des Offres d'abonnement (MSCA) » est destiné au(x) gestionnaire(s) de la boutique.



## 1 - Principes généraux du système w-HA

Ce paragraphe a pour objet d'identifier les "acteurs" impliqués lors d'une transaction avec w-HA et de décrire simplement les flux (flux d'information et flux financiers) qui circulent entre ces différents acteurs.

### 1.1 - Vocabulaire

#### **w-HA :**

Le terme "w-HA" peut désigner, selon le contexte :

- la société w-HA,
- la plate-forme technique w-HA,

#### **Marchand :**

Le "Marchand" (ou Editeur de Services) est un vendeur de biens téléchargeables ou services accessibles sur Internet.

#### **Opérateur Client :**

L' "Opérateur Client" est une entité qui a une relation commerciale et financière avec des clients, et qui a établi un partenariat avec w-HA.

Il peut s'agir :

- d'un Fournisseur d'Accès à Internet (FAI) :  
Ex : Orange, Alice, SFR, Mobistar
- d'un opérateur de téléphonie fixe :  
Ex : France Telecom
- d'un opérateur de téléphonie mobile :  
Ex : Orange France, Orange Réunion, Mobistar
- d'un opérateur de paiement sur carte bancaire :  
Ex : Orbeo

Pour plus d'informations sur les différentes offres commerciales s'appuyant sur la technologie w-HA, vous pouvez envoyer un mail à [contact.marchand@w-ha.com](mailto:contact.marchand@w-ha.com).

#### **Internaute :**

L'Internaute désigne le client final qui souscrit, sur Internet, à un abonnement en utilisant la solution w-HA.

#### **L'offre (oid) :**

Il s'agit d'un modèle d'abonnement, définissant notamment une période d'abonnement et un tarif, pour l'accès à un ou plusieurs produits ou services.

#### **L'abonnement (uoid) :**

Il s'agit d'une instance de l'offre, rattachée à un utilisateur.

#### **La transaction (trxid) :**

La souscription à un abonnement crée une transaction. Chaque reconduction crée également une transaction. Les transactions sont facturées à l'internaute.

#### **En résumé :**

Un internaute souscrit à une offre (modèle), il possède alors un abonnement (instance) valide. Il peut alors accéder aux produits ou services pendant la période de validité de l'abonnement.

## 1.2 - Schémas de principe (aspect commercial et marketing)

Les schémas de principe du système w-HA sont donnés ci-après.

### 1.2.1 - Processus d'abonnement (temps réel) vu du client

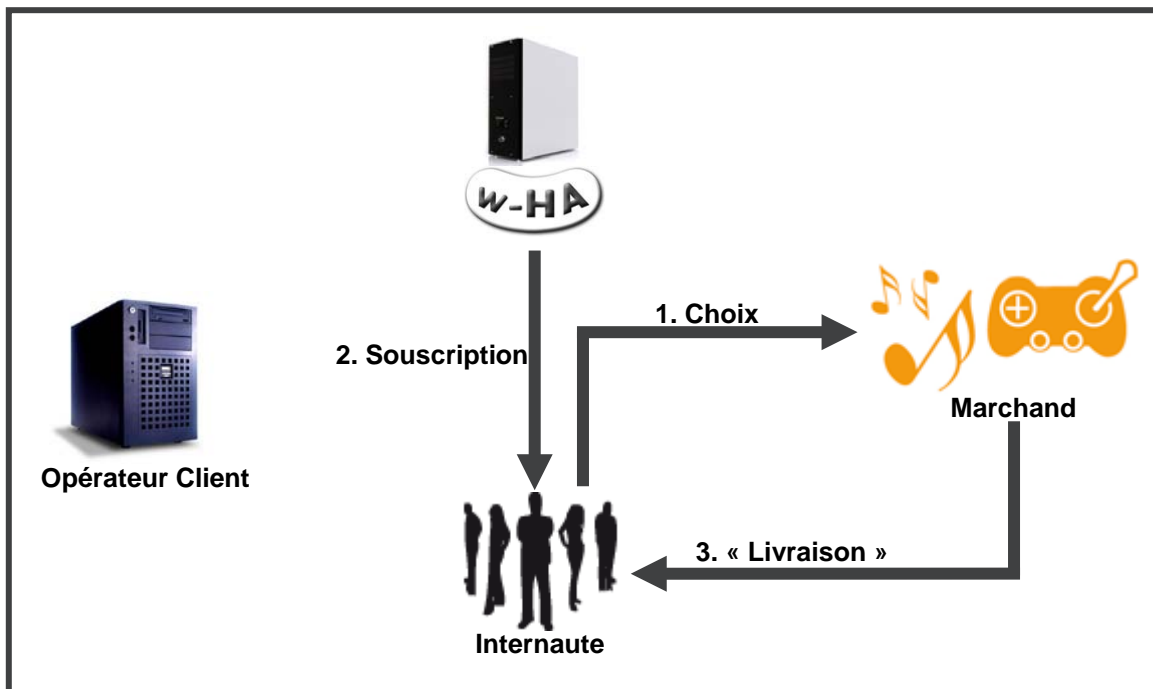


figure 1 : processus de souscription (vu de l'Internaute)

La souscription à l'abonnement vu de l'Internaute (en temps réel) est le suivant :

1. Ecran "marchand" : L'Internaute choisit un abonnement sur le site Web du marchand
2. Ecran "w-HA" : W-HA lui affiche une « panneau de paiement », sur lequel l'Internaute va confirmer son abonnement
3. Ecran "marchand" : Après autorisation de w-HA, l'Internaute est redirigé vers la page Web du marchand pour accéder au produit ou service auquel il a souscrit.

#### **Attention :**

Seuls les écrans Web vus par l'Internaute sont indiqués ici. En réalité, il existe des flux d'informations complémentaires entre les différents acteurs de la transaction. Le détail de ces flux est donné au paragraphe 1.3.

### 1.2.2 - Flux financiers (différé)

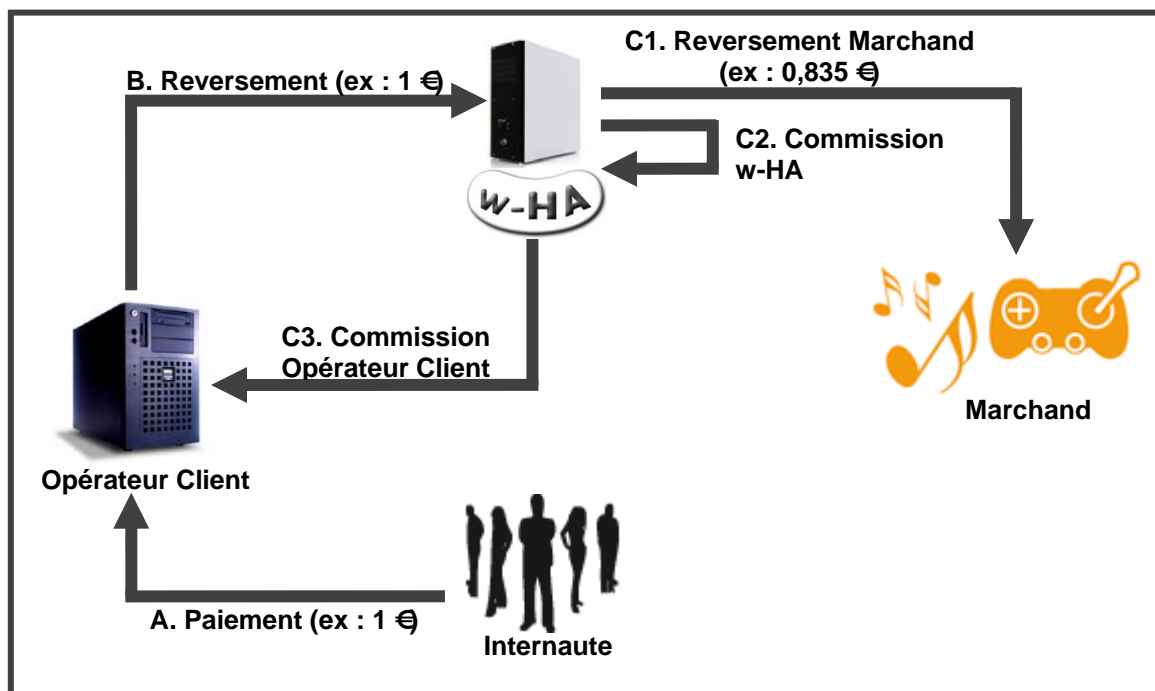


figure 2 : flux financiers entre les différents acteurs de la transaction

Les flux financiers entre les différents acteurs de la transaction (en différé) sont les suivants :

#### **Exemple pour un achat d'un euro :**

- A. L'Internaute est facturé par l'Opérateur Client (1 euro)
- B. L'Opérateur Client reverse l'intégralité de la somme facturée à w-HA
- C1. W-HA reverse au marchand une partie de la somme facturée (0,835 euro)
- C2. W-HA conserve une commission
- C3. W-HA reverse une commission à l'Opérateur Client



### 1.3 - Cinématique d'achat via le système de paiement w-HA (aspect technique)

Le schéma détaillé des flux d'informations, lors d'un achat via le système w-HA, est donné ci-après :

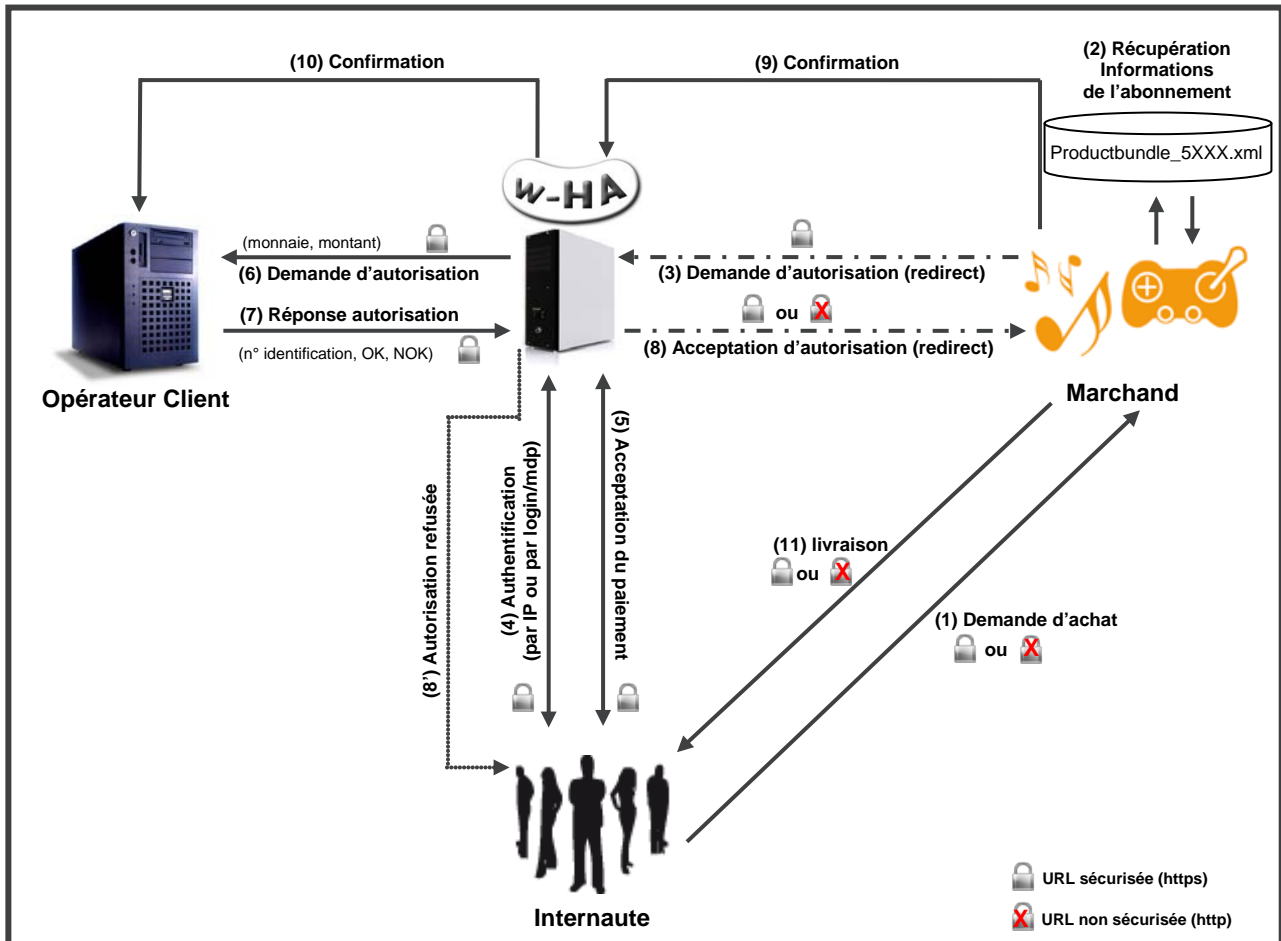


figure 3 : détail des flux d'information entre les différents acteurs de la transaction

## 1.4 - Inscription, Authentification et Paiement de l'Internaute

### 1.4.1 - Inscription

#### FAIs partenaires de w-HA

Dans la majorité des cas, l'**inscription** de l'internaute au système de paiement w-HA auprès de son FAI est **automatique** (optin) lors de la création de son accès internet. L'Internaute peut ensuite, s'il le souhaite, désactiver cette possibilité.

Il a également la possibilité d'**accepter ou non les paiements nomades** (cf. § suivant)

#### Autres FAIs

Si l'internaute est abonné à Internet auprès d'un **FAI non partenaire de w-HA**, ou s'il souhaite disposer d'un moyen de paiement supplémentaire il peut ouvrir un compte **Orbeo** avec sa **carte bancaire**, depuis le site <http://www.w-ha.com>.



### 1.4.2 - Authentification

#### Connexion Internet via FAI partenaire

Si l'internaute est connecté avec un accès d'un **FAI partenaire**, au moment de l'achat, celui-ci est **reconnu automatiquement via son adresse IP** et est immédiatement renvoyé vers le « panneau de paiement », aux couleurs de son FAI

#### Connexion Internet via un autre FAI

Si l'internaute est connecté avec un **autre accès Internet** (FAI non partenaire, ou connexion d'entreprise, par exemple), il se voit afficher un « **panneau de choix des opérateurs clients** ».

S'il a accepté les **paiements nomades**, il pourra choisir son FAI et **s'authentifier via login/mdp** avant d'être redirigé vers le « panneau de paiement ».

Il pourra également décider d'effectuer l'achat avec son **compte Orbeo**, à condition, bien sûr que le marchand ait choisi de proposer ce moyen de paiement, lors de la signature du contrat. L'internaute devra alors s'authentifier en utilisant le login/mdp choisi lors de l'ouverture de son compte Orbeo.

### 1.4.3 - Paiement

Une fois authentifié (automatiquement par son adresse IP ou après saisie d'un login/mdp), l'internaute est redirigé vers le « panneau de paiement ».

Ce panneau de paiement est aux couleurs de l'opérateur client (FAI ou Orbeo) et affiche les informations principales suivantes :

- nom du marchand
- logo du marchand
- descriptif de l'abonnement
- périodicité de l'abonnement
- montant de l'abonnement

L'internaute n'est qu'à **1 seul clic de sa souscription** (bouton de confirmation de l'abonnement), ce qui fait de w-HA l'une des solutions de paiement procurant le **meilleur taux de transformation du marché !**

## **1.5 - Les 3 grandes phases de l'abonnement**

### 1.5.1 - La Souscription (début de l'abonnement)

L'événement déclencheur de l'Abonnement est la **souscription à une Offre**.

L'internaute, en échange du paiement de son abonnement, obtient des droits d'accès à un produit ou service, dont les conditions sont définies dans l'Offre (en termes de durée, ...)

### 1.5.2 - Les Contrôles d'Accès ou Consommation (vie de l'abonnement)

Une fois que l'internaute dispose d'un Abonnement, il est nécessaire, à chaque fois que celui tente d'accéder au produit ou service constitutif de l'Offre, de réaliser un **contrôle d'accès**.

Dans le cadre de l'offre Internet+, le **contrôle d'accès est réalisé par l'éditeur** lui-même.

### 1.5.3 - La Résiliation ou Expiration (fin de l'abonnement)

Un abonnement se termine lorsqu'un des événements suivants se produit :

- l'Abonnement **arrive à son terme**,
- l'internaute demande une **Résiliation** de son Abonnement,
- l'internaute demande le **Remboursement** de son Abonnement



## 2 - Contrôle d'accès réalisé par l'éditeur

Lorsque le contrôle d'accès est réalisé par l'éditeur, celui-ci doit mettre en place une base de données sur son serveur, avec l'état de chaque abonnement.

Il est donc en mesure de suivre en temps réel le nombre d'abonnements en cours, le nombre de nouvelles souscriptions, le nombre de résiliations, grâce à son Back-Office (à développer).

De son côté w-HA gère également une base de données des abonnements pour chaque éditeur et envoie des notifications à l'éditeur lors des changements de statut.

Cela permet à l'éditeur d'assurer la synchronisation des bases de données (cf. § 0 et § **Erreur ! Source du renvoi introuvable.**)

En cas de désynchronisation des bases, il est de la **responsabilité de l'éditeur de faire en sorte que l'état de sa base corresponde à celle de w-HA**, étant donné que c'est la **base de données de w-HA qui fait foi**.

### 2.1 - Souscription (début de l'abonnement) / Contrôle accès EdS:

L'éditeur doit **associer** son propre **identifiant d'utilisateur** (ex : userId) à **l'identifiant de l'Abonnement** (uoid) attribué par w-HA.

L'internaute choisit un identifiant et un mot de passe, et l'application de l'éditeur y associe un identifiant d'utilisateur propre, par exemple UserId :

Exemple :

Login : [fabien.moreau@orange-ftgroup.com](mailto:fabien.moreau@orange-ftgroup.com)  
Password : \*\*\*\*\* } userID = fmoreau1234

La souscription à une Offre se fait en 4 étapes :

#### 2.1.1 - Etape #1 : Demande d'autorisation de souscription en utilisant la servlet : /pos\_bundle?action=authorizeOffer

#### Requêtes en "foreground", re-direction via le navigateur web de l'internaute

##### Cinématique orientée Offre : "souscription à une offre"

Souscription Offre #1 (01) (autoconfirm = 'true')	<input type="button" value="Valider"/>	/pos_bundle?action=authorizeOffer	Internetplus & Portail
Souscription Offre #2 (02) (autoconfirm = 'false')	<input type="button" value="Valider"/>	/pos_bundle?action=authorizeOffer	Internetplus & Portail
Souscription Offre #3 (03) (n'existe pas dans productbundle.xml)	<input type="button" value="Valider"/>	/pos_bundle?action=authorizeOffer	Internetplus & Portail
Souscription Offre #4 (04) (n'existe pas dans MSCA)	<input type="button" value="Valider"/>	/pos_bundle?action=authorizeOffer	Internetplus & Portail

#### **Attention !**

Une même Offre peut-être souscrite plusieurs fois par le même internaute.

Il convient donc que l'éditeur vérifie en amont si l'internaute est déjà abonné, et ne pas lui donner accès au lien de souscription, le cas échéant.



L'internaute est re-dirigé vers la plate-forme w-HA qui affiche un panneau de souscription.

### Exemple :

Paielement - Mozilla Firefox

https://wanadoo.w-ha.com/app-bundlepurchase/node?ReqID=b384bcd17304989bffa5f4b2a13222861fb0cbcbg=en

orange internet+ PLUS

**Votre commande**

Produit : Abonnement mensuel : Offre Liberté 1 enfant à 9,99 € par mois Edité par : Maxicours

Service : Maxicours

Prix : 9,99 € TTC chaque mois

Afficher les conditions spécifiques de cette offre

**Confirmer votre achat**

Le montant de votre achat sera ajouté à votre prochaine facture Internet Orange \*. Vous ne souhaitez pas souscrire ce service : > **annulez la commande**

\* service acheté aux Conditions de paiement Internet Orange que vous acceptez en confirmant votre achat.

© Copyright France Télécom Technologie w-HA

### Attention !

Lorsque l'internaute clique sur le bouton « Confirmer votre achat », et en l'absence d'erreur (ex : dépassement de plafond), l'**Abonnement** (uoid) est dans l'**état autorisé** dans la base de données w-HA, et à ce stade, **ne fera pas l'objet d'une facturation** (dans l'**attente du « confirm »** de l'étape #2).

### 2.1.2 - Etape #2 : 1<sup>ère</sup> Mise à Jour de la base de données EdS

Une fois la demande d'autorisation de souscription acquittée par la plate-forme w-HA, l'éditeur peut **mettre à jour sa base de données** avec l'**identifiant de l'Abonnement**, en précisant, par exemple :

\* son **statut** :

- Authorized (bouton « Confirmer votre achat »)
- Cancelled (lien « annulez la commande »)

\* sa **date de fin de validité**.

Par exemple :

O1 et O2 sont des Offres d'une durée d'un 1 mois à Tacite Reconduction (TR)

UserId	UserOfferId (uoid)	oid	pid	Status	Date Souscription	Date Reconduction	Date Fin
fmoreau12	6-U7141248844587211	O1	P1	Authorized	21/8/7 12:16	21/9/7 12:16	null
msantos012	6-U8241248844587211	O2	P2	Authorized	22/8/7 15:12	22/9/7 15:12	null
fgalle45	none	O3	P1	Cancelled	null	null	null
gcronimus	6-U8223448846757123	O5	P2	Authorized	23/8/7 23:34	23/11/7 23:34	null
fgalle45	none	O1	P1	Cancelled	null	null	null
fmoreau12	6-U0441248844587211	O3	P1	Authorized	24/8/7 07:12	24/9/7 07:12	null



### 2.1.3 - Etape #3 : Confirmation de souscription, en utilisant la servlet : /pos\_request?action=offerConfirm

L'éditeur envoie une requête de confirmation (serveur à serveur) vers la plate-forme w-HA

Exemple :

#### Requête en "background" : Confirmation différée de souscription

Si le paramètre "autoConfirm" est égal à "false" dans le fichier "productbundle\_XXX.xml", lorsque l'internaute accepte la souscription à l'offre, il est redirigé vers la servlet "pos\_bundle" qui le redirige vers la "fulfillmentUrl" (cette page).

La fulfillmentUrl doit être le point d'entrée dans la base de données de l'éditeur de service, pour la prise en compte de la souscription.

L'éditeur doit ensuite confirmer la souscription à w-HA (requête en background) via la servlet "/pos\_request?action=xxxxx"

La souscription est alors validée à la fois côté éditeur et côté w-HA, **évitant ainsi la désynchronisation des bases.**

L'appel de la servlet "/pos\_request?action=offerConfirm" peut être simulée via le formulaire ci-dessous :

merchantId (mid)	<input type="text" value="ex : 12"/>
identifiant abonnement (uoid)	<input type="text" value="ex : 6-U4111235544454210"/>
url requête (url)	<input type="text" value="ex : https://wanadoo.w-ha.com/app-node-mct/responder"/>
<input type="button" value="Confirm"/>	

#### **Attention !**

Afin de pouvoir **décoreller l'autorisation et la confirmation**, le paramètre « **autoconfirm** » de l'Offre en question doit être égal à « **false** », dans le fichier « **productbundle\_XXX.xml** »

#### **Attention !**

C'est seulement à réception de cette requête de « **confirm** » que l'**Abonnement** passe dans l'**état confirmé** dans la base de données w-HA, et qu'il fera l'objet d'une **facturation** de l'internaute.

#### **Attention !**

A partir du moment où un abonnement a été autorisé, l'EdS dispose d'un délai de **24 heures maximum pour « confirmer » cet abonnement.**

Au-delà de ce délai, il passe dans l'état annulé et ne pourra plus être confirmé.

### 2.1.4 - Etape #4 : 2<sup>nde</sup> Mise à Jour de la base de données EdS

C'est seulement après réception de l'acquiescement du « confirm » venant de la plate-forme w-HA, que la souscription à l'Abonnement doit être considérée comme effective.

L'éditeur peut alors **mettre à jour sa base de données**, par exemple, **en modifiant le statut de l'Abonnement :**

Du statut **Authorized**, l'Abonnement passe au statut **Confirmed**

Par exemple :

O1 et O2 sont des Offres d'une durée d'un 1 mois à Tacite Reconduction (TR)

UserId	UserOfferId (uoid)	oid	pid	Status	Date Souscription	Date Reconduction	Date Fin
fmoreau12	6-U7141248844587211	O1	P1	Confirmed	21/8/7 12:16	21/9/7 12:16	null
msantos012	6-U8241248844587211	O2	P2	Confirmed	22/8/7 15:12	22/9/7 15:12	null
fgalle45	none	O3	P1	Cancelled	null	null	null
gcronimus	6-U8223448846757123	O5	P2	Confirmed	23/8/7 23:34	23/11/7 23:34	null
fgalle45	none	O1	P1	Cancelled	null	null	null
fmoreau12	6-U0441248844587211	O3	P1	Confirmed	24/8/7 07:12	24/9/7 07:12	null

## 2.2 Contrôle d'accès / consommation (vie de l'abonnement)

Il s'agit de la cinématique orientée Produit.

La cinématique orientée Produit est une cinématique pour laquelle l'internaute cherche à **accéder (ou consommer) un produit/service**.

Deux cas sont alors à considérer :

- A - l'internaute n'a souscrit à aucune Offre contenant ce produit/service
- B - l'internaute a déjà souscrit à (au moins) une Offre contenant ce produit/service

Il est de la responsabilité de l'EdS de contrôler cela dans sa base de données, au moment où l'internaute tente d'accéder au produit/service en question.

### **Attention !**

Si l'éditeur ne fait pas le contrôle d'accès alors, l'internaute peut souscrire plusieurs fois à un même abonnement !

### **2.2.1 - L'internaute n'a souscrit aucune Offre contenant ce Produit**

Si l'internaute n'a souscrit aucune Offre contenant ce Produit, il est **à la charge de l'éditeur** d'afficher une page à l'internaute : en appelant la servlet /pos\_bundle?action=authorizeOffer (§ [3.1.1 - Demande Souscription / « pos\\_bundle?action=authorizeOffer »](#))

### **2.2.2 - L'internaute a déjà souscrit une (ou plusieurs) Offre(s) contenant ce Produit**

Si l'internaute a souscrit une (ou plusieurs) Offre(s) contenant ce Produit, et que celle(s)-ci est (sont) valide(s), **l'éditeur donne directement accès au Produit, sans re-direction vers la plate-forme w-HA.**

### **Attention !**

Il est de la responsabilité de l'éditeur de vérifier si l'abonnement de l'internaute est valide.

Bien entendu, il est de la **responsabilité de l'éditeur** de vérifier l'accès (ou consommation) possible du Produit dans une Offre et de mettre à jour sa base de données le cas échéant, selon qu'il s'agit d'un Produit :

- à accès illimité
- à accès unique
- à N accès
- .....

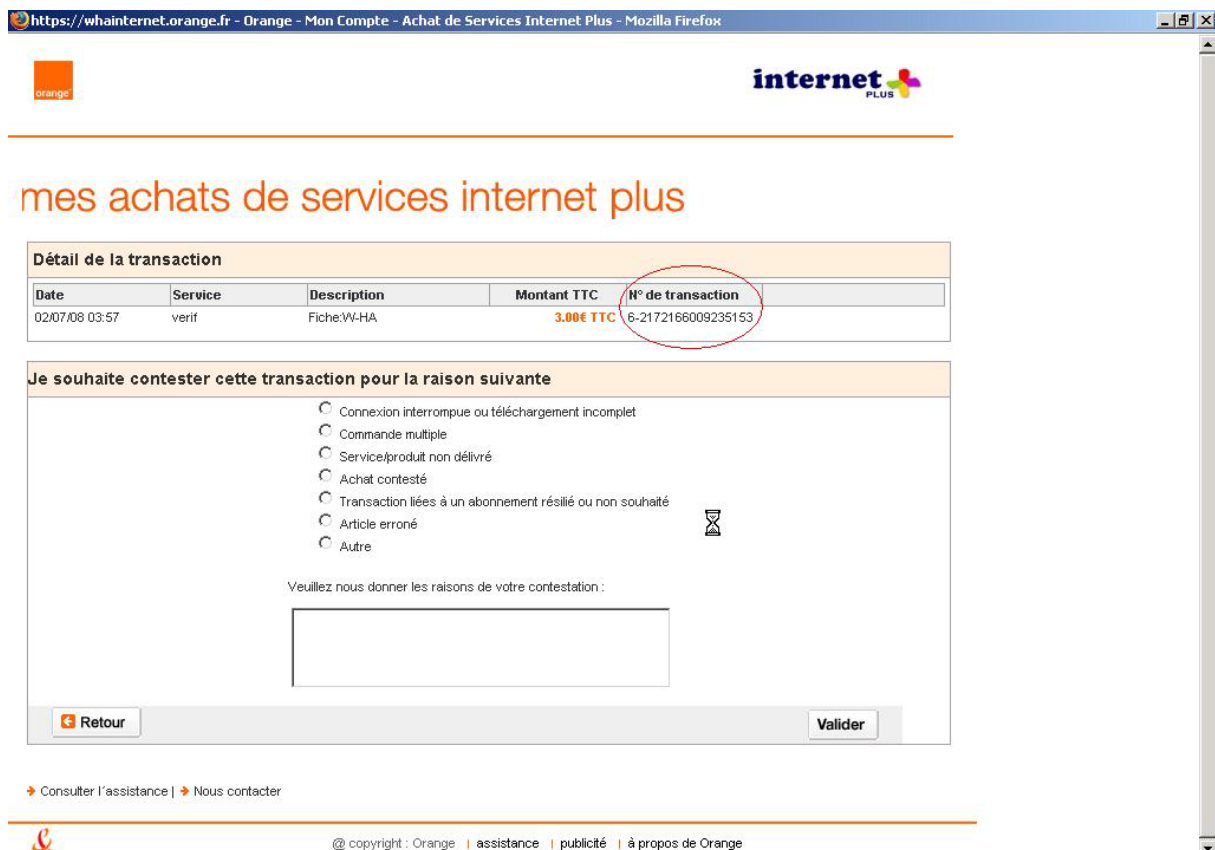
## 2.3 - Rbst / Résil. / Expiration (fin de l'abonnement)

Ce paragraphe traite de la « fin de vie d'un Abonnement » et a pour objet d'expliquer comment l'éditeur doit mettre à jours sa base de données lorsque cet événement se produit.

### 2.3.1 - Remboursement d'une transaction liée à un Abonnement

Un Internaute Orange Internet (par exemple) peut demander, **en ligne**, le remboursement d'une transaction, depuis le portail [www.orange.fr](http://www.orange.fr) > espace client > conso internet (voir le détail) > Achats de service internet plus (voir le détail) > Consulter votre relevé détaillé (accès direct : <https://wanadoo.w-ha.com/app-am/node>)

Dans l'onglet « **Mes achats** », un clic sur le lien « voir » aboutit à l'écran suivant :



The screenshot shows a web browser window with the URL <https://whainetnet.orange.fr>. The page title is "Orange - Mon Compte - Achat de Services Internet Plus - Mozilla Firefox". The page content includes the Orange logo and "internet PLUS" branding. The main heading is "mes achats de services internet plus".

**Détail de la transaction**

Date	Service	Description	Montant TTC	N° de transaction
02/07/08 03:57	verif	Fiche:W4-HA	3.00€ TTC	6-2172166009235153

**Je souhaite contester cette transaction pour la raison suivante**

- Connexion interrompue ou téléchargement incomplet
- Commande multiple
- Service/produit non délivré
- Achat contesté
- Transaction liées à un abonnement résilié ou non souhaité
- Article erroné
- Autre

Veuillez nous donner les raisons de votre contestation :

Buttons: [Retour](#) [Valider](#)

Footer: [Consulter l'assistance](#) | [Nous contacter](#)  
 @ copyright : Orange | assistance | publicité | à propos de Orange

**Remarque :** un **identifiant de transaction** Orange est de la forme **6-XXXXXXXXXXXXXXXXXX**

La demande est transmise au Service Clients de l'Opérateur qui décide d'accepter (ou non) le remboursement.

#### **Attention !**

**Si le remboursement est accepté** par l'opérateur, celui-ci peut déclencher également une **résiliation (immédiate) de l'Abonnement** à l'origine de la transaction.

## 2.3.2 - Résiliation d'un Abonnement

Un Abonnement peut être résilié de 2 manières :

### 2.3.2.1 - Résiliation d'un Abonnement par l'Internaute (online)

Désormais, l'éditeur peut re-diriger n'importe quel abonné qui souhaite résilier un abonnement, vers l'interface de gestion compte de son FAI.

**L'éditeur n'a pas besoin de connaître le FAI de l'internaute** (Orange, Neuf-Cegetel, ....).

Il suffit, sur son site Web, de faire un lien vers l'URL <https://route.w-ha.com/app-am/node>, en passant en paramètre l'identifiant de l'abonnement (uoid=X-UXXXXXXXXXXXXXXXXXX) en question.

Ex : <https://route.w-ha.com/app-am/node?uoid=8-U123451234512345>

Une fois authentifié (en mode nomade en choisissant son opérateur ou automatiquement via son adresse IP), le client est redirigé vers l'écran de détail de l'abonnement.

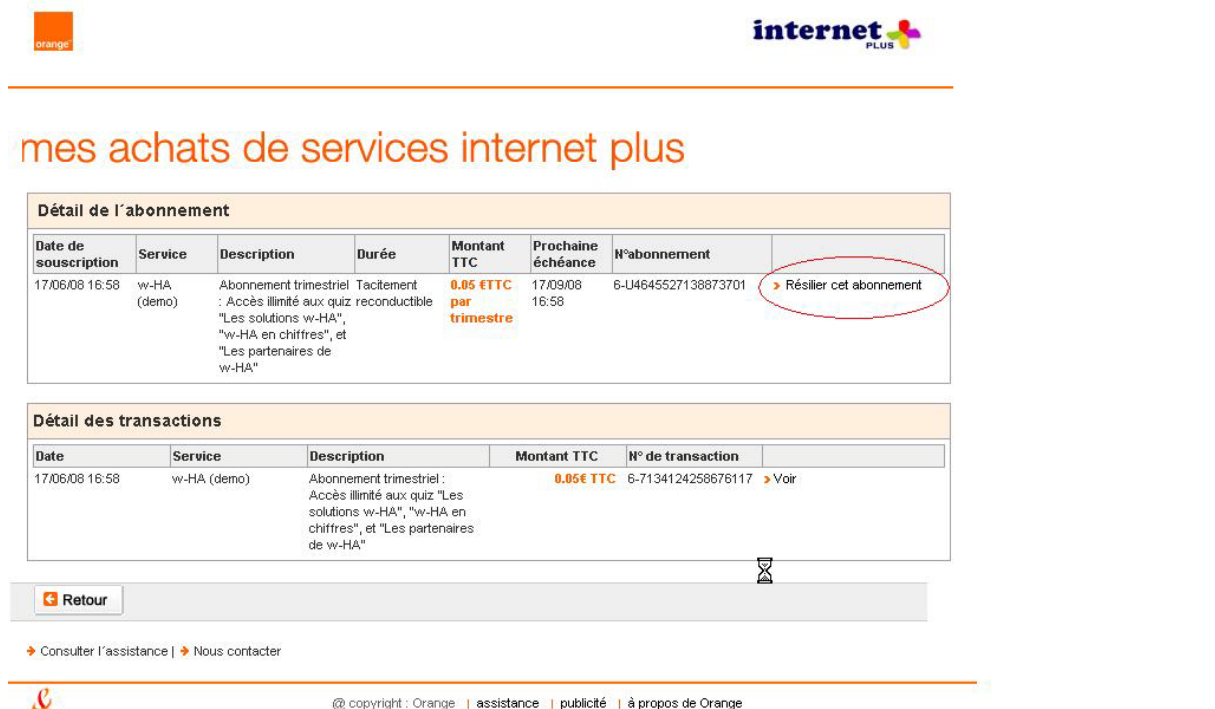
Différents cas peuvent se présenter :

- **pas de « uoid » transmis dans l'URL** : le client est redirigé vers la page d'accueil de l'application « Mon Compte » autrement dit vers la liste des achats de la période en cours.
- **l'« uoid » transmis ne correspondant à aucun abonnement valide** pour ce marchand : comme le cas précédent, le client est redirigé vers la liste des achats de la période en cours.
- **l'« uoid » transmis correspond à un abonnement valide** : le client est redirigé vers le détail de l'abonnement désiré.

Un Internaute Orange Internet (par exemple) peut demander, **en ligne**, la résiliation d'un Abonnement, depuis le portail [www.orange.fr](http://www.orange.fr) > espace client > conso internet (voir le détail) > Achats de service internet plus (voir le détail) > Consulter votre relevé détaillé (accès direct : <https://wanadoo.w-ha.com/app-am/node>)

Un clic sur l'onglet « **Mes abonnements** », aboutit à la page suivante :

 <https://whainetnet.orange.fr> - Orange - Mon Compte - Achat de Services Internet Plus - Mozilla Firefox



The screenshot shows the 'mes achats de services internet plus' page. It features the Orange logo and 'internet PLUS' branding. The main heading is 'mes achats de services internet plus'. Below this, there are two sections: 'Détail de l'abonnement' and 'Détail des transactions'.

**Détail de l'abonnement**

Date de souscription	Service	Description	Durée	Montant TTC	Prochaine échéance	N°abonnement	
17/06/08 16:58	w-HA (demo)	Abonnement trimestriel : Accès illimité aux quiz reconvertible "Les solutions w-HA", "w-HA en chiffres", et "Les partenaires de w-HA"	Tacitement	0.05 € TTC par trimestre	17/09/08 16:58	6-U4645527138873701	<a href="#">Résilier cet abonnement</a>

**Détail des transactions**

Date	Service	Description	Montant TTC	N° de transaction	
17/06/08 16:58	w-HA (demo)	Abonnement trimestriel : Accès illimité aux quiz "Les solutions w-HA", "w-HA en chiffres", et "Les partenaires de w-HA"	0.05 € TTC	6-7134124258676117	<a href="#">Voir</a>

At the bottom of the page, there is a 'Retour' button and links for 'Consulter l'assistance' and 'Nous contacter'. The footer contains the Orange logo and copyright information: '@ copyright : Orange | assistance | publicité | à propos de Orange'.



La demande de résiliation est automatiquement acceptée par la plate-forme w-HA.

**L'abonnement reste cependant actif jusqu'à la fin de sa période de validité (date anniversaire de la souscription) et les produits de l'abonnement doivent pouvoir être consommés jusqu'à cette date**

La demande de résiliation de l'internaute via cette interface va déclencher **2 notifications de résiliations** vers l'éditeur (servlet **/bundle-responder/responder**) (cf. §0) (pour un abonné Orange) :

- une notification (**code r=200**) est envoyée **immédiatement** à l'éditeur pour l'informer qu'une résiliation a été **demandée** (et acceptée)
- une notification (**code r=201**) est envoyée **à la fin de la période en cours** (date anniversaire) à l'éditeur pour l'informer que la résiliation est **effective** et qu'en conséquence, il ne faut plus donner accès au service

### **2.3.2.2 - Résiliation d'un Abonnement par le Service Clients Orange Internet**

Lorsqu'une demande de résiliation est effectuée auprès du Service Clients Orange Internet, l'opérateur a la possibilité de procéder :

- A - soit à une résiliation **immédiate** (liée à un remboursement, par exemple)
- B - soit à une résiliation **à la fin de la période en cours** (date anniversaire)

Dans les 2 cas, **une seule notification de résiliation** est envoyée vers l'éditeur, au moment où celle-ci est effective : donc immédiatement (cas A) ou à date anniversaire (cas B).

### **2.3.3 - Mode « push » : notifications de résiliation**

Lorsqu'une résiliation a lieu (cf. § 0) une requête de **notification de résiliation** est envoyée à l'éditeur sur la servlet **/bundle-responder/responder** (cf. 0)

L'éditeur peut alors **mettre à jour sa base de données, par exemple, en modifiant le statut de l'Abonnement** :

Du statut **Confirmed**, l'Abonnement passe, par exemple, aux statuts :

- Unsubscribed (pour une résiliation immédiate) + date fin (=date résiliation)
- To be unsubscribed (pour une résiliation à échéance) + date fin (=date anniversaire)

Par exemple :

O1 et O2 sont des Offres d'une durée d'un mois à Tacite Reconduction (TR)

L'abonnement 6-U7141248844587211 fait l'objet d'une résiliation immédiate le 28/09/2007 à 16h15

L'abonnement 6-U8241248844587211 fait l'objet d'une résiliation à échéance le 28/09/2007 à 16h15

UserId	UserOfferId (uoid)	oid	pid	Status	Date Souscription	Date (*) Reconduction	Date (**) Fin
fmoreau12	6-U7141248844587211	O1	P1	Unsubscribed	21/8/7 12:16	null	28/9/7 16:15
msantos012	6-U8241248844587211	O2	P2	To be Unsubscribed	22/8/7 15:12	22/9/7 15:12	22/10/7 15:12
fgalle45	none	O3	P1	Cancelled	null	null	null
gcronimus	6-U8223448846757123	O5	P2	Confirmed	23/8/7 23:34	23/11/7 23:34	null
fgalle45	none	O1	P1	Cancelled	null	null	null
fmoreau12	6-U0441248844587211	O3	P1	Confirmed	24/8/7 07:12	24/10/7 07:12	null



(\*) La colonne « Date Reconstitution » est utilisée pour indiquer, en fonction de la date du jour :

- soit la date du **dernier renouvellement** (ex : ligne 2) :
- soit la date du **prochain renouvellement** (ex : ligne 4 et 6)

(\*\*) La colonne « Date Fin » est utilisée pour indiquer, en fonction de la date du jour :

- soit la date à laquelle l'abonnement **a pris fin** (ex : ligne 1) :
- soit la date à laquelle l'abonnement **va prendre fin** (ex : ligne 2).

## 3 - Détail du fonctionnement des différentes fonctionnalités / servlets

### 3.1 - Servlets avec redirection de l'internaute vers la plate-forme w-HA

La fonctionnalité « souscription » (cf. 0) entraîne la re-direction (via son navigateur web) de l'internaute vers la plate-forme w-HA.

#### **Attention !**

La servlet « **pos\_bundle** » doit être **accessible depuis l'internet public (port 80)**.

#### **3.1.1 - Demande Souscription / « pos\_bundle?action=authorizeOffer »**

##### ***3.1.1.1 - Appel de la servlet***

Lorsque l'éditeur souhaite déclencher une **souscription** d'un utilisateur, il fait appel à la **servlet** « **pos\_bundle** », avec le paramètre « **action** » égal à « **authorizeOffer** ».

Cette servlet est appelée au niveau de la page Web de présentation des offres de l'éditeur :

##### ***3.1.1.2 - Exemple d'appel de la servlet***

```
http://192.168.1.8/bundle/pos\_bundle?  
action=authorizeOffer  
&oid=O1  
&mid=13  
&url=https://route.w-ha.com/app-bundlepurchase/node  
&sessionId=1234  
&userId=abcd
```

En rouge : paramètres obligatoires

En bleu : paramètres surchargeables

En vert : paramètres optionnels (merchant properties)



### 3.1.1.3 - Paramètres d'appel de la servlet

Pour le déclenchement d'une souscription, la servlet « pos\_bundle » est appelée avec les paramètres suivants :

Paramètres toujours obligatoires :

**action=authorizeOffer** : pour indiquer à la servlet qu'il s'agit d'une demande de souscription

**oid=Identifiant de l'offre** : identifiant de l'offre à laquelle l'utilisateur est invité à souscrire

Paramètres surchargeables :

**mid=Identifiant de boutique** : dans le cas où la boutique à utiliser n'est pas celle indiquée dans le fichier « web.xml », l'éditeur peut indiquer quel merchantId il souhaite utiliser à la place.

**url=Url du nœud w-HA** : ce paramètre est utilisé :

- soit lorsque l'éditeur utilise la boutique définie par défaut dans le fichier « web.xml » mais qu'il ne veut pas utiliser le nœud par défaut.
- soit de manière obligatoire, si l'éditeur a surchargé le paramètre mid (cf. ci-dessus). L'url du nœud à utiliser par défaut n'étant pas précisé dans le fichier « merchants.xml », il faut le préciser ici.

#### **Attention !**

Dans le cadre de l'offre **Internet+**, l'URL à paramétrer est :  
**<https://route.w-ha.com/app-bundlepurchase/node>**

Paramètres optionnels :

**paramètres supplémentaires de l'éditeur (ou « merchant properties » (mp)) :**

Ces paramètres ne sont pas obligatoires en théorie, mais dans la pratique, l'éditeur utilise au moins un paramètre du type « Identifiant Utilisateur », lui permettant d'associer un identifiant de client chez lui (ex : userId) avec un identifiant d'abonnement (uiod) souscrit auprès de w-HA.

Les « **merchant properties** » **seront récupérés au niveau de l'Url de prise en compte de la souscription (fulfillmentUrl)** qui doit être un exécutable (servlet, cgi, asp, php, .....)

### 3.1.1.4 - Fonctionnement interne de la servlet

Lorsqu'elle est appelée avec le paramètre action=authorizeOffer, la servlet « pos\_bundle » réalise les actions suivantes :

- récupère dans le fichier « web.xml » les informations relatives à la boutique (merchantId, URLs, .....)
- récupère dans le fichier « productbundle\_XXX.xml » les informations relatives à l'Offre qui doit être souscrite par l'utilisateur (OfferId ; ....)
- récupère, les paramètres de boutique (mctKeyId, mctKey, ...) dans le fichier « merchants.xml »,
- crée une URL sécurisée (https) et signée, appelant la plate-forme w-HA et contenant ces informations, ainsi que les paramètres (merchant properties) ajoutés par le marchand
- redirige l'internaute vers cette URL.
- met à jour le fichier de log « log\_request\_XXX.txt »



### 3.1.1.5 - Exemple d'URL générée par la servlet

L'URL vers laquelle l'internaute est redirigé est de la forme (URL encodée) :

```
https://route.w-ha.com/app-  
bundlepurchase/node?m=h%3D17c5d513e865a2995141ef60ef503ddd%3Bp%3D515%3Bk%3D515%  
3Bv%3D3%3A%7Bc%3DOfferAuthorizeReq%3Bv%3D%7BmUrl%3Dhttp%3A%2F%192.168.1.8%2F  
bundle%2Fpos_bundle%3Bpromo%3Dpromo%3Boid%3DO1%3Bmp%3D%7B_ap_userId%3Dabcd%  
3B_ap_sessionId%3D1234%3Bts%3D2008-04-  
14+14%3A38%3A15.25%3Bcur%3DEUR%3B%7D%3B%7D%7D
```

Pour plus de lisibilité, en URL décodée :

```
https://route.w-ha.com/app-bundlepurchase/node?  
m=h=17c5d513e865a2995141ef60ef503ddd;  
p=515;  
k=515;  
v={  
c=OfferAuthorizeReq;  
v={  
mUrl=http://192.168.1.8/bundle/pos_bundle;  
promo=promo;  
oid=O1;  
mp={  
_ap_userId=abcd;  
_ap_sessionId=1234;  
ts=2008-04-14 14:38:15.25;  
cur=EUR;}}}
```

Cette URL est sécurisée (https) et signée.

### 3.1.1.6 - Réponse de w-HA à la servlet

Lorsque l'utilisateur est redirigé vers la plate-forme w-HA pour une demande de souscription (cf. paragraphe précédent), celle-ci lui affiche un panneau de souscription :

orange

internet PLUS

## Votre commande

Produit : **Abonnement hebdomadaire : Accès illimité au Produit P1**

Service : FranceFootball.fr

Prix : **0.01 € TTC chaque semaine**

Edité par : **FRANCE FOOTBALL**

[afficher les conditions spécifiques de cette offre](#)

**Confirmer votre achat**

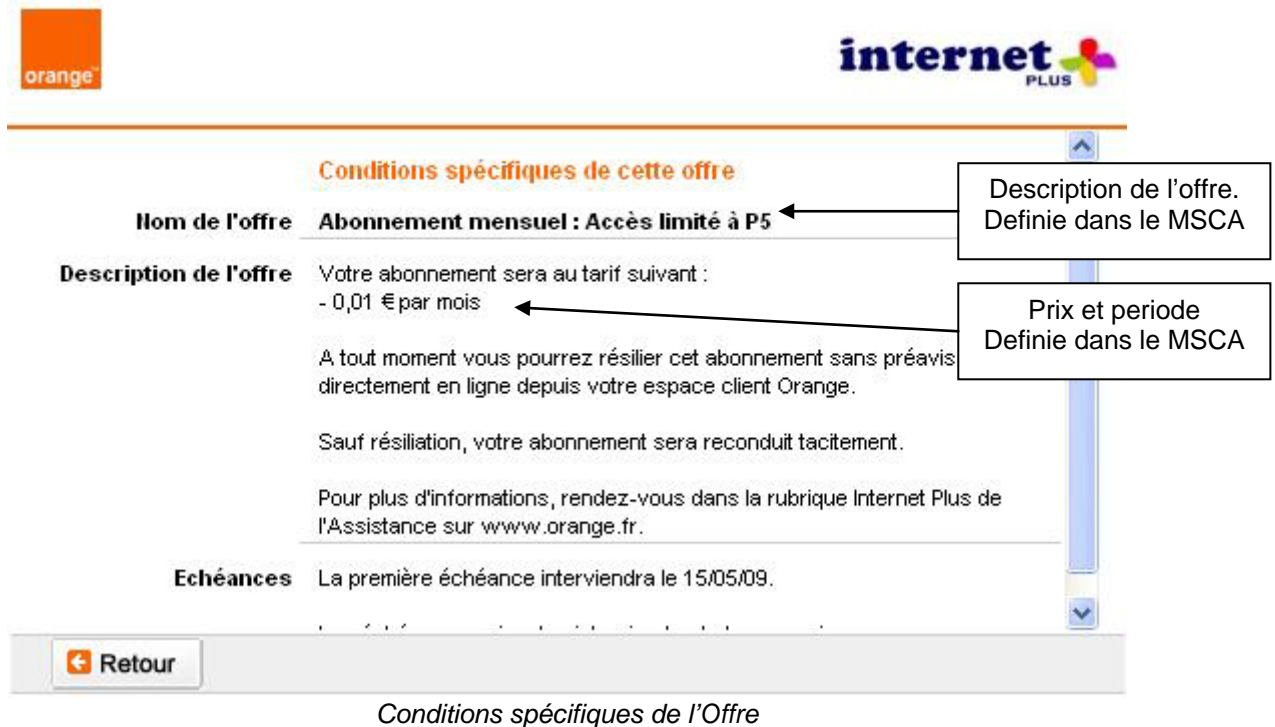
Le montant de votre achat sera ajouté à votre prochaine facture Internet Orange \*.  
Vous ne souhaitez pas souscrire ce service : **annulez la commande**

\* service acheté aux [Conditions de paiement Internet Orange](#) que vous acceptez en confirmant votre achat.

© Copyright France Télécom

Technologie w-HA

Panneau de souscription



**Conditions spécifiques de cette offre**

**Nom de l'offre** Abonnement mensuel : Accès limité à P5

**Description de l'offre** Votre abonnement sera au tarif suivant :  
- 0,01 € par mois

A tout moment vous pourrez résilier cet abonnement sans préavis directement en ligne depuis votre espace client Orange.

Sauf résiliation, votre abonnement sera reconduit tacitement.

Pour plus d'informations, rendez-vous dans la rubrique Internet Plus de l'Assistance sur [www.orange.fr](http://www.orange.fr).

**Echéances** La première échéance interviendra le 15/05/09.

[Retour](#)

*Conditions spécifiques de l'Offre*

L'utilisateur peut soit accepter la souscription, soit la refuser.

### **1er cas : refus de la souscription par l'utilisateur :**

Si l'utilisateur clique sur le lien « annulez la commande », la souscription n'est pas validée, au niveau de la plate-forme w-HA : celle-ci ne **génère pas d'identifiant de souscription**.

L'utilisateur est redirigé vers la servlet « pos\_bundle » de l'éditeur (*merchantCallbackUrl*), avec un message de type « **OfferAuthorizationCancel** »

Exemple d'URL (**pos\_bundle**) sur laquelle l'utilisateur est redirigé, en cas d'annulation sur le panneau de paiement :

```
http://192.168.1.8/bundle/pos_bundle?m=h%3D032e20600035df64a059644b9c12593a%3Bp%3D515%3Bk%3D515%3Bv%3D3%3A%7Bc%3DOfferAuthorizationCancel%3Bv%3D%7B_ap_userId%3Dabcd%3B_ap_sessionId%3D1234%3Bts%3D2008-04-14+14%3A46%3A44.343%3Bcur%3DEUR%3B%7D%7D
```

Pour plus de lisibilité, en URL décodée :

```
http://192.168.1.8/bundle/pos_bundle?
m=h=032e20600035df64a059644b9c12593a;
p=515;
k=515;
v=3:{
c=OfferAuthorizationCancel;
v={
_ap_userId=abcd;
_ap_sessionId=1234;
ts=2008-04-14 14:46:44.343;
cur=EUR;}}
```

Cette URL est sécurisée (https) ou non (http), et signée.



Les informations contenues dans cette URL sont :

- un hmac (h=), pour assurer l'intégrité du message,
- le message « OfferAuthorizeCancel »
- les paramètres supplémentaires de l'éditeur (avec le préfixe \_ap\_)

La servlet « pos\_bundle » met alors à jour le fichier de log « log\_response\_XXX.txt » et re-dirige l'utilisateur vers la **mcttrxCancelFromPaymentPanelUrl**, paramétrée au niveau du fichier « merchants.xml » :

Exemple d'URL (**mcttrxCancelFromPaymentPanelUrl**) sur laquelle l'utilisateur est redirigé, en cas d'annulation sur le panneau de paiement :

[http://192.168.1.4/demo/bundle/html/panel\\_cancel1.html?hmac=514f1bec55a3e95639a41a9e46af6b8b&amt=0.01&cur=EUR&sessionId=1234&ts=2008-04-14+14%3A46%3A44.343&userId=abcd](http://192.168.1.4/demo/bundle/html/panel_cancel1.html?hmac=514f1bec55a3e95639a41a9e46af6b8b&amt=0.01&cur=EUR&sessionId=1234&ts=2008-04-14+14%3A46%3A44.343&userId=abcd)

Pour plus de lisibilité, en URL décodée :

[http://192.168.1.4/demo/bundle/html/panel\\_cancel1.html?](http://192.168.1.4/demo/bundle/html/panel_cancel1.html?hmac=514f1bec55a3e95639a41a9e46af6b8b&cur=EUR&sessionId=1234&ts=2008-04-14+14:46:44.343&userId=abcd)  
hmac=514f1bec55a3e95639a41a9e46af6b8b  
&cur=EUR  
&sessionId=1234  
&ts=2008-04-14 14:46:44.343  
&userId=abcd

Cette URL est sécurisée (https) ou non (http), et signée.

Les informations contenues dans cette URL sont :

- un hmac, pour assurer l'intégrité du message,
- les merchant properties (amt + cur + ts + paramètres propres à l'éditeur)

### **2ème cas : refus de la souscription par l'opérateur (via la plate-forme w-HA) :**

Si l'utilisateur clique sur le bouton « Confirmer votre achat », mais que la souscription ne peut pas être acceptée (plafond mensuel atteint, paiement sur facture opérateur refusé, ...) par la plate-forme w-HA, celle-ci ne **génère pas d'identifiant de souscription**.

Un message d'erreur est affiché à l'utilisateur, directement sur le panneau de paiement.

Il n'y a pas de retour vers la servlet « pos\_bundle », pour préciser la cause de l'erreur.

En conséquence, aucune ligne n'est ajoutée au niveau du fichier « log\_response\_XXX.txt »

### **3ème cas : acceptation de la souscription par l'utilisateur :**

Si l'utilisateur clique sur le bouton « Confirmer votre achat », et que la souscription est acceptée par la plate-forme w-HA, celle-ci **génère un identifiant de souscription, au statut autorisé**.

L'utilisateur est redirigé vers la servlet « pos\_bundle » de l'éditeur (**merchantCallbackUrl**), avec un message de type « OfferAuthorizeSuccess ».

Exemple d'URL sur laquelle l'utilisateur est redirigé, en cas de confirmation d'achat sur le panneau de paiement :

[http://192.168.1.8/bundle/pos\\_bundle?m=h%3D66d1ce95dbd08060aa42c0f66696ca09%3Bp%3D515%3Bk%3D515%3Bv%3D3%3A%7Bc%3DOfferAuthorizationSuccess%3Bv%3D%7Bmp%3D%7B\\_ap\\_userId%3Dabcd%3B\\_ap\\_sessionId%3D1234%3Bts%3D2008-04-14+15%3A45%3A59.515%3Bcur%3DEUR%3B%7D%3Boid%3DO4%3Bru%3Dhttps%3A%2F%2Fwaa.nadoo.w-ha.com%2Fapp-node-mct%2Fresponder%3Bg\\_amt%3D0.01%3Bz%3D92442%3Bco%3DFR%3Buoid%3D6-U5117575881274524%3Bst%3DFR%3Bci%3DISSY+LES+MOULINEAUX+CEDEX%3B%7D%7D](http://192.168.1.8/bundle/pos_bundle?m=h%3D66d1ce95dbd08060aa42c0f66696ca09%3Bp%3D515%3Bk%3D515%3Bv%3D3%3A%7Bc%3DOfferAuthorizationSuccess%3Bv%3D%7Bmp%3D%7B_ap_userId%3Dabcd%3B_ap_sessionId%3D1234%3Bts%3D2008-04-14+15%3A45%3A59.515%3Bcur%3DEUR%3B%7D%3Boid%3DO4%3Bru%3Dhttps%3A%2F%2Fwaa.nadoo.w-ha.com%2Fapp-node-mct%2Fresponder%3Bg_amt%3D0.01%3Bz%3D92442%3Bco%3DFR%3Buoid%3D6-U5117575881274524%3Bst%3DFR%3Bci%3DISSY+LES+MOULINEAUX+CEDEX%3B%7D%7D)



Pour plus de lisibilité, en URL décodée :

```
http://192.168.1.8/bundle/pos_bundle?  
m=h=66d1ce95dbd08060aa42c0f66696ca09;  
p=515;  
k=515;  
v=3:{  
c=OfferAuthorizationSuccess;  
v={  
mp={  
_ap_userId=abcd;  
_ap_sessionId=1234;  
ts=2008-04-14 15:45:59.515;  
cur=EUR;};  
oid=O4;  
ru=https://wanadoo.w-ha.com/app-node-mct/responder;  
g_amt=0.01;  
z=92442;  
co=FR;  
uoid=6-U5117575881274524;  
st=FR;  
ci=ISSY LES MOULINEAUX CEDEX;}}
```

Cette URL est sécurisée (https) ou non (http), et signée.

Les informations importantes contenues dans cette URL sont :

- un hmac, pour assurer l'intégrité du message (h=)
- le message « OfferAuthorizeSuccess » (c=)
- le montant de la souscription (g\_amt=)
- les paramètres supplémentaires de l'éditeur (mp=)
- l'URL pour confirmer la souscription (ru=)
- l'identifiant de l'abonnement souscrit (uoid=)

A la réception de cette requête, la servlet « pos\_bundle » met alors à jour le fichier de log « log\_response\_XXX.txt »

Deux cas peuvent alors se présenter, selon la valeur du paramètre « autoConfirm » du fichier « productbundle\_XXX.xml », pour l'Offre en question :

#### **\* autoConfirm = true**

Si le paramètre « autoConfirm » de l'Offre est à la valeur 'true', dès réception du message de type « OfferAuthorizeSuccess », la servlet « pos\_bundle » déclenche automatiquement et immédiatement la **confirmation de souscription** (cf. § 0).

#### **Côté w-HA, la souscription passe dans le statut « confirmed ».**

Côté éditeur, la servlet « pos\_bundle » re-dirige l'utilisateur vers la **fulfillmentUrl** de l'offre, paramétrée dans le fichier « productbundle\_XXX.xml ».

Au niveau de la fulfillmentUrl, l'éditeur doit mettre en place un programme exécutable (CGI, php, jsp, coldfusion, ...) chargé de **prendre en compte la souscription dans sa base de données**.

#### **\* autoConfirm = false**

Si le paramètre « autoConfirm » de l'offre est à la valeur 'false', dès réception du message de type « OfferAuthorizeSuccess », la servlet « pos\_bundle » re-dirige l'utilisateur vers la **fulfillmentUrl** de l'offre, paramétrée dans le fichier « productbundle\_XXX.xml ».



Au niveau de la `fulfillmentUrl`, l'éditeur doit mettre en place un programme exécutable (CGI, php, jsp, coldfusion, ...) chargé de **prendre en compte la souscription**.

L'éditeur dispose de **24h maximum** (à partir de la date d'autorisation de l'offre) **pour confirmer la souscription** (la rendre effective, côté plate-forme w-HA), au moyen du SDK fourni par w-HA.

#### **Attention !!**

Si l'éditeur n'effectue pas la confirmation de la souscription, celle-ci sera valide côté éditeur, mais pas côté w-HA, entraînant la **désynchronisation des bases de données**.

### **3.1.1.7 - Prise en compte de la souscription (`fulfillmentUrl` de l'Offre)**

Lorsque la servlet « `pos_bundle` » reçoit une réponse positive de demande de souscription de la part de la plate-forme w-HA, elle redirige l'utilisateur vers l'URL de prise en compte de la souscription (« `fulfillmentUrl` » de l'Offre) paramétrée dans le fichier « `productbundle_XXX.xml` ».

Exemple d'URL de prise en compte de souscription, générée par la servlet « `pos_bundle` » :

```
http://192.168.1.8/demo/bundle/html/validOffre4.html?hmac=7633f0af106fbef2b0d644c0c9aefcd0&cur=EUR&oid=O4&sessionId=1234&ts=2008-04-14+15%3A54%3A26.093&uoid=6-U4645012218882221&userId=abcd
```

Pour plus de lisibilité, en URL décodée :

```
http://192.168.1.8/demo/bundle/html/validOffre4.html?
hmac=7633f0af106fbef2b0d644c0c9aefcd0
&cur=EUR
&oid=O4
&sessionId=1234
&ts=2008-04-14 15:54:26.093
&uoid=6-U4645012218882221
&userId=abcd
```

Cette URL est sécurisée (https) ou non et signée.

Les informations importantes contenues dans cette URL sont :

- un `hmac`, pour assurer l'intégrité du message (`h=`),
- l'identifiant de l'abonnement (`uoid=`)
- l'identifiant de l'offre (`oid=`)
- les paramètres supplémentaires de l'éditeur

#### **Attention !**

Pour le calcul du `hmac`, l'ensemble des paramètres doit être pris en compte, y compris l'identifiant de l'abonnement (`uoid`).

Au niveau de l'URL de prise en compte de l'abonnement, **l'éditeur doit associer l'identifiant de l'abonnement qui vient d'être souscrit (`uoid`) avec l'identifiant de l'utilisateur (ex : `userId`) dans sa propre application (back-office)**.

`UserId=fmoreau12` ←————→ `uoid=6-U7141248844587211`

Il peut alors mettre à jour sa base de données avec l'identifiant (w-HA) de l'abonnement souscrit, en précisant le statut de l'abonnement. (cf. § [2.1.2 - Etape #2 : 1ère Mise à Jour de la base de données EdS](#))





### 3.1.2 - Rechargement des fichiers de configuration locaux / « pos\_bundle?action=reload »

Lorsque l'éditeur modifie ces fichiers de configurations xml, il doit en principe re-démarrer l'application « bundle » pour prendre en compte les modifications, ce qui rend l'application indisponible le temps de l'arrêt-recharge.

L'appel de la servlet « **pos\_bundle** » avec le paramètre « **action=reload** » permet de recharger les différentes configurations stockées dans les fichiers XML de l'application.

Ainsi, un appel à cette action, entraîne une prise en compte de toutes les modifications effectuées dans ces fichiers depuis le dernier rechargement.

Les fichiers concernés sont :

- Le fichier « merchants.xml »
- Le(s) fichier(s) « productbundle\_XXX.xml »

## 3.2 - Servlets pour les requêtes de serveur à serveur

Les fonctionnalités suivantes sont gérées via des requêtes **httpS** de serveur à serveur ;;

- Confirmation différée (servlet « pos\_request » : EdS -> w-HA)
- Résiliation d'un Abonnement (servlet « pos\_request » : EdS -> w-HA)
- Remboursement d'une transaction (servlet « pos\_request » : EdS -> w-HA)
- notifications de résiliation « push » (servlet « responder » : w-HA → EdS)

### Attention !

Les servlets « responder » et « pos\_request » gérant ces fonctionnalités **NE doivent PAS** être accessibles depuis l'internet public (port 80).

### Attention !

S'agissant de requêtes **httpS**, les éventuels **équipements réseau** doivent être correctement configurés pour **laisser passer les requêtes** en direction et en provenance **de la plate-forme w-HA**.



### **3.2.1 - Confirmation « différée » / pos\_request?action=offerConfirm**

Contrairement à la demande de souscription et à la réponse à la demande de souscription, qui sont des re-directions http(s), **la confirmation de souscription est une requête https, de serveur à serveur.**

Elle permet de valider la souscription au niveau de la plate-forme w-HA : la souscription en question passe alors dans le statut « confirmed ».

Lorsque le paramètre « **autoConfirm** » est à la valeur 'true', c'est la servlet « pos\_bundle » qui génère **automatiquement et immédiatement une requête de « confirm »** vers la plate-forme w-HA.

Lorsque le paramètre « **autoconfirm** » relatif à une offre est égal à 'false' dans le fichier « productbundle\_XXX.xml », **l'éditeur de service doit « confirmer » la souscription à w-HA de manière différée.**

#### **3.2.1.1 - Appel de la servlet**

Lorsque l'internaute souscrit à une offre, celui-ci est redirigé vers l'URL de validation de la souscription définie dans le fichier « productbundle\_XXX.xml » (fulfillmentUrl).

L'arrivée de l'internaute sur cette URL entraîne le déclenchement d'un programme (à développer par l'éditeur en php, java, coldfusion, ...) qui va :

- 1°) **recupérer** l'identifiant de l'abonnement souscrit (**uuid**),
- 2°) **mettre à jour la base de données des abonnements de l'éditeur**
- 3°) appeler la servlet « **pos\_request ?action=offerConfirm** pour valider la souscription auprès de w-HA

#### **3.2.1.2 - Exemple d'appel de la servlet**

Après avoir mis à jour sa base de données des abonnements, l'éditeur va appeler la servlet, **via son client HTTP (à développer)**, en lui passant les paramètres nécessaires.

Pour ce faire, il utilisera une URL de la forme :

```
http://wha.marchand.com/bundle/pos\_request?  
action=offerConfirm  
&mid=10  
&uuid=6-U7141248844587211  
&url=https://wanadoo.w-ha.com/app-node-mct/responder
```

En rouge : paramètres obligatoires  
En bleu : paramètres surchargeables



### 3.2.1.3 - Paramètres d'appel de la servlet

Pour la confirmation différée de souscription, la servlet « pos\_request » est appelée avec les paramètres suivants :

**action=offerConfirm** : pour indiquer à la servlet qu'il s'agit d'une requête de confirmation de souscription

**mid=Identifiant de boutique** : identifiant de la boutique concernée par la requête de « confirmation » ; les valeurs de keyId et keyValue correspondantes sont récupérées par la servlet dans le fichier « merchants.xml ».

**uoid=Identifiant de l'abonnement** : identifiant de l'Abonnement souscrit par l'internaute et qui doit faire l'objet de la confirmation (=débit effectif).

**url=Url du nœud w-HA (responder)** : url du nœud w-HA qui va réceptionner la requête de confirmation de souscription.

Dans le cadre de l'offre Internet+, l'URL du nœud à appeler dépend de l'identifiant de l'Abonnement/Offre.

Les correspondances identifiant/opérateur/url, sont données dans le tableau ci-dessous :

Identifiant	Opérateur	URL (responder)
6-XXX...XXX	Orange Internet	<a href="https://wanadoo.w-ha.com/app-node-mct/responder">https://wanadoo.w-ha.com/app-node-mct/responder</a> (seul opérateur disponible pour la phase pilote)
7- XXX...XXX	Alice/Free	<a href="https://free.w-ha.com/app-node-mct/responder">https://free.w-ha.com/app-node-mct/responder</a>

### 3.2.1.4 - Fonctionnement interne de la servlet

Lorsqu'elle est appelée avec le paramètre action=offerConfirm, la servlet « pos\_request » réalise les actions suivantes :

- récupère dans le fichier « web.xml » les informations relatives à la boutique (merchantId, URLs, ....)
- récupère les paramètres de la boutique (merchantId, keyId, keyValue, ...) dans le fichier « merchants.xml »
- crée une URL sécurisée (https) et signée, vers la plate-forme w-HA et contenant ces informations, ainsi que l'identifiant de l'Abonnement (uoid) à confirmer
- déclenche une requête https (en background) vers la plate-forme w-HA.

### 3.2.1.5 - Exemple d'URL générée par la servlet

Exemple d'URL de confirmation de souscription, générée par la servlet « pos\_request » :

```
https://wanadoo.w-ha.com/app-node-mct/responder?m=h%3D8e94261071cba9dcaf53b81759f5ab2a%3Bp%3D515%3Bk%3D515%3Bv%3D3%3A%7Bc%3Dm_offerConfirm%3Bv%3D%7Buoid%3D6-U2143613233868231%3B%7D%7D
```



Pour plus de lisibilité, en URL décodée :

```
https://wanadoo.w-ha.com/app-node-mct/responder?  
m=h=8e94261071cba9dcaf53b81759f5ab2a;  
p=515;  
k=515;  
v=3:{  
c=m_offerConfirm;  
v={  
uoid=6-U2143613233868231;}}
```

Cette URL est sécurisée (https) et signée.

Les informations importantes contenues dans cette URL sont :

- un hmac, pour assurer l'intégrité du message (h=),
- la commande « m\_offerConfirm » (c=),
- l'identifiant de l'abonnement à confirmer (uoid=)

A réception de cette confirmation de souscription, la plate-forme w-HA renvoie un « acquittement » (ack) à la servlet « pos\_request ».

### 3.2.1.6 - Réponse de la plate-forme w-HA à une requête de confirmation différée

Lorsque la confirmation de la souscription à l'offre (=> abonnement) (uoid) est validée, la plate-forme w-HA retourne **c=ack**

Si une erreur se produit, la plate-forme retourne un code d'erreur spécifique (**e=code**), selon la nomenclature suivante :

Code	Libellé
0	Abonnement (uoid) inconnu : « User offer not found »
1	Souscription à l'abonnement (uoid) déjà confirmée : « Invalid user offer status ». Code retour générique lorsqu'un abonnement (uoid) n'est plus dans un statut actif. (ex : le confirm est envoyé au delà des 24h00 imparties après la souscription)

Exemple : e=1

### 3.2.1.7 - Exemple d'utilisation de la servlet : formulaire web

w-HA met également à disposition de l'éditeur une interface Web (formulaire) utilisant la servlet « pos\_request?action=offerConfirm ».

Elle permet à l'éditeur :

- de bien assimiler le fonctionnement de la servlet
- de réaliser des confirmations manuellement, si nécessaire.

Cette interface Web se trouve à l'URL suivante :

[http://wha.marchand.com/demo/bundle/html/confirm\\_differe.html](http://wha.marchand.com/demo/bundle/html/confirm_differe.html) <sup>(\*)</sup>;

(\*) remplacer *wha.marchand.com* par l'adresse IP ou le nom de domaine public du serveur sur lequel est installée l'application w-HA

**Requête "confirm" différé, de serveur à serveur (EdS --> w-HA)**

merchantId (mid)

identifiant abonnement (uoid)

url requête (url)



### **3.2.2 - Résiliation d'un Abonnement / « pos\_request ?action=cancelOffer**

L'éditeur a la possibilité de procéder à la résiliation d'un abonnement, en effectuant une requête (serveur à serveur) vers la plate-forme w-HA.

La résiliation peut être une :

- résiliation **immédiate (deferred=0=false)**
- résiliation à **échéance (deferred=1=true)**

Note :

- Lors d'une résiliation à échéance, l'abonnement s'arrêtera à la fin de la période d'engagement.
- Lors d'une résiliation immédiate, l'abonnement s'arrête immédiatement même s'il y a une période d'engagement.

#### **3.2.2.1 - Appel de la servlet**

Lorsqu'un internaute souhaite résilier un Abonnement, il peut :

- soit contacter le Service Clients d'Orange (Internet)
- soit contacter directement le Service Clients de l'éditeur

Après étude de la demande, l'éditeur peut alors déclencher une requête de résiliation d'abonnement vers la plate-forme w-HA en précisant l'**identifiant de l'Abonnement** (uoid) à résilier, le **moment** où la résiliation doit être effective (**immédiatement ou à échéance**), ainsi que des **commentaires**.

#### **3.2.2.2 - Exemple d'appel de la servlet**

Afin d'effectuer la résiliation d'abonnement, l'éditeur va appeler la servlet, **via son client HTTP** (à développer), en lui passant les paramètres nécessaires.

Pour ce faire, il utilisera une URL de la forme :

```
http://wha.marchand.com/bundle/pos_request?  
action=cancelOffer  
&mid=10  
&uoid=6-U1741248844587211  
&url=https://wanadoo.w-ha.com/app-node-mct/responder  
&deferred=0  
&userComment=commentaire client  
&adminComment=commentaire editeur  
&reasonCode=110
```

#### **3.2.2.3 - Paramètres d'appel de la servlet**

Pour la résiliation d'un abonnement, la servlet « pos\_request » est appelée avec les paramètres suivants :

**action=cancelOffer** : pour indiquer à la servlet qu'il s'agit d'une requête de résiliation d'un Abonnement

**mid=Identifiant de boutique** : identifiant de la boutique concernée par la requête de « confirmation » ; les valeurs de keyId et keyValue correspondantes sont récupérées par la servlet dans le fichier « merchants.xml ».

**uoid=Identifiant de l'abonnement** : identifiant de l'Abonnement à résilier.

**url=Url du nœud w-HA (responder)** : url du nœud w-HA qui va réceptionner la requête de résiliation d'Abonnement.

(ex : L'url du nœud Orange (Internet) est : https://wanadoo.w-ha.com/app-node-mct/responder)



**deferred=0/1** ou **deferred =false/true** : indique si la résiliation doit être effectuée immédiatement (deferred=0=false) ou à échéance (deferred=1=true)

**userComment=Commentaire du client** : commentaire du client sur la (les) raison(s) de sa demande de résiliation d'abonnement à l'éditeur (les caractères spéciaux doivent être encodés)

**adminComment=Commentaire de l'éditeur** : commentaire de l'éditeur sur la (les) raison(s) de sa demande de résiliation d'abonnement à w-HA (les caractères spéciaux doivent être encodés)

**reasonCode=Code du motif de la résiliation** :

Code (r=)	Commentaire
110	Résiliation suite à demande internaute (autre)
111	Résiliation suite à demande internaute (commande multiple)
112	Résiliation suite à demande internaute (ticket facturé plusieurs fois)
113	Résiliation suite à demande internaute (service non délivré)
114	Résiliation suite à demande internaute (achat contesté)
115	Résiliation suite à demande internaute (achat enfants)
116	Résiliation suite à demande internaute (transaction liée à abonnement résilié ou non souhaité)
117	L'abonnement est résilié : je souhaite le remboursement de la transaction associée
118	Le service délivré ne correspond pas à mes attentes
119	J'ai souscrit par erreur cet abonnement plusieurs fois

### 3.2.2.4 - Fonctionnement interne de la servlet

Lorsqu'elle est appelée avec le paramètre `action=cancelOffer`, la servlet « `pos_request` » réalise les actions suivantes :

- récupère dans le fichier « `web.xml` » les informations relatives à la boutique (merchantId, URLs)
- récupère les paramètres de la boutique (merchantId, keyId, keyValue) dans le fichier « `merchants.xml` »,
- crée une URL sécurisée (https) et signée, vers la plate-forme w-HA et contenant ces informations, ainsi que l'identifiant de l'Abonnement (uoid) pour lequel la demande de résiliation est effectuée.
- déclenche une requête https (en background) vers la plate-forme w-HA.

### 3.2.2.5 - Exemple d'URL générée par la servlet

Exemple d'URL de résiliation générée par la servlet « `pos_request` » :

```
https%3A/wanadoo.w-ha.com/app-node-mct/responder%3Fm%3Dh%3Dfd4685944ae52d91601e343f7f561d3c%3Bp%3D13%3Bk%3D13%3Bv%3D3%3A%7Bc%3Dm_userOfferUnsubscribe%3Bv%3D%7Badmincom%3DCommentaire%20editeur%3Bdc%3D0uoid%3D6-U714124844587211%3Breason%3D110%3Busercom%3DCommentaire%20client%3B%7D%7D
```

Pour plus de lisibilité, en URL décodée :

```
https://wanadoo.w-ha.com/app-node-mct/responder?
m=h=fd4685944ae52d91601e343f7f561d3c;
p=13;
k=13;
v=3:{
c=m_userOfferUnsubscribe;
v={
admincom=Commentaire editeur;
dc=0
uoid=6-U714124844587211;
reason=110;
usercom=Commentaire client;}}
```



Cette URL est sécurisée (https) et signée.

Les informations importantes contenues dans cette URL sont :

- un hmac, pour assurer l'intégrité du message (h=),
- la commande « m\_userOfferUnsubscribe » (c=),
- l'identifiant de l'Abonnement à résilier (uoid=)

### 3.2.2.6 - Réponse de w-HA à la servlet

Lorsque l'identifiant de l'Abonnement à résilier existe et l'Abonnement est dans un état permettant sa résiliation :

- la plate-forme w-HA retourne au kit :

**h=46e73ffadc98fa7f29d5fcb119b198f9;p=5XXX;k=5XXX;v=2:{c=ack}**

- le Kit renvoie : **e=0**

Si une erreur se produit, la plate-forme retourne un code d'erreur spécifique (**e=code**), selon la nomenclature suivante :

Code	Libellé
1	Echec de la résiliation
3	Mauvais hmac

Exemple : **e=1**

Remarque : Si on essaye de **résilier plusieurs fois le même abonnement**, la plate-forme retournera **toujours** un même message d'acquiescement :

**h=46e73ffadc98fa7f29d5fcb119b198f9;p=5XXX;k=5XXX;v=2:{c=ack}**

(Le kit renverra e=1)

### 3.2.2.7 - Exemple d'utilisation de la servlet : formulaire web

w-HA met également à disposition de l'éditeur une interface Web utilisant la servlet « pos\_request » (action=cancelOffer).

Elle permet à l'éditeur :

- de bien assimiler le fonctionnement de la servlet
- de réaliser des résiliations d'abonnement manuellement, si nécessaire

Cette interface Web se trouve à l'URL suivante :

[http://wha.marchand.com/demo/bundle/html/resiliation\\_abonnement.html](http://wha.marchand.com/demo/bundle/html/resiliation_abonnement.html) (\*):

(\*): remplacer *wha.marhand.com* par l'adresse IP ou le nom de domaine public du serveur sur lequel est installée l'application w-HA

**L'appel de la servlet "/pos\_request?action=cancelOffer" peut être simulée - via le formulaire ci-dessous :**

merchantId (mid)	<input type="text" value="12"/>
identifiant abonnement (uoid)	<input type="text" value="6-4111235544454210"/>
url du noeud (url)	<input type="text" value="https://wanadoo.w-ha.com/app-nod"/>
résiliation immédiate / à échéance (deferred)	<input type="text" value="0"/>
commentaire editeur (adminComment)	<input type="text" value="Commentaire editeur"/>
commentaire client (userComment)	<input type="text" value="Commentaire client"/>
code raison (reasonCode)	<input type="text"/>



**Attention !**

Cette page doit disposer d'un accès restreint pour l'éditeur, et **NE doit pas PAS être accessible aux autres utilisateurs.**

En effet, ils pourraient alors résilier des abonnements, via cette page de « demo » !





### **3.2.3 - Remboursement d'une transaction / « pos\_request ?action=refund »**

L'éditeur a la possibilité de procéder au remboursement d'une transaction, en effectuant une requête (serveur à serveur) vers la plate-forme w-HA.

#### **Attention !**

Afin de respecter le même processus que lors d'une demande de remboursement via le Service Clients d'Orange, **vous devez, après chaque remboursement abouti, procéder également à une requête de résiliation (immédiate).**

#### **3.2.3.1 - Appel de la servlet**

Lorsqu'un internaute souhaite se faire rembourser une transaction, il peut :

- soit contacter le Service Clients d'Orange (Internet)
- soit contacter directement le Service Clients de l'éditeur

Après étude de la demande, l'éditeur peut alors déclencher une requête de remboursement vers la plate-forme w-HA en précisant l'**identifiant de la transaction** à rembourser, ainsi que des **commentaires**.

#### **3.2.3.2 - Exemple d'appel de la servlet**

Afin d'effectuer le remboursement d'une transaction, l'éditeur va appeler la servlet, **via son client HTTP** (à développer), en lui passant les paramètres nécessaires.

Pour ce faire, il utilisera une URL de la forme :

```
http://wha.marchand.com/bundle/pos_request?  
action=refund  
&mid=10  
&trxid=6-17141248844587211  
&url=https://wanadoo.w-ha.com/app-node-mct/responder  
&userComment=commentaire client  
&adminComment=commentaire editeur  
&reasonCode=110
```

#### **3.2.3.3 - Paramètres d'appel de la servlet**

Pour le remboursement d'une transaction, la servlet « pos\_request » est appelée avec les paramètres suivants :

**action=refund** : pour indiquer à la servlet qu'il s'agit d'une requête de remboursement de transaction

**mid=Identifiant de boutique** : identifiant de la boutique concernée par la requête de « confirmation » ; les valeurs de keyId et keyValue correspondantes sont récupérées par la servlet dans le fichier « merchants.xml ».

**trxid=Identifiant de la transaction** : identifiant de la transaction à rembourser.

**url=Url du nœud w-HA (responder)** : url du nœud w-HA qui va réceptionner la requête de remboursement de transaction.

L'url du nœud Orange (Internet) est : https://wanadoo.w-ha.com/app-node-mct/responder

**userComment=Commentaire du client** : commentaire du client sur la (les) raison(s) de sa demande de remboursement de la transaction à l'éditeur



**adminComment=Commentaire de l'éditeur** : commentaire de l'éditeur sur la (les) raison(s) de sa demande de remboursement de la transaction à w-HA

**reasonCode=Code du motif du remboursement** :

Code (r=)	Commentaire
110	Remboursement suite à demande internaute (autre)
111	Remboursement suite à demande internaute (commande multiple)
112	Remboursement suite à demande internaute (ticket facturé plusieurs fois)
113	Remboursement suite à demande internaute (service non délivré)
114	Remboursement suite à demande internaute (achat contesté)
115	Remboursement suite à demande internaute (achat enfants)
116	Remboursement suite à demande internaute (transaction liée à abonnement résilié ou non souhaité)
117	L'abonnement est résilié : je souhaite le remboursement de la transaction associée
118	Le service délivré ne correspond pas à mes attentes
119	J'ai souscrit par erreur cet abonnement plusieurs fois

### 3.2.3.4 - Fonctionnement interne de la servlet

Lorsqu'elle est appelée avec le paramètre action=refund, la servlet « pos\_request » réalise les actions suivantes :

- récupère dans le fichier « web.xml » les informations relatives à la boutique (merchantId)
- récupère les paramètres de la boutique (merchantId, keyId, keyValue) dans le fichier « merchants.xml »,
- crée une URL sécurisée (https) et signé, vers la plate-forme w-HA et contenant ces informations, ainsi que l'identifiant de la transaction pour laquelle un remboursement est demandé.
- déclenche une requête https (en background) vers la plate-forme w-HA.

### 3.2.3.5 - Exemple d'URL générée par la servlet

Exemple d'URL de demande de remboursement générée par la servlet « pos\_request » :

```
https%3A%2F%2Fwanadoo.w-ha.com%2Fapp-node-mct%2Fresponder%3Fm%3Dh%3Dfd4685944ae52d91601e343f7f561d3c%3Bp%3D13%3Bk%3D13%3Bv%3D3%3A%7Bc%3Dm_fullRefund%3Bv%3D%7Badmincom%3DCommentaire%20editeur%3Brid%3Drq004442%3Btrxd%3D6-17141248844587211%3Breason%3D110%3Bd%3D0%3Busercom%3DCommentaire%20client%3B%7D%7D
```

Pour plus de lisibilité, en URL décodée :

```
https://wanadoo.w-ha.com/app-node-mct/responder?
m=h=fd4685944ae52d91601e343f7f561d3c;
p=5XXX;
k=5XXX;
v=3:{
c=m_fullRefund;
v={
admincom=Commentaire editeur;
rid=rq004442;
trxd=6-17141248844587211;
reason=110;
d=0;
usercom=Commentaire client;}}
```

*Cette URL est sécurisée (https) et signée.*

Les informations importantes contenues dans cette URL sont :

- » un hmac, pour assurer l'intégrité du message (h=)



- » la commande « m\_fullRefund » (c=)
- » l'identifiant de la transaction à rembourser (trxId=)

### 3.2.3.5 - Réponse de w-HA à la servlet

Lorsque l'identifiant de transaction à rembourser existe et est dans un état permettant son remboursement, la plate-forme w-HA retourne :

**L'identifiant de la demande de remboursement : trxid=RR1234567891234567**

Remarque : Il n'y a pas de lien entre l'identifiant de la demande de remboursement et l'identifiant de la transaction associée.

Si une erreur se produit, la plate-forme retourne un message spécifique, selon la nomenclature suivante :

Message	Libellé
<code>h=a0e198418d3a05ed657c1cfce78c0964;p=5XXX;k=5XXX;v=2:{c=ex;v={m=INVALID_MERCHANT_INFO;t=com.ipin.core.api.node.pos.MerchantTransactionRefundManager\$RefundException;c=2;}}</code>	Echec du remboursement
<code>e=3</code>	Mauvais hmac
<code>e=15</code>	Erreur dans les paramètres transmis (exemple : "trxid=" au lieu de "trxId=")

Exemple :

```
h=a0e198418d3a05ed657c1cfce78c0964;p=5XXX;k=5XXX;v=2:{c=ex;v={m=INVALID_MERCHANT_INFO;t=com.ipin.core.api.node.pos.MerchantTransactionRefundManager$RefundException;c=2;}}
```

Remarque : Si on essaye de **rembourser plusieurs fois la même transaction**, la plate-forme retournera pour la **première** demande de remboursement, par exemple, "trxid=RR1234567891234567", mais pour les **suivantes**, elle retournera "h=a0e198418d3a05ed657c1cfce78c0964;p=5XXX;k=5XXX;v=2:{c=ex;v={m=INVALID\_MERCHANT\_INFO;t=com.ipin.core.api.node.pos.MerchantTransactionRefundManager\$RefundException;c=2;}}", le remboursement étant déjà effectif.

### 3.2.3.7 - Exemple d'utilisation de la servlet : formulaire web

w-HA met également à disposition de l'éditeur une interface Web utilisant la servlet « pos\_request » (action=refund).

Elle permet à l'éditeur :

- de bien assimiler le fonctionnement de la servlet
- de réaliser des remboursements de transaction manuellement, si nécessaire

Cette interface Web se trouve à l'URL suivante :

[http://wha.marchand.com/demo/bundle/html/remboursement\\_transaction.html](http://wha.marchand.com/demo/bundle/html/remboursement_transaction.html) (\*) :

(\*) remplacer wha.marchand.com par l'adresse IP ou le nom de domaine public du serveur sur lequel est installée l'application w-HA



L'appel de la servlet `"/pos_request?action=refund"` peut être simulée - via le formulaire ci-dessous :

merchantId (mid)	<input type="text" value="12"/>
identifiant transaction (trxid)	<input type="text" value="6-4111235544454210"/>
url du noeud (url)	<input type="text" value="https://wanadoo.w-ha.com/app-nod"/>
commentaire client (userComment)	<input type="text" value="Commentaire utilisateur"/>
commentaire editeur (adminComment)	<input type="text" value="Commentaire éditeur"/>
code raison (reasonCode)	<input type="text" value="Code erreur"/>
<input type="button" value="Remboursement d'une Transaction"/>	

**Attention !**

Cette page doit disposer d'un accès restreint pour l'éditeur, et **NE doit pas PAS être accessible aux autres utilisateurs.**

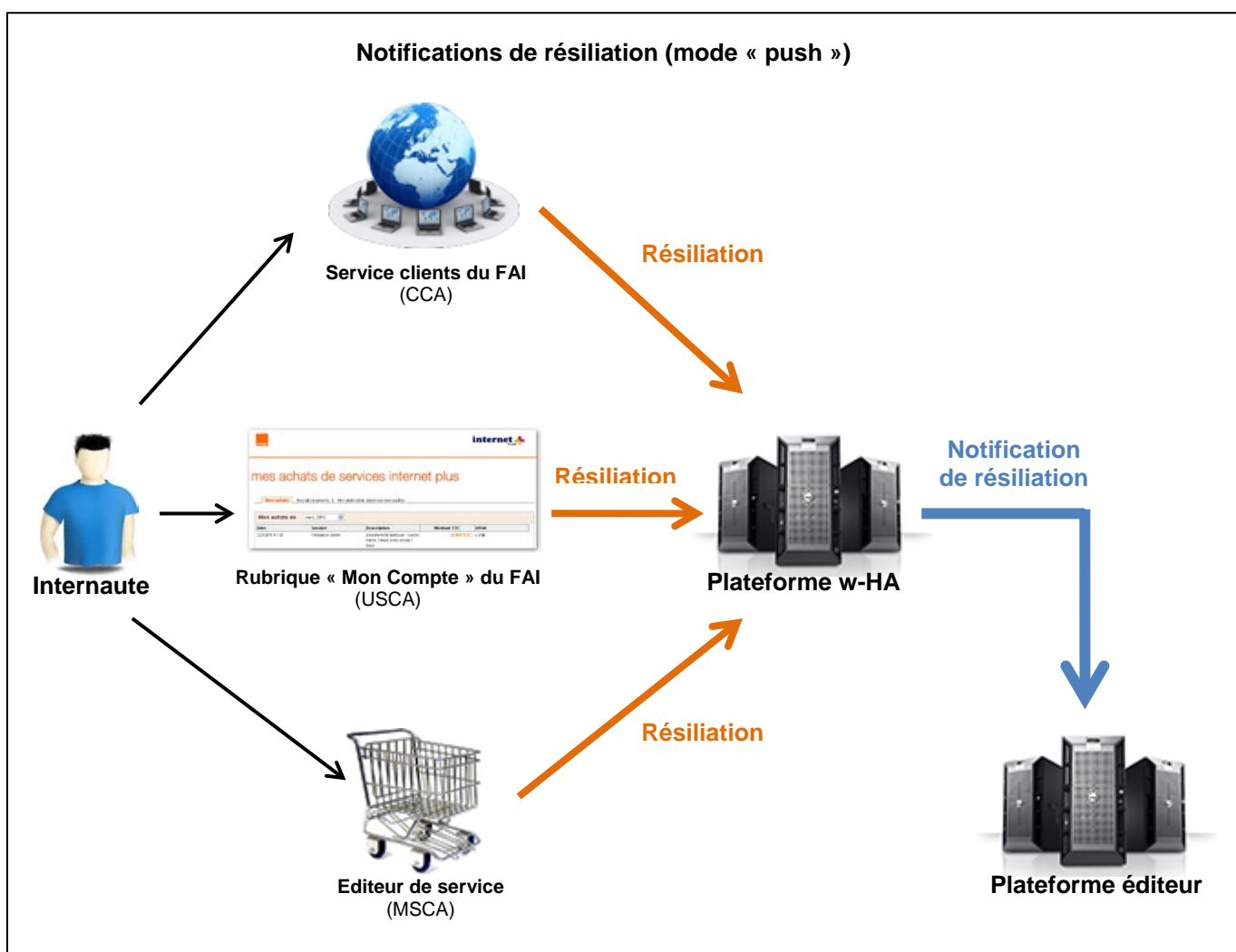
En effet, ils pourraient alors rembourser des transactions, via cette page de « demo » !

### 3.2.4 - Notification de résiliation / « responder ?c=NMPOC\_NEW »

Le contrôle d'accès étant réalisé par l'éditeur, celui-ci a besoin d'être informé, en temps réel, des résiliations d'abonnements en cours.

La plate-forme (v3.5) de w-HA prévoit un **mécanisme de « push » (w-HA vers Editeur)**, permettant à l'éditeur de mettre à jour immédiatement sa base de données avec le nouveau statut de l'abonnement.

Au niveau de l'application installée sur le serveur de l'éditeur, **une servlet « responder »** (ou un développement similaire autour du SDK), **est chargée de récupérer ces informations.**





### 3.2.4.1 - Appel de la servlet

Un client peut demander à tout instant la résiliation d'un de ses abonnements.  
Il existe 4 outils qui permettent de résilier un abonnement :

1. L'espace "Mon compte" du client
2. Le MSCA de l'éditeur
3. L'application marchande (action = cancelOffer)
4. Le CCA (outil de suivi des FAI)

Suivant l'outil utilisé pour la résiliation, l'éditeur ne va pas recevoir les mêmes notifications de résiliation.

Si la résiliation est demandée via l'espace "Mon Compte" de l'abonné, l'éditeur recevra alors 2 notifications :

- Une première notification annonçant la demande de résiliation
- Une seconde annonçant la résiliation effective.

Dans le cas de la résiliation **via les 3 autres outils**, l'éditeur ne recevra qu'une seule notification : lorsque la résiliation sera effective.

#### **CAS 1 : Résiliation via l'espace "Mon Compte" du FAI**

Une notification sera envoyée par W-ha **vers la servlet Responder** de l'éditeur (requête serveur à serveur) :

**\* Au moment où la résiliation est demandée par l'internaute via l'Espace Client de son opérateur.**

Un code est utilisé pour cette notification (**r=200**).

Cette notification intègre **le jour et l'heure à laquelle la demande a été faite**.

Le **champ commentaire** (c=) est utilisé pour transmettre cette information.

[http://www.w-ha.com/kit/responder.php?m=h=4f3d343dfca552db2e65e9f306cde62f;p=515;k=515:v=3:{c=NMPOC\\_NEW;v={uo=6-U2151766123994976;r=200;p=@515@Merchant515-1269871784986-product;o=test;c=29/03/10 16:53:28;}}](http://www.w-ha.com/kit/responder.php?m=h=4f3d343dfca552db2e65e9f306cde62f;p=515;k=515:v=3:{c=NMPOC_NEW;v={uo=6-U2151766123994976;r=200;p=@515@Merchant515-1269871784986-product;o=test;c=29/03/10 16:53:28;}})

**\* Au moment où la résiliation est effective :**

Le code utilisé pour cette notification est **r=201**.

Ici, le champ commentaire (c=) est utilisé pour transmettre **le jour et l'heure de la résiliation effective** de l'abonnement par w-HA.

[http://www.w-ha.com/kit/responder.php?m=h=4f3d343dfca552db2e65e9f306cde62f;p=515;k=515:v=3:{c=NMPOC\\_NEW;v={uo=6-U2151766123994976;r=201;p=@515@Merchant515-1269871784986-product;o=test;c=05/04/10 16:34:28;}}](http://www.w-ha.com/kit/responder.php?m=h=4f3d343dfca552db2e65e9f306cde62f;p=515;k=515:v=3:{c=NMPOC_NEW;v={uo=6-U2151766123994976;r=201;p=@515@Merchant515-1269871784986-product;o=test;c=05/04/10 16:34:28;}})

**Remarque :**

L'URL de la servlet « responder » aura été communiquée à w-HA au préalable.

**CAS 2 : Résiliation via le CCA, le MSCA ou l'application marchande**

Une notification sera envoyée par W-ha **vers la servlet Responder** de l'éditeur (requête serveur à serveur) **uniquement lorsque la résiliation sera effective.**

Le champ commentaire (c=) est utilisé pour transmettre **le jour et l'heure de la résiliation effective.**

[http://www.w-ha.com/kit/responder.php?m=h=994e932713c6e621fa57dc3039cfd38a;p=515;k=515;v=3:{c=NMPOC\\_NEW;v={uo=6-U8149894128200400;r=110;p=@515@acces\\_limite2;o=acces\\_limite\\_2;c=08/06/10 12:35:06;}}](http://www.w-ha.com/kit/responder.php?m=h=994e932713c6e621fa57dc3039cfd38a;p=515;k=515;v=3:{c=NMPOC_NEW;v={uo=6-U8149894128200400;r=110;p=@515@acces_limite2;o=acces_limite_2;c=08/06/10 12:35:06;}})

Chaque notification reçue par l'éditeur comporte un code de résiliation (**r=**) dont les équivalences sont données ci-dessous.

Code (r=)	Commentaire
100	Résiliation de l'abonnement via le CCA (Service Clients du FAI) motif = Connexion interrompue ou téléchargement incomplet
101	Résiliation effective de l'abonnement via le CCA (Service Clients du FAI) motif = Article erroné OU <b>Abonnement non confirmé dans le délai imparti (24 heures)</b>
102	Résiliation effective suite à fin de validité de l'abonnement (expiration de la licence)
103	Résiliation effective liée à échec de facturation de l'abonnement
104	Résiliation effective suite à fermeture du compte utilisateur par l'opérateur (via le bus de provisionning)
105	Résiliation effective suite à fermeture du compte utilisateur via le CCA (outil du Service Clients du FAI)
106	Résiliation effective suite à non paiement d'AAR par facture
107	Résiliation effective suite à non facturabilité de l'utilisateur
110	Résiliation effective de l'abonnement via le CCA (Service Clients du FAI) motif = autre
111	Résiliation effective suite à demande internaute (commande multiple)
112	Résiliation effective suite à demande internaute (ticket facturé plusieurs fois)
113	Résiliation effective suite à demande internaute (service non délivré)
114	Résiliation effective suite à demande internaute (achat contesté)
115	Résiliation effective suite à demande internaute (achat enfants)
116	Résiliation effective suite à demande internaute (transaction liée à abonnement résilié ou non souhaité)
117	L'abonnement est résilié : je souhaite le remboursement de la transaction associée
118	Le service délivré ne correspond pas à mes attentes
119	J'ai souscrit par erreur cet abonnement plusieurs fois
200	<b>Demande</b> de résiliation de l'abonnement par l'internaute via l'Espace Client de son Operateur (USCA)
201	Résiliation <b>effective</b> de l'abonnement par l'internaute (rubrique « Espace Client » de son opérateur (USCA)



### 3.2.4.2 - Exemple d'appel de la servlet

Lorsque la résiliation d'un abonnement se produit, l'URL appelée par la plate-forme w-HA est de la forme (URL encodée) :

```
http%3A%2F%2Fwha%2Emarchand%2Ecom%2Fbundle%2Dresponder%2Fresponder%3Fm%3Dh%3Df5938cde8dd6bfc67da921b4b8bf350%3B%0D%0A%3D10%3Bk%3D10%3Bv%3D3%3A%7Bc%3DNMPOC%5FNEW%3Bv%3D%7Buo%3D3D6%2DU7141248844587211%3Br%3D103%3Bp%3D%4010%40P1%7C%4010%40P2%7C%4010%40P4%7C%3B%0D%0Ao%3DCR005%3Bc%3DCancelled+due+to+charge+processing+functional+failure%3B%7D%7D%0D%0A
```

Pour plus de lisibilité, en URL décodée :

```
http://wha.marchand.com/bundle-responder/responder?  
m=  
h=f5938cde8dd6bfc67da921b4b8bf350;  
p=10 ;  
k=10 ;  
v=3 :{  
c=NMPOC_NEW;  
v={  
uo=6-U7141248844587211;  
r=103;  
p=@10@P1|@10@P2|@10@P4|;  
o=CR005;  
c=Cancelled due to charge processing functional failure;}}
```

*Cette URL est sécurisée (https) ou non, et signée.*

Les informations importantes contenues dans cette URL sont :

- un hmac pour assurer l'intégrité du message (h=),
- la commande « NMPOC\_NEW » (c=)
- l'identifiant de l'abonnement qui vient d'être résilié (uo=)
- le code de la résiliation (r=)
- l'offre concernée par la résiliation (o=)





### **[Optionnel] Développement d'un « responder » : précision sur le calcul du hmac**

Dans le cas où l'éditeur de service souhaite développer son propre « responder » en utilisant le protocole w-HA indiqué si dessus, il a besoin de calculer un hmac, afin de s'assurer de l'intégrité des informations reçus.

Le calcul du hmac se fait de la manière suivante :

Si le message (notification de résiliation) reçu est :

```
m=h=f5938cde8dd6bfc67da921b4b8bf350;p=10 ;k=10 ;v=3 :{c=NMPOC_NEW;v={uo=6-  
U7141248844587211;r=103;p=@10@P1|@10@P2|@10@P4|;o=CR005;c=Cancelled due to  
charge processing functional failure;}}
```

alors, le hmac (h=) se calcule de la façon suivante :

```
h= hmac_MD5("c=NMPOC_NEW;v={uo=6-  
U7141248844587211;r=103;p=@10@P1|@10@P2|@10@P4|;o=CR005;c=Cancelled due to  
charge processing functional failure;}", "Valeur_de_la_KeyValue")
```

où hmac\_MD5 est la fonction de calcul du HMAC/MD5.

Une description de l'algorithme HMAC/MD5, incluant des exemples, est disponible à l'URL :  
<http://www.ietf.org/rfc/rfc2104.txt?number=2104>

Un exemple de calcul de HMAC/MD5 en php est donné au paragraphe § 0

#### **3.2.4.3 - Fonctionnement interne de la servlet**

La servlet « responder » peut fonctionner :

##### **Mode « Logs »**

Le paramètre <urlRedirect> dans /bundle-responder/WEB-INF/merchants.xml **N'est PAS** renseigné.  
Le paramètre <logFileName>, dans /bundle-responder/WEB-INF/merchants.xml **est** renseigné.

##### **Mode « Redirection »**

Le paramètre <urlRedirect>, dans /bundle-responder/WEB-INF/merchants.xml **est** renseigné.  
Le paramètre <logFileName>, dans /bundle-responder/WEB-INF/merchants.xml **N'est PAS** renseigné.

##### **Mode « Redirection+Logs »**

Le paramètre <urlRedirect>, dans /bundle-responder/WEB-INF/merchants.xml **est** renseigné.  
Le paramètre <logFileName>, dans /bundle-responder/WEB-INF/merchants.xml **est** renseigné.

Lorsque la servlet reçoit une notification de résiliation, elle réalise les actions suivantes :

- écriture des informations de résiliation dans le fichier de log « log\_notification\_XXX.txt » renseigné au niveau du paramètre « logFilename » dans le fichier /bundle-responder/WEB-INF/merchants.xml
- redirection vers l'URL renseignée au niveau du paramètre « urlRedirect » dans le fichier /bundle-responder/WEB-INF/merchants.xml, si le paramètre est renseigné

#### **3.2.4.4 - Parsing du fichier « log\_notification\_XXX.txt »**

Afin de prendre en compte les notifications de résiliation des abonnements, l'éditeur doit **analyser régulièrement le fichier de log « log\_notification\_XXX.txt »** pour en extraire les identifiants d'Abonnements (uoid) résiliés et mettre à jour sa base de données.



### 3.2.4.5 - Redirection des notifications vers une URL

Si le paramètre « urlRedirect » du fichier /bundle-responder/WEB-INF/merchants.xml est renseigné, l'information de notification **sera récupérée au niveau de cette URL** qui doit être un exécutable (servlet, cgi, asp, php, .....)

L'url appelée par le Kit est dans la forme suivante (url encodée) :

```
http://wha.marchand.com/notification_resiliation.php?hmac%3Df5938cde8dd6bfc67da921b4b8bf350%26c%3DCommentaire%2Bpar%2Bdefault%26o%3DO1%26p%3D%405XXX%40P1%26r%3D110%26uo%3D6-U5112074834229514
```

URL decodée :

```
http://wha.marchand.com/notification_resiliation.php?
hmac=f5938cde8dd6bfc67da921b4b8bf350
&c=Commentaire+par+default
&o=O1
&p=@5XXX@P1
&r=110
&uo=6-U5112074834229514
```

### 3.2.4.6 - Réponse de la servlet à w-HA : acquittement

En mode « **Logs** » (paramètre `<urlRedirect>` non renseigné), la servlet **responder** envoie **automatiquement** un accusé de réception (ou acquittement) « ack » vers la plate-forme w-HA.

En mode « **Redirection** » (paramètre `urlRedirect` renseigné), c'est à l'éditeur d'envoyer cet acquittement vers la plate-forme w-HA.

Cet acquittement doit être réalisé à partir du « programme » placé au niveau de l'urlRedirect, et en charge de récupérer les notifications de résiliation et de mettre à jour en temps réel la base de données de l'éditeur.

La requête d'accusé de réception est de la forme suivante :

```
h=a45b921b4321fcb67da921b4b8bf350;p=5XXX;k=5XXX;v=3:{c=ack}
```

Exemple php pour envoyer un acquittement (ack) à w-HA (notification de résiliation) :

```
<?php
//Definition of a function which caculates hmac
function hmac($string, $key)
{
    $byte = 64; // byte length for md5
    if (strlen($key) > $byte) {
        $key = pack("H*",md5($key));
    }
    $key = str_pad($key, $byte, chr(0x00));
    $in_pad = str_pad("", $byte, chr(0x36));
    $out_pad = str_pad("", $byte, chr(0x5c));
    $k_in_pad = $key ^ $in_pad ;
    $k_out_pad = $key ^ $out_pad;

    return md5($k_out_pad . pack("H*",md5($k_in_pad . $string)));
}

//Ceation of the message (ack) to send to w-HA
$hmac = hmac("c=ack", "your Keyvalue");
$message = "h=".$hmac.";p=your Mctid;k=your Keyid;v=3:{c=ack}";
echo $message;
?>
```



Les informations importantes contenues dans cette URL sont :

- **h=** : le hmac permettant d'assurer l'intégrité du message
- **p=** : La valeur du paramètre « mctld » du fichier « merchants.xml » (cf. § 0)
- **k=** : La valeur du paramètre « mctKeyld » du fichier « merchants.xml » (cf. § 0)
- **v=3** : La version du format de la requête
- **c=ack** : Le message d'acquiescement {c=ack}

Les valeurs **en bleu**, sont à modifier en fonction des paramètres de la (des) boutique(s) de l'éditeur.

Le calcul du hmac se fait de la manière suivante :

Si le message (acquiescement de notification de résiliation) envoyé est :

***h=a45b921b4321fcb67da921b4b8bf350;p=10;k=10;v=3:{c=ack}***

alors, le hmac (h=) se calcule de la façon suivante :

***h=hmac\_MD5("c=ack","VALEUR\_DE\_LA\_MCTKEYVALUE")***

où hmac\_MD5 est la fonction de calcul du HMAC/MD5 et VALEUR\_DE\_LA\_MCTKEY est égal à la valeur du paramètre « MctKey » du fichier « merchants.xml » (cf. § 0).

Une description de l'algorithme HMAC/MD5, incluant des exemples, est disponible à l'URL : <http://www.ietf.org/rfc/rfc2104.txt?number=2104>

Un exemple de calcul de HMAC/MD5 en php est donné au paragraphe § 0

### ***3.2.4.7 - En cas de non réponse de la servlet***

Si la plate-forme w-HA ne reçoit pas l'accusé de réception de la résiliation, celle-ci déclenche de nouvelles tentatives : au total, **4 tentatives à 1 heure d'intervalle**.

D'autre part, toutes les notifications dont l'envoi a échoué après les 4 tentatives sont empilées et stockées **sur la plate-forme w-HA : les notifications sont bufférisées** en attendant la prochaine tentative d'émission.

La prochaine tentative survient dès qu'une nouvelle notification est transmise avec succès. Dans ce cas, toutes les notifications en attente sont ré-émises.

Il convient de noter que les notifications bufférisées ne sont pas automatiquement débloquées dès que la plate-forme de l'éditeur est à nouveau opérationnelle. En effet, tant que la plate-forme w-HA ne reçoit pas de nouvelle demande de notification, l'état de la plate-forme de l'éditeur ne sera pas testé et les notifications ne seront donc pas ré-émises.

## 4 - Ce que le marchand doit mettre en œuvre pour utiliser le Kit v3.5

### 4.1 - Création des offres via le MSCA

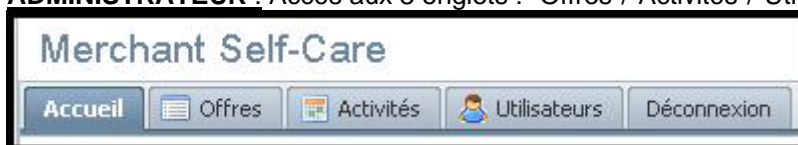
La première étape consiste à se connecter sur l'interface Web (MSCA) permettant à l'éditeur des créer ses offres au niveau de la plate-forme w-HA.

Cette interface est disponible à l'URL suivante : <https://mps.w-ha.com/app-msca/>

L'éditeur doit impérativement se **créer** au moins 1 **compte ADMINISTRATEUR** (avec un Login et un Mot de Passe de son choix) lors de sa première connexion au MSCA. Suite à la création de ce premier compte ADMINISTRATEUR, l'éditeur **ne doit plus utiliser** le login et le mot de passe transmis par w-HA pour la première connexion.

Il existe 3 types de comptes :

**ADMINISTRATEUR** : Accès aux 3 onglets : "Offres"/"Activités"/"Utilisateurs"



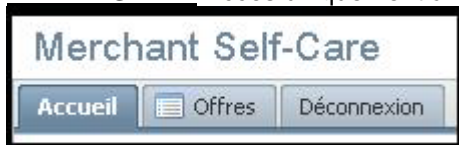
Le compte Administrateur permet de gérer les autres utilisateurs.

Il permet de créer et/ou de supprimer d'autres comptes : Administrateur, Service Client, Webmaster.

**SERVICE CLIENT** : Accès uniquement à l'onglet : "Activité"



**WEBMASTER** : Accès uniquement à l'onglet : "Offres"



### 4.2 - Paramétrage des fichiers de configuration « \*.xml »

L'éditeur devra paramétrer les fichiers de configuration de la servlet « pos\_bundle »

#### web.xml

Le fichier « web.xml » doit, à priori, être paramétré une seule fois par l'éditeur, avant le passage en production.

#### merchants.xml

Le fichier « merchants.xml » sera modifié par l'éditeur à chaque fois que celui-ci ouvrira une nouvelle boutique (merchantId) utilisant la fonctionnalité abonnement du Kit v3.5.

#### productbundle\_XXX.xml

Le fichier « productbundle\_XXX.xml », quant à lui, sera modifié par le marchand à chaque ajout, modification ou suppression de produits, **en cohérence avec les offres créées au niveau du MSCA.** (Ne pas oublier de redémarrer Tomcat après chaque modification dans un fichier « .xml »)

### 4.3 - Confirmation de souscription

Si l'éditeur décide de gérer une confirmation différée de la livraison (autoConfirm = false), il devra :

- soit **développer lui-même la requête de « confirm » vers la plate-forme w-HA**, à partir des APIs Java fournies par w-HA (cf. Annexe I 0).
- soit **développer un client http appelant la servlet « pos\_request » (action=offerConfirm)** (cf. § 0)

### 4.4 - Notifications de résiliation : mode « push »

L'éditeur décide d'utiliser la servlet « responder », faisant partie intégrante du Kit v3.5, il devra :

En mode « redirection » :

**Récupérer les notifications de résiliation en temps réel** en utilisant la redirection vers une URL spécifique et **envoyer un acquittement (c=ack)** vers la plate-forme w-HA, pour chaque notification reçue.

En mode « logs » :

**Analyser le fichier de log généré (log\_notification\_XXX.txt) afin de mettre à jour sa base de données des abonnements.**

### 4.5 - Réalisation de pages Web de présentation des offres

Sur son site Web, l'éditeur présente ses différentes offres d'abonnement :

<p>5 € / semaine</p> <p>Cette formule offre un accès illimité à :</p> <ul style="list-style-type: none"><li>- Mon Site Web</li></ul> <p><b>Suite &gt;&gt;</b></p> <p>Abonnement hebdomadaire à tacite reconduction, sans engagement de durée, et résiliable à tout moment.</p>	<p>10 € / mois</p> <p>Cette formule offre un accès illimité à :</p> <ul style="list-style-type: none"><li>- Mon Site Web</li></ul> <p><b>Suite &gt;&gt;</b></p> <p>Abonnement mensuel à tacite reconduction, sans engagement de durée, et résiliable à tout moment.</p>	<p>15 € / trimestre</p> <p>Cette formule offre un accès illimité à :</p> <ul style="list-style-type: none"><li>- Mon Site Web</li></ul> <p><b>Suite &gt;&gt;</b></p> <p>Abonnement trimestriel à tacite reconduction, sans engagement de durée, et résiliable à tout moment.</p>
authorizeOffer(O1)	authorizeOffer(O2)	authorizeOffer(O3)



## 4.6 - Affichage des écrans w-HA en page courante

### Côté éditeur de service :

Compte tenu de l'évolution permanente des navigateurs web (Internet Explorer, Firefox, Google Chrome ...), associés à des services de protection (antivirus, pop-up killer, ....) de plus en plus complexes, **l'affichage des pages liées au paiement Internet+ (choix de l'opérateur, authentification, panneau de paiement) doit se faire en page courante.**

Depuis la page de paiement, sur le site web du marchand, le code html à utiliser pour l'appel de la servlet de paiement w-HA (pos\_bundle) est le suivant :

```
<a href="/bundle/pos_bundle?action=authorizeOffer&oid=O1&wha_desc2=current">  
<image src="../images/bouton_paiement_internetplus.gif" /> (1)  
</a>
```

On utilise dans ce cas un lien *href* classique

On peut également utiliser le code html suivant :

```
<input type="image" src="../bouton_paiement_internetplus.gif"  
OnClick="window.location.href=  
'/bundle/pos_bundle?action=authorizeOffer&oid=O1&wha_desc2=current'"> (1)
```

La méthode *location.href* permet d'ouvrir directement le panel de paiement w-HA dans la page courante. Cependant, dans cette implémentation, on utilise du code javascript pour afficher les écrans (fonction *OnClick* et *location.href*).

Le Kit redirige l'internaute vers "route.w-ha.com" avec "wha\_desc2=current" en paramètre supplémentaire. Ces paramètres apparaissent dans les merchant properties (mp) et sont alors précédés de "\_ap\_" donc on obtient dans les mp : "\_ap\_wha\_desc2=current"

exemple :

```
https://route.w-ha.com/app-bundlepurchase/node?m=h=9db95d8f245ab1ceed02d114ffdbb45e:  
p=5XXX;k=5XXX;v=3;{c=OfferAuthorizeReq;v={mUrl=http://localhost:8080/bundle/pos_bundle;oid=O5;  
mp={_ap_userId=fmoreau;_ap_wha_desc2=current;cur=EUR;amt=0.01;};}}
```

### Côté plate-forme w-HA :

La plate-forme w-HA réalise un contrôle applicatif sur le paramètre **wha\_desc2=current**, afin d'optimiser la cinématique de paiement en mode « page courante ».

**Il est donc impératif pour l'éditeur d'associer le paramètre *wha\_desc2=current* à une cinématique de paiement en *page courante*.**

#### **Attention :**

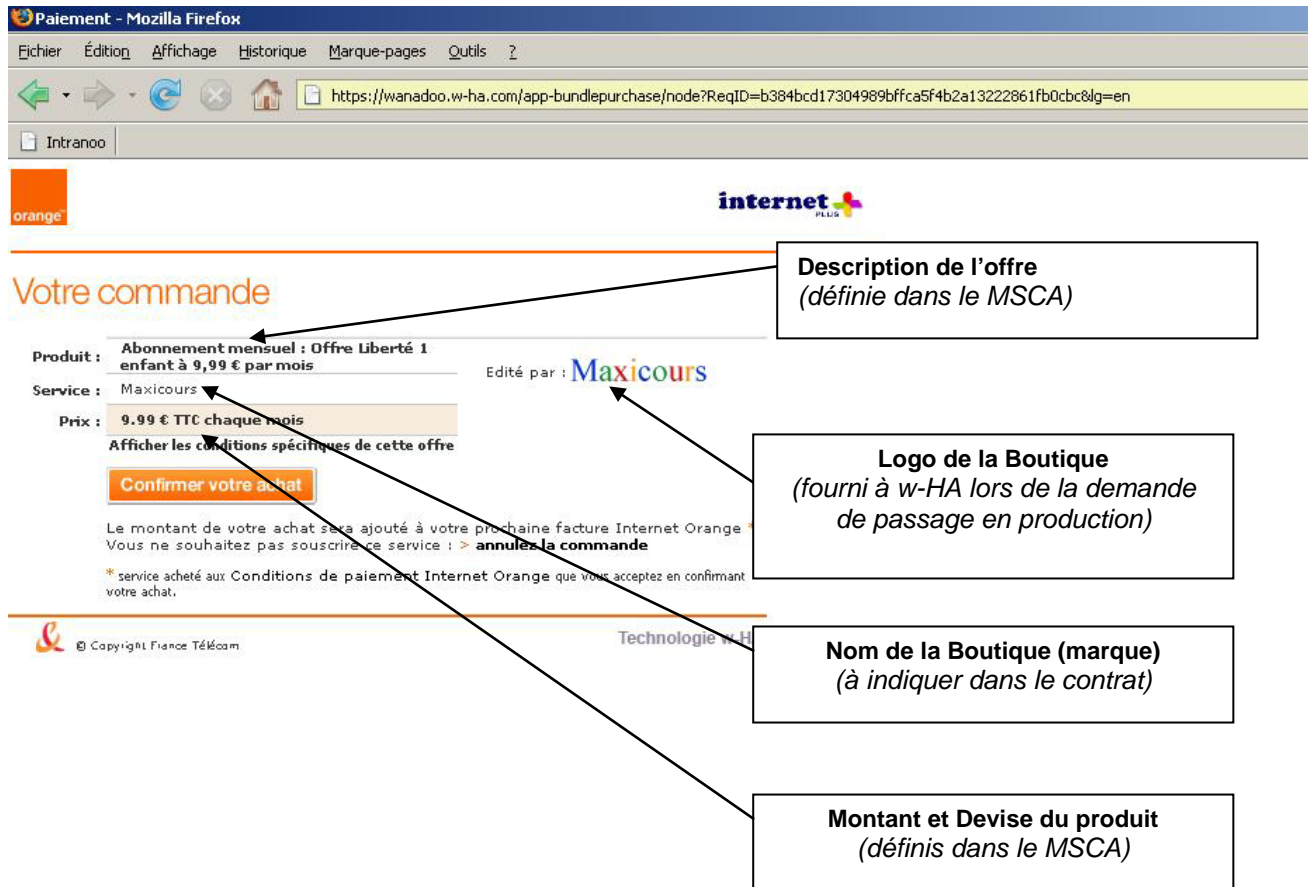
Vérifier que le nouveau paramètre "wha\_desc2=current" est bien pris en compte lors du calcul du hmac lors de la redirection du client vers :

- la **fulfillmenturl** (url paramétrée dans tomcat/webapps/bundle/WEB-INF/prodcutbundle\_5XXX.xml)
- la **mcttrxCancelFromPaymentPanelUrl** (url paramétrée dans tomcat/webapps/bundle/WEB-INF/merchant.xml)

(1) Le fichier « ../images/bouton\_paiement\_internetplus.gif » contient l'image du bouton de paiement Internet+ à afficher sur le site web du marchand.

L'ensemble des boutons utilisables est disponible sur [www.internetplus.fr](http://www.internetplus.fr) > boîte à outils

Exemple d'affichage d'un panneau de paiement Orange en page courante :



**Paieement - Mozilla Firefox**

Fichier Édition Affichage Historique Marque-pages Outils ?

https://wanadoo.w-ha.com/app-bundlepurchase/node?ReqID=b384bcd17304989bffa5f4b2a13222861fb0cbc&lg=en

Intranoo

orange internet PLUS

**Votre commande**

**Produit :** Abonnement mensuel : Offre Liberté 1 enfant à 9,99 € par mois

**Service :** Maxicours

**Prix :** 9.99 € TTC chaque mois

Confirmer votre achat

Le montant de votre achat sera ajouté à votre prochaine facture Internet Orange

Vous ne souhaitez pas souscrire ce service : > **annulez la commande**

\* service acheté aux Conditions de paiement Internet Orange que vous acceptez en confirmant votre achat.

© Copyright France Télécom Technologie w-HA

**Description de l'offre (définie dans le MSCA)**

**Logo de la Boutique (fourni à w-HA lors de la demande de passage en production)**

**Nom de la Boutique (marque) (à indiquer dans le contrat)**

**Montant et Devise du produit (définis dans le MSCA)**

Terminé

figure : affichage du « panneau de paiement » Orange



#### 4.7 - Protection du contenu (utilisation du hmac)

Toutes les informations échangées entre la plate-forme de l'éditeur de service et celle de w-HA sont **signées** (HMAC-MD5), ce qui permet **de garantir l'authentification des plates-formes et l'intégrité des informations échangées**.

La signature électronique repose sur une **clef de cryptage symétrique partagée** par la plate-forme du marchand et celle de w-HA : la **mctKey (ou KeyValue)**

A l'émission de l'information, la plate-forme "A" utilise un algorithme pour générer, à partir de sa clef symétrique et de ces informations, un MAC (Message Authentication Code) qu'il transmet en même temps à la plate-forme "B", dans l'URL.

A réception de l'information et du MAC, la plate-forme "B" utilise le même algorithme pour générer, à partir de la même clef symétrique et de ces informations, un autre MAC.

S'il est identique à celui reçu, il n'y a pas eu modification des informations : la transaction est autorisée.

S'il est différent de celui reçu, des informations ont été modifiées : la transaction est refusée.

**Afin d'empêcher un Internaute de bénéficier gratuitement d'un service en y accédant directement par son URL (fulfillmentUrl), il faut donc vérifier :**

- l'intégrité des paramètres supplémentaires (calcul d'un H-MAC)
- la valeur d'un ou plusieurs paramètres supplémentaires (ex : identifiant de session)

Le H-MAC est généré à partir de l'algorithme H-MAC MD5, avec comme paramètres :

- les paramètres récupérés au niveau de la fulfillmentUrl (en tant que « string » unique),
- la valeur du paramètre « mctKey » (ou KeyValue) du fichier « merchants.xml » (en tant que « string » unique),

Par exemple, si le retour sur l'URL finale (fulfillmentUrl) se fait de la manière suivante :

```
http://marchand.com/payant/paiement_ok.php?sessionId=1234&commandId=abcd&userId=toto  
&hmac=891284e23faa662c033a41dd9905cc10&uid=6-U5151292477228013
```

Le programme (ici "paiement\_ok.php") doit vérifier l'intégrité des paramètres en calculant le H-MAC relatif à ces paramètres, avant de vérifier leur valeur et d'afficher le produit/service acheté (ou en refuser l'accès si nécessaire).

Si la valeur du paramètre "mctKey" est "a1b2c3d4e5", alors le calcul du H-MAC se fait de la manière suivante :

```
mon-hmac = H-MAC ("commandId=abcd&sessionId=1234&uid=6-  
U5151292477228013&userId=toto","a1b2c3d4e5")
```

##### **Attention 1 :**

Les **paramètres supplémentaires** (ou « merchant properties » ou « mp ») passés à la servlet w-HA ne doivent contenir **ni accents, ni caractères spéciaux**.

En effet, l'URL-encodage de ces caractères aboutit à un calcul erroné du H-MAC, et dans ce cas, la livraison du produit/service n'aboutirait pas, alors que l'utilisateur final aura été débité.

##### **Attention 2 :**

L'ordre des paramètres à considérer pour le calcul du H-MAC par l'application w-HA est l'**ordre alphabétique**.



**Attention 3 :**

L'identifiant de l'abonnement (uoid), récupéré au niveau de la fulfillmentUrl, doit également être pris en compte dans le re-calcul du hmac.

Un exemple (basique) de calcul de H-MAC en PHP :

Si la fulfillment URL est :

http://marchand.com/payant/paiement\_ok.php?hmac=891284e23faa662c033a41dd9905cc10&commandId=abcd&sessionId=1234&userId=toto&uoid=6-U5151292477228013

Le calcul du hmac peut se faire de la façon suivante :

```
$hmac_verif = bin2hex(mhash(MHASH_MD5,"commandId=abcd&sessionId=1234&uoid=6-U5151292477228013&userId=toto","a1b2c3d4e5f"));  
if ($hmac_verif == $hmac)  
    print "ok"; else  
    print "no";
```

**Vérification de la valeur d'un ou plusieurs paramètres supplémentaires**

Afin d'empêcher un internaute d'utiliser plusieurs fois une même URL (et donc avoir accès au produit sans payer), l'éditeur doit **vérifier que la livraison liée à cette transaction n'a pas déjà été effectuée.**

Pour ce faire, il peut utiliser l'identifiant d'abonnement (uoid), qui est unique, ou un paramètre supplémentaire de son choix (lors de l'appel de la servlet de demande d'autorisation d'abonnement, cf [3.1.1 - Demande Souscription / « pos\\_bundle?action=authorizeOffer »](#)) (par exemple, une valeur qu'il incrémente à chaque souscription) afin de repérer l'unicité de l'URL de livraison (fulfillment URL).

Lorsque cette URL est appelée, il peut alors vérifier si cet uoid ou si ce paramètre supplémentaire définit une livraison déjà effectuée ou non.

**Au 1<sup>er</sup> appel de l'URL, l'éditeur redirige normalement vers la page de livraison.**

**Pour les éventuels appels ultérieurs, il bloque l'accès** (et affiche une page d'erreur par exemple).



## 5 – ANNEXE I : SDK

### 5.1 - Développement d'un « confirm » de souscription via le SDK

Si l'éditeur de service souhaite enclencher le « confirm » (ou le « cancel ») de la souscription après s'être assuré que celle-ci a bien été prise en compte (ou non) dans sa base de données, il doit utiliser les APIs Java contenues dans le SDK.

Une Javadoc (« javadoc.zip ») est mise à disposition des éditeurs qui en font la demande auprès de leur contact technique w-HA.

L'éditeur de service doit utiliser les classes Java suivantes :

- sdk-valista0.jar
- CORE-API-1-18-sp0r3-MERCHANT-SDK-lib.jar

#### 5.1.1 - Implémentation

L'implémentation se fait de la manière suivante :

La classe Java à construire doit hériter de la classe suivante :

#### **Proxy**

Les méthodes à implémenter pour la confirmation ou l'annulation sont :

#### CancelOffer

```
public void cancelOffer(java.lang.String nodeResponderURL,
                        long merchantId,
                        long keyId,
                        java.lang.String userOfferId)
    throws com.valista.core.api.node.pos.MerchantOfferManager.CancellationException,
           java.rmi.RemoteException
com.valista.core.api.node.pos.MerchantOfferManager.CancellationException
java.rmi.RemoteException
```

#### ConfirmOffer

```
public void confirmOffer(java.lang.String nodeResponderURL,
                        long merchantId,
                        long keyId,
                        java.lang.String userOfferId)
    throws com.valista.core.api.node.pos.MerchantOfferManager.ConfirmationException,
           com.valista.core.api.node.pos.MerchantOfferManager.ReplacementException,
           java.rmi.RemoteException
com.valista.core.api.node.pos.MerchantOfferManager.ConfirmationException
com.valista.core.api.node.pos.MerchantOfferManager.ReplacementException
java.rmi.RemoteException
```

### **5.1.2 - Exemple de code**

*Proxy proxyValista = new Proxy(keyTable) ;*

```
//CancelOffer
try
{
proxyValista.cancelOffer(nodeURL, merchantId, keyId, uoid);
forwardToMessageURL(request, response, "CancelSuccess.UserOfferId: " + uoid);
}
catch(com.valista.core.api.node.pos.MerchantOfferManager.CancellationException e)
{
forwardToMessageURL(request, response, "Cancel for" + uoid + "failed<br/>" +
e.toString());
}

//ConfirmOffer
try
{
proxyValista.confirmOffer(nodeURL, merchantId, keyId, uoid);
forwardToMessageURL(request, response, "ConfirmSuccess.UserOfferId: " + uoid);
}
catch(com.valista.core.api.node.pos.MerchantOfferManager.ConfirmationException e)
{
forwardToMessageURL(request, response, "Confirm for" + uoid + "failed<br/>" +
e.toString());
}
catch(com.valista.core.api.node.pos.MerchantOfferManager.ReplacementException e)
{
forwardToMessageURL(request, response, "Replacement Confirmation failed for" + uoid +
"failed<br/>" + e.toString());
}
```

## **5.2 - Développement d'un « responder » spécifique via le SDK**

Si l'éditeur de service souhaite créer son propre « responder » afin de récupérer les informations de résiliation et les intégrer directement dans sa base de données, il doit utiliser les APIs Java contenues dans le SDK sdk-valista.jar.

Une Javadoc (« javadocResponder.zip ») est mise à disposition des éditeurs qui ont font la demande auprès de leur contact technique w-HA.

L'éditeur de service doit faire hériter sa classe Java des APIs suivantes :

- core-api-merchant-sdk-lib.jar
- core-util-common.jar



### 5.2.1 - Implémentation

L'implémentation se fait de la manière expliquée ci-dessous.

La classe Java à construire doit hériter de la classe suivante :

#### **AbstractNotificationManagerCommand**

Type de commande traitée :

**NMPOC\_NEW**: NotificationManagerPushOfferCancellationData(data);

La méthode à implémenter pour ce type de commande est :

**pushOfferCancellation**(java.lang.String userOfferId, java.lang.String offerId, com.valista.core.api.merchant.node.NotificationManager.CancelledProductInfo[] products, java.lang.String reasonCode, java.lang.String comment)

Avec les interfaces suivantes :

#### **All Implemented Interfaces:**

com.valista.command.Command, com.valista.util.manager.Manager, com.valista.core.api.merchant.node.NotificationManager, java.rmi.Remote

### 5.2.2 - Exemple de code :

```
// NotificationManagerPushOfferCancellationData,
// Interface à implémenter NotificationManager
if(cmdType.equals(« NMPOC_NEW »)) {
    NotificationManagerPushOfferCancellationData data =
(NotificationManagerPushOfferCancellationData)cmdData;

    String reasonCode = data.getReasonCode();
    String comment = data.getComment();

    pushOfferCancellation(data.getUserOfferId(), data.getOfferId(), data.getCancelledProducts(),
reasonCode, comment);
    return MessageUtils.createAcknowledgementMessage(null);
}
```

#### **Description de l'interface NotificationManager :**

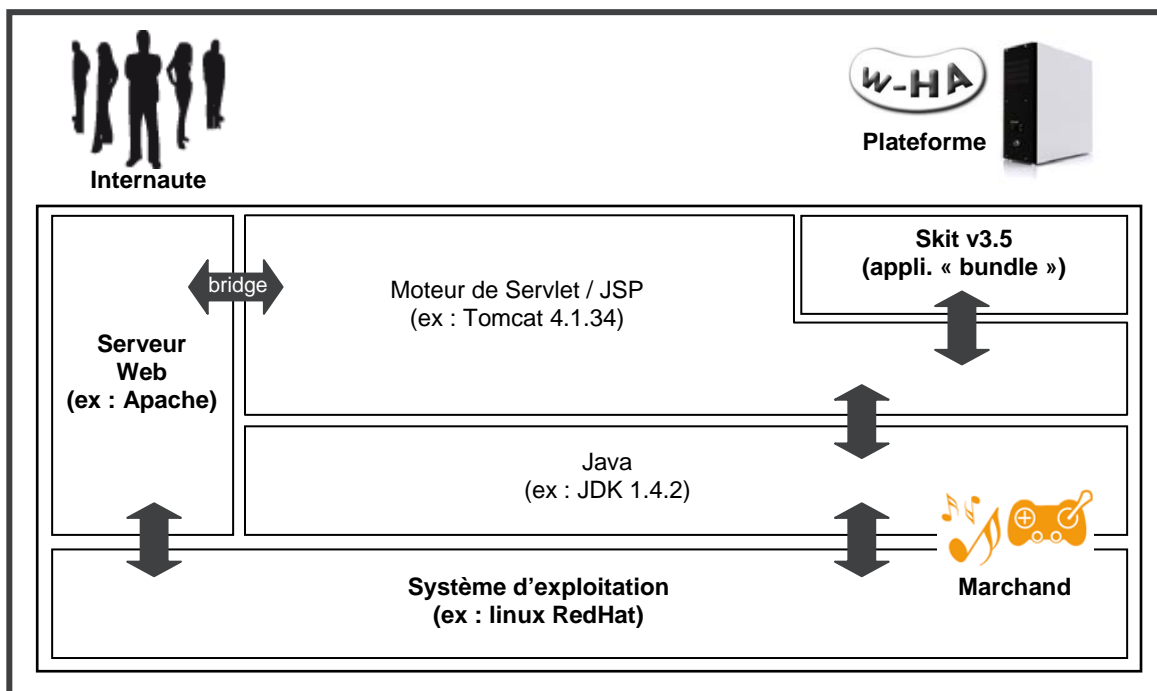
```
package com.valista.core.api.merchant.node ;
import com.valista.util.manager.Manager ;
import com.valista.message.ValistaMessageableException ;
import java.rmi.RemoteException ;
import java.io.Serializable ;

public interface NotificationManager extends Manager {
    public abstract void pushOfferCancellation(String s, String s1,
CancelledProductInfo[]cancelledProductInfos, String s2, String s3) throws RemoteException ;
}
```

## 6 - ANNEXE II : Pré-requis techniques pour le fonctionnement du Kit v3.5

La mise en œuvre de la solution w-HA nécessite l'installation, sur le serveur Web du marchand, d'une application Java « bundle ».

Le schéma suivant illustre les relations entre les différents composants logiciels et matériels nécessaires au fonctionnement de l'application w-HA « bundle ».



Composants nécessaires à l'application « bundle »



Le fonctionnement du système de paiement w-HA nécessite l'installation de l'application « w-HA » (application Java) sur la plate-forme d'hébergement de l'éditeur de service.

## 6.1 - Plate-forme d'hébergement

L'environnement technique (système d'exploitation et serveur web) de la plate-forme d'hébergement du site Web de l'éditeur doit être compatible avec l'application « w-HA ».

Celui-ci repose sur l'utilisation de Servlets Java, ce qui permet une compatibilité avec la majorité des plates-formes du marché.

### 6.1.1 - Système d'exploitation

W-HA garantit le bon fonctionnement du Kit v3.5 et en assure le support technique, pour les environnements décrits ci-dessous.

Si le système d'exploitation est :

**Sun Solaris version 2.6, version 2.7, ou versions supérieures**, w-HA assure le support technique pour les serveurs Web :

- Apache Web Server version 1.3.9 ou versions supérieures (avec support des modules DSO).  
Si Apache n'est pas installé, installer une version 1.3.x récente ou 2.x.

**Linux Kernel version 2.2.x (glibc/libc6 doit être installé)**, w-HA assure le support technique pour les serveurs Web :

- Apache Web Server version 1.3.9 ou versions supérieures (avec support des modules DSO).  
Si Apache n'est pas installé, installer une version 1.3.x récente ou 2.x

**Windows NT version 4.0 (le Service Pack 5 ou version supérieure doit être installé)**, w-HA assure le support technique pour les serveurs Web :

- Apache Web Server version 1.3.9 ou versions supérieures (avec support des modules DSO).  
Si Apache n'est pas installé, installer une version 1.3.x récente ou 2.x
- Internet Information Server version 4.x (IIS4)

**Windows 2000**, w-HA assure le support technique pour les serveurs Web :

- Apache Web Server version 1.3.9 ou versions supérieures (avec support des modules DSO).  
Si Apache n'est pas installé, installer une version 1.3.x récente ou 2.x.
- Internet Information Server version 5.x (IIS5)

**Windows 2003**, w-HA assure le support technique pour les serveurs Web :

- Apache Web Server version 1.3.9 ou versions supérieures (avec support des modules DSO).  
Si Apache n'est pas installé, installer une version 1.3.x récente ou 2.x.
- Internet Information Server version 6.x (IIS6)

**Pour les autres environnements** (autres couples Système d'Exploitation / Serveur Web), le fonctionnement de l'application w-HA est tout à fait envisageable, mais dans ce cas, l'installation de la version J2SDK 1.3.x (ou versions supérieures) de Java et la réalisation du « bridge » entre le serveur Web et le moteur de Servlet seront effectués par le marchand, sous son entière responsabilité.



### **6.1.2 - Machine Virtuelle Java**

Le Java 2 SDK version 1.4.x ou versions supérieures doit être installé, avant ou pendant l'intégration.

**W-HA préconise l'installation de la version « j2sdk1.4.x »** (par ex : j2sdk1.4.2)

Remarque :

Si la plate-forme est de type Linux/UNIX, il faut que les patches requis pour chaque composant soient installés (par exemple, lors de l'installation du j2sdk1.3.x sur Solaris 7, le document d'installation précise que les patches n° xxxxx doivent être installés).

### **6.1.3 - Module de gestion du protocole SSL : « JSSE »**

Le JSSE version 1.0.2 doit être installé, avant ou pendant l'intégration

Remarque :

Le JSSE est intégré par défaut aux J2SDK Java de versions 1.4.x et supérieures. Son installation n'est donc pas nécessaire.

### **6.1.4 - Moteur de Servlet**

Un moteur de Servlets/JSP respectant les spécifications SUN suivantes doit être installé, avant ou pendant l'intégration :

- **JavaServlet 2.2**
- **JavaServlets Pages (JSP) 1.1**

**W-HA recommande le moteur de Servlets/JSP « Tomcat 4.1.x »** (par ex : Tomcat 4.1.34).

Si le moteur de Servlets/JSP est Tomcat, il faut que le serveur Web soit l'un des suivants :

- Apache Web Server version 1.3.9 ou versions supérieures (avec support des modules DSO). Si Apache n'est pas installé, installer une version 1.3.x récente ou 2.x
- Internet Information Server 4 - IIS4
- Internet Information Server 5 - IIS5
- **Pour les autres Serveurs Web, ou si le moteur de Servlet du marchand n'est pas Jakarta-Tomcat**, le fonctionnement de l'application w-HA est tout à fait envisageable, à condition que le moteur de Servlets respecte bien les spécifications SUN énoncées ci-dessus.

Dans ce cas, l'intégration de l'application au sein du moteur de Servlet est à la charge et sous l'entière responsabilité du marchand. W-HA assure le support technique sur le fonctionnement et le paramétrage de l'application w-HA.



## 6.2 - Considérations Réseau

Pour que le Kit v3.5 puisse fonctionner, certains paramétrages des équipements réseaux sont à considérer (ouverture de ports, firewalls, proxies)

### 6.2.1 - Pendant l'installation de l'application w-HA

Il faut que la plate-forme d'hébergement de l'application w-HA :

- Soit **accessible** depuis l'extérieur **par le port du Serveur Web existant** (par défaut : port 80)
- Soit, si possible, accessible depuis l'extérieur par le port du Moteur de Servlet (par défaut : port 8080)
- **Puisse initier une communication SSL (sur le port 443) avec la plate-forme de prod w-HA**
- Les différents équipements réseaux (Firewalls, Proxies, ....) doivent donc, avant l'installation, être correctement configurés, pour permettre les accès précisés ci-dessus.

### 6.2.2 - En production

Il faut que la plate-forme d'hébergement de l'application w-HA :

- **Soit accessible depuis l'extérieur par le port du Serveur Web existant (par défaut : port 80)**
- **Puisse initier une communication SSL (sur le port 443) vers les nœuds :**

Nom de domaine	Opérateur	IP	Port	Type de Requêtes
<a href="https://wanadoo.w-ha.com">https://wanadoo.w-ha.com</a>	Orange Internet	193.28.205.2	443	Sortante
<a href="https://free.w-ha.com">https://free.w-ha.com</a>	Alice/Free	193.28.205.52	443	Sortante
<a href="https://sfr.w-ha.com">https://sfr.w-ha.com</a>	SFR	193.28.205.17	443	Sortante
<a href="http://secure.w-ha.com">http://secure.w-ha.com</a>	Orange Internet, Alice/Free	193.28.205.186	80	Entrante

Les différents équipements réseaux (Firewalls, Proxies, ....) doivent donc être correctement configurés, pour permettre les accès précisés ci-dessus.

#### Remarque :

Sous windows, pour connaître l'adresse IP correspondant à un nom de domaine, ouvrir une fenêtre « invite de commande » (démarrer > exécuter > cmd) et utiliser la commande : nslookup nom de domaine (*Exemple : nslookup wanadoo.w-ha.com*)

## 6.3 - Autres pré-requis pour l'intégration

### 6.3.1 - Utilitaire de décompression

Certains éléments nécessaires à l'installation du Kit v3.5 étant livrés sous la forme d'un fichier « .zip » ou « .tar.gz », un utilitaire de décompression, tel que **winzip, unzip ou tar est nécessaire.**

### 6.3.2 - Redémarrage ("re-boot") du Serveur Web

L'installation de l'application nécessite une modification de la configuration du (des) serveur(s) (serveur Http et Moteur de Servlets). Ces modifications nécessitent un redémarrage du (des) serveur(s), voire de la machine dans certains cas.

### 6.3.3 - Présence de l'administrateur système

Lors de l'installation de l'application et le redémarrage du serveur Web, la présence de l'administrateur système (droits "root") de la plate-forme d'hébergement est nécessaire.(1/2 à 1 journée environ)





## 7 - ANNEXE III : Composants du Kit v3.5

Pour les paiements à l'abonnement, le Kit v3.5 utilise les éléments suivants :

Application « bundle »	
Servlet « pos_bundle »	Gère : <ul style="list-style-type: none"><li>- la demande de souscription à la plate-forme w-HA</li><li>- le paiement à l'acte</li><li>- le contrôle d'accès « simple »</li><li>- le contrôle d'accès « upsell »</li></ul>
Servlet « pos_request »	Gère : <ul style="list-style-type: none"><li>- la confirmation différée de souscription</li><li>- la requête de contrôle d'accès « pull »</li><li>- le remboursement de transaction</li><li>- la résiliation d'abonnement</li><li>- le test du responder</li></ul>
Fichier de configuration « web.xml »	Contient les paramètres globaux et par défaut du Kit v3.5
Fichier de configuration « productbundle_XXX.xml »	Contient le Catalogue Produit
Fichier de configuration « merchants.xml »	Contient les paramètres des différentes boutiques du marchand
Un fichier de log : « log_request_XXX.txt »	Ecrit les informations concernant les « requêtes sortantes » du Kit v3.5
Un fichier de log : « log_response_XXX.txt »	Ecrit les informations concernant les « requêtes entrantes » du Kit v3.5

Application « bundle-responder »	
Servlet « responder »	Gère la réception d'informations en provenance (« push ») de la plate-forme w-HA lors d'événements de type « résiliation »
Fichier de configuration « web.xml »	Contient essentiellement le chemin d'accès au fichier « merchants.xml »
Fichier de configuration « merchants.xml »	Contient les paramètres des différentes boutiques du marchand
Un fichier de log des erreurs : « log_error_XXX.txt »	Ecrit les erreurs sur les événements de type « push »
Un fichier de log des notifications : « log_notification_XXX.txt »	Ecrit les informations sur les événements de type « push »

Application « demo »	
/demo/bundle/html/index.html	Permet de vérifier le bon fonctionnement de la servlet « pos_bundle » pour la souscription à une offre.
/demo/bundle/html/confirm_differe.html	Permet de vérifier le bon fonctionnement de la servlet « pos_request » pour la requête de confirmation différée de souscription
/demo/bundle/html/remboursement_transaction.html	Permet de vérifier le bon fonctionnement de la servlet « pos_request » pour la requête de remboursement de transaction
/demo/bundle/html/resiliation_abonnement.html	Permet de vérifier le bon fonctionnement de la servlet « pos_request » pour la requête de résiliation de souscription

Note : Lors du **paramétrage** des fichiers, **un seul paramètre est autorisé dans la définition des url** des fichiers ".xml" dans les fichiers « bundle » et « bundle-responder ».

Voici la liste des url à paramétrer dans les fichiers ".xml" :

- » dans tomcat/wepapps/bundle/WEB-INF/web.xml :
  - merchantHomeUrl
  - productUnavailableUrl
- » dans tomcat/wepapps/bundle/WEB-INF/productbundle\_5XXX.xml :
  - fulfillmentUrl
- » dans tomcat/wepapps/bundle/WEB-INF/merchant.xml :
  - mcttrxCancelFromPaymentPanelUrl
- » dans tomcat/wepapps/bundle-responder/WEB-INF/merchant.xml :
  - urlRedirect

Si l'url contient 0 paramètre :

(exemple d'url paramétrée dans un fichier ".xml" :

<http://wha.marchand.com/page.php>)

Alors lors de la redirection vers l'url paramétrée, les paramètres supplémentaires seront ajoutés à l'url après un "?" et seront séparés entre eux par des "&".

Si l'url contient 1 paramètre :

(exemple d'url paramétrée dans un fichier ".xml" :

<http://wha.marchand.com/page.php?parameter1=abc>)

Alors lors de la redirection vers l'url paramétrée, les paramètres supplémentaires seront ajoutés à l'url après un "&" et seront séparés entre eux par des "&".

Si l'url contient 2 paramètres ou plus :

(exemple d'url paramétrée dans un fichier ".xml" :

<http://wha.marchand.com/page.php?parameter1=abc&parameter2=cba>)

Alors une erreur 500 se produit (IllegalArgumentException) car le Kit v3.5 n'interprète pas ce type d'url.

Afin d'éviter les erreurs dues au paramétrage w-HA invite les éditeurs à passer les **paramètres supplémentaires** lors de l'appel du kit et **non dans les url paramétrées** dans les fichiers ".xml".



## 7.1 - Servlets « pos\_bundle » et « pos\_request »

### 7.1.1 - Fichier de configuration « web.xml »

#### 7.1.1.1 - Structure du fichier « web.xml »

Paramètre	Commentaires
<b>« pos_bundle »</b>	
merchantId (mctid)	<b>Identifiant de la boutique utilisée par défaut (fourni par w-HA).</b> Il permet à la plate-forme w-HA d'identifier la boutique. Les autres paramètres (mctKeyld, mctKey) relatifs à cette boutique sont présents dans le fichier défini par le paramètre xmlMarchandDatabase.
messageUrl	<b>Chemin d'accès à la JSP gérant les messages d'erreur au format html (ne pas modifier)</b> Les différents cas d'erreur sont envoyés à la JSP « message.jsp », au niveau de l'application « bundle ».
messageWmlUrl	<b>Chemin d'accès à la JSP gérant les messages d'erreur au format wml (ne pas modifier)</b> Les différents cas d'erreur sont envoyés à la JSP « message.jsp », au niveau de l'application « bundle ».
merchantCallbackUrl	<b>URL où se trouve la servlet « pos_bundle » sur le serveur du marchand (à adapter)</b> URL vers laquelle l'utilisateur est redirigé, après avoir accepté la souscription à l'offre, sur le panneau de paiement w-HA. Adapter avec l' <u>adresse IP publique</u> ou le <u>nom de domaine publique</u> du serveur sur lequel l'application « bundle » est installée.
nodePaymentPanelUrl	<b>URL du nœud (plate-forme) w-HA (fourni par w-HA)</b> L'URL du nœud de production est : <a href="https://route.w-ha.com/app-bundlepurchase/node">https://route.w-ha.com/app-bundlepurchase/node</a>
merchantCurrency	<b>Devise du marchand : EUR (ne pas modifier)</b> L'euro est la seule devise acceptée par la plate-forme w-HA La devise de l'éditeur doit donc être paramétré à la valeur EUR. Le montant de l'offre est affiché en euro sur le panneau de paiement w-HA.
xmlAllProductDatabase	<b>Chemin d'accès au fichier « productbundle_XXX.xml » (remplacer XXX par le mctid)</b> Liste des fichiers « productbundle_XXX.xml » contenant le catalogue produit (offres et produits) de chaque boutique de l'éditeur. La liste est de la forme : productbundle_XXX.xml;... ; productbundle_YYY.xml; où les XXX et YYY représentent les merchantId des boutiques de l'éditeur.
xmlMarchandsDatabase	<b>Chemin d'accès au fichier « merchant.xml » (ne pas modifier)</b> Le fichier « merchant.xml » contient les paramètres (mctKeyld, mctKey, ....) des différentes boutiques de l'éditeur.
timestampDelay	<b>Délai maximum pour le « confirm » de l'offre (en millisecondes, ne pas modifier)</b> Délai maximum entre la demande de souscription et la confirmation de souscription

<b>« pos_request »</b>	
merchantId	<b>Identifiant de la boutique utilisée par défaut (fourni par w-HA).</b> Il permet à la plate-forme w-HA d'identifier la boutique. Les autres paramètres (mctKeyld, mctKey) relatifs à cette boutique sont présents dans le fichier défini par le paramètre xmlMarchandDatabase.
messageUrl	<b>Chemin d'accès à la JSP gérant les messages d'erreur au format html (ne pas modifier)</b> Les différents cas d'erreur sont envoyés à la JSP « message.jsp », au niveau de l'application « bundle ».
messageWmlUrl	<b>Chemin d'accès à la JSP gérant les messages d'erreur au format wml (ne pas modifier)</b> Les différents cas d'erreur sont envoyés à la JSP « message.jsp », au niveau de l'application « bundle ».
merchantCallbackUrl	<b>URL où se trouve la servlet « pos_request » sur le serveur du marchand (NON UTILISEE)</b> Adapter avec l' <u>adresse IP publique</u> ou le <u>nom de domaine publique</u> du serveur sur lequel l'application « bundle » est installée.
xmlMarchandsDatabase	<b>Chemin d'accès au fichier « merchant.xml » (ne pas modifier)</b> Le fichier « merchant.xml » contient les paramètres (mctKeyld, mctKey, ....) des différentes boutiques de l'éditeur.
timestampDelay	<b>Délai maximum pour le « confirm » de l'offre (en millisecondes, ne pas modifier)</b> Délai maximum entre la demande de souscription et la confirmation de souscription



### 7.1.1.2 - Exemple de fichier « web.xml »

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN" "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
```

```
<web-app>
```

```
// pos_bundle //
```

```
<servlet>
  <servlet-name>MerchantServlet</servlet-name>
  <servlet-class>com.wha.core.merchant.simulator.MerchantWhaServlet</servlet-class>
  <init-param>
    <param-name>merchantId</param-name>
    <param-value>801</param-value>
  </init-param>
  <init-param>
    <param-name>messageUrl</param-name>
    <param-value>/jsp/message.jsp</param-value>
  </init-param>
  <init-param>
    <param-name>messageWmlUrl</param-name>
    <param-value>/jsp/wmlMessage.jsp</param-value>
  </init-param>
  <init-param>
    <param-name>merchantCallbackUrl</param-name>
    <param-value>http://wha.marchand.com/bundle/pos_bundle</param-value>
  </init-param>
  <init-param>
    <param-name>nodePaymentPanelUrl</param-name>
    <param-value>https://route.w-ha.com/app-bundlepurchase/node</param-value>
  </init-param>
  <init-param>
    <param-name>merchantCurrency</param-name>
    <param-value>EUR</param-value>
  </init-param>
  <init-param>
    <param-name>xmlAllProductDatabase</param-name>
    <param-value>/WEB-INF/productbundle_801.xml;/WEB-INF/productbundle_802;</param-value>
  </init-param>
  <init-param>
    <param-name>xmlMarchandsDatabase</param-name>
    <param-value>/WEB-INF/merchants.xml</param-value>
  </init-param>
  <init-param>
    <param-name>timestampDelay</param-name>
    <param-value>300000</param-value>
  </init-param>
</servlet>
```



## // pos\_request //

```
<servlet>
  <servlet-name>MerchantConfirmWHAServlet</servlet-name>
  <servlet-class>com.wha.core.merchant.simulator.MerchantConfirmWHAServlet</servlet-class>
  <init-param>
    <param-name>merchantId</param-name>
    <param-value>801</param-value>
  </init-param>
  <init-param>
    <param-name>messageUrl</param-name>
    <param-value>/jsp/message.jsp</param-value>
  </init-param>
  <init-param>
    <param-name>messageWmlUrl</param-name>
    <param-value>/jsp/wmlMessage.jsp</param-value>
  </init-param>
  <init-param>
    <param-name>merchantCallbackUrl</param-name>
    <param-value>http://wha.marchand.com/bundle/pos_confirm</param-value>
  </init-param>
  <init-param>
    <param-name>xmlMarchandsDatabase</param-name>
    <param-value>/WEB-INF/merchants.xml</param-value>
  </init-param>
  <init-param>
    <param-name>timestampDelay</param-name>
    <param-value>300000</param-value>
  </init-param>
</servlet>

<servlet>
  <load-on-startup></load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>MerchantServlet</servlet-name>
  <url-pattern>/pos_bundle</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>MerchantConfirmWHAServlet</servlet-name>
  <url-pattern>/pos_request</url-pattern>
</servlet-mapping>

</web-app>
```



## 7.1.2 - Fichier de configuration « productbundle\_XXX.xml »

### 7.1.2.1 - Structure du fichier « productbundle\_XXX.xml »

Paramètre	Commentaires
<b>Produits</b>	
productId	<b>(NON UTILISE)</b> Note : ce champs ne doit pas être vide
fulfillmentUrl	<b>(NON UTILISE)</b> Note : ce champs ne doit pas être vide
purchaseCase	<b>(NON UTILISE)</b> Note : ce champs ne doit pas être vide
autoConfirm	<b>(NON UTILISE)</b> Note : ce champs ne doit pas être vide
<b>Offres</b>	
offerId	<b>Identifiant de l'offre (tel que défini dans le MSCA)</b> Identifiant unique de l'offre, utilisé par la servlet « pos_bundle » pour réaliser la demande de souscription (re-direction)
fulfillmentUrl	<b>URL de prise en compte de la souscription (à adapter)</b> URL (de l'éditeur) vers laquelle l'utilisateur est redirigé, après avoir accepté la souscription de l'offre.
purchaseCase	<b>(NON UTILISE)</b>
autoConfirm	<b>Confirmation d'accès automatique (par défaut : true)</b> - <b>true</b> : la servlet « pos_bundle » confirme automatiquement la souscription puis redirige l'abonné vers la fulfillmentUrl - <b>false</b> : la servlet « pos_bundle » redirige l'abonné vers la fulfillmentUrl. L'éditeur dispose de 24 heures pour confirmer la souscription à la plate-forme w-HA

### 7.1.2.2 - Exemple de fichier « productbundle\_XXX.xml »

```
<?xml version='1.0'?>
<valista-product-database>
  <info>WEB Merchant Product Database</info>

  <product-list>
    <product>
      <productId>P1</productId>
      <fulfillmentUrl>http://www.marchand.com/homepage/acces.php </fulfillmentUrl>
      <purchaseCase></purchaseCase>
      <autoConfirm>true</autoConfirm>
    </product>
  </product-list>

  <offer-list>
    <offer>
      <offerId>O1</offerId>
      <fulfillmentUrl>http://www.marchand.com/abonnement/souscription.php</fulfillmentUrl>
      <purchaseCase></purchaseCase>
      <autoConfirm>true</autoConfirm>
    </offer>
    <offer>
      <offerId>O2</offerId>
      <fulfillmentUrl>http://www.marchand.com/abonnement/souscription.php</fulfillmentUrl>
      <purchaseCase></purchaseCase>
      <autoConfirm>true</autoConfirm>
    </offer>
  </offer-list>

</valista-product-database>
```



### 7.1.3 - Fichier de configuration « merchants.xml »

#### 7.1.3.1 - Structure du fichier « merchants.xml »

Paramètre	Commentaires
<b>mctId</b> (merchantId)	<b>Identifiant de la boutique (fourni par w-HA).</b> Il permet à la plate-forme w-HA d'identifier la boutique.
<b>mctKeyld</b> (keyId)	<b>Identifiant de la clef secrète de la boutique (fourni par w-HA)</b> L'identifiant de la clef est partagé entre le Kit v3.5 et la plate-forme w-HA. Associé à la « mctKey », il permet d'assurer l'intégrité des informations.
<b>mctKey</b> (keyValue)	<b>Valeur de la clef secrète de la boutique (fourni par w-HA)</b> La valeur de la clef est partagée entre le Kit v3.5 et la plate-forme w-HA. Associé au « mctKeyld », elle permet d'assurer l'intégrité des informations.
<b>mcttrxCancelFromPaymentPanelUrl</b>	<b>URL de retour en cas d'annulation depuis le panneau de paiement</b> Si l'internaute clique sur le bouton « Annuler » sur le panneau de paiement, il est redirigé vers cette URL
<b>requestFileName</b>	<b>Chemin d'accès (absolu) au fichier de log des requêtes sortantes</b>
<b>responseFileName</b>	<b>Chemin d'accès (absolu) au fichier de log des requêtes entrantes</b>

#### 7.1.3.2 - Exemple de fichier « merchants.xml »

```
<?xml version='1.0'?>
<valista-key-database>
<info>WEB Merchant key Database</info>
  <mct-list>
    <mct>
      <mctId>801</mctId>
      <mctKeyld>801</mctKeyld>
      <mctKey>Key for 801</mctKey>
      <mcttrxCancelFromPaymentPanelUrl>
http://www.marchand1.com/abonnement/refus\_paiement.php</mcttrxCancelFromPaymentPanelUrl>
      <requestFileName>/usr/local/tomcat/webapps/bundle/WEB-
      INF/log_request_801.txt</requestFileName>
      <responseFileName>/usr/local/tomcat/webapps/bundle/WEB-
      INF/log_response_801.txt</responseFileName>
    </mct>
    <mct>
      <mctId>802</mctId>
      <mctKeyld>802</mctKeyld>
      <mctKey>Key for 802</mctKey>
      <mcttrxCancelFromPaymentPanelUrl>
http://www.marchand2.com/abonnement/refus\_paiement.php</mcttrxCancelFromPaymentPanelUrl>
      <requestFileName>/usr/local/tomcat/webapps/bundle/WEB-
      INF/log_request_802.txt</requestFileName>
      <responseFileName>/usr/local/tomcat/webapps/bundle/WEB-
      INF/log_response_802.txt</responseFileName>
    </mct>
  </mct-list>
</mct-list>
```



### 7.1.4 - Fichier de log « log\_request\_XXX.txt »

Le chemin d'accès au **fichier de log « log\_request\_XXX »** est précisé dans le fichier « merchants.xml » du Kit v3.5.

Il s'agit du **chemin absolu**.

Par défaut : [TOMCAT\_HOME]/webapps/bundle/WEB-INF/log\_request\_XXX.txt

Il s'agit d'un fichier "texte" (\*.txt) qui est **mis à jour (une nouvelle ligne est ajoutée) à chaque nouvelle requête sortante (souscription) déclenchée** par la servlet « pos\_bundle ».

#### **Attention !**

Si de nombreuses requêtes sont réalisées, le fichier risque d'atteindre une taille de plusieurs Méga octets, entraînant une baisse de performance (voire un dysfonctionnement) de l'application.

**Aussi, la mise en place de logs rotatifs est-elle conseillée.**

#### 7.1.4.1 - Structure du fichier « log\_request\_XXX.txt »

Paramètre	Description
Date	Date à laquelle a eu lieu la requête (sortante)
Identifiant (Abonnement)	Au moment de la requête sortante, la plate-forme w-HA n'a pas encore attribué d'identifiant à l'Abonnement. Par conséquent, ce champ a toujours la valeur <b>NULL</b>
Statut (Abonnement)	Au moment de la requête sortante, la plate-forme w-HA n'a pas encore attribué d'identifiant à l'Offre. Par conséquent, ce champ a toujours la valeur <b>3</b>
URL	Les différentes valeurs possibles sont : - Cinématique Offre : URL de la plate-forme w-HA vers laquelle le client doit être redirigé ( <a href="https://opérateur.w-ha.com/app-bundlepurchase/node">https://opérateur.w-ha.com/app-bundlepurchase/node</a> )
Identifiant boutique	Identifiant de la boutique (parmi ceux définis dans le fichier « merchants.xml »)
Identifiant (Offre)	Les différentes valeurs possibles sont : - Cinématique Offre : Identifiant de l'Offre concernée (ex : <b>O1</b> )
Merchant Propertie #1	Paramètre supplémentaire de l'éditeur (#1)
.....	
Merchant Propertie #N	Paramètre supplémentaire de l'éditeur (#N)
Cas d'achat (purchase)	Les différentes valeurs possibles sont : - Souscription à une offre : <b>2</b>

#### 7.1.4.2 - Exemple de fichier « log\_request\_XXX.txt »

Chaque nouvelle demande de souscription entraîne l'ajout d'une nouvelle ligne contenant différents champs séparés par le caractère « pipe » (|) :

```
2008-04-14 14:38:15|NULL|3|https://route.w-ha.com/app-bundlepurchase/node|515|O1|NULL|_ap_userId=abcd|_ap_sessionId=1234|ts=2008-04-14 14:38:15.25|2|
2008-04-14 15:54:26|NULL|3|https://route.w-ha.com/app-bundlepurchase/node|515|O4|NULL|_ap_userId=abcd|_ap_sessionId=1234|ts=2008-04-14 15:54:26.093|2|
```





### 7.1.5 - Fichier de log « log\_response\_XXX.txt »

Le chemin d'accès au **fichier de log** « log\_response\_XXX.txt » est précisé dans le fichier « merchants.xml » du Kit v3.5.

Il s'agit du **chemin absolu**.

Par défaut : [TOMCAT\_HOME]/webapps/bundle/WEB-INF/log\_response\_XXX.txt

Il s'agit d'un fichier "texte" (\*.txt) qui est **mis à jour (une nouvelle ligne est ajoutée) à chaque nouvelle requête entrante (réponse à souscription) est reçue** par la servlet « pos\_bundle ».

#### **Attention !**

Si de nombreuses requêtes sont réalisées, le fichier risque d'atteindre une taille de plusieurs Méga octets, entraînant une baisse de performance (voire un dysfonctionnement) de l'application.

**Aussi, la mise en place de logs rotatifs est-elle conseillée.**

#### 7.1.5.1 - Structure du fichier « log\_response\_XXX.txt »

Paramètre	Description
Date	Date à laquelle a eu lieu la requête (entrante)
Identifiant (de l'Offre)	Les différentes valeurs possibles sont : - Souscription : Identifiant de l'Offre ( <b>6-UXX....XXXX</b> ) - Annulation du client depuis le panneau de paiement ( <b>void</b> )
Statut (de l'Offre)	Les différentes valeurs possibles sont : - Annulé ( <b>Cancel</b> ) - Confirmé ( <b>Confirm</b> ) - En attente de confirmation ( <b>NOT AUTO CONFIRM</b> )
URL	Les différentes valeurs possibles sont : - Annulation du client depuis le panneau de paiement : URL vers laquelle le client doit être redirigé ( <b>trxCancelFromPaymentPanelUrl</b> ) - Offre confirmée : URL vers laquelle doit être envoyée la requête de « confirm » ( <a href="https://opérateur.w-ha.com/app-node-mct/responder">https://opérateur.w-ha.com/app-node-mct/responder</a> )
Identifiant boutique	Identifiant de la boutique (parmi ceux définis dans le fichier « merchants.xml »)
Identifiant (Offre)	Les différentes valeurs possibles sont : - Cinématique Offre : Identifiant de l'Offre concernée (ex : <b>O1</b> )
Merchant Propertie #1	Paramètre supplémentaire de l'éditeur (#1)
.....	
Merchant Propertie #N	Paramètre supplémentaire de l'éditeur (#N)
Cas d'achat (purchase)	Les différentes valeurs possibles sont : - Souscription à une offre : <b>2</b>

#### 7.1.5.2 - Exemple de fichier « log\_response\_XXX.txt »

Chaque nouvelle réponse à une demande de souscription entraîne l'ajout d'une nouvelle ligne contenant différents champs séparés par le caractère « pipe » (|) :

```
2008-04-11
10:30:25|void|Annulation|http://192.168.1.8/demo/bundle/html/panel_cancel1.html|5231|void|_ap_userId=fmoreau
|ts=2008-04-11 10:30:11.671|cur=EUR|amt=0.01|2|2007-10-04 11:17:19|6-2008-04-14 15:14:24|6-
U4142633227858431|Confirm|https://wanadoo.w-ha.com/app-node-
mct/responder|515|O1|_ap_userId=abcd|_ap_sessionId=1234|ts=2008-04-14 15:14:08.656|cur=EUR|amt=0.01|2|
2008-04-14 15:46:13|6-U5117575881274524|NOT AUTO CONFIRM|https://wanadoo.w-ha.com/app-node-
mct/responder|515|O4|_ap_userId=abcd|_ap_sessionId=1234|ts=2008-04-14 15:45:59.515|cur=EUR|amt=0.01|2|
```



## 7.2 - Servlet « responder »

### 7.2.1 - Fichier de configuration « web.xml »

#### 7.2.1.1 - Structure du fichier « web.xml »

La servlet « responder » dispose d'un unique fichier de configuration « web.xml », dont la configuration se limite à quelques paramètres :

Paramètre	Commentaires
xmlMarchandsDatabase	<b>Chemin d'accès au fichier « merchants.xml » (ne pas modifier)</b> Chemin relatif par rapport au répertoire d'installation de la servlet (/bundle-responder)

#### 7.2.1.2 - Exemple de fichier « web.xml »

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
<context-param>
  <param-name>keyProviderClassName</param-name>
  <param-value>com.valista.message.SimpleKeyProvider</param-value>
</context-param>
<servlet>
<servlet-name>Responder</servlet-name>
<servlet-class>com.valista.communication.responder.HttpCommunicationResponder</servlet-class>
<init-param>
  <param-name>xmlMarchandsDataBase</param-name>
  <param-value>/WEB-INF/merchants.xml</param-value>
</init-param>
<init-param>
  <param-name>merchantCurrency</param-name>
  <param-value>EUR</param-value>
</init-param>

<!-- NOTIFICATION_MANAGER_PUSH_OFFER_CANCELLATION_CMD -->
<init-param>
<param-name>NMPOC_NEW</param-name>
<param-value>com.valista.core.api.merchant.node.impl.VoidNotificationManagerCommand</param-value>
</init-param>

<!-- NOTIFICATION_MANAGER_IS_USER_OFFER_AUTHORIZED_FOR_BILLING_CMD -->
<init-param>
<param-name>NMIUOAFB</param-name>
<param-value>com.valista.core.api.merchant.node.impl.VoidNotificationManagerCommand</param-value>
</init-param>

</servlet>
<servlet-mapping>
<servlet-name>Responder</servlet-name>
<url-pattern>/responder</url-pattern>
</servlet-mapping>
</web-app>
```



## 7.2.2 - Fichier de configuration « merchants.xml »

### 7.2.2.1 - Structure du fichier « merchants.xml »

Paramètre	Commentaires
<b>mctId</b> (merchantId)	<b>Identifiant de la boutique (fourni par w-HA).</b> Il permet à la plate-forme w-HA d'identifier la boutique.
<b>mctKeyId</b> (keyId)	<b>Identifiant de la clef secrète de la boutique (fourni par w-HA)</b> L'identifiant de la clef est partagé entre le Kit v3.5 et la plate-forme w-HA.  Associé à la « mctKey », il permet d'assurer l'intégrité des informations.
<b>mctKey</b> (keyValue)	<b>Valeur de la clef secrète de la boutique (fourni par w-HA)</b> La valeur de la clef est partagée entre le Kit v3.5 et la plate-forme w-HA.  Associé au « mctKeyId », elle permet d'assurer l'intégrité des informations.
<b>logErrorFileName</b>	<b>Chemin d'accès (absolu) au fichier de log des erreurs</b>
<b>logFileName</b>	<b>Chemin d'accès (absolu) au fichier de log des notifications</b>
<b>urlRedirect</b>	<b>URL de re-direction des informations de résiliation (OPTIONNEL)</b> URL vers laquelle sont re-dirigées les informations de résiliation

### 7.2.2.2 - Exemple de fichier « merchants.xml »

```
<?xml version='1.0'?>
<valista-key-database>
<info>WEB Merchant key Database</info>

  <mct-list>
    <mct>
      <mctId>801</mctId>
      <mctKeyId>801</mctKeyId>
      <mctKey>Key for 801</mctKey>
      <logErrorFileName>/usr/local/tomcat/webapps/bundle-responder/WEB-INF
      /log_notificationError_801.txt</logErrorFileName>
      <logFileName>/usr/local/tomcat/webapps/bundle-responder/WEB-INF/log_notification_801.txt
      </logFileName>
      <urlRedirect>http://wha.marchand.com/notification_resiliation.php</urlRedirect>
    </mct>
    <mct>
      <mctId>802</mctId>
      <mctKeyId>802</mctKeyId>
      <mctKey>Key for 802</mctKey>
      <logErrorFileName>/usr/local/tomcat/webapps/bundle-responder/WEB-INF
      /log_notificationError_802.txt</logErrorFileName>
      <logFileName>/usr/local/tomcat/webapps/bundle-responder/WEB-INF/log_notification_802.txt
      </logFileName>
      <urlRedirect >http://wha.marchand.com/notification_resiliation.php</urlRedirect >
    </mct>
  </mct-list>
```



### **7.2.3 - Fichier de log "log\_notification\_XXX.txt"**

#### **7.2.3.1 - Structure du fichier de log « log\_notification\_XXX.txt »**

Le fichier « log\_notification\_XXX.txt » permet de loguer les informations en entrée de la servlet « responder » :

<b>Paramètre</b>	<b>Description</b>
<b>Date</b>	Date et heure de réception de la notification de résiliation
<b>uo = &lt;user offer id&gt;</b>	Identifiant de l'abonnement (uoid)
<b>r = &lt;reason&gt;</b>	Code de la résiliation : cf. § 2.3.5.3
<b>P = &lt;product list&gt;</b>	Liste des produits, de la forme @<mctid>@<pid> @<mctid>@<pid> ...
<b>o = &lt;external offer id&gt;</b>	Identifiant de l'offre (oid)
<b>c = &lt;comment&gt;</b>	Description du motif de la résiliation

#### **7.2.3.2 - Exemple de fichier de log « log\_notification\_XXX.txt » :**

```
Fri Oct 05 19:25:03 2007: uo=6-U6141335211845451;r=200;p=@515@P1;o=O2;c=05/10/07 17:20:46
Sat Oct 06 15:59:16 2007: uo=6-U4643143178635305;r=101;p=@515@P1;o=O2;c=User billable timeout
Sat Oct 06 16:59:17 2007: uo=6-U4643143178635305;r=101;p=@515@P1;o=O2;c=User billable timeout
Sat Oct 06 17:59:19 2007: uo=6-U4643143178635305;r=101;p=@515@P1;o=O2;c=User billable timeout
Sat Oct 06 18:59:21 2007: uo=6-U4643143178635305;r=101;p=@515@P1;o=O2;c=User billable timeout
Mon Oct 08 15:15:38 2007: uo=6-U6141344148455402;r=107;p=@515@P1;o=O1;c=08/10/07 15:09:52
```