Descriptif technique du Kit v3.5 (Acte) w-HA pour Internet+

Résumé :	Ce document est destiné aux sociétés qui souhaitent vendre sur Internet des produits ou services (biens immatériels uniquement) en utilisant le système de paiement w-HA. Il traite des aspects suivants : » Principe de fonctionnement du système w-HA » Description technique du système w-HA (côté marchand) » Mise en oeuvre du système de paiement w-HA par le marchand

VALIDATION

	Nom	Département	Date
Validation	DIAKONOFF	Relation Client	28/08/2009

HISTORIQUE DU DOCUMENT

Version	Date	Modifications
1.10	28/08/2009	Homogénéisation doc acte & abo
1.11	19/10/2009	Mise à jour des nœuds w-HA + Corrections §2.2.1

DOCUMENTS DE REFERENCE

Titre	Typologie

TABLE DES MATIERES

1. PRINCIPE DE FONCTIONNEMENT DU SYSTEME 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 établit un partenariat avec w-HA.

Il peut s'agir :
» d'un Fournisseur d'Accès à Internet (FAI) :
Ex : Orange, AOL, Club-Internet, Alice, Neuf-Cegetel, 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.

Internaute
L'Internaute désigne le client final, qui achète, sur Internet, un bien téléchargeable ou un service, en utilisant la solution w-HA.

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'achat (temps réel) vu du client

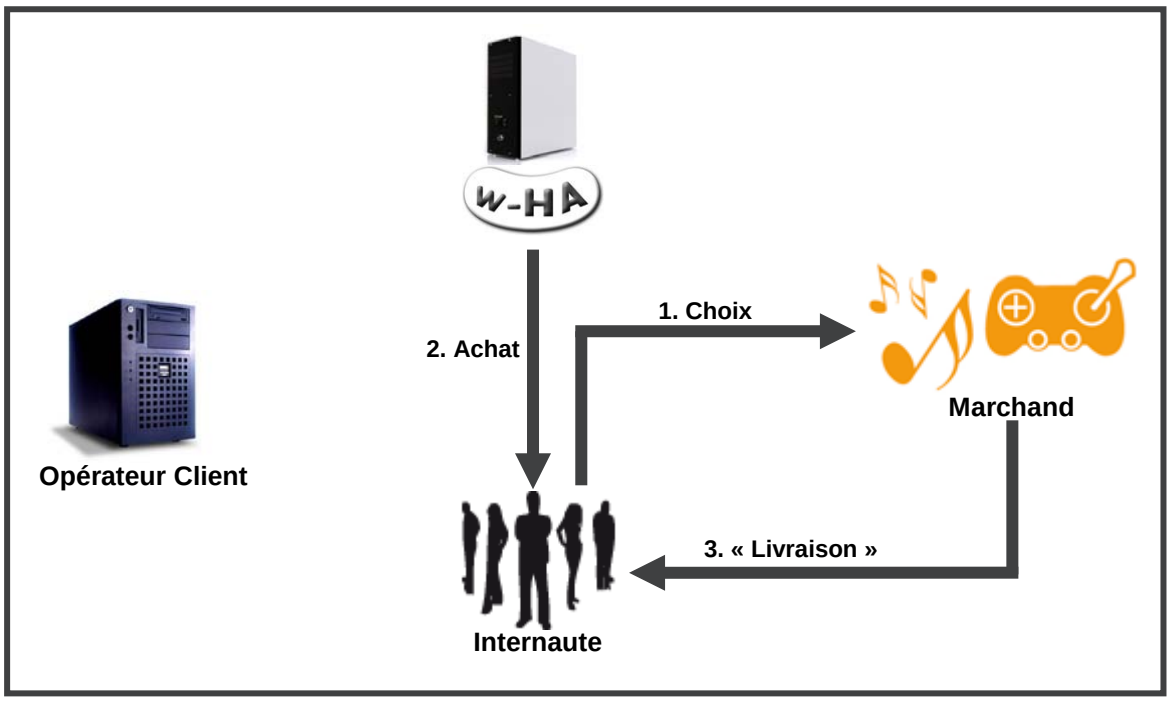


figure 1 : processus d'achat (vu de l'Internaute)

L'achat vu de l'Internaute (en temps réel) est le suivant :

1. Ecran "marchand" : L'Internaute choisit un produit (ou un service) 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 achat

3. Ecran "marchand" : Après autorisation de w-HA, l'Internaute est redirigé vers la page Web du marchand pour obtenir le produit (ou le service) acheté.

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)

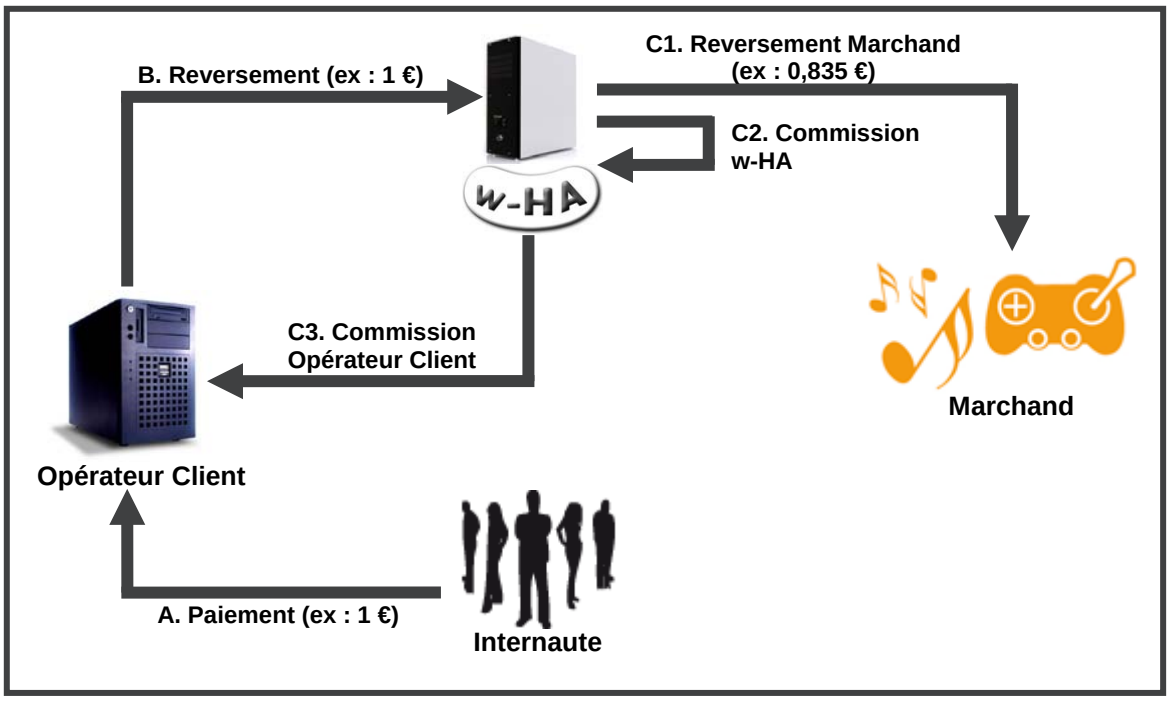


figure 2 : flux financiers entre les différents acteurs de la transaction

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 :

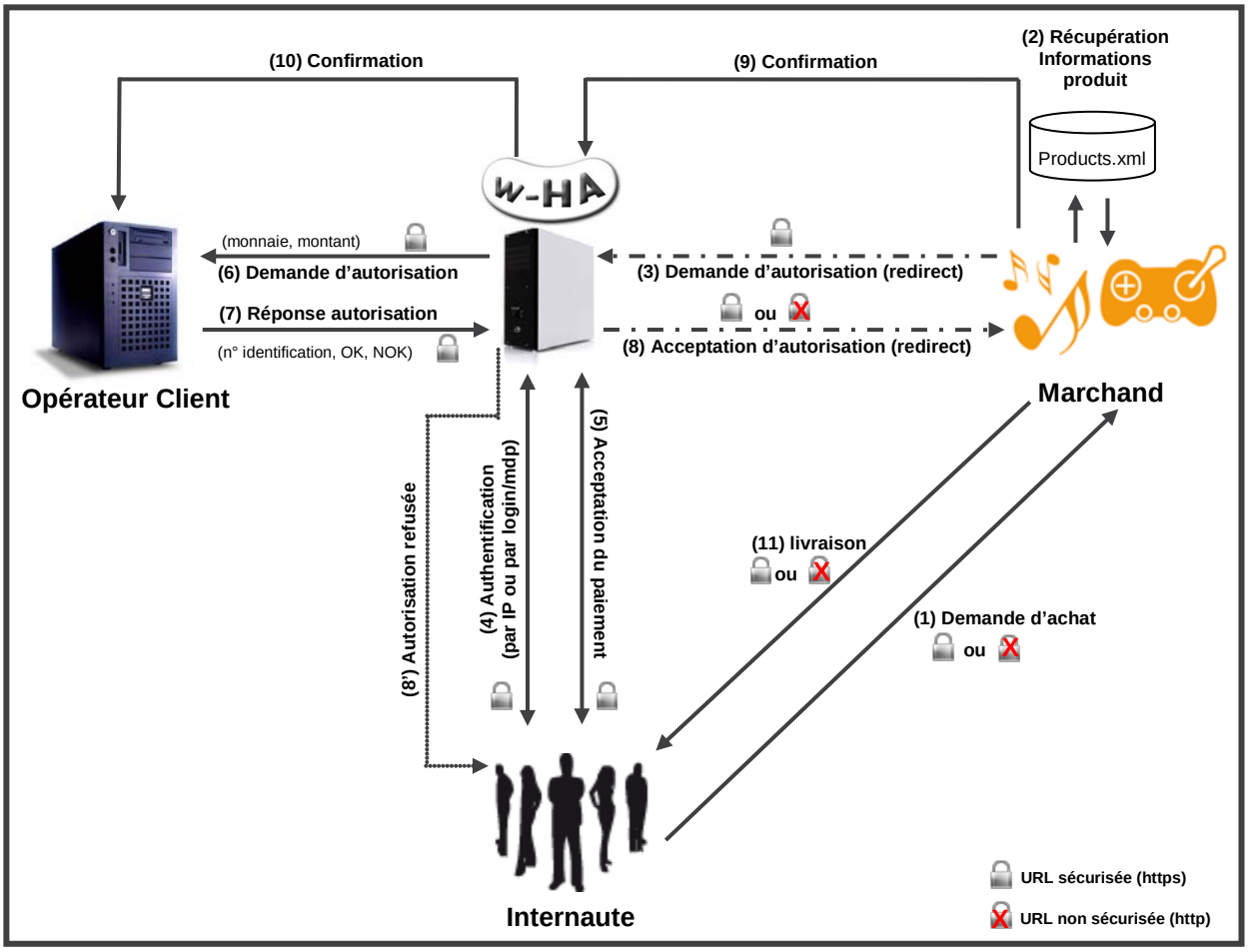


figure 3 : détail des flux d'information entre les différents acteurs de la transaction

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 du produit
- montant du produit

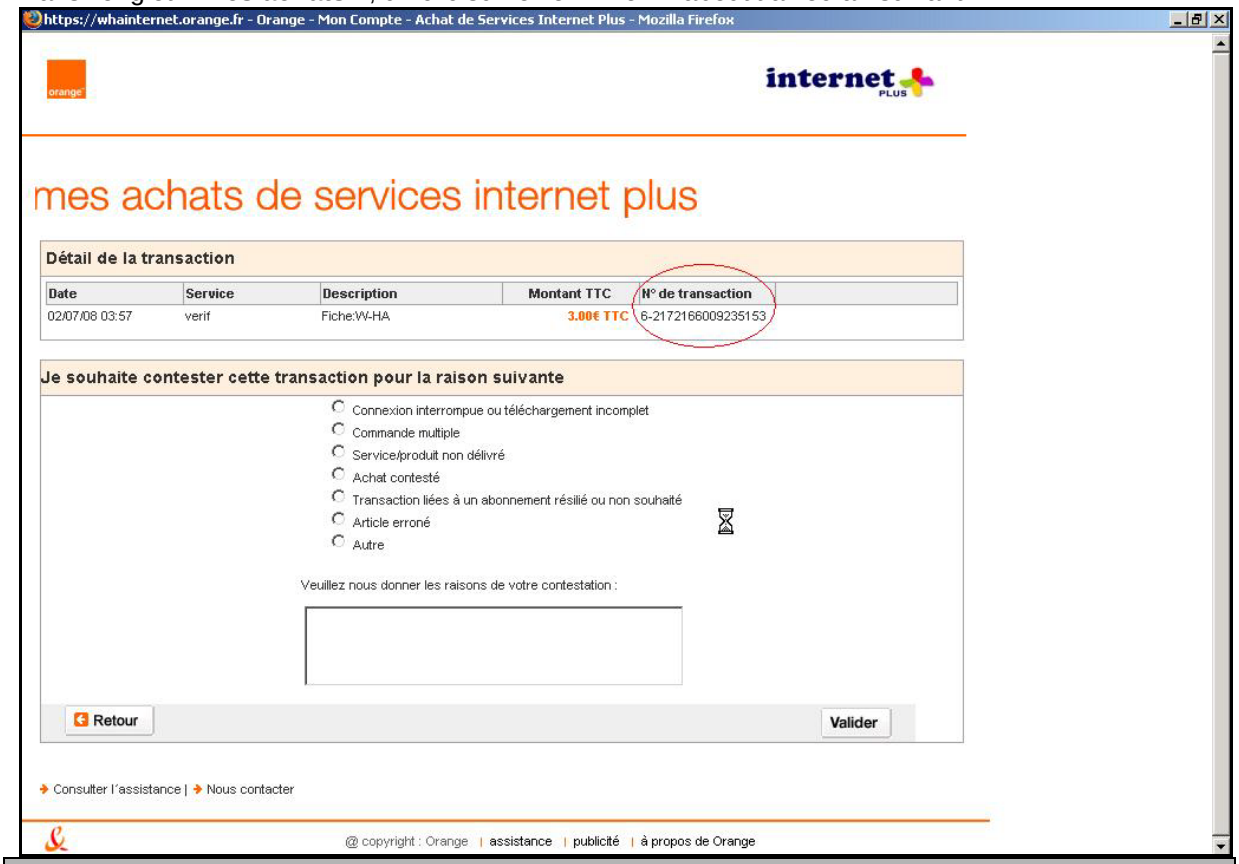
L'internaute n'est qu'à **1 seul clic de son achat** (bouton de confirmation d'achat), ce qui fait de w-HA l'une des solutions de paiement procurant **le meilleur taux de transformation du marché !**

Exemple : panneau de paiement Orange

1.4.4. Remboursement d'une transaction

Un Internaute Orange Internet (par exemple) peut **demander, en ligne, le remboursement** d'une transaction, depuis le portail 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 :



Remarque : un **identifiant de transaction** (trxid) Orange est de la forme 6-XXXXXXXXXXXXXXX

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

2. DETAIL DU FONCTIONNEMENT DES DIFFERENTES FONCTIONNALITES / SERVLETS

2.1. Servlet avec redirection de l'internaute vers la plate-forme w-HA

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

Attention !
La servlet « **pos_init** » doit être **accessible depuis l'internet public (port 80)**.

2.1.1. Servlet de demande d'autorisation d'achat / « pos_init ? action=authorize »

2.1.1.1 Appel de la Servlet

Cette Servlet est appelée au niveau des pages Web de présentation des produits/services mis en vente par le marchand avec le système de paiement w-HA.

Lorsque l'éditeur souhaite déclencher un **achat** d'un utilisateur, il fait appel à la servlet « **pos_init** », avec le paramètre « **action** » égal à « **authorize** ».

2.1.1.2 Exemple d'appel de la Servlet

http://*wha.marchand.com*/bundle/pos_init?
action=authorize
&pid=P1
&wha_desc2=current
&ParametresSupplementaires=abc

En rouge : paramètres obligatoires
En bleu : paramètres optionnels (merchant properties)

2.1.1.3 Paramètres d'appel de la servlet

Pour le déclenchement d'un achat, la servlet « pos_init » est appelée avec les paramètres suivants :

Paramètres toujours obligatoires :

action=authorize : pour indiquer à la servlet qu'il s'agit d'une demande d'achat à l'acte

pid=*Identifiant du produit*

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

Les paramètres supplémentaires (appelés aussi « **merchant properties** » ou « **mp** ») transmis à la servlet pos_init sont **récupérés à l'issue de la phase de paiement**, au niveau de l'URL de livraison (**fulfillmentUrl**).

L'éditeur peut alors faire passer, par exemple, un identifiant de partenaire (origine de l'achat), un identifiant d'utilisateur (profiling des achats), un identifiant de commande (suivi des commandes), un identifiant de session (utilisé pour la protection de l'Url de livraison), etc.

Attention !
Le paramètre « **wha_desc2=current** » est obligatoire si le panneau de paiement est affiché en page courante.

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

2.1.1.4 Fonctionnement interne de la servlet

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

- récupère dans le fichier « web.xml » les informations relatives à la boutique (merchantId, URLs, mctKeyId, mctKey …..)
- récupère dans le fichier « products.xml » les informations relatives au produit (productId ; ….)
- 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.

2.1.1.5 Exemple d'URL générée par la servlet

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

```
https%3A%2F%2Froute.w-ha.com%2Fapp-
authorization%2Fnode%3Fm%3Dh%3D8c2a104518b1252f92eef510545a8fa7%3Bp%3D515%3Bk%
3D515%3Bv%3D2%3A%7Bc%3DAuthorizeReq%3Bv%3D%7Bps%3D2%3Bamt%3D0%3Bmp%3D%
7Bpid%3DP1%3B_ap_wha_desc2%3Dcurrent%3Bts%3D2009-08-
25%2011%3A50%3A32.921%3B_ap_ParametresSupplementaires%3Dabc%3B%7D%3Bpi%3DP1%
3Bpg%3D0%3BmUrl%3Dhttp%3A%2F%2Flocalhost%3A8080%2Fbundle%2Fpos_init%3Bpd%3DPr
oduit%201%3Bpc%3DImage%3Bhr%3D1%3Bcur%3DEUR%3Bcl%3D-
1%3B%7D%7D%26lg%3Dfr%0A
```

Pour plus de lisibilité, en URL décodée :

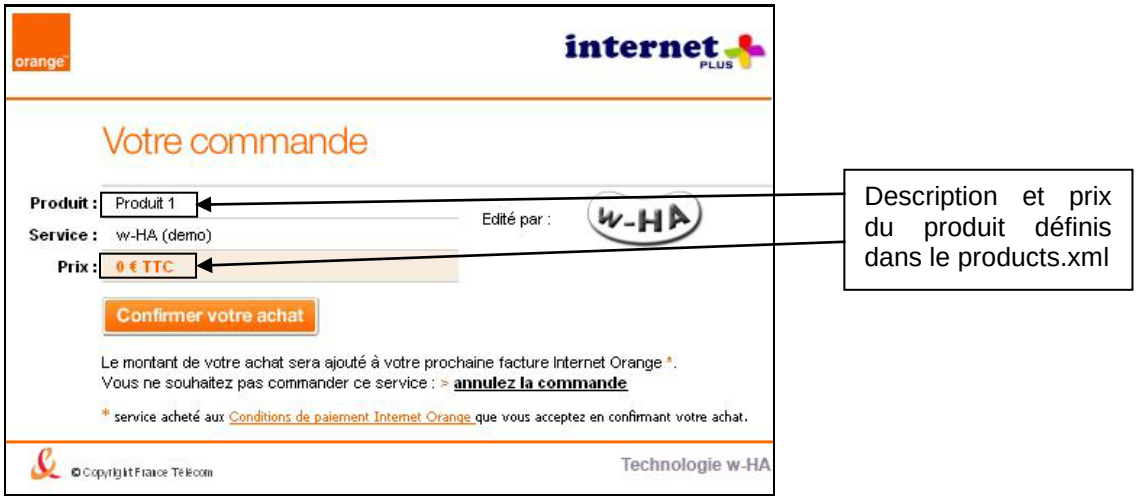
```
https://route.w-ha.com/app-authorization/node?
m=h=8c2a104518b1252f92eef510545a8fa7;
p=5XXX;
k=5XXX;
v=2:{
c=AuthorizeReq;
v={
ps=2;
amt=0;
mp={pid=P1;_ap_wha_desc2=current;ts=2009-08-25
11:50:32.921;_ap_ParametresSupplementaires=abc;};
pi=P1;
pg=0;
mUrl=http://wha.marchand.com/bundle/pos_init;
pd=Produit 1;
pc=Image;
hr=1;
cur=EUR;
cl=-1;}}&lg=fr
```

Cette URL est sécurisée (https) et signée.

En fait, il s'agit, d'une re-direction (directive "redirect") de l'internaute, via son navigateur Web., vers la plate-forme w-HA,

2.1.1.6 Réponse de w-HA à la servlet

Lorsque l'utilisateur est redirigé vers la plate-forme w-HA pour une demande d'achat à l'acte (cf. paragraphe précédent), celle-ci lui affiche un « panneau de paiement ».



panneau de paiement Orange

L'utilisateur peut soit « confirmer son achat », soit « annuler la commande ».

1er cas : annulation de la commande

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

- L'utilisateur est redirigé vers la servlet « pos_init » de l'éditeur (merchantUrl), avec un message «c=AuthorizeCancel »

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

```
http://localhost:8080/bundle/pos_init?m=h%3D6c9a10d076c5c141f334469704b1545c%3Bp%3D515
%3Bk%3D515%3Bv%3D3%3A%7Bc%3DAuthorizeCancel%3Bv%3D%7Bpid%3DP1%3B_ap_wha_d
esc2%3Dcurrent%3Bts%3D2009-08-
25+13%3A11%3A23.796%3B_ap_ParametresSupplementaires%3Dabc%3B%7D%7D
```

Pour plus de lisibilité, en URL décodée :

```
http://wha.marchand.com/bundle/pos_init?
m=h=6c9a10d076c5c141f334469704b1545c;
p=5XXX;
k=5XXX;
v=3:{
c=AuthorizeCancel;
v={pid=P1;_ap_wha_desc2=current;ts=2009-08-25
13:11:23.796;_ap_ParametresSupplementaires=abc;}}
```

Cette URL est signée.

Les informations contenues dans cette URL sont :
» un hmac (h=), pour assurer l'intégrité du message
» le message « AuthorizeCancel »
» les paramètres supplémentaires de l'éditeur (avec le préfixe _ap_)

- Puis, la servlet « pos_init » re-dirige l'utilisateur vers la trxCancelFromPaymentPanelUrl, paramétrée dans le fichier « web.xml » :

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

```
http://wha.marchand.com/demo/bundle/html/panel_cancel.html?
ParametresSupplementaires=abc
&wha_desc2=current
&hmac=9b79ef06de7305d54fc97986fb05554f
```

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

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

Si l'utilisateur clique sur le bouton « Confirmer votre achat », mais que la transaction 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 transaction.

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_init », pour préciser la cause de l'erreur.

3ème cas : acceptation de la transaction par l'utilisateur :

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

- L'utilisateur est redirigé vers la servlet « pos_init » de l'éditeur (merchantUrl), avec un message « c=AuthorizeSuccess ».

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

http%3A%2F%2Flocalhost%3A8080%2Fbundle%2Fpos_init%3Fm%3Dh%3Dc5b6f2ea0b77196357ab6c5ab5fe07c3%3Bp%3D515%3Bk%3D515%3Bv%3D3%3A%7Bc%3DAuthorizeSuccess%3Bv%3D%7Bst%3DFR%3Bv%3D86400%3Bmp%3D%7Bpid%3DP1%3B_ap_wha_desc2%3Dcurrent%3Bts%3D2009-08-25%2011%3A50%3A32.921%3B_ap_ParametresSupplementaires%3Dabc%3B%7D%3BtId%3D6-5150049942659815%3Bz%3D75000%3Bci%3DParis%3Brt%3Dhttps%3A%2F%2Fwanadoo.w-ha.com%2Fapp-node-mct%2Fresponder%3Bco%3DFR%3B%7D%7D%0A

Pour plus de lisibilité, en URL décodée :

http://*wha.marchand.com*/bundle/pos_init?
m=h=c5b6f2ea0b77196357ab6c5ab5fe07c3;
p=5XXX;
k=5XXX;
v=3:{
c=AuthorizeSuccess;
v={
st=FR;
v=86400;
mp={pid=P1;_ap_wha_desc2=current;ts=2009-08-25
11:50:32.921;_ap_ParametresSupplementaires=abc;};
tId=6-5150049942659815;
z=75000;
ci=Paris;
rt=https://wanadoo.w-ha.com/app-node-mct/responder;
co=FR;}}

Cette URL est signée.

En fait, il s'agit, d'une re-direction (directive "redirect") de l'internaute, via son navigateur Web., vers la plate-forme w-HA,

Les informations contenues dans cette URL sont :
- un HMAC (Message Authentification Code), pour assurer l'intégrité du message
- le message « c=AuthorizeSuccess »
- l'ensemble des informations du produit acheté par l'internaute,
- des informations relatives à l'origine géographique de l'internaute.
- les merchant properties (paramètres ajoutés par le marchand + l'identifiant tId)
- l'URL pour confirmer la transaction (rt=)

Le rôle de la Servlet "pos_init" est de :
- Vérifier le message d'autorisation de transaction retournée par la plate-forme w-HA,
- Déclencher la requête de confirmation du produit/service vendu à la plate-forme w-HA

2.1.1.7 Requête de confirmation de livraison

Lorsque la Servlet « pos_init » reçoit une réponse positive à la demande d'autorisation de la plate-forme w-HA, elle déclenche **automatiquement** une confirmation de livraison à cette dernière.

Exemple de requête de confirmation envoyée par la Servlet "pos_init" :

https://wanadoo.w-ha.com/app-node-mct/responder?m=h%3D33b3c2b73eb7d47944fa29cffa6d94f1%3B%0Ap%3D505%3B%0Ak%3D505%3B%0Av%3D2%3A%0A%7Bc%3DConfirm%3B%0Av%3D%7B%0As_rate%3D0%3B%0An_amt%3D1.00%3B%0Ag_amt%3D1.00%3B%0Av_amt%3D0%3B%0AtrxId%3D6-5150049942659815%3B%0Av_rate%3D0%3B%0As_amt%3D0%3B%0Acur%3DEUR%3B%7D%7D

Pour plus de lisibilité, en URL décodée :

https://wanadoo.w-ha.com/app-node-mct/responder?
m=h=33b3c2b73eb7d47944fa29cffa6d94f1;
p=5XXX;
k=5XXX;
v=2:{
c=Confirm;
v={
s_rate=0;
n_amt=0;
g_amt=0;
v_amt=0;
trxId=6-5150049942659815;
v_rate=0;
s_amt=0;
cur=EUR;}}

Cette URL est sécurisée (https) et signé.

Il s'agit, cette fois, d'une communication directe (en **https**) du serveur Web du marchand vers la plate-forme de paiement w-HA. Aussi, il est important de **configurer** les éventuels **proxys** et **firewalls** pour que la communication https puisse être réalisée.

Les informations contenues dans cette URL sont :
- la commande de confirmation de débit (« c=Confirm »)
- l'identifiant de la transaction à confirmer (trxId)
- le montant à débiter (n_amt, g_amt)

Le rôle de cette étape (requête de confirmation) est de :
- Confirmer le débit (déclenchement du processus de facturation).
- Rediriger l'internaute vers l'URL finale à laquelle se trouve le produit/service acheté
- Mettre à jour le fichier "journal des événements" (authorizations.txt),
- Mettre à jour le fichier "journal des transactions" (logs.txt)

Attention :
La « **confirmation** » de débit **précède** la **livraison** du produit vendu.

Si la « confirmation » se déroule **correctement** (la servlet reçoit un « ack ») :
- le **débit** de la transaction devient **effectif**,
- l'internaute est redirigé vers l'Url de livraison (**fulfillmentUrl**)

Si la « confirmation » ne se déroule **pas correctement** :
- la transaction n'est **pas débitée**
- l'internaute est redirigé vers la « **messageUrl** »

2.1.1.8 Livraison du produit/service acheté par l'Internaute

La livraison du produit acheté par l'internaute est déclenchée après que le serveur Web du marchand (via la sevlet pos_init) ait envoyé (automatiquement) un « c=Confirm » de débit à la plate-forme w-HA.

La Servlet « pos_init » de l'application redirige alors l'internaute vers l'URL de livraison (ffUrl).

Exemple d'URL finale d'accès au produit/service acheté par l'internaute (URL dynamique) :

```
http(s)//wha.marchand.com/...../cgi-bin/paiement_ok.cgi?
ParametresSupplementaires=abc
&wha_desc2=current
&hmac=9b79ef06de7305d54fc97986fb05554f
&trxId=30-5150049942659815
```

Cette URL est sécurisée (https) et signé.

Cette URL est un programme (servlet, php, asp, coldfusion, …..) qui reçoit (méthode GET) les paramètres suivants :

- Un hmac, permettant de vérifier l'intégrité des « mp »
- Les « merchants properties » (mp) : paramètres supplémentaires du marchand
- l'identifiant de transaction trxId,

2.2. Servlet pour les requêtes de serveur à serveur

La fonctionnalité suivante est gérée via des requêtes httpS de serveur à serveur :

- Remboursement d'une transaction (servlet « pos_request » : EdS → w-HA)

Attention !

La servlet « **pos_request** » gérant cette fonctionnalité **NE doit 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**.

2.2.1. 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.

2.2.1.1 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.

2.2.1.2 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=5XXX
&trxid=6-17141248844587211
&url=https://wanadoo.w-ha.com/app-node-mct/responder
&userComment=commentaire client
&adminComment=commentaire editeur
&reasonCode=110

2.2.1.3 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 souscris par erreur cet abonnement plusieurs fois

2.2.1.4 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 (URLs…)
- récupère dans le fichier « merchant.xml » les informations relatives à la boutique (merchantId, keyValue,…)
- 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.

2.2.1.5 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%3BtrxId%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;
trxId=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=)

2.2.1.6 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
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;}}	Echec du remboursement
e=3	Mauvais hmac
e=15	Erreur dans les paramètres transmis (exemple : "trx**i**d=" au lieu de "trx**I**d=")

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;}}

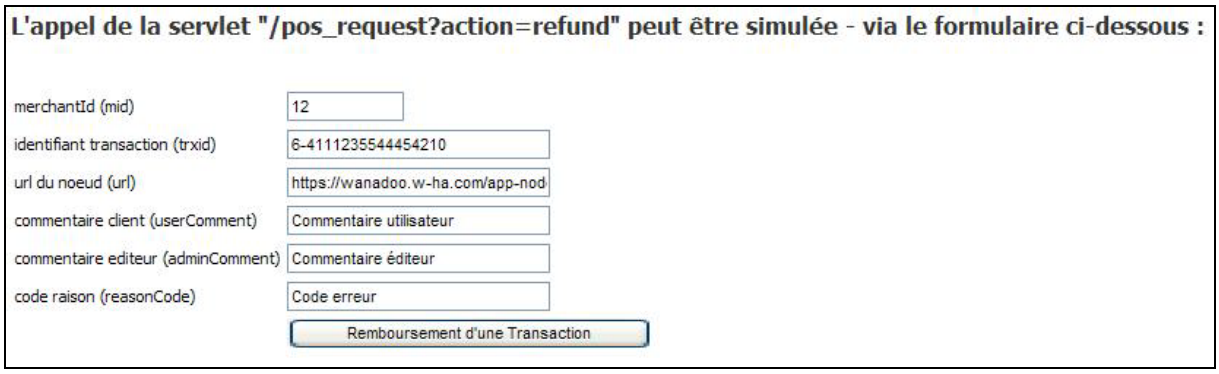
Remarque : Si on essaye de **rembourser plusieurs fois la même transaction**, la plate-forme retournera pour la **première** demande de remboursement, par exemple, "**trxid=RR1234567891234567**", mais pour les **suivantes**, elle retournera "**h=a0e198418d3a05ed657c1cfce78c0964;p=5XXX;k=5XXX;v=2:{c=ex;v={m=INVALID_MERCHANT_INFO;t=com.ipin.core.api.node.pos.Merch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

2.2.1.7 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 (*) :
(*) remplacer wha.marhand.com par l'adresse IP ou le nom de domaine public du serveur sur lequel est installée l'application w-HA



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. CE QUE LE MARCHAND DOIT METTRE EN ŒUVRE POUR UTILISER LE KIT V3.5

L'intégration du système de paiement w-HA au sein du site Web du marchand ne nécessite pas de développements informatiques complexes ni de connaissance de JAVA.

Les actions à mener par le marchand sont plutôt de l'ordre du paramétrage et de la création de pages Web. Seul un développement minime est nécessaire pour la protection de contenu.

3.1. Paramétrage des fichiers de configuration « *.xml »

web.xml
Le fichier "web.xml" doit, à priori, être paramétré une seule fois par le marchand, avant le passage en production.

products.xml
Le fichier "products.xml", quant à lui, sera modifié par le marchand à chaque ajout, modification ou suppression de produits.

Attention :

A chaque **modification** réalisée dans les fichiers "web.xml" ou "products.xml" (ou éventuellement « server.xml »), pour que celle-ci soit prise en compte, il faut **arrêter et redémarrer le Moteur de Servlet** !

3.2. Réalisation de pages Web

Des pages html doivent être réalisées par le marchand, pour informer l'internaute lorsque des événements particuliers se produisent.

D'autres pages doivent être réalisées pour présenter les produits/services du marchand et faire le "lien" avec l'application "acte"

Il est possible de gérer des événements qui peuvent se produire dans certains cas prévus par l'application.

Par exemple, lorsque l'utilisateur annule son achat au lieu de l'accepter (depuis le panneau de paiement), ou si le produit n'est pas présent dans le fichier "products.xml".

Lorsque ces événements se produisent, l'internaute est redirigé vers les URLs paramétrées dans les champs suivants du fichier "web.xml" :

<param-name>**merchantHomeUrl**</param-name>
<param-name>**trxCancelFromPaymentPanelUrl**</param-name>
<param-name>**productUnavailableUrl**</param-name>

Le marchand doit donc réaliser une page particulière, aux couleurs de son propre site, pour chacun des paramètres ci-dessus, présents dans le fichier "web.xml".

3.3. 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_init) est le suivant :

```
<a href="/bundle/pos_init?action=authorize&pid=P1&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_init?action=authorize&pid=P1&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-authorization/node?m=h=1ad42bde9f9812b062a2d11b79c00e83;p=5XXX;k=5XXX;v=2:{c=AuthorizeReq;v={ps=2;amt=0;mp={_ap_userId=fmoreau;pid=P1;**_ap_wha_desc2=current**;};pi=P1;pg=0;mUrl=http://localhost:8080/bundle/pos_init;pd=Produit#1;pc=Image;hr=1;cur=EUR;cl=-1;}}

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/prodcuts.xml)
- la **trxCancelFromPaymentPanelUrl** (url paramétrée dans tomcat/webapps/bundle/WEB-INF/web.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 > boîte à outils

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

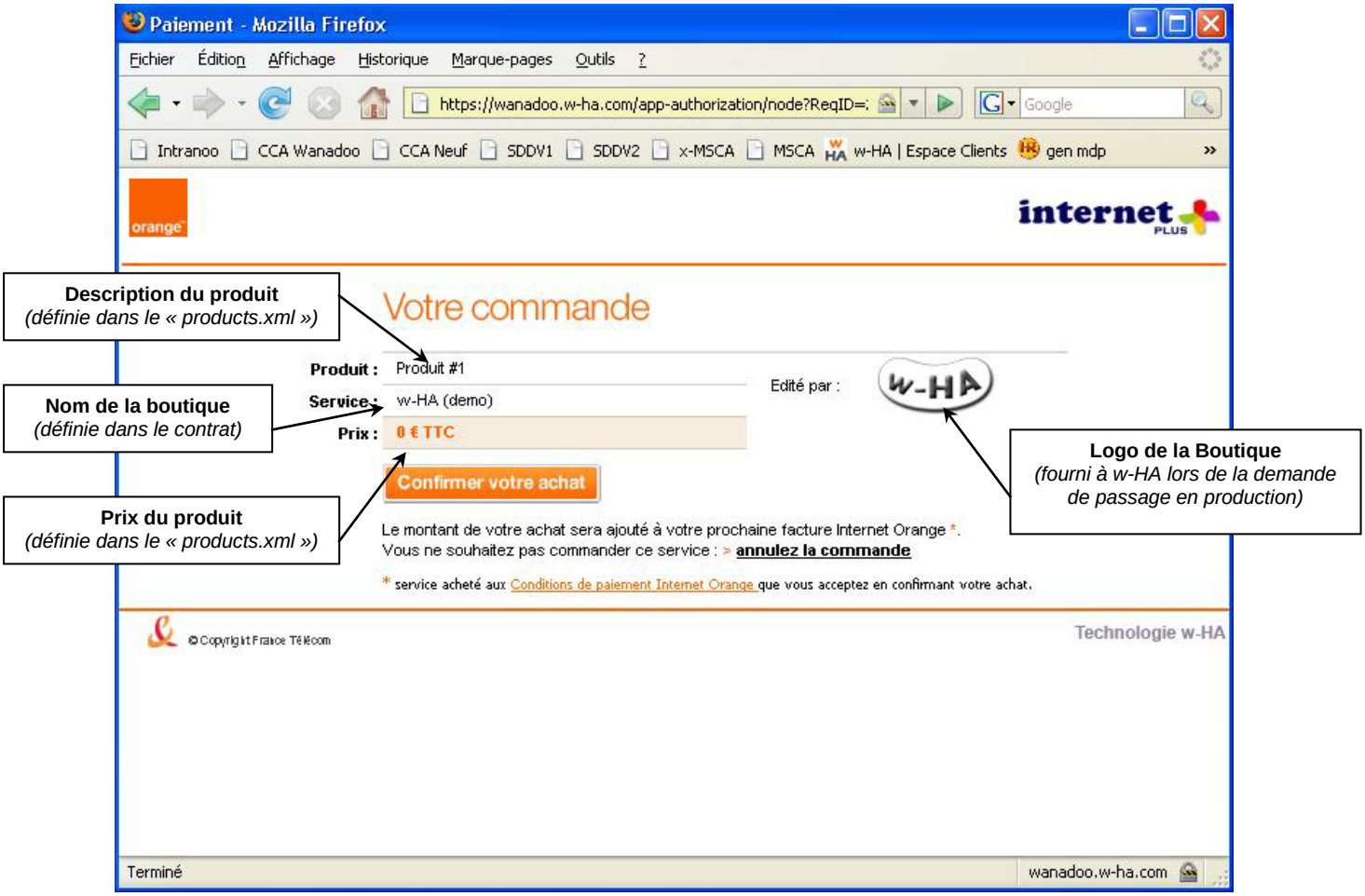


figure : affichage du « panneau de paiement » Orange

3.4. Utilisation du H-MAC pour la protection de contenu

Toutes les informations échangées entre la plate-forme du marchand et celle de w-HA sont **signées** (HMAC-MD5 puis SHA1), **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 Authentification 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 (par ex : un 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 "KeyValue" du fichier "web.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&commandeId=abcd&userId=toto&hmac=891284e23faa662c033a41dd9905cc10&trxId=6-7672821718212150

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 "KeyValue" est " a1b2c3d4e5f ", alors le calcul du H-MAC se fait de la manière suivante :

mon-hmac = H-MAC ("**commandeId=abcd&sessionId=1234&trxId=6-7672821718212150&userId=toto","a1b2c3d4e5f**")

Attention ! (1)
De la même façon que pour les paramètres "description" et "category" du fichier "products.xml", les **paramètres supplémentaires (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 la transaction (trxId)**, récupéré au niveau de la fulfillmentUrl, **NE doit PAS 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?
sessionId=1234&commandeId=abcd&userId=toto
&hmac=891284e23faa 662c033a41dd9905cc10&trxId=6-7672821718212150

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

```
$hmac_verif =
bin2hex(mhash(MHASH_MD5,"commandeId=abcd&sessionId=1234&userId=toto","a1b2c3d4e5f");
        if ($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), nous vous conseillons de **vérifier que la livraison liée à cette transaction n'a pas déjà été effectuée**.

Pour ce faire, vous pouvez utiliser l'identifiant de transaction (trxid), qui est unique, ou un paramètre supplémentaire (lors de l'appel de la servlet de demande d'autorisation d'achat, cf § 2.1.1.1 - Appel de la Servlet) de votre choix (par exemple, une valeur que vous incrémentez à chaque transaction) afin de repérer l'unicité de l'URL de livraison (fulfillment URL).

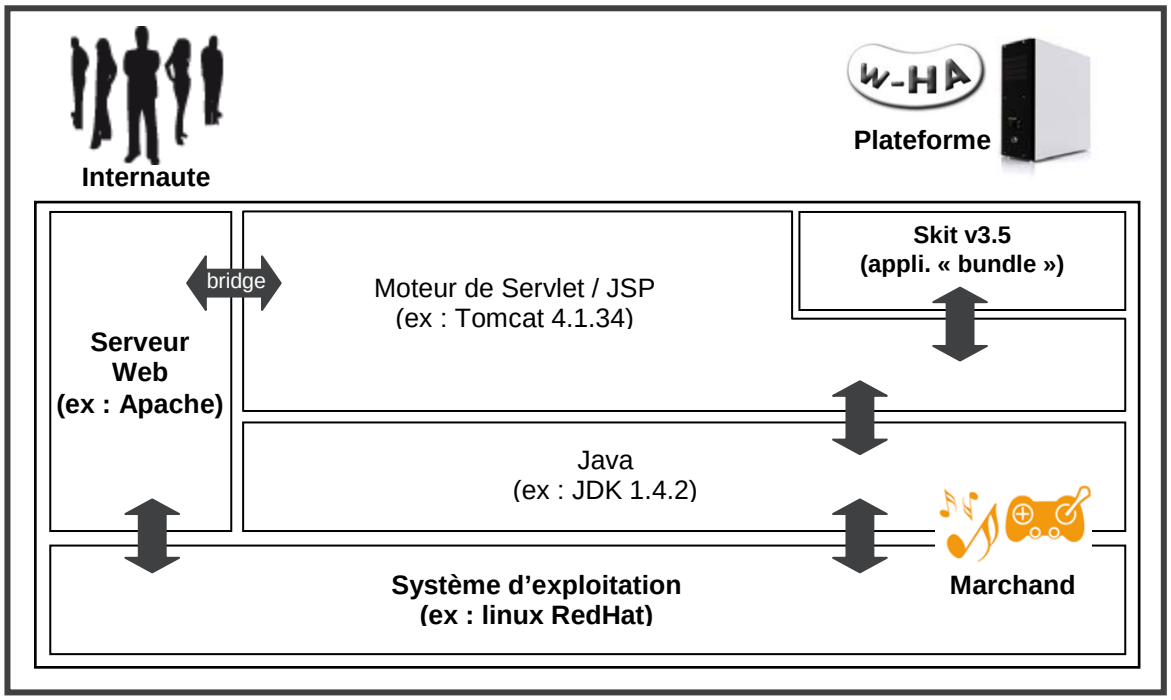
Lorsque cette URL est appelée, vous pouvez alors vérifier si cet trxid ou si ce paramètre supplémentaire définit une livraison déjà effectuée ou non.

Au 1er appel de l'URL, vous redirigez normalement vers la page de livraison.
Pour les éventuels appels ultérieurs, vous bloquez l'accès (et affichez une page d'erreur par exemple).

4. ANNEXE I : PRE-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.

4.1. 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.

4.1.1. 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.

4.1.2. 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 patchs 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 patchs n° xxxxx doivent être installés).

4.1.3. 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.

4.1.4. 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 : Tomat 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.

4.2. Considérations Réseau

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

4.2.1. 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.

4.2.2. 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
https://wanadoo.w-ha.com	Orange Internet	193.28.205.2	443	Sortante
https://free.w-ha.com	Alice/Free	193.28.205.52	443	Sortante
https://sfr.w-ha.com	SFR	193.28.205.17	443	Sortante
https://cb.w-ha.com	Orbeo	193.28.205.21	443	Sortante
https://qualif-marchand.w-ha.com	w-HA tests web	193.164.148.75	443	Sortante

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)

4.3. Autres pré-requis pour l'intégration

4.3.1. 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.

4.3.2. 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.

4.3.3. 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)

5. ANNEXE II : COMPOSANTS DU KIT V3.5

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

Application « bundle »	
Servlet « pos_init »	Gère : - la demande d'autorisation de débit vers w-HA - la réponse à la demande d'autorisation - la confirmation de débit - l'écriture des logs associés - la redirection vers l'URL de livraison (ffUrl)
Fichier de configuration « web.xml »	Contient les Paramètres globaux de la boutique
Fichier de configuration « products.xml »	Contient le Catalogue Produit
Un fichier de log « authorization.txt »	Ecrit les informations concernant les demandes d'autorisation
Un fichier de log « logs.txt » (c.à.d qui feront l'objet d'un débit)	Ecrit les informations sur paiements confirmés (c.à.d qui feront l'objet d'un débit)

Application « demo »	
/demo/acte/html/index.html	Permet de vérifier le bon fonctionnement de la servlet « pos_init » pour l'achat à l'acte d'un produit

5.1. Fichier de configuration "web.xml"

Le fichier de configuration "**web.xml**" de l'application « bundle » contient des éléments de configuration relatifs à la fois à la fonctionnalité « acte » et à la fonctionnalité « abonnement ».

Les éléments qui concernent la fonctionnalité « **acte** », qui fait l'objet de ce document, se trouvent entre les balises :

```
<servlet>
        <servlet-name>servlet_pos_init</servlet-name>
        <servlet-class>com.ipin.core.merchant.skit.WebAuthorizationServlet</servlet-class>
        …
</servlet>
```

Note : Lors du **paramétrage** des fichiers, **un seul paramètre est autorisé dans la définition des url** des fichiers ".xml".

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

» dans tomcat/wepapps/bundle/WEB-INF/web.xml :
- trxCancelFromPaymentPanelUrl
- merchantHomeUrl
- productUnavailableUrl

» dans tomcat/wepapps/bundle/WEB-INF/products.xml :
- fulfillmentUrl

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".

5.1.1. Structure du fichier « web.xml » : sevrlet « pos_init »

Paramètre	Commentaire
merchantLogDir	**Chemin d'accès aux fichiers authorizations.txt et logs.txt (ne pas modifier cette valeur)** Le fichier « authorization.txt » logue les événements concernant les demandes d'autorisation de débit Le fichier « logs.txt » logue les événements concernant les demandes de débit ou «Confirm»
MerchantId	**Identifiant du marchand : (fourni par w-HA)** Il permet à la plate-forme de paiement w-HA d'authentifier le marchand. - un identifiant de test, pour se connecter à la plate-forme de test w-HA est fourni au moment de l'installation. - un identifiant de production, sera fourni suite à la demande de passage en production.
KeyId	**Identifiant de la clef secrète : (fourni par w-HA, même valeur que merchantId)** La clef est partagé entre l'application w-HA et la plate-forme de paiement w-HA. Associé à la " keyValue ", il permet d'assurer l'intégrité des informations - un identifiant de clef de test est fourni au moment de l'installation - un identifiant de clef de production, sera fourni lors du passage en production.
KeyValue	**Valeur de la clef secrète : (fourni par w-HA)** La valeur de la clef est partagée entre l'application w-HA et la plate-forme de paiement w-HA. Associée au " keyId ", elle permet d'assurer l'intégrité des informations - une valeur de clef de test, est fournie au moment de l'installation - une valeur de clef de production, sera fournie lors du passage en production.
NodeAuthorizationUrl	**URL du nœud (plate-forme) w-HA (fourni par w-HA)** L'URL du nœud de test est : https://qualif-marchand.w-ha.com/app-authorization/node L'URL du nœud de production est : https://route.w-ha.com/app-authorization/node
MerchantLanguage	**Langue du marchand : fr (ne pas modifier cette valeur)** La langue du marchand doit être la valeur « fr » (français)
MerchantCurrency	**Devise du marchand : EUR (ne pas modifier cette valeur)** L'euro est la seule devise actuellement acceptée par la plate-forme de paiement w-HA. La devise du marchand doit donc être à la valeur EUR Le montant des produits vendus s'affiche sur le panneau de paiement en euros.
MessageUrl	**Chemin d'accès à la JSP gérant les messages d'erreur (ne pas modifier cette valeur)** Les différents cas d'erreur sont envoyés à la JSP " message.jsp ", au niveau de l'application « acte »
	Pour les 5 paramètres suivants : Si les requêtes http sont prises en charge par Tomcat, préciser le port sur lequel il fonctionne (exemple : http://www.marchand.com:8080/acte/pos_init) Si les requêtes http sont prises en charge par Apache ou IIS, il ne faut pas préciser de port (exemple : http://www.marchand.com/acte/pos_init)
MerchantUrl	**URL où se trouve la Servlet « pos_init » sur le serveur du marchand (à adapter)** Remplacer l'adresse IP par défaut par le Unom de domaineU du marchand.
MerchantHomeUrl	**URL de la Home Page du marchand (à adapter)** En cas d'erreur, la JSP « message.jsp » permet à l'internaute de revenir sur la « merchantHomeUrl »
TrxCancelFrom PaymentPanelUrl	**URL de retour en cas d'annulation de l'achat depuis le panneau de paiement (à adapter)** Si l'internaute clique sur le bouton « Annuler ma commande» sur le panneau de paiement, il est redirigé vers cette URL
ProductUnavailableUrl	**URL de retour si le produit n'est pas disponible (à adapter)** Si le service (pid) choisi par l'internaute n'est pas présent dans le fichier « products.xml », l'internaute est redirigé vers cette URL
XmlProductDatabase	**Chemin d'accès au fichier products.xml (ne pas modifier cette valeur)** Le fichier « products.xml » contient l'ensemble des informations relatives à chaque produit vendu via w-HA.
TimestampDelay	**Délai maximum pour le « Confirm » ou demande de débit (en millisecondes, ne pas modifier cette valeur)** Délai maximum entre la demande d'autorisation et la confirmation de débit, en millisecondes.
ContentProtectionUrl	NON UTILISE (ne pas modifier cette valeur)
ContentProtectionKey	NON UTILISE (ne pas modifier cette valeur)
ServiceUnavailableUrl	NON UTILISE (ne pas modifier cette valeur)

5.1.2. Exemple de fichier « web.xml »

```
<servlet>

        <servlet-name>servlet_pos_init</servlet-name>
        <servlet-class>com.ipin.core.merchant.skit.WebAuthorizationServlet</servlet-class>
        <init-param>
                <param-name>merchantLogDir</param-name>
                <param-value>c:\Tomcat\webapps\bundle\WEB-INF\logs\</param-value>
        <description></description>
        </init-param>
        <init-param>
                <param-name>merchantId</param-name>
                <param-value>5XXX</param-value>
                <description>CUSTOMIZE: Provide the correct merchant id (numeric).</description>
        </init-param>
        <init-param>
                <param-name>keyId</param-name>
                <param-value>5XXX</param-value>
                <description>CUSTOMIZE: Provide the correct keyId (numeric).</description>
        </init-param>
        <init-param>
                <param-name>keyValue</param-name>
                <param-value>abcdefghijklmnopqrstuvwxyz123456</param-value>
                <description>CUSTOMIZE: Provide the correct key.</description>
        </init-param>
        <init-param>
                <param-name>nodeAuthorizationUrl</param-name>
                <param-value>https://route.w-ha.com/app-authorization/node</param-value>
                <description>CUSTOMIZE: Provide the correct host name.</description>
        </init-param>
        <init-param>
                <param-name>merchantLanguage</param-name>
                <param-value>fr</param-value>
        </init-param>
        <init-param>
                <param-name>merchantCurrency</param-name>
                <param-value>EUR</param-value>
        </init-param>
        <init-param>
                <param-name>merchantUrl</param-name>
                <param-value>http://wha.marchand.com/bundle/pos_init</param-value>
                <description>CUSTOMIZE: Provide the correct host name.</description>
        </init-param>
        <init-param>
                <param-name>messageUrl</param-name>
                <param-value>/bundle/jsp/message.jsp</param-value>
        </init-param>
        <init-param>
                <param-name>merchantHomeUrl</param-name>
                <param-value>http://www.marchand.com/demo/bundle/html/index.html</param-value>
                <description>CUSTOMIZE: Provide the correct host name.</description>
        </init-param>
        <init-param>
<param-name>trxCancelFromPaymentPanelUrl</param-name>
<param-value>http://www.marchand.com/demo/bundle/html/panel_cancel.html</param-value>
<description>CUSTOMIZE: Provide the correct host name.</description>
        </init-param>
```

```
		<init-param>
<param-name>productUnavailableUrl</param-name>
<param-value>http://www.marchand.com/demo/bundle/html/productUnavailable.html</param-value>
<description>CUSTOMIZE: Provide the correct host name.</description>
		</init-param>
		<init-param>
			<param-name>xmlProductDatabase</param-name>
			<param-value>/WEB-INF/products.xml</param-value>
		</init-param>
		<init-param>
			<param-name>timestampDelay</param-name>
			<param-value>300000</param-value>
		</init-param>
		<init-param>
			<param-name>serviceUnavailableUrl</param-name>
			<param-value>NON UTILISE - NOT USED</param-value>
			<description>NON UTILISE / NOT USED</description>
		</init-param>
		<init-param>
			<param-name>contentProtectionUrl</param-name>
			<param-value>NON UTILISE - NOT USED</param-value>
			<description>NON UTILISE / NOT USED</description>
		</init-param>
		<init-param>
			<param-name>contentProtectionKey</param-name>
			<param-value>NON UTILISE - NOT USED</param-value>
			<description>NON UTILISE / NOT USED</description>
		</init-param>

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

</servlet>

<servlet-mapping>
		<servlet-name>servlet_pos_init</servlet-name>
		<url-pattern>/pos_init</url-pattern>
</servlet-mapping>
```

5.2. Fichier de configuration « products.xml »

L'ensemble des produits/services mis en vente par le marchand est référencé dans le fichier produits "products.xml". Il s'agit d'un **fichier "xml" dont chaque section xml décrit un produit/service**.

La Servlet « pos_init » récupère les informations relatives au produit/service sélectionné par l'internaute au moyen d'un identifiant produit unique, "productId".

Chaque section « xml » relative à un produit/service apporte une information particulière sur celui-ci.

Le **nombre de produits/services** dans la base produit "products.xml" **n'est pas limité**.

5.2.1. Structure du fichier « products.xml »

Paramètre	Commentaire
ProductId	**Identifiant du produit (alphanumérique, 50 caractères maximum, unique pour chaque produit)** Identifiant unique du produit, passé en paramètre de la fonction Javascript « authorize » sur la page Web du marchand
Amount	**Montant du produit (en euros, le séparateur décimal est le point, par exemple : 1.00)** Le montant du produit vendu, s'affiche sur le panneau de paiement en euros.
Description	**Description du produit (200 caractères maximum, pas d'accent ou de caractères spéciaux)** Description du produit, s'affichant sur le panneau de paiement
Category	**Catégorie de produit (texte libre, 50 caractères maximum)** Peut être utilisé par le marchand à des fins statistiques et marketing, cette information apparaissant dans les fichiers de log.
Class	**Classe de produit w-HA (-1=généraliste, 1=adulte, 2=jeux d'argent)** Utilisé par les internautes pour (s') interdire l'accès à certains contenus.
FulfillmentUrl	**URL finale d'accès au bien** URL sur laquelle l'internaute sera redirigée après son achat.
PaymentGuaranteed	**Paiement garanti par l'opérateur : false (ne pas modifier la valeur – clause contractuelle)** Indique que le marchand accepte tous les utilisateurs, que leur Opérateur Client garantisse ou non les paiements.
IPINHandleRefund	**Demandes de remboursement gérées par w-HA : true (ne pas modifier cette valeur – clause contractuelle)** Indique que la décision d'accorder ou non un remboursement à un utilisateur est prise par w-HA et non par le marchand.
AutoConfirm	**Confirmation (demande de débit) automatique : true (ne pas modifier cette valeur)** Indique que le "Confim" (demande effective de débit) se fait automatiquement, dès réception de la réponse positive à la demande d'autorisation
paymentService	**Type de service de paiement accepté par le marchand : 2 (ne pas modifier cette valeur – clause contractuelle)** Indique que le marchand accepte aussi bien les utilisateurs dont l'Opérateur Client débite les achats sur sa facture (post-payé) que ceux dont l'Opérateur Client débite les achats sur un compte pré-payé.

5.2.2. Exemple de fichier « products.xml »

```
<product-list>

<product>
        <productId>P1</productId>
        <amount>1.00</amount>
        <description>Produit #1</description>
        <category>Image</category>
        <class>-1</class>
        <fulfillmentUrl>http://www.marchand.com/demo/bundle/html/produit1.html</fulfillmentUrl>
        <paymentGuaranteed>false</paymentGuaranteed>
        <iPINHandleRefund>true</iPINHandleRefund>
        <autoConfirm>true</autoConfirm>
        <paymentService>2</paymentService>
</product>

</product-list>
```

5.3. Fichiers de Logs

5.3.1. Fichier "authorizations.txt"

Le chemin d'accès du fichier des transactions autorisées par la plate-forme w-HA (« authorizations.txt ») est précisé dans le fichier de configuration "web.xml" : **merchantLogDir** (par défaut : [TOMCAT_HOME]/webapps/bundle/WEB-INF/authorizations.txt)

Il s'agit d'un fichier texte (format .txt).

Le fichier « authorizations.txt » est mis à jour (une nouvelle ligne est ajoutée) à chaque fois que la servlet « pos_init » reçoit une réponse à la demande d'autorisation de débit.

Attention !

Les logs sont lus par l'application w-HA à chaque requête, afin d'éviter d'éventuel doublons.
En conséquence, au delà d'une certaine taille (plusieurs Mo),le délai de lecture de lecture devient trop important et empêche le bon déroulement des transactions.

Aussi, **il est vivement recommandé à l'éditeur de « purger » régulièrement ce fichier de log**, via l'utilisation de chrono-logs, par exemple.

En pratique, il est mis à jour lorsque l'internaute, sur le panneau de paiement, clique :
- soit sur le bouton « OK »,
- soit sur le bouton « Annuler »

5.3.2. Structure du fichier « authorizations.txt »

Chaque ligne est constituée des champs suivants :

Champs	Description	Type / Format
Transaction ID	Identification unique de la transaction	X-XXXXXXXXXXXXXXX où les X sont des chiffres
Status	Etat de la transaction	AUTHORIZED=1 CANCELLED=2 CONFIRMED=3 REFUNDED=5 REFUND DENIED=7
Reply URL	URL de la plate-forme w-HA utilisée pour la confirmation et l'annulation	URL du type htpps://xxx.w-ha.com/app-node-mct/responder (varie en fonction du FAI du client)
Validity	Durée de validité du produit	En secondes (cf. fichier « products.xml »)
Time stamp	Date et heure de l'achat	Format : aaaa-mm-jj hh:mm:ss
Description	Description du Produit	(cf. fichier « products.xml »)
Merchant ID	Identifiant unique du marchand	(cf. fichier « web.xml »)
Currency	Devise du produit	(cf. fichier « products.xml ») Code ISO de 3 lettres spécifiant la devise dans laquelle le produit est vendu. Il ne doit y avoir qu'une devise par fichier « products.xml »
Merchant Amount	Montant du produit	(cf. fichier « products.xml »)
User City	Ville de l'acheteur	Alphanumérique
User State	Etat de l'acheteur	Code 2 ou 3 lettres de l'Etat
User Zip Code	Code Postal de l'acheteur	Code Postal : numérique
User Country	Pays de l'acheteur	Alphanumérique
Key ID	Non utilisée	Non utilisé

5.3.3. Exemple de fichier "authorizations.txt"

Chaque « autorisation » fait l'objet d'une ligne qui contient différents champs séparés par le caractère "pipe" ("|"). Ceux-ci sont expliqués au paragraphe précédent.

6-7672821718212150|3|https://wanadoo.w-ha.com/app-node-mct/responder|86400|2008-11-06 16:38:45|Produit #1|505|EUR|0.0|PARIS|FR|75007|FR|505
6-587456287965542|3|https:// wanadoo.w-ha.com/app-node-mct/responder|86400|2008-11-06 16:39:07|Produit#3|505|EUR|0.0|PARIS|FR|75007|FR|505
8-452596245485771|3|https://club-internet.w-ha.com/app-node-mct/responder|86400|2008-11-19 17:25:33|Produit #1|505|EUR|0.0|PARIS|FR|75007|FR|505

5.3.4. Fichier "logs.txt"

Le chemin d'accès au fichier des transactions « confirmées » (demande de débit effectif) par le marchand (« logs.txt ») est précisé dans le fichier de configuration "web.xml".
(par défaut : [TOMCAT_HOME]/webapps/bundle/WEB-INF/logs.txt)

Il s'agit d'un fichier texte (format .txt).

A chaque transaction « confirmée » (ou si la « confirmation » n'a pas pu aboutir) par le marchand, une ligne est rajoutée en temps réel dans le fichier.

Les paramètres supplémentaires, passés à la servlet « pos_init » sont récupérés au niveau du fichier « logs.txt », et sont préfixé par '_ap_'

5.3.5. Structure du fichier « logs.txt »

Chaque ligne est constituée des champs suivants :

Champs	Description	Type / Format
Time stamp	Date et heure de la confirmation d'achat	
Status	Etat de la transaction	AUTHORIZED CANCELLED CONFIRMED REFUNDED REFUND DENIED
Transaction ID	Identification unique de la transaction	X-XXXXXXXXXXXXXXX où les X sont des chiffres
Validity	Durée de validité du produit	En secondes
User State	Etat de l'acheteur	Code 2 ou 3 lettres de l'Etat
User Zip Code	Code Postal de l'acheteur	Code Postal : numérique
User Country	Pays de l'acheteur	Alphanumérique
User City	Ville de l'acheteur	Alphanumérique
_ap_XXX	Paramètre supplémentaire du marchand et sa valeur	XXX est le nom du paramètre supplémentaire
pid=	Identifiant unique du produit	(cf. fichier « products.xml »)
ts=	Date et heure de la confirmation d'achat	Format aaaa-mm-jj hh:mm:ss.ccc

5.3.6. Exemple de fichier "logs.txt"

Chaque transaction fait l'objet d'une ligne qui contient différents champs séparés par le caractère "pipe" ("|"). Ceux-ci sont expliqués au paragraphe précédent.

```
Wed Feb 06 16:38:45 CET 2008|Confirm|6-7672821718212150|86400||75007|FR|PARIS|pid=P1|ts=2008-11-06 16:38:35.865
Wed Feb 06 16:39:07 CET 2008|Confirm|6-8541452112494723|86400||75007|FR|PARIS|pid=P3|ts=2008-11-06 16:38:58.648
Tue Feb 19 17:25:33 CET 2008|Confirm|6-5634298321514357|86400||75007|FR|PARIS|_ap_V=1D|_ap_URI=/pages/cuisine/recette.cfm?ID=278|pid=P1|_ap_Nb=2|ts=2008-11-19 17:24:16.956
```