

IPAC 500 Firmware Release NOTE V1.72

CODE PRODUIT

IPAC500

DATE

03/05/2018

Changements depuis la version V1.71

- Pages web sous design BootStrap.
- *IPAC 500 Défilement* :
 - Ajout de 4 numéros cycliques par résident
 - Ajout de logos / images
 - Ajout d'une plage horaire par résident.
- API de gestion Audio, LCD, VOIP.
- Supervision du portier par l'ASM ACCESS (service de management Amphitech, nécessite une connexion Internet sur le même réseau que le portier).
- Ajout de la synthèse vocale « ouverture de la porte » sur activation d'un relais depuis les entrées.

Problèmes résolus

- Entrées NO et NF inversées depuis la version Hardware 440B

Nouvelles Fonctionnalités

1 Création résident IPAC 500

Exemple IPAC 500 Défilement

- Création d'un résident, ajouter un « Nom » et « Prénom », dans le champ « Numéro » entrer un numéro au format P2P (adresse IP) ou au format plan de numérotation IP-PBX. Dans le champ « Compte SIP » choisir l'option appel P2P ou IPBX en sélectionnant un compte SIP valide (déjà renseigné dans le menu SIP).

- Cliquer sur 

Le numéro est alors ajouté dans la liste des numéros cycliques. Il est possible de créer jusqu'à 4 numéros cycliques par résidents (renouveler l'étape précédente).

- L'ordre d'appel peut être modifié, sélectionner un numéro dans la liste puis utiliser les boutons



pour changer l'ordre d'appel des numéros du résident.

- Le Bouton  permet de supprimer un numéro sélectionné dans la liste .
- Si des images ou des logos ont été importés dans la mémoire de l'IPAC à partir du menu **TELECHARGEMENTS / UPLOAD LOGO CONTACT**, il est possible d'associer une image ou un logo à un résident lors de la création du contact (format **.png uniquement** , taille max. 100 Ko) . :

TELECHARGEMENTS

[DOWNLOAD](#) [UPLOAD FICHIERS DE CONFIGURATION](#) [UPLOAD LOGO CONTACT](#) [LISTE DES CONTACTS](#)

Logo disponible

marc.jpeg
 logo_ratp.png
 Maison_Tully.png
 test 1.png
 Maison_Lannister.png
 Maison_Targaryen.png



preview

Upload fichier logo (100ko max, 200x200)

Sélectionner un fichier

Choose Files

No file chosen

Upload

Logo disponible


 marc.jpeg
 logo_ratp.png
 Maison_Tully.png
 test 1.png

➔



preview

VALIDER

Le logo s'affichera pendant le défilement quand ce résident sera affiché dans la liste des résidents de la page d'accueil, voir **PARAMETRES DE BASE/Paramètres portier/ECRAN D'ACCUEIL/Timer recherche résidents**.

- Une plage horaire peut être associée au résident Plage horaire off .
 Cette plage horaire est à créer à partir du menu **PARAMETRES DE BASE/Plages horaires** :

PLAGES HORAIRES

Plage 1 **Plage 2** Plage 3 Plage 4 +

Nom de la plage

	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	10h	11h	12h	13h	14h	15h	16h	17h
Lundi																		
Mardi																		
Mercredi																		
Jeudi																		
Vendredi																		
Samedi																		
Dimanche																		

VALIDER

APPLIQUER

Exemple IPAC 500 Boutons

- Une plage horaire peut être associée au(x) bouton(s) d'appel.
- L'affichage d'un logo ou d'une image est indisponible.

2 API de gestion du produit local

Ces API sont des API natives du produit, elles permettent une utilisation directe à partir d'un terminal présent sur le même réseau, capable d'envoyer des requêtes de Type GET et POST sous différents formats comme JSON ou URL ENCODED.

2.1 API PORTE

Le code porte API correspond à un des codes communs Relais 1 ou Relais 2. L'API porte est soumise

à la plage horaire ou au mode d'activation du code commun Relais 1 ou Relais 2 (Local/Distant).

Paramètres usine : user = admin / password = admin .

Dans l'exemple, *user = toto / password = titi.*

Le *code du relais = 1234* correspond à un des 4 codes communs Relais 1 ou Relais 2.

Num = numéro relais, soit 1 = RL1, 2 = RL2 , 3 = RL1 et RL2.

Authentification NONE :

`http://adresse_IP_IPAC/api/remote/?login=toto&password=titi&code=xxxx&relay=num` (GET)

`curl -d "code=1234&relay=num&login=toto&password=titi" -X POST http://adresse_IP_IPAC/api/remote` (POST)

Authentification BASIC :

`http://toto:titi@adresse_IP_IPAC/api/remote/?code=xxxx&relay=num` (GET)

`curl -d "code=1234&relay=num" -X POST http://toto:titi@adresse_IP_IPAC/api/remote` (POST)

Authentification DIGEST :

`http://toto:titi@adresse_IP_IPAC/api/remote/?code=xxxx&relay=num` (GET, mode Hashé) (GET)

`curl -d "code=1234&relay=num" -X POST http://toto:titi@adresse_IP_IPAC/api/remote --digest` (POST)

Il est possible pour les méthodes GET et POST d'utiliser le mode « https » dans la requête à la place du mode « http ».

Pour l'utilisation du mode « https » sous CURL, on ajoute - - insecure (certificat non signé).

Attention : Le mode d'authentification est sauvegardé dans le cash de la page durant toute l'ouverture de celle-ci.

Codes retours :

-200 OK = code OK

-403 Forbidden (mauvais code, type activation non distante)

-401 Unauthorized (plage horaire non active)

-423 LOCKED : Relais passés en PDL ou NETCUT

-480 Temporarily Unavailable (code en cours)

Exemples Formats JSON et URL Encoded :

POST:

```
curl -X POST -d '{"code":"1111","relay":"1"}' http://admin:admin@192.168.0.30/api/remote/ --digest --header "Content-Type: application/json"
```

POST:

```
curl -d "code=1111,relay=1" -H "Content-Type: application/x-www-form-urlencoded" -X POST http://admin:admin@192.168.0.30/api/remote/ --digest
```

GET:(auth NONE/BASIC)

```
curl -H "Content-Type: application/x-www-form-urlencoded" http://admin:admin@192.168.0.30/api/remote/?code=1111&relay=1
```

2.2 API VOIP

Cette API permet de contrôler à distance la partie téléphonie du produit.

- **Répondre à un appel entrant**

POST:

```
curl -d "type=answer" -H "Content-Type: application/x-www-form-urlencoded" -X POST  
http://admin:admin@192.168.0.30/api/voip/ --digest
```

GET: (auth NONE/BASIC)

```
curl -H "Content-Type: application/x-www-form-urlencoded"  
http://admin:admin@192.168.0.30/api/voip/?type=answer
```

- **Terminer une communication ou un appel entrant**

POST:

```
curl -d "type=terminate_all" -H "Content-Type: application/x-www-form-urlencoded" -X POST  
http://admin:admin@192.168.0.30/api/voip/ --digest
```

GET: (auth NONE/BASIC)

```
curl -H "Content-Type: application/x-www-form-urlencoded"  
http://admin:admin@192.168.0.30/api/voip/?type=terminate_all
```

- **Lancer un appel**

table:

Précise dans quelle table de base de donnée se situe le contact. Le format accepté est : liste – ldap – libre

id:

Précise l'index du contact dans sa base de données. Le format accepté est : 1 – 192.168.1.100 - 1000@proxy

Exemples Format JSON et URL Encoded :

POST:

```
curl -d '{"type":"call","table":"libre","id":"sip:192.168.0.22"}' -H "Content-Type: application/json"  
-X POST http://admin:admin@192.168.0.30/api/voip/ --digest
```

POST:

```
curl -d '{"type":"call","table":"liste","id":"4"}' -H "Content-Type: application/json" -X POST  
http://admin:admin@192.168.0.30/api/voip/ --digest
```

POST:

```
curl -d "type=call&table=libre&id=sip:192.168.0.22" -H "Content-Type: application/x-www-form-urlencoded" -X POST http://admin:admin@192.168.0.30/api/voip/ --digest
```

GET: (auth NONE/BASIC)

```
curl -H "Content-Type: application/x-www-form-urlencoded"
```

```
http://admin:admin@192.168.0.30/api/voip/?type=call&table=libre&id=192.168.0.22
```

2.3 API AUDIO

Cette API permet d'envoyer un fichier .wav encodé en base64 dans une URL adressée au portier, le fichier est ensuite diffusé dans le haut-parleur du portier (1 Mo max.).

- Lire un fichier Wav

loop:

Nombre de répétitions du fichier audio. Dans le cas d'un nombre nul ou non précisé, le son sera diffusé en boucle indéfiniment tant que le portier est au repos. Intervalle : 0 - 9999

Pour un fichier audio (payload.txt)

Exemple du contenu du fichier payload.txt :

```
{"type": "wav", "loop" : "2", "data": "UklGRmQfAABXQVZFZm10IBAAA.....ouJilmHhA=="}
```

La « value » de la clé « data » correspond à un fichier .wav converti en base64.

Exemple d'envoi d'une requête JSON contenant un fichier WAV encodé en base64:

```
curl -X POST -d `cat payload.txt` http://admin:admin@192.168.0.30/api/audio/ --digest --header "Content-Type: application/json" --header "Expect:"
ou
curl -X POST -d $(cat payload.txt) http://admin:admin@192.168.0.30/api/audio/ --digest --header "Content-Type: application/json" --header "Expect:"
```

- Stopper la diffusion du fichier audio

Indique l'arrêt de la diffusion du fichier audio.

Exemple commande stop format URL encoded:

POST:

```
curl -d "type=wav&data=stop" -H "Content-Type: application/x-www-form-urlencoded" -X POST http://admin:admin@192.168.0.30/api/audio/ --digest
```

GET: (auth NONE/BASIC)

```
curl -H "Content-Type: application/x-www-form-urlencoded" http://admin:admin@192.168.0.30/api/voip/?type=wav&data=stop
```

2.4 API LCD

Cette API permet d'envoyer un fichier image (.png, .jpeg ou .GIF). Le fichier est ensuite affiché sur l'afficheur LCD du produit, *uniquement* quand le portier n'est pas en mode « communication » ou « ouverture porte ».

- **Afficher une image**

tempo:

Durée d'affichage de l'image. Dans le cas d'un nombre nul ou non précisé, l'image sera affichée indéfiniment tant que le portier est au repos. Intervalle : 0 – 9999 sec.

Exemple d'envoi d'une requête JSON contenant un fichier png encodé en base64:

```
curl -X POST -d $(cat image.b64) http://admin:admin@192.168.0.48/api/video/ --digest --header "content-type: application/json" --header "Expect:"
```

Avec le fichier image. b64 contenant:

```
{"type":"img","tempo":"60","data":"iVBORw0KGgoAAAANSUhEUgAAAWQAAAEACAYAAACEfg... "}
```

La « value » de la clé « data » correspond à une image format .png convertie en base64.

- **Stopper l'affichage de l'image**

Indique l'arrêt de l'affichage de l'image.

Exemple commande stop format Json:

```
curl -X POST -d $(cat stop.txt) http://admin:admin@192.168.0.48/api/video/ --digest --header "content-type: application/json" --header "Expect:"
```

stop.txt :

```
{"type":"img","tempo":"60","data":"stop"}
```

3 Connexion serveur ASM

3.1 Paramétrage du produit :

ASM

PARAMETRES ASM
PROXY HTTP

PARAMETRES ASM

Identification produit	<input type="text" value="8963225"/>
Clé client	<input type="text" value="test"/>
uri	<input type="text" value="https://amphitech.fr"/>
Chemin	<input type="text" value="/api/v2/"/>
Status de la connexion	●
Dernière mise à jour	<input type="text" value="Success,Thu May 23 14:48:37 2019"/>

Ce menu permet de se connecter à un serveur de provisionnig ASM **ACCESS** ou tiers (nécessite une connexion internet sur le réseau), en utilisant les API de gestion du produit développées par Amphitech :

- Notify.
- Events.
- Settings.

Ces requêtes permettent de :

- Mettre à jour un produit à distance (paramètres, logo, certificats radius...).
- Notifier la présence du produit.
- Activer le / les relais.
- Etre informé des actions locales, de l'utilisation des codes clavier, de l'activation des relais, des appels sortants et entrants...

L'identification du produit vers le serveur utilise un modèle : Token / Id.

Les Champs à remplir :

- Identification du produit : utiliser le numéro de série du produit.
- Clé client : Identifiant du compte ASM ACCESS.
- URL : Adresse du serveur.
- Chemin : répertoire API Serveur.
- Status connexion : VERT : CONNECTE / ORANGE : Produit non activé sur serveur / ROUGE : non connecté (identifiant, clé client, ou adresse, incorrectes).
- Dernière mise à jour : Horodatage de la dernière synchronisation des configurations avec le serveur.

En cas d'utilisation d'un Proxy HTTP sur le réseau, utiliser les paramètres PROXY HTTP. Renseigner :

- Type de proxy
- Adresse du serveur
- Port
- Login et password si besoin.

ASM

PARAMETRES ASM PROXY HTTP

Type de proxy

Serveur

Port

Login

password

VALIDER

3.2 Connexion au serveur

 Connexion

Connexion

Identifiant ou email :

Mot de passe :

[Connexion](#)

[Mot de passe oublié](#) [Créer un compte](#)

Se connecter sur le serveur ASM ACCESS 2, se connecter ou Créer un compte. Si les produits ont été correctement paramétrés pour se connecter au serveur, les produits seront listés dans « produits activés » ou « produit non activé ». Un produit non activé signifie que celui-ci est en attente de validation par l'administrateur du compte.

Liste des produits

Code produit	Identité	Adresse installation	Version	Object ID	Console	Actions
<input type="text" value="Enter Code produ"/>	<input type="text" value="Enter Identité..."/>	<input type="text" value="Enter Adresse insi"/>	<input type="text" value="Enter Version..."/>	<input type="text" value="Enter Object ID..."/>		
IPAC50021	000000000		1.02	49352		
IPAC50021	000000000		1.71	93220		
IPAC50021	000000000		1.71	93360		

Configuration (IPAC50021)

Paramètres de base Paramètres avancés Utilisateurs Serveur ASM

Liste des contacts Relais Codes relais Entrées Messages vocaux Paramètres portier Plages horaires Réglages audio

Index	Prénom	Nom	Numéros	Plage horaire	Prioritaire	Image	Actions
<input type="text" value="Enter Index"/>	<input type="text" value="Enter Prénom"/>	<input type="text" value="Enter Nom..."/>	<input type="text" value="Enter Numé"/>	<input type="text" value="Enter Plage"/>			
<input checked="" type="checkbox"/>	001	P2P	marc	<input type="text" value="sip:192.168.0.23"/> <input type="text" value="sip:1000@1"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	002	gxx3275	marc	<input type="text" value="sip:1005@1"/> <input type="text" value="92.168.0.25"/> <input type="text" value="2"/>	<input checked="" type="checkbox"/>		

10 ▾

Liste des évènements

Export CSV

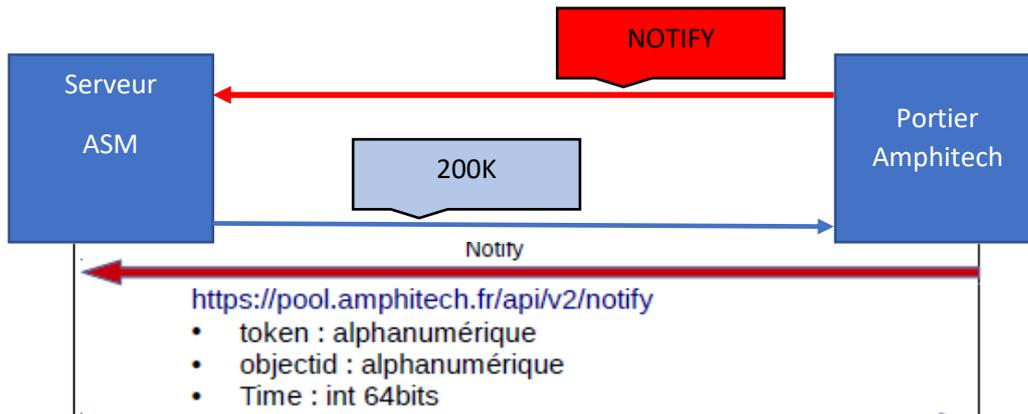
Date	object_id	Date	Event	value	numero	ip	request	code	terminated
<input type="text" value="Enter Da"/>	<input type="text" value="Enter obj"/>	<input type="text" value="Enter Da"/>	<input type="text" value="Enter Eve"/>	<input type="text" value="Enter val"/>	<input type="text" value="Enter num"/>	<input type="text" value="Enter ip.."/>	<input type="text" value="Enter rec"/>	<input type="text" value="Enter coc"/>	<input type="text" value="Enter ter"/>
19/04/2019 à 11:29:45	93360			outgoing:sip:192.168.0.23	sip:192.168.0.23		vocal	FA	false
19/04/2019 à 11:29:34	93360			outgoing:sip:192.168.0.23	sip:192.168.0.23		vocal	01	true
19/04/2019 à 11:28:59	93360			outgoing:sip:192.168.0.23:5060	sip:192.168.0.23:5060		vocal	FA	false
19/04/2019 à 11:28:51	93360			outgoing:sip:192.168.0.23:5060	sip:192.168.0.23:5060		vocal	01	true
19/04/2019 à 11:10:45	93360			1111;1			open	RL	false
19/04/2019 à 11:10:30	93360			1111;1			open	RL	false
19/04/2019 à 11:08:28	93360			outgoing:sip:192.168.0.23	sip:192.168.0.23		vocal	FA	false

4 Format des requêtes AMPHITECH PRODUIT-SERVEUR

Attention, l'utilisation des API de mises à jour du produit nécessite une grande prudence dans la manipulation des fichiers de configurations du produit. Veillez à respecter l'intégralité et la syntaxe des fichiers. Dans le cas d'une erreur de syntaxe, le produit ne chargera pas les paramètres envoyés, les paramètres précédents seront rechargés.

4.1 Connexion du produit au serveur

- Inscription d'un nouveau produit : envoi du premier Notify par le portier vers le serveur ASM Amphitech.



Exemple requête NOTIFY, méthode POST format JSON

```
POST /api/v2/notify/ HTTP/1.1
Accept-Encoding: identity
Content-Length: 66
Host: pool.amphitech.fr
Content-Type: application/json
Connection: close
User-Agent: STS-Embedded
```

```
{"token": "administrator", "objectid": "0000", "time": 1542289092}
```

CAS 1 : REPONSE DU SERVEUR : TOKEN valide mais Object ID non connu



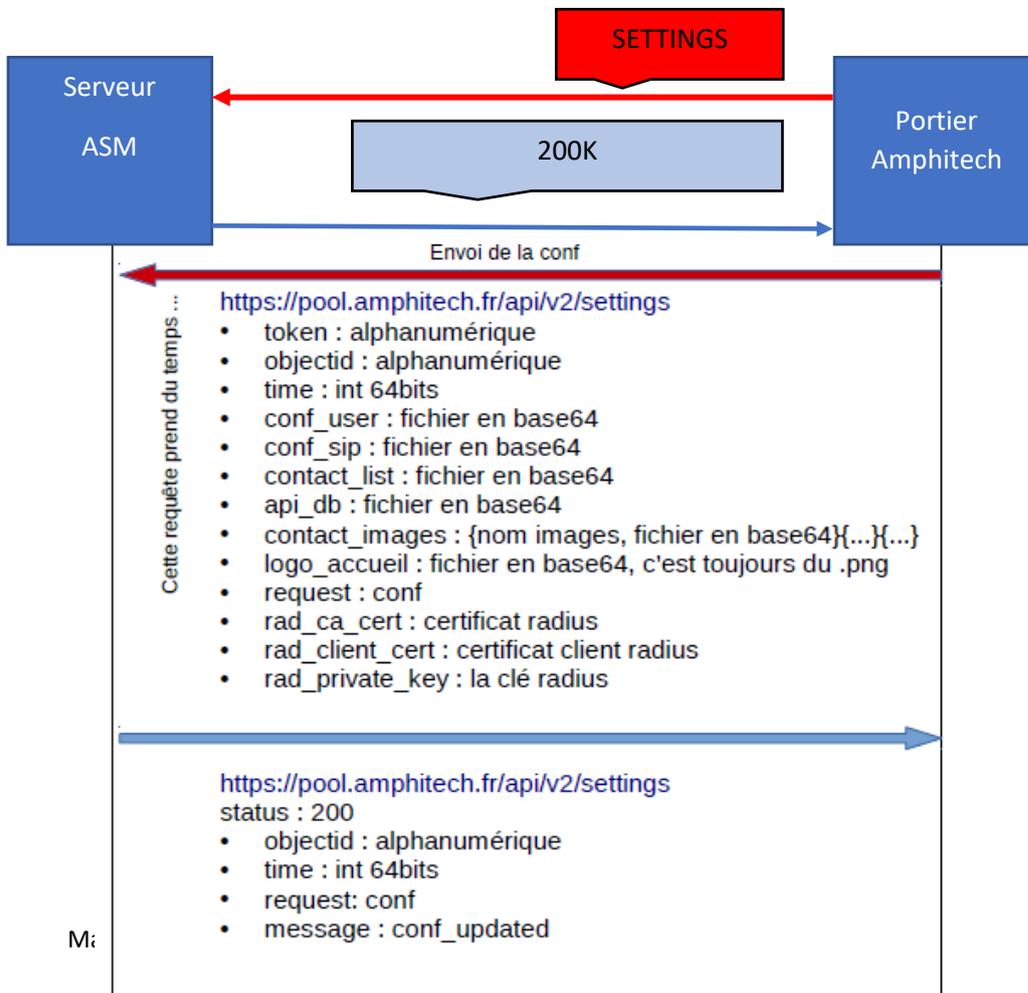
Exemple de réponse à la requête NOTIFY, méthode POST format JSON

```

HTTP/1.1 404 ERROR
Server: nginx/1.12.2
Date: Thu, 15 Nov 2018 12:39:10 GMT
Content-Type: application/json
Content-Length: 37
Connection: close
Vary: Accept-Encoding
Vary: Accept
Allow: POST, OPTIONS
X-Frame-Options: SAMEORIGIN

{"message":"unkown_objectid"}
  
```

Dans un premier temps, le serveur renvoie un 404 ERROR + Unknown ObjectID.
 Dans un second temps, le portier envoie au serveur toute sa configuration, acquitté par un message « conf_updated ».



Exemple requête SETTINGS, méthode POST format JSON

```
POST /api/v2/settings/ HTTP/1.0
Host: pool.amphitech.fr
X-Real-IP: 80.13.231.226
X-Forwarded-Proto: http
X-Forwarded-For: 80.13.231.226
X-Forwarded-Host: 80.13.231.226
Connection: close
Content-Length: 8334
Accept-Encoding: identity
Content-Type: application/json
User-Agent: STS-Embedded
```

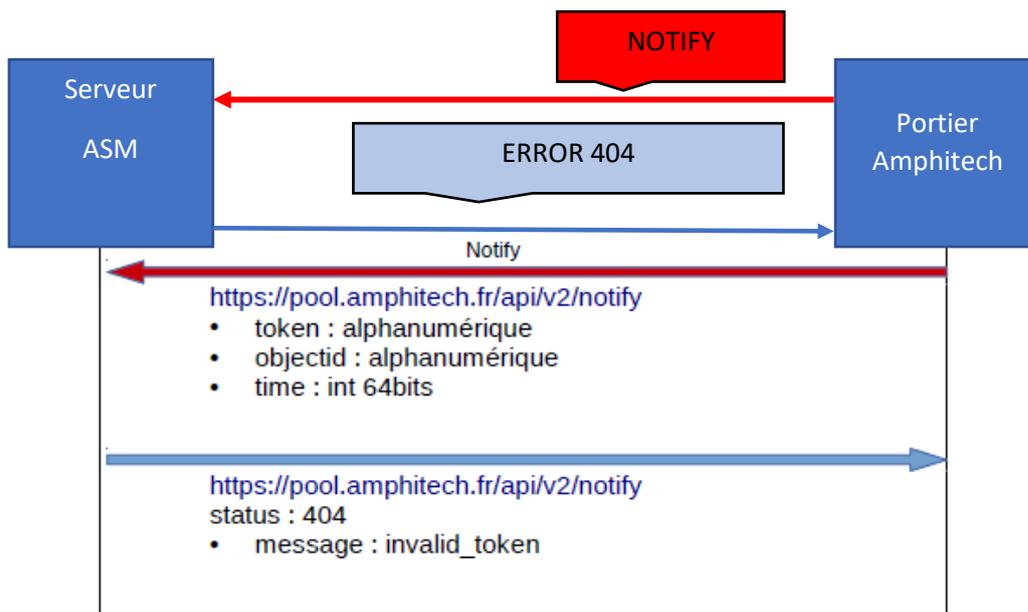
```
{"conf_user": "W0Zpcm13YXJ [...] cnVICg==", "token": "administrator", "objectid": "0000", "time": 1542289103, "conf_sip": "W2F1ZGlv [...] DAwMAo=", "request": "conf".....}
```

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 15 Nov 2018 12:39:22 GMT
Content-Type: application/json
Content-Length: 73
Connection: close
Vary: Accept-Encoding
Vary: Accept
Allow: POST, OPTIONS
X-Frame-Options: SAMEORIGIN
```

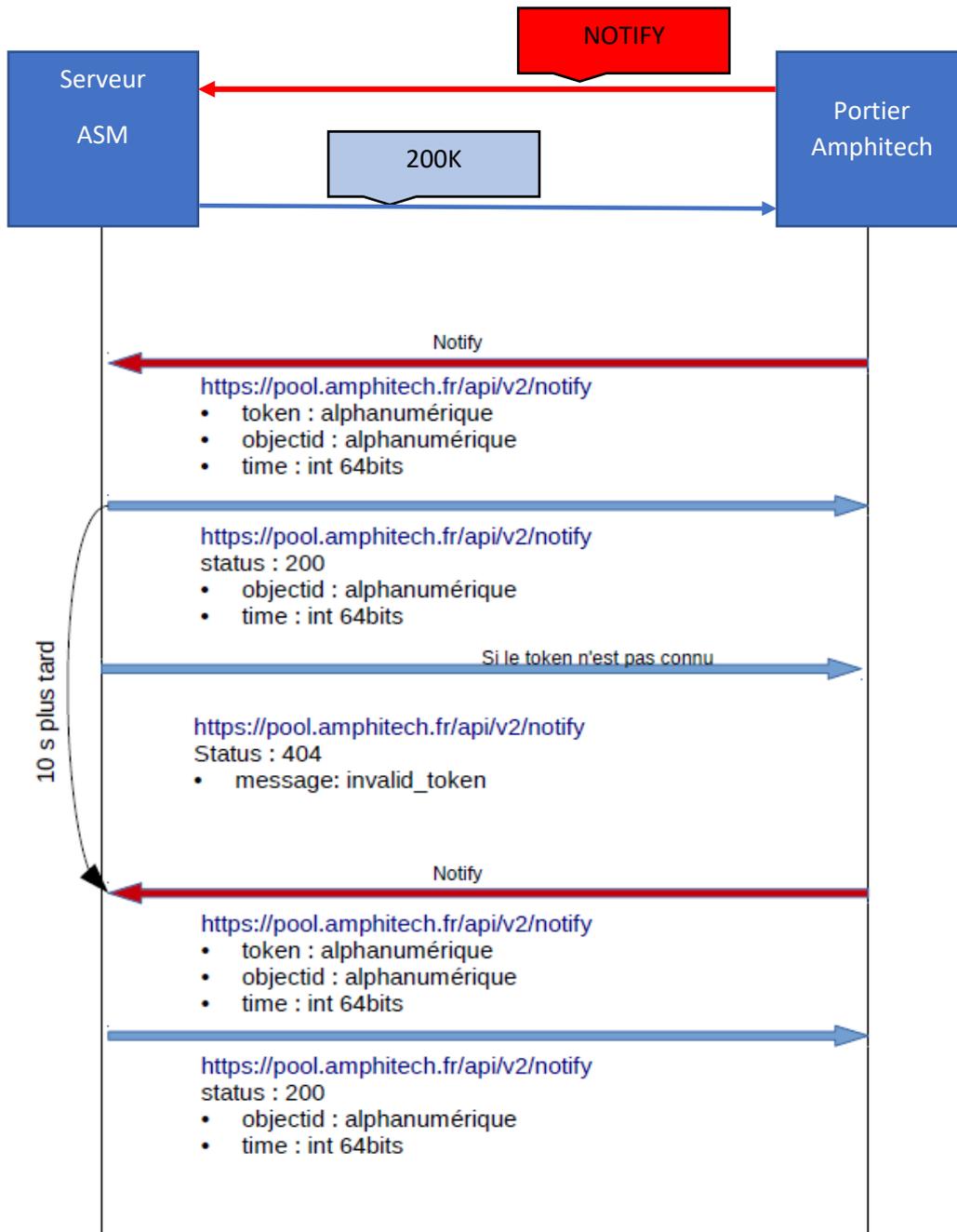
```
{"message": "conf_updated", "request": "conf", "time": "15664599", "objectid": "0000"}}
```

CAS 2 : TOKEN non valide

Le Token utilisé pour se connecter au serveur n'existe pas. Il faut créer un compte ASM et renseigner le Token de ce compte dans le produit (menu **PARAMETRES AVANCES / ASM**).



- Envoi cyclique du NOTIFY (produit vers serveur pour donner signe de vie au serveur ASM)



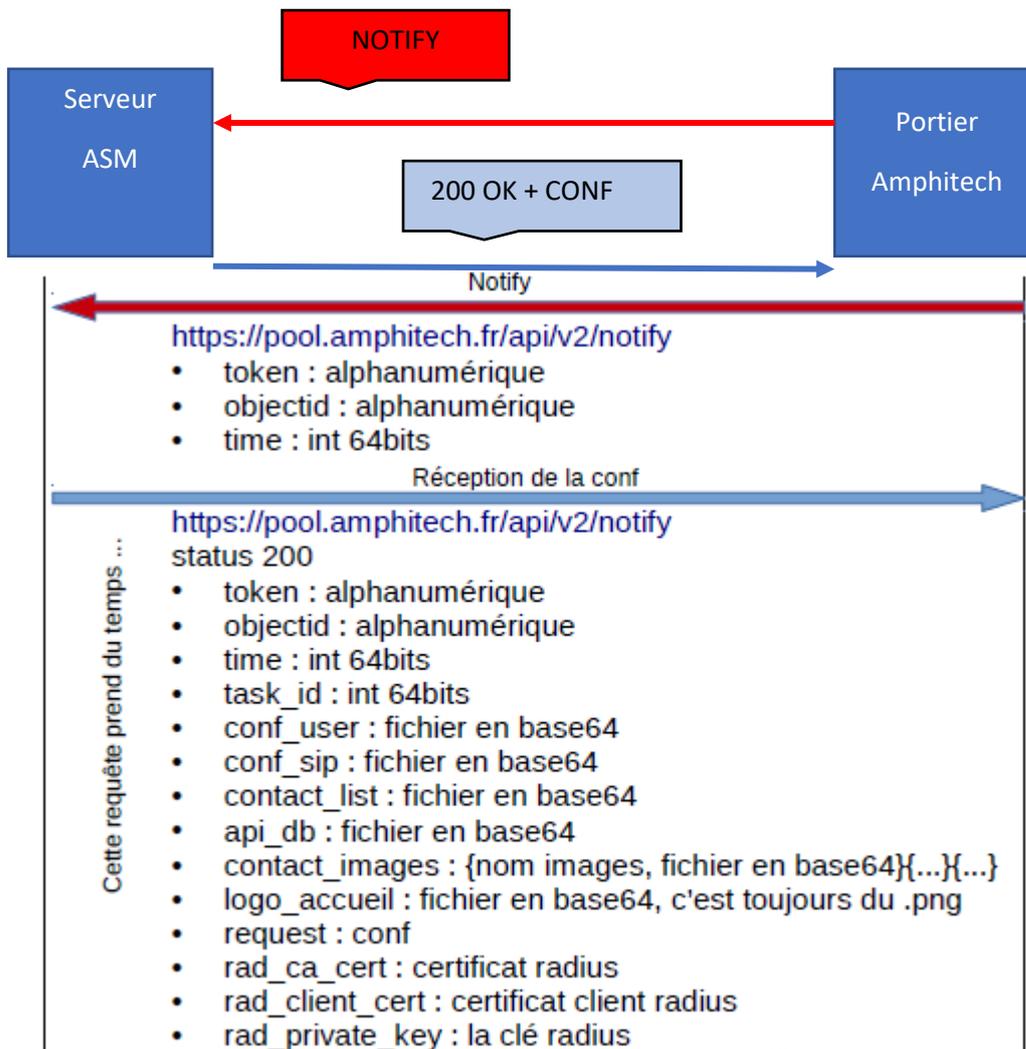
Si le Token est inconnu, le message « invalid_token » sera renvoyé par le serveur avec une erreur 404.

4.1.1 Modifications depuis le serveur ASM

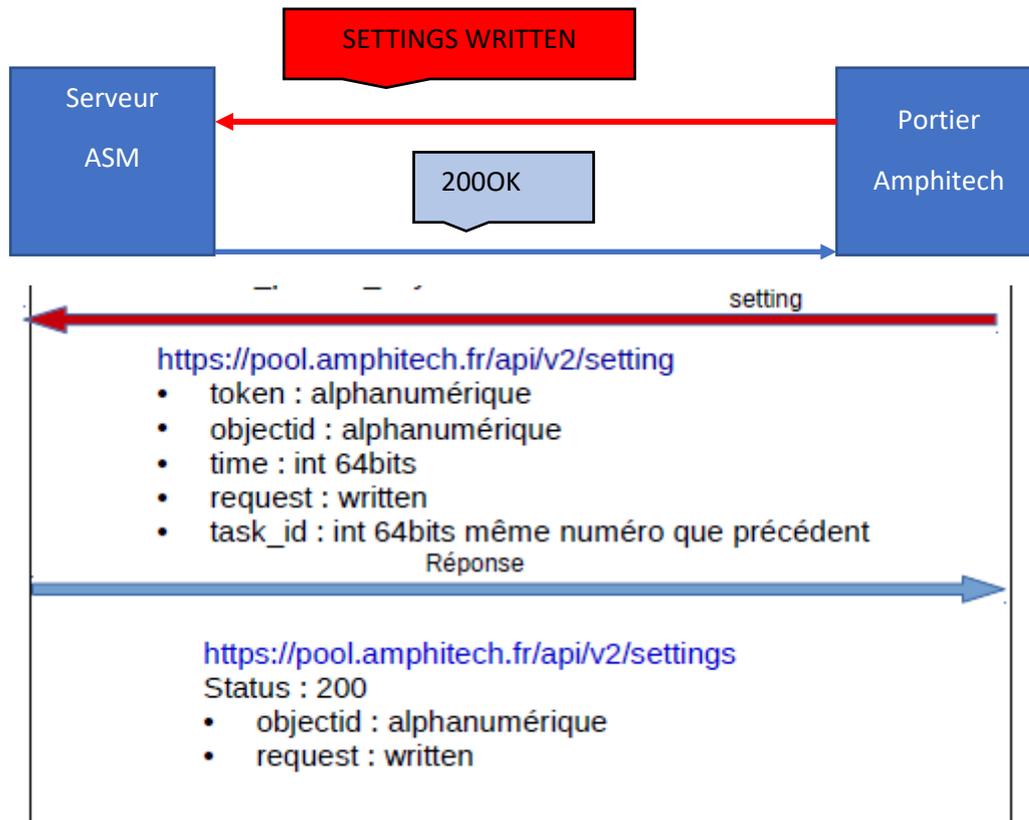
- Envoi cyclique du NOTIFY (produit vers serveur) : si une mise à jour est disponible sur le serveur (*action disponible task_id xxx*) le produit récupère la nouvelle configuration et renvoie un acquittement d'écriture de la nouvelle configuration afin que le serveur n'envoie plus la configuration au prochain notify. Si l'échange se passe mal, au prochain NOTIFY le produit peut toujours récupérer la configuration sur le serveur.

Lorsque la réception des paramètres dans le produit est terminée, le produit redémarre et met à jour les paramètres dans le Firmware. Le produit renvoie la configuration au serveur afin de valider la configuration précédemment envoyée.

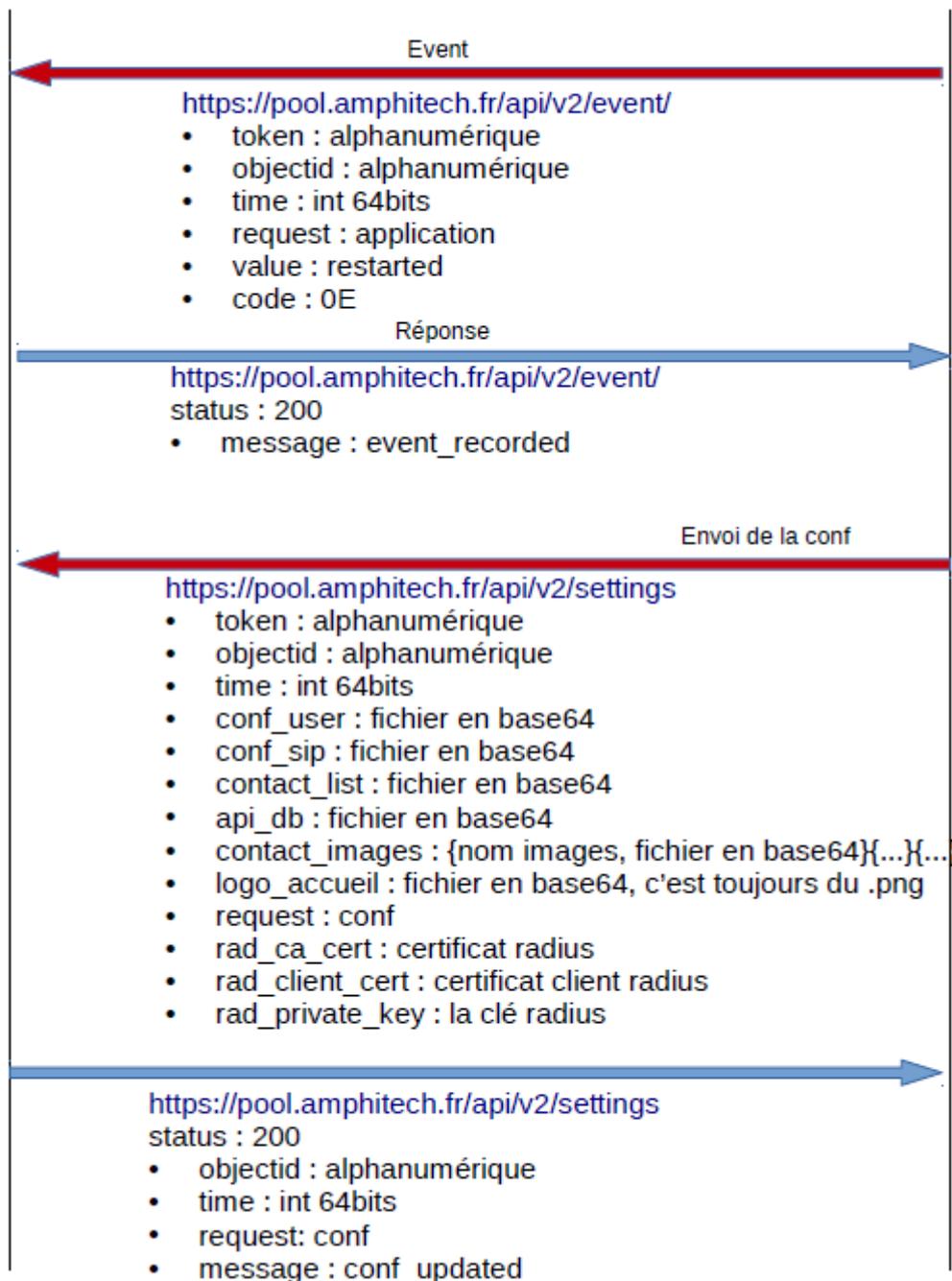
Exemple avec Object ID et TOKEN valides



Le produit renvoie un acquittement d'enregistrement de la nouvelle configuration avant de redémarrer pour prendre en compte les nouveaux paramètres.

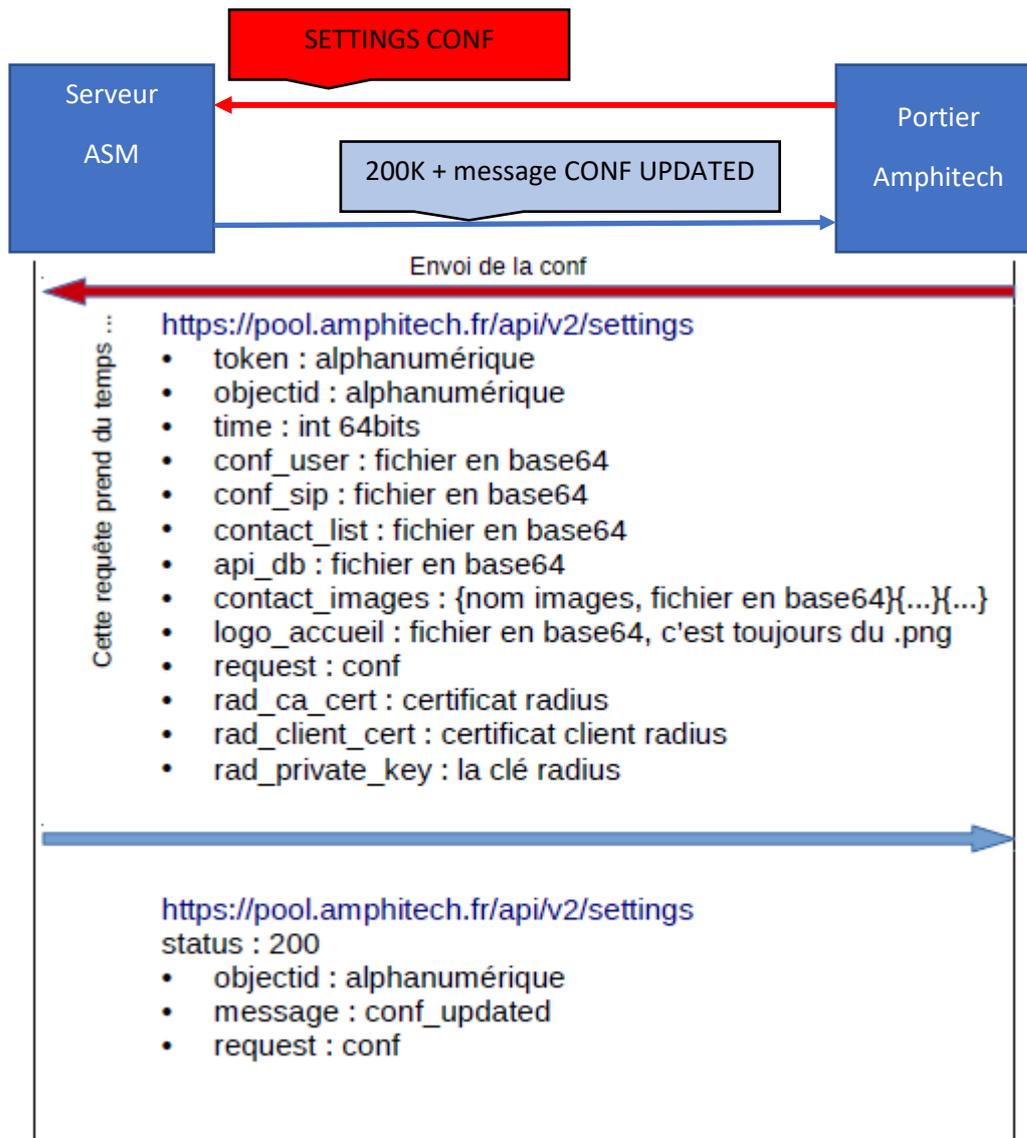


Le produit redémarre et envoie un évènement de redémarrage au serveur. Cet évènement est sauvegardé. Le produit renvoie alors sa configuration qui doit être en toute logique la même que celle envoyée précédemment par le serveur. Le serveur renvoie un message informant le produit que sa configuration actuelle est bien sauvegardée dans le serveur.



4.1.2 Modifications depuis le produit via ses pages web

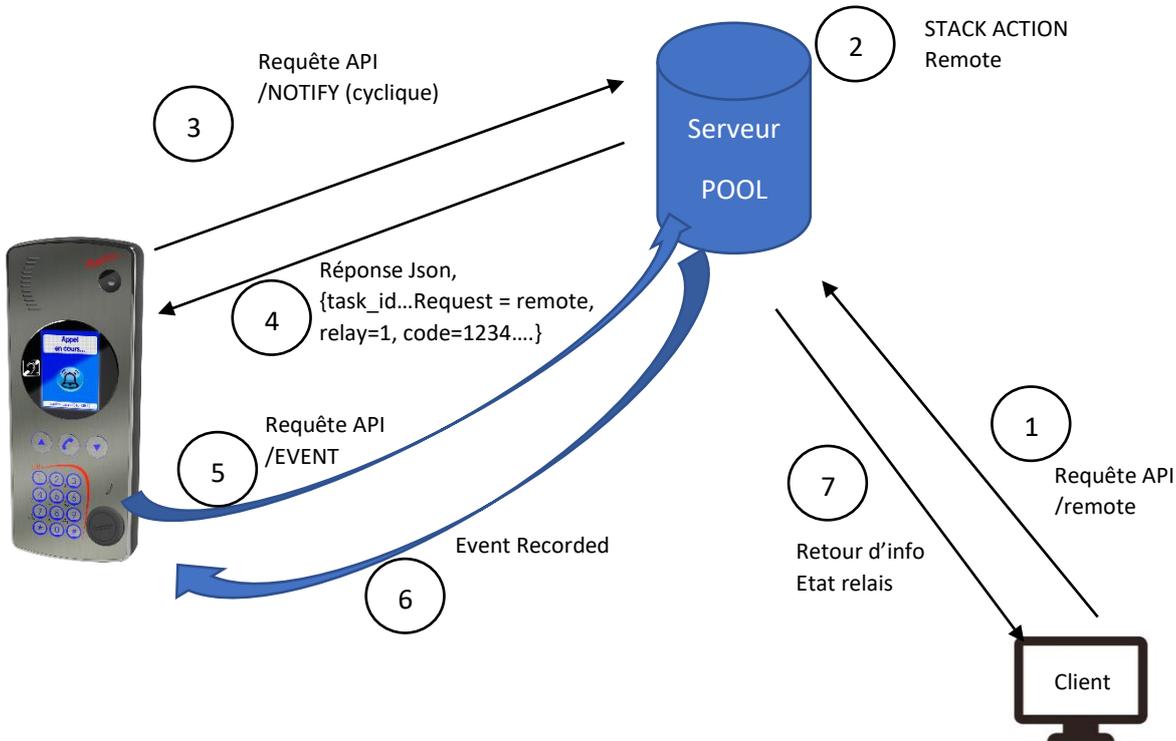
- Après une mise à jour locale du produit via les pages web, le produit envoie une requête de type SETTINGS au serveur suivi des paramètres (en base 64). La configuration du serveur sera en adéquation avec celle chargée sur le/les produits. Une fois le serveur mis à jour avec les paramètres, le message « conf_updated » est envoyé au produit pour informer de la bonne réception et de l'enregistrement des paramètres.



4.1.3 Pilotage du portier via le serveur ASM

4.1.3.1 Schéma de principe

Exemple pour un IPAC ObjectID =0000 et Token = administrator avec API REMOTE



4.1.3.2 Exemple service Remote

1. Un client via son PC / ASIP Stream (application AMPHITECH) peut donner un ordre d'ouverture porte au relais du portier (ObjectID 0000). **Attention l'application smartphone fonctionne avec un login et un password autorisant l'utilisation de l'API porte à cet utilisateur selon l'administration définie via ASM-ACCESS.**
2. Sur le serveur ASM une action task_id « remote » est en attente à destination de l'Object ID 0000 appartenant au TOKEN administrator.
3. Produit Object ID 0000 appartenant au TOKEN administrator envoie une requête cyclique NOTIFY au serveur Pool.

```
POST /api/v2/notify/ HTTP/1.0
Host: pool.amphitech.fr
X-Real-IP: 80.13.231.226
X-Forwarded-Proto: http
X-Forwarded-For: 80.13.231.226
X-Forwarded-Host: 80.13.231.226
Connection: close
Content-Length: 66
Accept-Encoding: identity
Content-Type: application/json
User-Agent: STS-Embedded
```

```
{"token": "administrator", "objectid": "0000", "time": 1542289095}
```

4. Le Serveur ASM répond par l'envoi d'une tâche « remote » à l'Object ID 0000 appartenant au TOKEN administrator.

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 15 Nov 2018 12:39:14 GMT
Content-Type: application/json
Content-Length: 8442
Connection: close
Vary: Accept-Encoding
Vary: Accept
Allow: POST, OPTIONS
X-Frame-Options: SAMEORIGIN
```

```
{"task_id":0,"token":"administrator","objectid":"0000","request":"remote","type":"open","relay":"1","code":"1234","time":15658999}
```

5. Le portier exécute l'ordre ou non en fonction des codes, plages horaires, type d'enclenchement, occupation du relais, etc. et renvoie au serveur ASM un EVENT code « RL » contenant la réponse de l'exécution de la tâche remote par le portier (code format http).

```
POST /api/v2/event/ HTTP/1.0
Host: pool.amphitech.fr
X-Real-IP: 80.13.231.226
X-Forwarded-Proto: http
X-Forwarded-For: 80.13.231.226
X-Forwarded-Host: 80.13.231.226
Connection: close
Content-Length: 123
Accept-Encoding: identity
Content-Type: application/json
User-Agent: STS-Embedded
```

```
{"task_id":0, "value": {1 :200, 2 :403}, "token": "administrator", "code": "RL", "objectid": "0000", "time": 1542289074}
```

Ajout du code EVENT RELAIS : RL

Détail de la clé value avec **code** retours: `value": {1 : code, 2 : code}`

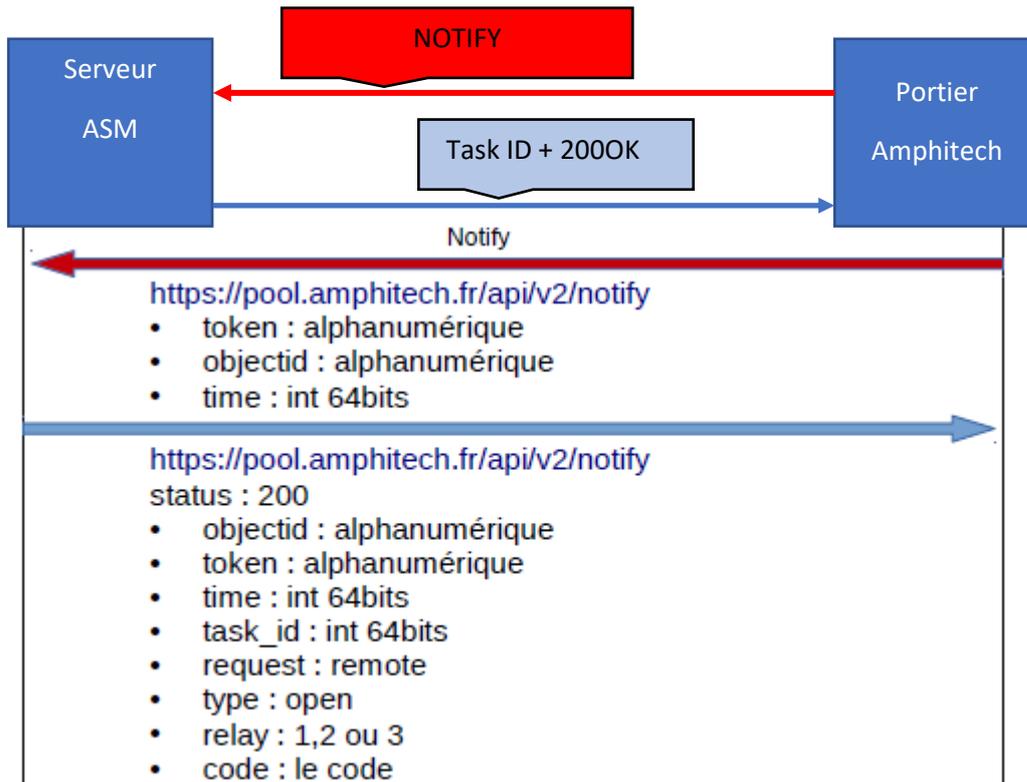
200 OK Code bon, ouverture de la porte à suivre.
423 LOCKED Relais de type PDL
480 TEMPORARLY UNAVAILABLE Code local en cours ou relais déjà actif
401 UNAUTHORIZED En dehors de la plage horaire
403 FORBIDDEN Code erroné et pas le relais concerné
400 BAD REQUEST Requête non comprise

6. Le serveur informe le portier que l'évènement à bien été stocké.

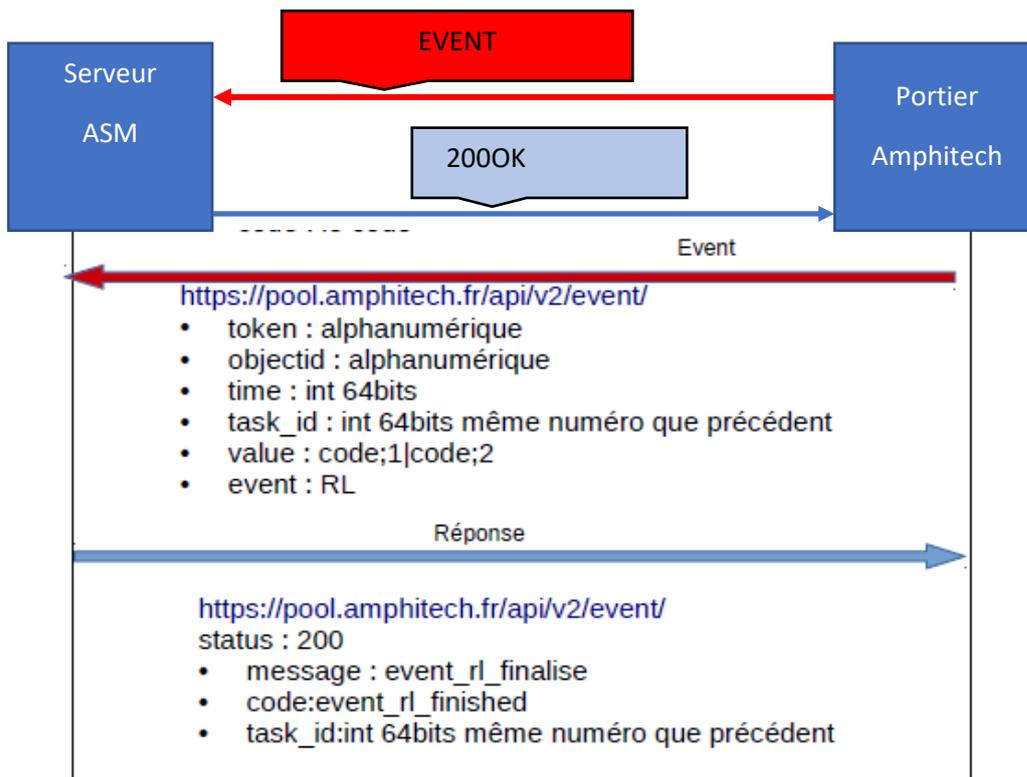
```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 15 Nov 2018 12:38:54 GMT
Content-Type: application/json
Content-Length: 28
Connection: close
Vary: Accept-Encoding
Vary: Accept
Allow: POST, OPTIONS
X-Frame-Options: SAMEORIGIN
```

```
{"code":"event_rl_finished","task_id":0,"message":"event_rl_finalise"}
```

7. Le client est informé de l'ouverture ou non du relais.



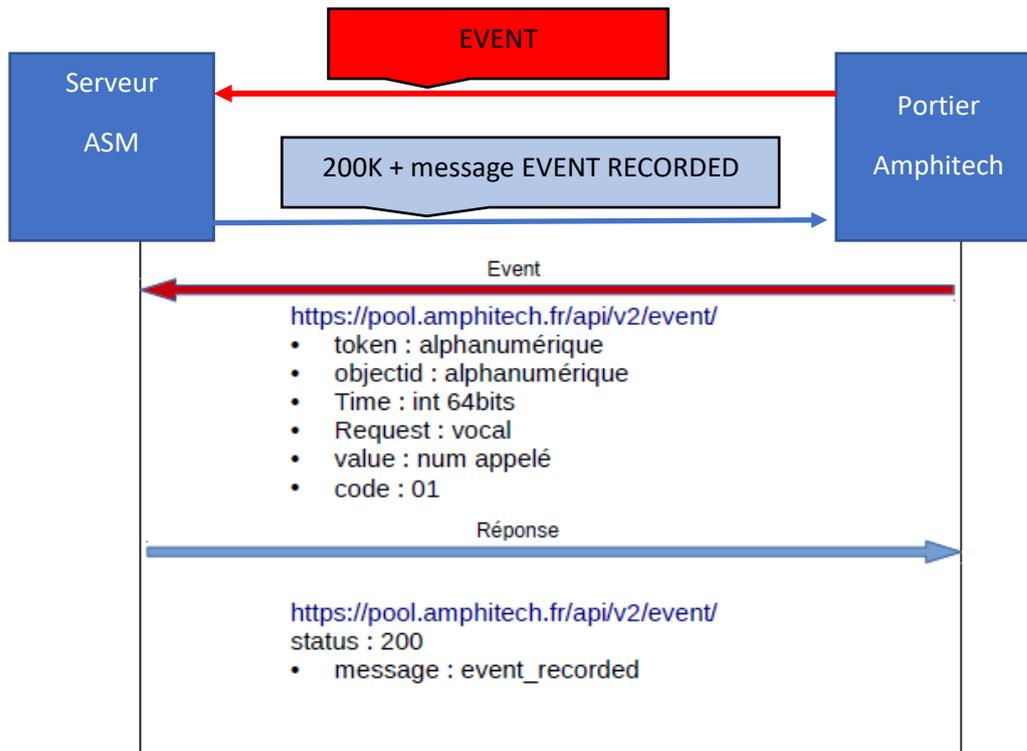
En fonction des plages horaires et du code envoyé le relais s'ouvre ou non.



4.2 Remontée des événements du produit vers le serveur ASM

Envoi de la requête EVENT vers le serveur ASM à la suite d'un évènement lié à la télésurveillance du produit (appel vocal, activation relais, code entrée sur le clavier...)

EXEMPLE : EVENT APPEL VOCAL CODE 01



```

POST /api/v2/event/ HTTP/1.0
Host: pool.amphitech.fr
X-Real-IP: 80.13.231.226
X-Forwarded-Proto: http
X-Forwarded-For: 80.13.231.226
X-Forwarded-Host: 80.13.231.226
Connection: close
Content-Length: 123
Accept-Encoding: identity
Content-Type: application/json
User-Agent: STS-Embedded
    
```

```

{"value": "outgoing ;sip:00143674252@192.168.0.252", "token": "administrator", "code": "01", "objectid": "0000", "request": "vocal", "time": 1542289074}
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Thu, 15 Nov 2018 12:38:54 GMT
Content-Type: application/json
Content-Length: 28
Connection: close
Vary: Accept-Encoding
Vary: Accept
Allow: POST, OPTIONS
X-Frame-Options: SAMEORIGIN
    
```

```

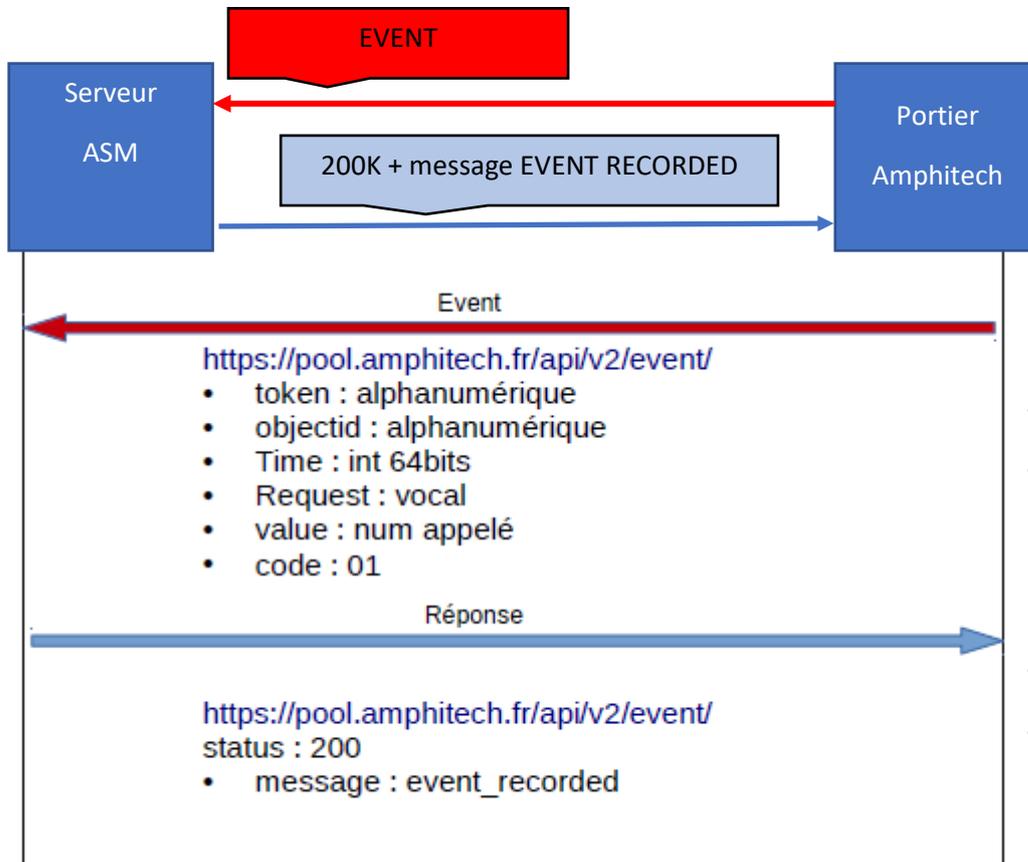
{"message": "event_recorded"}
    
```

Codes d'évènements possibles

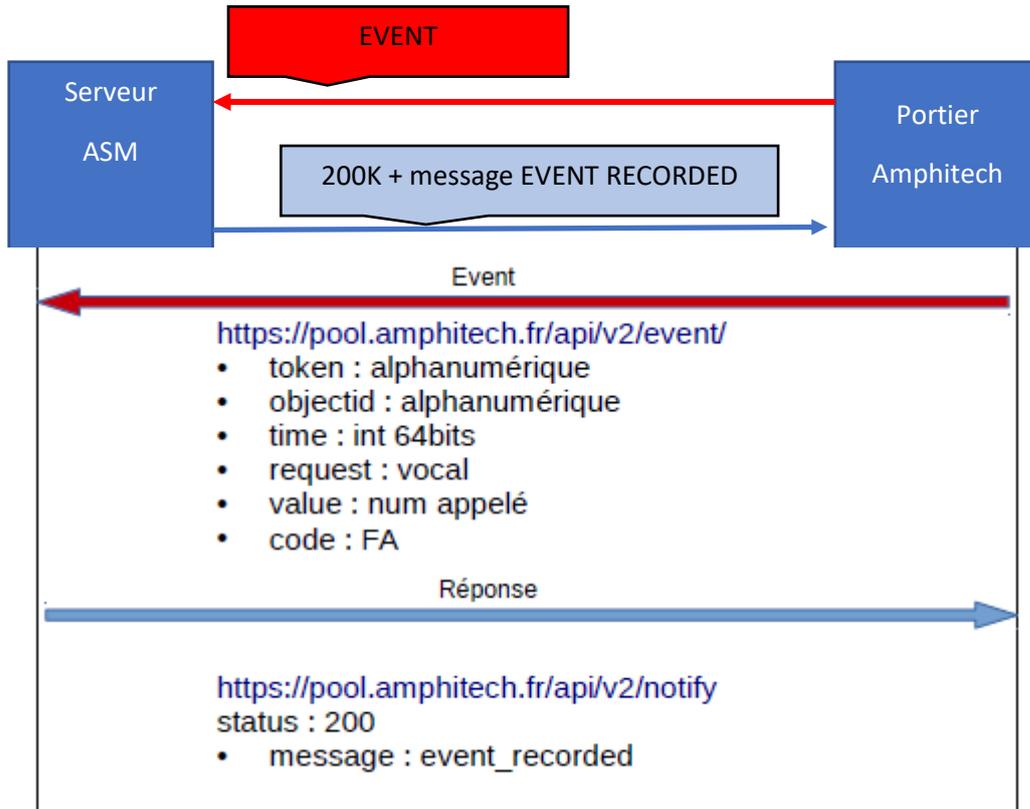
Codes [xx]	Raison d'appel	request	value
OE	Redémarrage soft	application	restarted
01	Appel vocal sortant	Vocal	Outgoing/incoming ; Numéro
FA	Fin d'appel vocal	Vocal	Outgoing/incoming ; Numéro
ST	Début de Stream vidéo vers ASM ACCESS	Stream	Adresse IP Numéro
FS	Fin de Stream vers ASM-ACCESS	Stream	Outgoing/incoming ; Num
RL	Relais enclenché	open	Code ; num relais
CL	Code entré localement au clavier	Ke yBoard	code

Autres exemples

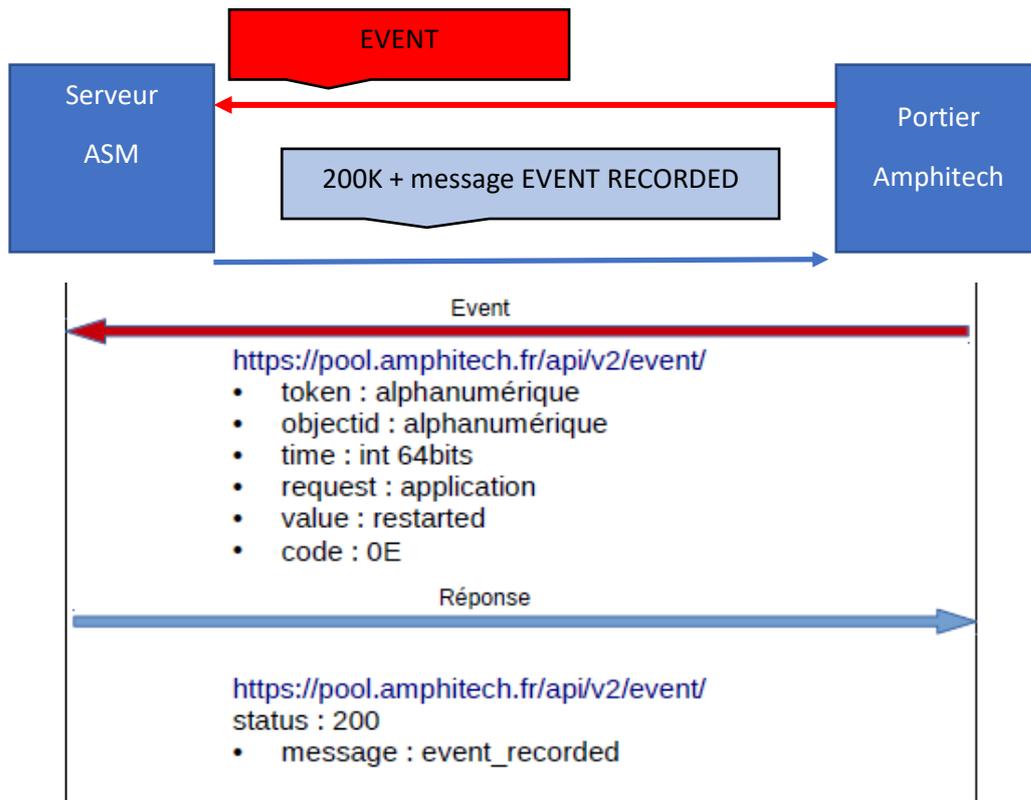
- Appel vocal sortant



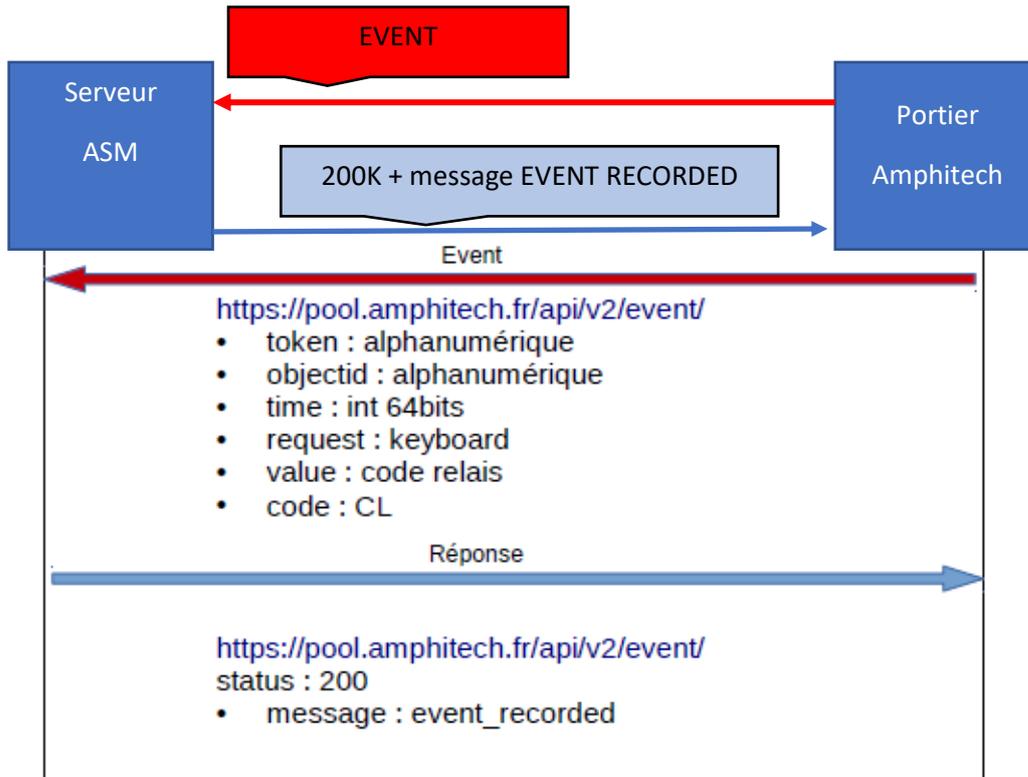
- Fin d'appel vocal



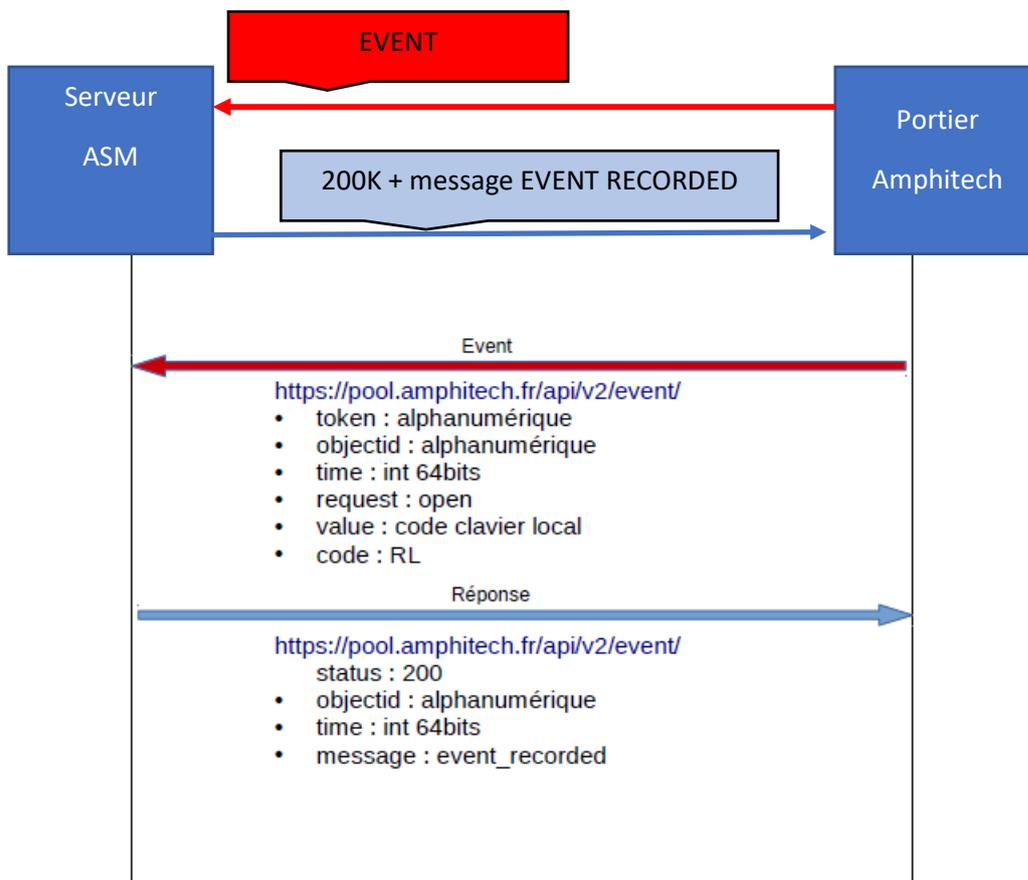
- Redémarrage du portier



- Saisie de code au clavier



- Ouverture du relais (distant ou local)



- Cas de réponse Token Invalide (pour tous les Events)



- Cas de réponse, ObjectID non connu (pour tous les Events)

