EIR (Registre d'Identité d'Équipement)

Vue d'ensemble

Le HSS comprend un EIR (Registre d'Identité d'Équipement) intégré qui fournit une vérification de l'identité des équipements pour les appareils mobiles. L'EIR valide les numéros IMEI (Identité Internationale d'Équipement Mobile) pour déterminer si l'équipement mobile est autorisé, volé ou sous observation avant de permettre l'accès au réseau.

Capacités clés

- **Interface S13** : Vérification de l'identité de l'équipement via le protocole Diameter
- **Validation IMEI** : Vérifier l'identité de l'équipement en utilisant IMEI/IMEISV
- **Correspondance flexible** : Correspondance de motifs basée sur Regex pour IMEI, IMEISV et IMSI
- **Classification à trois niveaux** : Support de la liste blanche, liste noire et liste grise
- **Évaluation ordonnée** : Les règles sont évaluées par ordre de priorité croissante, et une nouvelle règle peut être insérée à n'importe quelle position
- **Politiques configurables** : Comportement personnalisable pour les équipements inconnus
- **API REST** : Opérations CRUD complètes pour la gestion des règles EIR

Architecture

Interface Diameter

Interface	ID d'application	Pair	Objectif
S13	16,777,252	MME/SGSN	Vérification de l'identité de l'équipement

Base de données des règles d'équipement

L'EIR utilise un système de correspondance basé sur des règles flexible :

EIR_RULE		
uuid	id	PK
string	action	
string	regex	
int	precedence	
timestamp	inserted_at	
timestamp	updated_at	

Actions de règle :

- `whitelist` - Autoriser l'équipement
- `blacklist` - Bloquer l'équipement
- `greylist` - Surveiller l'équipement

Modèles Regex : Correspondre à IMEI, IMEISV ou IMSI

Valeurs d'état de l'équipement

État	Code	Signification	Action réseau
Liste blanche	0	Équipement approuvé	Autoriser l'accès au réseau
Liste noire	1	Équipement volé/bloqué	Refuser l'accès au réseau
Liste grise	2	Équipement sous observation	Autoriser avec surveillance

Interface S13

Opérations prises en charge

Demande de vérification d'identit�� d'équipement (ECR) / Réponse de vérification d'identité d'équipement (ECA)

Direction : MME/SGSN → HSS (EIR)

Déclencheur : MME vérifie l'identité de l'équipement lors de l'attachement ou de la mise à jour de la zone de suivi

AVPs de demande :

- Session-Id
- Origin-Host, Origin-Realm
- Destination-Realm
- Auth-Session-State
- Terminal-Information
  - IMEI (14 chiffres tel que transporté dans l'AVP IMEI S13 ; concaténé avec Software-Version, cela forme l'IMEISV de 16 chiffres. La longueur n'est pas validée)
  - Software-Version (2 chiffres, optionnel)

- User-Name (IMSI, optionnel)
- Vendor-Specific-Application-Id

Actions EIR :

1. Extraire IMEI, Software-Version (si présent) et IMSI (si présent)
2. Si IMSI fourni :
   - Valider que l'abonné existe et est activé
   - Mettre à jour l'état de l'abonné avec les informations de dernière vue (`last_seen_realm`, `last_received_diameter_message_type`, `last_seen_imei`, `last_seen_imei_software_version`, `last_seen_imei_timestamp`). L'identité de l'équipement est enregistrée indépendamment de l'état de l'équipement retourné.
3. Tenter une recherche d'équipement par ordre de priorité :
   - **Correspondance IMEISV** (IMEI + Software-Version concaténés)
   - **Correspondance IMEI** (IMEI uniquement)
   - **Correspondance IMSI** (si fourni dans la demande)
   - **Politique d'équipement inconnu** (comportement par défaut configuré)
4. Retourner l'état de l'équipement

AVPs de réponse :

- Session-Id (renvoyé de la demande)
- Result-Code : 2001 (succès)
- Equipment-Status : 0 (liste blanche) / 1 (liste noire) / 2 (liste grise)

Réponses d'erreur :

- Experimental-Result : 5422 (équipement/abonné non trouvé)
- Experimental-Result : 5012 (erreur générale)

Logique de correspondance d'équipement

Ordre de priorité

L'EIR utilise une stratégie de recherche en cascade pour maximiser la flexibilité de correspondance :

```
1. IMEISV (IMEI + Software-Version)
   ↓ (si aucune correspondance)
2. IMEI uniquement
   ↓ (si aucune correspondance)
3. IMSI (si fourni dans la demande)
   ↓ (si aucune correspondance)
4. Politique d'équipement inconnu
```

Algorithme de correspondance

Étape 1 : Correspondance IMEISV

- Concaténer IMEI + Software-Version : `"35979139461611" + "08" = "3597913946161108"`
- Tester contre les règles EIR par ordre de priorité croissante
- Retourner l'action ("whitelist", "blacklist", "greylist") de la première règle correspondante

Étape 2 : Correspondance IMEI (fallback)

- Utiliser uniquement l'IMEI : `"35979139461611"`
- Tester contre les règles EIR par ordre de priorité croissante
- Retourner l'action de la première règle correspondante

Étape 3 : Correspondance IMSI (fallback si IMSI fourni)

- Utiliser l'IMSI de la demande : `"999999876543210"`
- Tester contre les règles EIR par ordre de priorité croissante

- Retourner l'action de la première règle correspondante
- **Cas d'utilisation** : Bloquer tout l'équipement pour un abonné spécifique

Étape 4 : Politique d'équipement inconnu (dernier recours)

- Paramètre de configuration : `eir_unknown_equipment_behaviour`
- Options :
  - `:whitelist` - Autoriser l'équipement inconnu (permissif)
  - `:blacklist` - Bloquer l'équipement inconnu (restrictif)
  - `:greylist` - Observer l'équipement inconnu (modéré)
  - `:reject_unknown_equipment` - Retourner l'erreur 5422 (strict)

Précédence des règles

Chaque règle a une `précédence` entière de 1 à 65535. Dans chaque étape de correspondance, les règles sont testées par ordre de priorité croissante et la première correspondance l'emporte, donc les nombres inférieurs sont évalués en premier.

- Créer une règle sans une priorité l'ajoute après la priorité la plus élevée actuelle.
- Créer une règle avec une priorité explicite l'insère à cette position. Chaque règle existante avec une priorité égale ou supérieure descend d'un cran, que la valeur demandée ait déjà été utilisée ou non, donc les lacunes ne sont jamais comblées. Avec les règles `a`, `b`, `c` à 1, 2, 3, créer `d` avec une priorité 1 donne `d`, `a`, `b`, `c` à 1, 2, 3, 4.
- Mettre à jour la priorité d'une règle la déplace à cette position et déplace uniquement les règles entre ses anciennes et nouvelles positions, donc le reste de la liste conserve son ordre.
- Renvoyer la priorité actuelle d'une règle, par exemple un GET suivi d'un PUT du même corps, ne change rien.
- Supprimer une règle laisse un vide dans la numérotation. Les lacunes sont inoffensives car seul l'ordre relatif compte.
- La priorité ordonne les règles dans une étape de correspondance. Cela ne change pas l'ordre des étapes IMEISV, IMEI, IMSI ci-dessus.

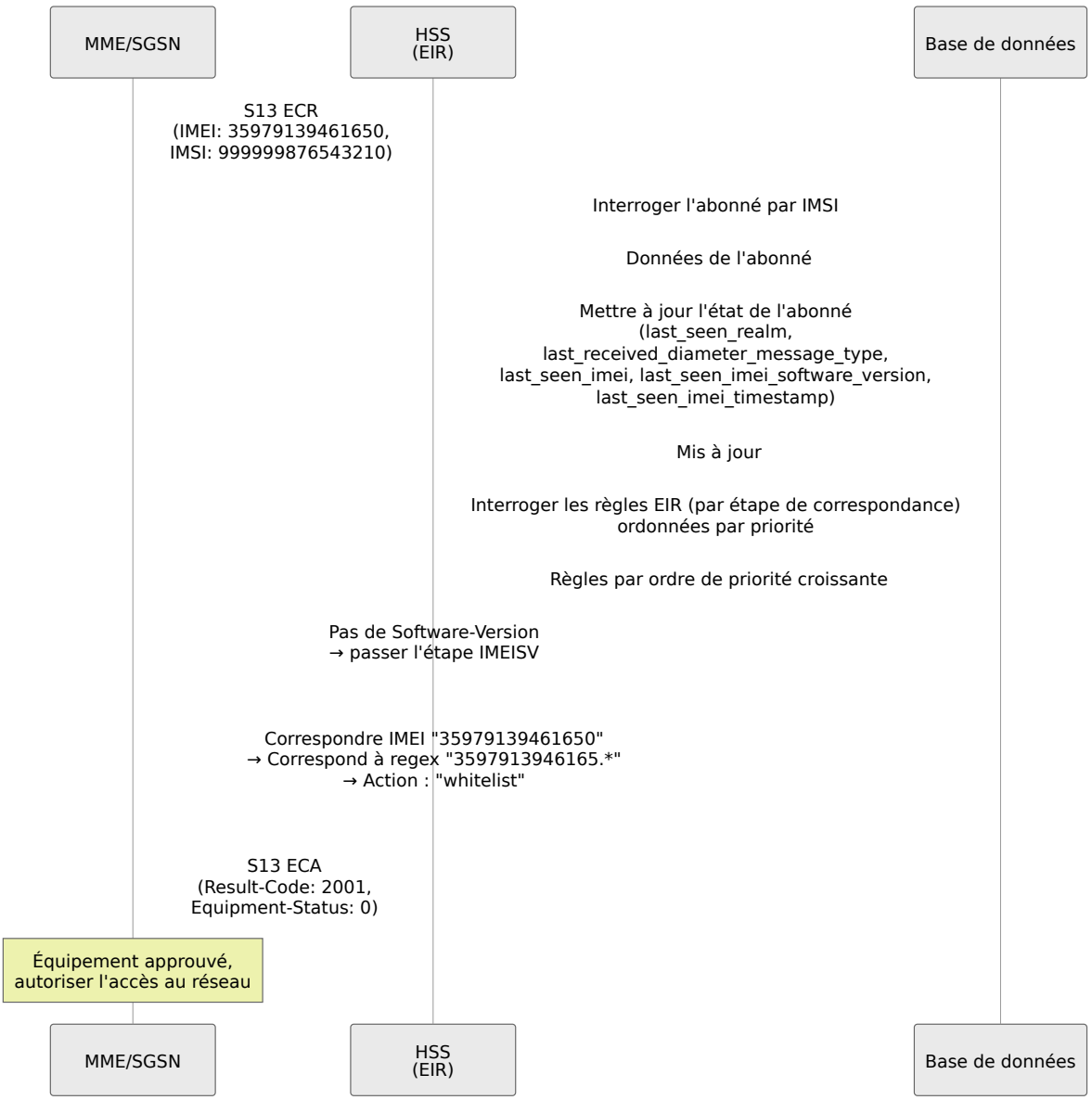
Les règles qui existaient avant l'introduction de la priorité sont numérotées de 1 à N dans l'ordre de création lors de la migration de la base de données. Si l'une de ces règles se chevauche, examinez leur priorité avec `GET /api/eir/rule` après la mise à niveau.

Exemples de modèles Regex

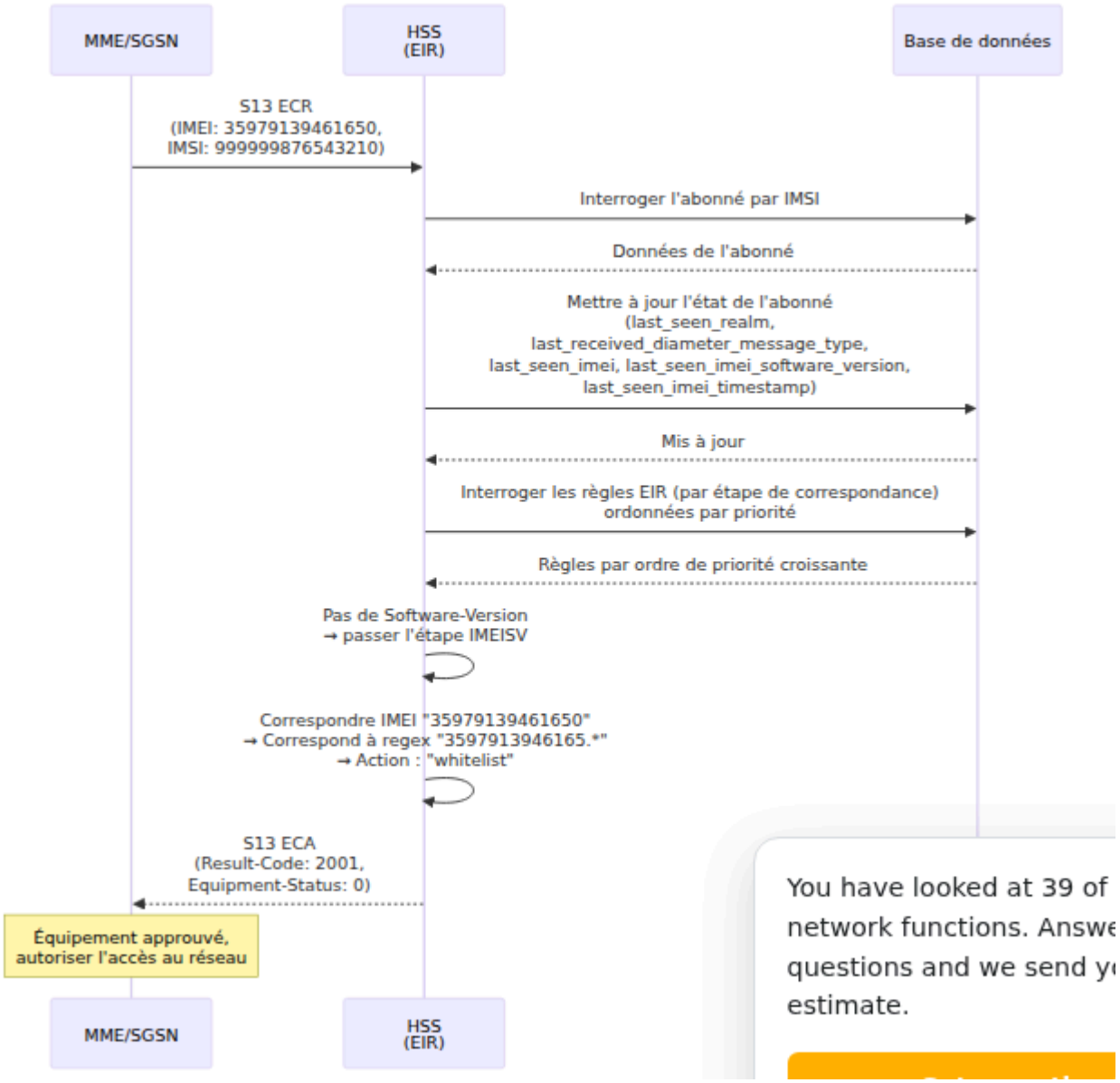
Modèle	Correspond à	Cas d'utilisation
`"35979139461650"`	IMEI exact	Liste blanche/liste noire d'un seul appareil
`"3597913946165.*"`	Wildcard de préfixe IMEI	Gamme de fabricants/modèles
`"3597913946161108"`	IMEISV exact	Appareil spécifique avec version logicielle
`"999999876543210"`	IMSI	Bloquer tout l'équipement pour l'abonné
`"359791.*"`	Wildcard TAC	Allocation de type d'appareil entier

Flux de messages courants

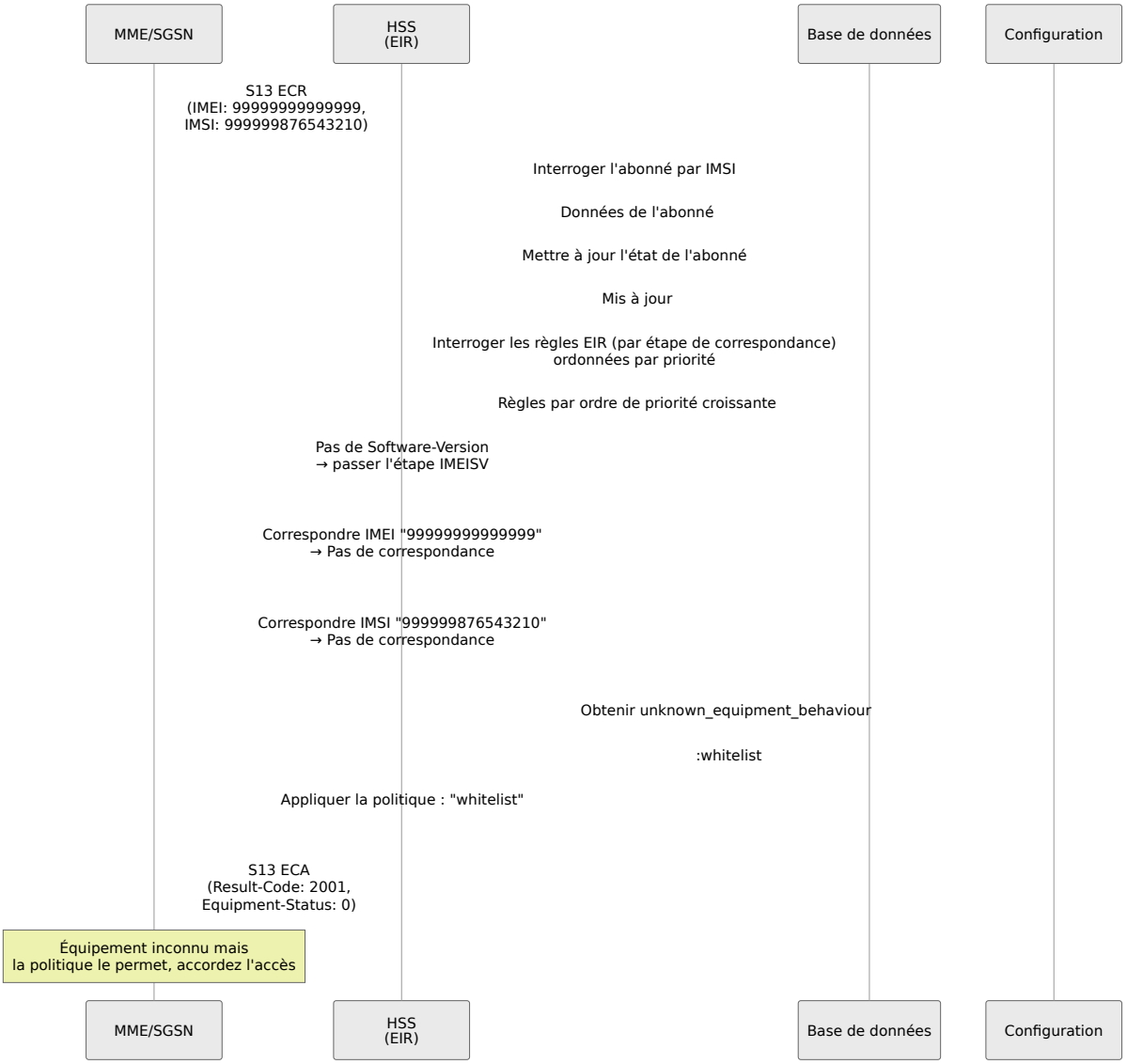
Flux 1 : Vérification de l'équipement - IMEI connu sur liste blanche



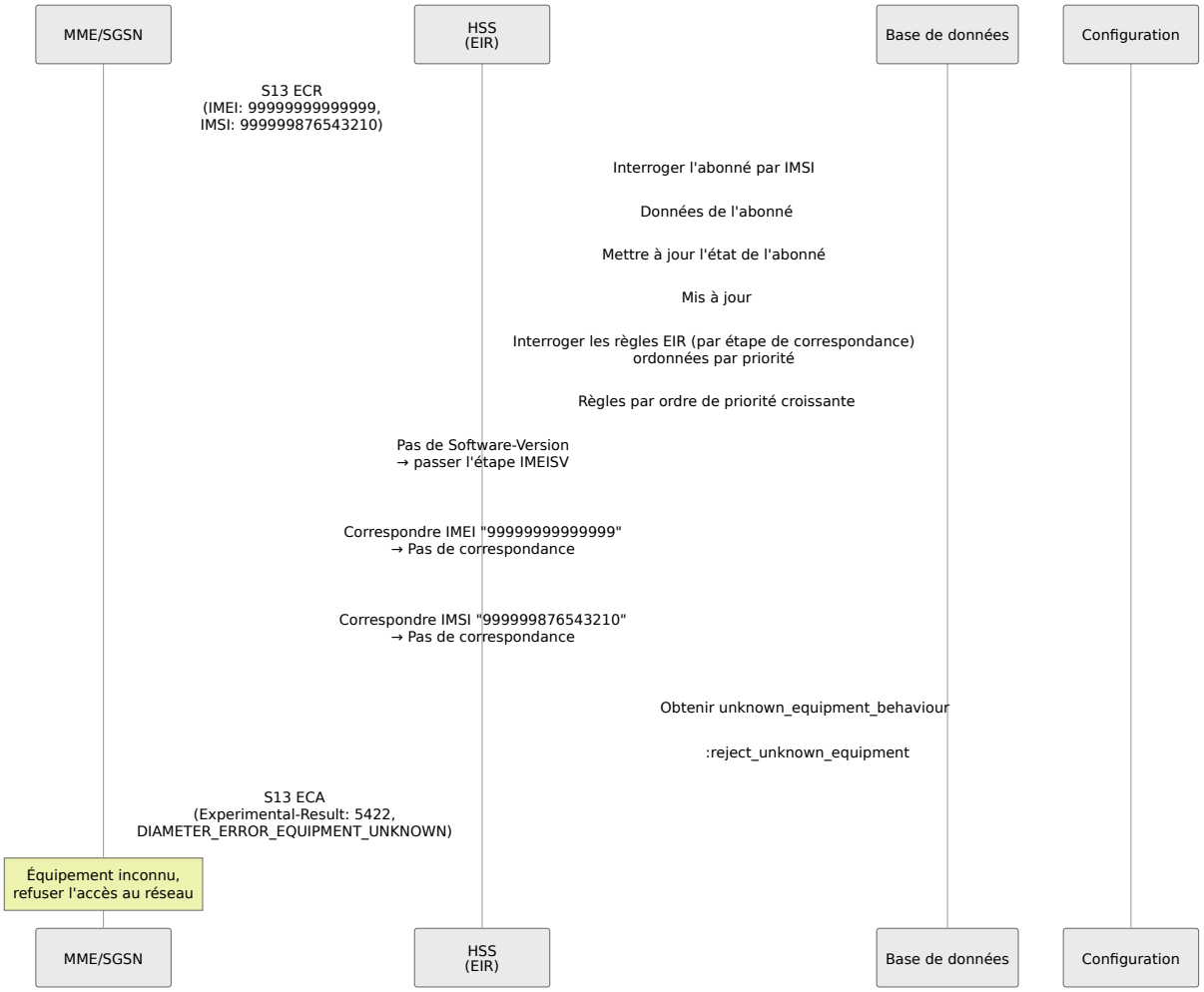
Flux 2 : Vérification de l'équipement - IMEI sur liste noire (Appareil volé)



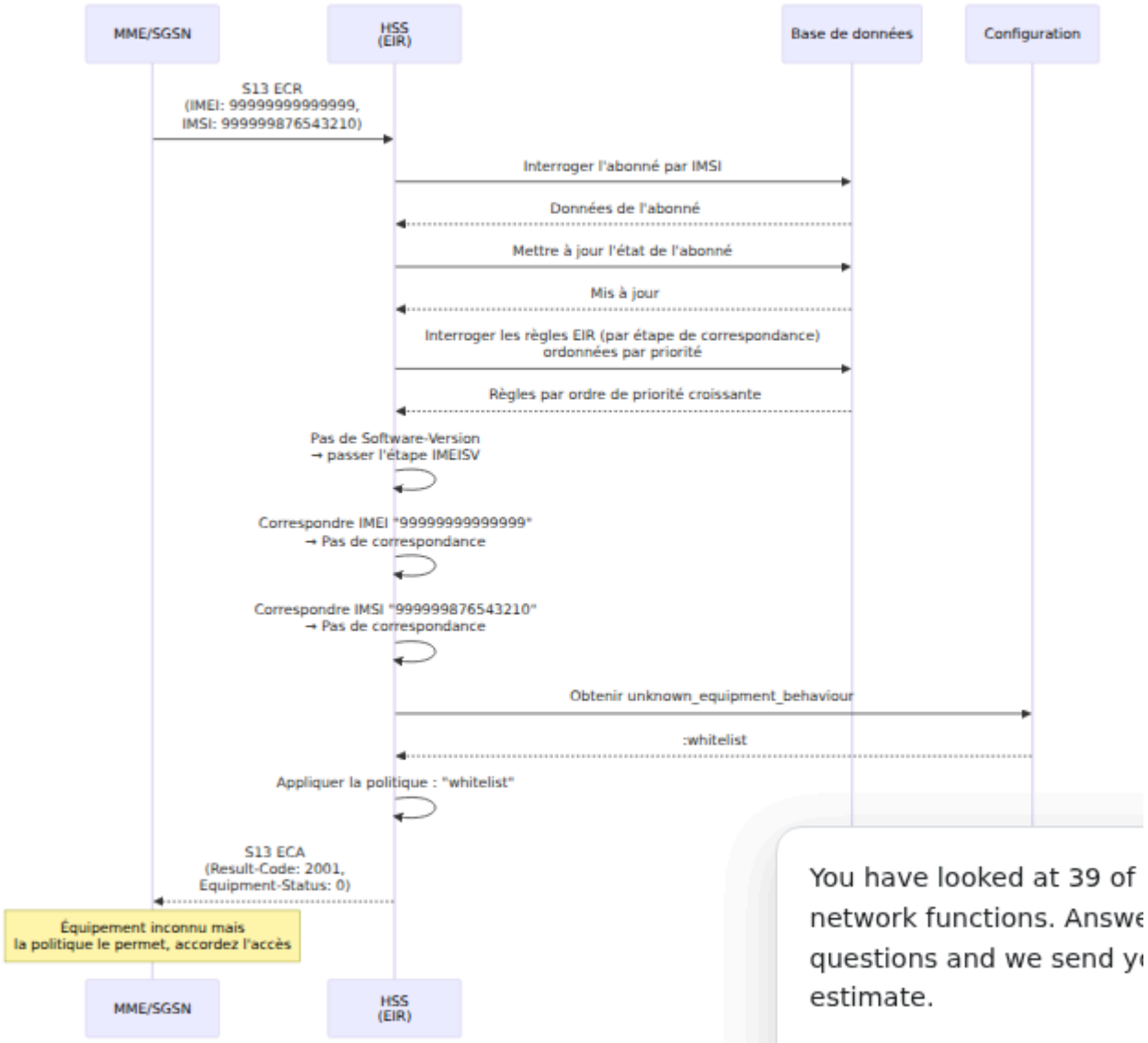
Flux 3 : Vérification de l'équipement - Équipement inconnu (Politique de liste blanche)



Flux 4 : Vérification de l'équipement - Équipement inconnu (Politique de rejet)

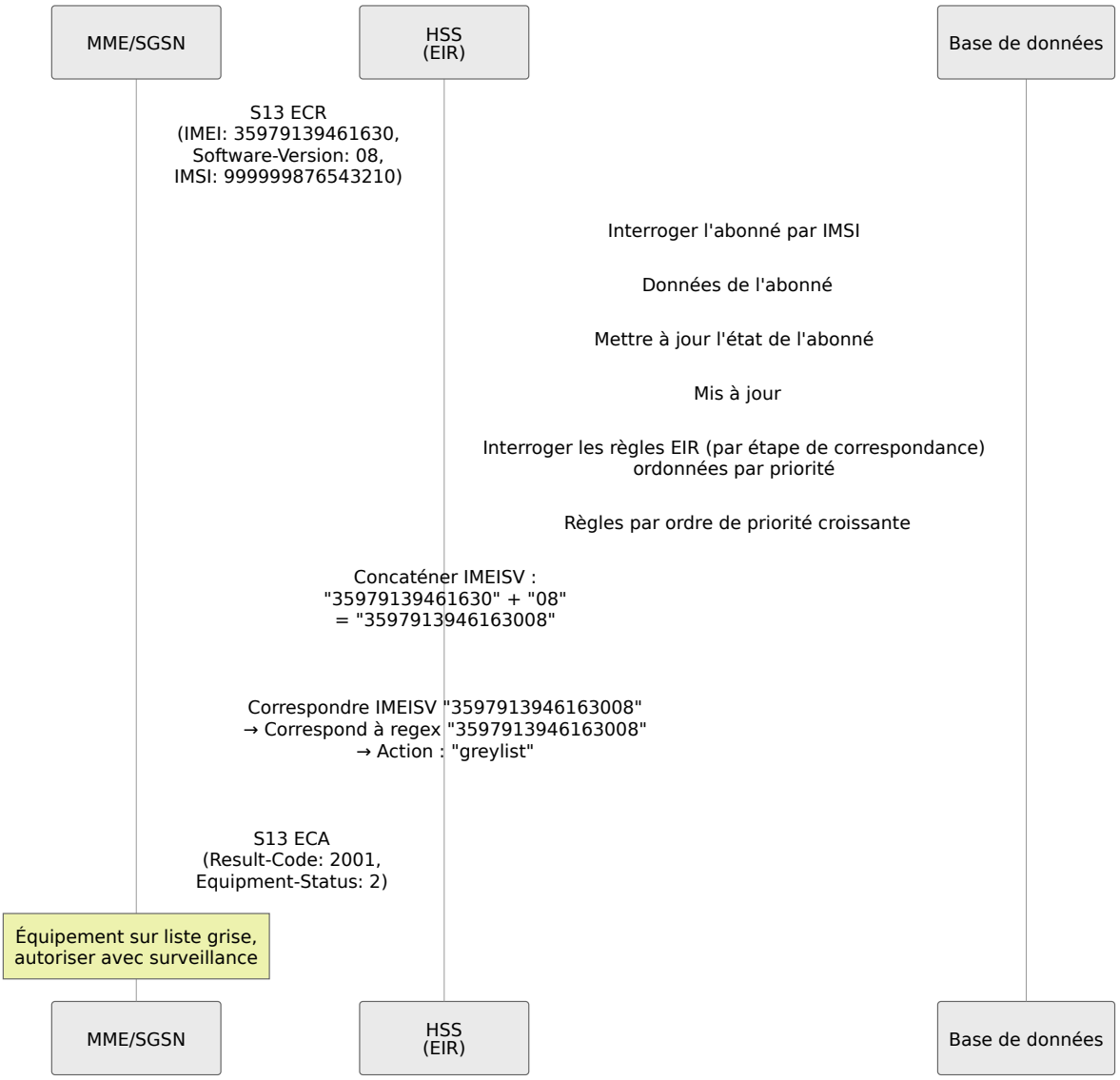


Flux 5 : Vérification de l'équipement - Abonné inconnu ou désactivé

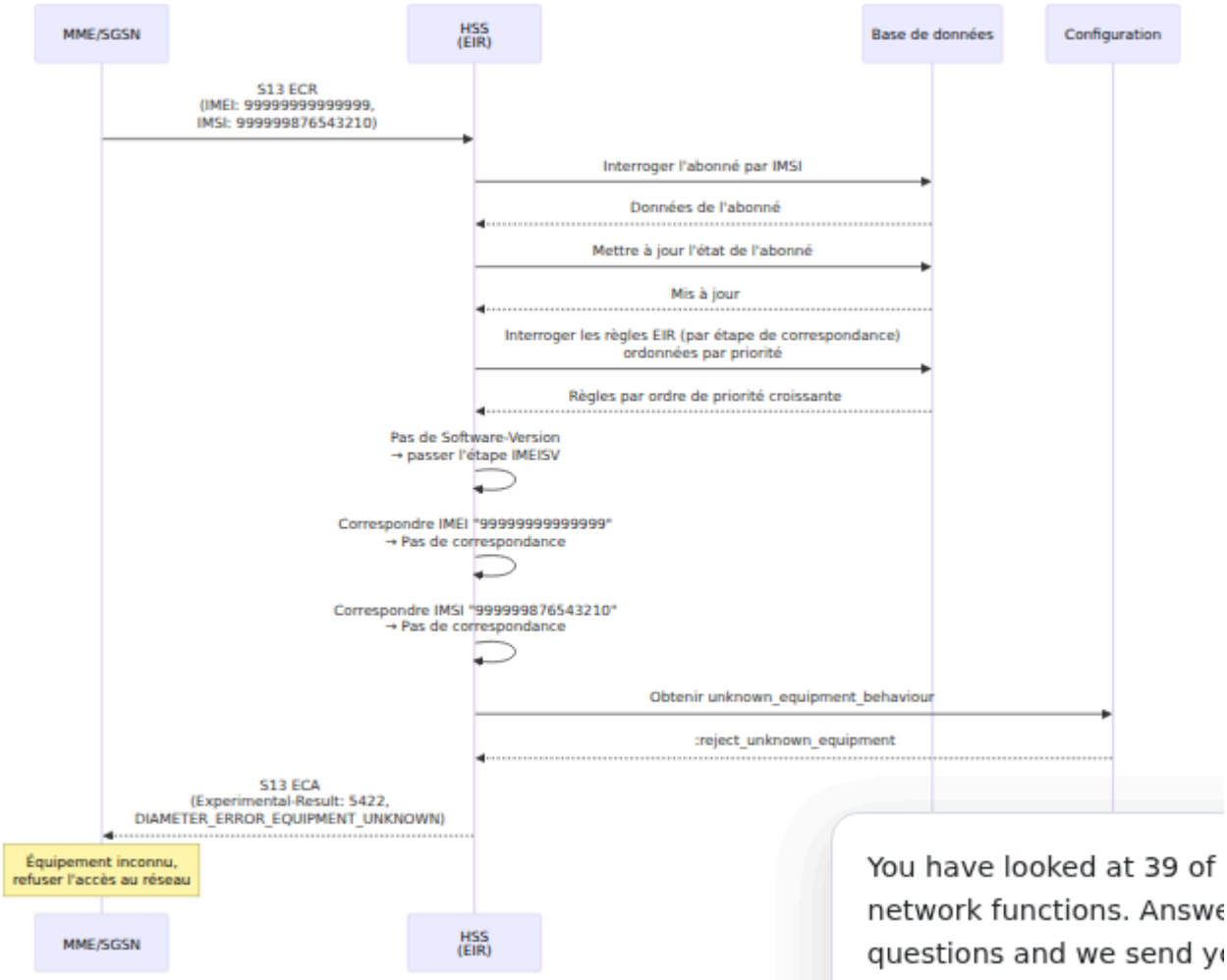


You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

Flux 6 : Vérification de l'équipement - Correspondance IMEISV



Flux 7 : Vérification de l'équipement - Blocage IMSI



API REST

Gestion des règles EIR

Chemin de base : `/api/eir/rule`

Les corps de demande sont des objets JSON plats avec les champs de règle au niveau supérieur (voir Format de demande). Les réponses utilisent l'enveloppe standard décrite dans la Référence API. Chaque règle porte `id`, `action`, `regex` et `precedence`.

Lister toutes les règles EIR

Demande :

```
GET /api/eir/rule
```

Réponse (HTTP 200) :

```
{
  "status": "success",
  "response": [
    {
      "id": "e7f6a5b4-c3d2-4e1f-8a9b-0c1d2e3f4a5b",
      "action": "whitelist",
      "regex": "3597913946165.*",
      "precedence": 2
    },
    {
      "id": "f8a7b6c5-d4e3-4f2a-9b0c-1d2e3f4a5b6c",
      "action": "blacklist",
      "regex": "35979139461640",
      "precedence": 1
    }
  ]
}
```

La liste est ordonnée par `id`, pas par priorité. Triez sur `precedence` pour voir l'ordre d'évaluation.

Obtenir une règle EIR spécifique

Demande :

```
GET /api/eir/rule/{id}
```

Réponse (HTTP 200) :

```
{
  "status": "success",
  "response": {
    "id": "e7f6a5b4-c3d2-4e1f-8a9b-0c1d2e3f4a5b",
    "action": "whitelist",
    "regex": "3597913946165.*",
    "precedence": 2
  }
}
```


Créer une règle EIR

Demande :

```
POST /api/eir/rule
Content-Type: application/json

{
  "action": "blacklist",
  "regex": "35979139461640",
  "precedence": 1
}
```

Validation :

- `action` : Requis, doit être "whitelist", "blacklist" ou "greylist"
- `regex` : Requis, doit être un modèle regex valide
- `precedence` : Entier optionnel de 1 à 65535. Omettez-le pour ajouter la règle après la priorité la plus élevée actuelle. Fournir une valeur insère la règle à cette position et déplace chaque règle avec une priorité égale ou supérieure d'un cran. Voir Précédence des règles.

Réponse (HTTP 201) :

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "blacklist",
    "regex": "35979139461640",
    "precedence": 1
  }
}
```


Réponse d'erreur (HTTP 422) :


```
{
  "status": "error",
  "response": {
    "invalid_fields": {
      "precedence": "must be greater than or equal to 1"
    }
  }
}
```


Mettre à jour une règle EIR (partielle)

Demande :

```
PATCH /api/eir/rule/{id}
Content-Type: application/json

{
  "action": "greylist"
}
```

Réponse (HTTP 200) :

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "greylist",
    "regex": "35979139461640",
    "precedence": 1
  }
}
```

Changer `precedence` déplace la règle à cette position et déplace uniquement les règles entre ses anciennes et nouvelles positions. Omettre `precedence`, ou envoyer sa valeur actuelle, laisse l'ordre inchangé. Ne pas envoyer `null` : cela efface la priorité de la règle et la règle est alors évaluée après chaque règle numérotée.

Mettre à jour une règle EIR (PUT)

Demande :

```
PUT /api/eir/rule/{id}
Content-Type: application/json

{
  "action": "whitelist",
  "regex": "359791394616.*",
  "precedence": 3
}
```

Réponse (HTTP 200) :

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "whitelist",
    "regex": "359791394616.*",
    "precedence": 3
  }
}
```

Un PUT applique les champs présents dans le corps, comme un PATCH ; les champs omis conservent leurs valeurs actuelles. Un PUT à un id qui n'existe pas crée la règle avec cet id, suivant le même placement de priorité qu'un POST.

Supprimer une règle EIR

Demande :

```
DELETE /api/eir/rule/{id}
```

Réponse (HTTP 200) :

```
{
  "status": "success",
  "response": ""
}
```

Supprimer une règle laisse un vide dans la numérotation de priorité. Les règles restantes conservent leur ordre relatif.

Configuration

Configuration du service Diameter

Application S13 (`config/runtime.exs`) :

```
%{
  application_name: :s13,
  application_dictionary: :diameter_gen_3gpp_s13,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_252}
  ]
}
```

Comportement des équipements inconnus

Configurez le comportement par défaut pour les équipements ne correspondant à aucune règle dans `config/runtime.exs` :

Exemple :

```
config :hss, :eir,
  unknown_equipment_behaviour: :whitelist
```

Valeurs valides :

- `:whitelist` - Autoriser les équipements inconnus (par défaut, permissif)
- `:blacklist` - Bloquer les équipements inconnus (restrictif)
- `:greylist` - Surveiller les équipements inconnus (modéré)
- `:reject_unknown_equipment` - Retourner l'erreur Diameter 5422 (strict)

Cas d'utilisation :

- **Développement/Test** : `:whitelist` - Autoriser tous les appareils
- **Production (permissif)** : `:whitelist` - Bloquer uniquement les appareils connus comme mauvais
- **Production (modéré)** : `:greylist` - Journaliser les appareils inconnus pour examen
- **Production (strict)** : `:reject_unknown_equipment` - Autoriser uniquement les appareils enregistrés

Gestion des erreurs

Code de résultat	Type	Signification	C
2001	Succès	DIAMETER_SUCCESS	Vérif de l'équ term
5422	Expérimental	DIAMETER_ERROR_EQUIPMENT_UNKNOWN	Abor trou équi inco rejet
5012	Expérimental	DIAMETER_UNABLE_TO_COMPLY	Erre trait

Cas d'utilisation

1. Gestion des appareils volés

Scénario : Appareil signalé volé

Action :

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "35979139461640"  # IMEI exact
}
```

Résultat : Appareil refusé l'accès au réseau lors de la prochaine attache

2. Liste blanche des fabricants

Scénario : Pré-approuver toute la gamme de modèles d'appareils

Action :

```
POST /api/eir/rule
{
  "action": "whitelist",
  "regex": "359791394.*"  # TAC pour fabricant/modèle
}
```

Résultat : Tous les appareils dans la plage TAC approuvés

3. Verrouillage de l'équipement de l'abonné

Scénario : Bloquer tout l'équipement pour un abonné spécifique (verrouillage SIM)

Action :

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "999999876543210"  # IMSI
}
```

Résultat : Tout équipement utilisé avec cette SIM est bloqué

4. Liste grise des équipements de test

Scénario : Surveiller les équipements de test en production

Action :

```
POST /api/eir/rule
{
  "action": "greylist",
  "regex": "35979139.*"  # Plage TAC des équipements de test
}
```

Résultat : Équipement autorisé mais signalé pour surveillance

5. Contrôle de version logicielle

Scénario : Bloquer une version de firmware vulnérable spécifique

Action :

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "359791394616.*05"  # Plage IMEI + Version logicielle
05
}
```

Résultat : Seuls les appareils avec la version logicielle "05" dans la plage IMEI sont bloqués

Détails de mise en œuvre

Composants internes

La fonctionnalité EIR est mise en œuvre à l'aide de plusieurs modules internes :

- **Gestionnaire de protocole S13** - Traitement des messages ECR/ECA
- **Moteur de correspondance d'équipement** - Correspondance IMEI/IMEISV/IMSI basée sur Regex dans l'ordre de priorité
- **Placement de priorité** - Logique d'insertion à la position qui déplace les règles EIR voisines lorsqu'une règle est créée à, ou déplacée à, une priorité

explicite. Les règles de facturation partagent uniquement la vérification des limites de 1 à 65535 et ne sont pas renumérotées
- **Base de données des règles EIR** - Stockage et recherche de motifs
- **Contrôleur d'API REST** - Points de terminaison de gestion des règles

Fonction de Recherche de Statut d'Équipement

La recherche de statut d'équipement suit cette logique en cascade :

1. **Correspondance IMEISV** : Vérifiez l'IMEI + la version du logiciel concaténés
2. **Correspondance IMEI** : Vérifiez uniquement l'IMEI
3. **Correspondance IMSI** : Vérifiez l'IMSI (si fourni)
4. **Équipement Inconnu** : Appliquez la politique par défaut configurée

Résultats Possibles :

- `whitelist` - Équipement autorisé
- `blacklist` - Équipement bloqué
- `greylist` - Équipement sous observation
- `reject_unknown_equipment` - Rejet strict

Considérations de Sécurité

Confidentialité de l'IMEI

Les numéros IMEI sont des identifiants sensibles. L'EIR :

- Ne consigne pas les valeurs IMEI en texte clair par défaut
- Sert l'API REST sans authentification à moins que l'authentification par clé partagée optionnelle ne soit configurée (`config :api_ex, api: %{auth: %{key: "..."}}`, présentée par les clients dans l'en-tête `x-api-key`). Activez-le partout où `/api/eir/rule` est accessible depuis des réseaux non fiables

Ordre des Règles

Les règles sont évaluées dans l'ordre croissant de `precedence` à chaque étape de correspondance, et la première correspondance l'emporte. Lorsque les motifs se chevauchent, donnez à la règle spécifique une priorité inférieure :

```
Précédence 1 : regex "35979139461640" action "blacklist"
(spécifique)
Précédence 2 : regex "359791.*" action "whitelist"  (large)
```

Créer la règle spécifique avec `"precedence": 1` alors que la règle large détient déjà la priorité 1 déplace automatiquement la règle large à 2. Voir Ordre des Règles pour le comportement complet de placement.

Limitation de Taux

Envisagez de mettre en œuvre une limitation de taux sur :

- Les requêtes S13 ECR provenant de pairs non fiables
- Les modifications de règles EIR de l'API REST
- Les requêtes de recherche IMEI pour prévenir les attaques d'énumération

Documentation Connexe

- Protocoles Diameter - Spécification du protocole S13
- Référence API - Documentation complète de l'API
- Architecture - Architecture globale du HSS
- Guide des Opérations - Procédures opérationnelles

Annexe : Structure de l'IMEI

Format IMEI (15 chiffres)

```
35 9791 394616 1
│  │    │      └─ Chiffre de contrôle (algorithme de Luhn)
│  │    └─ Numéro de série (6 chiffres)
│  └─ FAC (Code d'assemblage final, 4 chiffres)
└┘─ TAC (Code d'allocation de type, 8 chiffres au total y compris RBI)
   │  └─ RBI (Identifiant de l'organisme de rapport, 2 chiffres)
   └─ Fabricant/Modèle (6 chiffres)
```

Format IMEISV (16 chiffres)

```
35 9791 394616 1 08
│  │    │      │ └─ Version du logiciel (2 chiffres)
└─ IMEI (15 chiffres)
```

Exemples de Motifs

IMEI/IMEISV	Motif	Correspondances
359791394616108	`3597913946161.*`	Tous les appareils avec TAC+FAC+Numéro de série 359791394616*
359791394616140	`35979139461614.`	Tous les chiffres de contrôle pour le numéro de série 359791394616141-9
35979139461640	`35979139461640`	Correspondance exacte de l'IMEI
3597913946163008	`3597913946163008`	Correspondance exacte de l'IMEISV (IMEI + SV)

PCRF (Fonction de Règles de Politique et de Facturation)

Vue d'ensemble

Le HSS comprend un PCRF (Fonction de Règles de Politique et de Facturation) intégré qui fournit un contrôle de politique et des règles de facturation pour les sessions de données mobiles. Le PCRF contrôle les politiques de Qualité de Service (QoS), l'allocation de bande passante et les règles de facturation pour les porteuses par défaut et dédiées dans les réseaux LTE.

Capacités clés

- **Interface Gx** : Contrôle de politique pour PGW/PCEF (Passerelle de Données de Paquet / Fonction d'Application de Politique et de Facturation)
- **Interface Rx** : Autorisation et QoS pour les flux multimédias IMS (Système Multimédia IP)
- **Interface Sy** : Contrôle de la limite de dépenses des abonnés contre un OCS externe - voir Interface Sy
- **Gestion Dynamique des Politiques** : Mises à jour de politique en temps réel via des Requêtes de Ré-authentification (RAR)
- **Support VoLTE** : Création de porteuses dédiées pour les appels vocaux avec QoS garantie
- **Règles de Facturation** : Définir le comportement de facturation et les profils de vitesse en utilisant des Modèles de Flux de Trafic (TFT)
- **API REST** : Contrôle programmatique de l'application des politiques et de la gestion des règles

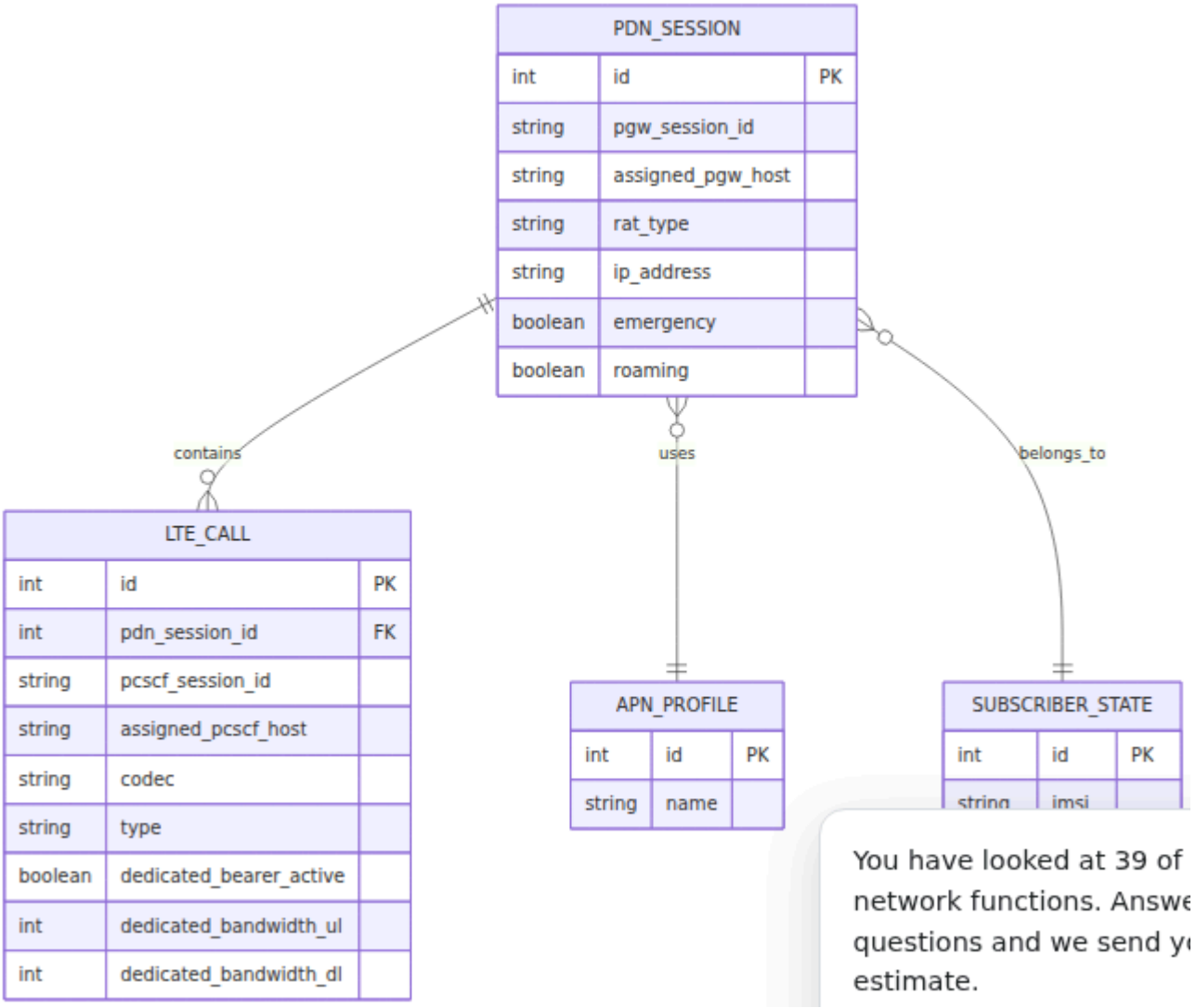
Architecture

Interfaces Diameter

Interface	ID d'application	Pair	But
Gx	16,777,238	PGW (PCEF)	Gestion de session PDN, application de QoS, règles de facturation
Rx	16,777,236	P-CSCF (AF)	Autorisation multimédia IMS, réservation de bande passante
Sy	16,777,302	OCS	Statut de la limite de dépenses des abonnés, politique basée sur les dépenses - voir Interface Sy

Gestion de l'état de session

Le PCRF maintient l'état de session pour les connexions PDN actives et les appels VoLTE :



Interface Gx

Opérations prises en charge

1. Requête de Contrôle de Crédit - Initiale (CCR-I)

Déclencheur : PGW crée une nouvelle connexion PDN pour l'abonné

AVPs de Requête :

- Session-Id
- Origin-Host, Origin-Realm
- Subscription-Id (contient l'IMSI)
- Called-Station-Id (nom de l'APN)

- IP-CAN-Type (type de réseau d'accès IP)
- RAT-Type (Technologie d'Accès Radio)
- Framed-IP-Address (adresse IP de l'UE)

Actions du PCRF :

1. Rechercher l'abonné par IMSI
2. Récupérer le profil APN et la configuration QoS
3. Créer une entrée de suivi de session
4. Construire des politiques QoS à partir du profil APN

AVPs de Réponse :

- Result-Code: 2001 (DIAMETER_SUCCESS)
- QoS-Information (limites de bande passante agrégées APN)
- Default-EPS-Bearer-QoS (QCI, ARP, priorité)
- Bearer-Control-Mode
- En ligne / Hors ligne (méthode de facturation par défaut au niveau de la session - voir Facturation en ligne/hors ligne au niveau de la session)

2. Requête de Contrôle de Crédit - Mise à jour (CCR-U)

Déclencheur : PGW signale des changements de session (mise à jour de localisation, changement de RAT, etc.)

Actions du PCRF :

1. Localiser la session existante par ID de session
2. Mettre à jour les paramètres de session (type de RAT, localisation, etc.)
3. Retourner les politiques mises à jour si nécessaire

Réponse : Result-Code 2001 avec des mises à jour de politique optionnelles

3. Requête de Contrôle de Crédit - Terminer (CCR-T)

Déclencheur : PGW termine la connexion PDN

Actions du PCRF :

1. Localiser la session par ID de session
2. Supprimer la session et les enregistrements d'appel associés
3. Confirmer la terminaison

Réponse : Result-Code 2001

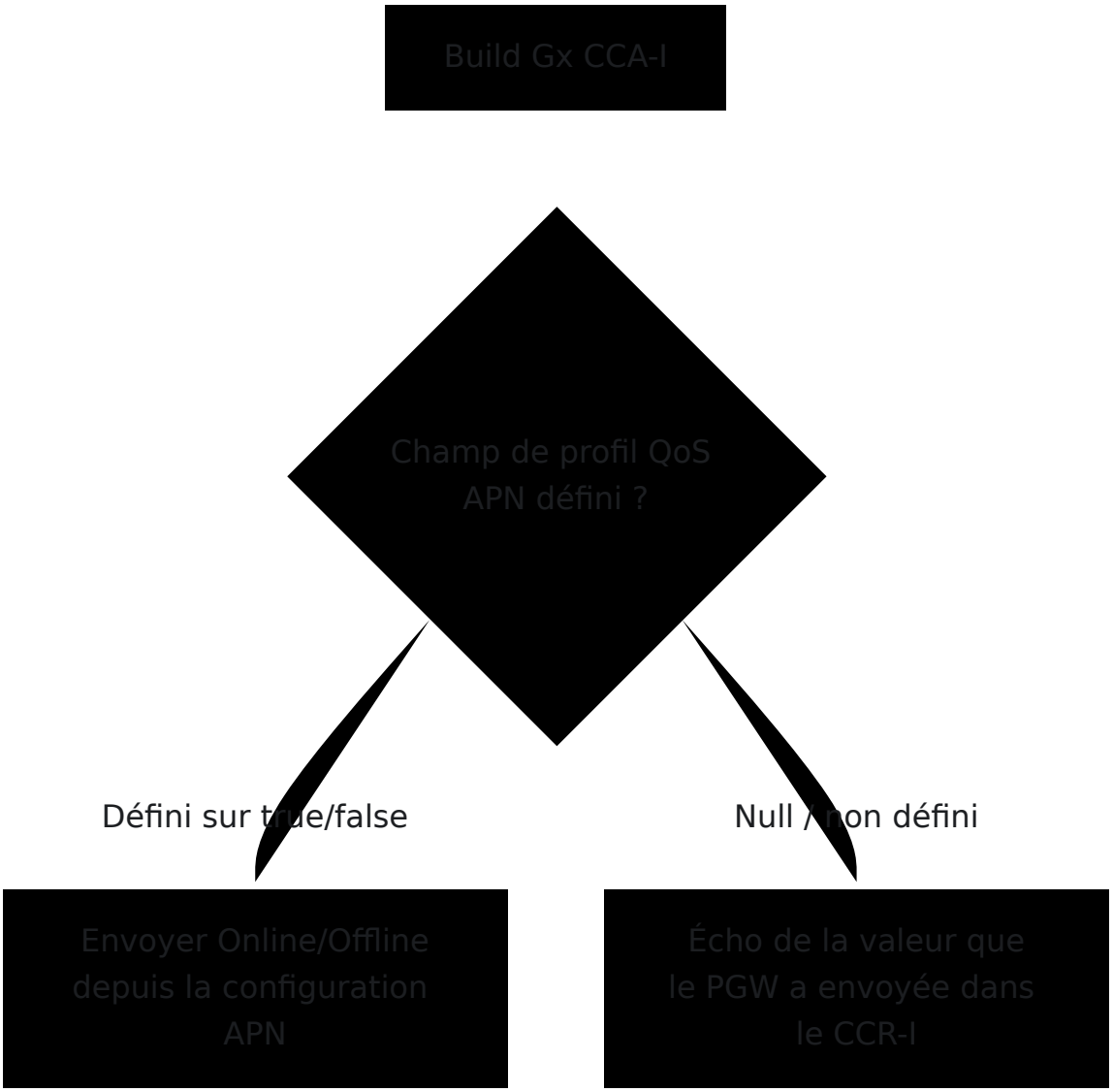
Facturation en ligne/hors ligne au niveau de la session

Les AVPs `Online` (AVP 1009) et `Offline` (AVP 1008) envoyés au niveau supérieur d'un Gx CCA-I définissent **la méthode de facturation par défaut pour l'ensemble de la session IP-CAN** (la porteuse par défaut), comme défini dans 3GPP TS 29.212. Cela est distinct de l'`Online`/`Offline` par règle porté *à l'intérieur* d'une Définition de Règle de Facturation, qui ne régit que le trafic correspondant à cette règle spécifique.

Par défaut, le PCRF **répète** les valeurs que le PGW/PCEF a annoncées dans le CCR-I. Le profil QoS APN peut plutôt **fixer** la méthode de facturation au niveau de la session, remplaçant l'écho. Cela est contrôlé par deux champs optionnels sur le profil QoS APN :

Champ	Type	AVP	Effet
`online_charging_enabled`	Boolean	`Online` (1009)	`true` → `ENABLE_ONLINE (1)`, `false` → `DISABLE_ONLINE (0)`
`offline_charging_enabled`	Boolean	`Offline` (1008)	`true` → `ENABLE_OFFLINE (1)`, `false` → `DISABLE_OFFLINE (0)`

Chaque champ est résolu indépendamment :



Valeurs AVP de Méthode de Facturation (TS 29.212):

AVP	Code	Valeur	Nom	Signification
`Online`	1009	0	DISABLE_ONLINE	La facturation en ligne (OCS / Gy) n'est pas appliquée à la session
`Online`	1009	1	ENABLE_ONLINE	La facturation en ligne est appliquée - le crédit doit être accordé avant que le trafic ne circule
`Offline`	1008	0	DISABLE_OFFLINE	La facturation hors ligne (CDR / Gz) n'est pas appliquée à la session
`Offline`	1008	1	ENABLE_OFFLINE	La facturation hors ligne est appliquée - l'utilisation est enregistrée pour la facturation

Comment cela fonctionne : Lorsque `online_charging_enabled`/`offline_charging_enabled` sont laissés non définis (la valeur par défaut pour chaque profil existant), le comportement reste inchangé - le PCRF reflète la demande du PGW, donc la posture de facturation existante du réseau est préservée. Définir l'un ou l'autre champ impose ce mode de facturation sur chaque CCR-I pour tout APN utilisant le profil, indépendamment de ce que le PGW annonce.

Cas d'utilisation : Fixer la facturation en ligne pour un APN prépayé (de sorte que le PGW doive consulter l'OCS même s'il a demandé hors ligne), ou forcer la facturation hors ligne sur un APN postpayé/IoT pour garantir la génération de CDR. Comme le paramètre se trouve sur le profil QoS APN, il s'applique uniformément à chaque APN qui référence ce profil.

> **Remarque :** Cela contrôle uniquement le *défaut de session*. La facturation par trafic est toujours régie par les champs

> `online_charging_enabled`/`offline_charging_enabled` sur des Règles de Facturation individuelles, qui apparaissent à l'intérieur des AVPs de Définition de Règle de Facturation.

4. Requête de Ré-authentification (RAR)

Direction : PCRF → PGW (HSS initie)

Déclencheur :

- Configuration d'appel IMS (Rx AAR déclenche Gx RAR)
- Fin d'appel IMS (Rx STR déclenche Gx RAR)
- Ré-auth manuelle via l'API REST

AVPs RAR :

- Session-Id (ID de session PGW)
- Auth-Application-Id : 16,777,238
- Re-Auth-Request-Type (0 = Autoriser seulement)
- Charging-Rule-Install/Remove
- QoS-Information (pour les porteuses dédiées)

Actions PGW : Créer/modifier/supprimer des porteuses dédiées en fonction des règles de facturation

Règles de Facturation et Modèles de Flux de Trafic

Le PCRF prend en charge la définition de règles de facturation avec des Modèles de Flux de Trafic (TFT) pour contrôler :

- **Facturation spécifique au service** - Tarifs différents pour la vidéo, les jeux, les réseaux sociaux, etc.
- **Profils de vitesse** - Limiter ou prioriser le trafic correspondant à des modèles spécifiques
- **Politiques basées sur l'utilisation** - Appliquer différentes QoS en fonction du type de trafic ou de la destination

Les règles de facturation peuvent être :

- Installées dynamiquement via Gx RAR en fonction de la détection d'application
- Prédéfinies et déclenchées par des conditions spécifiques (heure de la journée, localisation, quota)
- Associées à des TFT en utilisant des règles de filtre de paquets (5-tuple : protocole, IP source/destination, port source/destination)

Cas d'utilisation courants :

- **Zero-rating** - Accès illimité à des services spécifiques (Spotify, WhatsApp, Facebook) sans consommer de quota de données
- **Accès post-quota** - Autoriser le portail d'auto-assistance et les sites de support même après que l'abonné ait épuisé son allocation de données
- **Vitesse par paliers** - Haute vitesse pour les services premium, limitée pour le contenu standard
- **Politiques basées sur le temps** - Streaming illimité hors pointe, priorisation en période de pointe
- **Politiques de roaming** - Facturation différente pour l'utilisation de données internationale par rapport à nationale
- **SLA d'entreprise** - QoS garantie pour les applications critiques pour les entreprises

Structure de la Politique QoS

QoS de la Porteuse par Défaut (à partir du profil APN) :

```
{
  "QoS-Class-Identifier": 9,          // QCI (9 = porteuse par
défaut)
  "APN-Aggregate-Max-Bitrate-UL": 50000,  // kbps
  "APN-Aggregate-Max-Bitrate-DL": 100000, // kbps
  "Allocation-Retention-Priority": {
    "Priority-Level": 8,
    "Pre-emption-Capability": 1,      // Peut préempter
    "Pre-emption-Vulnerability": 1    // Peut être préempté
  }
}
```

QoS de la Porteuse Dédiée (pour VoLTE) :

```
{
  "QoS-Class-Identifier": 1,          // QCI 1 = Voix
Conversationnelle
  "Max-Requested-Bandwidth-UL": 128000,   // bps
  "Max-Requested-Bandwidth-DL": 128000,   // bps
  "Guaranteed-Bitrate-UL": 128000,
  "Guaranteed-Bitrate-DL": 128000
}
```

Interface Rx

Opérations prises en charge

1. Requête AA (AAR) / Réponse AA (AAA)

Déclencheur : P-CSCF demande une autorisation pour la session multimédia IMS (configuration d'appel VoLTE)

AVPs de Requête :

- Session-Id (identifiant de session P-CSCF)
- Subscription-Id (IMSI ou URI SIP)
- Media-Component-Description

    - Media-Type (audio, vidéo)
    - Max-Requested-Bandwidth-UL/DL
    - Codec-Data
    - Flow-Description (filtres de paquets 5-tuple)
- AF-Application-Identifier

Actions du PCRF :

1. Rechercher l'abonné par IMSI ou URI SIP
2. Trouver la session IMS active
3. Extraire les paramètres multimédias (codec, bande passante, règles de flux)
4. Créer une entrée de suivi d'appel
5. **Déclencher Gx RAR vers PGW** pour créer une porteuse dédiée
6. Attendre la réponse Gx RAA
7. Retourner Rx AAA avec le résultat de l'autorisation

AVPs de Réponse :

- Result-Code: 2001 (succès) ou 5063 (service non autorisé)

Appels d'urgence (SOS), y compris les appelants non authentifiés / sans SIM

Les appels d'urgence (`urn:service:sos`) sont autorisés même lorsque l'appelant n'est pas un abonné connu (un appareil sans SIM, ou un abonné en roaming sans identifiants enregistrés). Cela est nécessaire pour que les appels d'urgence obtiennent toujours la bonne QoS (TS 23.167).

Comment le PCRF reconnaît et autorise une AAR d'urgence :

1. **Détection d'urgence.** La demande est considérée comme une urgence si **soit** le `Subscription-Id` est une URI SIP contenant `sos`, **soit** l'`Framed-IP-Address` de l'AAR résout à une session PDN qui a déjà été créée comme une PDN d'urgence par le Gx CCR-I (le côté Gx signale l'urgence par l'APN / `Called-Station-Id` contenant `sos`). Le deuxième chemin est ce qui permet à une INVITE d'urgence **anonyme** (sans identité utilisable) d'être toujours autorisée.

2. **Résolution de l'abonné.** Recherché par `Subscription-Id`, sinon par `Framed-IP-Address`. Une PDN **orpheline d'urgence** (créée par Gx pour un abonné inconnu) n'a pas d'abonné (`subscriber_state_id: nil`) ; l'AAR prend alors le chemin orphelin plutôt que le chemin d'abonné normal.
3. **Profil APN SOS.** La QoS de la porteuse dédiée est prise à partir du profil APN SOS (voir "Profil APN SOS par défaut intégré" ci-dessous).
4. Un Gx RAR installe la porteuse vocale dédiée (QCI 1) sur la PDN (orpheline), et le AAA retourne `2001`.

> Remarque : un `Framed-IP-Address` ne donne une autorisation que s'il correspond à une PDN que Gx a déjà créée **comme urgence**. Une demande non urgente d'un UE non enregistré est toujours rejetée au P-CSCF (elle n'atteint jamais le PCRF en tant qu'AAR d'urgence).

Profil APN SOS par défaut intégré. L'autorisation d'urgence ne dépend plus d'un opérateur ayant préalablement provisionné un profil APN `sos` dans la base de données. Le PCRF résout le profil APN SOS comme :

- un profil APN défini par l'opérateur dont l'identifiant APN est `sos`, s'il existe ; sinon
- un **profil APN SOS par défaut intégré** fourni par OmniHSS.

Le par défaut intégré est un profil transitoire (pas de ligne DB, `apn_profile_id: nil` sur la PDN orpheline - la colonne est nullable) avec une QoS d'urgence :

Paramètre	Valeur
QCI de la porteuse SOS par défaut	5
QCI de la porteuse vocale dédiée	1 (priorité ARP 2)
Priorité d'allocation-rétention	1
Capacité de préemption	activée
Vulnérabilité à la préemption	désactivée
APN-AMBR UL/DL	10000 kbps

Pour remplacer ces valeurs (ou pour définir des règles de facturation par opérateur via un profil PCRF), créez un profil APN `sos` dans la base de données - il prend le pas sur le par défaut intégré.

2. Requête de Terminaison de Session (STR) / Réponse de Terminaison de Session (STA)

Déclencheur : P-CSCF termine la session IMS (fin d'appel)

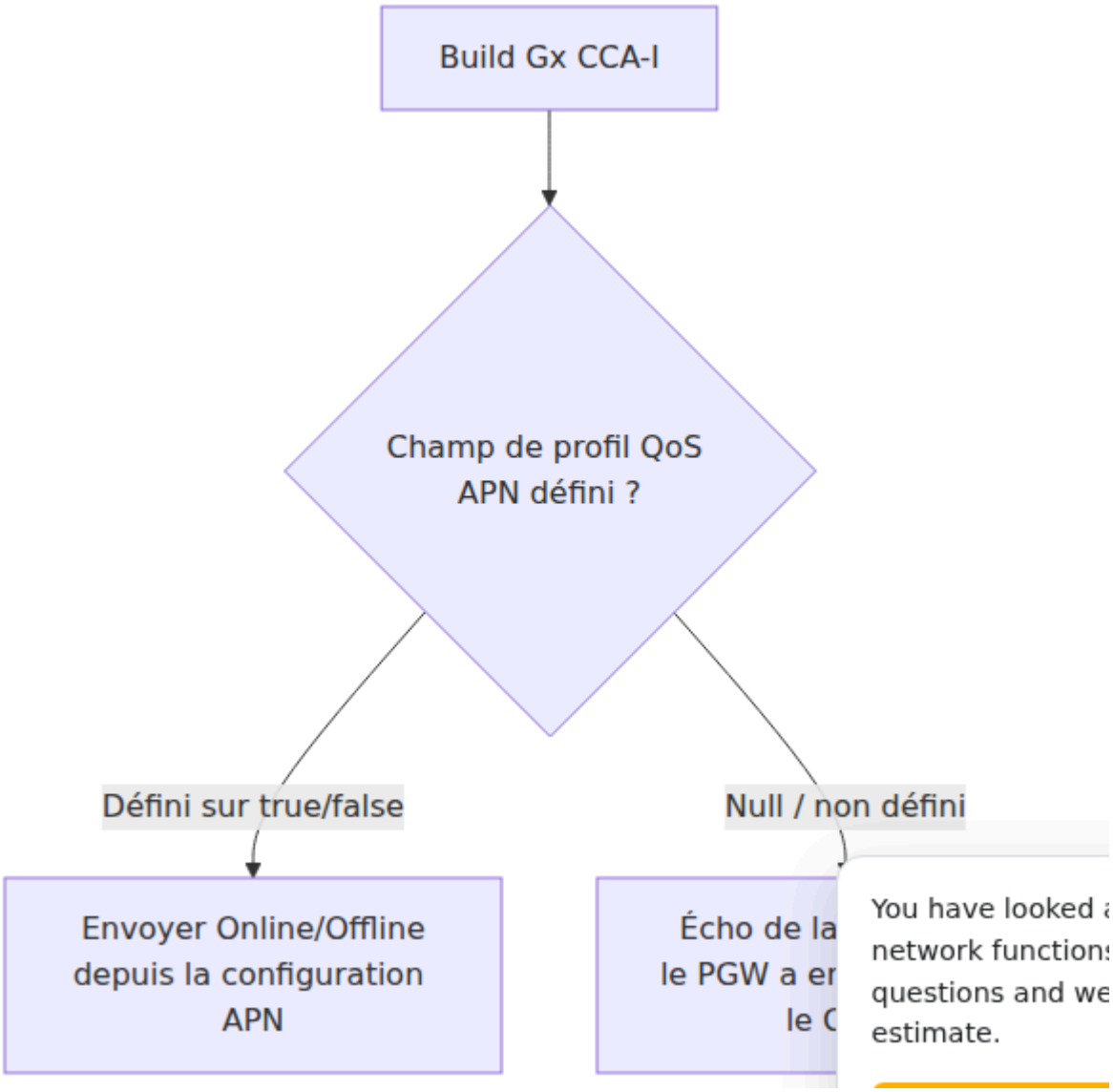
Actions du PCRF :

1. Localiser la session d'appel par ID de session P-CSCF
2. **Déclencher Gx RAR vers PGW** pour supprimer la porteuse dédiée
3. Supprimer l'entrée de suivi d'appel
4. Retourner la confirmation STA

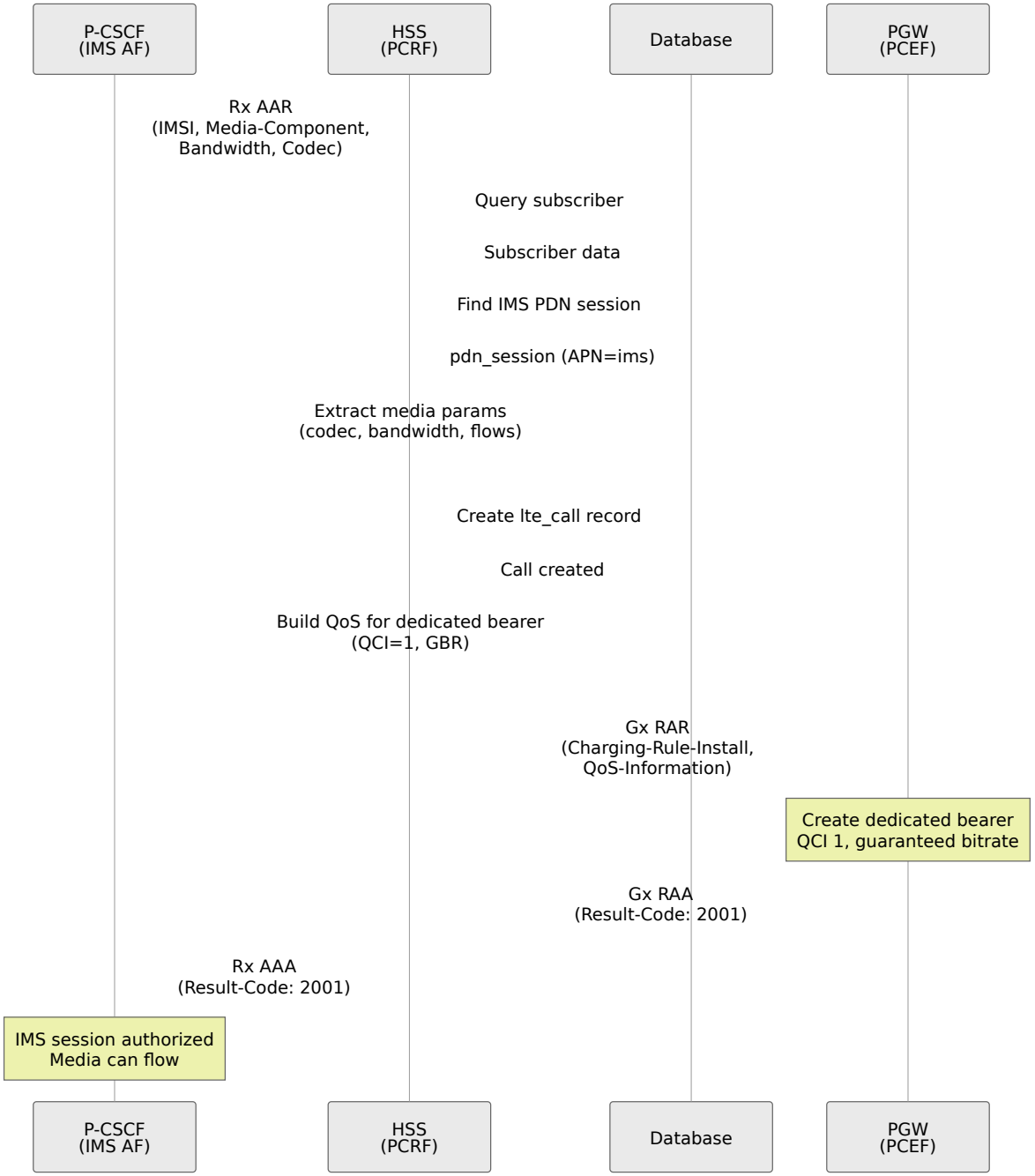
Réponse : Result-Code 2001

Flux de Messages Communs

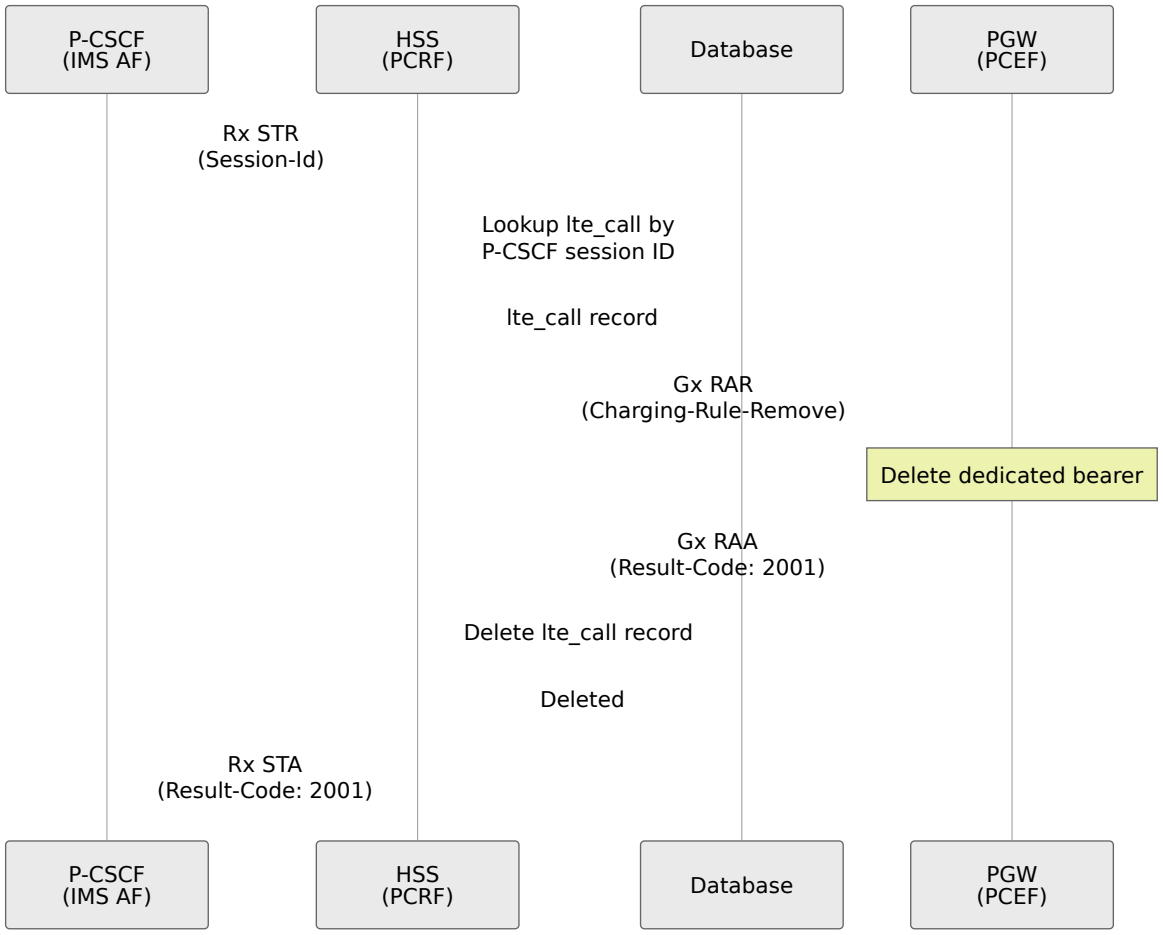
Flux 1 : Établissement de Session PDN



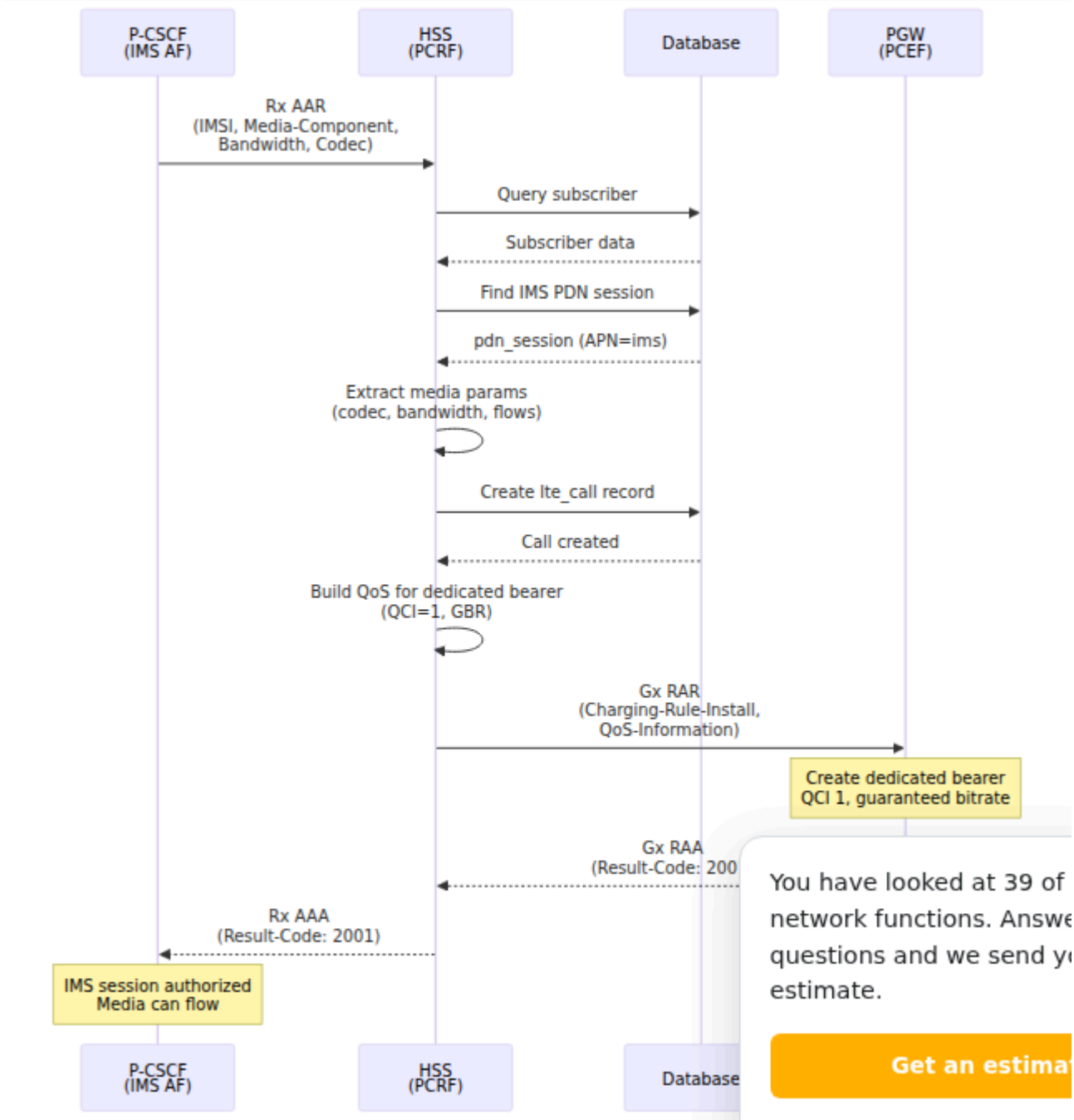
Flux 2 : Configuration d'Appel VoLTE (Rx AAR → Gx RAR)



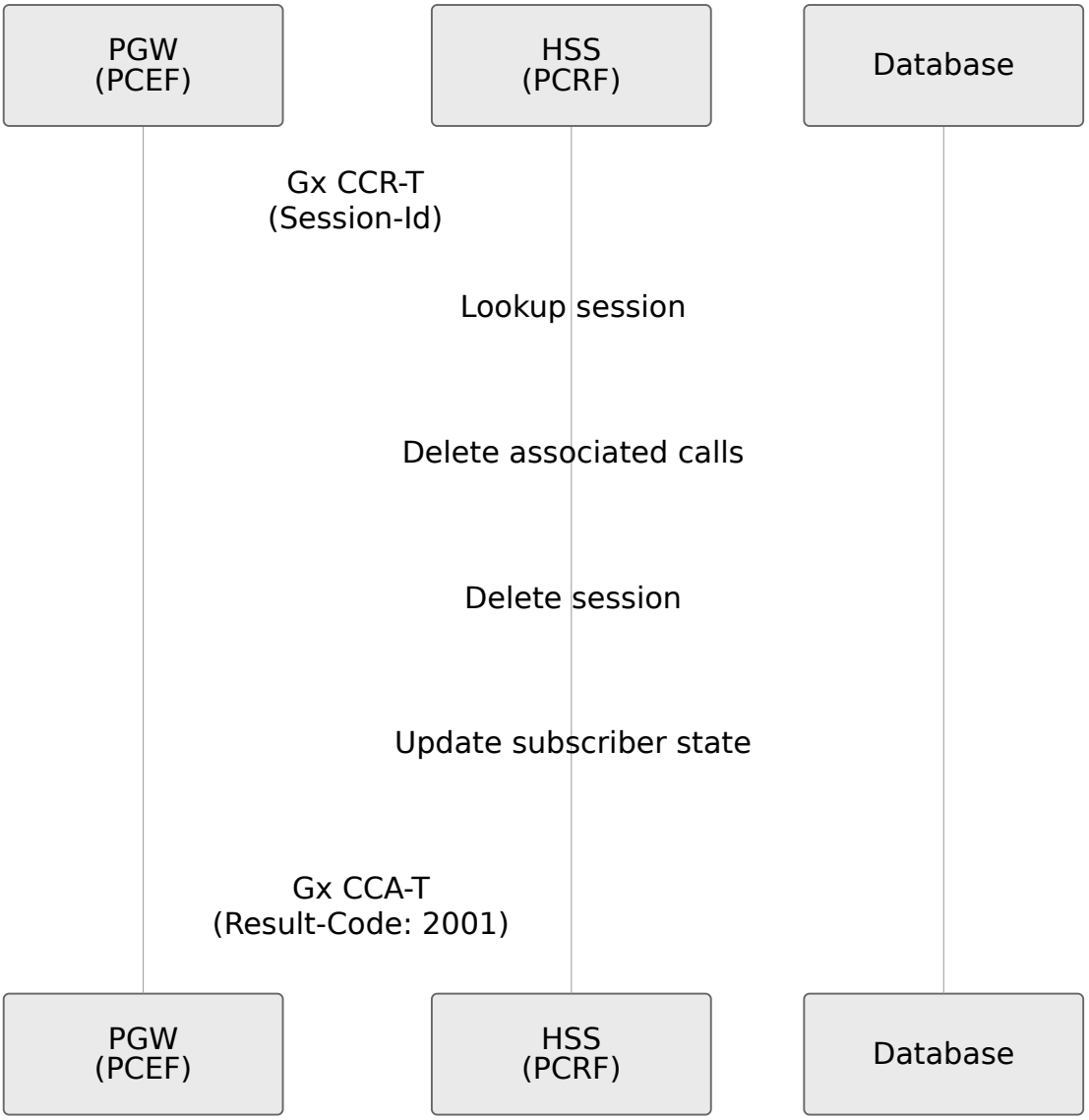
Flux 3 : Fin d'Appel VoLTE (Rx STR → Gx RAR)



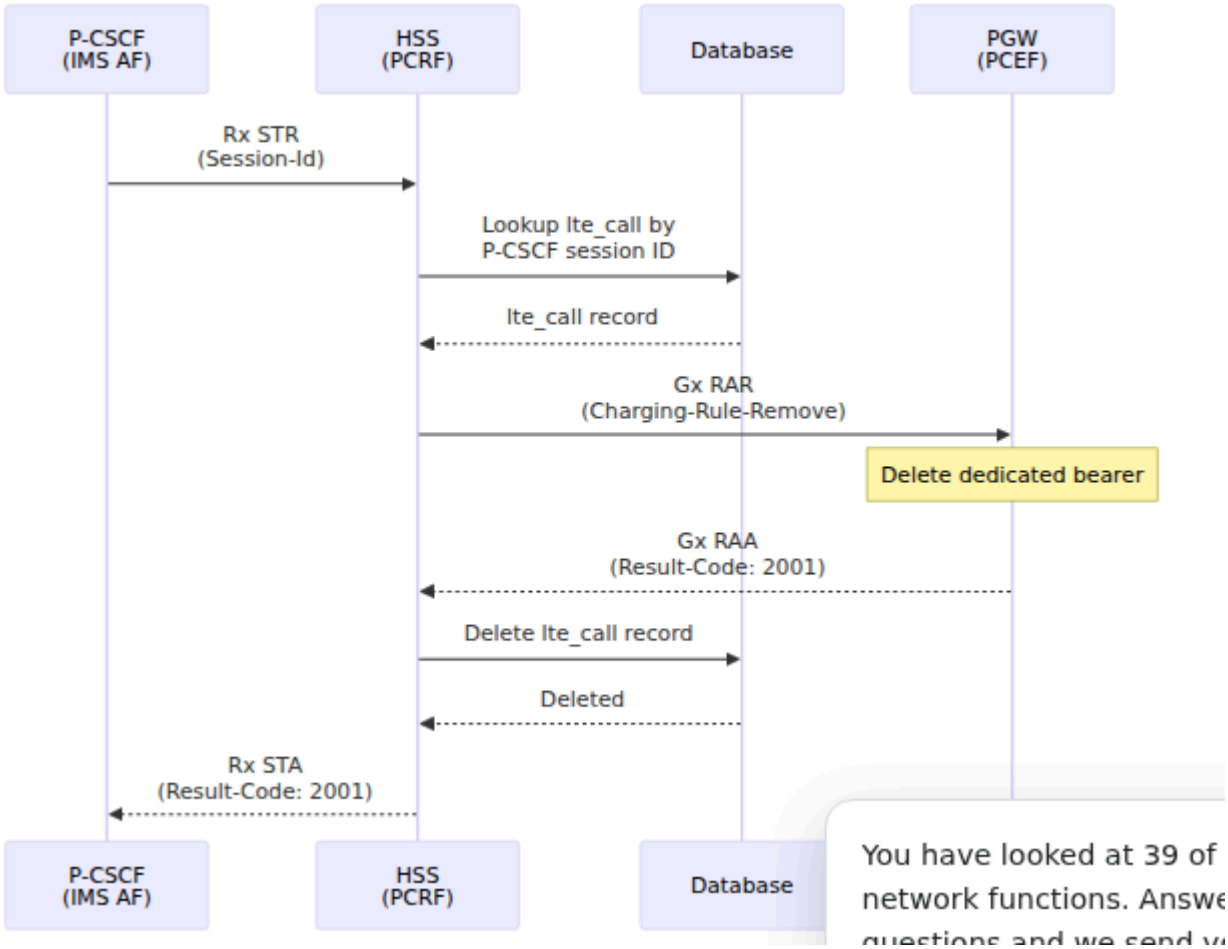
Flux 4 : Mise à jour de la Session PDN



Flux 5 : Terminaison de la Session PDN



Flux 6 : Ré-auth manuelle via l'API REST



API REST

Point de terminaison PCRF Re-Auth

Point de terminaison : `POST /api/operation/pcrf_re_auth`

But : Déclencher manuellement une Requête de Ré-auth Gx pour rafraîchir les politiques

Quand l'utiliser : Ce point de terminaison manuel est généralement utilisé pour le dépannage ou pour forcer le rafraîchissement des politiques sur des abonnés spécifiques. Pour les mises à jour de politique de routine (changement de profils QoS APN), le système déclenche automatiquement des ré-authentifications pour toutes les sessions affectées - aucune action manuelle n'est nécessaire.

Corps de la Requête :

```
{
  "imsi": "999999876543210",
  "apn": "ims"
}
```

Réponse de Succès (HTTP 200):

```
{
  "data": "Gx Re-Auth Request for 999999876543210 sent to
pgw.epc.mnc999.mcc999.3gppnetwork.org, Result-Code: 2001"
}
```

Réponse d'Erreur (HTTP 400):

```
{
  "error": "Unable to send Re-Auth Request for 999999876543210 on
APN ims, no active PDN Session found"
}
```

API de Configuration de Politique

Le PCRF récupère les politiques QoS à partir des configurations APN stockées dans la base de données. Ces politiques peuvent être créées et gérées via l'API REST.

Application Automatique des Politiques : Lorsque vous mettez à jour un profil QoS APN (par exemple, changement des limites de bande passante ou QCI), le système envoie automatiquement des Requêtes de Ré-auth Gx (RAR) à tous les PGW avec des sessions PDN actives utilisant cet APN. Cela garantit que les changements de politique sont appliqués immédiatement à tous les abonnés connectés sans intervention manuelle.

Architecture de Politique

Les politiques sont définies à travers une structure à trois niveaux :

```
Identifiant APN → Profil QoS APN → Profil APN
     ↓                  ↓                ↓
  "internet"      QCI, AMBR, ARP    Lien entre les deux
```

1. Créer un Identifiant APN

Définir le nom de l'APN et le support de version IP.

Point de terminaison : `POST /api/apn/identifier`

Corps de la Requête :

```
{
  "apn_identifier": {
    "apn": "internet",
    "ip_version": "ipv4v6"
  }
}
```

Options de Version IP :

- `"ipv4"` - IPv4 seulement
- `"ipv6"` - IPv6 seulement
- `"ipv4v6"` - Double pile (IPv4 et IPv6)
- `"ipv4_or_ipv6"` - Le réseau décide (soit IPv4 soit IPv6)

Réponse (HTTP 201):

```
{
  "data": {
    "id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
    "apn": "internet",
    "ip_version": "ipv4v6"
  }
}
```

Validation :

- `apn` : Requis, 1-254 caractères, unique
- `ip_version` : Requis, doit être l'une des quatre options ci-dessus

Lister les Identifiants APN : `GET /api/apn/identifier`

2. Créer un Profil QoS APN

Définir les paramètres QoS (bande passante, QCI, priorité).

Point de terminaison : `POST /api/apn/qos_profile`

Corps de la Requête :

```
{
  "apn_qos_profile": {
    "name": "Meilleur Effort Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "online_charging_enabled": false,
    "offline_charging_enabled": true
  }
}
```

Paramètres QoS :

Champ	Type	Plage	Description
`name`	string	1-254 chars	Nom du profil (unique)
`qci`	integer	1-254	Identifiant de Classe QoS (1-4 = GBR, 5-9 = Non-GBR)
`allocation_retention_priority`	integer	1-15	Niveau ARP (1 = priorité la plus élevée)
`apn_ambr_dl_kbps`	integer	1-4,294,967,293	Débit Maximum Agrégé APN en Descendant (kbps)
`apn_ambr_ul_kbps`	integer	1-4,294,967,293	Débit Maximum Agrégé APN en Montant (kbps)
`pre_emption_capability`	boolean	true/false	Peut préempter des porteuses de priorité inférieure

Champ	Type	Plage	Description
`pre_emption_vulnerability`	boolean	true/false	Peut être préempté par des porteuses de priorité supérieure
`online_charging_enabled`	boolean	true/false/null	AVP `Online` au niveau de la session pour le Gx CCA-I. `null` (par défaut) répète la valeur CCR-I du PGW. Voir Facturation en ligne/hors ligne au niveau de la session
`offline_charging_enabled`	boolean	true/false/null	AVP `Offline` au niveau de la session pour le Gx CCA-I. `null` (par défaut) répète la valeur CCR-I du PGW. Voir Facturation

Champ	Type	Plage	Description
			en ligne/hors ligne au niveau de la session

Valeurs QCI courantes :

- `1` - Voix Conversationnelle (VoLTE) - GBR, budget de délai de 100 ms
- `2` - Vidéo Conversationnelle - GBR, budget de délai de 150 ms
- `5` - Signalisation IMS - Non-GBR, budget de délai de 100 ms
- `9` - Porteuse par Défaut (Internet) - Non-GBR, budget de délai de 300 ms

Réponse (HTTP 201):

```
{
  "data": {
    "id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793",
    "name": "Meilleur Effort Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "online_charging_enabled": false,
    "offline_charging_enabled": true
  }
}
```

Lister les Profils QoS : `GET /api/apn/qos_profile`

3. Créer un Profil APN

Lier l'identifiant APN avec un profil QoS.

Point de terminaison : `POST /api/apn/profile`

Corps de la Requête :

```
{
  "apn_profile": {
    "name": "Profil APN Internet",
    "apn_identifier_id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
    "apn_qos_profile_id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793"
  }
}
```

Champs :

- `name` : Nom du profil (unique), utilisé pour référence
- `apn_identifier_id` : ID de Créer un Identifiant APN
- `apn_qos_profile_id` : ID de Créer un Profil QoS APN

Réponse (HTTP 201):

```
{
  "data": {
    "id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
    "name": "Profil APN Internet",
    "apn_identifier_id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
    "apn_qos_profile_id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793"
  }
}
```

Contraintes :

- `apn_identifier_id` et `apn_qos_profile_id` doivent référencer des enregistrements existants
- Chaque combinaison d'identifiant APN et de profil QoS doit être unique

Lister les Profils APN : `GET /api/apn/profile`

Exemple Complet de Configuration de Politique

Étape 1 : Créer la Politique APN IMS (VoLTE)

```
# 1. Créer un Identifiant APN
curl -X POST https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{
    "apn_identifier": {
      "apn": "ims",
      "ip_version": "ipv4v6"
    }
  }'
# Réponse : {"data": {"id": "2c3e4f5a-6b7c-4d8e-9f0a-
1b2c3d4e5f6a", ...}}

# 2. Créer un Profil QoS (Signalisation IMS)
curl -X POST https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_qos_profile": {
      "name": "QoS de Signalisation IMS",
      "qci": 5,
      "allocation_retention_priority": 2,
      "apn_ambr_dl_kbps": 5000,
      "apn_ambr_ul_kbps": 5000,
      "pre_emption_capability": true,
      "pre_emption_vulnerability": false
    }
  }'
# Réponse : {"data": {"id": "7b44a3e0-27cc-4b21-8d7a-
3f19e5c2b8d4", ...}}

# 3. Créer un Profil APN
curl -X POST https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_profile": {
      "name": "Profil APN IMS",
      "apn_identifier_id": "2c3e4f5a-6b7c-4d8e-9f0a-1b2c3d4e5f6a",
      "apn_qos_profile_id": "7b44a3e0-27cc-4b21-8d7a-3f19e5c2b8d4"
    }
  }'
# Réponse : {"data": {"id": "91c2f3a4-5d6e-4f7a-8b9c-
0d1e2f3a4b5c", ...}}
```

Étape 2 : Assigner à l'Abonné

Une fois créé, le profil APN est assigné aux abonnés via des profils EPC. Voir Référence API pour lier les profils APN aux abonnés.

Mise à jour et Suppression de Politique

Mettre à jour le Profil QoS :

```
PATCH /api/apn/qos_profile/{id}
PUT /api/apn/qos_profile/{id}
```

Exemple - Augmenter la Bande Passante pour Tous les Utilisateurs :

```
# Mettre à jour le profil QoS pour augmenter la bande passante
curl -X PATCH
https://hss.example.com:8443/api/apn/qos_profile/8d8ac610-566d-
4ef0-9c22-186b2a5ed793 \
  -H "Content-Type: application/json" \
  -d '{
    "apn_qos_profile": {
      "apn_ambr_dl_kbps": 150000,
      "apn_ambr_ul_kbps": 75000
    }
  }'
```

Ce qui se passe automatiquement :

1. Le profil QoS est mis à jour dans la base de données
2. Le système identifie toutes les sessions PDN actives utilisant des APNs liés à ce profil QoS
3. Pour chaque session active, un Gx RAR est envoyé au PGW correspondant
4. Les PGWs mettent à jour la QoS de la porteuse pour refléter les nouvelles limites de bande passante
5. Tous les abonnés connectés reçoivent immédiatement la politique mise à jour

Scénario d'exemple : Si 100 abonnés sont actuellement connectés sur l'APN "internet" utilisant ce profil QoS, tous les 100 auront leurs limites de bande passante mises à jour à 150 Mbps en descendant / 75 Mbps en montant dans les secondes suivant l'achèvement de l'appel API.

Remarque : Lorsque vous mettez à jour un profil QoS APN, le système **déclenche automatiquement une ré-auth** pour toutes les sessions PDN actives utilisant cet APN, appliquant immédiatement les nouvelles politiques aux abonnés attachés. Aucune ré-auth manuelle n'est requise.

Supprimer des Ressources :

```
DELETE /api/apn/identifier/{id}
DELETE /api/apn/qos_profile/{id}
DELETE /api/apn/profile/{id}
```

Contraintes de Suppression :

- Ne peut pas supprimer des identifiants APN ou des profils QoS référencés par des profils APN
- Ne peut pas supprimer des profils APN assignés à des abonnés actifs

Modèles de Politique

Internet Haute Vitesse (100 Mbps en descendant / 50 Mbps en montant) :

```
{
  "apn_qos_profile": {
    "name": "Internet Haute Vitesse",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

Internet Premium (500 Mbps en descendant / 100 Mbps en montant) :

```
{
  "apn_qos_profile": {
    "name": "Internet Premium",
    "qci": 8,
    "allocation_retention_priority": 5,
    "apn_ambr_dl_kbps": 500000,
    "apn_ambr_ul_kbps": 100000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}
```

IoT/M2M (Bande Passante Faible) :

```
{
  "apn_qos_profile": {
    "name": "IoT M2M",
    "qci": 9,
    "allocation_retention_priority": 10,
    "apn_ambr_dl_kbps": 1024,
    "apn_ambr_ul_kbps": 512,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

Services d'Urgence (Priorité Maximale) :

```
{
  "apn_qos_profile": {
    "name": "Profil APN d'Urgence",
    "qci": 5,
    "allocation_retention_priority": 1,
    "apn_ambr_dl_kbps": 10000,
    "apn_ambr_ul_kbps": 10000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}
```

Configuration

Configuration du Service Diameter

Application Gx (`config/runtime.exs`):

```
%{
  application_name: :gx,
  application_dictionary: :diameter_gen_3gpp_gx,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_238}
  ]
}
```

Application Rx (`config/runtime.exs`):

```
%{
  application_name: :rx,
  application_dictionary: :diameter_gen_3gpp_rx,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_236}
  ]
}
```

Paramètres QoS

Les paramètres QoS sont extraits de :

- **Porteuse par Défaut** : Configuration du profil APN dans la base de données
    - `apn_qos_profile.qci` (Identifiant de Classe QoS)
    - `apn_qos_profile.apn_ambr_ul_kbps` (Débit Maximum Agrégé en Montant)
    - `apn_qos_profile.apn_ambr_dl_kbps` (Débit Maximum Agrégé en Descendant)
    - `apn_qos_profile.priority_level` (Priorité d'Allocation-Rétention)
- **Porteuse Dédiée** : Extrait de la Description du Composant Multimédia AAR Rx
    - QCI : 1 (Voix Conversationnelle)
    - Débit Garanti : À partir des AVPs de Bande Passante Demandée Maximale
    - Filtres de Flux : À partir des AVPs de Description de Flux

Gestion des Erreurs

Code de Résultat	Type	Signification	Cause
2001	Succès	DIAMETER_SUCCESS	Demande traitée avec succès
5001	Expérimental	Utilisateur non trouvé	IMSI non présent dans la base de données des abonnés
5002	Expérimental	Session non trouvée	La session PDN n'existe pas pour mise à jour/terminer
5063	Expérimental	Service non autorisé	Autorisation multimédia IMS refusée

Détails de Mise en Œuvre

Gestion de Session

Le PCRF suit :

- **Sessions PDN Actives** - Une par APN, par abonné
- **Appels VoLTE** - Plusieurs appels par session IMS (prend en charge les appels en conférence)
- **Politiques QoS** - Appliquées dynamiquement en fonction de la configuration APN
- **Règles de Facturation** - Modèles de flux de trafic et politiques sp��cifiques au service

Fonctionnalités Avancées de Politique

Le PCRF prend en charge un contrôle de politique avancé, y compris :

- **Installation/Suppression de règles de facturation** via l'interface Gx
- **Correspondance de Modèle de Flux de Trafic (TFT)** pour la différenciation des services
- **Profils de vitesse dynamiques** basés sur l'application ou le type de trafic
- **Politiques conscientes du service** déclenchées par les conditions réseau ou le comportement des abonnés

Contactez votre administrateur système pour des informations sur la configuration des règles de facturation avancées et des politiques basées sur TFT.

Documentation Connexe

- Interface Sy - Contrôle de la limite de dépenses des abonnés contre un OCS externe
- Protocoles Diameter - Spécifications détaillées des protocoles
- Référence API - Documentation complète de l'API
- Architecture - Architecture globale du HSS
- Mapping des Données - Mappings de la base de données aux AVP Diameter

Gestion des erreurs de l'API

← Retour à la référence de l'API

Table des matières

- Réponses d'erreur courantes
- Flux de gestion des erreurs

Réponses d'erreur courantes

400 Mauvaise demande

```
{
  "error": "Invalid JSON format"
}
```

Causes :

- JSON mal formé
- Champs requis manquants
- Types de données invalides

404 Non trouvé

```
{
  "error": "Resource not found"
}
```

Causes :

- L'abonné/le profil/l'entité n'existe pas
- ID incorrect dans l'URL

422 Entité non traitable

```
{
  "errors": {
    "imsi": ["has already been taken"],
    "key_set_id": ["does not exist"]
  }
}
```

Causes :

- Échecs de validation
- Contraintes de base de données violées
- Références de clé étrangère inexistantes

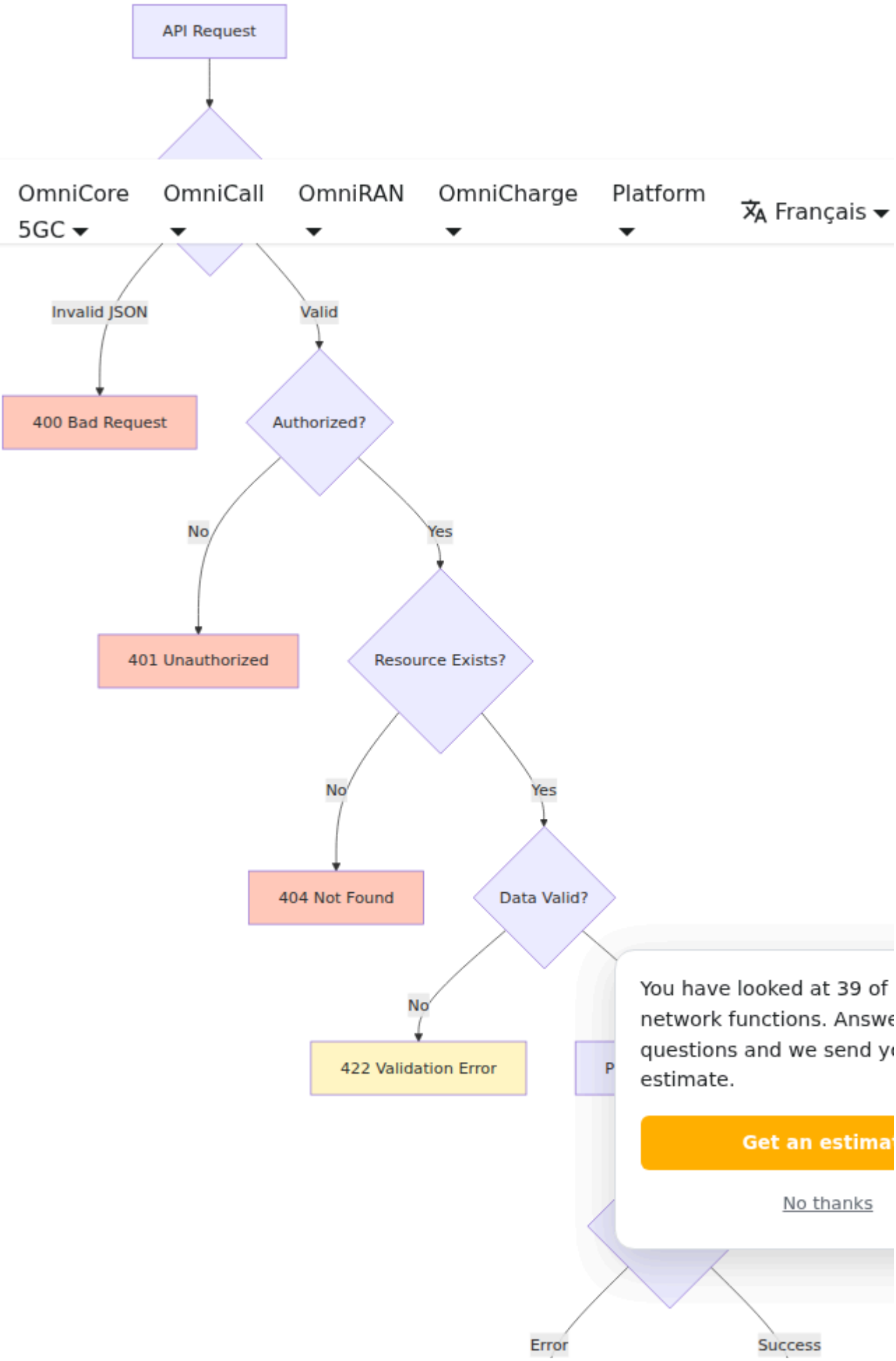
500 Erreur interne du serveur

```
{
  "error": "Internal server error"
}
```

Causes :

- Problèmes de connectivité à la base de données
- Erreurs d'application inattendues

Flux de gestion des erreurs



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



← Retour à la référence de l'API

Exemples d'utilisation de l'API

← Retour à la référence de l'API

Table des matières

- Provisionnement complet des abonnés
- Pré-provisionnement des abonnés dormants
- Provisionnement complet d'une IP statique

Provisionnement complet des abonnés

Cet exemple démontre le flux de travail complet pour le provisionnement d'un nouvel abonné à partir de zéro. Le processus implique la création de tous les profils et composants nécessaires avant de créer l'abonné.

Prérequis : Cet exemple utilise `jq` pour l'analyse JSON. Installez avec `apt-get install jq` ou `brew install jq`.

Sections connexes :

- Gestion des ensembles de clés
- Profils APN
- Profils EPC
- Gestion des abonnés

```
# 1. Créer l'ensemble de clés
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "authentication_algorithm": "milenage",
    "amf": "8000",
    "sqn": 0
  }' | jq -r '.response.id')

# 2. Créer le profil QoS APN
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Default Internet QoS",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 50000,
    "apn_ambr_ul_kbps": 25000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": true,
    "qci": 9
  }' | jq -r '.response.id')

# 3. Créer l'identifiant APN
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{
    "apn": "internet",
    "ip_version": "ipv4v6"
  }' | jq -r '.response.id')

# 4. Créer le profil APN
APN_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_identifier_id\": \"$APN_ID\",
    \"apn_qos_profile_id\": \"$APN_QOS_ID\",
    \"name\": \"Internet APN\"
```

```
  }" | jq -r '.response.id')

# 5. Créer le profil EPC
EPC_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_profiles\": [$APN_PROFILE_ID],
    \"name\": \"Standard Data Plan\",
    \"network_access_mode\": \"packet_only\",
    \"tracking_area_update_interval_seconds\": 600,
    \"ue_ambr_dl_kbps\": 100000,
    \"ue_ambr_ul_kbps\": 50000
  }" | jq -r '.response.id')

# 6. Créer l'abonné
SUBSCRIBER_ID=$(curl -k -X POST
https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"imsi\": \"001001123456789\",
    \"key_set_id\": \"$KEY_SET_ID\",
    \"epc_profile_id\": \"$EPC_PROFILE_ID\"
  }" | jq -r '.response.id')

echo "Abonné provisionné avec succès avec l'ID : $SUBSCRIBER_ID"
```

Ce que cela crée :

Ce flux de travail de provisionnement crée un abonné complet avec :

1. **Clés cryptographiques** (Ensemble de clés) - Pour l'authentification
2. **Profil de service de données** (Profil EPC) - Paramètres de bande passante et d'accès au réseau
3. **Configuration APN** (Profil APN) - Point d'accès avec QoS
4. **Enregistrement de l'abonné** (Abonné) - L'entité abonnée réelle

Prochaines étapes :

- Ajouter des numéros de téléphone : Voir Gestion des MSISDN
- Activer les services vocaux : Créer et assigner Profil IMS

- Configurer l'itinérance : Créer et assigner Profil d'itinérance
- Lier la SIM physique : Créer et assigner SIM

Voir aussi :

- Documentation Multi-MSISDN - Attribution de plusieurs numéros de téléphone
- Documentation des profils - Configuration avancée des profils

---

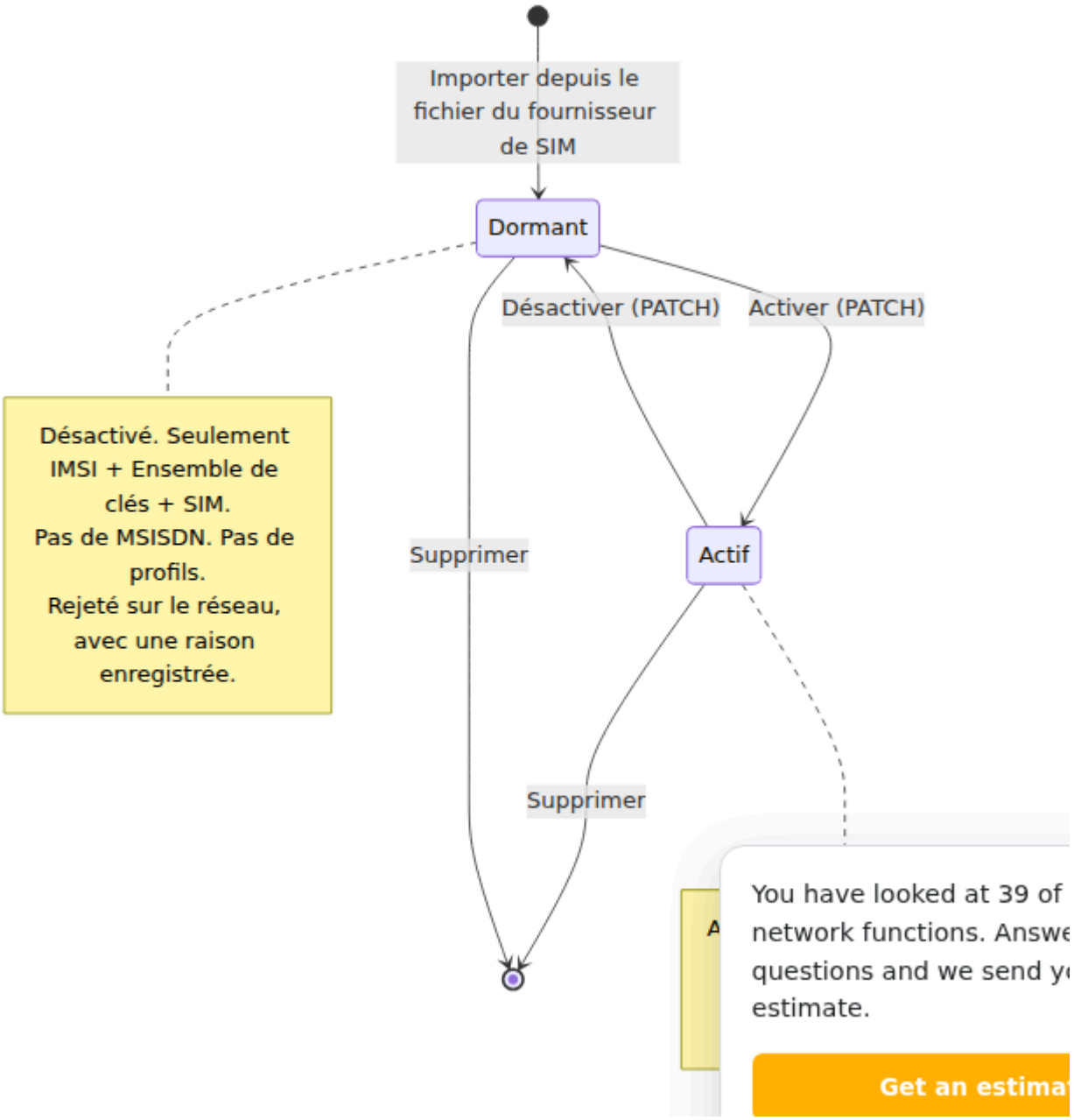
Pré-provisionnement des abonnés dormants

Utilisez ce flux de travail pour charger des abonnés en masse à partir d'un fichier de sortie d'un fournisseur de SIM, et pour activer chaque abonné plus tard. C'est le modèle recommandé pour la plupart des opérateurs.

Un abonné dormant ne conserve que son identité et ses clés d'authentification. Vous ajoutez le numéro de téléphone et les profils de service lors de l'activation. Cela correspond aux données qu'un fournisseur de SIM vous envoie. Le fichier du fournisseur de SIM contient l'ICCID, le PIN, les codes PUK et ADM, et les cl��s d'authentification (Ki, OPc). Le fichier ne contient pas de numéro de téléphone ni de plan de service.

La base de données a seulement besoin de l'IMSI et de l'ensemble de clés pour insérer un abonné. C'est ce qui rend un enregistrement dormant possible. Cela ne signifie pas que les autres objets sont optionnels pour le service. Chaque objet est requis pour son propre service : le profil EPC pour les données, le profil IMS et un MSISDN pour la voix, et une SIM pour l'authentification. Un abonné dormant ne fonctionne pas sur le réseau tant que vous n'avez pas assigné l'objet correspondant. Vous assignez ces objets lorsque le client active le service. Pour la raison que la SIM, l'ensemble de clés et l'abonné sont des objets séparés, voir Ensemble de clés, SIM et abonné.

Cycle de vie du provisionnement



Étape 1 : Importer (créer la SIM, l'ensemble de clés et un abonné désactivé)

Créez un ensemble de clés, une SIM et un abonné désactivé pour chaque ligne dans le fichier du fournisseur de SIM. Cet exemple utilise `jq` pour lire l'identifiant retourné de chaque réponse.

```
# 1. Créer l'ensemble de clés à partir du fichier du fournisseur
de SIM (Ki, OPc)
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "authentication_algorithm": "milenage",
    "amf": "8000",
    "sqn": 0
  }' | jq -r '.response.id')

# 2. Créer la SIM à partir du fichier du fournisseur de SIM
(ICCID, PIN/PUK/ADM)
SIM_ID=$(curl -k -X POST https://hss.example.com:8443/api/sim \
  -H "Content-Type: application/json" \
  -d '{
    "iccid": "89440000000000000001",
    "is_esim": false,
    "ind_length": 5
  }' | jq -r '.response.id')

# 3. Créer l'abonné dans un état désactivé.
#    imsi et key_set_id sont requis. Lier la SIM. Ajouter pas de
MSISDN et pas de profils.
SUBSCRIBER_ID=$(curl -k -X POST
https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"imsi\": \"001010000000001\",
    \"key_set_id\": \"$KEY_SET_ID\",
    \"sim_id\": \"$SIM_ID\",
    \"enabled\": false,
    \"ims_enabled\": false
  }" | jq -r '.response.id')
```

Un abonné dormant est sûr à conserver dans la base de données. L'abonné ne s'attache pas, ne s'authentifie pas et ne s'enregistre pas tant qu'il est désactivé. Si le réseau interroge un abonné dormant, le HSS rejette la demande. Voir Comportement des abonnés dormants sur le réseau.

Étape 2 : Activer (assigner le numéro et les profils, puis activer)

Pour activer un abonné, envoyez une requête `PATCH`. Ajoutez le MSISDN, assignez les profils et définissez `enabled` sur `true`.

```
curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d "{
    \"enabled\": true,
    \"ims_enabled\": true,
    \"msisdns\": [\"$MSISDN_ID\"],
    \"epc_profile_id\": \"$EPC_PROFILE_ID\",
    \"ims_profile_id\": \"$IMS_PROFILE_ID\",
    \"roaming_profile_id\": \"$ROAMING_PROFILE_ID\"
  }"
```

Une requête `PATCH` ne change que les champs que vous envoyez. L'IMSI, l'ensemble de clés et la SIM restent tels quels. Créez le MSISDN et les profils d'abord, ou réutilisez des identifiants existants. Voir Provisionnement complet des abonnés pour savoir comment créer les profils.

Étape 3 : Désactiver (désactiver l'abonné)

Pour désactiver un abonné, définissez `enabled` sur `false`. L'enregistrement de l'abonné et ses données restent dans la base de données.

```
curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d '{
    "enabled": false,
    "ims_enabled": false
  }'
```

Pour supprimer uniquement la voix IMS, et conserver le service de données, définissez `ims_enabled` sur `false` et laissez `enabled` sur `true`.

Comportement des abonnés dormants sur le réseau

Un abonné échoue en mode fermé tant qu'il est dormant ou désactivé. Le HSS rejette la demande sur l'interface réseau. Le HSS écrit également un journal d'avertissement avec la raison exacte. La raison est l'une des suivantes :

- L'abonné est désactivé (`enabled` est `false`).
- IMS est désactivé pour l'abonné (`ims_enabled` est `false`).
- Un objet requis n'est pas encore assigné. Par exemple, il n'y a pas de profil EPC pour une attache LTE, ou pas de profil IMS pour l'enregistrement IMS.

Le rejet utilise le code de résultat expérimental `DIAMETER_ERROR_USER_UNKNOWN (5001)` sur les interfaces S6a et Cx, conformément à 3GPP TS 29.272 et 3GPP TS 29.229. Le journal d'avertissement nomme l'IMSI et la raison exacte. Le journal vous permet de distinguer un abonné dormant d'un IMSI inconnu.

Provisionnement complet d'une IP statique

Cet exemple démontre le provisionnement d'un abonné avec une adresse IP statique à partir de zéro.

Scénario : Provisionner un abonné de dispositif IoT qui a besoin d'une adresse IPv4 statique sur l'APN "internet".

```
# Prérequis : jq doit être installé (apt-get install jq ou brew
install jq)

# 1. Créer l'ensemble de clés
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "authentication_algorithm": "milenage",
    "amf": "8000",
    "sqn": 0
  }' | jq -r '.response.id')

# 2. Créer le profil QoS APN
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "IoT Best Effort",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 10000,
    "apn_ambr_ul_kbps": 5000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": false,
    "qci": 9
  }' | jq -r '.response.id')

# 3. Créer l'identifiant APN
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{
    "apn": "internet",
    "ip_version": "ipv4"
  }' | jq -r '.response.id')

# 4. Créer le profil APN
APN_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
```

```
    \"apn_identifier_id\": \"$APN_ID\",
    \"apn_qos_profile_id\": \"$APN_QOS_ID\",
    \"name\": \"IoT Internet APN\"
  }" | jq -r '.response.id')

# 5. Créer une IP statique pour l'APN
STATIC_IP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_profile_id\": \"$APN_PROFILE_ID\",
    \"ipv4_static_ip\": \"100.64.1.100\"
  }" | jq -r '.response.id')

# 6. Créer le profil EPC
EPC_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_profiles\": [$APN_PROFILE_ID],
    \"name\": \"IoT Data Plan\",
    \"network_access_mode\": \"packet_only\",
    \"tracking_area_update_interval_seconds\": 600,
    \"ue_ambr_dl_kbps\": 10000,
    \"ue_ambr_ul_kbps\": 5000
  }" | jq -r '.response.id')

# 7. Créer un MSISDN (numéro de téléphone)
MSISDN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{
    "msisdn": "14155551000"
  }' | jq -r '.response.id')

# 8. Créer un abonné avec une IP statique
SUBSCRIBER_ID=$(curl -k -X POST
https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"imsi\": \"001001999999999\",
    \"key_set_id\": \"$KEY_SET_ID\",
    \"epc_profile_id\": \"$EPC_PROFILE_ID\",
    \"msisdns\": [$MSISDN_ID],
```

```
    \"static_ips\": [$STATIC_IP_ID]
  }" | jq -r '.response.id')

echo "Abonné IoT provisionné avec succès !"
echo "  ID de l'abonné : $SUBSCRIBER_ID"
echo "  IMSI : 001001999999999"
echo "  MSISDN : 14155551000"
echo "  IPv4 statique : 100.64.1.100 (sur l'APN 'internet')"
```

Ce que cela crée :

Ce flux de travail de provisionnement crée un abonné IoT complet avec :

1. **Clés cryptographiques** (Ensemble de clés) - Pour l'authentification
2. **Configuration APN** (Profil APN) - Point d'accès "internet"
3. **Attribution d'IP statique** (IP statique) - Adresse IPv4 fixe 100.64.1.100
4. **Profil de service de données** (Profil EPC) - Limites de bande passante optimisées pour l'IoT
5. **Numéro de téléphone** (MSISDN) - Pour l'identification de l'appareil
6. **Enregistrement de l'abonné** (Abonné) - L'entité abonnée complète

Résultat :

Lorsque cet abonné s'attache au réseau et se connecte à l'APN "internet", il recevra l'adresse IP statique `100.64.1.100` au lieu d'une adresse DHCP dynamique.

Prochaines étapes :

- Ajouter des APN supplémentaires avec des IP statiques : Répétez les étapes 2-5 pour chaque APN
- Activer les services vocaux : Créer et assigner Profil IMS
- Configurer l'itinérance : Créer et assigner Profil d'itinérance
- Lier la SIM physique : Créer et assigner SIM

Voir aussi :

- Gestion des IP statiques - Documentation détaillée sur les IP statiques

- Provisionnement complet des abonnés - Provisionnement de base sans IP statique
- Documentation Multi-MSISDN - Attribution de plusieurs numéros de téléphone

---

← Retour à la référence de l'API

Contrôle d'Accès API (RBAC)

Cette page vous explique comment contrôler l'accès à l'API REST OmniHSS. Elle explique les clés API, les rôles et les permissions. Elle fournit également des rôles suggérés pour les systèmes qui utilisent l'API, par exemple OmniCRM, OmniWeb, OmniHLR et les outils de surveillance.

Retour au Guide des Opérations. Pour les points de terminaison de l'API, voir la Référence API.

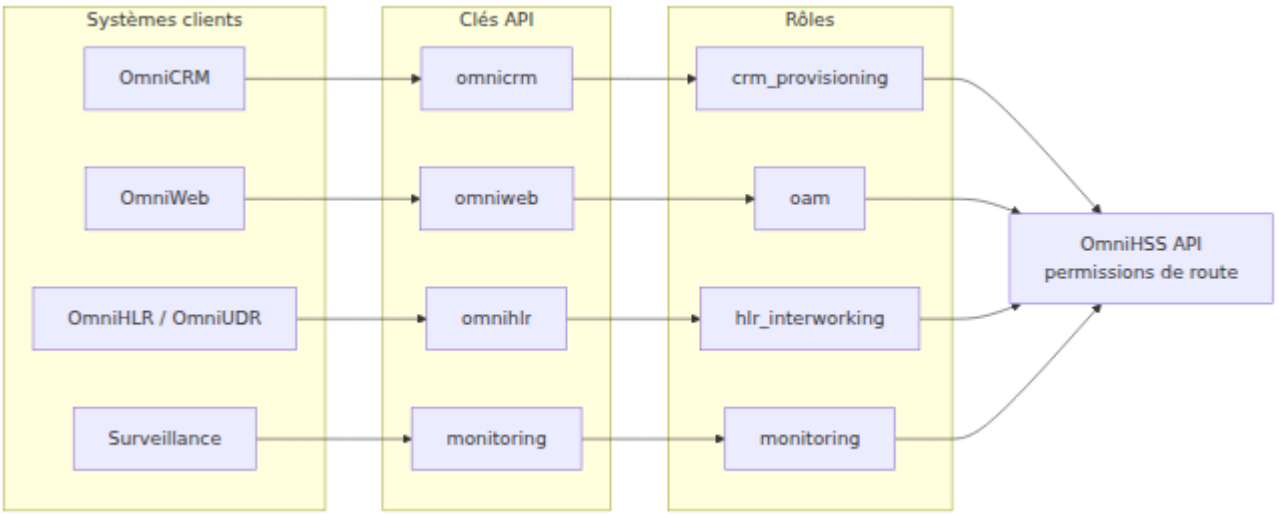
Table des Matières

- Aperçu
- Modes d'Authentification
- Comment OmniHSS Autorise une Demande
- Configuration
- Permissions
- Catalogue de Permissions OmniHSS
- Rôles Suggérés
- Exemple de Configuration Complet
- Utilisateur Agissant
- Appeler l'API avec une Clé
- Activer RBAC sur un Système en Cours d'Exécution
- Faire Pivoter une Clé
- Journalisation et Audit
- Dépannage

Aperçu

L'API OmniHSS peut changer les identifiants des abonnés, les numéros de téléphone et les profils de service. Elle peut également envoyer des opérations réseau, par exemple une Demande d'Annulation de Localisation à un MME. Ainsi, chaque système client doit obtenir uniquement l'accès dont il a besoin.

Le contrôle d'accès basé sur les rôles (RBAC) donne à chaque système client sa propre **clé API**. Chaque clé a un ou plusieurs **rôles**. Chaque rôle est une liste de **modèles de permission**. Chaque route API a exactement une **permission**. OmniHSS autorise une demande uniquement si un rôle de la clé correspond à la permission de la route. Toutes les autres demandes sont refusées. C'est ce qu'on appelle "refus par défaut".



Les interfaces Diameter (S6a, Cx, Sh, Gx, Rx, et autres) n'utilisent pas de clés API. Les pairs Diameter sont contrôlés par la liste des pairs dans la configuration Diameter. Voir Configuration Diameter.

Modes d'Authentification

L'API a trois modes d'authentification. Le mode provient du bloc `config :api_ex, :auth` dans `runtime.exs`. Le mode dépend des clés présentes dans ce bloc.

`config :api_ex, :auth`	Mode	Comportement
Non défini	Désactivé	L'API sert chaque route à chaque appelant sans clé. Un avertissement est écrit dans le journal au démarrage.
Contient `key` et pas `api_keys`	Héritage	Une clé partagée donne un accès complet à toutes les routes.
Contient `api_keys`	RBAC	Chaque client a une clé nommée avec des rôles. Les demandes que aucun rôle ne permet sont refusées.

> **Avertissement :** Ne pas exécuter un OmniHSS en production en mode désactivé. Si le bloc `config :api_ex, :auth` n'est pas défini, tout hôte pouvant atteindre le port 8443 peut lire et modifier toutes les données des abonnés.

Utilisez le mode RBAC pour tous les systèmes de production. Utilisez le mode héritage uniquement comme une étape courte lors de la migration.

Règles de mode :

- Une valeur vide ne désactive pas l'authentification. Par exemple, `key: ""` est une erreur de configuration. L'API répond alors à chaque demande avec `503`.
- OmniHSS définit le mode au démarrage. Si la configuration change pour un mode inférieur pendant que le service fonctionne, l'API répond à chaque demande avec `503`. Redémarrez le service pour utiliser un mode inférieur.
- En mode RBAC, une configuration invalide arrête l'API au démarrage. Le journal donne une liste numérotée de chaque problème. L'API ne démarre jamais sans protection.

Comment OmniHSS Autorise une Demande



OmniHSS prend la décision avant de lire le corps de la demande. Un appelant qui reçoit `401`, `403` ou `503` ne peut pas faire lire un grand corps de demande au serveur.

Configuration

Mettez le bloc `config :api_ex, :auth` dans le fichier de configuration d'exécution `runtime.exs`. Sur un hôte déployé, ce fichier est `/etc/omnihss/runtime.exs`. Redémarrez OmniHSS après avoir modifié le bloc.

Structure de Configuration

```
# runtime.exs
config :api_ex, :auth,
  header: "x-omnihss-key",
  user_header: "x-omnihss-user",
  public: ["status/api:read"],
  roles: %{
    admin: ["*"],
    monitoring: ["status/**:read", "subscriber/**:read"]
  },
  api_keys: [
    %{
      name: "monitoring",
      roles: [:monitoring],
      secrets: ["REPLACE-WITH-A-64-CHARACTER-RANDOM-HEX-SECRET-
FROM-OPENSSL-RAND"]
    },
    %{
      name: "break-glass",
      roles: [:admin],
      secrets: ["REPLACE-WITH-A-DIFFERENT-64-CHARACTER-RANDOM-HEX-
SECRET-VALUE"]
    }
  ],
  log: [deny: :warning, unauthenticated: :info, allow: false,
not_found: false]
```

Paramètres d'Auth

Paramètre	Type	Requis	Par défaut	Description
`header`	Chaîne	Non	`"x-api-key"`	L'en-tête HTTP qui transporte la clé API. Le nom n'est pas sensible à la casse. Utilisez `"x-omnihss-key"`. Ne pas utiliser le défaut `"x-api-key"`, car OmniWeb utilise `X-API-Key` pour ses propres clés. Voir OmniWeb.
`user_header`	Chaîne	Non	Non défini	L'en-tête HTTP qui nomme l'utilisateur humain pour lequel un client agit. Il doit être différent de `header`. Voir Utilisateur Agissant.
`public`	Liste de chaînes	Non	`[]`	Modèles de permission que les appelants peuvent utiliser sans clé. Utilisez ceci uniquement pour les vérifications de santé. Gardez les

Paramètre	Type	Requis	Par défaut	Description
				modèles publics en lecture seule.
`roles`	Carte	Oui (RBAC)	-	Noms de rôles et leurs modèles de permission. Chaque clé de la carte est un nom de rôle. Chaque valeur est une liste de modèles. Voir Permissions.
`api_keys`	Liste de cartes	Oui (RBAC)	-	Les clés API. La présence de ce paramètre sélectionne le mode RBAC. Voir Paramètres de Clé API.
`log`	Liste de mots-clés	Non	Voir description	Le niveau de journalisation pour chaque type de décision. Voir Paramètres de Journal.
`key`	Chaîne	Héritage uniquement	-	La clé partagée unique du mode héritage. Ne pas définir `key` et `api_keys` ensemble. Supprimez `key`

Paramètre	Type	Requis	Par défaut	Description
				lorsque vous passez à RBAC.

Paramètres de Clé API

Paramètre	Type	Requis	Par défaut	Description
`name`	Chaîne	Oui	-	Un nom unique pour la clé. Utilisez le nom du système client, par exemple `"omnicrm"`. Le nom apparaît dans les journaux. Le secret n'apparaît jamais dans les journaux.
`roles`	Liste d'atomes	Oui	-	Les rôles de la clé. Chaque rôle doit être défini dans `roles`. Une clé obtient les permissions de tous ses rôles.
`secrets`	Liste de chaînes	Oui	-	Un ou plusieurs secrets pour la clé. Le client envoie l'un de ces secrets dans l'en-tête de clé. Deux secrets vous permettent de faire pivoter une clé sans temps d'arrêt. Voir Faire Pivoter une Clé. Chaque secret doit être unique parmi toutes les clés.

Paramètre	Type	Requis	Par défaut	Description
`may_assert_user`	Booléen	Non	`false`	Si `true`, la clé peut nommer un utilisateur humain dans `user_header`. Définissez ceci uniquement pour les systèmes qui ont leur propre connexion utilisateur, par exemple OmniWeb et OmniCRM.
`metadata`	Carte	Non	`%{}`	Données libres sur la clé, par exemple un propriétaire ou un numéro de ticket. OmniHSS n'utilise pas ces données pour prendre des décisions.

Règles de secret :

- Utilisez un secret aléatoire long. Créez un secret avec la commande `openssl rand -hex 32`. Cela donne 64 caractères hexadécimaux.
- OmniHSS écrit un avertissement au démarrage pour un secret plus court que 16 octets. Seize octets est un minimum, pas une recommandation.
- Donnez à chaque système client sa propre clé. Ne partagez pas une clé entre deux systèmes. Si vous partagez une clé, vous ne pouvez pas retirer l'accès d'un seul système, et les journaux ne peuvent pas distinguer les deux systèmes.

Paramètres de Journal

Paramètre	Type	Requis	Par défaut	Description
`deny`	Niveau de journal ou `false`	Non	`:warning`	Le niveau de journal pour une demande que les rôles d'une clé ne permettent pas (`403`).
`unauthenticated`	Niveau de journal ou `false`	Non	`:info`	Le niveau de journal pour une demande sans clé ou avec une mauvaise clé (`401`).
`allow`	Niveau de journal ou `false`	Non	`false`	Le niveau de journal pour une demande autorisée. Définissez `:info` pour journaliser chaque appel API autorisé.
`not_found`	Niveau de journal ou `false`	Non	`false`	Le niveau de journal pour une demande d'une clé connue vers un chemin qui n'existe pas (`404`).

Un niveau de journal est `:debug`, `:info`, `:warning`, ou `:error`. La valeur `false` désactive la ligne de journal pour cette décision.

Permissions

Format de Permission

Chaque route a une permission au format `resource:verb`.

- La **ressource** est le chemin de l'API sans le préfixe `/api` et sans les paramètres de chemin. Par exemple, la route `GET /api/subscriber/imsi/:imsi` a la ressource `subscriber/imsi`.
- Le **verbe** provient de la méthode HTTP.

Méthode HTTP	Verbe
`GET` (liste et enregistrement unique)	`read`
`POST`	`write`
`PUT`, `PATCH`	`write`
`DELETE`	`delete`

Ainsi, `POST /api/operation/s6a/clr` a la permission `operation/s6a/clr:write`, et `DELETE /api/subscriber/:id` a la permission `subscriber:delete`.

Format de Modèle

Un rôle est une liste de modèles. Un modèle a le même format `resource:verb`, et il peut contenir des caractères génériques.

Caractère générique	Position	Correspond à
`*` (modèle entier)	Seul	Toutes les permissions.
`*` (segment de ressource)	N'importe quel segment	Exactement un segment. À la fin de la ressource, il correspond à un ou plusieurs segments.
`**`	Dernier segment uniquement	Zéro ou plusieurs segments.
`*` (verbe)	Après `:`	Tous les verbes.

Un segment est le texte entre deux caractères `/`. Un `*` n'est pas une correspondance de préfixe. Par exemple, `sub*` n'est pas un modèle valide.

Exemples de Modèle

Les caractères génériques `*` et `**` donnent des résultats différents. Utilisez ce tableau lorsque vous écrivez un rôle.

Modèle	`subscriber:read`	`subscriber/imsi:read`	`subscr`
`subscriber:read`	Oui	Non	Non
`subscriber:*`	Oui	Non	Oui
`subscriber/*:read`	Non	Oui	Non
`subscriber/**:read`	Oui	Oui	Non
`subscriber/**:*`	Oui	Oui	Oui
`*:read`	Oui	Oui	Non
`*`	Oui	Oui	Oui

Points clés :

- `subscriber/*:read` donne les routes de recherche (`subscriber/imsi`, `subscriber/msisdn`, et autres), mais pas la liste `subscriber`.
- `subscriber/**:read` donne les routes `subscriber` et toutes les routes en dessous de `subscriber`.
- Les rôles n'ajoutent que des permissions. Il n'y a pas de modèle "refuser". Pour retirer une permission d'un rôle, écrivez le rôle avec une liste de modèles plus petite.

Catalogue de Permissions OmniHSS

Ce tableau montre chaque permission de l'API OmniHSS. Pour le format de demande et de réponse de chaque route, voir la Référence API.

Statut et Documentation

Permission	Routes	Remarques
`status/api:read`	`GET /api/status/api`	Santé de l'API. Un bon candidat pour `public`.
`status/license:read`	`GET /api/status/license`	Statut de la licence.
`status/database:read`	`GET /api/status/database`	Santé de la base de données.
`docs:read`	`GET /api/docs`, `GET /docs`	Interface Swagger.
`schema:read`	`GET /api/schema`, `GET /schema`	Document OpenAPI.
`version:read`	`GET /version`	Version et build du logiciel.

Données des Abonnés

Permission	Routes
`subscriber:read`	`GET /api/subscriber`, `GET /api/subscriber/:id`
`subscriber:write`	`POST /api/subscriber`, `PUT` et `PATC` `/api/subscriber/:id`
`subscriber:delete`	`DELETE /api/subscriber/:id`
`subscriber/imsi:read`	`GET /api/subscriber/imsi/:imsi`
`subscriber/iccid:read`	`GET /api/subscriber/iccid/:iccid`
`subscriber/msisdn:read`	`GET /api/subscriber/msisdn/:msis`
`subscriber/ip:read`	`GET /api/subscriber/ip/:ip_addres`
`subscriber/repository_data:read`	`GET /api/subscriber/repository_data/:`
`subscriber/repository_data:write`	`PUT` et `PATCH` `/api/subscriber/repository_data/:`

Permission	Routes
`subscriber/repository_data:delete`	`DELETE /api/subscriber/repository_data/:`
`msisdn:read`	`GET /api/msisdn`, `GET /api/msisdn/`
`msisdn:write`	`POST /api/msisdn`, `PUT` et `PATCH /api/msisdn/:id`
`msisdn:delete`	`DELETE /api/msisdn/:id`
`msisdn/digits:read`	`GET /api/msisdn/digits/:digits`
`ims_subscription:read`, `:write`, `:delete`	`/api/ims_subscription`
`epc/static_ip:read`, `:write`, `:delete`	`/api/epc/static_ip`

Identifiants

Permission	Routes	Remarques
`key_set:read`	`GET /api/key_set`, `GET /api/key_set/:id`	La réponse ne contient pas Ki, OPc, OP, PSK, DES, ou le mot de passe SIP. Ces champs sont supprimés de chaque lecture.
`key_set:write`	`POST /api/key_set`, `PUT` et `PATCH /api/key_set/:id`	**Sensible.** Un appelant avec cette permission peut définir ou remplacer les clés d'authentification d'un abonné.
`key_set:delete`	`DELETE /api/key_set/:id`	**Sensible.**
`sim:read`	`GET /api/sim`, `GET /api/sim/:id`	La réponse ne contient pas KIc, KID, ou ADM1.
`sim:write`	`POST /api/sim`, `PUT` et `PATCH /api/sim/:id`	
`sim:delete`	`DELETE /api/sim/:id`	
`sim/iccid:read`	`GET /api/sim/iccid/:iccid`	Recherche par ICCID.

Profils et Règles

Chaque ressource dans ce tableau a les trois permissions `read`, `write`, et `delete`. Les routes sont `GET`, `POST`, `PUT`, `PATCH`, et `DELETE` sur le chemin. Un changement de profil peut changer le service de nombreux abonnés en même temps.

Ressource	Chemin	Descr
`apn/profile`	`/api/apn/profile`	Profils Voir G des Pr
`apn/identifier`	`/api/apn/identifier`	Noms
`apn/qos_profile`	`/api/apn/qos_profile`	QoS A
`epc/profile`	`/api/epc/profile`	Profils (donn
`ims/profile`	`/api/ims/profile`	Profils (voix).
`roaming/profile`	`/api/roaming/profile`	Profils roami Voir C de Roa
`roaming/rule`	`/api/roaming/rule`	Règles roami
`eir/rule`	`/api/eir/rule`	Règles Voir EI
`pcrf/profile`	`/api/pcrf/profile`	Profils Voir PC
`pcrf/charging_rule`	`/api/pcrf/charging_rule`	Règles factur PCRF.
`pcrf/flow_information_rule`	`/api/pcrf/flow_information_rule`	Règles flux PC

Interopérabilité HLR

Permission	Routes
`hlr/generate_auth_vectors:write`	`POST /api/hlr/generate_auth_`
`hlr/generate_eutran_auth_vectors:write`	`POST /api/hlr/generate_eutran_aut`
`hlr/circuit_session:write`, `:delete`	`PUT`, `PATCH`, `DELETE /api/hlr/circuit_session/:im`
`hlr/subscriber_state:write`	`PUT` et `PATCH /api/hlr/subscriber_state/:i`
`hlr/pdn_session:write`, `:delete`	`POST`, `PUT`, `PATCH`, `DELETE /api/hlr/pdn_session`

Opérations Réseau

Permission	Routes	Rema
`operation/count:read`	`GET /api/operation/count/:type`	Comptes enregistr
`operation/s6a/clr:write`	`POST /api/operation/s6a/clr`	Envoie u Demand d'Annula Localisat MME. L'a se détac Flux de Protocole
`operation/s6a/idr:write`	`POST /api/operation/s6a/idr`	Envoie u Demand d'Insertio Données d'Abonne MME.
`operation/pcrf/re_auth:write`	`POST /api/operation/pcrf/re_auth`	Envoie u Demand Authenti Gx au PG PCRF.

Rôles Suggérés

Ces rôles donnent à chaque système client les routes qu'il utilise, et pas plus. Commencez avec ces rôles. Ensuite, réduisez chaque rôle si un client n'utilise pas une fonctionnalité.

Rôle	Systèmes clients	Accès
`admin`	Clé de secours pour les urgences	Toutes les routes.
`crm_provisioning`	OmniCRM	Provisionnement d'abonnés, MSISDN, SIM, et ensembles de clés. Lecture seule pour les profils.
`oam`	OmniWeb	Toutes les données des abonnés, profils, et opérations réseau. Pas d'état de session HLR.
`hlr_interworking`	OmniHLR (OmniSS7), OmniUDR	Recherches d'abonnés, vecteurs d'authentification, état commuté par circuit.
`monitoring`	Exportateurs, tableaux de bord, scripts qui ne font que lire	Recherches d'abonnés et état en lecture seule.
`eir_lookup`	OmniEIR	Lire uniquement les r��gles EIR.

Admin (Clé de Secours)

```
admin: ["*"]
```

Comment ça fonctionne : La clé peut utiliser chaque route, y compris les nouvelles routes dans les versions ultérieures.

Cas d'utilisation : Accès d'urgence par un opérateur. Conservez le secret dans un coffre-fort de mots de passe. Ne configurez pas un système client avec

cette clé. Consultez les journaux pour chaque utilisation de cette clé.

OmniCRM (Provisionnement)

```
crm_provisioning: [
  "subscriber:*",
  "subscriber/*:read",
  "msisdn/**:*",
  "sim/**:read",
  "sim:write",
  "key_set:write",
  "status/**:read"
]
```

Comment ça fonctionne : OmniCRM crée, modifie et supprime des abonnés et des MSISDN. Il crée des SIM et des ensembles de clés lorsqu'il provisionne un nouveau service. Il recherche des abonnés par IMSI et par adresse IP UE. Il ne peut pas modifier des profils, des règles, ou des opérations réseau. Il ne peut pas supprimer des ensembles de clés ou des SIM.

Cas d'utilisation : Le CRM provisionne des services pour les clients. Définissez `may_assert_user: true` sur la clé OmniCRM afin que le CRM puisse nommer le membre du personnel qui a effectué le changement. Voir Utilisateur Agissant.

Si le CRM doit envoyer une Demande d'Annulation de Localisation après un échange de SIM, ajoutez `"operation/s6a/clr:write"`.

OmniWeb (OAM)

```
oam: [
  "*:read",
  "subscriber/**:*",
  "msisdn/**:*",
  "sim/**:*",
  "key_set:*",
  "ims_subscription:*",
  "apn/**:*",
  "epc/**:*",
  "ims/**:*",
  "roaming/**:*",
  "eir/**:*",
  "pcrf/**:*",
  "operation/**:write",
  "hlr/subscriber_state:write",
  "hlr/generate_auth_vectors:write",
  "hlr/generate_eutran_auth_vectors:write"
]
```

Comment ça fonctionne : OmniWeb gère toutes les données des abonnés, les profils, et les règles. Il envoie des opérations réseau (CLR, IDR, et Ré-Auth Gx). Il peut générer des vecteurs d'authentification de test. Il ne peut pas changer l'état de session commuté par circuit ou PDN que possède le HLR.

Cas d'utilisation : OmniWeb est le système OAM. OmniWeb a ses propres utilisateurs et son propre RBAC pour chaque élément. OmniWeb vérifie son propre RBAC avant d'envoyer une demande à OmniHSS. Ainsi, la clé OmniHSS d'OmniWeb doit couvrir chaque fonctionnalité que montre OmniWeb. Définissez `may_assert_user: true` sur la clé OmniWeb afin qu'OmniHSS puisse enregistrer l'utilisateur d'OmniWeb.

> **Avertissement :** Ne pas utiliser `x-api-key` comme en-tête de clé OmniHSS. OmniWeb utilise `X-API-Key` pour ses propres clés API personnelles, et le proxy OmniWeb envoie cet en-tête à OmniHSS. Définissez `header: "x-omnihss-key"`, afin que les deux clés ne se chevauchent pas.

OmniHLR et OmniUDR (Interopérabilité HLR)

```
hlr_interworking: [
  "subscriber/imsi:read",
  "subscriber/msisdn:read",
  "key_set:read",
  "hlr/generate_auth_vectors:write",
  "hlr/generate_eutran_auth_vectors:write",
  "hlr/circuit_session:*",
  "status/api:read"
]
```

Comment ça fonctionne : Le HLR et le UDR recherchent des abonnés, demandent des vecteurs d'authentification et mettent à jour la localisation commutée par circuit. Ils ne peuvent pas modifier les données ou les profils des abonnés.

Cas d'utilisation : OmniHLR (OmniSS7) pour 2G/3G, et OmniUDR pour 5G. Ces systèmes reçoivent des vecteurs contenant CK et IK. Ne donnez pas ces clés à d'autres systèmes.

Surveillance (Lecture seule)

```
monitoring: [
  "status/**:read",
  "subscriber/**:read",
  "operation/count:read"
]
```

Comment ça fonctionne : La clé peut lire l'état, les recherches d'abonnés et les comptes d'enregistrements. Elle ne peut pas modifier les données.

Cas d'utilisation : Exportateurs, tableaux de bord et rapports. Les lectures des abonnés contiennent des IMSIs et des MSISDN, qui sont des données personnelles. Si un outil a seulement besoin de l'état, utilisez `"status/**:read"` seul.

OmniEIR (Recherche EIR)

```
eir_lookup: ["eir/rule:read"]
```

Comment ça fonctionne : La clé peut uniquement lire les règles EIR.

Cas d'utilisation : OmniEIR lit les règles que stocke OmniHSS.

Outils de laboratoire et de test

Les outils de test, par exemple le simulateur RAN, créent et suppriment des abonnés et des profils. Dans un laboratoire, donnez à chaque outil de test sa propre clé avec le rôle `admin`. Ne pas installer de clés d'outil de test sur des systèmes de production.

Exemple de configuration complète

Cet exemple utilise tous les Rôles suggérés. Chaque paramètre est expliqué dans Configuration.

```
# runtime.exs
config :api_ex, :auth,
  # Ne pas utiliser x-api-key : OmniWeb envoie son propre en-tête
X-API-Key via son proxy
  header: "x-omnihss-key",
  user_header: "x-omnihss-user",

  # Vérification de l'état sans clé
  public: ["status/api:read"],

  roles: %{
    admin: ["*"],
    crm_provisioning: [
      "subscriber:*",
      "subscriber/*:read",
      "msisdn/**:*",
      "sim/**:read",
      "sim:write",
      "key_set:write",
      "status/**:read"
    ],
    oam: [
      "*:read",
      "subscriber/**:*",
      "msisdn/**:*",
      "sim/**:*",
      "key_set:*",
      "ims_subscription:*",
      "apn/**:*",
      "epc/**:*",
      "ims/**:*",
      "roaming/**:*",
      "eir/**:*",
      "pcrf/**:*",
      "operation/**:write",
      "hlr/subscriber_state:write",
      "hlr/generate_auth_vectors:write",
      "hlr/generate_eutran_auth_vectors:write"
    ],
    hlr_interworking: [
      "subscriber/imsi:read",
      "subscriber/msisdn:read",
      "key_set:read",
```

```
      "hlr/generate_auth_vectors:write",
      "hlr/generate_eutran_auth_vectors:write",
      "hlr/circuit_session:*",
      "status/api:read"
    ],
    monitoring: ["status/**:read", "subscriber/**:read",
"operation/count:read"],
    eir_lookup: ["eir/rule:read"]
  },

  api_keys: [
    %{
      name: "omnicrm",
      roles: [:crm_provisioning],
      may_assert_user: true,
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNICRM-KEY-00"]
    },
    %{
      name: "omniweb",
      roles: [:oam],
      may_assert_user: true,
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNIWEB-KEY-00"]
    },
    %{
      name: "omnihlr",
      roles: [:hlr_interworking],
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNIHLR-KEY-00"]
    },
    %{
      name: "omniudr",
      roles: [:hlr_interworking],
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNIUDR-KEY-00"]
    },
    %{
      name: "monitoring",
      roles: [:monitoring],
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
MONITOR-KEY-00"]
    },
    %{
```

```
      name: "break-glass",
      roles: [:admin],
      metadata: %{owner: "NOC", vault: "omnihss/break-glass"},
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-
BREAK-GLASS-KEY-00"]
    }
  ],

  log: [deny: :warning, unauthenticated: :info, allow: false,
not_found: false]
```

Comment ça fonctionne : OmniHSS démarre en mode RBAC, car `api_keys` est présent. Chaque système client a sa propre clé. Les équilibreurs de charge peuvent vérifier `GET /api/status/api` sans clé. Tous les autres itinéraires nécessitent une clé, et la clé doit avoir un rôle qui correspond à l'itinéraire.

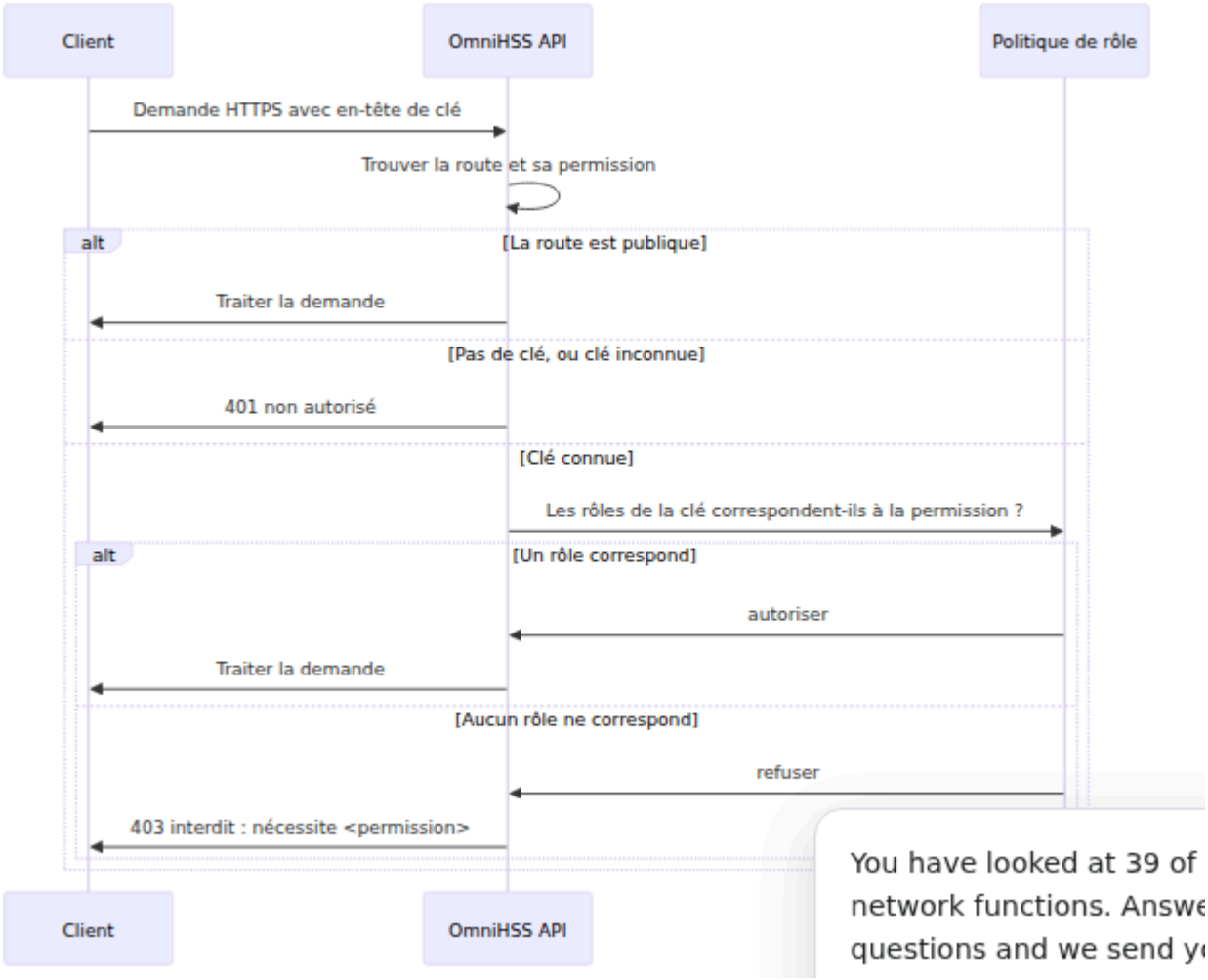
Lorsque le service démarre, le journal affiche une ligne de résumé. La ligne donne le mode, les noms des en-têtes, les noms des clés, les rôles et le nombre d'avertissements. Elle ne montre jamais de secret. Par exemple :

```
[info] ApiEx: authentication par clé partagée activée (mode :
rbac, en-tête : x-omnihss-key, user_header : x-omnihss-user,
skip_paths : [])
  clés : omnicrm [crm_provisioning, affirme l'utilisateur],
omniweb [oam, affirme l'utilisateur], omnihlr [hlr_interworking],
...
```

Vérifiez cette ligne après chaque changement. Si la ligne affiche `mode: legacy` ou un avertissement concernant une authentification désactivée, le paramètre `api_keys` n'est pas chargé.

Utilisateur agissant

Un système client peut agir pour un utilisateur humain. Par exemple, un utilisateur OmniWeb modifie un abonné, et OmniWeb envoie la demande avec sa propre clé. Le client peut envoyer le nom de l'utilisateur dans `user_header`.



Règles :

- OmniHSS accepte le nom d'utilisateur uniquement si la clé a `may_assert_user: true`. Pour d'autres clés, OmniHSS ignore l'en-tête.
- Le nom d'utilisateur doit être un UTF-8 valide, au maximum 256 octets, et ne doit contenir aucun caractère de contrôle.
- L'utilisateur agissant est un enregistrement d'audit. Ce n'est pas une connexion. OmniHSS fait confiance au système client pour envoyer le nom d'utilisateur correct. Ainsi, définissez `may_assert_user: true` uniquement pour les systèmes qui ont leur propre connexion utilisateur.

Appeler l'API avec une clé

Envoyez la clé dans l'en-tête configuré. Envoyez l'en-tête une seule fois. Si la demande a deux copies de l'en-tête, OmniHSS la refuse.

```
curl -k
https://hss.example.com:8443/api/subscriber/imsi/001010000000001 \
  -H "x-omnihss-key: <secret>"
```

Dans Swagger UI (`/api/docs`), cliquez sur **Autoriser** et collez le secret. Swagger UI envoie ensuite la clé avec chaque demande "Essayez-le". La clé a besoin de `docs:read` et `schema:read` pour ouvrir Swagger UI. Le modèle `*:read` contient les deux.

Codes de réponse

Toutes les erreurs utilisent le format API habituel `{"status":"error","response":"..."}`.

Statut	`response`	Cause
`401`	`unauthorized`	Pas de clé, une mauvaise clé ou deux copies de l'en-tête de clé.
`403`	`forbidden: requires <permission>`	Les rôles de la clé ne correspondent pas à la permission de l'itinéraire. La réponse nomme la permission.
`403`	`forbidden: acting user required`	L'itinéraire nécessite un utilisateur agissant, et la demande n'en a pas nommé un.
`503`	`authentication misconfigured`	La configuration n'est pas valide, ou elle a changé pour un mode inférieur pendant que le service était en cours d'exécution.
`404`	`Not Found`	Une clé connue a demandé un chemin qui n'existe pas.

Activer RBAC sur un système en cours d'exécution

Lorsque vous activez RBAC, chaque client qui n'envoie pas une clé valide reçoit `401`. Ainsi, préparez tous les clients avant de redémarrer OmniHSS.

1. Faites une liste de chaque système qui appelle l'API OmniHSS à ce site.
2. Sélectionnez un rôle parmi les Rôles suggérés pour chaque système.
3. Créez un secret pour chaque système avec `openssl rand -hex 32`.
4. Configurez chaque système client pour envoyer son secret dans l'en-tête `x-omnihss-key`.
5. Ajoutez le bloc `config :api_ex, :auth` à `runtime.exs` sur chaque hôte OmniHSS.
6. Redémarrez OmniHSS sur un hôte.
7. Vérifiez la ligne de journal de démarrage. Assurez-vous qu'elle affiche `mode: rbac` et les noms de clés corrects.
8. Recherchez les lignes `403` et `401` dans le journal. Chaque ligne `403` nomme la clé et la permission dont elle a besoin.
9. Si un client a besoin d'une permission que son rôle n'a pas, ajoutez le modèle au rôle. Ensuite, redémarrez OmniHSS.
10. Répétez les étapes 6 à 9 pour chaque autre hôte OmniHSS.

Si un système client ne peut pas encore envoyer un en-tête de clé, ne pas activer RBAC sur les hôtes que le client utilise. Ce client reçoit `401` pour chaque demande.

Dans un déploiement multi-sites, utilisez le même bloc `config :api_ex, :auth` sur tous les hôtes OmniHSS. Un client peut alors utiliser la même clé sur chaque site.

Faire tourner une clé

Une clé peut avoir plus d'un secret. Utilisez cela pour changer un secret sans temps d'arrêt.

1. Créez un nouveau secret avec `openssl rand -hex 32`.
2. Ajoutez le nouveau secret à la liste `secrets` de la clé. Gardez l'ancien secret dans la liste.
3. Redémarrez OmniHSS. Les deux secrets fonctionnent maintenant.
4. Modifiez le système client pour envoyer le nouveau secret.
5. Supprimez l'ancien secret de la liste `secrets`.
6. Redémarrez OmniHSS. Seul le nouveau secret fonctionne.

```
%{
  name: "omnicrm",
  roles: [:crm_provisioning],
  may_assert_user: true,
  secrets: [
    "REPLACE-WITH-THE-OLD-OMNICRM-SECRET-UNTIL-THE-CRM-USES-THE-
NEW-1",
    "REPLACE-WITH-THE-NEW-OMNICRM-SECRET-FROM-OPENSSL-RAND-HEX-32-
002"
  ]
}
```

Pour supprimer tout accès pour un système, retirez sa clé de `api_keys`. Ensuite, redémarrez OmniHSS.

Journalisation et audit

- Chaque ligne de journal qu'OmniHSS écrit après avoir identifié une clé contient `api_ex_principal=<nom de la clé>`. Utilisez cela pour trouver quel système a effectué un changement.
- Une ligne `403` donne le nom de la clé, le modèle d'itinéraire et la permission dont la clé a besoin. La ligne donne le modèle d'itinéraire, par exemple `/api/subscriber/:imsi`, et non le chemin complet.
- Les secrets n'apparaissent jamais dans les lignes de journal ou dans les réponses d'erreur.
- Pour journaliser chaque demande autorisée, définissez `log: [allow: :info]`. Cela donne une trace d'audit de tous les changements d'API, mais

cela ajoute une ligne de journal pour chaque demande.

Dépannage

Tous les clients obtiennent 401 après un redémarrage

Symptômes : Chaque client obtient `401 unauthorized`.

Causes possibles :

- Les clients n'envoient pas l'en-tête de clé.
- Les clients envoient la clé dans un en-tête différent, par exemple `x-api-key`.
- Les secrets dans les clients ne correspondent pas aux secrets dans `runtime.exs`.

Résolution :

1. Vérifiez la ligne de journal de démarrage pour le nom de l'en-tête configuré.
2. Assurez-vous que chaque client envoie sa clé dans cet en-tête.
3. Comparez le secret dans la configuration du client avec le secret dans `api_keys`.

Un client obtient 403

Symptômes : Un client obtient `403 forbidden: requires <permission>`.

Causes possibles :

- Le rôle de la clé ne contient pas de modèle pour cette permission.
- Le modèle utilise `*` là où `**` est nécessaire. Voir Exemples de modèles.

Résolution :

1. Trouvez la permission dans la réponse ou dans la ligne de journal.

2. Trouvez l'itinéraire dans le Catalogue des permissions OmniHSS.
3. Si le client doit utiliser l'itinéraire, ajoutez un modèle pour la permission au rôle de la clé.
4. Redémarrez OmniHSS.

L'API répond 503 à chaque demande

Symptômes : Chaque demande obtient `503 authentication misconfigured`.

Causes possibles :

- Le paramètre `key` hérité a une valeur vide.
- La configuration a changé pour un mode inférieur pendant que le service était en cours d'exécution.

Résolution :

1. Regardez dans le journal la ligne qui énumère les problèmes de configuration.
2. Corrigez le bloc `config :api_ex, :auth`.
3. Redémarrez OmniHSS.

OmniHSS ne démarre pas après le changement

Symptômes : OmniHSS s'arrête au démarrage. Le journal affiche `ApiEx: invalid RBAC configuration`.

Causes possibles :

- Une clé utilise un rôle qui n'est pas défini dans `roles`.
- Deux clés ont le même `name`.
- Deux clés partagent un secret.
- `user_header` est identique à `header`.
- Une clé n'a pas de secret.

Résolution :

1. Lisez la liste numérotée de problèmes sous la ligne de journal.

2. Corrigez chaque problème dans `runtime.exs`.
3. Redémarrez OmniHSS.

Le journal de démarrage affiche un avertissement

Symptômes : Le journal de démarrage contient des lignes `ApiEx auth warning:`.

Causes possibles :

- Un secret est plus court que 16 octets.
- Un modèle de rôle ne correspond à aucun itinéraire. C'est généralement une faute de frappe dans le modèle.
- Un modèle `public` ouvre un itinéraire qui peut modifier des données.

Résolution :

1. Lisez chaque ligne d'avertissement.
2. Créez des secrets longs avec `openssl rand -hex 32`.
3. Corrigez l'orthographe du modèle. Comparez-le avec le Catalogue des permissions OmniHSS.
4. Gardez les modèles `public` en lecture seule.

Référence API OmniHSS

← Retour au Guide des Opérations

Table des Matières

- Aperçu de l'API
- Authentification et Contrôle d'Accès
- Gestion des Clés
- Gestion des Abonnés
- Données du Répertoire Sh (Données Transparentes)
- Gestion des MSISDN
- Gestion des Abonnements IMS
- Gestion des SIM
- Gestion des Clés
- Gestion des Profils
- Gestion des IP Statique
- Gestion de l'Itinérance
- Gestion de l'EIR
- Statut et Santé
- Gestion des Erreurs
- Exemples d'Utilisation de l'API

Aperçu de l'API

URL de Base

```
https://[hostname]:8443/api
```

Authentification

Chaque requête doit contenir une clé API dans l'en-tête de clé configuré, par exemple `x-omnihss-key`. Chaque clé a des rôles qui contrôlent les routes qu'elle peut utiliser. Pour la configuration, les permissions de chaque route et les rôles suggérés pour chaque système client, voir Contrôle d'Accès API (RBAC).

Format de la Requête

- **Content-Type :** `application/json`
- **Protocole :** HTTPS uniquement
- **Port :** 8443 (configurable)

Important : Tous les points de terminaison de l'API attendent des charges utiles JSON "plates" sans objets enveloppants.

Format Correct :

```
{
  "name": "value",
  "field": "value"
}
```

Format Incorrect (À Ne Pas Utiliser) :

```
{
  "subscriber": {
    "name": "value",
    "field": "value"
  }
}
```

Exemple :

```
# ✓ Correct
curl -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{"name": "default", "ifc_template": "..."}'

# ✗ Incorrect
curl -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{"ims_profile": {"name": "default", "ifc_template": "..."}}'
```

Format de la Réponse

Toutes les réponses sont au format JSON avec la structure suivante :

Réponse de Succès :

```
{
  "status": "success",
  "response": { ... }
}
```

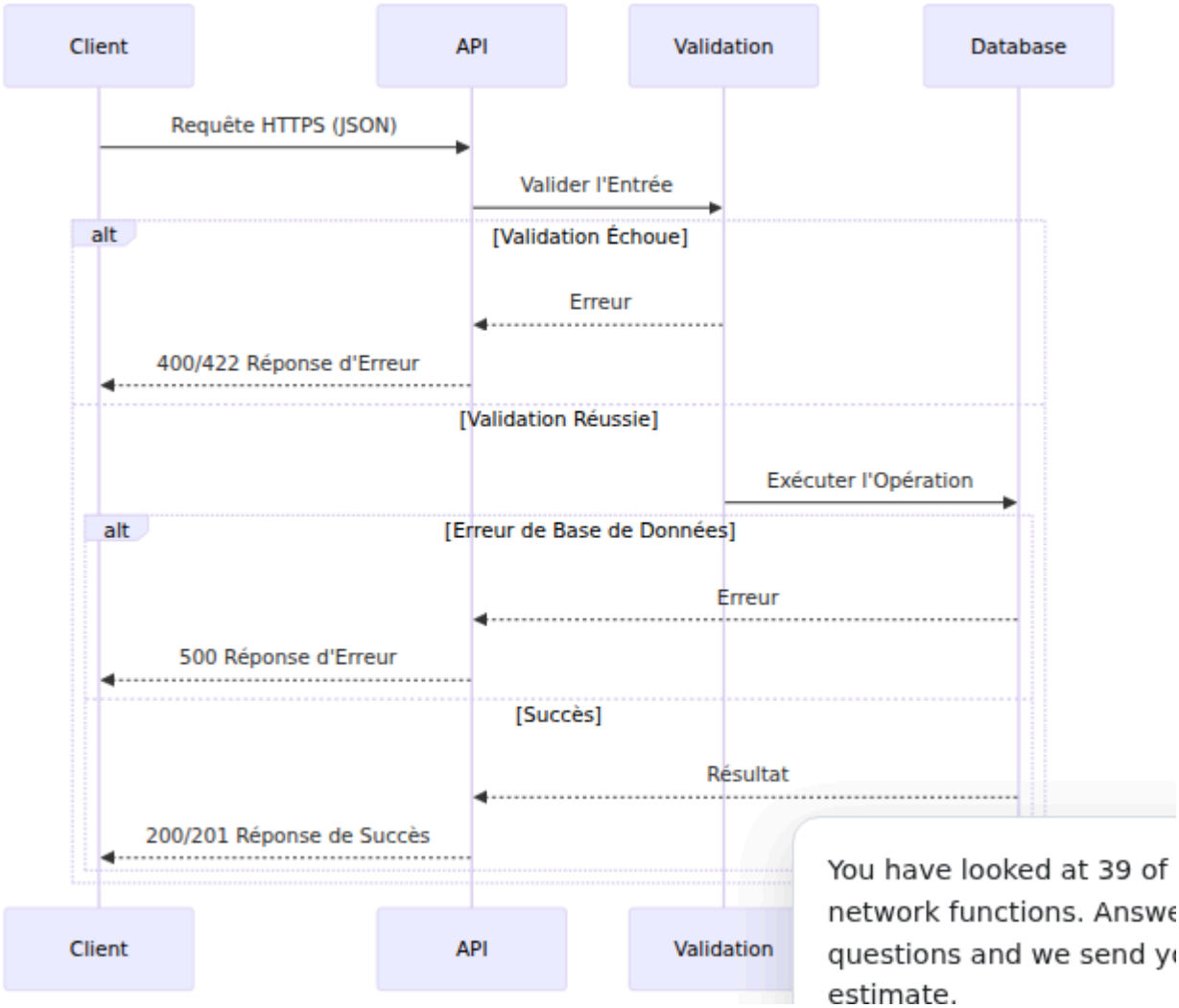
Réponse d'Erreur :

```
{
  "status": "error",
  "response": {
    "invalid_fields": {
      "field_name": "message d'erreur"
    }
  }
}
```

Codes d'État HTTP

Code	Signification	Cas d'Utilisation
200	OK	GET, PUT, DELETE réussi
201	Créé	POST réussi
400	Mauvaise Requête	Données d'entrée invalides
401	Non Autorisé	Pas de clé API, ou la clé n'est pas connue. Voir Contrôle d'Accès API
403	Interdit	Les rôles de la clé ne permettent pas la route. Voir Contrôle d'Accès API
404	Non Trouvé	La ressource n'existe pas
422	Entité Non Traitable	Erreur de validation
500	Erreur Interne du Serveur	Erreur côté serveur
503	Service Indisponible	La configuration du contrôle d'accès API n'est pas valide. Voir Contrôle d'Accès API

Flux de Requête API



Gestion des Abonnés

Lister les Abonnés

Récupérer tous les abonnés ou filtrer par critères.

Point de Terminaison : `GET /api/subscriber`

Paramètres de Requête :

Paramètre	Type	Description
`enabled`	boolean	Filtrer par statut activé
`ims_enabled`	boolean	Filtrer par statut IMS activé

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/subscriber
```

Exemple de Réponse :


```
{
  "data": [
    {
      "id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
      "imsi": "001001123456789",
      "enabled": true,
      "ims_enabled": true,
      "sim_id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
      "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
      "ims_profile_id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
      "roaming_profile_id": "9c858901-8a57-4791-81fe-
4c455b099bc9",
      "custom_attributes": {},
      "inserted_at": "2025-10-15T10:30:00Z",
      "updated_at": "2025-10-15T10:30:00Z"
    }
  ]
}
```


Obtenir un Abonné par ID

Récupérer un abonné spécifique par ID de base de données.

Point de Terminaison : `GET /api/subscriber/:id`

Paramètres de Chemin :

Paramètre	Type	Description
`id`	string (UUID)	ID de base de données de l'abonné

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-
4b01-90e6-d701748f0851
```

Obtenir un Abonné par IMSI

Récupérer un abonné par son IMSI.

Point de Terminaison : `GET /api/subscriber/imsi/:imsi`

Paramètres de Chemin :

Paramètre	Type	Description	Format
`imsi`	string	Identité Internationale d'Abonné Mobile	14-15 chiffres

Exemple de Requête :

```
curl -k
https://hss.example.com:8443/api/subscriber/imsi/001001123456789
```

Cas d'Utilisation : Dépannage d'un abonné spécifique par son IMSI.

Obtenir un Abonné par MSISDN

Récupérer un abonné par son numéro de téléphone.

Point de Terminaison : `GET /api/subscriber/msisdn/:msisdn`

Paramètres de Chemin :

Paramètre	Type	Description	Format
`msisdn`	string	Numéro ISDN de la Station Mobile	1-15 chiffres (E.164)

Exemple de Requête :

```
curl -k
https://hss.example.com:8443/api/subscriber/msisdn/14155551234
```

Cas d'Utilisation : Recherche d'informations sur un abonné lorsque vous n'avez que son numéro de téléphone.

Créer un Abonné

Provisionner un nouvel abonné.

Point de Terminaison : `POST /api/subscriber`

Corps de la Requête :

```
{
  "subscriber": {
    "imsi": "001001123456789",
    "enabled": true,
    "ims_enabled": true,
    "sim_id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
    "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
    "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
    "ims_profile_id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
    "roaming_profile_id": "9c858901-8a57-4791-81fe-4c455b099bc9",
    "custom_attributes": {
      "note": "Abonné de test"
    }
  }
}
```

Champs Requis :

- `imsi` - Doit être de 14 à 15 chiffres, unique
- `key_set_id` - Doit référencer un Ensemble de Clés existant
- `epc_profile_id` - Doit référencer un Profil EPC existant

Champs Optionnels :

- `enabled` - Par défaut : true
- `ims_enabled` - Par défaut : true
- `sim_id` - Référence à une carte SIM
- `ims_profile_id` - Référence à un Profil IMS (requis pour les services IMS)
- `roaming_profile_id` - Référence à un Profil d'Itinérance (requis pour le contrôle d'itinérance)
- `msisdns` - Tableau d'IDs de MSISDN (numéros de téléphone)
- `static_ips` - Tableau d'IDs de IP Statique pour les affectations APN
- `ims_subscription_id` - Référence à un Abonnement IMS pour regrouper plusieurs abonnés sous une identité publique partagée et S-CSCF
- `custom_attributes` - Paires clé-valeur personnalisées

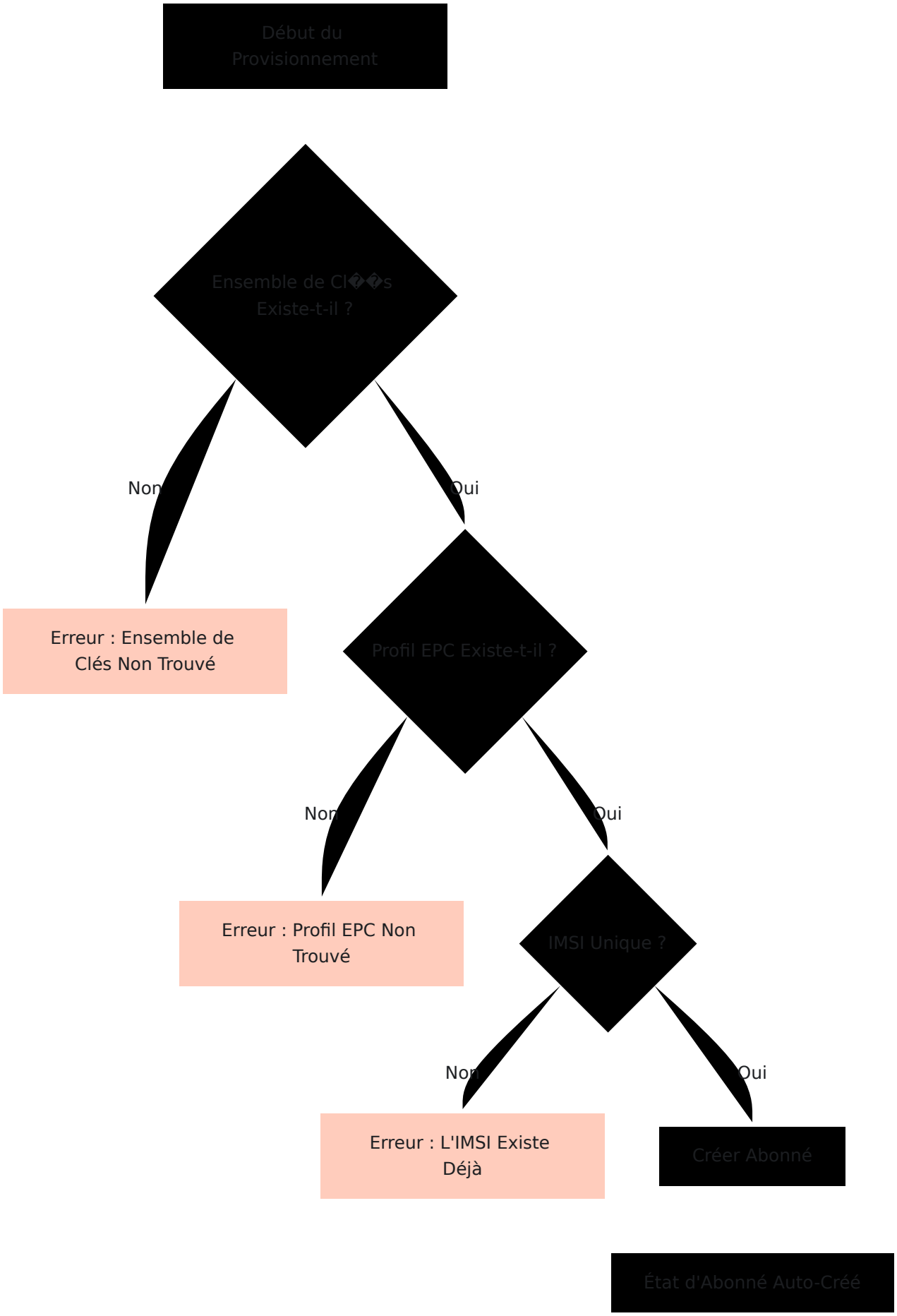
Voir Aussi :

- Exemple Complet de Provisionnement d'Abonné - Flux de travail de bout en bout
- Documentation Multi-MSISDN - Attribution de numéros de téléphone aux abonnés
- Gestion des IP Statique - Attribution d'IP statiques aux APNs

Exemple de Requête :

```
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3"
    }
  }'
```

Flux de Provisionnement :


Début du Provisionnement
Ensemble de Cl??s Existe-t-il ?
Non
Oui
Erreur : Ensemble de Clés Non Trouvé
Profil EPC Existe-t-il ?
Non
Oui
Erreur : Profil EPC Non Trouvé
IMSI Unique ?
Non
Oui
Erreur : L'IMSI Existe Déjà
Créer Abonné
État d'Abonné Auto-Créé

201 Créé

Mettre à Jour un Abonné

Modifier un abonné existant.

Point de Terminaison : `PUT /api/subscriber/:id`

Paramètres de Chemin :

Paramètre	Type	Description
`id`	string (UUID)	ID de base de données de l'abonné

Corps de la Requête :

```
{
  "subscriber": {
    "enabled": false,
    "ims_enabled": false,
    "epc_profile_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "custom_attributes": {
      "note": "Désactivé temporairement"
    }
  }
}
```

Champs Modifiables :

- `enabled` - Activer/désactiver tous les services
- `ims_enabled` - Activer/désactiver les services IMS
- `sim_id` - Changer l'attribution de la carte SIM
- `key_set_id` - Changer les clés cryptographiques (soyez prudent !)
- `epc_profile_id` - Changer le profil de service de données
- `ims_profile_id` - Changer le profil de service vocal
- `roaming_profile_id` - Changer la politique d'itinérance

- `msisdns` - Mettre à jour les numéros de téléphone attribués à l'abonné (`[]` supprime tout)
- `static_ips` - Mettre à jour les affectations d'IP statique aux APNs (`[]` supprime tout)
- `custom_attributes` - Mettre à jour les données personnalisées

Non Modifiables :

- `imsi` - Ne peut pas changer l'IMSI (supprimer et recréer à la place)
- `subscriber_state` - État d'exécution écrit par le HSS ; il ne peut pas être défini via ce point de terminaison. Définir ou effacer `assigned_scscf` avec `PATCH /api/hlr/subscriber_state/:imsi`

Voir Aussi :

- Gestion des Profils - Gestion des profils de service

Exemple de Requête :

```
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "enabled": false
    }
  }'
```

Cas d'Utilisation :

- Désactiver temporairement l'abonné : `{"enabled": false}`
- Désactiver uniquement les services vocaux : `{"ims_enabled": false}`
- Changer le profil de service : `{"epc_profile_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479"}` (voir Profils EPC)
- Mettre à jour la politique d'itinérance : `{"roaming_profile_id": "1b671a64-40d5-491e-99b0-da01ff1f3341"}` (voir Gestion de l'Itinérance)

Supprimer un Abonné

Retirer un abonné du système.

Point de Terminaison : `DELETE /api/subscriber/:id`

Paramètres de Chemin :

Paramètre	Type	Description
`id`	string (UUID)	ID de base de données de l'abonné

Exemple de Requête :

```
curl -k -X DELETE
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851
```

Avertissement : Cela supprime définitivement l'abonné et toutes les données d'état associées (sessions PDN, appels, etc.). L'IMSI peut être réutilisé après suppression.

Remarque : Supprimer un abonné ne supprime PAS les éléments associés :

- Ensemble de Clés - Peut être réutilisé pour d'autres abonnés
- SIM - Peut être réaffectée à un nouvel abonné
- Profils - Ressources partagées utilisées par plusieurs abonnés
- MSISDNs - Doivent être supprimés séparément si désiré

Annuler la Demande de Localisation (Détachement Forcé)

Envoyer une Demande d'Annulation de Localisation (CLR) pour forcer le détachement d'un abonné de son MME actuellement enregistré.

Point de Terminaison : `POST /api/subscriber/cancel_location`

Corps de la Requête :

```
{
  "imsi": "001001123456789"
}
```

Paramètres :

Paramètre	Type	Requis	Description
`imsi`	string	Oui	IMSI de l'abonné à détacher (14-15 chiffres)

Exemple de Requête :

```
curl -k -X POST
https://hss.example.com:8443/api/subscriber/cancel_location \
  -H "Content-Type: application/json" \
  -d '{"imsi": "001001123456789"}'
```

Réponse de Succès (200 OK) :

```
{
  "data": {
    "message": "Demande d'Annulation de Localisation envoyée avec
succès",
    "imsi": "001001123456789",
    "destination_host": "mme01.operator.com",
    "destination_realm": "epc.operator.com"
  }
}
```

Réponse d'Erreur (404 Non Trouvé) :

```
{
  "error": "Abonné non trouvé ou non actuellement enregistré à un
MME"
}
```

Comportement :

- Envoie un CLR S6a au MME où l'abonné est actuellement enregistré (`subscriber_state.last_seen_mme`)
- Utilise `Cancellation-Type: subscription_withdrawal` (force le détachement complet)
- Définit `CLR-Flags: {s6a_indicator: 1, reattach_required: 1}` (l'UE doit se ré-authentifier)
- Retourne 404 si l'abonné n'a jamais été enregistré ou si `last_seen_mme` est nul
- **Affecte tous les MSISDNs** associés à l'IMSI (même appareil physique/SIM)

Cas d'Utilisation :

- **Prévention de Fraude :** Détacher immédiatement un abonné suspect
- **Résiliation d'Abonnement :** Forcer la déconnexion lorsque le compte est désactivé
- **Dépannage :** Effacer l'enregistrement MME obsolète pour le débogage
- **Migration :** Forcer la ré-authentification pour appliquer de nouveaux paramètres de profil
- **Sécurité :** Déconnecter immédiatement un abonné compromis

Considérations Multi-IMSI :

Lors de l'utilisation de CLR avec des scénarios multi-MSISDN :

1. **Plusieurs MSISDNs, Un Seul IMSI :**

```
// L'abonné a l'IMSI 001001123456789 avec des MSISDNs
["+1234567890", "+9876543210"]
POST /api/subscriber/cancel_location
{"imsi": "001001123456789"}

// Résultat : Un CLR envoyé, les deux MSISDNs affectés (même
appareil)
```

2. **Différents IMSIs (Appareils Différents) :**

```
// Deux abonnés avec le même MSISDN mais différents IMSIs
(scénario de portabilité de numéro)
// Abonné A : IMSI 001001111111111, MSISDN "+1234567890"
// Abonné B : IMSI 001001222222222, MSISDN "+1234567890"

POST /api/subscriber/cancel_location
{"imsi": "001001111111111"}

// Résultat : Seul l'Abonné A détaché, l'Abonné B non affecté
```

Remarques Importantes :

- **Basé sur l'IMSI :** CLR est toujours envoyé par IMSI, pas par MSISDN
- **Asynchrone :** CLR est envoyé de manière asynchrone ; la réponse de succès signifie que le CLR a été envoyé, pas que le MME l'a traité
- **Pas de validation de l'état du MME :** CLR est envoyé même si le MME est injoignable (comportement standard du HSS)
- **Idempotent :** Sûr d'appeler plusieurs fois pour le même IMSI

Documentation Connexe :

- Flux de Protocole de Demande d'Annulation de Localisation
- Scénarios Multi-IMSI
- Architecture de l'Interface S6a

Données du Répertoire Sh (Données Transparentes)

Provisionnement d'accès aux documents transparents Sh d'un abonné (MMTEL-Services, IMS-ODB-Information, et autres). Ce sont les mêmes documents opaques qu'un Serveur d'Application lit et écrit via l'interface Diameter Sh. L'accès se fait via `GET`/`PUT`/`DELETE /api/subscriber/repository_data/:imsi`.

Voir Données du Répertoire Sh (Données Transparentes) pour le modèle de données, les flux de messages, les règles de concurrence SequenceNumber, et la référence complète de l'API REST.

Gestion des MSISDN

Les MSISDN (numéros de téléphone) peuvent être attribués aux abonnés pour activer les services vocaux. Voir Documentation Multi-MSISDN pour des détails sur l'attribution de plusieurs numéros à un seul abonné.

Le drapeau `shared`

Les réponses de lecture qui portent un MSISDN incluent un `shared` booléen calculé, en lecture seule. Il est `true` lorsque le numéro est détenu par **plus d'un abonné** (un numéro partagé multi-IMSI, par exemple un téléphone et une montre connectée sur des SIMs séparées), et `false` sinon. Il apparaît sur l'entité MSISDN (`GET /api/msisdn` et `GET /api/msisdn/:id`), sur chaque numéro intégré dans un abonné (`GET /api/subscriber/:id`), et sur les numéros des abonnés retournés par `GET /api/subscriber/msisdn/:msisdn`. Il n'est jamais accepté lors de l'écriture ; le HSS le calcule. Voir Résolution MSISDN à Travers les Interfaces.

Lister les MSISDN

Récupérer tous les numéros de téléphone.

Point de Terminaison : `GET /api/msisdn`

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/msisdn
```

Obtenir un MSISDN

Récupérer un numéro de téléphone spécifique.

Point de Terminaison : `GET /api/msisdn/:id`

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/msisdn/16fd2706-8baf-433b-82eb-8c7fada847da
```

Exemple de Réponse :

```
{
  "status": "success",
  "response": {
    "id": "16fd2706-8baf-433b-82eb-8c7fada847da",
    "msisdn": "14155551234",
    "shared": false
  }
}
```

Créer un MSISDN

Créer un nouveau numéro de téléphone.

Point de Terminaison : `POST /api/msisdn`

Corps de la Requête :

```
{
  "msisdn": {
    "msisdn": "14155551234"
  }
}
```

Validation :

- Doit être de 1 à 15 chiffres
- Doit être unique
- Doit suivre le format E.164 (format international sans le signe +)

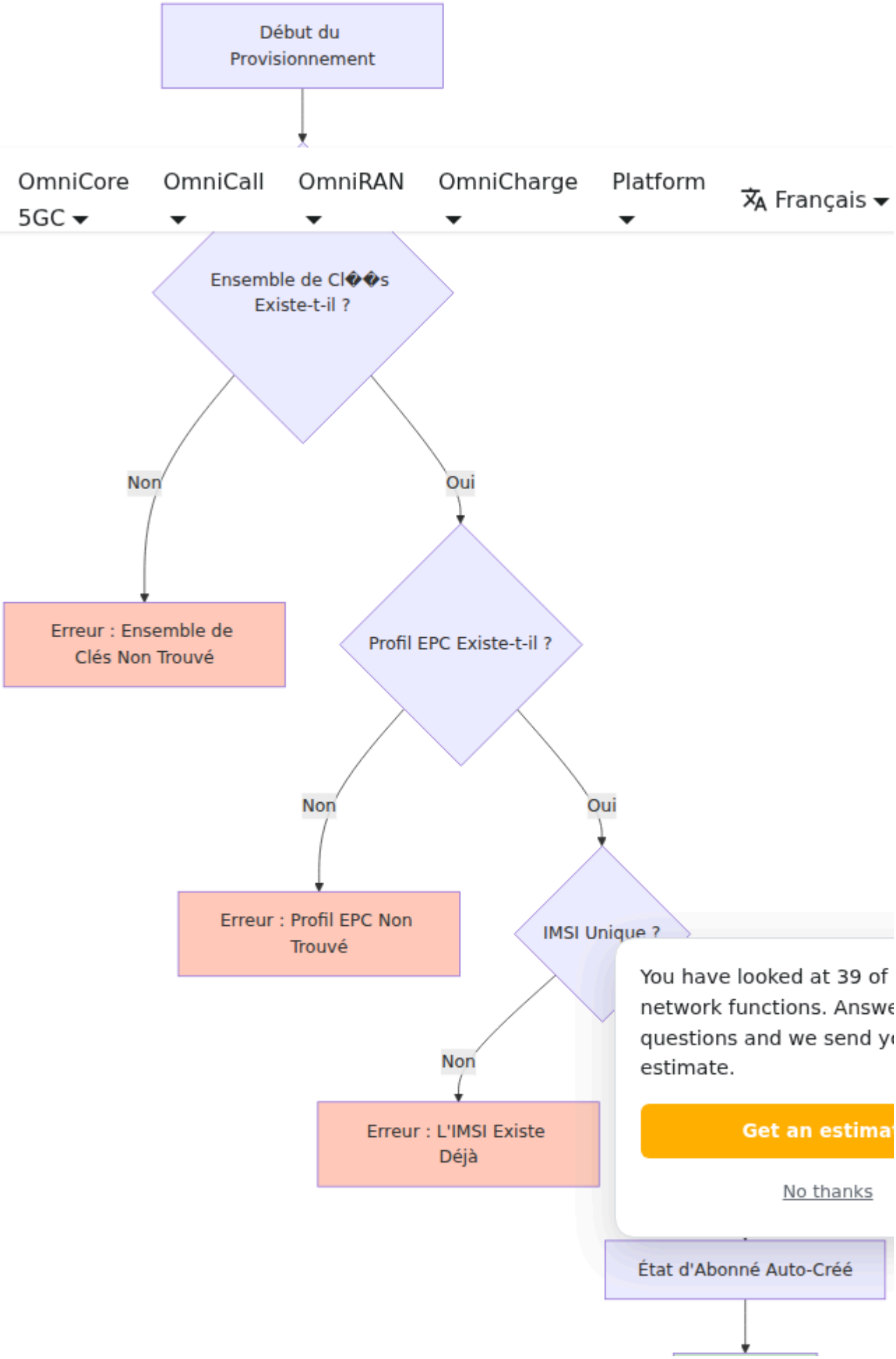
Exemple de Requête :

```
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{
    "msisdn": {
      "msisdn": "14155551234"
    }
  }'
```

Attribuer un MSISDN à un Abonné

Pour attribuer un numéro de téléphone à un abonné, vous devez créer un enregistrement de jointure. Cela se fait généralement via le point de terminaison de mise à jour de l'abonné ou par manipulation directe de la base de données.

Modèle Multi-MSISDN :


Début du Provisionnement
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Français
Ensemble de Cl��s Existe-t-il ?
Non
Oui
Erreur : Ensemble de Clés Non Trouvé
Profil EPC Existe-t-il ?
Non
Oui
Erreur : Profil EPC Non Trouvé
IMSI Unique ?
Non
Erreur : L'IMSI Existe Déjà
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks
État d'Abonné Auto-Créé



Voir Fonctionnalités Multi-MSISDN et Multi-IMSI pour une utilisation détaillée.

Supprimer un MSISDN

Retirer un numéro de téléphone.

Point de Terminaison : `DELETE /api/msisdn/:id`

Exemple de Requête :

```
curl -k -X DELETE
https://hss.example.com:8443/api/msisdn/16fd2706-8baf-433b-82eb-
8c7fada847da
```

Gestion des Abonnements IMS

Un Abonnement IMS regroupe plusieurs abonnés (IMSIs / IMPIs) en un seul abonnement IMS qui partage une identité publique et un S-CSCF de service unique. Voir Identité Publique Utilisateur Partagée.

Lister les Abonnements IMS

Point de Terminaison : `GET /api/ims_subscription`

```
curl -k https://hss.example.com:8443/api/ims_subscription
```

Obtenir un Abonnement IMS

Point de Terminaison : `GET /api/ims_subscription/:id`

```
curl -k
https://hss.example.com:8443/api/ims_subscription/16fd2706-8baf-
433b-82eb-8c7fada847da
```

Créer un Abonnement IMS

Point de Terminaison : `POST /api/ims_subscription`

```
curl -k -X POST https://hss.example.com:8443/api/ims_subscription
\
  -H "Content-Type: application/json" \
  -d '{"name": "Famille Smith - numéro partagé"}'
```

Champs :

- `name` - Nom lisible par l'homme (requis, unique)
- `ims_profile_id` - Profil IMS partagé optionnel pour les identités partagées du groupe

`assigned_scscf` et `last_seen_scscf_timestamp` sont des états d'exécution écrits par le HSS et ne sont pas définis par le client.

Mettre à Jour un Abonnement IMS

Point de Terminaison : `PATCH /api/ims_subscription/:id`

Supprimer un Abonnement IMS

Point de Terminaison : `DELETE /api/ims_subscription/:id`

La suppression d'un Abonnement IMS ne supprime pas ses abonnés ; leur `ims_subscription_id` est effacé. Pour ajouter un abonné à un groupe, définissez `ims_subscription_id` sur l'abonné (voir Mettre à Jour un Abonné).

Gestion des SIM

Les enregistrements de carte SIM stockent des informations physiques sur la carte SIM, y compris l'ICCID, les détails du fournisseur, les codes PIN/PUK et les clés OTA. Les enregistrements de SIM peuvent être liés en option à des abonnés.

Voir aussi :

- Documentation Multi-IMSI - Plusieurs abonnés sur une seule carte SIM physique

Lister les SIMs

Récupérer toutes les cartes SIM.

Point de terminaison : `GET /api/sim`

Exemple de requête :

```
curl -k https://hss.example.com:8443/api/sim
```

Obtenir une SIM

Récupérer une carte SIM spécifique. `adm1`-`adm10`, `kic` et `kid` sont en écriture seule : ils ne sont retournés que dans la réponse à la requête de création ou de mise à jour qui les a définis, et jamais dans une autre réponse API.

Point de terminaison : `GET /api/sim/:id`

Exemple de requête :

```
curl -k https://hss.example.com:8443/api/sim/3f2504e0-4f89-41d3-9a0c-0305e82c3301
```

Créer une SIM

Créer un nouvel enregistrement de carte SIM.

Point de terminaison : `POST /api/sim`

Corps de la requête :

```
{
  "sim": {
    "iccid": "8991101200003204510",
    "sim_vendor": "Gemalto",
    "batch_name": "2025-Q1-Batch-01",
    "is_esim": false,
    "pin1": "1234",
    "pin2": "5678",
    "puk1": "12345678",
    "puk2": "87654321",
    "adm1": "admin-code-1",
    "kic": "0123456789ABCDEF0123456789ABCDEF",
    "kid": "FEDCBA9876543210FEDCBA9876543210"
  }
}
```

Champs requis :

- `iccid` - 19-20 chiffres, unique

Champs optionnels mais importants :

- `sim_vendor` - Nom du fabricant
- `batch_name` - Pour le suivi
- `is_esim` - Indicateur booléen pour eSIM
- `pin1`, `pin2` - Codes PIN de l'utilisateur final
- `puk1`, `puk2` - Codes de déverrouillage PIN
- `adm1-adm10` - Codes administratifs
- `kic`, `kid` - Clés de sécurité OTA (chaîne hexadécimale)

Exemple de requête :

```
curl -k -X POST https://hss.example.com:8443/api/sim \
  -H "Content-Type: application/json" \
  -d '{
    "sim": {
      "iccid": "89911012000003204510",
      "sim_vendor": "Gemalto"
    }
  }'
```

Mettre à jour une SIM

Modifier les données de la carte SIM.

Point de terminaison : `PUT /api/sim/:id`

Exemple de requête :

```
curl -k -X PUT https://hss.example.com:8443/api/sim/3f2504e0-4f89-
41d3-9a0c-0305e82c3301 \
  -H "Content-Type: application/json" \
  -d '{
    "sim": {
      "batch_name": "Nom de lot mis à jour"
    }
  }'
```

Supprimer une SIM

Supprimer un enregistrement de carte SIM.

Point de terminaison : `DELETE /api/sim/:id`

Avertissement : Assurez-vous qu'aucun abonné ne référence cette SIM avant de la supprimer.

Gestion des ensembles de clés

Les ensembles de clés contiennent le matériel cryptographique (Ki, OPC/OP, AMF, SQN) utilisé pour l'authentification des abonnés via l'algorithme Milenage. Chaque abonné doit référencer un ensemble de clés.

Voir aussi :

- Flux de protocole - Procédures d'authentification utilisant des ensembles de clés

Lister les ensembles de clés

Récupérer tous les ensembles de clés cryptographiques.

Point de terminaison : `GET /api/key_set`

Exemple de requête :

```
curl -k https://hss.example.com:8443/api/key_set
```

Obtenir un ensemble de clés

Récupérer un ensemble de clés spécifique. `ki`, `opc`, `op`, `psk`, `des` et `sip_password` sont en écriture seule : ils ne sont retournés que dans la réponse à la requête de création ou de mise à jour qui les a définis, et jamais dans une autre réponse API.

Point de terminaison : `GET /api/key_set/:id`

Exemple de requête :

```
curl -k https://hss.example.com:8443/api/key_set/6ba7b810-9dad-
41d1-80b4-00c04fd430c8
```

Exemple de réponse :

```
{
  "data": {
    "id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
    "amf": "8000",
    "sqn": 0,
    "authentication_algorithm": "milenage",
    "ota_counter": 0
  }
}
```


Créer un ensemble de clés

Créer un nouvel ensemble de clés cryptographiques.

Point de terminaison : `POST /api/key_set`

Corps de la requête :


```
{
  "key_set": {
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "amf": "8000",
    "sqn": 0,
    "authentication_algorithm": "milenage"
  }
}
```


Champs requis :

- `ki` - Clé de 128 bits (32 caractères hexadécimaux)
- Soit `opc` OU `op` (L'OPC peut être dérivé de l'OP)
- `authentication_algorithm` - Actuellement seulement "milenage"

Champs optionnels :

- `amf` - Par défaut : "8000"
- `sqn` - Par défaut : 0

- `ota_counter` - Par défaut : 0

Format de clé :

- Toutes les clés sont des chaînes hexadécimales
- Ki, OPC, OP : 32 caractères hexadécimaux (128 bits)
- AMF : 4 caractères hexadécimaux (16 bits)

Exemple de requête :

```
curl -k -X POST https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "key_set": {
      "ki": "0123456789ABCDEF0123456789ABCDEF",
      "opc": "FEDCBA9876543210FEDCBA9876543210",
      "authentication_algorithm": "milenage"
    }
  }'
```

Avertissement de sécurité : Les ensembles de clés contiennent du matériel cryptographique hautement sensible. Protégez l'accès API en conséquence.

Mettre à jour un ensemble de clés

Modifier un ensemble de clés existant.

Point de terminaison : `PUT /api/key_set/:id`

Avertissement : Changer les clés pour un abonné actif entraînera des échecs d'authentification. Ne mettez à jour les clés que pendant les fenêtres de maintenance ou pour de nouveaux abonnés.

Impact : Les mises à jour affectent immédiatement tous les abonnés utilisant cet ensemble de clés. Les abonnés actifs échoueront à l'authentification lors de la prochaine tentative de connexion.

Supprimer un ensemble de clés

Supprimer un ensemble de clés.

Point de terminaison : `DELETE /api/key_set/:id`

Avertissement : Assurez-vous qu'aucun abonné ne référence cet ensemble de clés avant de le supprimer. Interrogez d'abord les abonnés pour vérifier les références.

Gestion des profils

Profils EPC

Les profils EPC (Evolved Packet Core) définissent les paramètres de service de données pour les abonnés. Ces profils sont référencés lors de la création des abonnés.

Lister les profils EPC

Point de terminaison : `GET /api/epc/profile`

Obtenir un profil EPC

Point de terminaison : `GET /api/epc/profile/:id`

Créer un profil EPC

Point de terminaison : `POST /api/epc/profile`

Corps de la requête :

```
{
  "apn_profiles": [],
  "name": "Plan de données standard",
  "network_access_mode": "packet_only",
  "tracking_area_update_interval_seconds": 600,
  "ue_ambr_dl_kbps": 100000,
  "ue_ambr_ul_kbps": 50000
}
```

Champs :

Champ	Description	Unités	
`name`	Nom du profil	Texte	l
`ue_ambr_dl_kbps`	Limite de bande passante de téléchargement	Kbps	1
`ue_ambr_ul_kbps`	Limite de bande passante de téléchargement	Kbps	5
`network_access_mode`	Type d'accès	Chaîne	" "
`tracking_area_update_interval_seconds`	Minuteur TAU	Secondes	6
`apn_profiles`	Liste des identifiants de profil APN	Tableau	[

Exemple de requête :

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_profiles": [],
    "name": "Premium 100Mbps",
    "network_access_mode": "packet_only",
    "tracking_area_update_interval_seconds": 600,
    "ue_ambr_dl_kbps": 100000,
    "ue_ambr_ul_kbps": 50000
  }'
```

Voir aussi :

- Documentation des profils - Guide de configuration détaillé des profils
- Provisionnement complet des abonnés - Utilisation des profils EPC dans le provisionnement

Mettre à jour un profil EPC

Point de terminaison : `PUT /api/epc/profile/:id`

Remarque : Les modifications des profils EPC affectent tous les abonnés utilisant ce profil. Les sessions actives peuvent devoir être rétablies.

Supprimer un profil EPC

Point de terminaison : `DELETE /api/epc/profile/:id`

Avertissement : Assurez-vous qu'aucun abonné ne référence ce profil avant de le supprimer.

Profils IMS

Les profils IMS (IP Multimedia Subsystem) définissent les paramètres de service vocal et les critères de filtrage initiaux (IFC) pour les abonnés. Ces profils sont référencés lors de la création des abonnés avec les services IMS activés.

Lister les profils IMS

Point de terminaison : `GET /api/ims/profile`

Créer un profil IMS

Point de terminaison : `POST /api/ims/profile`

Corps de la requête :

```
{
  "name": "VoLTE standard",
  "ifc_template": "<IMS-XML-Template-Here>"
}
```

Champs requis :

- `name` - Nom du profil (doit être unique)
- `ifc_template` - Modèle IFC (Critères de filtrage initiaux) XML avec des variables de modèle Liquid

Variables de modèle IFC :

Le modèle IFC prend en charge les variables de modèle Liquid suivantes qui sont substituées dynamiquement :

Variable	Description	Valeur d'exemple
`{{ imsi }}`	IMSI de l'abonné	`001001123456789`
`{{ msisdns }}`	Tableau de MSISDN (pour les boucles)	`["14155551234", "14155555678"]`
`{{ mcc }}`	Code de pays mobile	`001`
`{{ mnc }}`	Code de réseau mobile	`001`

Comment fonctionne le rendu du modèle :

Le modèle IFC est stocké comme un **modèle Liquid** (similaire à Jinja2) et est rendu **dynamiquement** lors des opérations IMS :

1. **Stockage** : Lorsque vous créez un profil IMS, le modèle est stocké tel quel avec des variables comme `{{ imsi }}` et `{% for msisdn in msisdns %}`
2. **Validation** : L'API valide le modèle en le rendant avec des données de test pour s'assurer que la syntaxe XML est valide
3. **Rendu à l'exécution** : Lorsqu'un abonné effectue une inscription IMS (MAA/SAA), le HSS :
   - Récupère le profil IMS de l'abonné
   - Rendu le modèle avec les données réelles de l'abonné :
     - `{{ imsi }}` → IMSI de l'abonné
     - `{{ msisdns }}` → numéros de téléphone de l'abonné
     - `{{ mcc }}` → Code de pays mobile configuré
     - `{{ mnc }}` → Code de réseau mobile configuré
   - Retourne le XML rendu au S-CSCF via Cx/Diameter

Syntaxe du modèle :

```
<!-- Substitution de variable simple -->
{{ imsi }}

<!-- Boucles pour tableaux -->
{% for msisdn in msisdns %}
  <MSISDN>{{ msisdn }}</MSISDN>
{% endfor %}

<!-- Combinaison de variables -->
{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc }}.3gppnetwork.org
```

Exemple de modèle IFC :

```
<?xml version="1.0" encoding="UTF-8"?>
<IMSSubscription>
<PrivateID>{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</PrivateID>
<ServiceProfile>
{% for msisdn in msisdns %}
<PublicIdentity>
<Identity>sip:{{ msisdn }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</Identity>
<Extension>
<IdentityType>0</IdentityType>
</Extension>
</PublicIdentity>
<PublicIdentity>
<Identity>tel:{{ msisdn }}</Identity>
<Extension>
<IdentityType>0</IdentityType>
</Extension>
</PublicIdentity>
{% endfor %}
<InitialFilterCriteria>
<Priority>10</Priority>
<TriggerPoint>
<ConditionTypeCNF>0</ConditionTypeCNF>
<SPT>
<ConditionNegated>0</ConditionNegated>
<Group>0</Group>
<Method>REGISTER</Method>
</SPT>
</TriggerPoint>
<ApplicationServer>
<ServerName>sip:as.ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</ServerName>
<DefaultHandling>0</DefaultHandling>
</ApplicationServer>
</InitialFilterCriteria>
</ServiceProfile>
</IMSSubscription>
```

Exemple de requête (curl) :

```
curl -k -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "default",
    "ifc_template": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<IMSSubscription><ServiceProfile>...</ServiceProfile>
</IMSSubscription>"
  }'
```

Exemple de requête (Python) :

```
import requests

response = requests.post(
    "https://hss.example.com:8443/api/ims/profile",
    json={
        "name": "default",
        "ifc_template": ifc_template_string
    },
    verify=False  # Pour les certificats auto-signés
)
```

Réponse de succès (201 Créé) :

```
{
  "status": "success",
  "response": {
    "id": "0f8fad5b-d9cb-469f-a165-70867728950e",
    "name": "default",
    "ifc_template": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>..."
  }
}
```

Validation :

- L'API valide que le modèle IFC est un XML valide
- Les variables de modèle sont rendues avec des données de test pour vérifier la syntaxe

- Le champ `name` doit être unique et non vide

Voir aussi :

- Documentation des profils - Détails et exemples de modèle IFC
- Flux de protocole - Inscription IMS et flux d'appels
- Modèle IFC par défaut - Mise en œuvre de référence

Profils APN

Les profils APN (Access Point Name) se composent de trois composants qui fonctionnent ensemble :

1. **Identifiant APN** - Définit le nom de l'APN et la version IP
2. **Profil QoS APN** - Définit les paramètres de qualité de service
3. **Profil APN** - Combine l'identifiant et le QoS, lié aux Profils EPC

Voir Documentation PCRF pour la configuration détaillée des politiques, la gestion du QoS et la ré-auth. automatique. Voir aussi Documentation des profils pour des exemples de configuration APN.

Lister les identifiants APN

Point de terminaison : `GET /api/apn/identifier`

Créer un identifiant APN

Point de terminaison : `POST /api/apn/identifier`

Corps de la requête :

```
{
  "apn": "internet",
  "ip_version": "ipv4v6"
}
```

Valeurs de version IP :

- `"ipv4"` - IPv4 uniquement

- `"ipv6"` - IPv6 uniquement
- `"ipv4v6"` - IPv4v6 (double pile)
- `"ipv4_or_ipv6"` - IPv4 ou IPv6 (choix du réseau)

Lister les profils QoS APN

Point de terminaison : `GET /api/apn/qos_profile`

Créer un profil QoS APN

Point de terminaison : `POST /api/apn/qos_profile`

Corps de la requête :

```
{
  "name": "Internet meilleur effort",
  "allocation_retention_priority": 8,
  "apn_ambr_dl_kbps": 50000,
  "apn_ambr_ul_kbps": 25000,
  "pre_emption_capability": false,
  "pre_emption_vulnerability": true,
  "qci": 9
}
```

Lister les profils APN

Point de terminaison : `GET /api/apn/profile`

Créer un profil APN

Point de terminaison : `POST /api/apn/profile`

Corps de la requête :

```
{
  "apn_identifier_id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
  "apn_qos_profile_id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793",
  "name": "APN Internet"
}
```

Champs requis :

- `apn_identifier_id` - Doit référencer un Identifiant APN existant
- `apn_qos_profile_id` - Doit référencer un Profil QoS APN existant

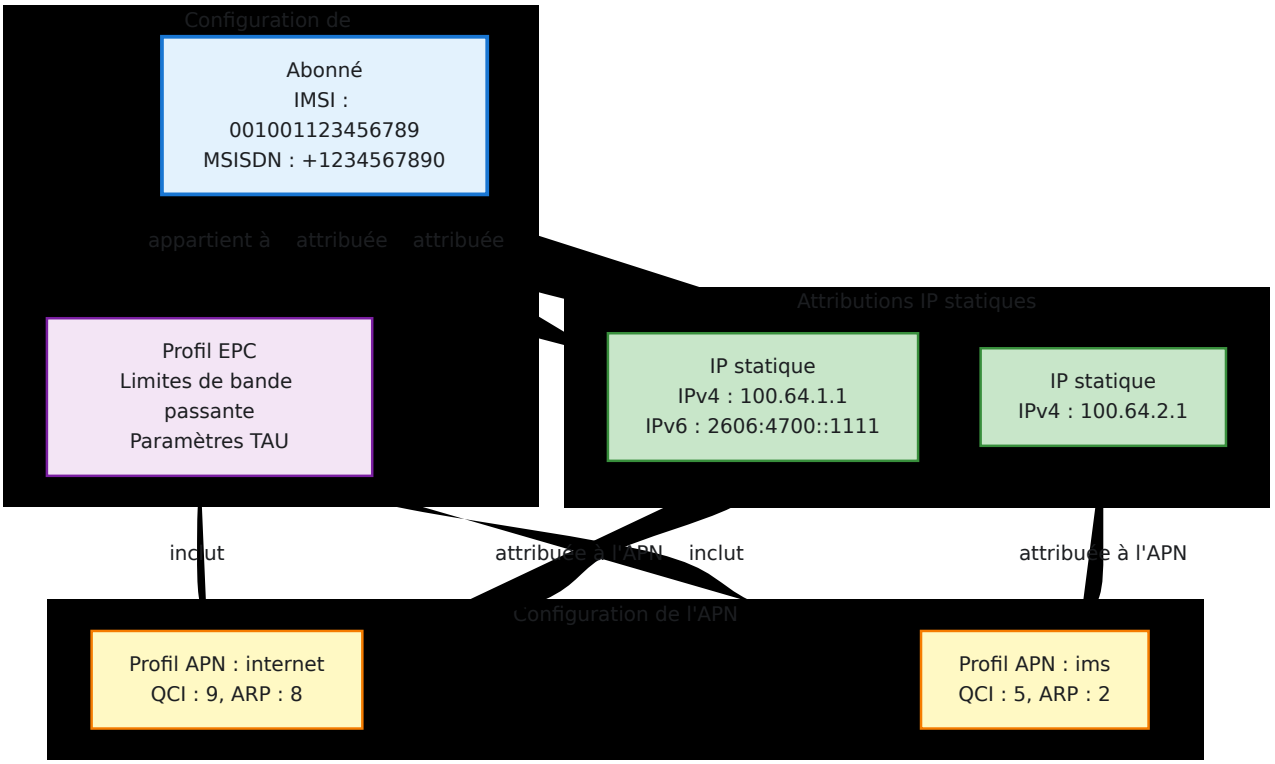
Voir aussi :

- Provisionnement complet des abonnés - Exemple complet incluant la configuration APN
- Profils EPC - Les profils APN sont liés aux profils EPC

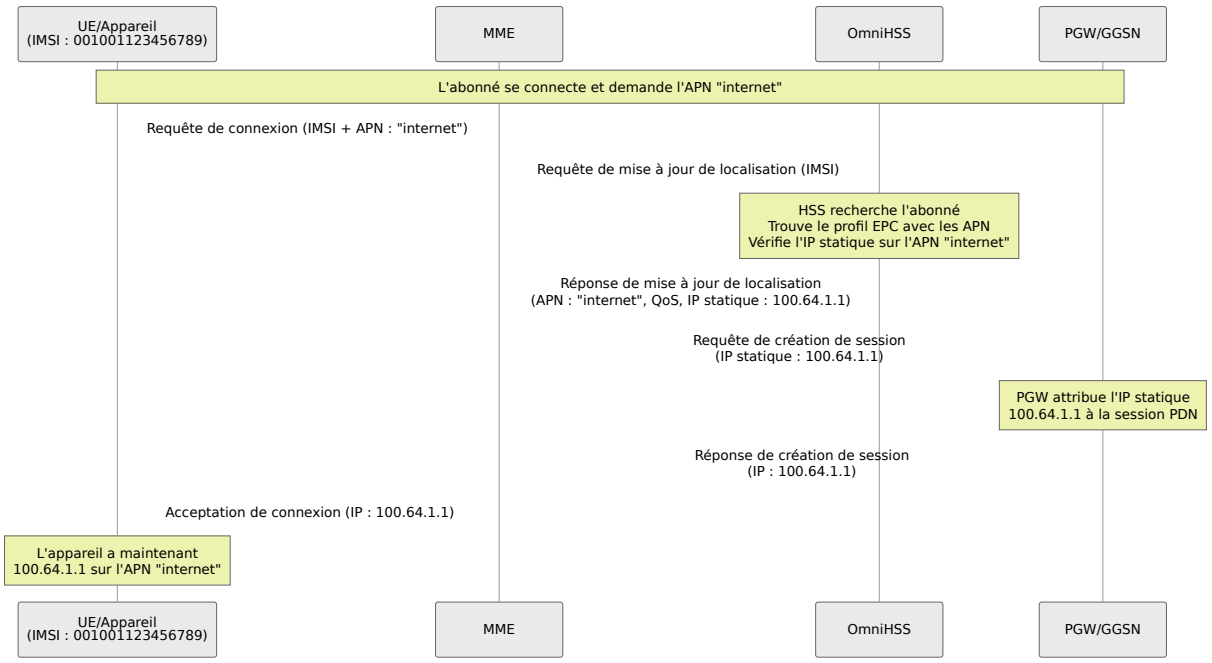
Gestion des IP statiques

Des adresses IP statiques peuvent être attribuées à des APN spécifiques pour des abonnés individuels. Cela permet aux abonnés de recevoir une adresse IPv4 et/ou IPv6 prédéterminée lors de la connexion à un APN particulier, plutôt que de recevoir une adresse dynamique d'un pool DHCP.

Architecture :

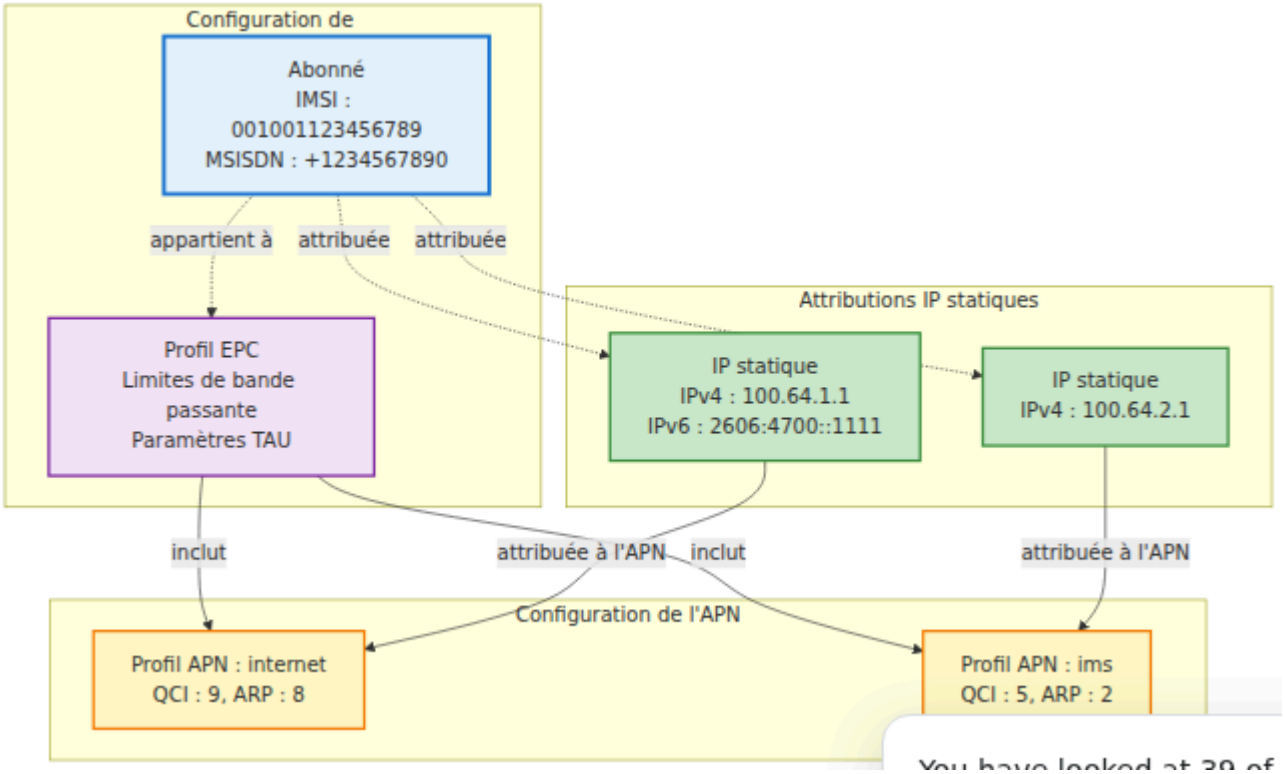


Flux de données lors de la connexion de l'abonné :



Réponse de mise à jour de localisation - Mappage des données de configuration APN :

Ce diagramme montre exactement d'où provient chaque champ dans l'AVP de configuration APN de la réponse de mise à jour de localisation S6a :



Concepts Clés :

- **Attribution par APN** : Chaque IP statique est liée à un Profil APN spécifique
- **Une IP par APN par Abonné** : Un abonné ne peut avoir qu'une seule attribution d'IP statique par APN
- **Support IPv4 et IPv6** : Les IP statiques peuvent être uniquement IPv4, uniquement IPv6, ou double pile
- **Unicité Globale des IP** : Chaque adresse IP doit être globalement unique à travers **tous** les enregistrements d'IP statiques dans le système
  - La même adresse IPv4 ou IPv6 ne peut pas être attribuée à plusieurs abonnés (même sur différents APN)
  - Cela empêche les conflits de routage et l'ambiguïté des adresses IP
  - Appliqué par des index uniques dans la base de données sur les champs `ipv4_static_ip` et `ipv6_static_ip`
- **Relation Plusieurs-à-Plusieurs** : Les abonnés et les IP statiques sont liés via une table de jointure

Cas d'Utilisation :

- Adresses IP fixes pour les dispositifs IoT
- Hébergement de serveurs sur des dispositifs mobiles (nécessite une IP statique pour les connexions entrantes)
- Applications héritées nécessitant des adresses IP spécifiques
- Routage de politique réseau basé sur l'IP source
- Conformité réglementaire nécessitant le suivi des adresses IP

Lister les IP Statiques

Récupérer toutes les attributions d'IP statiques.

Point de terminaison : `GET /api/epc/static_ip`

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/epc/static_ip
```

Exemple de Réponse :

```
{
  "data": [
    {
      "id": "c56a4180-65aa-42ec-a945-5fd21dec0538",
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111",
      "apn_profile": {
        "id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
        "name": "Profil APN Internet",
        "apn_identifier": {
          "apn": "internet",
          "ip_version": "ipv4v6"
        }
      },
      "inserted_at": "2025-11-15T10:30:00Z",
      "updated_at": "2025-11-15T10:30:00Z"
    }
  ]
}
```


Obtenir une IP Statique

Récupérer une attribution d'IP statique spécifique.

Point de terminaison : `GET /api/epc/static_ip/:id`

Paramètres de Chemin :

Paramètre	Type	Description
`id`	string (UUID)	ID de la base de données de l'IP statique

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/epc/static_ip/c56a4180-65aa-42ec-a945-5fd21dec0538
```

Créer une IP Statique

Créer une nouvelle attribution d'IP statique pour un APN.

Point de terminaison : `POST /api/epc/static_ip`

Corps de la Requête :

```
{
  "static_ip": {
    "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111"
  }
}
```

Champs Requis :

- `apn_profile_id` - Doit référencer un Profil APN existant
- Au moins l'un de `ipv4_static_ip` OU `ipv6_static_ip` doit être spécifié

Champs Optionnels :

- `ipv4_static_ip` - Adresse IPv4 (notation décimale pointée)
- `ipv6_static_ip` - Adresse IPv6 (notation standard)

Validation du Format IP :

- IPv4 : Format standard décimal pointé (ex. : `100.64.1.1`)
- IPv6 : Format standard hexadécimal séparé par des deux-points (ex. : `2606:4700:4700::1111`)
- Les adresses IPv4 et IPv6 doivent être **globalement uniques à travers tous les enregistrements d'IP statiques**
  - Cela empêche les conflits d'adresses IP dans le réseau
  - La même IP ne peut pas être attribuée à plusieurs abonnés, même sur différents APN
  - C'est une contrainte au niveau de la base de données appliquée par des index uniques

Options de Configuration :

Configuration	IPv4	IPv6	Exemple
IPv4 Seulement	✓	-	`{"ipv4_static_ip": "100.64.1.1"}`
IPv6 Seulement	-	✓	`{"ipv6_static_ip": "2606:4700:4700::1111"}`
Double Pile	✓	✓	Les deux champs spécifiés

Exemples de Requêtes :

IP Statique uniquement IPv4 :

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1"
    }
  }'
```

IP Statique uniquement IPv6 :

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "b2c3d4e5-f6a7-4b1c-9d2e-3f4a5b6c7d8e",
      "ipv6_static_ip": "2606:4700:4700::1111"
    }
  }'
```

IP Statique double pile :

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111"
    }
  }'
```

Réponse de Succès (201 Créé) :


```
{
  "data": {
    "id": "c56a4180-65aa-42ec-a945-5fd21dec0538",
    "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111",
    "inserted_at": "2025-11-15T10:30:00Z",
    "updated_at": "2025-11-15T10:30:00Z"
  }
}
```


Voir Aussi :

- Attribuer une IP Statique à un Abonné - Comment lier cela à un abonné
- Profils APN - Gestion des configurations APN

Mettre à Jour une IP Statique

Modifier une attribution d'IP statique existante.

Point de terminaison : `PUT /api/epc/static_ip/:id`

Paramètres de Chemin :

Paramètre	Type	Description
`id`	string (UUID)	ID de la base de données de l'IP statique

Corps de la Requête :

```
{
  "static_ip": {
    "ipv4_static_ip": "100.64.1.2",
    "ipv6_static_ip": "2606:4700:4700::1112"
  }
}
```

Champs Modifiables :

- `ipv4_static_ip` - Changer l'adresse IPv4
- `ipv6_static_ip` - Changer l'adresse IPv6
- `apn_profile_id` - Changer l'attribution APN

Non Modifiables :

- `id` - Clé primaire (lecture seule)

Avertissement : Changer l'adresse IP pour un abonné actif affectera sa prochaine connexion PDN. Les sessions PDN actives continueront à utiliser l'ancienne IP jusqu'à ce qu'elles se déconnectent et se reconnectent.

Exemple de Requête :

```
curl -k -X PUT
https://hss.example.com:8443/api/epc/static_ip/c56a4180-65aa-42ec-
a945-5fd21dec0538 \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "ipv4_static_ip": "100.64.1.2"
    }
  }'
```

Supprimer une IP Statique

Supprimer une attribution d'IP statique.

Point de terminaison : `DELETE /api/epc/static_ip/:id`

Paramètres de Chemin :

Paramètre	Type	Description
`id`	string (UUID)	ID de la base de données de l'IP statique

Exemple de Requête :

```
curl -k -X DELETE
https://hss.example.com:8443/api/epc/static_ip/c56a4180-65aa-42ec-
a945-5fd21dec0538
```

Comportement :

- Supprime l'attribution d'IP statique
- N'affecte PAS le Profil APN (l'APN reste disponible pour d'autres abonnés)
- Les abonnés utilisant cette IP statique recevront des IP dynamiques lors de leur prochaine connexion
- L'adresse IP devient disponible pour réutilisation après suppression

Avertissement : Si un abonné utilise activement cette IP statique, la supprimer entraînera qu'il reçoive une IP dynamique lors de sa prochaine connexion PDN. Assurez-vous que les abonnés sont hors ligne ou envoyez une Demande d'Annulation de Localisation avant de supprimer.

Attribuer une IP Statique à un Abonné

Pour attribuer une IP statique à un abonné, vous devez associer l'enregistrement d'IP statique avec le Abonné lors de la création ou de la mise à jour.

Modèle d'Attribution :

1. **Créer l'IP Statique** (voir Créer une IP Statique)
2. **Attribuer à l'Abonné** en utilisant le champ `static_ips`

Créer un Abonné avec une IP Statique :

```
# Étape 1 : Créer une IP statique pour l'APN "internet"
STATIC_IP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111"
    }
  }' | jq -r '.data.id')

# Étape 2 : Créer un abonné avec l'IP statique attribuée
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"subscriber\": {
      \"imsi\": \"001001123456789\",
      \"key_set_id\": \"6ba7b810-9dad-41d1-80b4-00c04fd430c8\",
      \"epc_profile_id\": \"e58ed763-928c-4155-bee9-
fdbaaadc15f3\",
      \"static_ips\": [$STATIC_IP_ID]
    }
  }"
```

Mettre à Jour un Abonné Existant avec une IP Statique :

```
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "static_ips": ["c56a4180-65aa-42ec-a945-5fd21dec0538",
"5a8f9c3e-2d4b-4e6f-8a1b-3c5d7e9f0a2b"]
    }
  }'
```

Multiples IP Statiques (Différents APNs) :

Un abonné peut avoir plusieurs IP statiques tant que chacune est pour un APN différent :

```
# Créer une IP statique pour l'APN "internet"
INTERNET_IP=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1"
    }
  }' | jq -r '.data.id')

# Créer une IP statique pour l'APN "ims"
IMS_IP=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "b2c3d4e5-f6a7-4b1c-9d2e-3f4a5b6c7d8e",
      "ipv4_static_ip": "100.64.2.1"
    }
  }' | jq -r '.data.id')

# Attribuer les deux à l'abonné
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"subscriber\": {
      \"imsi\": \"001001123456789\",
      \"key_set_id\": \"6ba7b810-9dad-41d1-80b4-00c04fd430c8\",
      \"epc_profile_id\": \"e58ed763-928c-4155-bee9-
fdbaaadc15f3\",
      \"static_ips\": [$INTERNET_IP, $IMS_IP]
    }
  }"
```

Règles de Validation :

- ✓ **Autorisé** : Plusieurs IP statiques pour différents APNs
- ✗ **Rejeté** : Plusieurs IP statiques pour le même APN

Exemple d'Erreur - APN Dupliqué :

```
# Cela échouera si les deux IP statiques font référence au même
APN
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "static_ips": ["c56a4180-65aa-42ec-a945-5fd21dec0538",
"5a8f9c3e-2d4b-4e6f-8a1b-3c5d7e9f0a2b"]
    }
  }'

# Réponse d'Erreur :
{
  "errors": {
    "static_ips": [
      "les ips statiques par apn par abonné doivent être uniques.
par exemple, un abonné ne peut pas se voir attribuer l'ip statique
100.64.1.1 pour internet et aussi 100.64.1.2 pour internet"
    ]
  }
}
```

Voir Aussi :

- Créer un Abonné - Provisionnement d'abonné
- Mettre à Jour un Abonné - Modification de la configuration de l'abonné
- Exemple Complet de Provisionnement d'IP Statique - Flux de travail de bout en bout

---

Gestion du Roaming

Les profils de roaming contrôlent si les abonnés peuvent accéder aux données et aux services IMS sur les réseaux visités. Les profils sont attribués aux abonnés et consistent en des règles correspondant au MCC/MNC.

Lister les Profils de Roaming

Point de terminaison : `GET /api/roaming/profile`

Créer un Profil de Roaming

Point de terminaison : `POST /api/roaming/profile`

Corps de la Requête :

```
{
  "roaming_profile": {
    "name": "Opérateurs Américains Uniquement",
    "data_action_if_no_rules_match": "deny",
    "ims_action_if_no_rules_match": "deny",
    "roaming_rules": []
  }
}
```

Valeurs d'Action :

- `"allow"` - Autoriser
- `"deny"` - Refuser

Actions par Défaut :

- `data_action_if_no_rules_match` - Action lorsque aucune règle de roaming ne correspond
- `ims_action_if_no_rules_match` - Action par défaut spécifique à l'IMS

Lister les Règles de Roaming

Point de terminaison : `GET /api/roaming/rule`

Créer une Règle de Roaming

Point de terminaison : `POST /api/roaming/rule`

Corps de la Requête :

```
{
  "roaming_rule": {
    "name": "Autoriser AT&T",
    "mcc": "310",
    "mnc": "410",
    "data_action": "allow",
    "ims_action": "allow"
  }
}
```

Champs :

- `mcc` - Code de Pays Mobile (3 chiffres)
- `mnc` - Code de Réseau Mobile (2-3 chiffres)
- `data_action` - `"allow"` ou `"deny"` pour les services de données
- `ims_action` - `"allow"` ou `"deny"` pour les services IMS/voix

Voir Aussi :

- Documentation sur le Roaming - Configuration détaillée et exemples
- Flux de Protocole - Comment fonctionne le contrôle de roaming dans les flux Diameter

---

Gestion de l'EIR

OmniHSS fonctionne comme un Registre d'Identité d'Équipement (EIR) via l'interface Diameter S13. Les règles EIR sont des expressions régulières correspondant à l'IMEISV d'un appareil, puis à son IMEI, puis à l'IMSI de l'abonné. À chaque étape, les règles sont évaluées dans l'ordre croissant de `precedence` et la première correspondance l'emporte.

Voir Documentation EIR pour un contrôle détaillé de l'identité de l'équipement, les flux d'interface S13, la validation de l'IMEI, et la priorité des règles.

Lister les Règles EIR

Récupérer toutes les règles EIR.

Point de terminaison : `GET /api/eir/rule`

La liste est retournée dans l'ordre `id`. Triez par `precedence` pour voir l'ordre d'évaluation.

Obtenir une Règle EIR

Récupérer une seule règle EIR.

Point de terminaison : `GET /api/eir/rule/:id`

Créer une Règle EIR

Ajouter une règle et la placer dans l'ordre d'évaluation.

Point de terminaison : `POST /api/eir/rule`

Corps de la Requête :

```
{
  "action": "blacklist",
  "regex": "35979139461640",
  "precedence": 1
}
```

Champs Requis :

- `action` - `"whitelist"` (autoriser), `"blacklist"` (refuser) ou `"greylist"` (autoriser mais suivre)
- `regex` - Expression régulière testée contre l'IMEISV, puis l'IMEI, puis l'IMSI. Doit se compiler

Champs Optionnels :

- `precedence` - Entier de 1 à 65535. Les valeurs plus basses sont évaluées en premier. Omettez-le pour ajouter la règle après la priorité actuelle la plus élevée. Fournir une valeur insère la règle à cette position : chaque règle existante avec une priorité égale ou supérieure descend d'une position, que la valeur ait déjà été utilisée ou non

Réponse de Succès (201 Créé) :

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "blacklist",
    "regex": "35979139461640",
    "precedence": 1
  }
}
```

La `precedence` dans la réponse est la position donnée à la règle. Les échecs de validation (action `unknown`, `regex` vide ou non compilable, `precedence` en dehors de 1 à 65535) retournent HTTP 422 dans le format d'erreur standard.

Mettre à Jour une Règle EIR

Changer l'action, le motif ou la position d'une règle.

Point de terminaison : `PATCH /api/eir/rule/:id` ou `PUT /api/eir/rule/:id`

Les deux méthodes n'appliquent que les champs présents dans le corps ; les champs omis conservent leurs valeurs actuelles. Un `PUT` à un id qui n'existe pas crée la règle avec cet id (HTTP 200), placée selon les mêmes règles de priorité qu'un `POST`.

Changer `precedence` déplace la règle à cette position et déplace uniquement les règles entre ses anciennes et nouvelles positions. Omettre `precedence`, ou envoyer sa valeur actuelle, laisse l'ordre inchangé. Ne pas envoyer `null` : cela efface la priorité de la règle et la règle est alors évaluée après chaque règle numérotée.

Supprimer une Règle EIR

Supprimer une règle.

Point de terminaison : `DELETE /api/eir/rule/:id`

Réponse : HTTP 200 avec `"response": ""`.

Supprimer une règle laisse un vide dans la numérotation de priorité. Les règles restantes conservent leur ordre relatif.

Cas d'Utilisation :

- Bloquer les dispositifs volés (mettre sur liste noire des IMEIs spécifiques)
- Restreindre les types de dispositifs (mettre sur liste noire par motif TAC)
- Autoriser uniquement les dispositifs approuvés (modèles sur liste blanche, avec `unknown_equipment_behaviour` changé de son défaut de `whitelist` à `blacklist` ou `reject_unknown_equipment` ; voir Comportement des Équipements Inconnus)
- Remplacer une règle large pour un dispositif en donnant à la règle spécifique une priorité plus basse

Voir Aussi :

- Documentation EIR - Logique de correspondance, priorité des règles et flux S13
- Flux de Protocole - Flux d'interface S13 et vérification EIR
- Aperçu de l'Architecture - Fonction EIR d'OmniHSS

Documentation Supplémentaire

Pour plus d'informations, consultez la documentation suivante :

- **Statut et Santé** - Points de terminaison de vérification de la santé de l'API
- **Gestion des Erreurs** - Erreurs courantes et dépannage

- **Exemples d'Utilisation de l'API** - Flux de travail de provisionnement complet

---

← Retour au Guide des Opérations | Suivant : Panneau de Contrôle →

État et Santé de l'API

← Retour à la Référence de l'API

État du Système

Vérifiez si l'API répond.

Point de terminaison : `GET /api/status`

Exemple de Requête :

```
curl -k https://hss.example.com:8443/api/status
```

Exemple de Réponse :

```
{
  "status": "ok"
}
```

Cas d'Utilisation : Vérification de la santé pour les équilibreurs de charge et les systèmes de surveillance.

← Retour à la Référence de l'API

Vue d'ensemble de l'architecture d'OmniHSS

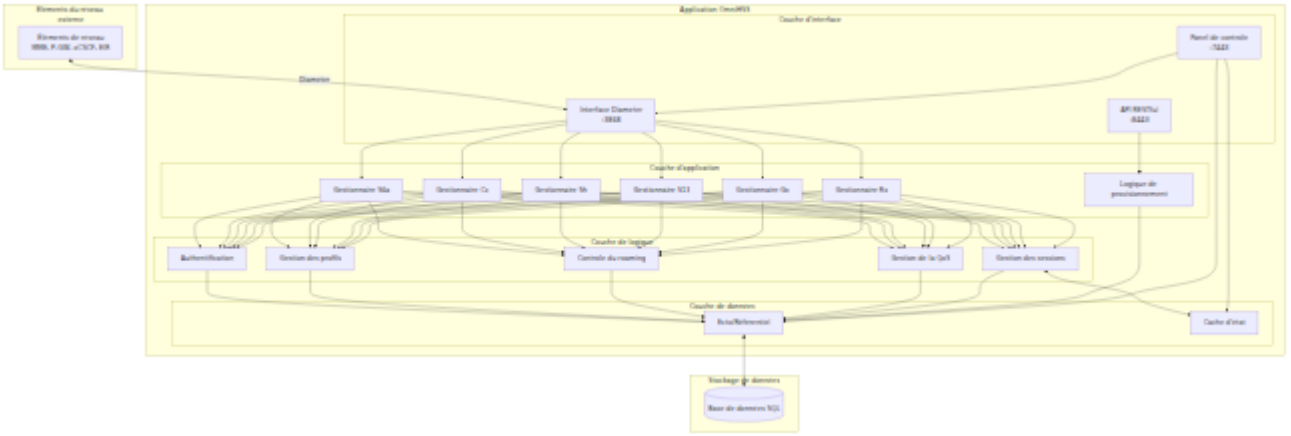
← Retour au Guide des Opérations

Table des Matières

- Vue d'ensemble du système
- Architecture des composants
- Pile Diameter
- Couche d'application
- Couche de données
- Interfaces externes
- Architecture de déploiement

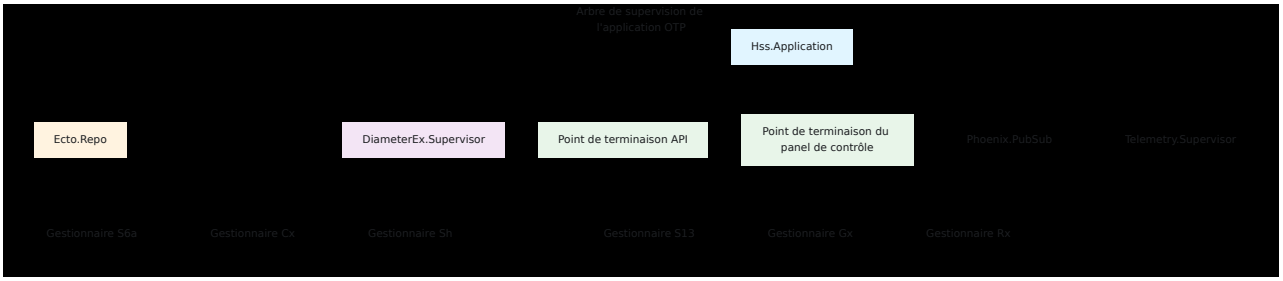
Vue d'ensemble du système

OmniHSS est construit sur Elixir et la plateforme Erlang/OTP, fournissant un système hautement concurrent et tolérant aux pannes, conçu pour des charges de travail de télécommunications. L'architecture suit une approche par couches avec une séparation claire des préoccupations.



Architecture des composants

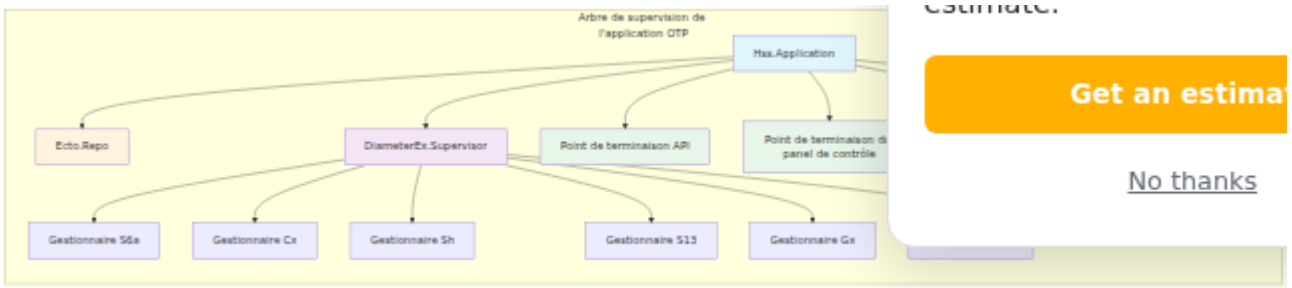
Composants principaux



Gestionnaires d'application Diameter

Chaque application Diameter (S6a, Cx, Sh, S13, Gx, Rx) est implémentée en tant que module gestionnaire DiameterEx qui :

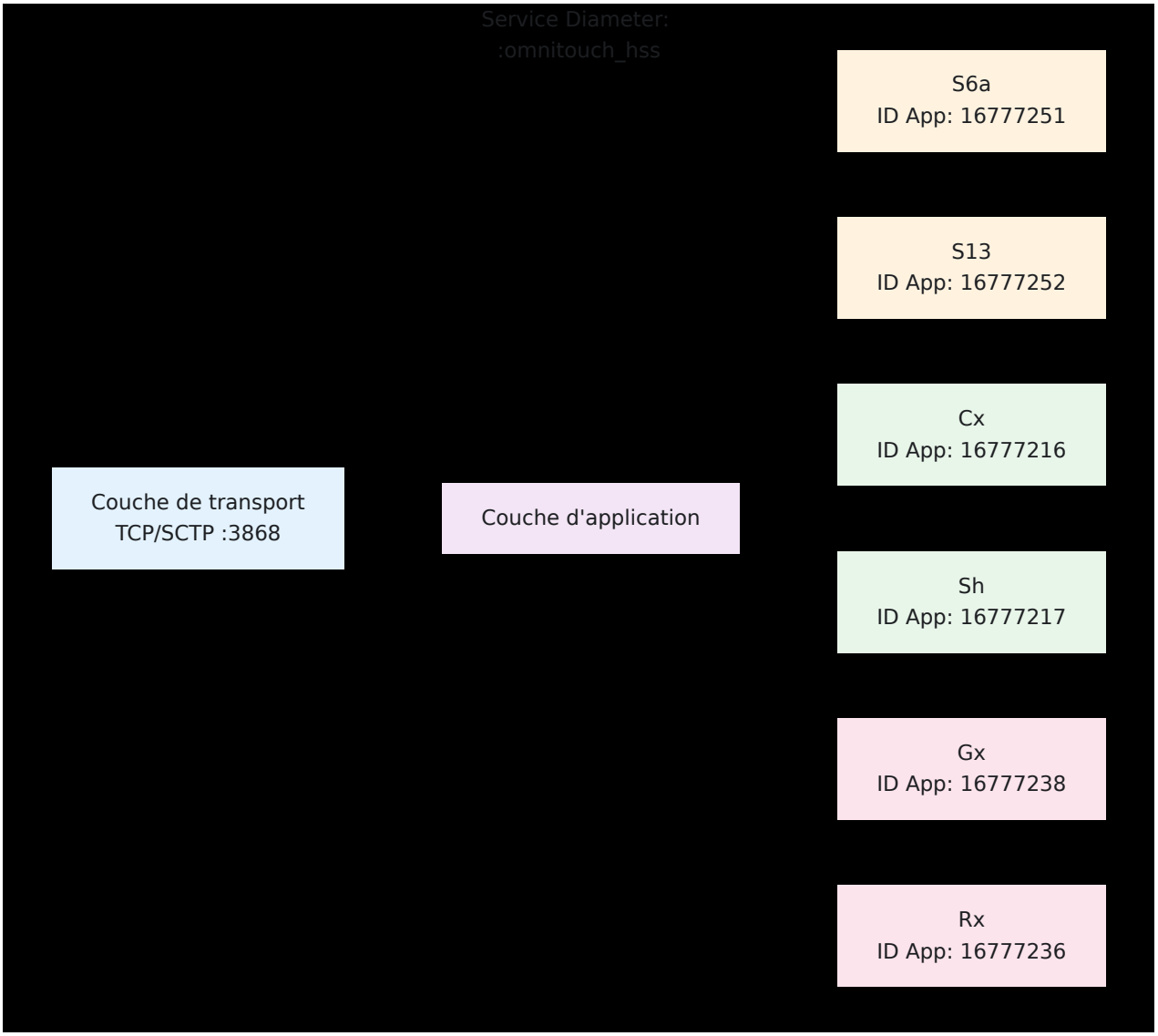
1. **S'inscrit auprès de DiameterEx** - S'abonne à des ID d'application Diameter spécifiques
2. **Valide les demandes** - Extrait les AVP, valide l'état de l'abonné
3. **Traite la logique métier** - Appelle les modules de logique métier appropriés
4. **Construit les réponses** - Génère des messages de réponse Diameter avec des AVP
5. **Gère les erreurs** - Retourne les codes de résultat Diameter appropriés



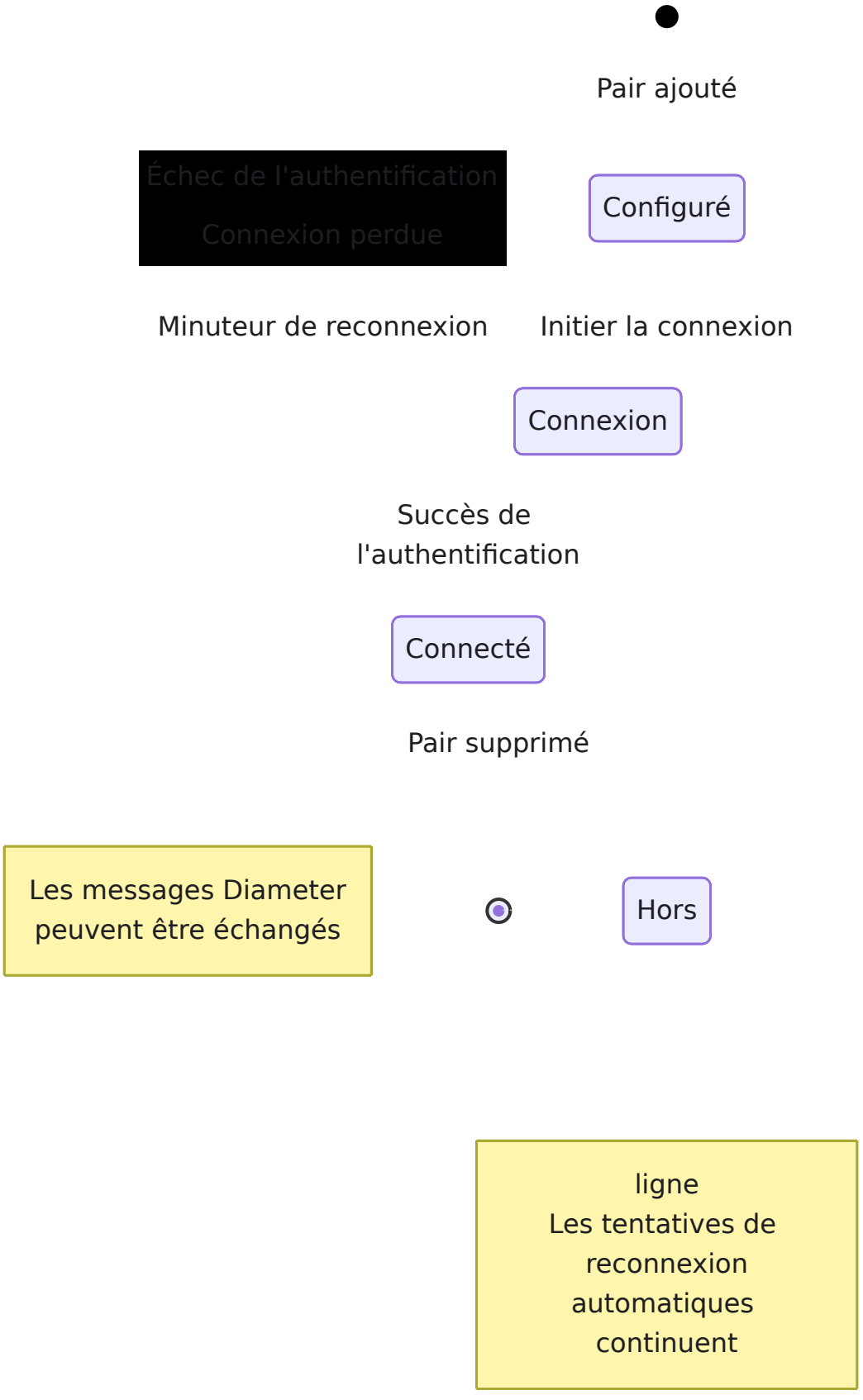
Pile Diameter

Configuration du service Diameter

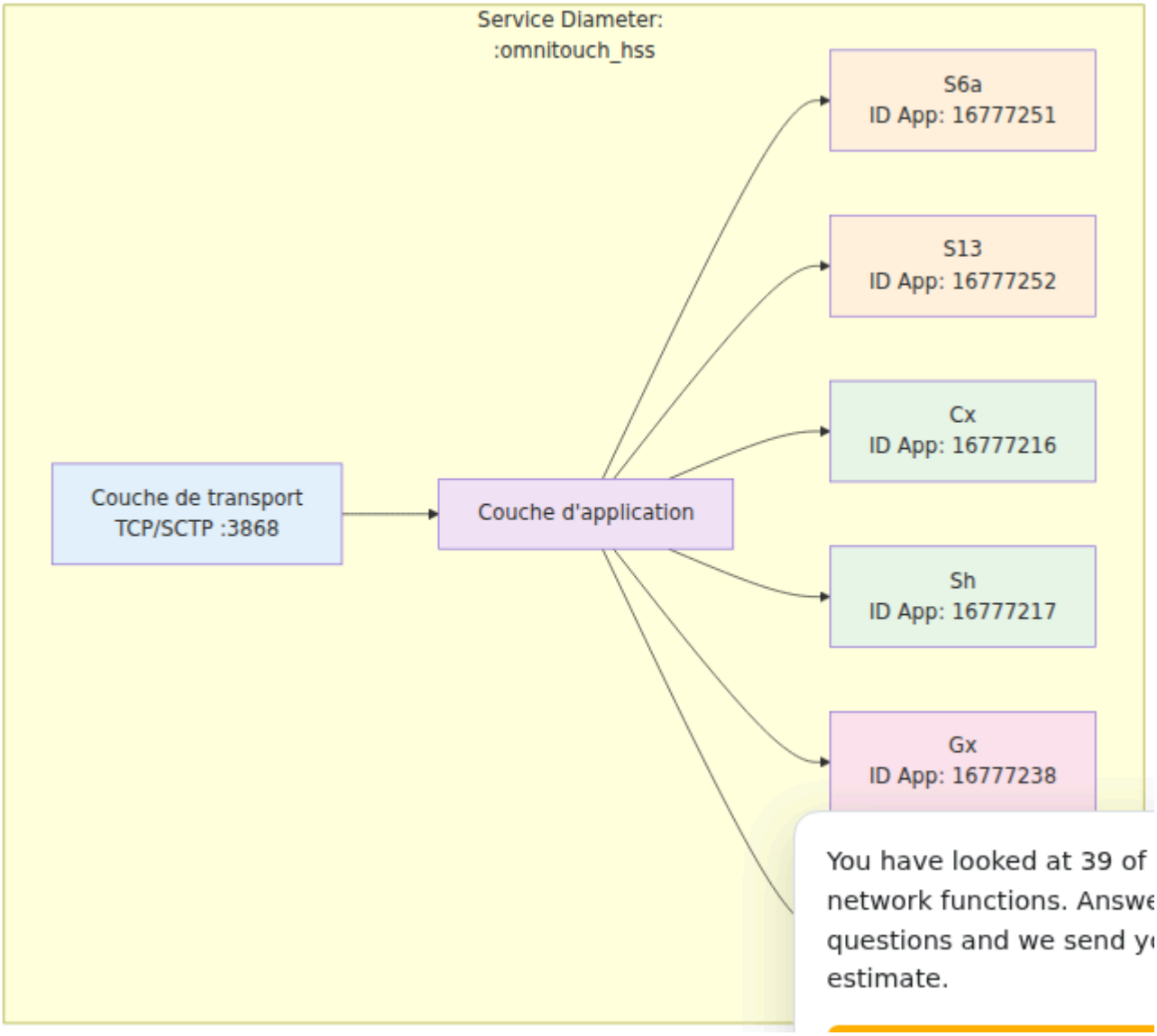
OmniHSS configure un seul service Diameter avec plusieurs applications prises en charge :


Service Diameter:
:omnitouch_hss
S6a
ID App: 16777251
S13
ID App: 16777252
Cx
ID App: 16777216
Couche de transport
TCP/SCTP :3868
Couche d'application
Sh
ID App: 16777217
Gx
ID App: 16777238
Rx
ID App: 16777236

Gestion de la connexion entre pairs



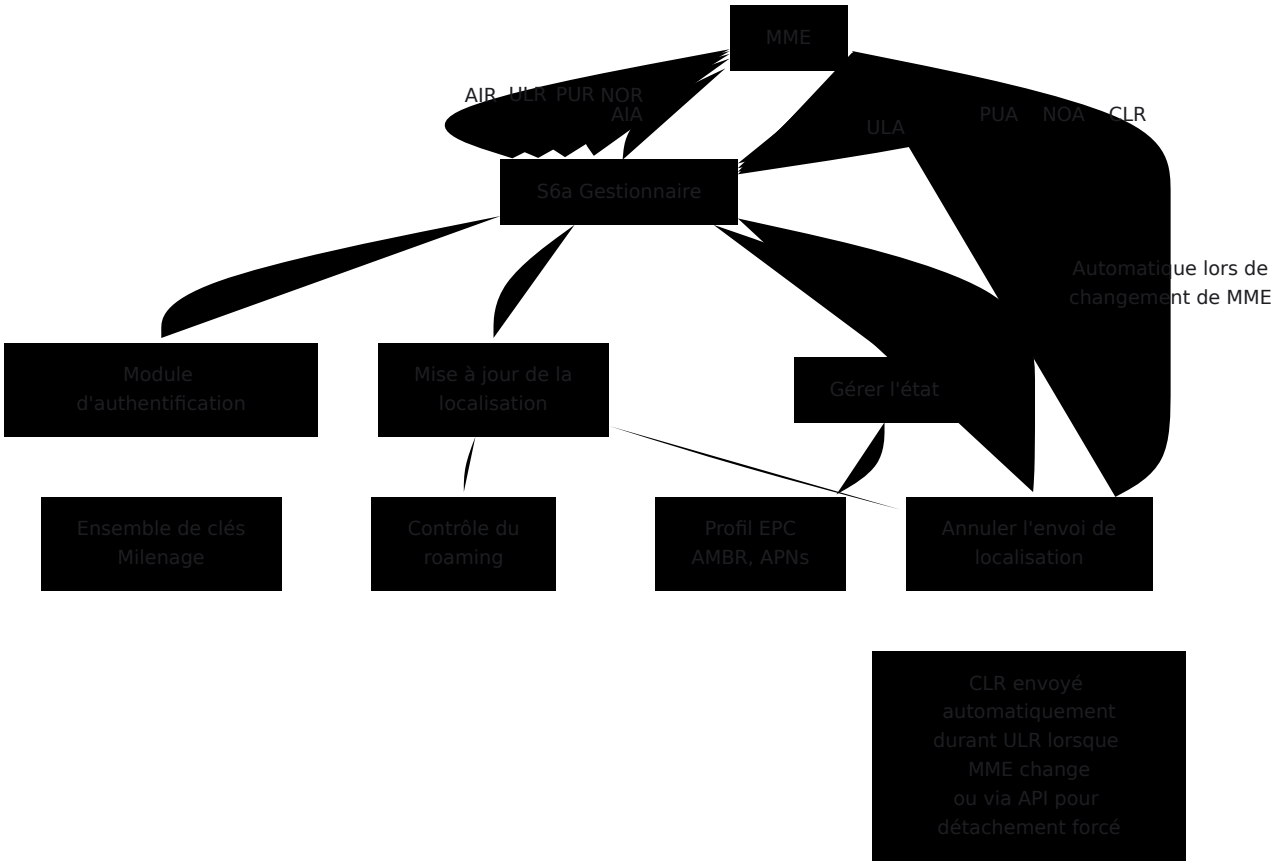
Flux de messages Diameter



Couche d'application

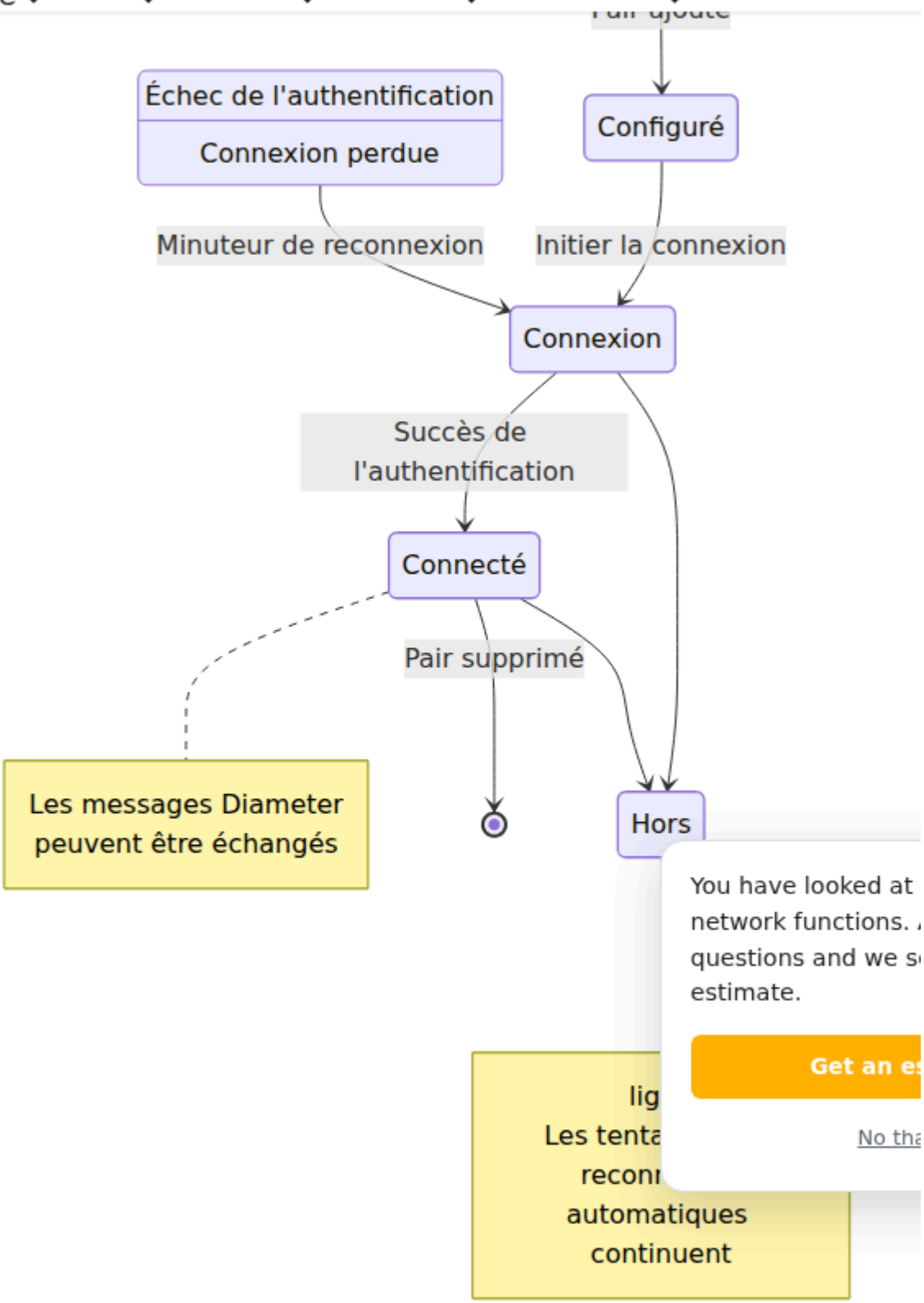
Interface S6a (LTE/EPC)

Gère l'authentification et la gestion de la mobilité pour les réseaux LTE.



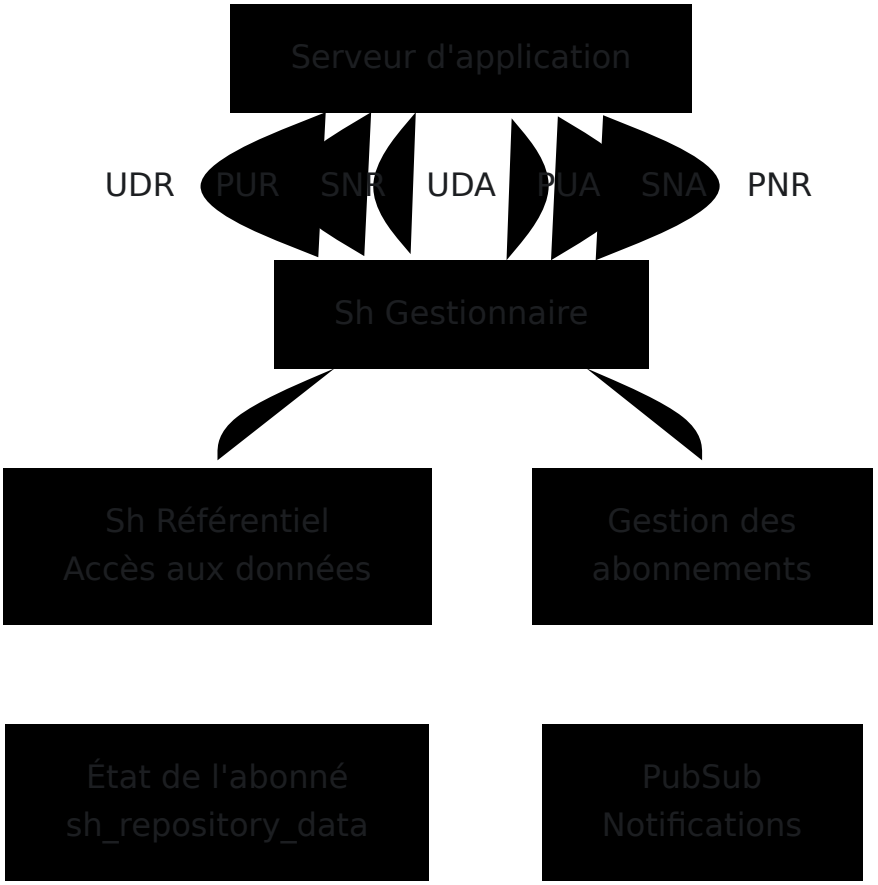
Interface Cx (IMS)

Gère l'enregistrement et l'authentification IMS.


Échec de l'authentification
Connexion perdue
Configuré
Minuteur de reconnexion
Initier la connexion
Connexion
Succès de l'authentification
Connecté
Pair supprimé
Les messages Diameter peuvent être échangés
Hors
You have looked at
network functions.
questions and we
estimate.
Get an
No tha
automatiques
continuent

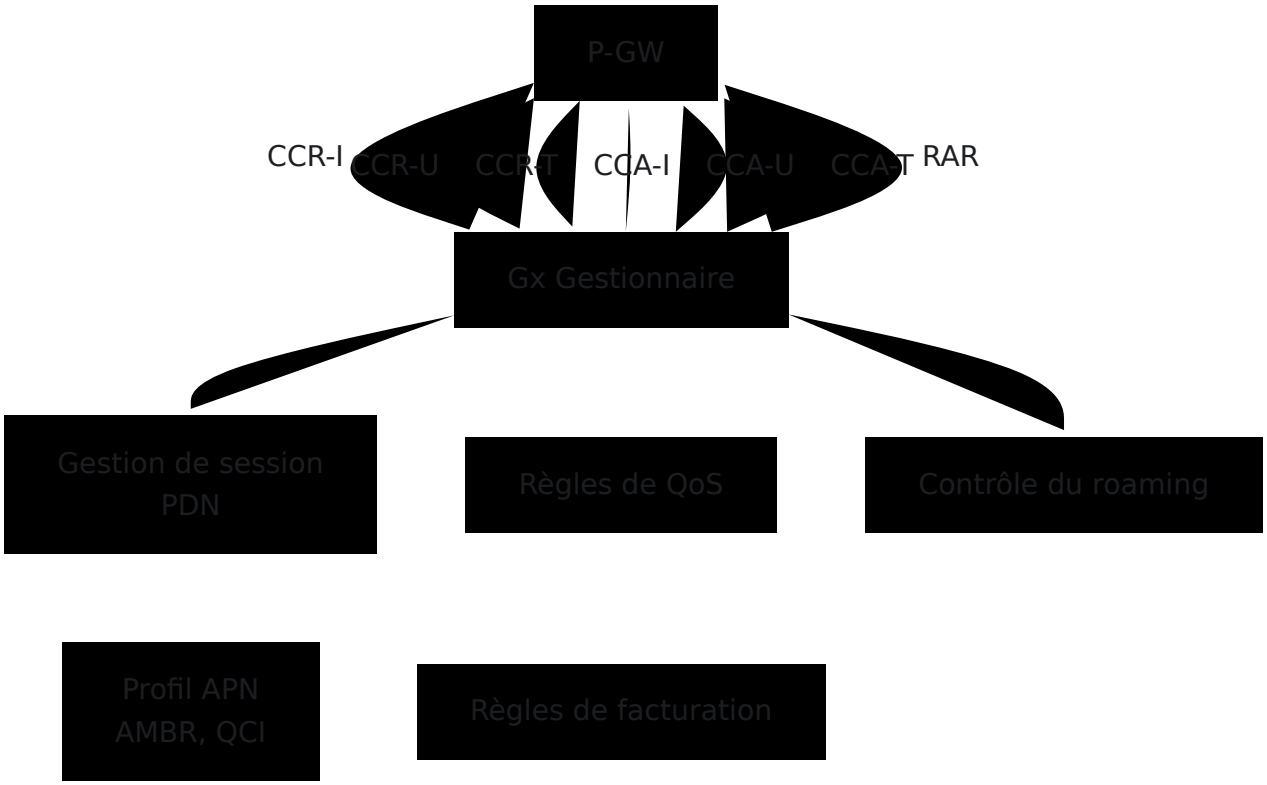
Interface Sh (Données de profil IMS)

Fournit aux serveurs d'application IMS un accès aux données de profil des abonnés.



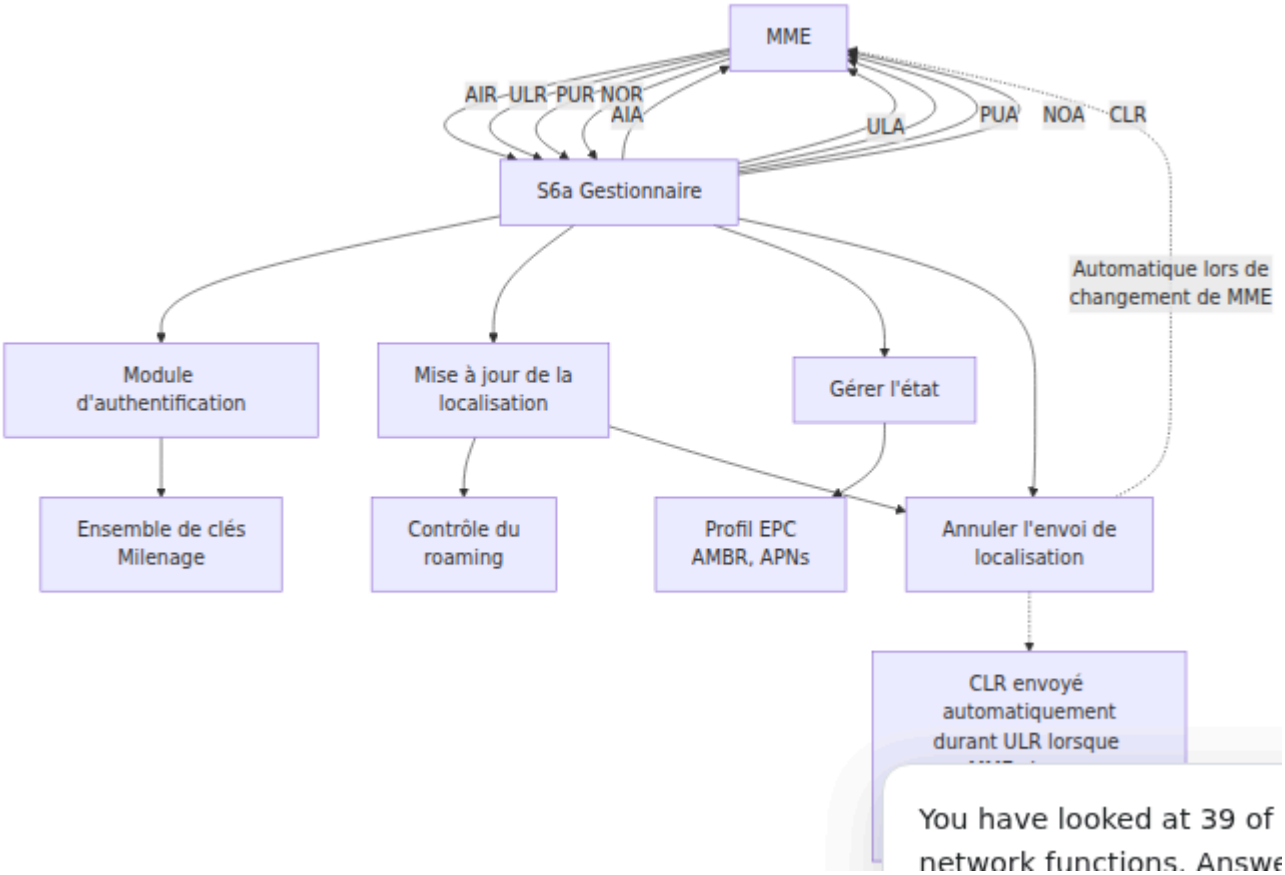
Interface Gx (Contrôle de politique)

Gère le contrôle de politique et de facturation pour les sessions de données.
Voir Documentation PCRF pour plus de détails.



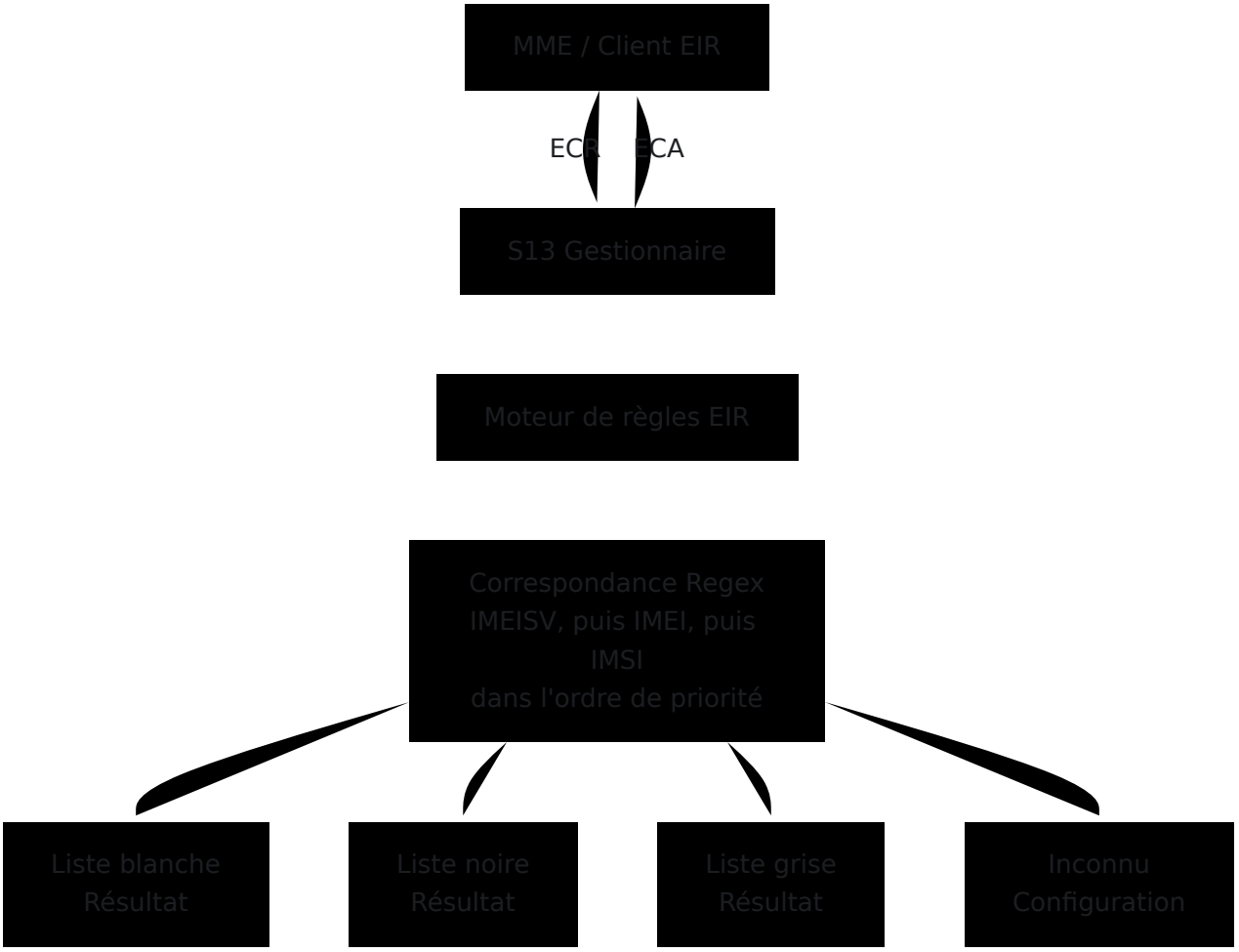
Interface Rx (Média IMS)

Contrôle la politique média IMS et les porteurs dédiés pour VoLTE. **Voir Documentation PCRF pour plus de détails.**



Interface S13 (EIR)

OmniHSS agit en tant qu'EIR. Il valide l'IMEISV et l'IMEI de l'appareil, et éventuellement l'IMSI de l'abonné lorsque l'ECR contient un AVP Nom d'utilisateur, selon les règles d'identité de l'équipement évaluées par ordre de priorité croissant. **Voir Documentation EIR pour plus de détails.**



Couche de données

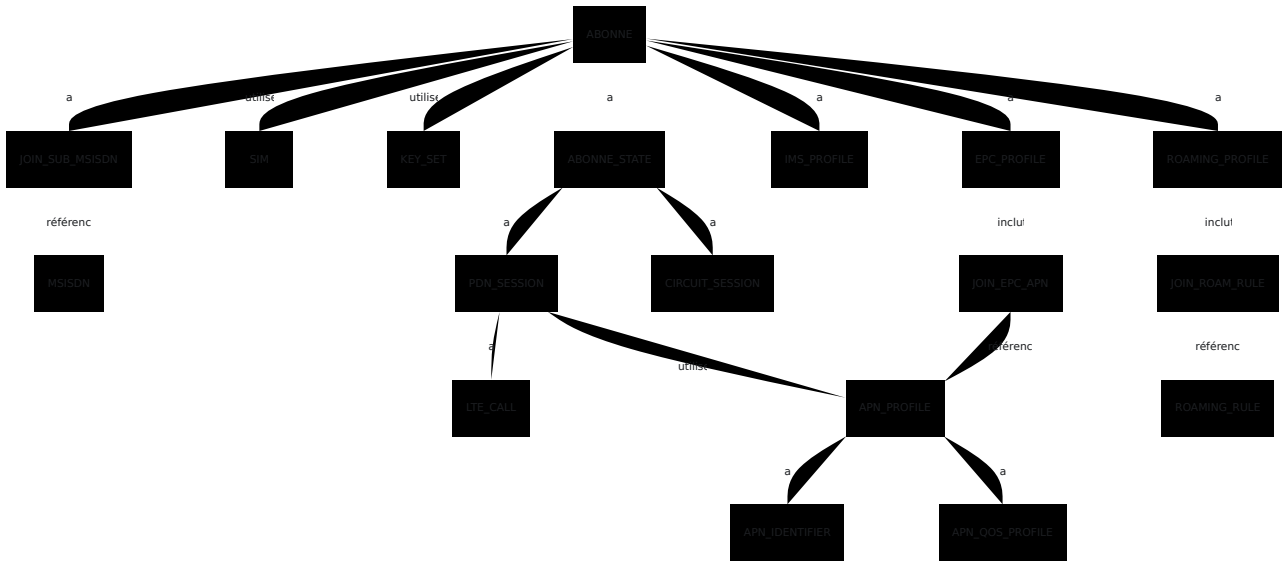
Backend de base de données

OmniHSS utilise **PostgreSQL** comme backend de base de données et **Ecto** comme couche d'abstraction de base de données. Toutes les tables utilisent des clés primaires UUID pour une identification unique à l'échelle mondiale à travers les sites.

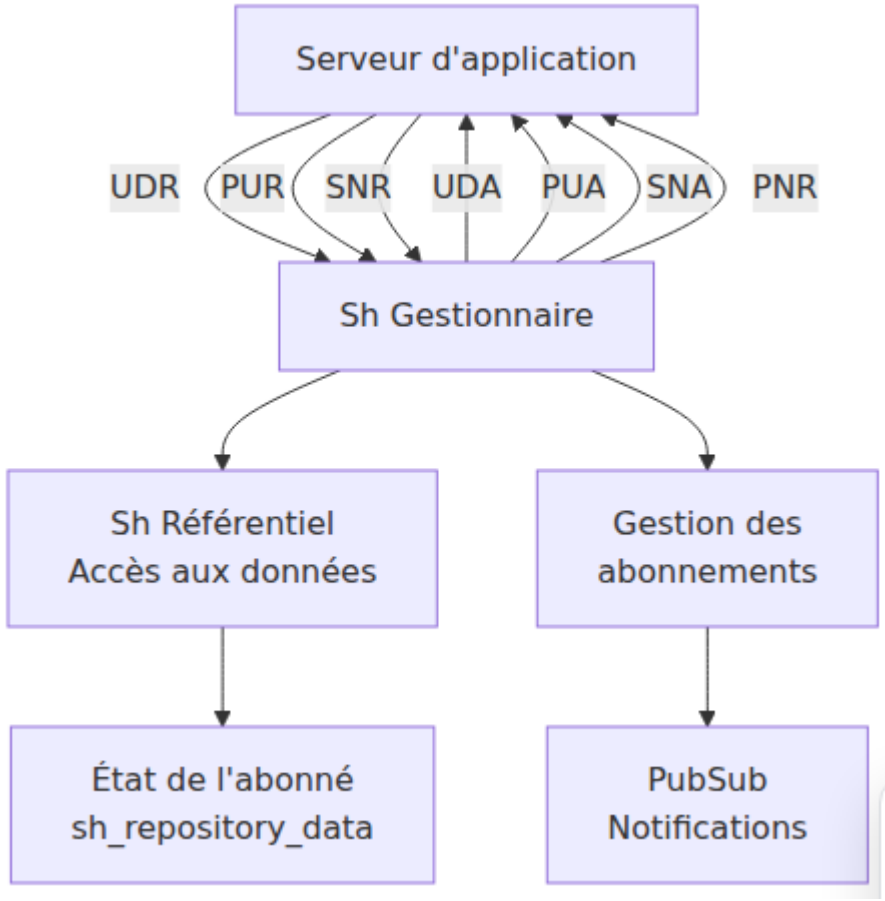
Pour les déploiements multi-sites, OmniHSS utilise la réplication logique bidirectionnelle de PostgreSQL. Chaque site fonctionne comme un primaire en lecture/écriture indépendant avec une réplication asynchrone basée sur WAL entre les sites.

Voir Réplication de base de données pour la configuration et les opérations de réplication multi-sites.

Vue d'ensemble du schéma de base de données

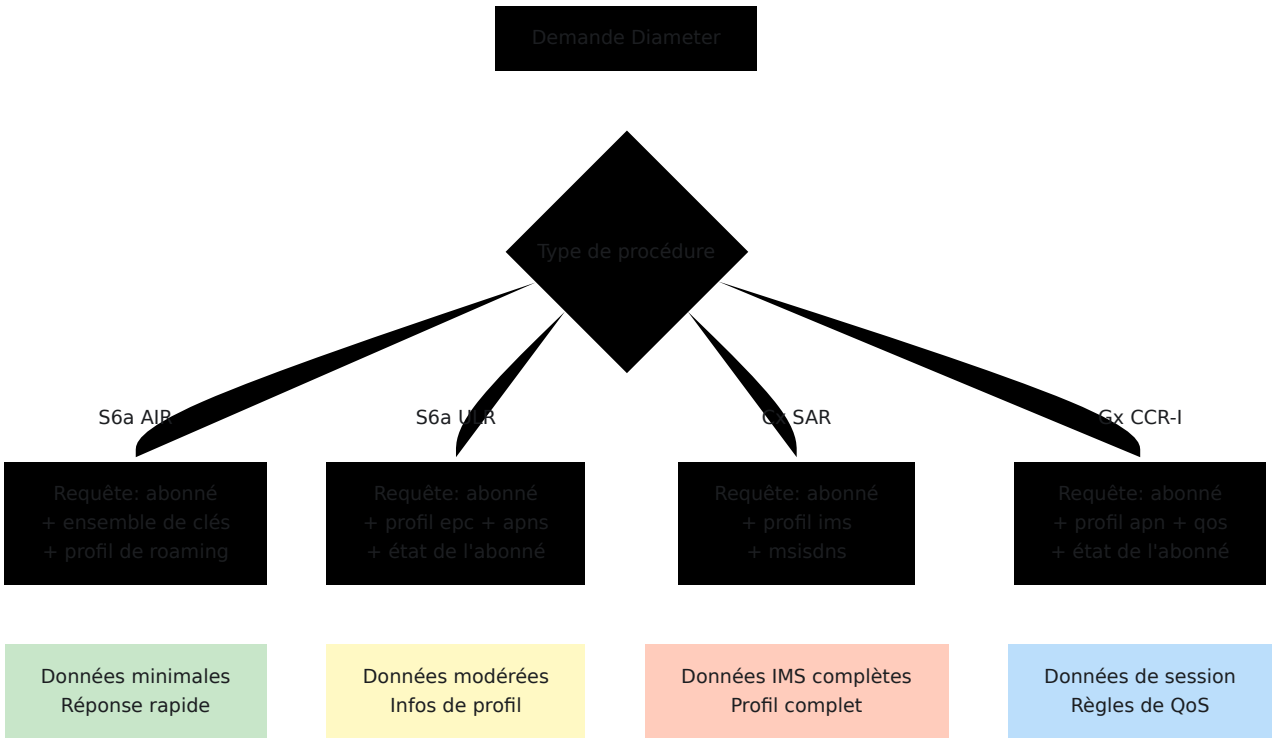


Modèle de référentiel Ecto



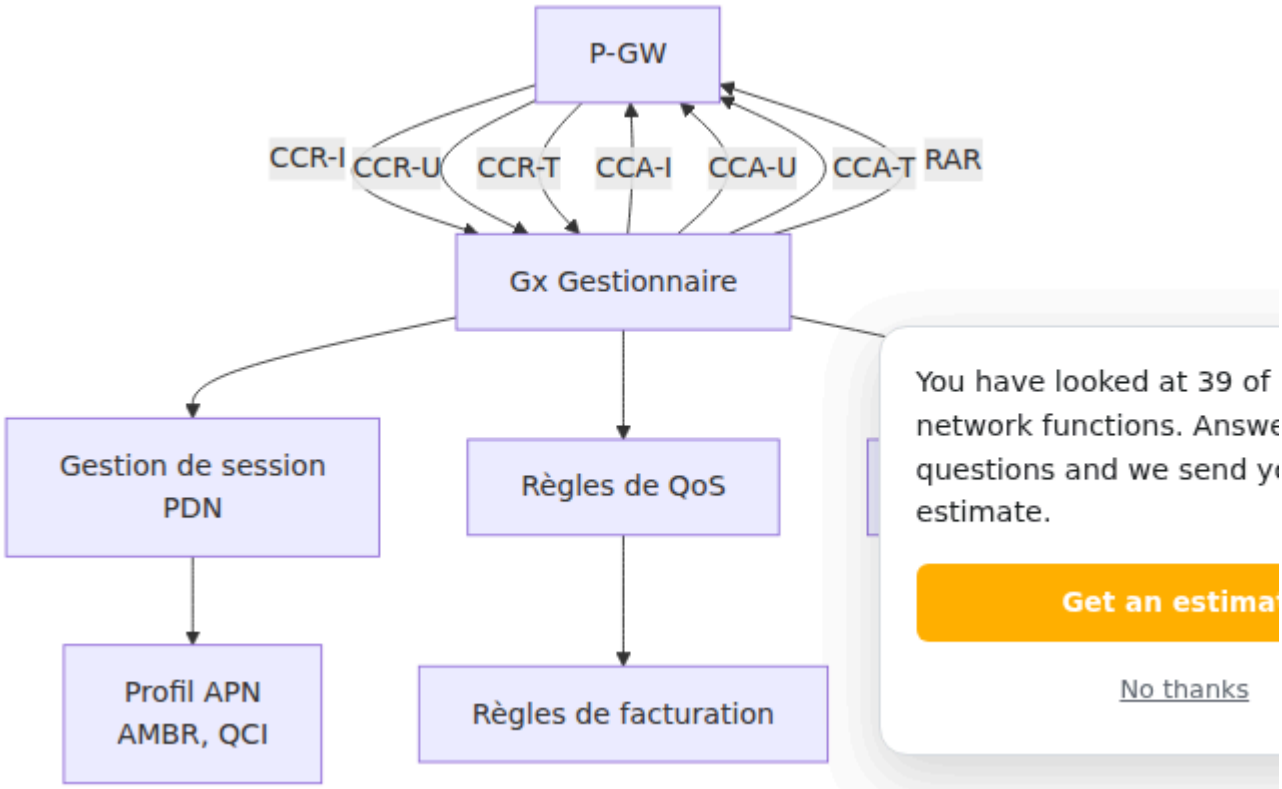
Stratégie de requête optimisée

Chaque procédure Diameter utilise des requêtes optimisées qui préchargent uniquement les associations nécessaires :

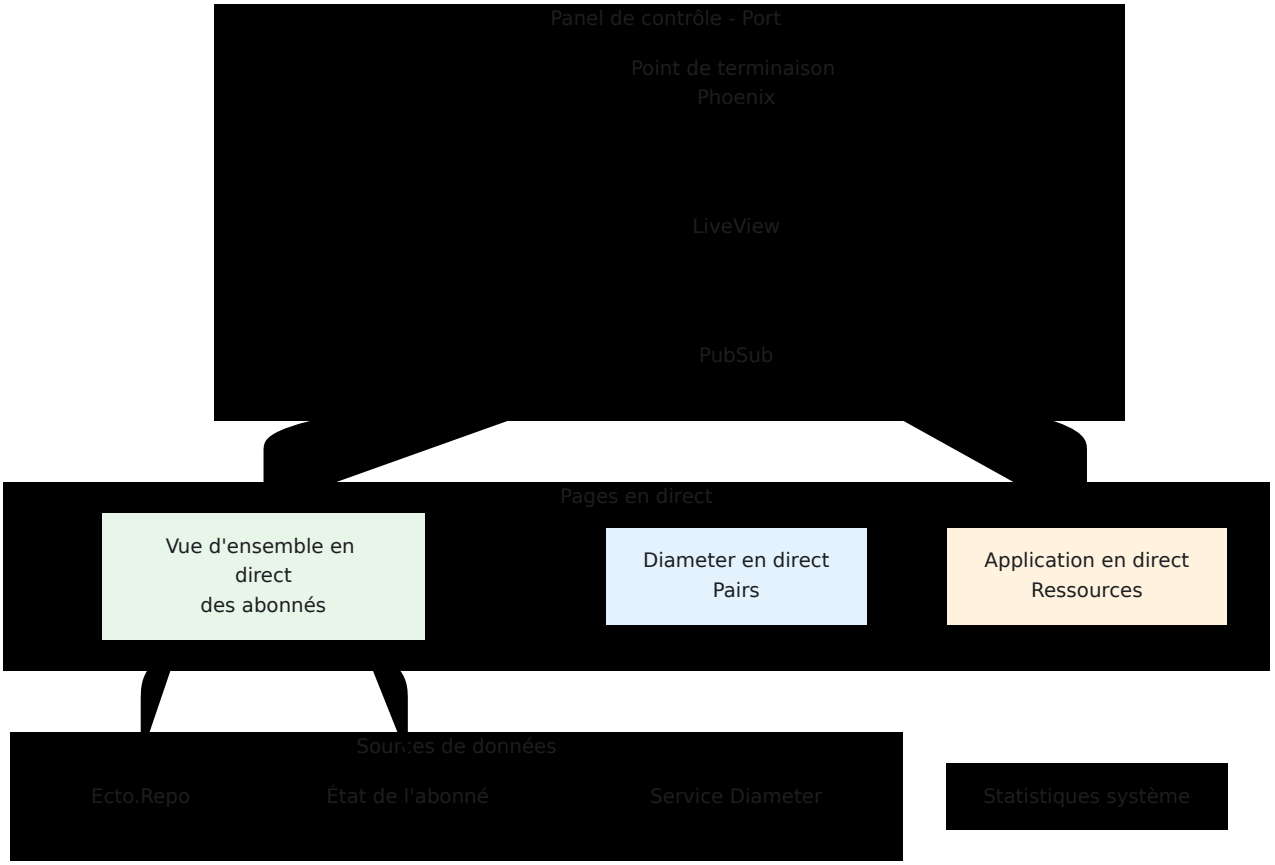

Demande Diameter
Type de procédure
S6a AIR
S6a ULR
Gx SAR
Gx CCR-I
Requête: abonné
+ ensemble de clés
+ profil de roaming
Requête: abonné
+ profil epc + apns
+ état de l'abonné
Requête: abonné
+ profil ims
+ msisdns
Requête: abonné
+ profil apn + qos
+ état de l'abonné
Données minimales
Réponse rapide
Données modérées
Infos de profil
Données IMS complètes
Profil complet
Données de session
Règles de QoS

Interfaces externes

Architecture API

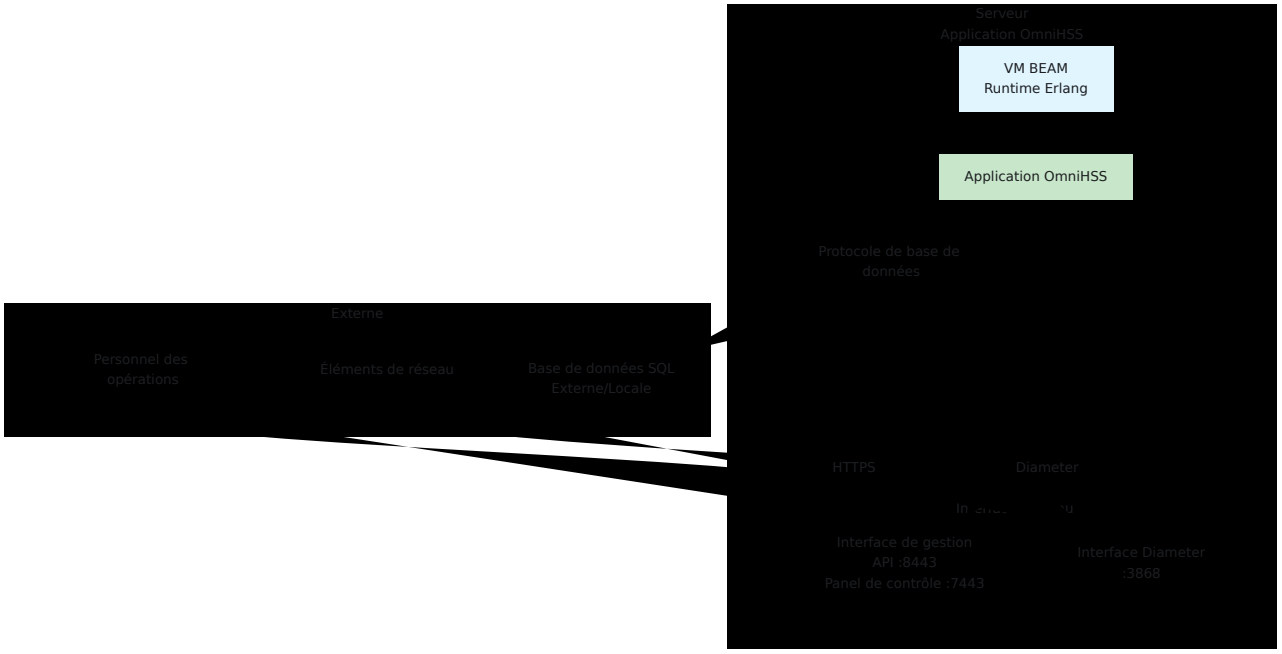


Architecture du panel de contrôle



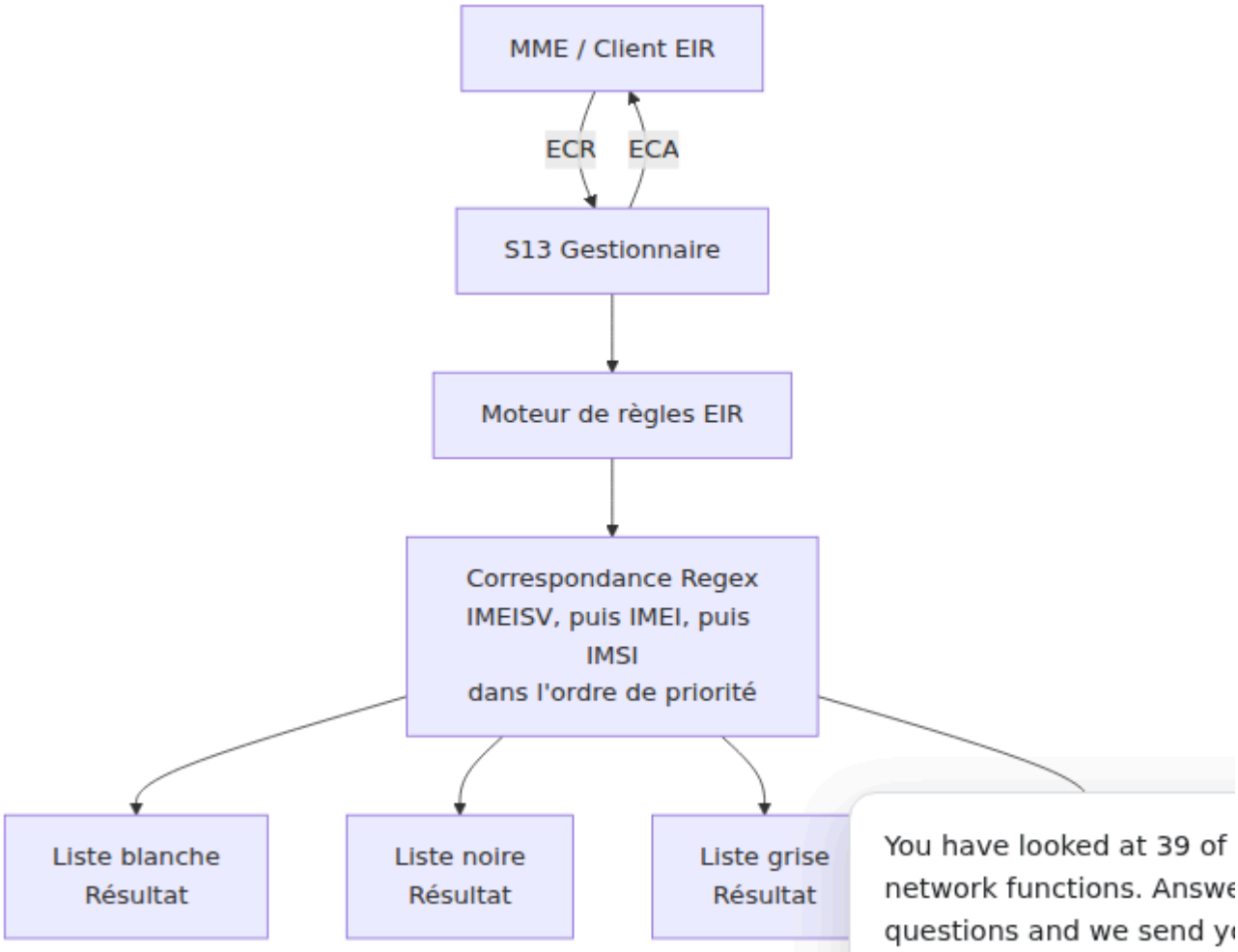
Architecture de déploiement

Déploiement sur un seul nœud



Déploiement HA multi-sites (Réplication logique)

Pour les déploiements à haute disponibilité à travers des sites géographiques, OmniHSS utilise la réplication logique bidirectionnelle de PostgreSQL. Chaque site dispose d'une base de données en lecture/écriture indépendante.



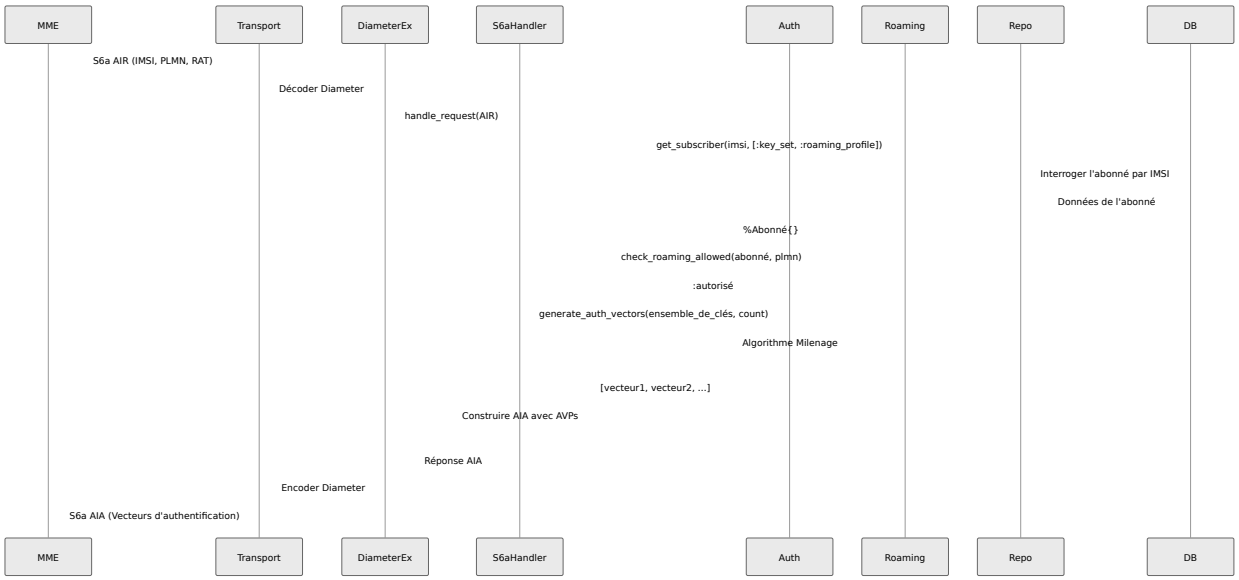
Caractéristiques clés :

- **Les deux sites en lecture/écriture** : Les écritures locales sont instantanées, jamais bloquées par le site distant
- **Réplication asynchrone** : Les modifications sont diffusées via WAL. La réplication n'ajoute aucun délai de latence d'écriture
- **Tolérant aux partitions** : Les deux sites continuent de fonctionner si le lien inter-site est coupé. Les files d'attente WAL localement et se rattrapent lors de la restauration
- **Pas de quorum** : Aucun nombre minimum de nœuds requis. Fonctionne de manière identique avec un site ou plusieurs
- **Clés primaires UUID** : Des ID uniques à l'échelle mondiale éliminent les conflits entre les sites

Voir Réplication de base de données pour la configuration et les opérations détaillées.

Exemple de flux de processus : Authentification

Cet exemple montre le flux complet d'une demande d'authentification :



Principes architecturaux clés

1. Tolérance aux pannes

- Les arbres de supervision Erlang/OTP redémarrent automatiquement les processus échoués
- Les gestionnaires Diameter isolés empêchent les pannes en cascade
- Pooling de connexions à la base de données avec reconnexion automatique

2. Concurrence

- Chaque demande Diameter est traitée dans son propre processus
- Aucun état partagé entre les gestionnaires de demandes
- Pooling de connexions à la base de données pour des requêtes parallèles

3. Modularité

- Chaque application Diameter dans un module séparé
- Séparation claire entre les couches d'interface, de logique métier et de données
- Algorithmes d'authentification pluggables

4. Performance

- Requêtes de base de données optimisées avec préchargement sélectif
- Transfert de données minimal pour chaque type de procédure
- Pooling de connexions et keepalive

5. Observabilité

- Surveillance en temps réel via le panel de contrôle
- Journalisation structurée dans toute l'application
- Suivi de l'état des pairs Diameter
- Suivi de l'état des abonnés avec des horodatages

← Retour au Guide des Opérations | Suivant : Configuration →

Guide de Configuration d'OmniHSS

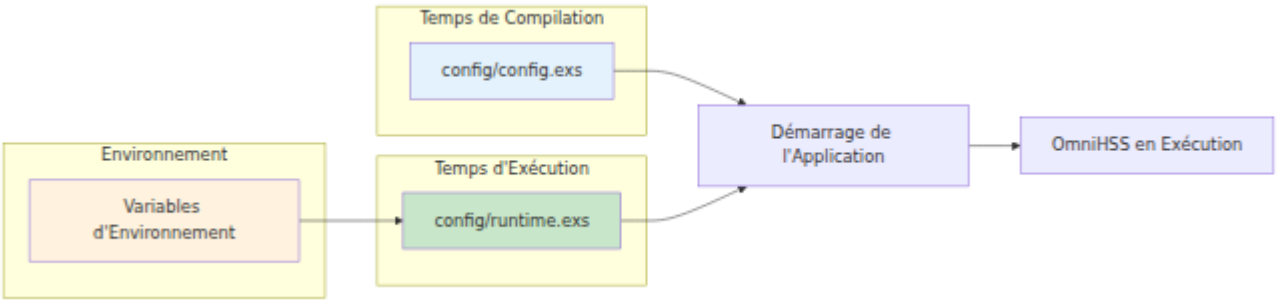
← Retour au Guide des Opérations

Table des Matières

- Aperçu du Fichier de Configuration
- Configuration du Client de Licence
- Configuration d'Exécution
- Configuration de la Base de Données
- Configuration de Diameter
- Configuration Réseau
  - Configuration du PLMN Domicile
  - Configuration du Noyau HSS
- Configuration IMS
- Configuration EIR
- Configuration de l'API et du Panneau de Contrôle
- Flux de Travail de Configuration

Aperçu du Fichier de Configuration

OmniHSS utilise deux fichiers de configuration principaux :



config/config.exs (Temps de Compilation)

Contient la configuration statique qui ne change pas entre les environnements :

- Configuration de la page du Panneau de Contrôle
- Configuration des points de terminaison de l'API
- Paramètres de télémétrie

config/runtime.exs (Temps d'Exécution)

Contient la configuration spécifique à l'environnement qui change par déploiement :

- Paramètres de connexion à la base de données
- Configuration des pairs Diameter
- Paramètres du PLMN Domicile
- Sélection de l'S-CSCF IMS
- Liaisons d'interface réseau

Configuration du Client de Licence

Le Client de Licence valide la licence HSS avec un serveur de licence distant :

```
# config/runtime.exs

config :license_client,
  # Points de terminaison de l'API du serveur de licence (liste
pour la redondance)
  license_server_api_urls:
["https://license.example.com:8443/api"],

  # Nom de l'organisation licenciée
  licensee: "Votre Nom d'Organisation",

  # Identifiant du produit
  product_name: "omnihss"
```

Paramètres de Configuration de Licence :

Paramètre	Description	Requis	Exemple
`license_server_api_urls`	Liste des URL du serveur de licence	Oui	`["https://10.0.0.1`
`licensee`	Nom de l'organisation sur la licence	Oui	`"ACME Telecom"`
`product_name`	Identifiant du produit pour la licence	Oui	`"omnihss"`

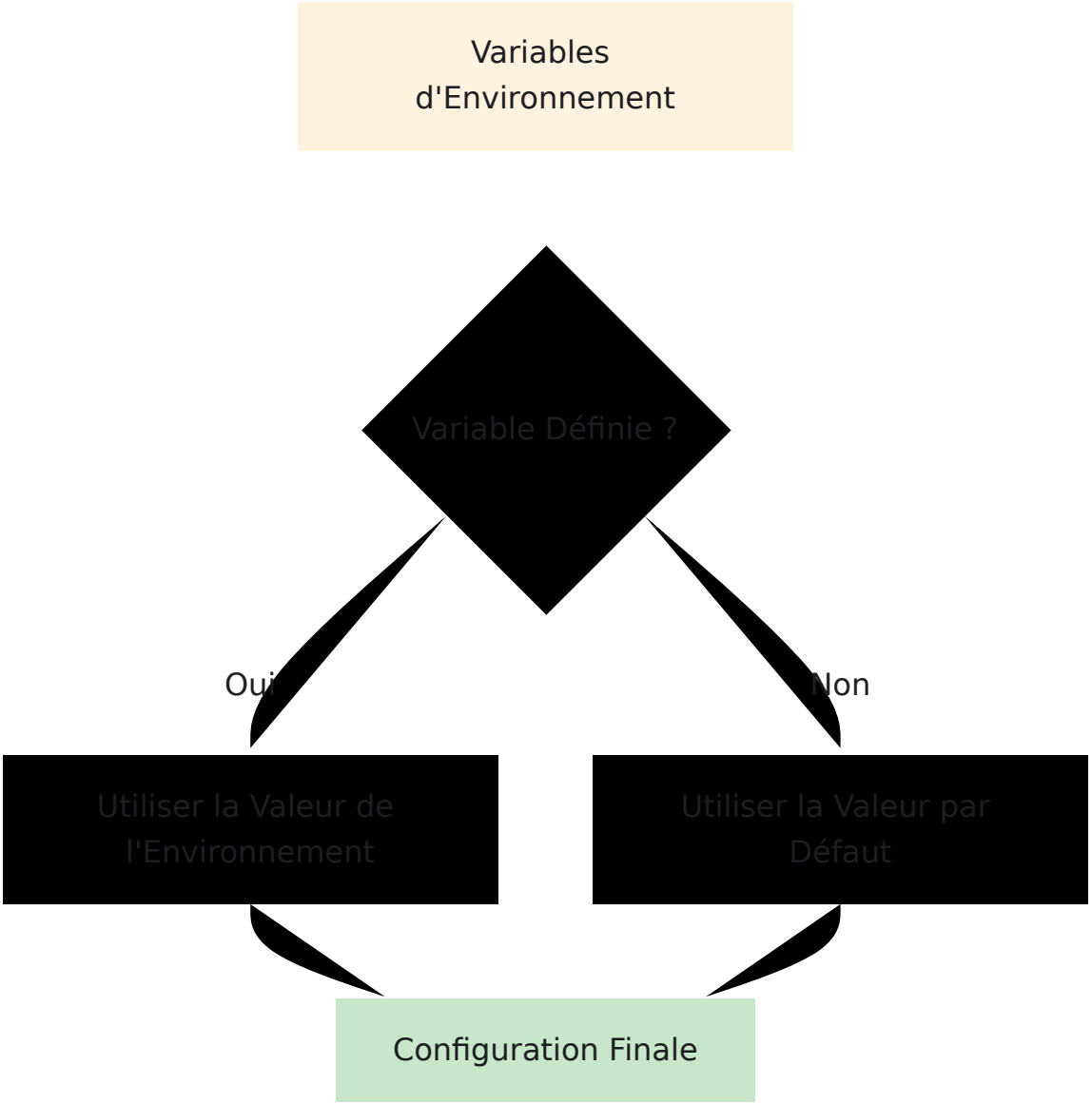
Notes Importantes :

- Le serveur de licence doit être accessible depuis HSS
- Utilisez HTTPS pour une validation de licence sécurisée
- Plusieurs URL fournissent une capacité de redondance
- La validation de la licence se produit au démarrage et périodiquement

---

Configuration d'Exécution

Priorité de Configuration



Modèle de Variable d'Environnement

OmniHSS suit ce modèle pour la configuration :

- Les noms des variables d'environnement sont en MAJUSCULES avec des underscores
- Les valeurs par défaut sont fournies dans runtime.exs

- Les identifiants de base de données doivent utiliser des variables d'environnement en production

---

Configuration de la Base de Données

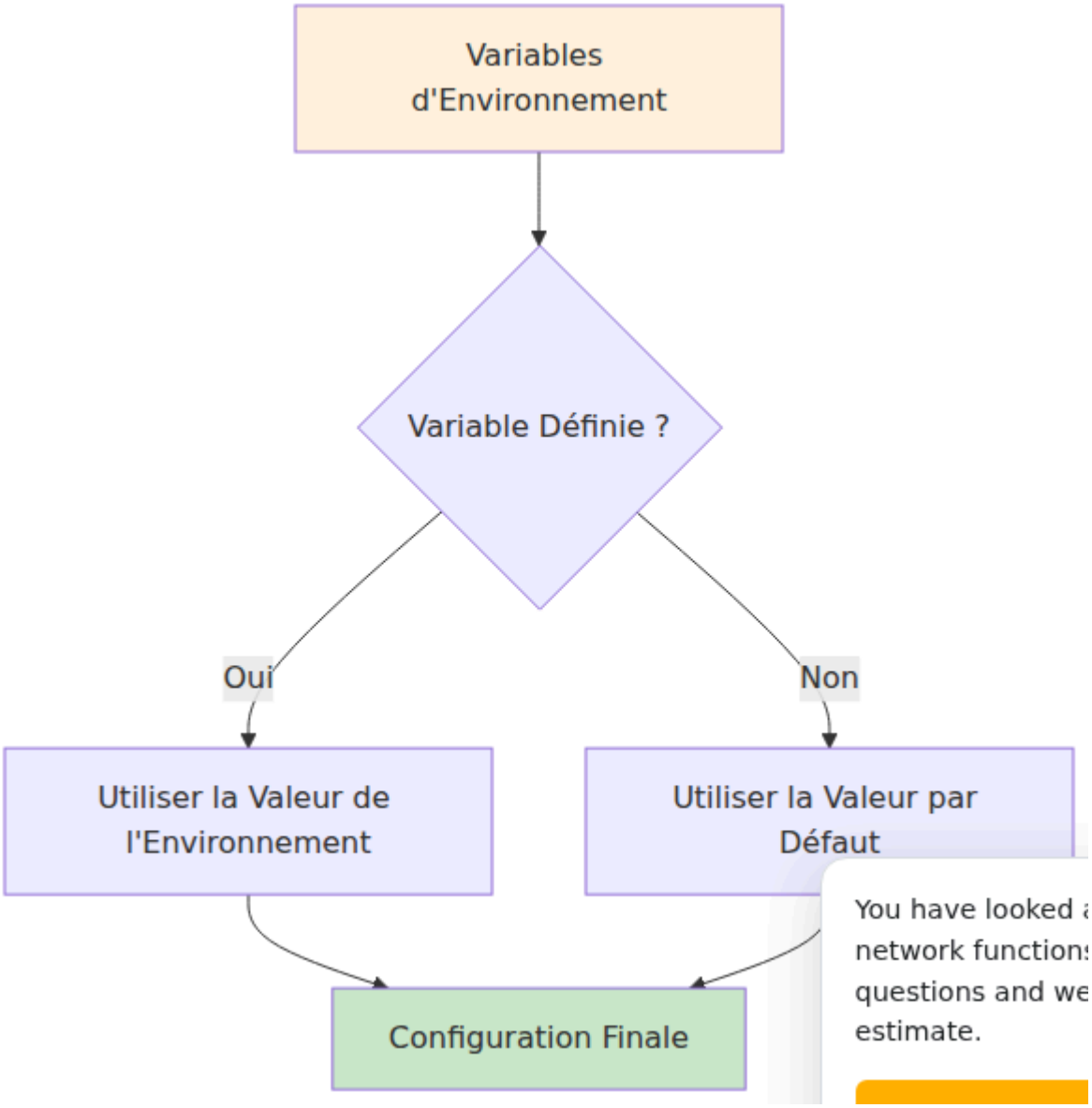
Configuration de Base de Données de Base

```
# config/runtime.exs

config :hss, Hss.Repo,
  # Paramètres de connexion à la base de données
  username: System.get_env("DATABASE_USERNAME", "root"),
  password: System.get_env("DATABASE_PASSWORD", "password"),
  hostname: System.get_env("DATABASE_HOSTNAME", "localhost"),
  database: System.get_env("DATABASE_NAME", "omnihss"),

  # Paramètres de pool de connexions
  pool_size:
String.to_integer(System.get_env("DATABASE_POOL_SIZE", "20")),

  # Délais d'attente (en millisecondes)
  timeout: 15_000,
  connect_timeout: 15_000,

  # Options supplémentaires
  show_sensitive_data_on_connection_error: false
```

Paramètres de Configuration de la Base de Données

Paramètre	Description	Par Défaut	Recommandation
`username`	Nom d'utilisateur de la base de données SQL	`"root"`	Utilisez un utilisateur dédié en production
`password`	Mot de passe de la base de données SQL	`"password"`	Utilisez un mot de passe fort, stockez dans une variable d'environnement
`hostname`	Nom d'hôte du serveur de base de données SQL	`"localhost"`	Utilisez un FQDN ou une IP en production
`database`	Nom de la base de données	`"omnihss"`	Gardez la valeur par défaut sauf si plusieurs instances
`pool_size`	Taille du pool de connexions	`20`	Ajustez en fonction de la charge (10-50 typique)

Réglage de la Taille du Pool



Directives :

- Commencez avec 20 connexions
- Surveillez les erreurs de "délai d'attente du pool de connexions"
- Augmentez de 10 si des délais d'attente se produisent sous une charge normale
- Chaque connexion utilise ~4 Mo de mémoire
- Trop de connexions peuvent dégrader les performances de la base de données SQL

Exemple : Configuration de la Base de Données de Production

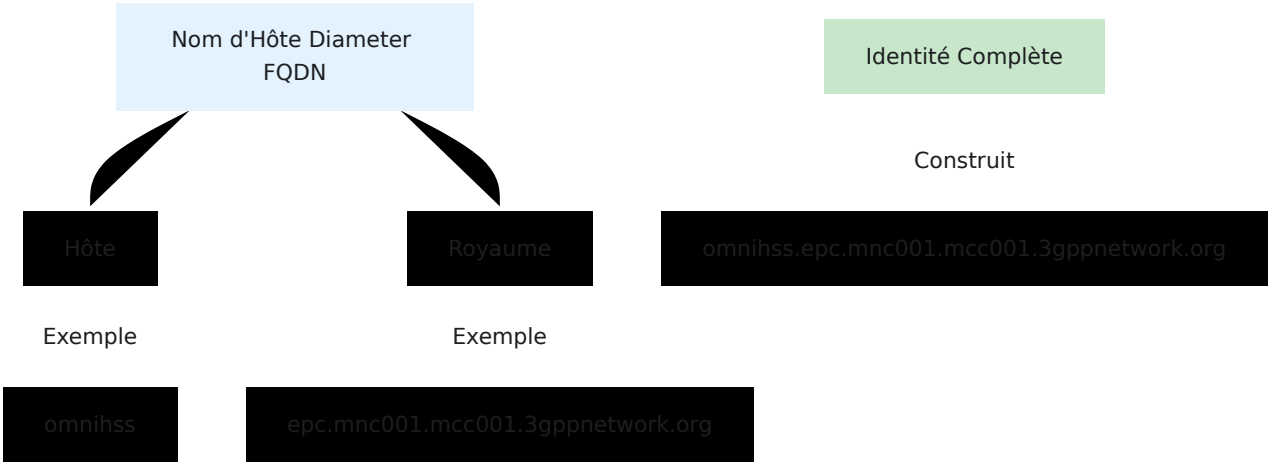
```
# config/runtime.exs - Exemple de Production

config :hss, Hss.Repo,
  username: System.fetch_env!("DATABASE_USERNAME"),      # Requis
en production
  password: System.fetch_env!("DATABASE_PASSWORD"),      # Requis
en production
  hostname: System.get_env("DATABASE_HOSTNAME",
"db.internal.example.com"),
  database: System.get_env("DATABASE_NAME", "omnihss"),
  port: String.to_integer(System.get_env("DATABASE_PORT",
"5432")),
  pool_size:
String.to_integer(System.get_env("DATABASE_POOL_SIZE", "30")),
  ssl: true,
  ssl_opts: [
    cacertfile: "/etc/ssl/certs/postgres-ca.pem",
    verify: :verify_peer
  ]
```

Configuration de Diameter

Configuration du Service Diameter

```
# config/runtime.exs

diameter_config = %{
  service_name: :omnitouch_hss,

  # Liaison réseau
  listen_ip: System.get_env("DIAMETER_LISTEN_IP", "10.7.25.186"),
  listen_port:
String.to_integer(System.get_env("DIAMETER_LISTEN_PORT", "3868")),

  # Identité Diameter
  host: System.get_env("DIAMETER_HOST", "omnihss"),
  realm: System.get_env("DIAMETER_REALM",
"epc.mnc001.mcc001.3gppnetwork.org"),

  # Identification du produit
  product_name: "OmniHSS",
  vendor_id: 10415,  # 3GPP
  supported_vendor_ids: [5535, 10415],

  # Paramètres de protocole
  request_timeout: 5000,

  # Configuration des pairs
  peers: [
    # Ajoutez ici les configurations des pairs
  ]
}

config :hss, :diameter, diameter_config
```

Configuration de l'Identité Diameter



Directives :

- **Hôte :** Nom d'hôte court du HSS (par exemple, "omnihss", "hss01")
- **Royaume :** Royaume Diameter correspondant à votre PLMN (par exemple, "epc.mnc001.mcc001.3gppnetwork.org")
- **Identité Complète :** Construit comme `{host}.{realm}`

Ajout de Pairs Diameter

Configuration de Pair Statique (Mode Connexion)

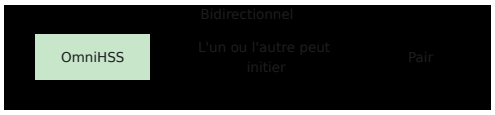
```
# config/runtime.exs

peers: [
  # Exemple de Pair MME
  %{
    host: "mme01.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.100",
    port: 3868,
    transport: :sctp,  # ou :tcp
    applications: [:s6a]
  },

  # Exemple de Pair P-GW
  %{
    host: "pgw01.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.101",
    port: 3868,
    transport: :sctp,
    applications: [:gx]
  },

  # Exemple de Pair I-CSCF
  %{
    host: "icscf01.ims.mnc001.mcc001.3gppnetwork.org",
    realm: "ims.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.102",
    port: 3868,
    transport: :tcp,
    applications: [:cx]
  }
]
```

Mode Écoute Seule

Pour les environnements où les pairs initient des connexions vers le HSS :

```
# config/runtime.exs

diameter_config = %{
  # ... autre config ...
  peers: []  # Vide - accepter uniquement les connexions entrantes
}
```

Modes de Connexion des Pairs Diameter







Sélection du Protocole de Transport

Transport	Avantages	Inconvénients	Recommandation
SCTP	Multi-streaming, meilleure détection des pannes	Nécessite un support du noyau, configuration du pare-feu	Préféré pour Diameter
TCP	Support universel, règles de pare-feu plus simples	Flux unique, détection des pannes plus lente	Utilisez si SCTP indisponible

---

Configuration Réseau

Configuration du PLMN Domicile

Le PLMN domicile identifie votre opérateur de réseau :

```
# config/runtime.exs

config :hss, :home_plmn, %{
  mcc: System.get_env("HOME_PLMN_MCC", "001"),  # Code Pays Mobile
  mnc: System.get_env("HOME_PLMN_MNC", "001")   # Code Réseau
Mobile
}
```

Configuration du Noyau HSS

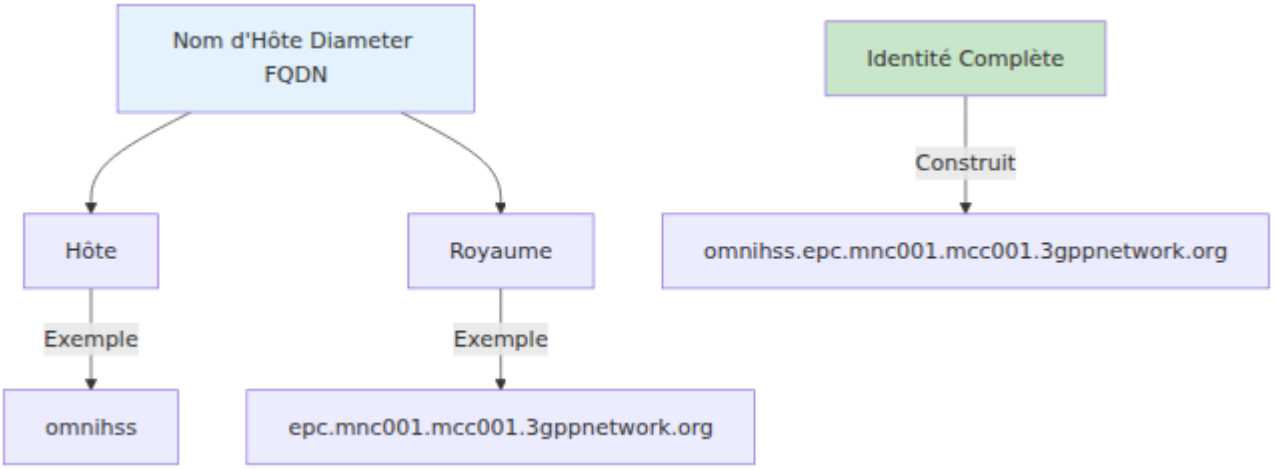
Ces paramètres contrôlent le comportement et les fonctionnalités du HSS :

```
# config/runtime.exs

config :hss,
  # Repositories Ecto pour les opérations de base de données
  ecto_repos: [Hss.Repo],

  # CLR (Cancel Location Request) lors du changement de MME
  send_clr_on_mme_change: true,

  # Arrêter le service Diameter lors des pannes de base de données
  stop_diameter_on_database_failure: true,

  # Configuration de l'application de licence
  license_enforced: true,
  license_module: LicenseClient
```

Paramètres du Noyau HSS :

Paramètre	Description	Par Défaut	Re
`ecto_repos`	Liste des repositories Ecto utilisés par l'application	`[Hss.Repo]`	Re op de
`send_clr_on_mme_change`	Envoyer une Demande d'Annulation de Localisation lorsque l'abonné change de MME	`true`	Ga un co
`stop_diameter_on_database_failure`	Désactiver le service Diameter si la base de données devient indisponible	`true`	Ac co do
`license_enforced`	Activer l'application de la licence	`true`	Re pr
`license_module`	Module gérant les vérifications de licence	`LicenseClient`	Ne

Format du Code PLMN



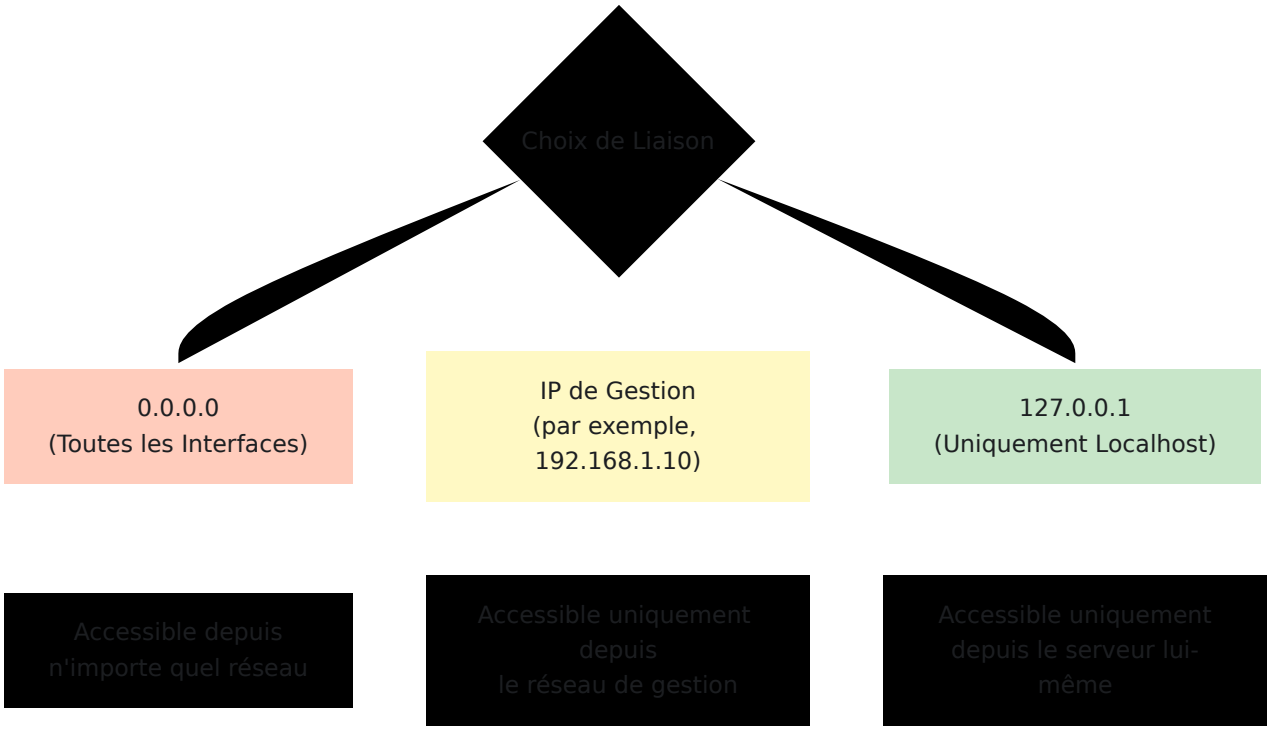
Exemples :

- AT&T (USA) : MCC=310, MNC=410
- Verizon (USA) : MCC=311, MNC=480
- Vodafone (UK) : MCC=234, MNC=15
- Réseau de Test : MCC=001, MNC=01

Liaison d'Interface Réseau

```
# config/runtime.exs

# Interface Diameter
listen_ip: System.get_env("DIAMETER_LISTEN_IP", "0.0.0.0"),  #
Toutes les interfaces
# Ou interface spécifique :
# listen_ip: "10.7.25.186",

# Interface API
config :hss, HssWeb.Api.Endpoint,
  http: [
    ip: {0, 0, 0, 0},  # Toutes les interfaces
    port: 8443
  ]

# Interface du Panneau de Contrôle
config :hss, HssWeb.ControlPanel.Endpoint,
  http: [
    ip: {0, 0, 0, 0},  # Toutes les interfaces
    port: 7443
  ]
```

Options de Liaison d'Interface :



Configuration IMS

Configuration de Sélection S-CSCF

Le HSS sélectionne un S-CSCF uniquement lors du premier REGISTER, lorsqu'aucun S-CSCF n'est enregistré pour l'abonné. Après qu'un nom soit stocké (voir État d'Attribution S-CSCF), le HSS renvoie le nom stocké et ne re-sélectionne pas. Cela maintient chaque REGISTER dans un cycle de défi IMS-AKA sur le même S-CSCF, de sorte que le cycle converge.

Le nom stocké est toujours un vrai nœud S-CSCF. Le HSS l'apprend du SAR, car le SAR est le premier message qui indique au HSS quel nœud a pris l'enregistrement. Chaque UAR de réenregistrement et chaque LIR de terminaison renvoient ensuite ce nœud. Un abonnement IMS partagé stocke un tel nœud pour tous ses membres. Voir Un S-CSCF Servant par Abonnement.

```
# config/runtime.exs

config :hss, :ims, %{
  scscf: %{
    # Méthode de sélection du premier REGISTER : :pool_uri ou
:random_peer
    selection_method: :pool_uri,

    # Nom du pool logique S-CSCF pour :pool_uri. Pas de port, donc
l'I-CSCF effectue une
    # recherche SRV (_sip._tcp/_udp.<name>) à travers le pool. Le
HSS renvoie ce
    # nom uniquement lors d'un premier REGISTER. Il ne le stocke
jamais comme un S-CSCF servant.
    pool_uri: "sip:scscf.ims.mnc001.mcc001.3gppnetwork.org",

    # Durée pendant laquelle un défi IMS-AKA peut rester en
attente avant que le HSS cesse
    # de renvoyer le S-CSCF défiant sur un UAR.
    auth_pending_ttl_seconds: 30,

    # Nœuds S-CSCF pour :random_peer (également l'inventaire S-
CSCF du déploiement)
    peers: [
      %{
        host: "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060"
      },
      %{
        host: "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060"
      }
    ]
  }
}
```

Paramètres :

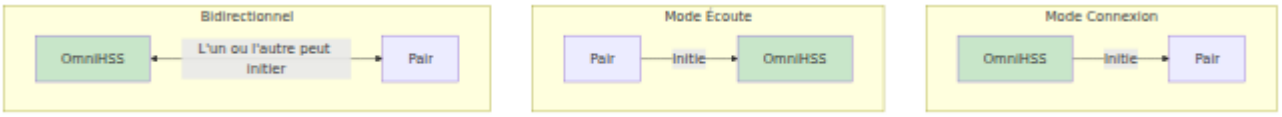
Paramètre	Type	Requis	Par Défaut	Des
`selection_method`	Atome	Oui	`:pool_uri`	Métho sélect du pre REGIS Utilise `:pool` `:rand` Voir M Sélect
`pool_uri`	Chaîne	Oui pour `:pool_uri`	-	Nom c logiqu renvo premi REGIS défini port, c CSCF une re SRV e répart à trav Le HS jamais comm serva abonn `pool_` pas de enreg averti revien nœud

Paramètre	Type	Requis	Par Défaut	Des
`auth_pending_ttl_seconds`	Entier	Non	30	Durée laquel honor en att d'auth en sec Après le HSS nom e utilise enreg place. d'Attri CSCF.
`peers`	Liste	Oui pour `:random_peer`	-	Liste d S-CSC tablea Param Pair ci

Paramètres de Pair :

Paramètre	Type	Requis	Par Défaut	Description
`host`	Chaîne	Oui	-	URI SIP S-CSCF, avec FQDN et port. L'I-CSCF doit être capable de le résoudre.

Méthodes de Sélection S-CSCF



Méthodes de Sélection :

Méthode	Description	Cas d'Utilisation
`:pool_uri`	Retourne le `pool_uri` unique. L'I-CSCF le résout par DNS SRV et le répartit à travers le pool. DNS répartit la charge et ignore un nœud mort lors du premier REGISTER.	Répartition de charge plus redondance au premier REGISTER. Recommandé par défaut.
`:random_peer`	Retourne un nœud de `peers`, choisi de manière déterministe par abonné (un hachage de l'IMSI sur la liste triée des pairs). Le même abonné correspond toujours au même nœud.	S-CSCF fixe par abonné, sans dépendance aux enregistrements SRV.

Le nom `:random_peer` est historique. La sélection est déterministe par abonné, pas aléatoire. Un mappage stable est requis, de sorte que chaque REGISTER dans un cycle de défi retourne le même S-CSCF.

Si `selection_method` contient une valeur inconnue, le HSS enregistre un avertissement et revient au comportement de `:random_peer`. Le HSS répond toujours au UAR.

La sélection s'applique uniquement au premier REGISTER. Le tableau ci-dessous montre ce que le UAA renvoie dans chaque état.

État de l'Abonné	Nom du Serveur UAA	Code de Résultat UAA
Aucun S-CSCF enregistré	Une nouvelle sélection : le `pool_uri`, ou un nœud de `peers`	2001 (première inscription)
Un défi IMS-AKA est en cours	Le nœud S-CSCF défiant	2002 (inscription suivante)
Enregistré, ou non enregistré avec le nom conservé	Le nœud S-CSCF servant	2002 (inscription suivante)
Un membre d'un abonnement IMS partagé qu'un autre membre sert déjà	Le nœud S-CSCF ancré de l'abonnement	2001 ou 2002

Configuration du Royaume IMS

Typiquement, l'IMS utilise un royaume séparé de l'EPC :

```
# Royaume EPC
"epc.mnc001.mcc001.3gppnetwork.org"

# Royaume IMS
"ims.mnc001.mcc001.3gppnetwork.org"
```

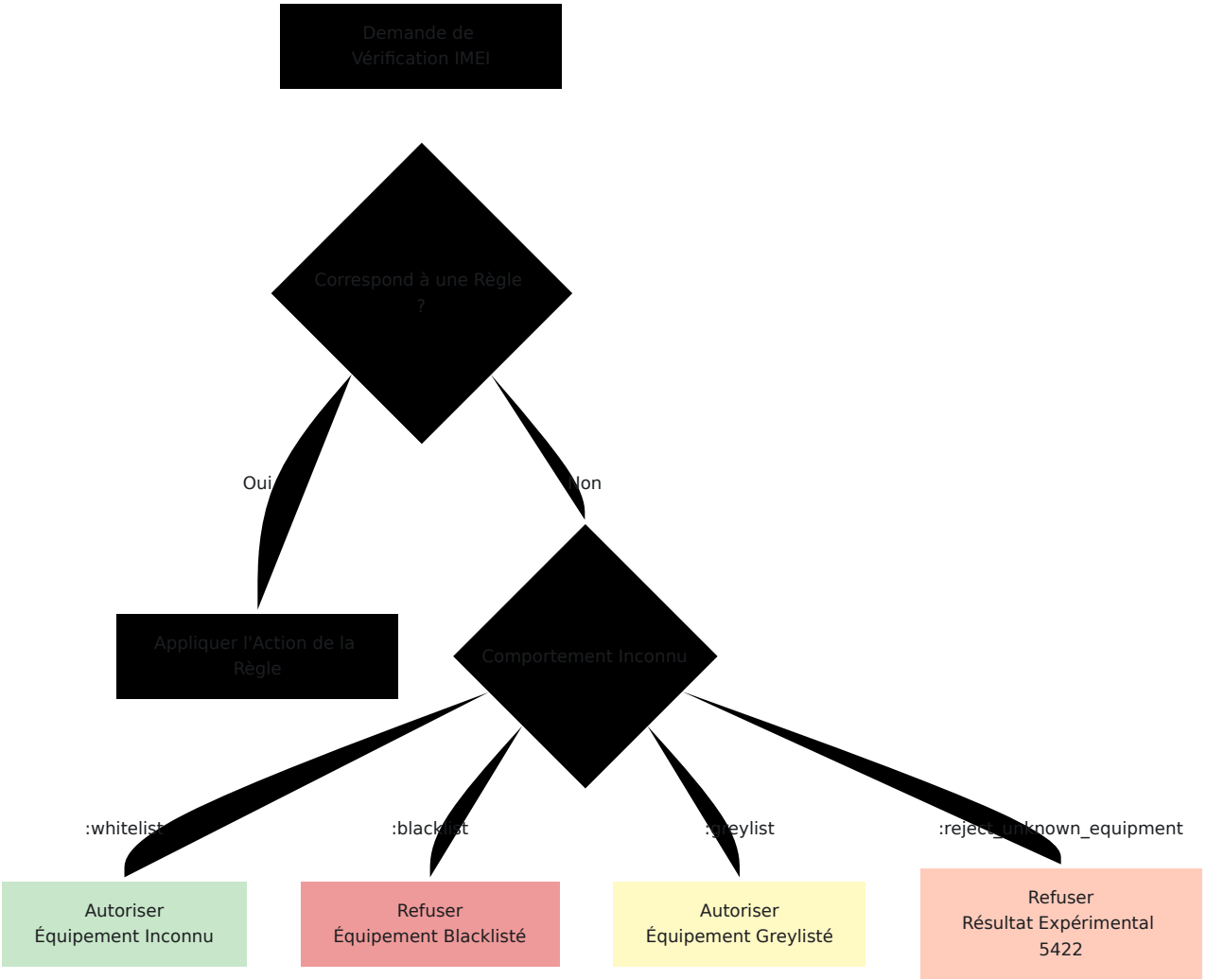
Configuration EIR

Voir Documentation EIR pour des détails complets sur la vérification de l'identité de l'équipement.

Paramètres du Registre d'Identité de l'Équipement

```
# config/runtime.exs

config :hss, :eir, %{
  # Comportement pour les équipements inconnus (aucune règle
correspondante)
  unknown_equipment_behaviour: :whitelist
  # Options :
  #   :whitelist - Autoriser les équipements inconnus
  #   :blacklist - Bloquer les équipements inconnus
  #   :greylist - Suivre mais autoriser les équipements inconnus
  #   :reject_unknown_equipment - Rejeter avec un code de résultat
spécifique
}
```

Comportement des Équipements Inconnus



Options de Comportement :

Option	Résultat	Cas d'Utilisation
`:whitelist`	Autoriser tous les IMEI inconnus	Réseau ouvert, test
`:blacklist`	Bloquer tous les IMEI inconnus	Sécurité modérée
`:greylist`	Autoriser mais suivre les IMEI inconnus	Mode de surveillance
`:reject_unknown_equipment`	Rejeter avec un code spécifique	Haute sécurité

Recommandation : Commencez avec `:whitelist` pendant les tests, passez à `:greylist` pour la surveillance en production, puis `:blacklist` pour une sécurité stricte.

Configuration de l'API et du Panneau de Contrôle

Configuration du Point de Terminaison de l'API

```
# config/config.exs

config :hss, HssWeb.Api.Endpoint,
  url: [host: "localhost"],
  render_errors: [view: HssWeb.ErrorView, accepts: ~w(json)],
  pubsub_server: Hss.PubSub,

  # Configuration HTTPS
  https: [
    port: 8443,
    cipher_suite: :strong,
    certfile: "priv/cert/omnitouch.crt",
    keyfile: "priv/cert/omnitouch.pem"
  ]
```

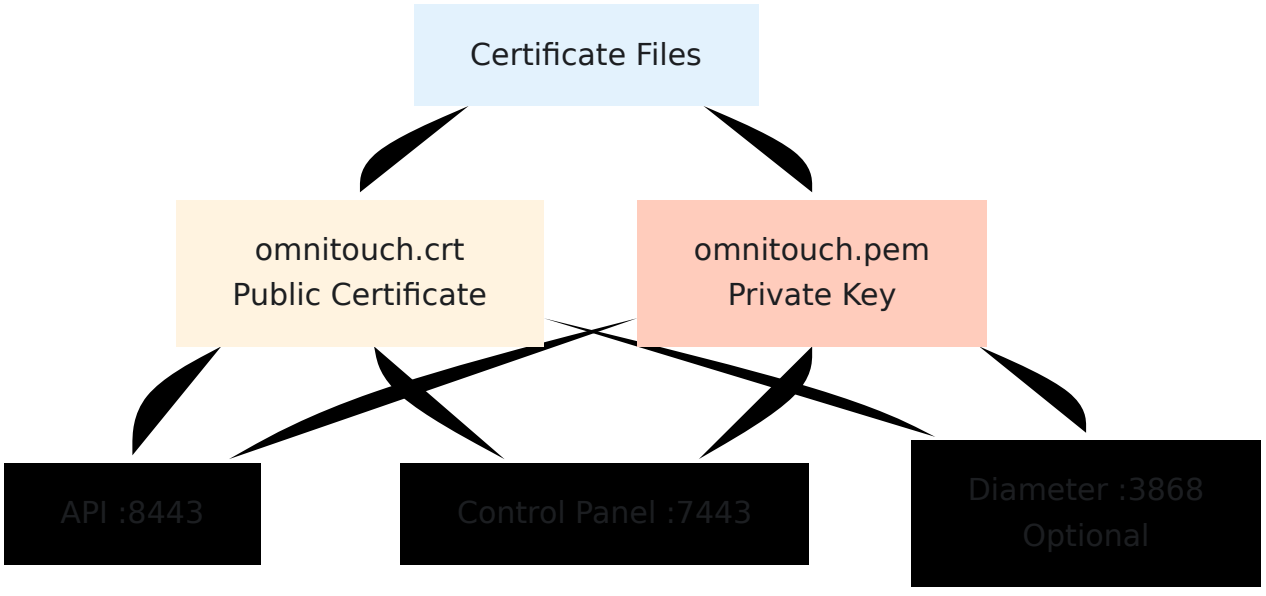
Contrôle d'Accès à l'API

Le bloc `config :api_ex, :auth` dans `runtime.exs` définit les clés et rôles de l'API. Pour tous les paramètres, voir Contrôle d'Accès à l'API (RBAC).

Configuration du Panneau de Contrôle

```
# config/config.exs

config :hss, HssWeb.ControlPanel.Endpoint,
  url: [host: "localhost"],
  render_errors: [view: HssWeb.ErrorView, accepts: ~w(html json)],
  pubsub_server: Hss.PubSub,
  live_view: [signing_salt: "some-secret"],

  # Configuration HTTPS
  https: [
    port: 7443,
    cipher_suite: :strong,
    certfile: "priv/cert/omnitouch.crt",
    keyfile: "priv/cert/omnitouch.pem"
  ]
```

Configuration du certificat TLS



Exigences du certificat :

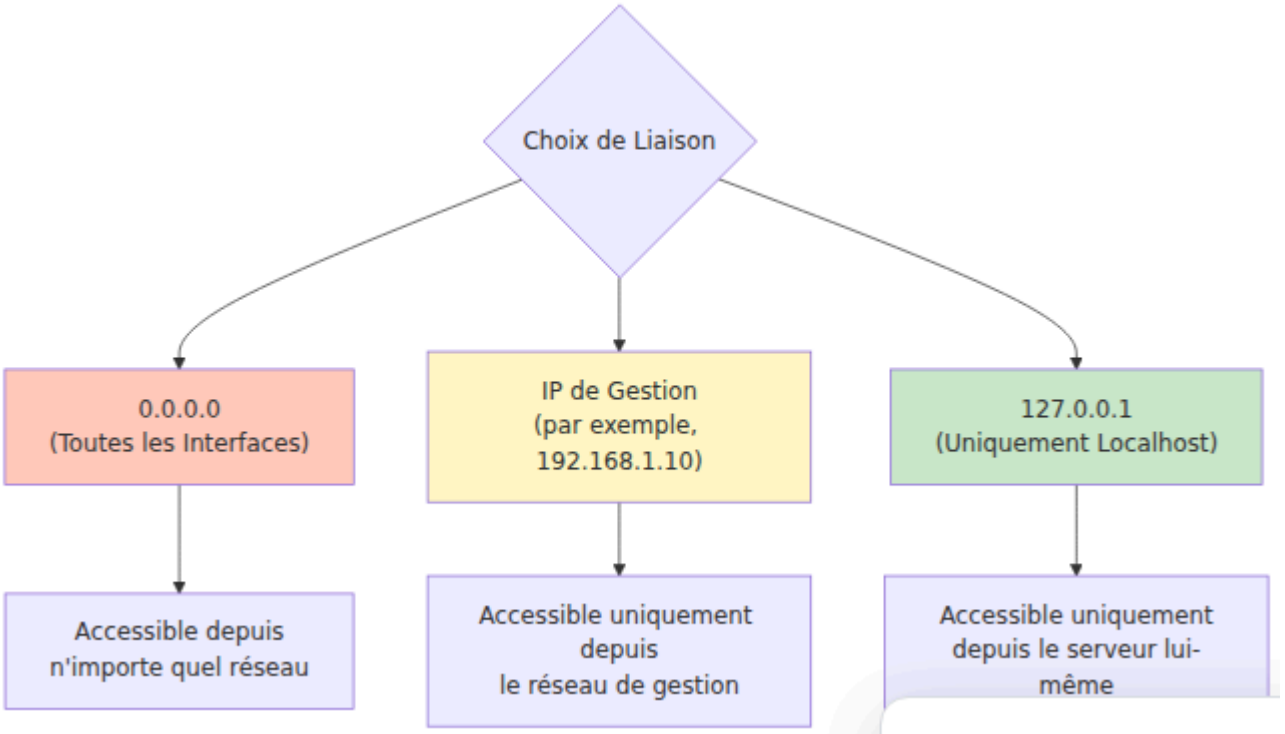
- Certificat X.509 valide
- Clé privée correspondante
- Inclure les certificats intermédiaires si nécessaire
- CN ou SAN doit correspondre au nom d'hôte

Pour la production :

```
https: [
  port: 8443,
  cipher_suite: :strong,
  certfile: System.get_env("TLS_CERT_FILE",
"/etc/ssl/certs/omnihss.crt"),
  keyfile: System.get_env("TLS_KEY_FILE",
"/etc/ssl/private/omnihss.key"),
  cacertfile: System.get_env("TLS_CA_FILE", "/etc/ssl/certs/ca-
bundle.crt")
]
```

Flux de configuration

Configuration de déploiement initial



Liste de vérification de configuration

Configuration essentielle

- [ ] Connexion à la base de données (nom d'hôte, identifiants)
- [ ] PLMN domicile (MCC, MNC)
- [ ] Hôte et domaine Diameter
- [ ] IP d'écoute et port Diameter
- [ ] Certificats TLS pour l'API et le panneau de contrôle
- [ ] Configuration du client de licence (URLs du serveur, licencié, nom_du_produit)
- [ ] Paramètres principaux HSS (send_clr_on_mme_change, stop_diameter_on_database_failure)

Intégration des éléments du réseau

- [ ] Pairs Diameter configurés (si en mode connect)
- [ ] Règles de pare-feu autorisent le trafic Diameter (port 3868)
- [ ] Règles de pare-feu autorisent le trafic HTTPS (ports 7443, 8443)
- [ ] Résolution DNS pour les identités Diameter

Configuration IMS (si utilisant les fonctionnalités IMS)

- [ ] Liste des pairs S-CSCF configurée
- [ ] Méthode de sélection S-CSCF choisie
- [ ] Domaine IMS configuré

Configuration optionnelle

- [ ] Comportement EIR configuré
- [ ] Taille du pool de base de données ajustée
- [ ] Liaison de l'interface réseau restreinte

Vérification de la configuration

Après avoir modifié la configuration :

1. **Vérification de la syntaxe :**

```
Check logs for configuration loading errors
```

2. **Accès au panneau de contrôle :**

```
Access https://[hostname]:7443
Verify Overview page loads
```

3. **Accès à l'API :**

```
curl -k https://[hostname]:8443/api/status
```

4. **Statut Diameter :**

```
Check Control Panel Diameter page
Verify peer connections
```

5. **Connectivité à la base de données :**

```
Check Control Panel for subscriber data
Or connect directly to SQL Database
```

Exemple complet de configuration

d'exécution

```
# config/runtime.exs - Exemple complet de production

import Config

#
==========================================================================
# CONFIGURATION DE LA BASE DE DONNÉES
#
==========================================================================
config :hss, Hss.Repo,
  username: System.fetch_env!("DATABASE_USERNAME"),
  password: System.fetch_env!("DATABASE_PASSWORD"),
  hostname: System.get_env("DATABASE_HOSTNAME", "db.omnihss.internal'
  database: System.get_env("DATABASE_NAME", "omnihss"),
  port: String.to_integer(System.get_env("DATABASE_PORT", "5432")),
  pool_size: String.to_integer(System.get_env("DATABASE_POOL_SIZE", '
  timeout: 15_000,
  connect_timeout: 15_000,
  ssl: true,
  ssl_opts: [
    cacertfile: "/etc/ssl/certs/postgres-ca.pem",
    verify: :verify_peer
  ]

#
==========================================================================
# CONFIGURATION DU CLIENT DE LICENCE
#
==========================================================================
config :license_client,
  license_server_api_urls: [System.get_env("LICENSE_SERVER_URL",
"https://license.example.com:8443/api")],
  licensee: System.get_env("LICENSE_ORGANIZATION", "Your Organizatior
  product_name: "omnihss"

#
==========================================================================
# CONFIGURATION DU PLMN DOMICILE ET DU NUCLEUS HSS
#
==========================================================================
```

```
config :hss,
  ecto_repos: [Hss.Repo],
  home_plmn: %{
    mcc: System.get_env("HOME_PLMN_MCC", "001"),
    mnc: System.get_env("HOME_PLMN_MNC", "001")
  },
  send_clr_on_mme_change: true,
  stop_diameter_on_database_failure: true,
  license_enforced: true,
  license_module: LicenseClient

#
============================================================================
# CONFIGURATION DIAMETER
#
============================================================================
diameter_config = %{
  service_name: :omnitouch_hss,
  listen_ip: System.get_env("DIAMETER_LISTEN_IP", "10.7.25.186"),
  listen_port: String.to_integer(System.get_env("DIAMETER_LISTEN_PORT
"3868")),
  host: System.get_env("DIAMETER_HOST", "omnihss01"),
  realm: System.get_env("DIAMETER_REALM",
"epc.mnc001.mcc001.3gppnetwork.org"),
  product_name: "OmniHSS",
  vendor_id: 10415,
  supported_vendor_ids: [5535, 10415],
  request_timeout: 5000,
  peers: [
    %{
      host: "mme01.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.7.25.100",
      port: 3868,
      transport: :sctp,
      applications: [:s6a]
    }
  ]
}

config :hss, :diameter, diameter_config

#
============================================================================
```

```
# CONFIGURATION IMS
#
============================================================================
config :hss, :ims, %{
  scscf: %{
    selection_method: :random_peer,
    peers: [
      %{host: "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060"},
      %{host: "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060"}
    ]
  }
}

#
============================================================================
# CONFIGURATION EIR
#
============================================================================
config :hss, :eir, %{
  unknown_equipment_behaviour: :whitelist
}

#
============================================================================
# CONFIGURATION DES POINTS DE TERMINAISON API
#
============================================================================
config :hss, HssWeb.Api.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8443],
  https: [
    port: 8443,
    cipher_suite: :strong,
    certfile: System.get_env("TLS_CERT_FILE", "/etc/ssl/certs/omnihss
    keyfile: System.get_env("TLS_KEY_FILE", "/etc/ssl/private/omnihss
  ],
  url: [host: System.get_env("API_HOST", "api.omnihss.internal"), por
8443]

#
============================================================================
# CONFIGURATION DES POINTS DE TERMINAISON DU PANNEAU DE CONTRÔLE
#
============================================================================
config :hss, HssWeb.ControlPanel.Endpoint,
```

```
  http: [ip: {0, 0, 0, 0}, port: 7443],
  https: [
    port: 7443,
    cipher_suite: :strong,
    certfile: System.get_env("TLS_CERT_FILE", "/etc/ssl/certs/omnihss
    keyfile: System.get_env("TLS_KEY_FILE", "/etc/ssl/private/omnihss
  ],
  url: [host: System.get_env("CP_HOST", "hss.omnihss.internal"), port
```

← Retour au guide des opérations | Suivant : Relations entre entités →

Guide du Panneau de Contrôle OmniHSS

← Retour au Guide des Opérations

Table des Matières

- Aperçu du Panneau de Contrôle
- Accéder au Panneau de Contrôle
- Page d'Aperçu
- Page Diameter
- Page Application
- Page de Configuration
- Navigation et Interface

Aperçu du Panneau de Contrôle

Le Panneau de Contrôle OmniHSS est une interface de surveillance basée sur le web qui fournit une visibilité en temps réel sur l'état du système, l'activité des abonnés et la connectivité Diameter. Construit avec Phoenix LiveView, il se met à jour automatiquement sans nécessiter de rafraîchissements de page.

Caractéristiques Clés

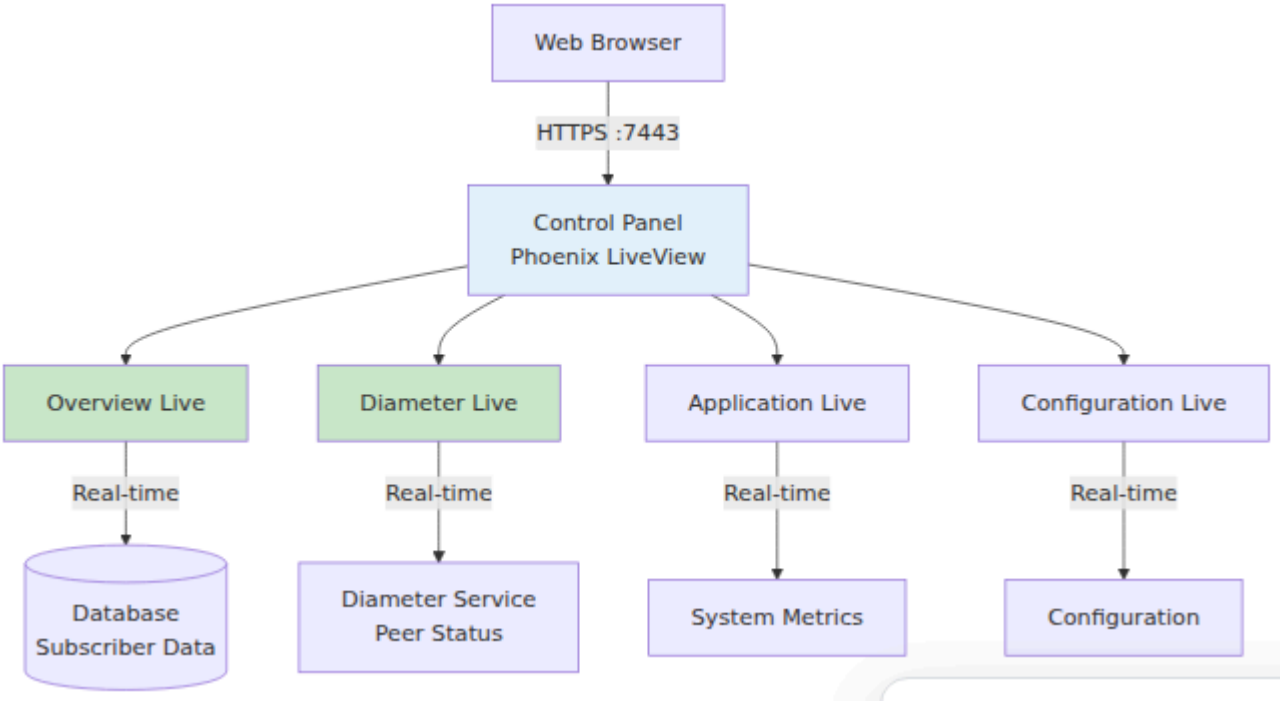
- **Mises à Jour en Temps Réel** - Se rafraîchit automatiquement chaque seconde
- **Surveillance des Abonnés** - Voir les abonnés actifs et leur état actuel
- **État Diameter** - Surveiller les connexions entre pairs en temps réel
- **Ressources Système** - Suivre la performance de l'application

- **Visionneuse de Configuration** - Inspecter la configuration à l'exécution

Informations d'Accès

```
URL: https://[hostname]:7443
Protocol: HTTPS uniquement
Port: 7443 (configurable)
Certificat: Configuré dans config/config.exs
```

Architecture du Panneau de Contrôle



Accéder au Panneau de Contrôle

Accès Initial

1. Ouvrez un navigateur web
2. Naviguez vers `https://[hostname]:7443`
3. Acceptez le certificat TLS (si auto-signé)
4. Vous serez présenté avec la page d'Aperçu par défaut

Avertissements de Certificat TLS

Si vous utilisez des certificats auto-signés, les navigateurs afficheront des avertissements de sécurité. Cela est attendu pour les déploiements internes.

Pour la Production : Utilisez des certificats signés par une Autorité de Certification de confiance.

Exigences Réseau

- **Le port 7443** doit être accessible depuis votre réseau de gestion
- **HTTPS** est obligatoire - HTTP n'est pas supporté
- **Les règles de pare-feu** doivent permettre le trafic vers le port 7443

Compatibilité des Navigateurs

Le Panneau de Contrôle utilise des technologies web modernes (LiveView, WebSockets) :

- Chrome/Chromium (recommandé)
- Firefox
- Safari
- Edge

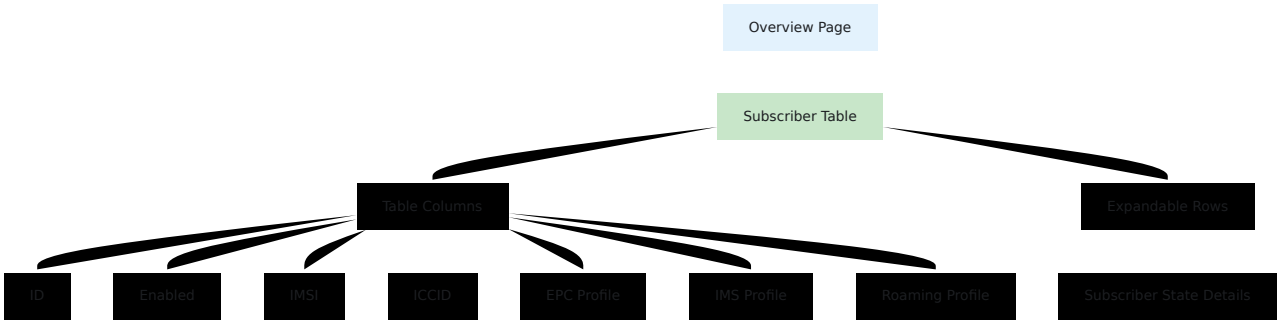
Remarque : Internet Explorer n'est pas supporté.

Page d'Aperçu

URL : `https://[hostname]:7443/overview`

La page d'Aperçu affiche tous les abonnés et leurs informations d'état en temps réel.

Mise en Page de la Page



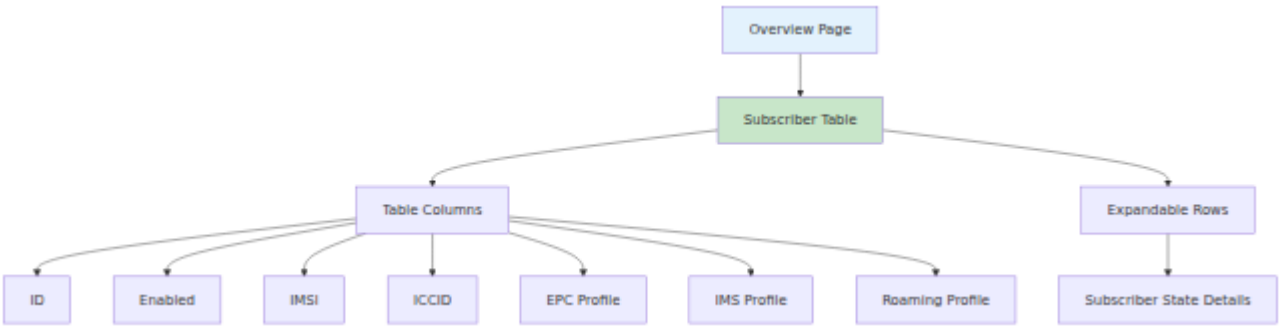
Colonnes du Tableau

Colonne	Description	Valeurs
ID	ID de la base de données des abonnés	Chaîne (UUID)
Enabled	État du service	✓ (activé) / ✗ (désactivé)
IMSI	Identité Mobile Internationale de l'Abonné	14-15 chiffres
ICCID	ID de la carte SIM	19-20 chiffres ou "N/A"
EPC Profile	Nom du profil de service de données	Nom ou ID de profil
IMS Profile	Nom du profil de service vocal	Nom de profil, ID, ou "N/A"
Roaming Profile	Nom de la politique de roaming	Nom de profil, ID, ou "N/A"

Détails des Lignes Dépliables

Cliquez sur n'importe quelle ligne pour déplier et voir l'état détaillé de l'abonné :

Informations de Localisation



Champs :

- **MCC** - Code Pays Mobile (3 chiffres)
- **MNC** - Code Réseau Mobile (2-3 chiffres)
- **TAC** - Code de Zone de Suivi
- **Cell ID** - Identifiant de la cellule de service
- **eNodeB ID** - Identifiant de la station de base
- **ECI** - Identifiant de Cellule E-UTRAN

Informations Réseau

Champs :

- **Dernier MME Vu** - Nom d'hôte MME de service actuel
- **Dernier Domaine Vu** - Domaine Diameter de MME
- **Type de RAT** - Technologie d'Accès Radio (par exemple, "E-UTRAN" pour LTE)
- **Dernière Vue À** - Horodatage du dernier message Diameter

Informations IMS

Champs :

- **S-CSCF Assigné** - URI SIP S-CSCF actuellement assigné

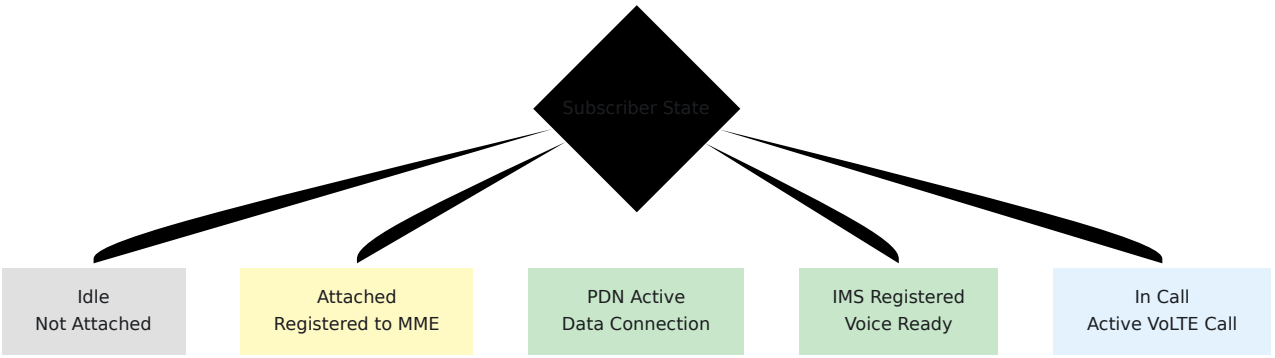
- **Identité Publique IMS** - URI SIP (par exemple, sip:+14155551234@ims.example.com)
- **Dernier P-CSCF Vu** - Dernier P-CSCF qui a contacté HSS
- **Dernier I-CSCF Vu** - Dernier I-CSCF qui a contacté HSS

Informations de Session

Champs :

- **Sessions PDN** - Nombre de connexions de données actives
- **Appels Actifs** - Nombre d'appels VoLTE actifs

Indicateurs d'État



Comment identifier l'état :

- **Idle :** Pas d'informations de localisation, pas de MME
- **Attached :** Dernier MME vu présent, informations de localisation disponibles
- **PDN Active :** Compte de sessions PDN > 0
- **IMS Registered :** S-CSCF assigné présent
- **In Call :** Compte d'appels actifs > 0

Auto-Rafraîchissement

La page d'Aperçu se rafraîchit automatiquement **toutes les 1 seconde** pour afficher des mises à jour en temps réel.

Indicateurs visuels :

- De nouvelles données apparaissent sans rechargement de page
- Les horodatages se mettent à jour en temps réel
- Aucun rafraîchissement manuel nécessaire

Cas d'Utilisation

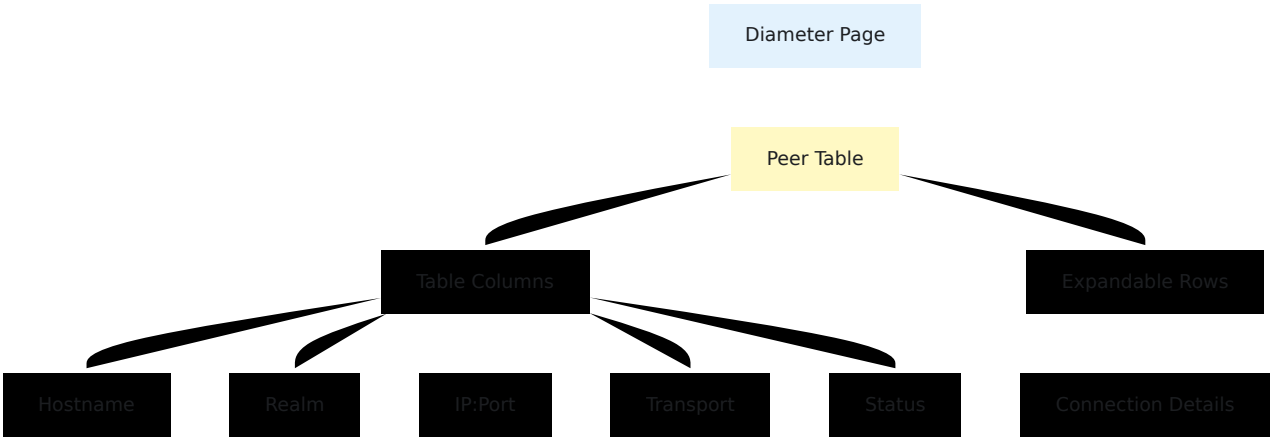
1. **Surveiller les Abonnés Actifs**
    - Voir quels abonnés sont actuellement attachés
    - Vérifier le réseau de service actuel (pour le roaming)
    - Vérifier l'état d'enregistrement IMS
2. **Dépannage**
    - Vérifier si l'abonné est activé
    - Vérifier l'horodatage de la dernière vue (l'abonné est-il réactif ?)
    - Confirmer les attributions de profil
    - Voir les informations de localisation actuelles
3. **Surveillance de Capacité**
    - Compter le nombre total d'abonnés attachés
    - Surveiller les comptes de sessions PDN
    - Suivre les appels VoLTE actifs

Page Diameter

URL : `https://[hostname]:7443/diameter`

La page Diameter montre l'état en temps réel de toutes les connexions entre pairs Diameter.

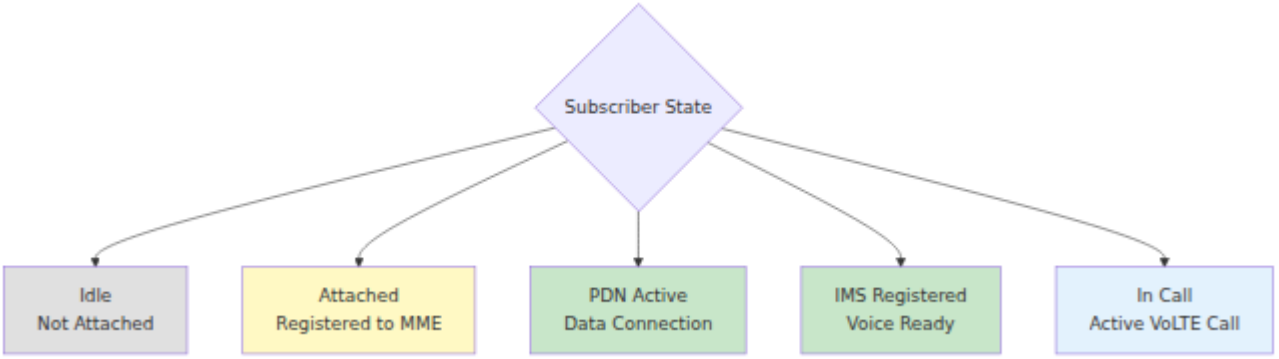
Mise en Page de la Page



Colonnes du Tableau

Colonne	Description	Valeurs
Hostname	Nom d'hôte du pair Diameter	FQDN
Realm	Domaine Diameter	Nom de domaine
IP:Port	Adresse réseau	Adresse IP et port
Transport	Protocole de transport	TCP ou SCTP
Status	État de la connexion	Connecté / Déconnecté

État de la Connexion



Détails des Lignes Dépliables

Cliquez sur n'importe quel pair pour voir des informations supplémentaires :

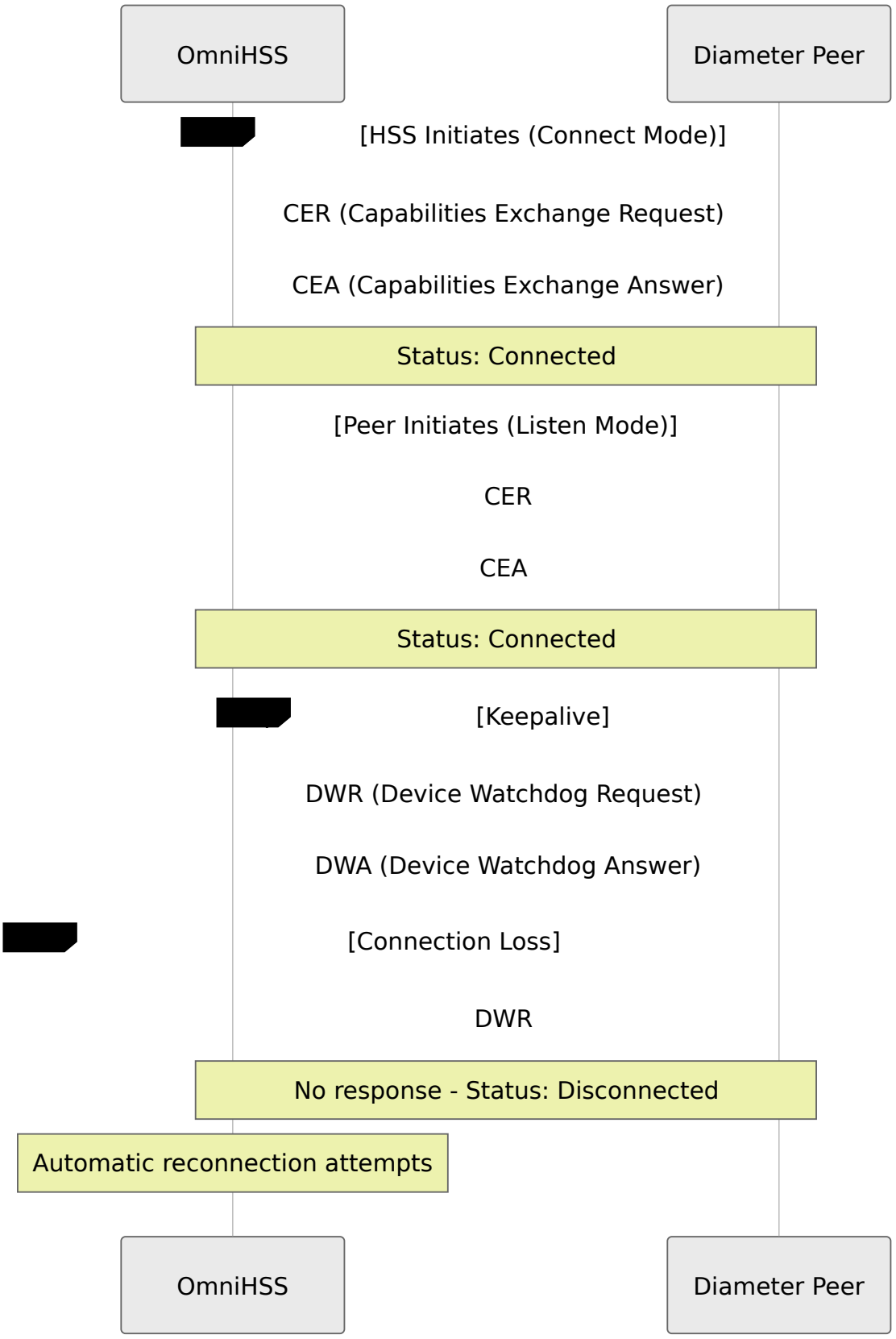
Informations de Connexion :

- **Type de Connexion** - Initié par HSS ou pair
- **Nom du Produit** - Identification du produit du pair
- **IDs d'Application** - Applications Diameter supportées

Exemples d'ID d'Application :

- 16777251 - S6a (MME)
- 16777238 - Gx (P-GW)
- 16777216 - Cx (I-CSCF, S-CSCF)
- 16777217 - Sh (Application Server)
- 16777236 - Rx (P-CSCF)
- 16777252 - S13 (client EIR, si externe)

Flux de Connexion entre Pairs



Auto-Rafraîchissement

La page Diameter se rafraîchit automatiquement **toutes les 1 seconde**.

Cas d'Utilisation

1. **Vérifier la Connectivité**
    - Assurez-vous que tous les pairs attendus sont connectés
    - Identifiez immédiatement les pairs déconnectés
    - Surveillez les connexions instables
2. **Dépannage**
    - Vérifiez si le pair est accessible
    - Vérifiez le protocole de transport (TCP vs SCTP)
    - Confirmez que les IDs d'application correspondent aux attentes
    - Identifiez quel côté a initié la connexion
3. **Planification de Capacité**
    - Comptez le nombre total de pairs connectés
    - Surveillez la stabilité des connexions
    - Planifiez une capacité supplémentaire pour les pairs

Problèmes Courants

Le Pair Montre Déconnecté

Causes Possibles :

1. Problème de connectivité réseau
2. Pair hors service ou redémarrage
3. Pare-feu bloquant le trafic
4. Mismatch de configuration Diameter
5. Problème de certificat (si utilisant TLS)

Étapes de Dépannage :

1. Vérifiez la connectivité réseau : `ping [peer-ip]`
2. Vérifiez si le port est accessible : `telnet [peer-ip] 3868`
3. Vérifiez les règles de pare-feu
4. Consultez les journaux HSS pour des messages d'erreur
5. Vérifiez que la configuration Diameter du pair correspond à celle de HSS

Le Pair Se Connecte et Se Déconnecte Répétitivement

Causes Possibles :

1. Instabilité du réseau
2. Mismatch de délai d'attente de keepalive
3. Problèmes de ressources du pair
4. Mismatch d'application Diameter

Étapes de Dépannage :

1. Vérifiez la stabilité du réseau
2. Consultez les minuteries de keepalive des deux côtés
3. Vérifiez les ressources système du pair
4. Vérifiez que les IDs d'application correspondent des deux côtés

---

Page Application

URL : `https://[hostname]:7443/application`

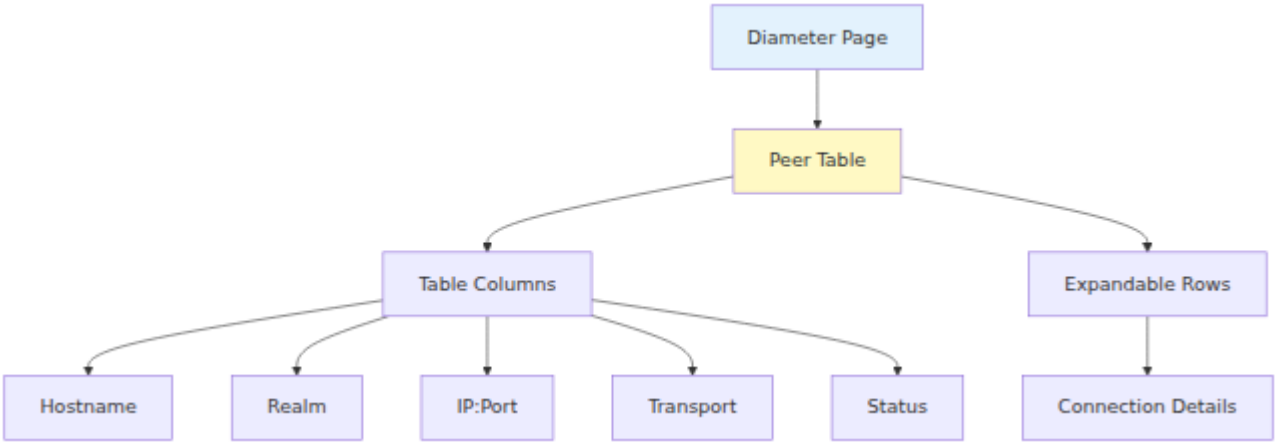
La page Application fournit des informations de surveillance au niveau du système et d'utilisation des ressources.

Caractéristiques

- **Informations sur le Processus** - Compte de processus de la VM Erlang et mémoire
- **Mémoire Système** - Mémoire totale et utilisée

- **Temps de Fonctionnement de l'Application** - Durée de fonctionnement d'OmniHSS
- **Version de la VM Erlang** - Informations sur la version d'exécution

Métriques Clés



Cas d'Utilisation

1. **Surveillance de la Santé**
    - Vérifiez que l'application fonctionne
    - Vérifiez les fuites de mémoire (augmentation de la mémoire au fil du temps)
    - Surveillez la croissance du compte de processus
2. **Planification de Capacité**
    - Suivez les tendances d'utilisation de la mémoire
    - Planifiez une montée en charge en fonction du compte de processus
    - Vérifiez que les ressources système sont adéquates
3. **Dépannage**
    - Identifiez l'épuisement des ressources
    - Vérifiez si un redémarrage est nécessaire
    - Vérifiez la version de la VM Erlang

Page de Configuration

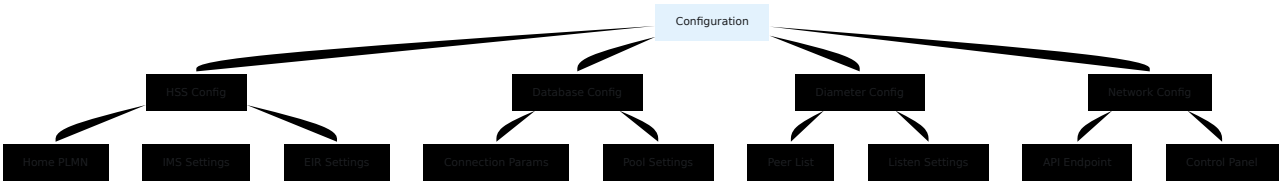
URL : `https://[hostname]:7443/configuration`

La page de Configuration affiche la configuration actuelle à l'exécution d'OmniHSS.

Caractéristiques

- **Voir la Configuration** - Inspecter tous les paramètres de configuration
- **Rechercher la Configuration** - Trouver des paramètres spécifiques
- **Variables d'Environnement** - Voir les valeurs résolues

Catégories de Configuration



Cas d'Utilisation

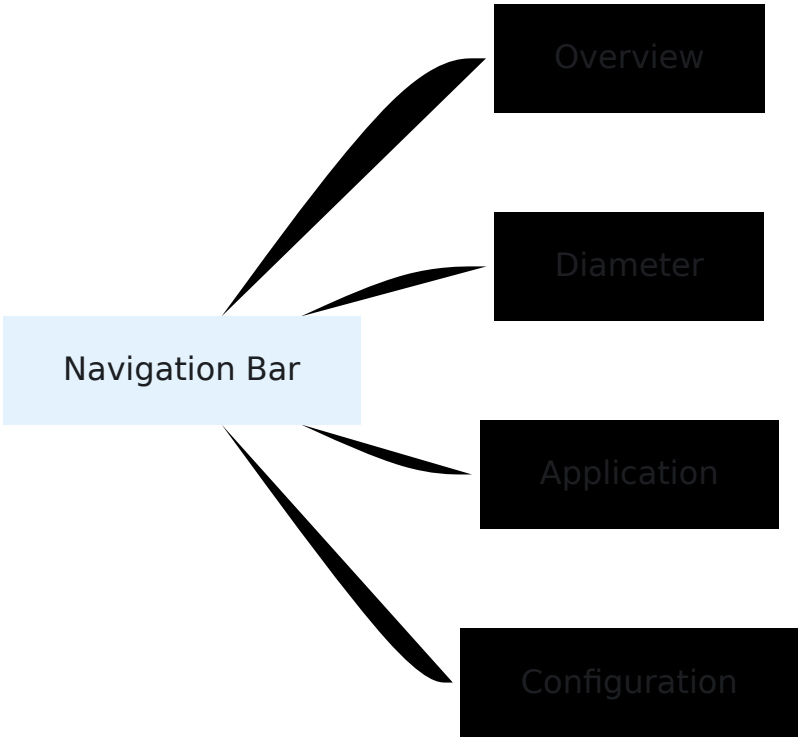
1. **Vérification de la Configuration**
    - Vérifiez que les paramètres de runtime.exs sont appliqués
    - Confirmez les paramètres de connexion à la base de données
    - Vérifiez la configuration des pairs Diameter
2. **Dépannage**
    - Identifiez les erreurs de configuration
    - Vérifiez que les variables d'environnement sont correctement définies
    - Comparez la configuration attendue à la configuration réelle
3. **Documentation**
    - Exportez la configuration actuelle pour documentation

- Partagez la configuration avec l'équipe de support

Remarque de Sécurité : La page de configuration peut afficher des informations sensibles (mots de passe de base de données, clés). Restreignez l'accès de manière appropriée.

Navigation et Interface

Barre de Navigation Supérieure



La navigation est toujours visible en haut de la page pour un accès rapide.

Raccourcis Clavier

Le Panneau de Contrôle n'implémente pas de raccourcis clavier personnalisés. Les raccourcis standards du navigateur fonctionnent :

- **Ctrl+R / F5** - Rafraîchissement manuel de la page (bien que l'auto-rafraîchissement rende cela inutile)
- **Ctrl+F** - Recherche sur la page

- **Ctrl+T** - Ouvrir un nouvel onglet (pour plusieurs pages)

Surveillance Multi-Onglets

Vous pouvez ouvrir plusieurs pages du Panneau de Contrôle dans des onglets de navigateur séparés pour une surveillance simultanée :

Exemple de Configuration :

- Onglet 1 : Page d'Aperçu (surveiller les abonnés)
- Onglet 2 : Page Diameter (surveiller la connectivité)
- Onglet 3 : Page Application (surveiller les ressources)

Tous les onglets se mettront à jour indépendamment.

Design Réactif

Le Panneau de Contrôle est optimisé pour les navigateurs de bureau. Les navigateurs mobiles sont supportés mais peuvent nécessiter un défilement horizontal pour les tableaux.

Résolution Recommandée : 1920x1080 ou plus pour un affichage confortable.

Meilleures Pratiques de Surveillance

Opérations Quotidiennes

1. **Début de Shift**
    - Ouvrez la page d'Aperçu du Panneau de Contrôle
    - Vérifiez que le nombre d'abonnés attendus est attaché
    - Vérifiez la page Diameter - tous les pairs connectés

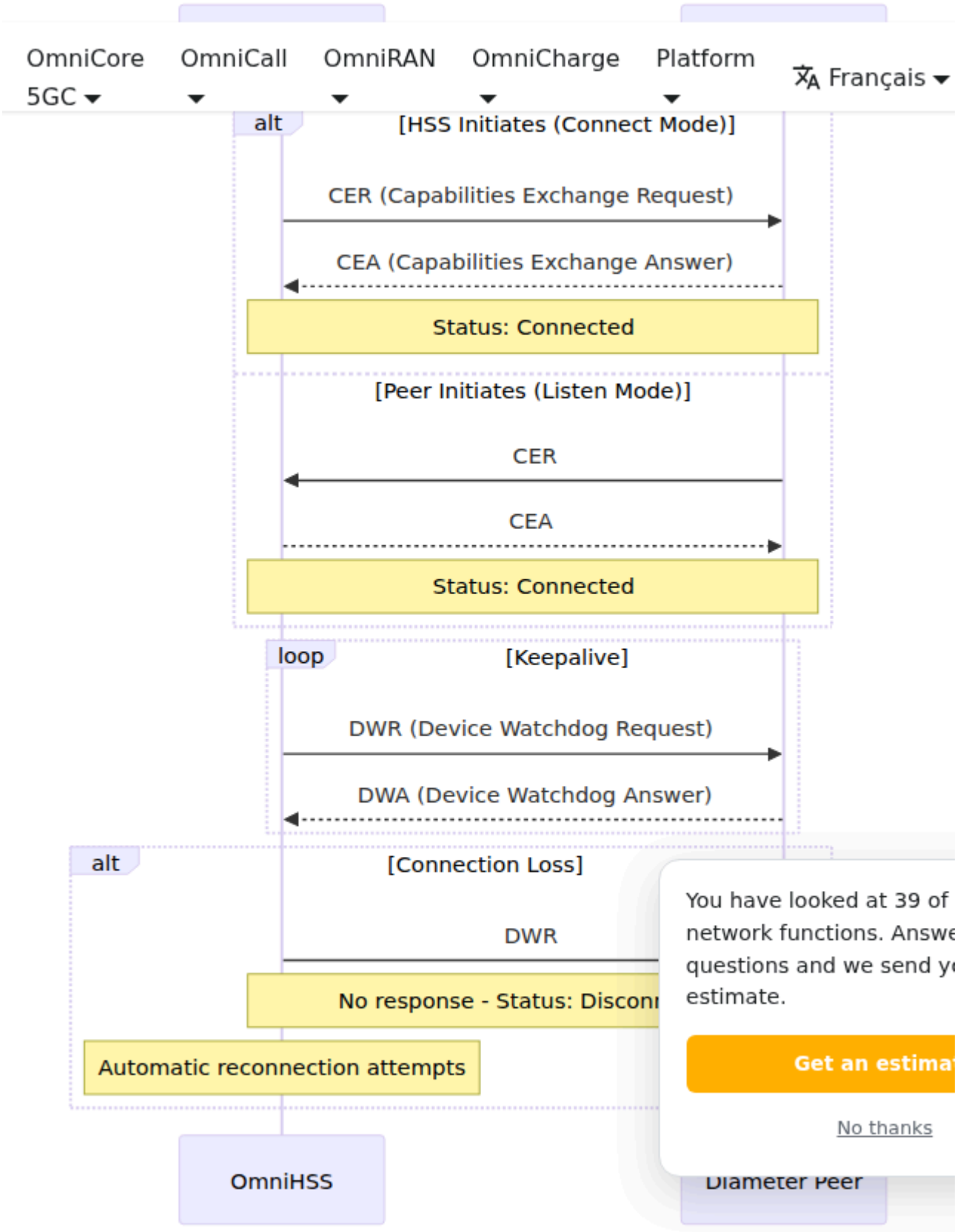
2. **Pendant le Shift**

   - Gardez la page d'Aperçu ouverte pour une surveillance en temps réel
   - Surveillez les changements d'état inhabituels
   - Surveillez les pairs déconnectés sur la page Diameter

3. **Fin de Shift**

   - Vérifiez que le système est stable
   - Vérifiez la page Application pour les tendances d'utilisation des ressources
   - Documentez toute anomalie

Flux de Travail de Dépannage



Seuils d'Alerte

Établissez des seuils de surveillance pour des alertes proactives :

Métrique	Avertissement	Critique
Pairs Diameter Déconnectés	1 pair	2+ pairs ou pair critique
Utilisation de la Mémoire	> 80%	> 90%
Échecs d'Authentification des Abonnés	> 5%	> 10%
Compte de Processus	> 80% de la limite	> 95% de la limite

← Retour au Guide des Opérations | Suivant : Métriques & Surveillance →

Réplication de Base de Données

← Retour au Guide des Opérations

Table des Matières

- Aperçu
- Comment Ça Fonctionne
- Architectures de Déploiement
- Référence de Configuration
- Gestion des Pairs
- Surveillance
- Dépannage

Aperçu

OmniHSS utilise **PostgreSQL** avec **réplication logique bidirectionnelle** pour des déploiements multi-sites à haute disponibilité. Chaque site fonctionne comme un primaire indépendant en lecture/écriture. Il n'y a pas de maître unique ou de répliques en lecture seule.

Caractéristiques Clés

- **Les Deux Sites en Lecture/Écriture** : Chaque site dispose d'une base de données primaire complète. Les écritures locales sont instantanées, quelle que soit la qualité du lien inter-site.
- **Réplication Asynchrone** : Les modifications se répliquent via le Write-Ahead Log (WAL) de PostgreSQL. Aucune écriture n'est bloquée en attendant qu'un site distant accuse réception.

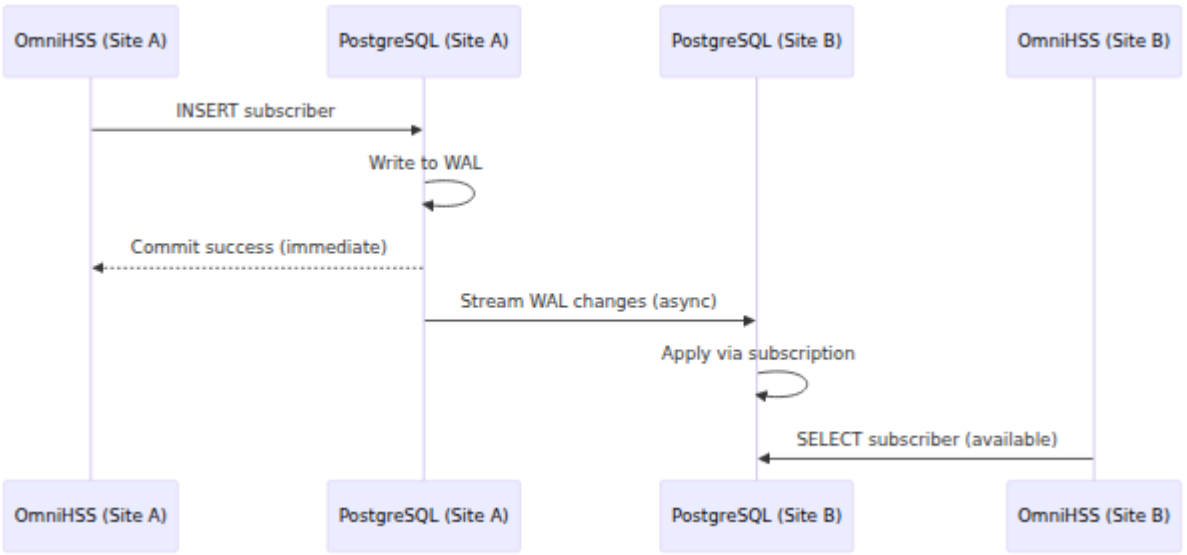
- **Tolérant aux Partitions** : Si le lien inter-site tombe, les deux sites continuent à fonctionner indépendamment. Les segments WAL s'accumulent localement et se rejouent automatiquement lorsque le lien se rétablit.
- **Pas de Quorum Requis** : Contrairement aux approches de cluster synchrones, il n'y a pas de concept de quorum. Un déploiement sur un seul site fonctionne de la même manière qu'un déploiement multi-sites.
- **Clés Primaires UUID** : Tous les enregistrements utilisent des clés primaires UUIDv4, éliminant les collisions d'ID entre les sites sans coordination.

Comment Cela Diffère de la Réplication Synchrone

	Synchrone	Réplication Logique Asynchrone
Latence d'écriture	Bloquée par le nœud le plus lent	Local uniquement : sous-millisecondes
Échec du lien	Le site minoritaire devient en lecture seule	Les deux sites continuent en lecture/écriture
Quorum nécessaire	Oui (majorité des nœuds)	Non
Cohérence des données	Forte (tous les nœuds identiques)	Éventuelle (brève latence de réplication)
Adapté aux liens satellites	Non (la latence nuit aux performances d'écriture)	Oui (les files d'attente WAL rattrapent)

Comment Ça Fonctionne

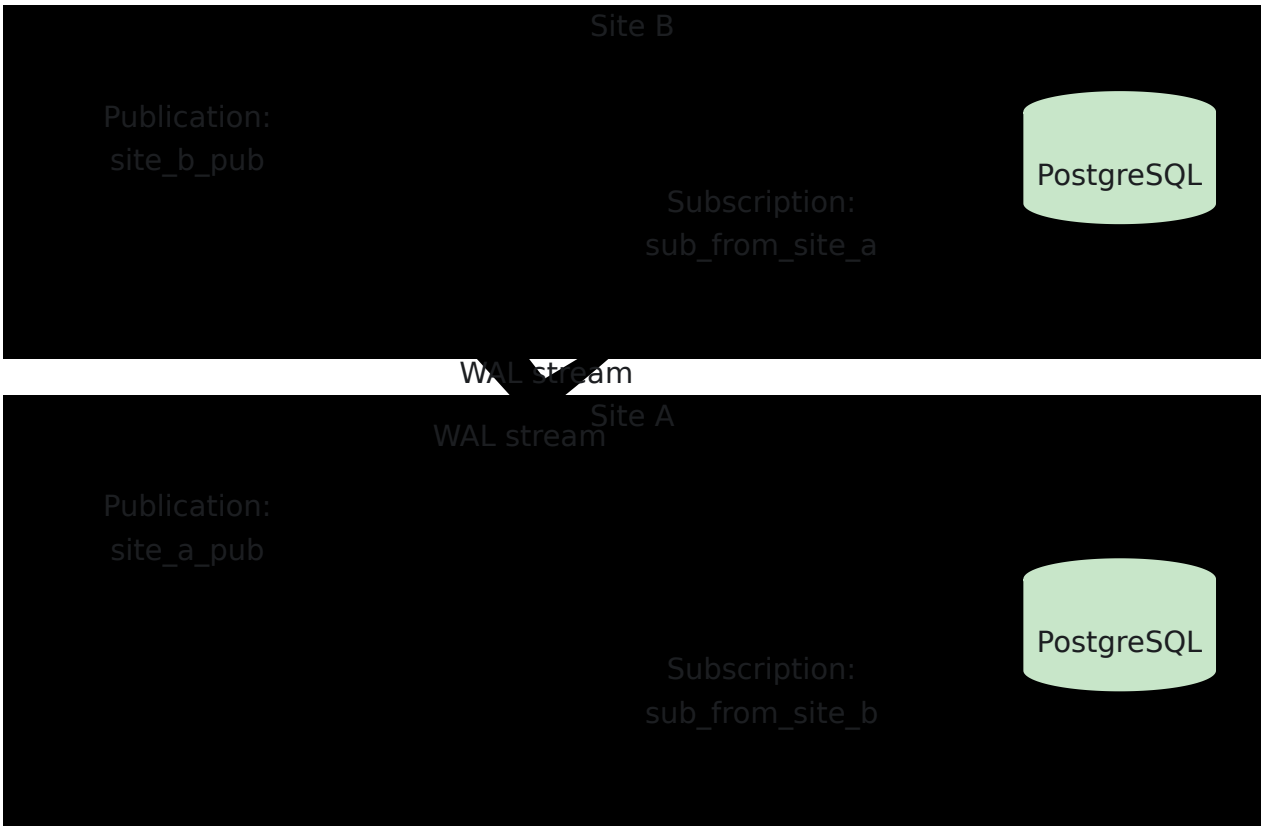
Flux de Réplication Logique



Modèle de Publication / Abonnement

Chaque site a :

- Une **publication** qui diffuse tous les changements de table (sauf `schema_migrations`)
- Un **abonnement** à la publication de chaque site pair



Mise en File d'Attente WAL Pendant une Panne de Lien

Lorsque le lien inter-site n'est pas disponible :

1. Les écritures locales continuent sans interruption
2. Les segments WAL s'accumulent du côté de la publication
3. Le slot de réplication suit où chaque abonné s'est arrêté
4. Lorsque le lien se rétablit, le WAL en file d'attente se rejoue depuis la dernière position reconnue
5. Les deux sites convergent vers un état identique

La conservation du WAL est limitée par `max_slot_wal_keep_size` pour éviter une croissance illimitée du disque pendant des pannes prolongées.

Prévention des Conflits

Les clés primaires UUID éliminent la principale source de conflits dans la réplication bidirectionnelle. Étant donné que chaque site génère des

identifiants uniques globalement de manière indépendante, les opérations INSERT ne se chevauchent jamais.

L'option d'abonnement `origin = none` empêche les boucles de réplication. Les modifications arrivées par réplication ne sont pas republées à d'autres abonnés.

La table `schema_migrations` est exclue des publications car les deux sites exécutent des migrations identiques de manière indépendante.

Last-Write-Wins pour l'État Dynamique

Les tables d'état dynamique (`subscriber_state`, `pdn_session`, `lte_call`) sont mises à jour fréquemment par le signalement Diameter. Chaque ULR met à jour le MME de service, chaque CCR met à jour la session PGW, et chaque AAR met à jour l'appel actif.

Dans un déploiement multi-sites, un abonné est servi par un site à la fois. Lorsque l'abonné se déplace entre les sites, les deux sites peuvent avoir des mises à jour de la même ligne `subscriber_state` à des moments différents. Sans résolution de conflit, l'ordre de réplication déterminerait quelle valeur survit. Cela pourrait écraser le MME de service actuel avec une valeur obsolète.

OmniHSS utilise des **triggers Last-Write-Wins (LWW)** sur les tables d'état dynamique :

```
CREATE TRIGGER lww_subscriber_state_trigger
  BEFORE UPDATE ON subscriber_state
  FOR EACH ROW
  EXECUTE FUNCTION lww_subscriber_state();
```

Le trigger compare le timestamp `updated_at` de la mise à jour répliquée entrante avec la ligne existante. Si la ligne existante est plus récente, la mise à jour est silencieusement supprimée. Cela garantit :

- Lorsque un abonné passe du Site A au Site B, la mise à jour la plus récente du Site B l'emporte toujours

- Les mises à jour obsolètes du Site A (avant le déplacement de l'abonné) sont rejetées
- Les deux sites convergent vers la même valeur : la plus récente écriture

Tables avec des triggers LWW :

Table	Mis à Jour Par	Fréquence
`subscriber_state`	ULR, SAR, AIR, PUR, ECR, CCR	Chaque mise à jour d'attache/localisation
`pdn_session`	CCR-I, CCR-T	Chaque création/suppression de session PDN
`lte_call`	AAR, STR	Chaque mise en place/fin d'appel VoLTE
`ims_subscription`	UAR, SAR (S-CSCF partagé)	Chaque enregistrement d'un membre groupé

Les données de provisionnement statiques (abonnés, profils, clés, APNs) n'ont pas besoin de LWW. L'API de provisionnement ne met à jour ces enregistrements que, typiquement, depuis un seul plan de gestion.

Convention de Nommage (Requise pour les Nouvelles Tables)

Les noms de fonction et de trigger LWW sont un **contrat**, pas un choix libre. Pour une table `foo`, la fonction doit être nommée `lww_foo()` et le trigger doit être nommé `lww_foo_trigger`.

Le rôle Ansible de réplication `omnihss` d'OmniCore **ne** conserve pas sa propre liste de ces tables. Il découvre chaque trigger dont le nom correspond à `lww_<table>_trigger` et s'assure que chacun est en mode `ENABLE REPLICA` à travers le maillage. Un trigger nommé autrement est invisible à cette réconciliation, donc la résolution de conflit inter-sites ne serait pas appliquée silencieusement sur cette table.

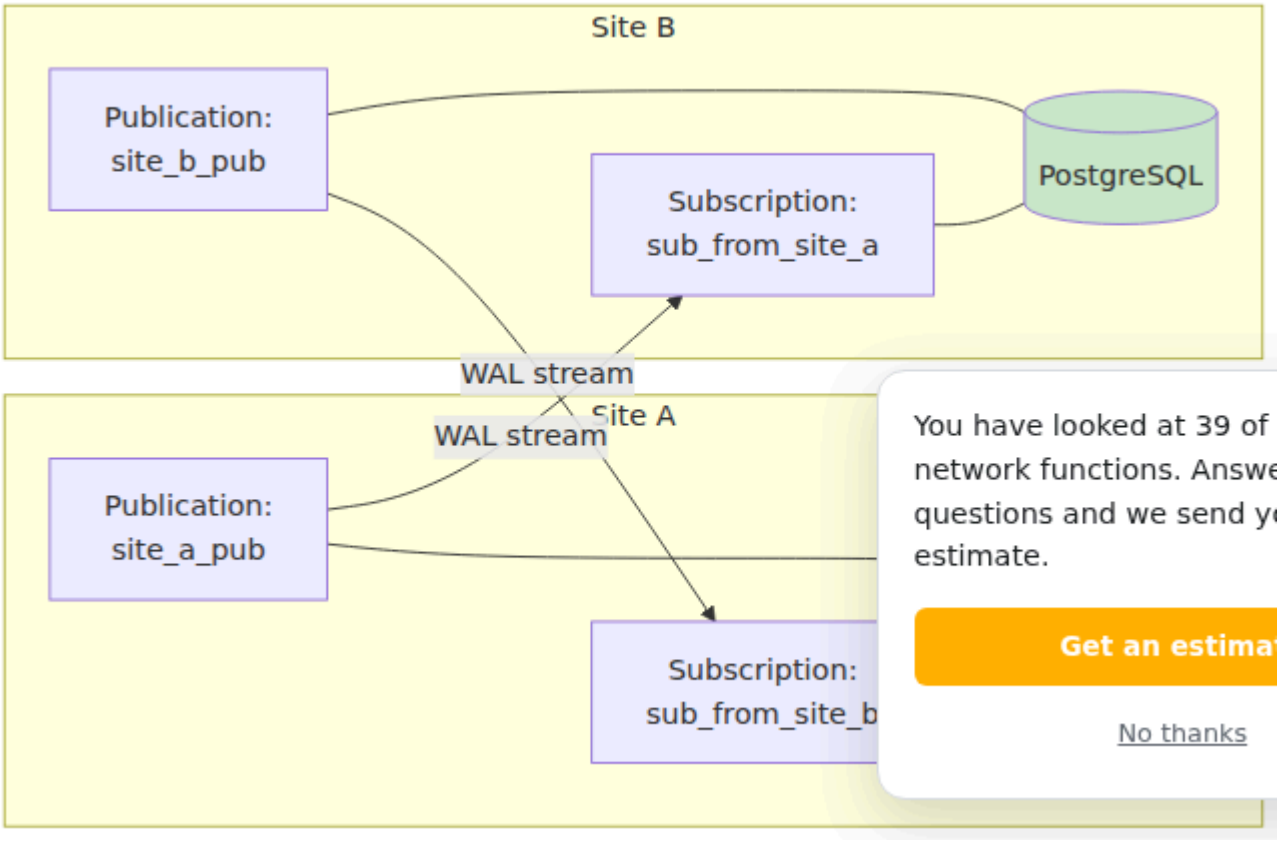
Lorsque vous ajoutez une nouvelle table d'état dynamique (une table écrite à partir du trafic Diameter en direct sur plus d'un site à la fois), suivez ces étapes :

1. Donnez à la table une colonne `updated_at` (le macro `timestamps()` d'Ecto l'ajoute).
2. Dans la même migration, créez `lww_<table>()` et `lww_<table>_trigger` avec exactement ces noms.
3. Dans la même migration, exécutez `ALTER TABLE <table> ENABLE REPLICA TRIGGER lww_<table>_trigger`, afin que le trigger se déclenche lors de l'application de la réplication, et non lors des écritures locales.

La table `ims_subscription` est l'exemple de référence : sa migration crée `lww_ims_subscription()` et `lww_ims_subscription_trigger`, puis active le trigger en mode réplique. Suivez cette même structure, et ne renommez pas la fonction ou le trigger.

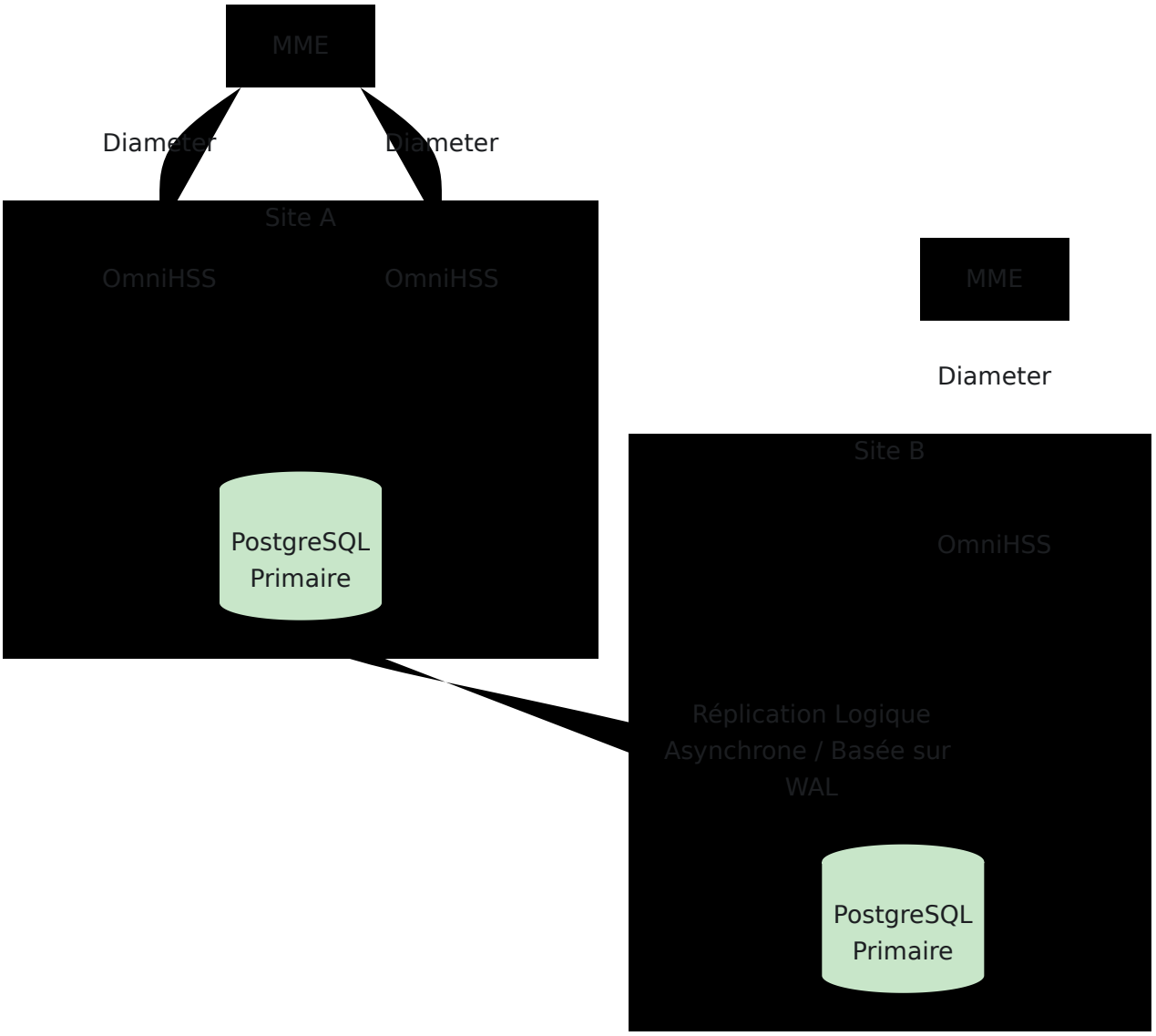
Architectures de Déploiement

Site Unique (Pas de Réplication)



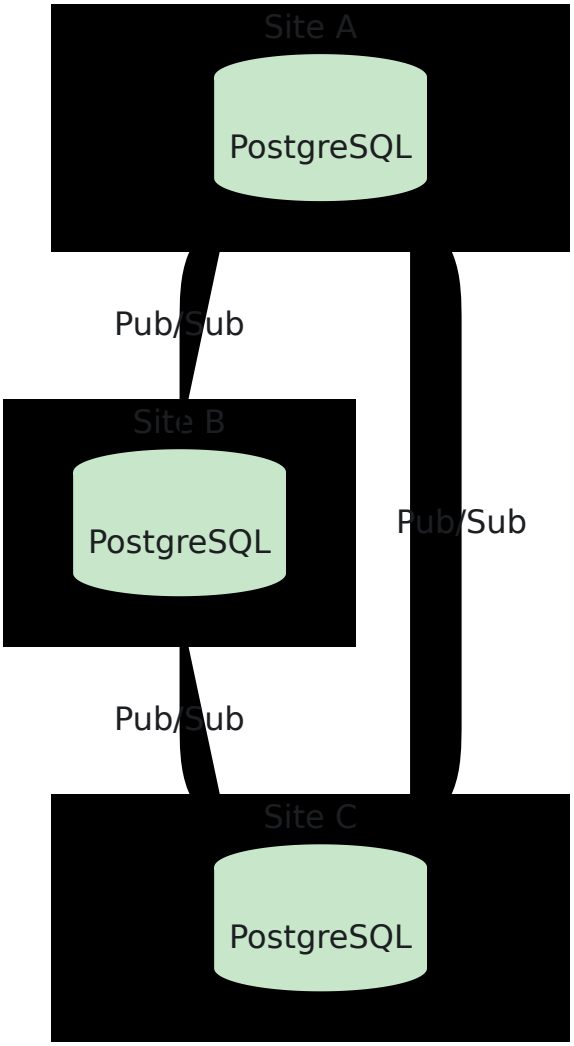
Plusieurs instances d'application HSS partagent une seule base de données PostgreSQL. La réplication de streaming PostgreSQL standard peut être utilisée pour un basculement local de la base de données si nécessaire.

Déploiement à Deux Sites



Les deux sites sont des primaires complets. Le provisionnement des abonnés sur l'un ou l'autre site se réplique à l'autre. Chaque MME se connecte à son HSS local. Il n'y a pas de dépendance Diameter inter-sites.

Déploiement Multi-Sites



Chaque site s'abonne à la publication de chaque autre site. Les pairs sont auto-découverts à partir de l'inventaire Ansible. Pour ajouter un site, ajoutez-le au groupe d'inventaire `hss` et exécutez le playbook.

Exigences R��seau

Port	Protocole	Objectif
5432	TCP	Connexions client PostgreSQL et de réplication

Référence de Configuration

Variables Ansible

La réplication est configurée par groupe HSS dans votre inventaire :

```yaml
hss:
  hosts:
    site-a-hss01:
      ansible_host: 10.80.12.140
    site-a-hss02:
      ansible_host: 10.80.12.141
  vars:
    omnihss:
      database_type: postgres
      database_host: localhost
      database_port: 5432
      database_name: omnihss
      database_username: omnihss
      database_password: "secure_password"
      replication:
        enabled: true
```

Paramètres Ansible

Paramètre	Type	Requis	Par Défaut	Description
`database_type`	Chaîne	Non	`postgres`	Backend de base de données. Doit être `postgres` pour la réplication.
`database_host`	Chaîne	Non	`localhost`	Hôte PostgreSQL. Utilisez `localhost` lorsque Postgres s'exécute sur le même hôte qu'OmniHSS.
`database_port`	Entier	Non	`5432`	Port PostgreSQL.
`database_name`	Chaîne	Non	`omnihss`	Nom de la base de données.
`database_username`	Chaîne	Oui	-	Utilisateur PostgreSQL. Doit avoir les privilèges `CREATEDB` et `REPLICATION`.
`database_password`	Chaîne	Oui	-	Mot de passe PostgreSQL.
`replication.enabled`	Booléen	Non	`false`	Activer la réplication logique

Paramètre	Type	Requis	Par Défaut	Description
				bidirectionnelle. Lorsque `true`, le rôle Ansible configure automatiquemen les publications, les abonnements et les entrées `pg_hba.conf`.

Découverte des Pairs

Les pairs sont découverts automatiquement à partir de deux sources :

1. **Groupe d'inventaire local** : Tous les autres hôtes dans le groupe Ansible `hss`
2. **Sites distants** : Entrées dans `connected_sites.hss` à partir de `prod_master_peers.yml`

Aucune configuration de pair par hôte n'est requise. Ajouter un hôte au groupe `hss` ou à `connected_sites.hss` et exécuter le playbook est suffisant.

Pairs de Sites Distants (prod_master_peers.yml)

Pour la réplication inter-site à travers des emplacements géographiques :

```yaml
# group_vars/prod_master_peers.yml
connected_sites:
  hss:
    - remote-site-hss01: 10.80.20.140
    - remote-site-hss02: 10.80.20.141
```

Configuration PostgreSQL

Le rôle Ansible configure automatiquement les paramètres PostgreSQL suivants lorsque la réplication est activée :

Paramètre	Valeur	Objectif
`wal_level`	`logical`	Requis pour la réplication logique. Inclut les données compl��tes des lignes dans le WAL.
`max_wal_senders`	`10`	Maximum de processus d'envoi WAL concurrents (un par abonné).
`max_replication_slots`	`10`	Maximum de slots de réplication. Chaque pair utilise un slot.
`listen_addresses`	`*`	Accepter les connexions des sites pairs.

Le rôle ajoute également des entrées `pg_hba.conf` pour permettre les connexions de réplication depuis chaque IP de pair découverte.

Gestion des Pairs

Ajouter un Nouveau Site

1. Ajoutez le(s) nouvel(s) hôte(s) au groupe d'inventaire `hss`, ou ajoutez-le à `connected_sites.hss` dans `prod_master_peers.yml`
2. Exécutez le playbook OmniHSS

Le rôle va :

- Installer PostgreSQL et OmniHSS sur le nouvel hôte
- Exécuter les migrations de base de données

- Créer une publication pour le nouveau site
- Créer des abonnements à tous les pairs existants
- Mettre à jour `pg_hba.conf` sur tous les pairs existants pour permettre le nouveau site
- Créer des abonnements des pairs existants au nouveau site

Supprimer un Site

```
ansible-playbook -i hosts site.yml --tags hss-remove-peer \
  -e "peer_name=site_b confirm_remove=yes"
```

Le paramètre `confirm_remove=yes` est requis comme mesure de sécurité. Sans cela, le playbook refuse de continuer et affiche ce qui se passerait.

Supprimer un pair :

- Supprime l'abonnement (arrête la réplication entrante de ce pair)
- Supprime l'IP du pair de `pg_hba.conf`
- Les données déjà répliquées depuis le pair restent dans la base de données locale
- L'abonnement du site pair à ce site doit être supprimé séparément sur le pair

Ajouter un Pair Ad-Hoc

Pour les cas où vous devez ajouter un pair en dehors du flux standard du playbook :

```
ansible-playbook -i hosts site.yml --tags hss-add-peer \
  -e "peer_name=site_b peer_host=10.80.20.140"
```

Surveillance

Métriques Prometheus

Le service **postgres_exporter** s'exécute sur chaque hôte HSS sur le port `9187`, exposant les métriques PostgreSQL à Prometheus.

Métriques de Réplication

Métrique	Type	Description
`pg_replication_slots_active`	Gauge	`1` si le slot de réplication est actif (pair connecté), `0` si inactif
`pg_replication_slots_pg_wal_lsn_diff`	Gauge	Latence de réplication en octets. Distance entre la position WAL actuelle et la dernière position confirmée du pair
`pg_stat_replication_pg_current_wal_lsn_bytes`	Gauge	Position WAL actuelle sur ce nœud

Exemples de Requêtes Prometheus

```
# Slot de réplication actif (1 = connecté, 0 = déconnecté)
pg_replication_slots_active{slot_name=~"sub_from_.*"}

# Latence de réplication en mégaoctets
pg_replication_slots_pg_wal_lsn_diff / 1048576

# Alerte : slot de réplication inactif depuis plus de 60 secondes
pg_replication_slots_active == 0
```

Alertes Recommandées

Alerte	Condition	Gravité	De
Slot de Réplication Inactif	`pg_replication_slots_active == 0` pendant 60s	Critique	Pa dé Le s'a
Latence WAL Excessive	`pg_replication_slots_pg_wal_lsn_diff > 104857600` pendant 5m	Avertissement	La de ré dé Mo pe ler lie dé
Slot WAL Perdu	L'état WAL est `lost`	Critique	Le inv pa né re- co

Validation Ansible

Le playbook OmniHSS valide la santé de la réplication à la fin de chaque exécution. Il vérifie :

1. Tous les travailleurs d'abonnement sont vivants
2. Tous les slots de réplication sont actifs
3. La latence WAL est dans le seuil (100 Mo par défaut)
4. L'état du slot WAL est `reserved` ou `extended` (pas `lost`)

Si une assertion échoue, le playbook échoue avec un message d'erreur clair identifiant le problème.

Validation Manuelle

```
# Vérifiez que les travailleurs d'abonnement sont vivants
sudo -u postgres psql -d omnihss -c \
  "SELECT subname, pid IS NOT NULL as worker_alive FROM
pg_stat_subscription;"

# Vérifiez les slots de réplication et la latence
sudo -u postgres psql -d omnihss -c \
  "SELECT slot_name, active,
    pg_wal_lsn_diff(pg_current_wal_lsn(), confirmed_flush_lsn) as
lag_bytes,
    wal_status
  FROM pg_replication_slots WHERE slot_type = 'logical';"

# Comparez les comptes d'abonnés entre les sites
sudo -u postgres psql -d omnihss -c "SELECT count(*) FROM
subscriber;"
```

Dépannage

Travailleur d'Abonnement Mort

Symptômes : `pg_stat_subscription` montre `pid IS NULL` pour un abonnement. La validation Ansible échoue avec "Travailleurs morts."

Causes Courantes :

- La table existe sur le publieur mais pas sur l'abonné (incompatibilité DDL)
- Conflit de clé dupliquée d'un enregistrement inséré manuellement
- Pair inaccessible pendant une période prolongée entraînant l'invalidation du slot WAL

Résolution :

1. Vérifiez les journaux PostgreSQL pour l'erreur spécifique :
   ```
   journalctl -u postgresql | grep "logical replication" | tail -20
   ```
2. Si incompatibilité DDL : appliquez la migration manquante sur ce site, puis le travailleur redémarrera automatiquement
3. Si clé dupliquée : résolvez le conflit manuellement, puis réactivez l'abonnement :
   ```
   ALTER SUBSCRIPTION sub_from_<peer> ENABLE;
   ```

Slot de Réplication Inactif

Symptômes : `pg_replication_slots` montre `active = false`. L'alerte Prometheus se déclenche.

Causes Courantes :

- Problème de connectivité réseau entre les sites
- PostgreSQL sur le site pair est arrêté

- Pare-feu bloquant le port 5432 depuis l'IP du pair

Résolution :

1. Vérifiez la connectivité réseau avec le pair :
   ```
   pg_isready -h <peer_ip> -U <db_user> -d omnihss
   ```
2. Vérifiez que PostgreSQL du site pair est en cours d'exécution
3. Vérifiez que les règles de pare-feu autorisent le port 5432 depuis l'IP de ce site

Le travailleur d'abonnement se reconnectera automatiquement lorsque le pair redeviendra accessible.

Slot WAL Perdu

Symptômes : `pg_replication_slots` montre `wal_status = 'lost'`. La validation Ansible échoue avec "WAL SLOT CRITIQUE."

Cause : Le pair a été déconnecté si longtemps que le WAL conservé a dépassé `max_slot_wal_keep_size` et a été nettoyé.

Résolution : Le pair nécessite un re-seeding complet :

1. Supprimez l'abonnement cassé sur ce site :
   ```
   DROP SUBSCRIPTION sub_from_<peer>;
   ```
2. Supprimez le slot cassé sur le site pair :
   ```
   SELECT pg_drop_replication_slot('sub_from_<this_site>');
   ```
3. Réexécutez le playbook OmniHSS. Il recrée l'abonnement avec `copy_data = true` pour effectuer une synchronisation initiale.

Latence WAL Excessive

Symptômes : `pg_replication_slots_pg_wal_lsn_diff` est grand et en croissance. L'alerte Prometheus se déclenche.

Causes Courantes :

- Lien inter-site à haute latence ou congestionné
- Site pair sous forte charge (le travailleur d'application ne peut pas suivre)
- Grande opération en bloc (par exemple, provisionnement massif) générant un volume WAL significatif

Résolution :

1. Vérifiez si la latence est en croissance ou stable :

```
SELECT slot_name,
  pg_wal_lsn_diff(pg_current_wal_lsn(), confirmed_flush_lsn) as
lag_bytes
FROM pg_replication_slots WHERE slot_type = 'logical';
```

2. Si stable : le pair applique des changements mais est en retard. Il rattrapera.
3. Si en croissance : enquêtez sur le lien réseau ou la charge du site pair.
4. Pour les opérations en bloc : la latence pendant l'opération est attendue et se résoudra après la fin de l'opération.

Coordination des Migrations de Schéma

Important : Les changements de schéma de base de données (migrations) ne se répliquent pas via la réplication logique. Lors du déploiement d'une nouvelle version d'OmniHSS qui inclut des changements de schéma :

1. Appliquez la migration sur **tous les sites** avant de déployer le code de l'application qui dépend de nouvelles colonnes/tables
2. Le playbook Ansible gère cela automatiquement. Il exécute des migrations sur chaque hôte

3. Si vous déployez manuellement, exécutez les migrations sur tous les sites d'abord :

```
/opt/omnihss/bin/hss rpc "Hss.Command.Database.migrate()"
```

Relations d'Entités OmniHSS

← Retour au Guide des Opérations

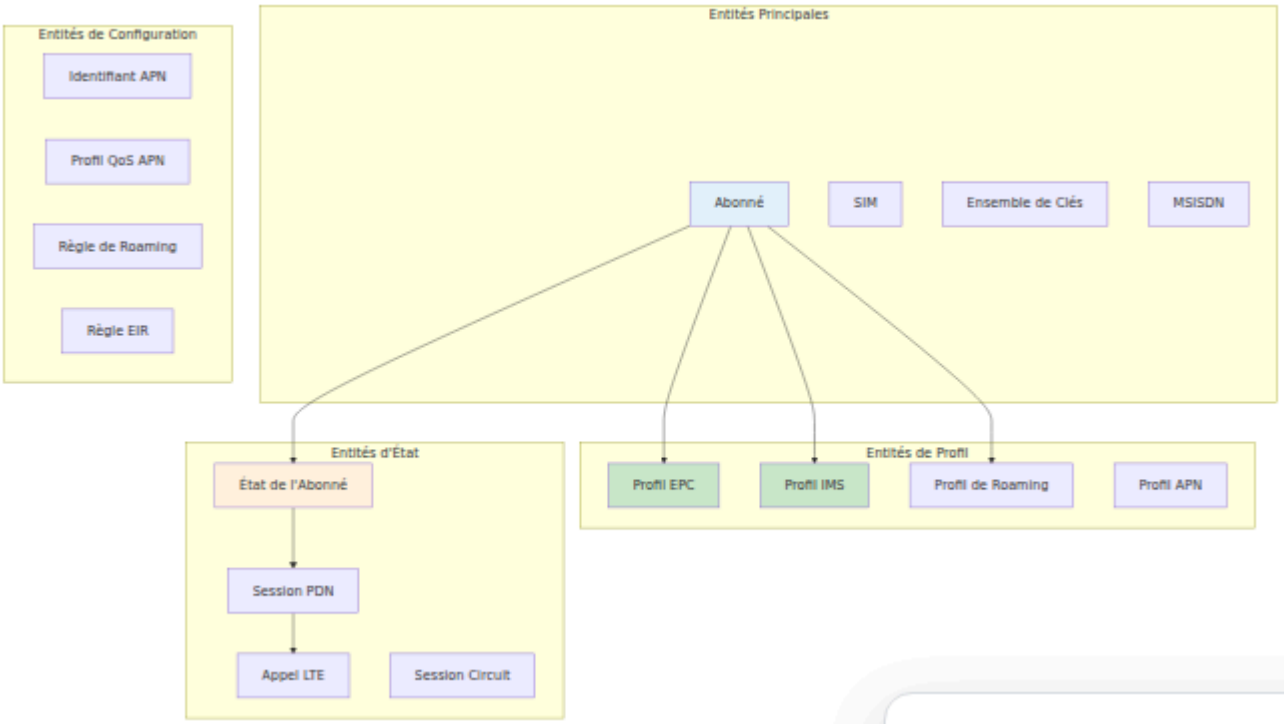
Table des Matières

- Aperçu des Entités
- Entités Principales
- Entités de Profil
- Entités d'État
- Diagrammes de Relations d'Entités
- Cycle de Vie des Entités
- Modèles de Flux de Données

Aperçu des Entités

OmniHSS organise les données des abonnés en entités logiques avec des relations claires. Comprendre ces entités est crucial pour des tâches opérationnelles telles que la provisionnement, le dépannage et la planification de capacité.

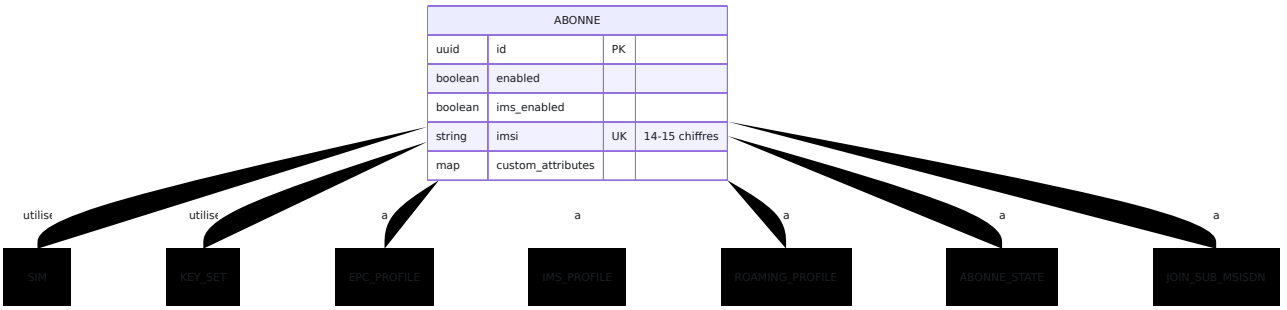
Catégories d'Entités



Entités Principales

Abonné

L'**Abonné** est l'entité centrale représentant un utilisateur mobile.



Champs :

Champ	Type	Description	Contraintes
`id`	uuid	Clé primaire	UUID, généré (`gen_random_uuid()`)
`enabled`	boolean	Indicateur de service activé	Par défaut : true
`ims_enabled`	boolean	Services IMS activés	Par défaut : true
`imsi`	string	Identité Internationale de l'Abonné Mobile	14-15 chiffres, unique
`custom_attributes`	map	Données personnalisées clé-valeur	Optionnel
`sim_id`	uuid	Clé étrangère vers SIM	Requis pour l'authentification (peut être nul pendant l'inactivité)
`key_set_id`	uuid	Clé étrangère vers Ensemble de Clés	Requis
`epc_profile_id`	uuid	Clé étrangère vers Profil EPC	Requis pour le service de données (peut être nul pendant l'inactivité)
`ims_profile_id`	uuid	Clé étrangère vers Profil IMS	Requis pour IMS/voix (peut être nul pendant l'inactivité)

Champ	Type	Description	Contraintes
`roaming_profile_id`	uuid	Clé étrangère vers Profil de Roaming	Optionnel ; si nul, l'abonné ne peut pas roaming (réseau domestique uniquement)
`subscriber_state_id`	uuid	Clé étrangère vers État de l'Abonné	Créé automatiquement
`ims_subscription_id`	uuid	Clé étrangère vers Abonnement IMS	Optionnel

La colonne "Contraintes" ci-dessus indique ce dont chaque objet est requis **pour un service**, et non ce dont la base de données a besoin pour insérer une ligne. La base de données accepte un abonné qui n'a qu'un IMSI et un Ensemble de Clés. Un tel abonné est *inactif* : il existe, mais aucun service ne fonctionne tant que vous n'assignez pas l'objet correspondant. Utilisez cela pour pré-provisionner des abonnés, puis activez-les plus tard. Voir Pré-Provisionnement d'Abonnés Inactifs.

Ce que chaque objet permet :

Pour faire fonctionner cela	L'abonné a besoin de
Authentification (attacher, authentification IMS)	`key_set` et `sim`
Service de données LTE (attachement S6a)	`epc_profile`
IMS / voix (enregistrement Cx)	`ims_profile`, et au moins un MSISDN
Roaming sur un réseau visité	un `roaming_profile` avec une règle correspondante

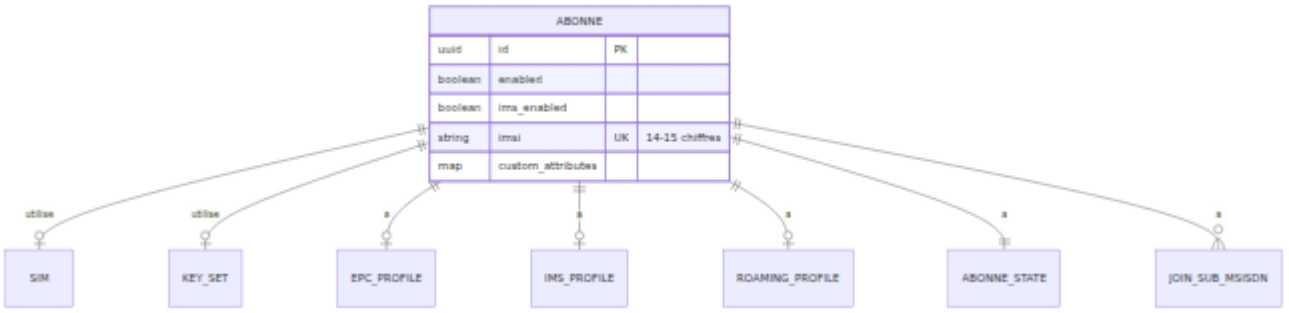
Si un objet requis est manquant lorsque l'abonné apparaît sur le réseau, le HSS rejette la demande et enregistre la raison exacte. Voir Comportement des Abonnés Inactifs sur le Réseau.

Points Clés :

- Chaque abonné doit avoir exactement un IMSI.
- L'IMSI doit comporter 14-15 chiffres (pas de lettres ni de caractères spéciaux).
- Un abonné peut avoir plusieurs MSISDN (numéros de téléphone).
- L'état de l'abonné est créé automatiquement lors de la création de l'abonné.
- Le drapeau `enabled` contrôle tous les services (données et IMS).
- Le drapeau `ims_enabled` contrôle uniquement les services IMS.
- La base de données a seulement besoin de l'IMSI et de l'Ensemble de Clés pour insérer un abonné. Chaque autre objet est nécessaire pour son propre service, comme indiqué ci-dessus. Un abonné sans profil IMS ne peut pas utiliser l'IMS, même si l'enregistrement est sauvegardé sans.
- Un abonné sans `ims_subscription_id` est autonome. Définissez `ims_subscription_id` pour regrouper plusieurs abonnés (IMSIs) dans un seul abonnement IMS qui partage une identité publique et un seul S-CSCF. Voir Abonnement IMS et Identité Publique Utilisateur Partagée.

SIM

L'entité **SIM** représente une carte SIM physique ou intégrée.



Champs :

Champ	Type	Description	Niveau de Sécurité
`iccid`	string	Identifiant de la Carte Circuits Intégrée	Public
`sim_vendor`	string	Fabricant de la SIM	Public
`batch_name`	string	Lot de fabrication	Public
`is_esim`	boolean	Indicateur de SIM intégrée	Public
`pin1`, `pin2`	string	Codes PIN	Sensible
`puk1`, `puk2`	string	Codes PUK	Sensible
`adm1` - `adm10`	string	Codes administratifs	Très Sensible
`kic`, `kid`	binary	Clés de sécurité OTA	Très Sensible

Points Clés :

- L'ICCID identifie de manière unique la carte SIM
- Une SIM peut être assignée à un abonné à la fois

- Les codes PIN/PUK sont pour le verrouillage de la SIM par l'utilisateur final
- Les codes ADM sont pour les opérations administratives de la SIM
- KIC/KID sont pour les mises à jour OTA (Over-The-Air) de la SIM

Ensemble de Clés

L'**Ensemble de Clés** contient des clés cryptographiques pour l'authentification.

KEY_SET			
uuid	id	PK	
binary	ki		128-bit
binary	opc		128-bit
binary	op		128-bit
binary	amf		16-bit
bigint	sqn		48-bit sequence
string	authentication_algorithm		

utilisé pa



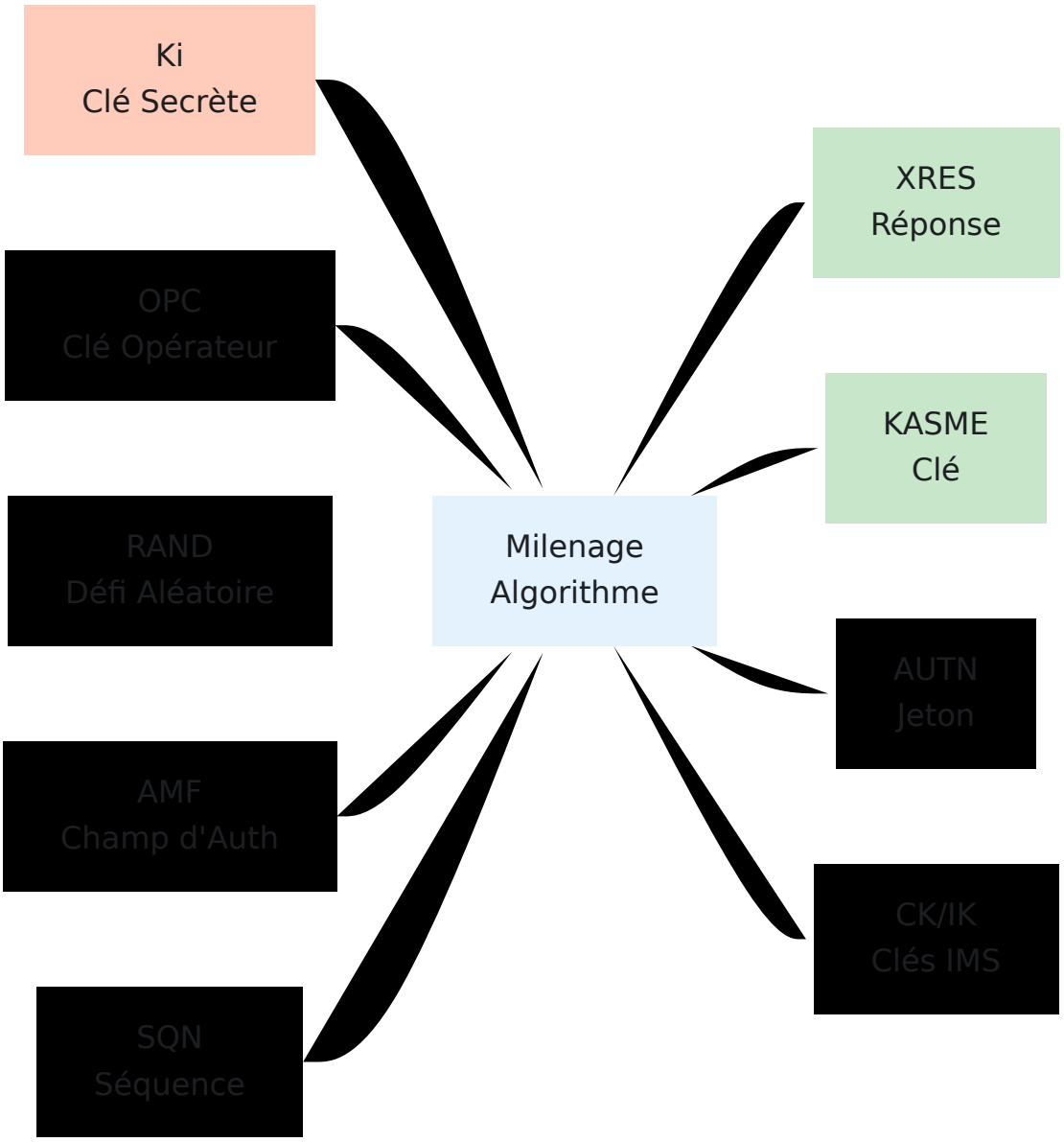
Champs :

Champ	Type	Description	Taille
`ki`	binary	Clé secrète	128 bits (16 octets)
`opc`	binary	Clé variante opérateur (dérivée)	128 bits
`op`	binary	Clé opérateur (pour dériver OPC)	128 bits
`amf`	binary	Champ de Gestion d'Authentification	16 bits (2 octets)
`sqn`	bigint	Numéro de séquence (anti-replay)	48 bits
`authentication_algorithm`	string	Nom de l'algorithme	Actuellement "milenage"
`ota_counter`	bigint	Compteur d'opération OTA	Entier

Points Clés :

- Plusieurs abonnés peuvent partager le même ensemble de clés
- Ki est le secret maître partagé avec la SIM
- Soit OPC soit OP doit être fourni (OPC peut être dérivé de OP)
- SQN est incrémenté à chaque authentification
- Milenage est actuellement le seul algorithme supporté

Algorithme d'Authentification :



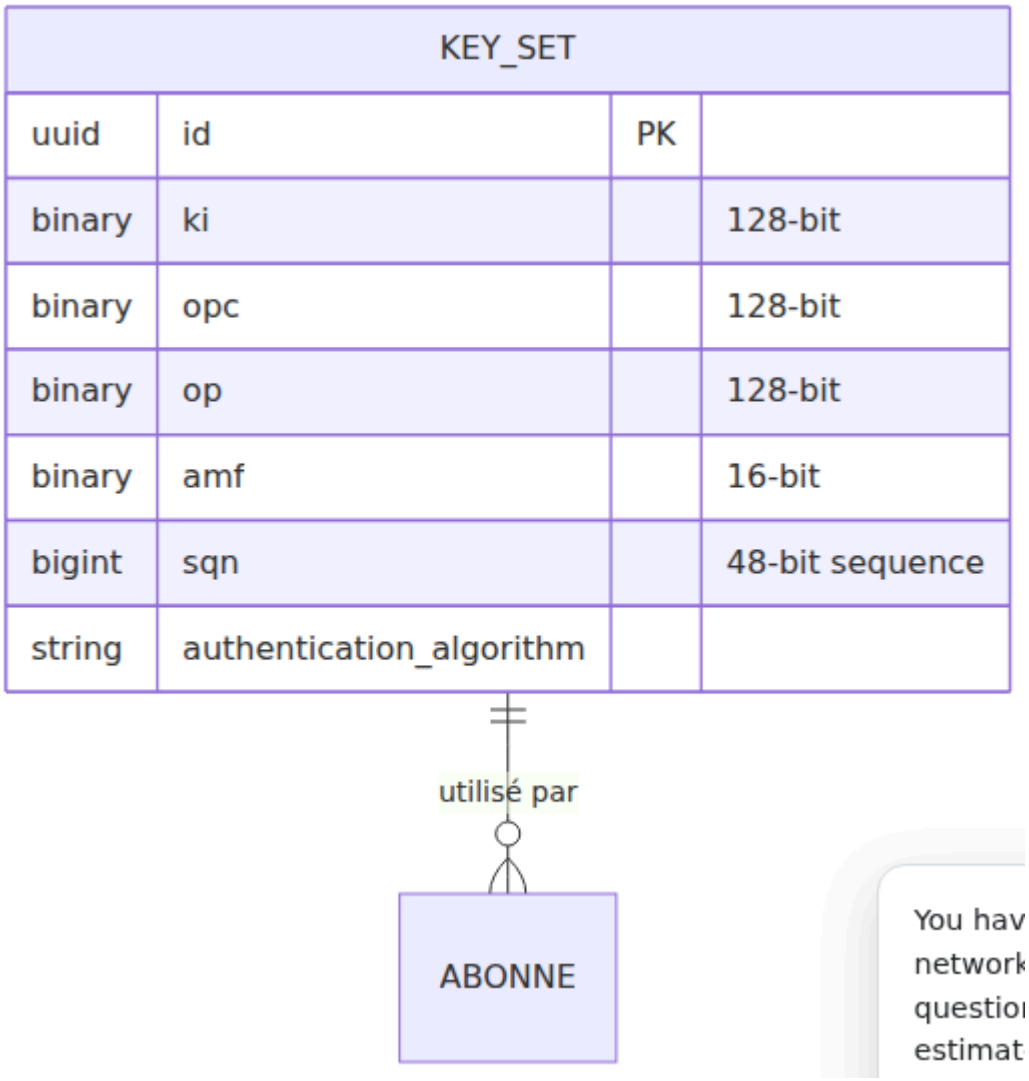
Ensemble de Clés, SIM et Abonné

La SIM, l'Ensemble de Clés et l'Abonné sont des objets séparés. Chaque objet a son propre cycle de vie. L'Abonné est l'enregistrement qui les relie.

- **SIM** - la carte physique ou l'eSIM. Elle contient l'ICCID et les codes PIN, PUK et ADM. C'est l'inventaire.
- **Ensemble de Clés** - les clés d'authentification (Ki, OPc, AMF, SQN). Le réseau s'authentifie contre ces clés.
- **Abonné** - l'identité réseau et le service (IMSI, MSISDN, profils, état activé).

Un Abonné référence une SIM (`sim_id`) et un Ensemble de Clés (`key_set_id`). Un Ensemble de Clés ne référence pas une SIM. Pour trouver à quelle SIM

appartient un Ensemble de Clés, lisez l'Abonné qui référence les deux.



Pourquoi les objets sont séparés

Les objets sont séparés car ils ne se mappent pas un à un dans tous les cas :

- **Changement de SIM.** Un abonné perd un téléphone et a besoin d'une nouvelle SIM. Vous conservez l'Abonné, avec son MSISDN, ses profils et ses données IMS. Vous le pointez vers une nouvelle SIM et un nouvel Ensemble de Clés. Le service continue.
- **SIM Multi-IMSI.** Un ICCID porte plusieurs IMSIs. Chaque IMSI est un Abonné séparé. Certains IMSIs partagent un Ensemble de Clés. Certains IMSIs ont leur propre Ensemble de Clés. Les clés appartiennent à un IMSI, pas à une carte. Voir Support SIM Multi-IMSI.
- **Roaming hébergé ou sponsorisé.** L'applet SIM s'authentifie avec un IMSI visité. Le réseau hôte traduit l'IMSI visité en votre IMSI domestique

avant que la demande n'atteigne le HSS. Les deux IMSIs partagent un Ensemble de Clés.

Un Ensemble de Clés peut donc servir plus d'un Abonné. C'est pourquoi l'Ensemble de Clés est un objet séparé, et non un champ sur la SIM.

Corrélation à l'importation

Le fichier fournisseur de SIM vous donne l'ICCID et les clés sur une ligne. À l'importation, vous créez l'Ensemble de Clés, créez la SIM, et créez l'Abonné qui référence les deux. L'Abonné est l'enregistrement connectif. Pour le flux de travail d'importation, voir Pré-Provisionnement d'Abonnés Inactifs.

MSISDN

Le **MSISDN** représente un numéro de téléphone.

MSISDN			
uuid	id	PK	
string	msisdn	UK	1-15 chiffres

assigné

JOIN_SUB_MSISDN		
uuid	subscriber_id	FK
uuid	msisdn_id	FK

appartient



Champs :

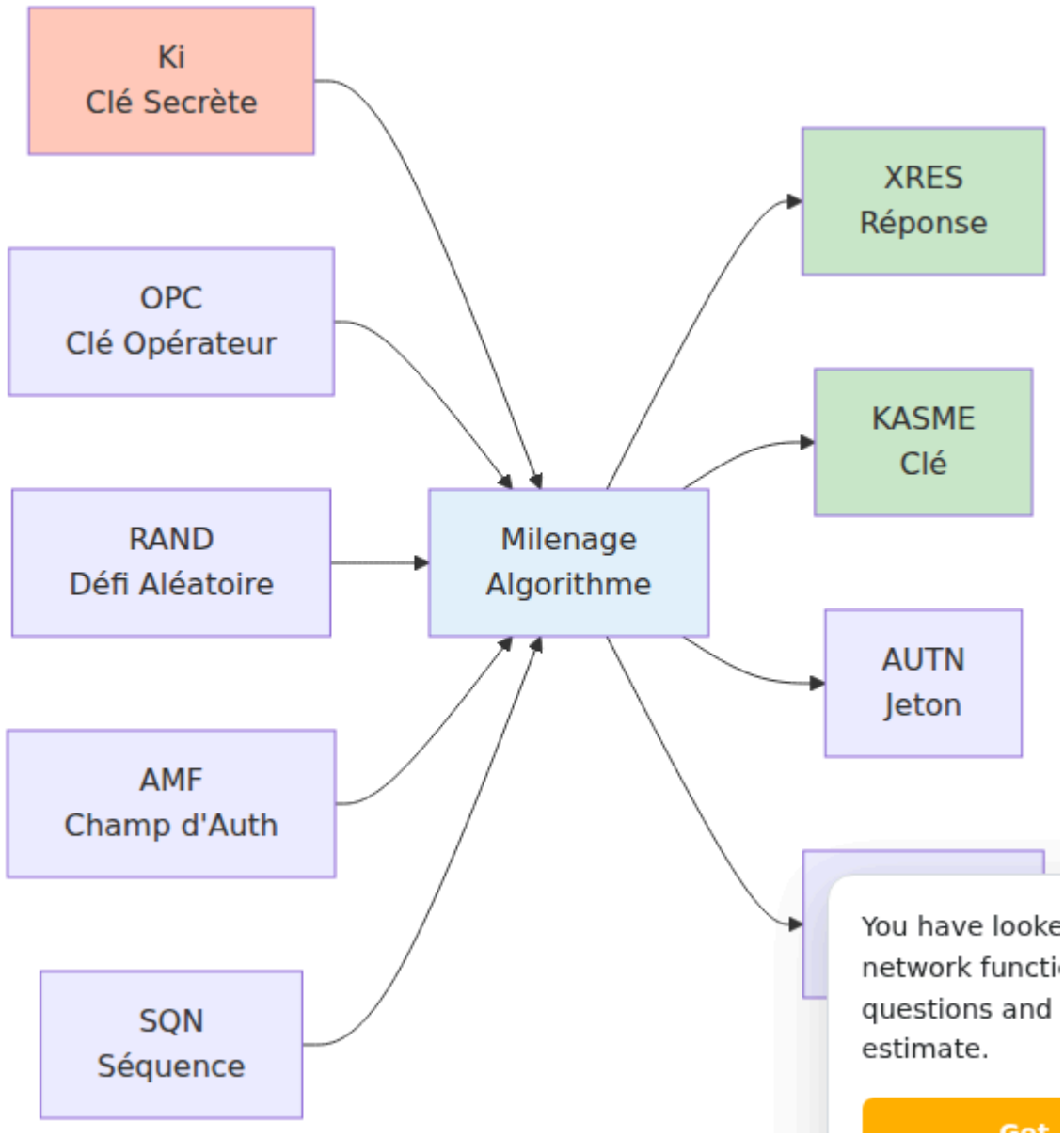
Champ	Type	Description	Format
`msisdn`	string	Numéro ISDN de la Station Mobile	1-15 chiffres, format E.164

Points Clés :

- MSISDN est le numéro de téléphone au format international.
- Un abonné peut avoir plusieurs MSISDN (un IMSI, plusieurs numéros).
- Un MSISDN peut être assigné à plusieurs abonnés (plusieurs IMSIs, un numéro). Utilisez cela pour un numéro partagé, par exemple un téléphone et une montre connectée. Le numéro est stocké comme un enregistrement MSISDN, et chaque abonné détient une ligne de jointure vers celui-ci.

- Une recherche inversée par MSISDN (par exemple, l'API `GET /subscriber/msisdn/:msisdn`, ou une recherche Diameter indexée sur le numéro) retourne un abonné. Lorsqu'un numéro est partagé, utilisez l'IMSI ou l'ICCID pour sélectionner un abonné spécifique.
- Format : Code pays + Numéro national, chiffres uniquement - pas de préfixe `+` (par exemple, `14155551234` et non `+14155551234`)
- Le préfixe `+` est automatiquement supprimé lors des recherches Diameter Cx/Rx lorsque le S-CSCF ou P-CSCF envoie un URI SIP avec ou sans le `+`

Modèle Multi-MSISDN :

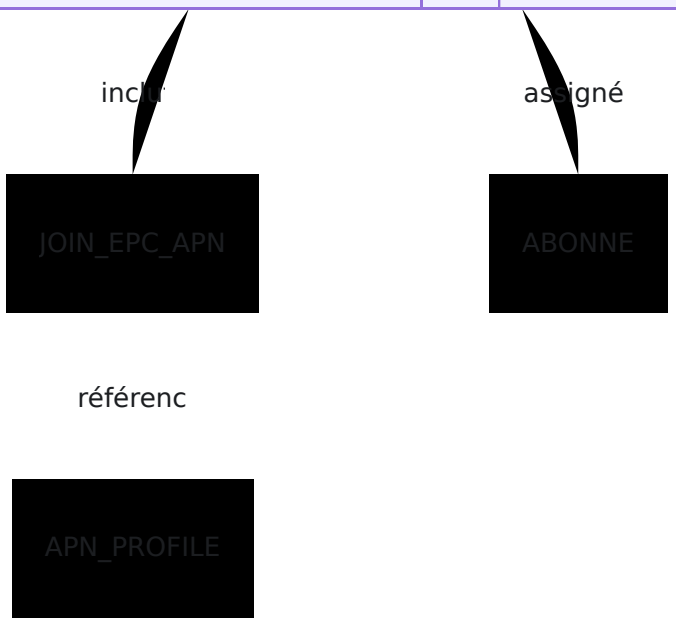


Entités de Profil

Profil EPC

Le **Profil EPC** définit les caractéristiques du service de données pour LTE.

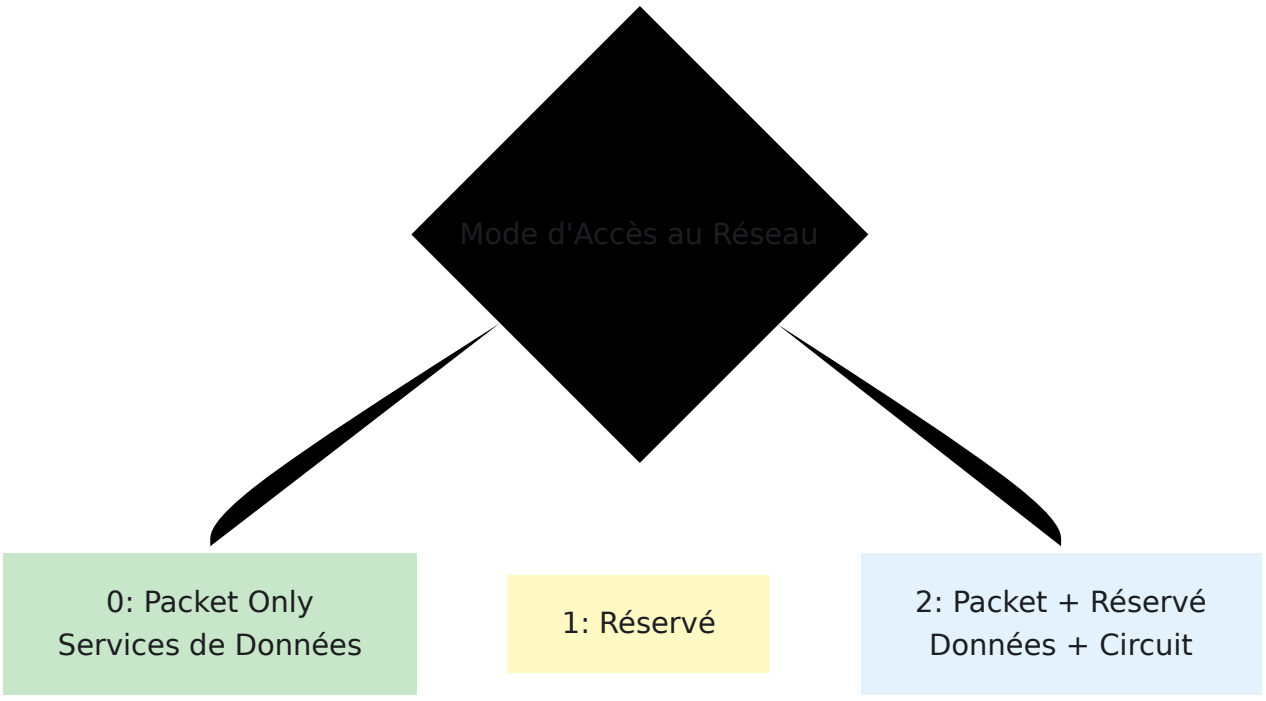
EPC_PROFILE			
uuid	id	PK	
string	name	UK	
integer	ue_ambr_dl_kbps		
integer	ue_ambr_ul_kbps		
string	network_access_mode		packet_only or packet_and_circuit
integer	tracking_area_update_interval_seconds		



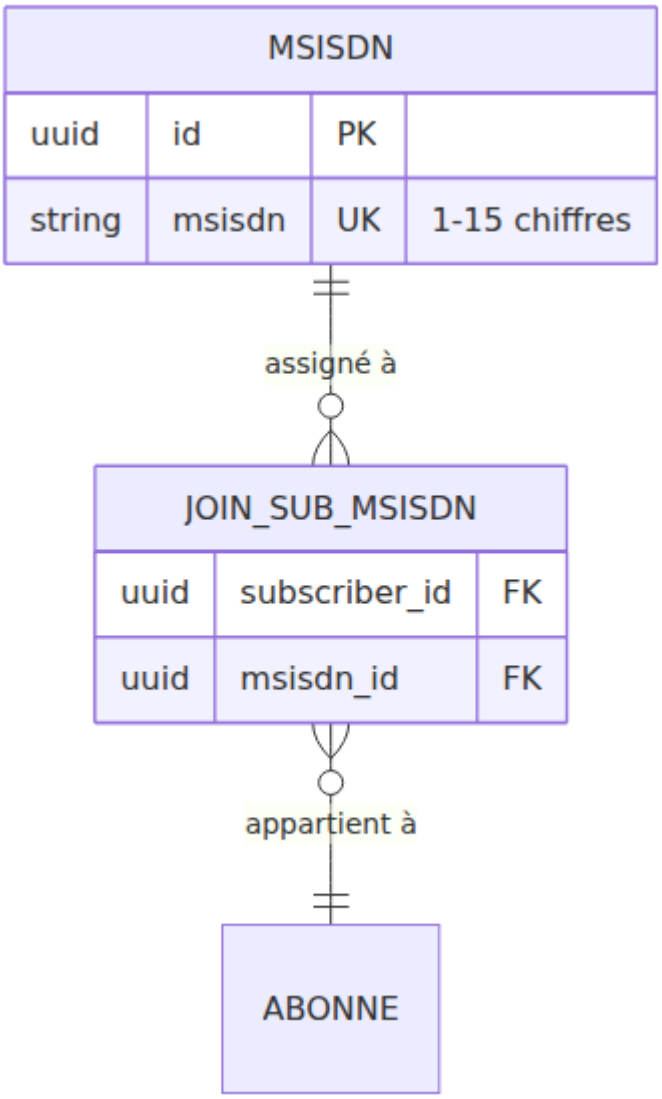
Champs :

Champ	Type	Description	
`name`	string	Nom du profil	Tex
`ue_ambr_dl_kbps`	integer	Limite de bande passante de téléchargement	Kbp
`ue_ambr_ul_kbps`	integer	Limite de bande passante de téléchargement	Kbp
`network_access_mode`	string	Restrictions d'accès	"pa "pa
`tracking_area_update_interval_seconds`	integer	Intervalle TAU	Sec

Modes d'Accès au Réseau :



AMBR (Débit Maximum Agrégé) :



Profil IMS

Le **Profil IMS** définit les caractéristiques des services voix/vidéo.

IMS_PROFILE		
uuid	id	PK
string	name	UK
text	ifc_template	

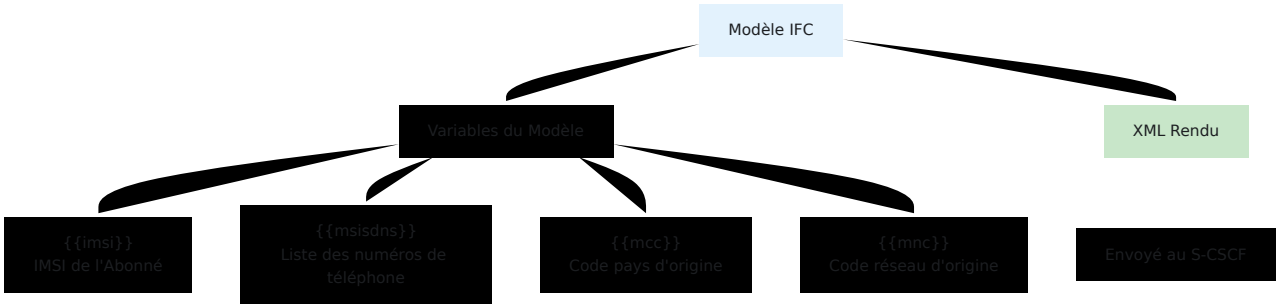
assigné



Champs :

Champ	Type	Description	Format
`name`	string	Nom du profil	Texte
`ifc_template`	text	Modèle XML de Critères de Filtrage Initial	XML avec variables

Variables du Modèle IFC :



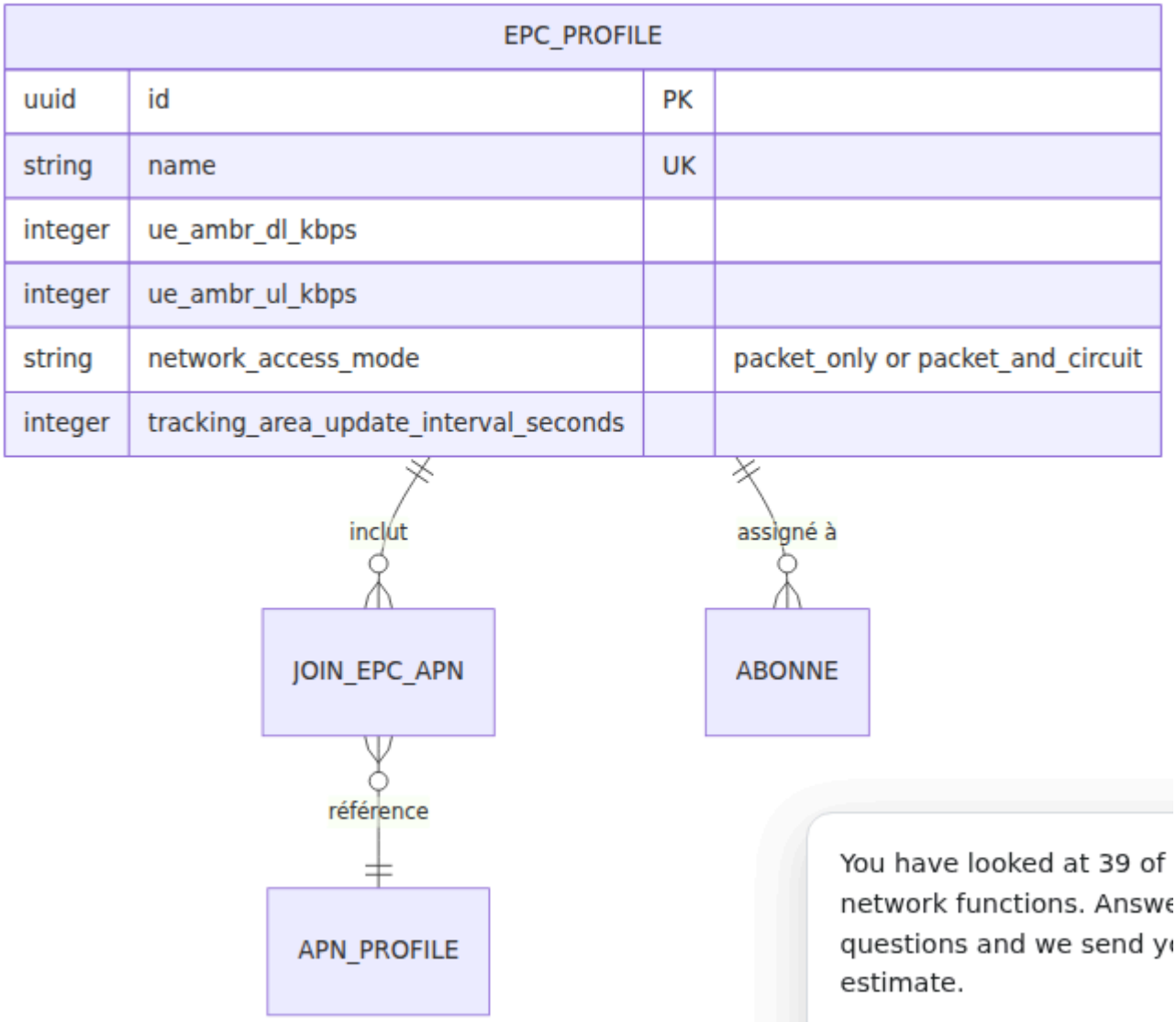
Points Clés :

- IFC (Critères de Filtrage Initial) contrôle le routage des appels dans l'IMS
- Le modèle est rendu lorsque l'abonné s'enregistre
- Les variables sont substituées par les données réelles de l'abonné

- Envoyé au S-CSCF lors de l'enregistrement IMS

Profil APN

Le **Profil APN** définit les caractéristiques pour un point d'accès de données spécifique.



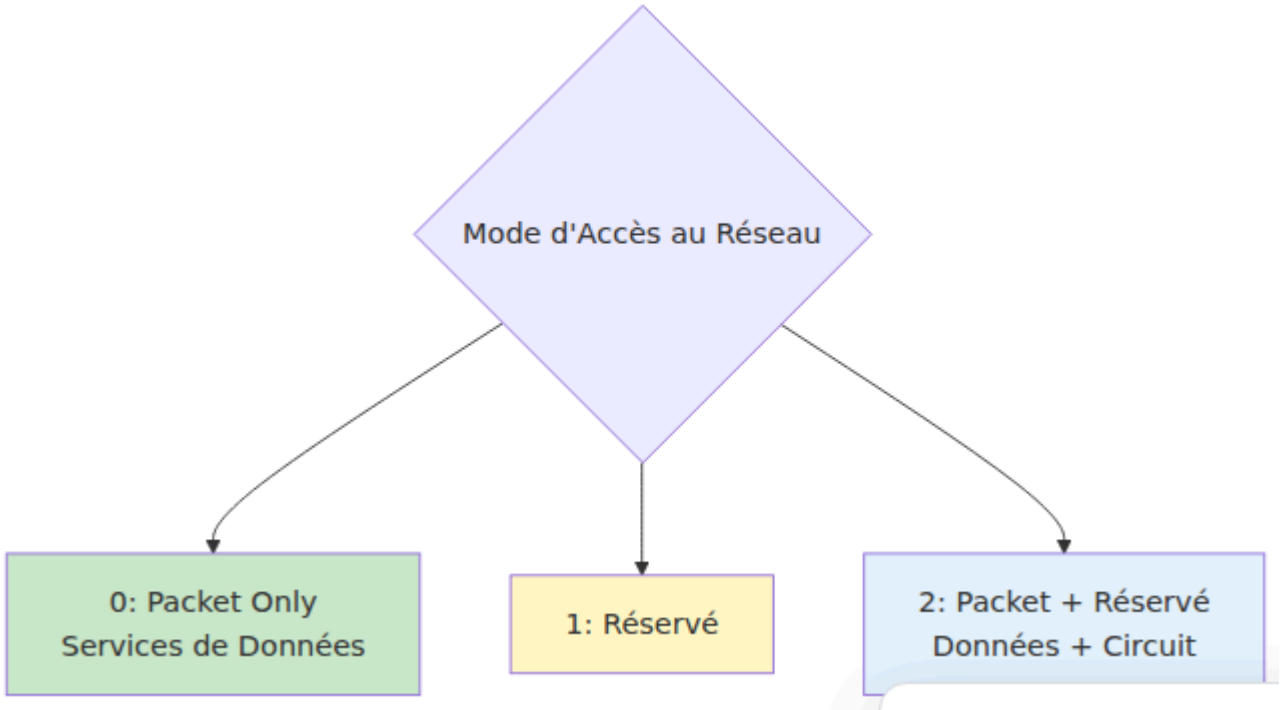
Entités Associées :

Identifiant APN

APN_IDENTIFIER			
uuid	id	PK	
string	apn	UK	
string	ip_version		ipv4, ipv6, ipv4v6, or ipv4_or_ipv6

Champ	Type	Description	Exemple
`apn`	string	Nom de l'APN	"internet", "ims", "mms"
`ip_version`	string	Support du protocole IP	Voir ci-dessous

Options de Version IP :



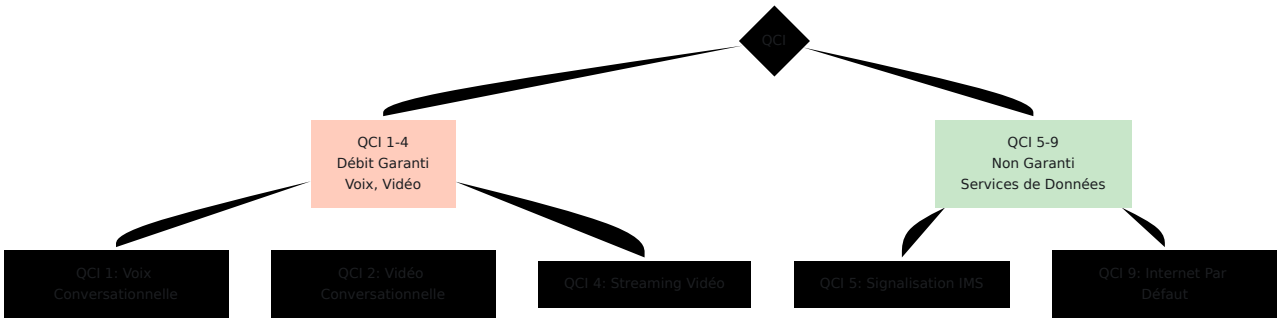
Profil QoS APN

APN_QOS_PROFILE		
uuid	id	PK
string	name	UK
integer	qci	
integer	allocation_retention_priority	
integer	apn_ambr_dl_kbps	
integer	apn_ambr_ul_kbps	
boolean	pre_emption_capability	
boolean	pre_emption_vulnerability	

Paramètres QoS :

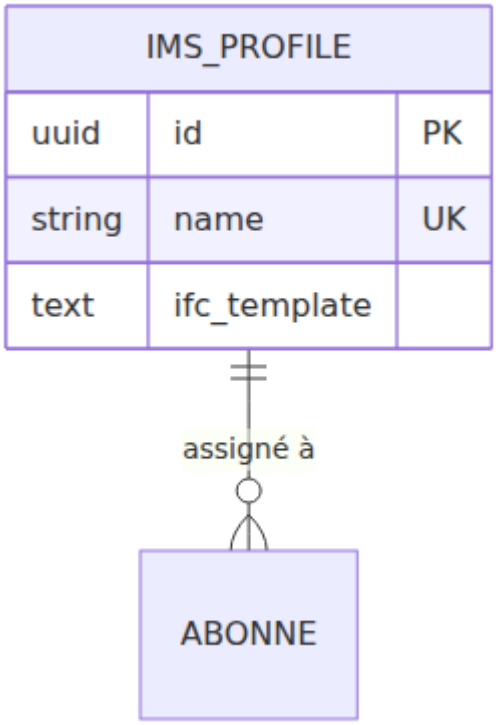
Paramètre	Description	Plage	Porteur Par Défaut
`qci`	Identifiant de Classe QoS	1-9	QCI 9 (Internet)
`allocation_retention_priority`	Priorité ARP	1-15	8 (priorité inférieure)
`apn_ambr_dl_kbps`	Limite de téléchargement APN	0+	Varie
`apn_ambr_ul_kbps`	Limite de téléchargement APN	0+	Varie
`pre_emption_capability`	Peut préempter d'autres	true/false	false
`pre_emption_vulnerability`	Peut être préempté	true/false	true

Valeurs QCI :



Profil de Roaming

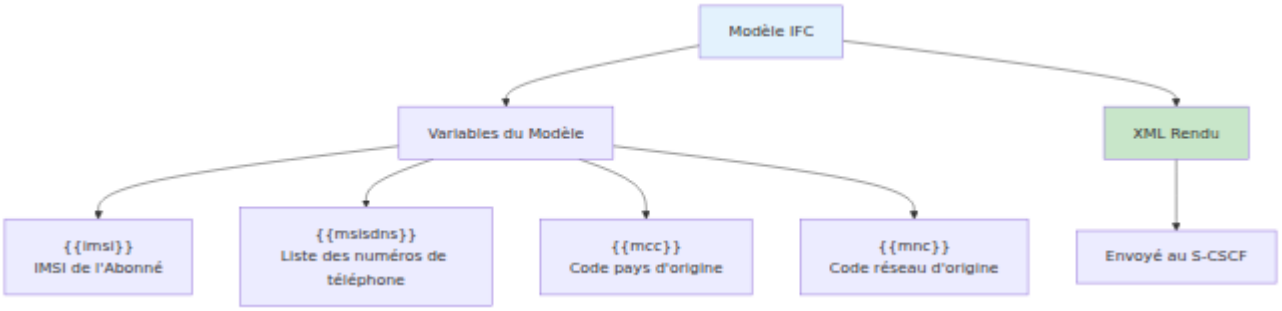
Le **Profil de Roaming** contrôle l'accès lorsque l'abonné visite d'autres réseaux.



Règle de Roaming :

ROAMING_RULE			
uuid	id	PK	
string	name	UK	
string	mcc		3 chiffres
string	mnc		2-3 chiffres
string	data_action		allow or deny
string	ims_action		allow or deny

Évaluation des Règles :



Règle EIR

Règles d'identité d'équipement évaluées par la fonction EIR S13. Les Règles EIR sont autonomes et ne sont pas attachées à un Abonné ou à un profil. Voir Documentation EIR.

EIR_RULE			
uuid	id	PK	
string	action		whitelist, blacklist or greylist
string	regex		testé contre IMEISV, puis IMEI, puis IMSI
integer	precedence		1-65535, plus bas est évalué en premier

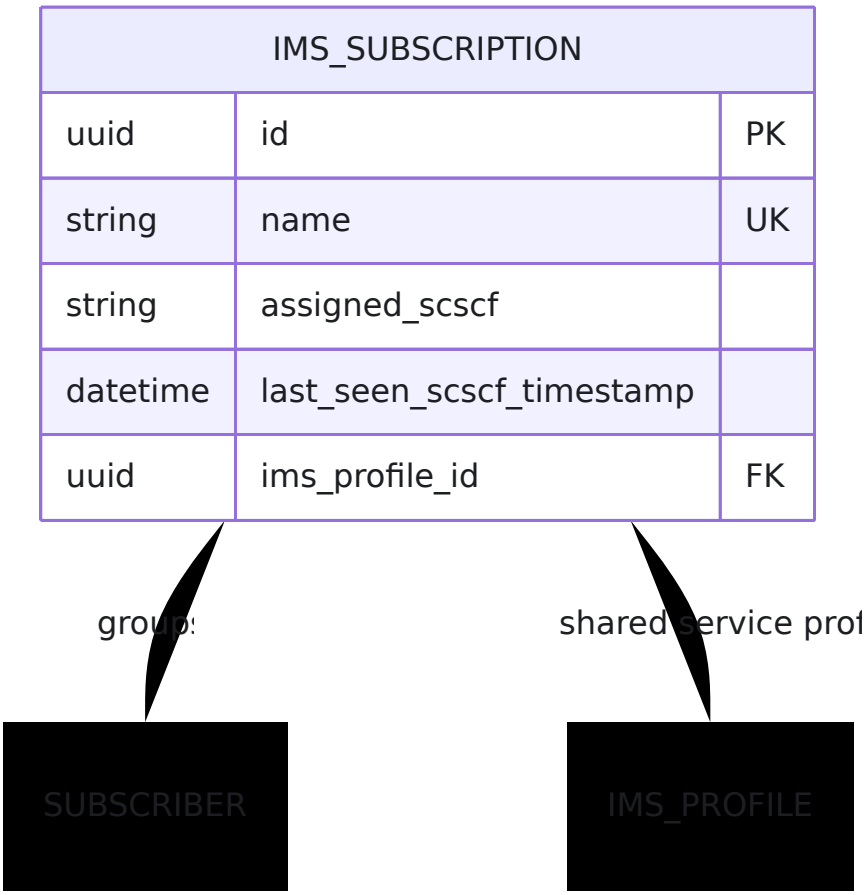
Champ	Type	Description	Exemple
`action`	string	Résultat retourné lorsque la règle correspond	"blacklist"
`regex`	string	Expression régulière testée contre l'IMEISV, puis l'IMEI, puis l'IMSI	"35979139461640"
`precedence`	integer	Ordre d'évaluation dans chaque étape de correspondance, le plus bas en premier. Omettre lors de la création pour ajouter ; fournir une valeur pour insérer à cette position et décaler les règles à ou après celle-ci d'une unité	1

Voir Précédence des Règles pour le comportement complet de placement.

Entités d'État

Abonnement IMS

L'**Abonnement IMS** regroupe plusieurs abonnés (IMPIs / IMSIs) en un seul abonnement IMS qui partage une Identité Publique Utilisateur et un S-CSCF unique. Utilisez-le pour servir un numéro sur plusieurs appareils, par exemple un téléphone et une montre connectée sur des SIM séparées.


IMS_SUBSCRIPTION
uuid
id
PK
string
name
UK
string
assigned_scscf
datetime
last_seen_scscf_timestamp
uuid
ims_profile_id
FK
SUBSCRIBER
IMS_PROFILE

Champs :

Champ	Type	Description	Contraintes
`name`	string	Nom lisible par l'homme	Requis, unique
`assigned_scscf`	string	S-CSCF partagé pour le groupe	État d'exécution, défini par le HSS
`last_seen_scscf_timestamp`	datetime	Quand le S-CSCF partagé a été écrit pour la dernière fois	État d'exécution
`ims_profile_id`	uuid	Profil de service partagé optionnel pour les identités partagées du groupe	Optionnel

Points clés :

- Les membres référencent le groupe via `subscriber.ims_subscription_id`.
- Le HSS assigne un S-CSCF lorsque le premier membre s'enregistre et le libère lorsque le dernier membre se désinscrit. Ne définissez pas `assigned_scscf` vous-même.
- Lorsque `ims_profile_id` est défini, le profil IMS du groupe rend les identités partagées ; sinon, chaque membre rend son propre profil IMS.
- La suppression d'une souscription IMS ne supprime pas ses abonnés ; leur `ims_subscription_id` est effacé.

Voir Identité Publique Utilisateur Partagée pour le modèle complet et la provision.

État de l'abonné

L'**État de l'abonné** suit le statut en temps réel de l'abonné.

APN_IDENTIFIER			
uuid	id	PK	
string	apn	UK	
string	ip_version		ipv4, ipv6, ipv4v6, or ipv4_or_ipv6

Champs clés :

Informations de localisation :

- `last_seen_mcc`, `last_seen_mnc` - Réseau visité
- `last_seen_tac` - Code de zone de suivi
- `last_seen_cell_id` - ID de cellule
- `last_seen_enodeb_id` - ID eNodeB
- `last_seen_eci` - Identifiant de cellule E-UTRAN

Éléments de réseau :

- `last_seen_mme` - MME actuel servant l'abonné
- `last_seen_realm` - Domaine Diameter du MME
- `last_seen_rat_type` - Technologie d'accès radio (LTE, 5G, etc.)

Informations IMS :

- `assigned_scscf` - S-CSCF actuel servant l'abonné
- `ims_public_identity` - URI SIP (par exemple, sip:+14155551234@ims.example.com)
- `sh_repository_data` - Données de profil IMS personnalisées

Équipement :

- `last_seen_imei` - IMEI signalé pour la dernière fois dans l'AVP d'informations terminales d'un ECR S13 ou d'un ULR S6a

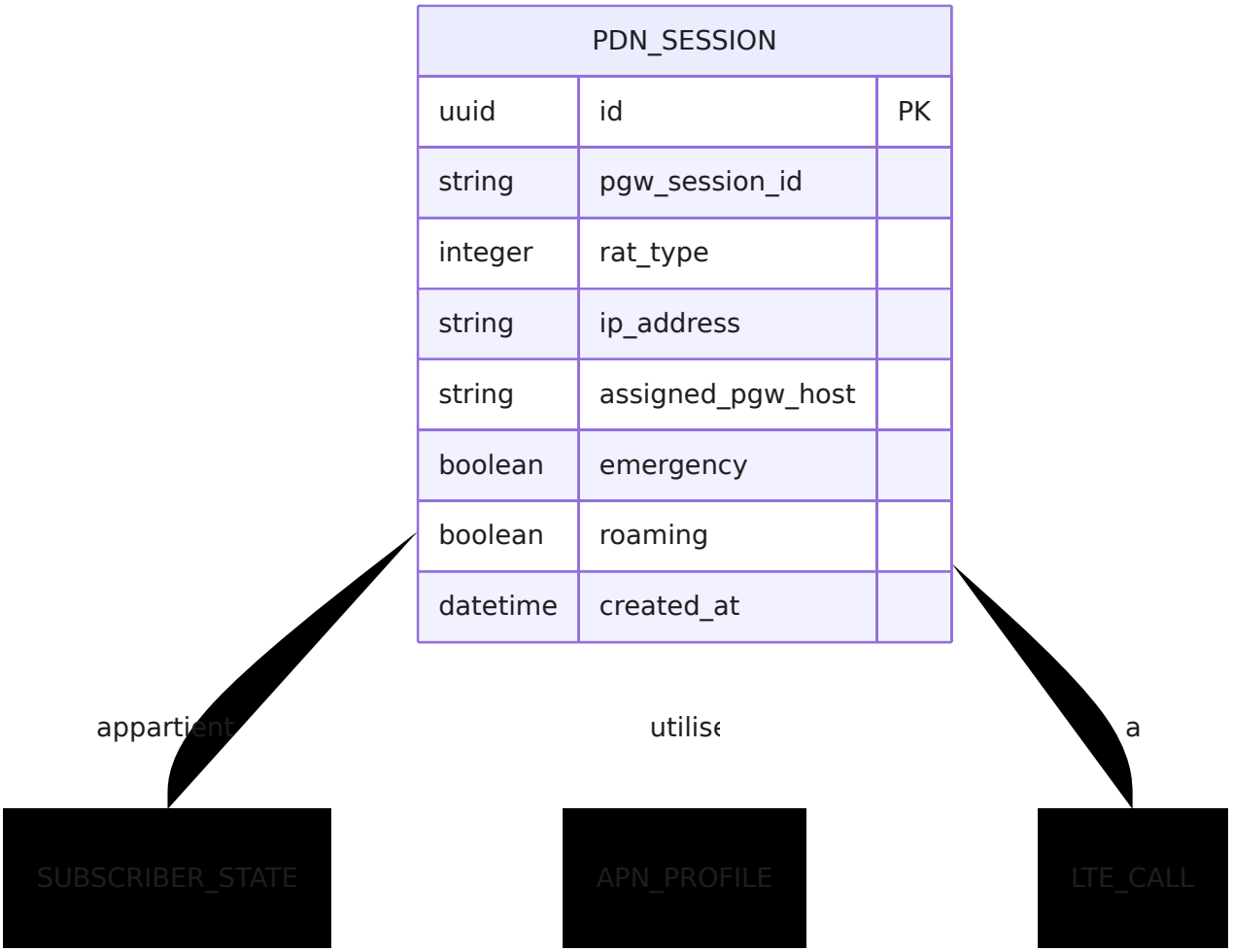
- `last_seen_imei_software_version` - Version du logiciel signalée avec cet IMEI (IMEISV = IMEI + Version du logiciel) ; conservée du rapport précédent lorsque le même IMEI arrive sans
- `last_seen_imei_timestamp` - Quand l'IMEI a été signalé pour la dernière fois

Horodatages :

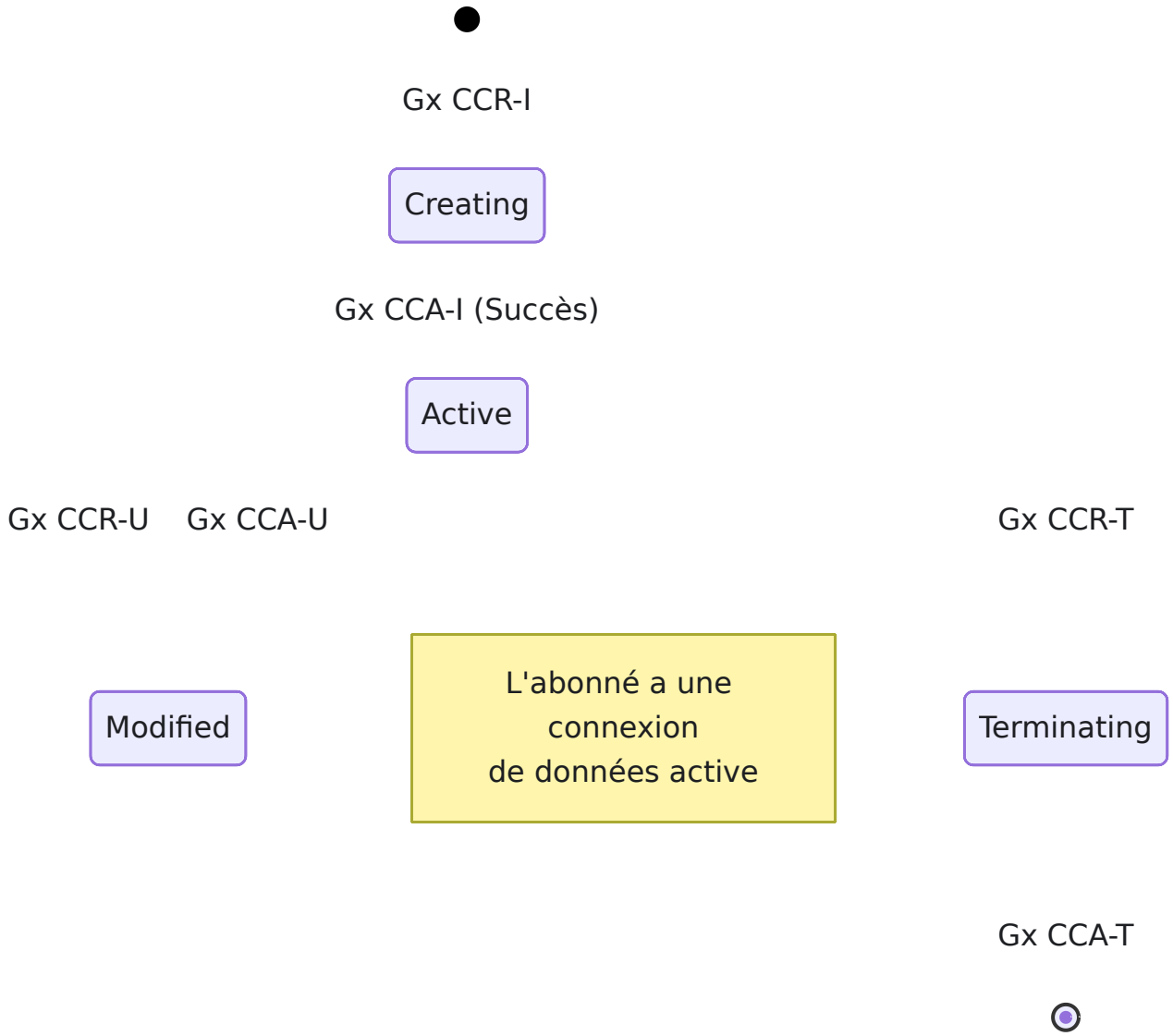
- `last_seen_at` - Dernier message Diameter reçu
- Divers horodatages `last_*_at` pour différentes procédures

Session PDN

La **Session PDN** représente une connexion de données active.



Cycle de vie de la session PDN :

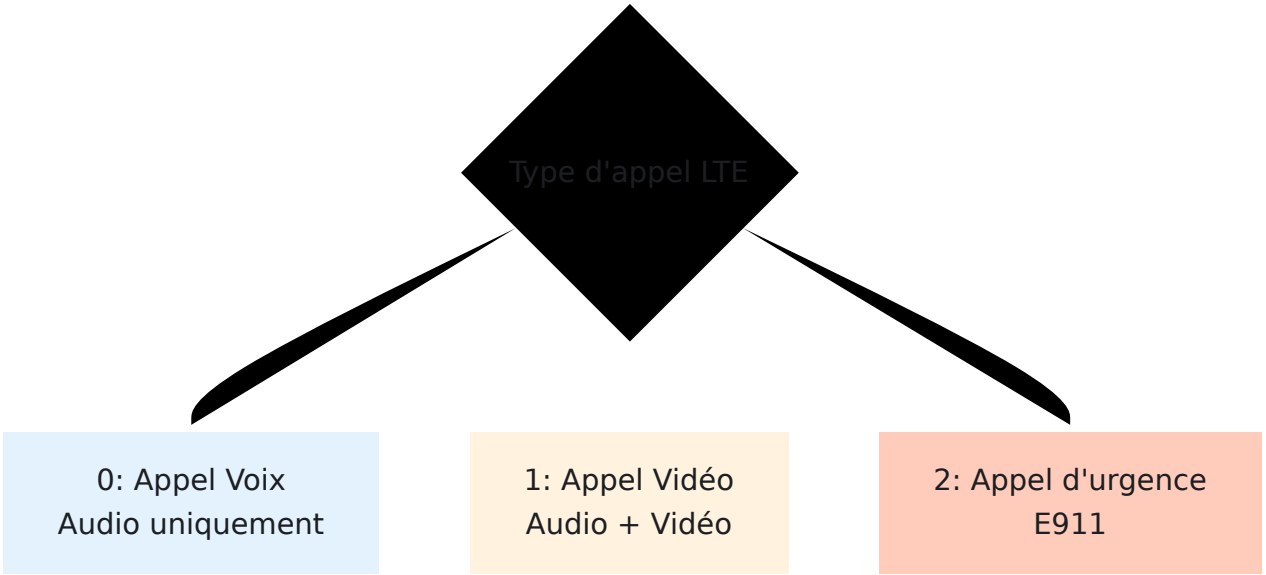


Appel LTE

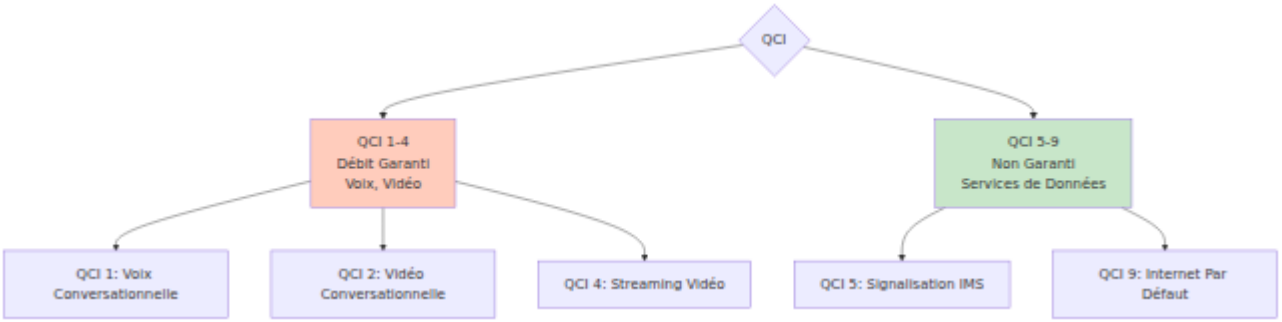
L'**Appel LTE** représente un appel VoLTE voix/vidéo actif.

APN_QOS_PROFILE		
uuid	id	PK
string	name	UK
integer	qci	
integer	allocation_retention_priority	
integer	apn_ambr_dl_kbps	
integer	apn_ambr_ul_kbps	
boolean	pre_emption_capability	
boolean	pre_emption_vulnerability	

Types d'appels :

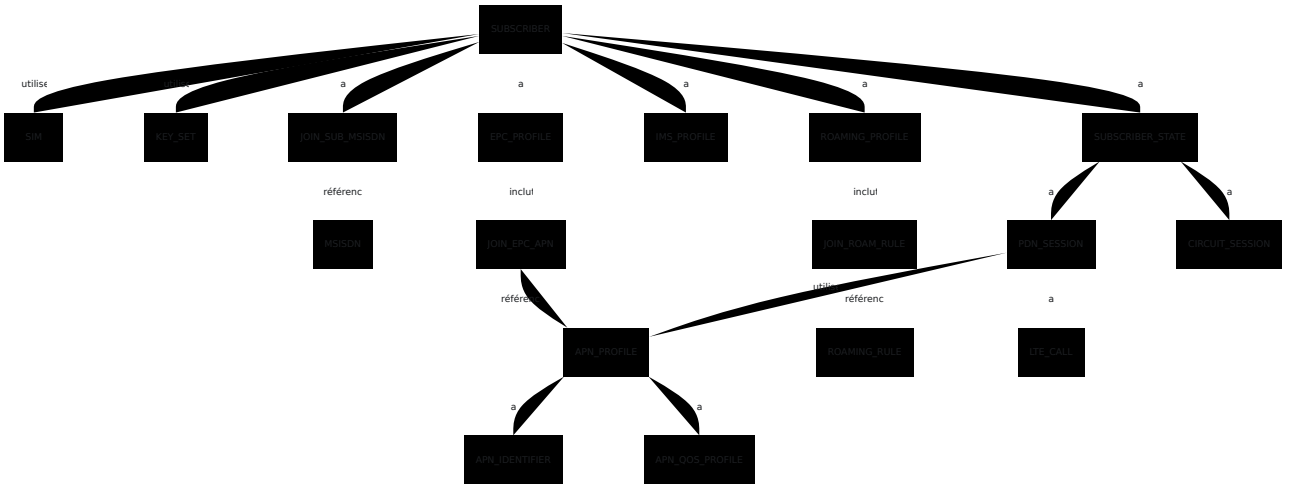


Flux d'appel VoLTE :



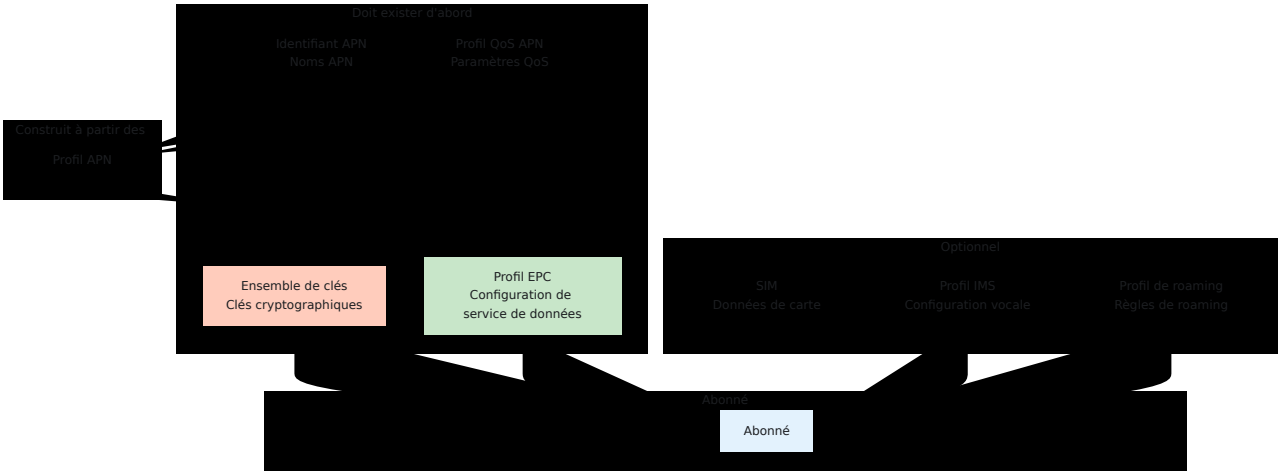
Diagrammes de relations d'entité

Relations d'entité complètes



Relations de provisionnement

Ce diagramme montre ce qui doit exister avant de créer un abonné :

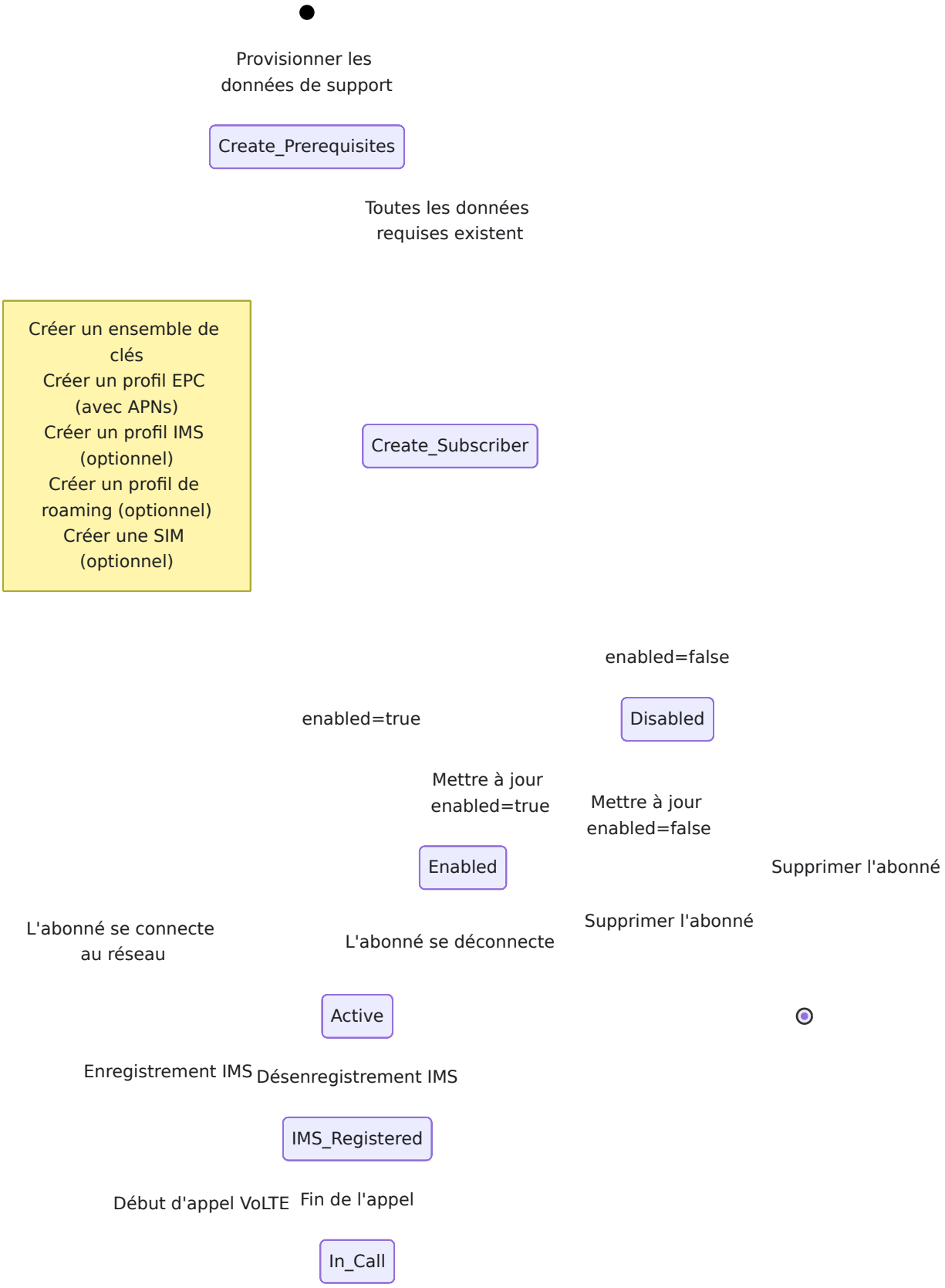


Relations d'état de session

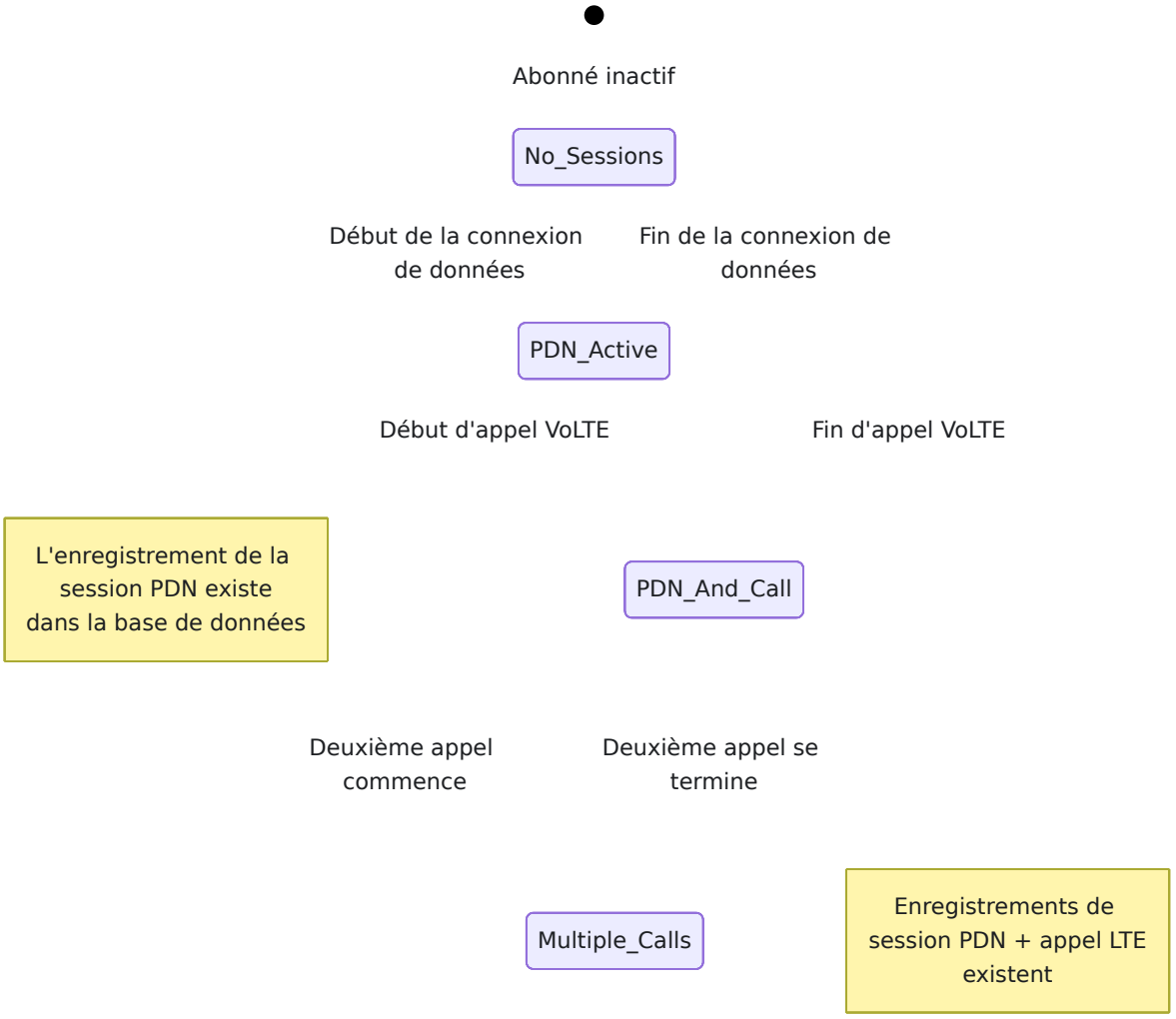
ROAMING_RULE			
uuid	id	PK	
string	name	UK	
string	mcc		3 chiffres
string	mnc		2-3 chiffres
string	data_action		allow or deny
string	ims_action		allow or deny

Cycle de vie de l'entité

Cycle de vie de provisionnement de l'abonné



Cycle de vie de session

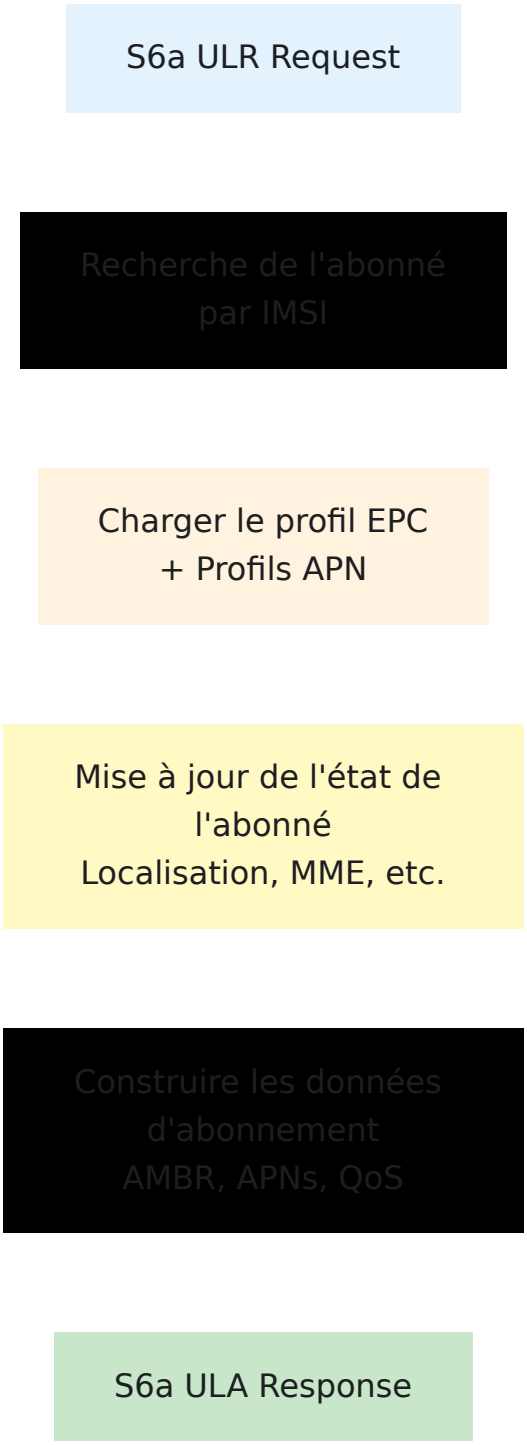


Modèles de flux de données

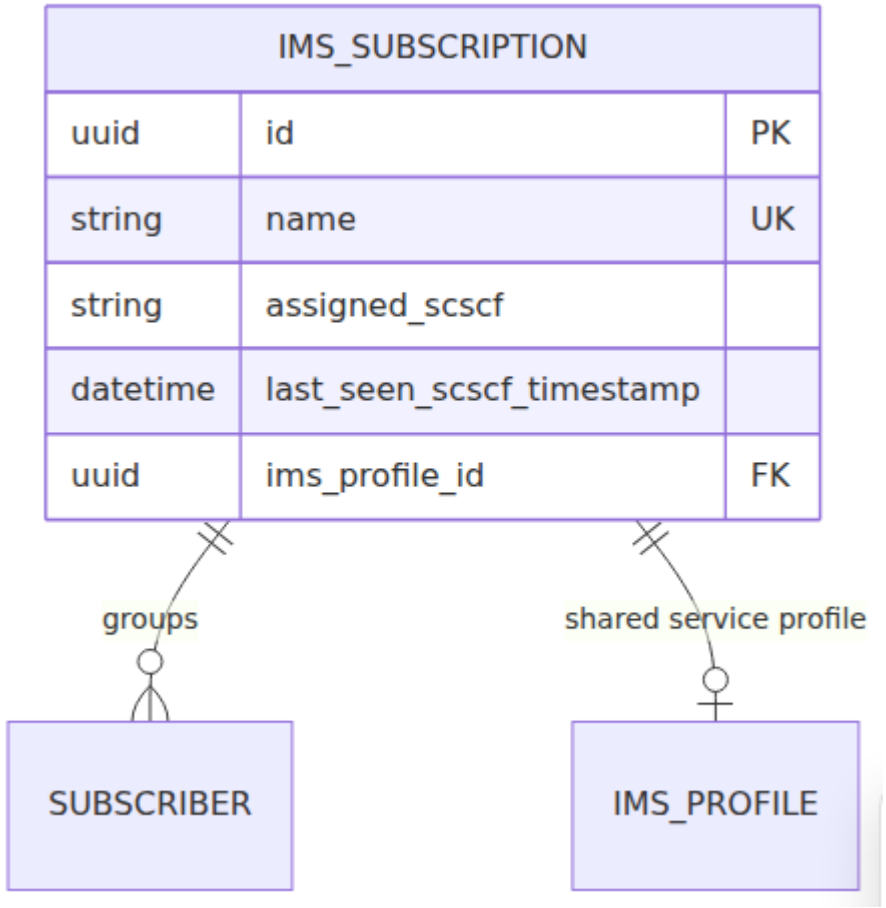
Flux d'authentification

EIR_RULE			
uuid	id	PK	
string	action		whitelist, blacklist or greylist
string	regex		testé contre IMEISV, puis IMEI, puis IMSI
integer	precedence		1-65535, plus bas est évalué en premier

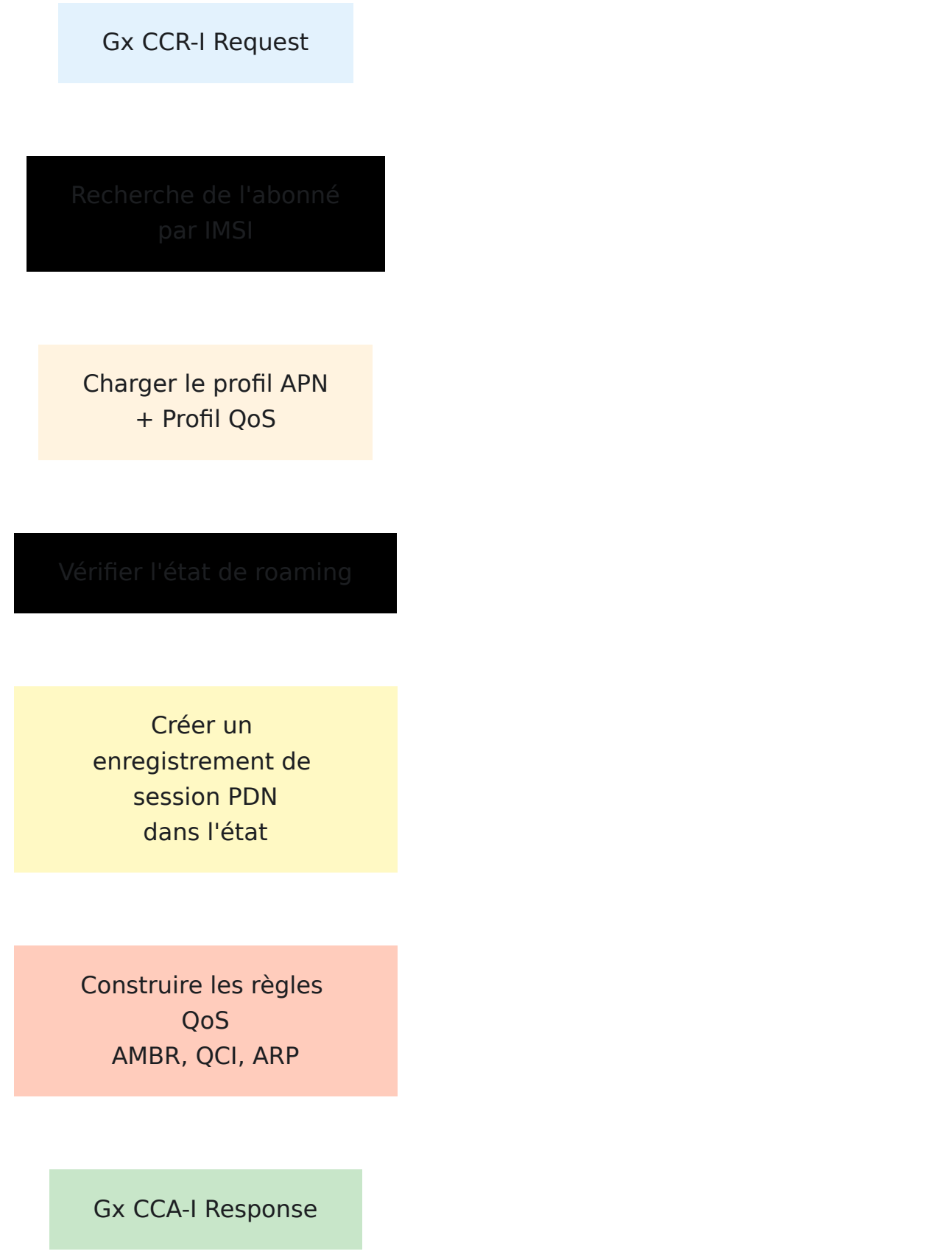
Flux de mise à jour de localisation



Flux d'enregistrement IMS



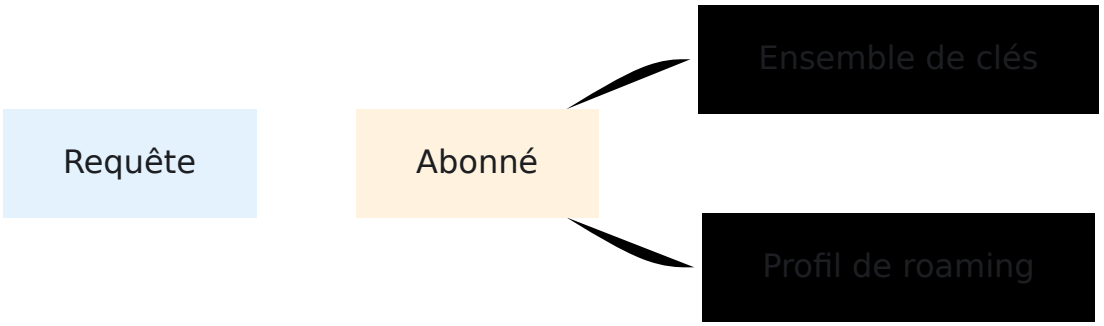
Flux d'établissement de session



Modèles d'optimisation de requête

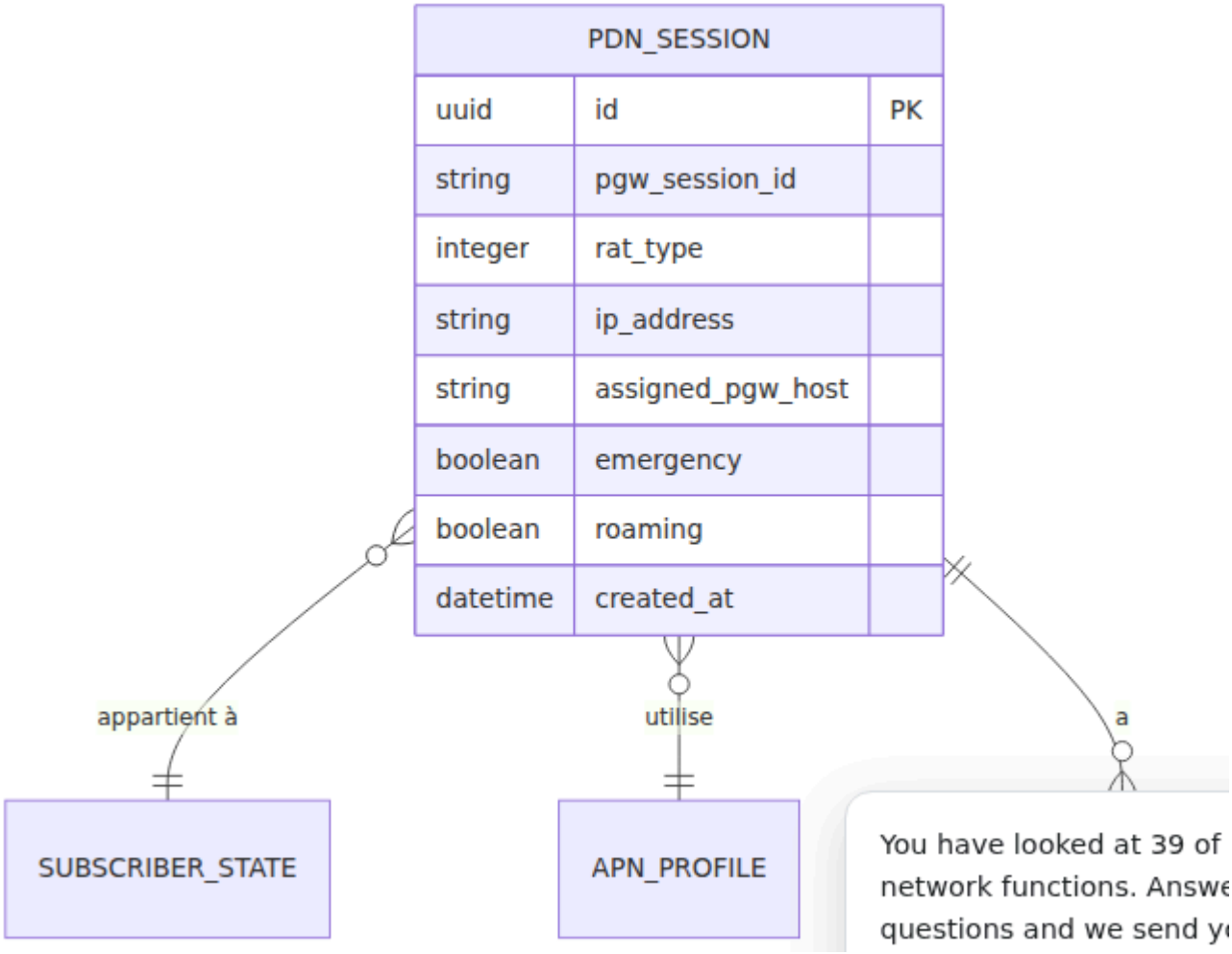
OmniHSS optimise les requêtes de base de données en préchargeant sélectivement uniquement les associations nécessaires pour chaque opération :

Requête minimale (Authentification)



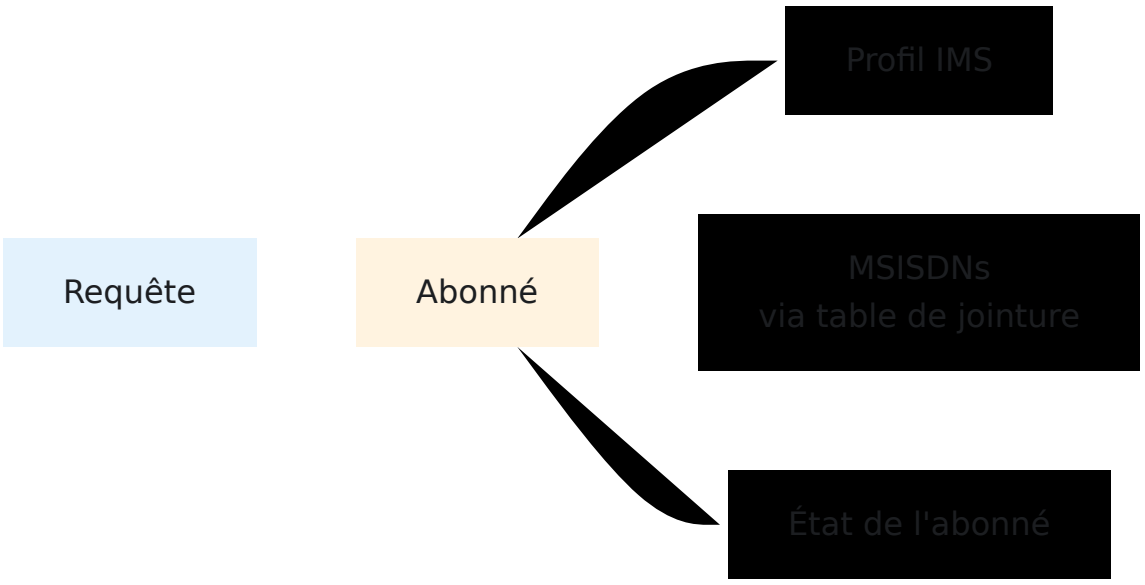
Cas d'utilisation : S6a AIR - Nécessite uniquement des clés cryptographiques et des règles de roaming

Requête modérée (Mise à jour de localisation)



Cas d'utilisation : S6a ULR - Nécessite des données complètes du profil EPC

Requête complète (Enregistrement IMS)



Cas d'utilisation : Cx SAR - Nécessite le profil IMS et tous les numéros de téléphone

---

← Retour au Guide des opérations | Suivant : Référence API →

Cartographie des Données de Réponse Diameter

← Retour au Guide des Opérations

Ce document fournit des diagrammes mermaid détaillés montrant d'où provient chaque champ dans les réponses du protocole Diameter dans le système OmniHSS.

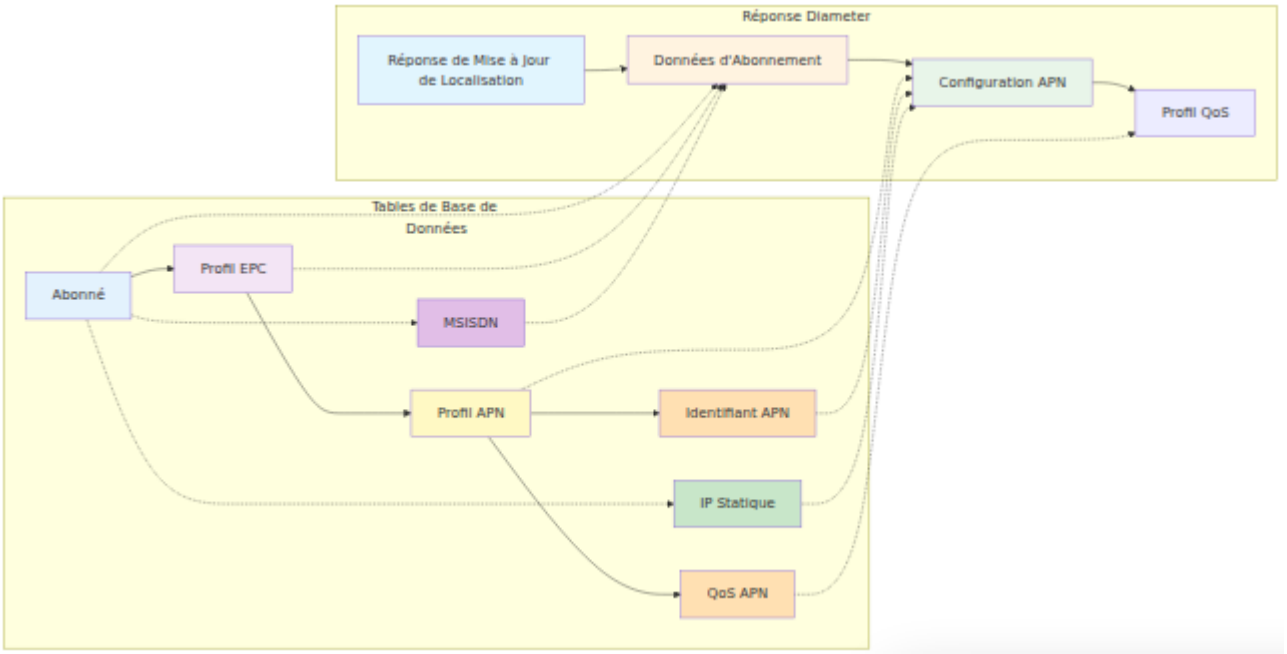
Table des Matières

- Réponse de Mise à Jour de Localisation (S6a ULA)
- Réponse d'Information d'Authentification (S6a AIA)
- Réponse d'Attribution de Serveur (Cx SAA)
- Réponse de Contrôle de Crédit (Gx CCA)
- Réponse de Données Utilisateur (Sh UDA)
- Réponse de Vérification d'Identité ME (S13 ECA)

Réponse de Mise à Jour de Localisation (S6a ULA)

La Réponse de Mise à Jour de Localisation est envoyée par le HSS au MME lors des procédures de connexion LTE. Ce diagramme montre le flux de données complet des tables de base de données aux AVP Diameter.

Cartographie des Sources de Données



Cartographie Détaillée des Champs

Source de Base de Données	Champ	D
subscriber.enabled	`true/false`	Su St
msisdn.msisdn	`'14155551234'`	MS
epc_profile.ue_ambr_ul_kbps	`50000`	Ma Re Ba UL
epc_profile.ue_ambr_dl_kbps	`100000`	Ma Re Ba DL
epc_profile.network_access_mode	`'packet_only'`	Ne Ac Mo
apn_identifier.apn	`'internet'`	Se Se
apn_identifier.ip_version	`'ipv4v6'`	PD
apn_qos_profile.qci	`9`	Qo Id

Source de Base de Données	Champ	D
apn_qos_profile.allocation_retention_priority	`8`	Pr Le
apn_qos_profile.pre_emption_capability	`false`	Pr en Ca
apn_qos_profile.pre_emption_vulnerability	`true`	Pr en Vu
apn_qos_profile.apn_ambr_ul_kbps	`25000`	AP UL
apn_qos_profile.apn_ambr_dl_kbps	`50000`	AP DL
static_ip.ipv4_static_ip	`'100.64.1.1'`	Se Pa Ad (IP
static_ip.ipv6_static_ip	`'2606:4700::1111'`	Se Pa Ad (IP

Transformations Clés:

1. **Bande passante AMBR**: La base de données stocke en kbps, Diameter attend en bps (multiplier par 1000)
2. **Encodage de la Version IP**: `0=IPv4, 1=IPv6, 2=IPv4v6, 3=IPv4_or_IPv6`

3. **Statut de l'Abonné**: `enabled: true → 0 (SERVICE_GRANTED)`, `enabled: false → 1 (OPERATOR_DETERMINED_BARRING)`
4. **Identifiant de Contexte**: Numérotation séquentielle (0, 1, 2...) pour chaque APN dans le profil
5. **IP Statique**: Inclus uniquement si attribué via `static_ips` relation plusieurs-à-plusieurs

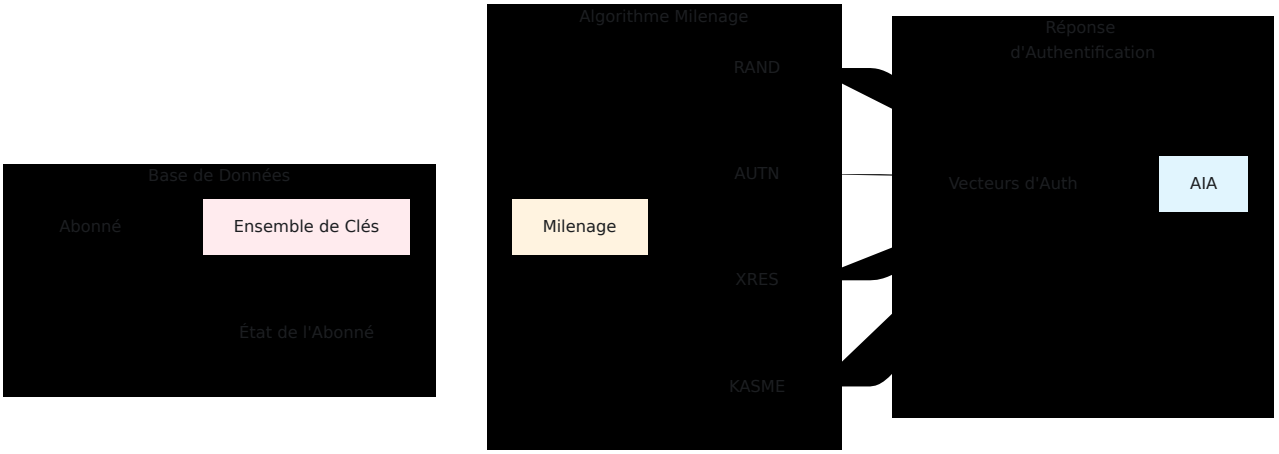
Validation de la Logique Métier:

- Vérification de Roaming: Correspondre le PLMN visité contre `roaming_profile.roaming_rules`
- Vérification de l'Abonné activé: `subscriber.enabled == true`
- Filtrer les APNs: Peut exclure les APNs IMS si la politique de roaming refuse IMS

Réponse d'Information d'Authentification (S6a AIA)

La Réponse d'Information d'Authentification fournit des vecteurs d'authentification pour les abonnés LTE/EPC.

Cartographie des Sources de Données



Composants Clés:

1. **Clés Cryptographiques**: Toutes les clés stockées sous forme de chaînes hexadécimales dans la table `key_set`
2. **Gestion de SQN**: Numéro de séquence incrémenté après chaque génération de vecteur d'authentification (prévention des attaques par rejeu)
3. **Algorithme Milenage**: 3GPP TS 35.206 - génère des vecteurs d'authentification
4. **Dérivation de KASME**: Clé dérivée de CK||IK en utilisant KDF selon TS 33.401

Fonctionnalités de Sécurité:

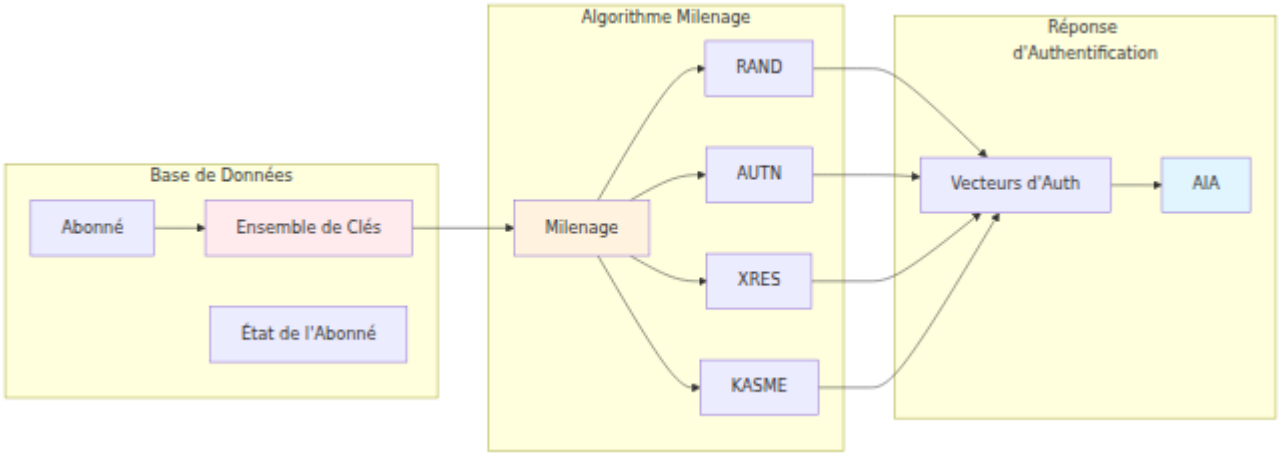
- SQN stocké par abonné (pas global)
- Ki/OPc ne quittent jamais le HSS (seules les valeurs dérivées sont transmises)
- AUTN inclut le numéro de séquence (SQN) et AMF pour l'authentification réseau
- L'algorithme Milenage fournit une authentification mutuelle entre UE et réseau

---

Réponse d'Attribution de Serveur (Cx SAA)

La Réponse d'Attribution de Serveur est envoyée par le HSS au S-CSCF lors de l'enregistrement IMS.

Cartographie des Sources de Données



Fonctionnalités Clés:

1. **Modèle IFC**: Modèle XML stocké dans `ims_profile.ifc_template`
2. **Substitution Dynamique**: Remplace `{{msisdn}}`, `{{imsi}}`, `{{impu}}` à l'exécution
3. **Attribution S-CSCF**: Stocke le S-CSCF attribué dans `subscriber_state.assigned_scscf`
4. **Identité Publique IMS**: Format: `sip:+{msisdn}@{ims_domain}` ou `tel:+{msisdn}`

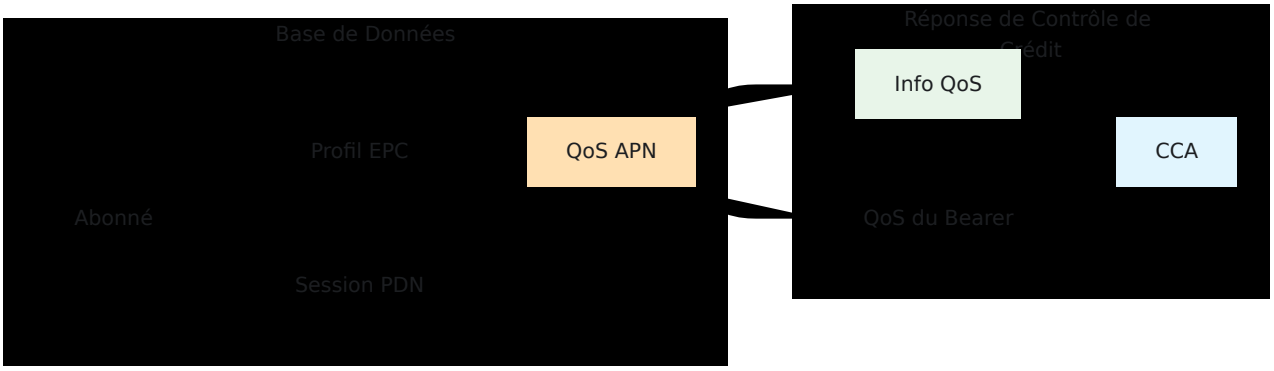
Paramètres du Modèle IFC:

- `{{msisdn}}` - Premier MSISDN de l'abonné
- `{{imsi}}` - IMSI de l'abonné
- `{{impu}}` - Identité Publique Utilisateur IMS (à partir de subscriber_state)
- `{{impi}}` - Identité Privée Utilisateur IMS (typiquement IMSI@realm)

Réponse de Contrôle de Crédit (Gx CCA)

La Réponse de Contrôle de Crédit est envoyée par la fonction PCRF au PGW lors de l'établissement du bearer.

Cartographie des Sources de Données



Fonctionnalités Clés:

1. **Suivi de Session**: Crée/met à jour l'enregistrement `pdn_session` pour chaque bearer
2. **Application de QoS**: Fournit QCI et limites de bande passante à partir du profil QoS APN
3. **Règles de Facturation**: Renvoie les règles de facturation par défaut pour l'intégration de facturation
4. **CC-Request-Type**: Gère INITIAL (1), UPDATE (2), TERMINATION (3)

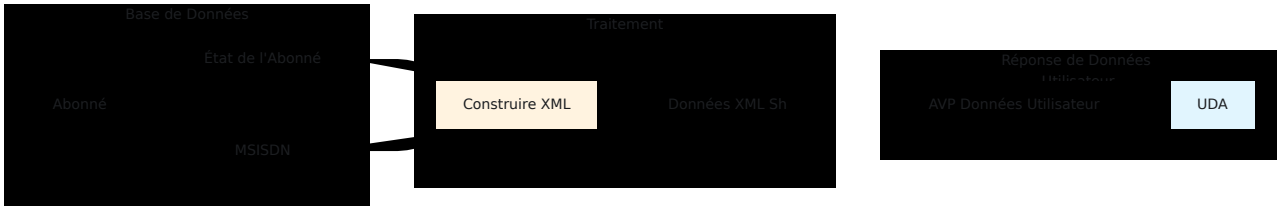
Gestion de l'État de Session:

- `INITIAL_REQUEST`: Crée un nouvel enregistrement de session PDN
- `UPDATE_REQUEST`: Met à jour la session PDN existante
- `TERMINATION_REQUEST`: Supprime l'enregistrement de session PDN

Réponse de Données Utilisateur (Sh UDA)

La Réponse de Données Utilisateur est envoyée par le HSS au AS (Serveur d'Application) via l'interface Sh.

Cartographie des Sources de Données



Fonctionnalités Clés:

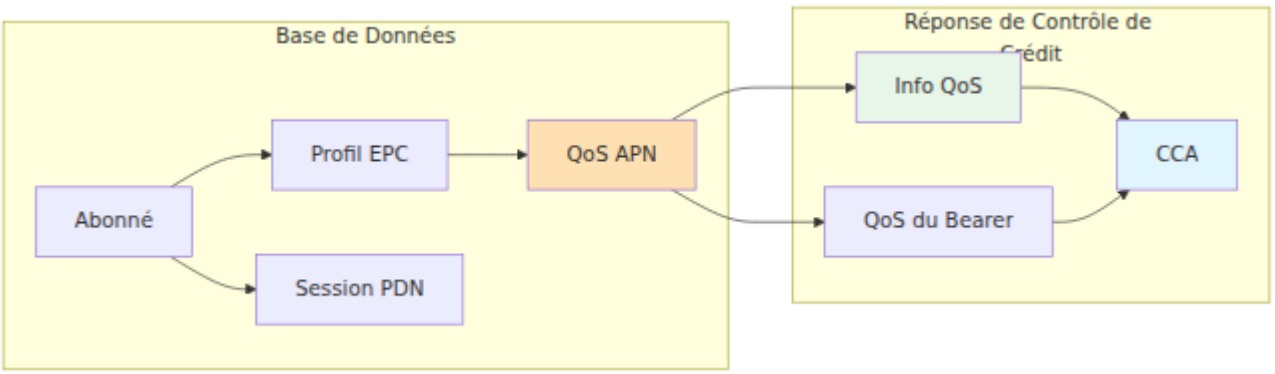
1. **Données de Référentiel**: Peut stocker des XML personnalisés dans `subscriber_state.sh_repository_data`
2. **Indication de Service**: Filtre les données par service demandé (par exemple, présence, messagerie)
3. **Identités Publiques**: Renvoie toutes les identités publiques IMS pour l'abonné
4. **Référence vs Transparent**: Prend en charge les modes de données de référence et transparent

---

Réponse de Vérification d'Identité ME (S13 ECA)

La Réponse de Vérification d'Identité ME est envoyée par la fonction EIR au MME pour la validation de l'IMEI.

Cartographie des Sources de Données



Fonctionnalités Clés:

1. **Correspondance Regex**: Les règles sont des expressions régulières testées contre l'IMEISV, puis l'IMEI, puis l'IMSI si la demande en contient une. À chaque étape, les règles sont évaluées par ordre de `priorité` et la première correspondance gagne. Voir Logique de Correspondance d'Équipement et Priorité des Règles
2. **Règles Basées sur TAC**: Peut correspondre au Code d'Attribution de Type (premiers 8 chiffres)
3. **Comportement Par Défaut**: `unknown_equipment_behaviour` sélectionne liste blanche, liste noire, liste grise ou rejeter (Résultat Expérimental 5422) pour les équipements ne correspondant à aucune règle. Voir Configuration EIR
4. **Valeurs de Statut de l'Équipement**:
   - `0` = LISTE BLANCHE (explicitement autorisé)
   - `1` = LISTE NOIRE (volé/bloqué)
   - `2` = LISTE GRISE (autorisé mais surveillé)
   - Pas de correspondance : le `unknown_equipment_behaviour` configuré sélectionne l'un des ci-dessus, ou avec `:reject_unknown_equipment` l'ECA porte le Code de Résultat Expérimental 5422 (DIAMETER_ERROR_EQUIPMENT_UNKNOWN) au lieu d'un Statut d'Équipement

Cas d'Utilisation:

- Bloquer les appareils volés par IMEI exact
- Bloquer les modèles d'appareils par motif TAC
- Autoriser uniquement les appareils approuvés
- Suivre les appareils du marché gris

Éléments de Réponse Communs

Toutes les réponses Diameter partagent ces AVP communs :

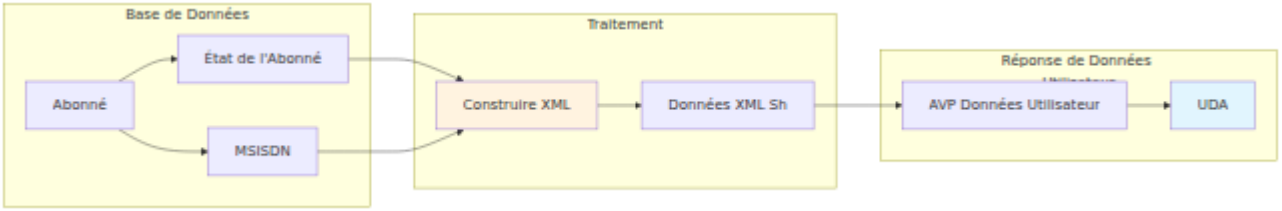


Exemple de Configuration:

```
config :diameter_ex,
  diameter_host: "hss",
  diameter_realm: "example.com",
  diameter_service_name: "OmniHSS"
```

Résumé du Flux de Données

Pipeline de Traitement des Demandes



Notes d'Implémentation

Gestionnaires de Protocole

Le système implémente des gestionnaires pour les protocoles Diameter suivants :

- **S6a** - Interface LTE/MME pour l'authentification et les mises à jour de localisation
- **Cx** - Interface IMS/CSCF pour l'enregistrement IMS et l'attribution de serveur
- **Sh** - Interface IMS/AS pour la récupération des données d'abonné
- **Gx** - Interface PCRF pour le contrôle de politique et de facturation
- **Rx** - Interface IMS/AF pour l'autorisation des médias
- **S13** - Interface EIR pour la validation de l'IMEI
- **SWx** - Interface WiFi/IMS pour l'authentification d'accès non-3GPP

Modèles de Données

Le schéma de base de données comprend les entités principales suivantes :

- **Abonné** - Enregistrement principal de l'abonné avec IMSI
- **Ensemble de Clés** - Clés cryptographiques pour l'authentification
- **Profil EPC** - Configuration de service LTE
- **Profil APN** - Configuration de point d'accès

- **Profil IMS** - Configuration de service IMS avec modèles IFC
- **Profil de Roaming** - Règles et restrictions de roaming
- **État de l'Abonné** - Suivi dynamique de session et d'état
- **Session PDN** - Suivi de session bearer active
- **IP Statique** - Attributions d'adresses IP statiques
- **Règle EIR** - Règles d'identité d'équipement (regex contre IMEISV, IMEI ou IMSI, évaluées par ordre de priorité)

← Retour au Guide des Opérations | Référence API → | Flux de Protocole →

Guide des Métriques et de la Surveillance d'OmniHSS

← Retour au Guide des Opérations

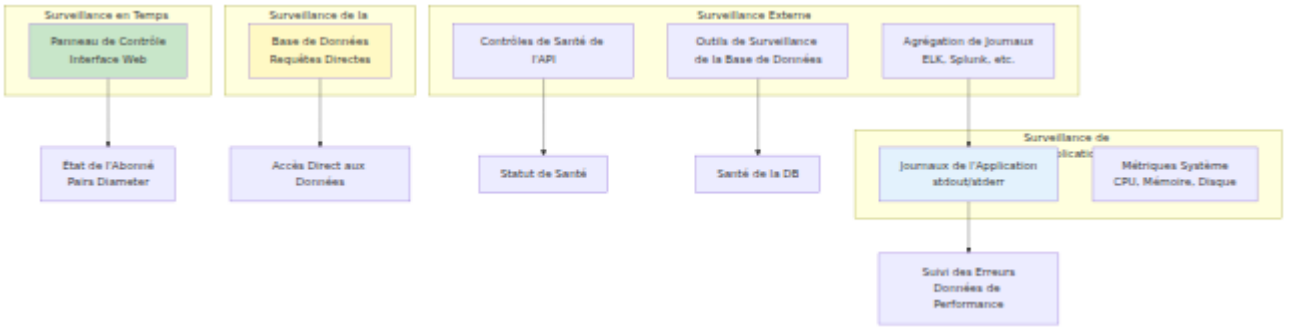
Table des Matières

- Aperçu de la Surveillance
- Surveillance du Panneau de Contrôle
- Surveillance de la Base de Données
- Surveillance des Journaux
- Intégration de la Surveillance Externe
- Indicateurs Clés de Performance
- Stratégies d'Alerte

Aperçu de la Surveillance

OmniHSS fournit plusieurs mécanismes pour surveiller la santé du système, la performance et l'activité des abonnés. Le personnel des opérations doit utiliser une combinaison de ces outils pour une visibilité complète.

Couches de Surveillance



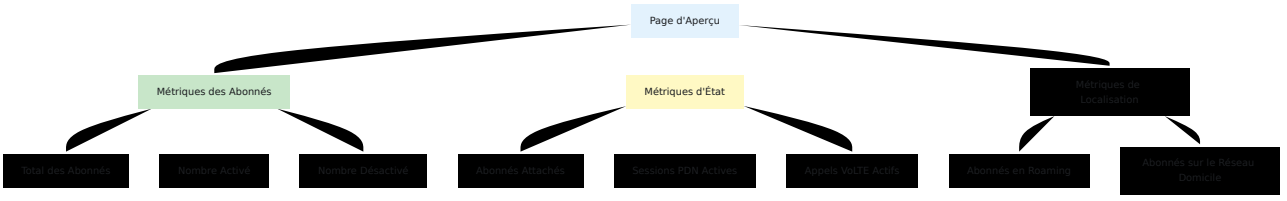
Surveillance du Panneau de Contrôle

Le Panneau de Contrôle fournit l'interface principale de surveillance en temps réel.

Surveillance de la Page d'Aperçu

URL : `https://[hostname]:7443/overview`

Métriques Clés Disponibles



États des Abonnés Surveillés

État	Indicateur	Ce que Cela Signifie
Inactif	Pas d'informations de localisation	Abonné éteint ou hors couverture
Attaché	MME présent	Abonné enregistré sur le réseau
PDN Actif	Nombre de sessions PDN > 0	Connexion de données active
IMS Enregistré	S-CSCF assigné	Services vocaux prêts
En Appel	Nombre d'appels actifs > 0	Appel VoLTE en cours

Extraction des Métriques de l'Aperçu

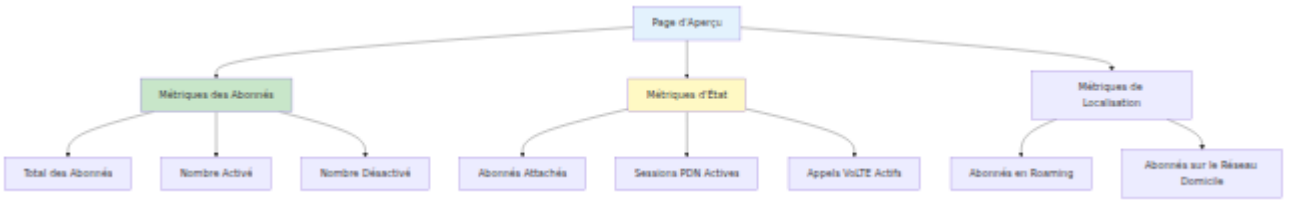
Le Panneau de Contrôle n'exporte pas les métriques directement. Vous pouvez :

1. **Compter les lignes visibles** pour le total des abonnés
2. **Scanner les coches vertes** pour compter les abonnés activés
3. **Examiner les détails étendus** pour les informations d'état
4. **Noter les horodatages de dernière vue** pour la réactivit��

Surveillance de la Page Diameter

URL : `https://[hostname]:7443/diameter`

Métriques Clés



Surveillance des Pairs Critiques

Identifiez les pairs critiques et surveillez leur statut :

Type de Pair	Criticité	Impact si Hors Service
MME	Élevée	Pas de nouvelles connexions LTE
P-GW	Élevée	Pas de sessions de données
S-CSCF	Élevée	Pas d'enregistrements IMS
P-CSCF	Élevée	Pas d'appels VoLTE
I-CSCF	Moyenne	Problèmes de routage IMS
AS	Faible-Moyenne	Service spécifique indisponible

Surveillance de la Page Application

URL : `https://[hostname]:7443/application`

Métriques Clés

Métrique	Description	Plage Normale	Seuil d'Action
Nombre de Processus	Processus Erlang actifs	Varie selon la charge	> 90% de la limite
Utilisation de la Mémoire	Mémoire totale consommée	< 80%	> 90%
Temps de Fonctionnement	Temps depuis le dernier redémarrage	N/A	Suivre pour la stabilité

Surveillance de la Base de Données

Requêtes Directes à la Base de Données

Connectez-vous à la Base de Données SQL pour extraire des métriques détaillées :

Comptes des Abonnés

Interrogez la base de données pour récupérer :

- Nombre total de tous les abonnés
- Nombre d'abonnés activés
- Nombre d'abonnés activés IMS

Statistiques de Session

Interrogez la base de données pour récupérer :

- Nombre de sessions PDN actives
- Nombre d'appels VoLTE actifs
- Répartition des sessions PDN par profil APN

Statistiques de Localisation

Interrogez la base de données pour récupérer :

- Nombre d'abonnés regroupés par réseau visité (combinaison MCC-MNC)
- Nombre d'abonnés actuellement en roaming (pas sur le PLMN domicile 001-001)
- Répartition des abonnés à travers différents réseaux visités

Activité Récente

Interrogez la base de données pour récupérer :

- Nombre d'abonnés vus dans la dernière heure
- Répartition des abonnés par MME de service

- Analyse des horodatages de la dernière activité des abonnés

Surveillance de la Santé de la Base de Données

Surveillez la santé de la base de données en interrogeant :

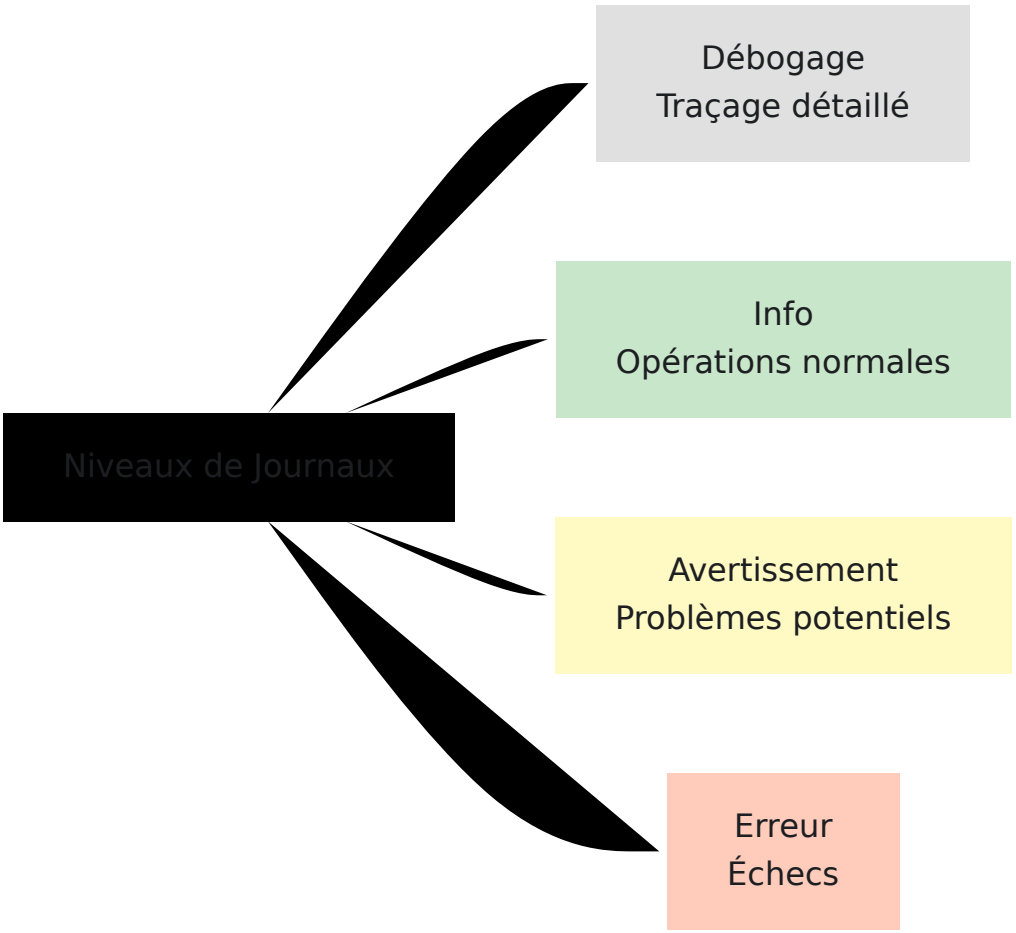
- Taille totale de la base de données et tendances de croissance
- Tailles individuelles des tables et nombres de lignes
- Nombre actuel de connexions à la base de données
- Performance des requêtes et utilisation des ressources

Surveillance des Journaux

Sortie des Journaux

OmniHSS sort les journaux vers **stdout/stderr**, qui doivent être capturés par votre gestionnaire de processus.

Niveaux de Journaux



Modèles de Journaux Clés à Surveiller

Événements des Pairs Diameter :

```
[info] Pair Diameter connecté : mme01.epc.example.com
[warn] Pair Diameter déconnecté : pgw01.epc.example.com
[error] Échec de la connexion au pair Diameter : délai d'attente
```

Événements de Base de Données :

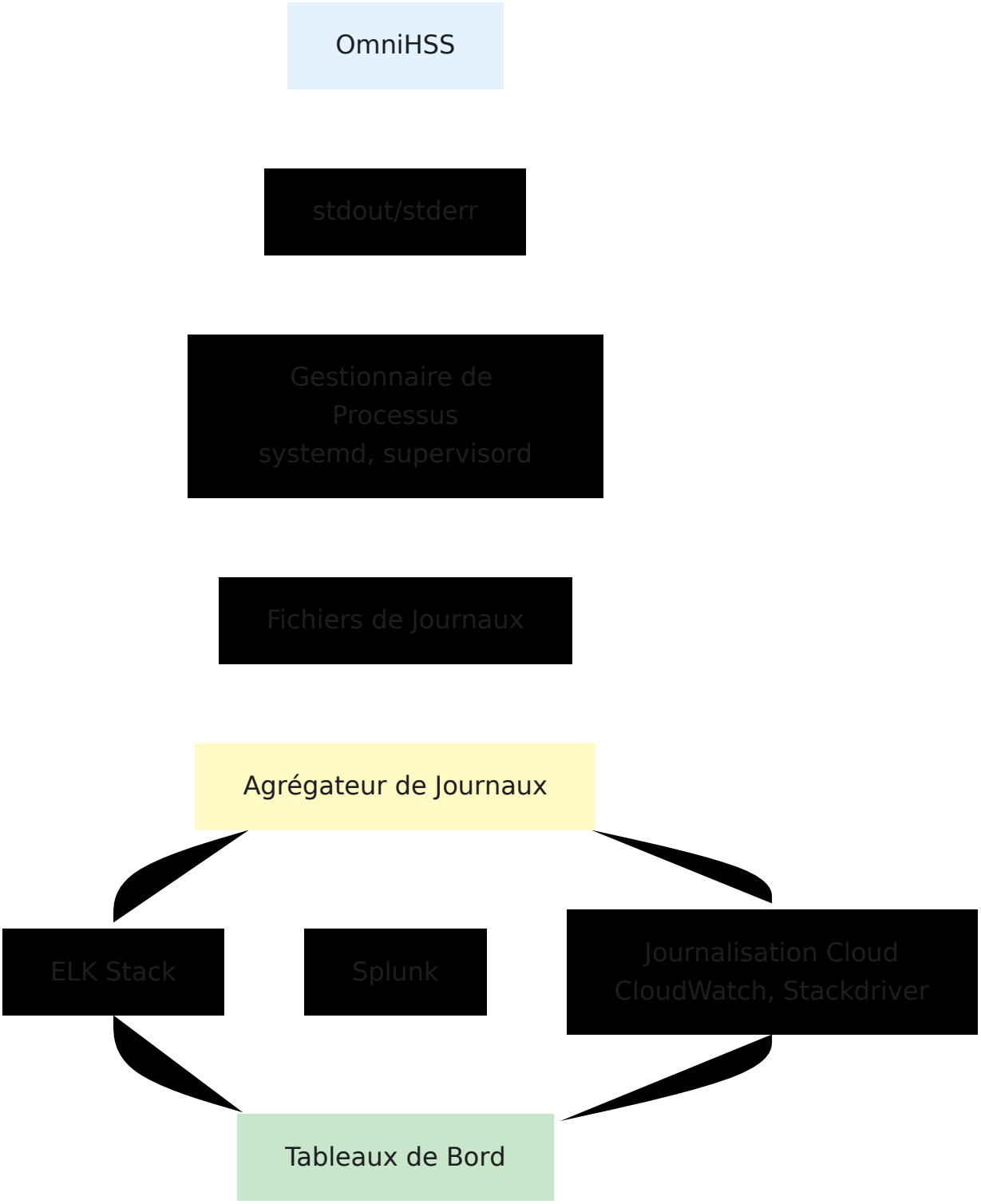
```
[info] Connexion à la base de données établie
[error] Connexion à la base de données perdue : délai d'attente
[error] Échec de la requête de base de données : interblocage
détecté
```

Événements d'Authentification :

```
[info] Authentification réussie : IMSI 001001123456789
[warn] Échec de l'authentification : IMSI 001001123456789, vecteur
invalide
[error] Roaming refusé : IMSI 001001123456789, MCC 310 MNC 410
```

Agrégation des Journaux

Pour les déploiements en production, mettez en œuvre l'agrégation des journaux :


OmniHSS
stdout/stderr
Gestionnaire de Processus
systemd, supervisord
Fichiers de Journaux
Agrégateur de Journaux
ELK Stack
Splunk
Journalisation Cloud
CloudWatch, Stackdriver
Tableaux de Bord

Intégration de la Surveillance Externe

Point de Vérification de Santé

Vérification de Santé de l'API : `GET /api/status`

```
curl -k https://hss.example.com:8443/api/status
```

Réponse Attendue :

```
{"status": "ok"}
```

Statut HTTP : 200 OK

Intégration des Outils de Surveillance

Exemple Nagios/Icinga

```
#!/bin/bash
# check_omnihss.sh

API_URL="https://hss.example.com:8443/api/status"

response=$(curl -k -s -o /dev/null -w "%{http_code}" "$API_URL" --
max-time 5)

if [ "$response" = "200" ]; then
    echo "OK - API OmniHSS répond"
    exit 0
else
    echo "CRITIQUE - API OmniHSS ne répond pas (HTTP $response)"
    exit 2
fi
```

Intégration Prometheus

OmniHSS expose un point de métriques Prometheus natif. Scrapez `http://<host>:9568/metrics`. Le port est défini par `prometheus_metrics_port` et par défaut à `9568`.

Le point inclut des métriques d'exécution BEAM, des métriques de pool de base de données, le statut des pairs Diameter, des compteurs de messages Diameter par application, et le compteur Cx ci-dessous.

Compteur d'Abandon d'Authentification S-CSCF

Métrique : `hss_cx_auth_abandon_total` **Type** : Compteur **Description** : Compte les SAR Cx qui terminent un défi IMS-AKA sans enregistrement réussi. Il s'agit d'un défi abandonné ou échoué. Le S-CSCF envoie un SAR `AUTHENTICATION_TIMEOUT` lorsque son `auth_vector_timeout` expire sur un défi abandonné, ou un SAR `AUTHENTICATION_FAILURE` sur une mauvaise réponse. Voir État d'Assignation S-CSCF. **Étiquettes** :

- `sar_type` - Le Type de Serveur d'Assignation : `authentication_timeout` ou `authentication_failure`.
- `outcome` - `name_cleared` lorsque le nom S-CSCF a été libéré, ou `kept_registered` lorsque un binding enregistré a été préservé. Le HSS libère le nom lorsque l'abonné n'est pas enregistré, et également lorsque le SAR provient d'un S-CSCF différent de celui enregistré. Le deuxième cas signifie que l'I-CSCF a re-forké le REGISTER car il ne pouvait pas atteindre le nœud enregistré, donc ce nom est obsolète.
- `origin_host` - Le S-CSCF qui a envoyé le SAR.

Exemples de requêtes :

```
# Taux de défi abandonné/échoué par seconde
rate(hss_cx_auth_abandon_total[5m])

# Répartition par cause
sum by (sar_type) (rate(hss_cx_auth_abandon_total[5m]))

# Meilleurs S-CSCF par défis abandonnés
topk(5, sum by (origin_host)
(rate(hss_cx_auth_abandon_total[5m])))
```

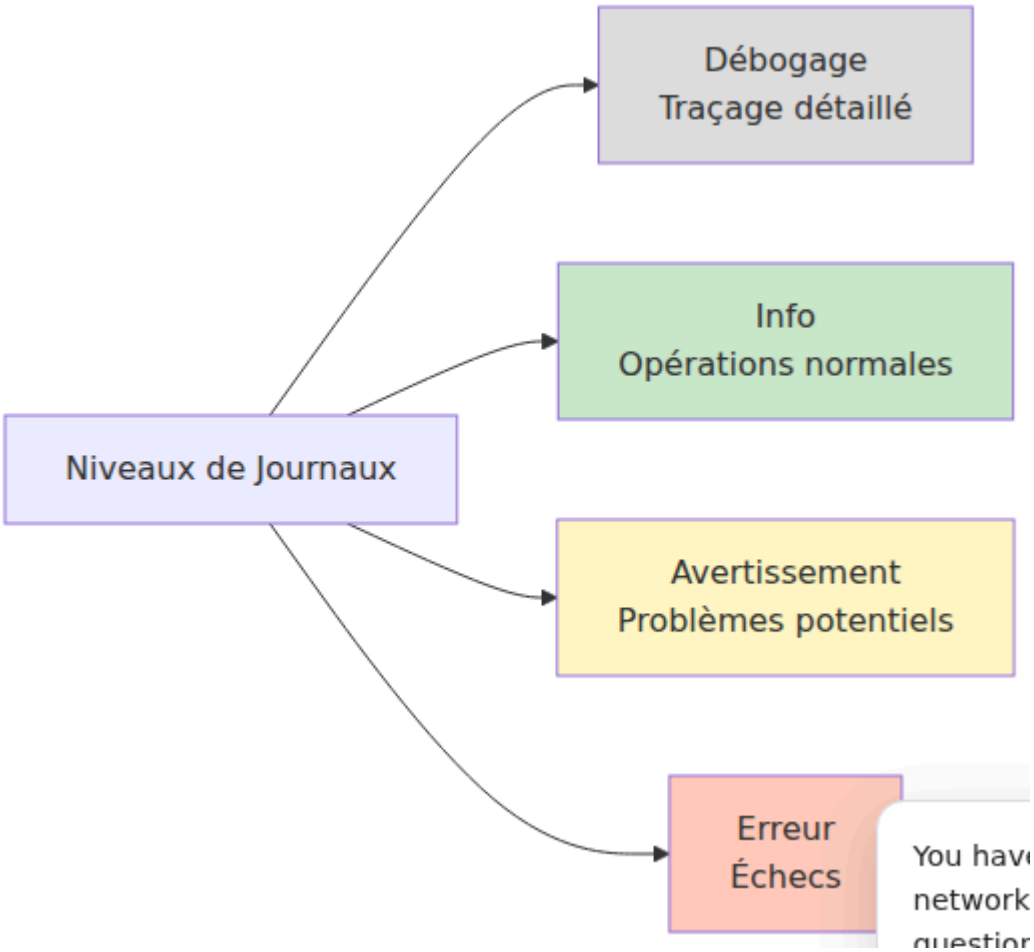
Un taux élevé sur un `origin_host` peut indiquer un S-CSCF qui échoue des défis, ou des abonnés qui abandonnent l'enregistrement sur ce nœud.

Intégration SNMP

Pour la surveillance basée sur SNMP, des scripts d'extension SNMP personnalisés peuvent interroger la base de données ou l'API pour des métriques et retourner des valeurs via des OID SNMP.

Indicateurs Clés de Performance

KPI Opérationnels



Seuils de KPI Recommandés

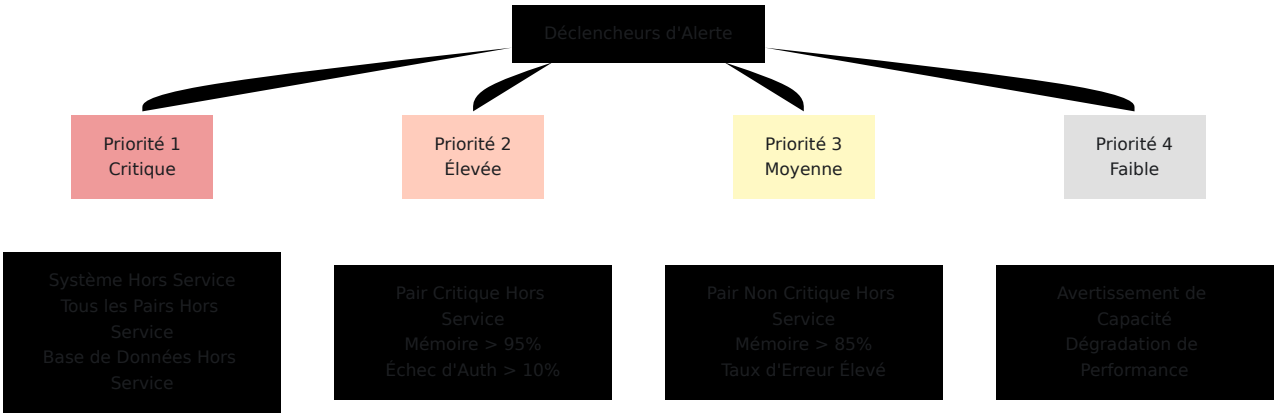
KPI	Cible	Avertissement	Critique
Temps de Fonctionnement du Système	99.99%	< 99.95%	< 99.9%
Temps de Fonctionnement des Pairs Diameter	99.9%	< 99.5%	< 99%
Taux de Réussite d'Authentification	> 99%	< 99%	< 95%
Temps de Réponse Diameter	< 100ms	> 200ms	> 500ms
Temps de Requête de Base de Données	< 50ms	> 100ms	> 500ms
Taux d'Erreur	< 0.1%	> 0.5%	> 1%

KPI de Capacité

Métrique	Surveiller	Planifier l'Action à
Total des Abonnés	Compte actuel	80% de la capacité attendue
Sessions PDN Concurrentes	Sessions actives	70% du maximum attendu
Taille de la Base de Données	Mo utilisés	80% du stockage alloué
Connexions à la Base de Données	Connexions actives	80% de la taille du pool

Stratégies d'Alerte

Priorités des Alertes



Définitions des Alertes

Alertes Critiques (P1)

Système Indisponible :

- Échec de la vérification de santé de l'API
- Panneau de Contrôle inaccessible
- Échec de connexion à la base de données
- Action : Enquête immédiate et escalade

Tous les Pairs Diameter Déconnectés :

- Aucun pair connecté
- Action : Vérifier le réseau, redémarrer si nécessaire

Base de Données Hors Service :

- Impossible de se connecter à la Base de Données SQL
- Action : Enquêter sur le serveur de base de données, redémarrer si nécessaire

Alertes de Haute Priorité (P2)

Pair Critique Hors Service :

- MME principal déconnecté
- P-GW principal déconnecté
- S-CSCF principal déconnecté
- Action : Enquêter sur la connectivité des pairs dans les 15 minutes

Utilisation Élevée de la Mémoire :

- Mémoire > 95%
- Action : Enquêter sur une fuite de mémoire, planifier un redémarrage

Taux d'Échec d'Authentification Élevé :

- > 10% des demandes d'authentification échouent
- Action : Vérifier la provision des abonnés, enquêter sur la cause

Alertes de Moyenne Priorité (P3)

Pair Non Critique Hors Service :

- Pair secondaire déconnecté
- Serveur d'Application déconnecté
- Action : Enquêter dans l'heure

Utilisation Élevée de la Mémoire :

- Mémoire > 85%
- Action : Surveiller la tendance, planifier une mise à niveau de capacité

Taux d'Erreur Élevé :

- Taux d'erreur > 1%
- Action : Examiner les journaux, identifier la cause profonde

Alertes de Faible Priorité (P4)

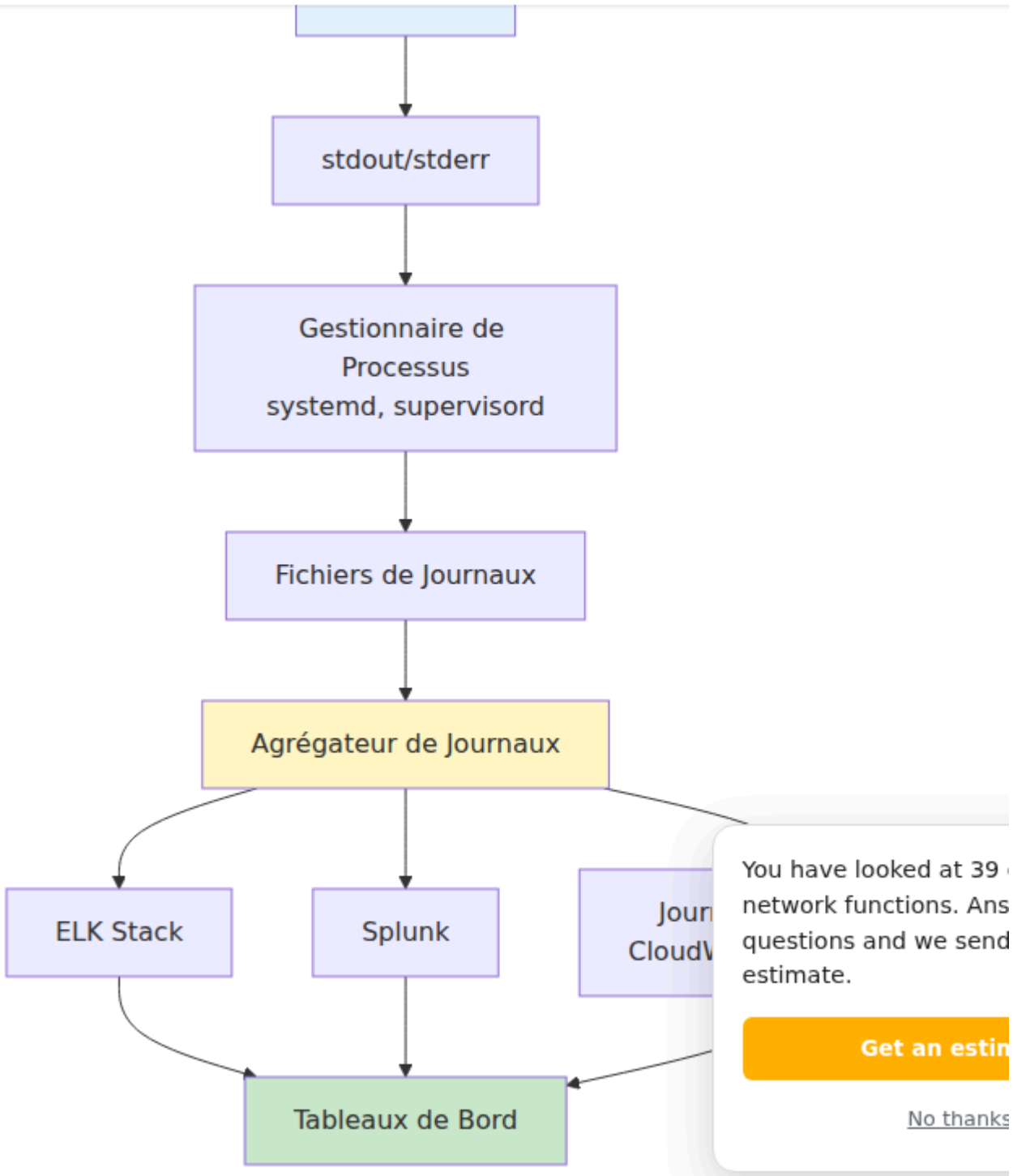
Avertissement de Capacité :

- Abonnés > 80% de la capacité
- Base de Données > 80% du stockage alloué
- Action : Planifier l'expansion de capacité

Dégradation de Performance :

- Temps de réponse élevés mais acceptables
- Action : Surveiller et optimiser les requêtes

Canaux de Notification d'Alerte



You have looked at 39
network functions. Ans
questions and we send
estimate.
Get an estim
No thanks

Liste de Vérification de Surveillance

Vérifications Quotidiennes

- [ ] Examiner l'Aperçu du Panneau de Contrôle - comptes d'abonnés normaux
- [ ] Examiner la page Diameter - tous les pairs critiques connectés
- [ ] Examiner la page Application - mémoire et processus dans les limites
- [ ] Vérifier les journaux d'erreurs - pas d'erreurs critiques dans les dernières 24 heures
- [ ] Vérifier que la sauvegarde a été effectuée avec succès

Vérifications Hebdomadaires

- [ ] Examiner les tendances de capacité - croissance des abonnés
- [ ] Examiner les tendances de performance - temps de réponse
- [ ] Examiner la taille de la base de données - taux de croissance acceptable
- [ ] Examiner les taux d'erreur - identifier les modèles
- [ ] Tester les notifications d'alerte - s'assurer qu'elles fonctionnent

Vérifications Mensuelles

- [ ] Revue de planification de capacité - projet à 6 mois d'avance
- [ ] Revue d'optimisation de performance - identifier les requêtes lentes
- [ ] Revue de sécurité - expiration des certificats, vulnérabilités
- [ ] Revue de documentation - mettre à jour les runbooks
- [ ] Test de récupération après sinistre - vérifier que les sauvegardes se restaurent correctement

---

← Retour au Guide des Opérations | Suivant : Multi-Features →

Fonctionnalités Multi-MSISDN et Multi-IMSI d'OmniHSS

← Retour au Guide des Opérations

Table des Matières

- Aperçu
- Multi-MSISDN : Plusieurs Numéros de Téléphone
- SIM Multi-IMSI : Plusieurs Identités Réseau
- Scénarios Combinés
- Résolution MSISDN à Travers les Interfaces
- Identité Utilisateur Publique Partagée
- Exemples de Configuration
- Procédures Opérationnelles

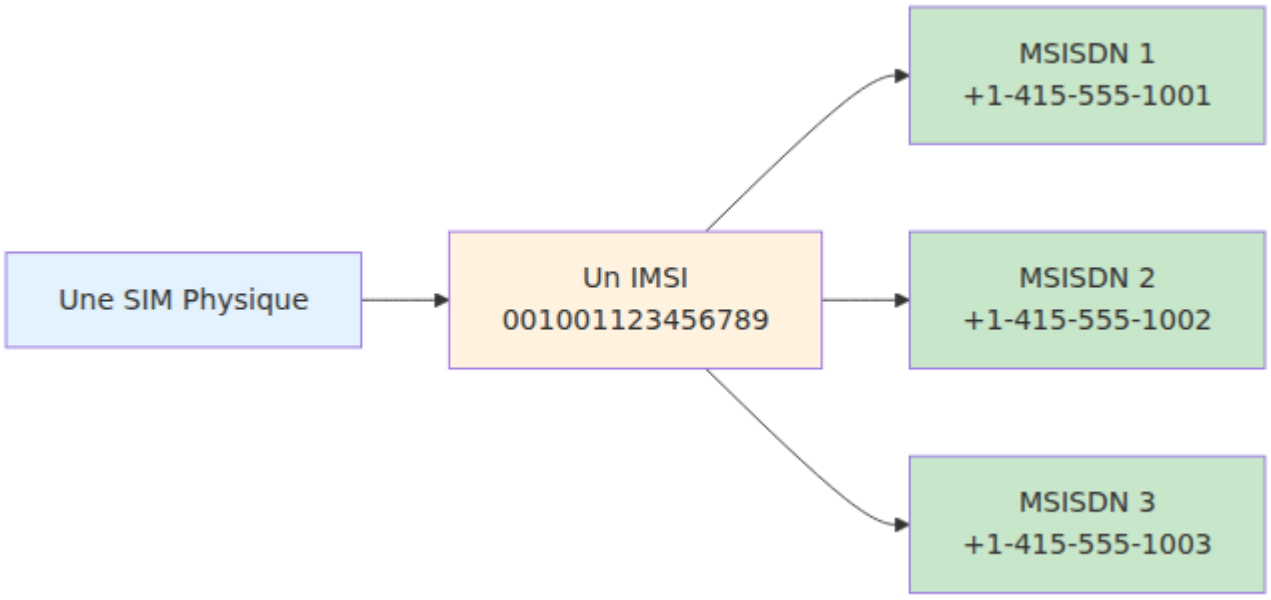
Aperçu

OmniHSS prend en charge des capacités de provisionnement avancées qui permettent des configurations de service flexibles :

Support Multi-MSISDN

Un IMSI → Plusieurs Numéros de Téléphone

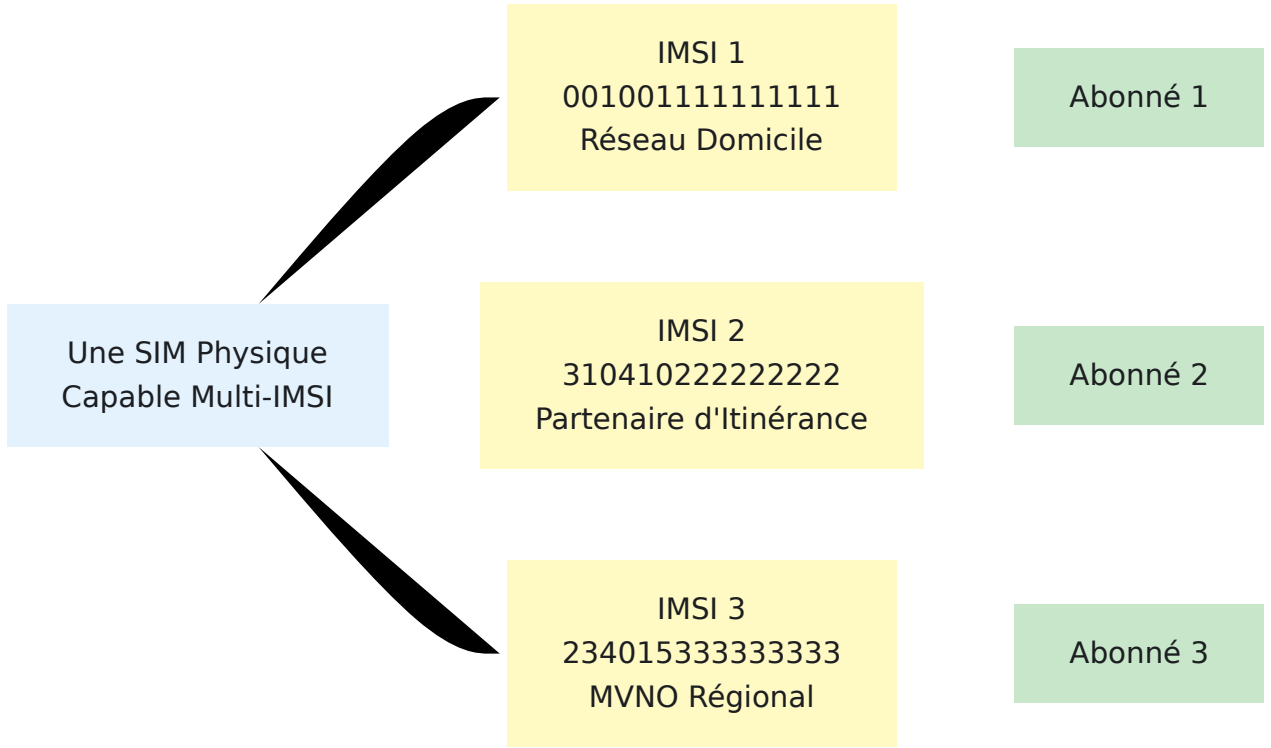
Un seul abonné (identifié par un IMSI) peut avoir plusieurs MSISDN (numéros de téléphone) attribués. Tous les numéros sonnent sur le même appareil et partagent les mêmes profils de service.



Support SIM Multi-IMSI

Une SIM → Plusieurs IMSIs

Une seule carte SIM physique peut contenir plusieurs IMSIs, permettant à l'appareil de se connecter à différents réseaux en utilisant différentes identités réseau. Cela est utile pour l'itinérance internationale et les scénarios MVNO.



Multi-MSISDN : Plusieurs Numéros de Téléphone

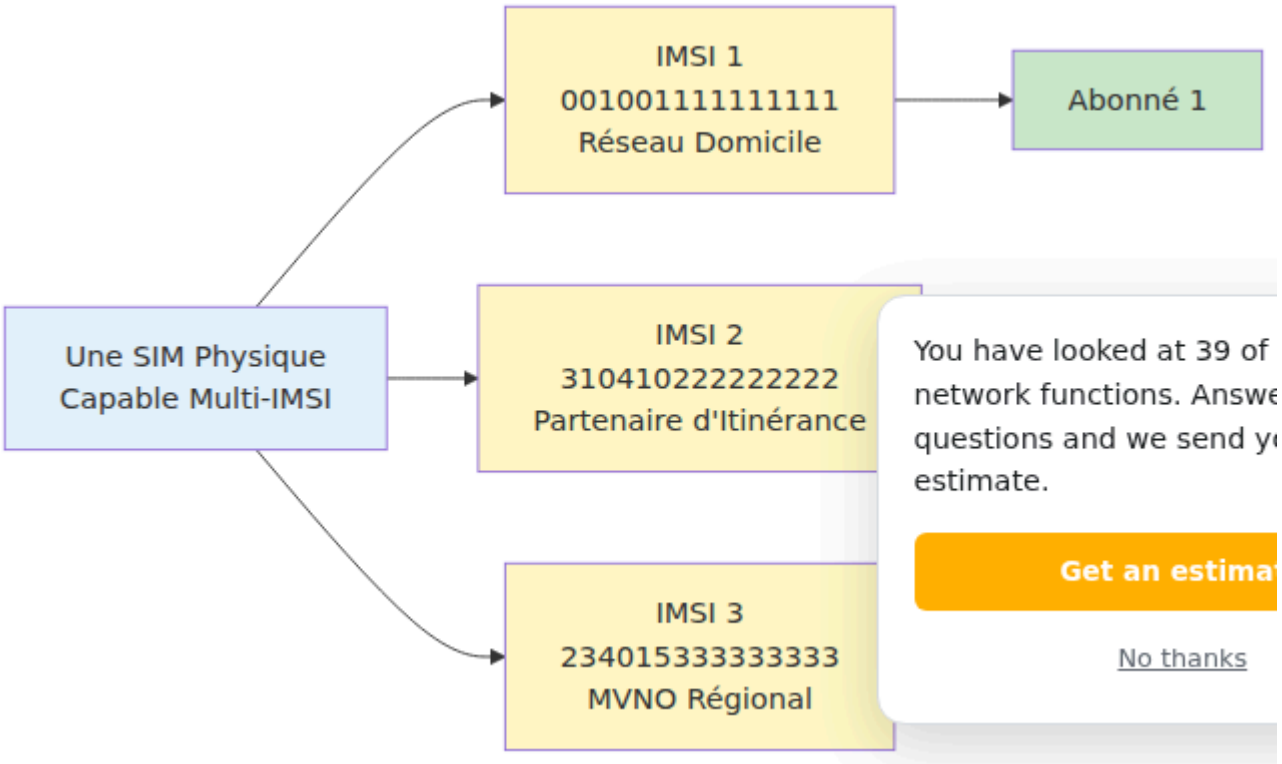
Comment Cela Fonctionne

Un enregistrement d'abonné dans le HSS a plusieurs MSISDN liés par une table de jointure. Lorsque l'abonné s'enregistre au IMS, tous les MSISDN sont inclus dans le profil IMS, permettant aux appels entrants à n'importe quel numéro d'atteindre l'appareil.

Caractéristiques Clés

- **Un IMSI** - L'abonné a un seul IMSI lié à sa carte SIM
- **Plusieurs MSISDNs** - L'abonné peut avoir plusieurs numéros de téléphone
- **Intégration IMS** - Tous les MSISDNs sont enregistrés dans IMS
- **Service Partagé** - Tous les numéros partagent les mêmes profils de service (EPC, IMS, Itinérance)

Modèle de Données

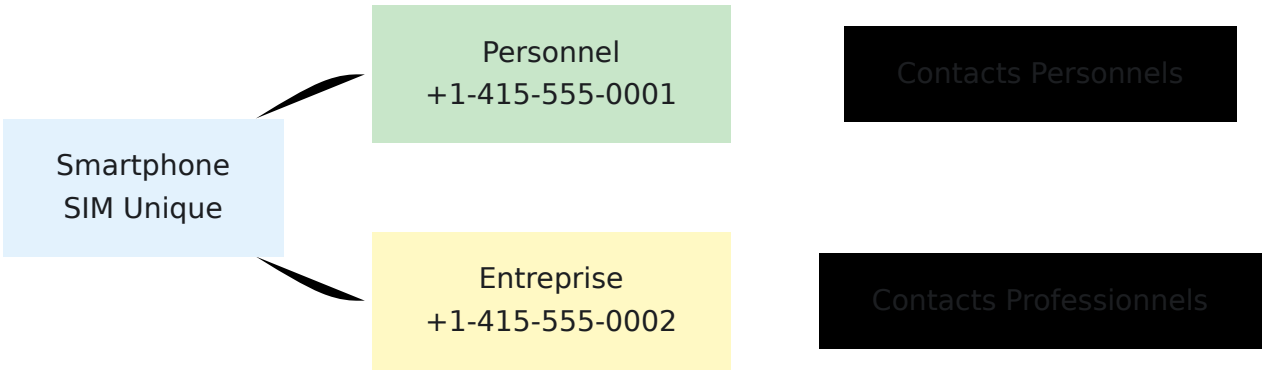


Important : Un MSISDN ne peut être attribué qu'à UN seul abonné à la fois. Cependant, un abonné peut avoir de NOMBREUX MSISDNs.

Cas d'Utilisation

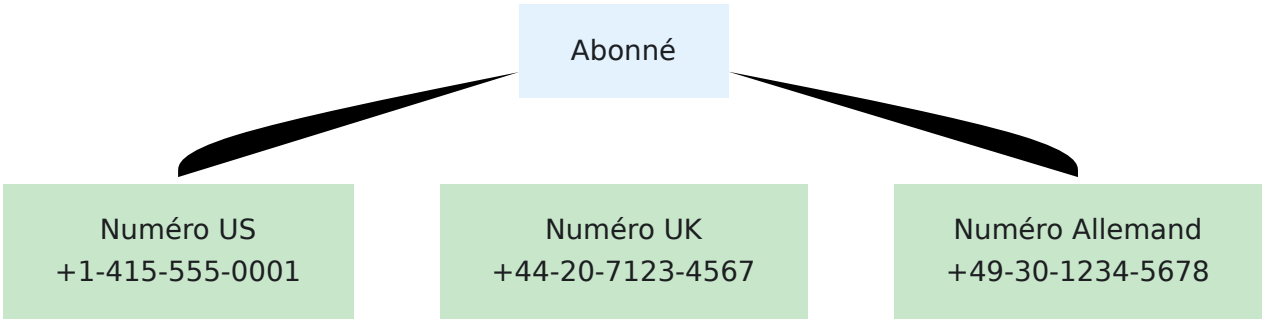
1. Lignes Professionnelles et Personnelles

Un abonné a à la fois des numéros de téléphone professionnels et personnels sur le même appareil :



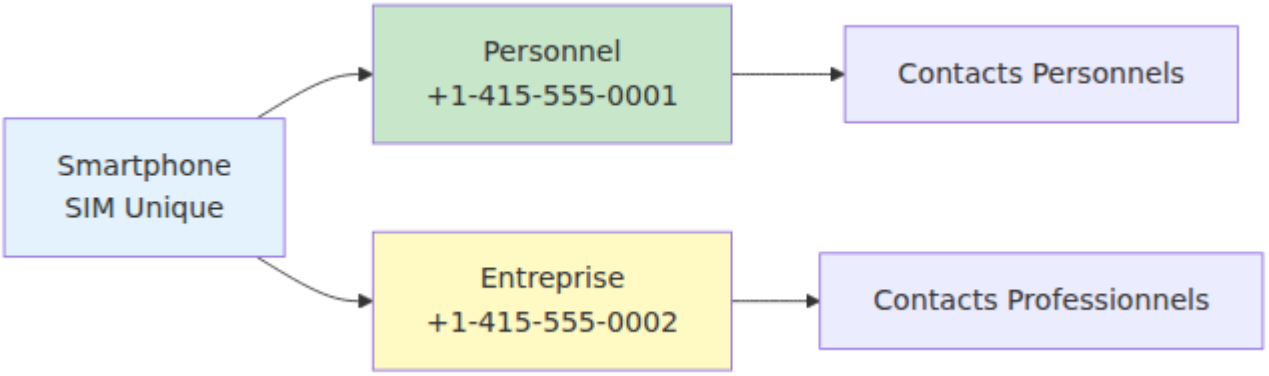
2. Numéros Internationaux

Un abonné qui voyage fréquemment a des numéros dans plusieurs pays :



3. Forfaits Familiaux

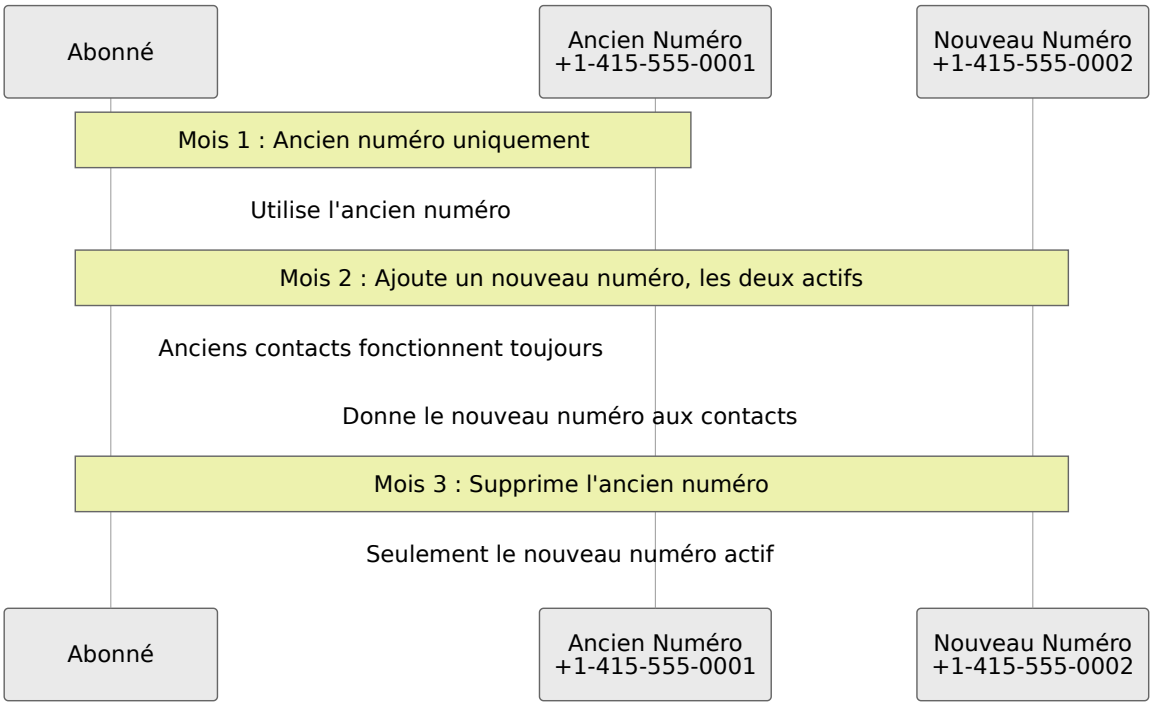
Un parent gère plusieurs numéros de membres de la famille :



Remarque : Dans OmniHSS, cela nécessiterait plusieurs abonnés (un par SIM/IMSI), chacun ayant potentiellement plusieurs MSISDNs.

4. Portage de Lignes Héritées

Lorsqu'un abonné change de numéro mais souhaite garder l'ancien numéro actif pendant la transition :



Configuration

Création de MSISDNs

Les MSISDNs doivent être créés avant d'être attribués aux abonnés.

```
# Créer le premier MSISDN
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{"msisdn": {"msisdn": "14155551001"}}'

# Créer le deuxi��me MSISDN
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{"msisdn": {"msisdn": "14155551002"}}'
```

Attribution de MSISDNs aux Abonnés

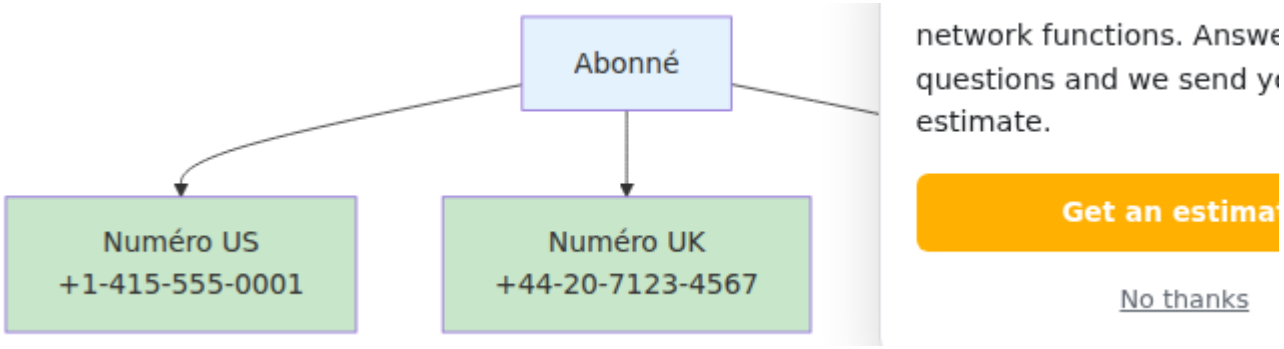
L'attribution se fait par le biais de la table de jointure dans la base de données.

Méthode de Base de Données :

1. Interroger la base de données pour obtenir l'ID de l'abonné pour l'IMSI cible
2. Interroger la base de données pour obtenir les IDs de MSISDN pour les numéros de téléphone
3. Insérer des enregistrements dans la table de jointure liant subscriber_id à chaque msisdn_id

Cela crée la relation plusieurs-à-plusieurs entre l'abonné et ses numéros de téléphone.

Flux de Provisionnement



Vérification de l'Attribution

Interroger la base de données pour récupérer l'abonné avec tous les MSISDNs liés en :

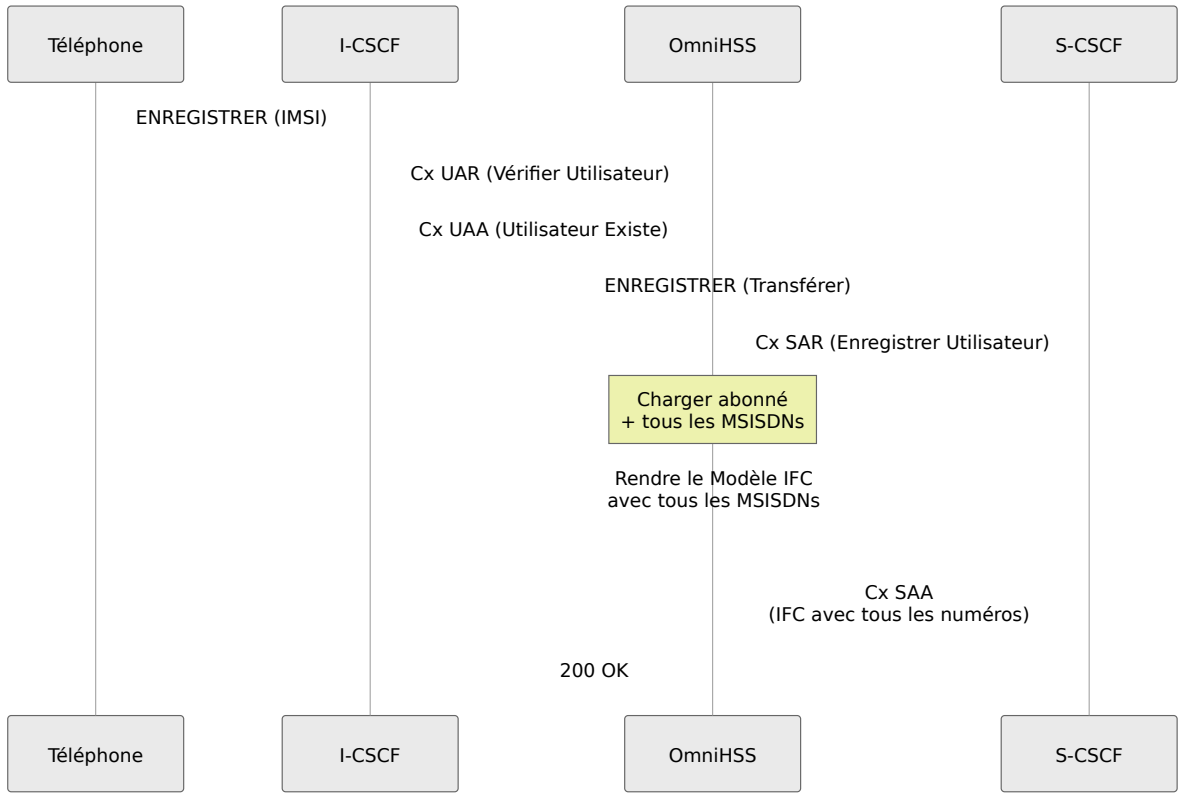
- Joignant la table des abonnés avec la table de jointure
- Joignant la table de jointure avec la table msisdn
- Groupant les résultats par abonné pour voir tous les numéros de téléphone ensemble

Cela affichera l'ID de l'abonné, l'IMSI et une liste de tous les MSISDNs attribués.

Intégration IMS

Enregistrement IMS

Lorsque un abonné s'enregistre au IMS, **tous les MSISDNs attribués sont inclus** dans le profil IMS envoyé au S-CSCF.



Rendu du Modèle IFC

Le modèle IFC IMS peut référencer tous les MSISDNs en utilisant la variable `{{msisdns}}`.

Exemple de Modèle IFC :

```
<ServiceProfile>
  <PublicIdentity>
    <Identity>sip:
{{imsi}}@ims.mnc{{mnc}}.mcc{{mcc}}.3gppnetwork.org</Identity>
  </PublicIdentity>
  <!-- Répéter pour chaque MSISDN -->
  <PublicIdentity>
    <Identity>sip:+14155551001@ims.example.com</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>tel:+14155551001</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>sip:+14155551002@ims.example.com</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>tel:+14155551002</Identity>
  </PublicIdentity>
  <!-- ... -->
</ServiceProfile>
```

Variable de Modèle :

- `{{msisdns}}` - Liste de tous les MSISDNs attribués à l'abonné

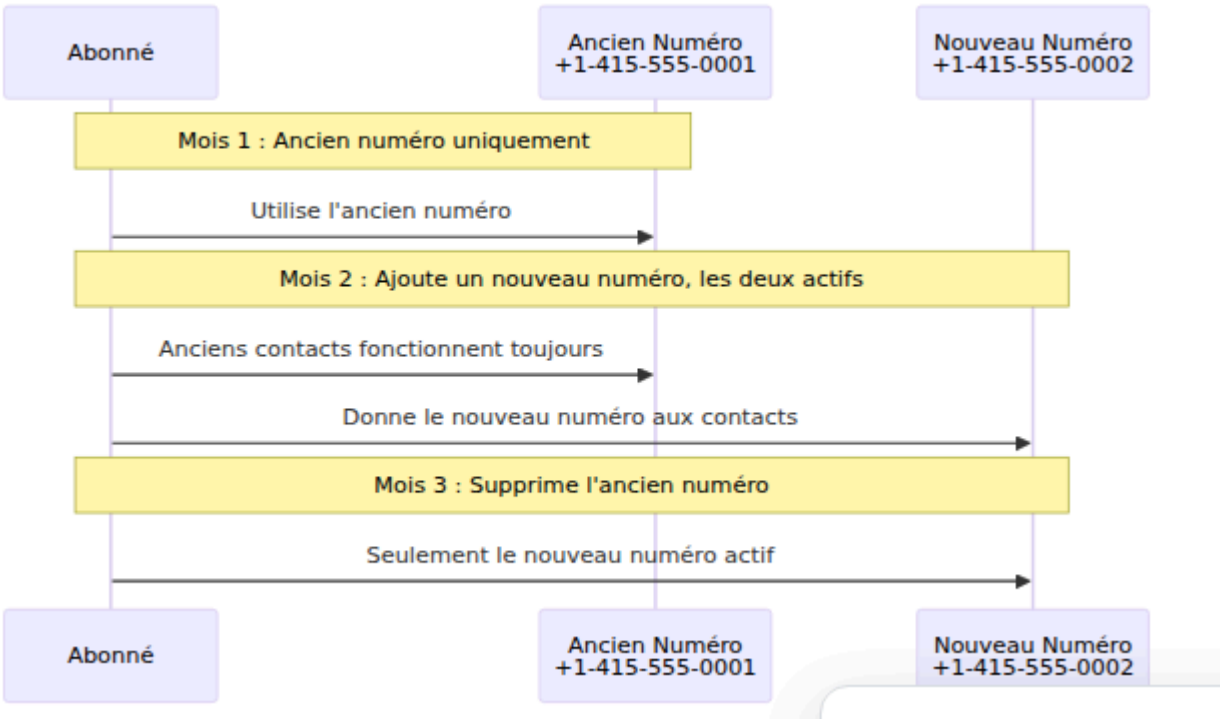
Identités Publiques

Chaque MSISDN entraîne généralement deux identités publiques IMS :



Routage des Appels Entrants

Lorsque quelqu'un appelle l'un des numéros de l'abonné, le réseau IMS route vers le bon URI SIP :



Présentation de l'Appel Sortant

Le téléphone peut choisir quel numéro présenter comme ID d'appelant pour les appels sortants.

Exemple SIP INVITE :

```
INVITE sip:+15105551234@ims.example.com SIP/2.0
From: "+14155551002" <sip:+14155551002@ims.example.com>;tag=123
To: <sip:+15105551234@ims.example.com>
P-Asserted-Identity: <sip:+14155551002@ims.example.com>
```

Les en-têtes `From` et `P-Asserted-Identity` indiquent lequel des numéros de l'abonné est utilisé.

Dépannage Multi-MSISDN

Problème : MSISDN N'apparaît Pas dans l'Enregistrement IMS

Symptômes :

- S-CSCF montre seulement une identité publique
- Les appels au deuxième numéro échouent

Étapes de Dépannage :

1. **Vérifier l'Attribution MSISDN dans la Base de Données :**
    - Interroger la base de données pour récupérer tous les MSISDNs liés à l'IMSI de l'abonné
    - Vérifier la table de jointure pour s'assurer que les relations existent
2. **Vérifier le Modèle de Profil IMS :**
    - Vérifier que le modèle inclut la variable `{{msisdns}}`
    - Confirmer que la syntaxe du modèle est un XML valide
3. **Examiner les Journaux HSS :**
    - Rechercher des messages d'enregistrement IMS (Cx SAR)
    - Vérifier que tous les MSISDNs sont inclus dans la réponse
4. **Tester l'Enregistrement IMS :**
    - Déclencher une nouvelle inscription sur le téléphone
    - Vérifier les journaux S-CSCF pour les identités publiques enregistrées

Problème : Impossible d'Attribuer un MSISDN à un Abonné

Symptômes :

- L'insertion dans la base de données échoue
- Erreur : "Entrée dupliquée" ou "Contrainte de clé étrangère"

Causes Possibles :

1. **MSISDN Déjà Attribué :**
    - Interroger la base de données pour vérifier si le MSISDN est déjà lié à un autre abonné

- **Solution :** Supprimer d'abord l'attribution existante, puis créer la nouvelle attribution

2. **MSISDN N'existe Pas :**

   - Interroger la base de données pour vérifier que l'enregistrement MSISDN existe
   - **Solution :** Créer d'abord l'enregistrement MSISDN via l'API ou l'insertion dans la base de données

Problème : Les Appels à Un Numéro Fonctionnent, L'Autre Ne Fonctionne Pas

Symptômes :

- Les appels au numéro principal fonctionnent
- Les appels au numéro secondaire échouent ou sont mal routés

Étapes de Dépannage :

1. **Vérifier les Deux Numéros dans l'Enregistrement IMS :**

   - Vérifier les identités publiques enregistrées S-CSCF
   - Confirmer que les deux URI SIP sont présents

2. **Vérifier les Règles de Routage IMS :**

   - Vérifier que les règles de routage du modèle IFC s'appliquent à toutes les identités
   - Vérifier si un numéro spécifique nécessite un routage spécial

3. **Tester les Deux Numéros :**

```
# Tester depuis un client SIP
sip:+14155551001@ims.example.com  # Devrait fonctionner
sip:+14155551002@ims.example.com  # Devrait aussi fonctionner
```

Problème : La Recherche API par MSISDN Renvoie le Mauvais Abonné

Symptômes :

- La requête API `/api/subscriber/msisdn/:msisdn` renvoie un abonné inattendu

Vérification :

Interroger la base de données pour trouver quel abonné le MSISDN est attribué. Cela devrait renvoyer exactement un abonné. S'il renvoie plusieurs ou le mauvais abonné, la table de jointure a des données incorrectes qui doivent être corrigées.

Meilleures Pratiques

Ordre de Provisionnement

1. Créer tous les MSISDNs d'abord
2. Créer l'abonné
3. Attribuer les MSISDNs à l'abonné
4. Vérifier l'attribution avant l'activation

Gestion des MSISDN

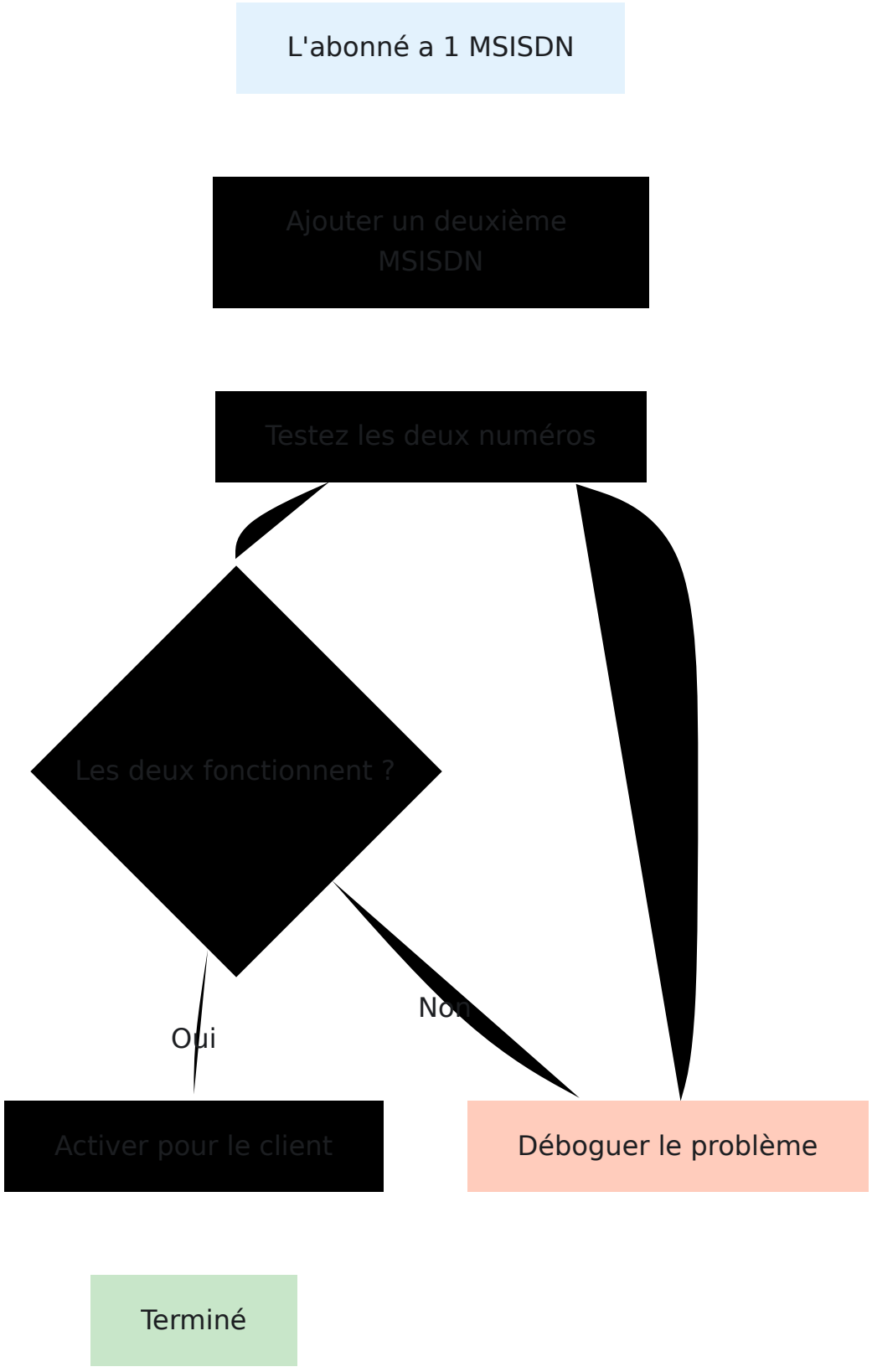
- **Documenter les numéros principaux vs secondaires** dans les custom_attributes de l'abonné
- **Porter les numéros séquentiellement** lors du portage pour éviter les interruptions de service
- **Tester tous les numéros** après le provisionnement avant de les donner au client

Configuration IMS

- S'assurer que le modèle IFC gère correctement plusieurs identités publiques
- Tester le routage entrant vers tous les numéros
- Vérifier la présentation de l'ID d'appel pour les appels sortants

Migration

Lors de la migration d'un MSISDN unique à multi-MSISDN :


L'abonné a 1 MSISDN
Ajouter un deuxième MSISDN
Testez les deux numéros
Les deux fonctionnent ?
Oui
Non
Activer pour le client
Déboguer le problème
Terminé

SIM Multi-IMSI : Plusieurs Identités Réseau

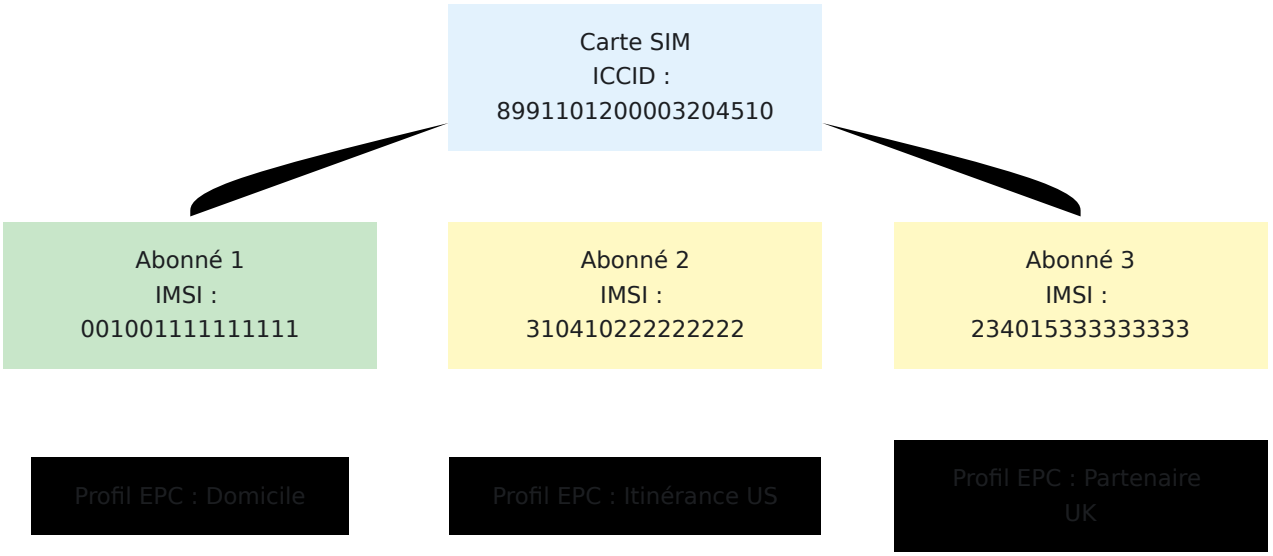
Comment Cela Fonctionne

Une SIM multi-IMSI contient plusieurs profils d'abonnés complets, chacun avec son propre IMSI, clés et identifiants. L'appareil peut passer d'IMSI en IMSI pour se connecter à différents réseaux, souvent automatiquement en fonction de l'emplacement ou de la disponibilité du réseau.

Important : Un **seul IMSI peut être actif à tout moment**. Lorsqu'un appareil passe à un IMSI différent sur la même carte SIM, le HSS désinscrit automatiquement l'IMSI précédemment actif.

Mise en Œuvre d'OmniHSS

Dans OmniHSS, chaque IMSI sur une SIM multi-IMSI est provisionné comme un **enregistrement d'abonné séparé**, mais tous font référence à la **même carte SIM** :



Cas d'Utilisation

1. Optimisation de l'Itinérance Internationale

- IMSI Domicile : 001-001 (tarifs réseau domicile)
- IMSI Itinérance US : 310-410 (tarifs locaux US)
- IMSI Itinérance UE : 234-015 (tarifs locaux UE)
- L'appareil change d'IMSI en fonction de l'emplacement

2. Service MVNO

- IMSI Principal : réseau MVNO (revendeur)
- IMSI de Repli : réseau hôte (opérateur parent)
- Basculement automatique si la couverture MVNO n'est pas disponible

3. IoT/M2M Multi-Réseau

- IMSI 1 : Opérateur principal
- IMSI 2 : Opérateur de secours pour redondance
- IMSI 3 : Repli d'urgence/coût réduit
- Les appareils critiques maintiennent la connectivité

4. Conformité Réglementaire

- Différents IMSIs pour différentes zones réglementaires
- Se conformer aux exigences locales de résidence des données
- Utiliser une identité réseau locale par juridiction

Fonctionnalités Multi-IMSI

Authentification Indépendante

- Chaque IMSI a son propre Ki, OPC et ensemble de clés
- Vecteurs d'authentification séparés par IMSI
- Différents identifiants de sécurité par réseau

Profils de Service Séparés

- Différents profils EPC (bande passante, APNs)
- Différents profils IMS (services vocaux)
- Différentes règles d'itinérance par IMSI

Identité Physique Partagée

- Tous les IMSIs font référence à la même SIM (via sim_id)
- Même ICCID à travers tous les enregistrements d'abonnés
- Regroupement logique via la carte SIM

Sélection de Réseau

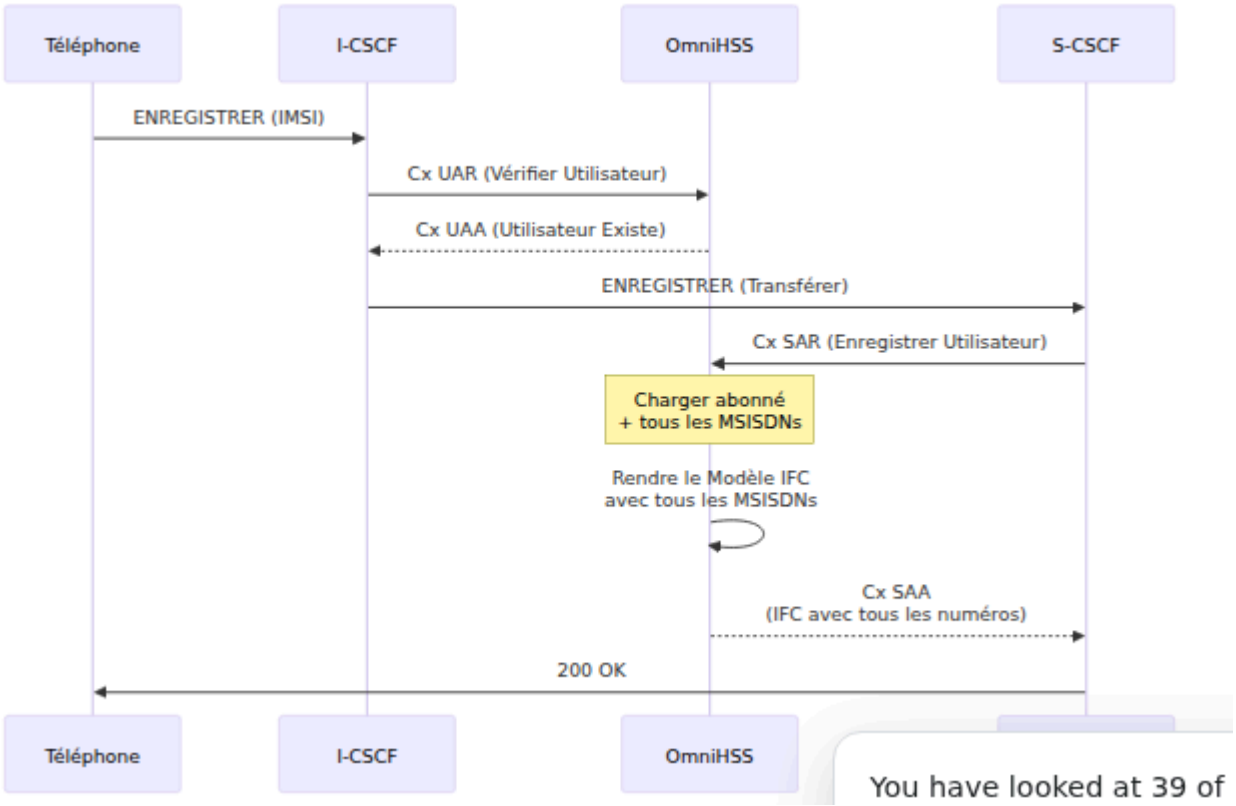
- L'appareil ou la carte SIM décide quel IMSI utiliser
- Basé sur les réseaux disponibles, l'emplacement, la politique
- Le HSS authentifie quel que soit l'IMSI que l'appareil présente

Configuration

```
# 1. Créer la carte SIM (capable multi-IMSI)
SIM_ID=$(curl -k -X POST https://hss.example.com:8443/api/sim \
  -d '{"sim": {"iccid": "8991101200003204510", "is_esim": false}}'
\
  | jq -r '.data.id')

# 2. Créer l'ensemble de clés pour l'IMSI 1 (réseau domicile)
KEYSET1=$(curl -k -X POST https://hss.example.com:8443/api/key_set
\
  -d '{"key_set": {"ki": "0123456789ABCDEF...", "opc":
"FEDCBA9876..."}}' \
  | jq -r '.data.id')

# 3. Créer l'abonné 1 (IMSI domicile)
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -d "{\"subscriber\": {
    \"imsi\": \"001001111111111\",
    \"sim_id\": \"$SIM_ID\",
    \"key_set_id\": \"$KEYSET1\",
    \"epc_profile_id\": \"e58ed763-928c-4155-bee9-fdbaaadc15f3\"
  }}"

# 4. Créer l'ensemble de clés pour l'IMSI 2 (partenaire
d'itinérance)
KEYSET2=$(curl -k -X POST https://hss.example.com:8443/api/key_set
\
  -d '{"key_set": {"ki": "1111111111111111...", "opc":
"2222222222..."}}' \
  | jq -r '.data.id')

# 5. Créer l'abonné 2 (IMSI d'itinérance)
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -d "{\"subscriber\": {
    \"imsi\": \"310410222222222\",
    \"sim_id\": \"$SIM_ID\",
    \"key_set_id\": \"$KEYSET2\",
    \"epc_profile_id\": \"f47ac10b-58cc-4372-a567-0e02b2c3d479\"
  }}"

# 6. Répéter pour d'autres IMSIs sur la SIM...
```

Flux d'Authentification

Lorsque un appareil multi-IMSI se connecte :



Le HSS n'a pas besoin de savoir qu'il s'agit d'une SIM multi-IMSI. Il authentifie quel que soit l'IMSI que l'appareil présente.

Changement d'IMSI et Désinscription Automatique

Lorsqu'une SIM multi-IMSI passe d'un IMSI à un autre, un seul IMSI peut être enregistré à la fois sur le réseau. OmniHSS gère automatiquement cela en envoyant une **Demande de Désinscription de Localisation (CLR)** pour désinscrire l'IMSI précédemment actif lorsqu'un nouvel IMSI de la même carte SIM s'enregistre.

Règle de l'IMSI Actif Unique

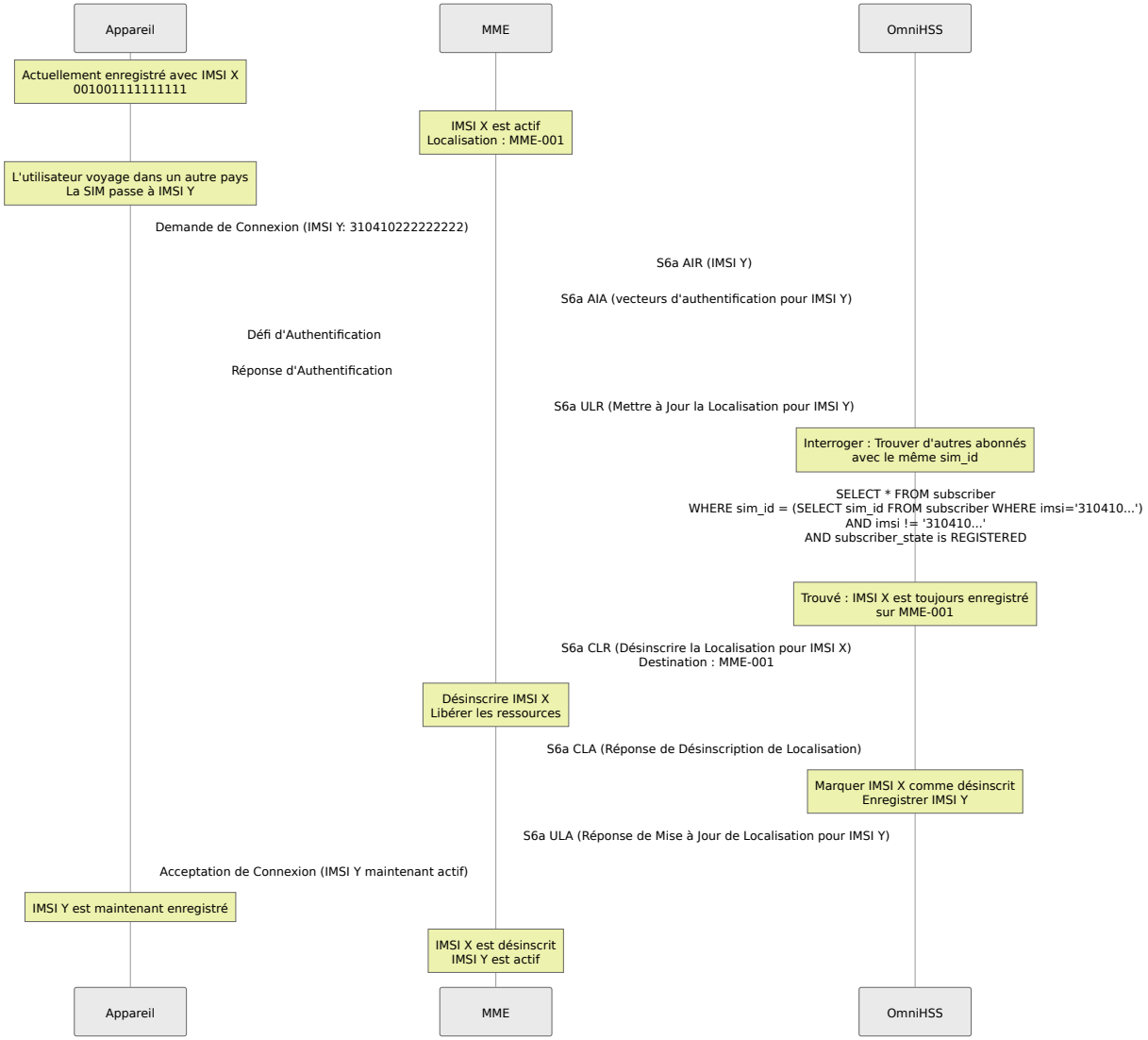
Concept Clé : Un seul abonné (IMSI) par carte SIM peut être actif à tout moment.

- Si un abonné est enregistré sur un MME avec **IMSI X**

- Et que le HSS reçoit une Demande de Mise à Jour de Localisation pour **IMSI Y** (sur la même SIM que l'IMSI X)
- Le HSS envoie automatiquement une **Demande de Désinscription de Localisation** pour désinscrire **IMSI X**

Cela garantit un passage propre entre les IMSIs et empêche les conflits dans le réseau.

Flux de Changement d'IMSI



Pourquoi cela importe

Intégrité du réseau :

- Empêche les enregistrements en double à partir de la même carte SIM physique

- Assure que les ressources du réseau sont correctement libérées
- Maintient des données de localisation précises des abonnés

Précision de la facturation :

- Un seul IMSI est facturé pour l'accès au réseau à la fois
- Limites de session claires entre les changements d'IMSI
- Génération précise de CDR (Call Detail Record)

Gestion des ressources :

- Les ressources MME pour l'ancien IMSI sont libérées
- Les contextes PDP et les porteurs sont nettoyés
- Le suivi de localisation reste précis

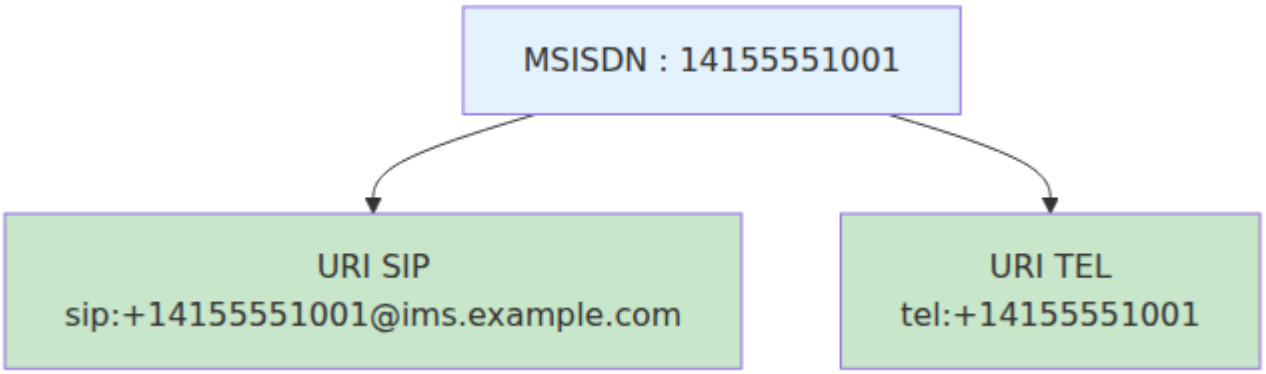
Déclencheurs de changement d'IMSI

L'appareil/SIM décide quand changer d'IMSI en fonction de :

1. **Disponibilité du réseau**
    - Réseau IMSI domestique non disponible
    - Changement vers l'IMSI du partenaire en itinérance
2. **Sélection manuelle**
    - L'utilisateur sélectionne manuellement le réseau
    - La SIM change vers l'IMSI correspondant
3. **Basé sur des politiques**
    - La carte SIM a des règles internes (par exemple, préférer l'IMSI local dans certains pays)
    - Changement automatique basé sur MCC/MNC
4. **Optimisation des coûts**
    - Changement vers l'IMSI avec des tarifs d'itinérance plus bas
    - Utiliser l'IMSI local pour éviter les frais d'itinérance

Considérations IMS

Le même comportement de demande d'annulation de localisation s'applique à l'enregistrement IMS :



Impact opérationnel

Pour le personnel des opérations :

1. **L'abonné apparaît hors ligne :** Lorsque l'IMSI change, l'ancien IMSI apparaîtra comme "désenregistré" dans le HSS. C'est un comportement normal.

2. **Deux enregistrements d'abonné pour une SIM :** Les SIMs multi-IMSI auront plusieurs enregistrements d'abonné partageant le même `sim_id`. Un seul sera dans un état "enregistré" à la fois.

3. **Suivi de localisation :** La table `subscriber_state` suit avec quel MME/SGSN chaque IMSI est enregistré. Lorsque l'IMSI change, l'ancienne localisation est effacée.

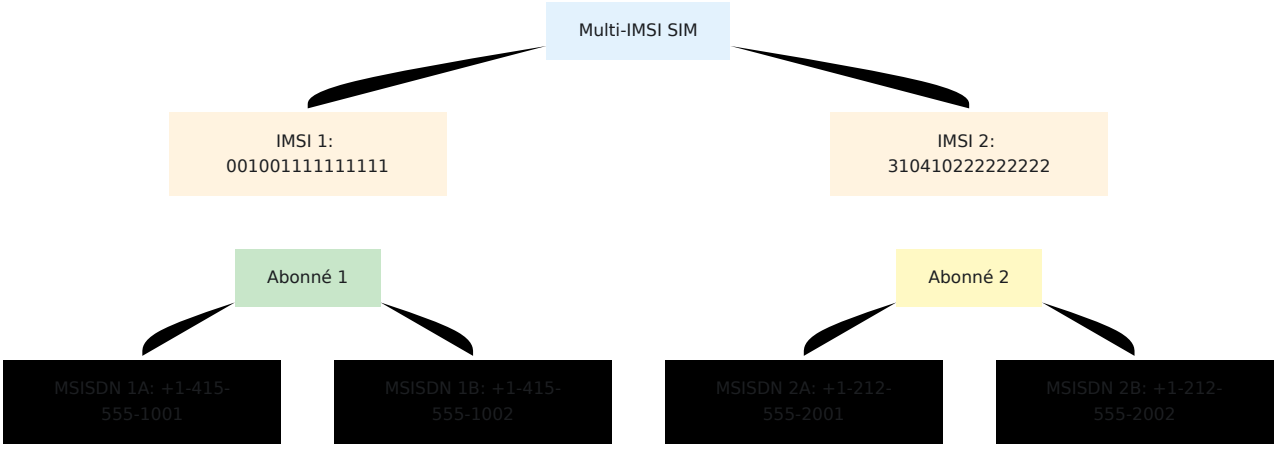
4. **Dépannage :** Si un appareil ne peut pas être atteint :

    - Vérifiez quel IMSI est actuellement enregistré
    - Vérifiez que le bon IMSI est utilisé pour le réseau actuel
    - Confirmez qu'un seul IMSI par SIM est dans un état enregistré

Scénarios combinés

Multi-IMSI + Multi-MSISDN

Vous pouvez combiner les deux fonctionnalités : plusieurs IMSIs sur une SIM, chacune avec plusieurs MSISDNs.



Exemple de cas d'utilisation :

- **Réseau domestique (IMSI 1) :**
  - Numéro personnel : +1-415-555-1001
  - Numéro professionnel : +1-415-555-1002
- **Réseau d'itinérance aux États-Unis (IMSI 2) :**
  - Numéro personnel : +1-212-555-2001
  - Numéro professionnel : +1-212-555-2002

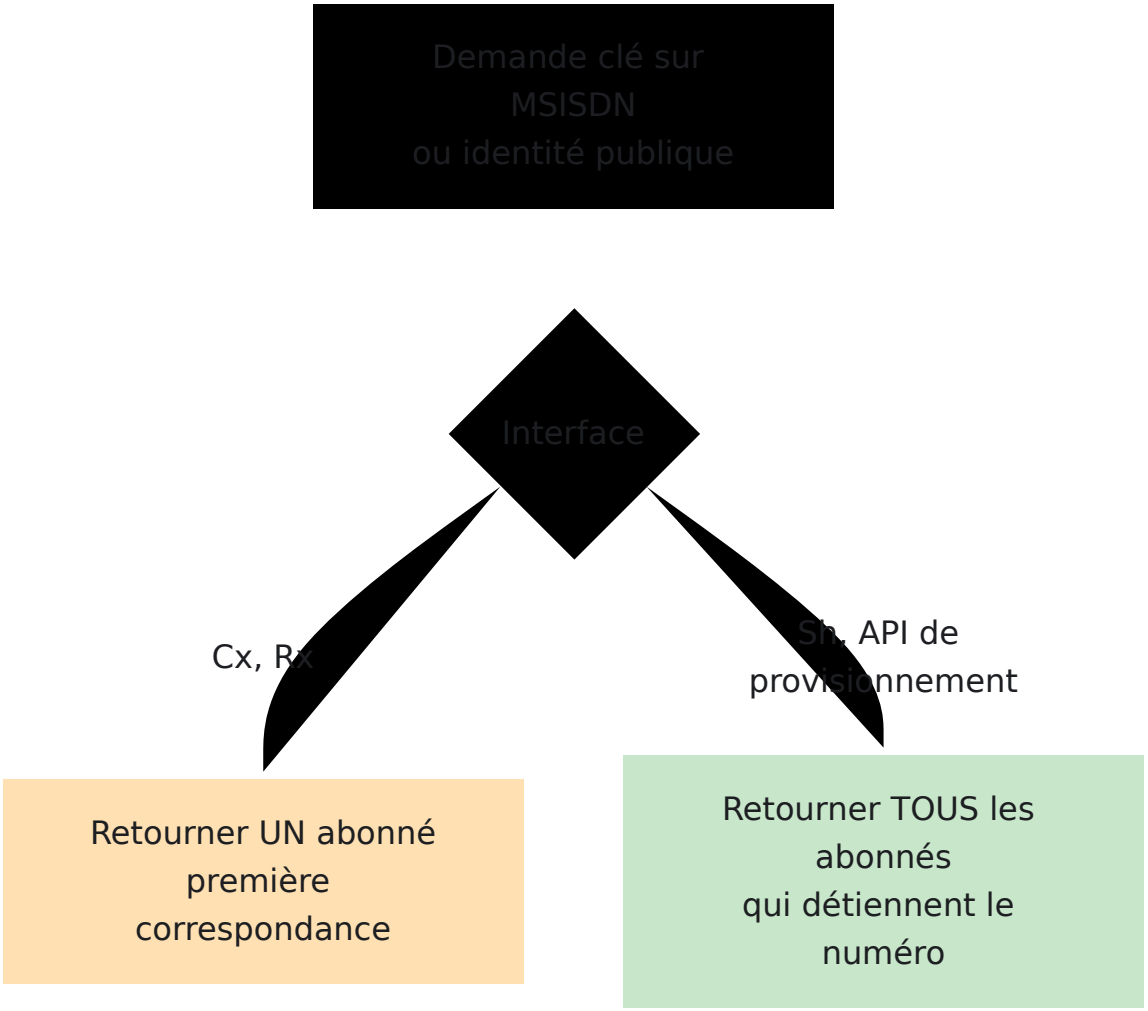
Lorsque l'appareil est sur le territoire domestique, il utilise l'IMSI 1 avec ses MSISDNs. Lorsqu'il est en itinérance aux États-Unis, il change vers l'IMSI 2 avec des MSISDNs différents optimisés pour le réseau américain.

Résolution MSISDN à travers les interfaces

Un MSISDN est une entité partagée dans le modèle de données. Un abonné peut détenir plusieurs MSISDNs (un IMSI, plusieurs numéros). Un MSISDN peut être détenu par plusieurs abonnés (plusieurs IMSIs, un numéro, par exemple un téléphone et une montre connectée). Voir les entités Abonné et MSISDN pour le modèle de données.

Parce qu'un numéro n'est pas toujours unique à un abonné, chaque interface doit décider comment résoudre une demande qui arrive avec une clé sur un MSISDN ou sur une identité publique qui contient un MSISDN. OmniHSS utilise deux règles de résolution différentes, et vous devez savoir quelle règle une interface utilise.

Deux règles de résolution



- **Retourner un.** La recherche retourne un seul abonné (la première correspondance). Les interfaces **Cx** et **Rx** utilisent cette règle.
- **Retourner tous.** La recherche retourne chaque abonné qui détient le numéro. L'interface **Sh** et l'API de provisionnement utilisent cette règle.

Comportement par interface

Interface	Clé sur	Multi-MSISDN par IMSI	Multi-IMSI par MSISDN (numéro partagé)
Cx UAR, MAR	Identité Privée (IMSI)	Correct. L'IMSI est exact.	Correct. Chaque IMSI est séparé.
Cx SAR	Identité Privée, ou Identité Publique	Correct. L'IFC liste chaque MSISDN de l'abonné.	Résout à un abonné si clé sur le numéro partagé. Utilisez l'Identité Privée (IMSI) pour sélectionner un abonné spécifique.
Cx LIR	Identité Publique	Correct.	Résout à un abonné **enregistré** (celui avec un S-CSCF assigné) lorsque le numéro est partagé, donc une demande de terminaison

Interface	Clé sur	Multi-MSISDN par IMSI	Multi-IMSI par MSISDN (numéro partagé)
			retourne un nœud de service atteignable.
Sh UDR, PUR, SNR	Identité Utilisateur (MSISDN ou Identité Publique)	Correct. Les données sont par abonné.	Retourne tous les abonnés qui détiennent le numéro. Un UDR agrège leurs données. Un PUR les met à jour tous.
Rx AAR	Abonné de la session de données	L'abonné peut avoir plusieurs numéros, mais la réponse porte un MSISDN comme Subscription-Id (le protocole permet un).	Correct. Le numéro est sélectionné à partir de la session de données, pas par recherche inversée.

Interface	Clé sur	Multi-MSISDN par IMSI	Multi-IMSI par MSISDN (numéro partagé)
API `GET /subscriber/msisdn/:msisdn`	MSISDN	Retourne l'abonné.	Retourne tous les abonnés qui détiennent le numéro.

Ce que cela signifie en pratique

- **L'enregistrement et l'authentification sont toujours corrects.** Le chemin d'enregistrement IMS (Cx MAR, SAR, UAR) est clé sur l'Identité Privée, qui contient l'IMSI. Chaque appareil s'enregistre avec son propre IMSI, donc un téléphone et une montre connectée qui partagent un numéro s'enregistrent et s'authentifient indépendamment.
- **Un abonné conserve tous ses numéros.** Pour un IMSI avec plusieurs MSISDNs, l'User-Data de Cx SAR liste chaque numéro comme une identité publique (`sip:` et `tel:`).
- **Un numéro partagé résout à un abonné sur Cx et à tous les abonnés partageant sur Sh.** C'est la différence de résolution dans le tableau ci-dessus. Sur le chemin de terminaison (Cx LIR), le HSS préfère un abonné partageant **enregistré**, donc il retourne le S-CSCF de service d'un appareil qui est atteignable. Comment l'appel est ensuite livré aux appareils enregistrés est une question de cœur IMS, pas de HSS.
- **Pour agir sur un abonné spécifique, utilisez l'IMSI ou l'ICCID.** Une recherche inversée par numéro n'est pas unique lorsque le numéro est partagé.

Spécifications

Les deux modèles ont un statut de norme différent. Sachez lequel vous utilisez.

Multi-MSISDN par IMSI est standardisé. 3GPP définit plus d'un numéro pour une seule souscription :

- **Option de numérotation multiple** (3GPP TS 23.008, clause 2.1.3). Le HLR/HSS stocke un MSISDN de base plus des MSISDN supplémentaires. Un indicateur de MSISDN de base marque le numéro que le VLR et le SGSN utilisent.
- **MSISDN supplémentaire (A-MSISDN)** (3GPP TS 23.003, clause 18.9). Un A-MSISDN est "assigné à un utilisateur avec une souscription PS en plus des MSISDN déjà assignés" et "doit être assigné de manière unique par IMSI".

Multi-IMSI par MSISDN est un modèle d'implémentation, pas une norme réseau unique. Il est réalisé sur la SIM par un applet multi-IMSI qui stocke plusieurs IMSIs et change entre eux (terme de l'industrie : Multiple-IMSI-Multiple-MSISDN, MIMM). Les directives GSMA connexes sont GSMA IR.73 Steering of Roaming, et pour eSIM les spécifications de profil SGP.2x/SGP.32. Du côté du réseau, 3GPP ne standardise pas un MSISDN à travers plusieurs souscriptions en tant que service ; au lieu de cela, il le contraint :

- **MSISDN de corrélation (C-MSISDN)** (3GPP TS 23.003, clause 18.7). Un C-MSISDN "est assigné de manière unique par IMSI et Identité Utilisateur Privée IMS". Seul un MSISDN "qui n'est pas partagé par plus d'une Identité Utilisateur Privée IMS" peut servir de C-MSISDN. Donc un numéro partagé ne peut pas être utilisé pour la corrélation de session SRVCC. Planifiez SRVCC et CS fallback avec un numéro par IMSI, pas le numéro partagé.

Spécifications d'interface :

- Cx (MAR, SAR, UAR, LIR) : 3GPP TS 29.228 et 3GPP TS 29.229.
- Sh (UDR, PUR, SNR) : 3GPP TS 29.328 et 3GPP TS 29.329.

Modèle	Statut	Référence
Multi-MSISDN par IMSI	Standardisé	TS 23.008 §2.1.3 (numérotation multiple) ; TS 23.003 §18.9 (A-MSISDN, unique par IMSI)
Multi-IMSI par MSISDN	Modèle d'applet SIM (MIMM) ; contraint sur le réseau	GSMA IR.73 ; TS 23.003 §18.7 (un MSISDN partagé ne peut pas être un C-MSISDN)

Identité Utilisateur Publique Partagée (un numéro, plusieurs appareils)

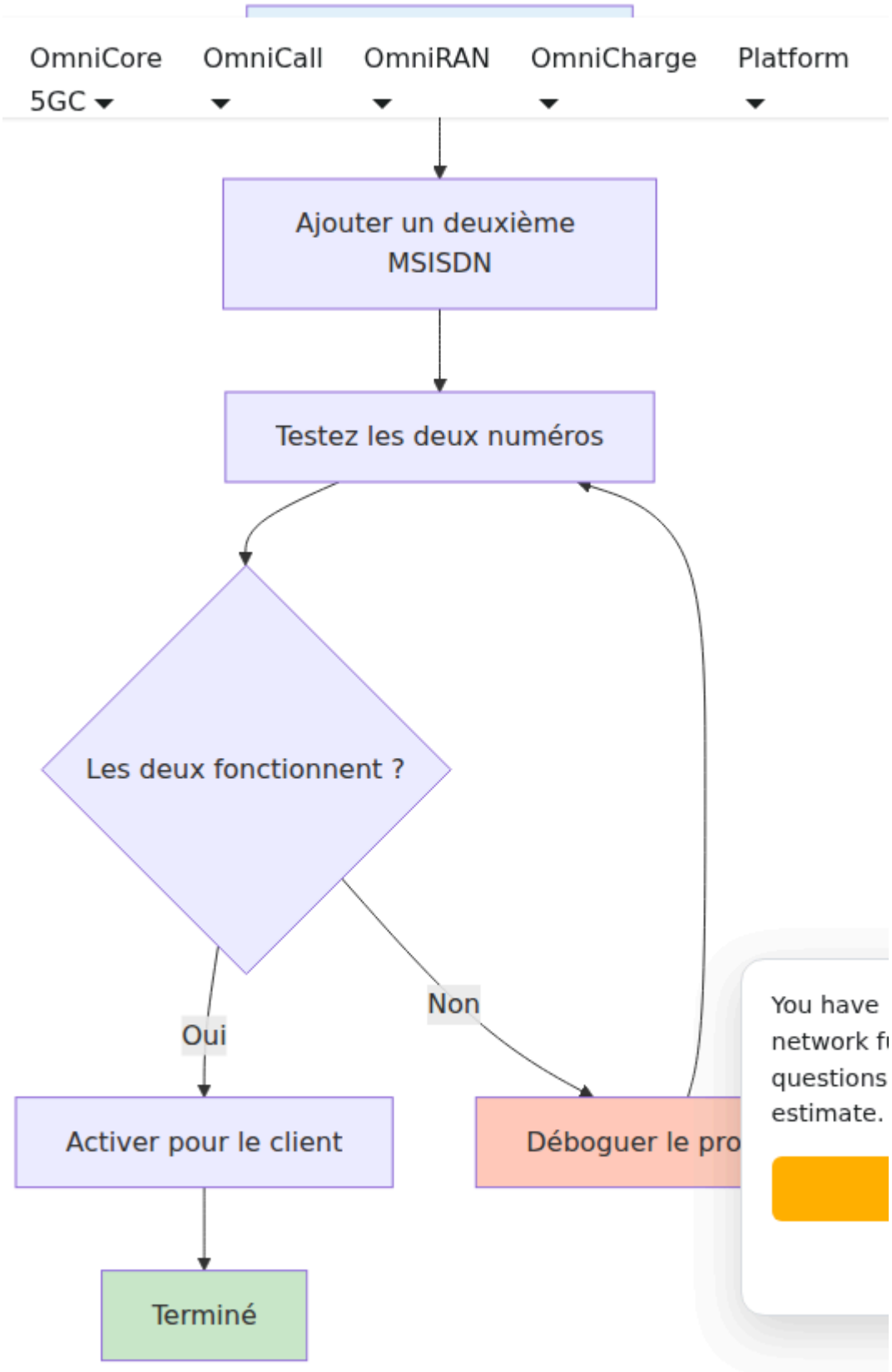
Pour faire sonner un numéro sur plusieurs appareils (par exemple un téléphone et une montre connectée sur des SIMs séparées), regroupez les appareils dans une **souscription IMS**. C'est le modèle 3GPP d'une Identité Utilisateur Publique partagée (3GPP TS 23.228) : plusieurs Identités Utilisateur Privées (IMPIs, une par IMSI) partagent une identité publique et, de manière critique, un seul S-CSCF de service. Le S-CSCF divise ensuite un appel de terminaison à chaque appareil enregistré.

Sans un groupe, deux abonnés qui partagent simplement un numéro sont deux souscriptions IMS séparées. Ils peuvent être assignés à différents S-CSCFs, donc un appel de terminaison n'atteint que les appareils sur l'un d'eux. Le groupe `ims_subscription` supprime cet écart en donnant aux membres un S-CSCF commun.

Comment cela fonctionne

- Un `ims_subscription` détient le S-CSCF partagé du groupe (`assigned_scscf`).
- Chaque abonné membre fait référence au groupe (`ims_subscription_id`).

- Lorsque le premier membre s'enregistre, le HSS sélectionne un S-CSCF. Le HSS ancre le groupe à ce nœud lorsque le Cx SAR du nœud confirme l'enregistrement. Chaque membre ultérieur est dirigé vers le **même** S-CSCF, quel que soit le S-CSCF que son enregistrement revendique.
- L'ancre détient toujours un véritable nœud S-CSCF. Avec `selection_method: :pool_uri`, le HSS retourne le nom de la piscine lors du premier REGISTER, mais il ancre le nœud que DNS a choisi, pas le nom de la piscine. Voir Un S-CSCF de service par souscription.
- Un nouvel enregistrement retourne le nœud ancré, jamais le nom de la piscine.
- Le S-CSCF partagé n'est libéré que lorsque le **dernier** membre se désenregistre.
- Lors d'une demande de terminaison (Cx LIR), le HSS retourne le S-CSCF partagé tant qu'un membre est enregistré.
- L'User-Data de Cx SAR liste chaque IMPI membre (l'ensemble d'enregistrement implicite), donc le S-CSCF enregistre l'ensemble. Un profil de service partagé optionnel sur le groupe rend une iFC pour les identités partagées.
- Lors d'une demande Sh pour l'identité publique partagée (Data-Reference `IMSPublicIdentity`), le HSS inclut un `<IMSPrivateUserIdentity>` par membre IMSI dans l'User-Data. Un Serveur d'Application (par exemple le MMTel TAS) compte ces derniers pour détecter un numéro partagé : un compte supérieur à un marque un numéro partagé multi-IMSI, donc le AS permet à l'appel de terminaison de se diviser sur chaque appareil. Un numéro ordinaire retourne un seul `<IMSPrivateUserIdentity>` et est livré à un contact.



Provisionnement

Créez le groupe, attachez chaque abonné de l'appareil et assignez le numéro partagé à chaque membre.

```
# 1. Créer la souscription IMS (le groupe). Un ims_profile_id
optionnel définit un
#    Profil de Service partagé pour les identités partagées du
groupe.
GROUP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/ims_subscription \
  -H "Content-Type: application/json" \
  -d '{"name": "Famille Smith - numéro partagé"}' | jq -r
'.response.id')

# 2. Attacher chaque abonné membre au groupe et lui donner le
MSISDN partagé.
curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$PHONE_SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d "{\"ims_subscription_id\": \"$GROUP_ID\", \"msisdns\":
[\"$SHARED_MSISDN_ID\"]}"

curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$WATCH_SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d "{\"ims_subscription_id\": \"$GROUP_ID\", \"msisdns\":
[\"$SHARED_MSISDN_ID\"]}"
```

Paramètres :

Paramètre	Type	Requis	Description
`name`	Chaîne	Oui	Nom lisible par l'homme pour la souscription IMS (unique).
`ims_profile_id`	UUID	Non	Profil de Service partagé (iFC) pour les identités partagées du groupe. Lorsqu'il est absent, chaque membre rend son propre profil IMS.
`ims_subscription_id`	UUID	Non	Sur un abonné : le groupe auquel il appartient. Lorsqu'il est absent, l'abonné est autonome (son propre S-CSCF par abonné, par défaut).

Le `assigned_scscf` et `last_seen_scscf_timestamp` sur le groupe sont des états d'exécution écrits par le HSS ; vous ne les définissez pas.

Remarques

- Un abonné sans `ims_subscription_id` est autonome et se comporte exactement comme auparavant.
- Donnez aux membres le même MSISDN afin que le numéro partagé résolve vers le groupe. Les membres peuvent conserver leurs propres numéros supplémentaires.
- Supprimer un groupe ne supprime pas ses abonnés ; leur `ims_subscription_id` est effacé et ils deviennent autonomes.

Procédures opérationnelles

Gestion des abonnés Multi-MSISDN

Voir tous les MSISDNs pour un abonné :

```
Interroger via l'API : GET /api/subscriber/imsi/:imsi
```

La réponse inclut tous les MSISDNs liés.

Dépannage Multi-IMSI

L'appareil ne s'attache pas avec le deuxième IMSI :

1. Vérifiez qu'un deuxième enregistrement d'abonné existe pour cet IMSI
2. Vérifiez que le key_set est configuré correctement pour cet IMSI
3. Vérifiez que le profil EPC est assigné
4. Confirmez que les règles d'itinérance permettent l'attachement

L'appareil change d'IMSI de manière inattendue :

- Cela est contrôlé par la logique de l'appareil/SIM, pas par le HSS
- Le HSS authentifie quel que soit l'IMSI présenté
- Vérifiez les paramètres de sélection d'IMSI de l'appareil

Dépannage Multi-MSISDN

Le deuxième numéro ne sonne pas :

1. Vérifiez que le MSISDN est lié dans la table de jointure
2. Vérifiez que le modèle de profil IMS inclut la variable `{{msisdns}}`
3. Confirmez que l'enregistrement IMS inclut toutes les identités publiques
4. Consultez les journaux S-CSCF pour les identités enregistrées

Les appels sortants n'affichent qu'un seul numéro :

- L'appareil sélectionne quel numéro présenter comme ID d'appelant
- C'est une configuration de l'appareil, pas du HSS
- Le HSS fournit toutes les identités ; l'appareil choisit

Résumé des avantages

Avantages Multi-MSISDN

✓ Une SIM, plusieurs numéros de téléphone
✓ Lignes professionnelles et personnelles séparées
✓ Présence locale internationale
✓ Gestion simplifiée des appareils
✓ Tous les numéros partagent le même service de données
✓ Facturation centralisée par IMSI

Avantages SIM Multi-IMSI

✓ Coûts d'itinérance optimisés
✓ Sélection automatique du réseau
✓ Redondance et basculement
✓ Identité réseau locale
✓ Conformité réglementaire
✓ Continuité de service à travers les réseaux

Avantages combinés

✓ Flexibilité maximale
✓ Ensembles de numéros différents par réseau
✓ Optimisé pour chaque cas d'utilisation
✓ Scénarios commerciaux complexes
✓ Optimisation internationale et locale

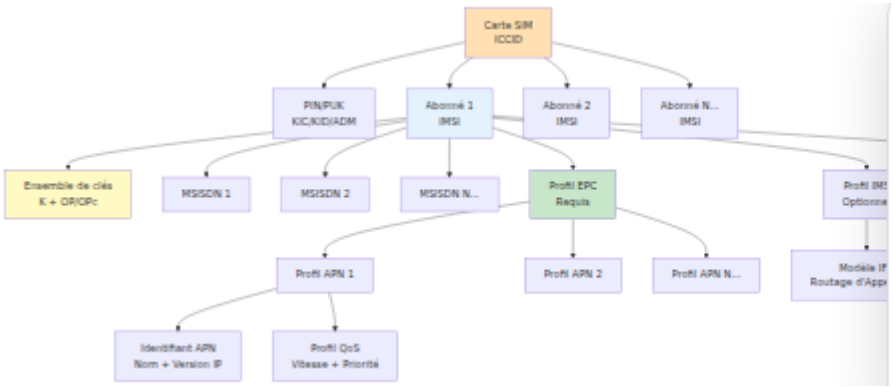
← Retour au Guide des opérations

Gestion des Profils OmniHSS

← Retour au Guide des Opérations

Vue d'ensemble

OmniHSS utilise des **profils** pour définir les caractéristiques de service pour les abonnés. Les profils vous permettent de créer des modèles de service réutilisables qui peuvent être attribués à plusieurs abonnés, simplifiant ainsi le provisionnement et garantissant la cohérence.

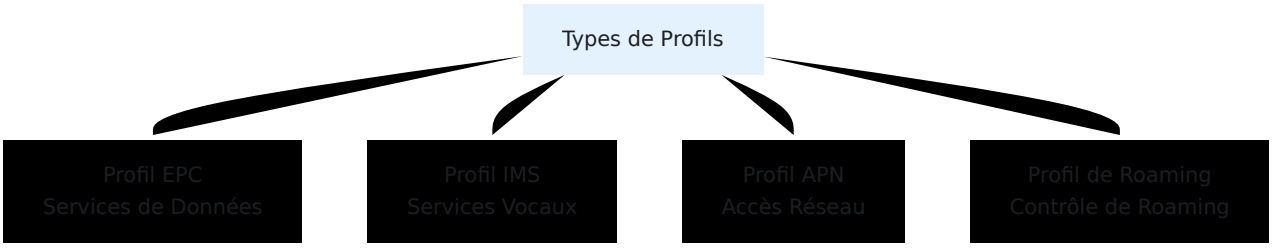


You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

Get an estima

No thanks

Types de Profils



Profils EPC

Les Profils EPC (Evolved Packet Core) définissent les caractéristiques de service de données pour les abonnés LTE.

Paramètres Clés

Paramètre	Description	Valeurs Typ
`ue_ambr_dl_kbps`	Limite de vitesse de téléchargement	10,000 - 1,00 Kbps
`ue_ambr_ul_kbps`	Limite de vitesse de téléversement	5,000 - 500,0 Kbps
`network_access_mode`	Type de service	"packet_only" "packet_and_
`tracking_area_update_interval_seconds`	Minuteur TAU	54 secondes (typique)

Création de Profils EPC

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_profiles": [],
    "name": "Premium 100Mbps",
    "network_access_mode": "packet_only",
    "tracking_area_update_interval_seconds": 600,
    "ue_ambr_dl_kbps": 100000,
    "ue_ambr_ul_kbps": 50000
  }'
```

Modèles de Profils EPC Courants

Internet de Base :

- Téléchargement : 10 Mbps (10,000 Kbps)
- Téléversement : 5 Mbps (5,000 Kbps)

Standard :

- Téléchargement : 50 Mbps (50,000 Kbps)
- Téléversement : 25 Mbps (25,000 Kbps)

Premium :

- Téléchargement : 100 Mbps (100,000 Kbps)
- Téléversement : 50 Mbps (50,000 Kbps)

Illimité :

- Téléchargement : 1 Gbps (1,000,000 Kbps)
- Téléversement : 500 Mbps (500,000 Kbps)

Profils IMS

Les Profils IMS définissent les caractéristiques de service vocal, principalement à travers des modèles IFC (Initial Filter Criteria).

Modèles IFC

Les modèles IFC sont des documents XML qui définissent les règles de routage des appels pour le S-CSCF.

Variables de Modèle :

- `{{imsi}}` - IMSI de l'abonné
- `{{msisdns}}` - Liste des numéros de téléphone
- `{{mcc}}` - Code du pays d'origine

- `{{mnc}}` - Code du réseau d'origine

Création de Profils IMS

```
curl -k -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{
    "ims_profile": {
      "name": "Standard VoLTE",
      "ifc_template": "<InitialFilterCriteria>...
</InitialFilterCriteria>"
    }
  }'
```

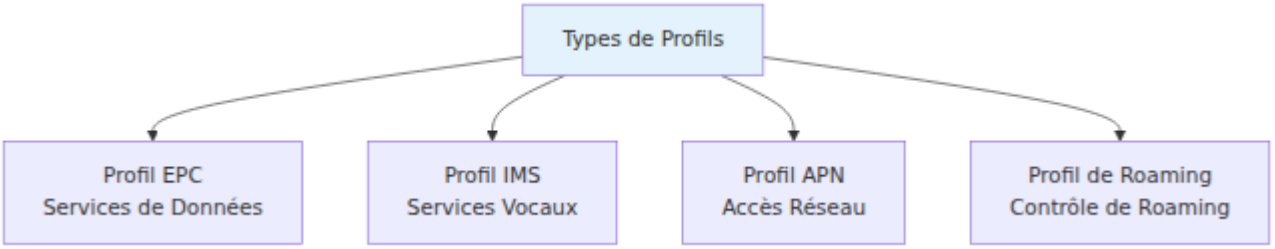
Exemple de Modèle IFC

```
<ServiceProfile>
  <PublicIdentity>
    <Identity>sip:
{{imsi}}@ims.mnc{{mnc}}.mcc{{mcc}}.3gppnetwork.org</Identity>
  </PublicIdentity>
  <InitialFilterCriteria>
    <Priority>0</Priority>
    <TriggerPoint>
      <ConditionTypeCNF>0</ConditionTypeCNF>
      <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
      </SPT>
    </TriggerPoint>
    <ApplicationServer>
      <ServerName>sip:as.ims.example.com</ServerName>
      <DefaultHandling>0</DefaultHandling>
    </ApplicationServer>
  </InitialFilterCriteria>
</ServiceProfile>
```

Profils APN

Les Profils APN (Access Point Name) définissent les points d'accès réseau pour les connexions de données.

Composants APN



Identifiant APN

Définit le nom APN et le support du protocole IP.

APNs Courants :

- `internet` - Accès Internet général
- `ims` - Signalisation IMS/VoLTE
- `mms` - Messagerie multim��dia
- `vzwadmin` - Spécifique à l'opérateur

Options de Version IP :

- `"ipv4"`: IPv4 uniquement
- `"ipv6"`: IPv6 uniquement
- `"ipv4v6"`: IPv4v6 (double pile)
- `"ipv4_or_ipv6"`: IPv4 ou IPv6 (choix du réseau)

Profil QoS APN

Définit les paramètres de qualité de service.

Valeurs QCI (QoS Class Identifier) :

QCI	Type	Cas d'utilisation	Priorité
1	GBR	Voix conversationnelle	Plus Élevée
2	GBR	Vidéo conversationnelle	Élevée
4	GBR	Streaming vidéo	Élevée
5	Non-GBR	Signalisation IMS	Moyenne
9	Non-GBR	Internet (par défaut)	Plus Basse

Méthode de Facturation au Niveau de la Session (optionnelle) :

Le profil QoS APN peut également fixer la méthode de facturation par défaut pour la session IP-CAN, envoyée comme les AVPs `Online`/`Offline` de niveau supérieur dans le Gx CCA-I.

Champ	Type	Par Défaut	Description
`online_charging_enabled`	boolean	`null`	Lorsqu'il est défini, force l'AVP `Online` de niveau session (`true` = activer, `false` = désactiver). Lorsque `null`, le PCRF renvoie la valeur que le PGW a envoyée dans le CCR-I.
`offline_charging_enabled`	boolean	`null`	Lorsqu'il est défini, force l'AVP `Offline` de niveau session (`true` = activer, `false` = désactiver). Lorsque `null`, le PCRF renvoie la valeur que le PGW a envoyée dans le CCR-I.

Voir PCRF : Facturation en Ligne/Hors Ligne au Niveau de la Session pour le comportement complet et la référence des valeurs AVP.

Création d'une Configuration APN Complète

```
# 1. Créer l'Identifiant APN
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{"apn": "internet", "ip_version": "ipv4v6"}' \
  | jq -r '.response.id')

# 2. Créer le Profil QoS APN
QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Best Effort",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 50000,
    "apn_ambr_ul_kbps": 25000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "qci": 9
  }' | jq -r '.response.id')

# 3. Créer le Profil APN
curl -k -X POST https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_identifier_id\": \"$APN_ID\",
    \"apn_qos_profile_id\": \"$QOS_ID\",
    \"name\": \"Profil APN Internet\"
  }"
```

Attribution des APNs au Profil EPC

Les APNs sont liés aux Profils EPC via la table `join_epc_profile_to_apn_profile`.

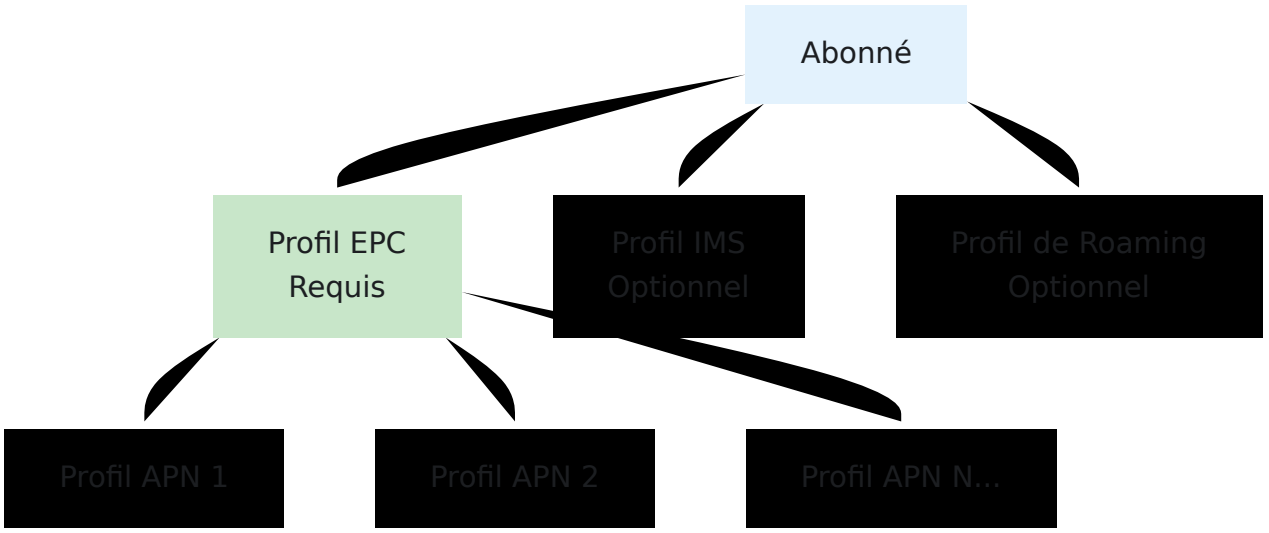
Insérez des enregistrements dans la table de jointure pour lier les identifiants de profil APN à l'identifiant de profil EPC. Plusieurs profils APN peuvent être attribués à un seul profil EPC.

Profils de Roaming

Voir la documentation détaillée dans le Guide de Contrôle de Roaming.

Attribution de Profils

Relations de Profils d'Abonnés



Attribution de Profils aux Abonnés

```
# Attribuer les profils EPC et IMS lors de la création de l'abonné
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
      "ims_profile_id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
      "roaming_profile_id": "9c858901-8a57-4791-81fe-4c455b099bc9"
    }
  }'

# Mettre à jour le profil de l'abonné
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "epc_profile_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479"
    }
  }'
```

Meilleures Pratiques de Gestion des Profils

Principes de Conception

1. **Créer des Profils Standards** - Définir des niveaux de service communs (Basique, Standard, Premium)
2. **Réutiliser les Profils** - Attribuer le même profil à plusieurs abonnés
3. **Documenter les Changements** - Suivre les modifications de profil

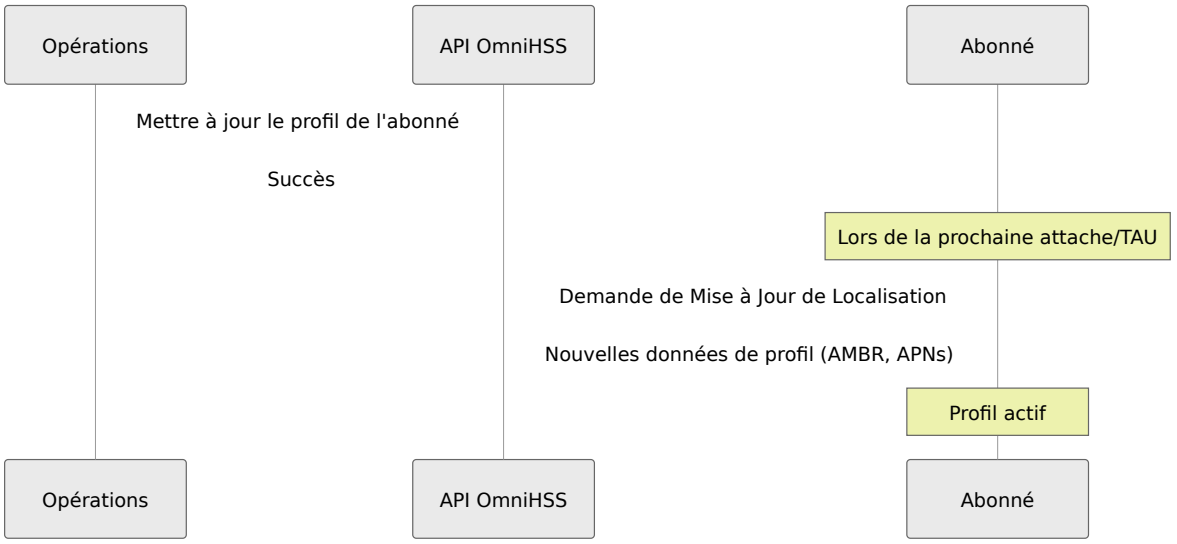
4. **Tester Avant Production** - Vérifier que le profil fonctionne d'abord avec un abonné de test

Convention de Nommage des Profils

```
[Niveau de Service]-[Vitesse]-[Fonctionnalités]

Exemples :
- "Basique-10Mbps-Internet"
- "Premium-100Mbps-VoLTE"
- "Entreprise-1Gbps-MultiAPN"
```

Migration de Profils

Lors du changement de profil d'un abonné :



Important : Les changements de profil prennent effet lors de la prochaine :

- Mise à Jour de Zone de Suivi (TAU)
- Attache
- Enregistrement IMS (pour les changements de profil IMS)

Dépannage des Problèmes de Profil

L'abonné ne reçoit pas la vitesse attendue :

1. Vérifiez les valeurs AMBR du profil EPC attribué
2. Vérifiez les valeurs AMBR du profil QoS APN
3. Vérifiez que le MME/P-GW applique correctement le QoS
4. Vérifiez la congestion du réseau

L'enregistrement IMS échoue :

1. Vérifiez le profil IMS attribué
2. Vérifiez la validité du XML du modèle IFC
3. Consultez les journaux S-CSCF pour les erreurs de traitement IFC
4. Confirmez la configuration de sélection S-CSCF

APN non disponible :

1. Vérifiez que le profil APN est lié au profil EPC
2. Vérifiez que l'identifiant APN correspond à la demande réseau
3. Consultez la demande de connectivité PDN de l'UE

← Retour au Guide des Opérations | Suivant : Contrôle de Roaming →

Flux de Protocole OmniHSS

← Retour au Guide des Opérations

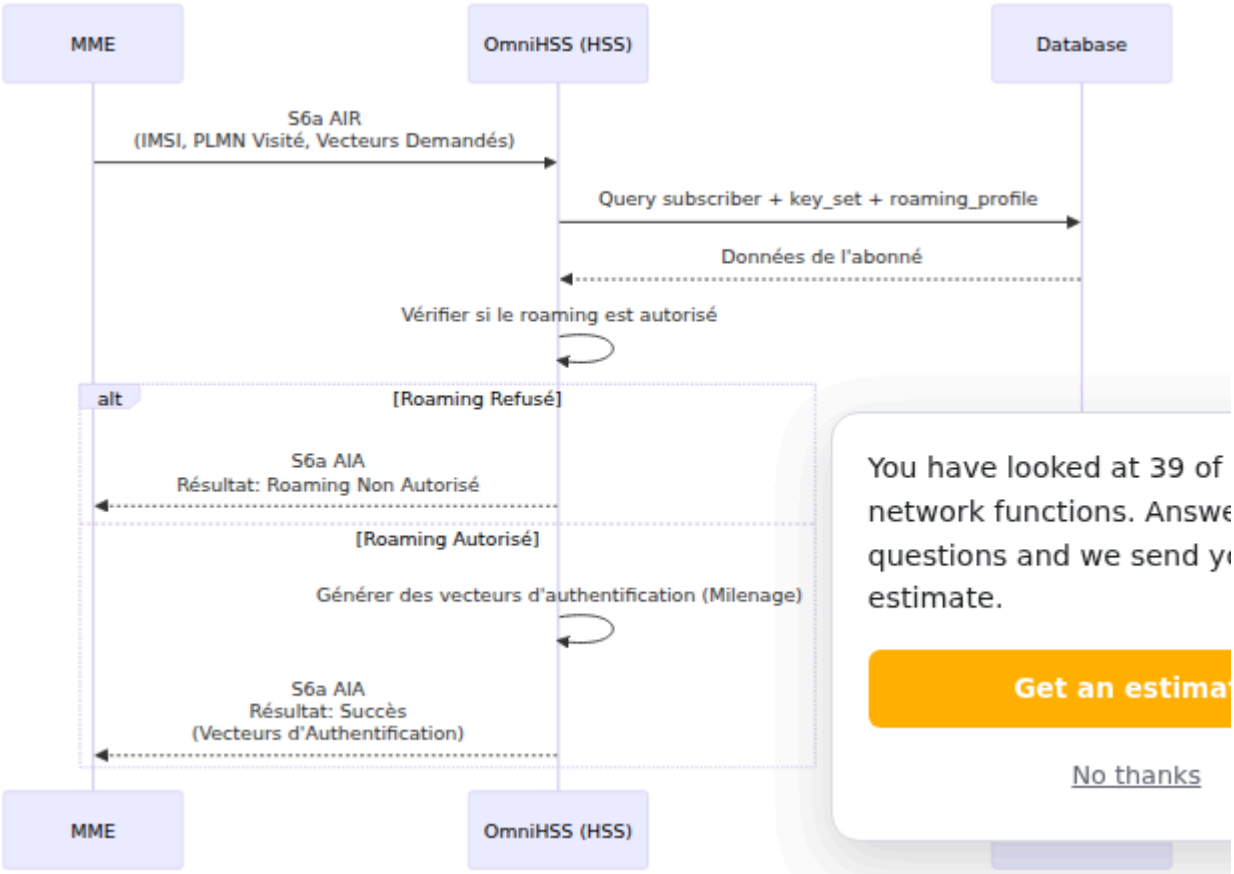
Vue d'ensemble

Ce document détaille les flux de messages du protocole Diameter pris en charge par OmniHSS. Comprendre ces flux est essentiel pour le dépannage et les opérations.

Interface S6a (LTE/EPC)

Demande d'Information d'Authentification (AIR/AIA)

Le MME demande des vecteurs d'authentification pour l'abonné.

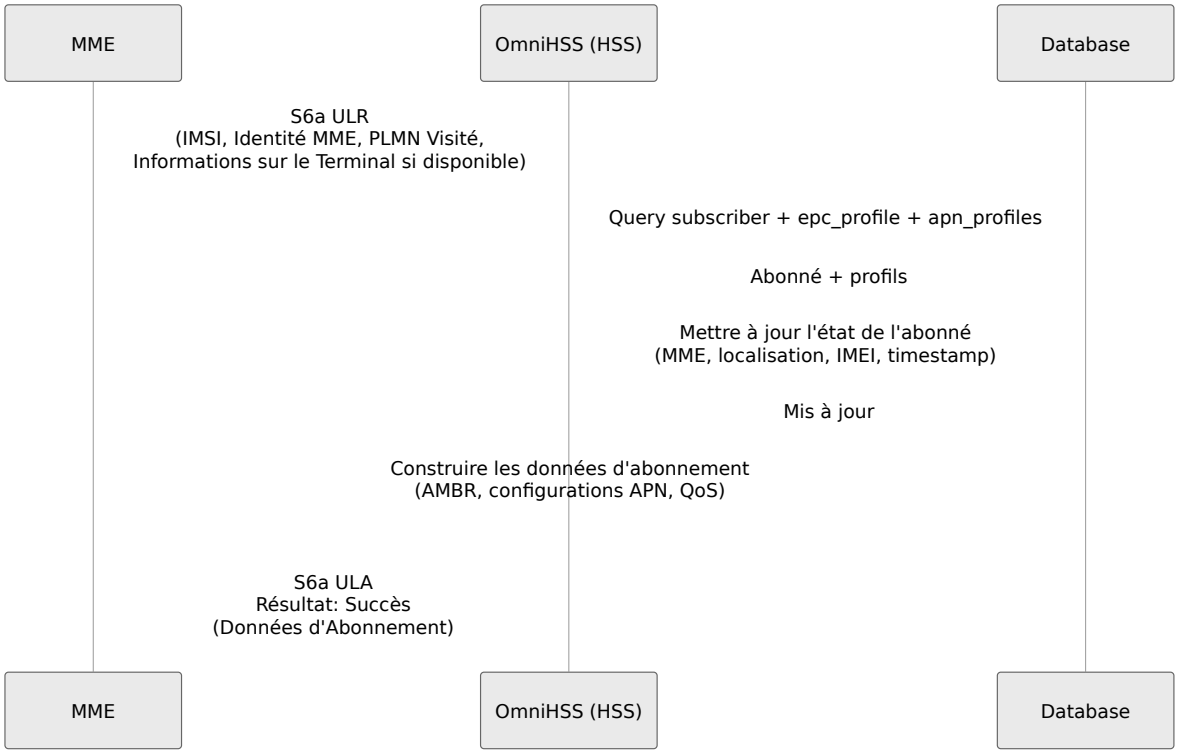


AVPs Clés:

- Demande: Nom d'Utilisateur (IMSI), Identifiant-PLMN-Visité, Nombre de Vecteurs Demandés
- Réponse: Information d'Authentification (RAND, AUTN, XRES, KASME)

Demande de Mise à Jour de Localisation (ULR/ULA)

Le MME notifie le HSS de la localisation de l'abonné et récupère les données d'abonnement.

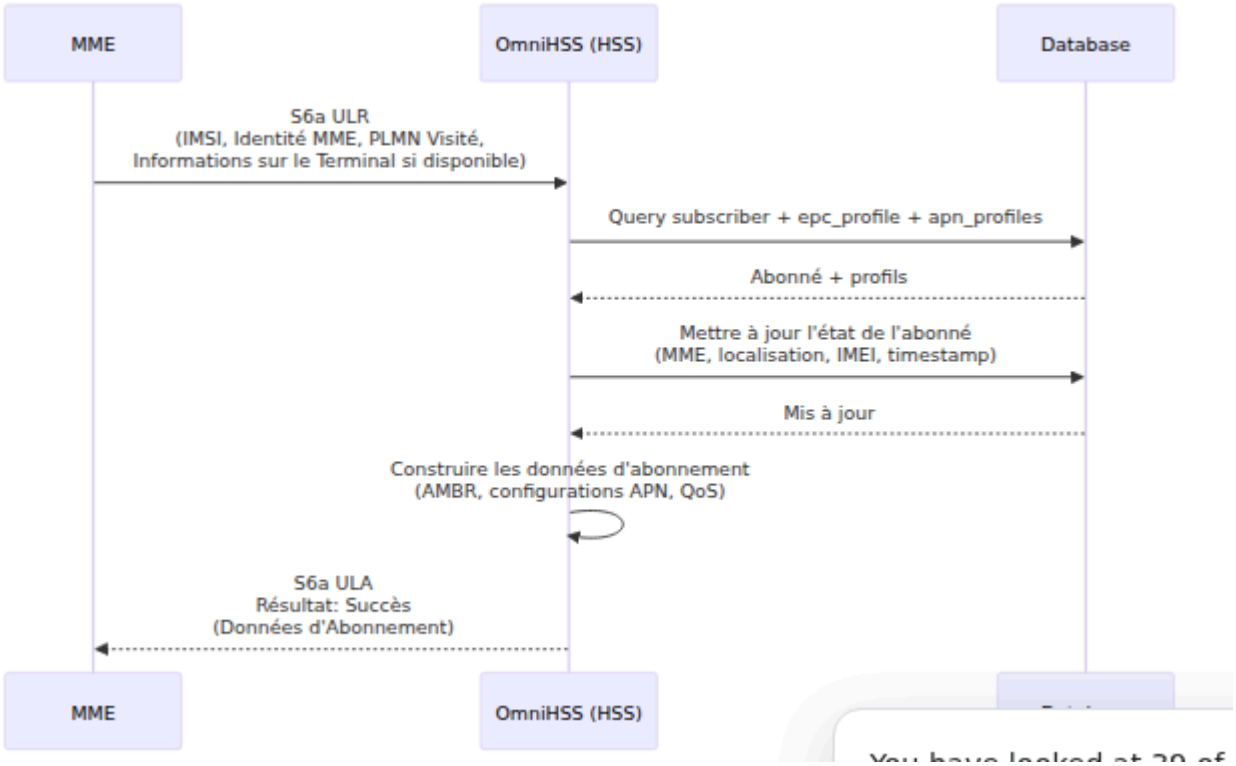


AVPs Clés:

- Demande: Nom d'Utilisateur (IMSI), Type-RAT, ULR-Flags, Identifiant-PLMN-Visité, Capacité-UE-SRVCC, Informations sur le Terminal (IMEI, Version-Logicielle; optionnel)
- Réponse: Données-D'abonnement (AMBR, Configuration-APN, Mode-D'accès-Réseau)

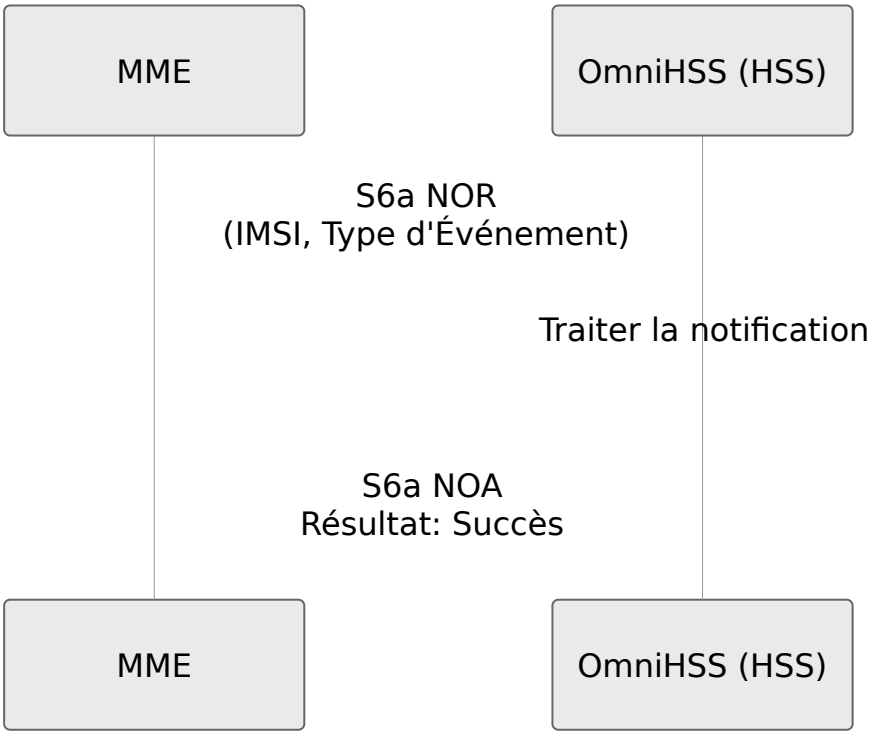
Demande de Purge UE (PUR/PUA)

Le MME notifie le HSS lorsque le contexte de l'abonné est supprimé.



Demande de Notification (NOR/NOA)

Le MME informe le HSS de divers événements.

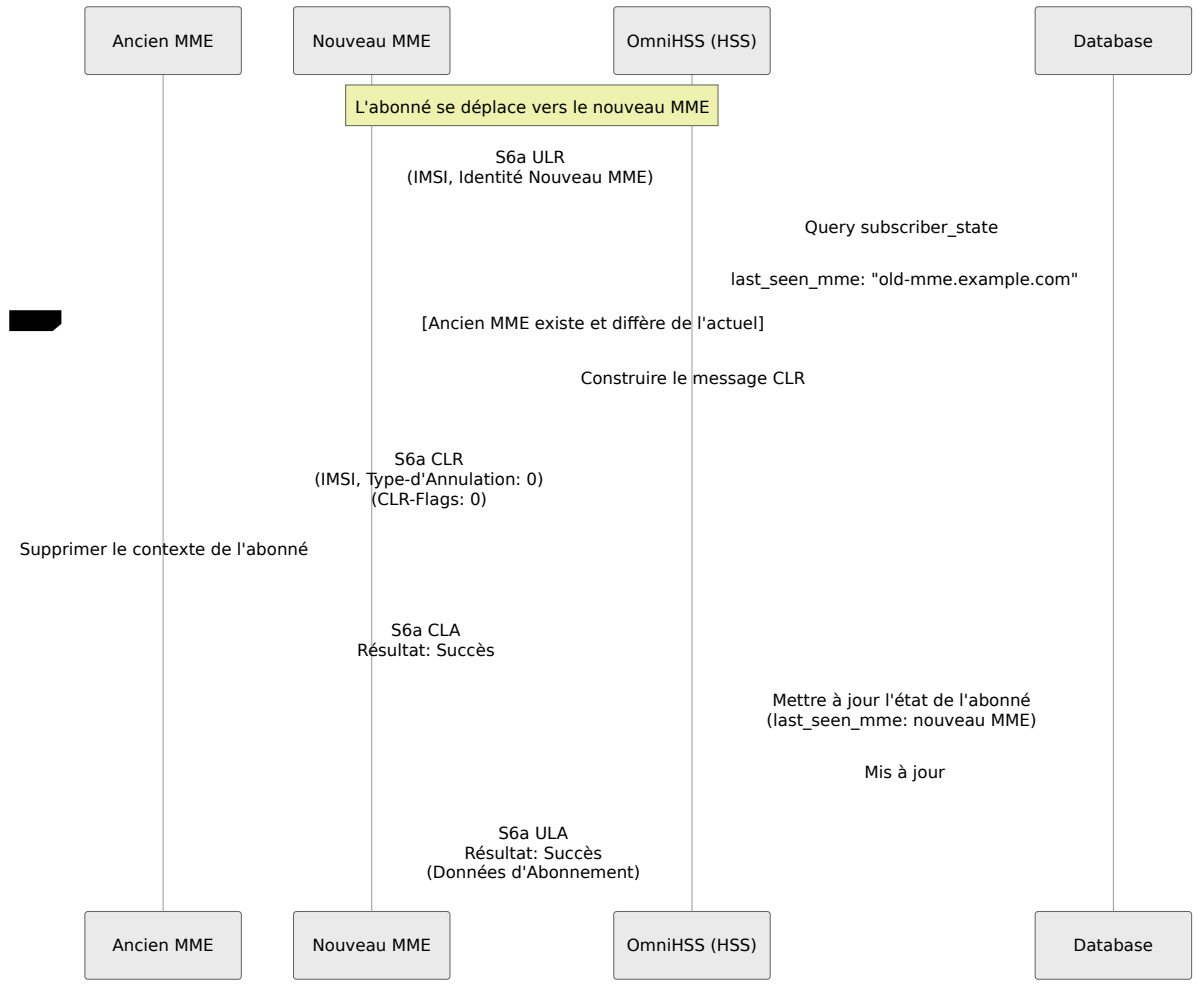


Demande d'Annulation de Localisation (CLR/CLA)

Le HSS initie l'annulation de localisation pour informer le MME que l'abonné doit être détaché. OmniHSS prend en charge l'envoi automatique et programmatique de CLR.

CLR Automatique (Transfert MME)

Lorsqu'un abonné effectue une Demande de Mise à Jour de Localisation depuis un nouveau MME, OmniHSS envoie automatiquement un CLR au précédent MME pour nettoyer les enregistrements obsolètes.



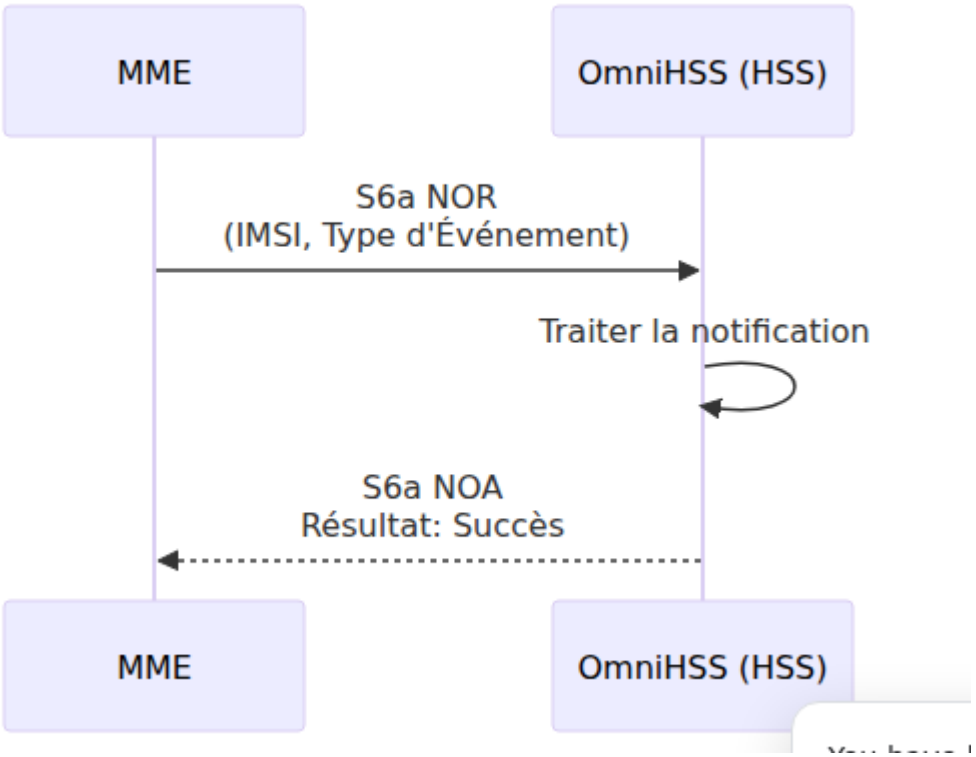
AVPs Clés (CLR Automatique):

- Nom d'Utilisateur: IMSI de l'abonné
- Hôte-Destination: Nom d'hôte de l'ancien MME
- Domaine-Destination: Domaine de l'ancien MME

- Type-d'Annulation: 0 (Procédure de Mise à Jour MME)
- CLR-Flags: 0
- Données-D'abonnement: Profil d'abonnement complet

CLR Programmatique (Déclenché par API)

Les administrateurs peuvent déclencher CLR via l'API programmatique pour détacher de force les abonnés (par exemple, pour un retrait d'abonnement, la prévention de fraude ou des actions administratives).



AVPs Clés (CLR Programmatique):

- Nom d'Utilisateur: IMSI de l'abonné
- Hôte-Destination: Nom d'hôte du dernier MME vu
- Domaine-Destination: Domaine du dernier MME vu
- Type-d'Annulation: `:subscription_withdrawal` (codé comme entier selon 3GPP TS 29.272)
- CLR-Flags:
  - s6a_indicator: 1 (indique l'interface S6a)
  - reattach_required: 1 (l'UE doit se ré-authentifier pour se rattacher)

Types d'Annulation

OmniHSS prend en charge plusieurs types d'annulation selon 3GPP TS 29.272 :

Type	Valeur	Description	Cas d'utilisation
Procédure de Mise à Jour MME	0	Changement normal de MME	Automatique lors de ULR depuis un nouveau MME
Procédure de Mise à Jour SGSN	1	Transfert SGSN	Scénarios de transfert 3G/2G
Retrait d'Abonnement	2	Résiliation administrative	Détachement manuel via API
Procédure de Mise à Jour IWF	3	Mise à jour de la fonction d'interconnexion	Interopérabilité avec le réseau hérité
Procédure de Rattachement Initial	4	Enregistrement frais	Forcer la ré-authentification

CLR-Flags

L'AVP CLR-Flags est un masque de bits avec les champs suivants :

Drapeau	Bit	Description
Indicateur S6a/S6d	0	1 = Interface S6a utilisée
Rattachement Requis	1	1 = L'UE doit effectuer un nouvel attachement

Exemple de Configuration CLR-Flags :

```
clr_flags: %{
  s6a_indicator: 1,        # Utilisation de l'interface S6a
  reattach_required: 1     # Forcer la ré-authentification
}
```

Scénarios Multi-IMSI

OmniHSS suit l'enregistrement MME **par abonné (IMSI)**, et non par MSISDN. Cela est crucial pour comprendre le comportement de CLR dans des scénarios multi-IMSI :

Scénario 1: Plusieurs MSISDN, Un Seul IMSI

```
Abonné A:
  - IMSI: 999000123456789
  - MSISDNs: ["+1234567890", "+9876543210"]
  - last_seen_mme: "mme01.operator.com"
```

Lorsque cet abonné se déplace vers un nouveau MME :

- **Un CLR envoyé** à "mme01.operator.com" avec IMSI 999000123456789
- Les deux MSISDNs sont affectés (même abonné, même SIM)
- L'AVP Nom d'Utilisateur contient l'IMSI, pas les MSISDNs

Scénario 2: Plusieurs Abonnés (IMSI Différents), Même MSISDN

OmniHSS impose une **contrainte MSISDN unique** (un MSISDN ne peut pas appartenir à plusieurs abonnés simultanément). Cependant, lors du portage/migration :

```
Abonné A:
  - IMSI: 999000111111111
  - MSISDN: "+1234567890"
  - last_seen_mme: "mme01.operator.com"

Abonné B (après portage):
  - IMSI: 999000222222222
  - MSISDN: "+1234567890"  # Même MSISDN, SIM/IMSI différente
  - last_seen_mme: "mme02.operator.com"
```

Lorsque l'Abonné B s'enregistre :

- **Aucun CLR envoyé** (IMSI différent = abonné différent)
- L'Abonné A reste enregistré à mme01
- L'Abonné B s'enregistre à mme02
- Les deux peuvent être actifs simultanément (différents appareils physiques)

Scénario 3: CLR Programmatique pour Abonné Multi-MSISDN

Résultat :

- **Un CLR envoyé** au dernier_mme_vu de l'abonné
- **Tous les MSISDNs** associés à cet IMSI sont effectivement détachés
- L'IMSI est la clé primaire pour suivre l'enregistrement MME

Notes Importantes

1. **L'IMSI est la Clé :** Les opérations CLR se font toujours **par IMSI**, jamais par MSISDN. La table `subscriber_state` suit `last_seen_mme` par abonné (IMSI).

2. **Opération Atomique :** Chaque abonné ne peut être enregistré qu'à un seul MME à la fois. Le CLR automatique garantit cela en nettoyant l'ancien enregistrement.

3. **Pas de CLR si Pas de MME Précédent :** Si `last_seen_mme` est `nil` (l'abonné n'a jamais été enregistré), aucun CLR n'est envoyé lors de ULR.

4. **Données d'Abonnement Incluses :** Le CLR automatique (durant ULR) inclut l'AVP complet `Subscription-Data` pour aider l'ancien MME à nettoyer correctement le contexte.

5. **Asynchrone :** Le CLR est envoyé de manière asynchrone (fire-and-forget). La réponse ULA au nouveau MME n'attend pas le CLA de l'ancien MME.

6. **Gestion du CLA :** OmniHSS reçoit les réponses CLA mais les rejette actuellement. Cela empêche les boucles de messages et est un comportement standard du HSS.

Interface SWx (Accès Non-3GPP / Appels WiFi)

L'interface SWx permet les appels WiFi en connectant l'ePDG (Evolved Packet Data Gateway) au HSS pour l'authentification et l'autorisation des abonnés sur des réseaux d'accès non-3GPP.

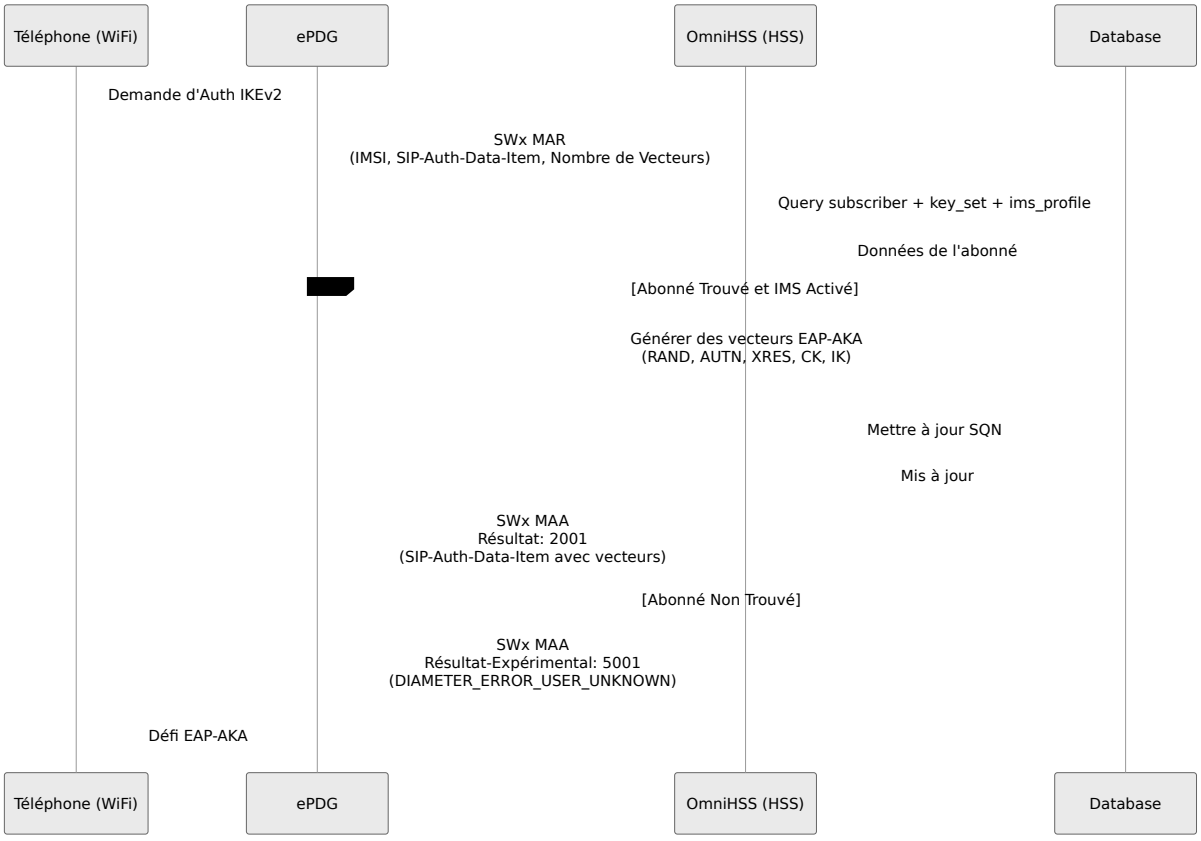
Demande d'Auth Multimédia (MAR/MAA)

L'ePDG demande des vecteurs d'authentification EAP-AKA pour l'enregistrement des appels WiFi. Un TWAG (trusted WLAN) demande des vecteurs EAP-AKA' via la même interface.

Le `SIP-Authentication-Scheme` dans la demande sélectionne le type de vecteur (clause 8.1.2.1 de 3GPP TS 29.273) :

Schéma	ANID	Vecteur retourné
`EAP-AKA`	Ignoré	Bit de séparation AMF 0. `Confidentiality-Key` et `Integrity-Key` transportent CK et IK.
`EAP-AKA'`	Requis	Bit de séparation AMF 1. `Confidentiality-Key` et `Integrity-Key` transportent CK' et IK', dérivés de CK, IK, SQN xor AK et l'ANID selon l'Annexe A.2 de 3GPP TS 33.402. CK et IK ne quittent pas le HSS.

Une demande avec `EAP-AKA'` et sans `ANID` est répondue par `Result-Code` 5005 (DIAMETER_MISSING_AVP). Tout autre schéma est répondu par 5004 (DIAMETER_INVALID_AVP_VALUE). L'ANID pour un réseau d'accès WLAN est `WLAN` (clause 8.1.1.2 de 3GPP TS 24.302).



AVPs Clés:

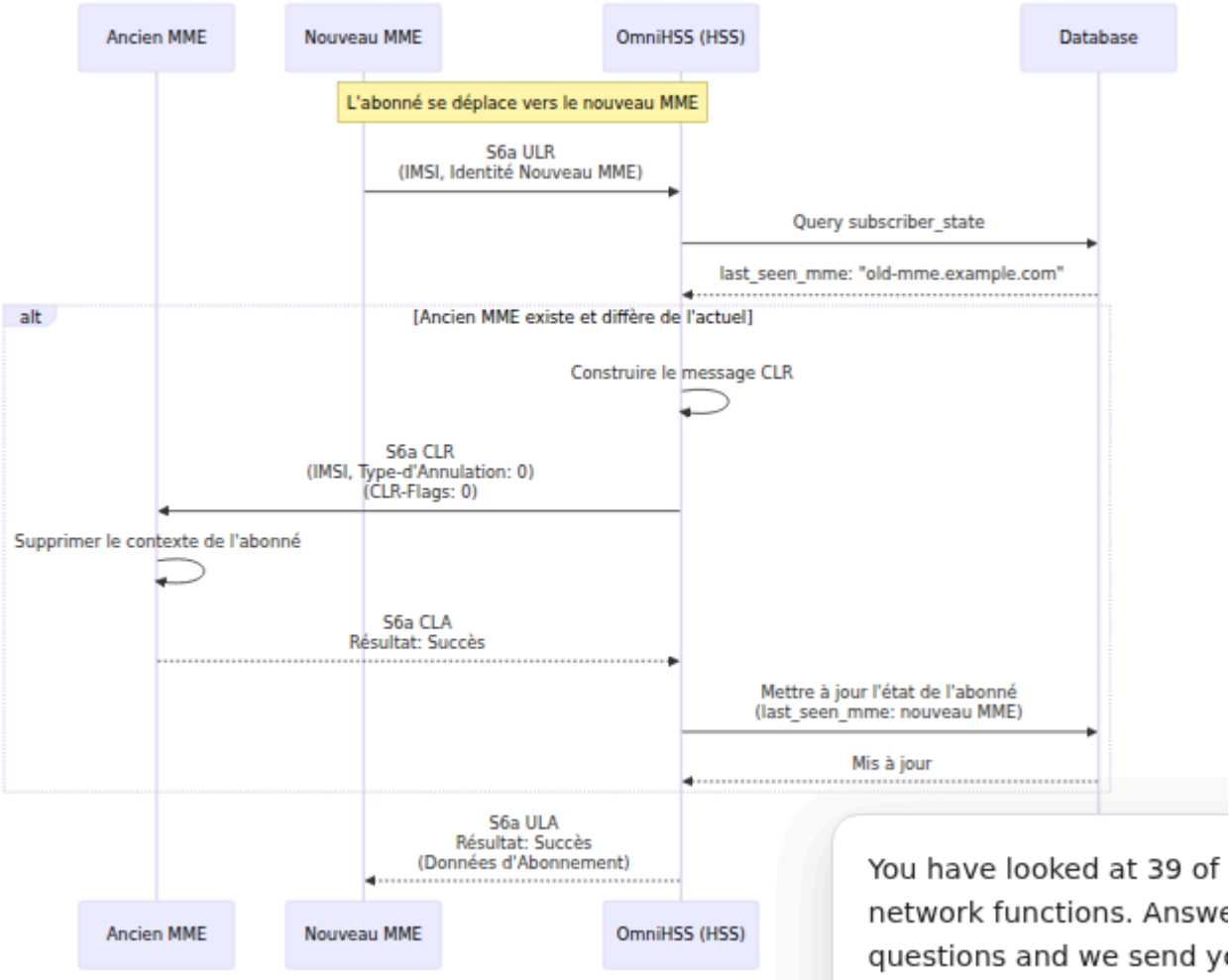
- Demande: Nom d'Utilisateur (IMSI ou IMSI@domaine), SIP-Auth-Data-Item (schéma d'authentification), SIP-Number-Auth-Items

- Réponse: SIP-Auth-Data-Item (SIP-Authenticate, SIP-Authorization, Confidentiality-Key, Integrity-Key)

Resynchronisation SQN: Lorsque l'UE détecte un échec d'authentification en raison d'un décalage SQN, l'ePDG inclut des données de resynchronisation dans l'AVP SIP-Authorization. OmniHSS extrait AUTS de ce champ et recalcule SQN en utilisant un calcul sensible à la longueur d'IND selon 3GPP TS 33.102.

Demande d'Attribution de Serveur (SAR/SAA)

L'ePDG enregistre ou désenregistre l'abonné pour les services d'appels WiFi.



Types d'Attribution de Serveur:

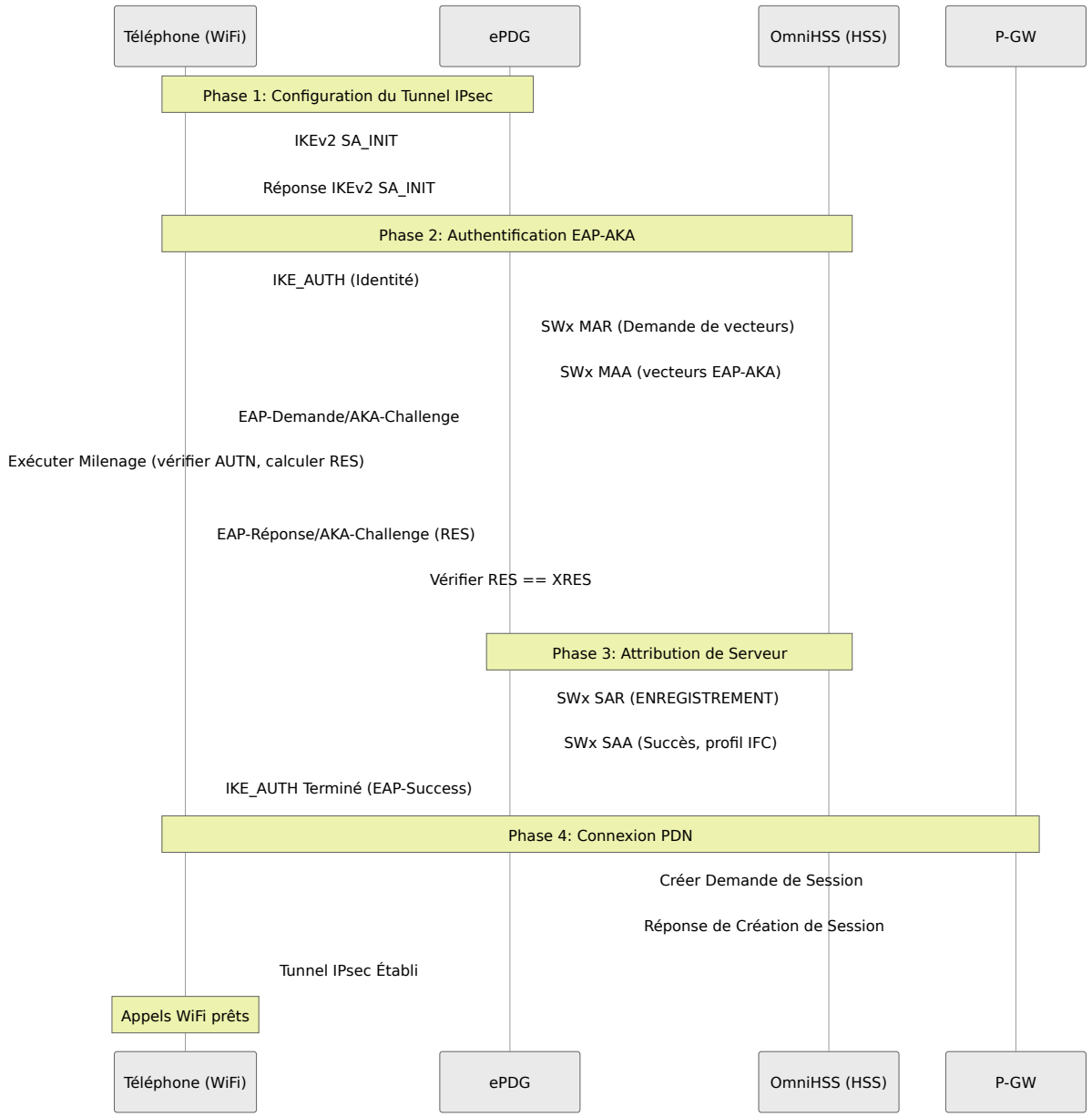
Type	Valeur	Description
NO_ASSIGNMENT	0	Query seulement, pas de changement d'état
ENREGISTREMENT	1	Enregistrement initial des appels WiFi
RE_ENREGISTREMENT	2	Rafraîchissement de l'enregistrement
UTILISATEUR_NON_ENREGISTRÉ	3	Services pour utilisateur non enregistré
TIMEOUT_DEREGISTRATION	4	Délai d'enregistrement
USER_DEREGISTRATION	5	Désenregistrement initié par l'utilisateur
AUTHENTICATION_FAILURE	7	Échec d'auth, effacer l'état
ADMINISTRATIVE_DEREGISTRATION	8	Désenregistrement initié par l'administrateur

AVPs Clés:

- Demande: Nom d'Utilisateur (IMSI), Type-Attribution-Serveur, Nom-Serveur (identité ePDG), Sélection-Service (APN)
- Réponse: User-Data (profil IFC XML), Non-3GPP-User-Data

Flux d'Enregistrement d'Appels WiFi

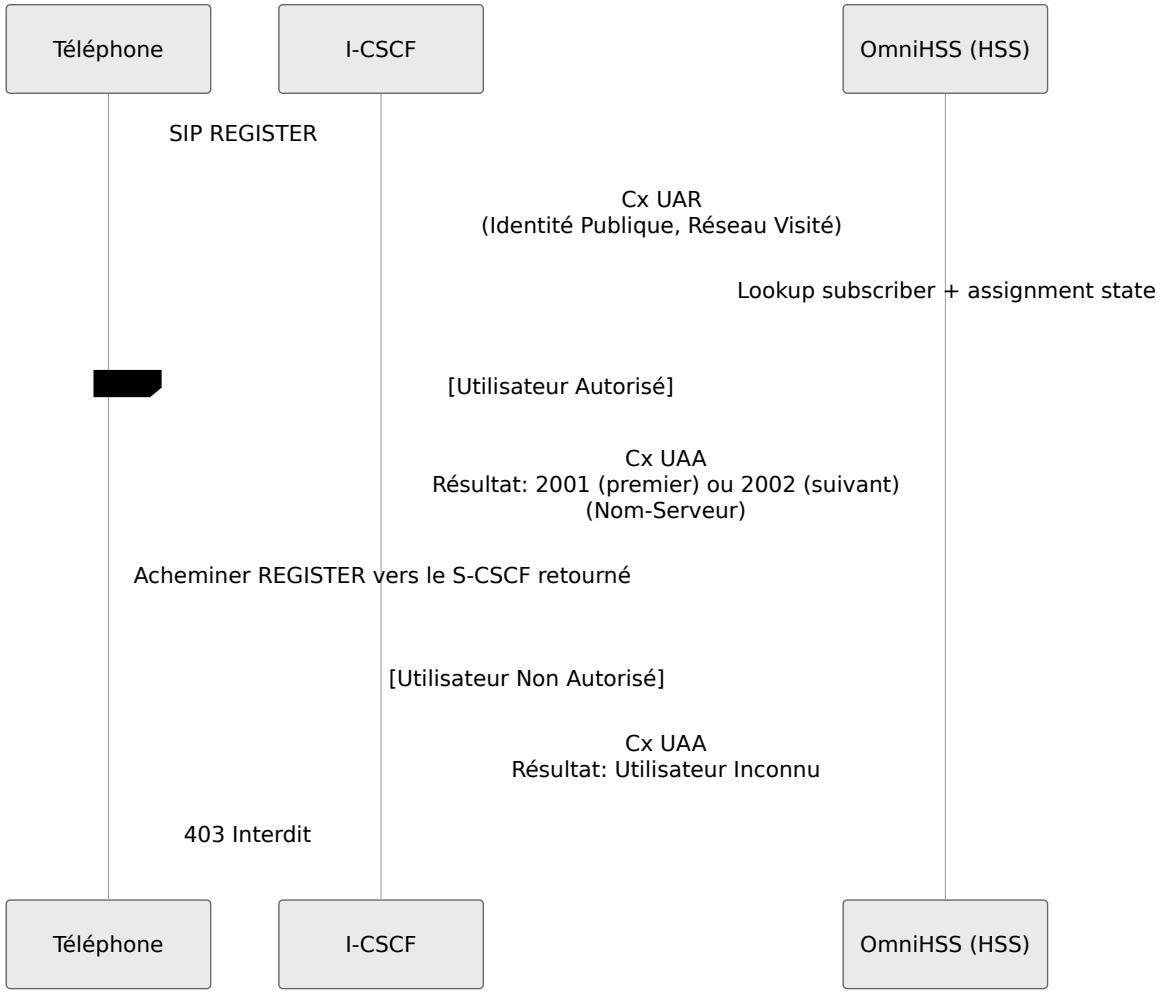
Enregistrement complet des appels WiFi de bout en bout.



Interface Cx (IMS)

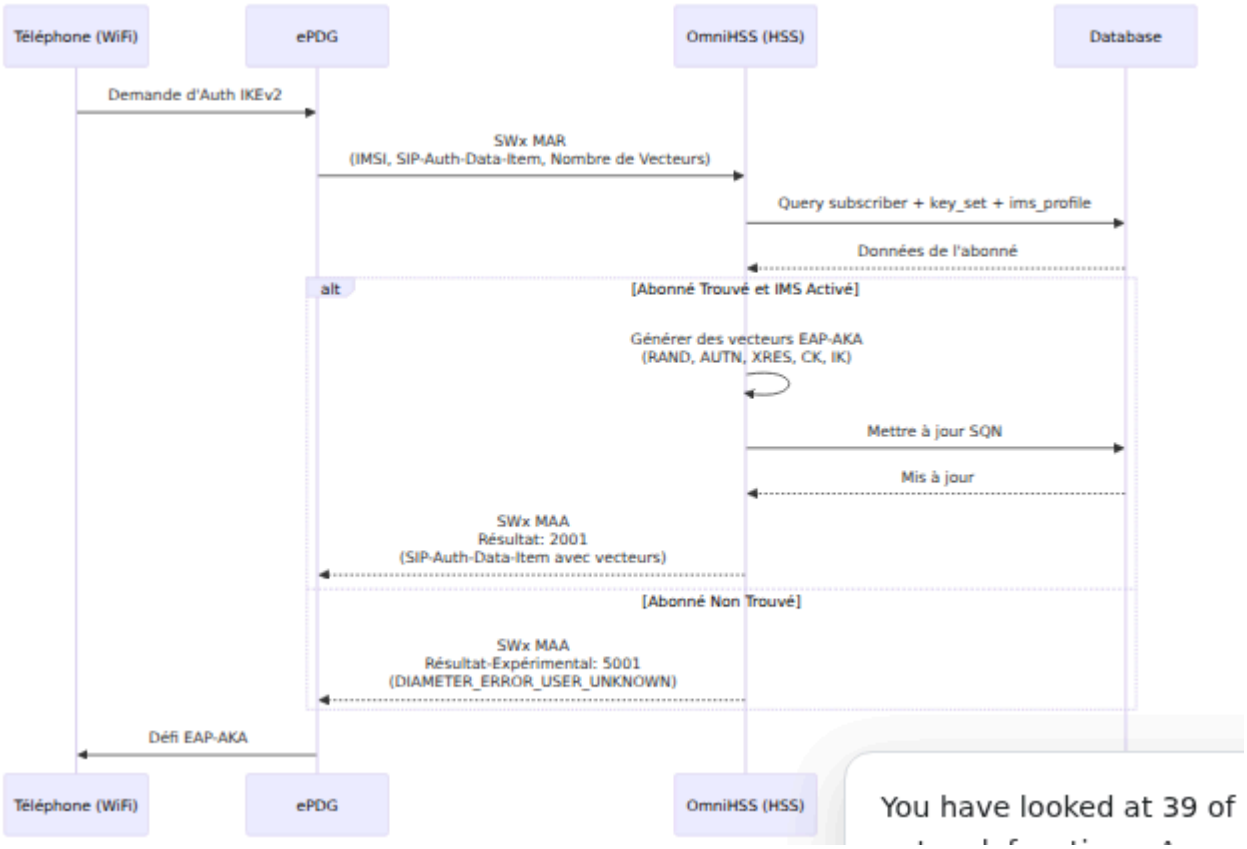
Demande d'Autorisation d'Utilisateur (UAR/UAA)

L'I-CSCF demande si l'utilisateur peut s'enregistrer, et quel S-CSCF utiliser. Le HSS renvoie un nom S-CSCF (un `Nom-Serveur`), pas un ensemble de capacités. Le nom et le code de résultat suivent les règles de État d'Attribution S-CSCF.



Demande d'Attribution de Serveur (SAR/SAA)

Le S-CSCF enregistre/désenregistre l'utilisateur et récupère le profil IMS.



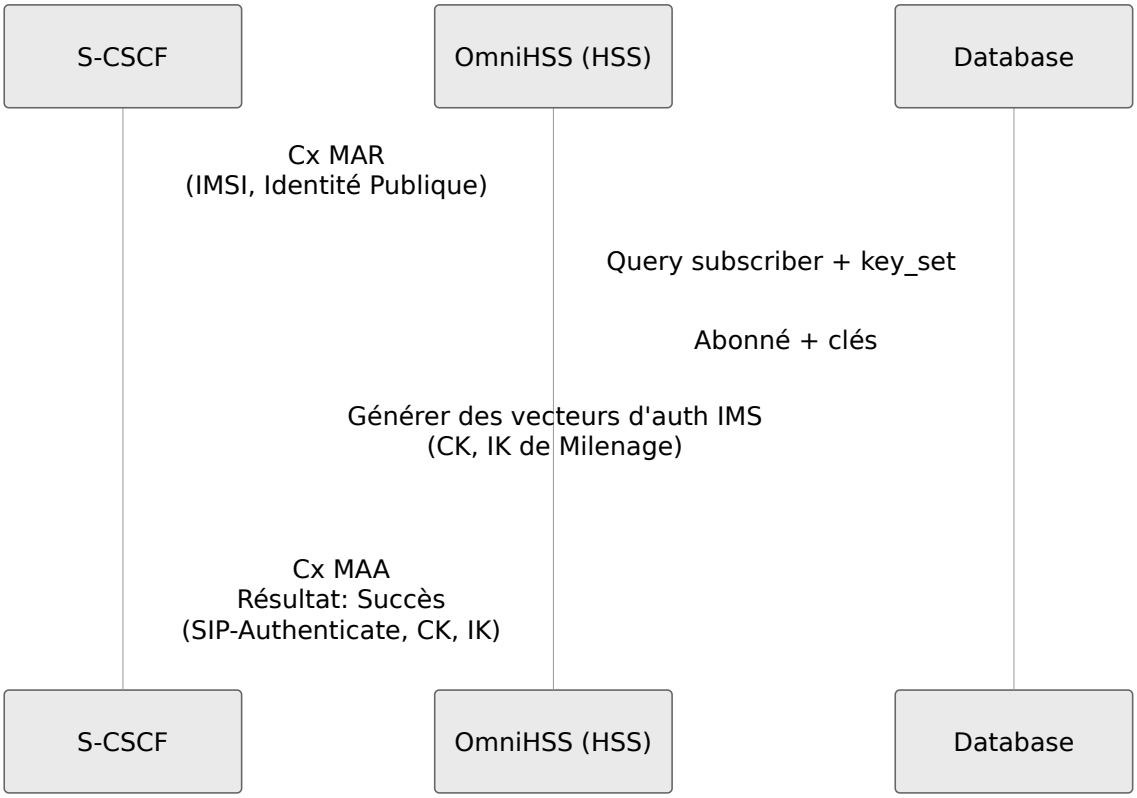
Rendu du Modèle IFC:

- `{{imsi}}` → IMSI réel
- `{{msisdns}}` → Liste des numéros de téléphone
- `{{mcc}}, {{mnc}}` → Codes PLMN d'origine

Pour savoir comment le SAR met à jour l'état d'enregistrement et l'état d'attente d'authentification, et comment il gère `AUTHENTICATION_FAILURE` et `AUTHENTICATION_TIMEOUT`, voir État d'Attribution S-CSCF.

Demande d'Auth Multimédia (MAR/MAA)

Le S-CSCF demande des vecteurs d'authentification pour l'enregistrement IMS.



État d'Attribution S-CSCF

Le HSS détient deux pièces d'état IMS séparées pour chaque abonné. L'état d'enregistrement et l'état d'attente d'authentification sont indépendants, selon 3GPP TS 29.228 §3.1.

Champ d'état	Défini à	Effacé à	Significat
`assigned_scscf`	Succès SAR (enregistrement)	Désenregistrement SAR, échec/timeout d'auth lorsque Non Enregistré, ou échec/timeout d'auth qui retire un nom obsolète	L'abonné enregistré ce S-CSCF
`auth_pending_scscf`	MAR	SAR (succès ou échec/timeout d'auth), désenregistrement UAR, ou expiration du TTL	Un défi IM AKA est en cours sur S-CSCF.
`auth_pending_since`	MAR	Avec `auth_pending_scscf`	Le momer où le défi commenc Le HSS l'utilise po expirer un défi qui ne termine jamais.

Le MAR stocke le nom S-CSCF et définit l'état d'attente d'authentification, selon TS 29.228 §6.3. Le SAR le confirme ou le retire, selon TS 29.228 §6.1.2.

Ordre de sélection S-CSCF UAA. Pour chaque UAR, le HSS retourne un nom S-CSCF et un code de résultat avec cette priorité :

1. Un S-CSCF d'attente d'authentification actif est enregistré : le retourner, avec `DIAMETER_SUBSEQUENT_REGISTRATION` (2002). Un défi est actif tant que `auth_pending_scscf` est défini et que `auth_pending_since` est à l'intérieur de `auth_pending_ttl_seconds`.

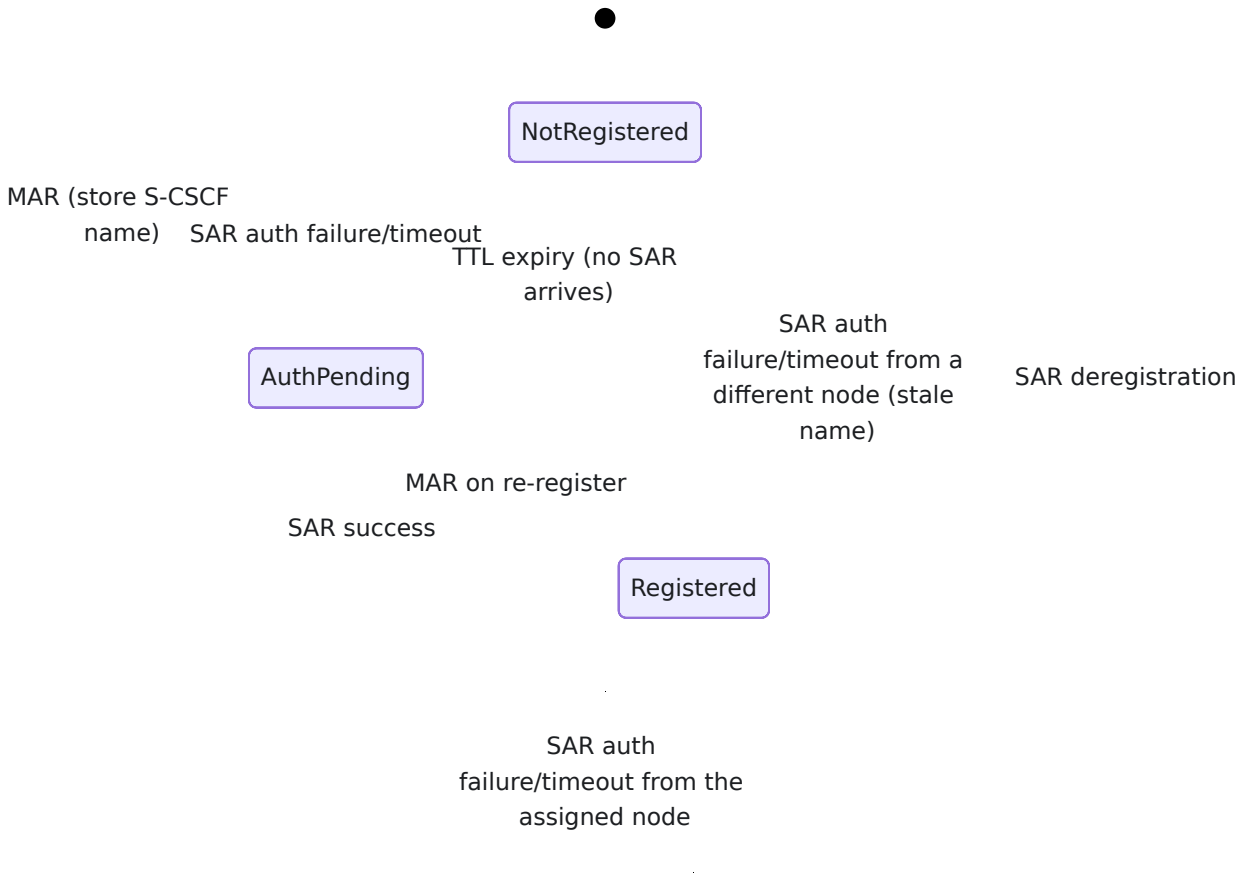
2. `assigned_scscf` détient un vrai nœud S-CSCF : le retourner, avec `DIAMETER_SUBSEQUENT_REGISTRATION` (2002).
3. Aucun des deux ne s'applique : sélectionner un nouveau S-CSCF (voir Configuration de Sélection S-CSCF), avec `DIAMETER_FIRST_REGISTRATION` (2001).

La règle 2 teste intentionnellement pour un vrai nœud. Une URI de pool est un nom pour l'ensemble du pool, donc elle n'identifie jamais le nœud qui sert l'abonné. Si une URI de pool est enregistrée d'une version antérieure, le HSS traite l'abonné comme non servi et répond 2001. Le prochain SAR stocke alors le vrai nœud.

Le réenregistrement retourne toujours le S-CSCF servant. La règle 2 rend l'attribution collante. Après qu'un nœud prenne l'enregistrement, chaque UAR de réenregistrement et chaque LIR terminant retourne ce nœud. Le HSS retourne une URI de pool uniquement pour un abonné qu'aucun nœud ne sert encore.

Le HSS préfère le S-CSCF d'attente d'authentification à `assigned_scscf`. Cela fait converger un cycle de défi en cours vers le nœud qui l'a émis. Pour un abonné enregistré en bonne santé, le nœud qui défie est le nœud attribué, donc la préférence n'a aucun effet. Cela n'a d'importance que pendant Récupération d'Échec S-CSCF.

L'état d'attente d'authentification expire. Un cycle de défi se termine en quelques secondes, et chaque résultat envoie un SAR qui efface l'état. Si le S-CSCF défiant n'envoie jamais ce SAR, le nom en attente reste enregistré. Le HSS ignore un nom en attente qui est plus ancien que `auth_pending_ttl_seconds` (30 secondes par défaut). Sans le TTL, un MAR errant ou abandonné surpasserait l'enregistrement enregistré pour toujours. Le trafic sortant irait vers un nœud qui ne détient rien, tandis que le trafic entrant continuerait d'aller vers `assigned_scscf`.



Codes de résultat Cx :

Code	Nom	Signification	Référen
2001	DIAMETER_FIRST_REGISTRATION	Aucun S-CSCF enregistré. Le nom retourné est une sélection.	TS 29.22
2002	DIAMETER_SUBSEQUENT_REGISTRATION	Un S-CSCF est enregistré. Le HSS retourne le nom stocké.	TS 29.22

Valeurs de type d'attribution de serveur pertinentes (SAR) :

Valeur	Nom	Action HSS
0	NO_ASSIGNMENT	Effacer le nom CSCF et l'enregistreme
1	REGISTRATION	Définir l'état d'enregistreme sur enregistré. Effacer l'authentificati en attente.
2	RE_REGISTRATION	Conserver l'enregistreme Effacer l'authentificati en attente.
4	TIMEOUT_DEREGISTRATION	Effacer le nom CSCF et l'enregistreme
5	USER_DEREGISTRATION	Effacer le nom CSCF et l'enregistreme
6	TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME	Terminer l'enregistreme Conserver le nom S-CSCF dans les enregistremen
7	USER_DEREGISTRATION_STORE_SERVER_NAME	Terminer l'enregistreme Conserver le

Valeur	Nom	Action HSS
		nom S-CSCF dans les enregistremen
8	ADMINISTRATIVE_DEREGISTRATION	Effacer le nom CSCF et l'enregistreme
9	AUTHENTICATION_FAILURE	Conserver une liaison enregistrée, à moins que le nom dans les enregistremen ne soit obsolèt Effacer le nom CSCF lorsque Non enregistré
10	AUTHENTICATION_TIMEOUT	Conserver une liaison enregistrée, à moins que le nom dans les enregistremen ne soit obsolèt Effacer le nom CSCF lorsque Non enregistré
11	DEREGISTRATION_TOO_MUCH_DATA	Effacer le nom CSCF et l'enregistreme

Les types 6 et 7 terminent l'enregistrement mais conservent le nom S-CSCF, conformément à TS 29.228 §6.1.2. Ce nœud continue de servir l'état non

enregistré de l'abonné, donc un UAR ultérieur le résout toujours.

Détection de nom obsolète en cas d'échec d'authentification. Un SAR `AUTHENTICATION_FAILURE` ou `AUTHENTICATION_TIMEOUT` conserve normalement la liaison d'un abonné enregistré. Il y a une exception. Si le SAR provient d'un S-CSCF différent de celui enregistré, l'I-CSCF doit avoir re-forké le REGISTER dans le pool, car il n'a pas pu atteindre le nœud enregistré. Le HSS efface alors ce nom comme obsolète. Sans cette règle, la Récupération d'échec S-CSCF ne converge pas : le HSS retournerait le nœud mort à chaque UAR ultérieur, et seul un défi que l'UE complète avant tout SAR de timeout récupérerait l'abonné.

Voir la métrique hss_cx_auth_abandon_total pour la visibilité des défis abandonnés ou échoués.

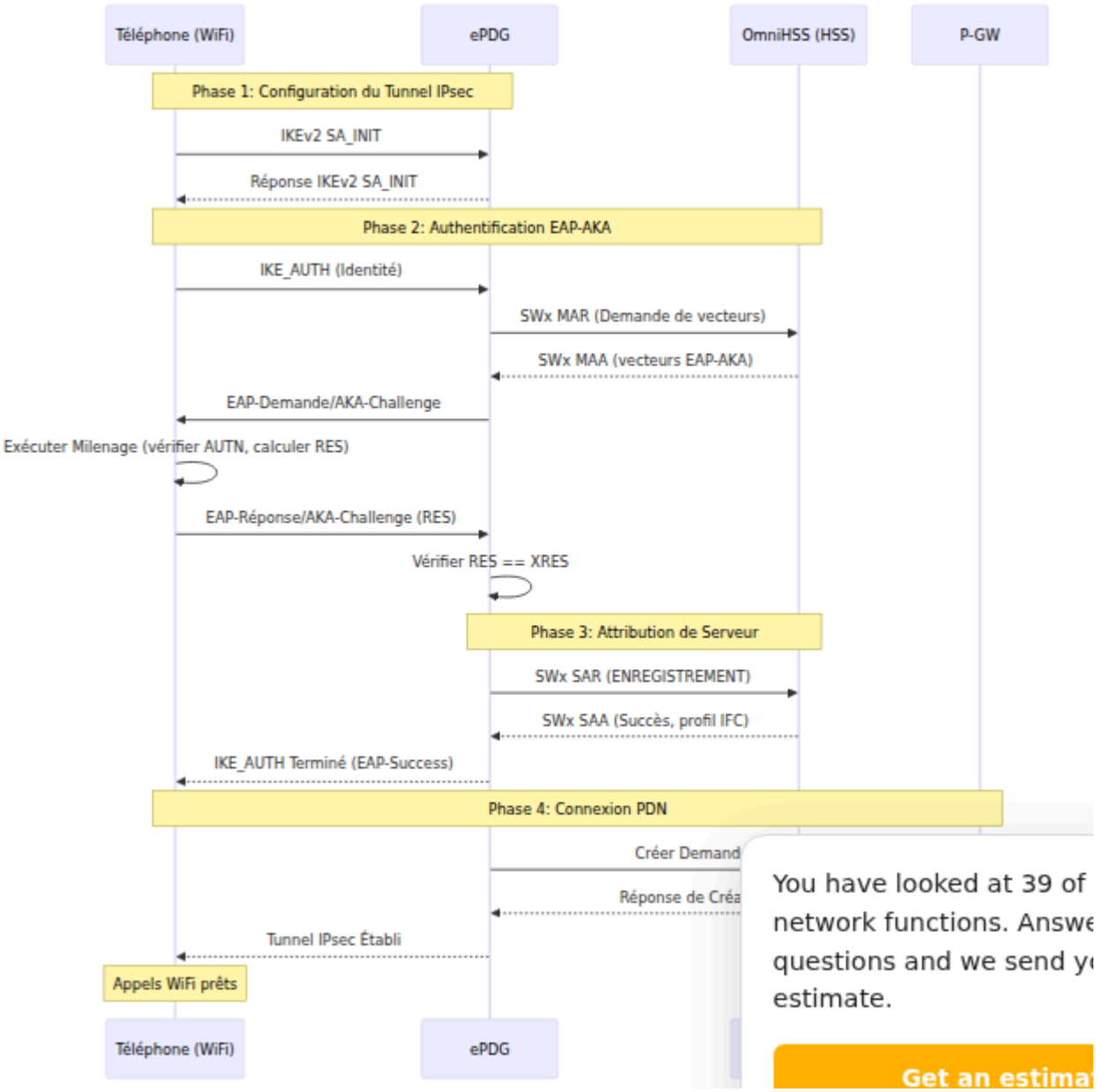
Un S-CSCF par abonnement

Un abonnement IMS partagé regroupe plusieurs IMSI, ou plusieurs MSISDN, sous une seule identité (voir Fonctionnalités Multi-MSISDN et Multi-IMSI). Un seul S-CSCF doit servir l'ensemble de l'abonnement. Si deux membres sont enregistrés sur des nœuds différents, un appel de terminaison atteindrait un nœud qui ne détient aucune liaison pour la partie appelée.

Le HSS détient un nom S-CSCF partagé sur l'abonnement IMS, appelé l'ancre :

1. Le premier membre s'enregistre. Aucune ancre n'est définie, donc le UAA retourne une sélection (une URI de pool, ou un nœud sous `:random_peer`).
2. Un nœud réel prend l'enregistrement et envoie son SAR. Le HSS ancre l'abonnement à ce nœud.
3. Chaque membre ultérieur lit l'ancre. Son UAA retourne le nœud ancré, donc chaque membre atterrit sur le même S-CSCF.
4. L'ancre est libérée lorsque le dernier membre se désinscrit.

Le HSS n'ancre jamais une URI de pool. Une URI de pool nomme le pool, pas un nœud, donc le HSS la conserve juste assez longtemps pour que l'I-CSCF la résolve. Si une ancre détient une URI de pool d'une version antérieure, le prochain SAR la remplace par le véritable nœud de service.



Récupération d'échec S-CSCF

Si le S-CSCF enregistré n'est pas accessible, l'I-CSCF renvoie le REGISTER au pool S-CSCF. Un S-CSCF actif sert alors l'abonné. Cela nécessite la route de secours du pool I-CSCF et la préférence `auth_pending_scscf` du HSS ensemble.

Il y a deux modes d'échec :

Mode d'échec	Comment il se présente	Récupération
Nom S-CSCF non résolvable (un nom décommissionné ou obsolète dans les données HSS)	DNS retourne NXDOMAIN. L'échec de transfert de l'I-CSCF se produit avant qu'il ne crée une branche.	L'I-CSCF renvoie au pool URI. DNS SRV sélectionne un S-CSCF actif.
S-CSCF résolvable mais mort (le service S-CSCF est hors service)	TCP connection-refused. La branche échoue de manière asynchrone.	L'I-CSCF renvoie au pool URI depuis sa route d'échec. `use_dst_blocklist` et `use_dns_failover` font que DNS ignore le nœud mort.



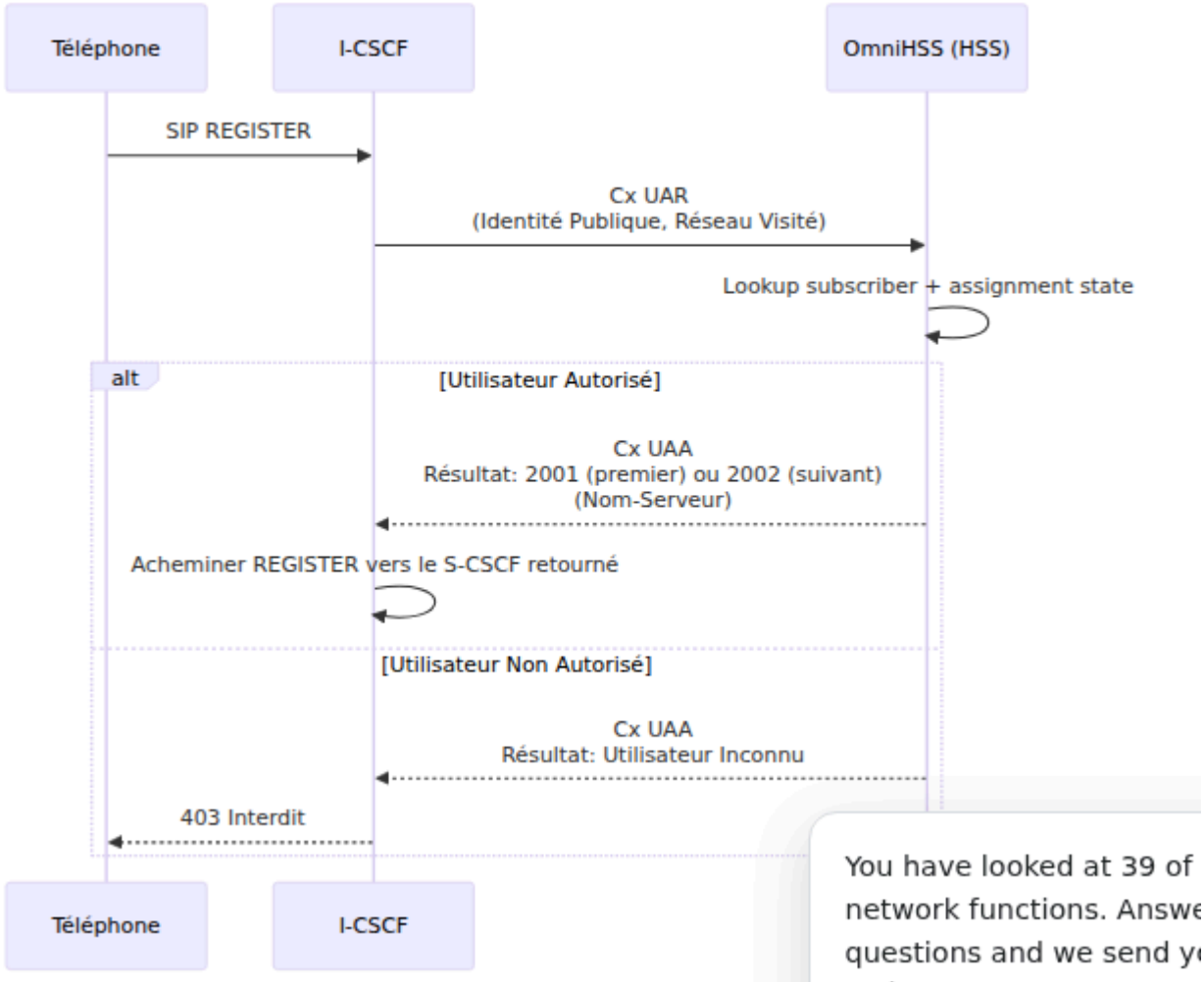
Le REGISTER authentifié doit atteindre le même S-CSCF actif qui a émis le défi. La préférence `auth_pending_scscf` dans le UAA permet cela : après que le nœud actif envoie son MAR, le HSS retourne ce nœud sur le prochain UAR, et non le nœud mort enregistré. Sans la préférence, les deux REGISTER iraient à des membres de pool différents et le cycle ne convergerait pas.

Limitation. Si le S-CSCF est tué pendant le cycle de défi IMS-AKA, l'UE est re-challengé par le nouveau S-CSCF avec un nouveau nonce. L'UE se rétablit lors de son prochain enregistrement. Cela est inhérent à un changement de S-CSCF en cours de défi, car le vecteur d'authentification en cours ne peut pas être transféré à un S-CSCF différent.

Pour permettre cette récupération, définissez `selection_method: :pool_uri` (voir Configuration de sélection S-CSCF) et déployez la route de secours I-CSCF.

Demande d'informations de localisation (LIR/LIA)

L'I-CSCF interroge quel S-CSCF sert l'utilisateur.

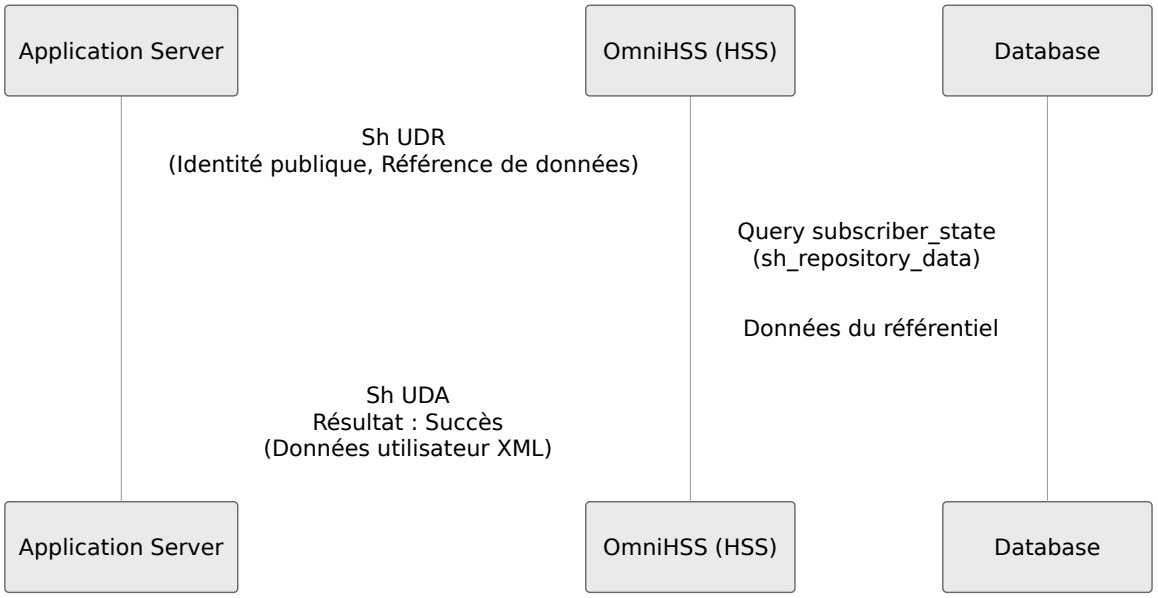


Interface Sh (Données de profil IMS)

Voir Sh RepositoryData (Données transparentes) pour le modèle de données, les règles de numéro de séquence/concurrence, et l'API de provisionnement REST pour les documents MMTEL-Services / IMS-ODB-Information.

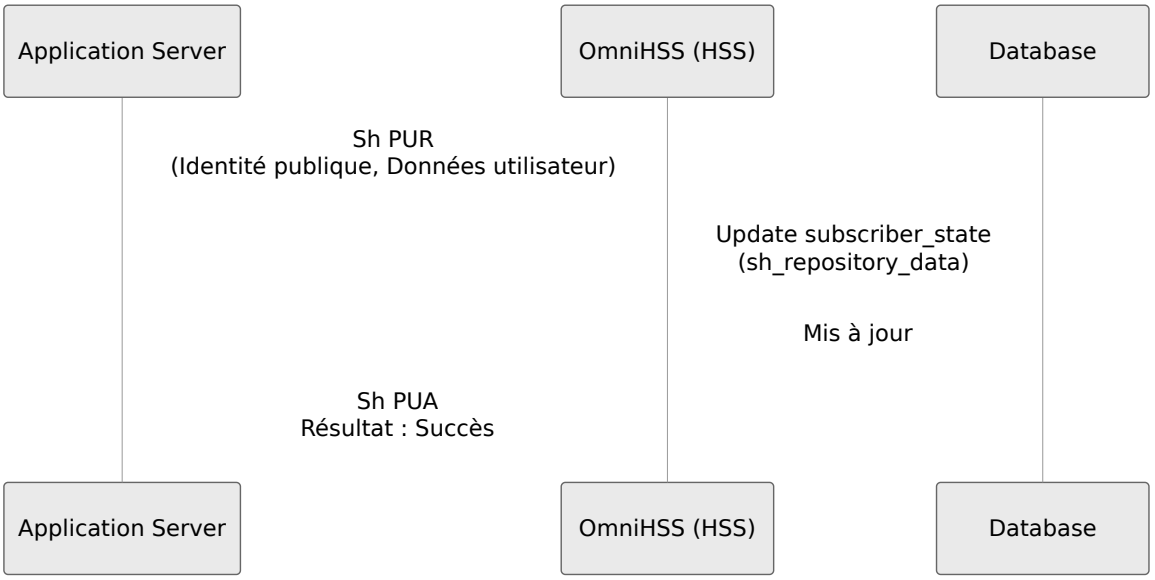
Demande de données utilisateur (UDR/UDA)

Le serveur d'application demande des données de profil d'abonné.



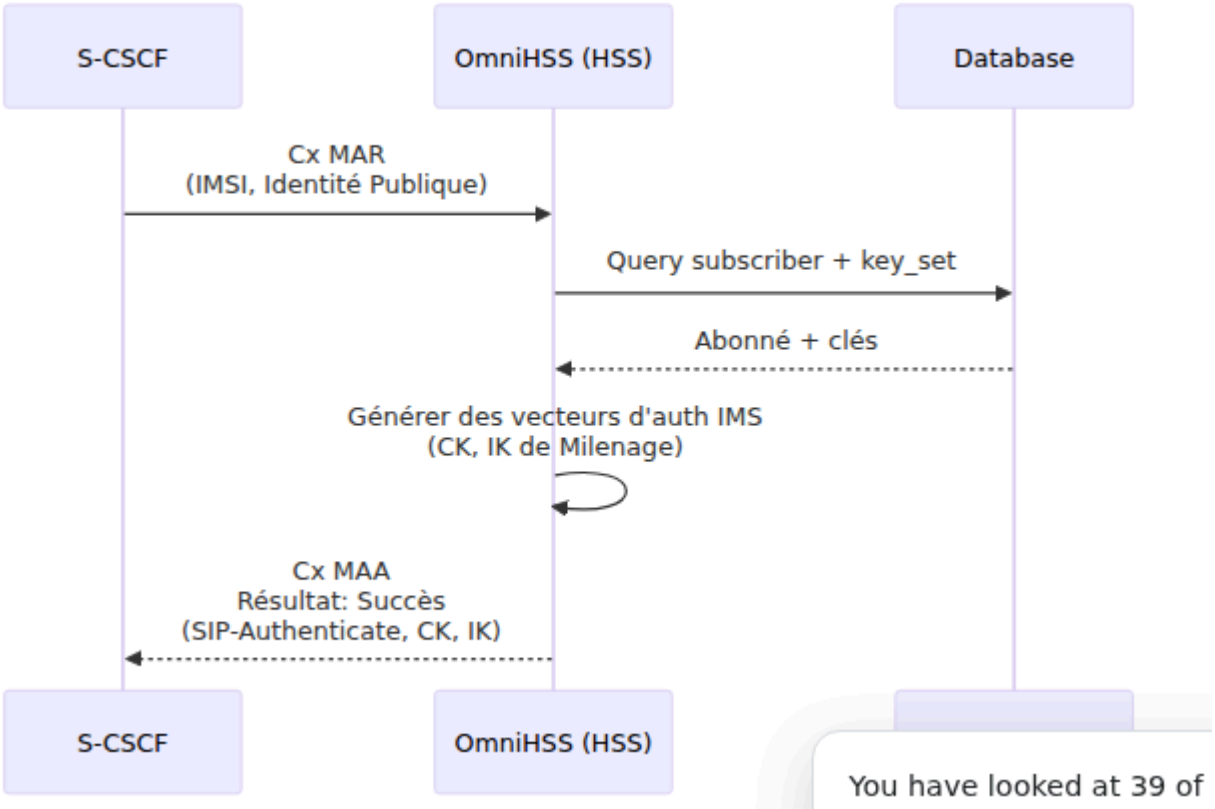
Demande de mise à jour de profil (PUR/PUA)

Le serveur d'application met à jour les données de profil d'abonné.



Demande de notifications d'abonnement (SNR/SNA)

Le serveur d'application s'abonne aux changements de profil.



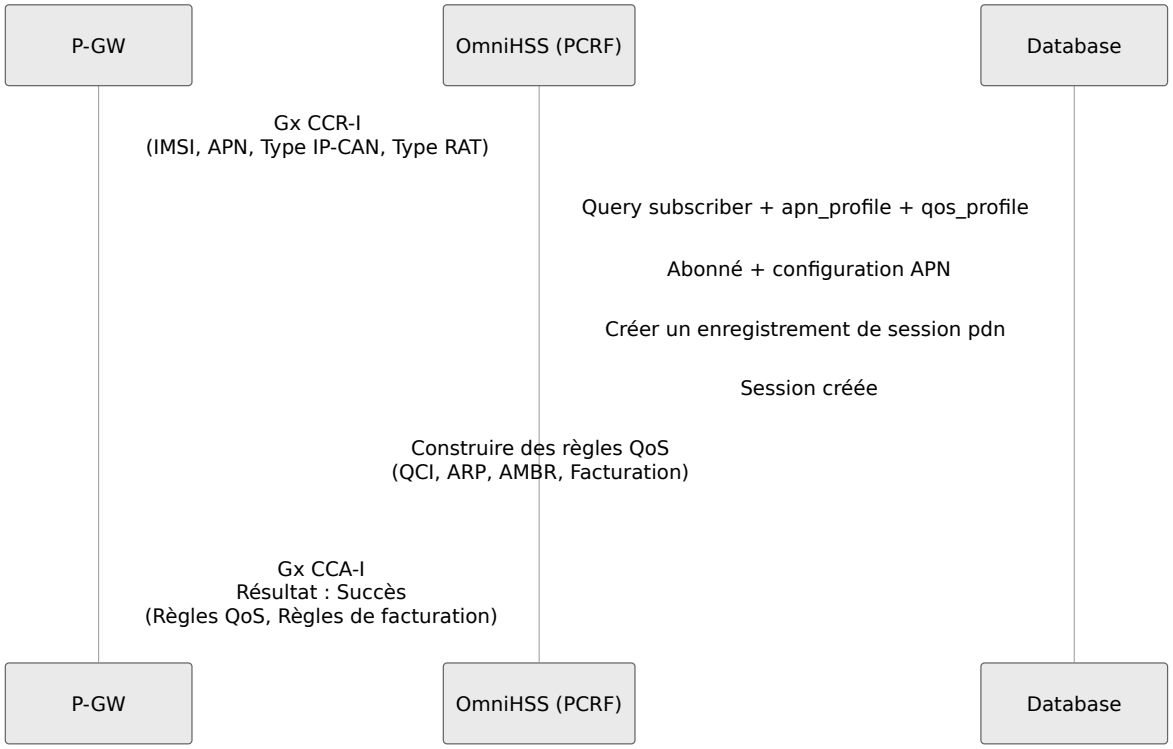
Interface Gx (Contrôle de politique)

OmniHSS fonctionne comme le PCRF (Fonction de règles de politique et de facturation) via l'interface Gx.

Voir Documentation PCRF pour l'architecture détaillée, la configuration de politique, et la gestion de QoS.

Demande de contrôle de crédit - Initiale (CCR-I/CCA-I)

Le P-GW demande des règles de politique lorsque la session PDN est établie.

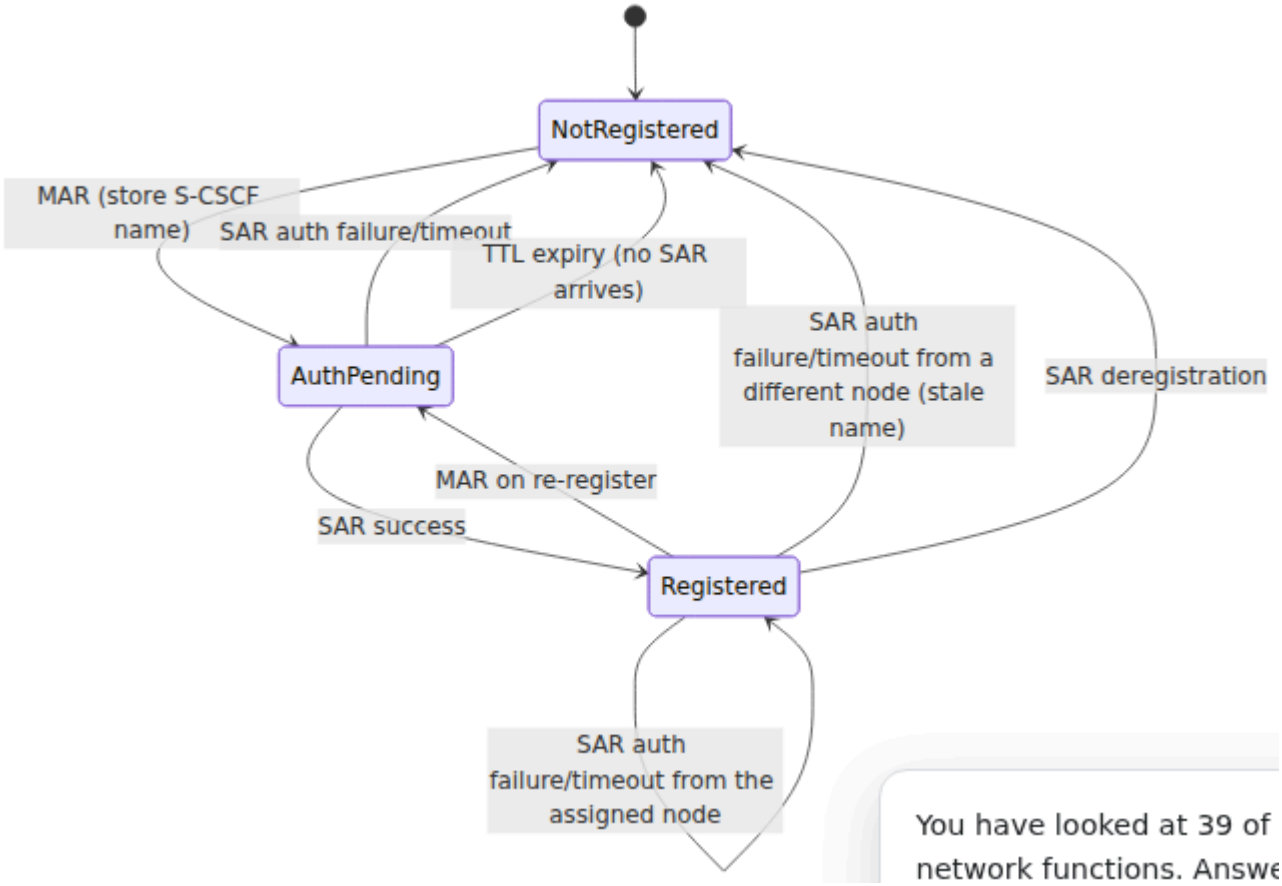


AVPs clés :

- Demande : Subscription-Id (IMSI), Called-Station-Id (APN), RAT-Type, IP-CAN-Type
- Réponse : QoS-Information (QCI, ARP, AMBR), Charging-Rule-Install

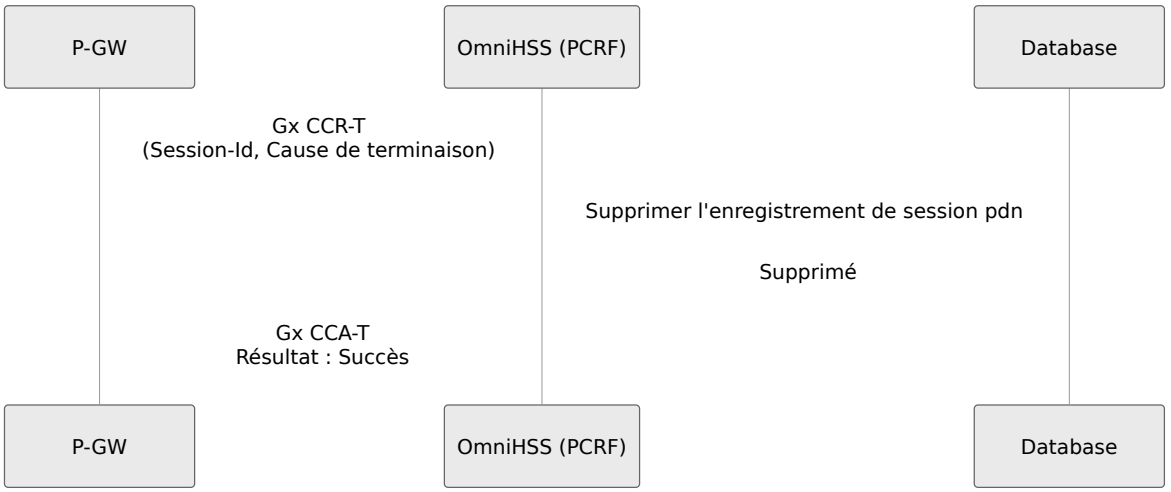
Demande de contrôle de crédit - Mise à jour (CCR-U/CCA-U)

Le P-GW notifie des changements de session.



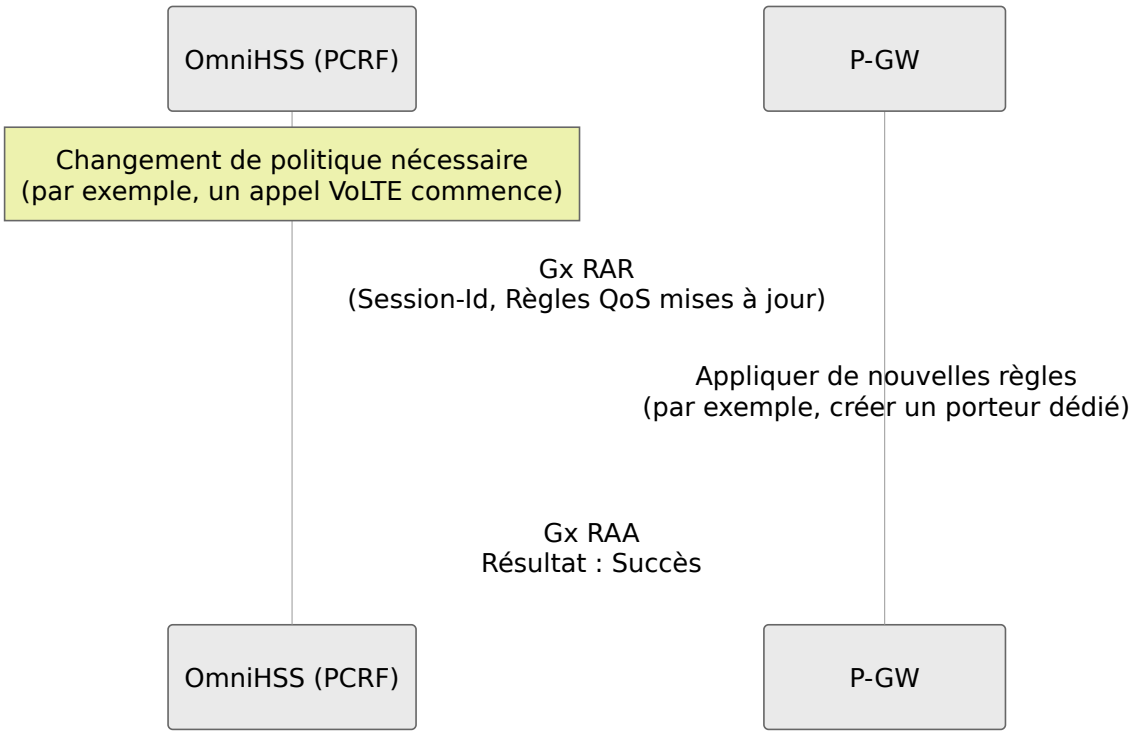
Demande de contrôle de crédit - Terminer (CCR-T/CCA-T)

Le P-GW notifie lorsque la session PDN se termine.



Demande de ré-authentification (RAR/RAA)

OmniHSS (PCRF) initie la mise à jour de politique au P-GW.



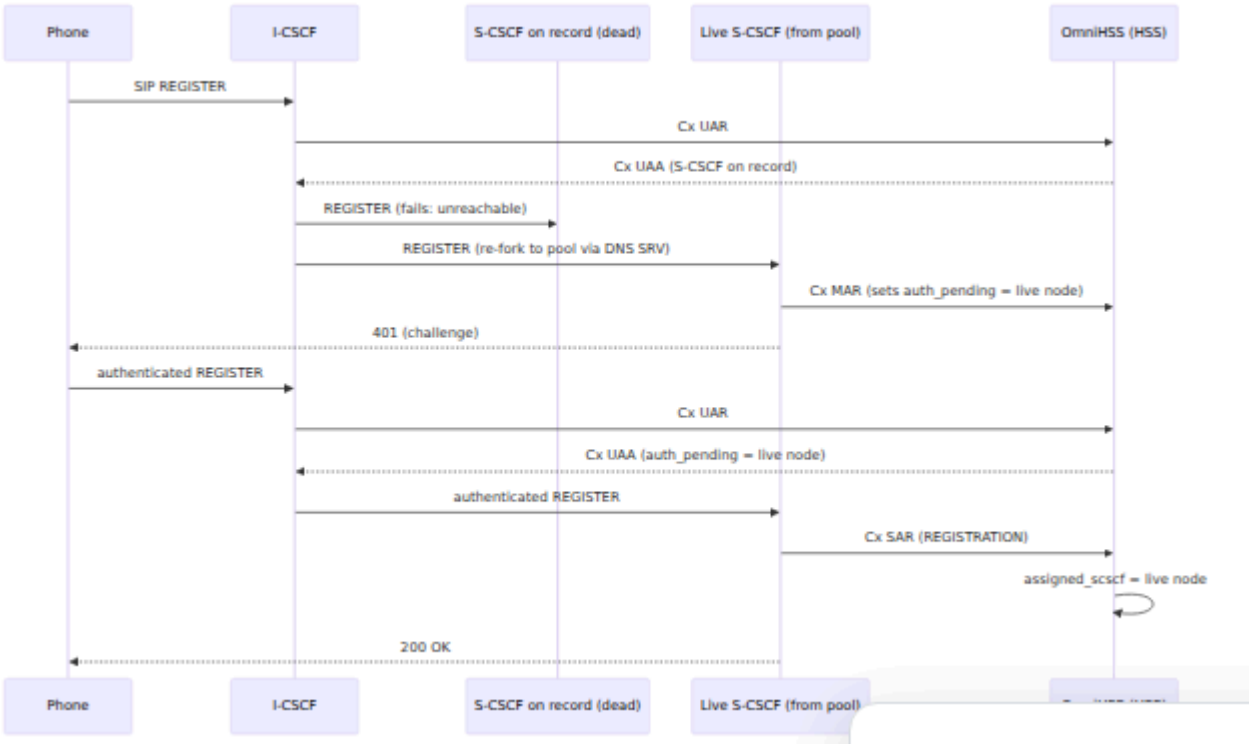
Interface Rx (Politique média IMS)

OmniHSS fonctionne comme le PCRF via l'interface Rx pour l'autorisation média IMS.

Voir Documentation PCRF pour les flux d'appels VoLTE détaillés et l'autorisation média.

Demande AA (AAR/AAA)

Le P-CSCF demande l'autorisation média pour la session IMS.

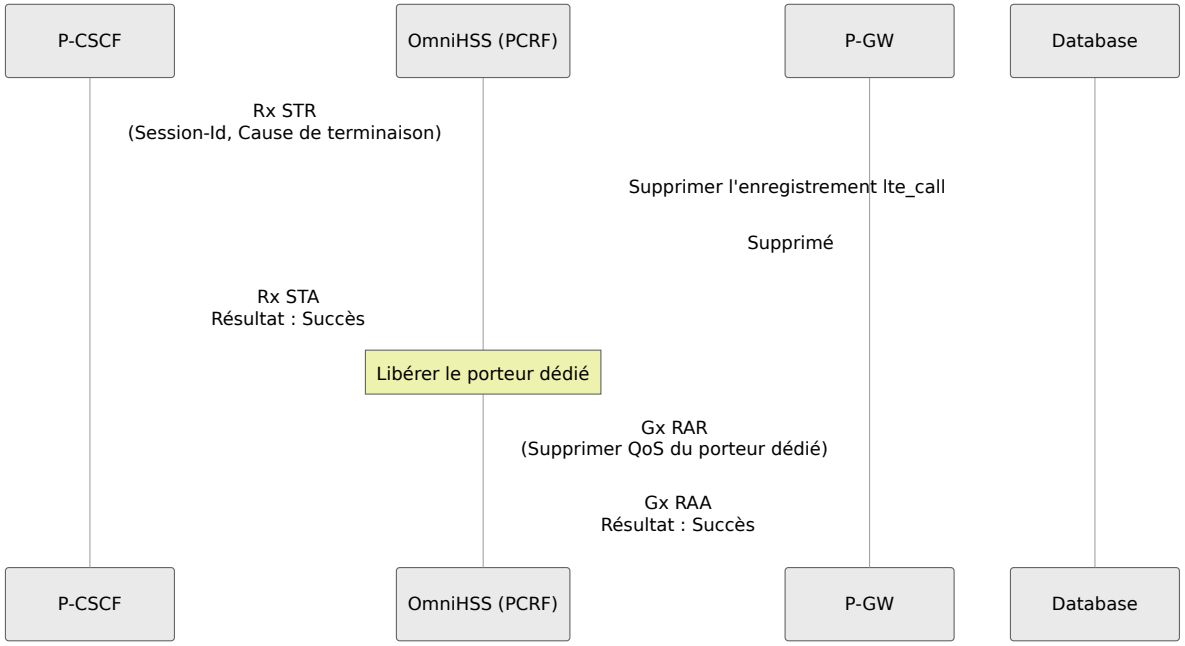


Informations clés :

- Analyser SDP pour déterminer le codec et la bande passante
- Calculer la bande passante requise (UL/DL)
- Créer des filtres SDF pour les flux média
- Déclencher le porteur dédié via Gx RAR

Demande de terminaison de session (STR/STA)

Le P-CSCF notifie lorsque la session IMS se termine.

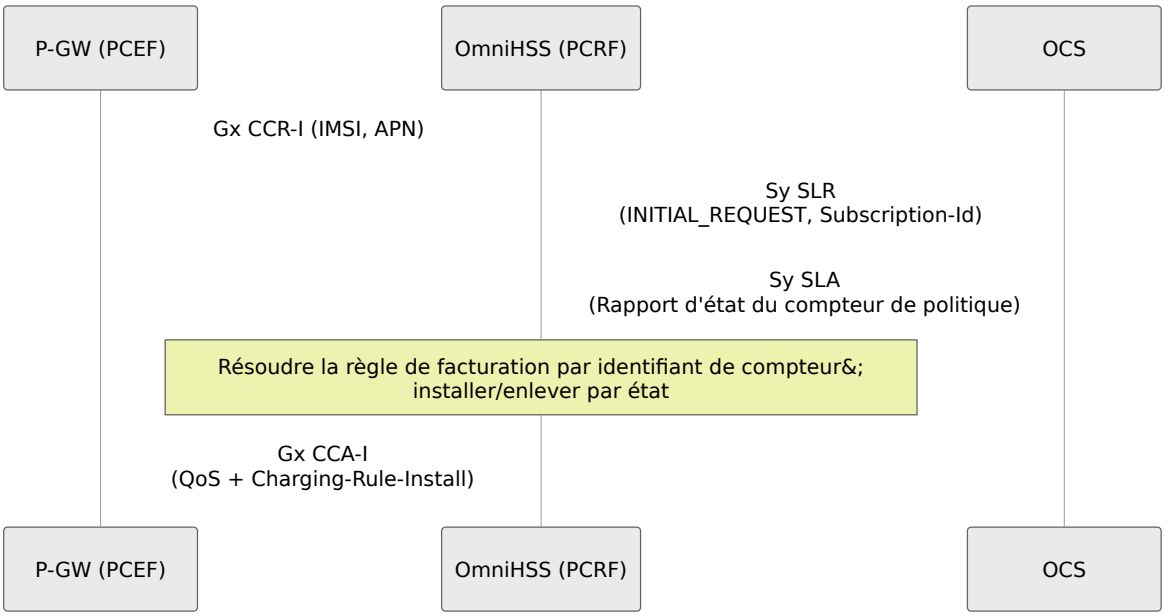


Interface Sy (Contrôle de limite de dépenses)

OmniHSS, agissant comme le PCRF, utilise l'interface Sy pour obtenir l'état de limite de dépenses de l'abonné d'un OCS externe (Système de facturation en ligne) et appliquer une politique basée sur les dépenses via Gx. Sy utilise l'identifiant d'application Diameter `16777302` (3GPP TS 29.219). Le HSS est le client Sy ; l'OCS est le serveur. Voir Interface Sy pour la référence complète (mapping des compteurs de politique, AVPs, configuration).

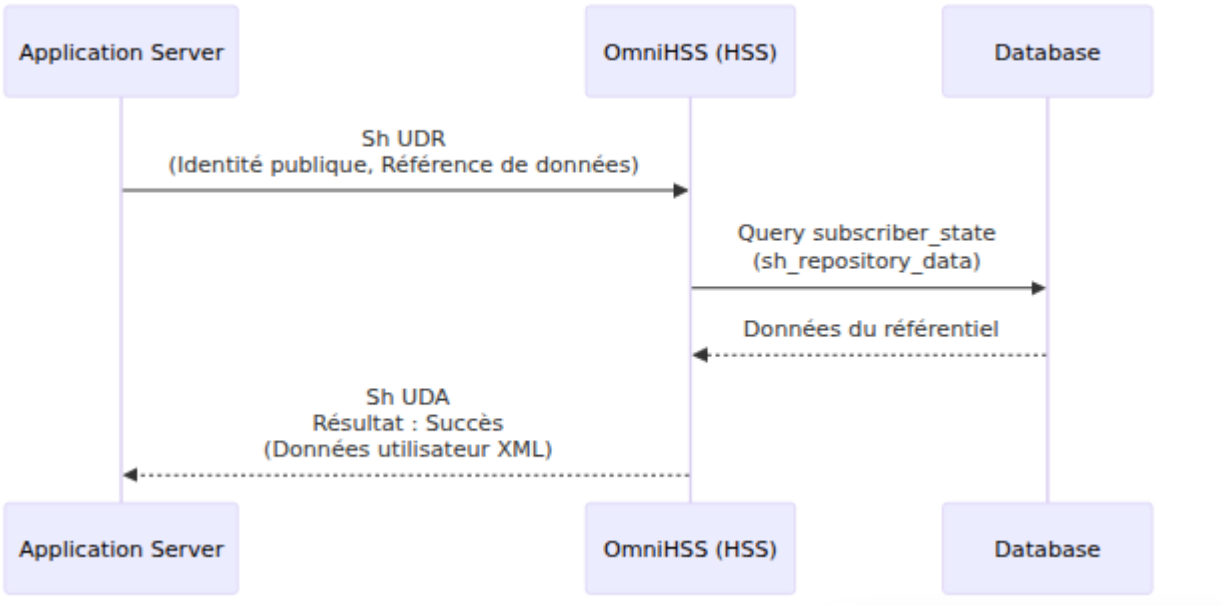
Demande de limite de dépenses (SLR/SLA)

Le PCRF ouvre une session Sy par abonné pour récupérer l'état actuel des compteurs de politique de l'OCS, puis intègre les règles résultantes dans le Gx CCA qu'il retourne déjà.



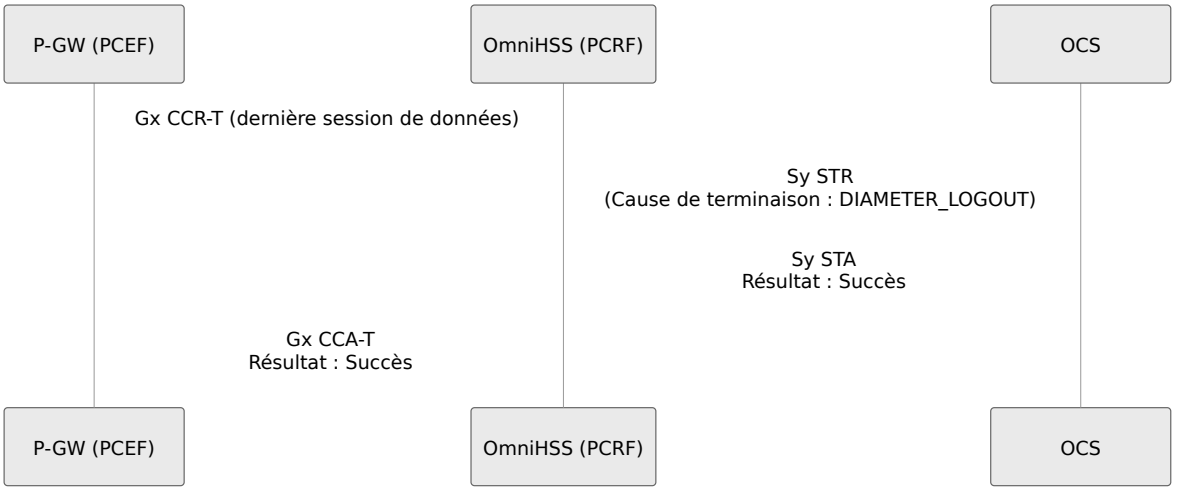
Notification d'état de dépenses (SNR/SNA)

Lorsqu'un état de compteur change (par exemple, une allocation de données est épuisée), l'OCS notifie le PCRF, qui pousse le changement de politique au PGW avec un Gx RAR.



Terminaison de session (STR/STA)

Lorsque la dernière session de données de l'abonné se termine, le PCRF ferme la session Sy.



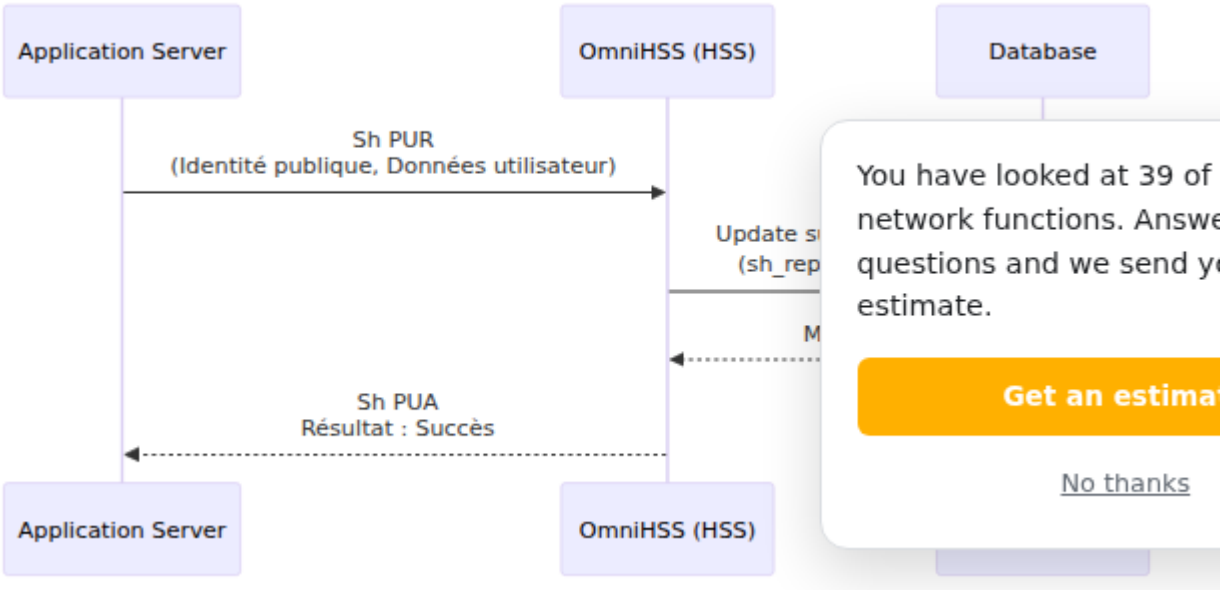
Interface S13 (EIR)

OmniHSS fonctionne comme l'EIR (Registre d'identité d'équipement) via l'interface S13.

Voir Documentation EIR pour le contrôle détaillé de l'identité d'équipement, la validation IMEI, la gestion de liste noire, et priorité des règles.

Demande de vérification d'identité ME (ECR/ECA)

Le client EIR externe (ou MME) demande la validation de l'équipement.



Valeurs d'état de l'équipement :

- **Liste blanche (0)** - Appareil autorisé
- **Liste noire (1)** - Appareil bloqué
- **Liste grise (2)** - Appareil autorisé mais suivi

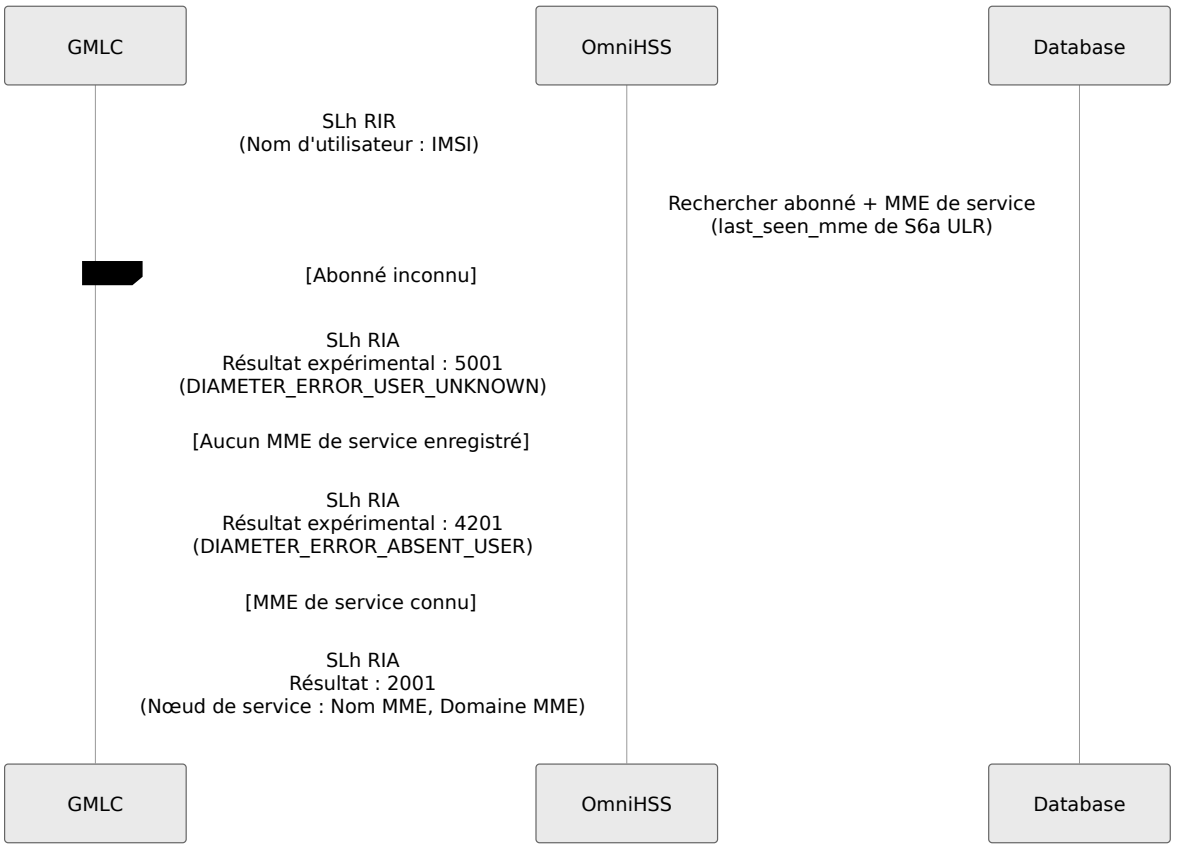
Si l'IMSI dans la demande est inconnu ou désactivé, ou si aucune règle ne correspond et que `unknown_equipment_behaviour` est `:reject_unknown_equipment`, l'ECA porte le code de résultat expérimental 5422 (DIAMETER_ERROR_EQUIPMENT_UNKNOWN) au lieu d'un État de l'équipement.

Interface SLh (LCS / Services de localisation)

OmniHSS répond à la demande `LCS-Routing-Info-Request` (RIR) SLh afin qu'un GMLC puisse découvrir quel MME sert actuellement un abonné avant de demander leur localisation via SLg. Le MME en service est celui enregistré lors de l'enregistrement S6a (Update-Location). SLh utilise l'identifiant d'application Diameter `16777291` (3GPP TS 29.173).

Demande d'informations de routage LCS (RIR/RIA)

Le GMLC demande le nœud de service pour un abonné.



Remarques :

- L'IMSI est pris à partir de `User-Name` ; une recherche MSISDN est utilisée comme solution de secours.
- Le `Nœud de service` retourné porte le `Nom MME`/`Domaine MME` enregistré en dernier par les gestionnaires S6a AIR/ULR (`subscriber_state.last_seen_mme` / `last_seen_realm`) - aucun provisionnement supplémentaire n'est requis.
- Un GMLC suit généralement un RIA réussi avec une demande SLg Provide-Location-Request au MME retourné.

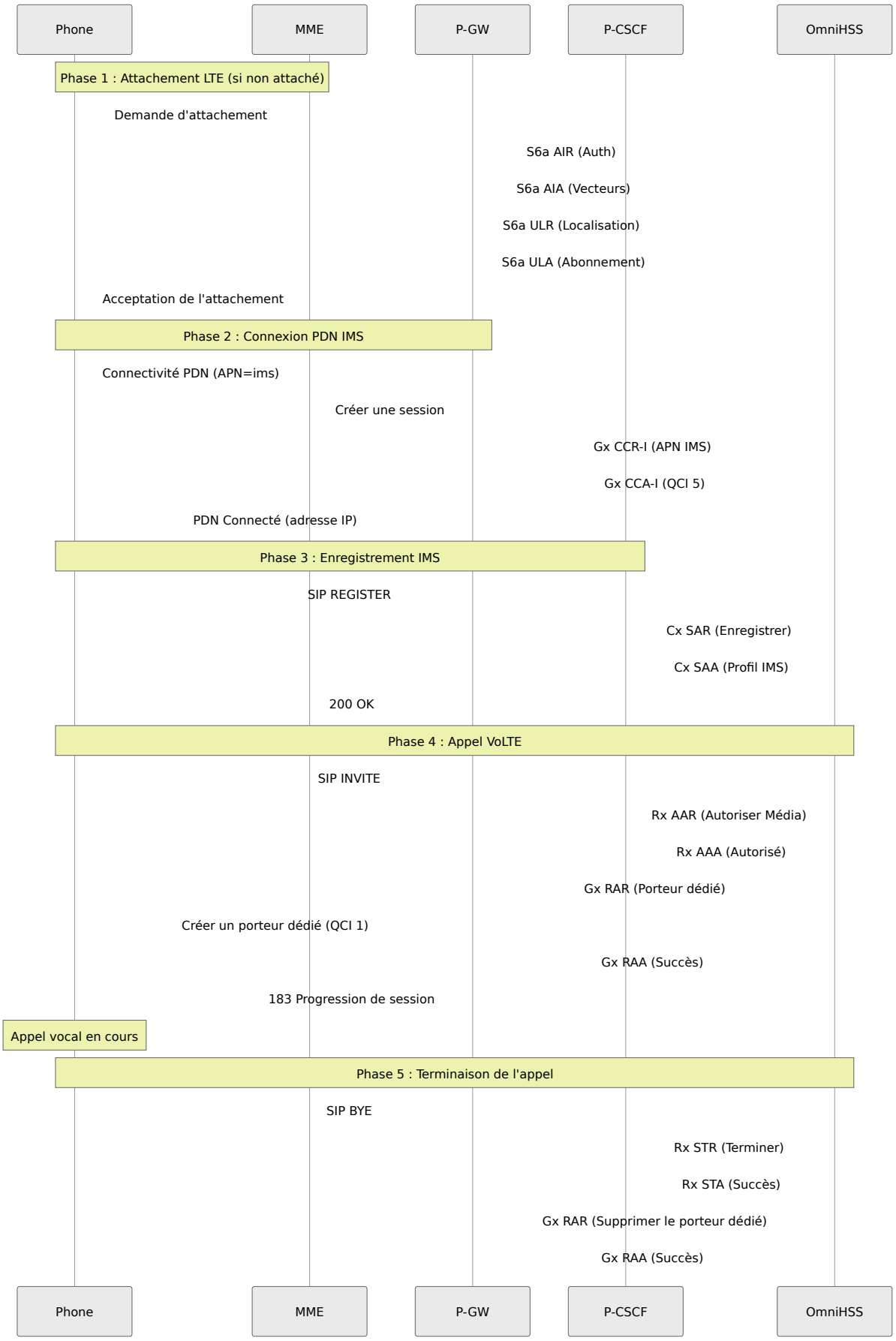
Codes de résultat :

- **2001** - Succès ; `Nœud de service` retourné.
- **5001** - `DIAMETER_ERROR_USER_UNKNOWN` ; IMSI/MSISDN non provisionné.

- **4201** - `DIAMETER_ERROR_ABSENT_USER` ; abonné connu mais aucun MME de service enregistré (jamais attaché, ou localisation annulée).

Flux d'appel complet : Appel VoLTE

Configuration d'appel VoLTE de bout en bout montrant plusieurs interfaces.

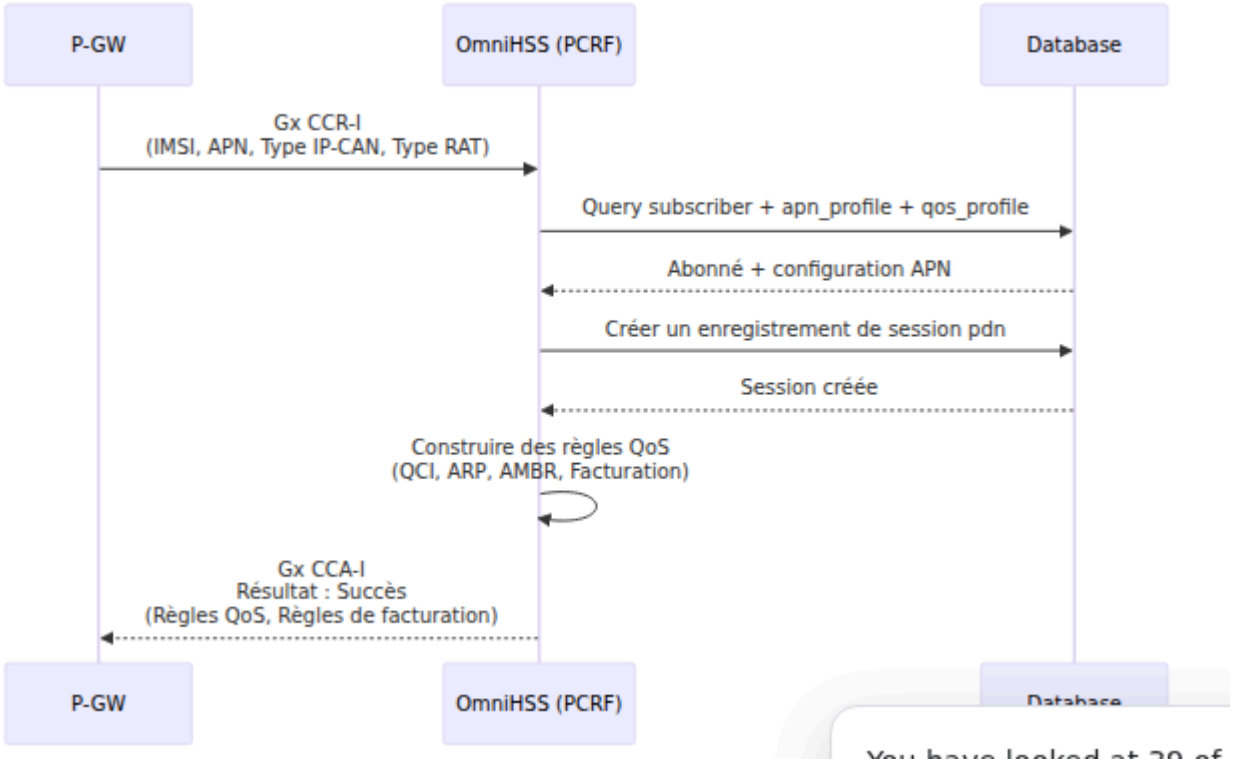

Phone
MME
P-GW
P-CSCF
OmniHSS
Phase 1 : Attachement LTE (si non attaché)
Demande d'attachement
S6a AIR (Auth)
S6a AIA (Vecteurs)
S6a ULR (Localisation)
S6a ULA (Abonnement)
Acceptation de l'attachement
Phase 2 : Connexion PDN IMS
Connectivité PDN (APN=ims)
Créer une session
Gx CCR-I (APN IMS)
Gx CCA-I (QCI 5)
PDN Connecté (adresse IP)
Phase 3 : Enregistrement IMS
SIP REGISTER
Cx SAR (Enregistrer)
Cx SAA (Profil IMS)
200 OK
Phase 4 : Appel VoLTE
SIP INVITE
Rx AAR (Autoriser Média)
Rx AAA (Autorisé)
Gx RAR (Porteur dédié)
Créer un porteur dédié (QCI 1)
Gx RAA (Succès)
183 Progression de session
Appel vocal en cours
Phase 5 : Terminaison de l'appel
SIP BYE
Rx STR (Terminer)
Rx STA (Succès)
Gx RAR (Supprimer le porteur dédié)
Gx RAA (Succès)
Phone
MME
P-GW
P-CSCF
OmniHSS

Interface Zh (Architecture de Démarrage Générique)

L'interface Zh permet l'Architecture de Démarrage Générique (GBA) en connectant la Fonction de Serveur de Démarrage (BSF) au HSS pour l'authentification des abonnés et la dérivation des clés.

Demande d'Authentification Multimédia (MAR/MAA)

Le BSF demande des vecteurs d'authentification GBA pour le démarrage.



AVPs Clés :

- Demande : Nom d'Utilisateur (IMPI), Identité-Publique, SIP-Auth-Data-Item (schéma d'authentification), Domaine-Destination
- Réponse : SIP-Auth-Data-Item (SIP-Authenticate, SIP-Authorization, Confidentiality-Key, Integrity-Key), GBA-UserSecSettings

Cas d'Utilisation GBA :

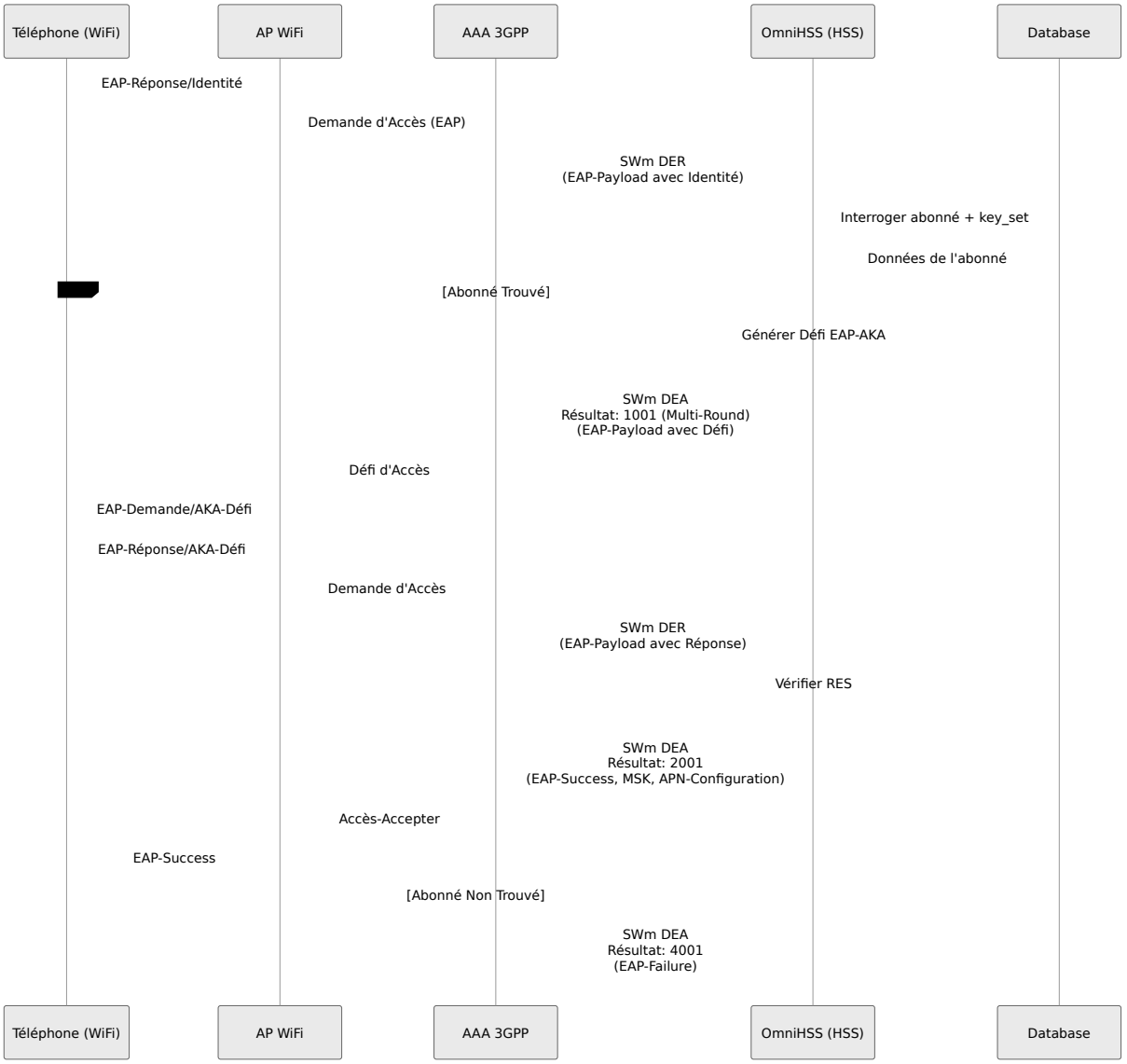
Application	Description
MBMS	Gestion des clés du Service de Diffusion Multicast Mobile
HTTPS	Provisionnement de certificats basé sur GBA
NAF	Authentification de la Fonction d'Application Réseau Générique
Présence	Authentification sécurisée du service de présence

Interface SWm (Accès Non Fiable Non-3GPP)

L'interface SWm permet l'authentification EAP-AKA pour les réseaux d'accès non fiables non-3GPP (par exemple, WiFi via ePDG) en connectant le Serveur AAA 3GPP au HSS.

Demande Diameter-EAP (DER/DEA)

Échange d'authentification EAP pour l'accès WiFi.

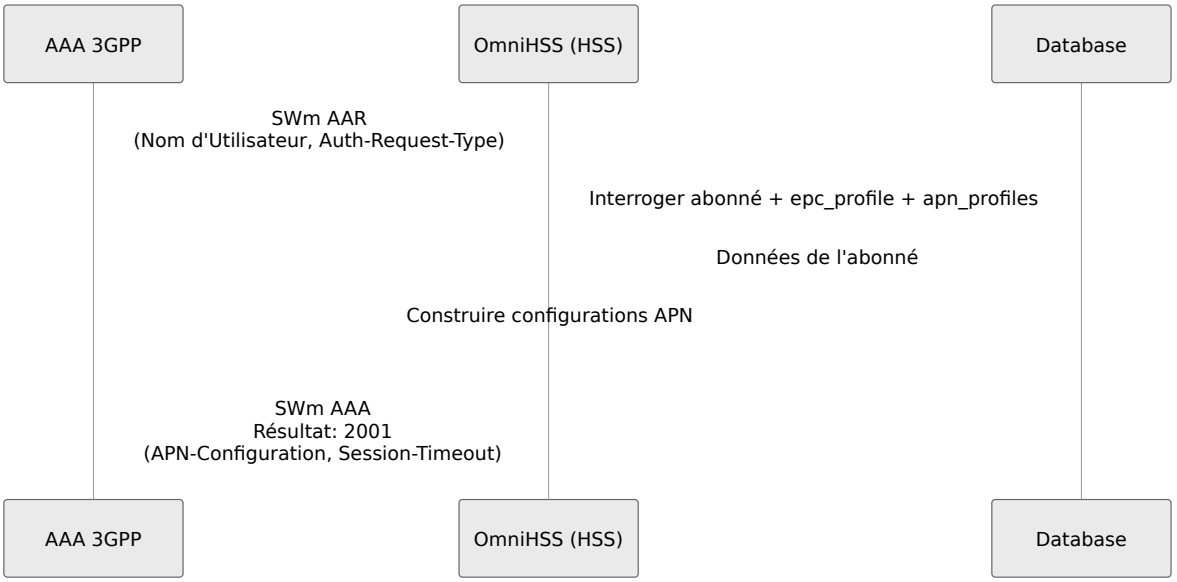


AVPs Clés :

- Demande : Session-Id, Auth-Application-Id, EAP-Payload, Auth-Request-Type, Nom d'Utilisateur (optionnel)
- Réponse : EAP-Payload, EAP-Master-Session-Key (MSK), APN-Configuration, Session-Timeout

Demande AA (AAR/AAA)

Autorisation après une authentification EAP réussie.

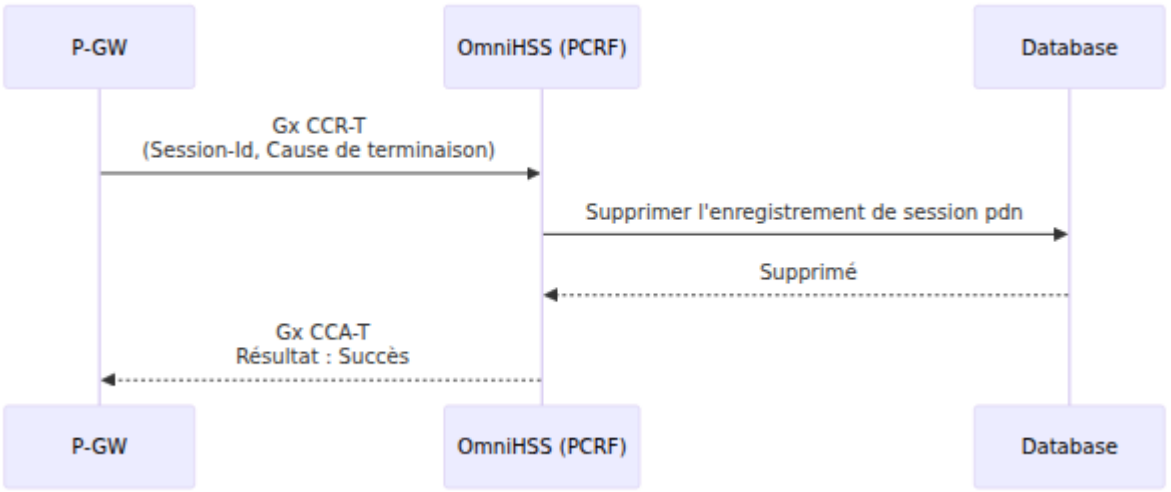


AVPs Clés :

- Demande : Session-Id, Auth-Application-Id, Nom d'Utilisateur, Auth-Request-Type, AAR-Flags
- Réponse : APN-Configuration, 3GPP-Charging-Characteristics, Session-Timeout

Demande de Terminaison de Session (STR/STA)

Notification de terminaison de session WiFi.



Causes de Terminaison :

Cause	Valeur	Description
DIAMETER_LOGOUT	1	Déconnexion de l'utilisateur
DIAMETER_SERVICE_NOT_PROVIDED	2	Service indisponible
DIAMETER_BAD_ANSWER	3	Erreur de protocole
DIAMETER_ADMINISTRATIVE	4	Terminaison administrative
DIAMETER_LINK_BROKEN	5	Connexion perdue
DIAMETER_AUTH_EXPIRED	6	Délai d'authentification
DIAMETER_USER_MOVED	7	Utilisateur déplacé vers un autre accès
DIAMETER_SESSION_TIMEOUT	8	Délai d'expiration de la session

Résolution des Problèmes de Protocole

Échecs d'Authentification (S6a AIR)

Vérifier :

1. Ensemble de clés configuré correctement (Ki, OPC, AMF)
2. Synchronisation SQN (en cas d'échecs répétés)
3. Règles de roaming autorisent le réseau visité

Échecs de Mise à Jour de Localisation (S6a ULR)

Vérifier :

1. Profil EPC existe et a des APNs configurés
2. Roaming autorisé pour les services de données
3. Format d'identité MME correct

Échecs d'Enregistrement IMS (Cx SAR)

Vérifier :

1. Profil IMS attribué à l'abonné
2. Modèle IFC XML valide
3. Sélection S-CSCF configurée
4. MSISDN attribués s'ils sont utilisés dans le modèle

Échecs de Connexion PDN (Gx CCR-I)

Vérifier :

1. APN existe dans la liste APN du profil EPC
2. Profil QoS APN configuré
3. Table de session PDN non pleine (si des limites existent)

← Retour au Guide des Opérations

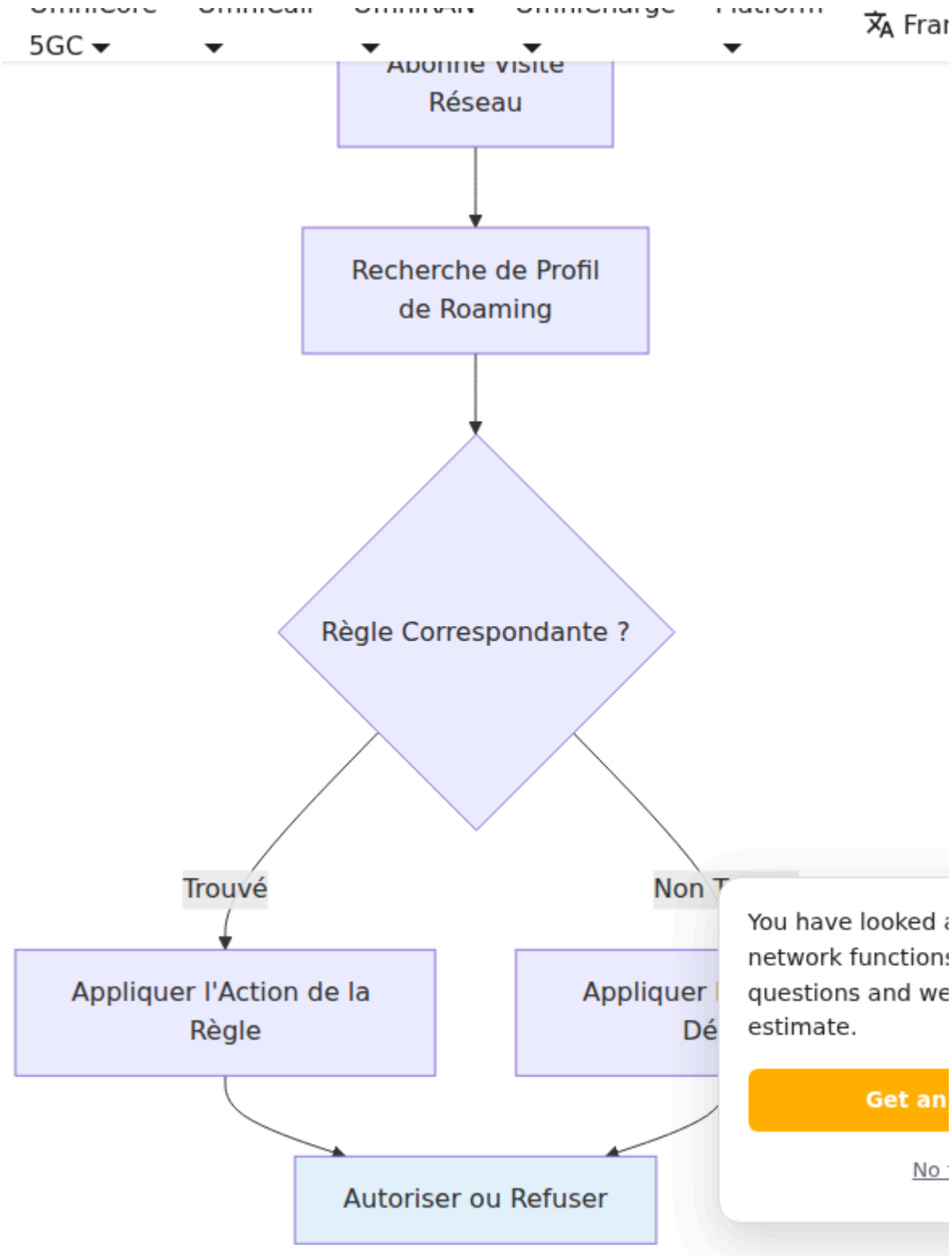
Contrôle de Roaming OmniHSS

← Retour au Guide des Opérations

Vue d'ensemble

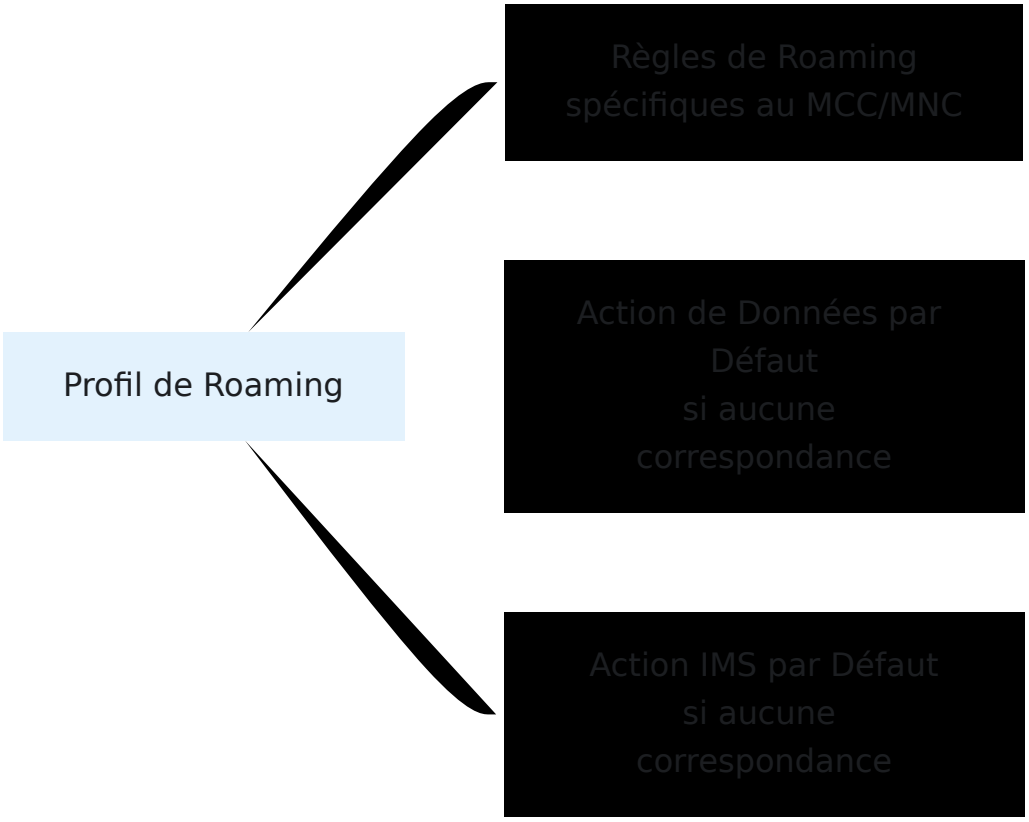
OmniHSS fournit un contrôle de roaming granulaire, vous permettant de définir quels réseaux les abonnés peuvent accéder pour les services de données et IMS lors du roaming.

Flux de Contrôle de Roaming



Structure du Profil de Roaming

Composants



Règle de Roaming

Chaque règle spécifie une action pour un réseau spécifique (combinaison MCC/MNC).

Champs :

- `name` - Nom descriptif
- `mcc` - Code Pays Mobile (3 chiffres)
- `mnc` - Code Réseau Mobile (2-3 chiffres)
- `data_action` - "allow" ou "deny"
- `ims_action` - "allow" ou "deny"

Profil de Roaming

Définit le comportement par défaut et lie aux règles.

Champs :

- `name` - Nom du profil
- `data_action_if_no_rules_match` - "allow" ou "deny"
- `ims_action_if_no_rules_match` - "allow" ou "deny"

Exemples de Configuration

Autoriser Tout le Roaming

```
# Créer un profil qui autorise tout
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_profile": {
      "name": "Autoriser Tout",
      "data_action_if_no_rules_match": "allow",
      "ims_action_if_no_rules_match": "allow",
      "roaming_rules": []
    }
  }'
```

Refuser Tout le Roaming

```
# Créer un profil qui bloque tout
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_profile": {
      "name": "Pas de Roaming",
      "data_action_if_no_rules_match": "deny",
      "ims_action_if_no_rules_match": "deny",
      "roaming_rules": []
    }
  }'
```

Autoriser des Réseaux Spécifiques (Liste Blanche)

```
# Créer une règle AT&T
RULE1=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Autoriser AT&T",
      "mcc": "310",
      "mnc": "410",
      "data_action": "allow",
      "ims_action": "allow"
    }
  }' | jq -r '.response.id')

# Créer une règle Verizon
RULE2=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Autoriser Verizon",
      "mcc": "311",
      "mnc": "480",
      "data_action": "allow",
      "ims_action": "allow"
    }
  }' | jq -r '.response.id')

# Créer un profil avec refus par défaut et lier les règles
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"roaming_profile\": {
      \"name\": \"Uniquement les Opérateurs Américains\",
      \"data_action_if_no_rules_match\": \"deny\",
      \"ims_action_if_no_rules_match\": \"deny\",
      \"roaming_rules\": [$RULE1, $RULE2]
    }
  }"
```

Autoriser les Données, Bloquer la Voix

```
# Créer une règle qui autorise les données mais bloque IMS
curl -k -X POST https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Données Seulement - T-Mobile",
      "mcc": "310",
      "mnc": "260",
      "data_action": "allow",
      "ims_action": "deny"
    }
  }'
```

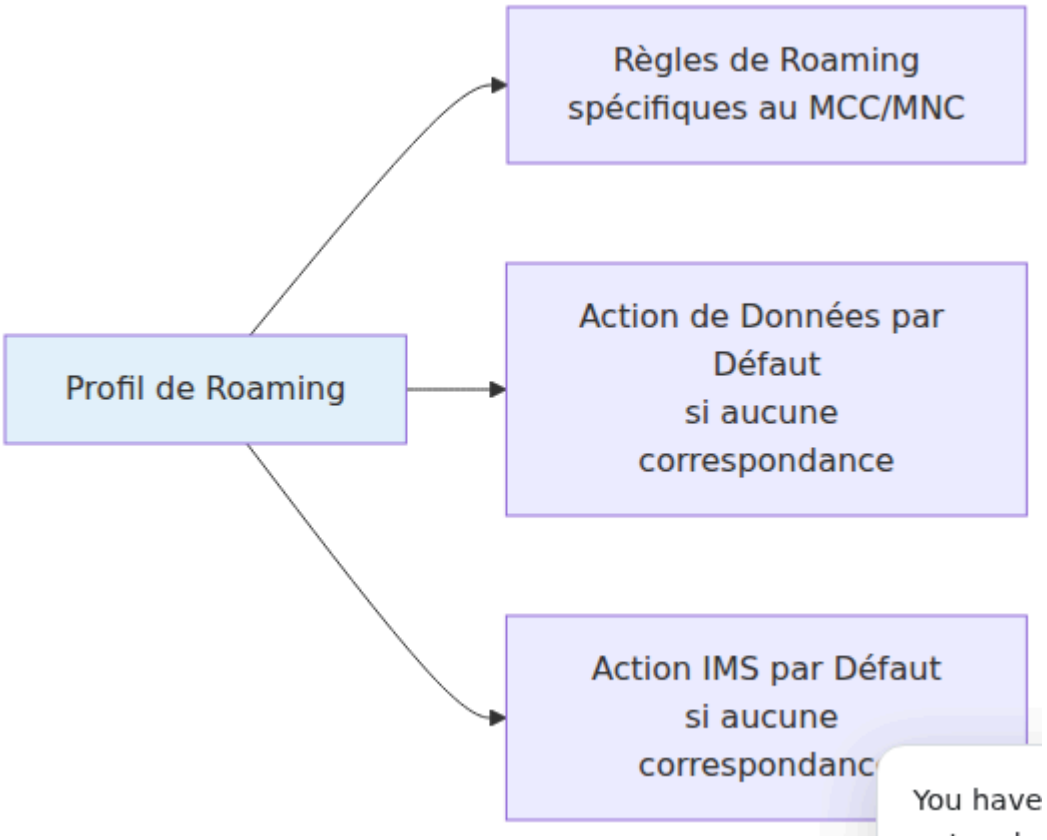
Bloquer des Réseaux Spécifiques (Liste Noire)

```
# Créer une règle de blocage de réseau coûteux
RULE=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Bloquer Réseau Coûteux",
      "mcc": "206",
      "mnc": "01",
      "data_action": "deny",
      "ims_action": "deny"
    }
  }' | jq -r '.response.id')

# Créer un profil avec autorisation par défaut
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"roaming_profile\": {
      \"name\": \"Bloquer les Réseaux Coûteux\",
      \"data_action_if_no_rules_match\": \"allow\",
      \"ims_action_if_no_rules_match\": \"allow\",
      \"roaming_rules\": [$RULE]
    }
  }"
```

Scénarios de Roaming Courants

Scénario 1 : Roaming Domestique Seulement

L'abonné peut roamer dans son pays d'origine mais pas à l'international.

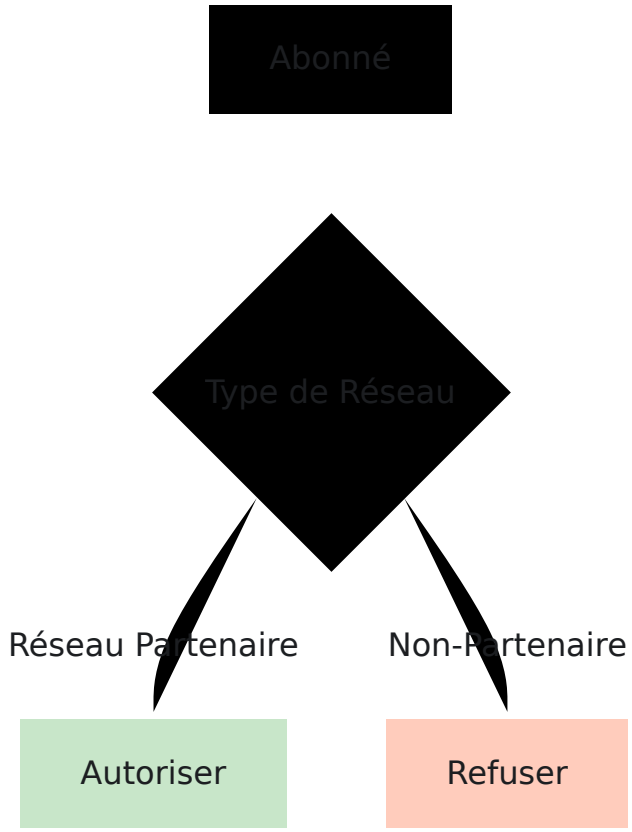


Configuration :

- Par défaut : Refuser tout
- Règles : Autoriser tous les codes MCC des USA (310, 311, 312, 313, 314, 315, 316)

Scénario 2 : Partenaires de Roaming Seulement

L'abonné ne peut roamer que sur des réseaux ayant des accords commerciaux.

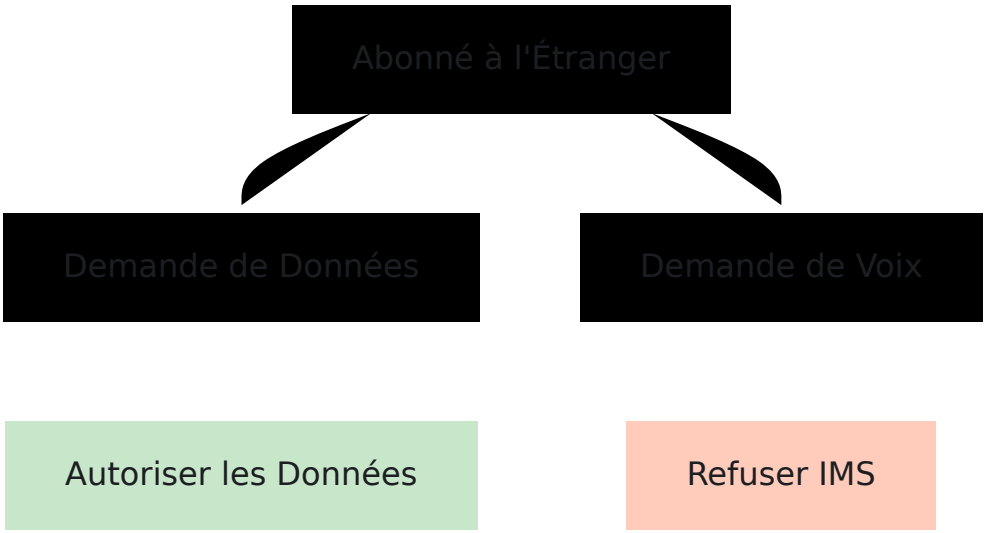


Configuration :

- Par défaut : Refuser tout
- Règles : Autoriser chaque réseau partenaire explicitement (par MCC/MNC)

Scénario 3 : Roaming de Données, Pas de Roaming Vocal

L'abonné peut utiliser des données à l'étranger mais doit utiliser le WiFi pour les appels vocaux.



Configuration :

- Règles : `data_action: "allow"`, `ims_action: "deny"`

Scénario 4 : Accès aux Services d'Urgence

Toujours autoriser les services d'urgence, même si le roaming est bloqué.

Remarque : La gestion des services d'urgence est généralement effectuée au niveau du MME/réseau. Les règles de roaming d'OmniHSS s'appliquent aux services normaux.

Référence MCC/MNC

Codes Pays Communs (MCC)

MCC	Pays	Réseaux
310-316	USA	AT&T, Verizon, T-Mobile, etc.
302	Canada	Rogers, Bell, Telus
234-235	Royaume-Uni	Vodafone, O2, EE
262	Allemagne	Deutsche Telekom, Vodafone
208	France	Orange, SFR, Bouygues
222	Italie	TIM, Vodafone, Wind
214	Espagne	Movistar, Vodafone

Opérateurs Américains Communs (MCC 310-316)

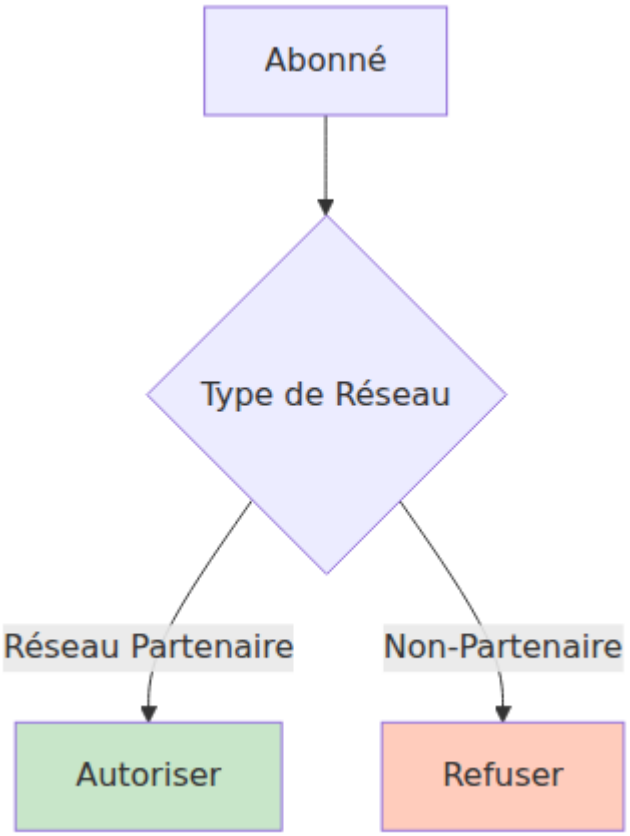
MCC	MNC	Opérateur
310	410	AT&T
311	480	Verizon
310	260	T-Mobile
310	120	Sprint
313	380	(Réseau de test exemple)

Listes Complètes : Voir ITU-T E.212 ou bases de données MCC/MNC

Points d'Application du Roaming

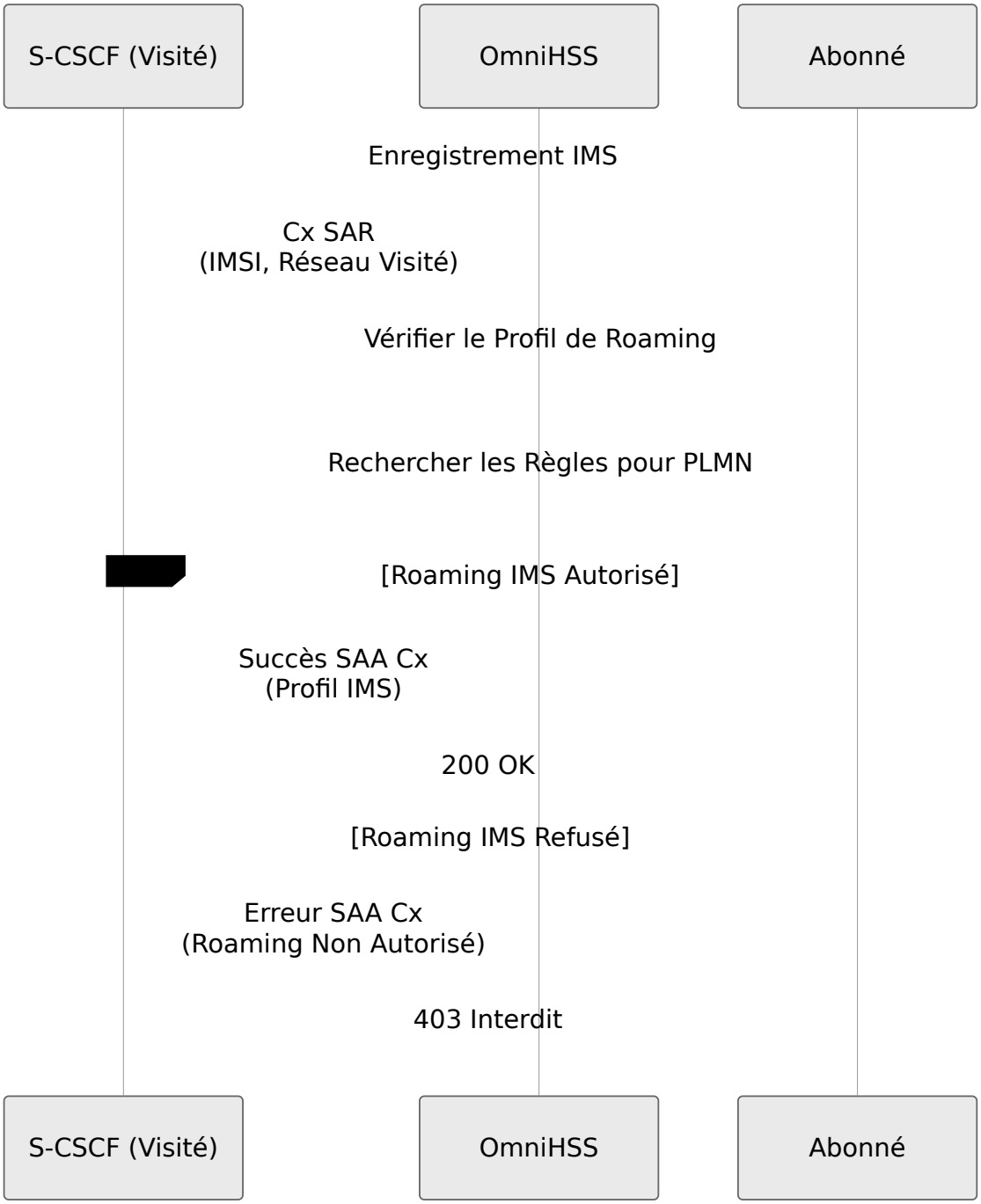
Interface S6a (Données)

Lorsque l'abonné se connecte au réseau visité :



Interface Cx (IMS)

Lorsque l'abonné s'enregistre à l'IMS dans le réseau visité :


S-CSCF (Visité)
OmniHSS
Abonné
Enregistrement IMS
Cx SAR
(IMSI, Réseau Visité)
Vérifier le Profil de Roaming
Rechercher les Règles pour PLMN
[Roaming IMS Autorisé]
Succès SAA Cx
(Profil IMS)
200 OK
[Roaming IMS Refusé]
Erreur SAA Cx
(Roaming Non Autorisé)
403 Interdit
S-CSCF (Visité)
OmniHSS
Abonné

Résolution des Problèmes de Roaming

L'Abonné Ne Peut Pas Se Connecter au Réseau Visité

Vérifiez l'attribution du profil de roaming :

- Interrogez la base de données pour voir le profil de roaming attribué à l'abonné
- Vérifiez le nom du profil et les paramètres d'action par défaut

Vérifiez si une règle existe pour le réseau visité :

- Interrogez la base de données pour les règles de roaming correspondant au MCC/MNC du réseau visité
- Vérifiez si une règle existe pour le profil de roaming de l'abonné
- Vérifiez la valeur de `data_action` pour ce réseau spécifique

L'Abonné Peut Se Connecter Mais Pas S'enregistrer à l'IMS

Vérifiez l'action IMS séparément :

- Interrogez les règles de roaming pour le réseau visité
- Vérifiez les valeurs de `data_action` et `ims_action`
- Recherchez les cas où les données sont autorisées mais IMS est refusé

Comportement de Roaming Inattendu

Examinez les journaux pour les vérifications de roaming :

```
[info] Vérification de roaming : IMSI 001001123456789, PLMN Visité
310-410
[info] Règle de roaming correspondante : "Autoriser AT&T"
[info] Action de données : autoriser, action IMS : autoriser
```

Meilleures Pratiques

Conception de Profil

1. **Commencer restrictif** - Refus par défaut, autoriser explicitement les partenaires
2. **Tester minutieusement** - Vérifier les règles en laboratoire avant la production
3. **Documenter les règles** - Maintenir une liste des réseaux autorisés et pourquoi
4. **Réviser régulièrement** - Mettre à jour au fur et à mesure que les accords de roaming changent

Gestion des Règles

1. **Utiliser des noms descriptifs** - "Autoriser-ATT-Données-Seulement" et non "Règle1"
2. **Vérifier MCC/MNC** - Vérifier les codes dans les bases de données officielles
3. **Considérer les deux services** - Penser aux données et à l'IMS séparément
4. **Surveiller l'utilisation** - Suivre quels réseaux les abonnés visitent réellement

Procédures Opérationnelles

1. **Changements d'Urgence** - Avoir une procédure pour activer/désactiver rapidement le roaming

2. **Mises à Jour en Masse** - Prévoir la mise à jour des profils de roaming de plusieurs abonnés
3. **Rapports** - Suivre l'utilisation du roaming et les tentatives refusées
4. **Communication avec les Clients** - Informer les clients des changements de politique de roaming

---

← Retour au Guide des Opérations | Suivant : Flux de Protocole →

Sh RepositoryData (Données Transparentes)

OmniHSS agit comme un **dépôt transparent** pour les données de service que les serveurs d'application (AS) stockent contre un abonné via l'interface Sh. Des documents tels que **MMTEL-Services** (services supplémentaires / renvoi d'appel, attente d'appel, OIP/OIR, interdiction) et **IMS-ODB-Information** (interdiction déterminée par l'opérateur) sont écrits par l'AS, conservés par le HSS et relus sur demande.

"Transparent" a une signification précise dans 3GPP TS 29.328 : le HSS comprend ces données **syntactiquement mais pas sémantiquement**. C'est un XML opaque appartenant à l'AS. OmniHSS stocke et renvoie la charge utile `ServiceData` **octet par octet**. Il ne reparse jamais ni ne re-sérialise la charge utile. Un document qu'un AS étiquette ou vérifie (par exemple XCAP/MMTel) est renvoyé exactement tel qu'écrit.

Les mêmes documents peuvent également être gérés hors bande par des systèmes de provisionnement/CRM via l'API de Provisionnement REST.

Table des Matières

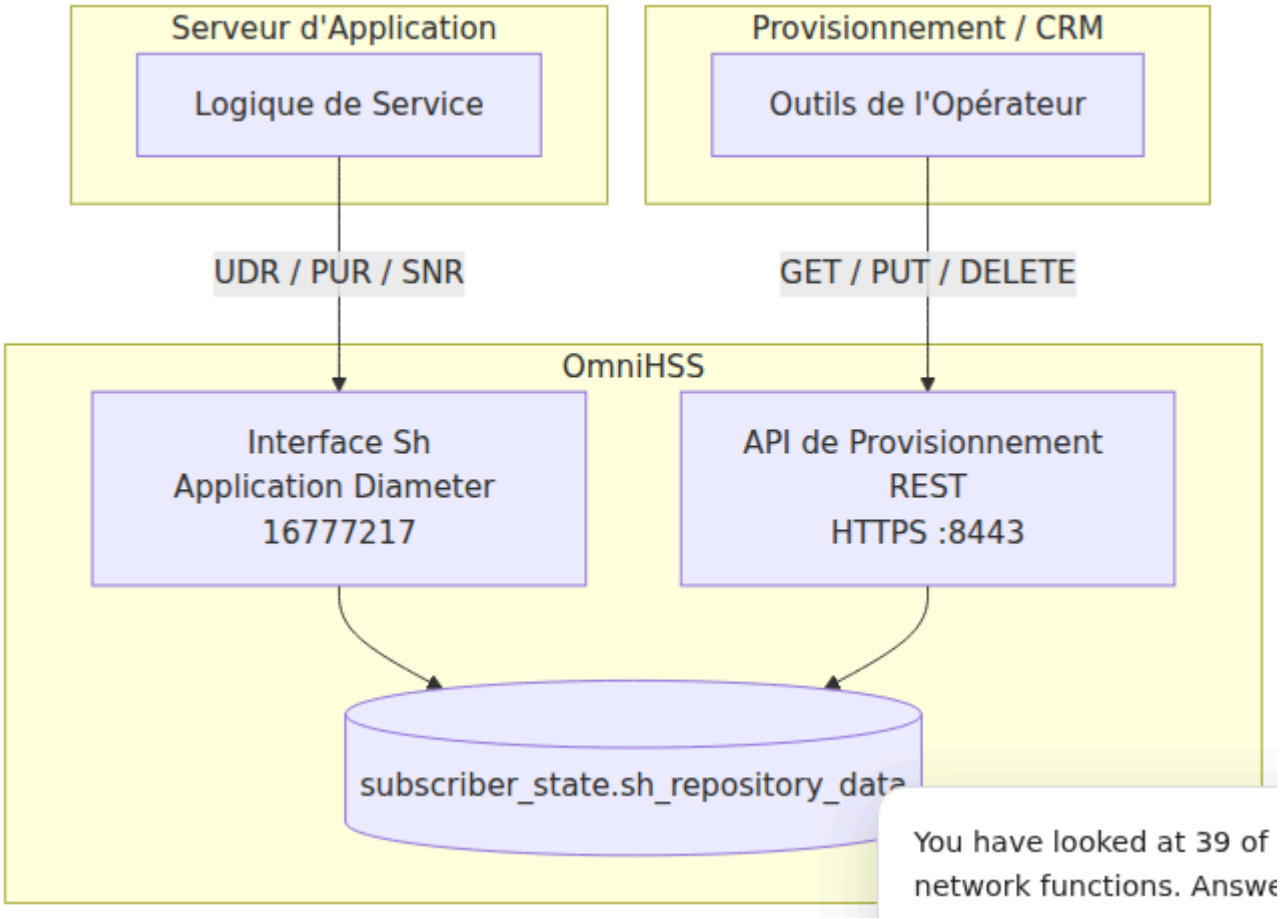
- Concepts
- Modèle de Données
- Flux de Messages Sh
- SequenceNumber et Concurrence
- API de Provisionnement REST
- Tables de Référence
- Dépannage

Concepts

Un abonné peut détenir un nombre quelconque de **documents transparents** indépendants, chacun identifié par une **Service-Indication** (le "nom de fichier"). Un document a trois parties :

Élément	Propriétaire	Signification
Service-Indication	Géré par le HSS	Nomme le document (par exemple `MMTEL-Services`, `IMS-ODB-Information`).
SequenceNumber	Géré par le HSS	Compteur de version utilisé pour la concurrence optimiste. Voir SequenceNumber et Concurrence.
ServiceData	Propriété de l'AS (opaque)	La charge utile XML elle-même. Stockée et renvoyée telle quelle.

Le HSS n'interprète que Service-Indication et SequenceNumber. Le corps `ServiceData` est traité comme des octets opaques.



Modèle de Données

Tous les documents transparents d'un abonné sont conservés dans un seul champ, `subscriber_state.sh_repository_data`, sous forme d'objet JSON indexé par Service-Indication. Chaque entrée stocke la version et la charge utile telle quelle :

```
{
  "MMTEL-Services":      { "sequence_number": "0", "service_data":
"<MMTelServices …>…</MMTelServices>" },
  "IMS-ODB-Information": { "sequence_number": "1", "service_data":
"<OdbForImsOrientedServices xmlns=\"odb.mmtel.omnitou.ch\"/>" }
}
```

Champ	Description
`sequence_number`	La version actuelle de ce document, telle que dernièrement acceptée par le HSS. Renvoie dans chaque `UDA` et utilisée pour valider les mises à jour.
`service_data`	Le corps XML opaque, stocké exactement comme l'AS l'a envoyé (pas de normalisation de l'espace de noms, de l'ordre des attributs ou des espaces). Un document sans élément `ServiceData` (les *données de dépôt vides* de la spécification) omet ce champ.

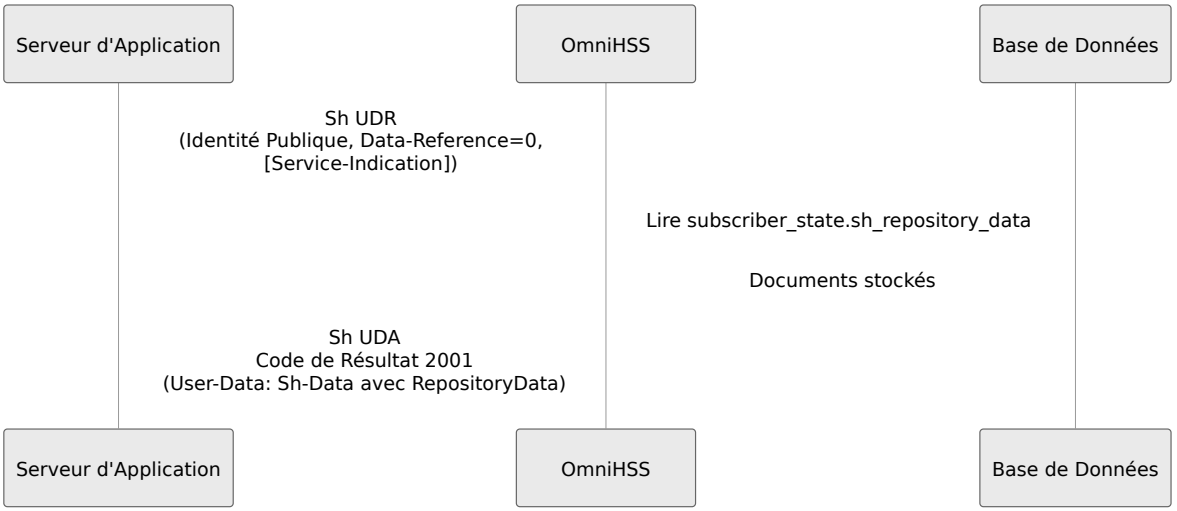
Ce modèle signifie qu'un seul abonné peut détenir de nombreux documents distincts, chacun versionné indépendamment, sans schéma fixe imposé par le HSS.

Flux de Messages Sh

RepositoryData est transporté dans l'AVP `User-Data` en tant que document XML `Sh-Data`. RepositoryData est sélectionné avec **valeur Data-Reference 0**. Un AVP **Service-Indication** optionnel restreint une demande à un seul document.

Demande de Données Utilisateur (UDR/UDA) : Lecture

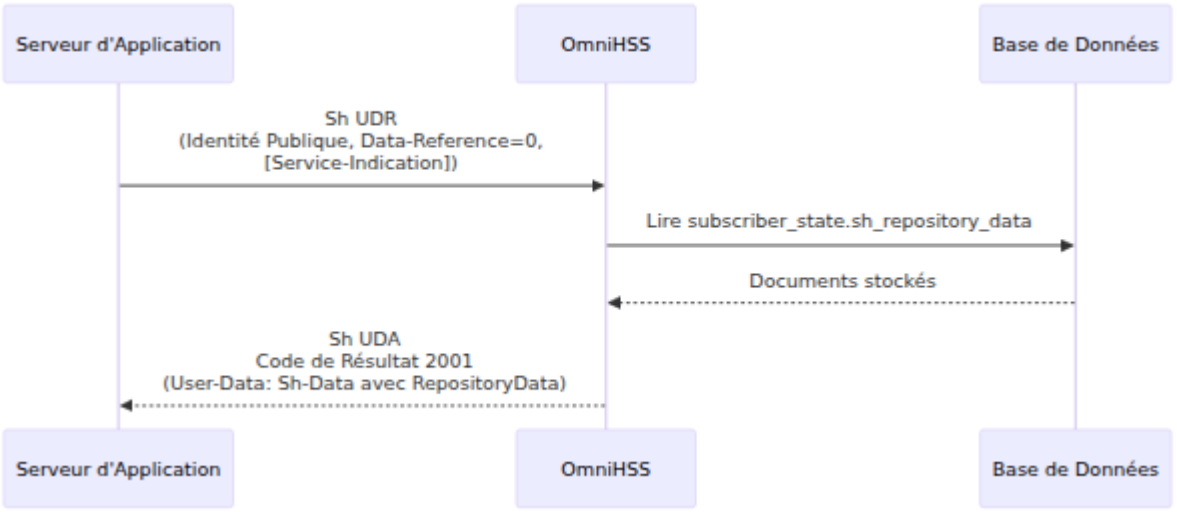
L'AS lit un ou tous les documents transparents. Avec une Service-Indication, seul le document correspondant est renvoyé ; sans, chaque document stocké est renvoyé, chacun comme son propre élément `<RepositoryData>`.



Un abonné inconnu ou non-IMS reçoit **Experimental-Result 5001 (DIAMETER_ERROR_USER_UNKNOWN)**.

Demande de Mise à Jour de Profil (PUR/PUA) : Écriture

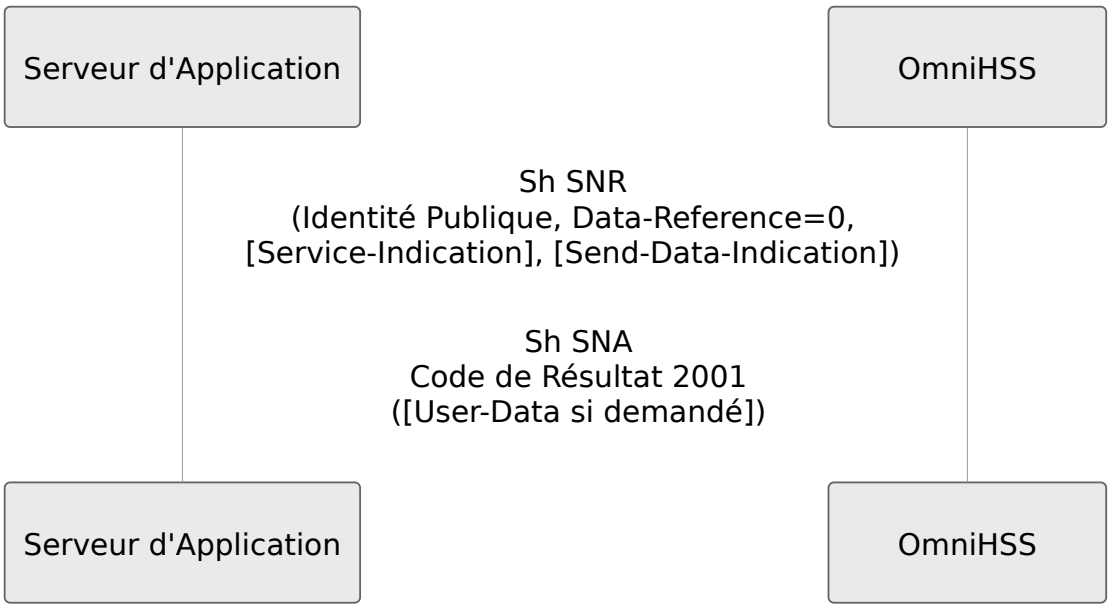
L'AS crée ou met à jour un seul document. La charge utile `ServiceData` est stockée telle quelle. La mise à jour est acceptée uniquement si son SequenceNumber est valide (voir ci-dessous).



Lorsqu'un PUR se résout à plusieurs abonnés (une Identité Publique avec joker), la mise à jour est appliquée de manière atomique : si un document est obsolète, aucun n'est persistant.

Demande d'Notifications d'Abonnement (SNR/SNA) : Abonnement

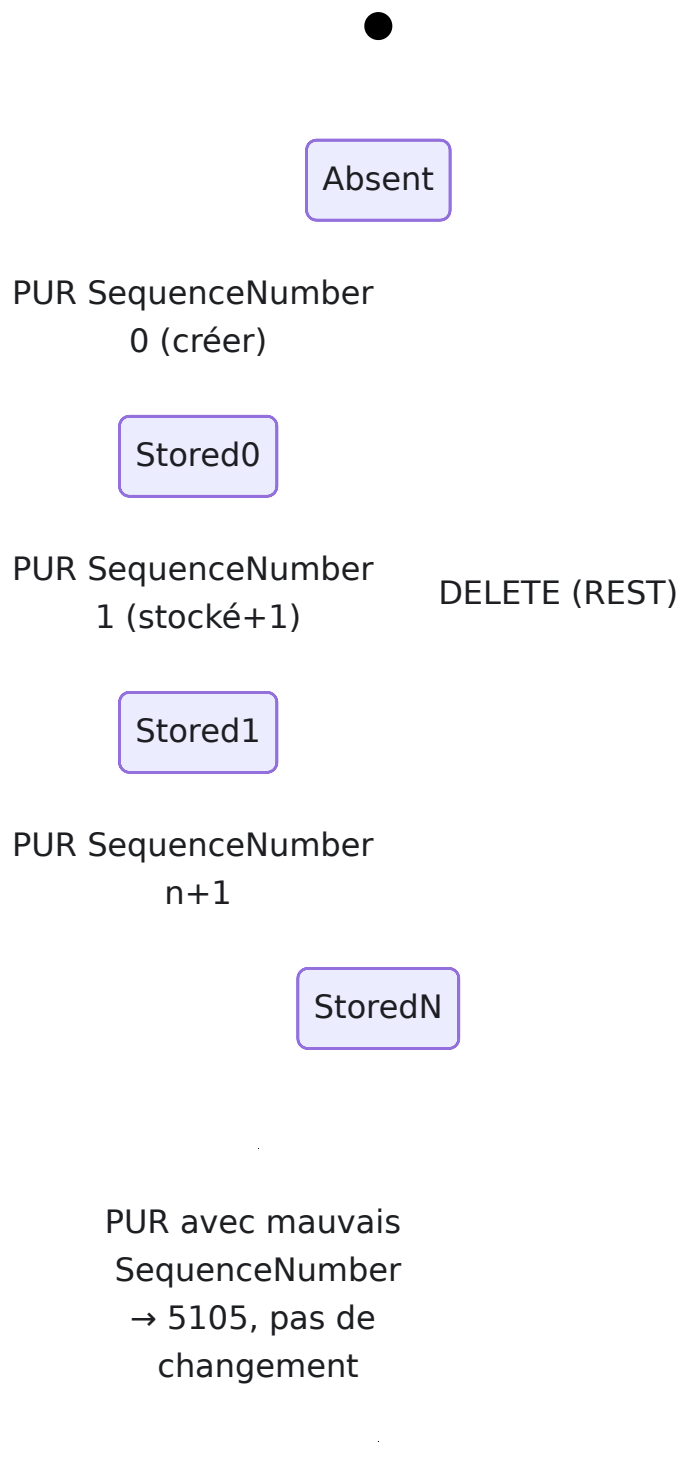
L'AS s'abonne aux données d'un abonné et peut demander que la valeur actuelle soit renvoyée immédiatement (`Send-Data-Indication`). OmniHSS répond au SNR et, lorsque demandé, inclut le RepositoryData actuel.



SequenceNumber et Concurrence

Le SequenceNumber fournit un contrôle de concurrence optimiste afin que deux serveurs d'application ne puissent pas se remplacer silencieusement, conformément à 3GPP TS 29.328.

- **Créer** un nouveau document utilise **SequenceNumber 0** (la valeur `0` est réservée exclusivement pour les données nouvellement ajoutées).
- **Mettre à jour** un document existant doit porter le **successeur immédiat** du SequenceNumber stocké (stocké + 1).
- Le compteur **se réinitialise de 65535 à 1** (pas 0, puisque 0 est réservé).
- Une mise à jour dont le SequenceNumber n'est pas le successeur attendu est rejetée avec **Experimental-Result 5105 (DIAMETER_ERROR_TRANSPARENT_DATA_OUT_OF_SYNC)** et le document stocké reste inchangé.



Exemple : un document est stocké à SequenceNumber `0`. Un PUR portant SequenceNumber `5` est rejeté avec `5105` et rien ne change. Un PUR portant SequenceNumber `1` est accepté et le document passe à la version `1`.

API de Provisionnement REST

Les mêmes documents peuvent être lus et écrits directement par des systèmes de provisionnement/CRM. L'abonné est adressé par IMSI dans le chemin ; le

document est identifié par `service_indication` (dans le corps de la demande pour les mises à jour, ou la chaîne de requête pour les lectures/suppressions). Le `service_data` opaque est échangé octet par octet, exactement comme via Sh.

Toutes les réponses utilisent l'enveloppe API standard :

```
{ "status": "success", "response": { } }
```

Contrairement au chemin Sh, l'API REST est un accès de provisionnement autoritaire et ne **s'applique pas** à la vérification de SequenceNumber hors synchronisation. Une écriture définit directement le document.

Lister Tous les Documents

Renvoie chaque document stocké pour l'abonné.

Point de terminaison : `GET /api/subscriber/repository_data/:imsi`

```
curl -k
https://hss.example.com:8443/api/subscriber/repository_data/001001123
```

```
{
  "status": "success",
  "response": {
    "MMTEL-Services":      { "sequence_number": "0",
"service_data": "<MMTelServices …/>" },
    "IMS-ODB-Information": { "sequence_number": "1",
"service_data": "<OdbForImsOrientedServices
xmlns=\"odb.mmtel.omnitou.ch\"/>" }
  }
}
```

Obtenir un Document Unique

Renvoie un document par Service-Indication.

Point de terminaison : `GET /api/subscriber/repository_data/:imsi?service_indication=:service_indication`

```
curl -k
"https://hss.example.com:8443/api/subscriber/repository_data/00100112
service_indication=MMTEL-Services"
```


```
{
  "status": "success",
  "response": { "sequence_number": "0", "service_data": "
<MMTelServices …/>" }
}
```


Créer ou Mettre à Jour un Document

Crée le document s'il est absent, ou le remplace.

Point de terminaison : `PUT /api/subscriber/repository_data/:imsi`

```
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/repository_data/001001123
\
  -H "Content-Type: application/json" \
  -d '{
        "service_indication": "MMTEL-Services",
        "service_data": "<MMTelServices version=\"1\"/>",
        "sequence_number": "0"
      }'
```

Champ	Type	Requis	Par Défaut	Description
`service_indication`	Chaîne	Oui	-	Le nom du document à créer/mettre à jour (par exemple `MMTEL-Services`).
`service_data`	Chaîne	Non	-	La charge utile XML opaque, stockée telle quelle. Omettre pour stocker un document vide (Service-Indication et SequenceNumber uniquement).
`sequence_number`	Chaîne	Non	`0`	La version à enregistrer pour le document.

La réponse renvoie le document stocké :

```
{
  "status": "success",
  "response": { "sequence_number": "0", "service_data": "
<MMTelServices version=\"1\"/>" }
}
```

Supprimer un Document

Supprime un document unique par Service-Indication.

Point de terminaison : `DELETE /api/subscriber/repository_data/:imsi`

```
curl -k -X DELETE
https://hss.example.com:8443/api/subscriber/repository_data/001001123
\
  -H "Content-Type: application/json" \
  -d '{ "service_indication": "MMTEL-Services" }'
```

Champ	Type	Requis	Par Défaut	Description
`service_indication`	Chaîne	Oui	-	Le document à supprimer.

La réponse renvoie les documents restants pour l'abonné.

Tables de Référence

Valeurs de Data-Reference (RepositoryData)

Valeur	Nom	Description	Référence
0	RepositoryData	Données de service transparentes identifiées par Service-Indication	3GPP TS 29.328 §7.6

Codes de Résultat et Experimental-Result

Code	Nom	Significa
2001	DIAMETER_SUCCESS	UDR/PUR/SI traités avec succès
5001	DIAMETER_ERROR_USER_UNKNOWN	Abonné nor trouvé ou n activé IMS
5105	DIAMETER_ERROR_TRANSPARENT_DATA_OUT_OF_SYNC	Le SequenceN de mise à jo n'est pas le successeur valeur stock

ID d'Application

ID	Interface	Description	Référence
16777217	Sh	AS ↔ HSS données utilisateur	3GPP TS 29.329

Indications de Service Courantes

Service-Indication	Contenu Typique
`MMTEL-Services`	Services supplémentaires MMTel (renvoi d'appel, attente/maintien/interdiction de communication, OIP/OIR, conférence) : XML simservs
`IMS-ODB-Information`	Informations d'interdiction déterminées par l'opérateur pour les services orientés IMS

> Les valeurs Service-Indication sont définies par le Serveur d'Application et son schéma XML correspondant ; le HSS n'impose pas de liste fixe.

Dépannage

Le Serveur d'Application reçoit 5105 lors de la mise à jour

Symptômes : Un PUR est répondu avec Experimental-Result `5105` et le document reste inchangé.

Causes possibles :

- L'AS a envoyé un SequenceNumber qui n'est pas le successeur immédiat de la valeur stockée au HSS (sa copie mise en cache est obsolète).
- Une mise à jour concurrente d'un autre AS a avancé le SequenceNumber.

Résolution :

1. Lire le document actuel avec un UDR pour obtenir le SequenceNumber à jour.
2. Réémettre le PUR avec SequenceNumber = stocké + 1.
3. Pour forcer une valeur administrativement, définissez-la via l'API REST, qui n'applique pas la vérification de concurrence.

Le ServiceData renvoyé diffère de ce qui a été écrit

Symptômes : L'AS croit que la charge utile a été modifiée par le HSS.

Causes possibles :

- Cela ne devrait pas se produire. OmniHSS stocke et renvoie `ServiceData` octet par octet. Les différences apparentes proviennent généralement de la propre bibliothèque XML de l'AS qui re-sérialise le document (modifiant les préfixes d'espace de noms, l'ordre des attributs ou les espaces) avant la comparaison.

Résolution :

1. Comparer les octets bruts renvoyés dans la réponse `UDA`/REST avec les octets bruts envoyés, pas une représentation re-parsée.
2. Confirmer que l'AS ne normalise pas le XML de son côté avant la comparaison.

Document non trouvé via REST

Symptômes : `GET …/repository_data/:imsi?service_indication=…` renvoie une erreur.

Causes possibles :

- Aucun document avec cette Service-Indication n'est stocké pour l'abonné.
- L'IMSI ne correspond pas à un abonné existant.

Résolution :

1. Lister tous les documents avec `GET /api/subscriber/repository_data/:imsi` pour voir ce qui est stocké.
2. Vérifier l'IMSI avec Obtenir l'Abonné par IMSI.

SRVCC - Continuité de l'Appel Vocal sur un Radio Unique

← Retour au Guide des Opérations

Aperçu

SRVCC (Continuité de l'Appel Vocal sur un Radio Unique) permet à un UE de transférer un appel VoLTE actif vers un porteur CS (Commuté) 2G/3G lorsque la couverture LTE échoue. Le MME orchestre le transfert via l'**interface Sv** (TS 29.280). Pour construire la demande Sv PS-to-CS, le MME a besoin de deux valeurs provenant du HSS :

AVP	Code	Objectif
`STN-SR`	1433	Numéro de Transfert de Session pour SRVCC. Le MSC-Server compose ce numéro E.164 pour atteindre le TAS/S-CSCF, qui exécute un SIP REFER pour déplacer la session VoLTE active vers le porteur CS.
`C-MSISDN`	1452	MSISDN de Corrélation. Le TAS l'utilise pour faire correspondre l'appel entrant STN-SR à la bonne session active lorsque l'abonné a plusieurs appels en cours.

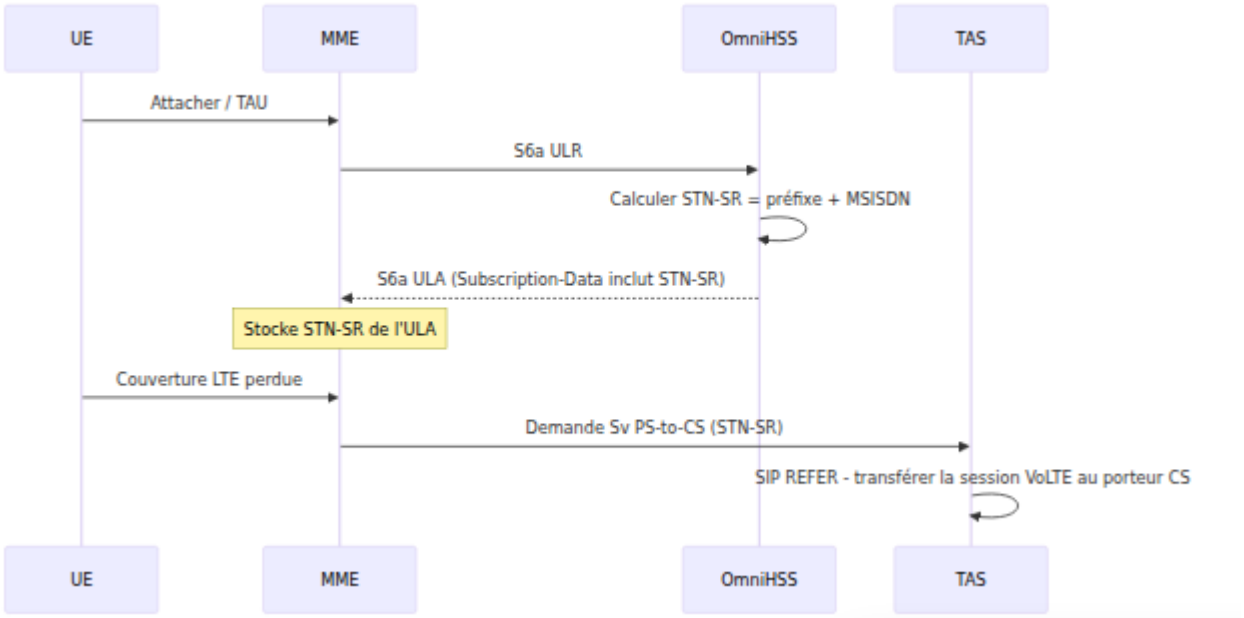
OmniHSS fournit `STN-SR` dans la réponse S6a **Update Location Answer** (ULA) AVP `Subscription-Data`. Le MME le lit depuis l'ULA et le transmet dans la demande Sv PS-to-CS.

Spécifications de référence :

Spécification	Titre
TS 23.216	Procédure SRVCC PS-to-CS
TS 29.272	Interface S6a (STN-SR §7.3.89, Subscription-Data §7.3.2)
TS 23.003	Format du numéro STN-SR (§13.4)

Comment ça fonctionne

OmniHSS utilise une approche basée sur un préfixe. Vous configurez un préfixe STN-SR unique pour l'ensemble de l'opérateur. Le HSS calcule le numéro STN-SR complet au moment de l'ULA en concaténant le préfixe avec le MSISDN de l'abonné. Le résultat est encodé en TBCD et placé dans l'AVP `STN-SR` à l'intérieur de `Subscription-Data`.



Cette approche ne nécessite aucun provisionnement par abonné. Tous les abonnés sur le HSS reçoivent le même préfixe, donc un seul changement de configuration active SRVCC pour l'ensemble de la base d'abonnés.

Si le préfixe n'est pas configuré, OmniHSS omet complètement l'AVP `STN-SR`. L'ULA n'est autrement pas affectée. Les déploiements non-SRVCC ne voient aucun changement de comportement.

Configuration

Ajoutez la clé `srvcc` à la configuration de l'application `:hss` dans `config/runtime.exs`.

```
config :hss, :srvcc,
  stn_sr_prefix: "9"
```

Paramètres

Paramètre	Type	Requis	Par défaut	Description
`stn_sr_prefix`	String	Non	(aucun)	Préfixe STN-SR attribué par l'opérateur, préfixé au MSISDN de chaque abonné pour former le numéro STN-SR E.164 complet. Lorsqu'il n'est pas défini, OmniHSS omet l'AVP `STN-SR` de toutes les réponses ULA.

Format du Numéro STN-SR

Le numéro STN-SR complet est :

```
STN-SR = stn_sr_prefix + MSISDN de l'abonné
```

Le numéro est encodé en **TBCD** (Téléphonie Binaire Codée Décimale, selon TS 24.008 §10.5.4.7) et placé dans l'AVP `STN-SR` (code 1433, fournisseur 10415).

Exemple :

Entrée	Valeur
Préfixe configuré	`9`
MSISDN de l'abonné	`61412345678`
STN-SR calculé	`961412345678`

Le MSC-Server compose `961412345678`. L'appel arrive au TAS, qui utilise la partie MSISDN pour localiser la session IMS active et émet le SIP REFER pour la transférer au porteur CS.

Exemple de Configuration

SRVCC Activé

```
# config/runtime.exs

config :hss, :srvcc,
  # Préfixe composé par le MSC-Server pour atteindre le TAS via
PSTN/CS.
  # STN-SR = stn_sr_prefix + MSISDN de l'abonné.
  stn_sr_prefix: "9"
```

Comment ça fonctionne : Chaque réponse ULA inclut `STN-SR` dans `Subscription-Data`. La valeur est l'encodage TBCD du préfixe concaténé avec le MSISDN de l'abonné. Le MME stocke la valeur et la transmet dans les demandes Sv PS-to-CS lorsque SRVCC est déclenché.

SRVCC Désactivé (par défaut)

Lorsque la clé `srvcc` est absente ou que `stn_sr_prefix` n'est pas défini, OmniHSS omet l'AVP `STN-SR`. L'ULA est autrement identique. Utilisez ceci pour les déploiements qui ne nécessitent pas SRVCC.

Interaction avec le MME

Le MME (OmniMME) lit `STN-SR` depuis l'ULA `Subscription-Data` et le stocke dans le contexte de l'abonné. Lorsque SRVCC est déclenché, il inclut `STN-SR` dans la demande Sv **PS-to-CS** envoyée au MSC-Server. Voir Omnitouch/omnimme#426 pour l'implémentation du MME.



Dépannage

STN-SR Non Présent dans l'ULA

Symptômes : Le MME ne peut pas remplir la demande Sv PS-to-CS. Les transferts SRVCC échouent.

Causes possibles :

- `stn_sr_prefix` n'est pas défini dans `config/runtime.exs`.
- Le service HSS a démarré avant que le changement de configuration ne prenne effet.

Résolution :

1. Vérifiez que le bloc `srvcc` est présent dans `config/runtime.exs` avec un `stn_sr_prefix` non vide.
2. Redémarrez le service OmniHSS pour appliquer le changement de configuration.
3. Déclenchez un nouvel ULR depuis le MME (par exemple en forçant un TAU). Confirmez que les `Subscription-Data` de l'ULA contiennent maintenant `STN-SR`.

Valeur STN-SR Incorrecte

Symptômes : Le MSC-Server reçoit un numéro malformé ou non routable.

Causes possibles :

- Le préfixe est incorrect ou manque de chiffres de tête.
- Le MSISDN de l'abonné stocké dans le HSS n'est pas au format international E.164 (chiffres uniquement, pas de `+`).

Résolution :

1. Vérifiez le MSISDN de l'abonné via l'API (`GET /api/subscriber/imsi/:imsi`).
2. Vérifiez le STN-SR attendu : concaténez `stn_sr_prefix` avec le MSISDN stocké.
3. Confirmez que la table de routage du MSC-Server gère cette plage de numéros.

← Retour au Guide des Opérations

Interface Sy (Contrôle de Limite de Dépense)

Aperçu

L'interface Sy connecte le HSS (agissant comme le **PCRF**, Fonction de Règles de Politique et de Facturation) à un **OCS** externe (Système de Facturation en Ligne) pour le **contrôle de limite de dépense des abonnés**. Elle permet au PCRF de baser ses décisions politiques - par exemple, quelle règle de facturation appliquer, ou s'il faut limiter un abonné - sur l'état des **compteurs de politique** que l'OCS maintient pour chaque abonné (quota de données consommé, dépenses par rapport à un plafond mensuel, etc.).

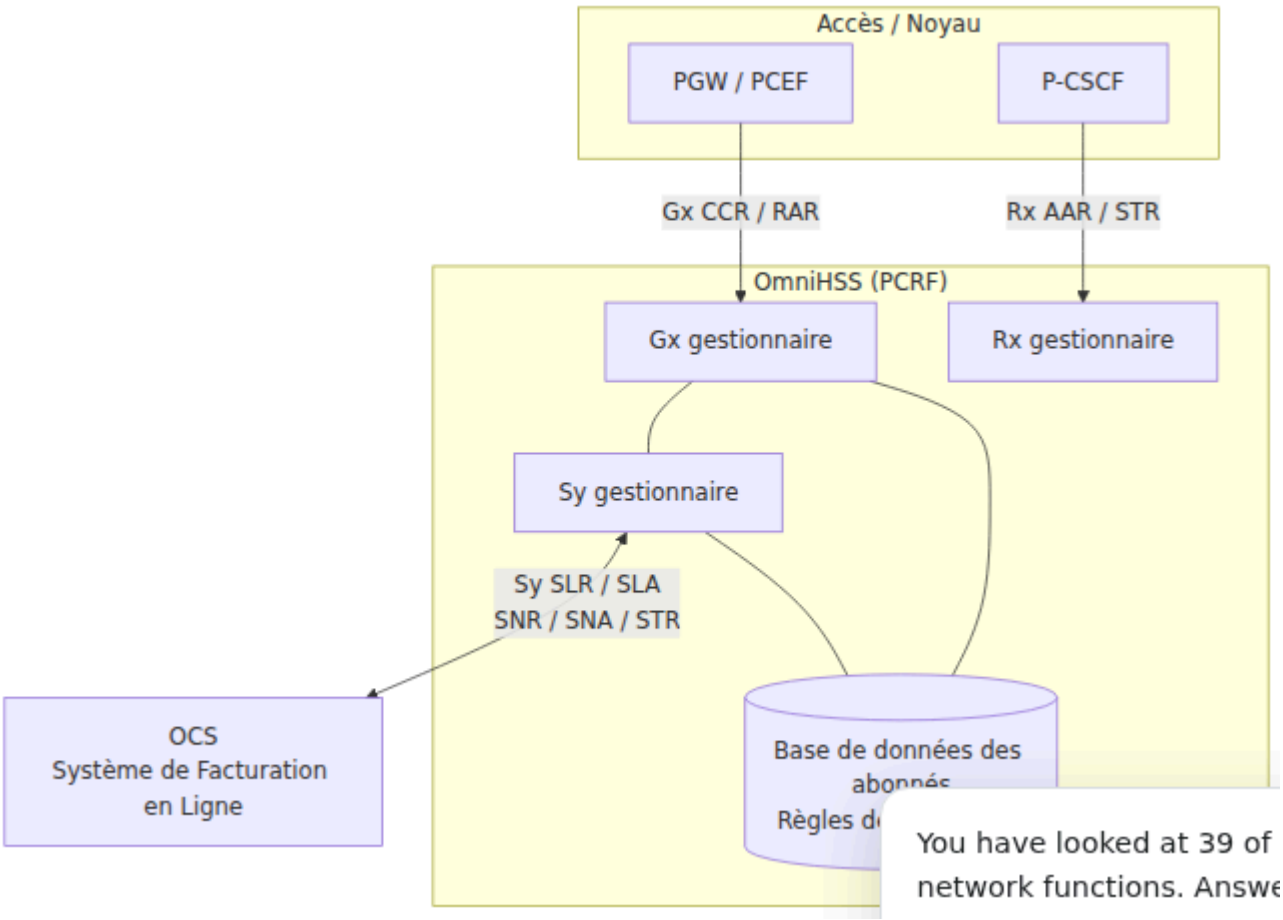
Le PCRF OmniHSS installe des règles de facturation et QoS via Gx et autorise les médias IMS via Rx. Via Sy, il s'abonne à l'état des compteurs de politique d'un abonné auprès de l'OCS et est notifié lorsqu'ils changent, afin que ces décisions Gx puissent tenir compte des dépenses et du quota actuels de l'abonné.

Sy est défini par 3GPP TS 29.219 et utilise l'ID d'application Diameter **16777302**.

Propriété	Valeur
ID d'application	16777302
Spécification	3GPP TS 29.219
Rôle HSS	PCRF (client Diameter pour SLR/STR, serveur pour SNR)
Pair	OCS (Système de Facturation en Ligne)
Transport	TCP ou SCTP, port 3868

> **Sy vs Gy.** Sy n'est pas une facturation en ligne. Gy (TS 32.299) est l'endroit où le PGW demande à l'OCS d'accorder un crédit pour le trafic en direct. Sy est une interface distincte et légère où le **PCRF** s'abonne à l'**état** des compteurs de politique de l'OCS et est notifié lorsqu'ils changent, afin qu'il puisse ajuster la politique. L'OCS est le serveur dans les deux cas.

Architecture



Le HSS est le **client Sy** : il ouvre une session Sy à l'OCS lorsqu'il a d'abord besoin de l'état de dépense d'un abonné, et la ferme lorsque la dernière session de données de l'abonné se termine. L'OCS est le **serveur Sy** : il répond aux demandes d'état et, chaque fois que l'état d'un compteur change, il notifie le PCRF afin qu'il puisse réappliquer la politique via Gx.

Compteurs de Politique et Règles de Facturation

L'OCS maintient un ou plusieurs **compteurs de politique** par abonné. Chaque compteur a :

- un **Identifiant de Compteur de Politique** (AVP 2901) - un nom stable pour le compteur, et
- un **État de Compteur de Politique** (AVP 2902) - une étiquette décrivant où se trouve actuellement le compteur (par exemple `actif`, `au-dessus_du_quota`, `limité`). L'ensemble des valeurs d'état n'est pas défini par le 3GPP ; il est configuré conjointement par l'opérateur sur l'OCS et le PCRF (TS 29.219 §3.1).

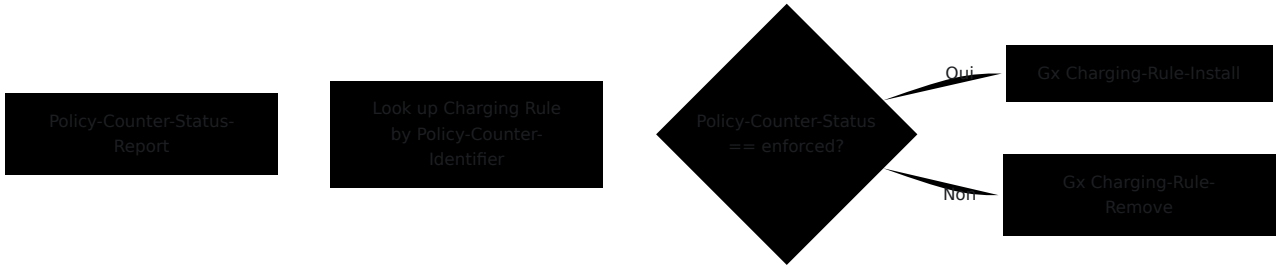
OmniHSS lie ceux-ci aux règles de facturation déjà présentes dans sa base de données :

AVP Sy	Nature	Correspond à	Rôle
`Policy-Counter-Identifier` (2901)	Statique - nomme le compteur	Un **nom de règle de facturation** dans la base de données HSS	Sélectionne *quelle* règle de facturation le compteur régit
`Policy-Counter-Status` (2902)	Dynamique - change aux seuils	Une **action d'application**	Décide *quoi faire* avec cette règle maintenant : installer, supprimer ou sélectionner un niveau

L'identifiant est stable pendant la durée de vie du compteur, donc à lui seul, il ne peut pas exprimer "sous quota" par rapport à "au-dessus du quota". Le **statut** est la valeur qui change lorsque l'abonné franchit un seuil, donc il déclenche l'action :

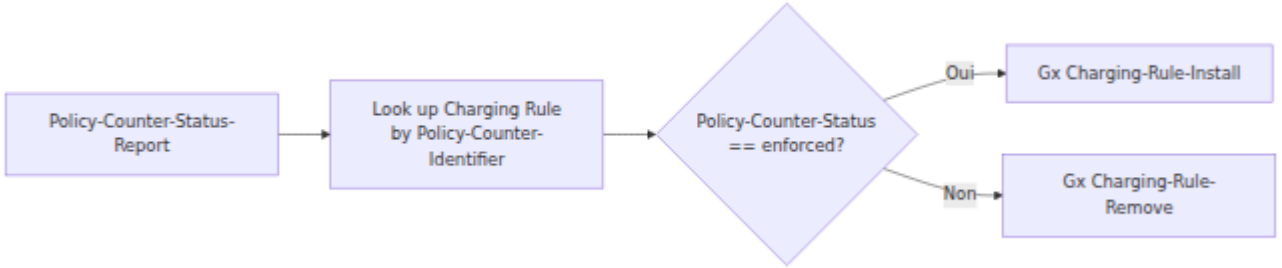
- le statut fixe **`enforced`** installe la règle de facturation nommée (`Charging-Rule-Install` via Gx) ;
- tout autre statut la supprime (`Charging-Rule-Remove`).

> **Sy ne transporte aucune règle de facturation propre.** Sy n'a pas d'AVPs de Règle de Facturation, de QoS ou de Flux - ceux-ci n'existent que sur Gx. L'OCS ne peut donc pas transmettre au PCRF une définition de règle ; il peut seulement *nommer* la règle (via l'identifiant du compteur) et signaler *quand* l'appliquer (via le statut). La définition de la règle elle-même est provisionnée dans le HSS. C'est pourquoi `Policy-Counter-Identifier` doit correspondre à un nom de règle de facturation, et l'OCS doit émettre ces noms exacts avec le statut `enforced` lorsque la règle doit être appliquée.



Cycle de Vie de la Session Sy

Une session Sy est **par abonné**, pas par porteur de données. Le PCRF garde une session Sy (identifiée par un unique `Session-Id` Diameter) pour un abonné tant que l'une des sessions de données de cet abonné dépend de l'état de dépense (TS 29.219 §4.4.1).

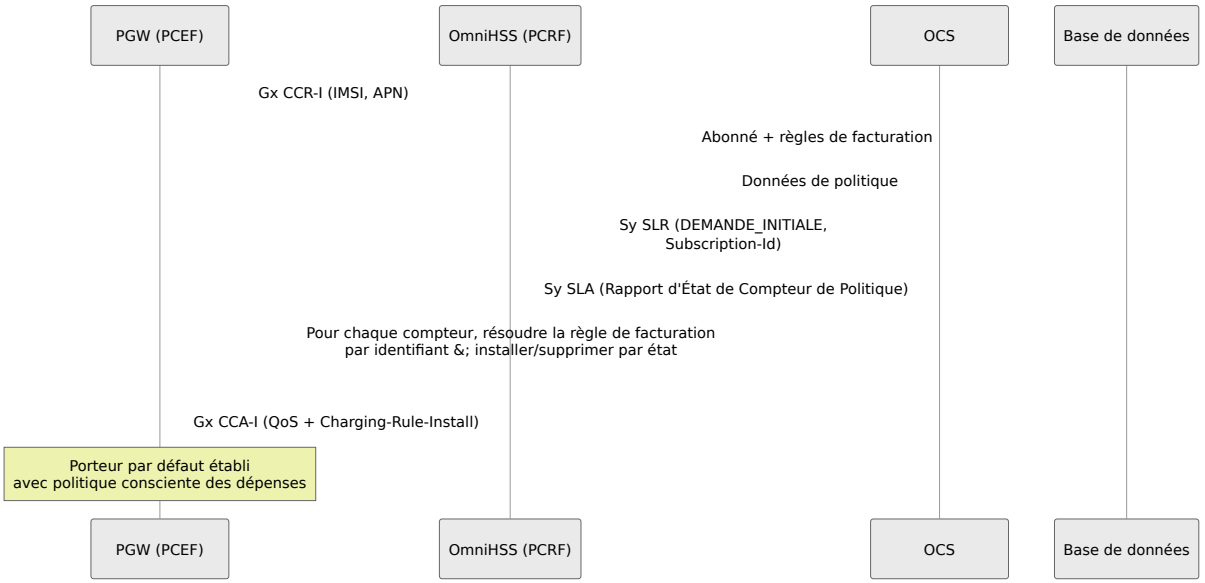


- **Établir (SLR, DEMANDE_INITIALE).** Lors du premier Gx CCR-I où la politique de l'abonné dépend de l'état de dépense, le PCRF ouvre la session Sy et demande les compteurs de politique pertinents. L'OCS répond avec l'état actuel de chacun dans la SLA.

- **Mettre à jour (SLR, DEMANDE_INTERMEDIAIRE).** Si le PCRF a besoin ultérieurement de compteurs supplémentaires, il réutilise la même session Sy avec une demande intermédiaire.
- **Notifier (SNR / SNA).** Lorsque l'état d'un compteur change, l'OCS envoie une Demande de Notification de Statut de Dépense. Le PCRF accuse réception avec un SNA et applique le changement via Gx.
- **Terminer (STR / STA).** Lorsque la dernière session de données de l'abonné se termine, le PCRF ferme la session Sy avec une Demande de Terminaison de Session.

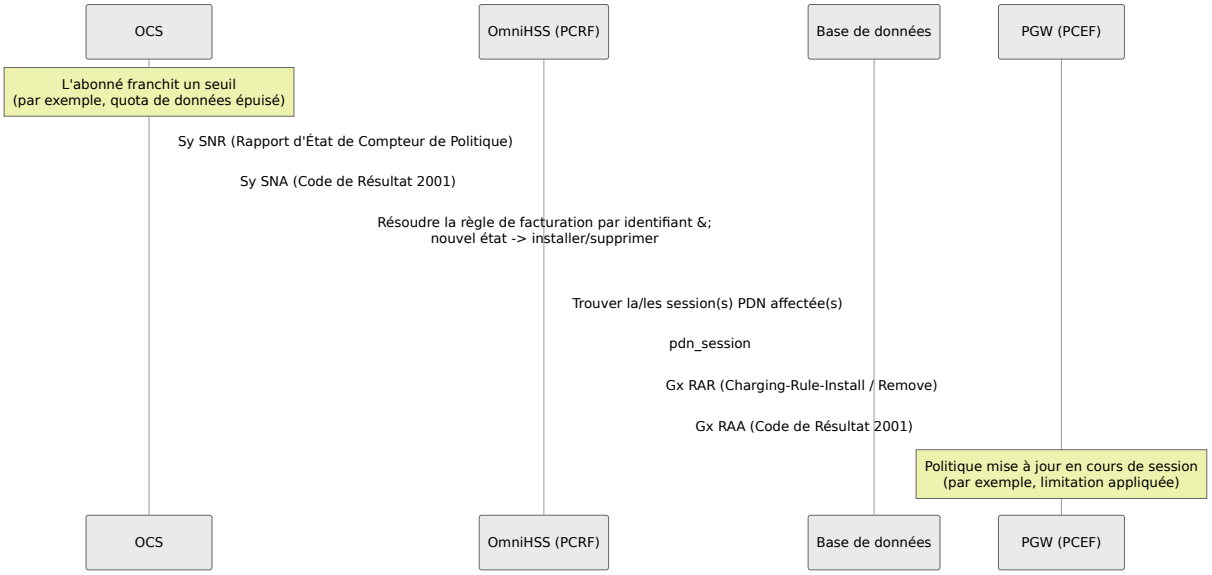
Flux de Messages

Flux 1 : Rapport Initial de Limite de Dépense lors de la Configuration PDN



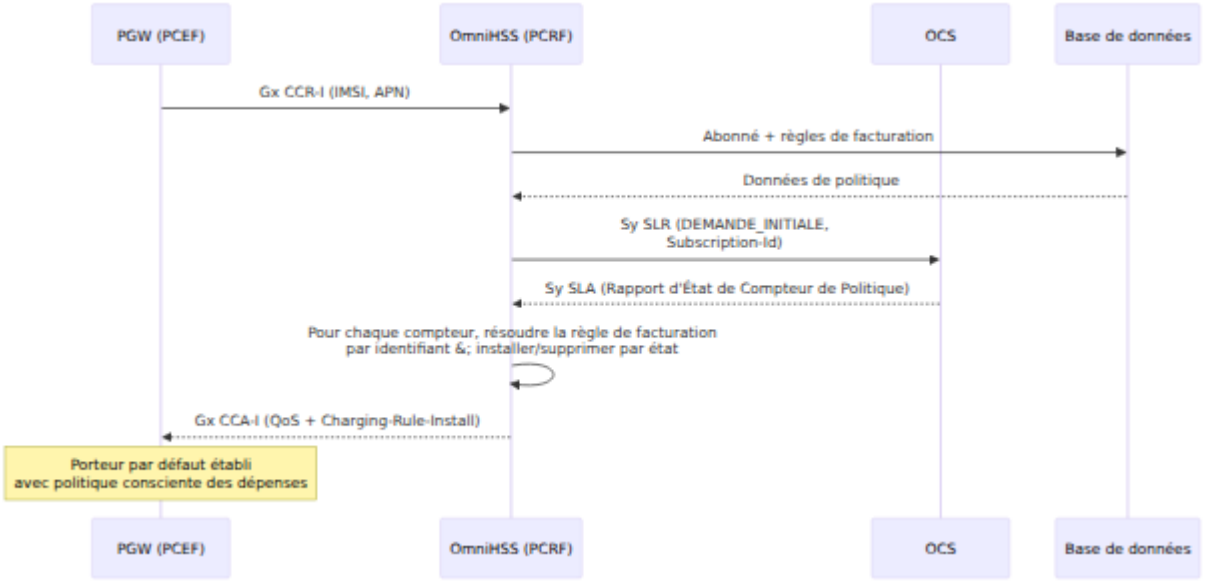
Le PCRF intègre les règles dérivées des dépenses dans le **même CCA-I** qu'il construit déjà pour la session Gx. Si l'OCS ne répond pas à temps, le PCRF revient à la politique par défaut de l'APN afin qu'une panne Sy ne bloque jamais l'attachement.

Flux 2 : Changement d'État de Dépense (OCS -> PCRF -> Gx RAR)



Cela réutilise la machinerie existante de Ré-Authentification Gx du PCRF - le même chemin qui installe des porteurs dédiés VoLTE - pour pousser le changement de politique basé sur les dépenses au PGW sans attendre le prochain CCR.

Flux 3 : Terminaison de la Session Sy



Configuration

Application Sy

L'application Sy est enregistrée sur le service Diameter HSS aux côtés de Gx et Rx (`config/runtime.exs`):

```
%{
  application_name: :sy,
  application_dictionary: :diameter_gen_3gpp_sy,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_302,
acct_application_id: nil}
  ]
}
```

Paramètre	Type	Requis	Par défaut	De
`application_name`	Atome	Oui	-	Nom inte l'interfac `:sy` pou Sy.
`application_dictionary`	Atome	Oui	-	Module d Diamete `:diamet` (3GPP TS
`vendor_specific_application_ids`	Liste	Oui	-	Annonce lors de l' capacité dessous.
`vendor_id`	Entier	Oui	-	ID du fou pour 3GP
`auth_application_id`	Entier	Oui	-	ID d'appl d'authen `16_777_`
`acct_application_id`	Entier | nil	Non	`nil`	ID d'appl facturati par Sy ;

Paramètres Sy

Le comportement de Sy est configuré sous la clé `:hss, :sy` (`config/runtime.exs`):

```
config :hss,
  sy: %{
    enabled: false
  }
```

Paramètre	Type	Requis	Par défaut	Description
`enabled`	Booléen	Non	`false`	Interrupteur principal. Lorsque `true`, le PCRF ouvre une session Sy par abonné sur Gx CCR-I. Lorsque `false`, Sy est complètement contourné.

La valeur `Policy-Counter-Status` qui installe la règle de facturation nommée est la convention fixe **`enforced`** ; toute autre valeur supprime la règle. Cela est convenu avec l'OCS et n'est pas configurable.

Routage OCS

Le HSS ne fixe **pas** un hôte ou un domaine OCS. Les SLR et STR d'origine portent le **propre** `Destination-Realm` du HSS et aucun `Destination-Host`, et sont routés par domaine plus l'ID d'application Sy (16777302). Quel que soit le pair qui annonce l'application Sy dans ce domaine, il reçoit la demande - typiquement le **DRA**, qui transfère à un OCS disponible. Cela fonctionne de manière identique que l'OCS soit atteint via le DRA ou directement en pair, et cela permet à la couche de routage de choisir parmi plusieurs OCS.

La seule exigence est qu'un pair annonçant l'application Sy existe dans le domaine. Dans un déploiement DRA, c'est le pair DRA déjà configuré pour le HSS ; aucune entrée de pair spécifique à Sy n'est nécessaire.

Mappage des Compteurs de Politique

Pour que le lien identifiant-règle fonctionne, l'opérateur configure **les deux** extrémités de manière cohérente :

- Sur l'**OCS** : l'`Identifiant de Compteur de Politique` de chaque compteur de politique est défini sur le `nom` exact d'une Règle de Facturation définie dans le HSS, et le compteur rapporte l'état `enforced` lorsque cette règle doit être appliquée.
- Sur le **HSS** : les règles de facturation référencées existent dans la base de données.

Voir Compteurs de Politique et Règles de Facturation pour le modèle de mappage.

Métriques

Sy expose des métriques Prometheus aux côtés des autres métriques d'interface HSS (voir Métriques & Surveillance).

Métrique: `hss_sy_messages_total` **Type**: Compteur **Description**: Messages Sy traités, par commande, direction et résultat. **Étiquettes**:

- `command` - `slr`, `sla`, `snr`, `sna`, `str`, ou `sta`
- `direction` - `incoming` (reçu de l'OCS) ou `outgoing` (envoyé à l'OCS)
- `result` - `sent`, `received`, `success`, ou `error`

Métrique: `hss_sy_policy_updates_total` **Type**: Compteur **Description**: Changements de règles de facturation Gx déclenchés par un état de compteur de politique Sy, par action et résultat. **Étiquettes**:

- `action` - `install` ou `remove`
- `result` - `success` ou `error` (le résultat du Gx RAR au PGW)

Exemples de requêtes:

```
# Taux de notification Sy de l'OCS
rate(hss_sy_messages_total{command="snr", direction="incoming"}
[5m])

# Taux d'installations de règles de limitation/application
déclenchées par des limites de dépenses
rate(hss_sy_policy_updates_total{action="install",
result="success"}[5m])

# Ratio d'échec de Sy Gx RAR
sum(rate(hss_sy_policy_updates_total{result="error"}[5m]))
  / sum(rate(hss_sy_policy_updates_total[5m]))
```

Documentation de Référence

Commandes

Commande	Abréviation	Code	Direction	But
Demande de Limite de Dépense	SLR	8388635	PCRF -> OCS	Demander l'état du compteur de politique ; s'abonner aux changements
Réponse de Limite de Dépense	SLA	8388635	OCS -> PCRF	Retourner l'état actuel du compteur
Demande de Notification de Statut de Dépense	SNR	8388636	OCS -> PCRF	Notifier un changement d'état de compteur
Réponse de Notification de Statut de Dépense	SNA	8388636	PCRF -> OCS	Accuser réception d'une notification
Demande de Terminaison de Session	STR	275	PCRF -> OCS	Fermer la session Sy (rapport final)
Réponse de Terminaison de Session	STA	275	OCS -> PCRF	Confirmer la fermeture de la session

Toutes les commandes portent l'ID d'application `16777302`, y compris les STR/STA réutilisés (RFC 6733 terminaison de session).

AVPs Clés

AVP	Code	Type	Description
`Policy-Counter-Identifier`	2901	UTF8String	Nomme un compteur de politique. Lié à un nom de Règle de Facturation dans OmniHSS.
`Policy-Counter-Status`	2902	UTF8String	Étiquette d'état définie par l'opérateur pour le compteur. Déclenche l'action d'application.
`Policy-Counter-Status-Report`	2903	Groupé	Identifiant d'un compteur + état (+ tout état en attente).
`SL-Request-Type`	2904	Énuméré	`INITIAL_REQUEST (0)` pour la première demande, `INTERMEDIATE_REQUEST (1)` par la suite.
`Pending-Policy-Counter-Information`	2905	Groupé	Un état futur et le moment où il devient actuel.
`Pending-Policy-Counter-Change-Time`	2906	Temps	Quand un état en attente prend effet.
`SN-Request-Type`	2907	Unsigned32	Masque de bits sur un SNR. Bit 0 = rapport normal, bit 1 = abort (l'OCS demande de mettre fin à la session Sy).
`Subscription-Id`	443	Groupé	Identité de l'abonné (IMSI et/ou MSISDN) sur le SLR.

Codes de Résultat

Code	Nom	Significatio
2001	DIAMETER_SUCCESS	Demande traitée avec succès
5030	DIAMETER_USER_UNKNOWN	L'OCS ne connaît pas l'abonné ; la session Sy n'est pas créée
5570	DIAMETER_ERROR_UNKNOWN_POLICY_COUNTERS	Un ou plusieurs compteurs d politique demandés sont inconnu de l'OCS
4241	DIAMETER_ERROR_NO_AVAILABLE_POLICY_COUNTERS	L'OCS n'a pa de compteur de politique pour l'abonn

Les codes 5570 et 4241 sont des valeurs de Code de Résultat Expérimental spécifiques à Sy (TS 29.219 §5.5) ; 5030 est défini par RFC 4006.

Documentation Connexe

- PCRF - Contrôle de politique Gx et Rx que Sy alimente
- Protocoles Diameter - Flux de messages à travers toutes les interfaces HSS
- Architecture - Architecture globale du HSS

Guide de Dépannage OmniHSS

← Retour au Guide des Opérations

Table des Matières

- Aperçu du Dépannage
- Échecs d'Authentification
- Problèmes de Connectivité Diameter
- Problèmes de Base de Données
- Échecs d'Enregistrement EPC
- Échecs d'Enregistrement IMS
- Échecs d'Appels VoLTE
- Problèmes de Roaming
- Problèmes EIR
- Problèmes de Performance
- Problèmes d'État des Abonnés
- Problèmes API
- Outils et Commandes de Diagnostic

Aperçu du Dépannage

Approche Générale du Dépannage



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

Get an estima

No thanks

Informations à Collecter

Avant de dépanner un problème, collectez :

1. **Informations sur l'Abonné** (si spécifique à l'abonné)
    - IMSI
    - MSISDN (numéro de téléphone)
    - Dernier état connu
    - Messages d'erreur du dispositif
2. **Informations de Timing**
    - Quand le problème a-t-il commencé ?
    - Est-il intermittent ou constant ?
    - Heure de la dernière opération réussie
3. **Portée de l'Impact**
    - Abonné unique ou multiple ?
    - Réseau spécifique ou tous les réseaux ?
    - Service spécifique (données/voix) ou les deux ?
4. **État du Système**
    - Vérifiez le Panneau de Contrôle pour l'état du système
    - Examinez l'état des pairs Diameter
    - Vérifiez la connectivité à la base de données

Échecs d'Authentification

Symptômes

- L'abonné ne peut pas se connecter au réseau
- Erreurs "Authentification rejetée"
- Tentatives d'authentification répétées

Causes et Solutions Courantes

Cause 1 : Ensemble de Clés Incorrect

Symptômes :

- Échec d'authentification constant pour un abonné spécifique
- Fonctionne pour d'autres abonnés avec le même profil

Étapes de Diagnostic :

1. Interroger l'abonné pour vérifier key_set_id :

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/[IMSI]
```

2. Vérifier que l'ensemble de clés existe et a les bonnes valeurs :

```
curl -k https://hss.example.com:8443/api/key_set/[KEY_SET_ID]
```

3. Comparer les valeurs Ki et OPC avec la documentation de la carte SIM

Solution :

- Mettre à jour l'abonné avec le bon ensemble de clés
- Si les clés sont correctes, la carte SIM peut être défectueuse

Cause 2 : SQN Hors Synchronisation

Symptômes :

- L'authentification échoue après avoir fonctionné précédemment
- Erreur : "Échec de synchronisation SQN"
- Fonctionne de manière intermittente

Étapes de Diagnostic :

1. Vérifier l'état de l'abonné pour la valeur SQN dans la base de données
2. Rechercher des erreurs liées à SQN dans les journaux

3. Vérifier la valeur SQN de l'ensemble de clés de l'abonné

Solution :

- SQN se resynchronisera automatiquement après que l'abonné envoie AUTS
- Si persistant, réinitialiser SQN à 0 dans l'ensemble de clés (nécessite une nouvelle connexion de l'abonné)

Avertissement : La réinitialisation de SQN peut causer des problèmes de sécurité. Ne le faire qu'en période de maintenance.

Cause 3 : Abonné Désactivé

Symptômes :

- Authentification rejetée immédiatement
- Aucun vecteur d'authentification généré

Étapes de Diagnostic :

1. Vérifier l'état d'activation de l'abonné :

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/[IMSI]
```

2. Vérifier que le champ `enabled` est `true`

Solution :

- Activer l'abonné :

```
curl -k -X PUT https://hss.example.com:8443/api/subscriber/[ID]
\
  -H "Content-Type: application/json" \
  -d '{"subscriber": {"enabled": true}}'
```

Cause 4 : Profil EPC Manquant

Symptômes :

- La recherche d'abonné réussit mais l'authentification échoue

- Erreur : "Aucun profil EPC assigné"

Étapes de Diagnostic :

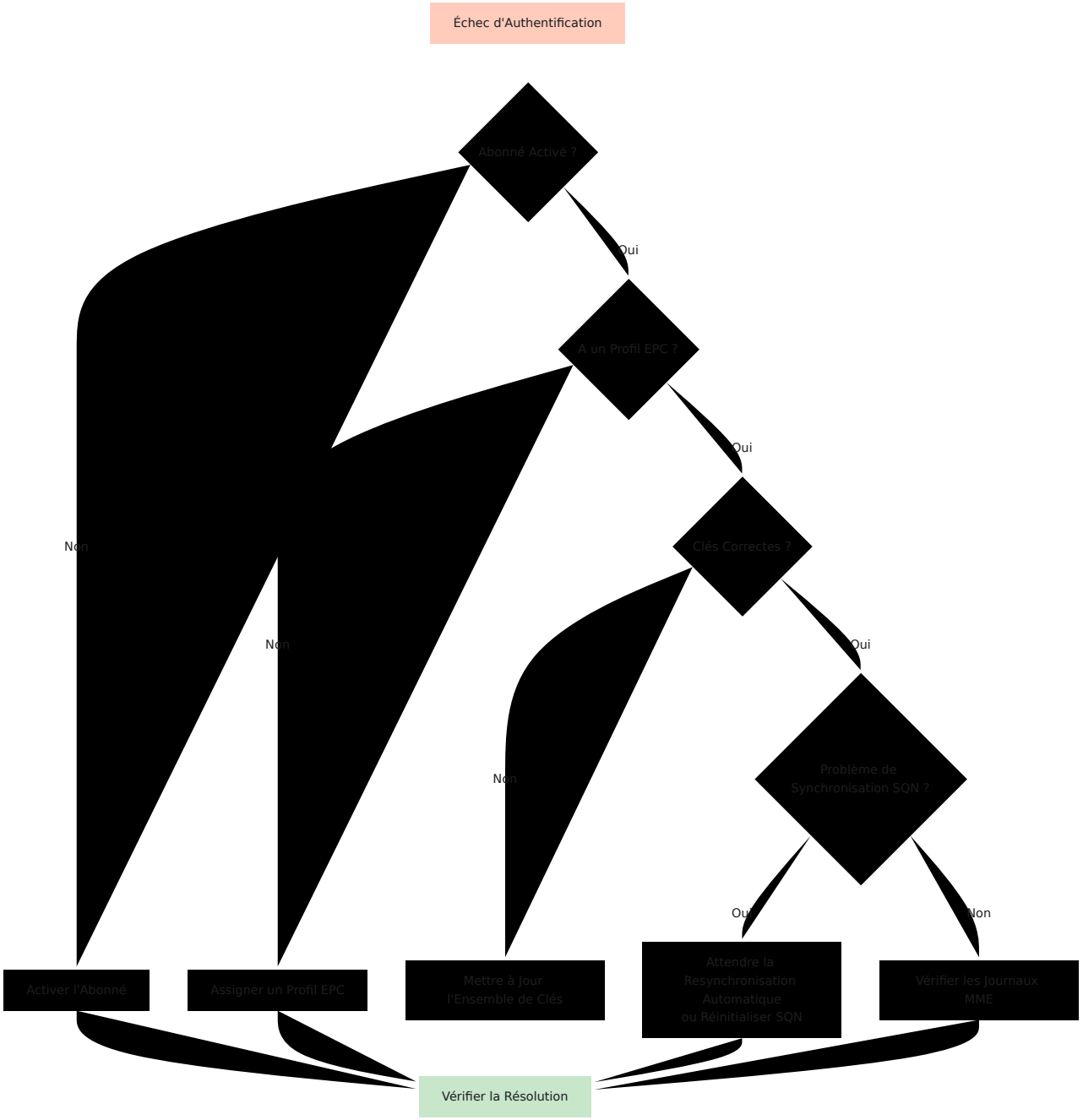
1. Vérifier le champ `epc_profile_id` de l'abonné
2. Vérifier que le profil EPC existe :

```
curl -k
https://hss.example.com:8443/api/epc/profile/[PROFILE_ID]
```

Solution :

- Assigner un profil EPC valide à l'abonné

Organigramme de Dépannage d'Authentification



Problèmes de Connectivité Diameter

Symptômes

- Les pairs Diameter apparaissent comme déconnectés dans le Panneau de Contrôle
- Erreurs "Aucun route vers l'hôte"
- Services échouent pour tous les abonnés

Causes et Solutions Courantes

Cause 1 : Connectivité Réseau

Symptômes :

- Le pair ne se connecte jamais
- Erreurs de délai d'attente de connexion
- Ping échoue vers le pair

Étapes de Diagnostic :

1. Vérifier la connectivité réseau de l'OmniHSS au pair :

   ```
   ping [PEER_IP]
   ```

2. Vérifier si le port Diameter est accessible :

   ```
   telnet [PEER_IP] 3868
   ```

3. Vérifier que les règles de pare-feu autorisent le trafic Diameter (port 3868)

Solution :

- Corriger le routage réseau

- Mettre à jour les règles de pare-feu
- Vérifier que le pair fonctionne et écoute

Cause 2 : Configuration Diameter Incorrecte

Symptômes :

- Les tentatives de connexion échouent
- L'échange CER/CEA échoue
- Le pair rejette la connexion

Étapes de Diagnostic :

1. Examiner la configuration Diameter dans runtime.exs :
    - Vérifier que origin_host du pair correspond à la valeur attendue
    - Vérifier la configuration origin_realm
    - Vérifier que l'adresse IP du pair est correcte
2. Vérifier les journaux pour les erreurs CER/CEA
3. Vérifier que la configuration du pair attend l'origin_host de l'OmniHSS

Solution :

- Mettre à jour runtime.exs avec la bonne configuration Diameter
- Redémarrer l'OmniHSS après modification de la configuration
- Coordonner avec l'administrateur du pair pour vérifier les paramètres

Cause 3 : Problèmes de Certificat (TLS Diameter)

Symptômes :

- La connexion échoue lors de la négociation TLS
- Erreurs de validation de certificat
- Erreurs "Certificat expiré" ou "Certificat invalide"

Étapes de Diagnostic :

1. Vérifier que les fichiers de certificat existent dans `priv/cert/`

2. Vérifier l'expiration du certificat :

```
openssl x509 -in priv/cert/diameter.crt -noout -dates
```

3. Vérifier que la chaîne de certificats est complète
4. Vérifier le certificat du pair si TLS mutuel

Solution :

- Renouveler les certificats expirés
- Installer la chaîne de certificats correcte
- Mettre à jour les fichiers de certificat et redémarrer l'OmniHSS

Cause 4 : Incompatibilité de Support d'Application du Pair

Symptômes :

- Le pair se connecte mais ne prend pas en charge les applications requises
- L'échange de capacités réussit mais les opérations échouent
- Erreurs "Application non supportée"

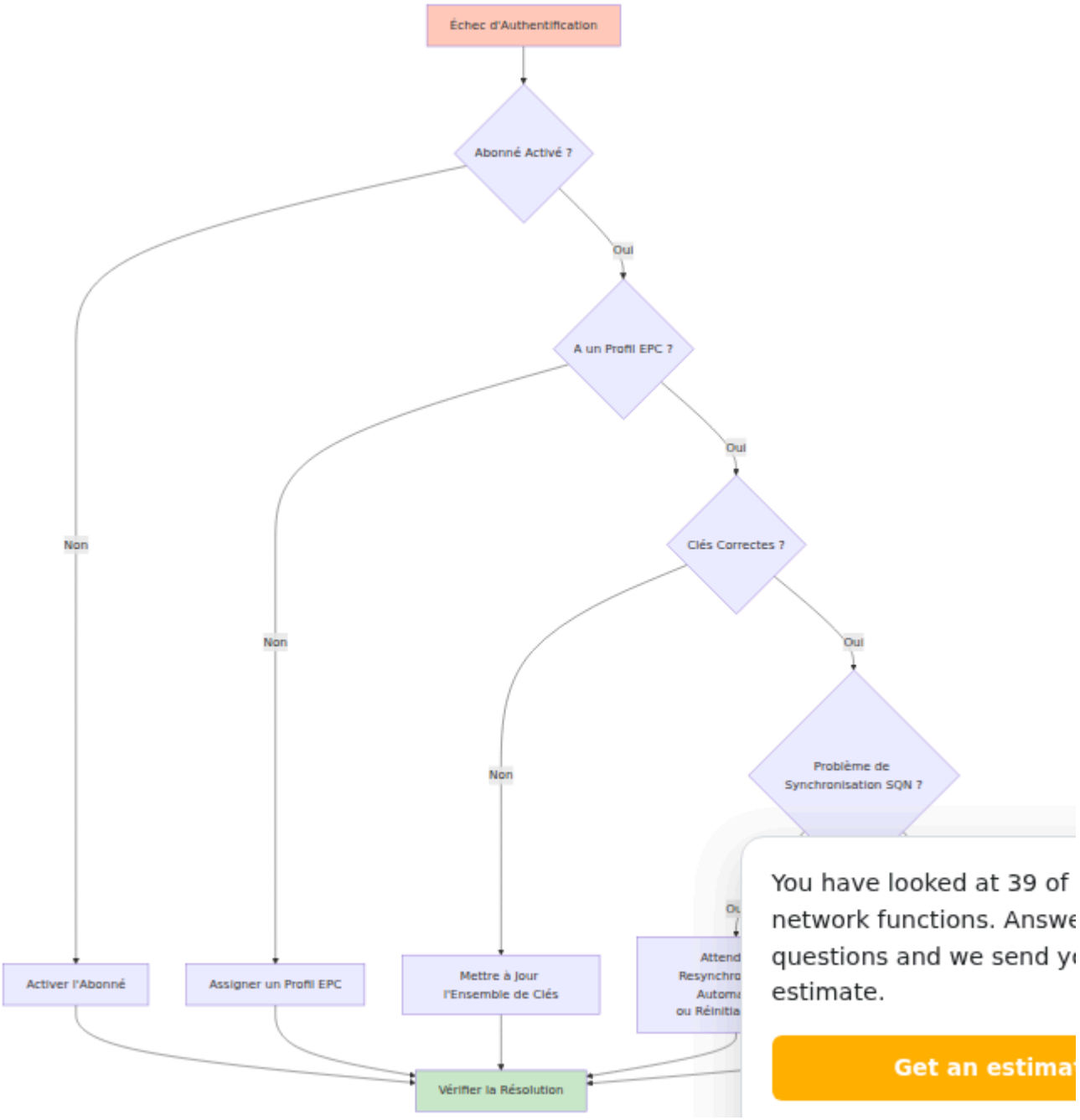
Étapes de Diagnostic :

1. Vérifier la page Diameter du Panneau de Contrôle pour les applications du pair
2. Vérifier que le pair prend en charge l'application requise (S6a, Cx, Sh, etc.)
3. Examiner l'échange CER/CEA dans les journaux

Solution :

- Vérifier que la configuration du pair inclut les applications Diameter requises
- Vérifier que le type de pair correspond à la fonctionnalité attendue :
  - MME doit prendre en charge S6a (16777251)
  - S-CSCF doit prendre en charge Cx (16777216)
  - P-GW doit prendre en charge Gx (16777238)

Organigramme de Dépannage Diameter



Problèmes de Base de Donn��es

Symptômes

- L'API renvoie des erreurs 500
- Le Panneau de Contrôle ne se charge pas

- Erreurs "Échec de la connexion à la base de données"
- Performance de requête lente

Causes et Solutions Courantes

Cause 1 : Serveur de Base de Données Hors Service

Symptômes :

- Tous les appels API échouent
- Le Panneau de Contrôle affiche une erreur
- Erreurs "Connexion refusée"

Étapes de Diagnostic :

1. Tester la connectivité à la base de données :

   ```
   # Se connecter à la base de données
   psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME]
   ```

2. Vérifier l'état du service de base de données sur le serveur de base de données

3. Vérifier la connectivité réseau vers le serveur de base de données

Solution :

- Démarrer le service de base de données
- Corriger les problèmes du serveur de base de données
- Vérifier le routage réseau vers le serveur de base de données

Cause 2 : Identifiants de Base de Données Incorrects

Symptômes :

- Erreurs "Authentification échouée"
- OmniHSS ne peut pas se connecter au démarrage

Étapes de Diagnostic :

1. Examiner la configuration de la base de données dans runtime.exs
2. Tester les identifiants manuellement avec le client de base de données
3. Vérifier les permissions de l'utilisateur de la base de données

Solution :

- Mettre à jour la configuration de la base de données dans runtime.exs
- Accorder les bonnes permissions à l'utilisateur de la base de données
- Redémarrer l'OmniHSS après modification de la configuration

Cause 3 : Pool de Connexion Épuisé

Symptômes :

- Erreurs 500 intermittentes
- Erreurs "Aucune connexion disponible"
- Les périodes de forte charge déclenchent des échecs

Étapes de Diagnostic :

1. Vérifier le nombre de connexions actuelles dans la base de données
2. Examiner la taille du pool de base de données dans runtime.exs
3. Surveiller l'utilisation des connexions pendant les périodes de forte charge

Solution :

- Augmenter la taille du pool dans la configuration runtime.exs
- Enquêter sur les fuites de connexion si le pool s'épuise de manière répétée
- Envisager une mise à l'échelle de la base de données si la charge est constamment élevée

Cause 4 : Requêtes Lentes

Symptômes :

- Réponses API très lentes
- Délais d'attente lors des recherches d'abonnés

- CPU de la base de données élevé

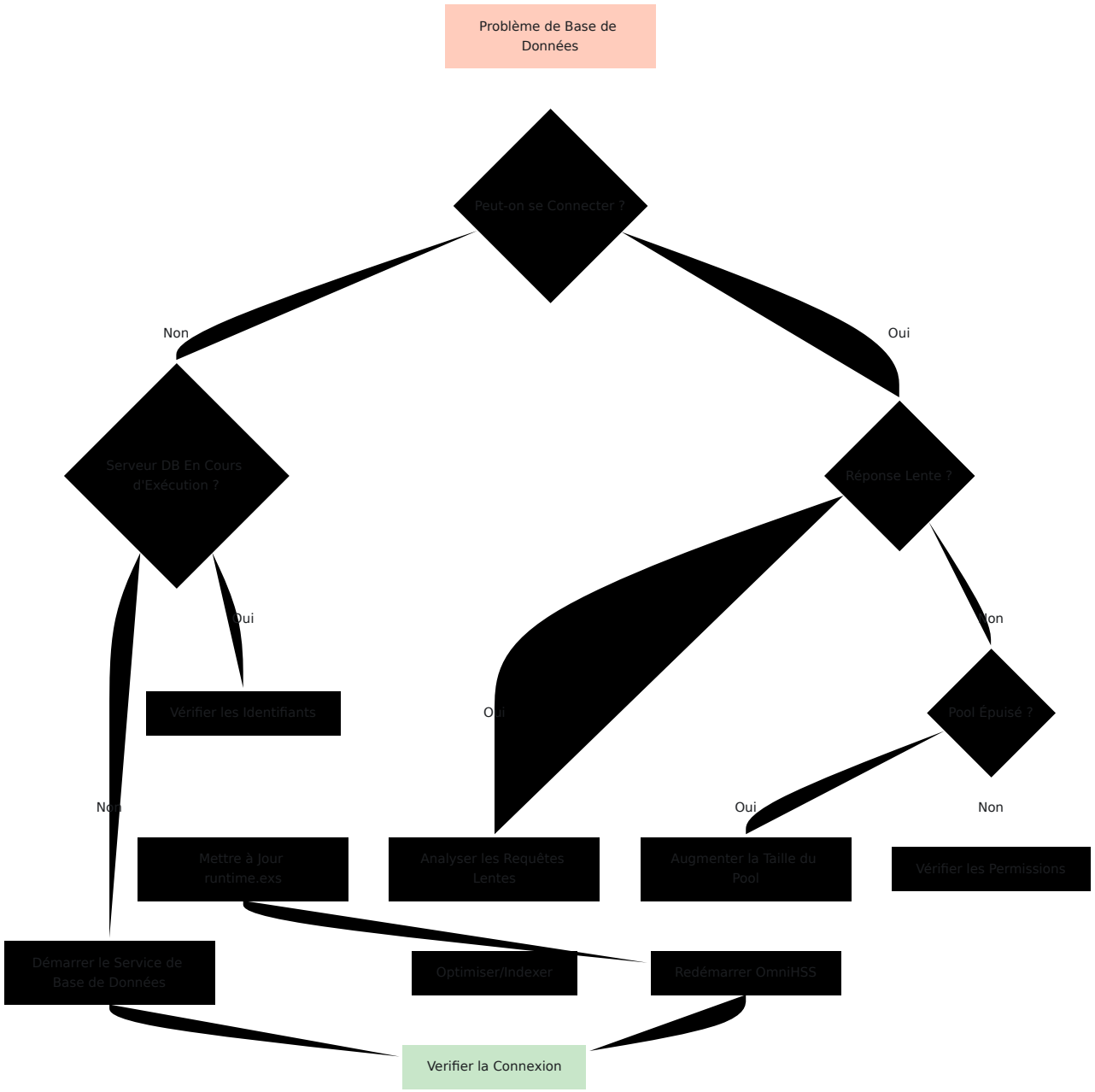
Étapes de Diagnostic :

1. Interroger la base de données pour le journal des requêtes lentes
2. Identifier les requêtes lentes spécifiques
3. Vérifier les index manquants
4. Vérifier le nombre d'abonnés et la taille des tables

Solution :

- Optimiser les requêtes lentes
- Ajouter des index manquants
- Envisager un réglage des performances de la base de données
- Planifier une mise à l'échelle de la base de données si nécessaire

Organigramme de Dépannage de Base de Données



Échecs d'Enregistrement EPC

Symptômes

- L'abonné ne peut pas se connecter au réseau LTE
- MME rejette la connexion

- Aucune session PDN établie

Causes et Solutions Courantes

Cause 1 : Roaming Refusé

Symptômes :

- L'abonné fonctionne sur le réseau domestique mais échoue en roaming
- Erreurs "Roaming non autorisé"
- Fonctionne pour certains réseaux mais pas d'autres

Étapes de Diagnostic :

1. Vérifier le champ roaming_profile_id de l'abonné
2. Interroger le profil de roaming et les règles
3. Vérifier le MCC/MNC du réseau visité
4. Vérifier si une règle de roaming existe pour ce réseau

Solution :

- Ajouter une règle de roaming pour le MCC/MNC du réseau visité
- Ou mettre à jour l'action par défaut du profil de roaming pour autoriser
- Voir la Documentation sur le Roaming pour la configuration

Cause 2 : Configuration APN Manquante

Symptômes :

- La connexion réussit mais la session PDN échoue
- Erreurs "APN inconnu" du MME
- L'abonné ne peut pas obtenir de connexion de données

Étapes de Diagnostic :

1. Vérifier que le profil EPC a des profils APN liés
2. Vérifier que l'identifiant APN correspond à ce que le dispositif demande
3. Interroger la configuration du profil APN

Solution :

- Lier les profils APN au profil EPC de l'abonn��
- S'assurer que le nom APN correspond à la configuration du dispositif
- Vérifier que le profil QoS APN existe

Cause 3 : MME Non Connecté

Symptômes :

- Tous les abonnés échouent à se connecter
- Aucune communication avec le MME
- Pair Diameter hors service

Étapes de Diagnostic :

1. Vérifier la page Diameter du Panneau de Contrôle
2. Vérifier que l'état du pair MME est "Connecté"
3. Vérifier que le MME prend en charge l'application S6a

Solution :

- Dépanner la connectivité Diameter
- Vérifier la configuration du MME
- Contacter l'administrateur du MME

Cause 4 : Corruption de l'État de l'Abonné

Symptômes :

- L'abonné apparaît comme connecté mais ne peut pas se reconnecter
- L'état ne correspond pas à la réalité
- La déconnexion et la reconnexion échouent

Étapes de Diagnostic :

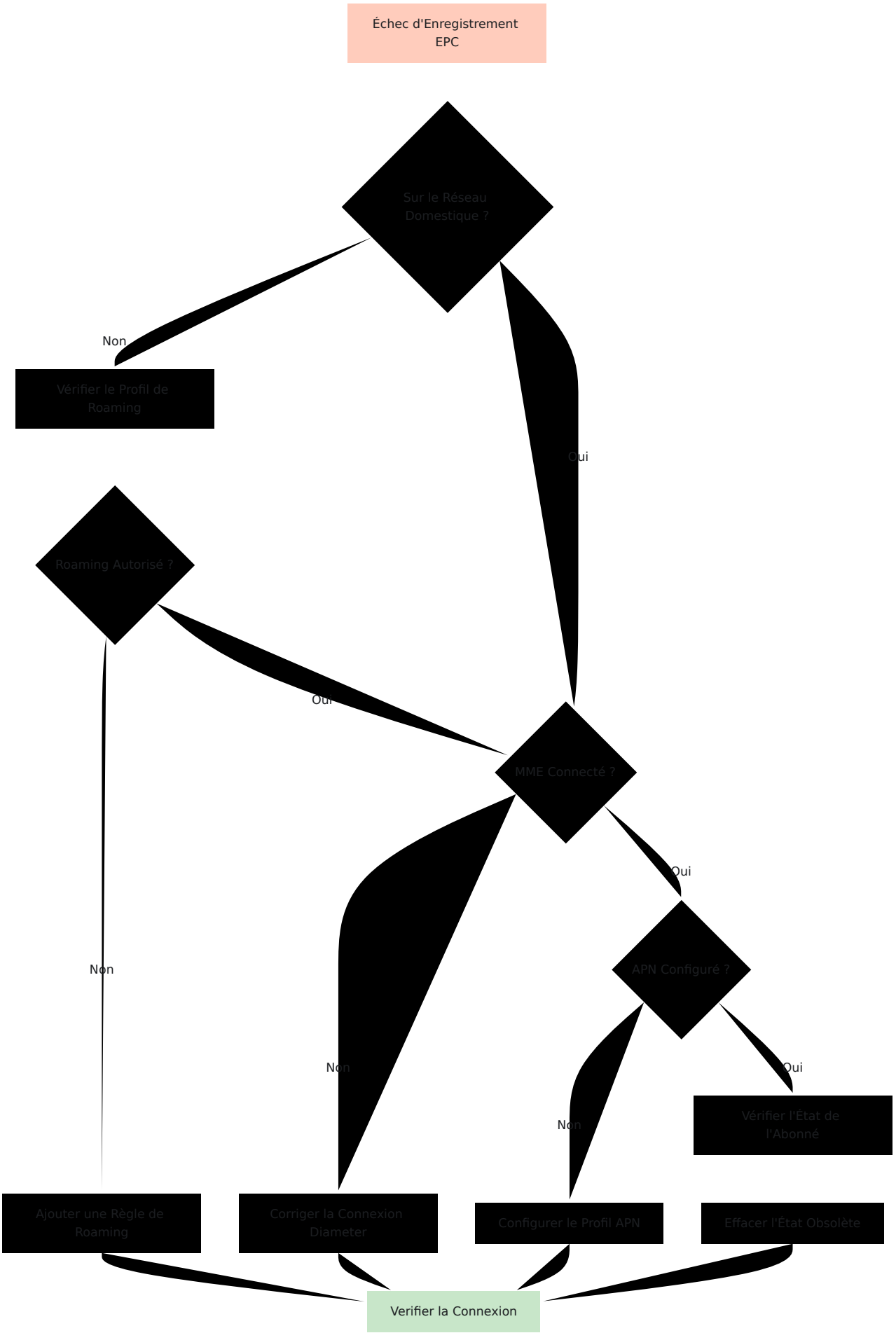
1. Interroger l'état de l'abonné dans la base de données
2. Vérifier les affectations MME obsolètes
3. Vérifier l'horodatage de la dernière mise à jour

Solution :

- Effacer l'état de l'abonné (procédure de déconnexion)
- Réinitialiser le MME de service dans l'état de l'abonné
- Peut nécessiter un cycle d'alimentation de l'abonné

Organigramme de Dépannage

d'Enregistrement EPC


Échec d'Enregistrement EPC
Sur le Réseau Domestique ?
Non
Vérifier le Profil de Roaming
Oui
Roaming Autorisé ?
Oui
MME Connecté ?
Oui
APN Configuré ?
Non
Non
Non
Oui
Vérifier l'État de l'Abonné
Ajouter une Règle de Roaming
Corriger la Connexion Diameter
Configurer le Profil APN
Effacer l'État Obsolète
Verifier la Connexion

Échecs d'Enregistrement IMS

Symptômes

- L'abonné ne peut pas s'enregistrer pour VoLTE
- "Échec de l'enregistrement IMS" sur le dispositif
- Les données fonctionnent mais la voix ne fonctionne pas

Causes et Solutions Courantes

Cause 1 : IMS Désactivé pour l'Abonné

Symptômes :

- L'abonné a des données mais pas d'IMS
- Enregistrement rejeté immédiatement

Étapes de Diagnostic :

1. Interroger l'abonné et vérifier le champ `ims_enabled`
2. Vérifier que l'abonné a un `ims_profile_id` assigné

Solution :

- Activer l'IMS pour l'abonné
- Assigner un profil IMS

Cause 2 : S-CSCF Non Connecté

Symptômes :

- Tous les enregistrements IMS échouent
- Aucun trafic Diameter lié à l'IMS

Étapes de Diagnostic :

1. Vérifier la page Diameter du Panneau de Contrôle

2. Vérifier que le pair S-CSCF est connecté
3. Vérifier que le S-CSCF prend en charge l'application Cx

Solution :

- Corriger la connectivité Diameter vers le S-CSCF
- Vérifier la configuration du S-CSCF

Cause 3 : Modèle IFC Manquant ou Invalide

Symptômes :

- L'enregistrement échoue pendant la Réponse d'Autorisation d'Utilisateur
- Erreurs liées à l'IFC dans les journaux

Étapes de Diagnostic :

1. Interroger le profil IMS de l'abonné
2. Vérifier que le modèle IFC est présent
3. Vérifier la syntaxe XML de l'IFC

Solution :

- Mettre à jour le profil IMS avec un modèle IFC valide
- Voir la Documentation des Profils pour des exemples d'IFC

Cause 4 : Roaming Refusé pour l'IMS

Symptômes :

- L'IMS fonctionne sur le réseau domestique
- Échoue en roaming
- Le roaming de données fonctionne mais pas l'IMS

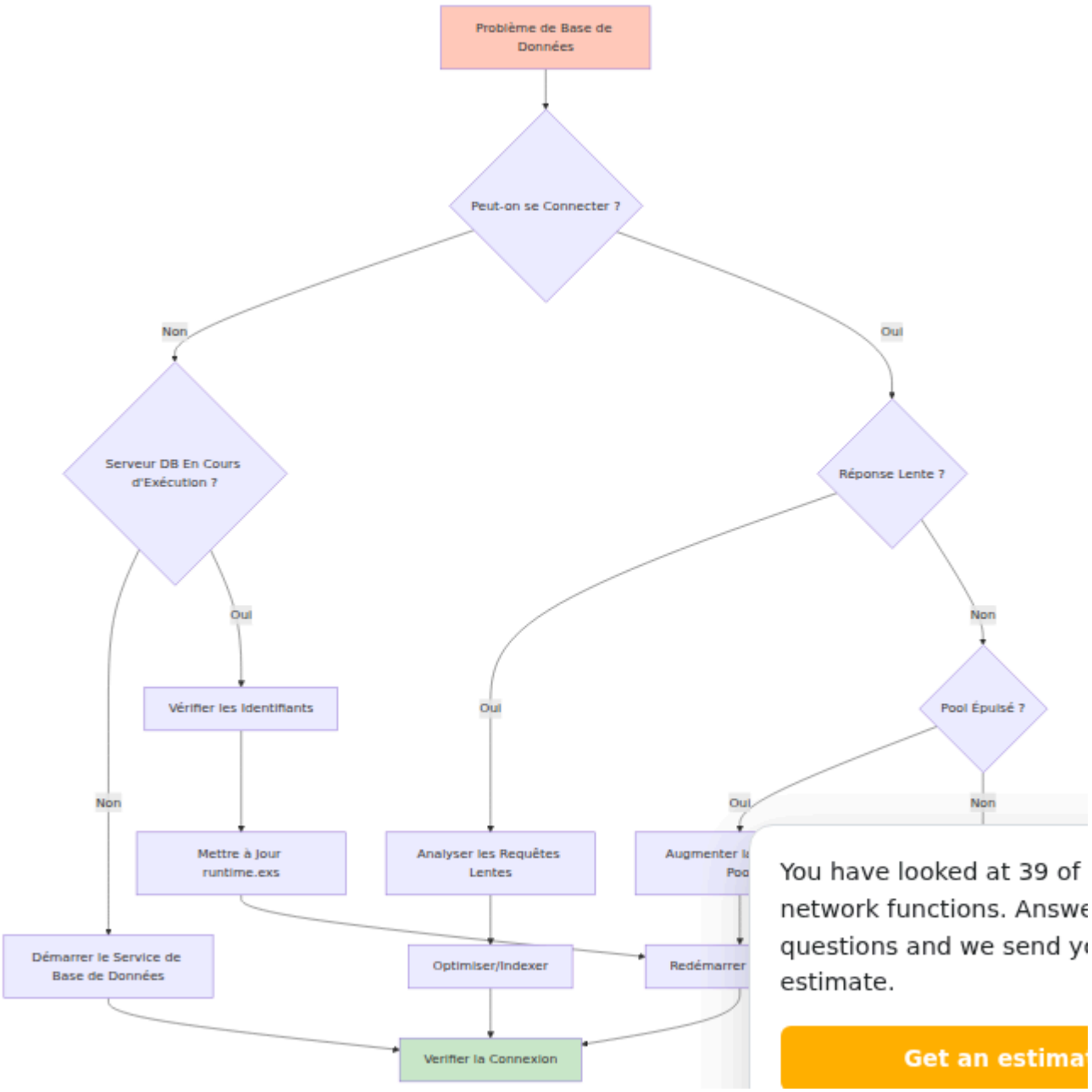
Étapes de Diagnostic :

1. Vérifier l'action IMS du profil de roaming
2. Vérifier que les règles de roaming ont le bon `ims_action`

Solution :

- Mettre à jour les règles de roaming pour autoriser l'IMS
- Ou mettre à jour l'action IMS par défaut du profil de roaming

Organigramme de Dépannage d'Enregistrement IMS



Échecs d'Appels VoLTE

Symptômes

- L'enregistrement IMS réussit mais les appels échouent
- Audio unidirectionnel
- L'appel se coupe immédiatement
- Erreur "Appel échoué" sur le dispositif

Causes et Solutions Courantes

Cause 1 : P-CSCF Non Connecté

Symptômes :

- L'enregistrement fonctionne mais les appels échouent
- L'autorisation des médias échoue

Étapes de Diagnostic :

1. Vérifier la page Diameter du Panneau de Contrôle
2. Vérifier que le pair P-CSCF est connecté
3. Vérifier que le P-CSCF prend en charge l'application Rx (fonction PCRF d'OmniHSS)

Solution :

- Corriger la connectivité Diameter vers le P-CSCF
- Vérifier que la configuration du P-CSCF pointe vers OmniHSS pour Rx

Cause 2 : Autorisation des Médias Manquante

Symptômes :

- La configuration de l'appel commence mais échoue
- Échange AAR/AAA échoue
- Erreurs d'interface Rx

Étapes de Diagnostic :

1. Vérifier les journaux pour les messages Diameter Rx
2. Vérifier que AAR (AA-Request) a été reçu
3. Vérifier la réponse AAA (AA-Answer)

Solution :

- Vérifier que le P-CSCF envoie AAR pour l'autorisation des médias
- Vérifier la configuration de l'application Rx d'OmniHSS
- Vérifier que l'abonné a un enregistrement IMS actif

Cause 3 : Problèmes de QoS/Bearer

Symptômes :

- L'appel se connecte mais pas d'audio
- Audio unidirectionnel
- Problèmes de qualité

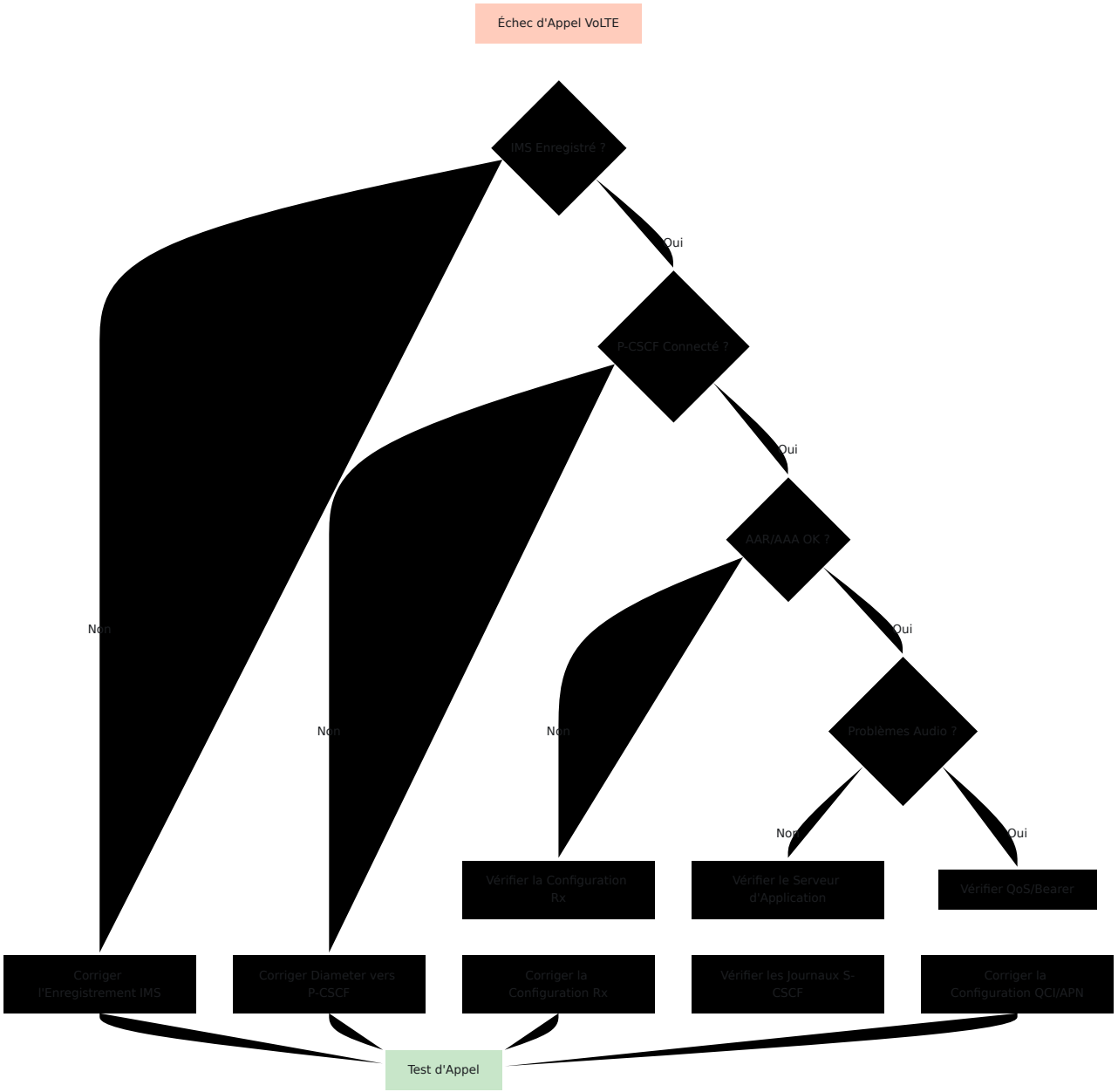
Étapes de Diagnostic :

1. Vérifier le profil QoS APN pour l'APN de voix
2. Vérifier que le QCI est correctement défini (généralement QCI 1 pour la voix)
3. Vérifier que le P-GW est connecté pour Gx (fonction PCRF)

Solution :

- Vérifier le profil QoS APN pour l'APN IMS
- S'assurer que QCI 1 est configuré pour le bearer vocal
- Corriger la connectivité Diameter vers le P-GW si nécessaire

Organigramme de Dépannage d'Appels VoLTE



Problèmes de Roaming

Symptômes

- L'abonné fonctionne à domicile mais pas en roaming
- Certains réseaux de roaming fonctionnent, d'autres non
- Le roaming de données fonctionne mais pas la voix (ou vice versa)

Causes et Solutions Courantes

Cause 1 : Aucun Profil de Roaming Assigné

Symptômes :

- Le roaming échoue pour l'abonné
- D'autres abonnés roament avec succès

Étapes de Diagnostic :

1. Interroger l'`roaming_profile_id` de l'abonné
2. Vérifier si le champ est nul

Solution :

- Assigner un profil de roaming à l'abonné

Cause 2 : Roaming Refusé par Politique

Symptômes :

- Le roaming échoue systématiquement sur un réseau spécifique
- L'erreur indique un rejet de la politique

Étapes de Diagnostic :

1. Identifier le MCC/MNC du réseau visité depuis l'appareil de l'abonné ou le MME
2. Interroger le profil de roaming de l'abonné
3. Vérifier les règles de roaming pour le MCC/MNC correspondant
4. Vérifier l'action par défaut du profil

Solution :

- Ajouter une règle de roaming pour autoriser le réseau visité :

```
curl -k -X POST https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Allow Visited Network",
      "mcc": "310",
      "mnc": "410",
      "data_action": "allow",
      "ims_action": "allow"
    }
  }'
```

Cause 3 : Données Autorisées mais IMS Refusé

Symptômes :

- Le roaming de données fonctionne
- Le roaming vocal/IMS échoue
- Disponibilité du service divisée

Étapes de Diagnostic :

1. Interroger les règles de roaming pour le réseau visité
2. Vérifier les valeurs de `data_action` par rapport à `ims_action`
3. Vérifier les actions par défaut du profil de roaming

Solution :

- Mettre à jour la règle de roaming pour autoriser l'IMS :
  - Définir `ims_action: "allow"`
- Ou mettre à jour `ims_action_if_no_rules_match` du profil à `"allow"`

Voir la Documentation sur le Roaming pour une configuration détaillée.

Problèmes EIR

Symptômes

- Appareils bloqués de manière inattendue
- Appareils volés non bloqués
- Vérification EIR échouée

Causes et Solutions Courantes

Cause 1 : Regex IMEI Incorrect

Symptômes :

- Mauvais appareils bloqués/autorisé
- La règle correspond incorrectement

Étapes de Diagnostic :

1. Interroger les règles EIR (`GET /api/eir/rule`). La liste est retournée dans l'ordre `id`, donc triez-la par `precedence` pour voir l'ordre d'évaluation
2. Identifier quelle règle correspond. Les règles sont testées d'abord contre l'IMEISV (IMEI + Version-Logiciel), puis l'IMEI, puis l'IMSI si la demande en contient un. À chaque étape, la règle correspondante avec la plus basse `precedence` l'emporte, donc une règle IMSI à faible priorité n'est jamais atteinte si une règle a correspondu à l'IMEISV ou à l'IMEI. Voir Logique de Correspondance d'Équipement
3. Tester le modèle regex contre l'IMEISV, l'IMEI et l'IMSI réels. Les modèles ne sont pas ancrés, donc `35979139461650` correspond également à l'IMEISV `3597913946165008`
4. Vérifier la priorité des règles (les valeurs plus basses sont évaluées en premier)

Solution :

- Mettre à jour la règle EIR avec le regex correct
- Tester le regex de manière approfondie avant de l'appliquer

- Considérer la priorité des règles (la règle correspondante avec la plus basse priorité l'emporte)

Cause 2 : MME Ne Pas Envoyer de Requêtes S13

Symptômes :

- La vérification EIR n'a jamais lieu
- Tous les appareils sont autorisés indépendamment des règles

Étapes de Diagnostic :

1. Vérifier si le MME est configuré pour utiliser l'interface S13
2. Vérifier que le pair Diameter du MME est connecté
3. Vérifier le support de l'application S13
4. Revoir la configuration du MME

Solution :

- Configurer le MME pour effectuer des vérifications EIR via S13
- Vérifier que le pair Diameter prend en charge l'application S13 (16777252)
- Contacter l'administrateur MME si nécessaire

Cause 3 : Pas de Règle par Défaut

Symptômes :

- Les appareils ne correspondant à aucune règle ont un comportement inattendu

Étapes de Diagnostic :

1. Interroger toutes les règles EIR
2. Vérifier si une règle de rattrapage existe
3. Vérifier la priorité des règles

Solution :

- L'équipement qui ne correspond à aucune règle obtient le paramètre `unknown_equipment_behaviour` de `config :hss, :eir` (voir Configuration

EIR). Vérifiez cela en premier
- Si vous préférez une règle de rattrapage explicite, ajoutez une règle avec le regex `.*` et l'action souhaitée, et donnez-lui le plus haut numéro de `precedence` afin qu'elle soit testée en dernier à chaque étape
- Une règle `.*` correspond à la première étape qui s'exécute (IMEISV, ou IMEI lorsque la demande ne contient pas de Version-Logiciel), donc les règles basées sur l'IMSI et le défaut configuré ne sont jamais atteints pour les demandes qui contiennent un IMEI

Problèmes de Performance

Symptômes

- Réponses API lentes
- Délai d'attente des requêtes Diameter
- Utilisation élevée du CPU ou de la mémoire
- Panneau de contrôle lent à charger

Causes et Solutions Courantes

Cause 1 : Charge Élevée sur la Base de Données

Symptômes :

- Toutes les opérations sont lentes
- CPU de la base de données élevé
- Délai d'attente des requêtes

Étapes de Diagnostic :

1. Vérifier l'utilisation des ressources du serveur de base de données
2. Identifier les requêtes lentes
3. Vérifier les index manquants
4. Surveiller les modèles de requêtes

Solution :

- Optimiser les requêtes lentes
- Ajouter des index à la base de données
- Augmenter les ressources de la base de données
- Envisager une mise à l'échelle de la base de données
- Voir Problèmes de Base de Données

Cause 2 : Nombre Élevé d'Abonnés

Symptômes :

- Performance dégradée au fil du temps
- La lenteur est corrélée à la croissance des abonnés
- Les opérations de liste sont particulièrement lentes

Étapes de Diagnostic :

1. Interroger le nombre total d'abonnés
2. Vérifier les tailles des tables
3. Revoir les plans d'exécution des requêtes
4. Surveiller les tendances d'utilisation des ressources

Solution :

- Planifier une mise à niveau de la capacité
- Optimiser les requêtes pour de grands ensembles de données
- Envisager la pagination pour de grands résultats
- Implémenter la mise en cache si nécessaire

Cause 3 : Problèmes de Pair Diameter

Symptômes :

- Les opérations Diameter sont lentes
- Délai d'attente sur un pair spécifique
- Certains pairs rapides, d'autres lents

Étapes de Diagnostic :

1. Vérifier la page Diameter du Panneau de Contrôle
2. Identifier le pair lent
3. Tester la latence réseau vers le pair
4. Vérifier l'utilisation des ressources du pair

Solution :

- Enquêter sur les problèmes de performance du pair
- Vérifier le chemin réseau pour la congestion
- Envisager d'ajouter des pairs redondants
- Augmenter le délai d'attente Diameter si nécessaire

Cause 4 : Problèmes de Mémoire

Symptômes :

- Utilisation de la mémoire OmniHSS élevée
- Erreurs de mémoire insuffisante
- La performance se dégrade au fil du temps

Étapes de Diagnostic :

1. Vérifier l'utilisation de la mémoire OmniHSS sur la page Application
2. Surveiller la tendance de la mémoire
3. Vérifier les fuites de mémoire
4. Revoir les paramètres de la VM Erlang

Solution :

- Redémarrer OmniHSS pour effacer la condition temporaire
- Enquêter sur la fuite de mémoire si l'utilisation augmente continuellement
- Ajuster les paramètres de mémoire de la VM Erlang dans runtime.exs
- Planifier une mise à niveau matérielle si constamment élevée

---

Problèmes d'État des Abonnés

Symptômes

- L'abonné apparaît comme attaché mais ne l'est pas
- Informations d'état obsolètes
- Informations de localisation incorrectes
- Impossible de détacher l'abonné

Causes et Solutions Courantes

Cause 1 : Crash/Redémarrage du MME

Symptômes :

- L'abonné montre un MME de service qui ne sert plus
- L'abonné ne peut pas s'attacher après le redémarrage du MME
- L'état est obsolète

Étapes de Diagnostic :

1. Vérifier l'état de l'abonné pour le MME de service
2. Vérifier si le MME a redémarré
3. Vérifier le dernier temps de connexion du MME

Solution :

- Attendre que l'abonné s'attache à nouveau (l'état sera mis à jour)
- Ou effacer manuellement l'état de l'abonné
- Le MME doit envoyer Cancel-Location lors du redémarrage

Cause 2 : Détachement Réseau Non Reçu

Symptômes :

- L'abonné éteint mais apparaît comme attaché
- Les sessions PDN restent dans la base de données

- La localisation n'est pas effacée

Étapes de Diagnostic :

1. Vérifier le timestamp `last_seen` de l'abonné
2. Vérifier si l'ancien état (heures ou jours) est obsolète
3. Vérifier si l'appareil de l'abonné est accessible

Solution :

- L'état sera effacé lorsque l'abonné s'attachera à nouveau
- Ou attendre le délai d'expiration de l'état (si implémenté)
- Un nettoyage manuel peut être nécessaire pour un état très obsolète

Cause 3 : Corruption de la Base de Données

Symptômes :

- État incohérent à travers les tables
- Violations de clés étrangères
- L'état n'a pas de sens

Étapes de Diagnostic :

1. Interroger l'état de l'abonné directement depuis la base de données
2. Vérifier les enregistrements orphelins
3. Vérifier l'intégrité référentielle

Solution :

- Identifier et corriger les données incohérentes
- Peut nécessiter un nettoyage manuel de la base de données
- Contacter le support si la corruption est généralisée

Problèmes API

Symptômes

- L'API renvoie des erreurs
- Réponses API lentes
- Impossible de créer/mettre à jour des entités
- Erreurs 500

Causes et Solutions Courantes

Cause 1 : Données de Demande Invalides

Symptômes :

- Erreurs 400 ou 422
- Messages d'erreur de validation
- Champ rejeté

Étapes de Diagnostic :

1. Revoir la réponse d'erreur pour des erreurs de champ spécifiques
2. Vérifier le format de la demande API
3. Vérifier que tous les champs requis sont présents
4. Vérifier les types de données

Solution :

- Corriger les données de la demande pour correspondre à la référence API
- S'assurer que tous les champs requis sont inclus
- Vérifier que les références de clés étrangères existent (ID de profil, etc.)

Cause 2 : Contrainte de Clé Étrangère

Symptômes :

- Impossible de créer un abonné

- Erreur : "key_set_id n'existe pas"
- Entité référencée non trouvée

Étapes de Diagnostic :

1. Identifier quelle clé étrangère échoue
2. Vérifier que l'entité référencée existe :
    - key_set_id → ensembles de clés
    - epc_profile_id → profils EPC
    - ims_profile_id → profils IMS

Solution :

- Créer d'abord l'entité référencée
- Ou utiliser l'ID d'entité existant
- Suivre le flux de provisioning complet

Cause 3 : Connectivité à la Base de Données

Symptômes :

- Erreurs 500
- Tous les appels API échouent
- Erreurs de connexion à la base de données

Solution :

- Voir Problèmes de Base de Données

Outils et Commandes de Diagnostic

Vérifications Rapides du Panneau de Contrôle

1. **Aperçu du Système**

- URL : `https://[hostname]:7443/overview`
- Vérifier : Comptes d'abonnés, sessions actives, état du système

2. **Statut Diameter**
   - URL : `https://[hostname]:7443/diameter`
   - Vérifier : Tous les pairs critiques connectés
3. **Santé de l'Application**
   - URL : `https://[hostname]:7443/application`
   - Vérifier : Utilisation de la mémoire, nombre de processus, temps de fonctionnement

Commandes de Diagnostic API

Vérifier la Santé du Système :

```
curl -k https://hss.example.com:8443/api/status
```

Interroger un Abonné :

```
# Par IMSI
curl -k
https://hss.example.com:8443/api/subscriber/imsi/001001123456789

# Par MSISDN
curl -k
https://hss.example.com:8443/api/subscriber/msisdn/14155551234

# Par ID
curl -k https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-
4b01-90e6-d701748f0851
```

Lister Tous les Abonnés :

```
curl -k https://hss.example.com:8443/api/subscriber
```

Vérifier la Configuration du Profil :

```
# Profil EPC
curl -k https://hss.example.com:8443/api/epc/profile/e58ed763-
928c-4155-bee9-fdbaaadc15f3

# Profil IMS
curl -k https://hss.example.com:8443/api/ims/profile/7c9e6679-
7425-40de-944b-e07fc1f90ae7

# Profil de Roaming
curl -k https://hss.example.com:8443/api/roaming/profile/9c858901-
8a57-4791-81fe-4c455b099bc9
```

Commandes de Diagnostic Réseau

Tester la Connectivité du Port Diameter :

```
telnet [PEER_IP] 3868
```

Vérifier le Certificat TLS :

```
openssl s_client -connect [hostname]:8443 -showcerts
```

Tester la Connectivité à la Base de Données :

```
psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME] -c "SELECT COUNT(*)
FROM subscriber;"
```

Analyse des Journaux

Rechercher des Journaux pour un IMSI Spécifique :

```
grep "001001123456789" /var/log/omnihss/omnihss.log
```

Trouver des Échecs d'Authentification :

```
grep "authentication.*fail" /var/log/omnihss/omnihss.log
```

Vérifier les Événements des Pairs Diameter :

```
grep "Diameter peer" /var/log/omnihss/omnihss.log
```

Trouver des Erreurs de Base de Données :

```
grep -i "database.*error" /var/log/omnihss/omnihss.log
```

Directives d'Escalade

Quand Escalader

Escalader au support technique/fournisseur lorsque :

1. **Pannes systémiques** qui ne peuvent pas être résolues avec des procédures documentées
2. **Corruption de données** ou état incohérent de la base de données
3. **Bugs logiciels suspects** ou comportement inattendu
4. **Problèmes de performance** qui ne peuvent pas être résolus par des réglages
5. **Incidents de sécurité** ou accès non autorisé
6. **Questions sur un comportement non documenté**

Informations à Fournir

Lors de l'escalade, inclure :

1. **Symptômes détaillés** - Ce qui échoue, quand, pour qui

2. **Étapes prises** - Quel dépannage vous avez déjà effectué
3. **Journaux** - Extraits de journaux pertinents montrant le problème
4. **Configuration** - Portions pertinentes de runtime.exs (cacher les données sensibles)
5. **Environnement** - Version d'OmniHSS, version de la base de données, version de l'OS
6. **Impact** - Combien d'abonnés affectés, impact commercial
7. **Exemples d'abonnés** - IMSIs spécifiques montrant le problème

Critique vs Non-Critique

Problèmes Critiques (Escalader Immédiatement) :

- Système complètement hors service
- Tous les abonnés incapables de s'attacher
- Corruption de la base de données
- Violation de la sécurité

Problèmes Non-Critiques (Documenter et Escalader Pendant les Heures de Bureau) :

- Problèmes d'abonnés uniques qui peuvent être contournés
- Dégradation de la performance qui est gérable
- Demandes d'amélioration
- Questions de documentation

Référence des Messages d'Erreur Courants

Erreurs d'Authentification

Message d'Erreur	Cause	Solution
"Échec de la génération des vecteurs d'authentification"	Ensemble de clés manquant ou invalide	Vérifier la configuration de l'ensemble de clés
"Échec de synchronisation SQN"	SQN hors synchronisation	Attendre la resynchronisation
"Abonné non trouvé"	IMSI invalide	Vérifier l'IMSI, provisionner l'abonné
"Abonné désactivé"	enabled=false	Activer l'abonné

Erreurs Diameter

Message d'Erreur	Cause	Solution
"Délai d'attente de connexion au pair Diameter"	Problème réseau	Vérifier la connectivité réseau
"Échange CER/CEA échoué"	Mismatch de configuration	Vérifier la configuration Diameter
"Application non supportée"	Le pair ne prend pas en charge l'application requise	Vérifier les applications du pair
"Échec de la poignée de main TLS"	Problème de certificat	Vérifier les certificats

Erreurs de Base de Données

Message d'Erreur	Cause	Solution
"Connexion refusée"	Base de données hors service	Démarrer la base de données
"Échec de l'authentification"	Mauvaises informations d'identification	Corriger les informations d'identification
"Aucune connexion disponible"	Pool épuisé	Augmenter la taille du pool
"D��lai d'attente de requête"	Requête lente	Optimiser les requêtes

Erreurs API

Message d'Erreur	Cause	Solution
"key_set_id n'existe pas"	Clé étrangère invalide	Créer l'ensemble de clés d'abord
"L'IMSI a déjà été pris"	IMSI dupliqué	Utiliser un IMSI différent ou supprimer l'existant
"Erreur de validation"	Entrée invalide	Vérifier le format et les exigences du champ

---

← Retour au Guide des Opérations | Suivant : Référence API →

Intégration des Webhooks OmniHSS

← Retour au Guide des Opérations

Table des Matières

- Aperçu
- Comment Fonctionnent les Webhooks
- Événements de Webhook
- Charge Utile de Webhook
- Configuration
- Cas d'Utilisation
- Considérations de Sécurité
- Dépannage

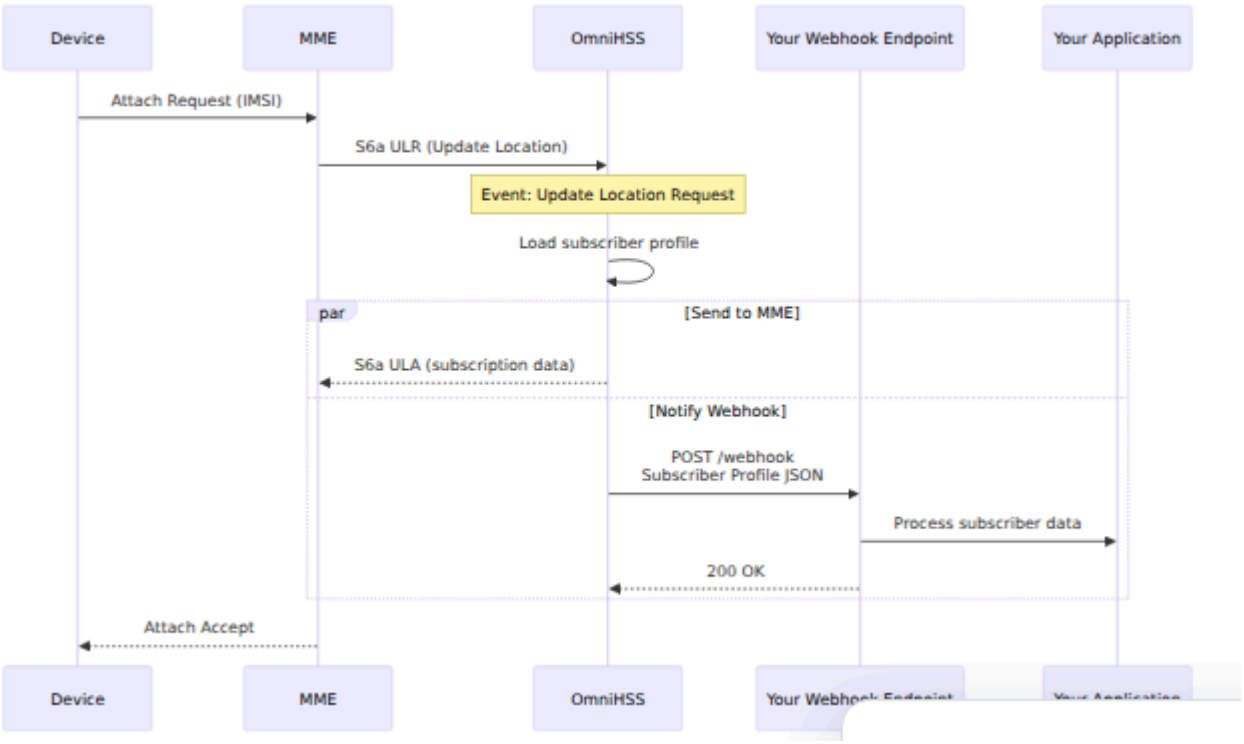
Aperçu

OmniHSS prend en charge les **webhooks** pour notifier les systèmes externes des événements des abonnés en temps réel. Lorsque des événements spécifiques se produisent (tels que des mises à jour de localisation, des demandes d'authentification ou des enregistrements IMS), OmniHSS peut envoyer une requête HTTP POST à votre point de terminaison de webhook configuré avec les données complètes du profil de l'abonné.

Qu'est-ce que les Webhooks ?

Les webhooks sont des rappels HTTP qui permettent à OmniHSS de pousser des notifications d'événements vers votre application au fur et à mesure qu'ils

se produisent, plutôt que d'exiger que votre application interroge l'API HSS pour des changements.



Avantages Clés

- **Notifications en temps réel** - Recevez des mises à jour instantanées lorsque des événements d'abonnés se produisent
- **Données complètes de l'abonné** - Chaque webhook inclut le profil complet de l'abonné (identique à `GET /api/subscriber`)
- **Automatisation basée sur les événements** - Déclenchez des flux de travail, des analyses ou des approvisionnements basés sur des événements réseau
- **Réduction des interrogations** - Pas besoin d'interroger continuellement l'API pour les changements de statut des abonnés
- **Flexibilité d'intégration** - Connectez OmniHSS à des systèmes de facturation, des plateformes d'analyse ou des applications personnalisées

---

Comment Fonctionnent les Webhooks

Flux d'Événements

1. **Un événement se produit** - Un abonné effectue une action (attachement, mise à jour de localisation, enregistrement IMS, etc.)
2. **HSS traite l'événement** - OmniHSS gère la requête/réponse Diameter normalement
3. **Webhook déclenché** - Si un webhook est enregistré pour ce type d'événement, HSS envoie un POST HTTP à votre point de terminaison
4. **Données de l'abonné incluses** - La charge utile du webhook contient le profil complet de l'abonné sous forme de JSON
5. **Votre application répond** - Votre point de terminaison doit renvoyer HTTP 200-299 pour accuser réception

Garanties de Livraison

- **Livraison en meilleur effort** - Les webhooks sont envoyés de manière asynchrone et ne bloquent pas les opérations réseau
- **Délai d'attente** - Les requêtes de webhook expirent après 5 secondes
- **Pas de nouvelles tentatives** - Si votre point de terminaison est indisponible ou renvoie une erreur, le webhook n'est pas réessayé
- **Ordre non garanti** - Les événements peuvent arriver dans le désordre sous une forte charge

Important : Les opérations réseau (authentification, mises à jour de localisation, etc.) **ne dépendent pas** de la livraison des webhooks. Si votre point de terminaison de webhook est hors ligne, le service des abonnés continue normalement.

Événements de Webhook

OmniHSS peut déclencher des webhooks pour les événements suivants :

Événements EPC/LTE

Événement	Déclencheur	Description
`update_location_request`	S6a ULR	L'abonné s'attache ou effectue une mise à jour de la zone de suivi
`authentication_information_request`	S6a AIR	Le réseau demande des vecteurs d'authentification pour l'abonné
`purge_request`	S6a PUR	MME supprime le contexte de l'abonné (appareil éteint, détaché)
`cancel_location_answer`	S6a CLA	MME accuse réception de la désinscription de l'abonné

Événements IMS

Événement	Déclencheur	Description
`ims_registration`	Cx SAR	L'abonné s'enregistre pour le service IMS/VoLTE
`ims_deregistration`	Cx SAR (de-reg)	L'abonné se désinscrit de l'IMS
`ims_profile_request`	Sh UDR	Le serveur d'application demande le profil IMS de l'abonné

Événements de Politique (PCRF)

Év��nement	Déclencheur	Description
`policy_request`	Gx CCR	P-GW demande une politique pour la session de données de l'abonné
`media_authorization`	Rx AAR	P-CSCF demande une autorisation de média pour un appel IMS

Événements Multi-IMSI

Événement	Déclencheur	Description
`imsi_switch`	ULR pour un IMSI différent sur la même SIM	L'appareil passe à un IMSI différent sur une SIM multi-IMSI

Charge Utile de Webhook

Format de Requête

Lorsqu'un événement se produit, OmniHSS envoie une requête HTTP POST à votre URL de webhook configurée :

```
POST /your-webhook-endpoint HTTP/1.1
Host: your-server.com
Content-Type: application/json
X-OmniHSS-Event: update_location_request
X-OmniHSS-Event-ID: 550e8400-e29b-41d4-a716-446655440000
X-OmniHSS-Timestamp: 2025-01-15T14:30:00Z

{
  "event": "update_location_request",
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "timestamp": "2025-01-15T14:30:00Z",
  "subscriber": {
    "id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "imsi": "001001123456789",
    "enabled": true,
    "ims_enabled": true,
    "msisdns": [
      {"id": "16fd2706-8baf-433b-82eb-8c7fada847da", "msisdn":
"14155551001"},
      {"id": "27ae3807-9cbf-4a4c-93fc-9d8afdb958eb", "msisdn":
"14155551002"}
    ],
    "sim": {
      "id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
      "iccid": "8991101200003204510",
      "is_esim": false
    },
    "key_set": {
      "id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "amf": "8000"
    },
    "epc_profile": {
      "id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
      "name": "Premium 100Mbps",
      "ue_ambr_dl_kbps": 100000,
      "ue_ambr_ul_kbps": 50000
    },
    "ims_profile": {
      "id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
      "name": "Standard VoLTE"
    },
    "roaming_profile": {
      "id": "9c858901-8a57-4791-81fe-4c455b099bc9",
```

```
      "name": "International Roaming Allowed"
    },
    "subscriber_state": {
      "mme_host": "mme-01.example.com",
      "mme_realm": "epc.mnc001.mcc001.3gppnetwork.org",
      "visited_plmn": "001001",
      "last_update": "2025-01-15T14:30:00Z"
    },
    "custom_attributes": {
      "account_type": "premium",
      "billing_plan": "unlimited"
    }
  },
  "event_context": {
    "visited_plmn": "310410",
    "mme_host": "mme-roaming.example.com",
    "location_update_type": "initial_attach"
  }
}
```

Structure de la Charge Utile

Champ	Type	Description
`event`	string	Type d'événement (par exemple, `update_location_request`)
`event_id`	string	UUID unique pour cette livraison de webhook
`timestamp`	string	Horodatage ISO 8601 lorsque l'événement s'est produit
`subscriber`	object	**Profil complet de l'abonné** (identique à `GET /api/subscriber/:id`)
`event_context`	object	Données contextuelles spécifiques à l'événement

Champs de Contexte d'Événement

L'objet `event_context` contient des informations spécifiques à l'événement :

Pour `update_location_request`:

```
{
  "visited_plmn": "310410",
  "mme_host": "mme-roaming.example.com",
  "mme_realm": "epc.mnc410.mcc310.3gppnetwork.org",
  "location_update_type": "initial_attach"
}
```

Pour `imsi_switch`:

```
{
  "previous_imsi": "001001111111111",
  "new_imsi": "310410222222222",
  "sim_id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
  "previous_mme_host": "mme-home.example.com",
  "new_mme_host": "mme-roaming.example.com"
}
```

Pour `ims_registration`:

```
{
  "scscf_host": "scscf-01.ims.example.com",
  "public_identities": [
    "sip:001001123456789@ims.mnc001.mcc001.3gppnetwork.org",
    "sip:+14155551001@ims.example.com",
    "tel:+14155551001"
  ]
}
```

En-têtes HTTP

En-tête	Description	Exemple
`Content-Type`	Toujours `application/json`	`application/json`
`X-OmniHSS-Event`	Type d'événement	`update_location_request`
`X-OmniHSS-Event-ID`	Identifiant unique de l'événement	UUID
`X-OmniHSS-Timestamp`	Horodatage de l'événement	Format ISO 8601
`User-Agent`	Version d'OmniHSS	`OmniHSS/1.0`

Configuration

Enregistrement des Webhooks

Les webhooks sont configurés via l'API OmniHSS.

Enregistrer un Webhook

```
curl -k -X POST https://hss.example.com:8443/api/webhook \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "url": "https://your-server.com/omnihss-webhook",
      "events": [
        "update_location_request",
        "ims_registration",
        "imsi_switch"
      ],
      "enabled": true,
      "description": "Webhook du système de facturation en
production"
    }
  }'
```

Réponse :

```
{
  "data": {
    "id": "4e5d6c7b-8a9f-4b0c-8d1e-2f3a4b5c6d7e",
    "url": "https://your-server.com/omnihss-webhook",
    "events": [
      "update_location_request",
      "ims_registration",
      "imsi_switch"
    ],
    "enabled": true,
    "description": "Webhook du système de facturation en
production",
    "created_at": "2025-01-15T14:00:00Z"
  }
}
```

Lister les Webhooks

```
curl -k https://hss.example.com:8443/api/webhook
```

Mettre à Jour un Webhook

```
curl -k -X PUT https://hss.example.com:8443/api/webhook/4e5d6c7b-
8a9f-4b0c-8d1e-2f3a4b5c6d7e \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "enabled": false
    }
  }'
```

Supprimer un Webhook

```
curl -k -X DELETE
https://hss.example.com:8443/api/webhook/4e5d6c7b-8a9f-4b0c-8d1e-
2f3a4b5c6d7e
```

Exigences du Point de Terminaison de Webhook

Votre point de terminaison de webhook doit :

1. **Accepter les requêtes POST** avec `Content-Type: application/json`
2. **Répondre rapidement** - Retourner HTTP 200-299 dans les 5 secondes
3. **Être idempotent** - Gérer les livraisons en double gracieusement
4. **Utiliser HTTPS** - Pour la sécurité, utilisez des points de terminaison TLS/SSL (recommandé)
5. **Valider les charges utiles** - Vérifier que la requête provient d'OmniHSS (voir section Sécurité)

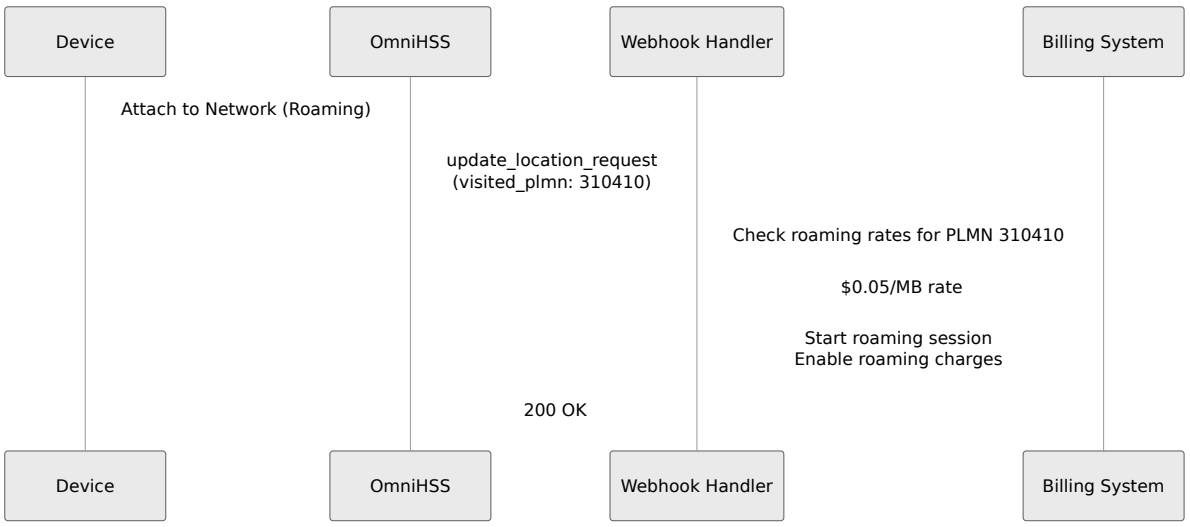
Exemple de Gestionnaire de Webhook (Node.js/Express) :

```
const express = require('express');
const app = express();

app.post('/omnihss-webhook', express.json(), (req, res) => {
  const { event, subscriber, event_context } = req.body;

  console.log(`Événement reçu : ${event}`);
  console.log(`IMSI de l'abonné : ${subscriber.imsi}`);

  // Traiter les données de l'abonné
  // ... votre logique métier ici ...

  // Répondre immédiatement pour accuser réception
  res.status(200).json({ received: true });

  // Gérer le traitement asynchrone après la réponse
  processWebhook(req.body).catch(console.error);
});

async function processWebhook(payload) {
  // Votre logique de traitement asynchrone
  // par exemple, mettre à jour le système de facturation,
déclencher des analyses, etc.
}

app.listen(3000);
```

Cas d'Utilisation

1. Suivi de Facturation et d'Utilisation en Temps Réel

Suivez l'utilisation du réseau par les abonnés et déclenchez des événements de facturation en temps réel.



Avantages :

- Détectez instantanément lorsque les abonnés se déplacent à l'international
- Appliquez des frais d'itinérance appropriés en temps réel
- Suivez avec précision les heures de début/fin de session
- Générez des alertes d'utilisation lorsque des seuils sont atteints

2. Analytique et Surveillance

Alimentez les données d'activité des abonnés dans des plateformes d'analyse pour des tableaux de bord et des rapports en temps réel.

Cas d'utilisation : Suivre les abonnés actifs par région

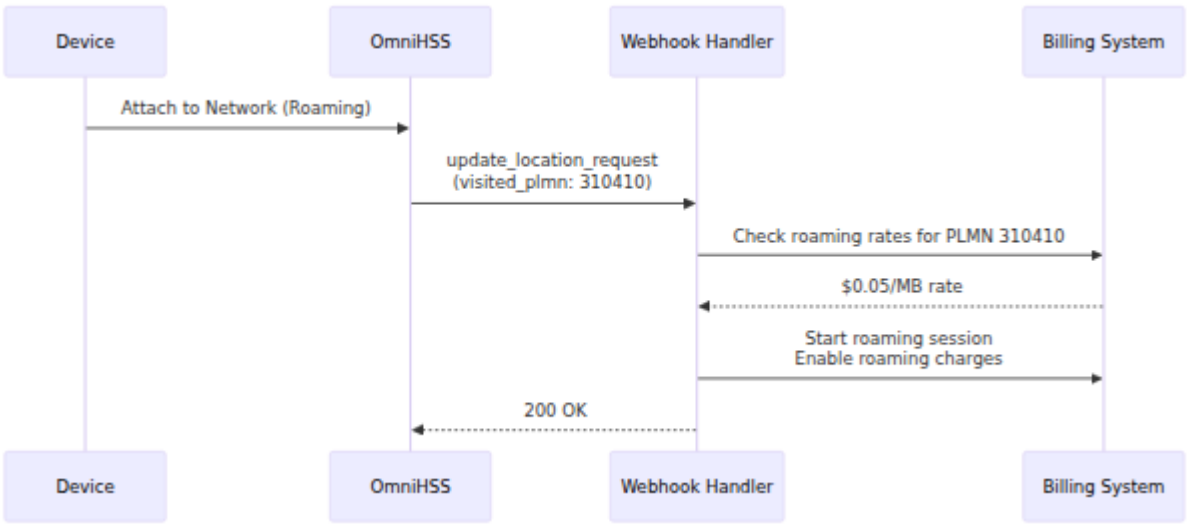
```
// Gestionnaire de webhook alimentant des données à la plateforme d'analyse
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber, event_context } = req.body;

  if (event === 'update_location_request') {
    await analytics.track({
      event: 'subscriber_location_update',
      imsi: subscriber.imsi,
      visited_plmn: event_context.visited_plmn,
      timestamp: req.body.timestamp,
      profile: subscriber.epc_profile.name
    });
  }

  res.status(200).send();
});
```

Tableau de Bord d'Analytique :

- Abonnés actifs par MME
- Abonnés en itinérance par pays
- Distribution des niveaux de service
- Taux de réussite des enregistrements IMS

3. Détection de Fraude et Sécurité

Détectez des modèles d'activité suspects en temps réel et déclenchez des réponses automatisées.



Scénarios de Détection de Fraude :

1. **Changements de Localisation Rapides**
    - L'abonné s'attache dans le pays A
    - 30 minutes plus tard, s'attache dans le pays B (physiquement impossible)
    - Action : Marquer le compte, envoyer une alerte à l'équipe de sécurité
2. **Abus de Changement d'IMSI**
    - Plusieurs changements rapides d'IMSI sur la même SIM
    - Possibilité de clonage de SIM ou d'utilisation non autorisée de multi-IMSI
    - Action : Désactiver tous les IMSIs sur la SIM, notifier l'équipe de fraude
3. **Itinérance Non Autorisée**
    - L'abonné se déplace vers un pays bloqué (sanctions, risque de fraude)
    - Action : Désactiver automatiquement l'abonné, bloquer l'accès au réseau

Exemple d'Implémentation :

```
@app.route('/omnihss-webhook', methods=['POST'])
def webhook_handler():
    data = request.json
    subscriber = data['subscriber']
    event_context = data.get('event_context', {})

    if data['event'] == 'update_location_request':
        visited_plmn = event_context.get('visited_plmn')

        # Vérifier les pays bloqués
        if visited_plmn in BLOCKED_PLMNS:
            disable_subscriber(subscriber['imsi'])
            alert_security_team(subscriber, 'Itinérance vers PLMN
bloqué')

        # Vérifier les voyages impossibles
        if is_impossible_travel(subscriber['imsi'], visited_plmn):
            flag_for_review(subscriber['imsi'])
            alert_fraud_team(subscriber, 'Voyage impossible
détecté')

    return jsonify({'status': 'ok'}), 200
```

4. Automatisation de l'Approvisionnement

Approvisionnez ou mettez à jour automatiquement les services des abonnés en fonction des événements réseau.

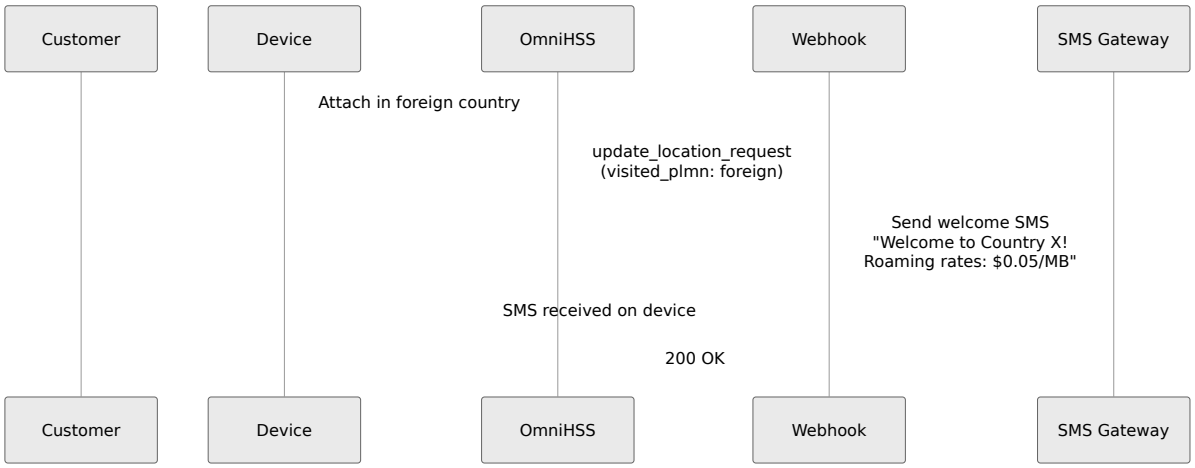
Cas d'utilisation : Activer automatiquement l'IMS lorsque l'abonné utilise pour la première fois VoLTE

```
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber } = req.body;

  if (event === 'ims_registration' && !subscriber.ims_enabled) {
    // Premier utilisateur IMS - activer l'IMS de manière
permanente
    await omnihss.updateSubscriber(subscriber.id, {
      ims_enabled: true,
      custom_attributes: {
        ...subscriber.custom_attributes,
        volte_activated_at: new Date().toISOString()
      }
    });

    // Mettre à jour le CRM
    await crm.updateCustomer(subscriber.imsi, {
      features: ['volte']
    });
  }

  res.status(200).send();
});
```

5. Notifications aux Clients

Envoyez des notifications en temps réel aux clients concernant leur service.

Cas d'utilisation : Message de bienvenue lors de l'itinérance à l'international



Exemples de Notifications :

- "Bienvenue à [Pays] ! Des frais d'itinérance s'appliquent."
- "Vous avez utilisé 80 % de votre allocation de données"
- "Le service VoLTE est maintenant actif sur votre appareil"
- "Votre compte a été mis à niveau vers Premium"

6. Gestion des SIM Multi-IMSI

Suivez et gérez les abonnés avec des SIM multi-IMSI, recevant des notifications lorsqu'ils changent d'IMSI.

```
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber, event_context } = req.body;

  if (event === 'imsi_switch') {
    const { previous_imsi, new_imsi, sim_id } = event_context;

    // Journaliser le changement d'IMSI pour l'analytique
    await db.logImsiSwitch({
      sim_id,
      from_imsi: previous_imsi,
      to_imsi: new_imsi,
      timestamp: req.body.timestamp
    });

    // Mettre à jour le système de facturation
    await billing.endSession(previous_imsi);
    await billing.startSession(new_imsi);

    // Alerter si changement excessif (risque de fraude)
    const switchCount = await db.getSwitchCount(sim_id, '24h');
    if (switchCount > 10) {
      await alertFraudTeam(`Changement excessif d'IMSI : SIM
${sim_id}`);
    }
  }

  res.status(200).send();
});
```

7. Intégration avec des Systèmes Externes

Connectez OmniHSS à des systèmes tiers sans interroger.

Exemples d'Intégrations :

- **Systèmes CRM** - Mettre à jour les dossiers clients avec l'utilisation des services
- **Surveillance Réseau** - Alimenter les données des abonnés dans des plateformes d'analyse réseau
- **Systèmes de Facturation** - Déclencher des frais basés sur des événements réseau
- **Systèmes de Billetterie** - Créer automatiquement des tickets pour des authentifications échouées
- **Entrepôts de Données** - Diffuser des événements d'abonnés pour une analyse de big data

---

Considérations de Sécurité

Secret/Signature de Webhook

Pour vérifier que les webhooks proviennent d'OmniHSS, implémentez la vérification de signature :

```
# Configurer le webhook avec un secret
curl -k -X POST https://hss.example.com:8443/api/webhook \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "url": "https://your-server.com/omnihss-webhook",
      "events": ["update_location_request"],
      "secret": "your-secret-key-here"
    }
  }'
```

OmniHSS inclura un en-tête `X-OmniHSS-Signature` :

```
X-OmniHSS-Signature:
sha256=5d7a8f9b2c1e3a4d6f7e8b9c0a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a
```

Vérifiez la signature :

```javascript
const crypto = require('crypto');

function verifyWebhook(req) {
  const signature = req.headers['x-omnihss-signature'];
  const secret = process.env.WEBHOOK_SECRET;
  const payload = JSON.stringify(req.body);

  const expectedSignature = 'sha256=' +
    crypto.createHmac('sha256', secret)
      .update(payload)
      .digest('hex');

  return crypto.timingSafeEqual(
    Buffer.from(signature),
    Buffer.from(expectedSignature)
  );
}

app.post('/omnihss-webhook', (req, res) => {
  if (!verifyWebhook(req)) {
    return res.status(401).json({ error: 'Signature invalide' });
  }

  // Traiter le webhook...
  res.status(200).send();
});
```

Meilleures Pratiques

1. **Utiliser HTTPS** - Utilisez toujours TLS pour les points de terminaison de webhook
2. **Valider les signatures** - Vérifiez les signatures des webhooks pour éviter le spoofing

3. **Limitation de taux** - Implémentez une limitation de taux sur les points de terminaison de webhook
4. **Liste blanche d'IP** - Restreignez l'accès au webhook aux adresses IP d'OmniHSS
5. **Surveiller les échecs** - Suivez les échecs de livraison de webhook et les erreurs
6. **Sanitiser les données** - Validez et nettoyez les charges utiles des webhooks avant traitement
7. **Sécuriser les identifiants** - Stockez les secrets de webhook dans une configuration sécurisée (variables d'environnement, gestionnaire de secrets)

Confidentialité des Données

Les charges utiles des webhooks contiennent des **informations sensibles sur les abonnés** :

- IMSI (identité de l'abonné)
- MSISDNs (numéros de téléphone)
- Données de localisation (PLMN visité, MME)
- Informations sur le profil de service

Exigences de Conformité :

- **RGPD** - Assurez-vous que les données des webhooks sont traitées en conformité avec le RGPD
- **Conservation des données** - Mettez en œuvre des politiques de conservation des données appropriées
- **Contrôle d'accès** - Restreignez l'accès au point de terminaison du webhook
- **Chiffrement** - Utilisez TLS pour le transport des webhooks
- **Journalisation d'audit** - Journalisez toutes les livraisons de webhooks pour la conformité

Dépannage

Webhook Non Reçu

Symptômes :

- Des événements se produisent mais le webhook n'est pas déclenché
- Le point de terminaison de webhook ne reçoit jamais de requêtes

Étapes de Dépannage :

1. **Vérifiez que le webhook est activé :**

   ```
   curl -k https://hss.example.com:8443/api/webhook
   # Vérifiez "enabled": true
   ```

2. **Vérifiez la configuration des événements du webhook :**

   - Assurez-vous que le type d'événement est inclus dans la liste `events` du webhook
   - Exemple : Si vous souhaitez des événements `ims_registration`, vérifiez qu'il est dans le tableau des événements

3. **Vérifier les journaux HSS :**

   - Vérifiez les erreurs de livraison de webhook
   - Recherchez des problèmes de connectivité réseau
   - Vérifiez qu'il n'y a pas d'échecs de résolution DNS

4. **Tester l'accessibilité de l'endpoint :**

   ```
   curl -X POST https://your-server.com/omnihss-webhook \
     -H "Content-Type: application/json" \
     -d '{"test": true}'
   ```

Délai d'attente du Webhook

Symptômes :

- Les journaux HSS montrent des erreurs de délai d'attente du webhook
- L'endpoint du webhook reçoit la demande mais HSS la marque comme échouée

Solution :

1. **Répondre immédiatement :**
   - Retourner HTTP 200 dans les 5 secondes
   - Traiter les données de manière asynchrone après avoir répondu
2. **Optimiser les performances de l'endpoint :**

```
// MAUVAIS - Traitement synchronisé lent
app.post('/webhook', (req, res) => {
  processData(req.body); // Bloque pendant 10 secondes
  res.status(200).send();
});

// BON - Traitement asynchrone après réponse
app.post('/webhook', (req, res) => {
  res.status(200).send(); // Répondre immédiatement
  processData(req.body); // Traitement asynchrone
});
```

Webhooks Dupliqués

Symptômes :

- Même événement livré plusieurs fois
- `event_id` est identique pour les livraisons dupliquées

Cause :

- Réessais réseau (bien qu'OmniHSS ne réessaie pas, l'infrastructure réseau pourrait le faire)
- Plusieurs webhooks enregistrés pour le même événement

Solution :

Implémenter l'idempotence en utilisant `event_id` :

```
const processedEvents = new Set();

app.post('/omnihss-webhook', (req, res) => {
  const eventId = req.body.event_id;

  if (processedEvents.has(eventId)) {
    // Déjà traité, ignorer
    return res.status(200).json({ status: 'duplicate' });
  }

  processedEvents.add(eventId);

  // Traiter le webhook...
  processWebhook(req.body);

  res.status(200).json({ status: 'processed' });
});
```

Le Webhook Renvoie une Erreur

Symptômes :

- L'endpoint renvoie HTTP 4xx ou 5xx
- Les journaux HSS montrent un échec de livraison du webhook

Erreurs Courantes :

1. **401 Non autorisé** - Échec de la vérification de la signature
    - Vérifiez que le secret du webhook correspond à la configuration
    - Vérifiez l'algorithme de calcul de la signature

2. **400 Mauvaise demande** - Charge utile invalide
    - Vérifiez l'analyse de la charge utile du webhook
    - Assurez-vous que l'en-tête Content-Type est géré
3. **500 Erreur interne du serveur** - L'endpoint a planté
    - Examinez les journaux d'erreurs de l'endpoint
    - Ajoutez une gestion des erreurs et des journaux

Solution :

Ajoutez une gestion des erreurs complète :

```
app.post('/omnihss-webhook', async (req, res) => {
  try {
    // Vérifier la signature
    if (!verifyWebhook(req)) {
      return res.status(401).json({ error: 'Invalid signature' });
    }

    // Valider la charge utile
    if (!req.body.event || !req.body.subscriber) {
      return res.status(400).json({ error: 'Invalid payload' });
    }

    // Traiter le webhook
    await processWebhook(req.body);

    res.status(200).json({ status: 'ok' });

  } catch (error) {
    console.error('Erreur de traitement du webhook :', error);
    // Retourner 200 pour éviter le réessai, enregistrer l'erreur
pour enquête
    res.status(200).json({ status: 'error', message: error.message
});
  }
});
```

Données de Souscripteur Manquantes

Symptômes :

- Webhook reçu mais l'objet souscripteur est incomplet
- Les champs attendus sont nuls ou manquants

Causes Possibles :

1. **Souscripteur pas complètement provisionné** - Certains profils peuvent être optionnels (IMS, itinérance)
2. **Condition de course de données** - Souscripteur mis à jour entre le déclenchement de l'événement et l'envoi du webhook

Solution :

Gérer les champs optionnels avec grâce :

```
const { subscriber } = req.body;

// Vérifier les champs optionnels
const imsProfile = subscriber.ims_profile || { name: 'No IMS' };
const roamingProfile = subscriber.roaming_profile || { name: 'No
Roaming' };

// Gérer les MSISDN manquants
const msisdns = subscriber.msisdns || [];
```

Surveillance et Observabilité

Métriques de Webhook

Suivez les performances et la fiabilité du webhook :

Métriques à Surveiller :

- Taux de livraison du webhook (réussi vs. échoué)

- Latence du webhook (temps de l'événement à la réponse de l'endpoint)
- Temps de réponse de l'endpoint
- Taux d'erreur par endpoint
- Événements par seconde

Exemple de Requête de Tableau de Bord (Prometheus/Grafana) :

```
# Taux de succès du webhook
rate(omnihss_webhook_success_total[5m]) /
rate(omnihss_webhook_attempts_total[5m])

# Latence du webhook
histogram_quantile(0.95, omnihss_webhook_duration_seconds)
```

Journaux de Webhook

Activez l'enregistrement détaillé des webhooks pour le dépannage :

Format de Journal :

```
{
  "timestamp": "2025-01-15T14:30:00Z",
  "level": "info",
  "component": "webhook",
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "webhook_id": "4e5d6c7b-8a9f-4b0c-8d1e-2f3a4b5c6d7e",
  "event_type": "update_location_request",
  "subscriber_imsi": "001001123456789",
  "endpoint": "https://your-server.com/omnihss-webhook",
  "http_status": 200,
  "duration_ms": 145,
  "error": null
}
```

← Retour au Guide des Opérations | Suivant : Référence API →