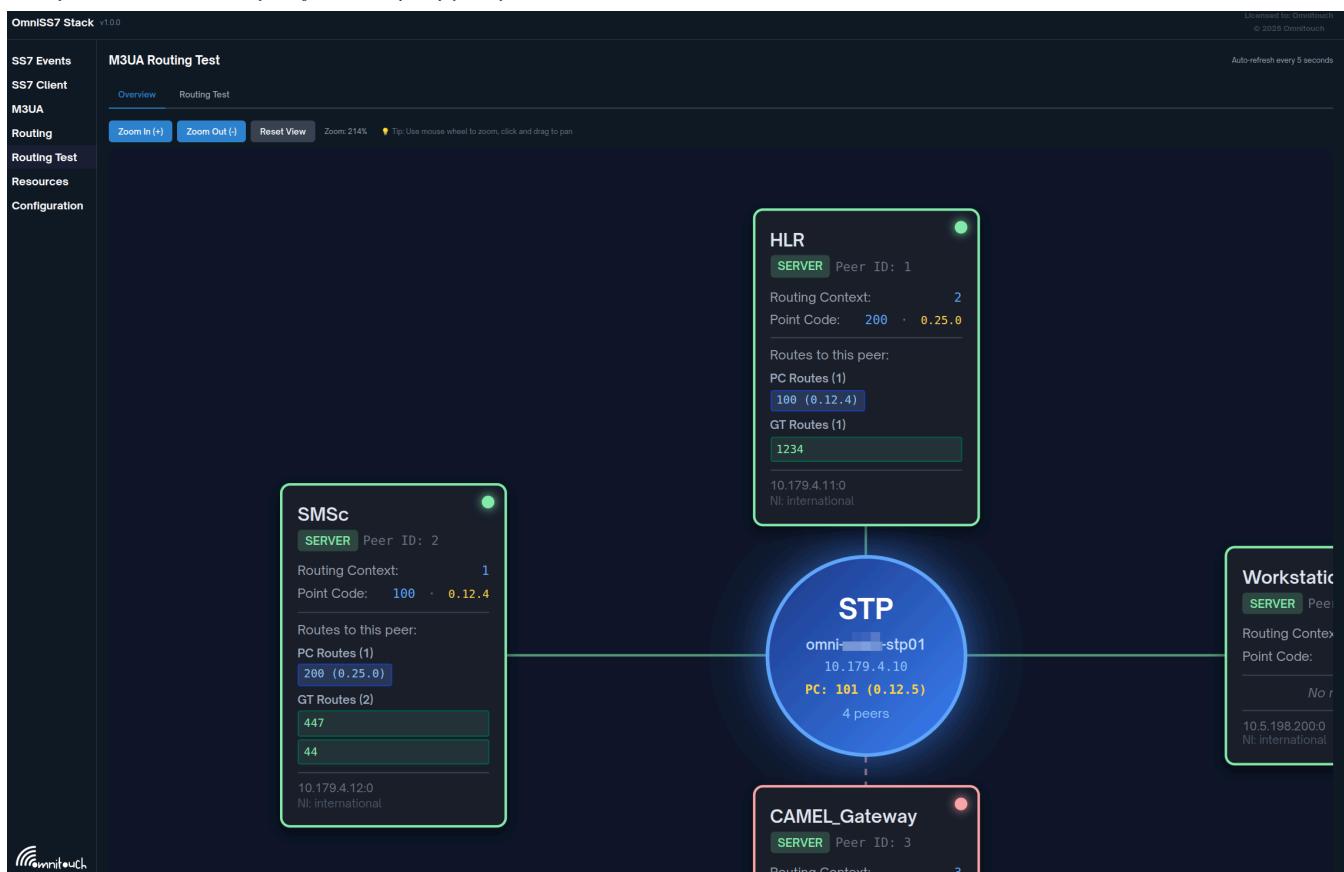


OmniSS7 - Guide de l'utilisateur

OmniSS7 par Omnitouch Network Services est une pile de signalisation SS7 complète et polyvalente qui offre une fonctionnalité flexible des éléments de réseau.



Aperçu de la documentation

Cette documentation est organisée par rôle d'élément de réseau. Choisissez le guide qui correspond à votre déploiement :

❖ Guides de configuration

- [Guide STP](#) - Configuration du Point de Transfert de Signal
 - Routage du trafic SS7 entre les paires de réseau
 - Routage par code de point et tiré global
 - Équilibrage de charge et dissimulation de topologie
 - Utilisez ceci si vous routez le trafic SS7 entre les réseaux
- [Guide Client MAP](#) - Configuration du Client MAP
 - Connectez-vous en tant que client MGUA pour envoyer des requêtes MAP
 - Requêtes HLR, authentication, informations de routage
 - Support du protocole MAP générique
 - Utilisez ceci si vous envoyez des requêtes MAP aux éléments de réseau
- [Guide Centre SMS](#) - Configuration du Centre SMS (SMSc)
 - Routage et livraison de messages SMS
 - Mise en file d'attente de messages avec base de données
 - Auto-flush et rapports de livraison
 - Utilisez ceci si vous exploitez un Centre SMS
- [Guide HLR](#) - Configuration du Répertoire de Localisation Abonné
 - Gestion de la base de données des abonnés
 - Génération de vecteurs d'authentification
 - Mises à jour de localisation et informations de routage
 - Utilisez ceci si vous exploitez un HLR/HSS
- [Guide Passerelle CAMEL](#) - Configuration de la Passerelle CAMEL
 - Services de réseau intelligent (CAMEL)
 - Contrôle d'appel en temps réel et facturation
 - Intégration OCS pour la facturation
 - Générateur de requêtes interactif et surveillance de session
 - Utilisez ceci si vous fournissez des services IN ou une facturation en temps réel

❖ Fonctionnalités communes

- [Guide des fonctionnalités communes](#) - Composants partagés
 - Aperçu et configuration de l'interface utilisateur Web
 - Documentation API
 - Surveillance et métriques (Prometheus)
 - Meilleures pratiques et dépannage

❖ Documentation de référence

- [Annexe](#) - Référence technique
 - Spécifications du protocole SS7
 - Codes d'opération MAP
 - Flux de transactions TCAP
 - Encodages et formats de caractères

Démarrage rapide

1. Aperçu du système

OmniSS7 peut fonctionner en différents modes selon vos besoins réseau :

3. Configuration

OmniSS7 peut fonctionner en 5 modes opérationnels différents. Le fichier de configuration config/runtime.exe contient des exemples complets et prêts à l'emploi.

Pour changer de mode :

1. Ouvrez config/runtime.exe
2. Découvrez la section de configuration souhaitée (STP, HLR, SMSc ou CAMEL GW)
3. Commentez les autres sections
4. Mettez à jour les adresses IP et les URL API si nécessaire
5. Redémarrez l'application

→ Voir les guides spécifiques au mode ci-dessous pour des instructions de configuration complètes

Exemples de configurations dans runtime.exs :

Mode STP :

```
config :omnis7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: false,
  camelgw_mode_enabled: false,
  map_client_m3ua: %{...}
```

Mode HLR :

```
config :omnis7,
  map_client_enabled: false,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,
  camelgw_mode_enabled: false,
  hlr_api_base_url: "...",
  map_client_m3ua: %{...}
```

Mode SMSC :

```
config :omnis7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,
  camelgw_mode_enabled: false,
  smsc_api_base_url: "...",
  auto_flush_enabled: true,
  map_client_m3ua: %{...}
```

Mode Passerelle CAMEL :

```
config :omnis7,
  map_client_enabled: true,
  camelgw_mode_enabled: true,
  ocs_enabled: true,
  ocs_url: "http://your-ocs-server/api/charging",
  cap_version: :v2, # Version CAP : :v1, :v2, :v3, ou :v4
  map_client_m3ua: %{...}
```

4. Accéder à l'interface Web

Naviguez vers <http://localhost> (ou votre nom d'hôte configuré)

Architecture du système

Matrice des fonctionnalités

Fonctionnalité	Mode STP	Client MAP	Mode SMSC	Mode HLR	Passerelle CAMEL
Routage par code de point	◊	◊	◊	◊	◊
Routage par titre global	◊	◊	◊	◊	◊
Réécriture du SSN	◊	◊	◊	◊	◊
Suppression multi-pair	◊	◊	◊	◊	◊
Requête MAP (envoyer)	◊	◊	◊	◊	◊
Réponses MAP (Recevoir)	◊	◊	◊	◊	◊
Gestion de file d'attente SMS	◊	◊	◊	◊	◊
Auto-flush SMS	◊	◊	◊	◊	◊
Base de données des abonnés	◊	◊	◊	◊	◊
Vecteurs d'authentification	◊	◊	◊	◊	◊
Mises à jour de localisation	◊	◊	◊	◊	◊
Suppression CAMEL	◊	◊	◊	◊	◊
Facturation en temps réel	◊	◊	◊	◊	◊
Contrôle d'appel (Services IN)	◊	◊	◊	◊	◊
Interface Web	◊	◊	◊	◊	◊
API REST	◊	◊	◊	◊	◊
Métriques Prometheus	◊	◊	◊	◊	◊

Opérations courantes

Accès à l'interface Web

- URL : <http://localhost> (ou nom d'hôte configuré)
- API Swagger : <http://localhost/swagger>
- Métriques : <http://localhost/metrics>

Surveillance

```
# Vérifier l'état des pairs M3UA
curl http://localhost/api/m3ua-status

# Voir les métriques Prometheus
curl http://localhost/metrics

# Vérifier la santé de l'application
curl http://localhost/api/health
```

Journaux

```
# Configurer le niveau de journalisation dans config/runtime.exs
config :logger,
  level: :debug # Options : :debug, :info, :warning, :error
```

Capacités clés

- Support complet du protocole MAP - Opérations MAP Phase 2/3/4
- Support du protocole CAP/CAMEL - CAP v1/v2/v3/v4 pour des services de réseau intelligent
- Signalisation M3UA/SCCP - Transmission SS7 et M3UA
- Routage en temps réel - Utilisation OCS pour la facturation prépayée/postpayée
- File d'attente de messages en temps réel - Livraison de SMS avec base de données
- Générateur de requêtes interactif - Interface Web pour les tests CAMEL/CAP
- Surveillance de session - Suivi des sessions d'appel CAMEL en temps réel
- Documentation API interactive - Interface Swagger pour les tests
- Métriques Prometheus - Observabilité complète
- Configuration multi-rôle - STP, Client MAP, SMSC, HLR, Passerelle CAMEL

Aperçu de la pile de protocoles

Exemples de cas d'utilisation

Passerelle réseau (STP)

Routage du trafic SS7 entre différents réseaux mobiles

- Connecter les réseaux d'opérateurs
- Passerelle M3UA/SCCP
- Équilibrage de charge entre HLR
- Traduction de titre global
- NAT SCCP (réutilisation intelligente de titre global)
- → [Guide STP](#)

Centre SMS (SMSC)

Livrer des messages SMS aux abonnés mobiles

- Livraison MT-SMS
- Origine MO-SMS
- Routage domicile SMS
- Masquage IMSI
- Partage de titre
- Gestion de file d'attente de messages
- Rapports de livraison
- Intégré dans OmniMessage pour gérer tous les SMS MAP
- → [Guide Centre SMS](#)

Client MAP

Interagir avec n'importe quel élément de réseau via MAP en utilisant une API RESTful simple

- PRN / SRI / ATI / etc
- Construisez vos propres applications SS7/MAP en utilisant des API RESTful
- Passerelles USSD
- Requête de vecteur d'authentification
- Recherche d'IMSI/MSISDN
- Requête d'informations de routage
- → [Guide Client MAP](#)

Base de données des abonnés (HLR)

Gérer les données des abonnés et l'authentification

- Mises à jour de localisation
- Génération d'authentification
- Provisionnement d'informations de routage
- Intégration complète dans OmniISS
- → [Guide HLR](#)

Plateforme de réseau intelligent (Passerelle CAMEL)

Contrôle d'appel en temps réel et facturation pour les opérateurs de télécommunications

- Facturation d'appels prépayée/postpayée
- Contrôle d'appel (cotation, délivrance, routage)
- Gestion de sessions et génération de CDR
- Génération de requêtes interactif pour les tests
- → [Guide Passerelle CAMEL](#)

Support et ressources

Documentation

Guides de configuration de base :

- [Guide de configuration STP](#) - Routage du Point de Transfer de Signal
- [Guide de configuration Client MAP](#) - Client du protocole MAP
- [Guide de configuration Centre SMS](#) - Routage et livraison de SMS
- [Guide de configuration HLR](#) - Base de données des abonnés
- [Guide de configuration Passerelle CAMEL](#) - Réseau intelligent & facturation

Intégration & Référence :

- [Guide du générateur de requêtes CAMEL](#) - Outil de test interactif
- [Guide des fonctionnalités communes](#) - Composants partagés & Interface Web
- [Référence technique](#) - Spécifications du protocole

Informations de contact

Produit : OmniSS7

Fabricant : Omnitouch Network Services

Version de la documentation : 2.0

Dernière mise à jour : 2025

Pour un support technique, une assistance à l'implémentation ou des demandes de vente, veuillez contacter Omnitouch Network Services.

Cette documentation couvre l'opération en temps réel d'OmniSS7 et la fonctionnalité utilisateur finale. Pour l'installation, le développement ou la configuration avancée, veuillez vous référer à la documentation technique.



Guide de l'API REST

[← Retour à la documentation principale](#)

Ce guide fournit une documentation complète pour l'**API REST** d'OmniSS7 et l'**interface Swagger**.

Table des matières

- [1. Aperçu](#)
 - [2. Configuration du serveur HTTP](#)
 - [3. Swagger UI](#)
 - [4. Points de terminaison de l'API](#)
 - [5. Authentification](#)
 - [6. Formats de réponse](#)
 - [7. Gestion des erreurs](#)
 - [8. Métriques \(Prometheus\)](#)
 - [9. Exemples de requêtes](#)
-

Aperçu

OmniSS7 fournit une API REST pour un accès programmatique aux opérations MAP (Mobile Application Part). L'API vous permet de :

- Envoyer des requêtes MAP (SRI, SRI-for-SM, UpdateLocation, etc.)
- Récupérer des réponses MAP
- Surveiller les métriques système via Prometheus

Architecture de l'API

Configuration du serveur HTTP

Détails du serveur

Paramètre	Valeur	Configurable
Protocole	HTTP	Non
Adresse IP	0.0.0.0 (toutes les interfaces)	Via code uniquement
Port	8080	Via code uniquement
Transport	Plug.Cowboy	Non

URL d'accès : `http://[server-ip]:8080`

Activation/Désactivation du serveur HTTP

Contrôlez si le serveur HTTP démarre :

```
config :omniss7,  
  start_http_server: true  # Mettre à false pour désactiver
```

Par défaut : `true` (activé)

Lorsqu'il est désactivé : Le serveur HTTP ne démarrera pas, et l'API REST/l'interface Swagger ne sera pas disponible.

Swagger UI

L'API comprend une **Swagger UI** pour la documentation interactive de l'API et les tests.

Accéder à Swagger UI

URL : `http://[server-ip]:8080/swagger`

Fonctionnalités :

- Documentation interactive de l'API
- Fonctionnalité d'essai pour tester les points de terminaison
- Schémas de requête/réponse
- Charges utiles d'exemple

Swagger JSON

La spécification OpenAPI est disponible à :

URL : `http://[server-ip]:8080/swagger.json`

Cas d'utilisation :

- Importer dans Postman ou d'autres clients API
 - Générer des bibliothèques clientes
 - Automatisation de la documentation de l'API
-

Points de terminaison de l'API

Tous les points de terminaison des opérations MAP suivent le modèle : `POST`

/api/{opération}

Résumé des points de terminaison

Point de terminaison	Méthode	But	Délai d'attente
/api/sri	POST	Envoyer des informations de routage	10s
/api/sri-for-sm	POST	Envoyer des informations de routage pour SM	10s
/api/send-auth-info	POST	Envoyer des informations d'authentification	10s
/api/MT-forwardSM	POST	SMS Mobile Terminé Forward SM	10s
/api/forwardSM	POST	Forward SM	10s
/api/updateLocation	POST	Mettre à jour la localisation	10s
/api/prn	POST	Fournir un numéro de roaming	10s
/metrics	GET	Métriques Prometheus	N/A
/swagger	GET	Swagger UI	N/A
/swagger.json	GET	Spécification OpenAPI	N/A

Remarque : Toutes les requêtes MAP ont un **délai d'attente de 10 secondes codé en dur**.

SendRoutingInfo (SRI)

Récupérer des informations de routage pour établir un appel à un abonné mobile.

Point de terminaison : POST /api/sri

Corps de la requête :

```
{  
  "msisdn": "1234567890",  
  "gmsc": "5551234567"  
}
```

Paramètres :

Champ	Type	Requis	Description
msisdn	String	Oui	MSISDN de la partie appelée
gmsc	String	Oui	Titre global du MSC de passerelle

Réponse (200 OK) :

```
{  
  "result": {  
    "imsi": "001001234567890",  
    "msrn": "5551234999",  
    "vlr_number": "5551234800",  
    ...  
  }  
}
```

Erreur (504 Gateway Timeout) :

```
{  
  "error": "timeout"  
}
```

Exemple cURL :

```
curl -X POST http://localhost:8080/api/sri \  
  -H "Content-Type: application/json" \  
  -d '{  
    "msisdn": "1234567890",  
    "gmsc": "5551234567"  
  }'
```

SendRoutingInfoForSM (SRI-for-SM)

Récupérer des informations de routage pour livrer un SMS à un abonné mobile.

Point de terminaison : POST /api/sri-for-sm

Corps de la requête :

```
{  
  "msisdn": "1234567890",  
  "service_center": "5551234567"  
}
```

Paramètres :

Champ	Type Requis	Description	
msisdn	String	Oui	MSISDN de destination
service_center	String	Oui	Titre global du centre de services

Réponse (200 OK) :

```
{  
  "result": {
```

```

    "imsi": "001001234567890",
    "msc_number": "5551234800",
    "location_info": {...},
    ...
}
}

```

Exemple cURL :

```
curl -X POST http://localhost:8080/api/sri-for-sm \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "service_center": "5551234567"
}'
```

SendAuthenticationInfo

Demander des vecteurs d'authentification pour un abonné.

Point de terminaison : POST /api/send-auth-info

Corps de la requête :

```
{
  "imsi": "001001234567890",
  "vectors": 3
}
```

Paramètres :

Champ	Type	Requis	Description
imsi	String	Oui	IMSI de l'abonné
vectors	Integer	Oui	Nombre de vecteurs d'authentification à générer

Réponse (200 OK) :

```
{
  "result": {
    "authentication_sets": [
      {
        "rand": "0123456789ABCDEF...",
        "xres": "...",
        "ck": "...",
        "ik": "...",
        "autn": "..."
      }
    ]
  }
}
```

```
  ],
  ...
}
```

Exemple cURL :

```
curl -X POST http://localhost:8080/api/send-auth-info \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "vectors": 3
}'
```

MT-ForwardSM

Livrer un SMS Mobile Terminé à un abonné.

Point de terminaison : POST /api/MT-forwardSM

Corps de la requête :

```
{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}
```

Paramètres :

Champ	Type Requis	Description
imsi	String Oui	IMSI de l'abonné de destination
destination_service_centre	String Oui	GT du centre de services de destination
originating_service_center	String Oui	GT du centre de services d'origine
smsPDU	String Oui	SMS TPDU au format hexadécimal

Remarque : smsPDU doit être une chaîne encodée en hexadécimal (majuscule ou minuscule).

Réponse (200 OK) :

```
{
  "result": {
    "delivery_status": "success",
    ...
}
```

```
    }  
}
```

Exemple cURL :

```
curl -X POST http://localhost:8080/api/MT-forwardSM \  
-H "Content-Type: application/json" \  
-d '{  
    "imsi": "001001234567890",  
    "destination_service_centre": "5551234567",  
    "originating_service_center": "5551234568",  
    "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"  
}'
```

ForwardSM

Transférer un message SMS (MO-SMS de l'abonné).

Point de terminaison : POST /api/forwardSM

Corps de la requête : Identique à MT-ForwardSM

Exemple cURL :

```
curl -X POST http://localhost:8080/api/forwardSM \  
-H "Content-Type: application/json" \  
-d '{  
    "imsi": "001001234567890",  
    "destination_service_centre": "5551234567",  
    "originating_service_center": "5551234568",  
    "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"  
}'
```

UpdateLocation

Notifier le HLR du changement de localisation de l'abonné (enregistrement VLR).

Point de terminaison : POST /api/updateLocation

Corps de la requête :

```
{  
    "imsi": "001001234567890",  
    "vlr": "5551234800"  
}
```

Paramètres :

Champ	Type	Requis	Description
imsi	String	Oui	IMSI de l'abonné
vlr	String	Oui	Adresse du titré global du VLR

Réponse (200 OK) :

```
{  
  "result": {  
    "hlr_number": "5551234567",  
    "subscriber_data": {...},  
    ...  
  }  
}
```

Remarque : En mode HLR, cela déclenche la séquence InsertSubscriberData (ISD) avec un délai d'attente de 10 secondes par ISD.

Exemple cURL :

```
curl -X POST http://localhost:8080/api/updateLocation \  
  -H "Content-Type: application/json" \  
  -d '{  
    "imsi": "001001234567890",  
    "vlr": "5551234800"  
  }'
```

ProvideRoamingNumber (PRN)

Demander le MSRN (Mobile Station Roaming Number) pour le routage des appels vers un abonné en itinérance.

Point de terminaison : POST /api/prn

Corps de la requête :

```
{  
  "msisdn": "1234567890",  
  "gmsc": "5551234567",  
  "msc_number": "5551234800",  
  "imsi": "001001234567890"  
}
```

Paramètres :

Champ	Type Requis	Description
msisdn	String Oui	MSISDN de l'abonné
gmsc	String Oui	GT du MSC de passerelle
msc_number	String Oui	Numéro MSC pour l'abonné
imsi	String Oui	IMSI de l'abonné

Réponse (200 OK) :

```
{
  "result": {
    "msrn": "5551234999",
    ...
  }
}
```

Exemple cURL :

```
curl -X POST http://localhost:8080/api/prn \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "gmsc": "5551234567",
  "msc_number": "5551234800",
  "imsi": "001001234567890"
}'
```

Authentification

État actuel : L'API ne nécessite pas d'authentification.

Considérations de sécurité :

- L'API est destinée à un usage sur un réseau interne/de confiance
- Envisagez d'utiliser des règles de pare-feu pour restreindre l'accès
- Pour les déploiements en production, envisagez de mettre en œuvre un middleware d'authentification

Formats de réponse

Toutes les réponses utilisent le format **JSON**.

Réponse de succès

Statut HTTP : 200 OK

Structure :

```
{  
  "result": {  
    // Données de réponse spécifiques à l'opération  
  }  
}
```

Réponse d'erreur

Statut HTTP :

- 400 Bad Request - Corps de requête invalide
- 504 Gateway Timeout - Délai d'attente de la requête MAP (10 secondes)
- 404 Not Found - Point de terminaison invalide

Structure :

```
{  
  "error": "timeout"  
}
```

ou

```
{  
  "error": "invalid request"  
}
```

Gestion des erreurs

Erreurs courantes

Erreur	Code HTTP	Description	Solution
JSON invalide	400	Le corps de la requête n'est pas un JSON valide	Vérifiez la syntaxe JSON
Champs manquants	400	Champs requis manquants	Incluez tous les paramètres requis
Délai d'attente	504	La requête MAP a dépassé le délai d'attente de 10s	Vérifiez la connectivité M3UA, la disponibilité HLR/VLR
Non trouvé	404	Point de terminaison invalide	Vérifiez l'URL du point de terminaison

Comportement de délai d'attente

Toutes les requêtes MAP ont un **délai d'attente de 10 secondes codé en dur** :

1. Requête envoyée au MapClient GenServer
2. Attend une réponse jusqu'à 10 secondes
3. Si aucune réponse → retourne 504 Gateway Timeout
4. Si réponse reçue → retourne 200 OK avec résultat

Dépannage des délais d'attente :

- Vérifiez l'état de la connexion M3UA (Web UI → page M3UA)
- Vérifiez que l'élément réseau (HLR/VLR/MSC) est accessible
- Vérifiez la configuration de routage
- Consultez les journaux d'événements SS7 pour les erreurs

Métriques (Prometheus)

L'API expose des métriques Prometheus pour la surveillance.

Point de terminaison des métriques

URL : `http://[server-ip]:8080/metrics`

Format : Format texte Prometheus

Exemple de sortie :

```
# HELP map_requests_total Total MAP requests
# TYPE map_requests_total counter
map_requests_total{operation="sri"} 42
map_requests_total{operation="sri_for_sm"} 158
map_requests_total{operation="updateLocation"} 23

# HELP cap_requests_total Total CAP requests
# TYPE cap_requests_total counter
cap_requests_total{operation="initialDP"} 87
cap_requests_total{operation="requestReportBCSMEvent"} 91

# HELP map_request_duration_milliseconds Duration of MAP request/
responses in ms
# TYPE map_request_duration_milliseconds histogram
map_request_duration_milliseconds_bucket{operation="sri",le="10"} 5
map_request_duration_milliseconds_bucket{operation="sri",le="50"} 12
map_request_duration_milliseconds_bucket{operation="sri",le="100"} 35
...
.

# HELP map_pending_requests Number of pending MAP TID waiters
# TYPE map_pending_requests gauge
map_pending_requests 3
```

Métriques disponibles

Métrique	Type	Étiquettes	Description
map_requests_total	Compteur	operation	Nombre total de requêtes MAP par type d'opération
cap_requests_total	Compteur	operation	Nombre total de requêtes CAP par type d'opération
map_request_duration_milliseconds	Histogramme	operation	Durée de la requête en millisecondes
map_pending_requests	Jauge	-	Nombre de transactions MAP en attente

Configuration de Prometheus

Ajoutez à votre `prometheus.yml` :

```
scrape_configs:
  - job_name: 'omniss7'
    static_configs:
      - targets: ['server-ip:8080']
    metrics_path: '/metrics'
    scrape_interval: 15s
```

Exemples de requêtes

Exemple Python

```
import requests
import json

# Requête SRI-for-SM
url = "http://localhost:8080/api/sri-for-sm"
payload = {
    "msisdn": "1234567890",
    "service_center": "5551234567"
}

response = requests.post(url, json=payload, timeout=15)

if response.status_code == 200:
```

```

        result = response.json()
        print(f"Succès : {result}")
    elif response.status_code == 504:
        print("Délai d'attente - aucune réponse du réseau")
    else:
        print(f"Erreur : {response.status_code} - {response.text}")

```

Exemple JavaScript

```

const axios = require('axios');

async function sendSRI() {
    try {
        const response = await axios.post('http://localhost:8080/api/
sri', {
            msisdn: '1234567890',
            gmsc: '5551234567'
        }, {
            timeout: 15000
        });

        console.log('Succès :', response.data);
    } catch (error) {
        if (error.code === 'ECONNABORTED') {
            console.error('Délai d\'attente - aucune réponse du réseau');
        } else {
            console.error('Erreur :', error.response?.data ||
error.message);
        }
    }
}

sendSRI();

```

Exemple Bash/cURL

```

#!/bin/bash

# Requête UpdateLocation
response=$(curl -s -w "\n%{http_code}" -X POST http://localhost:8080/
api/updateLocation \
-H "Content-Type: application/json" \
-d '{
    "imsi": "001001234567890",
    "vlr": "5551234800"
}')

http_code=$(echo "$response" | tail -n 1)

```

```
body=$(echo "$response" | sed '$d')

if [ "$http_code" -eq 200 ]; then
  echo "Succès : $body"
elif [ "$http_code" -eq 504 ]; then
  echo "Délai d'attente - aucune réponse du réseau"
else
  echo "Erreur $http_code : $body"
fi
```

Diagrammes de flux

Flux de requête API

Résumé

L'API REST d'OmniSS7 fournit :

- ◊ **Opérations MAP** - Support complet pour SRI, SRI-for-SM, UpdateLocation, livraison SMS, authentification
- ◊ **Swagger UI** - Documentation interactive de l'API et tests
- ◊ **Métriques Prometheus** - Surveillance et observabilité
- ◊ **Délai d'attente codé en dur** - Délai d'attente de 10 secondes pour toutes les requêtes MAP
- ◊ **Serveur HTTP** - Fonctionne sur le port 8080 (configurable via `start_http_server`)

Pour l'accès à l'interface Web, consultez le [Guide de l'interface Web](#).

Pour les détails de configuration, consultez la [Référence de configuration](#).



Référence Technique (Annexe)

[← Retour à la Documentation Principale](#)

Référence technique pour les protocoles SS7 et l'implémentation d'OmniSS7.

Pile de Protocoles SS7

Codes d'Opération MAP

Opération	Opcode	But
updateLocation	2	Enregistrer la localisation de l'abonné
cancelLocation	3	Désenregistrer du VLR
provideRoamingNumber	4	Demander MSRN
sendRoutingInfo	22	Interroger le routage des appels
mt-forwardSM	44	Livrer un SMS à l'abonné
sendRoutingInfoForSM	45	Interroger le routage des SMS
mo-forwardSM	46	Transférer un SMS de l'abonné
sendAuthenticationInfo	56	Demander des vecteurs d'authentification

Types de Messages TCAP

- **BEGIN** - Démarrer la transaction
 - **CONTINUE** - Milieu de transaction
 - **END** - Réponse finale
 - **ABORT** - Annuler la transaction
-

Addressage SCCP

Formats de Titre Global

- **E.164** - Numéro de téléphone international (par exemple, 447712345678)
- **E.212** - Format IMSI (par exemple, 234509876543210)
- **E.214** - Format de code de point

Numéros de Sous-Système (SSN)

- **SSN 6**: HLR

- **SSN 7:** VLR
 - **SSN 8:** MSC/SMSC
 - **SSN 9:** GMLC
 - **SSN 10:** SGSN
-

TPDU SMS

Types de Messages

- **SMS-DELIVER** (MT) - Réseau vers mobile
- **SMS-SUBMIT** (MO) - Mobile vers réseau
- **SMS-STATUS-REPORT** - Statut de livraison
- **SMS-COMMAND** - Commande à distance

Encodages de Caractères

- **GSM7** - Alphabet GSM 7 bits (160 caractères par SMS)
 - **UCS2** - Unicode 16 bits (70 caractères par SMS)
 - **8-bit** - Données binaires (140 octets par SMS)
-

États M3UA

- **DOWN** - Pas de connexion SCTP
 - **CONNECTING** - Connexion SCTP en cours
 - **ASPUP_SENT** - En attente de l'ACK ASPUP
 - **INACTIVE** - ASP actif mais non opérationnel
 - **ASPAC_SENT** - En attente de l'ACK ASPAC
 - **ACTIVE** - Prêt pour le trafic
-

Codes de Point SS7 Communs

Les codes de point sont généralement des valeurs de 14 bits (UIT) ou de 24 bits (ANSI).

Format d'Exemple (UIT) :

- Réseau : 3 bits
 - Cluster : 8 bits
 - Membre : 3 bits
-

Codes d'Erreur SCCP

- **0** - Pas de traduction pour l'adresse
 - **1** - Pas de traduction pour une adresse spécifique
 - **2** - Congestion du sous-système
 - **3** - Échec du sous-système
 - **4** - Utilisateur non équipé
 - **5** - Échec MTP
 - **6** - Congestion du réseau
 - **7** - Non qualifié
 - **8** - Erreur dans le transport du message
-

Codes d'Erreur MAP

Code	Erreur	Description
1	unknownSubscriber	Abonné non présent dans le HLR
27	absentSubscriber	Abonné non joignable
34	systemFailure	Échec du réseau
35	dataMissing	Données requises non disponibles
36	unexpectedDataValue	Valeur de paramètre invalide

Documentation Connexe

- [← Retour à la Documentation Principale](#)
 - [Guide STP](#)
 - [Guide Client MAP](#)
 - [Guide Centre SMS](#)
 - [Guide HLR](#)
 - [Fonctionnalités Communes](#)
-

Guide de Configuration du Passerelle CAMEL

Vue d'ensemble

Le mode **Passerelle CAMEL (CAMELGW)** transforme OmniSS7 en une plateforme de Réseau Intelligent (IN) qui fournit des services de contrôle d'appel et de facturation en temps réel en utilisant le protocole CAMEL Application Part (CAP).

Qu'est-ce que CAMEL ?

CAMEL (Applications Personnalisées pour la Logique Améliorée des Réseaux Mobiles) est un ensemble de normes conçues pour fonctionner sur un réseau de cœur GSM ou UMTS. Il permet aux opérateurs de fournir des services nécessitant un contrôle en temps réel des appels, tels que :

- **Appels prépayés** - Vérification et facturation en temps réel du solde
- **Services à tarif premium** - Facturation spéciale pour les services à valeur ajoutée
- **Contrôle du routage des appels** - Routage dynamique des destinations en fonction du temps/emplacement
- **Réseaux privés virtuels** - Plans de numérotation d'entreprise
- **Filtrage des appels** - Autoriser/bloquer les appels en fonction de critères

Versions du Protocole CAP

OmniSS7 CAMELGW prend en charge plusieurs versions de CAP :

Version	Phase	Fonctionnalités
CAP v1	Phase CAMEL 1	Contrôle d'appel de base, opérations limitées
CAP v2	Phase CAMEL 2	Opérations améliorées, support SMS
CAP v3	Phase CAMEL 3	Support GPRS, opérations supplémentaires
CAP v4	Phase CAMEL 4	Fonctionnalités avancées, support multimédia

Par défaut : CAP v2 (le plus largement déployé)

Architecture

Exemple de Flux d'Appel

Configuration

Prérequis

- OmniSS7 installé et en cours d'exécution
- Connectivité M3UA avec MSC/GMSC (gsmSSF)
- Système de Facturation en Ligne (OCS) avec point de terminaison API (optionnel, pour la facturation en temps réel)

Activer le Mode Passerelle CAMEL

Modifiez config/runtime.exe et configurez la section Passerelle CAMEL :

```
config ocsms7,
  # Drapeaux de mode - Activer les fonctionnalités CAP/CAMEL
  cap_client_enabled: true,
  camelgw_mode_enabled: true,
  # Désactiver d'autres modes
  map_client_enabled: false,
  hlr_mode_enabled: false,
  smsc_mode_enabled: false,
  # Configuration de la Version CAP/CAMEL
  # Détermine quelle version CAP utiliser pour les requêtes sortantes et le dialogue
  # Options : :v1, :v2, :v3, :v4
  cap_version: :v2,
  # Intégration OCS (pour la facturation en temps réel)
  ocs_enabled: true,
  ocs_url: "http://your-ocs-server/api/charging",
  ocs_timeout: 5000, # millisecondes
  ocs_auth_token: "your-api-token" # Optionnel, si OCS nécessite une authentification
  # Configuration de la Connexion M3UA pour CAMEL
  # Se connecter en tant qu'ASP (Application Server Process) pour les opérations CAP
  cap_client_m3ua: {
    mode: "ASP",
    callback: (:capclient, :handle_payload, []),
    process_name: :camelgw_client_asp,
  }
  # Point de terminaison local (système CAMELGW)
  local_ip: (10, 179, 4, 13),
  local_port: 2905,
```

```

# Point de terminaison distant (MSC/GMSC - gsmSSF)
remote_ip: (10, 179, 4, 10),
remote_port: 2965,
# Paramètres M3UA
routing_context: 1,
network_appearance: 0,
ssp_identifier: 15
}

```

Configurer les Pages de l'Interface Web

L'Interface Web comprend des pages spécialisées pour les opérations CAMEL :

```

config :control_panel,
use_additional_pages: [
  {events: "events", "events": "Événements S57"},  

  {S57_Web_TestClientLive, "/client", "Client S57"},  

  {S57_Web_M3UAStatusLive, "/m3ua", "M3UA"},  

  {S57_Web_CAMELSessionsLive, "/camel_sessions", "Sessions CAP"},  

  {S57_Web_CAMELRequestLive, "/camel_request", "Requêtes CAP"}
],
page_order: ["events", "/client", "/m3ua", "/camel_sessions",
  "/camel_request", "/application", "/configuration"]

```

Opérations CAP Supportées

Opérations Entrantes (de gsmSSF → gsmSCF)

Opération	Opcode	Description	Gestionnaire
InitialDP	0	Point de Détection Initial - notification de mise en place d'appel	handle_initial_dp/1
EventReportBCSM	6	Événement du Modèle d'Etat d'Appel de Base (réponse, déconnexion, etc.)	handle_event_report_bcsm/1
ApplyChargingReport	71	Rapport de facturation de gsmSSF	handle_apply_charging_report/1
AssistRequestInstructions	16	Demande d'assistance de gsmSCF	handle_assist_request_instructions/1

Opérations Sortantes (de gsmSCF → gsmSSF)

Opération	Opcode	Description	Générateur
Connect	20	Connecter l'appel au numéro de destination	CapRequestGenerator.connect_request/2
Continue	31	Continuer le traitement de l'appel sans modification	CapRequestGenerator.continue_request/1
ReleaseCall	22	Libérer/terminer l'appel	CapRequestGenerator.release_call_request/2
RequestReportBCSMEEvent	23	Demander une notification des événements d'appel	CapRequestGenerator.request_report_bcsm_event_request/2
ApplyCharging	35	Appliquer la facturation à l'appel	CapRequestGenerator.apply_charging_request/3

Fonctionnalités de l'Interface Web

Page des Sessions CAMEL

URL : http://localhost/camel_sessions

Surveillance en temps réel des sessions d'appel CAMEL actives :

Fonctionnalités :

- **Liste des sessions en direct** - Se refresh automatiquement toutes les 2 secondes
- **Détails de la session** - OTID, ID d'appel, État, Durée
- **Version CAP** - Affiche la version du protocole (CAP v1/v2/v3/v4) détectée à partir de l'InitialDP
- **Informations sur l'appel** - IMSI, Numéro A, Numéro B, Clé de Service
- **Suivi de l'état** - Initié, Répondu, Terminé
- **Chronomètre de durée** - Affichage de la durée d'appel en temps réel

Colonnes du tableau :

- ID d'appel, État, Version, IMSI, Numéro appelant, Numéro appelé, Clé de Service, Durée, Heure de début, OTID

États de la session :

- **Initié** - InitialDP reçu, en attente de réponse
- **Répondu** - Appel répondu, facturation en cours
- **Terminé** - Appel terminé, CDR généré

Détection de la Version CAP : Le système détecte automatiquement la version du protocole CAP à partir de la portion de dialogue de l'InitialDP et l'affiche dans la colonne Version. Cela aide à identifier quelle version CAP chaque MSC utilise.

Constructeur de Requêtes CAMEL

URL : http://localhost/camel_request

Outil interactif pour construire et envoyer des requêtes CAP :

Fonctionnalités :

- **Sélecteur de type de requête** - InitialDP, Connect, ReleaseCall, etc.
- **Changement de formulaire dynamiques** - S'adapte au type de requête sélectionné
- **Options SCCP/M3UA** - Configuration d'adressage avancée
- **Historique des requêtes** - Dernières 20 requêtes avec statut
- **Suivi de session** - Maintient l'OTID pour les requêtes de suivi
- **Retour d'information en temps réel** - Messages de succès/erreur

Types de Requêtes :

1. **InitialDP** - Démarrer une nouvelle session d'appel
 - Clé de Service (entier)
 - Numéro appelant (A-partie)
 - Numéro appelé (B-partie)
2. **Connect** - Routage de l'appel vers la destination
 - Numéro de destination
3. **ReleaseCall** - Terminer l'appel
 - Code de Cause (16=Normal, 17=Occupé, 31=Non spécifié)
4. **RequestReportBCSMEEvent** - Demander des notifications d'événements
 - Événements : Answer, tDisconnect, tAnswer, tDisconnect
5. **Continue** - Continuer l'appel sans modification
 - Aucun paramètre requis
6. **ApplyCharging** - Appliquer des limites de durée d'appel
 - Durée (secondes, 1-864000)
 - Libération sur Timeout (booléen)
 - Voir [Guide du Constructeur de Requêtes CAMEL](#) pour une utilisation détaillée

Options SCCP Avancées :

- Titre Global de la Partie Appelée
- Titre Global de la Partie Appelante
- SSN Appelée (par défaut : 146 = gsmSSF)
- SSN Appelante (par défaut : 146)

Options M3UA :

- OPC (Code de Point d'Origine, par défaut : 5013)
- DPC (Code de Point de Destination, par défaut : 5011)

Intégration avec OCS

Cycle de Vie de l'Appel avec Facturation

1. Initiation de l'Appel (InitialDP)

Lorsque le MSC envoie l'InitialDP, CAMELGW :

1. **Déetecte la version CAP** - Examine la portion de dialogue pour identifier CAP v1/v2/v3/v4
2. **Décode le message CAP** - Extrait l'IMSI, les numéros appelant/appelé
3. **Appelle OCS** - API InitiateSession
4. **Reçoit l'autorisation** - MaxUsage (par exemple, 30 secondes)
5. **Stocke la session** - Dans SessionStore (table ETS) avec la version CAP
6. **Repond au MSC** - RequestReportBCSMEEvent + Continue (en utilisant la même version CAP)

Exemple :

```

# Données InitialDP décodées
%{
imsi: "310150123456789",
called_party_number: "14155551234",
called_party_number: "14155556789",
service_key: 1,
msc_address: "19216800123",
cap_version: :w2 # DéTECTé à partir du dialogue
}

# Réponse OCS
{:ok, %{:max_usage: 30}} # 30 secondes autorisées

```

```
# Entrée SessionStore
%{
call_id: "CAMEL-40800173",
initial_dp_data: %(...),
cap_version: :w2, # Stocké pour la génération de réponse
start_time: 1730246400,
state: :initiated
}%
```

2. Réponse à l'Appel (EventReportBCSM - oAnswer)

Lorsque l'appel est répondu :

1. **Reçoit l'événement oAnswer** - Du MSC
2. **Met à jour OCS - UpdateSession avec usage=0**
3. **Démarre la boucle de débit** - OCS commence la facturation
4. **Met à jour l'état de la session** : answered dans SessionStore
5. **Continue l'appel** - Envoie Continue au MSC

3. Mises à Jour Périodiques (Optionnel)

Pour les appels longs, demander un crédit supplémentaire :

```
# Toutes les 30 secondes
OCS.Client.update_session(call_id, {}, current_usage)
```

Si MaxUsage retourne 0, l'abonné n'a pas de crédit → Envoyer ReleaseCall

4. Terminaison de l'Appel (EventReportBCSM - oDisconnect)

Lorsque l'appel se termine :

1. **Reçoit l'événement oDisconnect** - Du MSC
2. **Calcule la durée totale** - À partir de l'heure de début de la session
3. **Termine la session OCS** - API TerminateSession
4. **CDR généré** - Par OCS avec le coût final
5. **Nettoie la session** - Supprime de SessionStore
6. **Envole ReleaseCall** - Confirme la terminaison au MSC

Analyse des CDR

Les CDR sont générés par votre OCS et incluent généralement :

Champs CDR de CAMEL :

- Account - IMSI ou numéro appelleur
- Destination - Numéro de la partie appelée
- OriginID - Identifiant d'appel unique (CAMEL-OTID)
- Usage - Durée totale de l'appel (secondes)
- Cost - Coût calculé
- IMSI - IMSI du client
- CallingNumber - A-partie
- CalledPartyNumber - B-partie
- MSCAddress - Code de point MSC servant
- ServiceKey - Clé de service CAMEL

Tests

Test Manuel avec le Constructeur de Requêtes

1. **Naviguer vers le Constructeur de Requêtes** :

```
http://localhost/camel_request
```

2. **Envoyer InitialDP :**

- Sélectionner "InitialDP" dans le menu déroulant
- Clé de Service : 108
- Numéro Appelant : 1415551234
- Numéro Appelé : 14155556789
- Cliquer sur "Envoyer la Requête InitialDP"
- Noter l'OTID généré

3. **Surveiller la Session** :

- Ouvrir un nouvel onglet : http://localhost/camel_sessions
- Voir la session active avec l'état "Initié"

4. **Simuler la Réponse à l'Appel** :

- Retourner au Constructeur de Requêtes
- Sélectionner "EventReportBCSM"
- Type d'Événement : oAnswer
- Cliquer sur "Envoyer la Requête EventReportBCSM"
- L'état de la session change en "Répondu"

5. **Terminer l'Appel** :

- Sélectionner "ReleaseCall"
- Code de Cause : 16 (Normal)
- Cliquer sur "Envoyer la Requête ReleaseCall"
- L'état de la session change en "Terminé"

Test avec un MSC Réel

Configurer le Service CAMEL du MSC

Sur votre MSC/VLR, configurez le service CAMEL :

```
# Exemple de configuration MSC Huawei
ADD CAMELSERVICE:
SERVICENAME=1,
SERVICEKEY=100,
GSMSCFAID=05512341234", # Titre Global CAMELGW
DEFAULTCALLHANDLING=CONTINUE;

ADD CAMELSUBSCRIBER:
IMSI="310150123456789",
SERVICEKEY=1,
TRIGGERTYPE=TERMCALL;
```

Surveiller les Journaux

Regardez les journaux CAMELGW pour les messages CAP entrants :

```
# Voir les journaux en temps réel
tail -f /var/log/omniss7/omniss7.log

# Filtrer pour les événements CAP
grep "CAP:" /var/log/omniss7/omniss7.log

# Voir le journal des événements (format JSON)
curl http://localhost/api/events | jq '.[] | select(.map_event | startswith("CAP:"))'
```

Test de Charge

Utilisez le Constructeur de Requêtes dans une boucle pour les tests de charge :

```
# Envoyer 100 requêtes InitialDP
for i in {1..100}; do
  curl -X POST http://localhost/api/camel/initial_dp \
  -H "Content-Type: application/json" \
  -d '{
    "service_key": 100,
    "calling_number": "14155556789",
    "called_number": "14155556789"
  }'
done
sleep 0.1
```

Surveillance & Opérations

Métriques Prometheus

CAMELGW expose des métriques à <http://localhost:8080/metrics> :

Métriques spécifiques à CAP :

- cap_requests_total(operation) - Total des requêtes CAP par type d'opération (par exemple, initialDP, requestReportBCSMEvent)

Métriques MAP/API supplémentaires :

- map_requests_total(operation) - Total des requêtes MAP par type d'opération
- map_request_duration_milliseconds(operation) - Histogramme de durée des requêtes
- map_pending_requests - Nombre de transactions MAP en attente

Métriques STP M2UA (si le mode STP est activé) :

- m2ua_stp_messages_received_total(peer_name,point_code) - Messages reçus des pairs
- m2ua_stp_messages_sent_total(peer_name,point_code) - Messages envoyés aux pairs
- m2ua_stp_routing_failures_total(reason) - Echecs de routage par raison

Exemples de requêtes :

```
# Requêtes CAP
curl http://localhost:8080/metrics | grep cap_requests_total

# Total des InitialDP reçus
curl http://localhost:8080/metrics | grep 'cap_requests_total(operation="initialDP")'

# Requêtes MAP en attente
curl http://localhost:8080/metrics | grep map_pending_requests
```

Vérifications de Santé

```
# Vérifier la connectivité M3UA
curl http://localhost/api/m3ua-status

# Vérifier la connectivité OCS
curl http://localhost/api/ocs-status

# Vérifier les sessions actives
curl http://localhost/api/camel/sessions/count
```

Configuration des Journaux

Ajustez le niveau de journalisation dans config/runtime.exs :

```
config :logger
  level: :info # Options : :debug, :info, :warning, :error
# Activer la journalisation de débogage CAP
config :logger, :console,
  metadata: [:cap_operation, :otid, :call_id]
```

Dépannage

Problème : Aucun message CAP reçu

Symptômes : Le Constructeur de Requêtes fonctionne, mais le MSC n'envoie pas l'InitialDP

Vérifiez :

- État du lien M3UA : curl http://localhost/api/m3ua-status
- Configuration du service CAMEL du MSC (Cle de Service, adresse gsmSCF)
- Routeage SCCP (Le Titre Global doit router vers CAMELGW)
- Règles de pare-feu (autorisier le port SCTP 2905)

Solution :

```
# Vérifier la connectivité M3UA
tcpdump -i eth0 sctp

# Vérifier si le MSC peut atteindre CAMELGW
ss -tuln | grep 2905
```

Problème : Erreurs OCS

Symptômes : INSUFFICIENT_CREDIT ou erreurs de délai d'attente

Vérifiez :

- OCS est accessible : curl http://your-ocs-server/api/health
- Le compte a un solde dans OCS
- Plan de tarification configuré dans OCS
- Connectivité réseau vers OCS
- Le jeton d'authentification est valide (si requis)

Solution :

- Vérifiez la configuration de l'URL OCS dans runtime.exs
- Vérifiez les journaux OCS pour des erreurs
- Testez l'API OCS manuellement avec curl
- Vérifiez que les règles de pare-feu permettent la connectivité

Problème : Session non trouvée

Symptômes : EventReportBCSM échoue avec "Session non trouvée"

Cause : Mismatch d'OTID ou session expirée

Solution :

- Vérifiez l'OTID dans les journaux
- Vérifiez le délai d'expiration de la session (par défaut : pas d'expiration)
- Assurez-vous que le DTID correspond à l'OTID dans les messages Continue/End

```
# Vérifiez les sessions actives
iex> CAMELGW.SessionStore.list_sessions()
```

Problème : Erreurs de décodage

Symptômes : Failed to decode InitialDP dans les journaux

Cause : Mismatch de version CAP ou message malformé

Solution :

- Vérifiez que la configuration de la version CAP correspond au MSC
- Vérifiez que l'encodage ASN.1 est correct
- Capturez PCAP et analysez avec Wireshark

```
# Capturez les messages CAP
tcpdump -i eth0 -w cap_trace.pcap sctp port 2905
```

```
# Analysez avec Wireshark (filtre : m3ua)
wireshark cap_trace.pcap
```

Configuration Avancée

Plusieurs Versions CAP

Supportez différentes versions CAP par clé de service :

```
config :omniss7,
  cap_version_map: %{
    100 => :v2, # La clé de service 100 utilise CAP v2
    200 => :v3, # La clé de service 200 utilise CAP v3
    300 => :v4 # La clé de service 300 utilise CAP v4
  },
  cap_version: :v2 # Par défaut
```

Résumé

Le mode Passerelle CAMEL permet à OmniSS7 de fonctionner comme une plateforme complète de Réseau Intelligent avec :

- Support complet du protocole CAP (v1/v2/v3/v4)
- Facturation en temps réel via l'intégration OCS
- Opérations de contrôle d'appel (Connect, Release, Continue)
- Gestion des sessions avec intégration ETSI
- Tests interactifs via le Constructeur de Requêtes de l'Interface Web
- Surveillance en direct des sessions d'appel actives
- Génération de CDR pour la facturation et l'analyse
- Performance et fiabilité prêtes pour la production

Pour des informations supplémentaires :

- Documentation du Constructeur de Requêtes CAMEL
- Référence Technique - Opérations CAP

Produit : Passerelle CAMEL Omnis7
Version de la Documentation : 1.0
Dernière Mise à Jour : 2023-10-26

CAMEL Request Builder - Résumé de l'implémentation

Aperçu

Un nouveau composant LiveView a été créé pour construire et envoyer des requêtes CAMEL/CAP des fins de test. Cela fournit une interface utilisateur interactive pour créer des opérations InitialDP et d'autres opérations CAMEL.

Nouveaux composants

1. CAMEL Request Builder LiveView

Fonctionnalités :

- Interface utilisateur interactive basée sur un formulaire pour construire des requêtes CAMEL.
- Support pour plusieurs types de requêtes:
 - InitialDP** - Point de Détection Initiale (notification de configuration d'appel)
 - Connect** - Connecter l'appel à la destination
 - ReleaseCall** - Libérer/terminer l'appel
 - RequestReportBCSMEvent** - Demander des notifications d'événements
 - Continue** - Continuer le traitement de l'appel
 - ApplyCharging** - Appliquer des limites de facturation/durée aux appels

Capacités clés :

- Menu déroulant de sélection du type de requête
- Champs de formulaire dynamiques en fonction du type de requête sélectionné
- Options SCCP/M3UA pour les destinations (sous-système/point de terminaison)
- Paramètres de la partie appelée/appelante
 - Configuration du SSN (Numéro de sous-système)
 - Paramètres OPC/DPC (Code de point)
- Historique des requêtes en temps réel (20 dernières requêtes)
- Suivi de session via OTID
- Retour d'information sur le succès/erreur
- Suivi de la taille de la requête

Route : /camel_request

2. Journal d'événements amélioré avec support CAMEL

Nouvelles fonctions :

- `paklog_camel/2` - Journalisation dédiée des messages CAMEL/CAP
- `lookup_cap_opcode_name/1` - Recherche du code d'opération CAP
- `find_cap_opcode/1` - Extraire le code d'opération CAP à partir de JSON
- `extract_cap_tids/1` - Extraire OTID/DTID des messages CAP
- `format_cap_to_json/1` - Convertir les PDU CAP en format JSON

Codes d'opération CAP pris en charge :

```
0 => "initialDP"
5 => "connect"
6 => "releaseCall"
7 => "requestReportBCSMEvent"
8 => "eventReportBCSM"
10 => "connect"
13 => "furnishChargingInformation"
35 => "applyCharging"
...
(47 opérations au total)
```

Fonctionnalités :

- Journalisation JSON de toutes les requêtes/réponses CAMEL
- Détection et suivi des actions TCAP (Début/Continuer/Fin/Abandon)
- Émission d'adresse message SCCP
- Gestion des erreurs pour les messages malformés
- Traitement des tâches en arrière-plan (non-bloquant)
- Événement préfixé par "CAP:" pour un filtrage facile

3. CapClient mis à jour

Modifications :

- Ajout d'appels `paklog_camel/2` pour les messages entrants et sortants
- Journalisation double : MAP (paklog) et CAP (paklog_camel) pour la compatibilité
- Messages sortants journalisés dans `scpp_m3ua_maker/2`
- Messages entrants journalisés dans `handle_payload/1`

Configuration

Les nouvelles pages LiveView ont été ajoutées à la configuration d'exécution :

```
# Fichier : config/runtime.exs

config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "Événements SS7"},
    {SS7.Web.TestClientLive, "/client", "Client SS7"},
    {SS7.Web.M3UASessionLive, "/m3ua", "M3UA*"},
    {SS7.Web.HLRSessionLive, "/hlr", "Sessions HLR*"},
    {SS7.Web.CAMELsessionsLive, "/camel_sessions", "Sessions CAMEL*"},
    {SS7.Web.CAMELrequestLive, "/camel_request", "Constructeur de requêtes CAMEL*"}
  ],
  page_order: ["events", "/client", "/m3ua", "/hlr_links",
    "/camel_sessions", "/camel_request",
    "/application", "/configuration"]
]
```

Utilisation

Accéder au Constructeur de Requêtes

1. Naviguez vers : https://votre-server:8087/camel_request
2. Sélectionnez le type de requête dans le menu déroulant
3. Remplissez les paramètres requis
4. Optionnellement, développez "Options SCCP/M3UA Avancées" pour un réglage fin
5. Cliquez sur "Envoyer [RequestType] Requête"

Flux de Requête

InitialDP (Nouvel Appel)

1. Définir la Clé de Service (par exemple, 100)
2. Définir le Numéro Appelant (A-Party)
3. Définir le Numéro Appelé (B-Party)
4. Envoyer la requête → Génère un nouvel OTID
5. OTID stocké dans la session pour les requêtes de suivi

Requêtes de Suivi (Connect, ReleaseCall, etc.)

1. Doit avoir un OTID actif de l'InitialDP
2. La requête utilise automatiquement l'OTID stocké
3. Avertissement affiché si aucun OTID actif

Paramètres de Requête

InitialDP :

- Clé de Service (entier)
- Numéro Appelant (format ISDN)
- Numéro Appelé (format ISDN)

Connect :

- Numéro de Destination (où acheminer l'appel)

ReleaseCall :

- Code de Cause (16 = Normal, 17 = Occupé, 31 = Non spécifié)

RequestReportBCSMEvent :

- Événements BCSM (séparés par des virgules : oAnswer, oDisconnect, etc.)

Continue :

- Aucun paramètre (utilise l'OTID actif)

ApplyCharging :

- Durée (secondes, 1-864000) - Durée maximale de l'appel avant action
- Libérer en cas de dépassement de délai (booléen) - Si l'appel doit être libéré lorsque la durée expire

Options Avancées

Adresse SCCP :

- GT de la Partie Appelée (Tire Global)
- GT de la Partie Appelante
- SSN Appelé (par défaut 146 = gsmSSF)
- SSN Appelant (par défaut 146)

Codes de Point M3UA :

- OPC (Code de Point d'origine, par défaut 5013)
- DPC (Code de Point de destination, par défaut 5011)

Journalisation JSON

Tous les messages CAMEL sont désormais journalisés au format JSON dans le journal des événements avec :

- Direction : entrant/sortant
- Action TCAP : Début/Continuer/Fin/Abandon
- Opération CAP : par exemple, "CAP:initialDP", "CAP:connect"
- Adresse CCGC : Information sur la Partie Appelée/Appelante
- TIDs : OTID/DTID pour la corrélation
- Message Complet : PDU CAP encodé en JSON

Exemple d'Entrée de Journal

```
{
  "map_event": "CAP:initialDP",
  "direction": "outgoing",
  "tcap_action": "Begin",
  "otid": "ABC04",
  "ccgc": "1234567890",
  "sccp_called": "1234567890",
  "ssn": 146,
  "ssn_global": {
    "GlobalTitle": "GlobalTitle1234567890",
    "Dnis": "12345678901234567890",
    "NumberingPlan": "isdn_tele",
    "NatureOfAddress_Indicator": "international"
  },
  "event_message": "{ ... full CAP PDU ... }"
}
```

Historique des Requêtes

L'interface utilisateur affiche les 20 dernières requêtes avec :

- Horodatage
- Type de requête (avec badge code par couleur)
- OTID (premiers 8 caractères hexadécimaux)
- Statut (envoyé/erreur)
- Taille du message en octets

Suivi de Session

Panneau d'Informations de Session Actuelle :

- Affiche l'OTID actif
- Montre la taille du dernier message en octets
- Visible uniquement lorsque la session est active

Flux de Travail de Test

1. **Démarrer un Nouvel Appel :**
 - Envoyer InitialDP → Obtenir OTID
 - Le système crée une session
2. **Contrôler l'Appel :**
 - Envoyer RequestReportBCSMEvent → Demander des notifications
 - Envoyer ApplyCharging → Définir la limite de durée de l'appel (par exemple, 290 secondes)
 - Envoyer Connect → Acheminer vers la destination
 - OU Envoyer ReleaseCall → Terminer
3. **Voir les Résultats :**
 - Vérifier l'historique des requêtes
 - Surveiller la page des Sessions CAMEL
 - Examiner les journaux d'événements avec le préfixe "CAP:*

ApplyCharging - Contrôle de Durée d'Appel

Aperçu

L'opération ApplyCharging vous permet de définir une durée maximale d'appel et éventuellement de libérer l'appel lorsque cette durée expire. Cela est généralement utilisé pour des scénarios de facturation prépayée ou pour imposer des limites de temps sur les appels.

Cas d'Utilisation

- **Facturation Prépayée** : Limiter la durée d'appel en fonction du solde de l'abonné
- **Facturation Basée sur le Temps** : Imposer des intervalles de facturation périodiques

- **Gestion des Ressources** : Empêcher les appels de se prolonger indéfiniment
- **Intégration OCS** : Coordonner avec les systèmes de facturation en ligne pour le contrôle de crédit en temps réel

Paramètres

Durée (maxCallPeriodDuration)

- **Type** : Entier (1-864000 secondes)
- **Description** : Nombre maximum de secondes que l'appel peut durer avant que le minuteur expire
- **Exemples**
 - 60 = 1 minute
 - 299 = 4 minutes 59 secondes (valeur de test courante)
 - 3600 = 1 heure
 - 86400 = 24 heures

Libérer en cas de dépassement de délai (releaseIfDurationExceeded)

- **Type** : Booléen (vrai/faux)
- **Par défaut** : vrai
- **Description** : Que se passe-t-il lorsque la durée expire :
 - vrai : Libérer/déconnecter automatiquement l'appel
 - faux : Envoyer une notification mais garder l'appel actif (permet à gsmSCF d'agir)

Structure du Message

Le message ApplyCharging est encodé en tant que TCAP Continue avec :

- **TCAP** : Message Continue (utilise la transaction existante)
- **Opcode** : 35 (applyCharging)
- **Paramètres** : ApplyChargingArg contenant :
 - aChargingCharacteristics : Informations de facturation basées sur le temps
 - aChargingCharacteristics.timeDurationCharging : Durée maximale et indicateur de libération
 - partyToCharge : Quelle partie est facturée (par défaut : sendingSideID)

Exemple d'Utilisation

Scénario : Appel prépayé avec limite de 5 minutes

1. Envoyer **InitialDP** pour commencer la surveillance de l'appel

```
Clé de Service : 100
Appelé : 4477069980123
Appelé : 447706998456
→ OTID : A1B2C3D4
```

2. Envoyer **ApplyCharging** pour définir la limite de 5 minutes

```
Durée : 300 (secondes)
Libérer en cas de dépassement de délai : vrai
→ Utilise OTID : A1B2C3D4
```

3. Envoyer **Connect** pour compléter l'appel

```
Destination : 447706998456
→ Utilise OTID : A1B2C3D4
```

4. Après 5 minutes (300 secondes) :

- L'appel est automatiquement libéré par le réseau
- gsmSCF reçoit une notification de déconnexion

Meilleures Pratiques

1. Envoyer toujours **ApplyCharging** AVANT Connect
 - Assure que la facturation est active lorsque l'appel se connecte
 - Empêche les segments d'appel non facturés
2. Utiliser avec **RequestReportBCSMEvent**
 - Demander les événements **oAnswer** et **oDisconnect**
 - Permet de suivre la durée réelle de l'appel
 - Permet de réappliquer la facturation si nécessaire
3. Définir des durées raisonnables
 - Trop court : Opérations de facturation fréquentes, mauvaise expérience utilisateur
 - Trop long : Risque de perte de revenus sur les appels prépayés
 - Typique : 60-300 secondes pour prépayé, plus long pour postpayé
4. Gérer le dépassement de délai de manière élégante
 - Si **release=false**, être prêt à gérer les notifications d'expiration du minuteur
 - Implémenter une logique pour prolonger la durée ou libérer l'appel

Gestion des Erreurs

Problèmes courants :

- **Aucun OTID actif** : Doit envoyer InitialDP d'abord
- **Durée invalide** : Doit être de 1 à 864000 secondes
- **Support réseau** : Certaines implémentations SSF peuvent ne pas prendre en charge ApplyCharging
- **Précision du minuteur** : La résolution du minuteur réseau est généralement de 1 seconde, mais peut varier

Surveillance

Suivre les opérations ApplyCharging via :

- **Historique des Requêtes** : Montrer les requêtes ApplyCharging envoyées
- **Journal des Événements** : Rechercher "CAP-applyCharging"
- **Sessions CAMEL** : Surveiller les sessions actives avec facturation appliquée
- **Trace TCAP** : Déboguer les problèmes d'encodage/décodage

Détails de l'Implémentation

Gestion de l'État

- **LiveView** assigne l'état du formulaire de suivi
- OTID stocké dans les assignments de socket
- Historique des requêtes limité à 20 entrées
- Actualisation automatique désactivée (envoi manuel uniquement)

Génération de Requêtes

- Utilise le module existant **CapRequestGenerator**
- Construit les structures TCAP/CAP appropriées
- Encodé avec le codage : **:TCAPMessage**
- Enveloppe dans SCCP via **CapClient.sccp_m3ua_maker/2**

Mécanisme d'Envoi

- Envoie via M3UA à :**camelw_client_asp**
- Utilise le contexte de routage 1
- Encapsulation automatique SCCP/M3UA

Gestion des Erreurs

- Validation du formulaire avec retour d'information utilisateur
- Gestion élégante de l'OTID manquant
- Erreurs d'analyse affichées dans l'interface utilisateur
- Echecs d'encodage journalisés

Améliorations Futures

Ajouts potentiels :

1. Modules/paquets de requêtes
2. Corrélation et affichage des réponses
3. Visualisation du flux d'appel
4. Détails de session en profondeur
5. Exporter l'historique des requêtes
6. Tests de charge (requêtes en masse)
7. Export PCAP des messages générés
8. Validation des paramètres CAP

Notes d'Intégration

- Compatible avec la journalisation MAP existante (paklog)
- Partage la même base de données pour les événements avec les événements MAP
- Utilise même infrastructure SCCPM3UA
- Fonctionne avec CAMELSessionLive pour la surveillance
- S'intègre avec le routage M3UA existant

Fichiers Modifiés

- **config/runtime.exe** - MIS À JOUR

Dépendances

- **CapRequestGenerator** existant
- **CapClient** pour l'envoi M3UA
- **M3UA** pour la transmission de paquets

- EventLog pour la journalisation des messages
- Cadre Phoenix LiveView
- Panneau de Contrôle pour l'infrastructure de l'interface utilisateur



Guide des Fonctionnalités Communes

[Retour à la Documentation Principale](#)

Ce guide couvre les fonctionnalités communes à tous les modes de fonctionnement d'OmniSS7.

Table des Matières

1. [Aperçu de l'Interface Web](#)
2. [Documentation de l'API](#)
3. [Surveillance et Métriques](#)
4. [Meilleures Pratiques](#)

Aperçu de l'Interface Web

L'interface Web est accessible via l'adresse de votre serveur web configuré.

OmniSS7 Stack CAMEL GW v1.0.0

Last updated: 2025-10-25 23:57:55 UTC [Refresh](#)

[Documentation](#) © 2025 OmniTouch

SS7 Events	M3UA Status	Actions																					
SS7 Client																							
M3UA	<table border="1"> <thead> <tr> <th>Name</th> <th>PID</th> <th>Status</th> <th>ASP State</th> <th>Assoc/SCTP</th> <th>Local</th> <th>Remote</th> <th>RC</th> <th>24h Uptime</th> <th colspan="2">Actions</th> </tr> </thead> <tbody> <tr> <td>:camelgw_client_asp</td> <td>#PID=0.1322.0></td> <td>Up</td> <td>active</td> <td>established</td> <td>10.5.198.200:2905</td> <td>10.179.4.10:2905</td> <td>4</td> <td>0.1%</td> <td colspan="2"> ▼ </td> </tr> </tbody> </table>	Name	PID	Status	ASP State	Assoc/SCTP	Local	Remote	RC	24h Uptime	Actions		:camelgw_client_asp	#PID=0.1322.0>	Up	active	established	10.5.198.200:2905	10.179.4.10:2905	4	0.1%	▼	
Name	PID	Status	ASP State	Assoc/SCTP	Local	Remote	RC	24h Uptime	Actions														
:camelgw_client_asp	#PID=0.1322.0>	Up	active	established	10.5.198.200:2905	10.179.4.10:2905	4	0.1%	▼														
CAP Sessions																							
CAP Requests																							
Resources																							
Configuration																							

M3UA Status

24-Hour Availability Timeline

Uptime: 0.1% Total Up: 1m Total Down: 23h 58m

24h ago 18h ago 12h ago 6h ago Now

Up: Up Down: Down

Basic Information

Name:	camelgw_client_asp	PID:	#PID=0.1322.0>	Status:	Up	Mode:	ASP	ASP State:	active	Association State:	established
Routing Context:	4										

Network Configuration

Local IP:	10.5.198.200	Local Port:	2905	Remote IP:	10.179.4.10	Remote Port:	2905
-----------	--------------	-------------	------	------------	-------------	--------------	------

Additional Details

[Raw Data](#) (click to expand)



Navigation Principale

- **Événements** - Événements de signalisation SS7 en temps réel et journaux de messages
- **Application** - États et configurations de l'exécution
- **Configuration** - Visualiser ou configurer du système
- **Etat M3UA** - Connexions de pairs M3UA (mode STP)
- **File d'attente SMS** - Messages SMS sortants (mode SMSc)

Accéder à l'Interface Web

1. Ouvrez votre navigateur web
2. Naviguez vers le nom d'hôte configuré (par exemple, <http://localhost>)
3. Consultez le tableau de bord de l'état du système

Documentation de l'API Swagger

Documentation API interactive :

<http://your-server/swagger>

Configuration de l'Interface Web

Configurer dans config/runtime.exs :

```
config :control_panel,
  # Ordre des pages dans le menu de navigation
  page_order: ["events", "/application", "/configuration"],

  # Paramètres du serveur web
  web: %{
    listen_ip: "0.0.0.0", # IP à lier (0.0.0.0 pour toutes les interfaces)
    port: 80, # Port HTTP (443 pour HTTPS)
    hostname: "localhost", # Nom d'hôte du serveur pour la génération d'URL
    enable_tls: false, # Mettre vrai pour activer HTTPS
    tls_cert: "cert.pem", # Chemin vers le fichier de certificat TLS
    tls_key: "key.pem" # Chemin vers la clé privée TLS
  }
```

Paramètres de Configuration :

Paramètre	Type	Par Défaut	Description
page_order	Liste	["/events", "/application", "/configuration"]	Ordre des pages dans le menu de navigation
listen_ip	Chaine	"0.0.0.0"	Adresse IP à lier au serveur web
port	Port	80	Port pour l'interface HTTP (443 pour HTTPS)
hostname	Chaine	"localhost"	Nom d'hôte du serveur pour la génération d'URL
enable_tls	Booléen	false	Activer HTTPS avec TLS
tls_cert	Chaine	"cert.pem"	Chemin vers le certificat TLS (lorsque TLS activé)
tls_key	Chaine	"key.pem"	Chemin vers la clé privée TLS (lorsque TLS activé)

Configuration du Logger

Configurer le niveau de journalisation dans config/runtime.exs :

```
config :logger,
  level: :debug # Options : :debug, :info, :warning, :error
```

Niveaux de Journalisation :

- :debug - Informations détaillées de débogage
- :info - Messages d'information généraux
- :warning - Messages d'avertissement pour des problèmes potentiels

- :error - Messages d'erreur uniquement

Documentation de l'API

URL de Base de l'API

<http://your-server/api>

Codes de Réponse

- 200 - Succès
- 400 - Mauvaise Demande
- 504 - Délai d'Attente de la Passerelle

Spécification OpenAPI

<http://your-server/swagger.json>

Surveillance et Métriques

Point de Terminaison des Métriques Prometheus

<http://your-server/metrics>

Catégories de Métriques Clés

Métriques M3UA/SCTP :

- Changements d'état d'association SCTP
- Transitions d'état ASP M3UA
- Unités de données de protocole envoyées/reçues

Métriques M2PA :

- Transitions d'état de lien (DOWN → ALIGNMENT → PROVING → READY)
- Messages et octets envoyés/reçus par lien
- Erreurs spécifiques au lien (décodage, encodage, SCTP)

Métriques STP :

- Messages reçus/envoyés par paire
- Échecs de routage par raison
- Distribution du trafic entre les paires

Métriques Client MAP :

- Requêtes MAP par type d'opération
- Histogrammes de durée des requêtes
- Jauge des transactions en attente

Métriques CAP :

- Requêtes CAP par type d'opération
- Opérations de passerelle CAMEL

Métriques SMSC :

- Profondeur de la file d'attente
- Taux de livraison
- Messages échoués

Intégration Grafana

Les métriques d'OmniSS7 sont compatibles avec Prometheus et Grafana.

Meilleures Pratiques

Recommendations de Sécurité

1. Isolation du Réseau

- Déployer dans un VLAN dédié
- Règles de pare-feu pour restreindre l'accès
- Autoriser SCTP uniquement à partir d'adresses connues

2. Sécurité de l'Interface Web

- Activer TLS pour la production
- Utiliser un proxy inverse avec authentification
- Restreindre aux IPs de gestion

3. Sécurité de l'API

- Mettre en œuvre une limitation de débit
- Utiliser des clés API ou OAuth
- Journaliser toutes les requêtes pour audit

Optimisation des Performances

1. Limites TPS

- Configurer un TPS approprié
- Surveiller la charge du système
- Ajuster les tampons SCTP

2. Optimisation de la Base de Données

- Ajouter des index
- Archiver les anciens messages
- Surveiller le pool de connexions

3. Ajustement M3UA

- Ajuster les intervalles de battement SCTP
- Configurer les valeurs de délai d'attente
- Utiliser plusieurs liens pour la redondance

Surveillance et Alerte

Métriques Clés :

- État de connexion M3UA
- Taux de réussite des requêtes MAP
- Temps de réponse de l'API
- Profondeur de la file d'attente de messages

Seuils d'Alerte :

- M3UA hors service > 1 minute
- Taux de délai d'attente MAP > 10%
- Profondeur de la file d'attente > 1000
- Taux d'erreur de l'API > 5%

Référence Complète de Configuration

Tous les Paramètres de Configuration

Cette section fournit une référence complète de tous les paramètres de configuration disponibles dans tous les modes de fonctionnement.

Configuration du Logger (:logger)

```
config :logger,
  level: :debug # :debug | :info | :warning | :error
```

Configuration de l'Interface Web (:control_panel)

```
config :control_panel,
  page_order: ["/events", "/application", "/configuration"],
  web: %{
    listen_ip: "0.0.0.0",
    port: 80,
    hostname: "localhost",
    enable_tls: false,
    tls_cert: "cert.pem",
    tls_key: "key.pem"
  }
```

Paramètre	Type	Requis	Par Défaut	Description
page_order	Liste de Chaine	Non	["/events", "/application", "/configuration"]	Ordre des pages du menu de navigation
web.listen_ip	Chaine	Oui	"0.0.0.0"	Adresse IP à lier au serveur web
web.port	Entier	Oui	80	Numéro de port HTTP/HTTPS
web.hostname	Chaine	Oui	"localhost"	Nom du serveur
web.enable_tls	Booléen	Non	false	Activer HTTPS
web.tls_cert	Chaine	Si TLS activé	"cert.pem"	Chemin vers le certificat TLS
web.tls_key	Chaine	Si TLS activé	"key.pem"	Chemin vers la clé privée TLS

Configuration STP M3UA (:omiss7)

```
config :omiss7,
  m3ua_stp: %{
    enabled: false,
    local_ip: {127, 0, 0, 1},
    local_port: 2905
  },
  enable_gt_routing: true,
  m3ua_peers: [...],
  m3ua_routes: [...],
  m3ua_gt_routes: [...]
```

Paramètre	Type	Requis	Par Défaut	Description
m3ua_stp.enabled	Booléen	Oui	false	Activer le mode STP au démarrage
m3ua_stp.local_ip	Tuple	Oui	{127, 0, 0, 1}	IP à lier pour M3UA entrant
m3ua_stp.local_port	Entier	Oui	2905	Port SCTP pour M3UA
enable_gt_routing	Booléen	Non	false	Activer le routage de titre global

Paramètres de Pair M3UA :

Paramètre	Type	Requis	Description
peer_id	Entier	Oui	Identifiant unique du pair
name	Chaine	Oui	Nom unique du pair
role	Atome	Oui	:client ou :server
local_ip	Tuple	Si :clientIP locale à lier	
local_port	Entier	Si :clientPort local (0 pour dynamique)	
remote_ip	Tuple	Oui	IP du pair distant
remote_port	Entier	Si :clientPort du pair distant	
routing_context	Entier	Oui	Contexte de routage M3UA
point_code	Entier	Oui	Code de point SS7
network_indicator	Atome	Non	:international ou :national

Paramètres de Route M3UA :

Paramètre	Type	Requis	Description
dest_pc	Entier	Oui	Code de point de destination
peer_id	Entier	Oui	Pair à travers lequel router
priority	Entier	Oui	Priorité de la route (plus bas = plus haute priorité)
network_indicator	Atome	Non	:international ou :national

Paramètres de Route GT M3UA :

Paramètre	Type	Requis	Description
gt_prefix	Chaine	Oui	Préfixe de Titre Global à correspondre
peer_id	Entier	Oui	Pair de destination
priority	Entier	Oui	Priorité de la route
description	Chaine	Non	Description de la route pour journalisation
source_ss	Entier	Non	Correspondre uniquement si SSN source correspond
dest_ss	Entier	Non	Réécrire le SSN de destination à cette valeur

Configuration Client MAP (:omiss7)

```
config :omiss7,
  map_client_enabled: false,
  map_client_m3ua: %{
    mode: "ASP"
    callback: (:MapClient, :handle_payload, []),
    process_name: :map_client_asp,
    local_ip: {10, 0, 0, 100},
    local_port: 2905
    remote_ip: {10, 0, 0, 1},
    remote_port: 2905,
    routing_context: 1
  }
```

Paramètre	Type	Requis	Par Défaut	Description
map_client_enabled	Booléen	Oui	false	Activer le mode Client MAP
map_client_m3ua.mode	Chaine	Oui	"ASP"	Mode de connexion M3UA ("ASP" ou "SGP")
map_client_m3ua.callback	Tuple	Oui	(:MapClient, :handle_payload, [])	Gestionnaire de rappel de message
map_client_m3ua.process_name	Atome	Oui	:map_client_asp	Nom de processus enregistré
map_client_m3ua.local_ip	Tuple	Oui	-	Adresse IP locale
map_client_m3ua.local_port	Entier	Oui	2905	Port SCTP local
map_client_m3ua.remote_ip	Tuple	Oui	-	IP STP/SGP distante
map_client_m3ua.remote_port	Entier	Oui	2905	Port SCTP distant
map_client_m3ua.routing_context	Entier	Oui	-	Contexte de routage M3UA

Configuration du Centre SMS (:omiss7)

```
config :omiss7,
  auto_flush_enabled: false,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: nil,
  auto_flush_tps: 10
```

Paramètre	Type	Requis	Par Défaut	Description
auto_flush_enabled	Booléen	Non	false	Activer le vidage automatique de la file d'attente SMS
auto_flush_interval	Entier	Non	10000	Intervalle de sondage de la file d'attente (millisecondes)
auto_flush_dest_smsc	Chaine	nil	nil	Filtrer par SMSC de destination (nil = tous)

Configuration de l'API HTTP (:omiss7)

Le backend SMS utilise maintenant l'API HTTP au lieu de connexions directes à la base de données.

```
config :omiss7,
  smsc_api_base_url: "https://10.5.198.200:8443",
  frontend_name: "omni-smsc01" # Optionnel : par défaut au nom d'hôte_SMSC
```

Paramètres de l'API :

Paramètre	Type	Requis	Par Défaut	Description
smsc_api_base_url	Chaine	Oui	"https://10.5.198.200:8443"	URL de base pour l'API backend SMS

Point de Terminaison de l'API Utilisés :

- POST /api/frontends - Enregistrer cette instance de frontend avec le backend
- POST /api/messages/raw - Insérer de nouveaux messages SMS
- GET /api/messages - Récupérer la file d'attente de messages (avec l'en-tête smsc)
- PATCH /api/messages/{id} - Mander le message comme livré
- PUT /api/messages/{id} - Mettre à jour le statut du message
- POST /api/events - Ajouter le suivi des événements
- GET /api/status - Point de terminaison de vérification de santé

Enregistrement du Frontend :

Le système s'enregistre automatiquement auprès de l'API backend au démarrage et se réenregistre toutes les 5 minutes. L'enregistrement inclut :

- Nom et type de frontend (SMSC)
- Nom d'hôte
- Temps de fonctionnement en secondes
- Détails de configuration (format JSON)

Remarques de Configuration :

- La vérification SSL est désactivée par défaut pour les certificats auto-signés
- Les requêtes HTTP expirent après 5 secondes
- Tous les horodatages sont au format ISO 8601
- L'API utilise JSON pour les corps de requête/réponse

Documentation Connexe

- [Retour à la Documentation Principale](#)
- [Guide STP](#)
- [Guide Client MAP](#)
- [Guide Centre SMS](#)
- [Guide HLR](#)

Référence de Configuration

[← Retour à la Documentation Principale](#)

Ce document fournit une référence complète pour tous les paramètres de configuration d'OmniSS7.

Table des Matières

1. [Aperçu](#)
2. [Drapeaux de Mode Opérationnel](#)
3. [Paramètres de Mode HLR](#)
4. [Paramètres de Mode SMS](#)
5. [Paramètres de Mode STP](#)
6. [Paramètres de NAT de Titre Global](#)
7. [Paramètres de Connexion M3UA](#)
8. [Paramètres du Serveur HTTP](#)
9. [Paramètres de Base de Données](#)
10. [Valeurs Codées en Dur](#)

Aperçu

La configuration d'OmniSS7 est gérée via `config/runtime.exs`. Le système prend en charge trois modes opérationnels :

- **Mode STP** - Point de Transfert de Signal pour le routage
- **Mode HLR** - Registre de Localisation Domiciliaire pour la gestion des abonnés
- **Mode SMS** - Centre SMS pour la livraison de messages

Fichier de Configuration : `config/runtime.exs`

Drapeaux de Mode Opérationnel

Contrôle quelles fonctionnalités sont activées.

Paramètre	Type	Par Défaut	Description	Modes
<code>map_client_enabled</code>	Boolean	false	Activer le client MAP et la connectivité M3UA	Tous
<code>hlr_mode_enabled</code>	Boolean	false	Activer les fonctionnalités spécifiques au HLR	HLR
<code>smsc_mode_enabled</code>	Boolean	false	Activer les fonctionnalités spécifiques au SMS	SMS

Exemple :

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false
```

Paramètres de Mode HLR

Configuration pour le mode HLR (Registre de Localisation Domiciliaire).

Configuration de l'API HLR

Paramètre	Type	Par Défaut	Requis	Description
hlr_api_base_url	String-		Oui	URL du point de terminaison de l'API HLR backend (vérification SSL codée en dur désactivée)
hlr_service_center_gt_address	String-		Oui	Adresse de Titre Global HLR renvoyée dans les réponses UpdateLocation
smsc_service_center_gt_address	String-		Oui	Adresse GT SMS Sc renvoyée dans les réponses SRI-for-SM

Exemple :

```
config :omniss7,
  hlr_api_base_url: "https://10.180.2.140:8443",
  hlr_service_center_gt_address: "55512341111",
  smsc_service_center_gt_address: "55512341112"
```

Mapping MSISDN ↔ IMSI

Configuration pour la génération d'IMSI synthétiques à partir de MSISDN. Pour une explication technique détaillée de l'algorithme de mapping, voir [Mapping MSISDN ↔ IMSI dans le Guide HLR](#).

Paramètre	Type	Par Défaut	Requis	Description
hlr_imsi_plmn_prefix	String	"50557"	Non	Préfixe PLMN (MCC+MNC) pour la génération d'IMSI synthétiques
hlr_msisdn_country_code	String	"61"	Non	Préfixe de code pays pour le mapping inverse IMSI→MSISDN
hlr_msisdn_nsn_offset	Integer	0	Non	Décalage dans le MSISDN où commence le NSN (typiquement la longueur du code pays)
hlr_msisdn_nsn_length	Integer	9	Non	Longueur du Numéro d'Abonné National à extraire du MSISDN

Exemple (code pays à 2 chiffres) :

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",          # MCC 505 + MNC 57
  hlr_msisdn_country_code: "99",           # Exemple de code pays à 2 chiffres
  hlr_msisdn_nsn_offset: 2,                # Ignorer le code pays à 2 chiffres
  hlr_msisdn_nsn_length: 9                 # Extraire le NSN à 9 chiffres
```

Exemple (code pays à 3 chiffres) :

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",          # MCC 505 + MNC 57
  hlr_msisdn_country_code: "999",           # Exemple de code pays à 3 chiffres
  hlr_msisdn_nsn_offset: 3,                # Ignorer le code pays à 3 chiffres
  hlr_msisdn_nsn_length: 8                 # Extraire le NSN à 8 chiffres
```

Important : Définissez `nsn_offset` sur la longueur de votre code pays pour extraire correctement le NSN. Par exemple :

- Code pays "9" (1 chiffre) → `nsn_offset: 1`
- Code pays "99" (2 chiffres) → `nsn_offset: 2`
- Code pays "999" (3 chiffres) → `nsn_offset: 3`

Configuration InsertSubscriberData (ISD)

Configuration pour les données de provisionnement des abonnés envoyées aux VLR lors de UpdateLocation. Pour une explication détaillée de la séquence ISD et du flux de messages, voir [Configuration InsertSubscriberData dans le Guide HLR](#).

Paramètre	Type	Par Défaut	Requis	Description
isd_network_access_mode	Atom	:packetAndCircuit	Non	Type d'accès réseau : :packetAndCircuit, :packetOnly, ou :circuitOnly
isd_send_ss_data	Boolean	true	Non	Envoyer ISD #2 avec les données des Services Supplémentaires
isd_send_call_barring	Boolean	true	Non	Envoyer ISD #3 avec les données de Blocage d'Appels

Exemple :

```
config :omniss7,
  isd_network_access_mode: :packetAndCircuit,
  isd_send_ss_data: true,
  isd_send_call_barring: true
```

Configuration CAMEL

Configuration pour le routage d'appels intelligents basé sur CAMEL. Pour une explication détaillée de l'intégration CAMEL et des clés de service, voir [Intégration CAMEL dans le Guide HLR](#).

Paramètre	Type	Par Défaut	Requis	Description
camel_service_key	Integer	11_110	Non	Clé de service CAMEL pour les réponses SRI
camel_trigger_detection_point	Atom	:termAttemptAuthorized	Non	Point de déclenchement CAMEL : :termAttemptAuthorized, :tBusy, :tNoAnswer, :tAnswer
camel_gsmSCF_gt_address	String	(utilise le GT appelé)	Non	Titre Global gsmSCF par défaut pour les réponses CAMEL (peut être remplacé par GT NAT)

Exemple :

```
config :omniss7,
  camel_service_key: 11_110,
  camel_trigger_detection_point: :termAttemptAuthorized
```

Préfixes VLR Domiciliaires

Configuration pour distinguer les abonnés domiciliés des abonnés en itinérance. Pour une explication détaillée de la détection domicile/itinérance et des opérations PRN, voir [Gestion des Abonnés en Itinérance dans le Guide HLR](#).

Paramètre	Type	Par Défaut	Requis	Description
home_vlr_prefixes	List	["5551231"]	Non	Préfixes GT VLR considérés comme réseau "domiciliaire"

Exemple :

```
config :omniss7,
```

```
home_vlr_prefixes: ["5551231", "5551234"]
```

Paramètres de Mode SMS

Configuration pour le mode Centre SMS.

Configuration de l'API SMS

Paramètre	Type	Par Défaut	Requis	Description
smsc_api_base_url	String -		Oui	URL du point de terminaison de l'API SMS backend (vérification SSL codée en dur désactivée)
smsc_name	String "{hostname}_SMS	Non		Identifiant SMS pour l'enregistrement backend
smsc_service_center_gt_address	String -		Oui	Adresse de Titre Global du Centre de Service

Exemple :

```
config :omniss7,
  smsc_api_base_url: "https://10.179.3.219:8443",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112"
```

Remarque : L'enregistrement frontend a lieu toutes les **5 minutes** (codé en dur) via le module SMS.FrontendRegistry.

Configuration d'Auto-Vidage

Paramètre	Type	Par Défaut	Requis	Description
auto_flush_enabled	Boolean	true	Non	Activer le traitement automatique de la file d'attente SMS
auto_flush_interval	Integer	10_000	Non	Intervalle de traitement de la file d'attente en millisecondes
auto_flush_dest_smss	String	-	Oui	Nom du SMS de destination pour l'auto-vidage
auto_flush_tps	Integer	10	Non	Taux de traitement des messages (transactions/seconde)

Exemple :

```
config :omniss7,
  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smss: "ipsmgw",
  auto_flush_tps: 10
```

Paramètres de Mode STP

Configuration pour le mode Point de Transfert de Signal M3UA. Pour une configuration de routage détaillée et des exemples, voir le [Guide de Configuration STP](#).

Serveur STP Autonome

Paramètre	Type	Par Défaut	Requis	Description
m3ua_stp.enabled	Boolean	false	Non	Activer le serveur STP M3UA autonome
m3ua_stp.local_ip	Tuple	{127, 0, 0, 1}	Non	Adresse IP pour écouter les connexions
m3ua_stp.local_port	Integer	2905	Non	Port à écouter
m3ua_stp.point_code	Integer	-	Oui (si activé)	Code de point SS7 de ce STP

Exemple :

```
config :omniss7,
  m3ua_stp: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 2905,
    point_code: 100
  }
```

Routage par Titre Global

Paramètre	Type	Par Défaut	Requis	Description
enable_gt_routing	Boolean	false	Non	Activer le routage par GT en plus du routage par PC

Exemple :

```
config :omniss7,
  enable_gt_routing: true
```

Paramètres de NAT de Titre Global

La Traduction d'Adresse de Réseau de Titre Global permet des GT de réponse différents en fonction du préfixe de la partie appelante. Pour une explication détaillée et des exemples, voir le [Guide de NAT de Titre Global](#).

Paramètre	Type	Par Défaut	Requis	Description
gt_nat_enabled	Boolean	false	Non	Activer/désactiver la fonctionnalité de NAT GT
gt_nat_rules	List of Maps	[]	Oui (si activé)	Liste des mappings préfixe-à-GT

Format de Règle : Chaque règle dans gt_nat_rules doit être une carte avec :

- `calling_prefix`: Préfixe de chaîne à faire correspondre avec le GT appelant
- `response_gt`: Titre Global à utiliser dans les réponses

Exemple :

```
config :omniss7,
  gt_nat_enabled: true,
  gt_nat_rules: [
    # Lorsqu'appelé depuis un GT commençant par "8772", répondre avec "55512341112"
    %{calling_prefix: "8772", response_gt: "55512341112"},
    # Lorsqu'appelé depuis un GT commençant par "8773", répondre avec "55512341111"
    %{calling_prefix: "8773", response_gt: "55512341111"},
    # Fallback par défaut (préfixe vide correspond à tous)
    %{calling_prefix: "", response_gt: "55512311555"}
  ]
```

Voir Aussi : [Guide de NAT GT](#) pour une utilisation détaillée et des exemples.

Paramètres de Connexion M3UA

Configuration de connexion M3UA pour le mode client MAP. Pour une utilisation détaillée et des exemples, voir le [Guide Client MAP](#).

Paramètre	Type	Par Défaut	Requis	Description
map_client_m3ua.mode	String	-	Oui	Mode de connexion : "ASP" ou "SGP"
map_client_m3ua.callback	Tuple	-	Oui	Module/fonction de rappel : {MapClient, :handle_payload, []}
map_client_m3ua.process_name	Atom	-	Oui	Nom du processus pour l'enregistrement
map_client_m3ua.local_ip	Tuple	-	Oui	Adresse IP locale à lier
map_client_m3ua.local_port	Integer	2905	Oui	Port SCTP local
map_client_m3ua.remote_ip	Tuple	-	Oui	Adresse IP STP/SGW distante
map_client_m3ua.remote_port	Integer	2905	Oui	Port SCTP distant
map_client_m3ua.routing_context	Integer	-	Oui	ID de contexte de routage M3UA

Exemple :

```
config :omniss7,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
```

Paramètres du Serveur HTTP

Configuration pour le serveur HTTP de l'API REST.

Paramètre	Type	Par Défaut	Requis	Description
start_http_server	Boolean	true	Non	Activer/désactiver le serveur HTTP (port 8080)

Valeurs Codées en Dur (non configurables) :

- **IP** : 0.0.0.0 (toutes les interfaces)
- **Port** : 8080
- **Transport** : Plug.Cowboy

Exemple :

```
config :omniss7,
  start_http_server: true  # Définir sur false pour désactiver
```

Points de Terminaison de l'API :

- API REST : `http://[server-ip]:8080/api/*`
- Swagger UI : `http://[server-ip]:8080/swagger`
- Métriques Prometheus : `http://[server-ip]:8080/metrics`

Voir [Guide de l'API](#) pour plus de détails.

Paramètres de Base de Données

Configuration pour la persistance de la base de données Mnesia.

Paramètre	Type	Par Défaut	Requis	Description
mnesia_storage_type	Atom	:disc_copies	Non	Type de stockage Mnesia : :disc_copies ou :ram_copies

Exemple :

```
config :omniss7,
  mnesia_storage_type: :disc_copies  # Production
  # mnesia_storage_type: :ram_copies  # Test uniquement
```

Types de Stockage :

- :disc_copies - Stockage persistant sur disque (survit aux redémarrages) - **Recommandé pour la production**
- :ram_copies - Seulement en mémoire (perdu lors du redémarrage) - Pour test uniquement

Tables Mnesia :

- m3ua_peer - Connexions de pairs M3UA
- m3ua_route - Routes de Code de Point
- m3ua_gt_route - Routes de Titre Global

Emplacement : Répertoire Mnesia.{node_name}/

Valeurs Codées en Dur

Les valeurs suivantes sont **codées en dur dans le code source** et ne peuvent pas être modifiées via la configuration.

Délais

Valeur	Impact	Solution de Contournement
Délai de demande MAP : 10 secondes	Toutes les opérations MAP expirent après 10s	Modifier le code source
Délai ISD : 10 secondes	Chaque message ISD expire après 10s	Modifier le code source

Serveur HTTP

Valeur	Impact	Solution de Contournement
IP HTTP : 0.0.0.0	Le serveur écoute sur toutes les interfaces	Modifier le code source
Port HTTP : 8080	L'API REST fonctionne sur le port 8080	Modifier le code source

Vérification SSL

Valeur	Impact	Solution de Contournement
API HLR SSL : désactivé	Vérification SSL toujours désactivée	Modifier le code source
API SMSc SSL : désactivé	Vérification SSL toujours désactivée	Modifier le code source

Intervales d'Enregistrement

Valeur	Impact	Solution de Contournement
Enregistrement frontend : 5 minutes	Le SMS Sc s'enregistre avec le backend toutes les 5 min	Modifier le code source

Actualisation Automatique de l'Interface Web

Page	Intervalle
Gestion du Routage	5 secondes
Abonnés Actifs	2 secondes

Exemples de Configuration

Configuration HLR Minimale

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,

  hlr_api_base_url: "https://10.180.2.140:8443",
  hlr_service_center_gt_address: "55512341111",
  smsc_service_center_gt_address: "55512341112",

  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
```

Configuration SMS Sc Minimale

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,

  smsc_api_base_url: "https://10.179.3.219:8443",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112",

  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smss: "ipsmgw",
  auto_flush_tps: 10,

  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :stp_client_asp,
    local_ip: {10, 179, 4, 12},
```

```

    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
}

```

STP avec Serveur Autonome

```

config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: false,

  enable_gt_routing: true,
  mnesia_storage_type: :disc_copies,

  m3ua_stp: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 2905,
    point_code: 100
  },
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :stp_client_asp,
    local_ip: {10, 179, 4, 10},
    local_port: 2906,
    remote_ip: {10, 179, 4, 11},
    remote_port: 2905,
    routing_context: 1
  }
}

```

Résumé

Total des Paramètres de Configuration : 32

Par Catégorie :

- Mode Opérationnel : 3 paramètres
- Mode HLR : 13 paramètres
- Mode SMS : 7 paramètres
- Mode STP : 5 paramètres
- Connexion M3UA : 8 paramètres
- Serveur HTTP : 1 paramètre
- Base de Données : 1 paramètre

Paramètres Requis (doivent être définis) :

- `hlr_api_base_url` (mode HLR)
- `hlr_service_center_gt_address` (mode HLR)
- `smsc_api_base_url` (mode SMS)
- `smsc_service_center_gt_address` (mode SMS/HLR)
- Tous les paramètres `map_client_m3ua.*`
- `m3ua_stp.point_code` (si STP activé)

Documentation Connexe

- [**Guide HLR**](#) - Configuration spécifique au HLR
- [**Guide SMSc**](#) - Configuration spécifique au SMSc
- [**Guide STP**](#) - Configuration de routage STP
- [**Guide de l'API**](#) - Référence de l'API REST
- [**Guide de l'Interface Web**](#) - Documentation de l'interface web



Guide de NAT de Titre Global

Vue d'ensemble

La traduction d'adresse de titre global (GT NAT) est une fonctionnalité qui permet à OmniSS7 de répondre avec différentes adresses de titre global en fonction du préfixe GT de la partie appelante, du préfixe GT de la partie appelée, ou d'une combinaison des deux. Cela est essentiel lors de l'opération avec plusieurs titres globaux et de la nécessité de s'assurer que les réponses utilisent le bon GT en fonction du réseau ou du pair qui appelle et/ou du GT qu'ils ont appelé.

Quoi de neuf (GT NAT amélioré)

La fonctionnalité GT NAT a été améliorée avec de nouvelles capacités puissantes :

Nouvelles fonctionnalités

1. **Correspondance de préfixe de partie appelée** : Les règles peuvent désormais correspondre sur `called_prefix` en plus de `calling_prefix`
2. **Correspondance combinée** : Les règles peuvent correspondre à la fois sur les préfixes appelants ET appelés simultanément
3. **Priorisation basée sur le poids** : Les règles utilisent désormais un champ `weight` (plus bas = priorité plus élevée) au lieu de simplement la longueur du préfixe
4. **Correspondance flexible** : Vous pouvez désormais créer des règles avec :
 - Seulement le préfixe appelant
 - Seulement le préfixe appelé
 - Les préfixes appelants et appelés
 - Aucun (règle de secours)

Nouveau format de règle

Champs requis :

- `weight` : Priorité entière (plus bas = priorité plus élevée)
- `response_gt` : Le GT à utiliser pour la réponse

Champs optionnels (au moins un recommandé pour une correspondance spécifique) :

- `calling_prefix` : Correspondre au préfixe GT de la partie appelante
- `called_prefix` : Correspondre au préfixe GT de la partie appelée

Exemple :

```
gt_nat_rules: [
    # Règle spécifique avec les deux préfixes - priorité la plus élevée
    %{calling_prefix: "8772", called_prefix: "555", weight: 1,
response_gt: "111111"},

    # Règles spécifiques - priorité moyenne
    %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, 
    %{called_prefix: "555", weight: 10, response_gt: "333333"},

    # Règle de secours wildcard - priorité la plus basse
    %{weight: 100, response_gt: "999999"}
]
```

Cas d'utilisation

Opération multi-réseaux

Lorsque vous avez plusieurs réseaux pairs et que chacun s'attend à des réponses d'un GT spécifique :

- **Réseau A** appelle votre GT 111111 et s'attend à des réponses de 111111
- **Réseau B** appelle votre GT 222222 et s'attend à des réponses de 222222

Sans GT NAT, vous auriez besoin d'instances séparées ou de routage complexe. Avec GT NAT, une seule instance d'OmniSS7 peut gérer cela intelligemment.

Scénarios de roaming

Lorsque vous opérez en tant qu'HLR ou SMSc avec des accords de roaming :

- Les abonnés du **réseau domestique** utilisent le GT 555000
- Le **partenaire de roaming 1** utilise le GT 555001
- Le **partenaire de roaming 2** utilise le GT 555002

GT NAT garantit que chaque partenaire reçoit des réponses du bon GT auquel ils sont configurés pour router.

Tests et migration

Lors des migrations de réseau ou des tests :

- Migrer progressivement le trafic de l'ancien GT vers le nouveau GT
- Maintenir les deux GT pendant la période de transition
- Router les réponses en fonction du GT utilisé par l'appelant

Comment ça fonctionne

Flux de traduction d'adresse

1. **Demande entrante** : OmniSS7 reçoit un message SCCP avec :
 - GT de la partie appelée : 55512341112 (votre GT)
 - GT de la partie appelante : 877234567 (leur GT)
2. **Recherche GT NAT** : Le système vérifie le GT appelant 877234567 par rapport aux règles de préfixe configurées
3. **Correspondance de préfixe** : Trouve le préfixe correspondant le plus long (par exemple, 8772 correspond à 877234567)
4. **Sélection du GT de réponse** : Utilise response_gt de la règle correspondante (par exemple, 55512341112)
5. **Réponse envoyée** : La réponse SCCP utilise :
 - GT de la partie appelée : 877234567 (inversé - leur GT)
 - GT de la partie appelante : 55512341112 (GT NAT)

Types de réponses affectés

GT NAT s'applique à plusieurs couches de la pile SS7 :

Couche SCCP (Toutes les réponses)

- Adresses GT appelées/appelantes SCCP dans tous les messages de réponse
- Accusés de réception ISD (InsertSubscriberData)
- Réponses UpdateLocation
- Réponses d'erreur

Couche MAP (Opération spécifique)

- **Réponses SRI-for-SM** : networkNode-Number (adresse GT SMSc)
- **UpdateLocation** : hlr-Number dans les réponses
- **InsertSubscriberData** : GT HLR dans les messages ISD

Configuration

Configuration de base

Ajoutez à config/runtime.exs :

```

config :omniss7,
  # Activer GT NAT
  gt_nat_enabled: true,

  # Définir les règles GT NAT
  gt_nat_rules: [
    # Règle 1 : Les appels du préfixe "8772" obtiennent une réponse
    du "55512341112"
    %{calling_prefix: "8772", response_gt: "55512341112"},

    # Règle 2 : Les appels du préfixe "8773" obtiennent une réponse
    du "55512341111"
    %{calling_prefix: "8773", response_gt: "55512341111"},

    # Règle par défaut (préfixe vide correspond à tout)
    %{calling_prefix: "", response_gt: "55512311555"}
  ]

```

Paramètres de configuration

Pour une référence complète de configuration, voir [Paramètres de Titre Global NAT dans la Référence de Configuration](#).

Paramètre	Type	Requis	Description
gt_nat_enabled	Booléen	Oui	Activer/désactiver la fonctionnalité GT NAT
gt_nat_rules	Liste de Maps	Oui (si activé)	Liste des règles de correspondance de préfixes

Format de règle

Chaque règle est une carte avec les clés suivantes :

```

%{
  calling_prefix: "8772",      # (Optionnel) Préfixe à faire
correspondre avec le GT appelant
  called_prefix: "555",        # (Optionnel) Préfixe à faire
correspondre avec le GT appelé
  weight: 10,                  # (Requis) Valeur de priorité (plus
bas = priorité plus élevée)
  response_gt: "55512341112"  # (Requis) GT à utiliser dans les
réponses
}

```

Champs de règle :

- **calling_prefix** (Optionnel) : Préfixe de chaîne à faire correspondre avec le GT appelant entrant

- La correspondance se fait par `String.startsWith?/2`
- La chaîne vide "" ou `nil` agit comme un wildcard (correspond à n'importe quel GT appelant)
- Peut être omis pour correspondre à n'importe quel GT appelant
- **called_prefix** (Optionnel) : Préfixe de chaîne à faire correspondre avec le GT appelé entrant
 - La correspondance se fait par `String.startsWith?/2`
 - La chaîne vide "" ou `nil` agit comme un wildcard (correspond à n'importe quel GT appelé)
 - Peut être omis pour correspondre à n'importe quel GT appelé
- **weight** (Requis) : Valeur de priorité entière
 - Poids inférieur = priorité plus élevée (traité en premier)
 - Doit être ≥ 0
 - Utilisé comme critère de tri principal pour les règles de correspondance
- **response_gt** (Requis) : L'adresse de Titre Global à utiliser dans les réponses
 - Doit être une chaîne de numéro valide E.164
 - Doit correspondre à l'un de vos GT configurés

Au moins l'un de `calling_prefix` ou `called_prefix` doit être spécifié pour un routage spécifique. Les deux peuvent être omis pour une règle wildcard/ de secours.

Logique de correspondance de règle

Les règles sont évaluées par **poids d'abord (croissant), puis par spécificité de préfixe combinée** :

Algorithme de correspondance :

1. Filtrer les règles où tous les préfixes spécifiés correspondent
 - Si `calling_prefix` est défini, il doit correspondre au GT appelant
 - Si `called_prefix` est défini, il doit correspondre au GT appelé
 - Si les deux sont définis, les deux doivent correspondre
 - Si aucun n'est défini, la règle agit comme un wildcard
2. Trier les règles correspondantes par :
 - **Primaire** : Poids (croissant - valeurs plus basses d'abord)
 - **Secondaire** : Longueur de préfixe combinée (décroissante - plus long = plus spécifique)
3. Retourner la première règle correspondante

Exemples :

```

# Exemples de règles
gt_nat_rules: [
    # Poids 1 : Priorité la plus élevée - correspond aux deux préfixes
    %{calling_prefix: "8772", called_prefix: "555", weight: 1,
response_gt: "111111"},

    # Poids 10 : Priorité moyenne - règles spécifiques
    %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, #
Appelant seulement
    %{called_prefix: "555", weight: 10, response_gt: "333333"}, #
Appelé seulement

    # Poids 100 : Priorité la plus basse - wildcard de secours
    %{weight: 100, response_gt: "444444"} # Correspond à tout
]

# Exemples de correspondance :
# Appelant : "877234567", Appelé : "555123" -> "111111" (poids 1, les
deux correspondent)
# Appelant : "877234567", Appelé : "999999" -> "222222" (poids 10,
appelant seulement)
# Appelant : "999999999", Appelé : "555123" -> "333333" (poids 10,
appelé seulement)
# Appelant : "999999999", Appelé : "888888" -> "444444" (poids 100,
wildcard)

```

Exemples

Exemple 1 : Deux partenaires réseau

Scénario : Vous exploitez un SMS avec deux partenaires réseau. Chacun s'attend à des réponses d'un GT différent.

```

config :omniss7,
    gt_nat_enabled: true,

    # GT SMS par défaut (utilisé lorsque GT NAT est désactivé ou
aucune règle ne correspond)
    smsc_service_center_gt_address: "5551000",

    # Règles GT NAT pour les partenaires
    gt_nat_rules: [
        # Partenaire A (préfixe 4412) s'attend à des réponses du GT
        5551001
        %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},

        # Partenaire B (préfixe 4413) s'attend à des réponses du GT
        5551002
    ]

```

```

    %{calling_prefix: "4413", weight: 10, response_gt: "5551002"},

    # Par défaut : utiliser le GT SMSc standard (wildcard de secours)
    %{weight: 100, response_gt: "5551000"}
]

```

Flux de trafic :

SRI-for-SM entrant de 44121234567 :
 GT appelé : 5551001 (votre GT que le Partenaire A utilise)
 GT appelant : 44121234567 (GT du Partenaire A)

Recherche GT NAT :
 "44121234567" correspond au préfixe "4412"
 GT de réponse sélectionné : "5551001"

Réponse SRI-for-SM à 44121234567 :
 GT appelé : 44121234567 (inversé)
 GT appelant : 5551001 (NAT'd)
 networkNode-Number : 5551001 (dans la réponse MAP)

Exemple 2 : HLR avec GT régionaux

Scénario : HLR national avec différents GT par région.

```

config :omniss7,
  gt_nat_enabled: true,
  hlr_service_center_gt_address: "555000", # GT HLR par défaut

  gt_nat_rules: [
    # VLRs de la région nord (préfixe 5551)
    %{calling_prefix: "5551", weight: 10, response_gt: "555100"},

    # VLRs de la région sud (préfixe 5552)
    %{calling_prefix: "5552", weight: 10, response_gt: "555200"},

    # VLRs de la région ouest (préfixe 5553)
    %{calling_prefix: "5553", weight: 10, response_gt: "555300"},

    # Par défaut pour d'autres régions (wildcard)
    %{weight: 100, response_gt: "555000"}
]

```

Exemple 3 : Scénario de migration

Scénario : Migration progressive de l'ancien GT vers le nouveau GT.

```
config :omniss7,
```

```

gt_nat_enabled: true,
  hlr_service_center_gt_address: "123456789", # Ancien GT (par
défaut)

  gt_nat_rules: [
    # Réseaux migrés (ont déjà mis à jour leurs configurations)
    %{calling_prefix: "555", weight: 10, response_gt: "987654321"},

# Nouveau GT
    %{calling_prefix: "666", weight: 10, response_gt: "987654321"},

# Nouveau GT

    # Tout le monde utilise encore l'ancien GT (wildcard)
    %{weight: 100, response_gt: "123456789"} # Ancien GT
  ]

```

Exemple 4 : Correspondance de préfixe de partie appelée (NOUVEAU)

Scénario : Vous avez plusieurs GT pour différents services et souhaitez répondre avec le bon GT en fonction de quel GT a été appelé.

```

config :omniss7,
  gt_nat_enabled: true,

  gt_nat_rules: [
    # Lorsqu'ils appellent votre GT SMS (5551xxx), répondez avec ce
GT
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},

    # Lorsqu'ils appellent votre GT Voix (5552xxx), répondez avec ce
GT
    %{called_prefix: "5552", weight: 10, response_gt: "555200"},

    # Lorsqu'ils appellent votre GT Données (5553xxx), répondez avec
ce GT
    %{called_prefix: "5553", weight: 10, response_gt: "555300"},

    # Règle de secours par défaut
    %{weight: 100, response_gt: "555000"}
  ]

```

Flux de trafic :

Demande entrante au GT appelé : 555100 (votre GT SMS)
GT appelant : 441234567 (n'importe quel appelant)

Recherche GT NAT :

GT appelé "555100" correspond au préfixe "5551"

```
GT de réponse sélectionné : "555100"
```

```
La réponse utilise le GT appelant : 555100 (correspond à ce qu'ils  
ont appelé)
```

Exemple 5 : Correspondance combinée de préfixes appellants + appelés (AVANCÉE)

Scénario : Différents partenaires appellent différents GT, et vous souhaitez un contrôle précis.

```
config :omniss7,  
  gt_nat_enabled: true,  
  
  gt_nat_rules: [  
    # Partenaire A appelant votre GT SMS - priorité la plus élevée  
(poids 1)  
    %{calling_prefix: "4412", called_prefix: "5551", weight: 1,  
response_gt: "555101"},  
  
    # Partenaire B appelant votre GT SMS - priorité la plus élevée  
(poids 1)  
    %{calling_prefix: "4413", called_prefix: "5551", weight: 1,  
response_gt: "555102"},  
  
    # Quiconque appelant votre GT SMS - priorité moyenne (poids 10)  
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},  
  
    # Partenaire A appelant n'importe quel GT - priorité moyenne  
(poids 10)  
    %{calling_prefix: "4412", weight: 10, response_gt: "555200"},  
  
    # Règle de secours par défaut - faible priorité (poids 100)  
    %{weight: 100, response_gt: "555000"}  
  ]
```

Exemples de correspondance :

```
# Partenaire A appelle le GT SMS  
Appelant : "441234567", Appelé : "555100"  
→ Correspond à la règle de poids 1 (les deux préfixes) → "555101"
```

```
# Partenaire A appelle le GT Voix  
Appelant : "441234567", Appelé : "555200"  
→ Correspond à la règle de poids 10 (appelant seulement) → "555200"
```

```
# Appelant inconnu appelle le GT SMS  
Appelant : "999999999", Appelé : "555100"
```

```

→ Correspond à la règle de poids 10 (appelé seulement) → "555100"
# Appelant inconnu appelle le GT Voix
Appelant : "999999999", Appelé : "555200"
→ Correspond à la règle de poids 100 wildcard → "555000"

```

Modes opérationnels

GT NAT fonctionne dans tous les modes opérationnels d'OmniSS7 :

Mode HLR

GT NAT affecte :

- Réponses UpdateLocation (GT HLR dans la réponse)
- Messages InsertSubscriberData (GT HLR en tant que partie appelante)
- Réponses SendAuthenticationInfo
- Réponses Cancel Location

Pour plus d'informations sur les opérations HLR, voir le [Guide de Configuration HLR](#).

Configuration :

```

config :omniss7,
  hlr_mode_enabled: true,
  hlr_service_center_gt_address: "5551234567",  # GT HLR par défaut

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "331", weight: 10, response_gt: "5551234568"},

# France
    %{calling_prefix: "44", weight: 10, response_gt: "5551234569"},

# Royaume-Uni
    %{weight: 100, response_gt: "5551234567"}  # Wildcard par défaut
  ]

```

Mode SMSc

GT NAT affecte :

- Réponses SRI-for-SM (champ networkNode-Number) - voir [Détails SRI-for-SM](#)
- Accusés de réception MT-ForwardSM

Pour plus d'informations sur les opérations SMSc, voir le [Guide de Configuration SMSc](#).

Configuration :

```
config :omniss7,
  smsc_mode_enabled: true,
  smsc_service_center_gt_address: "5559999",  # GT SMS Sc par défaut

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "1", weight: 10, response_gt: "5559991"},  #
  Amérique du Nord
    %{calling_prefix: "44", weight: 10, response_gt: "5559992"},  #
  Royaume-Uni
    %{calling_prefix: "86", weight: 10, response_gt: "5559993"},  #
  Chine
    %{weight: 100, response_gt: "5559999"} # Wildcard par défaut
  ]
```

Mode Passerelle CAMEL

GT NAT affecte :

- Toutes les réponses au niveau SCCP (GT gsmSCF en tant que partie appelante)
- Réponses des opérations CAMEL/CAP (InitialDP, EventReportBCSM, etc.)
- Accusés de réception RequestReportBCSMEvent
- Réponses ApplyCharging
- Réponses Continue

Configuration :

```
config :omniss7,
  camelgw_mode_enabled: true,
  camel_gsmSCF_gt_address: "55512341112",  # GT gsmSCF par défaut

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "555", weight: 10, response_gt: "55512341111"},  #
  # Réseau A
    %{calling_prefix: "666", weight: 10, response_gt: "55512311555"},  #
  # Réseau B
    %{weight: 100, response_gt: "55512341112"} # Wildcard par défaut
  ]
```

Cas d'utilisation : Lorsque vous opérez en tant que gsmSCF (Service Control Function) pour plusieurs réseaux, chaque gsmSSF de réseau peut s'attendre à des réponses d'un GT gsmSCF spécifique. GT NAT garantit que le bon GT est utilisé en fonction de quel gsmSSF appelle.

Journalisation et Débogage

Activer la journalisation GT NAT

GT NAT inclut la journalisation automatique de toutes les traductions :

```
# Dans les journaux, vous verrez :  
[info] GT NAT [réponse SRI-for-SM] : GT appelant 877234567 -> GT de  
réponse 55512341112  
[info] GT NAT [UpdateLocation ISD] : GT appelant 331234567 -> GT de  
réponse 55512341111  
[info] GT NAT [réponse MAP BEGIN] : GT appelant 441234567 -> GT de  
réponse 55512311555
```

Le champ de contexte montre où le NAT a été appliqué :

- "réponse SRI-for-SM" - Dans le gestionnaire SRI-for-SM
- "UpdateLocation ISD" - Dans les messages InsertSubscriberData
- "UpdateLocation END" - Dans la réponse UpdateLocation END
- "réponse MAP BEGIN" - Réponses MAP BEGIN génériques
- "ISD ACK" - Accusé de réception ISD
- "réponse d'erreur HLR" - Réponse d'erreur de l'HLR
- "réponse CAMEL" - Réponses d'opérations CAMEL/CAP (gsmSCF)

Validation

Le système valide la configuration GT NAT au démarrage :

```
# Vérifier la config GT NAT  
iex> GtNat.validate_config()  
{:ok, [  
  %{calling_prefix: "8772", weight: 10, response_gt: "55512341112"},  
  %{calling_prefix: "8773", weight: 10, response_gt: "55512341111"}  
]}
```



```
# Vérifier si activé  
iex> GtNat.enabled?()  
true
```



```
# Obtenir toutes les règles  
iex> GtNat.get_rules()  
[  
  %{calling_prefix: "8772", weight: 10, response_gt: "55512341112"},  
  %{calling_prefix: "8773", weight: 10, response_gt: "55512341111"}  
]
```

Tester GT NAT

Testez la logique GT NAT de manière programmatique :

```
# Tester la traduction avec uniquement le GT appelant (called_gt est nil)
iex> GtNat.translate_response_gt("877234567", nil, "default_gt")
"55512341112"

# Tester la traduction avec les deux GT appelant et appelé
iex> GtNat.translate_response_gt("877234567", "555123", "default_gt")
"55512341112"

# Tester avec journalisation (GT appelé nil)
iex> GtNat.translate_response_gt_with_logging("877234567", nil,
"default_gt", "test")
# Journaux : GT NAT [test] : GT appelant 877234567 -> GT de réponse
55512341112
"55512341112"

# Tester avec journalisation (les deux GT)
iex> GtNat.translate_response_gt_with_logging("877234567", "555123",
"default_gt", "test")
# Journaux : GT NAT [test] : GT appelant 877234567, GT appelé 555123
-> GT de réponse 55512341112
"55512341112"

# Tester sans correspondance (retourne par défaut)
iex> GtNat.translate_response_gt("999999999", "888888", "default_gt")
"default_gt"
```

Dépannage

Problème : GT NAT ne fonctionne pas

Vérification 1 : Est-il activé ?

```
iex> Application.get_env(:omniss7, :gt_nat_enabled)
true # Devrait être vrai
```

Vérification 2 : Les règles sont-elles configurées ?

```
iex> Application.get_env(:omniss7, :gt_nat_rules)
[%{calling_prefix: "8772", response_gt: "55512341112"}, ...] # Devrait retourner une liste
```

Vérification 3 : Vérifiez les journaux

Recherchez "GT NAT" dans les journaux pour voir si des traductions se produisent.

Problème : Mauvais GT dans les réponses

Symptôme : Les réponses utilisent une adresse GT inattendue

Cause : La correspondance de préfixe de règle pourrait être trop large ou la règle par défaut attrape le trafic

Solution : Révisez les poids et les préfixes des règles :

```
# MAUVAIS : Wildcard avec un poids bas (attrape tout en premier)
gt_nat_rules: [
    %{weight: 1, response_gt: "111111"},                      # Cela
correspond à tout en premier !
    %{calling_prefix: "8772", weight: 10, response_gt: "222222"} # 
Jamais atteint
]

# BON : Règles spécifiques avec un poids inférieur, wildcard avec un
poids plus élevé
gt_nat_rules: [
    %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, # 
Spécifique, poids bas
    %{weight: 100, response_gt: "111111"} # Wildcard, poids élevé
(secours)
]
```

Problème : GT NAT non appliqué à un type de message spécifique

Symptôme : Certaines réponses utilisent le GT NAT'd, d'autres non

Couverture actuelle :

- ◊ GT appelant SCCP (toutes les réponses)
- ◊ Réponses SRI-for-SM (networkNode-Number)
- ◊ Messages ISD UpdateLocation (GT HLR)
- ◊ Réponses UpdateLocation END
- ◊ Accusés de réception ISD
- ◊ Réponses MAP BEGIN

Si un type de message spécifique n'utilise pas GT NAT, il se peut qu'il ne soit pas encore implémenté. Vérifiez le code source ou contactez le support.

Considérations de performance

Performance de recherche

GT NAT utilise une correspondance de préfixe simple avec une complexité $O(n)$ où n est le nombre de règles.

Conseils de performance :

- Gardez le nombre de règles en dessous de 100 pour la meilleure performance
- Utilisez des préfixes spécifiques pour réduire le nombre de règles
- La règle par défaut (préfixe vide) doit être la dernière

Benchmark (système typique) :

- 10 règles : $< 1\mu\text{s}$ par recherche
- 50 règles : $< 5\mu\text{s}$ par recherche
- 100 règles : $< 10\mu\text{s}$ par recherche

Utilisation de la mémoire

Chaque règle nécessite ~ 100 octets de mémoire :

- 10 règles $\approx 1\text{ Ko}$
- 100 règles $\approx 10\text{ Ko}$

Meilleures pratiques

1. Inclure toujours une règle de secours wildcard

```
gt_nat_rules: [
  %{calling_prefix: "8772", weight: 10, response_gt: "111111"}, 
  %{calling_prefix: "8773", weight: 10, response_gt: "222222"}, 
  %{weight: 100, response_gt: "default_gt"} # Avoir toujours un
wildcard avec un poids élevé
]
```

2. Utiliser des préfixes et des poids significatifs

```
# BON : Préfixes clairs et spécifiques avec des poids appropriés
%{calling_prefix: "331", weight: 10, response_gt: "..." } # France
%{calling_prefix: "44", weight: 10, response_gt: "..." } # Royaume- 
Uni

# MAUVAIS : Préfixes trop larges ou poids déroutants
%{calling_prefix: "3", weight: 5, response_gt: "..." }      # Trop de
```

```
pays
%{calling_prefix: "331", weight: 100, response_gt: "..."} # Le poids
devrait être plus bas pour les règles spécifiques
```

3. Documentez vos règles

```
gt_nat_rules: [
    # Partenaire XYZ - réseau britannique (plage GT : 4412xxxxxx)
    # Poids 10 : Priorité standard du partenaire
    %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},

    # Partenaire ABC - réseau français (plage GT : 33123xxxxx)
    # Poids 10 : Priorité standard du partenaire
    %{calling_prefix: "33123", weight: 10, response_gt: "5551002"}
]
```

4. Testez avant le déploiement

```
# Testez dans iex avant de déployer
iex> GtNat.translate_response_gt("44121234567", nil, "default")
"5551001" # Résultat attendu

# Testez avec le GT appelé
iex> GtNat.translate_response_gt("44121234567", "555123", "default")
"5551001" # Résultat attendu
```

5. Surveillez les journaux

Activez la journalisation de niveau INFO pour voir toutes les traductions GT NAT en production.

Intégration avec d'autres fonctionnalités

Mode STP

GT NAT fonctionne indépendamment du routage STP. Le STP route en fonction des codes de point et des GT de destination, tandis que GT NAT gère l'adressage des réponses.

Pour plus d'informations sur le routage STP, voir le [Guide de Configuration STP](#).

Intégration CAMEL

GT NAT est **entièrement intégré** avec les opérations CAMEL/CAP :

Couche SCCP :

- GT de la partie appelante dans toutes les réponses CAMEL
- Appliqué automatiquement en fonction du GT gsmSSF entrant

Configuration :

- `camel_gsmSCF_gt_address` - GT gsmSCF par défaut (optionnel)
- Si non configuré, utilise le GT de la partie appelée de la demande entrante
- Les règles GT NAT remplacent le défaut en fonction du préfixe de la partie appelante

Exemple :

```
# Lorsque gsmSSF 555123456 appelle votre gsmSCF
# Entrant : Appelé=55512341112, Appelant=555123456
# Recherche GT NAT : "555" -> response_gt="55512341111"
# Réponse : Appelé=555123456, Appelant=55512341111
```

Équilibrage de charge

GT NAT peut être combiné avec l'équilibrage de charge M3UA pour une gestion avancée du trafic.

Guide de migration

Activer GT NAT sur un système existant

1. Préparer la configuration

```
# Ajouter à runtime.exs (garder désactivé au départ)
config :omniss7,
  gt_nat_enabled: false, # Commencer désactivé
  gt_nat_rules: [
    # Vos règles ici avec des poids
    %{calling_prefix: "877", weight: 10, response_gt: "111111"}, # Wildcard de secours
    %{weight: 100, response_gt: "999999"} # Wildcard de secours
  ]
```

2. Tester la configuration

```
# Valider que la config compile
mix compile

# Tester dans iex
iex -S mix
iex> GtNat.validate_config()
```

3. Activer en staging

```
gt_nat_enabled: true # Changer à vrai
```

4. Surveiller les journaux

```
tail -f log/omniss7.log | grep "GT NAT"
```

5. Déployer en production

- Déployer pendant la fenêtre de maintenance
- Surveiller les 24 premières heures de près
- Avoir un plan de retour prêt (définir gt_nat_enabled: false)

Support

Pour des problèmes ou des questions :

- Vérifiez les journaux pour les messages "GT NAT"
- Validez la config avec `GtNat.validate_config()`
- Consultez la section de dépannage de ce guide
- Contactez le support OmniSS7 avec des extraits de journaux

Guide de Configuration HLR

[Retour à la Documentation Principale](#)

Ce guide fournit la configuration pour utiliser OmniSS7 comme un **Home Location Register (HLR/HSS)** avec **OmniHSS** comme base de données d'abonnés en backend.

Intégration OmniHSS

Le mode HLR d'OmniSS7 fonctionne comme un **frontend de signalisation SS7** qui interagit avec **OmniHSS**, un serveur d'abonnés à domicile (HSS) complet. Cette architecture sépare les préoccupations :

- **OmniSS7 (Frontend HLR)** : Gère toute la signalisation du protocole SS7/MAP, le routage SCCP et la communication réseau
- **OmniHSS (Backend HSS)** : Gère les données des abonnés, l'authentification, le provisioning et les fonctionnalités avancées

Pourquoi OmniHSS ?

OmniHSS fournit une gestion des abonnés de qualité opérateur avec des fonctionnalités incluant :

- **Support Multi-IMSI** : Chaque abonné peut avoir plusieurs IMSI associés à un seul MSISDN pour le roaming international, le changement de réseau et le provisioning eSIM
- **Authentification Multi-Profile** : Soutien pour les appareils avec plusieurs profils (3GPP, GPRS et COMP128)
- **Suivi des Session Circuits & Paquets** : Suivi indépendant des enregistrements réseau CS (circuit-switched) et PS (packet-switched)
- **Provisioning Avancé** : Profils de service personalisables, services supplémentaires et données d'abonnement CAMEL
- **Conception API-First** : API HTTP RESTful pour l'intégration avec les systèmes de facturation, CRM et provisioning
- **Mises à Jour en Temps Réel** : Suivi de localisation, gestion de session et génération de vecteurs d'authentification

Toutes les données des abonnés, les informations d'authentification et les configurations de service sont stockées et gérées dans OmniHSS. OmniSS7 interroge OmniHSS via des appels API HTTPS pour répondre aux opérations MAP comme `UpdateLocation`, `SendAuthenticationInfo` et `SendRoutingInfo`.

Important : Le mode HLR d'OmniSS7 est un **frontend de signalisation uniquement**. Toute la logique de gestion des abonnés, les algorithmes d'authentification, les règles de provisioning et les opérations de base de données sont gérées par OmniHSS. Ce guide couvre la configuration du protocole SS7/MAP dans OmniSS7. Pour des informations sur le provisioning des abonnés, la configuration de l'authentification, les profils de service et les opérations administratives, [reportez-vous à la documentation d'OmniHSS](#).

Support Multi-IMSI

OmniHSS prend en charge nativement les configurations Multi-IMSI, permettant à un seul abonné (identifié par MSISDN) d'avoir plusieurs IMSI. Cela permet :

- **Profils de Roaming International** : Différents IMSI pour différentes régions afin de réduire les coûts de roaming
- **eSIM Multi-Profile** : Plusieurs profils réseau sur un seul appareil compatible eSIM
- **Changement de Réseau** : Changement transparent entre les réseaux sans changer de MSISDN
- **Coordination Dual SIM** : Coordination entre plusieurs SIM physiques ou virtuelles
- **Tests & Développement** : Plusieurs IMSI de test pointant vers le même abonné

Comment ça fonctionne :

- Chaque IMSI a ses propres informations d'authentification (Ki, OPC, algorithme)
- Chaque IMSI peut avoir des enregistrements de session circuits et paquets indépendants
- Les services et profils des abonnés peuvent être partagés ou personnalisés par IMSI
- OmniSS7 interroge OmniHSS par IMSI, et OmniHSS retourne les données appropriées de l'abonné
- Les systèmes de facturation peuvent suivre l'utilisation par IMSI tout en associant tous les IMSI à un seul compte

Exemple de scénario Multi-IMSI :

Abonné MSISDN : 41-335-123-4567
└─ IMSI 1 : 3102601234567890 (Réseau National US - Authentication Milenage)
└─ IMSI 2 : 2880811234567890 (Profil de Roaming France - Authentication Milenage)
└─ IMSI 3 : 440281234567891 (Profil de Roaming UK - Authentication COMP128)

Tous les trois IMSI peuvent être utilisés indépendamment pour l'enregistrement réseau, mais ils appartiennent tous au même compte d'abonné. OmniHSS gère la correspondance IMSI-abonné et assure une authentification et un provisioning appropriés pour chaque IMSI.

OmniSS7 Stack v1.0.0

SS7 Events

SS7 Client

M3UA

HLR Links

Active Subscribers

Resources

Configuration

Active Subscribers

Active Subscribers: Subscribers who have sent an `UpdateLocation` request and have not yet sent a `CancelLocation`. This table shows the IMSI, current VLR, MSC, registration time, and duration for each active subscriber.

Total: 0 Auto-refresh every 2 seconds

IMSI	VLR Number	MSC Number	Updated At	Duration
No Active Subscribers				
Subscribers will appear here when they send <code>UpdateLocation</code> requests.				

OmniTouch

Table des Matières

1. [Intégration OmniHSS](#)
 2. [Support Multi-IMSI](#)
 3. [Qu'est-ce que le Mode HLR ?](#)
 4. [Activité du Mode HLR](#)
 5. [Bases de Données des Abonnés](#)
 6. [Vecteurs d'Authentification](#)
 7. [Mises à Jour de Localisation](#)
 8. [Intégration CAMEL](#)
 9. [Gestion des Abonnés en Roaming](#)
 10. [Opérations HLR](#)
- [Mapping des champs de Réponse](#)
 - `GetRoutingInfoForMSI`
 - `UpdateLocation / ISDN`
 - `SendRoutingInfoForMS`
 - [Résumé des Sources de Champs](#)

Qu'est-ce que le Mode HLR ?

Le Mode HLR permet à OmniSS7 de fonctionner comme un Home Location Register pour :

- **Gestion des Abonnés** : Stocker et gérer les données des abonnés
- **Authentification** : Générer des vecteurs d'authentification pour l'accès au réseau
- **Suivi de Localisation** : Traiter les mises à jour de localisation provenant des VLR
- **Informations de Routage** : Fournir des informations de routage pour les appels et les SMS

Architecture HLR

Omniss7 peut fonctionner en différents modes. Pour l'utiliser comme HLR, vous devez activer le mode HLR dans la configuration.

Activation du Mode HLR

Le fichier config/runtime.exe d'Omniss7 contient trois modes opérationnels préconfigurés. Pour activer le mode HLR :

1. **Ouvrez config/runtime.exe**
2. **Trouvez les trois sections de configuration (lignes 53-174) :**
 - Configuration 1 : Mode STP (lignes 53-85)
 - Configuration 2 : Mode HLR (lignes 86-123)
 - Configuration 3 : Mode VLR (lignes 124-174)
3. **Commentez la configuration actuellement active (ajoutez # à chaque ligne)**
4. **Décommentez la configuration HLR (supprimez # des lignes 87-123)**
5. **Personnalisez les paramètres de configuration selon vos besoins**
6. **Redémarrez l'application : iex -S mix**

Configuration du Mode HLR

La configuration HLR complète ressemble à ceci :

```
config :omniss7,
# Flags de mode - Activer uniquement les fonctionnalités HLR
map_client_enabled: true,
hlr_mode_enabled: true,
smsc_mode_enabled: false,
# Configuration de l'API Backend OmniHSS
hlr_api_base_url: "https://10.180.2.140:8443",
# Adresse GT du Centre de Service HLR pour les opérations SMS
hlr_service_center_gt_address: "1234567890",
# Configuration de Mapping MSISDN - IMSI
# Voir la section Mapping MSISDN - IMSI pour les détails
hlr_imsi_plmn_prefix: "50557",
hlr_msisdn_country_code: "61",
hlr_msisdn_nsn_offset: 0,
hlr_msisdn_nsn_length: 9,
# Configuration InsertSubscriberData
# Mode d'Accès Réseau : packetAndCircuit, :packetOnly, ou :circuitOnly
isd_network_access_mode: :packetAndCircuit,
# Envoyer ISD #2 (données de Services Supplémentaires)
isd_send_ss_data: true,
# Envoyer ISD #3 (données de Barrage d'Appels)
isd_send_call_barring: true,
# Configuration CAMEL (pour les réponses SendRoutingInfo)
# Clé de Service pour l'initialisation de service CAMEL
camel_service_key: 11_11b,
# Point de Détection de Déclenchement CAMEL
# Options : :termAttemptAuthorized, :tBusy, :tNoAnswer, :tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,
# Préfixes VLR à Domicile
# Liste des préfixes d'adresses VLR considérés comme réseau "domestique"
# Si le VLR de l'abonné commence par l'un de ces préfixes, utilisez la réponse SRI standard
# Sinon, l'abonné est en roaming et nous devons envoyer PRN pour obtenir MSRN
home_vlr_prefixes: ["123456"],
# Configuration de Connexion M3UA
# Se connecter en tant qu'ASP pour recevoir les opérations MAP (UpdateLocation, SendAuthInfo, etc.)
map_client_asua: {
  mode: "ASP",
  callback: [MapClient, :handle_payload, []],
  process_name: :hlr_client_asp,
  # Point de terminaison local (système HLR)
  local_ip: {10, 179, 4, 11},
  local_port: 2905,
  # Point de terminaison STP distant
  remote_ip: {10, 179, 4, 10},
  remote_port: 2905,
  routing_context: 1
}

# Configuration de Connexion M3UA
```

The screenshot shows the OmniSS7 Stack interface. In the top right corner, it says "Licensed to: Omnitouch" and "© 2028 Omnitouch". The top bar includes "OmniSS7 Stack" and "1.0.0", "Last updated: 2025-10-26 00:00:54 UTC", and a "Check now" button. The main content area has a dark background with white text. It shows the "HLR API Connection Status" section with a latency of 46.14 ms, a host of https://10.180.2.140:8443, and a note about a self-signed certificate. Below this is the "Configuration" section, which includes "Subscriber Lookup" and "Generate Authentication Vectors". The "Subscriber Lookup" section has a dropdown for "Lookup Type" set to "IMSI" with a value of "e.g. 00101000000001" and a "Lookup" button. The "Generate Authentication Vectors" section has an "IMSI" input field with "e.g. 00101000000001" and a "Generate" button.

Paramètres de Configuration à Personnaliser

Pour une référence complète de tous les paramètres de configuration, voir la [Référence de Configuration](#).

Paramètre	Type	Par Défaut	Description	Exemple
hlr_api_base_url	String	Requis	OmniHSS point d'API backend	"https://10.179.3.219:8443"
hlr_service_center_gt_address	String	Requis	Adresse GT HLR utilisée dans les réponses UpdateLocation	"5551234568"
smsc_service_center_gt_address	String	Requis	Adresse GT SMSC retournée dans les réponses SRI-for-SM	"5551234567"
hlr_smsc_alert_gts	Liste	[]	Liste des GTs SMSC pour envoyer alertServiceCenter après UpdateLocation	["15559876543", "15559876544"]
hlr_alert_location_expiry_seconds	Integer	172800	Temps d'expiration de la localisation (secondes) avant que le SMSC reçoit alertServiceCenter	86400
hlr_imsi_plan_prefix	String	"595357"	Préfixe de plan d'IMSI pour les demandes de NSSN (voir MSISDN -> IMSI Mapping)	"00101000000001"
hlr_msisdn_country_code	String	"61"	Préfixe du code pays pour le mapping inverse IMSI->MSISDN (voir MSISDN -> IMSI Mapping)	"1"
hlr_msisdn_nsn_offset	Integer	0	Décalage dans MSISDN pour l'extraction NSN (voir MSISDN -> IMSI Mapping)	0
hlr_msisdn_nsn_length	Integer	9	Longueur du Numéro d'Abonné National à extraire (voir MSISDN -> IMSI Mapping)	10
isid_network_access_mode	Atom	packetAndCircuit	Mode d'accès réseau pour InsertSubscriberData	:packetOnly
isid_send_ss_data	Boolean	true	Envoyer ISD #2 avec des données de Services Supplémentaires	false
isid_send_call_barring	Boolean	true	Envoyer ISD #3 avec des données de Barrage d'Appels	false
camel_service_key	Integer	11_110	Clé de service CAMEL pour les réponses SendRoutingInfo	100
camel_diameter_detection_point	Atom	2200	Point de détection CAMEL	:easy
home_vlr_detection_point	Liste	["5551231"]	Liste des préfixes d'adresses VLR considérés comme réseau "domestique"	["555123"]
local_ip	Tuple	Requis	Adresse IP de votre système HLR	(10, 179, 4, 12)
local_port	Integer	2905	Port SCTP local	2905
remote_ip	Tuple	Requis	Adresse IP STP pour la connectivité SS7	(10, 179, 4, 10)
remote_port	Integer	2905	Port SCTP distant	2905
routing_context	Integer	1	ID de contexte de routage M3UA	1

Que se Passe-t-il Lorsque le Mode HLR est Actifé

Lorsque `hlr_mode_enabled`: true, l'interface web affichera :

- Événements SS7 - Journalisation des événements
- Client SS7 - Test des opérations MAP
- M3UA - État de la connexion
- Liens HLR - Etat de l'API HLR + gestion des abonnés - Spécifique au HLR
- Ressources - Surveillance du système
- Configurateur - Visualiseur de configuration

Les onglets **Routage**, **Test de Routage** et **Liens SMSC** seront cachés.

Notes Importantes

- Configuration Requise : Le paramètre `hlr_service_center_gt_address` est **obligatoire**. L'application échouera à démarrer s'il n'est pas configuré.
- Backend OmniHSS : L'API OmniHSS doit être accessible à l'URL `hlr_api_base_url` configurée
- Délai d'Attente des Requêtes API : Toutes les requêtes API OmniHSS ont un **délai d'attente codé en dur de 5 secondes**
- Délai d'Attente des Réponses MAP : Toutes les réponses MAP (UpdateLocation, SendAuthInfo, etc.) ont un **délai d'attente codé en dur de 10 secondes**
- Délai d'ISD : Chaque message InsertSubscriberData (ISD) dans une session UpdateLocation a un **délai d'attente codé en dur de 10 secondes**
- La connexion M3UA au STP est requise pour recevoir les opérations MAP
- Après avoir changé de modes, vous devez redémarrer l'application pour que les modifications prennent effet
- Interface Web : Voir le [Guide de l'Interface Web](#) pour des informations sur l'utilisation de l'interface web
- Accès API : Voir le [Guide API](#) pour la documentation de l'API REST et l'accès à Swagger UI

Base de Données des Abonnés

OmniHSS gère toutes les données des abonnés y compris les identités, les informations d'authentification, les profils de service et les informations de localisation. OmniSS7 récupère ces données via des appels API RESTful.

Modèle d'Abonné OmniHSS

OmniHSS stocke des informations complètes sur les abonnés :

- Plusieurs IMSI par abonné : Support pour les configurations Multi-IMSI (eGPRS, profils de roaming, changement de réseau)
- Informations d'authentification : Sélection de Kc, OPe et algorithmes (Milenage ou COMP128)
- Profils de service : Catégorie d'abonné, services autorisés, paramètres QoS
- Suivi de localisation : Suivi indépendant de l'information de localisation actuelle VLR/MSC (session circuit) et SGSN/GGSN (session paquet)
- Données d'abonnement CAMEL : Clés de service, points de déclenchement et adresses gsmSCF
- Services supplémentaires : Transfert d'appels, barrages, attente, configurations CLIP/CLIR
- Etat administratif : Activé/désactivé, restrictions de service, dates d'expiration

Vecteurs d'Authentification

Générer des Vecteurs d'Authentification

OmniHSS génère des vecteurs d'authentification en utilisant les algorithmes Milenage ou COMP128 en fonction de la méthode d'authentification configurée pour chaque abonné. Lorsque OmniSS7 reçoit des requêtes MAP `sendAuthenticationInfo` :

- OmniSS7 extrait l'IMSI de la requête MAP
- OmniSS7 appelle l'API OmniHSS pour générer des vecteurs d'authentification
- OmniHSS récupère les informations d'authentification Ki et OPe de l'abonné
- OmniHSS génère le nombre demandé de vecteurs (RAND, XRES, CK, IK, AUTN)
- OmniSS7 encode les vecteurs au format MAP et les retourne au VLR/GSN demandeur

Intégration de l'API OmniHSS

OmniSS7 communique avec OmniHSS via l'API REST HTTPS pour récupérer des informations sur les abonnés, mettre à jour les données de localisation et générer des vecteurs d'authentification :

```
config :omnis7,
  hlr_api_base_url: "https://omnihss-server:8443"
```

Lorsque OmniSS7 reçoit des opérations MAP du réseau SS7, il interroge OmniHSS pour :

- Récupérer les données de l'abonné par IMSI ou MSISDN
- Générer des vecteurs d'authentification en utilisant les informations KU/OPc stockées
- Mettre à jour la localisation de la session circuit lorsque les abonnés effectuent UpdateLocation
- Vérifier l'état de l'abonné et les droits de service

Mises à Jour de Localisation

Traitement de la Mise à Jour de Localisation

Lors de la réception des requêtes MAP `updateLocation`, OmniSS7 coordonne avec OmniHSS pour enregistrer l'abonné à un nouveau VLR :

1. Extraire les informations de localisation de la requête UpdateLocation (IMSI, nouveau GT VLR, nouveau GT MSC)
2. Interroger OmniHSS pour vérifier que l'abonné existe et est actif
3. Mettre à jour la session circuit dans OmniSS7 et la nouvelle localisation VLR/MSC
4. Envoyer des messages `MSISDN_AlertData` (ISD) pour prévisionner l'abonné au nouveau VLR
5. Retourner la réponse `UpdateLocation` au VLR (incluant le GT HLR de `hlr_service_center_gt_address`)
6. Envoyer `alertServiceCenter` aux GTs SMSc configurés (si `hlr_smc_alert_gts` est peuplé)

Remarque : Le paramètre de configuration `hlr_service_center_gt_address` spécifie le Global Title du HLR qui est retourné dans les réponses `UpdateLocation`. Cela permet au VLR/MSC d'identifier et de router les messages de retour `MSISDN_AlertData` ce HLR.

Intégration du Centre de Service d'Alerte

Après une mise à jour de localisation réussie, le HLR peut automatiquement notifier les systèmes SMSc qu'un abonné est maintenant joignable en envoyant des messages `alertServiceCenter` (opcode MAP 64). Pour des informations sur la façon dont le SMSc gère ces alertes, voir [Gestion du Centre de Service d'Alerte dans le Guide SMSc](#).

Configuration

Configurez la liste des Global Titles SMSc à notifier :

```
config :omnis7,
# Liste des GTs SMSc pour envoyer alertServiceCenter après UpdateLocation
  hlr_smc_alert_gts: [
    "15559876543",
    "15559876544",
  ],
# Temps d'expiration de localisation lorsque le SMSc reçoit alertServiceCenter (par défaut : 48 heures)
  hlr_alert_location_expiry_seconds: 172800
```

Diagramme de Flux

Comportement

Lorsqu'un abonné effectue une mise à jour de localisation :

1. Le HLR envoie `alertServiceCenter` à **chaque** GT SMSc dans la liste `hlr_smc_alert_gts`
2. Le message inclut le MSISDN de l'abonné
3. Le HLR utilise `hlr_service_center_gt_address` comme GT de l'appelant
4. Adressage SCCP : SSN appelant=6 (HLR), SSN appelée=8 (SMSc)

Le SMSc reçoit l'alerte et :

- Supprime le préfixe TON/NFI du MSISDN (par exemple, "19123123213" → "23123213")
- Marque l'abonné comme joignable dans sa base de données de localisation (via POST à `/api/location`)
- Définit le champ `user_agent` sur le GT HLR lors de l'appel à l'API (pour le suivre de quel HLR a envoyé l'alerte)
- Définit le temps d'expiration de localisation basé sur `hlr_alert_location_expiry_seconds`
- Suit l'abonné dans le Tracker d'Abonnés du SMSc pour la surveillance

Tests

Utilisez la page **Abonnés Actifs** dans l'interface Web pour envoyer manuellement des messages `alertServiceCenter` pour les tests :

1. Accédez à l'onglet "Abonnés Actifs"
2. Trouvez la section "Tester le Centre de Service d'Alerte"
3. Entrez MSISDN, GT SMSc et GT HLR (les valeurs par défaut sont pré-remplies à partir de la config)
 - GT SMSc par défaut est la première entrée dans `hlr_smc_alert_gts`
 - GT HLR par défaut est `hlr_service_center_gt_address`
4. Cliquez sur "Envoyer alertServiceCenter"

Ceci est utile pour tester la gestion des alertes SMSc sans nécessiter un flux complet de mise à jour de localisation. Le formulaire utilise la validation `phx-blur` pour éviter d'afficher des erreurs pendant la saisie.

Configuration InsertSubscriberData (ISD)

Après une mise à jour de localisation réussie, le HLR envoie des données de provisioning des abonnés au VLR en utilisant des messages `InsertSubscriberData` (ISD). La configuration ISD vous permet de personnaliser quelles données sont envoyées et comment.

Pour la référence des paramètres de configuration, voir [Configuration ISD dans la Référence de Configuration](#).

Séquence ISD

Le HLR peut envoyer jusqu'à 3 messages ISD séquentiels :

1. **ISD #1** (Toujours envoyé) - Données de base de l'abonné :
 - IMSI
 - MSISDN
 - Catégorie d'abonné
 - Etat de l'abonné (serviceGranted)
 - Liste des services de support
 - Liste des télesservices
 - Mode d'accès réseau
2. **ISD #2 (Optionnel)** - Données des Services Supplémentaires (SS) :
 - Paramètres de transfert d'appels (inconditionnel, occupé, pas de réponse, non joignable)
 - Appels en attente
 - Mode en attente d'appel
 - Service multi-parties
 - État et fonctionnalités des services supplémentaires
3. **ISD #3 (Optionnel)** - Données de Barrage d'Appels :
 - Barrage de tous les appels sortants (BACO)
 - Barrage des appels internationaux sortants (BOIC)
 - Données de restriction d'accès

Options de Configuration

```
# Configuration InsertSubscriberData
# Mode d'Accès Réseau : :packetAndCircuit, :packetOnly, ou :circuitOnly
isdn_network_access_mode: :packetAndCircuit,
# Envoyer ISD #2 (données de Services Supplémentaires)
isdn_send_ss_data: true,
# Envoyer ISD #3 (données de Barrage d'Appels)
isdn_send_call_barring: true,
```

Mode d'Accès Réseau

Le paramètre `isdn_network_access_mode` contrôle quel type d'accès réseau l'abonné est autorisé :

Valeur	Description	Cas d'utilisation
:packetAndCircuit	À la fois communiqué par paquets (GPRS/LTE) et communiqué par circuits (voix)	Par défaut - Abonnés à service complet
:packetOnly	Communiqué par paquets uniquement (données/LTE)	Cartes SIM uniquement pour données, appareils IoT
:circuitOnly	Communiqué par circuits uniquement (voix/SMS)	Appareils anciens, plans uniquement vocaux

Contrôle des Messages ISD

Vous pouvez contrôler quels messages ISD sont envoyés en fonction de vos besoins réseau :

Envoyer tous les ISD (Par défaut - Ensemble de fonctionnalités complet) :

```
isdn_send_ss_data: true,
isdn_send_call_barring: true,
```

Envoyer uniquement les données de base de l'abonné (Provisioning minimal) :

```
isdn_send_ss_data: false,
isdn_send_call_barring: false,
```

Envoyer de base + services supplémentaires (Pas de barrage d'appels) :

```
isdn_send_ss_data: true,
isdn_send_call_barring: false,
```

Exemple de Flux ISD

Lorsqu'une mise à jour de localisation est reçue :

```
VLR → HLR: UpdateLocation (DÉBUT)
HLR → VLR: InsertSubscriberData #1 (CONTINUER) - Données de base
VLR → HLR: ISD #1 ACK (CONTINUER)
```

```

HLR -> VLR: Insert5SubscriberData #2 (CONTINUER) - Données SS [si activé]
VLR -> HLR: ISD #2 ACK (CONTINUER)
HLR -> VLR: Insert5SubscriberData #3 (CONTINUER) - Barrage d'appels [si activé]
HLR -> VLR: ISD #3 ACK (CONTINUER)
HLR -> VLR: Réponse (FIN)

```

Si `isd_send_ss_data` ou `isd_send_call_barring` sont définis sur `false`, ces messages ISD sont ignorés, et la FIN de la mise à jour de localisation est envoyée plus tôt.

Meilleures Pratiques

- **Configuration par Défaut** : Utilisez `:packetAndCircuit` et activez tous les ISD pour une compatibilité maximale
- **IoT/M2M** : Utilisez `:packetOnly` et désactivez les données SS/barrage d'appels pour les appareils uniquement pour données
- **Interopérabilité** : Certains VLR plus anciens peuvent ne pas prendre en charge tous les services supplémentaires - désactivez `isd_send_ss_data` si vous rencontrez des problèmes
- **Performance** : Désactiver les ISD inutilisés réduit la surcharge des messages et accélère les mises à jour de localisation

Intégration CAMEL

Configuration CAMEL pour SendRoutingInfo

Lors de la réponse aux requêtes `SendRoutingInfo` (SRI) d'un GMSC (Gateway MSC), le HLR peut demander au GMSC d'invoquer des services CAMEL pour un routage d'appels intelligent et un contrôle de service.

Pour la référence des paramètres de configuration, voir [Configuration CAMEL dans la Référence de Configuration](#).

Qu'est-ce que CAMEL ?

CAMEL (Applications Personalisées pour la Logique Améliorée des Réseaux Mobiles) est un protocole qui permet des services réseau intelligents dans les réseaux GSM/UMTS. Il permet aux opérateurs de réseau de mettre en œuvre des services à valeur ajoutée tels que :

- Facturation prépayée
- Barrage d'appels
- Réseau Privé Virtuel (VPN)
- Services à tarif premium
- Transfert d'appels avec logique personnalisée
- Services basés sur la localisation

Options de Configuration

```

# Configuration CAMEL (pour les réponses SendRoutingInfo)
# Clé de Service pour l'initiation de service CAMEL
camel_service_key: 11_110

# Point de Détection de Déclenchement CAMEL
# Options : :termAttemptAuthorized, :tBusy, :tNoAnswer, :tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,

```

Clé de Service

La `camel_service_key` identifie quel service CAMEL doit être invoqué au gsmSCF (Service Control Function). Il s'agit d'un identifiant numérique configuré dans votre réseau :

Clé de Service	Cas d'utilisation Typique
11_110	Contrôle des appels entrants prépayés (par défaut)
100	Service prépayé d'origine
200	Transfert d'appels avec logique personnalisée
300	Réseau Privé Virtuel (VPN)
Personnalisé	Services spécifiques à l'opérateur

Exemple de Configuration :

```

# Pour le contrôle des appels entrants prépayés
camel_service_key: 11_110,
# Pour le service VPN,
camel_service_key: 300,

```

Point de Détection de Déclenchement

Le `camel_trigger_detection_point` spécifie quand le service CAMEL doit être déclenché lors de la configuration de l'appel :

Point de Détection	Description	Quand Déclenché
:termAttemptAuthorized	Tentative d'appel autorisée (par défaut)	Avant que l'appel ne soit routé vers l'abonné
:tBusy	Ocupé à la terminaison	Lorsque l'abonné est occupé
:tNoAnswer	Pas de réponse à la terminaison	Lorsque l'abonné ne répond pas
:tAnswer	Réponse à la terminaison	Lorsque l'abonné répond à l'appel

Exemples de Configuration :

Contrôle prépayé standard (déclenchement avant le routage) :

`camel_trigger_detection_point: :termAttemptAuthorized,`

Gestion personnalisée des occupations (déclenchement lorsque occupé) :

`camel_trigger_detection_point: :tBusy,`

Facturation basée sur la réponse (déclenchement à la réponse) :

`camel_trigger_detection_point: :tAnswer,`

Réponse SRI avec CAMEL

Lorsqu'il est configuré, les réponses `SendRoutingInfo` incluent des informations d'abonnement CAMEL :

```

GMSC -> HLR: SendRoutingInfo (DÉBUT)
HLR -> GMSC: Réponse SRI (FIN) avec :
  - IMSI
  - Numéro VLR
  - Etat de l'abonné
  - Infos de routage CAMEL :
    * Clé de Service : 11_110
    * Adresse gsmSCF : <adresse configurée>
  - Point de Détection de Déclenchement : termAttemptAuthorized
  - Gestion des Appels par Défaut : continueCall

```

Le GMSC contacte gsmSCF au point de déclenchement pour exécuter le service CAMEL

Meilleures Pratiques

- **Réseaux de Production** : Utilisez des clés de service standardisées convenues avec votre fournisseur gsmSCF
- **Tests** : Utilisez `:termAttemptAuthorized` pour des tests les plus complets
- **Services Prépayés** : La clé de service `11_110` est une norme industrielle courante pour les appels entrants prépayés
- **Gestion de Repli** : `:defaultCallHandling: :continueCall` garantit que les appels se poursuivent si le gsmSCF est ininjoignable

Gestion des Abonnés en Roaming

Détection VLR à Domicile vs VLR en Roaming

Lorsque le HLR reçoit une requête `SendRoutingInfo` (SRI), il doit déterminer si l'abonné est sur un VLR "domestique" (dans votre réseau) ou sur un VLR en roaming (visite d'un autre réseau). Le comportement diffère en fonction de cette détermination :

Pour la référence des paramètres de configuration, voir [Préfixes VLR à Domicile dans la Référence de Configuration](#).

- **VLR à Domicile** : Retourner une réponse SRI standard avec des informations de routage CAMEL
- **VLR en Roaming** : Envoyer une requête `Provide Roaming Number` (PRN) pour obtenir un MSRN, puis le retourner dans la réponse SRI

Configuration

```

# Préfixes VLR à Domicile
# Liste des préfixes d'adresses VLR considérés comme réseau "domestique"
# Si l'adresse VLR de l'abonné commence par l'un de ces préfixes, utiliser la réponse SRI standard
# Sinon, l'abonné est en roaming et nous devons envoyer PRN pour obtenir MSRN
home_vlr_prefixes: ["555123"],

# Exemple de Configuration :

```

```

# Un seul opérateur de réseau domestique
home_vlr_prefixes: ["555123"],

# Plusieurs opérateurs de réseau domestique (par exemple, différentes régions ou filiales)
home_vlr_prefixes: ["555123", "555124", "555125"],

# Comment Ça Fonctionne

```

1. Flux Abonné à Domicile (Standard)

Lorsque l'adresse VLR de l'abonné commence par un préfixe domestique configuré :

```

GMSC -> HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR interroge l'ABR basé sur les préfixes de l'abonné
HLR vérifie l'adresse VLR : "5551234567"
HLR détermine : VLR commence par "555123" - Réseau domestique
HLR -> GMSC: Réponse SRI avec infos de routage CAMEL :
  - IMSI
  - Numéro VLR : "5551234567"
  - Adresse gsmSCF (MSC) : "5551234567"
  - Clé de service CAMEL : 11_110
  - Point de détection de déclenchement : termAttemptAuthorized

```

2. Flux Abonné en Roaming (PRN Requis)

Lorsque l'adresse VLR de l'abonné ne correspond à aucun préfixe domestique :

```

GMSC -> HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR vérifie l'origine VLR et renvoie les données de l'abonné
HLR vérifie l'origine VLR : "49170123456"
HLR détermine que VLR ne commence pas par "555123" -> Roaming
HLR -> MSC: ProvideRoamingNumber (PRN) :
- MSISDN : "1234567890"
- IMSI : "999999876543210"
- Numéro MSC : "49170123456"
- Adresse GMSC : "555123456"
MSI -> HLR: Réponse PRN avec MSRN : "49170999888777"
HLR -> GMSC: Réponse SRI avec infos de routage :
- IMSI
- Numéro VLR : "49170123456"
- Numéro en Roaming (MSRN) : "49170999888777"

```

Differences de Structure de Réponse

Réponse SRI Abonné à Domicile

```
%{
  imsi: "999999876543210",
  extendedRoutingInfo: {
    :cAMELRoutingInfo, %{
      gsmCAMELSubscriptionInfo: %{
        *-CS1: %{
          serviceKey: "110",
          gatewayAddress: "491701234561",
          defaultCallHandling: :continueCall,
          *-BcmrTriggerDetectionPoint: :termAttemptAuthorized
        }
      }
    },
    subscriberInfo: %{
      locationInformation: %{"vlr-number": "5551234567"},
      subscriberState: {notProvidedFromVLR, :NULL}
    }
}
```

Réponse SRI Abonné en Roaming

```
%{
  imsi: "999999876543210",
  extendedRoutingInfo: {
    :routingInfo, %{
      roamingNumber: "49170999888777" # MSRN de PRN
    }
  },
  subscriberInfo: %{
    locationInformation: %{"vlr-number": "49170123456"},
    subscriberState: {notProvidedFromVLR, :NULL}
  }
}
```

Opération Provide Roaming Number (PRN)

Structure de la Requête PRN

La requête PRN envoyée au MSC/VLR contient :

Champ	Source	Description
MSISDN	Requête SRI Numéro de téléphone de l'abonné	
IMSI	API HLR IMSI de l'abonné	
Numéro MSC	API HLR MSC servant l'abonné en roaming (serving_msc)	
Adresse GMSC	Requête SRI GMSC effectuant la requête SRI originale	
Numéro de Référence d'Appel	Identifiant de référence d'appel	
Phases CAMEL Supportées	Statique	Phases CAMEL prises en charge par le GMSC

Gestion de la Réponse PRN

Le HLR attend une réponse PRN contenant :

- MSRN (Numéro de Roaming de la Station Mobile) : Un numéro temporaire alloué par le réseau visité pour le routage de l'appel

Gestion des Erreurs :

- Si le PRN expire -> Retourne l'erreur 27 (Abonné Absent) dans la réponse SRI
- Si le PRN échoue -> Retourne l'erreur 27 (Abonné Absent) dans la réponse SRI
- Si le MSRN ne peut pas être extrait -> Retourne l'erreur 27 (Abonné Absent) dans la réponse SRI

Exemples de Configuration

Opérateur de Réseau Domestique Unique

```
# Tous les adresses VLR commençant par "555123" sont considérées comme domestiques
home_vlr_prefixes: ["555123"],
  • VLR 5551234567 -> Domicile (réponse CAMEL)
  • VLR 5551235801 -> Domicile (réponse CAMEL)
  • VLR 49170123456 -> Roaming (PRN + réponse MSRN)
```

Opérateur Multi-Région

```
# Plusieurs réseaux domestiques dans différentes régions
home_vlr_prefixes: ["555123", "555124", "555125"],
  • VLR 5551234567 -> Domicile (région 1)
  • VLR 5552342123 -> Domicile (région 2)
  • VLR 5553411111 -> Domicile (région 3)
  • VLR 44201234567 -> Roaming (international)
```

Configuration de Test

Pour tester la fonctionnalité PRN, définissez une liste vide pour traiter tous les VLR comme en roaming :

```
# Tous les VLR sont traités comme en roaming (pour tester le flux PRN)
home_vlr_prefixes: [];
```

Meilleures Pratiques

- Selection de Préfixe : Utilisez le préfixe unique le plus court qui identifie les VLR de votre réseau (par exemple, code pays + code réseau)
- Multiples Préfixes : Incluez tous les préfixes VLR de votre réseau, y compris différentes régions et filiales
- Accords de Roaming : Assurez-vous que le PRN est correctement pris en charge par les réseaux partenaires en roaming
- Tests : Testez à la fois les scénarios domestiques et en roaming de manière approfondie avant le déploiement en production
- Surveillance : Surveillez les taux de délai d'attente du PRN pour identifier les problèmes de connectivité avec les partenaires en roaming

Dépannage

Symptôme : Tous les abonnés traités comme en roaming

- Cause : home_vlr_prefixes non configuré ou préfixes non correspondant pas aux adresses VLR
- Solution : Vérifiez les adresses VLR dans votre base de données et mettez à jour les préfixes en conséquence

Symptôme : Les requêtes PRN expirent

- Cause : Problèmes de connectivité réseau vers le MSC/VLR partenaire en roaming
- Solution : Vérifiez le routage M3UA/SCCP vers les adresses MSC distantes

Symptôme : MSRN invalide dans la réponse SRI

- Cause : Format de réponse PRN du partenaire en roaming ne correspondant pas à la structure attendue
- Solution : Examinez les journaux de réponse PRN et ajustez extract_msrn_from_prn/l si nécessaire

Opérations HLR

Opérations MAP Supportées

- updateLocation (Opode 2) - Enregistre la localisation VLR
- sendAuthenticationInfo (Opode 50) - Considère les valeurs d'auth
- sendLocationInfo (Opode 22) - Fournir MSRN pour les appels avec support CAMEL
- sendRoutingInfoForSM (Opode 45) - Fournir GT MSC pour SMS
- cancelLocation (Opode 3) - Désenregistre du VLR ancien
- insertSubscriberData (Opode 7) - Pousse le profil de l'abonné

Mapping des Champs de Réponse

Cette section détaille d'où provient chaque champ dans les réponses HLR.

Réponse SendRoutingInfo (SRI)

But : Fournit des informations de routage pour les appels entrants vers un abonné.

Le HLR fournit deux types de réponses différents en fonction de si l'abonné est sur un VLR domestique ou en roaming :

Réponse Abonné à Domicile (Routage CAMEL)

Utilisé lorsque l'adresse VLR de l'abonné commence par une valeur configurée dans home_vlr_prefixes.

Structure de Réponse :

Champ	Source	Description	Exemple
IMSI	API OmniHSS	IMSI de l'abonné provenant de la base de données OmniHSS	"999999876543210"
Numéro VLR	API OmniHSS VLR actuel servant l'abonné (<code>circuit_session.assigned_vlr</code>)	":notProvidedFromVLR	"9999998765432456"
Etat de l'abonné	Statique	Toujours <code>notProvidedFromVLR</code>	"9999998765432456"
extendedRoutingInfo	Type : <code>routingInfo</code>	Type : <code>routingInfo</code>	"9999998765432456"
Adresse gsmSCF	API OmniHSS	MSC servant l'abonné (<code>circuit_session.assigned_msc</code>)	"5551234561"
Cle de Service	runtime.exs	Identifiant de service (<code>camel_service_key</code>)	"11_118"
Point de Détection de Déclenchement	runtime.exs	Quand déclencher CAMEL (<code>camel_trigger_detection_point</code>)	"termAttemptAuthorized"
Gestion des Capacités CAMEL	Statique	Niveau de support de phase CAMEL	3
Gestion des Appels par Défaut	Statique	Repli si gsmSCF injoignable	":continueCall"

Reponse Abonné en Roaming (Routage MSRN)

Utilisé lorsque l'adresse VLR de l'abonné ne correspond à aucun préfixe `home_vlr_prefixes` configuré.

Structure de Réponse :

Champ	Source	Description	Exemple
IMSI	API OmniHSS	IMSI de l'abonné provenant de la base de données OmniHSS	"999999876543210"
Numéro VLR	API OmniHSS VLR actuel servant l'abonné (<code>circuit_session.assigned_vlr</code>)	":notProvidedFromVLR	"9999998765432456"
Etat de l'abonné	Statique	Toujours <code>notProvidedFromVLR</code>	"9999998765432456"
extendedRoutingInfo	-	Type : <code>routingInfo</code>	"9999998765432456"
Numéro en Roaming (MSRN)	Réponse PRN	MSRN obtenu de la requête <code>ProvideRoamingNumber</code>	"49179999888777"

Logique de Décision de Routage :

- OmniS5S7 reçoit la requête `SendRoutingInfo`
- OmniS5S7 interroge les données de l'abonné depuis l'API OmniHSS
- OmniS5S7 vérifie l'adresse VLR par rapport aux `home_vlr_prefixes` :

 - Si le VLR commence par un préfixe domestique :
 - Retourner les infos de routage CAMEL (flux abonné domestique)
 - Si le VLR ne correspond à aucun préfixe domestique :
 - Envoyer `ProvideRoamingNumber` (PRN) au MSC
 - Extraire le MSRN de la réponse PRN
 - Retourner les infos de routage avec MSRN (flux abonné en roaming)

Flux de Données :

- OmniS5S7 interroge OmniHSS pour les informations sur l'abonné
- OmniHSS retourne IMSI, localisation actuelle VLR/MSC et état de l'abonné
- OmniS5S7 utilise ces données pour construire la réponse PRN

Exigences de Configuration :

```
# Dans runtime.exs
home_vlr_prefixes: ["555123"], # Liste des préfixes VLR à domicile
```

Réponses d'Erreur :

- Si `serving_vlr` et `serving_msc` sont null : Retourne l'erreur 27 (Abonné Absent)
- Si l'abonné n'est pas trouvé : Retourne l'erreur 1 (Abonné Inconnu)
- Si la requête PRN expire (cas en roaming) : Retourne l'erreur 27 (Abonné Absent)
- Si la réponse PRN est invalide (cas en roaming) : Retourne l'erreur 27 (Abonné Absent)

Réponse UpdateLocation avec InsertSubscriberData

But : Enregistre l'abonné à un nouveau VLR et provisionne les données de l'abonné.

Réponse UpdateLocation FIN

Champ	Source	Description	Exemple
Numéro HLR	runtime.exs	Global Title de ce HLR (<code>hlr_service_center_gt_address</code>)	"5551234568"
Type de Message	TCAP Statique	Réponse finale après tous les ISD	FIN

InsertSubscriberData #1 (Données de Base de l'Abonné)

Champ	Source	Description	Exemple
IMSI	Requête	Provenant de la requête <code>UpdateLocation</code>	"999999876543210"
MSISDN	API OmniHSS	Numéro de téléphone de l'abonné provenant d'OmniHSS	"555123456"
Catégorie	Statique	Catégorie d'abonné	"\n"(0x0A)
Etat de l'abonné	Statique	Etat de service	":serviceGranted"
Liste des Services de Support	Statique	Services de support pris en charge	["<lt;>"]
Liste des Téléservices	Statique	Téléservices pris en charge	["<lt;1,17>"," ","]
Mode d'Accès Réseau	runtime.exs	Accès par paquet/circuit (<code>isd_network_access_mode</code>)	":packetAndCircuit"

InsertSubscriberData #2 (Services Supplémentaires) - Optionnel

Champ	Source	Description	Contrôlé Par
SS Provenance	Statique	Données des services supplémentaires	isd_send_ss_data: true
Transfert d'Appels	Statique	Configurations de transfert (conditionnel, occupé, pas de réponse, non joignable)	Config activé
Appel en Attente	Statique	Etat du service d'appel en attente	Config activé
Service Multi-Parties	Statique	Support d'appel conférence	Config activé

ISD #2 inclut :

- Transfert d'appels inconditionnel (code SS 21)
- Transfert d'appels en cas d'occupation (code SS 41)
- Transfert d'appels en cas de non réponse (code SS 42)
- Transfert d'appels en cas de non joignabilité (code SS 62)
- Appel en attente (code SS 43)
- Services multi-parties (code SS 51)
- Services CLIP/CLIR

InsertSubscriberData #3 (Barrage d'Appels) - Optionnel

Champ	Source	Description	Contrôlé Par
Infos de Barrage d'Appels	Statique	Configurations de barrage d'appels	isd_send_call_barring: true
BAOC	Statique	Barrage de tous les appels sortants (code SS 146)	Config activé
BOIC	Statique	Barrage des appels internationaux sortants (code SS 147)	Config activé
Données de Restriction d'Accès	Statique	Restrictions d'accès au réseau	Config activé

Contrôle de Séquence ISD :

- ISD #1 : **Toujours envoyé** - Contient les données essentielles de l'abonné
- ISD #2 : Envoyé uniquement si `isd_send_ss_data: true` dans `runtime.exs`
- ISD #3 : Envoyé uniquement si `isd_send_call_barring: true` dans `runtime.exs`

Réponse SendRoutingInfoForSM (SRI-for-SM)

But : Fournit des informations de routage MSC/SMSC pour la livraison de SMS. Lorsqu'un SMSc doit livrer un SMS à un abonné, il envoie une requête SRI-for-SM au HLR pour déterminer où router le message.

Structure de Réponse :

Champ	Source	Description	Comment Généré	Exemple
IMSI	Calculé	IMSI synthétique dérivé de MSISDN	PLMN_PREFIX + zero_padded_MSISDN	"0010015551234567"
Numéro de Noeud Réseau	runtime.exs	Adresse GT SMSC pour le routage	SMSc_service_center_gt_address	"5551234567"

Paramètres de Configuration (depuis `runtime.exs`) :

```
# Adresse GT du Centre de Service (retournée dans les réponses SRI-for-SM)
# Cela indique au SMSC demandeur où envoyer les messages MT-ForwardSM
smsc_service_center_gt_address: "5551234567", # Requis

# Configuration de Mapping MSISDN - IMSI
# Prefixe PLMN : MCC (001 = Réseau de Test) + MNC (01 = Opérateur de Test)
# Prefixe PLMN : MCC (001 = Réseau de Test) + MNC (01 = Opérateur de Test)
htr_imsi_plmn_prefix: "001001", # Seul paramètre de config nécessaire !
```

Mapping MSISDN -- IMSI

Paramètres de Configuration :

Ces paramètres contrôlent comment OmniS5S7 génère des IMSI synthétiques à partir des MSISDN pour les réponses SRI-for-SM :

- `htr_imsi_plan_prefix` : Le préfixe MCC+MNC à utiliser lors de la construction des IMSI synthétiques (par exemple, "50057" pour MCC=500, MNC=57)
- `htr_msisdn_country_code` : Code du pays à préfixer lors du mappage inverse MSISDN-->IMSI (par exemple, "61" pour les États-Unis/Canada)
- `htr_msisdn_ns_offset` : Position du caractère où commence le Numéro d'Abonné National (NSN) dans le MSISDN (typiquement 0 si le MSISDN n'inclut pas le code pays, ou longueur du code pays s'il l'inclut)
- `htr_msisdn_ns_length` : Nombre de chiffres à extraire du MSISDN comme NSN

Pour des détails supplémentaires sur la configuration, voir [Mapping MSISDN -- IMSI dans la Référence de Configuration](#).

Pourquoi le Mapping MSISDN vers IMSI est-il nécessaire ?

Le protocole MAP pour **SendRoutingInfoForSM** (SRI-for-SM) exige que le HLR retourne un **IMSI** (Identité Internationale d'Abonné Mobile) dans sa réponse. Cependant, le SMSC demandeur ne connaît que le **MSISDN** (numéro de téléphone).

Dans un réseau traditionnel :

- Le SMSC envoie SRI-for-SM avec le MSISDN de destination (par exemple, "5551234567")
- Le HLR doit rechercher l'abonné dans sa base de données pour trouver son IMSI
- Le HLR retourne l'IMSI dans la réponse SRI-for-SM
- Le SMSC utilise ensuite cet IMSI lors de l'envoi de MT-ForwardSM au MSC/VLR

Approche d'OmniS5S7 - IMSIs Synthétiques :

Au lieu de maintenir une base de données complète des abonnés avec des mappings MSISDN vers IMSI, OmniS5S7 utilise un schéma d'encodage simple pour **calculer** des IMSI synthétiques directement à partir des MSISDN. Cette approche offre deux avantages clés :

- `hlr_service_center_gt_address` - Utilisé dans les réponses `UpdateLocation`
- `smsc_service_center_gt_address` - Utilisé dans les réponses `SRI-for-SM` (où les MT-ForwardSM doivent être routés)

Optionnel dans `runtime.exe` (avec valeurs par défaut) :

- `camel_service_key` - Par défaut: `11_110`
- `camel_triggrg_acctdetect` - Par défaut: `:termAttemptAuthorized`
- `isd_send_ss_access_mode` - Par défaut: `:packetAndCircuit`
- `isd_send_ss_data` - Par défaut: `true`
- `isd_send_call_barring` - Par défaut: `true`
- `hlr_imsi_plan_prefix` - Par défaut: `"001001"` (préfixe PLMN pour le mapping MSISDN→IMSI)

Requis de la part d'OmniHSS :

OmniHSS doit fournir des points d'API REST pour :

- Recherche d'abonné par IMSI et MSISDN
- Mises à jour de localisation de session circuit (assигнация VLR/MSC)
- Génération de vecteurs d'authentification
- Requête sur l'état de l'abonné et les profils de service

Documentation Connexe

Documentation OmniSS7 :

- [Retour à la Documentation Principale](#)
- [Guide des Fonctionnalités Communes](#)
- [Guide Client MAP](#)
- [Référence Technique](#)
- [Référence de Configuration](#)

Documentation OmniHSS : Pour la gestion des abonnés, le provisioning, la configuration de l'authentification et les opérations administratives, reportez-vous à la **documentation produit OmniHSS**. OmniHSS contient toute la logique de base de données des abonnés, les algorithmes d'authentification, les règles de provisioning de service et les capacités de gestion Multi-IMSI.

OmniSS7 par Omnitouch Network Services

Guide de Configuration du Client MAP

[Retour à la Documentation Principale](#)

Ce guide fournit une configuration détaillée pour utiliser OmniSS7 en tant que **Client MAP** pour envoyer des requêtes du protocole MAP aux éléments du réseau.

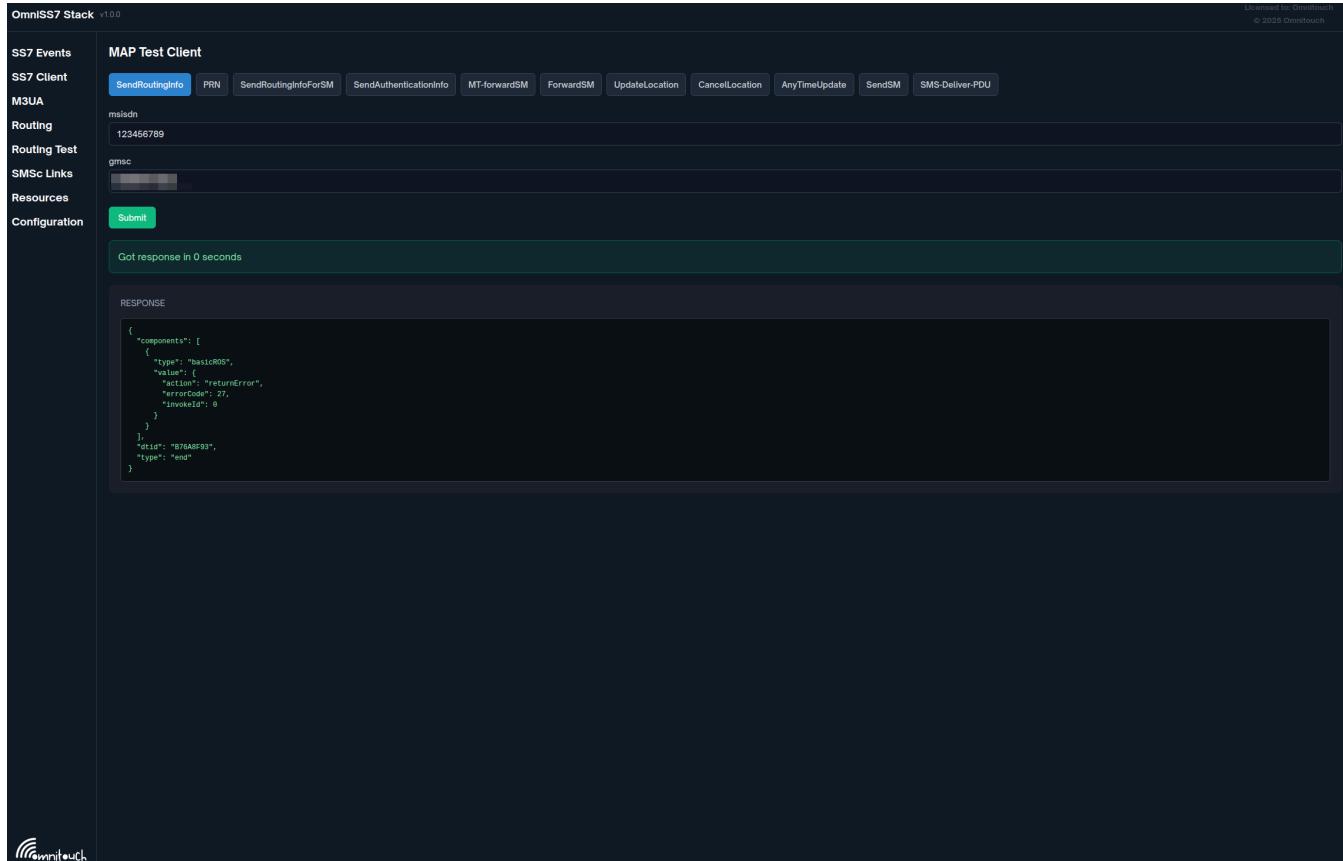


Table des Matières

1. [Qu'est-ce que le Mode Client MAP ?](#)
2. [Activation du Mode Client MAP](#)
3. [Connexion M3UA](#)
4. [Envoyer de Remarques via l'API](#)
5. [Métriques et Surveillance](#)
6. [Dépannage](#)

Qu'est-ce que le Mode Client MAP ?

Mode Client MAP permet à OmniSS7 de se connecter en tant que **Processus Serveur d'Application (ASP)** à un pair M3UA (STP ou SGP) et d'envoyer/recevoir des messages **MAP (Mobile Application Part)** pour des services tels que :

- **Requêtes HLR : SRI** (SRI-for-ASP, Authentication Info)
- **Mises à Jour de Localisation** : Update Location, Cancel Location
- **Gestion des Abonnés** : Provide Roaming Number (PRN), Insert Subscriber Data

Architecture Réseau

Activation du Mode Client MAP

Éditez config/runtime.exs et configurez les paramètres du client MAP. Pour une référence de configuration complète, voir [Paramètres de Connexion M3UA dans la Référence de Configuration](#).

Configuration de Base

```
config :omnis7,  
  # Activer le mode Client MAP  
  map_client_enabled: true,  
  
  # Connexion M3UA pour le Client MAP (se connecte en tant qu'ASP à un STP/SGP distant)  
  map_client_m3ua: %  
    mode: "ASP" # Mode M3UA : "ASP" (client) ou "SGP" (serveur)  
    callback: &{MapClient, :handle_payload, []}, # Callback pour les messages entrants  
    process_name: :map_client_asp, # Nom du processus enregistré  
    local_ip: (10, 0, 0, 100), # Adresse IP locale  
    local_port: 2905, # Port SCTP local  
    remote_ip: (10, 0, 0, 1), # IP STP/SGP à connecter  
    remote_port: 2905, # Port STP/SGP distant  
    routing_context: 1 # Contexte de routage M3UA  
  }
```

Exemple de Configuration en Production

```
config :omnis7,  
  # Activer le Client MAP pour la production  
  map_client_enabled: true,  
  
  # Connexion M3UA en production  
  map_client_m3ua: %  
    mode: "ASP"  
    callback: &{MapClient, :handle_payload, []},  
    process_name: :map_client_asp,  
    local_ip: (10, 0, 0, 100),  
    local_port: 2905,  
    remote_ip: (10, 0, 0, 1), # IP STP en production  
    remote_port: 2905,  
    routing_context: 1  
    
  config :control_panel,  
    web: %  
      listen_ip: "0.0.0.0",  
      port: 443,  
      host: "s7-gateway.example.com",  
      enable_tls: true,  
      tls_cert: "/etc/ssl/certs/gateway.crt",  
      tls_key: "/etc/ssl/private/gateway.key"  
    
```

Opérations MAP Disponibles

MAP Tester API 1.0.0 OAS 3.0
/swagger.json

default

POST /api/MT-forwardSM MT-forwardSM MAP Request (For sending SMS for delivery by remote MSC/SMSi)

POST /api/deliverPDU Utility: Build SMS-DELIVER TPDU from originating address + GSM7

POST /api/forwardSM forwardSM MAP Request

POST /api/prn ProvideRoamingNumber (PRN) MAP Request

POST /api/send-auth-info SendAuthenticationInfo MAP Request

POST /api/sendSM Utility: Perform SRI-for-SM + MT-forwardSM from MSISDN and GSM7

POST /api/sri SendRoutingInfo MAP Request

POST /api/sri-for-sm SendRoutingInfoForSM MAP Request

POST /api/updateLocation UpdateLocation MAP Request

GET /metrics Prometheus metrics

GET /swagger.json OpenAPI spec

Schemas

AuthInfoRequest >

ErrorResponse >

ForwardSMRequest >

PRNRequest >

SMSDeliverPDURequest >

SMSDeliverPDUResponse >

SRIForSMRequest >

SRIRequest >

SRIResponse >

SendSMRequest >

UpdateLocationRequest >

1. Envoyer des Informations de Routage pour SM (SRI-for-SM)

Interroge le HLR pour déterminer le MSC servant pour la livraison de SMS. Pour des informations détaillées sur la façon dont le HLR traite les requêtes SRI-for-SM, voir [SRI-for-SM dans le Guide HLR](#).

Point de terminaison API : POST /api/sri-for-sm

Requête :

```
{  
  "msisdn": "447712345678",  
  "serviceCenter": "447999123456"  
}
```

Réponse :

```
{  
  "result": {  
    "imsi": "234567890123456789",  
    "locationInformation": {  
      "networkNodeNumber": "447999555111"  
    }  
  }  
}
```

Exemple cURL :

```
curl -X POST http://localhost/api/sri-for-sm \  
-H "Content-Type: application/json" \  
-d '{  
  "msisdn": "447712345678",  
  "serviceCenter": "447999123456"  
}'
```

2. Envoyer des Informations de Routage (SRI)

Interroge le HLR pour des informations de routage d'appel vocal.

Point de terminaison API : POST /api/sri

Requête :

```
{  
  "msisdn": "447712345678",  
  "gmsc": "447999123456"  
}
```

Réponse :

```
{  
  "result": {  
    "imsi": "234567890123456789",  
    "extendedRoutingInfo": {  
      "routingInfo": {  
        "roamingNumber": "447999555222"  
      }  
    }  
  }  
}
```

3. Fournir un Numéro de Roaming (PRN)

Demande un numéro de roaming temporaire (MSRN) au MSC servant.

Point de terminaison API : POST /api/prn

Requête :

```
{  
  "msisdn": "447712345678",  
  "gsc": "447999123456",  
  "lsc": "447999555111",  
  "imsi": "234509876543210"  
}
```

4. Envoyer des Informations d'Authentification

Demande des vecteurs d'authentification au HLR pour l'authentification des abonnés.

Point de terminaison API : POST /api/send-auth-info

Requête :

```
{  
  "imsi": "234509876543210",  
  "vectors": 5  
}
```

Réponse :

```
{  
  "result": {  
    "authenticationSetList": [  
      {  
        "rand": "0123456789ABCDEF0123456789ABCDEF",  
        "kres": "ABCDEF0123456789",  
        "ek": "0123456789ABCDEF0123456789ABCDEF",  
        "ik": "FEDCBA9876543210FEDCBA9876543210",  
        "auth": "0123456789ABCDEF0123456789ABCDEF"  
      }  
    ]  
  }  
}
```

5. Mettre à Jour la Localisation

Enregistre la localisation actuelle d'un abonné auprès du HLR. Pour des informations détaillées sur le traitement de UpdateLocation et les séquences d'InsertSubscriberData, voir [Mises à Jour de Localisation dans le Guide HLR](#).

Point de terminaison API : POST /api/updateLocation

Requête :

```
{  
  "imsi": "234509876543210",  
  "vlr": "447999555111"  
}
```

Résumé des Opérations MAP

Envoyer de Requêtes via l'API

Utilisation de Swagger UI

L'interface Swagger UI fournit une interface interactive pour envoyer des requêtes SS7.

Accéder à Swagger UI :

1. Naviguez vers <http://your-server/swagger>
2. Parcourez les points de terminaison API disponibles
3. Cliquez sur n'importe quel point de terminaison pour développer ses détails

Envoyer une Requête :

1. Cliquez sur le point de terminaison que vous souhaitez utiliser (par exemple, /api/sri-for-sm)
2. Cliquez sur le bouton "Try it out"
3. Remplissez les paramètres requis dans le corps de la requête
4. Cliquez sur "Execute"
5. Consultez la réponse ci-dessous

Codes de Réponse de l'API

- 200 - Succès, résultat retourné dans le corps de la réponse
- 400 - Mauvaise Requête, paramètres invalides
- 504 - Délay d'Attente de la Passerelle, pas de réponse du réseau SS7 dans les 10 secondes

Métriques du Client MAP

Métriques Disponibles

Métriques de Requête :

- map_requests_total - Nombre total de requêtes MAP envoyées
 - Étiquettes: operation (valeurs: sri, sri_for_sm, prn, authentication_info, etc.)
- map_request_errors_total - Nombre total d'erreurs de requêtes MAP
 - Étiquettes: operation
- map_request_duration_milliseconds - Histogramme des durées des requêtes MAP
 - Étiquettes: operation
- map_pending_requests - Nombre actuel de requêtes MAP en attente (gauge)

Exemples de Requêtes Prometheus

```
# Total des requêtes SRI-for-SM au cours de la dernière heure  
increase(map_requests_total{operation='sri_for_sm'})1h  
  
# Temps de réponse moyen pour les requêtes SRI  
rate(map_request_duration_milliseconds_sum{operation="sri"}[5m]) /  
rate(map_request_duration_milliseconds_count{operation="sri"}[5m])  
  
# Taux d'erreur pour toutes les opérations MAP  
sum(rate(map_request_errors_total[5m])) by (operation)  
  
# Requêtes en attente actuelles  
map_pending_requests
```

Dépannage Client MAP

Problème : Délai d'Attente des Requêtes

Symptômes :

- L'API retourne 504 Gateway Timeout
- Pas de réponse du HLR/MSC

Vérifications :

1. Vérifiez que la connexion M3UA est ACTIVE :

```
# Dans la console IEx  
:sys:get_state(:map_client_asp)
```
2. Vérifiez la connectivité réseau vers le STP
3. Vérifiez le contexte de routage et l'adressage SCCP
4. Vérifiez les journaux pour des erreurs SCCP

Problème : Erreurs SCCP

Symptômes :

- L'API retourne des réponses d'erreur SCCP
- Les journaux montrent des messages "SCCP unidata service"

Codes d'Erreur SCCP Courants :

- Pas de Traduction : Titre Global non trouvé dans la table de routage STP
- Échec de Sous-système : Sous-système de destination (HLR SSN 6) indisponible
- Échec Réseau : Congestion ou échec du réseau

Solutions :

- Contacter l'administrateur STP pour vérifier la configuration de routage
- Vérifiez que le Titre Global de destination est accessible
- Vérifiez si le sous-système de destination est opérationnel

Documentation Connexe

- [Retour à la Documentation Principale](#)
- [Guide des Fonctionnalités](#) - Interface Web, API, Surveillance
- [Guide de l'Administration de routeurs](#)
- [Guide du Centre SMS](#) - Livraison de SMS
- [Référence Technique](#) - Spécifications du protocole



Guide de Configuration du Centre SMS (SMS)

[← Retour à la Documentation Principale](#)

Ce guide fournit une configuration détaillée pour utiliser OmniSS7 comme un **Centre SMS (SMS)** en tant que frontend avec **OmniMessage** comme plateforme de stockage et de livraison des messages en backend.

Intégration d'OmniMessage

Le mode SMS d'OmniSS7 fonctionne comme un frontend de signalisation SS7 qui s'interface avec **OmniMessage**, une plateforme SMS de niveau opérateur. Cette architecture sépare les préoccupations :

- **OmniSS7 (Frontend SMS)** : Gère tous les signaux de protocole SS7/MAP, le routage SCCP et la communication réseau
- **OmniMessage (Backend SMS)** : Gère le stockage des messages, la mise en file d'attente, la logique de réessai, le suivi de livraison et les décisions de routage

Pourquoi OmniMessage ?

OmniMessage fournit des capacités de messagerie SMS de niveau opérateur avec des fonctionnalités incluant :

- **Gestion de la File d'Attente des Messages** : Stockage persistant avec une logique de réessai configurable et une mise en file d'attente par priorité
- **Suivi de Livraison** : Statut de livraison en temps réel, rapports de livraison (DLR) et suivi des raisons d'échec
- **Support Multi-SMS** : Plusieurs instances de frontend peuvent se connecter à un seul backend OmniMessage pour l'équilibrage de charge et la redondance
- **Intelligence de Routage** : Règles de routage avancées basées sur la destination, l'expéditeur, le contenu du message et l'heure de la journée
- **Limitation de Taux** : Contrôles TPS (transactions par seconde) par route pour prévenir la congestion du réseau
- **Conception API-First** : API HTTP RESTful pour l'intégration avec les systèmes de facturation, les portails clients et les applications tierces
- **Analytique & Reporting** : Statistiques sur le volume des messages, taux de succès de livraison et métriques de performance

Toutes les données de message, l'état de livraison et les configurations de routage sont stockées et gérées dans OmniMessage. OmniSS7 interroge OmniMessage via des appels API HTTPS pour récupérer les messages en attente, mettre à jour l'état de livraison et s'enregistrer en tant que frontend actif.

Important : Le mode SMS d'OmniSS7 est un **frontend de signalisation uniquement**. Toute la logique de routage des messages, la gestion des files d'attente, les algorithmes de réessai, le suivi de livraison et les règles métier sont gérés par OmniMessage. Ce guide couvre la configuration du protocole SS7/MAP dans OmniSS7. Pour des informations sur le routage des messages, la configuration des files d'attente, les rapports de livraison, la limitation de taux et l'analytique, **reportez-vous à la documentation d'OmniMessage**.

Table des Matières

1. [Intégration d'OmniMessage](#)
2. [Qu'est-ce que le Mode Centre SMS ?](#)
3. [Activation du Mode SMS](#)
4. [Configuration de l'API HTTP](#)
5. [Flux de Messages SMS](#)
6. [Prévention des Boucles](#)
7. [Suivi des Abonnés SMS](#)
8. [Configuration de Flush Automatique](#)
9. [Métriques et Surveillance](#)
10. [Dépannage](#)

Qu'est-ce que le Mode Centre SMS ?

Remarque : Cette section couvre uniquement la configuration de signalisation SS7 d'OmniSS7. Pour les règles de routage des messages, la gestion des files d'attente, le suivi de livraison et la configuration de la logique métier, voir la **documentation produit d'OmniMessage**.

Le Mode Centre SMS permet à OmniSS7 de fonctionner comme un SMSc pour :

- **Livraison MT-SMS** : Livraison de SMS Terminés vers les abonnés
- **Gestion MO-SMS** : Réception et routage de SMS Origines Mobiles
- **Mise en File d'Attente des Messages** : File d'attente de messages soutenue par une base de données avec logique de réessay
- **Flush Automatique** : Livraison automatique de SMS depuis la file d'attente
- **Rapports de Livraison** : Suivi de l'état de livraison des messages

Architecture du Centre SMS

Activation du Mode SMSc

OmniSS7 peut fonctionner en différents modes. Pour l'utiliser comme un SMSc, vous devez activer le mode SMSc dans la configuration.

Changer vers le Mode SMSc

Le fichier config/runtime.exs d'OmniSS7 contient trois modes opérationnels préconfigurés. Pour activer le mode SMSc :

1. **Ouvrir** config/runtime.exs
2. **Trouver** les trois sections de configuration (lignes 53-204) :
 - Configuration 1 : Mode STP (lignes 53-95)
 - Configuration 2 : Mode HLR (lignes 97-142)
 - Configuration 3 : Mode SMSc (lignes 144-204)
3. **Commenter** toute autre configuration active (ajouter # à chaque ligne)
4. **Décommenter** la configuration SMSc (retirer # des lignes 144-204)
5. **Personnaliser** les paramètres de configuration selon les besoins
6. **Redémarrer** l'application : iex -S mix

Configuration du Mode SMSc

La configuration complète du SMSc ressemble à ceci :

```
config :omniss7,
  # Drapeaux de mode - Activer les fonctionnalités STP + SMSc
  # Remarque : map_client_enabled est vrai car le SMSc a besoin de capacités de routage
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,

  # Configuration de l'API Backend OmniMessage
  smsc_api_base_url: "https://10.179.3.219:8443",
  # Identification du SMSc pour l'enregistrement avec le backend
  smsc_name: "ipsmgw",
  # Adresse GT du Centre de Service pour les opérations SMS
  smsc_service_center_gt_address: "5551234567",

  # Configuration de Flush Automatique (traitement de la file d'attente SMS en arrière-plan)
  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smss: "ipsmgw",
  auto_flush_tps: 10,

  # Configuration de Connexion M3UA
  # Se connecter en tant qu'ASP pour envoyer/recevoir des opérations MAP SMS
  map_client_m3ua: %{
```

```

mode: "ASP",
callback: {MapClient, :handle_payload, []},
process_name: :stp_client_asp,
# Point de terminaison local (système SMSc)
local_ip: {10, 179, 4, 12},
local_port: 2905,
# Point de terminaison STP distant
remote_ip: {10, 179, 4, 10},
remote_port: 2905,
routing_context: 1
}

config :control_panel,
use_additional_pages: [
  {SS7.Web.EventsLive, "/events", "Événements SS7"},
  {SS7.Web.TestClientLive, "/client", "Client SS7"},
  {SS7.Web.M3UAStatusLive, "/m3ua", "M3UA"},
  {SS7.Web.RoutingLive, "/routing", "Routage"},
  {SS7.Web.RoutingTestLive, "/routing_test", "Test de Routage"},
  {SS7.Web.SmscLinksLive, "/smsc_links", "Liens SMSc"}
],
page_order: ["/events", "/client", "/m3ua", "/routing", "/routing_test", "/smsc_links",
"/application", "/configuration"]

```

Paramètres de Configuration à Personnaliser

Pour une référence complète de tous les paramètres de configuration, voir la [Référence de Configuration](#).

Paramètre	Type	Par Défaut	Description	Exemple
smsc_api_base_url	String	<i>Requis</i>	Point de terminaison de l'API backend OmniMessage	"https://10.179.3.219:8443"
smsc_name	String	"{hostname}_SMSc"	Votre identifiant SMS pour l'enregistrement	"ipsmgw"
smsc_service_center_gt_address	String	<i>Requis</i>	Titre Global du Centre de Service	"5551234567"
auto_flush_enabled	Boolean	true	Activer le traitement automatique de la file d'attente	false
auto_flush_interval	Integer	10_000	Intervalle de traitement de la file d'attente en millisecondes	5_000
auto_flush_dest_smsc	String	<i>Requis</i>	Nom du SMS de destination pour le flush automatique	"ipsmgw"
auto_flush_tps	Integer	10	Taux de traitement des messages (transactions/seconde)	20
local_ip	Tuple	<i>Requis</i>	Adresse IP de votre système SMSc	{10, 179, 4, 12}
local_port	Integer	2905	Port SCTP local	2905
remote_ip	Tuple	<i>Requis</i>	Adresse IP STP pour la connectivité SS7	{10, 179, 4, 10}
remote_port	Integer	2905	Port SCTP	2905

Paramètre	Type	Par Défaut	Description	Exemple
routing_context	Integer	1	distant ID de contexte de routage M3UA	1

Que Se Passe-t-il Lorsque le Mode SMS est Activé

Lorsque `smsc_mode_enabled: true` et `map_client_enabled: true`, l'interface web affichera :

- ◊ **Événements SS7** - Journalisation des événements
- ◊ **Client SS7** - Test des opérations MAP
- ◊ **M3UA** - État de la connexion
- ◊ **Routage** - Gestion de la table de routage (STP activé)
- ◊ **Test de Routage** - Test de routage (STP activé)
- ◊ **Liens SMS** - État de l'API SMS + gestion de la file d'attente SMS ← *Spécifique au SMS*
- ◊ **Ressources** - Surveillance du système
- ◊ **Configuration** - Visualiseur de configuration

L'onglet **Liens HLR** sera masqué.

Notes Importantes

- Le mode SMS nécessite `map_client_enabled: true` pour les capacités de routage
- Backend OmniMessage** : L'API OmniMessage doit être accessible à l'URL `smsc_api_base_url` configurée
- Enregistrement du Frontend** : Le système s'enregistre automatiquement auprès d'OmniMessage toutes les **5 minutes** via le module `SMS.FrontendRegistry`
- Délai d'Attente des Requêtes API** : Toutes les requêtes API OmniMessage ont un **délai d'attente codé en dur de 5 secondes**
- Délai d'Attente des Requêtes MAP** : Toutes les requêtes MAP (SRI-for-SM, MT-ForwardSM, etc.) ont un **délai d'attente codé en dur de 10 secondes**
- Le flush automatique traite automatiquement la file d'attente SMS en arrière-plan
- La connexion M3UA au STP est requise pour envoyer/recevoir des opérations MAP SMS
- Après avoir changé de mode, vous devez redémarrer l'application pour que les changements prennent effet
- Interface Web** : Voir le [Guide de l'Interface Web](#) pour des informations sur l'utilisation de l'interface web
- Accès API** : Voir le [Guide API](#) pour la documentation de l'API REST et l'accès à l'interface Swagger

Configuration de l'API HTTP

Configuration du Backend OmniMessage

OmniSS7 communique avec OmniMessage via l'API REST HTTPS pour gérer la livraison des messages, suivre l'état des abonnés et s'enregistrer en tant que frontend actif :

```
config :omniss7,
  # URL de base de l'API OmniMessage
  smsc_api_base_url: "https://10.5.198.200:8443",
  # Identifiant du nom SMS pour l'enregistrement (par défaut à hostname_SMS si vide)
  smsc_name: "omni-smsc01",
  # Adresse GT du Centre de Service pour les opérations SMS
  smsc_service_center_gt_address: "5551234567"
```

Paramètres de Configuration :

Paramètre	Type Requis	Par Défaut	Description
<code>smsc_api_base_url</code>	String Oui	" <code>https://localhost:8443</code> "	URL de base pour l'API OmniMessage
<code>smsc_name</code>	String Non	"" (utilise <code>{hostname}_SMS</code>)	Identifiant SMS pour l'enregistrement et la gestion de la file d'attente
<code>smsc_service_center_gt_address</code>	String Non	"5551234567"	Adresse GT du Centre de Service renvoyée dans les réponses SRI-

Paramètre	Type Requis	Par Défaut	Description
			for-SM. Cela indique aux autres éléments du réseau où router les messages MT-ForwardSM. Voir le Guide SRI-for-SM pour plus de détails.

Enregistrement du Frontend

Le système s'enregistre automatiquement auprès d'OmniMessage au démarrage et **se réenregistre toutes les 5 minutes** via le module `SMS.FrontendRegistry`. Cela permet à OmniMessage de :

- Suivre les frontends actifs pour l'équilibrage de charge
- Surveiller le temps de fonctionnement et l'état de santé
- Collecter des informations de configuration
- Gérer le routage SMS distribué à travers plusieurs frontends

Détails de Mise en Œuvre :

- **Intervalle d'Enregistrement** : 5 minutes (codé en dur)
- **Processus** : Démarré automatiquement lorsque `smsc_mode_enabled: true`

Charge Utile d'Enregistrement :

```
{
  "frontend_name": "omni-smsc01",
  "configuration": "...",
  "frontend_type": "SS7",
  "hostname": "smsc-server01",
  "uptime_seconds": 12345
}
```

Remarque : Le nom du frontend est pris du paramètre de configuration `smsc_name`. S'il n'est pas défini, il par défaut à "`{hostname}_SMS`".

Communication avec l'API OmniMessage

Lorsque OmniSS7 reçoit des opérations MAP du réseau SS7 ou traite la file d'attente de messages, il communique avec OmniMessage pour :

- **S'enregistrer en tant que frontend actif** et signaler l'état de santé
- **Soumettre des messages d'origine mobile (MO)** reçus des abonnés
- **Récupérer des messages terminés mobiles (MT)** de la file d'attente pour livraison
- **Mettre à jour l'état de livraison** avec des rapports de succès/échec
- **Interroger des informations de routage** pour le transfert de messages

Point de terminaison	Méthode	But	Corps de la Requête
/api/frontends	POST	Enregistrer une instance de frontend	{"frontend_name": "...", "frontend_type": "SMS", "hostname": "...", "uptime_seconds": ...}
/api/messages_raw	POST	Insérer un nouveau message SMS	{"source_msisdn": "...", "source_smsc": "...", "message_body": "..."}
/api/messages	GET	Obtenir la file d'attente de messages	En-tête : <code>smsc: <smsc_name></code>
/api/messages/{id}	PATCH	Marquer le message comme livré	{"deliver_time": "...", "dest_smsc": "..."}
/api/messages/{id}	PUT	Mettre à jour l'état du message	{"dest_smsc": null}
/api/locations	POST	Insérer/mettre à jour la localisation de	{"msisdn": "...", "imsi": "...", "location": "...", "ims_capable": true, "csfb": false, "expires": "...", "user_agent": "...", "ran_location": "...", "imei": "...", ...}

Point de terminaison	Méthode	But	Corps de la Requête
/api/events	POST	l'abonné Ajouter un suivi d'événement	"registered": "..." {"message_id": ..., "name": "...", "description": "..."} -
/api/status	GET	Vérification de santé	-

Format de Réponse de l'API

Toutes les réponses de l'API utilisent le format JSON avec les conventions suivantes :

- **Réponses de succès** : HTTP 200-201 avec un corps JSON contenant les données de résultat
 - **Réponses d'erreur** : HTTP 4xx/5xx avec des détails d'erreur dans le corps de la réponse
 - **Horodatages** : Format ISO 8601 (par exemple, "2025-10-21T12:34:56Z")
 - **IDs de Message** : Identifiants entiers ou chaînes

Modules Client API

Le système SMS se compose de trois modules principaux :

1. SMSc.APIClient

Module principal du client API fournissant toute la communication HTTP API avec OmniMessage :

- `frontend_register/4` - Enregistrer le frontend auprès d'OmniMessage
 - `insert_message/3` - Insérer un message SMS brut (version compatible Python à 3 paramètres)
 - `insert_location/9` - Insérer/mettre à jour les données de localisation de l'abonné
 - `get_message_queue/2` - Récupérer les messages en attente de la file d'attente
 - `mark_dest_smss/3` - Marquer le message comme livré ou échoué
 - `add_event/3` - Ajouter un suivi d'événement pour les messages
 - `flush_queue/2` - Traiter les messages en attente (SRI-for-SM + MT-forwardSM)
 - `auto_flush/2` - Boucle de traitement continu de la file d'attente

2. SMS.FrontendRegistry

Gère l'enregistrement périodique du frontend auprès du backend :

- S'enregistre automatiquement au démarrage
 - Se réenregistre toutes les 5 minutes
 - Utilise `smsc_name` de la config (retombe sur le nom d'hôte)
 - Collecte des informations de configuration et de temps de fonctionnement du système

3. SMS_Utils

Fonctions utilitaires pour les opérations SMS :

- generate tp scts/0 - Générer un horodatage SMS au format TPDU

Flux de Messages SMS

Flux SMS Entrant (Mobile-Origin)

Flux SMS Sortant (Mobile-Terminé)

Étapes Clés Expliquées :

- **Demande SRI-for-SM** : Le SMSc interroge le HLR avec le MSISDN de destination pour déterminer où router le message SMS. Le HLR répond avec :
 - Un IMSI synthétique (calculé à partir du MSISDN pour la confidentialité) - voir [MSISDN → IMSI Mapping](#)
 - L'adresse GT du SMSc (numéro de noeud réseau) où le MT-ForwardSM doit être envoyé

- Pour des détails complets sur le fonctionnement de cela, voir [SRI-for-SM dans le Guide HLR](#)
- **Demande MT-forwardSM** : Une fois les informations de routage obtenues, le SMSc envoie le message SMS réel au MSC/VLR servant l'abonné

Structure du SMS TPDU

Gestion du Centre de Service d'Alerte

Le SMSc peut recevoir des messages **alertServiceCenter** du HLR pour suivre l'état de disponibilité des abonnés.

Pour des informations sur la façon dont le HLR envoie des messages **alertServiceCenter**, voir [Intégration du Centre de Service d'Alerte dans le Guide HLR](#).

Qu'est-ce que **alertServiceCenter** ?

Lorsqu'un abonné effectue une **UpdateLocation** au HLR (c'est-à-dire, s'enregistre avec un nouveau VLR/MSC), le HLR peut notifier les systèmes SMSc que l'abonné est maintenant joignable en envoyant un message **alertServiceCenter** (opcode MAP 64).

Configuration

Le temps d'expiration de la localisation est configuré dans le HLR :

```
config :omniss7,
  # Temps d'expiration de la localisation lorsque le SMSc reçoit alertServiceCenter (par défaut
: 48 heures)
  hlr_alert_location_expiry_seconds: 172800
```

Comportement

Lorsque le SMSc reçoit un message **alertServiceCenter** :

1. **Décoder MSISDN** : Extraire le MSISDN de l'abonné du message (format TBCD)
2. **Supprimer le préfixe TON/NPI** : Enlever les préfixes courants comme "19", "11", "91" (par exemple, "19123123213" → "123123213")
3. **Calculer IMSI** : Générer un IMSI synthétique en utilisant le même mappage que SRI-for-SM
4. **POST à /api/location** : Mettre à jour la base de données de localisation avec :
 - **msisdn**: Numéro de téléphone de l'abonné (nettoyé)
 - **imsi**: IMSI synthétique
 - **location**: Nom du SMSc (par exemple, "ipsmgw")
 - **expires**: Heure actuelle + **hlr_alert_location_expiry_seconds**
 - **csfb**: true (abonné joignable via Circuit-Switched Fallback)
 - **ims_capable**: false (il s'agit d'un enregistrement CS 2G/3G, pas IMS/VoLTE)
 - **user_agent**: GT du HLR qui a envoyé l'alerte (pour le suivi)
 - **ran_location**: "SS7"
5. **Suivre dans le Suivi des Abonnés SMSc** : Enregistrer l'abonné avec GT HLR, status=actif, compteurs de messages à 0
6. **Envoyer ACK** : Répondre au HLR avec un accusé de réception **alertServiceCenter**

Gestion des Abonnés Absents

Lorsque le SMSc tente de livrer un message et reçoit une erreur "abonné absent" lors de SRI-for-SM (pour plus d'informations sur SRI-for-SM, voir [SRI-for-SM dans le Guide HLR](#)) :

1. **Déetecter l'absence** : SRI-for-SM renvoie l'erreur **absentSubscriberDiagnosticSM**
2. **Expirer la localisation** : POST à /api/location avec **expires=0** pour marquer l'abonné comme injoignable
3. **Agent utilisateur** : Défini sur "SS7_AbsentSubscriber" pour identifier la source
4. **Mettre à jour le tracker** : Marquer l'abonné comme **failed** dans le Suivi des Abonnés SMSc

Cela garantit que la base de données de localisation et le tracker reflètent avec précision l'état de disponibilité des abonnés.

Diagramme de Flux

Point de Terminaison API

POST /api/location

```
{  
  "msisdn": "15551234567",  
  "imsi": "001010123456789",  
  "location": "ipsmgw",  
  "ims_capable": false,  
  "csfb": true,  
  "expires": "2025-11-01T12:00:00Z",  
  "user_agent": "15551111111",  
  "ran_location": "SS7",  
  "imei": "",  
  "registered": "2025-10-30T12:00:00Z"  
}
```

Remarque : Le champ `user_agent` contient le GT du HLR qui a envoyé l'alertServiceCenter, permettant au SMSc de suivre quel HLR fournit des mises à jour de localisation.

Pour les abonnés absents, `expires` est défini sur l'heure actuelle (expiration immédiate).

Prévention des Boucles

Le SMSc met en œuvre une **prévention automatique des boucles** pour éviter des boucles de routage de messages infinies lorsque les messages proviennent de réseaux SS7.

Pourquoi la Prévention des Boucles est Importante

Lorsque le SMSc reçoit des messages SMS d'origine mobile (MO) du réseau SS7, il les insère dans la file d'attente de messages avec un champ `source_smsc` identifiant leur origine (par exemple, "SS7_GT_15551234567"). Sans prévention des boucles, ces messages pourraient être :

1. Reçus du réseau SS7 → Mis en file d'attente avec `source_smsc` contenant "SS7"
2. Récupérés de la file d'attente → Traitement pour livraison
3. Renvois au réseau SS7 → Créant une boucle

Comment Cela Fonctionne

Le SMSc détecte et prévient automatiquement les boucles lors du traitement des messages :

Mise en Œuvre

Lors du traitement des messages de la file d'attente, le SMSc vérifie le champ `source_smsc` :

- **Si `source_smsc` contient "SS7" :**
 - Le message est ignoré
 - Événement ajouté : "Prévention des Boucles" avec une description expliquant la raison de l'ignorance
 - Message marqué comme échoué via une requête PUT
 - Journalisé avec un niveau d'avertissement
- **Sinon :**
 - Le message est traité normalement
 - Les opérations SRI-for-SM et MT-ForwardSM se poursuivent

Valeurs Source SMSC

Les messages peuvent avoir diverses valeurs `source_smsc` :

Source	Valeur Exemple	Action
Réseau SS7 (MO-FSM)	"SS7_GT_15551234567"	Ignoré - Prévention des boucles
API Externe/SMPP	"ipsmgw" ou "api_gateway"	Traité normalement
Autre SMSc	"smsc-node-01"	Traité normalement

Suivi des Événements

Lorsqu'un message est ignoré en raison de la prévention des boucles, un événement est enregistré :

```
{
  "message_id": 12345,
  "name": "Prévention des Boucles",
  "description": "Message ignoré - source_smssc 'SS7_GT_15551234567' contient 'SS7', empêchant la
boucle de message"
}
```

Cet événement est visible dans :

- **Interface Web** : Page des Événements SS7 (/events)
- **Base de Données** : Table events via l'API
- **Journaux** : Entrées de journal au niveau d'avertissement

Configuration

La prévention des boucles est **toujours activée** et ne peut pas être désactivée. C'est une fonctionnalité de sécurité critique pour prévenir les perturbations du réseau dues aux boucles de messages.

Scénario Exemple

Scénario : Un abonné mobile envoie un SMS via le réseau SS7

1. Téléphone mobile → MSC/VLR → SMSc (via M0-ForwardSM)
2. SMSc reçoit M0-FSM de GT 15551234567
3. SMSc insère dans la file d'attente : source_smssc = "SS7_GT_15551234567"
4. Le flush automatique récupère le message de la file d'attente
5. SMSc détecte "SS7" dans source_smssc → IGNORER
6. Événement enregistré : "Prévention des Boucles"
7. Message marqué comme échoué
8. Aucun SRI-for-SM ou MT-ForwardSM envoyé (boucle empêchée)

Sans prévention des boucles, l'étape 8 renverrait le message au réseau SS7, créant potentiellement une boucle infinie.

Suivi des Abonnés SMSc

Le SMSc comprend un GenServer **Suivi des Abonnés** qui maintient l'état en temps réel des abonnés basé sur les messages alertServiceCenter et les tentatives de livraison de messages.

Objectif

Le tracker fournit :

- **Surveillance de la Disponibilité** : Quels abonnés sont actuellement joignables
- **Suivi HLR** : Quel HLR a envoyé l'alertServiceCenter pour chaque abonné
- **Compteurs de Messages** : Nombre de messages envoyés/reçus par abonné
- **Suivi des Échecs** : Marquer les abonnés comme échoués lorsque les tentatives de livraison échouent
- **Visibilité de l'Interface Web** : Tableau de bord en temps réel montrant tous les abonnés suivis

Informations Suivies

Pour chaque abonné, le tracker stocke :

Champ	Description	Exemple
msisdn	Numéro de téléphone de l'abonné (clé)	"15551234567"

Champ	Description	Exemple
imsi	IMSI de l'abonné	"001010123456789"
hlr_gt	GT HLR qui a envoyé alertServiceCenter	"15551111111"
messages_sent	Compte des messages MT-FSM envoyés	5
messages_received	Compte des messages MO-FSM reçus	2
status	:active ou :failed	:active
updated_at	Horodatage Unix de la dernière mise à jour	1730246400

Transitions d'État

Comportement

Lorsque alertServiceCenter est reçu :

- Créer ou mettre à jour l'entrée de l'abonné
- Définir status = :active
- Enregistrer le GT HLR
- Réinitialiser ou préserver les compteurs de messages

Lorsque SRI-for-SM réussit :

- Incrémenter le compteur messages_sent
- Mettre à jour l'horodatage updated_at

Lorsque SRI-for-SM échoue :

- Définir status = :failed
- Garder dans le tracker pour surveillance

Lorsque l'abonné est supprimé :

- Supprimer de la table ETS
- N'apparaît plus dans l'interface Web

Interface Web - Page des Abonnés SMS

Chemin : /smsc_subscribers **Actualisation automatique :** Toutes les 2 secondes

Remarque : Cette page n'est disponible que lorsque vous exécutez en mode SMS. Après avoir décommenté la configuration SMS dans config/runtime.exs, vous devez redémarrer l'application pour que la route devienne disponible.

La page **Abonnés SMS** fournit une surveillance en temps réel de tous les abonnés suivis :

Fonctionnalités

1. Tableau des Abonnés

- MSISDN, IMSI, HLR GT
- Compteurs de messages envoyés/reçus
- Badge de statut (Actif/Échoué) avec codage couleur
- Horodatage de la dernière mise à jour et durée
- Bouton de suppression pour les abonnés individuels

2. Statistiques Résumées

- Total des abonnés suivis
- Compte des abonnés actifs
- Compte des abonnés échoués
- Nombre de HLR uniques

3. Actions

- Tout Effacer : Supprimer tous les abonnés suivis
- Supprimer : Supprimer un abonné individuel

Exemple de Vue

Abonnés Suivis SMSc			Total : 3	
MSISDN	IMSI	HLR GT	Msgs S/R	Statut
15551234567	001010123456789	15551111111	5/2	● Actif
15559876543	001010987654321	15551111111	0/0	● Actif
15551112222	001010111222233	15552222222	3/1	○ Échoué

Résumé : Total : 3 | Actifs : 2 | Échoués : 1 | HLR Uniques : 2

Fonctions API

Le tracker expose ces fonctions pour un accès programmatique :

```
# Appelé lorsque alertServiceCenter est reçu
SMSc.SubscriberTracker.alert_received(msisdn, imsi, hlr_gt)

# Incrémenter les compteurs de messages
SMSc.SubscriberTracker.message_sent(msisdn)
SMSc.SubscriberTracker.message_received(msisdn)

# Marquer comme échoué (échec SRI-for-SM)
SMSc.SubscriberTracker.mark_failed(msisdn)

# Supprimer du suivi
SMSc.SubscriberTracker.remove_subscriber(msisdn)

# Fonctions de requête
SMSc.SubscriberTracker.get_active_subscribers()
SMSc.SubscriberTracker.get_subscriber(msisdn)
SMSc.SubscriberTracker.count_subscribers()
SMSc.SubscriberTracker.clear_all()
```

Intégration

Le tracker est automatiquement intégré avec :

- **Gestionnaire alertServiceCenter** : Appelle `alert_received/3` lors de la mise à jour de localisation réussie
- **Gestionnaire SRI-for-SM** : Incrémente `messages_sent` lors du routage réussi
- **Gestionnaire Abonné Absent** : Appelle `mark_failed/1` lorsque l'abonné est absent
- **Erreurs d'abonnés inconnus** : Appelle `mark_failed/1` lorsque SRI-for-SM échoue

Configuration de Flush Automatique de la File d'Attente SMS

Le service **Flush Automatique** traite automatiquement les messages SMS en attente.

Pour la référence des paramètres de configuration, voir [Configuration de Flush Automatique dans la Référence de Configuration](#).

Configuration

```
config :omniss7,
  auto_flush_enabled: true,          # Activer/désactiver le flush automatique
  auto_flush_interval: 10_000,        # Intervalle de sondage en millisecondes
  auto_flush_dest_smsc: nil,         # Filtre : nil = tous
  auto_flush_tps: 10                # Max transactions par seconde
```

Comment Cela Fonctionne

1. **Sondage** : Toutes les `auto_flush_interval` millisecondes, interroge l'API pour les messages en attente
2. **Filtrage** : Filtrer éventuellement par `auto_flush_dest_smsc`
3. **Limitation de Taux** : Traiter jusqu'à `auto_flush_tps` messages par cycle
4. **Livraison** : Pour chaque message :
 - Envoyer **SRI-for-SM** (Envoyer les Informations de Routage pour le Message Court) au HLR pour obtenir des informations de routage
 - Le HLR renvoie un IMSI synthétique calculé à partir du MSISDN
 - Le HLR renvoie l'adresse GT du SMSC où le MT-ForwardSM doit être envoyé
 - Voir [Détails SRI-for-SM dans le Guide HLR](#) pour la documentation complète
 - En cas de succès, envoyer **MT-forwardSM** au MSC/VLR
 - Mettre à jour l'état du message via l'API (livré/échoué)
 - Ajouter un suivi d'événement via l'API

❖ **Plongée Technique** : Pour une explication complète de la façon dont SRI-for-SM fonctionne, y compris le mappage MSISDN à IMSI, la configuration de l'adresse GT du centre de service et la génération d'IMSI synthétique préservant la confidentialité, voir la [section SRI-for-SM dans le Guide de Configuration HLR](#).

Métriques SMSc

Métriques Disponibles

Métriques de la File d'Attente SMS :

- `smsc_queue_depth` - Nombre actuel de messages en attente
- `smsc_messages_delivered_total` - Total des messages livrés avec succès
- `smsc_messages_failed_total` - Total des messages ayant échoué à la livraison
- `smsc_delivery_duration_milliseconds` - Histogramme des temps de livraison

Exemples de Requêtes :

```
# Profondeur de la file d'attente actuelle
smsc_queue_depth

# Taux de succès de livraison (dernières 5 minutes)
rate(smsc_messages_delivered_total[5m]) /
(rate(smsc_messages_delivered_total[5m]) + rate(smsc_messages_failed_total[5m]))

# Temps de livraison moyen
rate(smsc_delivery_duration_milliseconds_sum[5m]) /
rate(smsc_delivery_duration_milliseconds_count[5m])
```

Dépannage SMSc

Problème : Messages Non Livrés

Vérifications :

1. Vérifiez que le flush automatique est activé
2. Vérifiez la connexion à la base de données
3. Surveillez les journaux pour des erreurs
4. Vérifiez que la connexion M3UA est ACTIVE
5. Vérifiez les limites TPS

Problème : Haute Profondeur de File d'Attente

Causes Possibles :

- Limite TPS trop basse
- Problèmes de délai d'attente HLR
- Problèmes de connectivité réseau
- Numéros de destination invalides

Solutions :

- Augmenter `auto_flush_tps`
 - Vérifier la disponibilité du HLR
 - Examiner les journaux des messages échoués
-

API MT-forwardSM

Envoyer un SMS via l'API

Point de terminaison API : POST /api/MT-forwardSM

Requête :

```
{  
  "imsi": "234509876543210",  
  "destination_serviceCentre": "447999555111",  
  "originating_serviceCenter": "447999123456",  
  "smsPDU": "040B917477218345F600001570301857140C0BD4F29C0E9281C4E1F11A"  
}
```

Réponse :

```
{  
  "result": "success",  
  "message_id": "12345"  
}
```

Documentation Connexe

Documentation OmniSS7 :

- [← Retour à la Documentation Principale](#)
- [Guide de Configuration HLR](#) - Configuration et opérations en mode HLR
 - [Détails Techniques SRI-for-SM](#) - Documentation complète sur le mappage MSISDN à IMSI et la configuration du centre de service
- [Guide des Fonctionnalités Communes](#) - Interface Web, API, Surveillance
- [Guide Client MAP](#) - Opérations MAP
- [Référence Technique](#) - Spécifications de protocole

Documentation OmniMessage : Pour la configuration du routage des messages, la gestion des files d'attente, le suivi de livraison, la limitation de taux et l'analytique, reportez-vous à la **documentation produit d'OmniMessage**. OmniMessage contient toute la logique de routage des messages, les algorithmes de réessay de files d'attente, la gestion des rapports de livraison et le moteur de règles métier.

OmniSS7 par Omnitouch Network Services



Guide de Configuration M3UA STP

[Retour à la Documentation Principale](#)

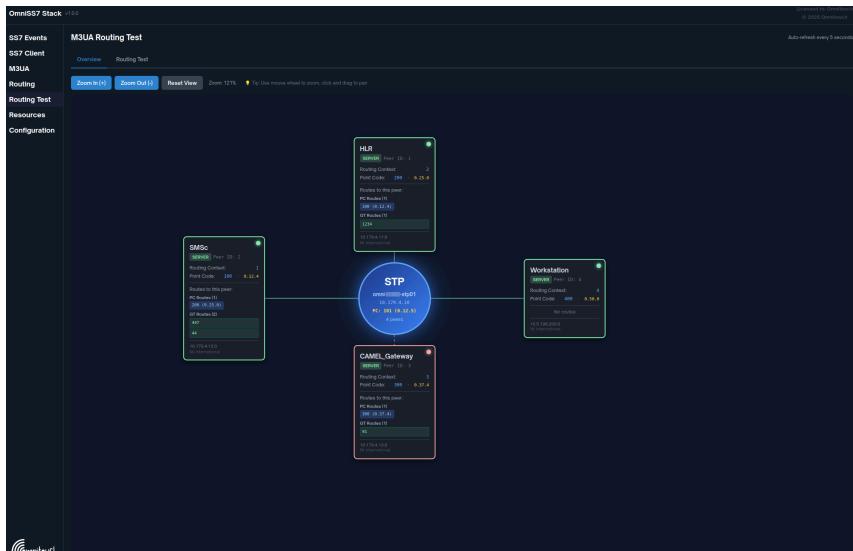
Ce guide fournit une configuration détaillée pour utiliser OmnisSST comme un **Point de Transfert de Signification (STP)**.

Table des Matières

1. [Ouverture session STP](#)
2. [Rôles Réseau STP](#)
3. [Activation du Mode STP](#)
4. [Configuration des Protocoles](#)
 - 5. [Gestion des Protocoles](#)
 - 5.1. [M3UA et MTP](#)
 - 5.1.1. [Gestion de M3UA via l'Interface Web](#)
 - 5.1.2. [Gestion de M3UA via l'Interface Web](#)
 - 6. [Résolution par Code de Point](#)
 - 7. [Fonctionnalités de Gestion des Routes](#)
 - 7.1. [Règles IPROUTE : Présentation des Règles des Routages](#)
 - 8. [Gestion de l'authentification](#)
 - 9. [Gestion des utilisateurs](#)
 - 10. [Gestion des groupes](#)
 - 11. [Gestion des autorisations](#)
 - 12. [Génération des journaux](#)

Qu'est-ce qu'un Point de Transfert de Signification (STP) ?

Un **Point de Transfert de Signification (STP)** est un élément réseau critique dans les réseaux de signification SS7 et basé sur IP qui achemine les messages de signification entre les nœuds du réseau.



Fonctions du STP

- **Routage de Messages** : Achemine le trafic de signification SS7 en fonction du Code de Point de Destination (PC) ou du Titre Global (GT)
- **Traduction de Protocole** : Relie les réseaux SS7 traditionnels aux réseaux M3UA/SIP basés sur IP
- **Management de Protocole** : Gère les protocoles de signification et de routage, et les règles de routage sur la priorité
- **Passerelle Réseau** : Connecte différents réseaux de signification et fournisseurs de services
- **Management de l'Authentification** : Gère les utilisateurs et les autorisations pour les différents protocoles et réseaux

Diagramme Réseau STP

Rôles Réseau STP Expliqués

- ASP (Processus Serveur d'Application)**
- **Rôle** : Client se connectant à un SGCP/STP distant
 - **Direction** : Connexion sortante
 - **Cas d'Utilisation** : Un STP se connecte à un STP d'un réseau partenaire

SGP (Processus Passerelle de Signification)

- **Rôle** : Serveur accueillant les connexions des ASP
- **Direction** : Connexion entrante
- **Cas d'Utilisation** : Plusieurs réseaux partenaires se connectent à votre STP

AS (Serveur d'Application)

- **Définition** : Regroupement logique d'un ou plusieurs ASP
- **Bal** : Fournit redondance et partage de charge
- **Cas d'Utilisation** : Plusieurs ASP servent la même destination

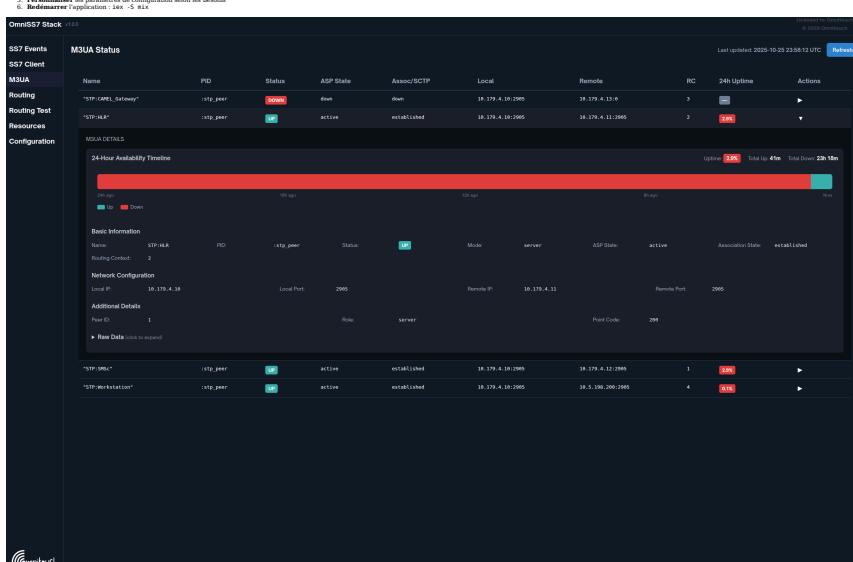
Activation du Mode M3UA STP

OmnisSST peut fonctionner en différents modes. Pour l'utiliser comme un STP vous devez activer le mode STP dans la configuration.

Changement en Mode STP

Le fichier config.routine.txt d'OmnisSST contient trois modes opérationnels préconfigurés. Pour activer le mode STP :

1. `Overwrite config.routine.txt`
2. Trouver les trois sections de configuration (lignes 53-174) :
 - Configuration 1 : Mode HLR (lignes 53-62)
 - Configuration 2 : Mode HLR (lignes 67-123)
 - Configuration 3 : Mode CAMEL Gateway (lignes 128-174)
3. **Commenter la configuration actuellement active (ajouter // à chaque ligne)**
4. **Activer la configuration 1 (lignes 53-62)**
5. **Personnaliser les paramètres de configuration selon les besoins**
6. **Réduire la taille de la configuration : ligne 5-84**



Configuration du Mode STP

La configuration STP complète ressemble à ceci :

```
config :omnisSST
# Définir le mode : Activer uniquement les fonctionnalités STP
stp_client_enabled: true
stp_node_enabled: false
stp_node_available: false
```



```

point_code: 300,
network_indicator: :international
},
{
peer_id: 4,
name: "M3UA Client",
role: :server,
remote_ip: "10.4.1.6, 8, 40",
remote_port: 0,
remote_point_code: 4,
point_code: 400,
network_indicator: :international
}
}

```

OmniS57 Stack v1.6.5

SS7 Events **M3UA Status** Last updated 2025-10-25 23:58:12 UTC Refresh

SS7 Client **M3UA**

Routing **Routing Test**

Resources **Configuration**

M3UA

Name	PID	Status	ASP State	Assoc/SCTP	Local	Remote	RC	24h Uptime	Actions
"STP-MLR-Gateway"	stp-peer	DOWN	down	down	10.179.4.16:2905	10.179.4.13:0	3		
"STP-MLR"	stp-peer	UP	active	established	10.179.4.16:2905	10.179.4.11:2905	2		

24-Hour Details

24-Hour Availability Timeline

UpTime: 245m Total Up: 45m Total Down: 22h15m

Basic Information

Name: STP-MLR PID: 100 Status:

Routing Context: 2 Mode: server ASP State: active Association State: established

Network Configuration

Local IP: 10.179.4.38 Local Port: 2905 Remote IP: 10.179.4.31 Remote Port: 2905

Additional Details

Peer ID: 1 Role: server Port Code: 2905

M3UA

Name	PID	Status	ASP State	Assoc/SCTP	Local	Remote	RC	24h Uptime	Actions
"STP-MLR"	stp-peer	UP	active	established	10.179.4.16:2905	10.179.4.13:2905	1		
"STP-Workstation"	stp-peer	UP	active	established	10.179.4.16:2905	10.5.190.200:2945	4		

Paramètres de Configuration des Pairs

Paramètre	Type	Requis	Description
peer_id	String	Non	Identifiant unique pour ce pair.
name	String	Obligatoire	Nom visible pour l'honneur pour les journaux et la surveillance.
role	String	Obligatoire	Client (client) ou :server (entrant).
local_ip	String	Obligatoire	Adresse IP (client) / Port local (0 pour dynamique)
local_port	String	Obligatoire	Port de pair local (0 pour entrant = accepter de n'importe quel port source)
remote_ip	String	Obligatoire	Port de pair distant (0 pour entrant = accepter de n'importe quel port source)
routing_context	String	Obligatoire	Contexte de routage de l'application de routage M3UA
point_code	String	Obligatoire	Code de point de SS7 de ce port
network_indicator	String	Obligatoire	:international ou :national

Filtrage du Port Source pour les Connexions Entrantes

Pour les connexions entrantes (rôle : :server), le paramètre `remote_port` contrôle le filtrage de port source :

- Port Spécifique** (par exemple, `remote_port: 2905`) : Accepter que les connexions de ce port source exact
- Toutes les Sources** : Accepter toutes les connexions de port source
- N'importe quel Port** (`remote_port: 0`) : Accepter les connexions de n'importe quel port source
- Utilise le port distant** : Utilise des port sources dynamiques/éphémères
- Ne valide pas l'adresse IP source**
- Plus facile mais également moins sécurisé**

Exemple :

```

# Accepter uniquement de 10.5.108.200:2905 (port spécifique)
peer_id: 1
name: "M3UA_Peer"
role: :server
remote_ip: "10.5.108.200",
remote_port: 2905
# ... autre config

# Accepter de 10.5.108.200 avec n'importe quel port source
peer_id: 2
name: "M3UA_Peer"
role: :server
remote_ip: "10.5.108.200",
remote_port: 0, # Accepter de n'importe quel port source
# ... autre config
}

```

Support du Protocole M2PA

OmniSS7 prend en charge à la fois les protocoles **M3UA** et **M2PA** pour le transport de signification SS7.

Qu'est-ce que M2PA ?

M2PA (MTP User Peer-to-Peer Adaptation Layer) est un protocole standardisé par l'IETF (RFC 4165) pour transporter les messages MTP3 SS7 sur des réseaux IP en utilisant SCTP.

M3UA vs M2PA : Principales Différences

Fonctionnalité	M3UA	M2PA
Architecture	Client-Server (MLR/MLR)	Pair à Pair
Cas d'utilisation	Passerelle entre SS7 et IP	Passerelle entre SS7 et IP
Gestion de l'État de Liaison	Applications (ASUP/ISUP/Q7MFC) / SS7 Direct (24h) / Verifications (PdH)	Applications (ASUP/ISUP/Q7MFC) / SS7 Direct (24h) / Verifications (PdH)
Nombre de Pairs	Plusieurs (MLR/MLR)	Plusieurs (MLR/MLR)
Déploiement Typique	Passerelle SS7 vers IP / STP	Liens de signification directs entre nœuds
RFC	RFC 4466	RFC 4165

Conseils de Sélection de Protocole

Recommendation : Utilisez M3UA par défaut. N'utilisez M2PA que lorsque cela est spécifiquement requis.

Quand Utiliser M3UA (Recommendé)

M3UA est le protocole recommandé pour la plupart des déploiements :

- Déploiement STP** : Connexions directes point à point entre Points de Transfert de Signification dans un réseau multi-SS7.
- Remplacement TDM Héritage** : Remplacez les liens TDM SST traditionnels lorsque le système distant exige spécifiquement M2PA.
- Connexions d'éléments de Réseau** : Connectez HLR, MSC, SMSC, et tous les éléments de réseau à votre STP.
- Passerelle entre SS7 et IP** : Utilisez M3UA pour les connexions entre les systèmes de transmission de plusieurs Services d'Application.
- Opérateurs Flexibles** : Architectures client/serveur avec contrôle centralisé
- Interconnection avec les réseaux de l'opérateur** : Utilisez M3UA pour connecter les éléments de réseau (HLR, MSC, SMSC, VLR, etc.) à votre STP.

Utilisez M3UA pour connecter les éléments de réseau (HLR, MSC, SMSC, VLR, etc.) à votre STP.

Quand Utiliser M2PA (Cas Spéciaux Unique)

M2PA ne doit être utilisé que dans des scénarios spécifiques :

- Liens STP à STP** : Connexions directes point à point entre Points de Transfert de Signification dans un réseau multi-SS7.
- Remplacement TDM Héritage** : Remplacez les liens TDM SST traditionnels lorsque le système distant exige spécifiquement M2PA.
- Connexions d'éléments de Réseau** : Connectez HLR, MSC, SMSC, et tous les éléments de réseau à votre STP.
- Passerelle entre SS7 et IP** : Utilisez M2PA pour les connexions entre les systèmes de transmission de plusieurs Services d'Application.
- Opérateurs Flexibles** : Architectures client/serveur avec contrôle centralisé
- Interconnection avec les réseaux de l'opérateur** : Utilisez M2PA pour connecter les éléments de réseau (HLR, MSC, SMSC, VLR, etc.) à votre STP.

Important : Ne utilisez M2PA pour connecter des éléments de réseau (HLR, MSC, SMSC, VLR, etc.) à votre STP : M2PA est conçu pour les interconnexions STP à STP où les deux côtés fonctionnent comme des nœuds de routage.

Configuration des Pairs M2PA

Tous les pairs M2PA sont configurés de la même manière que les pairs M3UA, avec un paramètre protocole supplémentaire.

Configuration des Pairs M2PA

Ajoutez des pairs M2PA à votre configuration `sta_peers` dans `config/runtime.exe` (ou, si partagent la même section de configuration malgré le fait qu'ils soient des protocoles différents) :

Paramètres Clés pour M2PA :

Paramètre	Valeur	Description
protocol	:m2pa	Spécifie le protocole M2PA (par défaut :cha si omis)
role	:client ou :server	Dirigeant de la connexion
local_ip	Entrez	Port SCTP distant (par défaut M2PA est 3545)
remote_ip	Entrez	Port SCTP distant (par défaut M2PA est 3545)
point_code	Entrez	Votre code de point
adjacent_codepoint	Entrez	Code de point distant (spécifique à M2PA)

Remarque : M2PA utilise le port **3545** comme nom de l'indicateur (différent du port 2905 de M3UA).

Etat de Link M2PA

Tous les liens M2PA proposent à travers plusieurs états lors de l'utilisation :

- Down** : Aucune connexion établie.
- Alignment** : Phase de synchronisation initiale (-1 secondes)
- Established** : Connexion établie (durée de vie moyenne de 1-2 secondes)
- Ready** : Link actif et prêt pour le traffic.

La progression de l'état via une signalisation fiable avant l'échange de trafic.

Gestion des Pairs M2PA via l'Interface Web

La page **Routage** dans l'Interface Web fournit un support complet pour la gestion des pairs M2PA :

Navigation vers la page de Routage

Modifier les Pairs M2PA

Créer un "Ajouter un Nouveau Pair"

- Remplir la configuration du pair** avec les informations de base et les paramètres détaillés
- Protocole** : M2PA
- Role** : :client ou :server
- Code de Point** (votre PC)
- Adresse IP** (votre PC)
- Port Local** (votre PC)
- Port Lucas/Distant** (typiquement 3545 pour M2PA)
- Indicateur** (code de point international ou national)

Cliquer sur "Enregistrer le Pair"

La table des pairs affiche le type de protocole avec un code couleur :

Blanc : pair M3UA

Violet : pair M2PA

Comportement de Routage M2PA

Les pairs M2PA s'intègrent parfaitement en système de routage d'OmniSS7 :

- Routage de Code de Point** : Fonctionnement identique pour M2PA et M3UA
- Routage de Titre Global** : Entièrement supportées sur les liens M2PA

- Priorité de Route** : Les paires M2PA et MS3UA peuvent être mélangées dans les mêmes tables de routage
- schéma de Message** : Les messages peuvent arriver sur M2PA et être routés vers MS3UA, et vice versa

Métriques M2PA

M2PA fournit des métriques Prometheus compilées pour surveiller la santé et le trafic des liens :

- Métriques de Trafic :
- `m2pa_message_sent_total` : Total des messages MTP2 envoyés par lien
 - `m2pa_message_received_total` : Total des messages MTP2 reçus par lien
 - `m2pa_bitrate_sent` : Taux de transmission sur le lien
 - `m2pa_bitrate_received_total` : Total des octets reçus sur M2PA

Toutes les métriques de trafic sont étiquetées par `link_name, point_code, adjacent_pc`

Métriques d'état de lien :

- `m2pa_link_state_change_total` : Transitions d'état de lien (DOWN → ALIGNMENT → PROVING → READY)
- `m2pa_link_state_link_name_from_ipk_to_ipk`

Métriques d'erreur :

- `m2pa_errors_total` : Total des erreurs par type
- `decode_error` : Erreur de décodage de message M2PA
- `receive_error` : Erreur de réception de message M2PA
- `scsi_send_error` : Erreurs de transmission SCSI
- `transmit_error` : Erreurs de transmission

Accès aux Métriques :

- Point de terminaison Prometheus : <http://your-server:8080/metrics>
- Les métriques s'enregistrent automatiquement au démarrage de l'application

Méilleures Pratiques M2PA

1. Sélection de Port : Utilisez un port 1565 pour M2PA (norme de l'industrie)

2. Sélection de Port : Assurez-vous que les changements d'état de liens via les métriques

3. Règles de Pare-feu : Assurez-vous que SCTP protocole IP (122) est autorisé

4. Règles de Pare-feu : Assurez-vous que les règles de pare-feu sont correctement configurées des deux côtés

5. Indicateur de Réseau : Doit correspondre entre les paires (international ou national)

6. Indicateur de Réseau : Utilisez la paire de test de Routage pour vérifier la connectivité après configuration

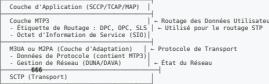
Configuration du Routage par Code de Point

Le routage par Code de Point dirige les messages en fonction de **Code de Point de Destination (DPC)** dans l'en-tête MTP3.

Comprendre les Codes de Point dans la Pile de Protocole SS7

Les codes de point existent à différents niveaux de la pile de protocole SS7. Comprendre cette distinction est important :

Couches de la Pile de Protocole :



Divers Types de Codes de Point :

- Codes de Point de Couche MTP3** (Utilisés pour le Routage) :
 - Situés dans l'en-tête de routage MTP3 (DPC, OPC)
 - Présents dans le paramètre Domains de Protocole (tag 528)
 - Présents dans les messages de Domains Utilisés M2PA
 - Le MTP utilise ces valeurs DPC pour les décisions de routage
 - Utilisées pour la recherche des routes

- Codes de Point de Couche MS3A** (Utilisés pour la Gestion de Réseau) :
 - Présents dans les messages de Domains MS3A (DUNA, DAVA, SCON, DUP)
 - Indiquent les codes de point affectés pour l'état de réseau
 - Utilisées pour la recherche des routes
 - Non utilisées pour la recherche des domaines utilisateurs

Comment Fonctionne le Routage

- Pour les messages MS3A DATA, le STP extrait le message MTP3 du paramètre Domains du Protocole (tag 528), qui contient l'en-tête de routage MTP3 (DPC, OPC, SLS). Le DPC de la couche MTP3 est utilisé pour rechercher des routes.
- Pour les messages de Domains Utilisés M2PA, le STP extrait le message MTP3 du champ de données utilisateur M2PA, puis le DPC de l'en-tête de routage MTP3.
- Le STP utilise ces valeurs DPC pour les décisions de routage

Routes de Code de Point de Base

Agence des routes à config/routing.exe :

```
config routing
  class routes:
    # Route tout le trafic pour le PC 100 vers le pair 1 (STP Partenaire)
    {100, 1} route 100
    # Route tout le trafic pour le PC 200 vers le pair 2 (STP Partenaire)
    {200, 2} route 200
    # Route tout le trafic pour le PC 300 avec priorité 1
    {300, 1} route 300 priority 1
    # Route tout le trafic pour le PC 300 avec priorité 2
    {300, 2} route 300 priority 2
    # Exemple d'équilibrage de charge : PC 300 avec routes principales et de secours
    {300, 1} route 300
    {300, 2} route 300 priority 1
    {300, 3} route 300 priority 2
    {300, 4} route 300 priority 3
    {300, 5} route 300 priority 4
    {300, 6} route 300 priority 5
    {300, 7} route 300 priority 6
    {300, 8} route 300 priority 7
    {300, 9} route 300 priority 8
    {300, 10} route 300 priority 9
    {300, 11} route 300 priority 10
    {300, 12} route 300 priority 11
    {300, 13} route 300 priority 12
    {300, 14} route 300 priority 13
    {300, 15} route 300 priority 14
    {300, 16} route 300 priority 15
    {300, 17} route 300 priority 16
    {300, 18} route 300 priority 17
    {300, 19} route 300 priority 18
    {300, 20} route 300 priority 19
    {300, 21} route 300 priority 20
    {300, 22} route 300 priority 21
    {300, 23} route 300 priority 22
    {300, 24} route 300 priority 23
    {300, 25} route 300 priority 24
    {300, 26} route 300 priority 25
    {300, 27} route 300 priority 26
    {300, 28} route 300 priority 27
    {300, 29} route 300 priority 28
    {300, 30} route 300 priority 29
    {300, 31} route 300 priority 30
    {300, 32} route 300 priority 31
    {300, 33} route 300 priority 32
    {300, 34} route 300 priority 33
    {300, 35} route 300 priority 34
    {300, 36} route 300 priority 35
    {300, 37} route 300 priority 36
    {300, 38} route 300 priority 37
    {300, 39} route 300 priority 38
    {300, 40} route 300 priority 39
    {300, 41} route 300 priority 40
    {300, 42} route 300 priority 41
    {300, 43} route 300 priority 42
    {300, 44} route 300 priority 43
    {300, 45} route 300 priority 44
    {300, 46} route 300 priority 45
    {300, 47} route 300 priority 46
    {300, 48} route 300 priority 47
    {300, 49} route 300 priority 48
    {300, 50} route 300 priority 49
    {300, 51} route 300 priority 50
    {300, 52} route 300 priority 51
    {300, 53} route 300 priority 52
    {300, 54} route 300 priority 53
    {300, 55} route 300 priority 54
    {300, 56} route 300 priority 55
    {300, 57} route 300 priority 56
    {300, 58} route 300 priority 57
    {300, 59} route 300 priority 58
    {300, 60} route 300 priority 59
    {300, 61} route 300 priority 60
    {300, 62} route 300 priority 61
    {300, 63} route 300 priority 62
    {300, 64} route 300 priority 63
    {300, 65} route 300 priority 64
    {300, 66} route 300 priority 65
    {300, 67} route 300 priority 66
    {300, 68} route 300 priority 67
    {300, 69} route 300 priority 68
    {300, 70} route 300 priority 69
    {300, 71} route 300 priority 70
    {300, 72} route 300 priority 71
    {300, 73} route 300 priority 72
    {300, 74} route 300 priority 73
    {300, 75} route 300 priority 74
    {300, 76} route 300 priority 75
    {300, 77} route 300 priority 76
    {300, 78} route 300 priority 77
    {300, 79} route 300 priority 78
    {300, 80} route 300 priority 79
    {300, 81} route 300 priority 80
    {300, 82} route 300 priority 81
    {300, 83} route 300 priority 82
    {300, 84} route 300 priority 83
    {300, 85} route 300 priority 84
    {300, 86} route 300 priority 85
    {300, 87} route 300 priority 86
    {300, 88} route 300 priority 87
    {300, 89} route 300 priority 88
    {300, 90} route 300 priority 89
    {300, 91} route 300 priority 90
    {300, 92} route 300 priority 91
    {300, 93} route 300 priority 92
    {300, 94} route 300 priority 93
    {300, 95} route 300 priority 94
    {300, 96} route 300 priority 95
    {300, 97} route 300 priority 96
    {300, 98} route 300 priority 97
    {300, 99} route 300 priority 98
    {300, 100} route 300 priority 99
    {300, 101} route 300 priority 100
    {300, 102} route 300 priority 101
    {300, 103} route 300 priority 102
    {300, 104} route 300 priority 103
    {300, 105} route 300 priority 104
    {300, 106} route 300 priority 105
    {300, 107} route 300 priority 106
    {300, 108} route 300 priority 107
    {300, 109} route 300 priority 108
    {300, 110} route 300 priority 109
    {300, 111} route 300 priority 110
    {300, 112} route 300 priority 111
    {300, 113} route 300 priority 112
    {300, 114} route 300 priority 113
    {300, 115} route 300 priority 114
    {300, 116} route 300 priority 115
    {300, 117} route 300 priority 116
    {300, 118} route 300 priority 117
    {300, 119} route 300 priority 118
    {300, 120} route 300 priority 119
    {300, 121} route 300 priority 120
    {300, 122} route 300 priority 121
    {300, 123} route 300 priority 122
    {300, 124} route 300 priority 123
    {300, 125} route 300 priority 124
    {300, 126} route 300 priority 125
    {300, 127} route 300 priority 126
    {300, 128} route 300 priority 127
    {300, 129} route 300 priority 128
    {300, 130} route 300 priority 129
    {300, 131} route 300 priority 130
    {300, 132} route 300 priority 131
    {300, 133} route 300 priority 132
    {300, 134} route 300 priority 133
    {300, 135} route 300 priority 134
    {300, 136} route 300 priority 135
    {300, 137} route 300 priority 136
    {300, 138} route 300 priority 137
    {300, 139} route 300 priority 138
    {300, 140} route 300 priority 139
    {300, 141} route 300 priority 140
    {300, 142} route 300 priority 141
    {300, 143} route 300 priority 142
    {300, 144} route 300 priority 143
    {300, 145} route 300 priority 144
    {300, 146} route 300 priority 145
    {300, 147} route 300 priority 146
    {300, 148} route 300 priority 147
    {300, 149} route 300 priority 148
    {300, 150} route 300 priority 149
    {300, 151} route 300 priority 150
    {300, 152} route 300 priority 151
    {300, 153} route 300 priority 152
    {300, 154} route 300 priority 153
    {300, 155} route 300 priority 154
    {300, 156} route 300 priority 155
    {300, 157} route 300 priority 156
    {300, 158} route 300 priority 157
    {300, 159} route 300 priority 158
    {300, 160} route 300 priority 159
    {300, 161} route 300 priority 160
    {300, 162} route 300 priority 161
    {300, 163} route 300 priority 162
    {300, 164} route 300 priority 163
    {300, 165} route 300 priority 164
    {300, 166} route 300 priority 165
    {300, 167} route 300 priority 166
    {300, 168} route 300 priority 167
    {300, 169} route 300 priority 168
    {300, 170} route 300 priority 169
    {300, 171} route 300 priority 170
    {300, 172} route 300 priority 171
    {300, 173} route 300 priority 172
    {300, 174} route 300 priority 173
    {300, 175} route 300 priority 174
    {300, 176} route 300 priority 175
    {300, 177} route 300 priority 176
    {300, 178} route 300 priority 177
    {300, 179} route 300 priority 178
    {300, 180} route 300 priority 179
    {300, 181} route 300 priority 180
    {300, 182} route 300 priority 181
    {300, 183} route 300 priority 182
    {300, 184} route 300 priority 183
    {300, 185} route 300 priority 184
    {300, 186} route 300 priority 185
    {300, 187} route 300 priority 186
    {300, 188} route 300 priority 187
    {300, 189} route 300 priority 188
    {300, 190} route 300 priority 189
    {300, 191} route 300 priority 190
    {300, 192} route 300 priority 191
    {300, 193} route 300 priority 192
    {300, 194} route 300 priority 193
    {300, 195} route 300 priority 194
    {300, 196} route 300 priority 195
    {300, 197} route 300 priority 196
    {300, 198} route 300 priority 197
    {300, 199} route 300 priority 198
    {300, 200} route 300 priority 199
    {300, 201} route 300 priority 200
    {300, 202} route 300 priority 201
    {300, 203} route 300 priority 202
    {300, 204} route 300 priority 203
    {300, 205} route 300 priority 204
    {300, 206} route 300 priority 205
    {300, 207} route 300 priority 206
    {300, 208} route 300 priority 207
    {300, 209} route 300 priority 208
    {300, 210} route 300 priority 209
    {300, 211} route 300 priority 210
    {300, 212} route 300 priority 211
    {300, 213} route 300 priority 212
    {300, 214} route 300 priority 213
    {300, 215} route 300 priority 214
    {300, 216} route 300 priority 215
    {300, 217} route 300 priority 216
    {300, 218} route 300 priority 217
    {300, 219} route 300 priority 218
    {300, 220} route 300 priority 219
    {300, 221} route 300 priority 220
    {300, 222} route 300 priority 221
    {300, 223} route 300 priority 222
    {300, 224} route 300 priority 223
    {300, 225} route 300 priority 224
    {300, 226} route 300 priority 225
    {300, 227} route 300 priority 226
    {300, 228} route 300 priority 227
    {300, 229} route 300 priority 228
    {300, 230} route 300 priority 229
    {300, 231} route 300 priority 230
    {300, 232} route 300 priority 231
    {300, 233} route 300 priority 232
    {300, 234} route 300 priority 233
    {300, 235} route 300 priority 234
    {300, 236} route 300 priority 235
    {300, 237} route 300 priority 236
    {300, 238} route 300 priority 237
    {300, 239} route 300 priority 238
    {300, 240} route 300 priority 239
    {300, 241} route 300 priority 240
    {300, 242} route 300 priority 241
    {300, 243} route 300 priority 242
    {300, 244} route 300 priority 243
    {300, 245} route 300 priority 244
    {300, 246} route 300 priority 245
    {300, 247} route 300 priority 246
    {300, 248} route 300 priority 247
    {300, 249} route 300 priority 248
    {300, 250} route 300 priority 249
    {300, 251} route 300 priority 250
    {300, 252} route 300 priority 251
    {300, 253} route 300 priority 252
    {300, 254} route 300 priority 253
    {300, 255} route 300 priority 254
    {300, 256} route 300 priority 255
    {300, 257} route 300 priority 256
    {300, 258} route 300 priority 257
    {300, 259} route 300 priority 258
    {300, 260} route 300 priority 259
    {300, 261} route 300 priority 260
    {300, 262} route 300 priority 261
    {300, 263} route 300 priority 262
    {300, 264} route 300 priority 263
    {300, 265} route 300 priority 264
    {300, 266} route 300 priority 265
    {300, 267} route 300 priority 266
    {300, 268} route 300 priority 267
    {300, 269} route 300 priority 268
    {300, 270} route 300 priority 269
    {300, 271} route 300 priority 270
    {300, 272} route 300 priority 271
    {300, 273} route 300 priority 272
    {300, 274} route 300 priority 273
    {300, 275} route 300 priority 274
    {300, 276} route 300 priority 275
    {300, 277} route 300 priority 276
    {300, 278} route 300 priority 277
    {300, 279} route 300 priority 278
    {300, 280} route 300 priority 279
    {300, 281} route 300 priority 280
    {300, 282} route 300 priority 281
    {300, 283} route 300 priority 282
    {300, 284} route 300 priority 283
    {300, 285} route 300 priority 284
    {300, 286} route 300 priority 285
    {300, 287} route 300 priority 286
    {300, 288} route 300 priority 287
    {300, 289} route 300 priority 288
    {300, 290} route 300 priority 289
    {300, 291} route 300 priority 290
    {300, 292} route 300 priority 291
    {300, 293} route 300 priority 292
    {300, 294} route 300 priority 293
    {300, 295} route 300 priority 294
    {300, 296} route 300 priority 295
    {300, 297} route 300 priority 296
    {300, 298} route 300 priority 297
    {300, 299} route 300 priority 298
    {300, 300} route 300 priority 299
    {300, 301} route 300 priority 300
    {300, 302} route 300 priority 301
    {300, 303} route 300 priority 302
    {300, 304} route 300 priority 303
    {300, 305} route 300 priority 304
    {300, 306} route 300 priority 305
    {300, 307} route 300 priority 306
    {300, 308} route 300 priority 307
    {300, 309} route 300 priority 308
    {300, 310} route 300 priority 309
    {300, 311} route 300 priority 310
    {300, 312} route 300 priority 311
    {300, 313} route 300 priority 312
    {300, 314} route 300 priority 313
    {300, 315} route 300 priority 314
    {300, 316} route 300 priority 315
    {300, 317} route 300 priority 316
    {300, 318} route 300 priority 317
    {300, 319} route 300 priority 318
    {300, 320} route 300 priority 319
    {300, 321} route 300 priority 320
    {300, 322} route 300 priority 321
    {300, 323} route 300 priority 322
    {300, 324} route 300 priority 323
    {300, 325} route 300 priority 324
    {300, 326} route 300 priority 325
    {300, 327} route 300 priority 326
    {300, 328} route 300 priority 327
    {300, 329} route 300 priority 328
    {300, 330} route 300 priority 329
    {300, 331} route 300 priority 330
    {300, 332} route 300 priority 331
    {300, 333} route 300 priority 332
    {300, 334} route 300 priority 333
    {300, 335} route 300 priority 334
    {300, 336} route 300 priority 335
    {300, 337} route 300 priority 336
    {300, 338} route 300 priority 337
    {300, 339} route 300 priority 338
    {300, 340} route 300 priority 339
    {300, 341} route 300 priority 340
    {300, 342} route 300 priority 341
    {300, 343} route 300 priority 342
    {300, 344} route 300 priority 343
    {300, 345} route 300 priority 344
    {300, 346} route 300 priority 345
    {300, 347} route 300 priority 346
    {300, 348} route 300 priority 347
    {300, 349} route 300 priority 348
    {300, 350} route 300 priority 349
    {300, 351} route 300 priority 350
    {300, 352} route 300 priority 351
    {300, 353} route 300 priority 352
    {300, 354} route 300 priority 353
    {300, 355} route 300 priority 354
    {300, 356} route 300 priority 355
    {300, 357} route 300 priority 356
    {300, 358} route 300 priority 357
    {300, 359} route 300 priority 358
    {300, 360} route 300 priority 359
    {300, 361} route 300 priority 360
    {300, 362} route 300 priority 361
    {300, 363} route 300 priority 362
    {300, 364} route 300 priority 363
    {300, 365} route 300 priority 364
    {300, 366} route 300 priority 365
    {300, 367} route 300 priority 366
    {300, 368} route 300 priority 367
    {300, 369} route 300 priority 368
    {300, 370} route 300 priority 369
    {300, 371} route 300 priority 370
    {300, 372} route 300 priority 371
    {300, 373} route 300 priority 372
    {300, 374} route 300 priority 373
    {300, 375} route 300 priority 374
    {300, 376} route 300 priority 375
    {300, 377} route 300 priority 376
    {300, 378} route 300 priority 377
    {300, 379} route 300 priority 378
    {300, 380} route 300 priority 379
    {300, 381} route 300 priority 380
    {300, 382} route 300 priority 381
    {300, 383} route 300 priority 382
    {300, 384} route 300 priority 383
    {300, 385} route 300 priority 384
    {300, 386} route 300 priority 385
    {300, 387} route 300 priority 386
    {300, 388} route 300 priority 387
    {300, 389} route 300 priority 388
    {300, 390} route 300 priority 389
    {300, 391} route 300 priority 390
    {300, 392} route 300 priority 391
    {300, 393} route 300 priority 392
    {300, 394} route 300 priority 393
    {300, 395} route 300 priority 394
    {300, 396} route 300 priority 395
    {300, 397} route 300 priority 396
    {300, 398} route 300 priority 397
    {300, 399} route 300 priority 398
    {300, 400} route 300 priority 399
    {300, 401} route 300 priority 400
    {300, 402} route 300 priority 401
    {300, 403} route 300 priority 402
    {300, 404} route 300 priority 403
    {300, 405} route 300 priority 404
    {300, 406} route 300 priority 405
    {300, 407} route 300 priority 406
    {300, 408} route 300 priority 407
    {300, 409} route 300 priority 408
    {300, 410} route 300 priority 409
    {300, 411} route 300 priority 410
    {300, 412} route 
```

```

%{
dest_pc: 1000;
peer_id: 1;
priority: 1;
mask: 1;
network_indicator: :international
}
# Masque par défaut à 14 - Correspond à : Seulement PC 1000
# Masque explicite (même résultat)
%{
dest_pc: 1000;
peer_id: 1;
priority: 1;
mask: 1;
network_indicator: :international
}
# Correspond à : Seulement PC 1000

```

Exemple 2 : Petite Plage

```

%{
dest_pc: 1000;
peer_id: 2;
priority: 1;
mask: 12;
network_indicator: :international
}
# Correspond à : PC 1000, 1001, 1002, 1003

```

```

Example 3 : Plage Moyenne
%{
dest_pc: 1000;
peer_id: 1;
priority: 1;
mask: 15;
network_indicator: :international
}
# Correspond à : PC 1000-1063 (64 codes de point consécutifs)

```

Exemple 4 : Route par Défaut/De Secours

```

%{
dest_pc: 0;
peer_id: 10;
priority: 10;
mask: 1;
network_indicator: :international
}
# Correspond à : Tous les codes de point (0-16383)
# Utiliser comme une route de secours/route-all avec une faible priorité

```

Combinaison de Routes Spécifiques et Masques

Vous pouvez combiner des routes spécifiques avec des routes masquées pour un routage flexible :

```

config omnisis!
  # Route spécifique pour le PC 1000 (prend la priorité)
  %{
  dest_pc: 1000;
  peer_id: 1;
  priority: 1;
  mask: 1;
  network_indicator: :international
  }
  # masque par défaut à 14 (correspondance exacte)
  %{
  dest_pc: 1000;
  peer_id: 1;
  priority: 1;
  mask: 15;
  network_indicator: :international
  }
  # Route de plage pour les PCs 1000-1063
  %{
  dest_pc: 1000;
  peer_id: 1;
  priority: 1;
  mask: 5;
  network_indicator: :international
  }
  # Route par défaut/de secours pour tous les autres PCs
  %{
  dest_pc: 0;
  peer_id: 10;
  priority: 10;
  mask: 1;
  network_indicator: :international
  }

```

Définition de Routage pour DPC 1000 :

1. Correspond à la route masque /14 (PC 1000 exactement) - **Sélectionnée** (la plus spécifique)
2. Correspond également à la route masque /8 (plage PC 1000-1063) - Ignorée (la moins spécifique)
3. Correspond également à la route masque /0 (tous les PCs) - Ignorée (la moins spécifique)

Définition de Routage pour DPC 1015 :

1. Correspond à la route masque /14 (PC 1000 exactement)
2. Correspond à la route masque /8 (plage PC 1000-1063) - **Sélectionnée** (correspondance la plus spécifique)
3. Correspond également à la route masque /0 (tous les PCs) - Ignorée (la moins spécifique)

Définition de Routage pour DPC 5000 :

1. Ne correspond pas à la route masque /14
2. Ne correspond pas à la route masque /8
3. Correspond à la route masque /0 (tous les PCs) - **Sélectionnée** (seule correspondance, route de secours)

Méilleures Pratiques

- Obtenir des routes pour les Destinations Uniques : Pour des correspondances exactes de route de point, utiliser complètement le champ **mask** (par défaut à 15)
- Utiliser /14 Explicitement Seulement Route Nécessaire : Ne spécifiez **mask** : 14 que lorsque vous devez le clarifier dans la documentation ou lorsque vous mélangez avec des routes de plage
- Utiliser des Masques de Plage pour des Blocs Réseaux : Routage de segments de réseaux entiers vers des pairs spécifiques avec des masques /6 à /15
- Utiliser des masques de plage pour les routes de destination
- La Plus Spécifique Gagne : Le moteur de routage sélectionne toujours la correspondance de route la plus spécifique (valeur de masque la plus élevée) en premier
- Préfixe comme Critère de Détermination : Si plusieurs routes ont la même masque, le moteur de prétend la plus bas (empêtre)

Configuration du Routage par Titre Global (GT)

Le routage par Titre Global permet un routage basé sur le **contenu** en utilisant des numéros de téléphone ou des valeurs IMSI au lieu de codes de point. Pour un routage avancé par Titre Global basé sur l'appelant/à l'appelé, voir le [Guide NAT du Titre Global](#).

Prérequis

- Activer le routage GT : `enable_gt_routing: true` dans `config/runtime.exe`

Configuration des Routes GT

```

config omnisis!
  # Activation du routage GT
  enable_gt_routing: true
  mbaa_gt_routes:
    %{
    gt_prefix: "11"; # Préfixe de Titre Global à correspondre
    peer_id: 1; # Paire de destination
    priority: 1; # Priorité (plus élevée = plus haut)
    description: "Numéros UK"; # Description pour la journalisation
    }
    # Route les numéros US (préfixe 1) vers la paire 1
    %{
    gt_prefix: "1"; # Correspondance de préfixe la plus longue gagne
    peer_id: 2; # Paire de destination
    priority: 1; # Priorité (plus élevée = plus haut)
    description: "Numéros US"; # Description pour la journalisation
    }
    # Route plus spécifique : numéros mobiles UK commençant par 447
    %{
    gt_prefix: "447"; # Correspondance de préfixe la plus longue gagne
    peer_id: 3; # Paire de destination
    priority: 1; # Priorité (plus élevée = plus haut)
    description: "Numéros mobiles UK"; # Description pour la journalisation
    }
    # Route spécifique à SSN (optionnel)
    %{
    gt_prefix: "555"; # Correspondance de préfixe la plus longue gagne
    source_ssn: 8; # Correspond uniquement si SSN source = 8 (SMSC)
    peer_id: 4; # Paire de destination
    priority: 1; # Priorité (plus élevée = plus haut)
    description: "Traffic SMS pour préfixe 61"; # Description pour la journalisation
    }

```

Logique de Routage GT

L'algorithme de routage GT suit ce processus décisionnel :

- Correspondance du Préfixe le Plus Long : Le STP trouve toutes les routes GT où le préfixe correspond au début du Titre Global
 - Exemple : GT 447712345678 correspond à "447" et "447" gagne (correspondance la plus longue)
- Correspondance SSN (Optionnel) :
 - Si source_ssn est spécifié, la route ne correspond que lorsque la SSN de la Partie Appelée SCCP est égale à cette valeur
 - Si source_ssn est null, la route correspond si n'importe quel SSN (vidéocall)

3. Correspondance TT/NPUNAI (Optionnel) :

- Si source_tt, source_npi ou source_nai sont spécifiées, les routes doivent correspondre à ces indicateurs
- Les routes doivent correspondre aux indicateurs des wildcards (correspondent à l'importe quelle valeur)

4. Sélection Basée sur la Spécificité :

- Les routes avec des critères de correspondance plus spécifiques gagnent sur les wildcards
- Ordre de priorité : Longueur de Préfixe GT = SSN = TT = NPI = NAI = Numéro de Priorité

5. Sélection des Indicateurs (Optionnel) :

- Si dest_tt, dest_npi ou dest_nai sont spécifiées, le STP sélecte ces indicateurs
- Indicateur de Plan de Numérotation (NPI) : Spécifie le format d'adresse (par exemple : International, National, Absent)

6. Retour des Codes de Point :

- Si aucun route GT ne correspond, le STP revient au routage par Code de Point en utilisant le DPC

Routage Avancé GT / Type de Traduction, NPI / NAI

En plus de la correspondance de préfixe GT et de SSN, le STP prend en charge le routage et la transformation basés sur les indicateurs de Titre Global SCCP :

Type de Traduction (TT) :

Identifie le plan de numérotation et le type d'adresse

Indicateur de Plan de Numérotation (NPI) :

Définit le plan de numérotation (par exemple : SSIN, Données, Télex)

Indicateur de Nature d'Adresse (NAI) :

Spécifie le format d'adresse (par exemple : International, National, Absent)

OmniSS7 Stock STP

Correspondance (Indicateurs Source) :

Les routes peuvent correspondre sur les indicateurs des messages entrants :

- source_tt : Correspondre aux messages avec un Type de Traduction spécifique
- source_npi : Correspondre aux messages avec un Indicateur de Plan de Numérotation spécifique
- source_nai : Correspondre aux messages avec un Indicateur de Nature d'Adresse spécifique
- Value null = wildcard (correspond à l'importe quelle valeur)

Transformation (Indicateurs de Destination) :

Les routes peuvent réécrire les indicateurs lors du transfert :

- dest_tt : Transforme le Type de Traduction en nouvelle valeur
- dest_npi : Transformer l'Indicateur de Plan de Numérotation en nouvelle valeur
- dest_nai : Transformer l'Indicateur de Nature d'Adresse en nouvelle valeur
- Value null = préserver la valeur d'origine (pas de transformation)

Sélection Basée sur la Spécificité :

Lorsque plusieurs routes correspondent, la route la plus spécifique est sélectionnée en utilisant cet ordre de priorité :

- Correspondance de préfixe GT le plus long
- SSN spécifique sur TT wildcard
- TT source spécifique sur TT wildcard
- NPI source spécifique sur NPI wildcard
- NAI source spécifique sur NAI wildcard
- Nombre de caractères plus bas

Exemples de Configuration :

```
config.conf
enable_gt_routing true;
nhss_gt_routes:
    {
        # Exemple 1 : Correspondre NPI spécifique et transformer NAI
        {
            gt_prefix: "944",
            peer_id: 1,
            source_npi: "44", # Correspondre NPI=4 (ISDN/Telephone)
            source_nai: "1", # Correspondre NAI=1 (International)
            dest_nai: "3", # Transformer en TT=3 (National)
            priority: 1,
            description: "Numéros UK : transformation TT 8-3"
        },
        # Exemple 2 : Correspondre NPI spécifique et transformer NAI
        {
            gt_prefix: "91",
            peer_id: 2,
            source_npi: "44", # Correspondre NPI=4 (ISDN/Telephone)
            source_nai: "4", # Correspondre NAI=4 (International)
            dest_nai: "1", # Transformer en NAI=1 (National)
            priority: 1,
            description: "Numéros US : International-National NAI"
        },
        # Exemple 3 : Routage combiné SSN et Indicateur
        {
            gt_prefix: "93",
            source_tt: "944", # Correspondre au trafic SMS
            source_tt_0: "0", # Correspondre à l'importe quel TT (wildcard)
            source_nai: "6", # Correspondre à l'importe quel NAI (wildcard)
            dest_tt: 2, # Transformer en TT=2 (National)
            dest_nai: "1", # Transformer en NAI=1 (National)
            dest_nai_4: "4", # Définir NAI=4 (International)
            priority: 1,
            description: "SMS Français : Normalisation complète"
        },
        # Exemple 4 : Wildcard TT, NPI spécifique
        {
            gt_prefix: "944",
            source_tt: "944", # Correspondre à l'importe quel TT (wildcard)
            source_nai: "6", # Correspondre à l'importe quel NAI (wildcard)
            dest_tt: 0, # Transformer en NPI=6 (Données)
            dest_nai: "1", # Transformer en NAI=1 (National)
            peer_id: 4,
            priority: 1,
            description: "Normalisation du réseau de données allemand"
        }
    }
}

values.conf
values of TT/NPUNAI

```

Types de Traduction (TT) :

- 0 = Incognito
- 1 = International
- 2 = National
- 3 = Numéro de Réseau

Indicateur de Plan de Numérotation (NPI) :

- 0 = Incognito
- 1 = ISDN/Telephone (E164)
- 2 = Données (K.121)
- 4 = Télex (E.60)
- 5 = Numéro de Réseau (E.212)

Indicateur de Nature d'Adresse (NAI) :

- 0 = Incognito
- 1 = Numéro d'abonné
- 2 = Numéro pour Utilisation Nationale
- 3 = Numéro Supplémentaire National
- 4 = Numéro d'abonné

Exemple de Décision de Routage :

Pour un message entrant avec :

GT : +447123456789

SSN : 8

TT : 1

NPI : 1

NAI : 1

Avant ces routes configurées :

Route A : Wildcard TT

{gt_prefix: "944", peer_id: 1, priority: 1}

Route B : TT spécifique

{gt_prefix: "91", source_tt: 0, peer_id: 2, priority: 1}

Route C : TT spécifique + NPI

{gt_prefix: "944", source_tt: 0, source_npi: 1, peer_id: 3, priority: 1}

Résultat : La Route C est sélectionnée (la plus spécifique : correspond à GT + TT + NPI)

Le message est transféré avec des indicateurs transférés selon les valeurs dest_tt, dest_npi, dest_nai de la Route.

Exemple de Routage GT

GT : +91 SSN Source TT/NPUNAI Route Correspondante

4412345678906 91 Correspondance de préfixe le plus long

4412345678906 91 - - "944" = pair 1

4412345678906 91 - - "944" = pair 1

5558812345678906 91 Correspondance GT + SSN, décrir SSN à 6

5558812345678906 91 - - "555" (SSN 8) = pair 2

4412345678906 91 Correspondance GT + TT, décrir TT à 3

4412345678906 91 - - "44" (TT=0) = pair 1

12125501238 91 Correspondance GT + TT + NPI + NAI

Cas d'Utilisation Courants pour le Routage TT/NPUNAI

1. Normalisation de l'interconnexion Réseau

- Divers réseaux peuvent utiliser des conventions d'indicateurs différents

- Transfert de l'indicateur de plan de numérotation pour assurer la compatibilité

- Exemple : Le réseau partenaire utilise TT=0 pour international, votre réseau utilise TT=1

- 2. Conversion de Protocole
 - Convertir entre les types de numérotation lors du routage entre différents types de réseaux
 - Exemple : Traduire l'adresse mobile (NPV=4) vers le PTIN (NPV=1)
- 3. Standardisation du Format d'Adresse
 - Normaliser tout le trafic entrant pour utiliser des valeurs NAI cohérentes
 - Exemple : Convertir tous les formats internationaux (NAI=4) en format national (NAI=3) pour le routage domestique
- 4. Routage Spécifique au Fournisseur
 - Basé en fonction du type de traduction vers différents fournisseurs de services
 - Exemple : Routage vers le fournisseur A, T1+ vers le fournisseur B
- 5. Intégration de Systèmes Hérités
 - Les systèmes modernes peuvent utiliser des valeurs d'indicateur différentes de celles des systèmes hérités
 - Transformer en STP pour maintenir la compatibilité avec les anciens systèmes

Fonctionnalités de Gestion des Routes

Désactivation des Routes

Les routes peuvent être temporairement désactivées sans être supprimées. Cela est utile pour les tests, la maintenance ou la gestion du trafic.

Drappeur Actif

Les routes par Code de Point et par Titre Global prennent en charge un drappeur enabled optionnel :

```
config {omissis};
  nsha_gt_routes: [
    # Route active
    {
      dest_pc: 100,
      peer_id: 1,
      priority: 1,
      network_indicator: "international",
      enabled: true # Route active (par défaut si omis)
    },
    # Route désactivée (non évaluée lors du routage)
    {
      dest_pc: 200,
      peer_id: 2,
      priority: 1,
      network_indicator: "international",
      enabled: false # Route désactivée
    }
  ];
nsha_gt_routes: [
  # Route GT désactivée
  {
    peer_id: "44",
    dest_pc: 100,
    priority: 1,
    description: "Numéros UK - temporairement désactivés",
    enabled: false # Numéros UK - temporairement désactivés
  }
];
```

Comportement par Défaut :

- Si aucun id n'est spécifié, les routes sont par défaut enabled : True
- Les routes désactivées sont complètement ignorées lors de la recherche de routes
- Utiliser l'interface Web pour activer/désactiver les routes sans modifier la configuration

Cas d'Utilisation :

- Utiliser les numérotations de flux de trafic
- Permettre les manipulations temporaires
- Utiliser ARI de différents chemins de routage
- Définir des règles pour empêcher les routes

Routes DROP - Prévention des Boucles de Routage

Les routes DROP (avec peer_id=0) rejettent silencieusement le trafic au lieu de le transférer. Cela empêche les boucles de routage et permet un filtrage avancé du trafic.

Configuration des Routes DROP

```
config {omissis};
  nsha_gt_routes: [
    # Route DROP pour un code de point spécifique
    {
      dest_pc: 999,
      peer_id: 0, # peer_id=0 signifie DROP
      priority: 1,
      network_indicator: "international",
      enabled: true
    }
  ];
nsha_gt_routes: [
  # Route DROP pour préfixe GT
  {
    gt_prefix: "999",
    peer_id: 0, # peer_id=0 signifie DROP
    priority: 99,
    description: "Bloquer la plage de test"
  }
];
```

Comment Fonctionnent les Routes DROP

Lorsqu'un message correspond à une route DROP :

1. Le routeur de routage identifie peer_id = 0
2. Le message est **silencieusement rejetté** (non transféré)
3. Les messages de routage peuvent être correspondants pour DPC 999 ou "Route DROP correspondante pour GT 999"
4. La recherche de routage retourne {error, dropped}

Important : Le trafic rejeté est enregistré au niveau INFO pour la surveillance et le dépannage.

Cas d'Utilisation Courant : Liste Blanche de Préfixes

L'une des utilisations les plus puissantes des routes DROP est la liste blanche de préfixes : permettant uniquement des numéros spécifiques au sein d'une grande plage tout en bloquant tous les autres.

Le Modèle :

1. Créez une route DROP pour l'ensemble du préfixe avec un numéro de priorité élevé (par exemple, 99)
2. Créez une route pour une plage de numéros spécifiques avec une priorité basse (par exemple, 1)
3. Étant donné que les numéros de priorité plus bas sont évalués en premier, les routes autorisées correspondent avant la route DROP
4. Le trafic est explicitement autorisé et attiré par la route DROP

Scénario Exemple :

Vous avez un préfixe GT 1234 qui représente une plage de 10 000 numéros (1234000000 - 1234999999), mais vous ne souhaitez router que 3 numéros spécifiques : 1234567890, 1234555999, et 1234111222.

```
config {omissis};
nsha_gt_routes: [
  # Route DROP pour priorité HAUTE (évaluée en dernier)
  {
    dest_pc: 999,
    peer_id: 0, # peer_id=0 signifie DROP
    priority: 99, # peer_id élevé = faible priorité + évalué en dernier
    network_indicator: "international",
    enabled: true
  }
];
nsha_gt_routes: [
  # Route DROP pour préfixe GT
  {
    gt_prefix: "999",
    peer_id: 0, # peer_id=0 signifie DROP
    priority: 99,
    description: "Bloquer la plage de test"
  }
];
```

Comportement de Routage :

GT Entrant	Routes Correspondantes	Route Sélectionnée	Action
1234567890	+1234 DHCIP (priorité 99)	+1234567890 (la plus spécifique, la plus haute priorité) Routé vers le pair 1	
1234555999	+1234 DHCIP (priorité 99) +1234555999 (la plus spécifique, la plus haute priorité) Routé vers le pair 1		
1234111222	+1234 DHCIP (priorité 99) +1234111222 (la plus spécifique, la plus haute priorité) Routé vers le pair 1		
1234000000 - 1234999999	+1234 DHCIP (priorité 99) +1234 DHCIP (seule correspondance)		Rejeté + enregistré

Résultat :

- Seuls 3 numéros spécifiques sont routés vers le pair 1
- Tous les autres numéros sont rejetés
- Tout le trafic rejeté est enregistré pour la surveillance

Journaux Généraux :

[INFO] Route DROP correspondante pour GT 1234999999

[INFO] Route DROP correspondante pour GT 1234000000

Routes DROP pour les Codes de Point

Le même modèle de liste blanche fonctionne pour le routage par Code de Point :

```
config {omissis};
nsha_routes: [
  # DHCIP toute la plage /8 (64 codes de point : 1000-1063)
  {
    dest_pc: 1000,
    peer_id: 0,
    priority: 99,
    mask: 8,
    network_indicator: "international",
    enabled: true
  }
];
# Autoriser des PCs spécifiques
{
  dest_pc: 1010, peer_id: 1, priority: 1, network_indicator: "international",
  dest_pc: 1020, peer_id: 1, priority: 1, network_indicator: "international",
  dest_pc: 1030, peer_id: 1, priority: 1, network_indicator: "international"
};
```

Résultat : Seules les PCs 1010, 1020 et 1030 sont routées. Tous les autres PCs dans la plage 1000-1063 sont rejettés.

Surveillance des Routes DROP

Verifiez les Journaux :

Suivez le trafic rejeté
tail -f log/app.log | grep "Route DROP correspondante"

Sortie Attente :

[INFO] Route DROP correspondante pour GT 1234999999

[INFO] Route DROP correspondante pour DPC 1059

Via l'interface Web :

- Navigation à l'onglet Journaux du Système
- Filtrer par niveau INFO
- Rechercher "Route DROP correspondante"

Meilleures Pratiques :

1. Assurez-vous régulièrement les journaux pour vous assurer que les routes DROP ne bloquent pas le trafic légitime
2. Utilisez des champs description décris pour documenter pourquoi les routes sont rejetées
3. Utilisez la fonctionnalité de recherche pour trouver rapidement les routes
4. Testez le comportement des routes DROP avant de déployer en production
5. Configurez des alertes pour des augmentations inattendues du trafic rejeté

Routage Avancé : Routage Basé sur SSN et Réécriture

Numéros de Sous-Système (SSN)

Les Numéros de Sous-Système identifient la couche d'application :

- SSN 4 : HLR (Répertoire de Localisation de l'Abonné)
- SSN 7 : VLR (Répertoire de Localisation Visiteur)
- SSN 8 : SMSC (Centre de Messagerie)
- SSN 9 : GMLC (Centre de Localisation Mobile Passerelle)

Exemple de Routage Basé sur SSN

Routage du trafic SMS vers différents HLR en fonction du préfixe de numéro :

```
config {omissis};
nsha_gt_routes: [
  # Routage vers les numéros UK vers HLR UK, réécriture SSN de 8 (SMS) à 6 (HLR)
  {
    dest_pc: "44",
    source_ssn: 8, # Correspondre au SSN entrant 8 (SMS)
    peer_id: 1, priority: 1, network_indicator: "international",
    dest_ssn: 6, # Réécritre à SSN 6 (HLR)
    priority: 1,
    description: "SMS UK vers HLR"
  }
];
```

```
}.
# Route du trafic vocal pour les numéros UK (SSN 6) sans réécriture
#_gt_peer = M3UA_Peer()
gt_prefix: "44",
local_ssn: 6, # Correspondre au SSN entrant 6 (HLR)
peer_id: 1,
peer_ip: "192.168.1.11",
priority: 1,
description: "Trafic vocal UK"
}
```

Tester la Configuration de Routage STP

Après avoir configuré les paires et les routes, vérifier votre configuration :

1. Vérifier l'état des Pairs

Via l'interface Web :

- Naviguez vers <http://localhost>

- Vérifiez la page d'état M3UA

- Assurez-vous que les paires affichent l'état : ACTIF

Via la Console (x) :

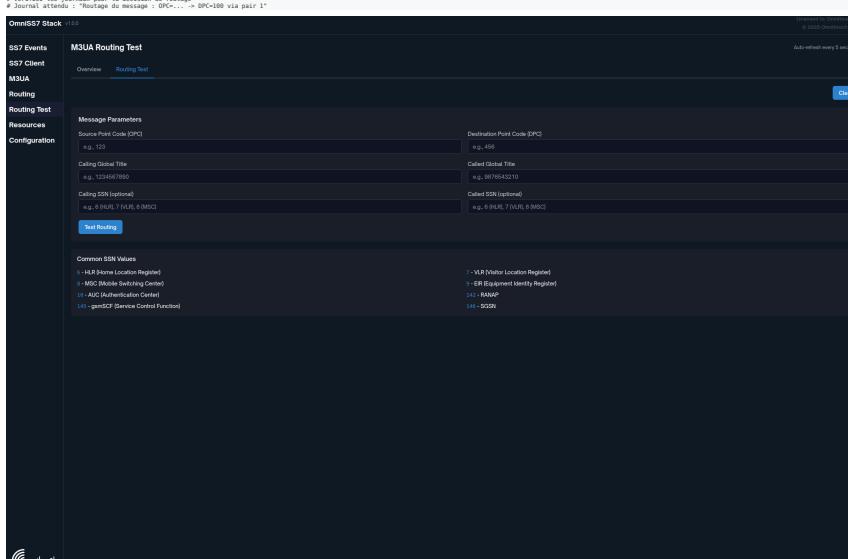
```
# Obtenir tous les états des paires
M3UA_STP.get_peers_status()

# Sortie attendue :
# [
#   {
#     peer_id: 1, name: "Partner_STP_Next", status: :active, point_code: 100, ...},
#   {
#     peer_id: 2, name: "Local_HLR", status: :active, point_code: 200, ...}
# ]
```

2. Tester le Routage par Code de Point

```
# Envoyer un message M3UA de test au MPC 100
test_message = M3UA_Peer()
test_message.send("44", "192.168.1.11", "200", "100", "100", 0)
M3UA_STP.route_by_pc(100, test_message, 0)

# Vérifiez les journaux pour la décision de routage
# Journal d'erreur : "Message de message : MPC<--> MPC-100 via pair 1"
```



3. Tester le Routage par Titre Global

```
# Rechercher manuellement la route GT
# M3UA_STP.lookup_peer_by_gt(44712345678)

# Sortie attendue :
# [
#   {
#     peer: {id: 3, "name": "Mobile_Peer", ...}, nil
#   }
# ]
```

```
# Rechercher la route GT avec SSN
# M3UA_STP.lookup_peer_by_gt(ssn: "555888234567", 8)

# Sortie attendue :
# [
#   {
#     peer: {id: 4, "name": "HLR_Peer", ...}, 6
#   }
# ]
```

4. Surveiller les Métriques de Routage

```
Accédez aux métriques Prometheus à /metrics

Métriques clés :
# Messages reçus par pair
#_m3ua_stp_message_recv_total{peer_name="Partner_STP_Next",point_code="100"} 1523
# Messages envoyés par pair
#_m3ua_stp_message_send_total{peer_name="Local_HLR",point_code="200"} 1468
```

```
# Effectifs de routage
#_m3ua_stp_routing_successes_total{reason="no_route"} 5
#_m3ua_stp_routing_failures_total{reason="no_gt_route"} 2
```

Métriques et Surveillance STP

Métriques Disponibles

Métriques de Trafic par pair :

- `m3ua_stp_message_recv_total{peer_name="Partner_STP_Next",point_code="100"}` Total des messages reçus de chaque paire
- `m3ua_stp_message_send_total{peer_name="Local_HLR",point_code="200"}` Total des messages transférés à chaque paire

**Métriques d'effacement



Guide de l'interface utilisateur Web

[← Retour à la documentation principale](#)

Ce guide fournit une documentation complète pour l'utilisation de l'OmniSS7 **Web UI** (interface Phoenix LiveView).

Table des matières

1. [Aperçu](#)
 2. [Accéder à l'interface Web](#)
 3. [Page de gestion du routage](#)
 4. [Page des abonnés actifs](#)
 5. [Opérations courantes](#)
 6. [Comportement d'auto-rafraîchissement](#)
-

Aperçu

L'interface Web OmniSS7 est une application **Phoenix LiveView** qui fournit des capacités de surveillance et de gestion en temps réel. Les pages disponibles dépendent du mode opérationnel actif (STP, HLR ou SMSc).

Architecture de l'interface Web

Configuration du serveur

- **Protocole:** HTTPS
- **Port:** 443 (configuré dans config/runtime.exs)
- **IP par défaut:** 0.0.0.0 (écoute sur toutes les interfaces)
- **Certificats:** Situés dans priv/cert/

URL d'accès: [https://\[server-ip\]:443](https://[server-ip]:443)

Accéder à l'interface Web

Prérequis

1. **Certificats SSL:** Assurez-vous que des certificats SSL valides sont présents dans priv/cert/:

- `omnitouch.crt` - Fichier de certificat
- `omnitouch.pem` - Fichier de clé privée

2. **Application en cours d'exécution:** Démarrez l'application avec `ieexec -S mix`

3. **Pare-feu:** Assurez-vous que le port 443 est ouvert pour le trafic HTTPS

Pages disponibles par mode

Page	Mode STP	Mode HLR	Mode SMSc	Description
Événements SS7	◊	◊	◊	Journalisation des événements et capture des messages SCCP
Client SS7	◊	◊	◊	Tests manuels des opérations MAP
M3UA	◊	◊	◊	État de la connexion M3UA
Routage	◊	◊	◊	Gestion de la table de routage M3UA
Test de routage	◊	◊	◊	Tests et validation de routes
Liens HLR	◊	◊	◊	État de l'API HLR et gestion des abonnés
Abonnés actifs	◊	◊	◊	Suivi de la localisation des abonnés en temps réel (HLR)
Liens SMSc	◊	◊	◊	État de l'API SMSc et gestion des files d'attente
Abonnés SMSc	◊	◊	◊	Suivi des abonnés en temps réel (SMSc)
Application	◊	◊	◊	Ressources système et surveillance
Configuration	◊	◊	◊	Visualiseur de configuration

Gestion du routage

Page: /routing

Modes: STP, SMSc

Auto-rafrâchissement: Toutes les 5 secondes

La page de gestion du routage fournit une interface à onglets pour gérer les tables de routage M3UA.

Mise en page de la page

Onglet Pairs

Gérez les connexions de pairs M3UA (autres STP, HLR, MSC, SMSC).

Colonnes du tableau des pairs

Colonne	Description	Exemple
ID	Identifiant unique du pair	1
Nom	Nom du pair lisible par l'homme "STP_West"	
Rôle	Rôle de la connexion	client, server, stp
Code de point	Code de point SS7 du pair	100
Distant	IP:Port distant	10.0.0.10:2905
Statut	Statut de la connexion	active, aspup, down
Actions	Boutons Modifier/Supprimer	-

Ajouter un pair

1. **Cliquez** sur l'onglet Pairs
2. **Remplissez** les champs du formulaire :
 - **ID du pair**: Généré automatiquement s'il est laissé vide
 - **Nom du pair**: Nom descriptif (obligatoire)
 - **Rôle**: Sélectionnez client, server ou stp
 - **Code de point**: Code de point SS7 (obligatoire)
 - **IP locale**: Adresse IP de votre système
 - **Port local**: 0 pour une attribution de port dynamique
 - **IP distante**: Adresse IP du pair
 - **Port distant**: Port du pair (généralement 2905)
 - **Contexte de routage**: ID de contexte de routage M3UA
 - **Indicateur de réseau**: international ou national
3. **Cliquez** sur "Ajouter un pair"

Persistance: Le pair est immédiatement enregistré dans Mnesia et survit au redémarrage.

Modifier un pair

1. **Cliquez** sur le bouton "Modifier" de la ligne du pair
2. **Modifiez** les champs du formulaire si nécessaire
3. **Cliquez** sur "Mettre à jour le pair"

Remarque: Si vous changez l'ID du pair, l'ancien pair est supprimé et un nouveau est créé.

Supprimer un pair

1. **Cliquez** sur le bouton "Supprimer" de la ligne du pair
2. **Confirmez** la suppression (toutes les routes utilisant ce pair seront également supprimées)

Indicateurs de statut des pairs

Statut Couleur	Description
active ◊ Vert	Le pair est connecté et routant des messages
aspup ◊ Jaune	L'ASP est en ligne mais pas encore actif
down ◊ Rouge	Le pair est déconnecté

Onglet Routes de Code de Point

Configurez des règles de routage basées sur les codes de point de destination.

Colonnes du tableau des routes

Colonne	Description	Exemple
PC de destination	Code de point cible (format zone.area.id)	1.2.3 (100)
Masque	Masque de sous-réseau pour la correspondance de PC	/14 (exact), /8 (plage)
ID du pair	Pair cible pour cette route	1
Nom du pair	Nom du pair cible	"STP_West"
Priorité	Priorité de la route (1 = la plus élevée)	1
Réseau	Indicateur de réseau	international
Actions	Boutons Modifier/Supprimer	-

Ajouter une route de code de point

1. **Cliquez** sur l'onglet "Routes de Code de Point"
2. **Remplissez** les champs du formulaire :
 - **Code de point de destination**: Entrez au format zone.area.id (par exemple, 1.2.3) ou entier (0-16383)
 - **Masque**: Sélectionnez le masque /14 pour une correspondance exacte, des valeurs inférieures pour des plages
 - **ID du pair**: Sélectionnez le pair cible dans le menu déroulant
 - **Priorité**: Entrez la priorité (1 = la plus élevée, par défaut)
 - **Indicateur de réseau**: Sélectionnez international ou national
3. **Cliquez** sur "Ajouter une route"

Format de code de point: Vous pouvez entrer des codes de point dans deux formats :

- **Format 3-8-3**: zone.area.id (par exemple, 1.2.3)
- **Format entier**: 0-16383 (par exemple, 1100)

Le système convertit automatiquement entre les formats.

Comprendre les masques

Les codes de point sont des valeurs de 14 bits (0-16383). Le masque spécifie combien de bits les plus significatifs doivent correspondre :

Masque	PC correspondants	Cas d'utilisation
/14	1 (correspondance exacte)	Route vers une destination spécifique
/13	2 PC	Petite plage
/8	64 PC	Plage moyenne
/0	Tous les 16 384 PC	Route par défaut/de secours

Exemples :

- PC 1000 /14 → Correspond uniquement au PC 1000
- PC 1000 /8 → Correspond aux PC 1000-1063 (64 PC consécutifs)
- PC 0 /0 → Correspond à tous les codes de point (route par défaut)

Carte de référence du masque de code de point

La page Web comprend une référence interactive montrant toutes les valeurs de masque et leurs plages.

Onglet Routes de Titre Global

Configurez des règles de routage basées sur les adresses de Titre Global SCCP.

Exigence: Le routage par Titre Global doit être activé dans la configuration :

```
config :omniss7,  
  enable_gt_routing: true
```

Colonnes du tableau des routes

Colonne	Description	Exemple
Préfixe GT	Préfixe GT du parti appelé (vide = secours)	"1234", ""
SSN source	Correspondre au SSN du parti appelé (optionnel)	6 (HLR), any
ID du pair	Pair cible	1
Pair	Nom du pair	"HLR_West (1)"
SSN de destination	Réécrire le SSN lors du transfert (optionnel)	6, preserve
Priorité	Priorité de la route	1
Description	Description de la route	"US numbers"
Actions	Boutons Modifier/Supprimer	-

Ajouter une route de titre global

1. **Cliquez** sur l'onglet "Routes de Titre Global"
2. **Remplissez** les champs du formulaire :
 - **Préfixe GT**: Laissez vide pour la route de secours, ou entrez des chiffres (par exemple, "1234")
 - **SSN source**: Optionnel - filtrer par SSN du parti appelé
 - **ID du pair**: Sélectionnez le pair cible
 - **SSN de destination**: Optionnel - réécrire le SSN lors du transfert
 - **Priorité**: Priorité de la route (1 = la plus élevée)
 - **Description**: Description lisible par l'homme
3. **Cliquez** sur "Ajouter une route"

Routes de secours: Si le préfixe GT est vide, la route agit comme un attrape-tout pour les GT qui ne correspondent à aucune autre route.

Valeurs SSN courantes

La page comprend une carte de référence avec des valeurs SSN courantes :

SSN	Élément de réseau
6	HLR (Home Location Register)
7	VLR (Visitor Location Register)
8	MSC (Mobile Switching Center)
9	EIR (Equipment Identity Register)
10	AUC (Authentication Center)
142	RANAP
145	gsmSCF (Service Control Function)
146	SGSN

Réécriture de SSN

- **SSN source**: Correspondre au SSN du parti appelé dans les messages entrants
- **SSN de destination**: Si défini, réécrit le SSN du parti appelé lors du transfert
 - Vide = préserver le SSN original
 - Valeur = remplacer par ce SSN

Cas d'utilisation: Router des messages avec SSN=6 (HLR) vers un pair, et réécrire vers SSN=7 (VLR) du côté sortant.

Persistante de la table de routage

Toutes les routes sont stockées dans Mnesia et survivent aux

redémarrages de l'application.

Comment les routes persistent

1. **Modifications de l'interface Web:** Toutes les opérations d'ajout/ modification/suppression sont immédiatement enregistrées dans Mnesia
2. **Redémarrage de l'application:** Les routes sont chargées depuis Mnesia au démarrage
3. **Fusion de runtime.exs:** Les routes statiques de config/runtime.exs sont fusionnées avec les routes Mnesia (pas de doublons)

Priorité de la route

Lorsque plusieurs routes correspondent à une destination :

1. **Plus spécifique d'abord:** Les valeurs de masque plus élevées (plus spécifiques) prennent la priorité
 2. **Champ de priorité:** Les numéros de priorité inférieurs routent en premier (1 = priorité la plus élevée)
 3. **Statut du pair:** Seules les routes vers des pairs actifs sont utilisées
-

Abonnés actifs

Page: /subscribers

Mode: HLR uniquement

Auto-rafraîchissement: Toutes les 2 secondes

Affiche le suivi en temps réel des abonnés qui ont envoyé des demandes UpdateLocation.

Fonctionnalités de la page

Colonnes du tableau des abonnés

Colonne	Description	Exemple
IMSI	IMSI de l'abonné	"50557123456789"
Numéro VLR	Adresse GT VLR actuelle	"555123155"
Numéro MSC	Adresse GT MSC actuelle	"555123155"
Mis à jour à	Dernière horodatage UpdateLocation	"2025-10-25 14:23:45 UTC"
Durée	Temps depuis l'enregistrement	"2h 15m 34s"

Résumé des statistiques

Lorsque des abonnés sont présents, une carte de résumé affiche :

- **Total Actif:** Nombre total d'abonnés enregistrés
- **VLRs Uniques:** Nombre d'adresses VLR distinctes
- **MSCs Uniques:** Nombre d'adresses MSC distinctes

Effacer les abonnés

Bouton Effacer Tout: Supprime tous les abonnés actifs du tracker.

Confirmation: Nécessite une confirmation avant de vider (ne peut pas être annulé).

Cas d'utilisation: Effacer les enregistrements d'abonnés obsolètes après une maintenance ou un test du réseau.

Auto-rafraîchissement

La page se rafraîchit automatiquement toutes les **2 secondes** pour afficher les mises à jour des abonnés en temps réel.

Abonnés SMSc

Page: /smsc_subscribers

Mode: SMSc uniquement

Auto-rafraîchissement: Toutes les 2 secondes

Affiche le suivi en temps réel des abonnés basé sur les messages alertServiceCenter reçus des HLR, l'état de livraison des messages et le suivi des échecs.

Fonctionnalités de la page

Colonnes du tableau des abonnés

Colonne	Description	Exemple
MSISDN	Numéro de téléphone de l'abonné	"15551234567"
IMSI	IMSI de l'abonné	"001010123456789"
HLR GT	HLR GT qui a envoyé alertServiceCenter	"15551111111"
Msgs Envoyés	Nombre de messages MT-FSM envoyés	5
Msgs Reçus	Nombre de messages MO-FSM	2

Colonne	Description	Exemple
Statut	reçus	● Actif
Dernière mise à jour	Actif ou Échoué (codé par couleur)	"2025-10-30 14:23:45 UTC"
Durée	Dernière horodatage de mise à jour	"15m 34s"

Indicateurs de statut

- **Actif** (Vert): L'abonné est joignable, dernier alertServiceCenter reçu avec succès
- **Échoué** (Rouge): Dernière tentative de livraison échouée (SRI-for-SM ou erreur d'abonné absent)

Résumé des statistiques

Lorsque des abonnés sont présents, une carte de résumé affiche :

- Total Suivi**: Nombre total d'abonnés suivis
- Actifs**: Nombre d'abonnés avec un statut actif
- Échoués**: Nombre d'abonnés avec un statut échoué
- HLRs Uniques**: Nombre de HLR distincts envoyant des alertes

Gestion des abonnés

Bouton Supprimer: Supprime un abonné individuel du suivi.

Bouton Effacer Tout: Supprime tous les abonnés suivis.

Confirmation: Effacer Tout nécessite une confirmation avant de vider (ne peut pas être annulé).

Cas d'utilisation :

- Supprimer les entrées obsolètes après des problèmes de réseau
- Effacer les données de test après le développement
- Surveiller quels HLR envoient des alertes

Compteurs de messages

Le tracker incrémente automatiquement les compteurs :

- Messages Envoyés**: Incrémenté lorsque SRI-for-SM réussit et MT-FSM est envoyé
- Messages Reçus**: Incrémenté lorsque MO-FSM est reçu de l'abonné

Auto-rafraîchissement

La page se rafraîchit automatiquement toutes les **2 secondes** pour afficher les mises à jour en temps réel des abonnés et des statuts.

Opérations courantes

Recherche et filtrage

Actuellement, l'interface Web ne comprend pas de fonctionnalité de recherche/filtrage intégrée. Pour trouver des routes spécifiques :

1. Utilisez la fonction de recherche de votre navigateur (Ctrl+F / Cmd+F)
2. Recherchez des noms de pairs, des codes de point ou des préfixes GT

Opérations en masse

Pour effectuer des modifications de routes en masse :

1. **Option 1:** Utilisez l'[API REST](#) pour un accès programmatique
2. **Option 2:** Modifiez config/runtime.exs et redémarrez l'application
3. **Option 3:** Utilisez l'interface Web pour des modifications de routes individuelles

Exportation/Importation

Remarque: L'interface Web ne prend actuellement pas en charge l'exportation ou l'importation de tables de routage. Les routes sont :

- Stockées dans des fichiers de base de données Mnesia
- Configurées dans config/runtime.exs

Pour sauvegarder les routes :

1. **Mnesia:** Sauvegardez le répertoire Mnesia.{node_name}/
 2. **Configuration:** Contrôle de version de config/runtime.exs
-

Comportement d'auto-rafraîchissement

Différentes pages ont différents intervalles de rafraîchissement :

Page	Intervalle de rafraîchissement	Raison
Gestion du	5 secondes	Les changements de route sont peu

Page	Intervalle de rafraîchissement	Raison
routage		fréquents
Abonnés actifs	2 secondes	L'état des abonnés change fréquemment
État M3UA	Varie selon la page	Surveillance de l'état de connexion

Connexion WebSocket: Toutes les pages utilisent les connexions WebSocket de Phoenix LiveView pour des mises à jour en temps réel.

Interruption réseau: Si la connexion WebSocket est perdue, la page tentera de se reconnecter automatiquement.

Dépannage

Page ne se charge pas

1. **Vérifiez le certificat HTTPS:** Assurez-vous que `priv/cert/omnitouch.crt` et `.pem` sont présents
2. **Vérifiez le port 443:** Vérifiez que les règles de pare-feu autorisent le trafic HTTPS
3. **Application en cours d'exécution:** Confirmez que l'application est en cours d'exécution avec `iex -S mix`
4. **Console du navigateur:** Vérifiez les erreurs de certificat SSL (avertissements de certificat auto-signé)

Routes ne persistant pas

1. **Vérifiez le stockage Mnesia:** Vérifiez `mnesia_storage_type:disc_copies` dans la configuration
2. **Répertoire Mnesia:** Assurez-vous que le répertoire Mnesia est accessible en écriture
3. **Vérifiez les journaux:** Recherchez des erreurs Mnesia dans les journaux de l'application

Auto-raffraîchissement ne fonctionnant pas

1. **Connexion WebSocket:** Vérifiez la console du navigateur pour des erreurs WebSocket
 2. **Réseau:** Vérifiez la connexion réseau stable
 3. **Rafraîchissement de la page:** Essayez de rafraîchir la page (F5)
-

Documentation connexe

- [Guide STP](#) - Configuration détaillée du routage
 - [Guide HLR](#) - Gestion des abonnés
 - [Guide API](#) - API REST pour un accès programmatique
 - [Référence de configuration](#) - Tous les paramètres de configuration
-

Résumé

L'interface Web OmniSS7 fournit une gestion intuitive et en temps réel des tables de routage et du suivi des abonnés :

- ◊ **Mises à jour en temps réel** - L'auto-raffraîchissement maintient les données à jour
- ◊ **Stockage persistant** - Mnesia garantit que les routes survivent aux redémarrages
- ◊ **UI basée sur les rôles** - Les pages s'adaptent au mode opérationnel (STP/HLR/SMSc)
- ◊ **Gestion interactive** - Ajouter, modifier, supprimer des routes sans redémarrage
- ◊ **Surveillance de l'état** - État de connexion et statut des pairs en direct

Pour des opérations avancées ou de l'automatisation, consultez le [Guide API](#).