

# Référence de configuration d'Omnismf

La configuration modifiable par l'opérateur est fournie via `config/runtime.exs`, qui est largement influencée par les variables d'environnement du système d'exploitation. La plupart des valeurs ci-dessous ont une valeur par défaut intégrée et une variable d'environnement qui la remplace avant le démarrage du SMF. Quelques clés sont définies uniquement par la configuration de l'application et n'ont pas de variable d'environnement (`chf_uri`, le bloc `charging`, `pgw_c_addr`, et le bloc `pco`); celles-ci sont signalées là où elles apparaissent. La clé d'environnement de l'application compilée est `:omnismf`.

```
config :omnismf,
  sbi_scheme: "http",           # SBI_SCHEME
  sbi_addr:   "127.0.0.4",      # SBI_ADDR
  sbi_port:   7777,             # SBI_PORT
  nrf_uri:    "http://127.0.0.1:7777", # NRF_URI
  udm_uri:    "http://127.0.0.12:7777", # UDM_URI
  pcf_uri:    "http://127.0.0.13:7777", # PCF_URI
  amf_uri:    "http://127.0.0.5:7777", # AMF_URI
  upf_addr:   "10.179.1.24",     # UPF_ADDR
  upf_pfcp_port: 8805,          # UPF_PFCP_PORT
  pfcp_addr:  "0.0.0.0",        # PFCP_ADDR
  mcc: "999",                  # MCC
  mnc: "70",                   # MNC
  dns: ["8.8.8.8", "8.8.4.4"],  # DNS
  dns6: ["2001:4860:4860::8888"], # DNS6
  mtu: 1400,                   # MTU
  prometheus_metrics_port: 9568, # PROMETHEUS_PORT
  heartbeat_interval: 10_000,    # HEARTBEAT_INTERVAL
  ue_ip_pool: [subnet_map: %{default: ["10.45.0.0/16"]} ] #
UE_IP_SUBNET_MAP / UE_IP_POOL
```

# Paramètres SBI, Pair et PLMN

| Paramètre  | Type   | Requis | Par défaut               |    |
|------------|--------|--------|--------------------------|----|
| sbi_scheme | chaîne | Non    | "http"                   | SB |
| sbi_addr   | chaîne | Non    | "127.0.0.4"              | SB |
| sbi_port   | entier | Non    | 7777                     | SB |
| nrf_uri    | chaîne | Non    | "http://127.0.0.1:7777"  | NR |
| udm_uri    | chaîne | Non    | "http://127.0.0.12:7777" | UD |

| Paramètre          | Type           | Requis | Par défaut               |    |
|--------------------|----------------|--------|--------------------------|----|
| pcf_uri            | chaîne         | Non    | "http://127.0.0.13:7777" | PC |
| amf_uri            | chaîne         | Non    | "http://127.0.0.5:7777"  | AM |
| upf_addr           | chaîne         | Non    | "10.179.1.24"            | UP |
| upf_pfcf_port      | entier         | Non    | 8805                     | UP |
| pfcf_addr          | chaîne         | Non    | "0.0.0.0"                | PF |
| mcc                | chaîne         | Non    | "999"                    | MC |
| mnc                | chaîne         | Non    | "70"                     | MN |
| heartbeat_interval | entier<br>(ms) | Non    | 10000                    | HE |

| Paramètre | Type | Requis | Par défaut |  |
|-----------|------|--------|------------|--|
|           |      |        |            |  |

# Interconnexion N26

| Paramètre  | Type   | Requis | Par défaut   | Var Env                                              | Descr                                                                                                                                                                                                                                                                                                                       |
|------------|--------|--------|--------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pgw_c_addr | chaîne | Non    | "127.0.0.11" | Aucun<br>(configuration de l'application uniquement) | L'adresse IP du PGW-C du PGW autonome jumelé, renvoyée à la vue E1 de l'interconnexion N26 afin de la création d'une session entre l'MME/SCS et l'aterrissement PGW-C qui se fait au point d'attache UPF que l'opérateur a définie dans l'environnement d'application : omnism n'a pas de variable d'environnement pour cet |

# Facturation convergente N40 (Nchf, TS 32.290 / TS 32.291)

Le SMF peut agir en tant que consommateur de facturation convergente vers le CHF via N40 (Nchf\_ConvergedCharging). Lorsqu'il est activé, il ouvre une demande de données de facturation à chaque session PDU établie, la met à jour pendant la session à partir des rapports d'utilisation URR N4, et la ferme lors de la libération. Le CHF est d'abord résolu via la découverte NRF, revenant à la `chf_uri` configurée lorsque la découverte ne résout pas une. La facturation convergente est désactivée par défaut ; lorsqu'elle est désactivée, le chemin de facturation est une opération no-op et aucun appel CHF n'est effectué.

La clé `chf_uri` de niveau supérieur et le bloc `charging` sont définis uniquement par la configuration de l'application `:omnismf` ; aucun d'eux n'a de variable d'environnement.

```
config :omnismf, :charging,  
  enabled: false,  
  chf_uri: "http://127.0.0.20:7777"
```

| Paramètre                         | Type    | Requis | Par défaut                                                  | Var Env                                                 |
|-----------------------------------|---------|--------|-------------------------------------------------------------|---------------------------------------------------------|
| <code>chf_uri</code>              | chaîne  | Non    | <code>"http://127.0.0.20:7777"</code>                       | Aucun<br>(configurat<br>de<br>l'applicatio<br>uniquemer |
| <code>charging<br/>enabled</code> | booléen | Non    | <code>false</code>                                          | Aucun<br>(configurat<br>de<br>l'applicatio<br>uniquemer |
| <code>charging<br/>chf_uri</code> | chaîne  | Non    | (retourne à la <code>chf_uri</code> de<br>niveau supérieur) | Aucun<br>(configurat<br>de                              |

| Paramètre | Type | Requis | Par défaut | Var Env                   |
|-----------|------|--------|------------|---------------------------|
|           |      |        |            | l'applicatio<br>uniquemer |

## Session-AMBR vers le PCF

Lors de la création de l'association de politique SM, le Session-AMBR souscrit (lu à partir du champ de configuration DNN UDM `sessionAmbr`) est envoyé au PCF sous **les deux** noms de champ spécifiés `subsSessAmbr` (TS 29.512 `SmPolicyContextData`) et la clé héritée `sessionAmbr`. L'envoi uniquement de `sessionAmbr` laisse `subsSessAmbr` non défini, ce qui amène le PCF à revenir à son défaut d'environ ~1000 Kbps et à limiter chaque session. Aucune configuration d'opérateur ne régit cela ; c'est documenté ici afin que les opérateurs comprennent quel champ le PCF doit honorer.

## Valeurs par défaut PCO orientées UE (clés plates)

Ces clés plates remplissent les Options de Configuration de Protocole renvoyées à l'UE dans l'Acceptation de l'Établissement de Session PDU. Elles sont remplacées par le bloc structuré `:pco` ci-dessous lorsqu'il est présent.

| Paramètre | Type             | Requis | Par défaut                 | Var Env | De                                              |
|-----------|------------------|--------|----------------------------|---------|-------------------------------------------------|
| dns       | liste de chaînes | Non    | [ "8.8.8.8", "8.8.4.4" ]   | DNS     | Se<br>DN<br>an<br>l'U<br>Sé<br>de:<br>da<br>en' |
| dns6      | liste de chaînes | Non    | [ "2001:4860:4860::8888" ] | DNS6    | Se<br>DN<br>an<br>l'U<br>Sé<br>de:<br>da<br>en' |
| mtu       | entier           | Non    | 1400                       | MTU     | MT<br>IPv<br>an<br>l'U                          |

# Pool IP UE

| Paramètre  | Type                    | Requis | Par défaut                                    | Var Env                                                   |
|------------|-------------------------|--------|-----------------------------------------------|-----------------------------------------------------------|
| ue_ip_pool | mot-clé<br>(subnet_map) | Non    | <code>%{default:<br/>["10.45.0.0/16"]}</code> | <code>UE_IP_SUBNET</code><br>/<br><code>UE_IP_POOL</code> |

Le JSON `UE_IP_SUBNET_MAP` est indexé par le nom exact du DNN, le motif `^regex`, ou `"default"` ; chaque valeur est une liste de CIDR IPv4 et/ou IPv6. La priorité de correspondance est **DNN exact** → **regex** → **default**. Exemple :

```
{
  "internet":    ["10.45.0.0/16", "2001:db8:45::/64"],
  "ims":        ["10.47.0.0/24"],
  "^enterprise": ["10.99.0.0/16"],
  "default":    ["10.45.0.0/16"]
}
```

Si ni `UE_IP_SUBNET_MAP` ni `UE_IP_POOL` n'est défini, le pool par défaut est `10.45.0.0/16`. Voir [Attribution d'adresse IP UE](#) pour le comportement d'attribution et de réservation d'IP statique.

# P-CSCF et Options de Configuration de Protocole (:pco)

Pour VoNR/IMS, le SMF renvoie l'adresse P-CSCF (et les autres éléments PCO : DNS, NBNS, MTU de lien) à l'UE à l'intérieur de l'Acceptation de l'Établissement de Session PDU ePCO. Ce bloc utilise **le même agencement et la même logique de sélection que le PGW-C 4G**, de sorte qu'un opérateur configure les deux nœuds de manière identique. Lorsqu'ils sont présents, les valeurs :pco prennent la priorité sur les clés plates dns/dns6/mtu ci-dessus. Ce bloc est défini uniquement dans la configuration de l'application :omnismf et n'a pas de variable d'environnement.

```
config :omnismf, :pco,  
  p_cscf_ipv4_address_list: ["10.179.4.165"],  
  p_cscf_ipv6_address_list: [],  
  p_cscf_discovery_enabled: true,  
  p_cscf_discovery_fqdn:  
    "pcscf.ims.mnc070.mcc999.3gppnetwork.org",  
  p_cscf_discovery_dns_server: "10.179.2.10",  
  p_cscf_discovery_timeout_ms: 5000,  
  primary_dns_server_address: "8.8.8.8",  
  secondary_dns_server_address: "8.8.4.4",  
  primary_nbns_server_address: nil,  
  secondary_nbns_server_address: nil,  
  ipv4_link_mtu_size: 1400
```

| Paramètre                             | Type             | Requis | Par               |
|---------------------------------------|------------------|--------|-------------------|
| <code>p_cscf_ipv4_address_list</code> | liste de chaînes | Non    | <code>[]</code>   |
| <code>p_cscf_ipv6_address_list</code> | liste de chaînes | Non    | <code>[]</code>   |
| <code>p_cscf_discovery_enabled</code> | booléen          | Non    | <code>true</code> |

| Paramètre                                | Type   | Requis | Par                            |
|------------------------------------------|--------|--------|--------------------------------|
|                                          |        |        |                                |
| <code>p_cscf_discovery_fqdn</code>       | chaîne | Non    | <code>pcscf.mnc{MNC}.mc</code> |
| <code>p_cscf_discovery_dns_server</code> | chaîne | Non    | résolveur système              |

| Paramètre                     | Type            | Requis | Par        |
|-------------------------------|-----------------|--------|------------|
| p_cscf_discovery_timeout_ms   | entier<br>(ms)  | Non    | 5000       |
| primary_dns_server_address    | chaîne          | Non    | (plat dns) |
| secondary_dns_server_address  | chaîne          | Non    | (plat dns) |
| primary_nbns_server_address   | chaîne  <br>nil | Non    | nil        |
| secondary_nbns_server_address | chaîne  <br>nil | Non    | nil        |
| ipv4_link_mtu_size            | entier          | Non    | (plat mtu) |

# Clés de secours plates P-CSCF héritées

Les clés plates `pcscf` et `pcscf6` sont des secours hérités conservés pour la parité avec les anciennes configurations PGW-C. Elles ne sont consultées que lorsque les listes d'adresses P-CSCF `:pco` structurées (`p_cscf_ipv4_address_list` / `p_cscf_ipv6_address_list`) ne sont pas définies. La forme structurée `:pco` est préférée pour les nouveaux déploiements. Chaque clé accepte une seule chaîne d'adresse ou une liste de chaînes d'adresse. Ces clés sont définies uniquement par la configuration de l'application `:omnismf` et n'ont pas de variable d'environnement.

| Paramètre           | Type                       | Requis | Par défaut      | Var Env                                           | Descripti                                                                                                                          |
|---------------------|----------------------------|--------|-----------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <code>pcscf</code>  | chaîne ou liste de chaînes | Non    | <code>[]</code> | Aucun (configuration de l'application uniquement) | Secours statique IPv4 hérité, utilisé uniquement lorsque <code>p_cscf_ipv4_address_list</code> est non défini. La forme structurée |
| <code>pcscf6</code> | chaîne ou liste de chaînes | Non    | <code>[]</code> | Aucun (configuration de l'application uniquement) | Secours statique IPv6 hérité, utilisé uniquement lorsque <code>p_cscf_ipv6_address_list</code> est non défini. La forme structurée |

# Ports de métriques et de gestion

| Paramètre                                       | Type   | Requis | Par défaut        | Var Env                       |
|-------------------------------------------------|--------|--------|-------------------|-------------------------------|
| <code>prometheus_metrics_port</code>            | entier | Non    | <code>9568</code> | <code>PROMETHEUS_PORT</code>  |
| port API de gestion<br>( <code>:api_ex</code> ) | entier | Non    | <code>8443</code> | <code>API_PORT</code>         |
| Port HTTPS du panneau de contrôle               | entier | Non    | <code>7443</code> | <code>CONTROL_PANEL_PO</code> |

L'API de gestion et le panneau de contrôle sont servis via TLS en utilisant le certificat à `priv/cert/omnitouch.crt` et la clé à `priv/cert/omnitouch.pem`. Seuls les ports d'écoute peuvent être ajustés à l'exécution via les variables d'environnement ci-dessus.

# API de gestion / OAM

L'API de gestion (TLS, port par défaut 8443, env API\_PORT) expose les points de terminaison suivants pour les opérations et les diagnostics. Tous les points de terminaison de gestion sont montés sous le préfixe /api.

| Méthode | Chemin                | But                                                                |
|---------|-----------------------|--------------------------------------------------------------------|
| GET     | /api/status/api       | Statut du service API.                                             |
| GET     | /api/status/nrf       | Statut de l'enregistrement NRF.                                    |
| GET     | /api/status/nf        | Résumé du statut NF (SMF).                                         |
| GET     | /api/pdu_session      | Liste des sessions PDU actives.                                    |
| POST    | /api/pdu_session      | Créer une session PDU (tests opérationnels).                       |
| GET     | /api/pdu_session/{id} | Afficher une seule session PDU.                                    |
| DELETE  | /api/pdu_session/{id} | Supprimer une seule session PDU.                                   |
| GET     | /api/upf              | Vue de l'association UPF / PCF.                                    |
| POST    | /api/eps_handover     | Gérer l'interconnexion N26 EPS (adopter / changer / rechercher).   |
| GET     | /api/ip_pool          | Capacité IPv4 globale plus un résumé des pools par DNN.            |
| GET     | /api/ip_pool/detail   | Pools par DNN avec chaque plage CIDR, totaux et adresses allouées. |

| Méthode   | Chemin                     | But                                                         |
|-----------|----------------------------|-------------------------------------------------------------|
| GET       | /api/statistics            | Statistiques de session et opérationnelles.                 |
| GET       | /api/oam/config            | Lire la configuration d'exécution.                          |
| PATCH/PUT | /api/oam/config/{id}       | Mettre à jour une valeur de configuration d'exécution.      |
| GET       | /api/oam/resources         | Instantané de l'utilisation des ressources.                 |
| GET       | /api/oam/diagnostics       | Instantané des diagnostics.                                 |
| GET       | /api/oam/peers             | Statut des pairs SBI (UDM/PCF/AMF/NRF).                     |
| GET       | /api/oam/pfcp_peers        | Statut des pairs PFCP (UPF).                                |
| GET       | /api/oam/pfcp_session/{id} | Inspecter une session PFCP.                                 |
| POST      | /api/oam/upf               | Gérer l'association UPF.                                    |
| POST      | /api/oam/pfcp/reassociate  | Relancer la configuration de l'association PFCP avec l'UPF. |
| POST      | /api/oam/cache/clear       | Effacer les caches internes.                                |

| Méthode | Chemin                                       | But                                                                 |
|---------|----------------------------------------------|---------------------------------------------------------------------|
| POST    | <code>/api/oam/log_level</code>              | Changer le niveau de journalisation à l'exécution sans redémarrage. |
| POST    | <code>/api/oam/nrf/reregister</code>         | Forcer la réinscription NRF.                                        |
| DELETE  | <code>/api/oam/session/{id}</code>           | Vider une seule session.                                            |
| DELETE  | <code>/api/oam/session/by_dnn/{id}</code>    | Vider les sessions par DNN.                                         |
| DELETE  | <code>/api/oam/session/by_supi/{id}</code>   | Vider les sessions par SUPI.                                        |
| DELETE  | <code>/api/oam/session/by_snssai/{id}</code> | Vider les sessions par S-NSSAI.                                     |
| DELETE  | <code>/api/oam/session/by_upf/{id}</code>    | Vider les sessions par UPF.                                         |
| POST    | <code>/api/oam/session/modify</code>         | Modifier une session (tests opérationnels).                         |

## Journalisation

Le SMF émet des journaux JSON structurés. Le formatteur de journal par défaut est configuré dans `runtime.exs` et n'est pas destiné à être modifié par déploiement :

```
config :logger, :default_formatter,  
  format: {OmniLogger.JsonFormatter, :format},  
  metadata: :all
```

| Paramètre             | Type        | Requis | Par défaut                            | Descriptif                                                                                                                                             |
|-----------------------|-------------|--------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>format</code>   | formatteur  | Non    | <code>OmniLogger.JsonFormatter</code> | Émet chaque ligne de journalisation tant qu'un formatteur JSON personnalisé n'est pas ingéstré dans le pipeline de journalisation.                     |
| <code>metadata</code> | atome/liste | Non    | <code>:all</code>                     | Inclut toutes les métadonnées du Logger, l'ID de session, le PDU, la référence au contexte DNN, la procédure de journalisation dans chaque ligne JSON. |

Le niveau de journalisation à l'exécution peut être changé sans redémarrage via le point de terminaison de l'API de gestion `POST /api/oam/log_level`.  
Filtrer les journaux par SUPI pour tracer le cycle de vie complet d'un abonné.

[← Vue d'ensemble](#)

# Métriques OmniSMF

Les métriques sont exposées à `GET /metrics` sur `prometheus_metrics_port` (par défaut `9568`, env `PROMETHEUS_PORT`). Tous les noms de métriques ci-dessous sont préfixés comme indiqué par l'exportateur.

# Métriques de session PDU

| Métrique                                         | Type     | Étiquettes                             | Description                                                       |
|--------------------------------------------------|----------|----------------------------------------|-------------------------------------------------------------------|
| <code>omni_smf_pdu_session_create_count</code>   | compteur | <code>result</code>                    | Opérations de création de session PDU.                            |
| <code>omni_smf_pdu_session_modify_count</code>   | compteur | <code>result</code>                    | Opérations de modification de session PDU.                        |
| <code>omni_smf_pdu_session_release_count</code>  | compteur | <code>result</code>                    | Opérations de libération de session PDU.                          |
| <code>omni_smf_pdu_session_creates_total</code>  | compteur | <code>result</code> , <code>dnn</code> | Total des opérations de création de session PDU par réseau DNN.   |
| <code>omni_smf_pdu_session_releases_total</code> | compteur | <code>result</code>                    | Total des opérations de libération de session PDU par réseau DNN. |
| <code>omni_smf_active_pdu_sessions_count</code>  | jauge    | <code>dnn</code>                       | Nombre de sessions PDU actives par réseau DNN.                    |

| Métrique                          | Type         | Étiquettes | Description                                                                                                                    |
|-----------------------------------|--------------|------------|--------------------------------------------------------------------------------------------------------------------------------|
|                                   |              |            | le ma<br>conte<br>(la so<br>vérité<br>sessio<br>active                                                                         |
| omni_smf_session_duration_seconds | distribution | aucun      | Durée<br>sessio<br>en se<br>émise<br>la libé<br>mesu<br>depuis<br>l'étab<br>de la<br>(seau<br>15, 30<br>300, 9<br>3600,<br>s). |

Exemple (taux de succès d'établissement) :

```
sum(rate(omni_smf_pdu_session_creates_total{result="success"}
[5m]))
/ sum(rate(omni_smf_pdu_session_creates_total[5m]))
```

Exemple (sessions actives par DNN) :

```
sum by (dnn) (omni_smf_active_pdu_sessions_count)
```

# Métriques PFCP / UPF

| Métrique                                     | Type     | Étiquettes             |
|----------------------------------------------|----------|------------------------|
| <code>omni_smf_pfcpsessions_total</code>     | compteur | <code>operation</code> |
| <code>omni_smf_pfcassociations_total</code>  | compteur | <code>result</code>    |
| <code>omni_smf_pfcheartbeats_total</code>    | compteur | <code>result</code>    |
| <code>omni_smf_upf_health</code>             | jauge    | aucun                  |
| <code>omni_smf_n4_usage_reports_total</code> | compteur | aucun                  |

| Métrique                                    | Type     | Étiquettes |
|---------------------------------------------|----------|------------|
| omni_smf_n4_usage_report_volume_bytes_total | compteur | aucun      |

Les deux métriques `omni_smf_n4_*` sont les seules métriques de volume du plan utilisateur du SMF : elles se remplissent à partir des rapports d'utilisation URR que l'UPF renvoie sur N4, donc elles n'avancent que lorsque l'UPF a des règles de rapport d'utilisation installées et fournit des rapports.

Exemple (alerte lorsque l'association UPF tombe) :

```
omni_smf_upf_health == 0
```

Exemple (débit de volume du plan utilisateur à travers toutes les sessions) :

```
sum(rate(omni_smf_n4_usage_report_volume_bytes_total[5m]))
```

Exemple (taux de rapports d'utilisation reçus de l'UPF) :

```
sum(rate(omni_smf_n4_usage_reports_total[5m]))
```

Exemple (échecs de battements de cœur PFCP vers l'UPF) :

```
sum(rate(omni_smf_pfcpc_heartbeats_total{result="failure"}[5m]))
```

# Métriques de facturation N40

| Métrique                                 | Type     | Étiquettes             | Description                                                                                                  |
|------------------------------------------|----------|------------------------|--------------------------------------------------------------------------------------------------------------|
| <code>omni_smf_n40_charging_total</code> | compteur | <code>operation</code> | Total des opérations de facturation convergée N40 Nchf vers le CHF par événement de facturation (TS 32.290). |

Exemple (taux d'opérations de facturation convergée par événement) :

```
sum by (operation) (rate(omni_smf_n40_charging_total[5m]))
```

# Métriques NRF

| Métrique                                      | Type  | Étiquettes           | Description                                                     |
|-----------------------------------------------|-------|----------------------|-----------------------------------------------------------------|
| <code>omni_smf_nrf_registration_status</code> | jauge | <code>nf_type</code> | État de l'enregistreur NRF (1 = enregistré, 0 = non enregistré) |

Exemple :

```
omni_smf_nrf_registration_status == 0
```

# Métriques VM BEAM

| Métrique                                | Type  | Description                                    |
|-----------------------------------------|-------|------------------------------------------------|
| <code>beam_memory_total</code>          | jauge | Mémoire totale BEAM (octets).                  |
| <code>beam_memory_processes</code>      | jauge | Mémoire allouée aux processus Erlang.          |
| <code>beam_memory_processes_used</code> | jauge | Mémoire réellement utilisée par les processus. |
| <code>beam_memory_system</code>         | jauge | Mémoire système.                               |
| <code>beam_memory_atom</code>           | jauge | Mémoire totale des atomes.                     |
| <code>beam_memory_atom_used</code>      | jauge | Mémoire des atomes utilisée.                   |
| <code>beam_memory_binary</code>         | jauge | Mémoire binaire.                               |
| <code>beam_memory_code</code>           | jauge | Mémoire de code.                               |
| <code>beam_memory_ets</code>            | jauge | Mémoire des tables ETS.                        |
| <code>beam_processes_count</code>       | jauge | Nombre de processus Erlang.                    |
| <code>beam_ports_count</code>           | jauge | Nombre de ports Erlang.                        |
| <code>beam_atom_count</code>            | jauge | Nombre d'atomes.                               |
| <code>beam_vm_uptime</code>             | jauge | Temps de fonctionnement de la VM (secondes).   |

Le nombre de processus actifs sert également de signal à l'échelle de la session, puisque chaque session PDU est un processus de travail dédié. Une divergence soutenue entre `beam_processes_count` et

`omni_smf_active_pdu_sessions_count` peut indiquer des travailleurs de session orphelins (voir [Dépannage](#)).

# Guide des opérations OmniSMF

OmniSMF est la fonction de gestion de session (SMF) du cœur 5G. Il gère l'ensemble du cycle de vie des sessions PDU : allocation d'adresses, association de politiques, programmation et démantèlement du plan utilisateur. Ce guide couvre le rôle 3GPP, les interfaces SBI et N4 qu'il expose, et les procédures clés qu'il pilote, chacune avec un diagramme de séquence.

# Rôle 3GPP et références de spécification

| Spécification         | Pertinence                                                                                                                  |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------|
| TS 23.501             | Architecture système : rôle du SMF, concept de session PDU, modèle QoS                                                      |
| TS 23.502             | Procédures : Établissement de session PDU (§4.3.2), Modification (§4.3.3), Libération (§4.3.4), Demande de service (§4.2.3) |
| TS 29.502             | API de service Nsmf_PDUSession                                                                                              |
| TS 29.244             | Interface N4 : protocole PFCP entre SMF et UPF                                                                              |
| TS 29.503             | Nudm_SubscriberDataManagement (sm-data) et Nudm_UEContextManagement (smf-registrations)                                     |
| TS 29.512             | Npcf_SMPolicyControl : association de politique SM et notification vers PCF                                                 |
| TS 29.518             | Namf_Communication : N1N2MessageTransfer vers AMF                                                                           |
| TS 32.290 / TS 32.291 | Nchf_ConvergedCharging (N40) : facturation convergente vers le CHF                                                          |
| TS 24.501             | 5GS NAS (5GSM) : établissement/modification de session PDU, type de session et gestion des causes 5GSM (§6.4.1.3)           |
| TS 29.571             | Types de données communs : <code>IpAddress</code> , S-NSSAI                                                                 |

# Interfaces

| Interface | Pair  | Protocole                     | Objectif                                                                                                                                                                                               |
|-----------|-------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| N11       | AMF   | Nsmf_PDUSession (SBI, HTTP/2) | Créer / récupérer / mettre à jour / libérer le contexte SM, N1N2MessageTransfer                                                                                                                        |
| N7        | PCF   | Npcf_SMPolicyControl (SBI)    | Créer/supprimer l'association politique SM + rappel de notation politique                                                                                                                              |
| N10       | UDM   | Nudm_SDM + Nudm_UECM (SBI)    | Données d'abonnement de subscriber et enregistrement SMF                                                                                                                                               |
| N4        | UPF   | PFCP (UDP)                    | Association PFCP + établissement/modification/termination de session                                                                                                                                   |
| N40       | CHF   | Nchf_ConvergedCharging (SBI)  | Facturation convergente : ouvrir / à jour/fermer une demande de facturation sur le cycle de vie de session PDU (CHF résolu via découverte NRF avec une URI de secours statique ; désactivé par défaut) |
| N26       | PGW-C | Interconnexion interne        | Adopter / changer / rechercher la mobilité 4G/5G                                                                                                                                                       |
| n/a       | NRF   | Nnrf_NFManagement (SBI)       | Enregistrement NF + heartbeat / découverte sortante)                                                                                                                                                   |

# Points de terminaison SBI

Tous les points de terminaison SBI sont servis sous l'URL de base `{SBI_SCHEME}://{SBI_ADDR}:{SBI_PORT}`.

| Méthode | Chemin                                                               | Service         |                                                                                                                                                                          |
|---------|----------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST    | <code>/nsmf-pdusession/v1/sm-contexts</code>                         | Nsmf_PDUSession | Crée SM (code de session)                                                                                                                                                |
| POST    | <code>/nsmf-pdusession/v1/sm-contexts/{smContextRef}/retrieve</code> | Nsmf_PDUSession | Récupère les informations de contexte SM (code de session) spécifiées dans l'URI (GET)                                                                                   |
| POST    | <code>/nsmf-pdusession/v1/sm-contexts/{smContextRef}/modify</code>   | Nsmf_PDUSession | Mettre à jour les informations de contexte SM (code de session) N2, code de session d'établissement de la transmission, modification du flux de données de l'utilisateur |
| POST    | <code>/nsmf-pdusession/v1/sm-contexts/{smContextRef}/release</code>  | Nsmf_PDUSession | Libère les ressources SM (code de session) (démarrage de la session)                                                                                                     |
| POST    | <code>/nsmf-callback/sm-policy-notify/{smContextRef}</code>          | Rappel Npcf     | Reçoit les informations de mise à jour de la politique de l'utilisateur (SmPolicy) du PNF dédié                                                                          |

## Créer un contexte SM

Champs de requête obligatoires : `supi`, `sNssai`, `servingNetwork`, `dnn`, `smContextStatusUri`, `pduSessionId`, `anType`, `servingNfId`. Retourne `201 Created` avec `Location: {base}/nsmf-pdusession/v1/sm-contexts/{ref}`.  
Mappage des erreurs : `400 MANDATORY_IE_MISSING`, `404 DNN_DENIED`, `500 SYSTEM_FAILURE`, `503 NF_DISCOVERY_FAILURE` (UDM inaccessible).

## Récupérer le contexte SM

L'AMF lit le contexte SM actuel en le référenciant (TS 29.502 §5.2.2.6 utilise `POST .../sm-contexts/{smContextRef}/retrieve`, pas `GET`). Le SMF retourne `200 OK` avec la vue actuelle du contexte SM pour une référence connue, ou `404 CONTEXT_NOT_FOUND` pour une inconnue.

## Mettre à jour le contexte SM

Le gestionnaire de modification dispatch sur la première clé reconnue dans le corps :

| Clé du corps                 | Scénario                                                                                                                                       |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| n2SmInfo +<br>n2SmInfoType   | Informations SM N2 du gNB ( PDU_RES_SETUP_RSP active le tunnel ; PDU_RES_REL_RSP le désactive)                                                 |
| upCnxState:<br>"DEACTIVATED" | Libération AN / entrée au repos UE (FAR descendant vers BUFFER)                                                                                |
| upCnxState:<br>"ACTIVATING"  | Demande de service ; la réponse porte N2<br>PDU_RES_SETUP_REQ                                                                                  |
| hoState                      | Transition d'état de transfert : PREPARING, PREPARED, COMPLETED (réactive le tunnel), CANCELLED                                                |
| qosFlows                     | Ajout/modification de flux QoS après établissement (retourne 204 No Content ; les changements sont transmis au gNB via les informations SM N2) |
| release: true                | Libération initiée par l'AMF (retourne 204 No Content)                                                                                         |
| servingNfId                  | Changement d'instance AMF pendant la mobilité (retourne 204 No Content)                                                                        |

## Libérer le contexte SM

Le corps peut inclure ueLocation pour l'enregistrement de la localisation finale. Retourne 204 No Content.

# Procédures clés

## Établissement de session PDU (TS 23.502 §4.3.2)



Le N1N2MessageTransfer est envoyé de manière asynchrone, donc le `201 Created` retourne d'abord à l'AMF. Le message SM N1 est une acceptation d'établissement de session PDU NAS et les informations SM N2 sont un `PDU Session Resource Setup Request Transfer` NGAP, transporté comme un corps `multipart/related` (TS 29.518) avec les parties NAS/NGAP binaires référencées par `contentId`.

## Sourcing de configuration DNN

Le comportement d'un DNN est défini à deux endroits, et un opérateur doit définir les deux de manière cohérente pour qu'un DNN fonctionne correctement :

| Attribut DNN               | Sourced from                                         | Comment il est provisionné                                                                                                                                   |
|----------------------------|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S-NSSAI autorisé           | Abonnement AMF/UDM                                   | L'AMF fournit <code>sNssai</code> lors de la création du contexte SM ; le DNN doit exister sous un S-NSSAI correspondant dans les données SM de l'abonné UDM |
| Session-AMBR               | UDM<br><code>dnnConfiguration.sessionAmbr</code>     | Provisionné dans le HSS → UDM/UDR                                                                                                                            |
| QoS par défaut (5QI + ARP) | UDM<br><code>dnnConfiguration.5gQosProfile</code>    | Provisionné dans le HSS → UDM/UDR                                                                                                                            |
| Type(s) de session PDU     | UDM<br><code>dnnConfiguration.pduSessionTypes</code> | <code>defaultSessionType</code> + <code>allowedSessionTypes</code>                                                                                           |
| Mode SSC                   | UDM <code>dnnConfiguration.sscModes</code>           | <code>defaultSscMode</code> + <code>allowedSscModes</code> (TS 23.501 §5.6.9.2)                                                                              |

| Attribut DNN              | Sourced from                            | Comment il est provisionné        |
|---------------------------|-----------------------------------------|-----------------------------------|
|                           |                                         |                                   |
| IP statique UE            | UDM<br>dnnConfiguration.staticIpAddress | Provisionné dans le HSS → UDM/UDR |
| Pool IP UE (dynamique)    | <b>SMF</b> ue_ip_pool subnet_map        | Configuration SMF                 |
| DNS / MTU / P-CSCF (ePCO) | <b>SMF</b> dns / dns6 / mtu / :pco      | Configuration SMF                 |

La règle de base : **la politique par abonné et par DNN (S-NSSAI, AMBR, QoS par défaut, type de session, mode SSC, IP statique) vit dans l'abonnement UDM ; les ressources réseau remises à l'UE (pool IP dynamique, DNS, MTU, P-CSCF) vivent dans la configuration SMF.** Un DNN présent dans le subnet\_map SMF mais absent des données SM de l'abonné UDM est rejeté avec 404 DNN\_DENIED ; un DNN présent dans l'UDM mais sans entrée subnet\_map correspondante et sans pool default échoue à l'allocation IP avec :no\_pool\_for\_dnn.

# Allocation d'adresse IP UE

Chaque session PDU se voit attribuer une adresse IP (IPv4, IPv6, ou les deux) avant que la session PCF ne soit établie avec l'UPF. Le pool IP reflète le modèle d'allocation PGW-C 4G. Les sources sont essayées dans l'ordre de priorité :

1. **Adresse adoptée N26.** Pour une session d'origine 4G passant à 5G via N26 (un `adoptedIpv4` dans la demande de création), l'IP UE déjà allouée par le PGW-C autonome est réutilisée telle quelle ; l'adresse appartient au pool du PGW-C, donc elle est préservée lors du transfert, pas réallouée. Pour une session à double pile, la portion IPv6 est alors allouée dynamiquement.
2. **IP statique (pinned par l'abonné).** Lorsque les données d'abonnement SM UDM contiennent une `staticIpAddress` pour le DNN demandé, cette adresse fixe est honorée pour chaque session sur ce DNN. Cela reflète le comportement d'IP statique HSS/PGW-C 4G et est finalement provisionné dans le HSS et affiché via l'UDM/UDR. Pour une session à double pile, la portion IPv6 est allouée dynamiquement.
3. **Pool dynamique.** Sinon, une adresse est allouée à partir du pool configuré du DNN. L'allocation choisit une adresse hôte libre aléatoire (excluant le réseau/broadcast pour IPv4), réessaie en cas de collision, et est dédoublée à travers toutes les sessions actives.

L'IP UE est retournée au pool lors de **tout** arrêt de travailleur (libération explicite, crash, ou arrêt du superviseur), donc un CPE bloqué dans une boucle d'établissement/libération/réessai ne peut pas fuir lentement les adresses du pool.

## Pools sensibles au DNN ( `subnet_map` )

Les pools sont sélectionnés par DNN via le `subnet_map`. Les clés sont des noms de DNN exacts, des motifs `^regex`, ou `:default` ; les valeurs sont des listes de CIDR IPv4 et/ou IPv6. La priorité de correspondance est **DNN exact** → **regex** → **:default**. Pour une session `IPV4V6`, le SMF alloue à partir des plages IPv4 et IPv6 du pool correspondant. Voir [Pool IP UE](#) dans la référence de configuration.

## Champ IP statique et réservation

Le champ `staticIpAddress` suit TS 29.503 (`DnnConfiguration`) et TS 29.571 (`IpAddress`) : un tableau d'objets, chacun portant éventuellement `ipv4Addr` (un seul objet est également accepté). Une adresse statique est réservée dans le pool afin qu'elle ne soit jamais remise dynamiquement :

- Une adresse statique **à l'intérieur** d'une plage configurée est comptée contre ce pool et exclue de l'allocation dynamique.
- Une adresse statique **à l'extérieur** de chaque plage configurée est suivie uniquement pour la détection des doublons et ne consomme pas la capacité du pool (les IP statiques peuvent vivre sur leur propre sous-réseau routé).
- Si l'adresse statique est **déjà utilisée** par une autre session active, la nouvelle session PDU est rejetée (`:static_ip_in_use`) plutôt que d'émettre un doublon, correspondant au PGW-C 4G. Cela ne devrait se produire que si la même IP statique est mal configurée contre deux abonnés.

Pour provisionner une IP statique, définissez `staticIpAddress` dans la configuration DNN de l'abonné (via le HSS → UDM/UDR) ; aucun changement de configuration SMF n'est requis.

### Réconciliation du type de session PDU (TS 24.501 §6.4.1.3)

Lorsqu'une session à double pile (`IPV4V6`) est demandée mais que le pool correspondant ne fournit qu'une seule famille d'adresses, le SMF **réconcilie le type de session signalé vers la famille qu'il a réellement allouée** avant de construire l'Acceptation d'établissement N1 et le

`PDUSessionResourceSetupRequestTransfer` N2 :

- IPv4 alloué, pas d'IPv6 → signaler `IPV4` avec la cause 5GSM **#50 (type de session PDU IPv4 uniquement autorisé)**.
- IPv6 alloué, pas d'IPv4 → signaler `IPV6` avec la cause 5GSM **#51 (type de session PDU IPv6 uniquement autorisé)**.
- Les deux familles présentes → signaler `IPV4V6` inchangé.

Ceci est requis pour l'interopérabilité : annoncer `IPV4V6` tout en transportant uniquement une adresse IPv4 fait que le RAN rejette la configuration des ressources N2. Les opérateurs servant des abonnements à double pile à partir

de pools uniquement IPv4 verront donc ces sessions se lever comme IPv4 avec la cause NAS correspondante, plutôt que d'échouer.

## Surveillance

- `GET /api/ip_pool` : capacité IPv4 globale plus un résumé de pool par DNN.
- `GET /api/ip_pool/detail` : pools par DNN avec chaque plage CIDR, total/alloué/disponible, et la liste complète des adresses actuellement allouées (dynamiques + statiques).

**Remarque IPv6** : Les adresses IPv6 sont allouées, réservées, libérées et rapportées par le pool.

## Découverte et publicité P-CSCF (VoNR / IMS)

Pour une session PDU IMS/VoNR, l'UE doit apprendre un P-CSCF contre lequel s'enregistrer. Le SMF le retourne dans l'**ePCO** de l'Acceptation d'établissement de session PDU (conteneur P-CSCF IPv4 `0x000C`, conteneur IPv6 `0x0001`), un conteneur par adresse, identique au PGW-C 4G. Le P-CSCF n'est retourné que lorsque le DNN commence par `ims`. Ordre de sélection :

1. **Découverte DNS + vérification de santé (préférée).** Lorsque `p_cscf_discovery_enabled`, le SMF résout `_sip._tcp.<fqdn>` (SRV) en A/AAAA, puis vérifie la santé de chaque candidat avec un SIP `OPTIONS` (TCP puis UDP). Seuls les serveurs qui répondent sont annoncés, et l'ensemble est actualisé sur un cycle fixe de 60 secondes, donc un P-CSCF qui cesse de répondre aux `OPTIONS` est retiré de l'ensemble annoncé dans environ une minute, et un récupéré est réajouté au cycle suivant. L'actualisation fonctionne indépendamment de l'établissement de session ; les nouvelles sessions lisent toujours le dernier ensemble sain.
2. **Secours statique.** Si la découverte est désactivée ou ne retourne rien de sain, `p_cscf_ipv4_address_list` / `p_cscf_ipv6_address_list` sont utilisés.

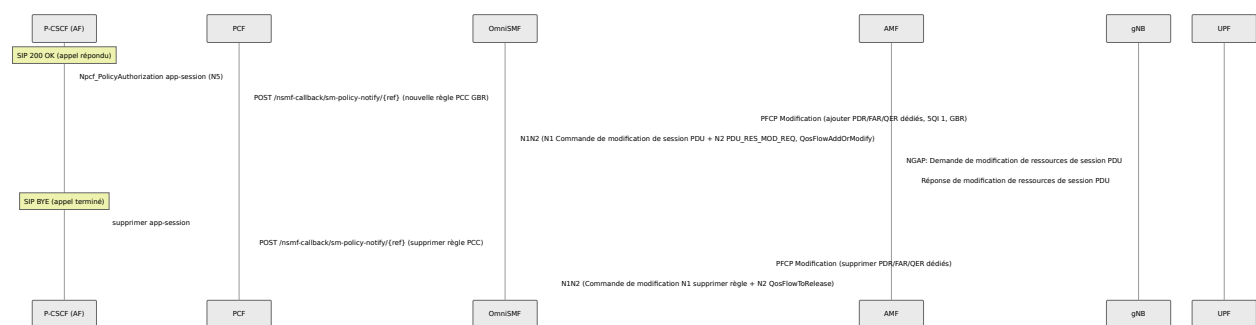
Cela reflète exactement le PGW-C, donc un déploiement IMS 4G/5G mixte configure le P-CSCF de la même manière sur les deux nœuds. Voir [P-CSCF et options de configuration de protocole](#).

# Flux QoS dédiés / Porteur GBR (VoNR, 5QI-1)

VoNR a besoin d'un flux QoS GBR dédié (5QI 1) pour les médias vocaux, aux côtés du flux non-GBR par défaut. Un chemin **statique** (provisionné) et un chemin **dynamique** (piloté par l'appel) sont pris en charge.

**Statique** : la règle GBR est provisionnée dans la politique (la règle de facturation PCRF HSS vers UDR `sm-policy-data` vers la règle PCC PCF) et retournée sur la décision `sm-policies`. Elle est stockée et installée une fois la session PFCP existante, donc l'UPF marque le flux avec son 5QI/GBR dès que la session est établie.

**Dynamique (piloté par l'appel)** : le flux est ajouté lorsque l'appel est répondu et supprimé sur BYE, piloté par le rappel de notification de politique PCF :



La notification est livrée à `/nsmf-callback/sm-policy-notify/{smContextRef}` (le SMF lit `smPolicyDecision` du corps) : une règle PCC non nulle installe le flux, une règle à valeur nulle le supprime (TS 29.512, une règle PCC à valeur nulle est une règle à supprimer). L'appariement de l'application-session AF à la politique SM (par IP UE) est effectué sur le PCF.

## Modification de session PDU : Mise à jour du flux QoS

Un AMF peut également piloter une modification de flux QoS directement via le point de terminaison de modification avec une liste `qosFlows`, plutôt que via le PCF. Chaque entrée met à jour un flux existant (correspondant par `qfi`) ou alloue un nouveau QFI. L'UPF est reprogrammé avec le QoS modifié, et comme `SmContextUpdatedData` ne définit aucun membre JSON `qosFlows`, la réponse

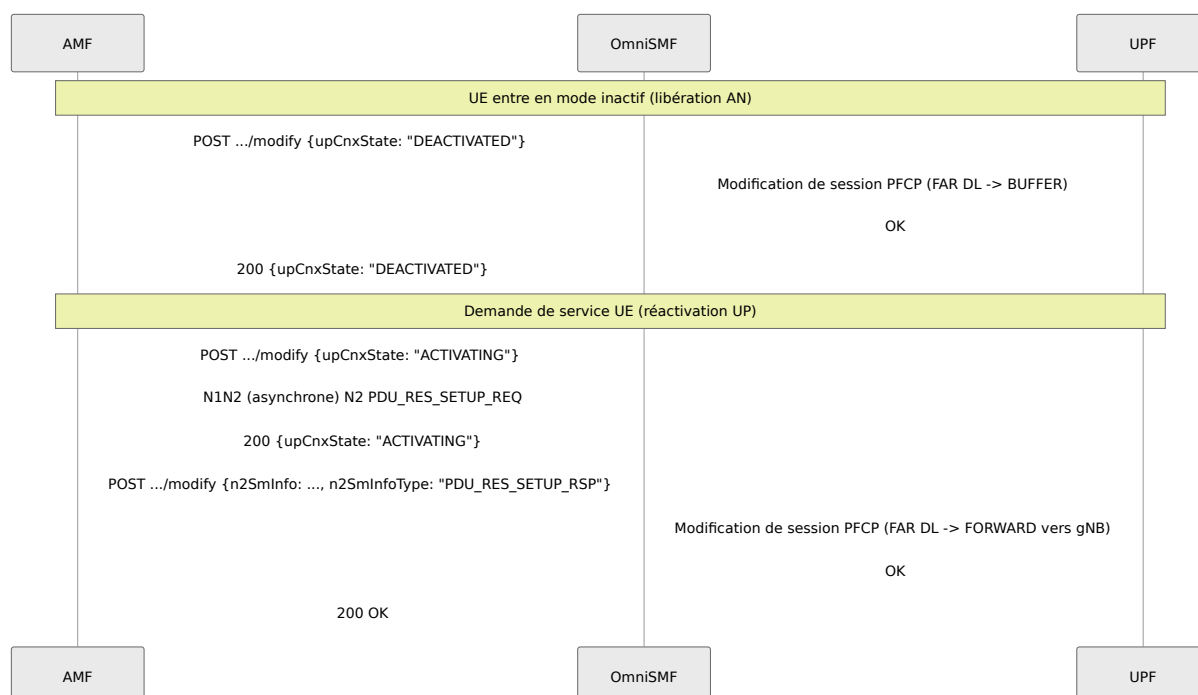
est `204 No Content` ; le changement de QoS est transmis au gNB via les informations SM N2.

## Libération de session PDU (TS 23.502 §4.3.4)



L'ordre de démantèlement est suppression de politique PCF → suppression de session PCF → désenregistrement UDM → notification de statut du contexte SM (`resourceStatus: RELEASED`) au rappel AMF ; le travailleur libère ensuite l'IP UE de retour au pool et s'arrête. Le même démantèlement s'exécute pour une modification `release: true` initiée par l'AMF.

## UE au repos / Demande de service : État de connexion UP (TS 23.502 §4.2.3)



Lors de `ACTIVATING`, le SMF met à jour le PTI à partir de la dernière demande d'établissement de l'UE et envoie un `N1N2MessageTransfer` à l'AMF pour

déclencher le `PDUSessionResourceSetup` sur le gNB ; la réponse `200` porte `upCnxState: ACTIVATING`. Le plan utilisateur est réorienté vers le gNB uniquement une fois que le F-TEID gNB arrive dans le suivi `PDU_RES_SETUP_RSP`.

## Transitions d'état de transfert

Un transfert initié par l'AMF fait progresser la session à travers les valeurs `hoState` sur le point de terminaison de modification :

- `PREPARING` / `PREPARED` : l'état est enregistré ; le plan utilisateur n'est pas modifié.
- `COMPLETED` : le tunnel est réactivé (modification PFCP, FAR descendant vers le gNB cible) et l'état de transfert est effacé.
- `CANCELLED` : l'état de transfert est effacé sans changement de plan utilisateur.

## Structure de session PFCP N4

Chaque session PDU installe les IEs PFCP suivants sur l'UPF (TS 29.244) :

| IE               | Direction    | Objectif                                                                                      |
|------------------|--------------|-----------------------------------------------------------------------------------------------|
| PDR (montant)    | Accès → Cœur | Correspondre au trafic GTP-U du gNB sur le F-TEID N3 de l'UPF                                 |
| PDR (descendant) | Cœur → Accès | Correspondre au trafic de N6 par adresse IP UE                                                |
| FAR (montant)    | Cœur         | Transférer vers N6 (sans en-tête extérieur)                                                   |
| FAR (descendant) | Accès        | Initialement BUFFER ; mis à jour vers GTP-U FORWARD après <code>PDU_RES_SETUP_RSP</code>      |
| QER              | Les deux     | Faire respecter l'AMBR de session (MBR montant et descendant) ; QER dédié ajouté par flux GBR |
| URR              | Les deux     | Rapport d'utilisation basé sur le temps (seuil dérivé de <code>heartbeat_interval</code> )    |

Le seul URR utilise une mesure basée sur la durée avec un déclencheur de seuil temporel. Le seuil est dérivé de `heartbeat_interval` (six fois sa valeur, par exemple 60 s avec le heartbeat par défaut de 10 s), donc il n'est pas indépendamment réglable ; augmentez `heartbeat_interval` pour allonger la cadence de rapport. Les rapports d'utilisation que l'UPF renvoie sont enregistrés et affichés comme les métriques

`omni_smf_n4_usage_reports_total` et `omni_smf_n4_usage_report_volume_bytes_total` (voir [Métriques](#)), et, lorsque la facturation convergente N40 est activée, transférés au CHF en tant que demandes de données de facturation intermédiaires.

L'association PFCP avec l'UPF est établie (avec des heartbeats) avant qu'une demande de session ne soit envoyée. Si l'association est perdue, elle peut être réinitialisée via l'API de gestion (`POST /api/oam/pfcp/reassociate`).

## Interconnexion EPS N26 (4G ↔ 5G)

Le SMF et le PGW-C autonome partagent un UPF et coordonnent via un chemin d'interconnexion interne N26 afin qu'un UE conserve son adresse IP lors d'un transfert :

- **Adopter (4G → 5G) :** une session EPS d'origine 4G est adoptée dans 5GS, réutilisant l'IP UE déjà allouée par le PGW-C. La réadoption d'une session existante retourne la session 5G existante plutôt que de créer un doublon.
- **Changer :** le côté d'accès d'une session existante est changé en `:eps` (5G → 4G) ou `:fivegs` (4G → 5G) en mettant à jour le tunnel d'accès via une modification PFCP (S-GW S5/S8-U TEID/adresse pour EPS, gNB N3 TEID/adresse pour 5GS) tout en conservant la même IP UE et la même session PFCP sur l'UPF. Le changement retourne une vue EPS (IP UE préservée, DNN, AMBR de session, identifiant de porteur EPS par défaut, et le F-TEID PGW-U) que le PGW-C utilise pour répondre au SGW.
- **Recherche :** les sessions PDU actives d'un UE peuvent être lues comme des vues de connexion PDN EPS par SUPI/IMSI.

L'API de gestion expose `POST /api/eps_handover` pour piloter l'interconnexion. L'adresse S5/S8-C du PGW-C apparié est annoncée via la clé de configuration `pgw_c_addr` (voir [Référence de configuration](#)).

# Dépannage OmniSMF

## Établissement de la session PDU échoue avec 404 DNN\_DENIED

- **Symptôme** : l'AMF reçoit `404` avec la cause `DNN_DENIED`.
- **Cause** : les données d'abonnement SM du UDM n'incluent pas le DNN demandé.
- **Résolution** : vérifiez que les données d'abonnement du UDM pour le SUPI contiennent une entrée `dnnConfigurations` correspondant au DNN demandé (la correspondance est insensible à la casse). Le journal SMF montre `dnn_not_found`.

## Établissement de la session PDU échoue avec 503 NF\_DISCOVERY\_FAILURE

- **Symptôme** : l'AMF reçoit `503` avec la cause `NF_DISCOVERY_FAILURE`.
- **Cause** : le SMF n'a pas pu atteindre le UDM à `udm_uri`.
- **Résolution** : vérifiez la connectivité de l'hôte SMF à `udm_uri`, et que le port SBI du UDM est accessible. Le journal SMF montre la raison sous-jacente.

## Établissement de session PFCP rejeté

- **Symptôme** : le UPF renvoie une cause non acceptée dans la réponse d'établissement de session PFCP.
- **Causes / résolution** :

- `upf_addr` / `upf_pfcf_port` incorrect : vérifiez que le UPF écoute sur le port PFCF.
- Association PFCF non établie : le SMF doit compléter la configuration de l'association PFCF avant d'envoyer des demandes de session. Vérifiez les journaux de démarrage pour les messages d'association ; si nécessaire, relancez via `POST /api/oam/pfcf/reassociate`.
- `pfcf_addr` non routable depuis le UPF : définissez-le sur une adresse que le UPF peut atteindre.

## UE ne reçoit pas d'adresse IP

- **Symptôme** : la session échoue lors de l'attribution d'IP.
- **Causes / résolution** :
  - `:no_pool_for_dnn` : le DNN demandé n'a pas d'entrée `subnet_map` correspondante et pas de `default` ; ajoutez-en un.
  - `:ipv4_pool_exhausted` / `:ipv6_pool_exhausted` : la plage correspondante est pleine ; agrandissez ou ajoutez une plage.
  - `:static_ip_in_use` : la même `staticIpAddress` est provisionnée pour deux abonnés ; corrigez les données d'abonnement.
  - `:static_ip_invalid` : l'`staticIpAddress.ipv4Addr` provisionné n'est pas une adresse IPv4 valide ; corrigez les données d'abonnement.
- **Double pile servie comme une seule pile** : une demande de double pile d'un pool IPv4 uniquement apparaît intentionnellement comme IPv4 avec la cause 5GSM #50 (voir [réconciliation du type de session PDU](#)).

## Session AMBR limitée à ~1 Mbps

- **Symptôme** : le débit de chaque session est limité à environ 1 Mbps, quel que soit l'AMBR souscrit.
- **Cause** : le PCF a reçu uniquement une clé `sessionAmbr` non spécifiée et a laissé `subsSessAmbr` non défini, donc il est retombé sur son défaut d'environ 1000 Kbps.
- **Résolution** : le SMF envoie à la fois `subsSessAmbr` (le nom de champ TS 29.512) et `sessionAmbr` lors de l'association de politique SM, et lit l'AMBR

souscrit à partir du champ `sessionAmbr` de la configuration DNN du UDM. Vérifiez que la configuration DNN du UDM porte le bon `sessionAmbr`, et que le PCF respecte `subsSessAmbr`. Voir [Session-AMBR vers le PCF](#).

## N1N2MessageTransfer vers l'AMF échoue

- **Symptôme** : le contexte SM et la session PFCP sont créés mais l'UE ne reçoit jamais l'Acceptation d'Établissement.
- **Cause** : le N1N2MessageTransfer asynchrone vers l'AMF a échoué ; pas de nouvelle tentative automatique.
- **Résolution** : vérifiez `amf_uri` et la connectivité de l'AMF. Pour récupérer, libérez le contexte SM et réessayez l'établissement.

## Flux VoNR dédié non créé

- **Symptôme** : un appel vocal s'établit mais aucun flux GBR 5QI-1 dédié n'apparaît.
- **Causes / résolution** :
  - Le rappel de notification de politique du PCF n'est pas arrivé : confirmez que le PCF est configuré avec l'URI de notification `/nsmf-callback/sm-policy-notify/{smContextRef}` et que l'application de session AF (P-CSCF) a atteint le PCF.
  - Pas de QFI disponible : le journal montre `No QFI available for PCC rule ...` ; trop de flux dédiés existent déjà sur la session.
  - La règle PCC ne portait pas de 5QI / GBR : seules les règles avec des valeurs `5qi` et GBR résolubles deviennent des flux dédiés ; vérifiez la règle de facturation / la décision QoS dans la politique.

# Nombre de sessions élevé / Croissance de la mémoire

- **Symptôme** : le nombre de travailleurs de session augmente sans libérations correspondantes.
- **Cause** : les libérations initiées par l'AMF/UE manquées (par exemple, après un redémarrage de l'AMF) laissent des travailleurs de session orphelins ; il n'y a pas de délai d'expiration de session.
- **Résolution** : suivez le nombre de sessions actives en direct avec le compteur `omni_smf_active_pdu_sessions_count` (décomposé par DNN), ou lisez les points de terminaison de gestion `GET /api/statistics` et `GET /api/pdu_session` (voir [Métriques](#)). Les sessions peuvent être vidées par DNN, SUPI, S-NSSAI ou UPF via les points de terminaison de gestion `/api/oam/session/by_*`. L'adresse IP de l'UE est toujours retournée au pool lorsqu'un travailleur sort, donc vider les travailleurs orphelins récupère leurs adresses.

## Corrélation des journaux

Les journaux sont des JSON structurés avec des métadonnées (SUPI, identifiant de session PDU, référence de contexte SM, DNN, procédure). Filtrez par SUPI pour tracer le cycle de vie de la session d'un abonné. Ajustez le niveau de journalisation d'exécution via `POST /api/oam/log_level` sans redémarrage.

