← Vue d'ensemble

Configuration d'OmniUDM

OmniUDM est configuré à partir des variables d'environnement au démarrage via `config/runtime.exs`, sous la clé d'application `:omniudm`. Chaque valeur a une valeur par défaut intégrée, donc le processus commence sans environnement défini ; les déploiements en production doivent définir au moins les valeurs SBI, NRF, UDR, PLMN et `hnet_key_dir`.

Exemple de configuration

```
config :omniudm,
  udr_uri: "http://127.0.0.22:7777",
  sbi_scheme: "http",
  sbi_addr: "127.0.0.12",
  sbi_port: 7777,
  nrf_uri: "http://127.0.0.1:7777",
  mcc: "999",
  mnc: "70",
  prometheus_metrics_port: 9568,
  heartbeat_interval: 10_000,
  hnet_key_dir: "/etc/omniudm/hnet"

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

Référence des paramètres

Paramètre	Type	Requis	Par défaut
`udr_uri`	chaîne (URI)	Non	`http://127.0.0.22:7777`
`sbi_scheme`	chaîne	Non	`http`
`sbi_addr`	chaîne (IP)	Non	`127.0.0.12`
`sbi_port`	entier	Non	`7777`

Paramètre	Type	Requis	Par défaut
`nrf_uri`	chaîne (URI)	Non	`http://127.0.0.1:7777`
`mcc`	chaîne	Non	`999`
`mnc`	chaîne	Non	`70`
`prometheus_metrics_port`	entier	Non	`9568`
`heartbeat_interval`	entier (ms)	Non	`10000`

Paramètre	Type	Requis	Par défaut
`hnet_key_dir`	chaîne (chemin)	Non	`/etc/omniudm/hnet`

Clés d'ajustement optionnelles

Ces clés sont lues à partir de la configuration de l'application et ne sont pas exposées en tant que variables d'environnement dans `runtime.exs` ; définissez-les dans la configuration si un déploiement doit remplacer les valeurs par défaut.

Paramètre	Type	Par défaut	Description
`default_nssai`	carte	tranches eMBB / URLLC / mIoT lab (voir ci-dessous)	NSSAI par défaut de secours retourné par `GET /{supi}/nssai` lorsque la réponse `am-data` du UDR ne contient pas de `nssai` explicite. Définissez ceci pour annoncer les vraies tranches du réseau, y compris les S-NSSAI qualifiés SD
`nf_capacity`	entier (0-65535)	Par défaut de l'application	Capacité de traitement relative de cette instance NF annoncée dans le profil NF enregistré avec le NRF. Les consommateurs pèsent la sélection NF par capacité. Ajustable à l'exécution via le point de terminaison PATCH de configuration OAM (`PATCH /api/oam/config/{id}`) ; n'a généralement pas de variable d'environnement dédiée
`nf_priority`	entier (0-65535)	Par défaut de l'application	Priorité de sélection relative de cette instance NF annoncée dans le profil NF enregistré avec le NRF (valeur inférieure = priorité plus élevée). Ajustable à l'exécution via le point de terminaison PATCH de configuration OAM (`PATCH /api/oam/config/{id}`) ; n'a généralement pas de variable d'environnement dédiée

Le secours intégré `default_nssai` annonce `defaultSingleNssais` de `sst 1` (et `sst 1 / sd 000001`) et `singleNssais` couvrant `sst 1`, `sst 1 / sd 000001`, `sst 2 / sd 000002` (URLLC), et `sst 3 / sd 000003` (mIoT). Un abonné proposé une tranche que l'AMF n'est pas configuré pour servir est rejeté avec la cause 5GMM #62, donc ce secours doit correspondre à ce que le réseau sert réellement.

La valeur est un objet `Nssai` (TS 29.503 / TS 29.571) avec les sous-champs suivants :

Champ	Type	Sémantique
`defaultSingleNssais`	liste de S-NSSAI	Tranches que l'UE est autorisé à utiliser *sans* les demander explicitement - l'ensemble par défaut abonné que l'AMF applique lorsque l'UE ne demande aucun NSSAI. Doit être un sous-ensemble de `singleNssais`
`singleNssais`	liste de S-NSSAI	L'ensemble complet des tranches abonnées auxquelles l'UE a droit sur ce PLMN

Chaque entrée S-NSSAI est une carte :

Champ	Type	Sémantique
`sst`	entier (0–255)	Type de tranche/service. Valeurs standardisées : `1` = eMBB, `2` = URLLC, `3` = mIoT
`sd`	chaîne (6 chiffres hexadécimaux)	Différenciateur de tranche optionnel qualifiant le `sst`. Omettre pour une tranche uniquement SST ; un SST que le réseau sert *uniquement* sous forme qualifiée SD doit être listé avec son `sd` (une entrée non qualifiée ne correspondra pas)

Un S-NSSAI dans `defaultSingleNssais` qui n'est pas également présent dans `singleNssais`, ou une tranche ici qu'aucun AMF n'est configuré pour servir, conduit à ce que l'AMF rejette l'UE avec la cause 5GMM #62.

Clés du réseau domestique (`hnet_key_dir`)

`hnet_key_dir` est le répertoire contenant les clés privées du réseau domestique utilisées pour le déchiffrement SUCI du Profil ECIES A/B. Chaque carte SIM est provisionnée avec une clé **publique** du réseau domestique et son identifiant de clé (`home_network_public_key_id`) ; OmniUDM doit détenir la clé **privée** correspondante sous ce même identifiant.

Les fichiers de clés sont chargés une fois à la première utilisation et mis en cache en mémoire. Un répertoire qui n'existe pas est toléré (un avertissement est enregistré et le déchiffrement ECIES est effectivement désactivé - les SUCI en Null-scheme fonctionnent toujours).

Nommage des fichiers. L'identifiant de clé et la courbe sont extraits du nom de fichier. Deux conventions sont acceptées :

Convention	Profil A (Curve25519)	Profil B (P-256)
Pointé `<id>.<curve>.key`	`1.x25519.key`	`2.secp256r1.key`
Tireté `<curve>-<id>.key`	`curve25519-1.key` ou `x25519-1.key`	`secp256r1-2.key`

L'identifiant numérique `<id>` doit correspondre à l'`home_network_public_key_id` provisionné sur la SIM. Les fichiers qui ne correspondent pas à un modèle reconnu, et les fichiers sans extension `.key`, sont ignorés avec un avertissement.

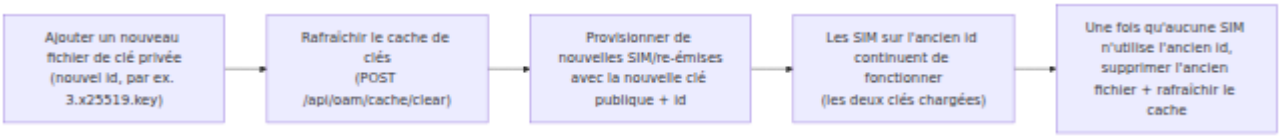
Contenu des fichiers. Chaque fichier contient la clé privée brute dans l'un des trois encodages, détectés automatiquement :

- **PEM** - PKCS#8 (`-----BEGIN PRIVATE KEY-----`) ou SEC1 (`-----BEGIN EC PRIVATE KEY-----`). Le scalaire privé brut est extrait automatiquement. C'est le format de champ recommandé.
- **Hex** - une seule chaîne hexadécimale des octets de la clé privée brute.
- **Raw** - les octets de la clé privée brute tels quels.

Un seul fichier de clé corrompu ou mal formé est ignoré (enregistré comme un avertissement) sans empêcher le chargement des autres clés.

Comment un identifiant de clé est mappé à une SIM. Une SUCI porte l'identifiant de clé que la SIM a utilisé pour dissimuler son SUPI : `suci-0-<mcc>-<mnc>-<routing-ind>-<scheme>-<home_nw_pub_key_id>-<scheme_output>`. OmniUDM lit `<home_nw_pub_key_id>` à partir de la SUCI et sélectionne le fichier de clé privée dont l'identifiant numérique `<id>` correspond. Plusieurs identifiants de clé peuvent donc être servis en même temps - un fichier par identifiant utilisé à travers le parc de SIM - et une SUCI dont l'identifiant n'a pas de clé chargée échoue avec `unknown_home_network_key_id`.

Rotation d'une clé de réseau domestique (sans temps d'arrêt). Étant donné que le déchiffrement est basé sur l'identifiant et que plusieurs identifiants coexistent, une clé est renouvelée en introduisant un *nouvel* identifiant aux côtés de l'ancien plutôt qu'en écrasant un fichier sur place :



Écraser le fichier pour un identifiant encore utilisé sur des SIM actives casse le déchiffrement pour ces SIM (elles présentent l'ancienne clé publique, la vérification MAC échoue). Toujours retirer un identifiant uniquement après que chaque SIM l'utilisant a été reprovisionnée sur un nouveau.

Précautions opérationnelles.

- Traitez le contenu de `hnet_key_dir` comme un matériel secret hautement sensible - la compromission d'une clé privée du réseau domestique permet à toute partie de déchiffrer les SUCI pour cet identifiant de clé. Restreindre les permissions de répertoire et de fichier uniquement au compte de service OmniUDM.

- La clé privée ici doit être le contrepartie de la clé publique provisionnée sur les SIM pour le même identifiant de clé. Un décalage entraîne un échec de la vérification MAC et le rejet de l'authentification.
- Les clés sont mises en cache après le premier chargement. Après avoir ajouté, remplacé ou supprimé des fichiers de clés, le cache doit être rafraîchi (voir l'action de gestion de nettoyage du cache OAM, `POST /api/oam/cache/clear`) ou le processus redémarré pour que le changement prenne effet.

Gestion / API OAM

Une API de gestion/OAM HTTPS est servie (port par défaut `8443`, TLS activé) pour l'inspection et le contrôle opérationnels. Elle est distincte de l'écouteur SBI.

Méthode	Chemin	But
GET	`/api/status/api`	Statut de l'API de gestion
GET	`/api/status/license`	Statut de la licence
GET	`/api/status/nrf`	Statut de l'enregistrement NRF
GET	`/api/status/nf`	Résumé du statut NF
GET	`/api/ue`	Liste des contextes UE actifs en mémoire
GET	`/api/ue/{supi}`	Inspecter un seul contexte UE par SUPI
GET	`/api/statistics`	Statistiques de service, y compris un compte en direct des contextes UE actifs en mémoire
GET	`/api/oam/config`	Lire la configuration effective actuelle
PATCH	`/api/oam/config/{id}`	Mettre à jour une valeur de configuration (PUT est également accepté)
DELETE	`/api/oam/ue_cache/{id}`	Évincer un contexte UE du cache en mémoire
POST	`/api/oam/ue/sdm-notify`	Déclencher manuellement une notification de changement de données SDM pour un SUPI. POST un `ModificationNotification` à chaque `callbackReference` d'abonnement SDM actif (par ex.

Méthode	Chemin	But
		après des changements de provisionnement hors bande)
DELETE	`/api/oam/ue/amf-registration/{id}`	Forcer la désinscription AMF pour un UE. Efface l'enregistrement 3GPP AMF stocké *et* notifie l'AMF de service à son `deregCallbackUri` afin qu'il libère l'UE (raison de désinscription `SUBSCRIPTION_WITHDRAWN`) ; retourne 404 si l'UE n'a pas d'enregistrement AMF
POST	`/api/oam/cache/clear`	Effacer les caches (y compris le cache de clés du réseau domestique)
POST	`/api/oam/log_level`	Changer le niveau de journalisation à l'exécution (`{"level": ...}`)
GET	`/api/oam/resources`	Inventaire des ressources
GET	`/api/oam/diagnostics`	Instantané de diagnostics
POST	`/api/oam/nrf/reregister`	Forcer la réinscription avec le NRF

Journalisation

Paramètre	Valeur	Description
`format`	`{OmniLogger.JsonFormatter, :format}`	Les lignes de journal sont émises sous forme de JSON structuré via le formateur OmniLogger partagé, adapté à l'ingestion par un agrégateur de journaux
`metadata`	`:all`	Toutes les métadonnées de journal (y compris le contexte de session par demande tel que SUPI et procédure) sont incluses dans chaque ligne

Le niveau de journalisation à l'exécution peut être changé à la volée via l'API de gestion/OAM (`POST /api/oam/log_level`). Les matériaux clés (CK/IK/KAUSF/SQN⊕AK) ne sont jamais enregistrés ; seul le contexte non secret tel que le nom du réseau de service et le SUPI déchiffré apparaît dans les journaux.

← Vue d'ensemble

Métriques OmniUDM

Les métriques sont exposées au format Prometheus sur `prometheus_metrics_port` (par défaut `9568`) à `/metrics`.

Métriques du service UDM

Métrique	Type	Étiquettes	
`omni_udm_auth_requests_count`	compteur	`supi`, `result`	Re ve d'a pa (`s` `fa`
`omni_udm_auth_vector_generations_total`	compteur	`result`	Te gé ve d'a pa
`omni_udm_auth_result_count`	compteur	`supi`, `result`	Év rés d'a en `au` (`r` va ra
`omni_udm_uecm_registrations_total`	compteur	`access_type`	En UE tra ac ch d'e AM 3G inc co éti

Métrique	Type	Étiquettes	
			`ac` (`3` `no`
`omni_udm_sdm_queries_total`	compteur	`data_type`	Re do typ (`a` `da` se `ns` co `da`
`omni_udm_active_ue_contexts_count`	jauge	aucun	No co ac me lor et su co
`omni_udm_nrf_registration_status`	jauge	`nf_type`	Sta d'e NR en no

Requêtes utiles

```
# Taux d'échec d'authentification sur 5 minutes
sum(rate(omni_udm_auth_requests_count{result="failure"}[5m]))
  / sum(rate(omni_udm_auth_requests_count[5m]))

# Taux de requêtes SDM par type de données
sum by (data_type) (rate(omni_udm_sdm_queries_total[5m]))

# Taux d'enregistrement UECM réparti par type d'accès (3gpp /
non_3gpp)
sum by (access_type) (rate(omni_udm_uecm_registrations_total[5m]))

# Taux d'échec de génération de vecteur d'authentification
sum(rate(omni_udm_auth_vector_generations_total{result="failure"}
[5m]))
  / sum(rate(omni_udm_auth_vector_generations_total[5m]))

# Alerte si UDM n'est pas enregistré avec le NRF
min(omni_udm_nrf_registration_status) < 1
```

`omni_udm_active_ue_contexts_count` suit le nombre en direct de contextes UE actifs en mémoire ; il est mis à jour à chaque création et suppression de contexte UE. Les points de terminaison OAM `/api/statistics` et `/api/status/nf` rapportent le même compte à la demande.

Métriques BEAM VM

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

← Vue d'ensemble

Opérations OmniUDM

OmniUDM est la fonction de Gestion Unifiée des Données (UDM) du cœur Omnitouch 5G. Elle présente l'identité de l'abonné, les données d'abonnement et de contexte au reste du cœur via les services Nudm standard, sauvegardant toutes les données autorisées dans le Référentiel de Données Unifié (OmniUDR) via `nudr-dr`.

Rôle et Références de Spécification 3GPP

Aspect	Référence
Rôle fonctionnel de l'UDM dans le 5GC	TS 23.501
Services Nudm (SDM / UEAU / UECM)	TS 29.503
Nudr_DataRepository (consommé)	TS 29.504
Schémas de dissimulation et de protection SUCI / SUPI	TS 33.501 §6.12 et Annexe C
Structure et encodage SUCI / SUPI	TS 23.003 §2.2B
Hiérarchie d'authentification et de clé 5G-AKA	TS 33.501 §6.1.3, Annexe A
Encodage de l'identité mobile (SUCI/MSIN BCD)	TS 24.501

Interfaces et Points de Terminaison SBI

OmniUDM enregistre les services suivants auprès du NRF et les sert sur l'écouteur SBI (`sbi_scheme://sbi_addr:sbi_port`):

Service	Version	Objectif
`nudm-sdm`	v2	Gestion des données d'abonné
`nudm-ueau`	v1	Génération de vecteurs d'authentification UE
`nudm-uecm`	v1	Gestion du contexte UE (enregistrement AMF/SMF)

Toutes les données d'abonné et de contexte autorisées sont lues depuis / écrites dans OmniUDR via `nudr-dr/v2` (chemin de base `.../nudr-dr/v2/subscription-data/{ueId}/...`). Tous les corps de requête/réponse sont en JSON. Les erreurs sont retournées sous forme de `application/problem+json` ProblemDetails.

Nudm_UEAuthentication (`nudm-ueau/v1`)

Méthode	Chemin	Description
POST	`/{supiOrSuci}/security-information/generate-auth-data`	Générer un vecteur d'authentification 5G-AKA. Accepte un SUPI ou SUCI ; dé-dissimulera d'abord la SUCI. Nécessite `servingNetworkName` et `ausfInstanceId`
POST	`/{supi}/auth-events`	Enregistrer un événement de résultat d'authentification. Nécessite `nfInstanceId`, `success`, `timeStamp`, `authType`, `servingNetworkName`
PUT	`/{supi}/auth-events/{authEventId}`	Mettre à jour / supprimer un événement d'authentification (le corps AuthEvent, `authRemovalInd=true` le supprime)

Nudm_UECM (`nudm-uecm/v1`)

Méthode	Chemin	Description
PUT	`/{supi}/registrations/amf-3gpp-access`	AMF enregistre l'accès 3GPP pour un UE
PATCH	`/{supi}/registrations/amf-3gpp-access`	Mettre à jour les paramètres de l'enregistrement 3GPP AMF stocké (merge-patch)
GET	`/{supi}/registrations/amf-3gpp-access`	Récupérer l'enregistrement 3GPP AMF stocké
PUT	`/{supi}/registrations/amf-non-3gpp-access`	AMF enregistre l'accès non-3GPP pour un UE
GET	`/{supi}/registrations/amf-non-3gpp-access`	Récupérer l'enregistrement non-3GPP AMF stocké
PUT	`/{supi}/registrations/smf-registrations/{pduSessionId}`	SMF enregistre un contexte de session PDU
DELETE	`/{supi}/registrations/smf-registrations/{pduSessionId}`	SMF désenregistre un contexte de session PDU

Nudm_SDM (`nudm-sdm/v2`)

Méthode	Chemin	Description
GET	`/{supi}`	GetDataSets - récupérer plusieurs ensembles de données sélectionnés par le paramètre de requête obligatoire `dataset-names` (séparés par des virgules `AM`, `SM`, `SMF_SEL`, `UEC_SMF`)
GET	`/{supi}/am-data`	Données d'abonnement à l'accès et à la mobilité (requête `plmn-id` optionnelle)
GET	`/{supi}/sm-data`	Données d'abonnement à la gestion de session (requête `plmn-id` optionnelle)
GET	`/{supi}/smf-select-data`	Données d'abonnement à la sélection SMF (requête `plmn-id` optionnelle)
GET	`/{supi}/nssai`	Données d'abonnement à la sélection de tranche réseau
GET	`/{supi}/ue-context-in-smf-data`	Contexte UE dans SMF (sessions PDU actives) : voir la note ci-dessous
POST	`/{supi}/sdm-subscriptions`	S'abonner aux notifications de

Méthode	Chemin	Description
		changement de données. Nécessite `nfInstanceId`, `callbackReference`, `monitoredResourceUris`
GET	`/{supi}/sdm-subscriptions/{subscriptionId}`	Récupérer un abonnement SDM stocké
PATCH	`/{supi}/sdm-subscriptions/{subscriptionId}`	Modifier un abonnement SDM (merge-patch)
DELETE	`/{supi}/sdm-subscriptions/{subscriptionId}`	Supprimer un abonnement SDM

Dissimulation SUCI / SUPI

Pour protéger l'identité permanente de l'abonné sur l'air, l'UE envoie un **SUCI** (Identifiant d'Abonnement Dissimulé) au lieu du **SUPI** (Identifiant Permanent d'Abonnement, c'est-à-dire `imsi-<IMSI>`). Lorsque l'AUSF demande à OmniUDM de générer un vecteur d'authentification pour un SUCI, OmniUDM **dé-dissimulera la SUCI au SUPI** avant de pouvoir rechercher les données de l'abonné dans l'UDR.

OmniUDM prend en charge les trois schémas de protection définis dans TS 33.501 §6.12 / Annexe C :

Schéma	Nom	Courbe / Crypto	Nécessite une clé HNET ?
0	Schéma nul	Aucun - MSIN est en clair	Non
1	Profil ECIES A	Curve25519 (X25519) + AES-128-CTR + HMAC-SHA-256	Oui
2	Profil ECIES B	secp256r1 / P-256 + AES-128-CTR + HMAC-SHA-256	Oui

Pour le **schéma nul**, le MSIN est transporté en clair ; il est **BCD-décodé** à partir de la sortie du schéma pour reconstruire l'IMSI (selon TS 24.501 - les chiffres MSIN sont encodés en BCD, pas en hexadécimal). Pour **le Profil A et le Profil B**, la dé-dissimulation effectue un ECDH éphémère-statique contre la **clé privée du Réseau d'Accueil** de l'opérateur, dérive les clés de chiffrement/MAC (ANSI-X9.63-KDF avec SHA-256), vérifie le MAC, déchiffre le MSIN dissimulé, et le BCD-décodage pour reconstruire l'IMSI.

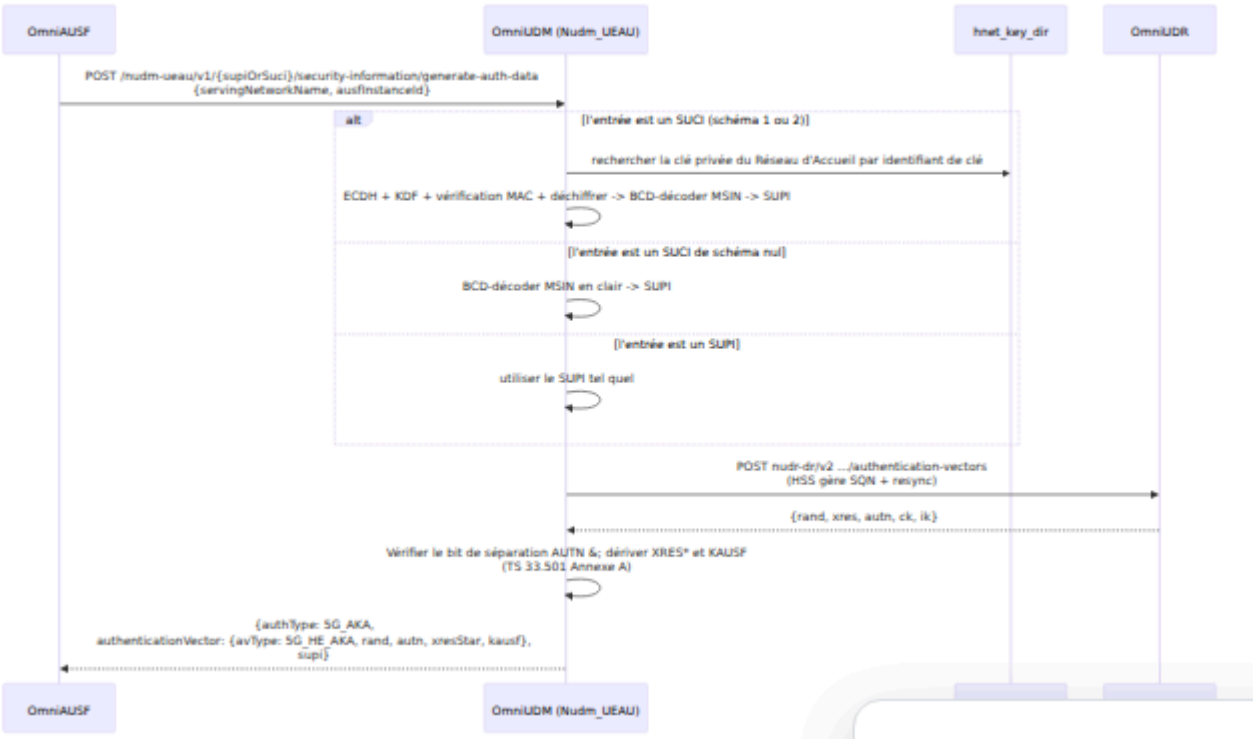
Les clés privées du Réseau d'Accueil pour le Profil A/B sont chargées depuis le répertoire `hnet_key_dir`. La nomination des fichiers, les encodages, et les exigences de sécurité et de rafraîchissement du cache sont documentés dans Configuration → Clés du Réseau d'Accueil (`hnet_key_dir`).

Procédures Clés

Dé-dissimulation SUCI et Génération de Vecteur d'Authentification

L'AUSF demande un vecteur d'authentification ; OmniUDM résout l'identité (dé-dissimulant un SUCI si nécessaire), demande un vecteur au HSS via UDR, puis dérive les clés 5G à partir de celui-ci. **Ki/OPc, gestion de SQN et toute resynchronisation restent dans le HSS derrière UDR** - OmniUDM vérifie que l'AUTN du HSS porte le bit de séparation 5G (bit AMF 0, TS 33.501 §6.1.3.2, pris de l'AMF provisionné de l'abonné) et dérive `XRES*` et `KAUSF` à partir du

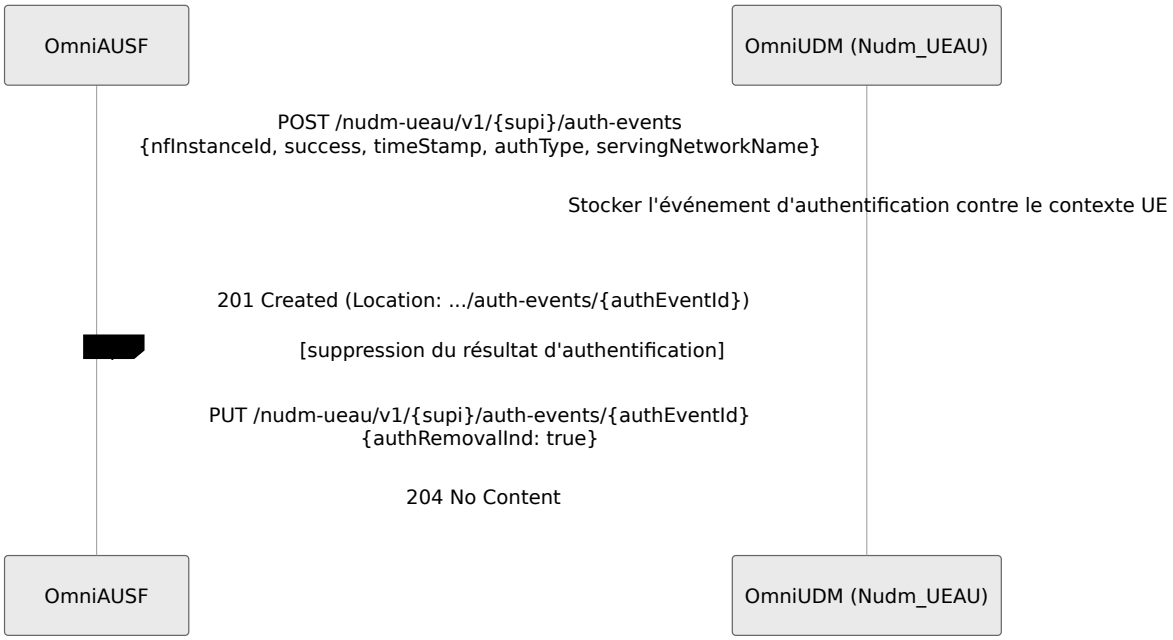
`xres`, `ck` et `ik` retournés. Un `resynchronizationInfo` fourni par l'AUSF est transmis au HSS via UDR.



Les échecs de dé-dissimulation retournent HTTP 403 (`Échec de la dé-dissimulation SUCI`), un schéma de protection non pris en charge retourne HTTP 501, un abonné manquant ou désactivé retourne HTTP 404 (`USER_NOT_FOUND`), et un vecteur HSS invalide (y compris un AUTN sans le bit de séparation) ou une génération de vecteur HSS échouée retourne HTTP 500.

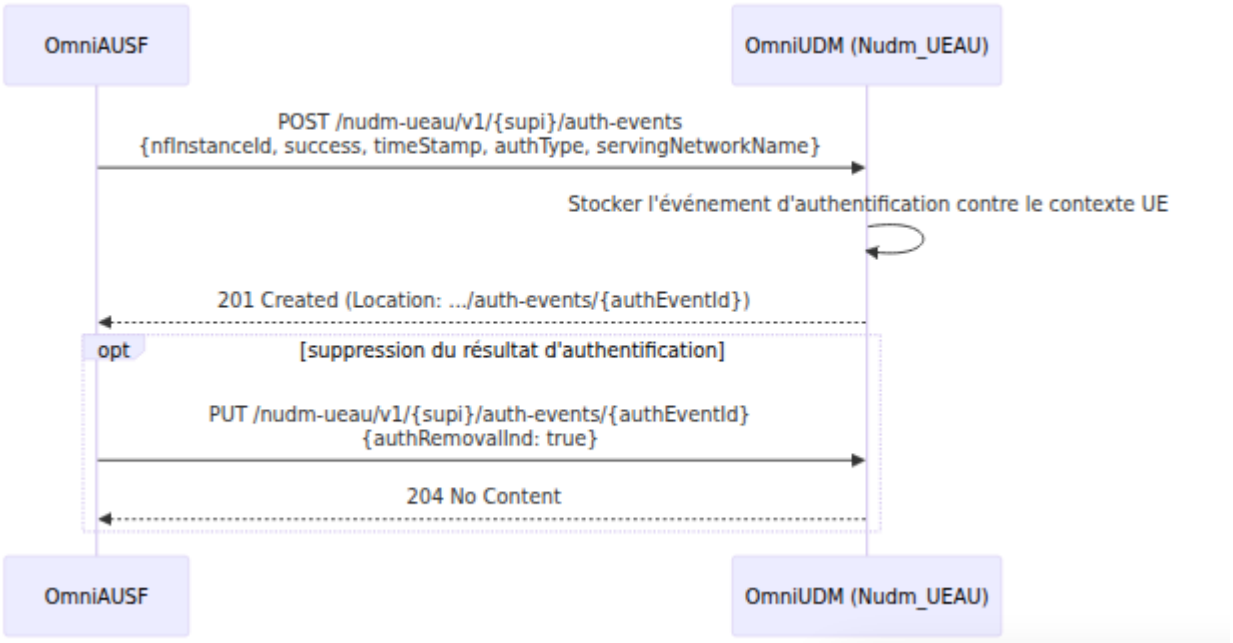
Enregistrement de l'Événement de Résultat d'Authentification

Après que l'AUSF ait terminé l'authentification, il enregistre le résultat. OmniUDM stocke l'événement contre le contexte UE et retourne un `Location` pour l'événement créé ; l'AUSF peut plus tard `PUT` un `AuthEvent` avec `authRemovalInd=true` pour le supprimer (204 No Content).



Récupération des Données d'Abonné SDM (`sm-data`)

Après l'enregistrement, l'AMF/SMF récupère les données d'abonnement. OmniUDM proxy la requête vers UDR, ajoutant le PLMN de service, et retourne la réponse UDR. Pour `sm-data`, un seul objet UDR est enveloppé dans un tableau selon TS 29.503 `SmSubsData`.



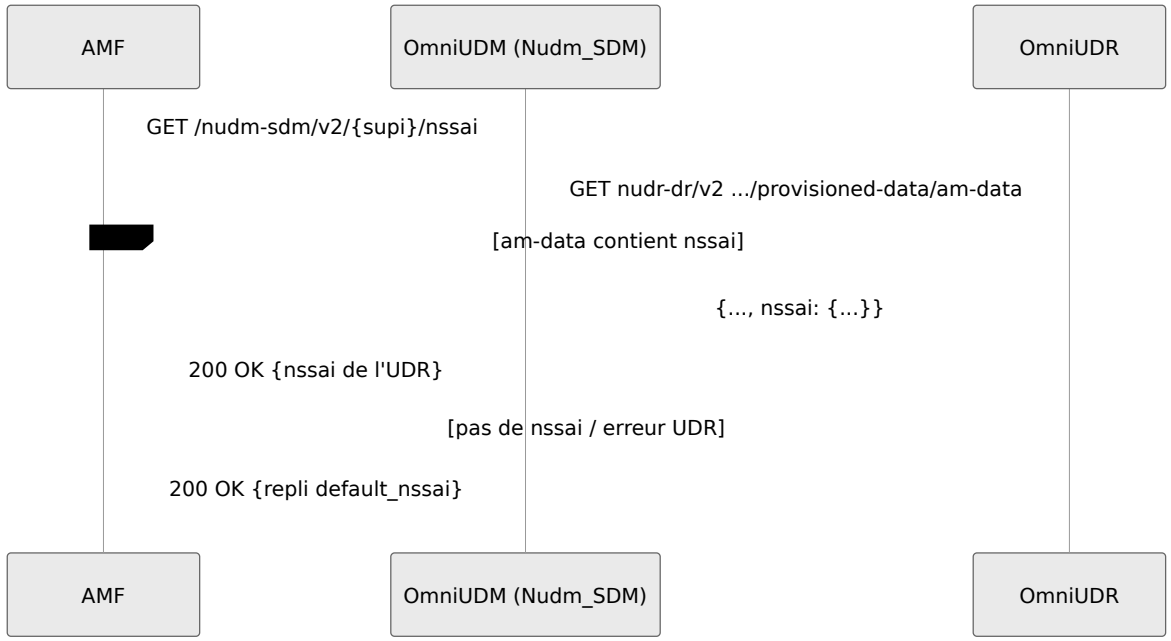
Les flux `am-data`, `smf-select-data`, et GetDataSets (`GET /{supi}`) suivent le même modèle de proxy contre UDR. GetDataSets assemble uniquement les

ensembles de données demandés qu'il peut résoudre, en sautant ceux qui échouent. `nssai` est dérivé de la réponse `am-data` de l'UDR (voir ci-dessous).

> **Le contexte-UE-dans-SMF retourne le contexte de session PDU en direct.** `GET /{supi}/ue-context-in-smf-data` et l'ensemble de données `UEC_SMF` dans GetDataSets retournent les contextes de session PDU actifs de l'UE tels que rapportés par le SMF via la procédure d'enregistrement SMF Nudm_UECM. Chaque enregistrement SMF stocké est projeté dans la carte `pduSessions` (indexée par l'ID de session PDU) selon TS 29.503. La carte est vide (`{"pduSessions": {}}`) uniquement lorsque l'UE ne détient réellement aucun enregistrement SMF.

Récupération NSSAI avec Repli par Défaut

`GET /{supi}/nssai` récupère `am-data` depuis UDR et extrait le `nssai` intégré. Si UDR retourne `am-data` sans `nssai`, ou si la recherche UDR échoue, OmniUDM retourne le repli configurable `default_nssai` afin que l'AMF reçoive toujours une liste de tranches.



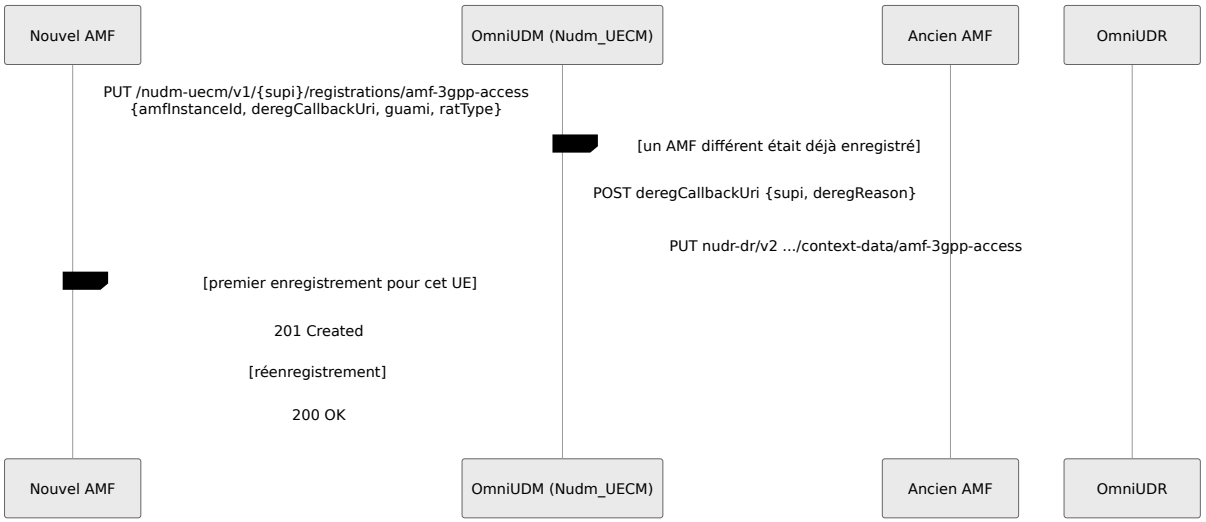
Le repli `default_nssai` doit correspondre aux tranches que le réseau sert réellement - un abonné proposé une tranche que l'AMF n'est pas configuré pour servir est rejeté avec la cause 5GMM #62. Voir Configuration → Clés d'Ajustement Optionnelles.

Notifications de Changement SDM

Les abonnés avec un abonnement SDM actif sont informés des changements ultérieurs : OmniUDM POSTe une `ModificationNotification` à chaque `callbackReference` d'abonnement (asynchrone) lorsque les ressources surveillées changent. Une notification peut également être déclenchée manuellement pour un SUPI via l'API de Gestion/OAM.

Enregistrement AMF UECM et Mobilité

Lorsque un UE s'enregistre, l'AMF s'enregistre pour cet UE auprès d'OmniUDM. Si un AMF différent a été enregistré précédemment, OmniUDM notifie le `deregCallbackUri` de l'ancien AMF afin qu'il puisse libérer l'UE, puis persiste le nouvel enregistrement dans UDR.



Les IEs obligatoires sont validés selon TS 29.503 : l'enregistrement 3GPP AMF nécessite `amfInstanceId`, `deregCallbackUri`, `guami`, `ratType` ; l'enregistrement SMF nécessite `smfInstanceId`, `pduSessionId`, `singleNssai`, `plmnId`, et le corps `pduSessionId` doit correspondre au chemin (une discordance retourne `MANDATORY_IE_INCORRECT`). Les IEs manquants retournent HTTP 400.

← Vue d'ensemble

Dépannage OmniUDM

L'authentification échoue avec 404 Utilisateur non trouvé

OmniUDM a résolu l'identité mais UDR n'a retourné aucune souscription d'authentification. Confirmez que `udr_uri` est accessible, qu'OmniUDR peut atteindre la base de données des abonnés, et que l'IMSI est provisionné. Pour une entrée SUCI, confirmez que le déchiffrement a produit l'IMSI attendu (vérifiez la ligne de journal "De-concealed SUCI to ...").

L'authentification échoue avec 403 Échec du déchiffrement SUCI

OmniUDM n'a pas pu déchiffrer la SUCI. Le champ de détail indique la cause spécifique :

- `unknown_home_network_key_id` - aucune clé privée pour l'identifiant de clé de la SUCI n'est chargée. Confirmez qu'un fichier de clé pour ce `home_network_public_key_id` existe dans `hnet_key_dir` et suit une convention de nommage reconnue, et que le cache de clés a été actualisé depuis l'ajout du fichier.
- `ecies_mac_verification_failed` - la clé privée chargée ne correspond pas à la clé publique sur la SIM (clé incorrecte ou échangée), ou la SUCI est corrompue. Vérifiez la paire de clés provisionnée sur les SIM par rapport à la clé privée déployée pour cet identifiant.
- `ecies_profile_a_scheme_output_too_short` / `ecies_profile_b_scheme_output_too_short` - la sortie du schéma SUCI est tronquée / malformée.
- `invalid_suci_format` - la chaîne SUCI n'est pas bien formée.

L'authentification échoue avec 501 Schéma de protection non pris en charge

La SUCI a utilisé un schéma de protection autre que 0 (Null), 1 (Profil A) ou 2 (Profil B). Seuls ces trois sont pris en charge.

L'authentification échoue avec 500 Erreur interne

Les champs de vecteur HSS (`rand`/`xres`/`ck`/`ik`/`autn`) étaient absents, non valides en hexadécimal, ou de mauvaise longueur ; l'AUTN manquait du bit de séparation 5G ; ou la génération de vecteur HSS a échoué. La ligne de journal `Invalid auth key material ... separation_bit_not_set` signifie que l'AMF provisionné de l'abonné dans l'ensemble de clés HSS n'a pas son bit supérieur défini (utilisez `8000`). Sinon, vérifiez l'ensemble de clés de l'abonné dans le HSS et qu'OmniUDR peut atteindre le HSS.

`am-data` / `sm-data` / `smf-select-data` retourne 404 Non trouvé

La recherche UDR a échoué pour cet ensemble de données. Confirmez que les données provisionnées de l'abonné existent dans UDR pour le PLMN de service, que `mcc`/`mnc` (ou le `plmn-id` de la requête) correspondent à la façon dont les données sont provisionnées, et que `udr_uri` est accessible.

UE rejeté par AMF avec cause 5GMM #62 (aucun slice pris en charge)

Le NSSAI servi à l'AMF n'inclut pas un slice que l'AMF est configuré pour servir. Si les `am-data` de l'UDR de l'abonné ne contiennent pas de `nssai`, OmniUDM retourne le `default_nssai` de secours - définissez `default_nssai` (ou provisionnez `nssai` dans UDR) pour correspondre aux véritables slices du réseau, y compris tout S-NSSAI qualifié SD.

Contexte UE ou enregistrement non présent après redémarrage

L'état d'exécution par UE (contexte d'authentification, enregistrements AMF/SMF, souscriptions SDM) est conservé dans un magasin de contexte en mémoire et est reconstruit par ré-enregistrement ; les données d'abonné autoritaires persistent dans OmniUDR. Après un redémarrage, AMF/SMF se ré-enregistrent lors de la prochaine mise à jour d'enregistrement périodique, de rétablissement de session PDU, ou de cycle d'alimentation de l'UE. Si un enregistrement spécifique manque plus tôt que prévu, déclenchez la mise à jour d'enregistrement du NF consommateur plutôt que d'attendre le minuteur périodique.

Notification de changement de données SDM non reçue

Confirmez que l'URI `callbackReference` de l'abonnement est accessible depuis l'hôte OmniUDM. Les notifications sont envoyées de manière asynchrone ; recherchez des avertissements "SDM notification to ... failed". Une notification peut être déclenchée manuellement pour un SUPI via l'API de gestion/OAM (`POST /api/oam/ue/sdm-notify`).

Après avoir changé les clés du réseau domestique, le déchiffrement utilise toujours l'ancienne clé

Les clés du réseau domestique sont mises en cache en mémoire après le premier chargement. Rechargez-les à chaud via l'action de gestion OAM (`POST /api/oam/hnet/reload`) après avoir ajouté ou remplacé des fichiers de clés dans `hnet_key_dir` ; la réponse indique le nombre de clés chargées et leurs identifiants de clé. Aucun redémarrage de processus n'est requis.

UDM non découvrable par d'autres NF

Vérifiez `omni_udm_nrf_registration_status`. Si `0`, confirmez que `nrf_uri` est accessible et que le heartbeat réussit ; l'API de gestion OAM propose une action de ré-enregistrement NRF (`POST /api/oam/nrf/reregister`).