← Vue d'ensemble

Configuration d'OmniSCP

Tous les paramètres réglables par l'opérateur sont fournis via `config/runtime.exs`, chacun étant soutenu par une variable d'environnement afin que les valeurs puissent être remplacées au moment du déploiement sans modifier le fichier. Les valeurs indiquées ci-dessous sont les valeurs par défaut appliquées lorsque la variable d'environnement correspondante n'est pas définie.

```
import Config

config :omniscp,
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr: System.get_env("SBI_ADDR", "127.0.0.17"),
  sbi_port: String.to_integer(System.get_env("SBI_PORT", "7777")),
  nrf_uri: System.get_env("NRF_URI", "http://127.0.0.1:7777"),
  mcc: System.get_env("MCC", "999"),
  mnc: System.get_env("MNC", "70"),
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000")),
  prometheus_metrics_port:
String.to_integer(System.get_env("PROMETHEUS_PORT", "9568")),
  discovery_cache_ttl:
String.to_integer(System.get_env("DISCOVERY_CACHE_TTL", "60000")),
  lb_strategy: String.to_atom(System.get_env("LB_STRATEGY",
"round_robin")),
  max_retries: String.to_integer(System.get_env("MAX_RETRIES",
"2")),
  upstream_timeout:
String.to_integer(System.get_env("UPSTREAM_TIMEOUT", "5000"))

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

Paramètres de `:omniscp`

Paramètre	Type	Requis	Par défaut
`sbi_scheme`	String	Non	`"http"`
`sbi_addr`	String	Non	`"127.0.0.17"`
`sbi_port`	Integer	Non	`7777`
`sbi_certfile`	String	Non	*(non défini)*

Paramètre	Type	Requis	Par défaut
`sbi_keyfile`	String	Non	*(non défini)*
`nrf_uri`	String	Non	`"http://127.0.0.1:7777"`
`mcc`	String	Non	`"999"`
`mnc`	String	Non	`"70"`

Paramètre	Type	Requis	Par défaut
`heartbeat_interval`	Integer (ms)	Non	`10000`
`prometheus_metrics_port`	Integer	Non	`9568`
`discovery_cache_ttl`	Integer (ms)	Non	`60000`

Paramètre	Type	Requis	Par défaut
`lb_strategy`	Atom	Non	`round_robin`
`max_retries`	Integer	Non	`2`

Paramètre	Type	Requis	Par défaut
`upstream_timeout`	Integer (ms)	Non	`5000`

Capacité/priorité du profil NRF

Ces deux valeurs font partie du propre profil NRF de l'SCP. Elles ne sont **pas** définies dans `runtime.exs` et n'ont pas de variable d'environnement dédiée, donc elles prennent normalement leurs valeurs par défaut, mais elles sont réglables à l'exécution via l'API de gestion (`PATCH /api/oam/config/{id}`), ce qui déclenche également une nouvelle inscription NRF afin que la nouvelle valeur soit annoncée.

Paramètre	Type	Requis	Par défaut	Variable d'environnement	Des
`nf_capacity`	Integer (0-65535)	Non	`100`	(aucun, uniquement OAM)	Valeur d annoncé profil NR Les consomr peuvent de la por l'SCP.
`nf_priority`	Integer (0-65535)	Non	`0`	(aucun, uniquement OAM)	Valeur d annoncé profil NR (plus bas haute pr

> **Valeurs par défaut de secours compilées.** L'application contient également des valeurs par défaut de secours compilées utilisées uniquement si la clé `:omniscp` est entièrement absente, et celles-ci diffèrent des valeurs par défaut d'environnement `runtime.exs` ci-dessus (par exemple, `sbi_addr` revient à `127.0.0.200`, `nrf_uri` à `http://127.0.0.10:7777`, et `max_retries` à `1`). En fonctionnement normal, `runtime.exs` fournit chaque clé, donc les valeurs dans les tableaux ci-dessus sont les valeurs par défaut effectives.

Réglages du répartiteur de charge et de surcharge

Ces clés de configuration d'application `:omniscp` régissent le suivi de santé basé sur les échecs et le contrôle de surcharge. Elles n'ont **pas de variable d'environnement** et ne sont pas exposées via l'API OAM, donc elles sont définies uniquement via la configuration de l'application. Chacune est lue en direct à l'exécution, donc un changement dans la configuration de l'application en cours prend effet lors de la prochaine utilisation du chemin de requête sans

recompilation. Les valeurs par défaut indiquées s'appliquent lorsque la clé n'est pas définie.

Paramètre	Type	Requis	Par défaut	Description
`health_check_interval`	Integer (ms)	Non	`15000`	Intervalle du balayage de santé/récupératic en arrière-plan su les instances de producteurs.
`failure_threshold`	Integer	Non	`3`	Échecs de transfert consécutifs avant qu'une instance soit marquée comme non saine et retirée de la rotation.
`recovery_cooldown_ms`	Integer (ms)	Non	`30000`	Temps de récupération aprè lequel une instance non sair est éligible pour revenir à la rotation.
`oci_reduction_threshold`	Integer	Non	`50`	Seuil de contrôle de surcharge régissant quand un producteur surchargé (selon sa métrique de réduction `3gpp-Sbi-Oci`) est reti de l'ensemble de candidats avant l

Paramètre	Type	Requis	Par défaut	Description
				sélection du répartiteur de charge. Voir Opérations → Routage et Découverte.

Réglage de l'écouteur SBI

Le proxy SBI est servi sur HTTP/2. Les paramètres de transport suivants sont fixes dans l'application (non réglables par l'environnement) et sont documentés ici car ils régissent le comportement de concurrence de l'SCP sous charge :

Réglage	Valeur	Description
`max_concurrent_streams`	`5000`	Maximum de flux HTTP/2 concurrents par connexion entrante. Limite le nombre de requêtes SBI en vol qu'une seule connexion de consommateur peut multiplexe à travers le proxy.
`num_acceptors`	`100`	Taille du pool d'acceptation de l'écouteur SBI pour accepter de nouvelles connexions TCP entrantes.

Les requêtes sortantes (amont) vers les producteurs utilisent un pool de connexions partagé Finch fourni par le runtime Omni5G commun, réutilisé pour tout le trafic transféré et indexé par l'hôte du producteur. La taille de son pool n'est pas exposée comme un paramètre d'OmniSCP, donc la concurrence en amont est régie par ce pool partagé ainsi que par `upstream_timeout` (qui limite chaque requête transférée individuelle) et la limite d'entrée `max_concurrent_streams` ci-dessus. Pour faire évoluer un SCP occupé, les

leviers que vous contrôlez sont `upstream_timeout` (des délais d'attente plus courts libèrent les connexions vers le pool plus tôt) et `discovery_cache_ttl` (un ratio de réussite plus élevé élimine les allers-retours NRF du chemin de requête) ; `max_concurrent_streams` limite combien une seule connexion de consommateur peut multiplexe à travers le proxy.

API de gestion / OAM

OmniSCP expose une API de gestion/OAM sur **HTTPS sur le port 8443** pour un statut en lecture seule et un contrôle opérationnel en direct. Ces points de terminaison ne transportent pas de trafic d'abonné ; restreignez l'accès au réseau des opérations. Tous les chemins ci-dessous sont servis sous le chemin de base `/api` de l'écouteur de gestion.

Points de terminaison de statut (lecture seule)

Méthode	Chemin	Description
`GET`	`/api/status/api`	Statut du service API.
`GET`	`/api/status/license`	Statut de la licence.
`GET`	`/api/status/nrf`	Statut de l'inscription NRF pour l'SCP.
`GET`	`/api/status/nf`	Statut NF de l'SCP : adresse/port SBI, taille du cache de découverte, stratégie `lb_strategy` active, et santé du répartiteur de charge par instance.
`GET`	`/api/api/cache`	Contenu du cache de découverte : par entrée le type NF cible, nom du service, nombre d'instances, et TTL restant.
`GET`	`/api/api/health`	Santé du répartiteur de charge : par instance suivie, le drapeau sain et le nombre d'échecs, plus les totaux sains/non sains.
`GET`	`/api/api/statistics`	Statistiques de l'SCP : taille du cache de découverte et nombres d'instances suivies/saines/non saines.
`GET`	`/api/oam/config`	Configuration d'exécution actuelle effective (SBI, NRF, TTL du cache, `lb_strategy`, `max_retries`,

Méthode	Chemin	Description
		`upstream_timeout`, `nf_capacity`, `nf_priority`, niveau de journalisation).
`GET`	`/api/oam/cache`	Vue du cache de découverte (comme `/api/api/cache`).
`GET`	`/api/oam/upstream/drain`	Liste des instances NF en amont actuellement drainées.

Points de terminaison de contrôle

Méthode	Chemin	Description
`PATCH`	`/api/oam/config/{id}`	Met à jour les paramètres d'exécution sélectionnés en direct : `discovery_cache_ttl`, `lb_strategy`, `max_retries`, `upstream_timeout`, `nf_capacity`, `nf_priority`, `log_level`. Les valeurs sont validées (`lb_strategy` doit être l'un de `round_robin`, `weighted`, `priority` ; les clés entières doivent être des entiers). Changer `nf_capacity`/`nf_priority` déclenche également une nouvelle inscription NRF.
`DELETE`	`/api/oam/cache/{id}`	Vide le cache de découverte pour un type NF spécifique, ou passe `all` pour vider l'ensemble du cache.
`POST`	`/api/oam/upstream/drain`	Draine un amont afin que le proxy cesse de le sélectionner. Corps : `{"nf_instance_id": "..."}` (alias `"target"`), optionnel `"reason"`.

Méthode	Chemin	Description
`DELETE`	`/api/oam/upstream/drain/{id}`	Restaure un amont par ID d'instance NF, le rendant à nouveau éligible à la sélection.
`POST`	`/api/oam/log_level`	Change le niveau de journalisation actif à l'exécution. Corps : `{"level": "..."}` (`emergency`, `alert`, `critical`, `error`, `warning`/`warn`, `notice`, `info`, `debug`).
`POST`	`/api/oam/nrf/reregister`	Force une nouvelle inscription immédiate de l'SCP auprès du NRF.

Drainage en amont est un état administratif, séparé du suivi de santé basé sur les échecs. Les instances drainées sont filtrées de la liste des candidats *avant* la sélection du répartiteur de charge, donc un producteur drainé ne reçoit jamais de trafic transféré. Si **chaque** candidat pour un service est drainé, le proxy n'a pas de prochain saut utilisable et les requêtes vers ce service retournent `504 NF_DISCOVERY_FAILURE` (voir Opérations → Réponses d'erreur du proxy et Dépannage).

Journalisation

Paramètre	Type	Requis	Par défaut	Descriptio
`format`	Tuple	Non	`{OmniLogger.JsonFormatter, :format}`	Formate les lignes de journal en tant que JS structuré vi le formateu `OmniLogger` partagé, adapté à l'ingestion par un pipeline de journalisati
`metadata`	Atom / Liste	Non	`:all`	Métadonné de journal à émettre. `:all` inclut chaque clé métadonné attachée à une entrée journal.

Le niveau de journal actif peut être changé à l'exécution sans redémarrage via l'API de gestion, soit `POST /api/oam/log_level` ou `PATCH /api/oam/config/{id}` avec une valeur `log_level`.

← Vue d'ensemble

Métriques OmniSCP

Les métriques sont exposées sur l'écouteur Prometheus (`prometheus_metrics_port`, par défaut `9568`) à `/metrics`. Les valeurs des tags de résultat sur les métriques de proxy sont : `success` (2xx/3xx), `client_error` (4xx), `server_error` (5xx), `error` (connexion/délai d'attente).

Métriques de proxy

Métrique : `omni_scp_proxy_requests_count`
Type : Compteur
Description : Compte des résultats de proxy par requête.
Étiquettes : `target_nf_type`, `result`

Métrique : `omni_scp_proxy_requests_duration_ms`
Type : Résumé
Description : Durée de proxy par requête en millisecondes.
Étiquettes : `target_nf_type`

Métrique : `omni_scp_proxy_requests_total`
Type : Compteur
Description : Total des requêtes proxy par type de NF cible et résultat.
Étiquettes : `target_nf_type`, `result`

Métrique : `omni_scp_proxy_request_duration_ms`
Type : Histogramme (distribution)
Description : Durée de la requête proxy. Seaux : 1, 5, 10, 25, 50, 100, 250, 500, 1000 ms.
Étiquettes : `target_nf_type`

Le SCP est un proxy de transfert sans état et ne conserve aucun état par association, donc il n'expose aucune métrique "associations actives". L'activité en direct est observée à travers le compteur de requêtes proxy ci-dessus (taux, ratio d'erreur) et les chiffres de santé du répartiteur de charge sur l'API OAM.

```
# Taux de requêtes proxy par type de NF cible
sum by (target_nf_type) (rate(omni_scp_proxy_requests_total[5m]))

# Ratio d'erreur serveur à travers tous les producteurs
sum(rate(omni_scp_proxy_requests_total{result="server_error"}
[5m]))
  / sum(rate(omni_scp_proxy_requests_total[5m]))

# Latence proxy p95 par type de NF
histogram_quantile(0.95, sum by (le, target_nf_type)
(rate(omni_scp_proxy_request_duration_ms_bucket[5m])))
```

Métriques du cache de découverte

Métrique : `omni_scp_discovery_cache_hits`
Type : Compteur
Description : Hits du cache de découverte, par service.
Étiquettes : `target_nf_type`, `service_name`

Métrique : `omni_scp_discovery_cache_misses`
Type : Compteur
Description : Misses du cache de découverte, par service.
Étiquettes : `target_nf_type`, `service_name`

Métrique : `omni_scp_cache_hits_total`
Type : Compteur
Description : Hits agrégés du cache de découverte.
Étiquettes : aucune

Métrique : `omni_scp_cache_misses_total`
Type : Compteur
Description : Misses agrégées du cache de découverte.
Étiquettes : aucune

```
# Ratio global de hits du cache (cible > 0.9 dans une topologie
stable)
sum(rate(omni_scp_cache_hits_total[5m]))
  / (sum(rate(omni_scp_cache_hits_total[5m])) +
sum(rate(omni_scp_cache_misses_total[5m])))

# Taux de misses par service (valeurs élevées signifient
redécouverte fréquente de NRF)
sum by (target_nf_type, service_name)
(rate(omni_scp_discovery_cache_misses[5m]))
```

Métrique d'enregistrement NRF

Métrique : `omni_scp_nrf_registration_status`
Type : Jauge
Description : Statut d'enregistrement SCP avec le NRF (`1` = enregistré, `0` = non enregistré).
Étiquettes : `nf_type`

```
# Alerte si le SCP perd son enregistrement NRF
min(omni_scp_nrf_registration_status) < 1
```

Métriques BEAM VM

L'environnement d'exécution exporte également des jauges hôte/VM : `beam_memory_total`, `beam_memory_processes`, `beam_memory_processes_used`, `beam_memory_system`, `beam_memory_atom`, `beam_memory_atom_used`, `beam_memory_binary`, `beam_memory_code`, `beam_memory_ets` (tous en octets), et `beam_processes_count`, `beam_ports_count`, `beam_atom_count`, `beam_vm_uptime` (secondes). Celles-ci sont utiles pour la surveillance de la capacité et des fuites du processus SCP lui-même.

> Les noms de métriques ci-dessus utilisent la normalisation de point à soulignement de Prometheus ; les séries d'histogramme exposent les suffixes habituels `_bucket`, `_sum`, et `_count`.

← Vue d'ensemble

Opérations OmniSCP

Ce guide couvre le rôle de l'SCP dans le 3GPP, les interfaces et points de terminaison qu'il expose, son comportement de routage et de découverte des requêtes, ainsi que les principales procédures d'exécution avec des diagrammes de séquence. Pour chaque paramètre réglable, voir Configuration ; pour les métriques, voir Métriques ; pour la gestion des échecs, voir Dépannage.

Rôle 3GPP et Références de Spécification

L'SCP fournit une **communication indirecte** comme défini dans le TS 23.501. Dans le **Modèle C**, le consommateur effectue lui-même la découverte (via le NRF) mais délègue le transfert des requêtes à l'SCP. Dans le **Modèle D**, le consommateur délègue à la fois la découverte et le transfert à l'SCP : il envoie la requête avec les paramètres `3gpp-Sbi-Discovery-*` et l'SCP effectue la découverte NRF en son nom. OmniSCP prend en charge les deux modèles, ainsi que le transfert direct lorsque le consommateur connaît déjà la cible.

Spécification	Section	Pertinence
TS 23.501	§6.3.1, Annexe E	Rôle de l'SCP et les modèles de communication indirecte (C et D) dans le 5G SBA.
TS 23.501	§7.3	Comportement de traitement des requêtes de l'SCP au sein du SBA.
TS 29.500	§6.10	Routage SBI via l'SCP ; les en-têtes de routage/découverte `3gpp-Sbi-*`.
TS 29.500	§6.10.3	Ensemble d'en-têtes `3gpp-Sbi-Discovery-*` transporté sur les requêtes de découverte déléguées.
TS 29.500	§6.10.3.2	En-tête `3gpp-Sbi-Target-apiRoot` (cible de routage direct) ; sélection du nom de service principal.
TS 29.500	§6.10.3.3	En-tête de réponse `3gpp-Sbi-Producer-Id` (liaison consommateur-producteur).
TS 29.500	§5.2.2.2	L'en-tête `User-Agent` transmet le type NF du demandeur.
TS 29.510	§6.2	Nnrf_NFDiscovery, le service de découverte NRF consommé par l'SCP.
TS 29.510	§6.1	Nnrf_NFManagement, auto-enregistrement de l'SCP et notifications de statut NF.

Référence : TS 23.501, TS 29.500, TS 29.510.

Interfaces et Écouteurs

OmniSCP est un proxy inverse HTTP polyvalent. Les consommateurs NF dirigent leur trafic SBI vers l'écouteur SBI de l'SCP au lieu de directement vers les producteurs. Pour chaque requête, l'SCP résout un producteur cible (à partir d'un en-tête de requête, d'un résultat de découverte NRF mis en cache, ou en interrogeant le NRF), sélectionne une instance via la stratégie de répartition de charge configurée, transfère la requête, et en cas d'échec, réessaie contre une autre instance.

Écouteur	Port par défaut	Schéma	Objec
Proxy SBI	`7777` (`sbi_port`)	HTTP / HTTPS (`sbi_scheme`)	Accepte les req des consomma reçoit les notific statut NRF. Tous chemins non-nc sont transférés producteurs. De `sbi_scheme=ht` `sbi_certfile`/ pour terminer T l'écouteur.
Métriques Prometheus	`9568` (`prometheus_metrics_port`)	HTTP	Sert le point de de collecte `/me` Métriques.
API de gestion / OAM	`8443`	HTTPS	Points de termi statut en lectur de contrôle OAI Configuration – gestion.

L'écouteur SBI est servi sur HTTP/2 avec une limite de flux par connexion et un pool d'acceptation fixe ; ces valeurs de réglage de transport sont décrites dans Configuration ►► Réglage de l'écouteur SBI.

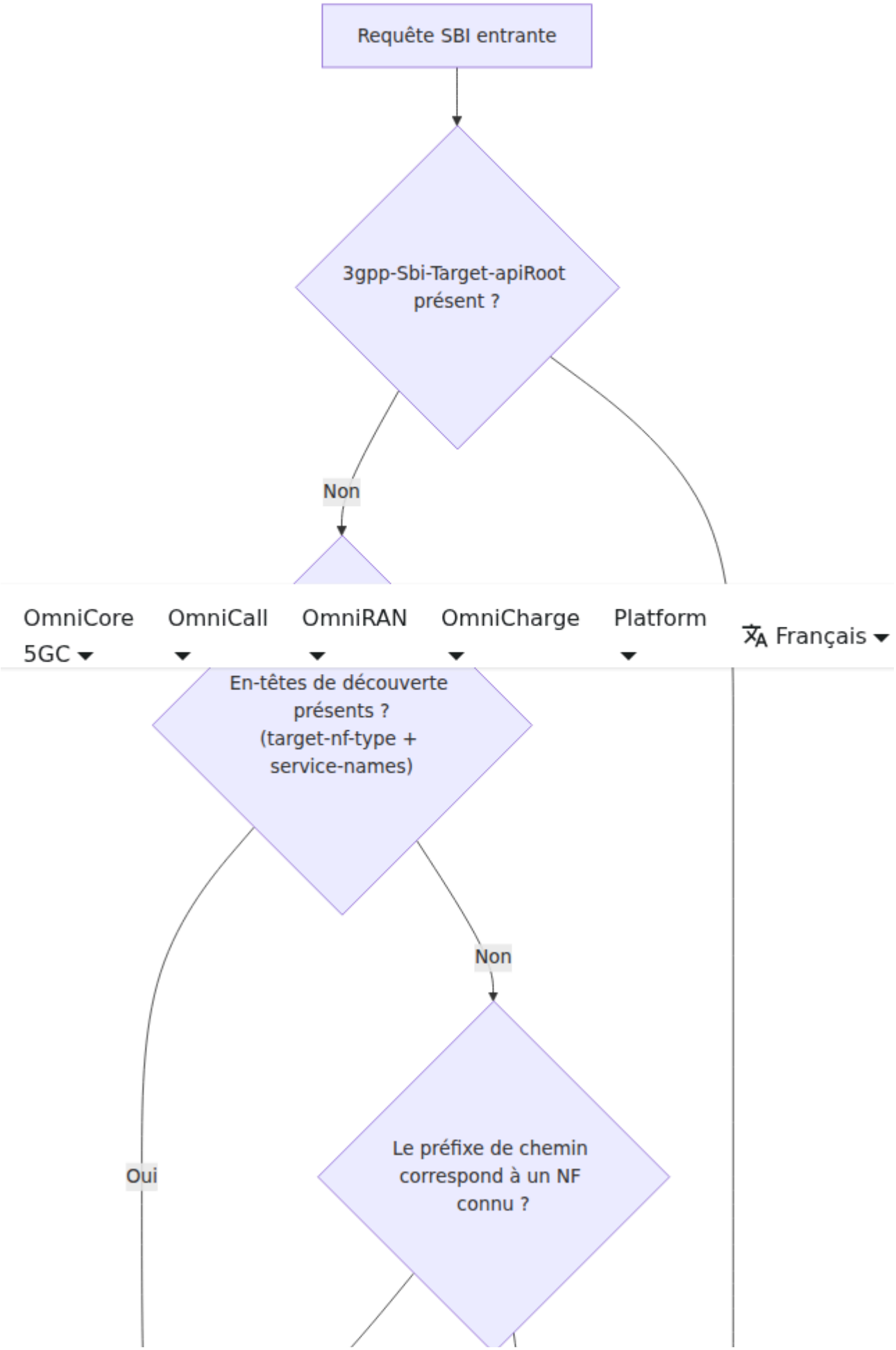
Tableau des points de terminaison SBI

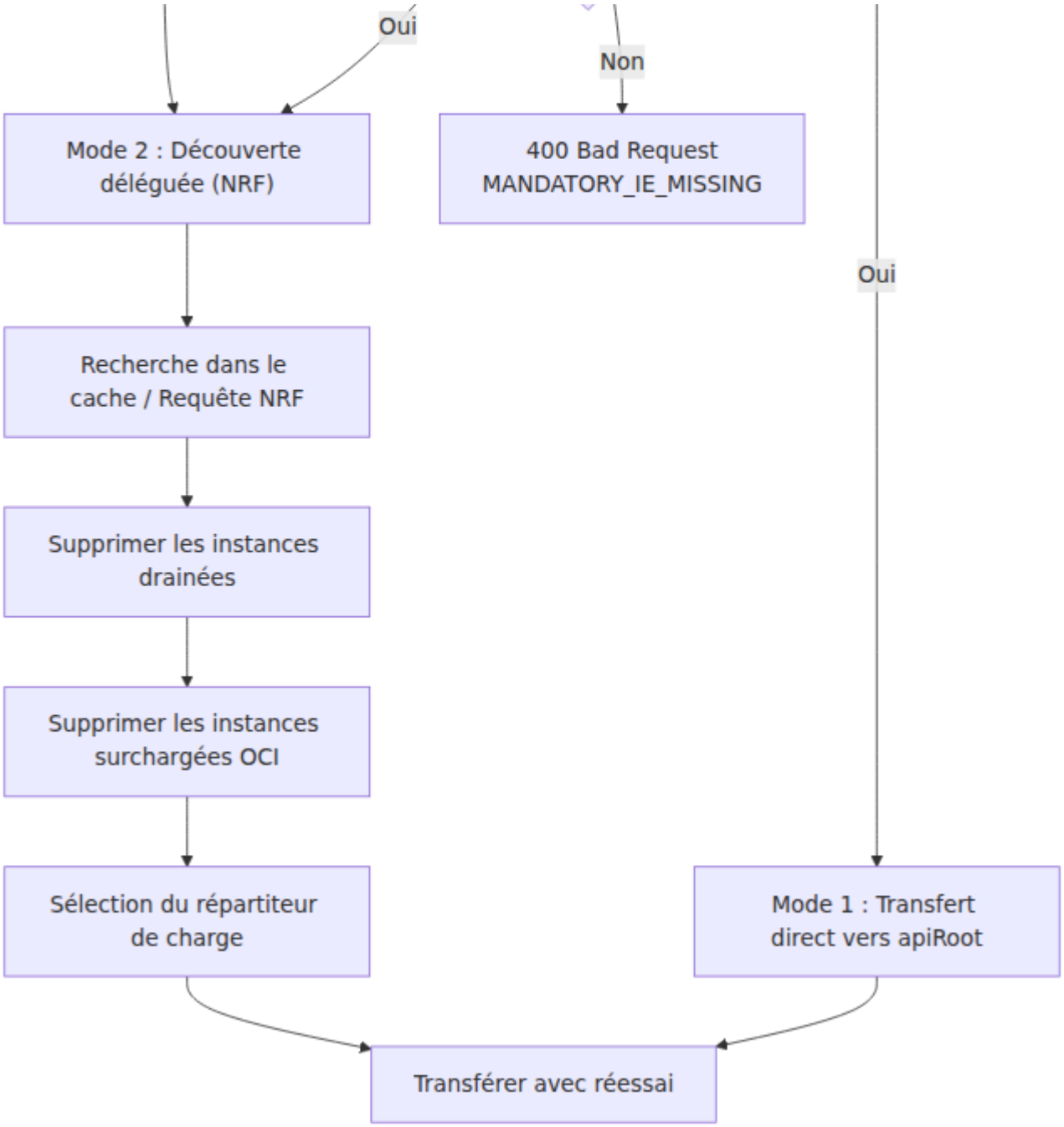
OmniSCP est transparent : exactement un chemin est géré localement, et chaque autre requête est transférée.

Méthode	Chemin	Géré localement	Description
`POST`	`/nnrf-nfm/v1/nf-status-notify`	Oui	Reçoit les notifications de statut NF du NRF. Sur `NF_DEREGISTERED` ou `NF_PROFILE_CHANGED`, le cache de découverte est invalidé pour la seule instance affectée identifiée par son `nfInstanceId` (retombant sur l'éviction d'un seul type NF uniquement si aucun identifiant d'instance n'est présent, et ne faisant rien si aucun des deux n'est présent). Le cache entier n'est jamais effacé. Retourne `204 No Content`.
`*`	tous les autres chemins	Non (transféré)	Transféré au producteur résolu selon le mode de routage actif.

Routage et Découverte

À chaque requête entrante, l'SCP inspecte les en-têtes de requête et le chemin pour décider **comment** résoudre le producteur cible. Trois modes de routage sont évalués par ordre de priorité :


Requête SBI entrante
3gpp-Sbi-Target-apiRoot présent ?
Non
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Français
En-têtes de découverte présents ? (target-nf-type + service-names)
Non
Oui
Le préfixe de chemin correspond à un NF connu ?



Mode 1 : Transfert direct

Lorsque le consommateur envoie `3gpp-Sbi-Target-apiRoot`, il connaît déjà la racine API du producteur. L'SCP supprime les en-têtes spécifiques à l'SCP et transfère la requête directement à cette URI de base sans recherche NRF, cache ou répartition de charge.

Mode 2 : Découverte déléguée (Modèles C/D)

Lorsque le consommateur envoie `3gpp-Sbi-Discovery-target-nf-type` et `3gpp-Sbi-Discovery-service-names`, l'SCP résout lui-même l'ensemble des

producteurs :

1. **Recherche dans le cache** : indexée par `{type NF cible, nom de service principal}`. Le premier nom dans `3gpp-Sbi-Discovery-service-names` est considéré comme le service principal (TS 29.500 §6.10.3.2). En cas de succès dans le TTL, la liste des instances mises en cache est utilisée et aucune requête NRF n'est effectuée.
2. **Découverte NRF** : en cas d'échec, l'SCP interroge le NRF (Nnrf_NFDiscovery), transmettant les paramètres de découverte qu'il a reçus : type NF demandeur, liste PLMN cible, liste S-NSSAI demandeur, ID d'ensemble NF, et ID d'instance NF cible. Le résultat est mis en cache sous la même clé pour `discovery_cache_ttl`.
3. **Filtrage des drains** : toute instance dont le `nfInstanceId` est administrativement drainé (voir Configuration → API de gestion) est retirée de l'ensemble des candidats avant la sélection.
4. **Contrôle de surcharge** : toute instance signalant actuellement une surcharge via son en-tête de réponse `3gpp-Sbi-Oci` (Overload Control Information) à ou au-dessus de `oci_reduction_threshold` pour sa période de validité est retirée de l'ensemble des candidats (TS 29.500 §6.3). C'est un effort de bonne foi : si chaque candidat est surchargé, l'ensemble complet est conservé plutôt que de rejeter la requête.

5. **Sélection du répartiteur de charge** : une instance est choisie parmi les candidats restants en utilisant `lb_strategy`.
6. **Transférer avec réessai** : la requête est transférée ; en cas d'échec, une autre instance est essayée, jusqu'à `max_retries`. À chaque réponse, le `3gpp-Sbi-Oci` annoncé par le producteur est enregistré afin que les requêtes suivantes puissent alléger la charge de celui-ci.

Mode 3 : Inférence basée sur le chemin

Lorsque aucun en-tête de routage n'est présent, l'SCP infère le type NF cible à partir du préfixe de nom de service du chemin de la requête (format de chemin `/<nom-de-service>/<version>/...`) et procède exactement comme au Mode 2.

Préfixe de chemin	Type NF
`nudm-`	UDM
`nausf-`	AUSF
`namf-`	AMF
`nsmf-`	SMF
`npcf-`	PCF
`nudr-`	UDR
`nnssf-`	NSSF
`nbsf-`	BSF
`nnrf-`	NRF
`nchf-`	CHF
`nnef-`	NEF
`naf-`	AF

Si le préfixe de chemin n'est pas dans ce tableau, l'SCP retourne `400 Bad Request` avec la cause `MANDATORY_IE_MISSING`.

Stratégies de répartition de charge

La stratégie de sélection est définie par `lb_strategy` et s'applique aux Modes 2/3 (ensembles d'instances découvertes) :

Stratégie	Logique de sélection
`round_robin`	Fait défiler les instances candidates saines dans l'ordre. Un compteur de rotation séparé est conservé par groupe `{nfType, serviceName}` afin que différents services n'interfèrent pas. C'est la valeur par défaut.
`weighted`	Sélectionne l'instance avec le meilleur score `load - capacity` de son profil NRF, préférant les producteurs à haute capacité et faible charge. `load` manquant par défaut à 50 et `capacity` manquant à 100.
`priority`	Sélectionne l'instance avec la valeur `priority` la plus basse (priorité la plus élevée) de son profil NRF. `priority` manquant par défaut à 65535. Adapté aux déploiements actifs/de secours.

> **D'où proviennent les poids.** Les stratégies `weighted` et `priority` lisent `load`, `capacity` et `priority` du **profil NRF de chaque producteur** ; l'SCP n'a pas de configuration de pondération par producteur propre. Pour influencer la sélection, vous devez régler ces valeurs sur les producteurs (ou dans le NRF), pas sur l'SCP. Les paramètres `nf_capacity`/`nf_priority` de l'SCP (Configuration → Capacité/priority du profil NRF) sont un concept distinct : ils décrivent *l'SCP lui-même* dans son propre profil afin que les pairs puissent pondérer l'SCP, et n'ont aucun effet sur la façon dont l'SCP pèse les producteurs.

Suivi de santé basé sur les échecs. Indépendamment de la stratégie, le répartiteur de charge suit les résultats par instance. Une instance est marquée comme non saine après **3 échecs consécutifs** et est exclue de la sélection ; un balayage de santé en arrière-plan s'exécute toutes les **15 secondes** et renvoie une instance au service une fois qu'un **refroidissement de 30 secondes** s'est écoulé depuis son dernier échec. Si chaque candidat est non sain, l'SCP revient à la liste complète plutôt que d'échouer complètement (favorisant la disponibilité plutôt que l'exclusion stricte). Ce suivi automatique de la santé est distinct de l'état de drainage géré par l'opérateur.

Ces trois seuils (nombre d'échecs, intervalle de balayage et refroidissement) ne sont pas exposés en tant que variables d'environnement et par défaut comme indiqué ci-dessus, mais ce sont de vraies clés de configuration de l'application `:omniscp` (`failure_threshold`, `health_check_interval`, `recovery_cooldown_ms`) qui sont lues en direct à l'exécution. Aucun point de terminaison OAM ne les modifie à l'exécution. Voir Configuration → Réglages du répartiteur de charge et de surcharge. Pour retirer un producteur de la rotation selon un calendrier que vous contrôlez (plutôt que d'attendre le seuil d'échec), utilisez plutôt drainage administratif. Un échec à cette fin est une réponse 5xx ou une erreur de connexion/délai ; toute réponse avec un statut inférieur à 500 (y compris 4xx) est considérée comme un succès et réinitialise le compteur d'échecs de l'instance.

Comportement SBI / Proxy

En-têtes consommés (et supprimés avant le transfert)

Ceux-ci contrôlent le routage et ne sont jamais transférés au producteur.

En-tête	Objectif
`3gpp-Sbi-Target-apiRoot`	Cible de routage direct du Mode 1.
`3gpp-Sbi-Discovery-target-nf-type`	Type NF à découvrir (par ex. `UDM`).
`3gpp-Sbi-Discovery-service-names`	Noms de services séparés par des virgules ; le premier est le principal (utilisé comme clé de cache et service de requête NRF).
`3gpp-Sbi-Discovery-requester-nf-type`	Type NF demandeur ; limite la portée de la requête NRF.
`3gpp-Sbi-Discovery-target-plmn-list`	Liste PLMN cible ; transmise à la requête NRF.
`3gpp-Sbi-Discovery-requester-snssai-list`	Liste S-NSSAI demandeur ; transmise à la requête NRF.
`3gpp-Sbi-Discovery-nf-set-id`	Filtre d'ID d'ensemble NF ; transmis à la requête NRF.
`3gpp-Sbi-Discovery-target-nf-instance-id`	ID d'instance NF cible spécifique ; transmis à la requête NRF.
`3gpp-Sbi-Discovery-requester-nf-instance-id`	ID d'instance demandeur. Consommé et supprimé, mais actuellement non inclus dans la requête de découverte NRF.

L'en-tête `User-Agent` est également inspecté : selon le TS 29.500 §5.2.2.2, il commence par le type NF du demandeur (par ex. `AMF-...`), qui est utilisé comme type NF demandeur si aucun en-tête de découverte explicite n'en

fournit un. Si aucun des deux n'est présent, l'SCP par défaut le type NF demandeur à `SCP` pour la requête NRF.

En-têtes produits

En-tête	Objectif
`3gpp-Sbi-Producer-Id`	Ajouté à la réponse, portant le `nfInstanceId` du producteur qui a servi la requête, afin que le consommateur puisse lier les requêtes suivantes au même producteur (TS 29.500 §6.10.3.3).
`3gpp-Sbi-Lci`	Informations de contrôle de charge du producteur, relayées de bout en bout depuis la réponse du producteur afin que le consommateur reçoive son indice de charge via l'SCP (TS 29.500 §6.3). L'SCP relaie la valeur du producteur et ne synthétise pas la sienne.
`3gpp-Sbi-Load`	L'en-tête compagnon de charge `3gpp-Sbi-Lci`, relayé de bout en bout depuis la réponse du producteur avec `3gpp-Sbi-Lci` (TS 29.500 §6.3).

Les en-têtes de réponse hop-by-hop (`transfer-encoding`, `connection`, `content-length`) ne sont pas renvoyés au consommateur.

Réponses d'erreur du proxy

Lorsque l'SCP ne peut pas compléter une opération, il retourne un corps `ProblemDetails` RFC 7807 selon le TS 29.500.

Statut HTTP	Cause	Condition
`400 Bad Request`	`MANDATORY_IE_MISSING`	Pas d'informations de routage : pas de target-apiRoot, pas d'en-têtes de découverte, et le préfixe de chemin ne correspond à aucun service connu.
`502 Bad Gateway`	`TARGET_NF_NOT_REACHABLE`	Aucune URI SBI n'a pu être résolue à partir d'un profil NF découvert.
`504 Gateway Timeout`	`NF_DISCOVERY_FAILURE`	La découverte (NRF + cache) n'a donné aucune instance NF utilisable, y compris lorsque **chaque** instance candidate pour le service est administrativement drainée.
`500 Internal Server Error`	`SYSTEM_FAILURE`	Erreur interne de proxy réellement inattendue non couverte par les causes plus spécifiques ci-dessus.

> **Remarque sur le cas de tous les drains.** Si chaque candidat pour un service est drainé, le proxy n'a aucun prochain saut utilisable et retourne `504 NF_DISCOVERY_FAILURE` (pas `500`). Dédrain au moins une instance pour restaurer le service. Voir Dépannage.

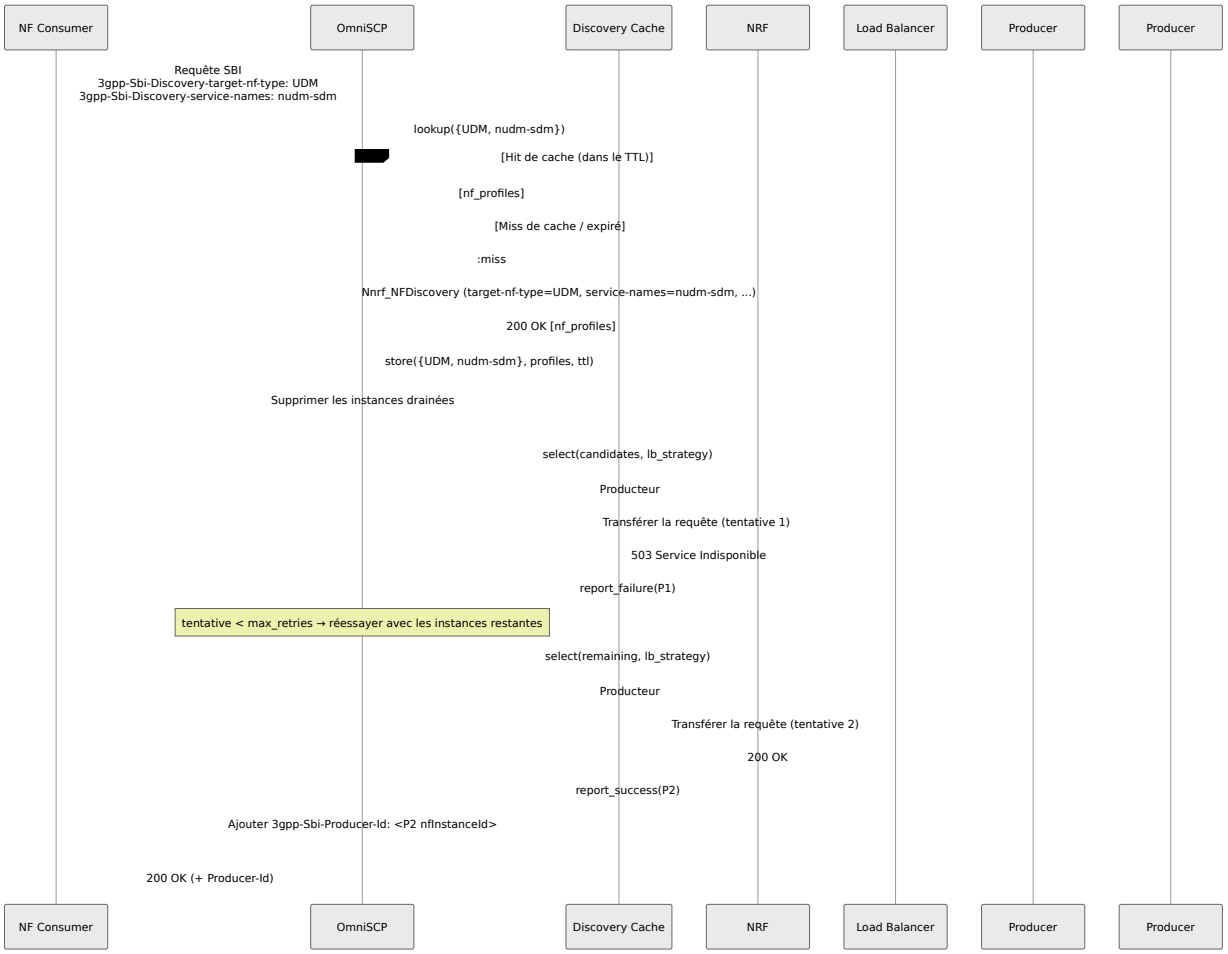
Sémantique de réessai. Une réponse de producteur avec un statut **< 500** (y compris 4xx) est considérée comme une réponse valide et renvoyée au consommateur telle quelle. Une réponse **5xx** ou une **erreur de connexion/délai** compte comme un échec : l'instance échouée est signalée au répartiteur de charge et retirée de l'ensemble des candidats, et une autre instance est essayée. Les réessais continuent jusqu'à ce qu'une réponse non

échouée soit obtenue, que `max_retries` soit atteint, ou que la liste des candidats soit épuisée, auquel cas la dernière réponse (ou erreur) est renvoyée.

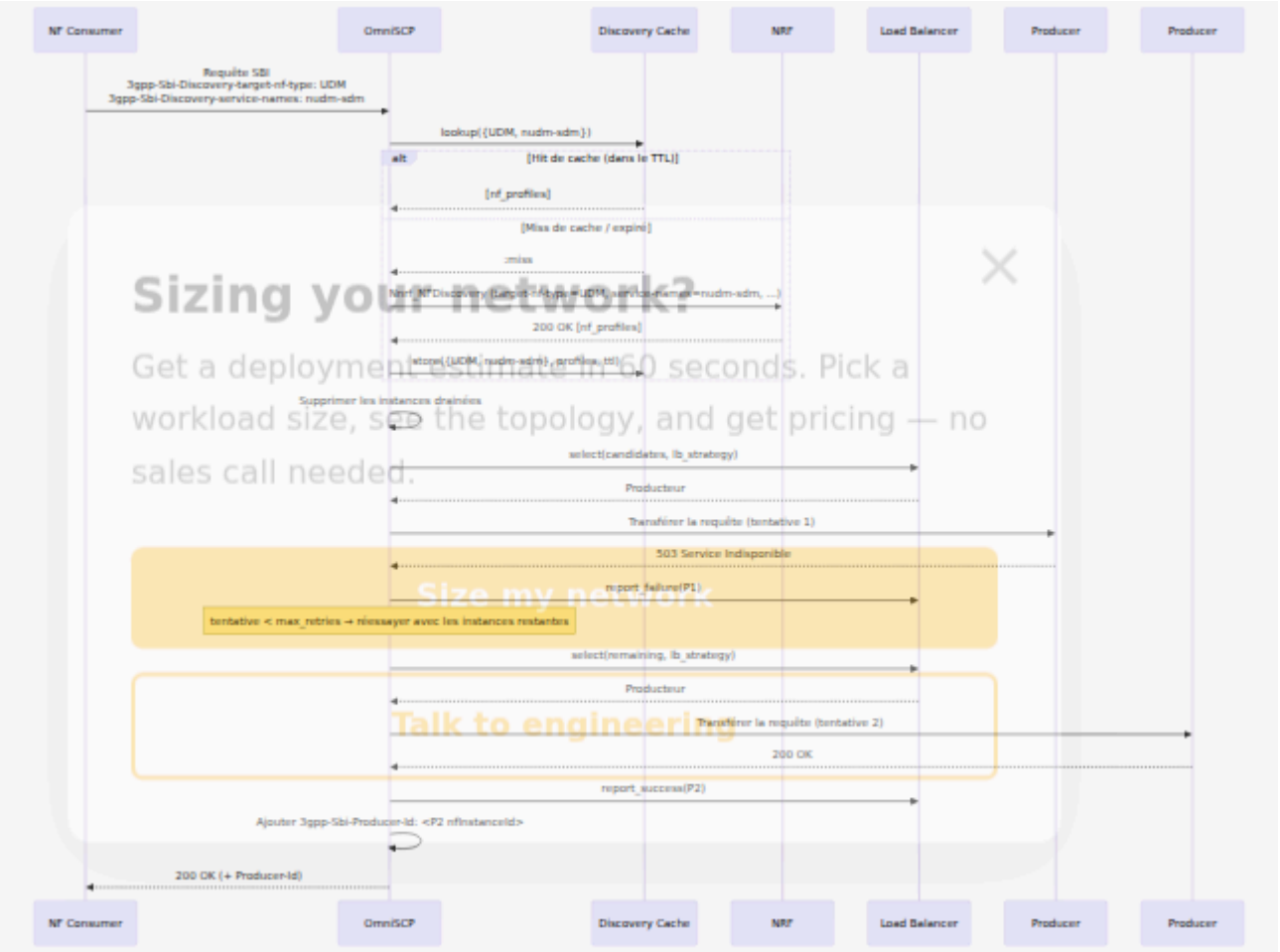
Principales Procédures

Requête transférée, équilibrée avec un réessai (Mode 2/3)

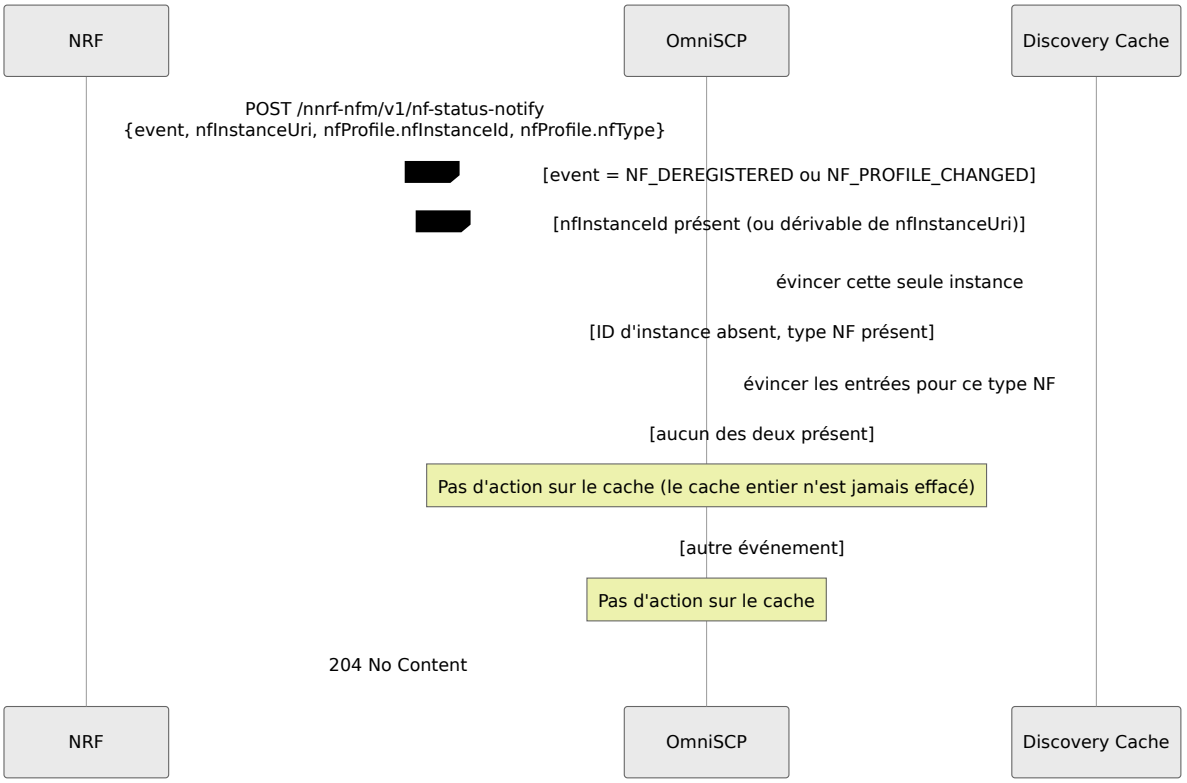
La séquence ci-dessous montre la découverte déléguée, la sélection du répartiteur de charge, une première tentative qui échoue avec un 5xx, et un réessai réussi contre une deuxième instance.



Transfert direct (Mode 1)

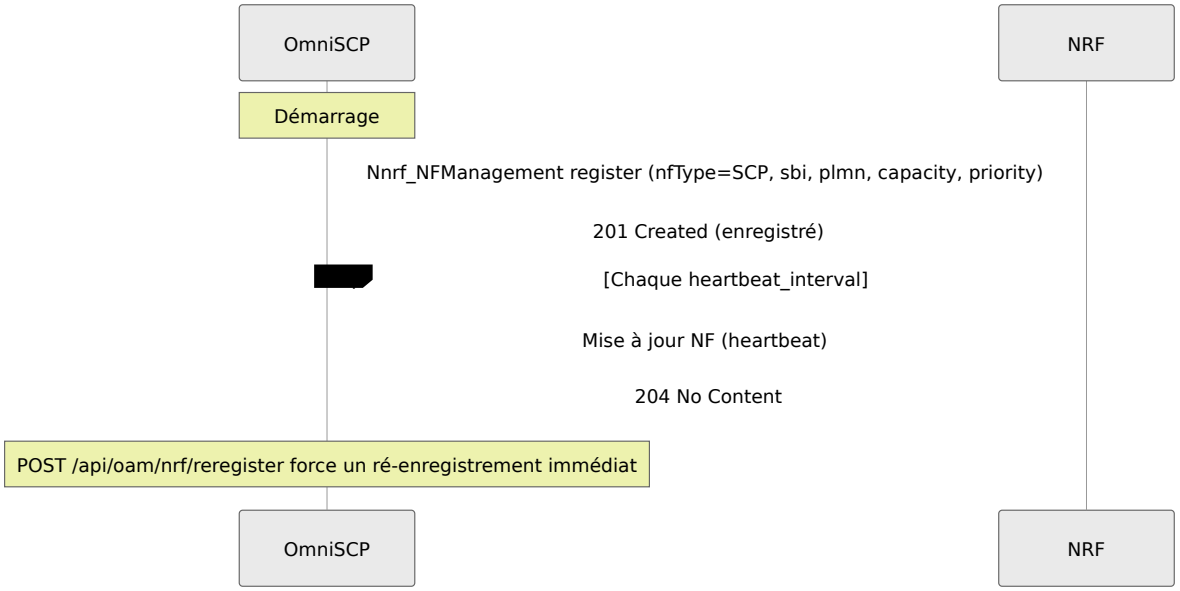


Notification de statut NRF (invalidation du cache)



Auto-enregistrement et heartbeat NRF

L'SCP s'enregistre auprès du NRF en tant que NF `SCP` au démarrage et maintient l'enregistrement actif avec des heartbeats périodiques. Le profil annoncé contient l'adresse/port/schéma SBI, le PLMN (`mcc`/`mnc`), et la `capacity`/`priority` NF. L'enregistrement peut être forcé immédiatement via l'API OAM.

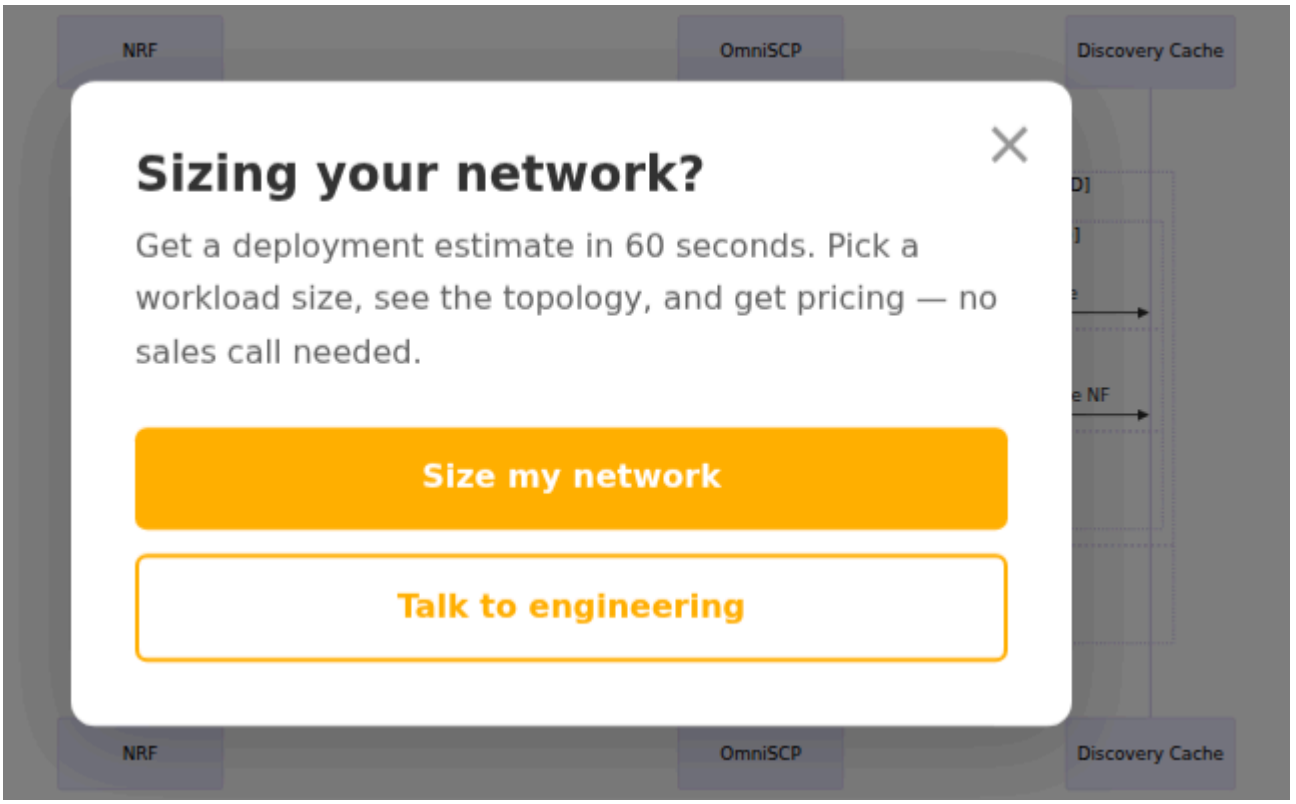


Drainage administratif en amont

Le drainage est un état administratif, distinct du suivi de santé basé sur les échecs automatique. Les instances drainées sont filtrées de la liste des candidats *avant* la sélection du répartiteur de charge, de sorte qu'un producteur drainé ne reçoive jamais de trafic transféré. Utilisez-le pour retirer gracieusement un producteur de la rotation (pour maintenance) sans le désenregistrer du NRF.

Flux de travail de maintenance. Obtenez le `nfInstanceId` du producteur cible auprès du NRF (la valeur avec laquelle le producteur s'est enregistré), puis `POST /api/oam/upstream/drain` avec cet ID et un `reason` optionnel. Les requêtes en cours déjà transférées au producteur ne sont pas interrompues ; seules les *nouvelles* sélections l'ignorent, donc laissez les requêtes existantes se terminer avant de servir le producteur. Confirmez qu'il est hors rotation avec `GET /api/oam/upstream/drain`, et lorsque la maintenance est terminée, `DELETE /api/oam/upstream/drain/{id}` pour le remettre en service.

> **L'état de drainage est en mémoire.** Les entrées de drainage sont conservées dans une table ETS et **ne sont pas persistées lors d'un redémarrage de l'SCP** : si le processus SCP redémarre alors qu'un producteur est drainé, ce producteur devient à nouveau sélectionnable et vous devez réémettre le drainage. Planifiez les fenêtres de maintenance en conséquence, et vérifiez à nouveau `GET /api/oam/upstream/drain` après tout redémarrage de l'SCP.


NRF
OmniSCP
Discovery Cache
Sizing your network?
Get a deployment estimate in 60 seconds. Pick a workload size, see the topology, and get pricing — no sales call needed.
Size my network
Talk to engineering
NRF
OmniSCP
Discovery Cache

← Vue d'ensemble

Dépannage OmniSCP

Les causes d'erreur et les codes d'état mentionnés ici sont décrits dans Opérations → Réponses d'erreur du proxy. Les clés de configuration et les points de terminaison OAM se trouvent dans Configuration; les métriques dans Métriques.

400 Bad Request: MANDATORY_IE_MISSING

Symptômes : Le consommateur reçoit `400` avec la cause `MANDATORY_IE_MISSING`.

Causes possibles :

- La requête ne contenait ni `3gpp-Sbi-Target-apiRoot` ni le couple `3gpp-Sbi-Discovery-target-nf-type` + `3gpp-Sbi-Discovery-service-names`.
- Dépendre de l'inférence basée sur le chemin (Mode 3) pour un préfixe de service qui n'est pas dans la carte intégrée.

Résolution :

1. Confirmer que le consommateur envoie un target-apiRoot ou les deux en-têtes de découverte.
2. Si vous dépendez de l'inférence par chemin, vérifiez que le préfixe de chemin se trouve dans le tableau Mode 3. Pour un préfixe qui n'est pas dans ce tableau, envoyez des en-têtes de découverte explicites.

504 Gateway Timeout: NF_DISCOVERY_FAILURE

Symptômes : Le consommateur reçoit `504` avec la cause `NF_DISCOVERY_FAILURE`.

Causes possibles :

- Le NRF n'a retourné aucune instance pour le type/service NF demandé.
- Le NRF est inaccessible.

Résolution :

1. Vérifiez `nrf_uri` et la connectivité réseau vers le NRF.
2. Confirmez que le type NF cible est enregistré dans le NRF.
3. Vérifiez si une notification d'état NRF récente a invalidé le cache pour le type NF et que le producteur ne s'est pas réenregistré.

504 Gateway Timeout: NF_DISCOVERY_FAILURE (toutes les instances drainées)

Symptômes : Le consommateur reçoit `504` avec la cause `NF_DISCOVERY_FAILURE` pour un service qui a des producteurs enregistrés.

Causes possibles :

- Chaque instance candidate pour le service est administrativement drainée, donc le proxy n'a pas de prochain saut utilisable. Cette condition correspond à `504 NF_DISCOVERY_FAILURE`, le même statut que la découverte ne retournant aucune instance.

Résolution :

1. Vérifiez `GET /api/oam/upstream/drain`. Si toutes les instances pour le service sont drainées, dédrainiez celles nécessaires via `DELETE`

`/api/oam/upstream/drain/{id}`.
2. Si aucune instance n'est drainée et que le service est enregistré, considérez cela comme un cas d'échec de découverte (voir la section NF_DISCOVERY_FAILURE ci-dessus).

500 Internal Server Error: SYSTEM_FAILURE

Symptômes : Le consommateur reçoit `500` avec la cause `SYSTEM_FAILURE`.

Causes possibles :

- Une véritable erreur interne inattendue du proxy non couverte par les causes plus spécifiques (`400`, `502`, `504`) ci-dessus.

Résolution :

1. Examinez les journaux SCP pour les détails de l'erreur interne.

502 Bad Gateway: TARGET_NF_NOT_REACHABLE

Symptômes : Le consommateur reçoit `502` avec la cause `TARGET_NF_NOT_REACHABLE`.

Causes possibles :

- Aucun URI SBI n'a pu être résolu à partir d'un profil NF découvert.
- Tous les producteurs sélectionnés ont retourné des erreurs de connexion après des tentatives de réessai.

Résolution :

1. Confirmez que les producteurs fonctionnent et sont accessibles aux adresses SBI dans leurs profils NRF.
2. Si les producteurs sont lents, augmentez `upstream_timeout`.

3. Examinez `max_retries`; `0` transforme un échec unique en une erreur immédiate.
4. Vérifiez `GET /api/api/health` pour les instances marquées comme non saines après des échecs répétés.

Délais d'attente en amont / haute latence du proxy

Symptômes : Durée élevée `omni_scp_proxy_request_duration_ms`; les requêtes échouent de manière intermittente avec un résultat `error`.

Causes possibles :

- Les producteurs répondent plus lentement que `upstream_timeout`.
- Faible taux de réussite du cache forçant un aller-retour NRF sur de nombreuses requêtes.

Résolution :

1. Inspectez l'histogramme de latence et le taux d'erreur par type NF.
2. Comparez `omni_scp_cache_hits_total` vs `omni_scp_cache_misses_total`; un taux de raté élevé signifie une découverte NRF fréquente, donc envisagez d'augmenter `discovery_cache_ttl`.
3. Ajustez `upstream_timeout` pour correspondre aux temps de réponse réalistes des producteurs. N'oubliez pas qu'une tentative expirée ajoute le temps d'attente complet à la latence de bout en bout avant un réessai.

Routage obsolète / obsolescence du cache de découverte

Symptômes : Les requêtes sont transmises à des producteurs qui ont déménagé ou redémarré, provoquant des erreurs jusqu'à ce que les entrées expirent.

Causes possibles :

- Un producteur a changé d'adresse ou redémarré sans se désinscrire du NRF, donc le SCP conserve un profil obsolète jusqu'à l'expiration du TTL.

Résolution :

1. Réduisez `discovery_cache_ttl` pour raccourcir la fenêtre d'obsolescence.
2. Assurez-vous que les producteurs se désinscrivent du NRF lors de l'arrêt afin que le SCP reçoive une notification `NF_DEREGISTERED` et invalide le type NF affecté.
3. Videz explicitement via `DELETE /api/oam/cache/{nf_type}` (ou `DELETE /api/oam/cache/all`).

Enregistrement NRF non maintenu

Symptômes : `omni_scp_nrf_registration_status` indique `0`.

Causes possibles :

- `nrf_uri` incorrect ou NRF inaccessible.
- Mismatch PLMN (`mcc`/`mnc`) par rapport à la configuration NRF.

Résolution :

1. Vérifiez `nrf_uri` et la connectivité.
2. Confirmez que `mcc`/`mnc` correspondent à la configuration PLMN du NRF.
3. Forcez une réinscription avec `POST /api/oam/nrf/reregister` et examinez les journaux de démarrage.