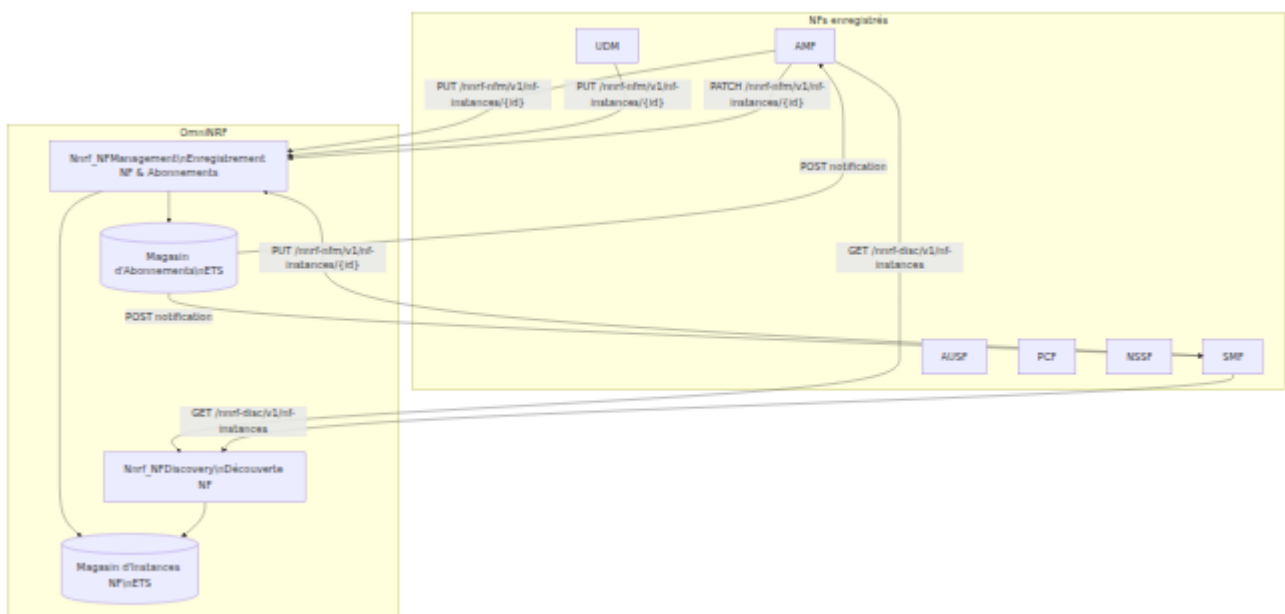


Opérations OmniNRF

1. Vue d'ensemble des composants

OmniNRF est la Fonction de Répertoire de Réseau (NRF) pour le cœur 5G d'Omnicore. Il fournit des services centralisés d'enregistrement de NF, de découverte et de notification de changement d'état à tous les autres NF dans le 5GC. Tous les instances de NF s'enregistrent auprès du NRF au démarrage et maintiennent leur vivacité via des battements de cœur périodiques. Les consommateurs interrogent le NRF pour résoudre les points de terminaison NF avant d'émettre des appels SBI.



2. Rôle 3GPP et Références de Spécification

Aspect	Référence
Définition fonctionnelle du NRF	TS 23.501 Section 6.2.6
Service Nnrf_NFManagement	TS 29.510 Section 5.2
Service Nnrf_NFDiscovery	TS 29.510 Section 5.3
Modèle de données NFProfile	TS 29.510 Section 6.1
Notification d'état NF	TS 29.510 Section 5.2.2.8
Procédure de battement de cœur	TS 29.510 Section 5.2.2.4
Modèle de données SearchResult	TS 29.510 Section 6.2

OmniNRF implémente le rôle de producteur SBI pour `Nnrf_NFManagement` et `Nnrf_NFDiscovery` tel que défini dans TS 29.510.

3. Points de terminaison SBI

Tous les points de terminaison sont HTTP/1.1 avec `Content-Type: application/json`.

Nnrf_NFManagement (TS 29.510 Section 5.2)

Méthode	Chemin	Description	Statut
PUT	<code>/nnrf-nfm/v1/nf-instances/{nfInstanceId}</code>	Enregistrer ou ré-enregistrer un NF	201 Created / 204 OK
PATCH	<code>/nnrf-nfm/v1/nf-instances/{nfInstanceId}</code>	Battement de cœur (corps vide) ou mise à jour JSON Patch	200 OK / 204 OK
DELETE	<code>/nnrf-nfm/v1/nf-instances/{nfInstanceId}</code>	Désenregistrer NF	200 OK / 204 OK
GET	<code>/nnrf-nfm/v1/nf-instances/{nfInstanceId}</code>	Récupérer le profil NF	200 OK
GET	<code>/nnrf-nfm/v1/nf-instances</code>	Lister toutes les instances NF	200 OK
POST	<code>/nnrf-nfm/v1/subscriptions</code>	S'abonner aux notifications d'état NF	201 Created
DELETE	<code>/nnrf-nfm/v1/subscriptions/{subscriptionId}</code>	Se désabonner	200 OK

Nnrf_NFDiscovery (TS 29.510 Section 5.3)

Méthode	Chemin	Description	Succès
GET	/nnrf-disc/v1/nf-instances	Découvrir des instances NF par type et filtres	200 OK

Paramètres de requête de découverte

Paramètre	Requis	Description
target-nf-type	Oui	Type de NF recherché (par exemple UDM, SMF)
requester-nf-type	Oui	Type de NF effectuant la demande (par exemple AMF)
service-names	Non	Liste de noms de services séparés par des virgules pour filtrer
plmn-id	Non	ID PLMN encodé en JSON, par exemple {"mcc": "999", "mnc": "70"}
limit	Non	Nombre maximum de résultats à retourner

4. Référence de Configuration

OmniNRF est configuré via l'environnement d'application Elixir sous la clé `:omninrf`. Dans une version, ces valeurs sont généralement définies dans `config/runtime.exs` ou via des variables d'environnement.

Exemple de Configuration

```
config :omninrf,  
  sbi_scheme: "http",  
  sbi_addr: "127.0.0.10",  
  sbi_port: 7777,  
  mcc: "999",  
  mnc: "70",  
  heartbeat_timeout: 30_000,  
  cleanup_interval: 10_000,  
  plmn_list: [%{mcc: "999", mnc: "70"}]
```

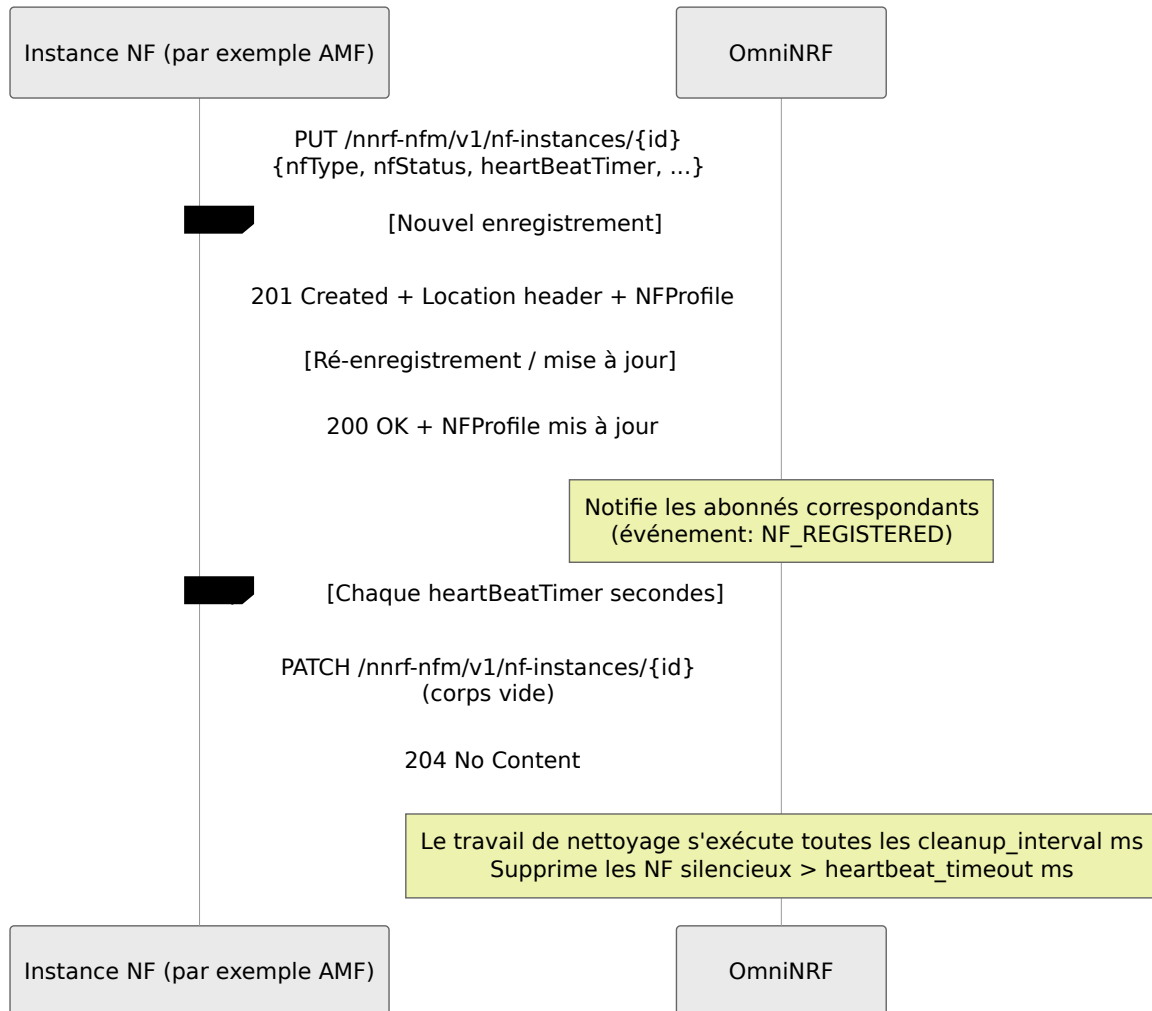
Tableau des Paramètres

Paramètre	Type	Par défaut	Description
sbi_scheme	chaîne	"http"	Schéma URI pour l'écouteur SBI (<code>http</code> ou <code>https</code>)
sbi_addr	chaîne	"127.0.0.10"	Adresse IP à laquelle le serveur HTTP SBI se lie
sbi_port	entier	7777	Port TCP sur lequel le serveur HTTP SBI écoute
mcc	chaîne	"999"	Code de Pays Mobile pour le PLMN servi
mnc	chaîne	"70"	Code de Réseau Mobile pour le PLMN servi
heartbeat_timeout	entier (ms)	30000	Temps après le dernier battement de cœur avant qu'une instance NF soit considérée comme obsolète et supprimée. Le NRF utilise également cette valeur pour définir <code>heartBeatTimer</code> dans le <code>NFProfile</code> si le NF enregistrant n'en fournit pas

Paramètre	Type	Par défaut	Description
<code>cleanup_interval</code>	entier (ms)	<code>10000</code>	Fréquence d'exécution du travail de nettoyage des enregistrements obsolètes
<code>plmn_list</code>	liste de cartes	<code>[%{mcc: mcc, mnc: mnc}]</code>	Liste des PLMN servis par ce NRF, utilisée pour le filtrage PLMN lors de la découverte

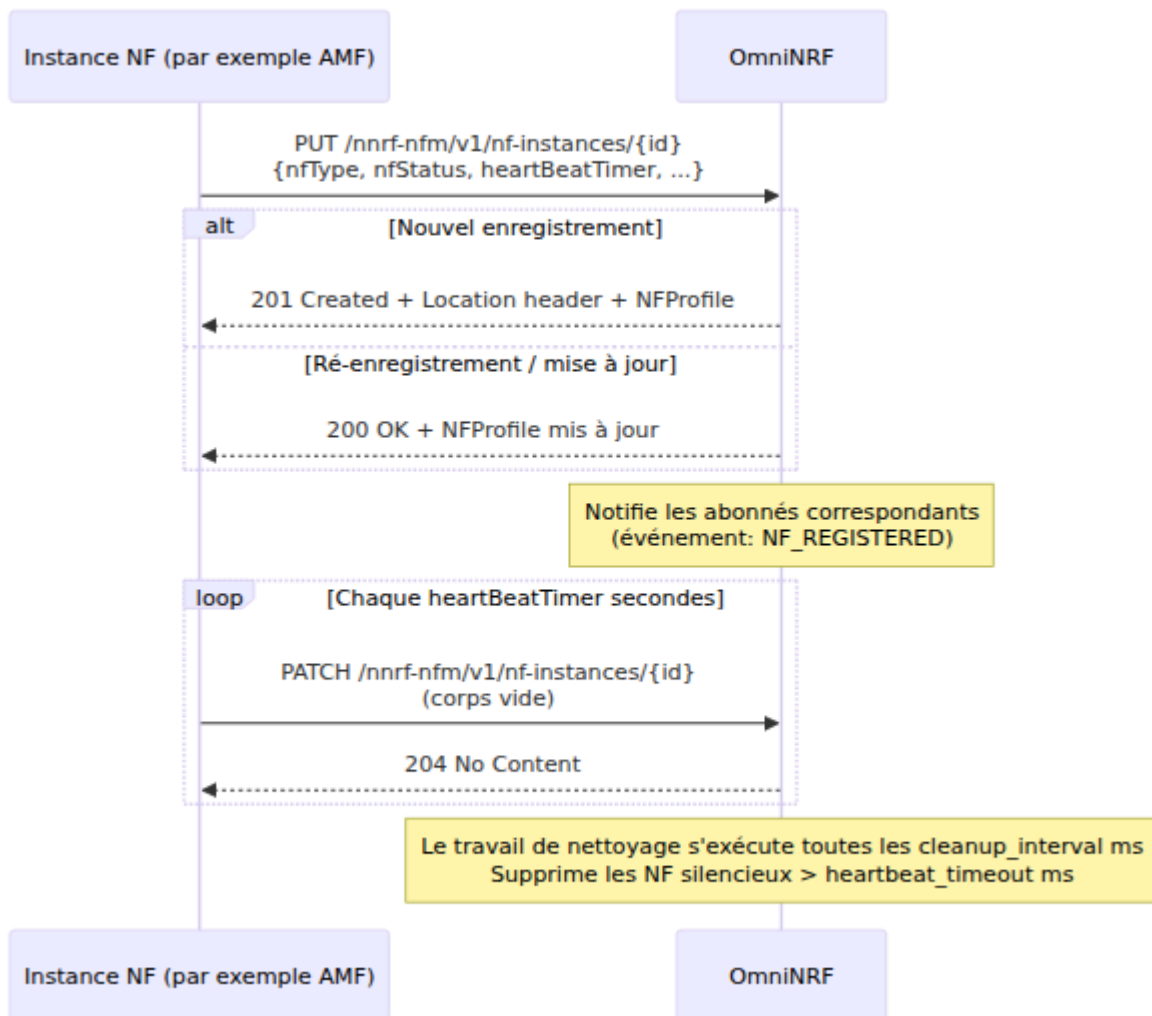
5. Procédures Clés

5.1 Enregistrement NF et Battement de Cœur



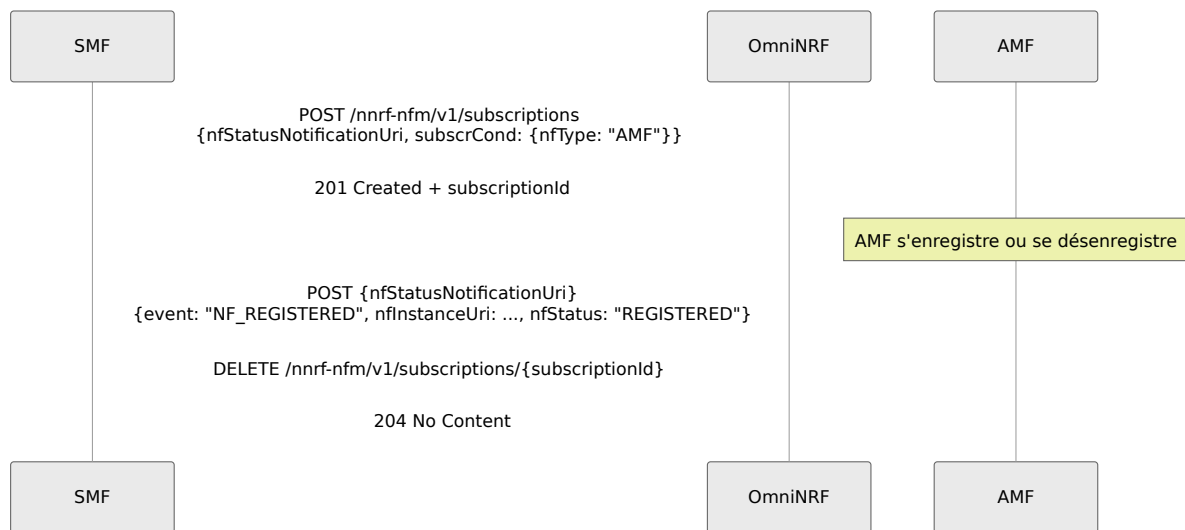
Après qu'un NF obsolète soit supprimé, le NRF envoie une notification **NF_DEREGISTERED** à tous les abonnés correspondants.

5.2 Découverte NF



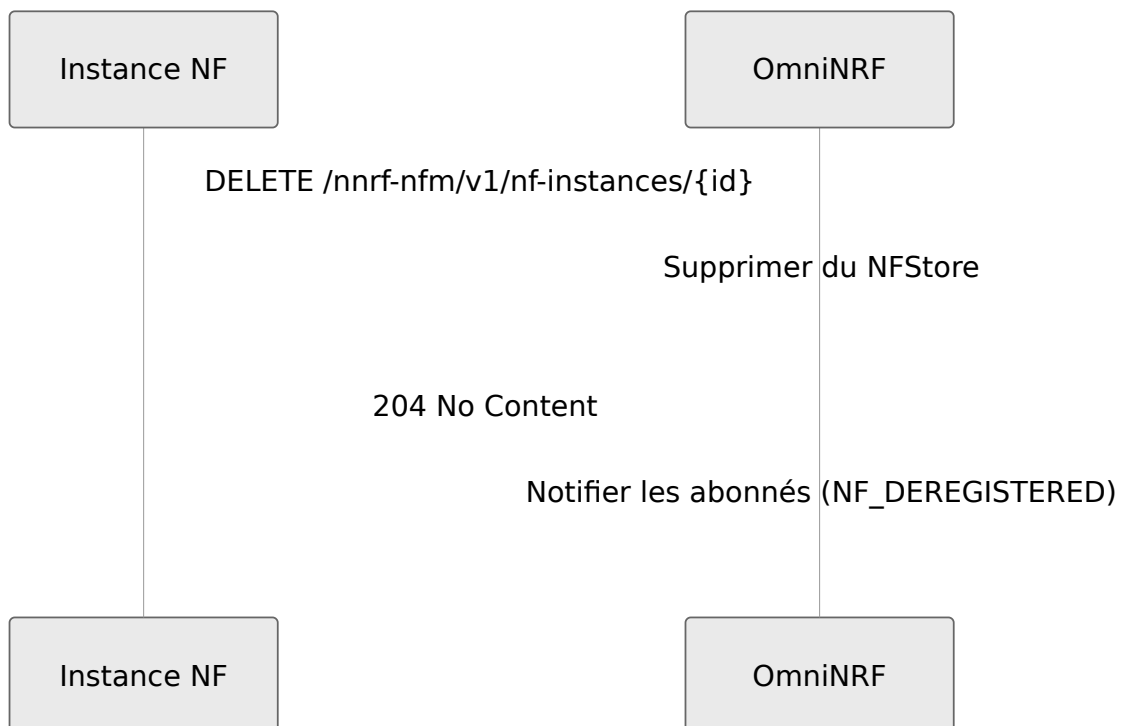
La `validityPeriod` retournée est codée en dur à `3600` secondes (voir NRF-L5 dans les Limitations Connues).

5.3 Abonnements d'État NF



Les notifications sont envoyées sans garantie de livraison (pas de réessai en cas d'échec de livraison). Les abonnements peuvent être filtrés par `subscrCond.nfType`.

5.4 Désenregistrement NF



6. Métriques Prometheus

Métriques NRF

Métrique	Type	Tags	
omni_nrf.nf.registered.count	gauge	nf_type	Nombre de NF par type
omni_nrf.discovery.requests.count	counter	--	Nombre de demandes de découverte
omni_nrf.registration.requests.count	counter	--	Nombre de demandes d'enregistrement
omni_nrf.nf_registrations.total	counter	nf_type	Total des enregistrements
omni_nrf.nf_deregistrations.total	counter	nf_type	Total des déregistrements
omni_nrf.discovery_requests.total	counter	target_nf_type	Total des demandes de découverte par cible
omni_nrf.heartbeats.total	counter	nf_type	Total des battements de cœur par type de NF
omni_nrf.active_nf_instances.count	gauge	nf_type	Nombre de instances de NF actives

Métriques BEAM VM

Métrique	Type	Description
<code>beam.memory.total</code>	gauge	Mémoire totale BEAM en octets
<code>beam.memory.processes</code>	gauge	Mémoire utilisée par les processus Erlang
<code>beam.memory.processes_used</code>	gauge	Mémoire réellement utilisée par les processus
<code>beam.memory.system</code>	gauge	Mémoire système
<code>beam.memory.atom</code>	gauge	Mémoire totale des atomes
<code>beam.memory.atom_used</code>	gauge	Mémoire des atomes utilisée
<code>beam.memory.binary</code>	gauge	Mémoire binaire
<code>beam.memory.code</code>	gauge	Mémoire de code
<code>beam.memory.ets</code>	gauge	Mémoire des tables ETS
<code>beam.processes.count</code>	gauge	Nombre de processus Erlang
<code>beam.ports.count</code>	gauge	Nombre de ports Erlang
<code>beam.atom.count</code>	gauge	Nombre d'atomes
<code>beam.vm.uptime</code>	gauge	Temps de fonctionnement de la VM en secondes

7. Limitations Connues

Ces limitations reflètent l'état actuel de l'implémentation et représentent des lacunes par rapport à la spécification complète TS 29.510.

ID	Domaine	Description
NRF-C2	Filtres de découverte	La découverte prend en charge uniquement <code>target-nf-type</code> , <code>requester-nf-type</code> , <code>service-names</code> , <code>plmn-id</code> , et <code>limit</code> . Les filtres pour <code>snssai</code> , <code>dnn</code> , <code>tai</code> , <code>guami</code> , <code>preferred-collocated-nf-types</code> , et d'autres définis dans la Section 6.2.3.2.3.1 de TS 29.510 ne sont pas implémentés
NRF-H2	Corps de notification	Les notifications de changement d'état manquent de l'objet complet <code>NfProfile</code> et du champ <code>subscriptionId</code> dans le corps selon la Section 5.2.2.8 de TS 29.510
NRF-H3	Expiration de l'abonnement	Les abonnements n'ont pas d'application de <code>validityTime</code> ou <code>reqPeriodicRegTimer</code> . Les abonnements n'expirent jamais
NRF-H4	Validation de NFProfile	Seul <code>nfType</code> est validé lors de l'enregistrement. Les champs tels que <code>nfInstanceId</code> , <code>nfStatus</code> , les listes PLMN et la capacité ne sont pas validés
NRF-H5	PATCH d'abonnement	Il n'y a pas de point de terminaison <code>PATCH /nnrf-nfm/v1/subscriptions/{subscriptionId}</code> pour mettre à jour un abonnement existant
NRF-M1	JSON Patch	Le gestionnaire PATCH applique uniquement des opérations à clé plate (<code>replace</code> , <code>add</code> , <code>remove</code>). Les chemins JSON Pointer imbriqués (par exemple <code>/nfServices/0/serviceStatus</code>) ne sont pas résolus correctement
NRF-M2	Concurrence optimiste	La gestion des requêtes conditionnelles ETag / If-Match / If-None-Match n'est pas implémentée

ID	Domaine	Description
NRF-M4	GET d'abonnement	Il n'y a pas de point de terminaison <code>GET /nnrf-nfm/v1/subscriptions/{subscriptionId}</code>
NRF-L5	Validité de la période de découverte	<code>validityPeriod</code> dans les réponses de découverte est codé en dur à <code>3600</code> secondes plutôt que dérivé des données de profil NF ou de la configuration NRF
NRF-L6	Pagination de découverte	Aucune pagination d'en-tête <code>Link</code> ou itération basée sur un curseur n'est prise en charge pour de grands ensembles de résultats de découverte

8. Dépannage

L'enregistrement NF échoue avec 400 Bad Request

Le corps de la requête manque de `nfType`. Tous les corps NFProfile doivent inclure au moins `nfType`. Vérifiez la charge utile de la requête PUT.

NF disparaît de l'enregistrement de manière inattendue

Le NF n'envoie pas de battements de cœur assez fréquemment. Le `heartbeat_timeout` par défaut est de 30 secondes. Assurez-vous que le NF enregistrant envoie un battement de cœur PATCH à un intervalle plus court que cette valeur. Le NRF définit `heartBeatTimer` dans la réponse d'enregistrement à `heartbeat_timeout / 1000` secondes comme indice.

La découverte retourne une liste `nfInstances` vide

- Vérifiez que le NF est enregistré : `GET /nnrf-nfm/v1/nf-instances/{id}` doit retourner un profil avec `nfStatus: "REGISTERED"`.
- Vérifiez que `target-nf-type` correspond exactement à `nfType` dans le NFProfile stocké (sensible à la casse).
- Si `plmn-id` est spécifié, vérifiez qu'il s'agit d'un JSON valide et qu'il correspond à `plmnList` dans le NFProfile, ou que le NFProfile n'a pas de `plmnList` (ce qui le fait correspondre à tous les PLMN).

Les notifications d'état ne sont pas reçues

- Confirmez que `nfStatusNotificationUri` dans le corps de l'abonnement est accessible depuis l'hôte NRF.
- La livraison de notification est sans garantie ; vérifiez les journaux d'application pour des avertissements `Failed to send status notification to`.
- Si `subscrCond.nfType` est défini, la notification n'est envoyée que lorsqu'un NF de ce type change d'état.

Forte rotation d'enregistrement ou messages de journalisation fréquents de nettoyage obsolète

Réduisez `cleanup_interval` et `heartbeat_timeout` si un nettoyage agressif est souhaité, ou augmentez `heartbeat_timeout` si des NFs sont expirés à tort. Les événements de nettoyage obsolète sont enregistrés au niveau `info` avec le message `NRF: Expiring stale NF instance {id}`.