← Vue d'ensemble

Référence de configuration OmniNSSF

La configuration de l'opérateur est lue à partir de la clé d'environnement de l'application `:omnissf`. La configuration d'exécution est fournie via `config/runtime.exs`, qui mappe les variables d'environnement du système d'exploitation aux clés de configuration au démarrage. La topologie de tranche (liste NSI, NSSAI autorisés, mappage AMF-set, NSSAI configurés) est fournie par la configuration structurelle de l'application (voir Configuration de la topologie de tranche) et ajustée à l'exécution via l'API OAM.

Configuration d'exécution (`config/runtime.exs`)

```
config :omnissf,
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr: System.get_env("SBI_ADDR", "127.0.0.14"),
  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"),
  prometheus_metrics_port:
String.to_integer(System.get_env("PROMETHEUS_PORT", "9568")),
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000"))

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

Paramètre	Type	Requis	Par défaut
`sbi_scheme`	chaîne	Non	`"http"`
`sbi_addr`	chaîne	Non	`"127.0.0.14"`
`sbi_port`	entier	Non	`7777`
`nrf_uri`	chaîne	Non	`"http://127.0.0.1:7777"`
`mcc`	chaîne	Non	`"999"`

Paramètre	Type	Requis	Par défaut
`mnc`	chaîne	Non	`"70"`
`prometheus_metrics_port`	entier	Non	`9568`
`heartbeat_interval`	entier (ms)	Non	`10000`

> Remarque : la valeur par défaut compilée pour `nrf_uri` dans la configuration d'exécution est `http://127.0.0.1:7777`, tandis que la valeur par défaut de l'accesseur interne est `http://127.0.0.10:7777`. Lorsque `runtime.exs` est appliqué, la valeur par défaut effective est `http://127.0.0.1:7777`. Définissez `NRF_URI` explicitement pour éviter toute ambiguïté.

Comportement et paramètres de notification

Les clés de configuration de l'application `:omnissf` suivantes ajustent le comportement de sélection de tranche et de notification. Aucune d'entre elles n'a de variable d'environnement `runtime.exs`, elles sont donc définies par la configuration de l'application plutôt que par une variable d'environnement du système d'exploitation.

Paramètre	Type	Requis	Par défaut	Desc
`dynamic_nrf_discovery`	Booléen	Non	`false`	Lorsque NSSF de NRF ser tranche l'exécut partir d racine v Nnrf_NF au lieu uniquer liste `ns` statique chaque de séle tranche interrog racine p NSSAI d et revie NRF de `nsi_lis` configu lorsque découv désacti inacces ne retou
`strict_availability`	Booléen	Non	`false`	Mode d'applic disponil Lorsque (permis TAI sans

Paramètre	Type	Requis	Par défaut	Desc
				enregist de disp explicit toujours autorisé configu Lorsque (fermé) TAI est le NSSA est reto
`nssf_notification_max_attempts`	Entier	Non	`3`	Nombre maximu tentativ livraison notifica change disponil avant q soit aba et que l soit enr
`nssf_notification_backoff_ms`	Liste d'entiers	Non	`[200, 1000]`	Horaire par tent millisec pour les notifica change disponil premièr est app avant la deuxièr

Paramètre	Type	Requis	Par défaut	Desc
				tentativ deuxièr la troisi dernière est réut pour tou tentativ

Journaliseur

Paramètre	Type	Requis	Par défaut	Var Env
`format`	tuple de formatteur	Non	`{OmniLogger.JsonFormatter, :format}`	-
`metadata`	atome / liste	Non	`:all`	-

Le niveau de journal actif peut être changé à l'exécution sans redémarrage via l'API OAM (`PATCH /oam/config/{id}` avec `log_level`, ou `POST /oam/log_level`). Les niveaux valides sont `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info`, et `debug`.

Configuration de la topologie de tranche

La topologie de tranche est chargée dans le magasin `SliceConfig` au démarrage à partir des clés de configuration de l'application `:omnissf` suivantes. Chacune a une valeur par défaut intégrée qui modélise une seule tranche eMBB (SST 1, SD `0x000001`). Ces clés sont structurelles (cartes et listes) et sont définies par la configuration de l'application plutôt que par une seule variable d'environnement ; elles peuvent également être modifiées à l'exécution via l'API OAM (voir Gestion des tranches à l'exécution).

```
config :omnissf,
  nsi_list: [
    %{s_nssai: %{sst: 1, sd: "0x000001"}, nrf_uri:
"http://127.0.0.10:7777", nsi_id: "1"}
  ],
  allowed_nssai: %{
    "999-70" => [%{sst: 1, sd: "0x000001"}]
  },
  amf_set_mapping: %{
    "1-0x000001" => ["1"]
  },
  configured_nssai: %{}
```

`nsi_list` - Entrées d'instance de tranche réseau

Chaque entrée lie un S-NSSAI au NRF qui sert la découverte NF spécifique à la tranche pour cette tranche. Le NSSF fait correspondre le S-NSSAI demandé contre cette liste ; une correspondance renvoie les informations NSI retournées à l'AMF, et aucune correspondance renvoie `403`.

Paramètre	Type	Requis	Par défaut	Var Env	
`nsi_list`	liste de cartes	Non	entrée eMBB unique (ci-dessous)	-	Ens d'e au
`s_nssai.sst`	entier	Oui	`1`	-	Typ de
`s_nssai.sd`	chaîne	Non	`"0x000001"`	-	Diff tra om cor que
`nrf_uri`	chaîne	Oui	`"http://127.0.0.10:7777"`	-	URI ser ret com `nsi`
`nsi_id`	chaîne	Non	`"1"`	-	Ide de ret `nsi` Om res

La correspondance SD est consciente des jokers : un SD `nil` sur l'entrée configurée ou la demande correspond à n'importe quel SD, et les SD non-nil sont comparés sans tenir compte de la casse.

`allowed_nssai` - NSSAI autorisés par PLMN

Mappe une clé PLMN `"mcc-mnc"` à la liste des S-NSSAIs autorisés pour les UE de ce PLMN. Lorsqu'une requête pour un PLMN n'a pas d'entrée, le NSSF revient au

PLMN home (`mcc-mnc` de la configuration d'exécution). La liste résultante est retournée comme `allowedNssaiList` (type d'accès `3GPP_ACCESS`) et, lorsqu'un TAI est fourni, est filtrée pour les S-NSSAIs signalés disponibles à ce TAI.

Paramètre	Type	Requis	Par défaut	Var Env	Description
`allowed_nssai`	carte	Non	`{"999-70" => [...]}`	-	Tableau NSSAI autorisés indexé par PLMN
clé de carte	chaîne `"mcc-mnc"`	Oui	`"999-70"`	-	Identifiant PLMN auquel l'ensemble autorisé s'applique
`sst`	entier	Oui	`1`	-	SST d'un S-NSSAI autorisé
`sd`	chaîne	Non	`"0x000001"`	-	SD d'un S-NSSAI autorisé

`amf_set_mapping` - Ensemble AMF cible par S-NSSAI

Mappe une clé S-NSSAI aux ensembles AMF capables de servir cette tranche. Lorsqu'il est présent, le premier ensemble AMF est retourné comme `targetAmfSet`, indiquant à l'AMF de rediriger s'il ne peut pas servir la tranche lui-même.

Paramètre	Type	Requis	Par défaut	Var Env	Description
`amf_set_mapping`	carte	Non	`{"1-0x000001" => ["1"]}`	-	Tableau d'ensemble-AMF cible indexé par S-NSSAI
clé de carte	chaîne `"sst"` ou `"sst-sd"`	Oui	`"1-0x000001"`	-	Clé S-NSSAI à laquelle le mappage s'applique
valeur	liste de chaînes	Oui	`["1"]`	-	Identifiants de l'ensemble AMF cible pour la tranche

> La correspondance des clés pour `amf_set_mapping` et `configured_nssai` est **exacte**, contrairement à `nsi_list`. La clé de recherche est construite à partir du S-NSSAI dans la requête : `"sst"` lorsque le S-NSSAI demandé n'a pas de SD, `"sst-sd"` lorsqu'il en a un. Une requête qui porte un SD ne correspond qu'à une clé `"sst-sd"` ; une clé `"sst"` nue ne la correspondra pas, et vice-versa. Il n'y a pas de jokers SD ici. Si vous servez une tranche avec un SD, utilisez la forme `"sst-sd"` (par exemple `"1-0x000001"`); si le S-NSSAI demandé n'a pas de SD, utilisez la forme nue `"sst"` (par exemple `"1"`).

`configured_nssai` - Mappage NSSAI configuré (itinérance)

Mappe une clé S-NSSAI à une valeur NSSAI configurée retournée à l'AMF pour des scénarios d'itinérance/mappage. Vide par défaut ; lorsqu'elle est définie, la

valeur est retournée telle quelle comme `configuredNssai`. Cela est utilisé pour donner à un UE en itinérance le mappage entre un S-NSSAI du PLMN servant (visité) et le S-NSSAI correspondant de son PLMN home, afin que l'UE puisse présenter un S-NSSAI home-PLMN lorsqu'il se déplace plus tard. Le NSSF n'interprète pas la valeur - il passe à travers n'importe quel objet que vous configurez - donc il doit être façonné exactement comme l'AMF s'y attend selon la TS 29.531.

Paramètre	Type	Requis	Par défaut	Var Env	Description
`configured_nssai`	carte	Non	`%{}` (vide)	-	Tableau NSSAI configurés indexé par S-NSSAI
clé de carte	chaîne `"sst"` ou `"sst-sd"`	Non	(aucun)	-	Clé S-NSSAI à laquelle le mappage configuré s'applique (correspondance exacte, voir note ci-dessus)
valeur	objet	Non	(aucun)	-	Valeur NSSAI configurée retournée telle quelle comme `configuredNssai`

Exemple - retourner un mappage S-NSSAI home-PLMN pour la tranche PLMN visitée `1-0x000001` :

```
config :omnissf,
  configured_nssai: %{
    "1-0x000001" => %{"sst" => 1, "sd" => "0x0000ff"}
  }
```

Laissez `configured_nssai` vide (`%{}`) pour les déploiements non-itinérants ; la réponse omet alors complètement `configuredNssai`.

Définir une tranche de bout en bout

Une seule tranche n'est pas décrite par une seule clé de configuration - les quatre clés ci-dessus sont des tableaux indépendants, et une réponse de sélection de tranche est assemblée à partir de celles qui contiennent une entrée correspondante. Pour rendre une nouvelle tranche entièrement utilisable, vous remplissez normalement deux, et éventuellement les quatre :

Clé	Requis pour que la tranche fonctionne ?	Ce qu'elle contrôle dans la réponse
`nsi_list`	**Oui**	Si le S-NSSAI se résout ou non. Une demande sans entrée NSI correspondante est rejetée avec `403 SNSSAI_NOT_FOUND`. Fournit `nsiInformation` (NRF cible + ID NSI optionnel)
`allowed_nssai`	**Oui**, par PLMN servant	Si l'UE est informée qu'elle peut utiliser la tranche. Sans une entrée PLMN correspondante (et sans retour au PLMN home), la réponse omet `allowedNssaiList`, donc l'AMF n'a pas de NSSAI autorisé à donner à l'UE
`amf_set_mapping`	Optionnel	Ajoute `targetAmfSet` afin que l'AMF puisse rediriger lorsqu'il ne peut pas servir la tranche lui-même. Omettre pour les déploiements à ensemble-AMF unique
`configured_nssai`	Optionnel (itinérance uniquement)	Ajoute `configuredNssai` pour le mappage S-NSSAI visité-à-home

Exemple pratique - ajouter la tranche SST `2` (URLLC), SD `0x000010`, servie par le NRF `http://10.0.0.20:7777`, autorisée pour le PLMN home `999-70`, redirigeable vers l'ensemble AMF `2` :

```
config :omnissf,
  nsi_list: [
    %{s_nssai: %{sst: 2, sd: "0x000010"}, nrf_uri:
"http://10.0.0.20:7777", nsi_id: "2"}
  ],
  allowed_nssai: %{
    "999-70" => [%{sst: 2, sd: "0x000010"}]
  },
  amf_set_mapping: %{
    "2-0x000010" => ["2"]
  }
```

Rappelez-vous l'asymétrie de correspondance SD : `nsi_list` et `allowed_nssai` comparent les SD sans tenir compte de la casse avec `nil` agissant comme un joker, tandis que `amf_set_mapping` et `configured_nssai` sont indexés par une chaîne exacte `"sst-sd"`. Gardez la forme SD cohérente (par exemple toujours `"0x000010"`) à travers les quatre clés afin que les clés s'alignent.

La même tranche peut également être ajoutée à l'exécution avec `POST /oam/slice` (qui écrit uniquement `nsi_list`) ; les tableaux NSSAI autorisés, AMF-set et NSSAI configurés sont uniquement de configuration et ne peuvent pas être modifiés via l'API OAM. Les ajouts NSI à l'exécution sont annulés au redémarrage.

Ports de gestion et de métriques

Les écouteurs suivants orientés opérateur sont démarrés en même temps que le point de terminaison SBI :

Écouteur	Par défaut	Var Env	Description
API de gestion / OAM	TCP `8443` (HTTPS)	-	API REST pour l'état, l'inspection des tranches, les statistiques et les actions OAM à l'exécution
Métriques Prometheus	TCP `9568`	`PROMETHEUS_PORT`	Point de terminaison de récupération Prometheus exposant les métriques NSSF et BEAM

API de gestion / OAM

Une API de gestion/OAM REST est servie sur HTTPS sur le port `8443` (lien `0.0.0.0`). Elle fournit un état en lecture seule et des statistiques ainsi que des contrôles OAM à l'exécution. Toutes les réponses sont en JSON, enveloppées dans une enveloppe standard `success` / `error`. Les chemins ci-dessous sont relatifs à la base de l'API.

État et inspection

Méthode	Chemin	Description
GET	`/status/api`	Vivacité du service API
GET	`/status/nrf`	État d'enregistrement du NRF
GET	`/status/nf`	État NF du NSSF
GET	`/slice_config`	Instantané de la configuration de tranche actuelle
GET	`/api/slices`	S-NSSAIs configurés avec des mappages NSI, ensembles AMF cibles, NSSAI autorisés par PLMN, et mappage d'ensemble AMF
GET	`/api/nssai_availability`	Disponibilité NSSAI par AMF actuellement stockée
GET	`/api/statistics`	Statistiques opérationnelles du NSSF
GET	`/oam/config`	Configuration d'exécution effective (adresse/port SBI, URI NRF, MCC/MNC, capacité/priorité NF, niveau de journal)

`GET /api/statistics` retourne un instantané des comptes utile pour un rapide contrôle de santé/capacité sans récupérer Prometheus :

Champ	Signification
`total_slices`	Nombre d'entrées NSI actuellement chargées (taille de `nsi_list`)
`total_allowed_nssai_plmns`	Nombre de PLMNs avec une entrée `allowed_nssai`
`total_amf_set_mappings`	Nombre de clés S-NSSAI dans `amf_set_mapping`
`nssai_availability_count`	Nombre d'AMFs ayant signalé la disponibilité par TAI
`nssai_availability_subscription_count`	Nombre d'abonnements actifs au changement de disponibilité

Gestion des tranches à l'exécution

Méthode	Chemin	Description
POST	`/oam/slice`	Ajouter ou mettre à jour un mappage NSI. Corps : `sst` (requis), `nrf_uri` (requis), `sd` optionnel, `nsi_id`
DELETE	`/oam/slice/{id}`	Supprimer un mappage NSI. `id` est la clé S-NSSAI `"sst"` ou `"sst-sd"`. Retourne `404` si aucun NSI correspondant n'existe
DELETE	`/oam/nssai_availability/{id}`	Invalider la disponibilité NSSAI stockée pour un AMF (`id` = `nfId`)

Les changements de tranche à l'exécution s'appliquent immédiatement aux requêtes de sélection de tranche suivantes mais ne sont pas persistés ; ils reviennent à la topologie configurée au redémarrage.

Contrôles de configuration à l'exécution

Méthode	Chemin	Description
PATCH	`/oam/config/{id}`	Mettre à jour la configuration d'exécution. Clés autorisées : `nrf_uri`, `nf_capacity`, `nf_priority`, `log_level`. Les clés/valeurs invalides retournent `400`
POST	`/oam/nrf/reregister`	Forcer la réinscription du NRF
POST	`/oam/log_level`	Changer le niveau de journal. Corps : `level` (un des `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info`, `debug`)

Pour `PATCH /oam/config/{id}`, `nf_capacity` et `nf_priority` doivent être des entiers et `log_level` doit être une chaîne de niveau valide ; sinon, la demande retourne `400`.

← Vue d'ensemble

Métriques OmniNSSF

Les métriques sont exposées au format Prometheus sur le port des métriques (par défaut `9568`, `PROMETHEUS_PORT`). Les métriques spécifiques à NSSF sont listées ci-dessous ; les métriques de la VM/mémoire BEAM (`beam.memory.*`, `beam.processes.count`, `beam.ports.count`, `beam.atom.count`, `beam.vm.uptime`) sont également exportées.

Métriques NSSF

Métrique	Type	É
`omni_nssf_nsselection_requests_count`	Compteur	`r` `n`
`omni_nssf_ns_selection_requests_total`	Compteur	`r`
`omni_nssf_nssai_availability_update_count`	Compteur	`n`
`omni_nssf_nssai_availability_delete_count`	Compteur	`n`
`omni_nssf_nssai_availability_subscribe_count`	Compteur	(a
`omni_nssf_nssai_availability_unsubscribe_count`	Compteur	(a
`omni_nssf_nssai_availability_updates_total`	Compteur	`n`
`omni_nssf_nssai_availability_patch_total`	Compteur	`n`
`omni_nssf_nssai_availability_patch_subscription_total`	Compteur	(a
`omni_nssf_nrf_registration_status`	Jauge	`n`

> Deux compteurs de sélection de tranche presque identiques existent et ne doivent pas être additionnés. `omni_nssf_nsselection_requests_count` utilise les valeurs `result` `success`/`error` et porte également `nf_type`, tandis que `omni_nssf_ns_selection_requests_total` utilise les valeurs `result` `success`/`failure` et n'a pas de `nf_type`. Choisissez-en un comme votre source de vérité pour éviter de compter deux fois les mêmes requêtes.

> `omni_nssf_nssai_availability_updates_total` (pluriel `updates`) et `omni_nssf_nssai_availability_update_count` (singulier) sont tous deux connectés au même événement de mise à jour PUT, donc ils suivent ensemble. L'un ou l'autre peut être utilisé pour le volume total de mise à jour de disponibilité.

Métriques BEAM VM

Métrique	Type	Description
`beam_memory_total`	Jauge	Mémoire totale allouée par la VM (octets)
`beam_memory_processes` / `beam_memory_processes_used`	Jauge	Mémoire allouée à / utilisée par les processus (octets)
`beam_memory_system`	Jauge	Mémoire allouée au système VM (octets)
`beam_memory_atom` / `beam_memory_atom_used`	Jauge	Mémoire allouée à / utilisée par la table des atomes (octets)
`beam_memory_binary`	Jauge	Mémoire allouée aux binaires (octets)
`beam_memory_code`	Jauge	Mémoire allouée au code chargé (octets)
`beam_memory_ets`	Jauge	Mémoire allouée aux tables ETS (octets)
`beam_processes_count`	Jauge	Nombre de processus actifs
`beam_ports_count`	Jauge	Nombre de ports ouverts
`beam_atom_count`	Jauge	Nombre d'atomes dans la table des atomes
`beam_vm_uptime`	Jauge	Temps de fonctionnement de la VM (secondes)

Exemple PromQL

Taux d'erreur de sélection de tranche sur 5 minutes :

```
sum(rate(omni_nssf_nsselection_requests_count{result="error"}
[5m]))
  / sum(rate(omni_nssf_nsselection_requests_count[5m]))
```

Taux de requêtes de sélection de tranche par type de NF demandé :

```
sum by (nf_type) (rate(omni_nssf_nsselection_requests_count[5m]))
```

Santé de l'enregistrement NRF (alerte lorsque non enregistré) :

```
min(omni_nssf_nrf_registration_status) < 1
```

Taux de mise à jour de disponibilité NSSAI par AMF :

```
sum by (nf_id)
(rate(omni_nssf_nssai_availability_update_count[5m]))
```

← Vue d'ensemble

Guide des opérations OmniNSSF

OmniNSSF implémente la fonction de sélection de tranche de réseau (NSSF) du cœur 5G. Il résout les requêtes de sélection de tranche de l'AMF en une réponse de sélection de tranche autorisée et collecte et autorise la disponibilité des tranches par zone de suivi rapportée par les AMF.

La topologie des tranches (mappings S-NSSAI vers NRF, NSSAI autorisés par PLMN, mappings de l'ensemble AMF et NSSAI configurés) est conservée dans un magasin `SliceConfig` en mémoire qui est chargé à partir de la configuration de l'application au démarrage et peut être ajusté à l'exécution via l'API de gestion OAM. La disponibilité NSSAI par AMF et les abonnements de disponibilité sont conservés dans un magasin `Context` en mémoire. Il s'agit d'un état d'exécution : il est reconstruit à partir de la configuration et des rapports AMF en direct après un redémarrage.

OmniNSSF s'enregistre auprès du NRF au démarrage et maintient son enregistrement avec des battements de cœur périodiques. Par défaut, les cibles NRF spécifiques aux tranches sont configurées par l'opérateur dans `nsi_list` et retournées à l'AMF en tant qu'informations NSI. La découverte dynamique du NRF est également disponible en tant que comportement facultatif : lorsque `dynamic_nrf_discovery` est activé, à chaque requête de sélection de tranche, le NSSF résout le NRF servant la tranche à l'exécution à partir du NRF racine via Nnrf_NFDiscovery, revenant à la `nsi_list` configurée statiquement lorsque la découverte est inaccessible ou ne donne rien. Cela est désactivé par défaut.

Rôle 3GPP et références de spécification

Spécification	Pertinence
TS 23.501 §5.15	Concepts de découpage de réseau - S-NSSAI, types de NSSAI (Demandé / Autorisé / Configuré / Abonné), Instance de tranche de réseau, rôle de NSSF
TS 23.502	Procédures - sélection de tranche lors de l'enregistrement et de l'établissement de session PDU
TS 29.531	APIs de services Nnssf_NSSelection et Nnssf_NSSAIAvailability (HTTP/2 SBI)
TS 23.003	Format S-NSSAI - SST (1 octet) et SD optionnel (3 octets)

Concepts S-NSSAI et NSSAI

Une **Single Network Slice Selection Assistance Information (S-NSSAI)** identifie une tranche de réseau. Selon le TS 23.003, elle se compose de :

- **SST (Type de tranche/service)** - obligatoire, une valeur de 8 bits. Les valeurs standardisées incluent `1` = eMBB, `2` = URLLC, `3` = MIoT, `4` = V2X.
- **SD (Différenciateur de tranche)** - optionnel, une valeur de 24 bits qui distingue plusieurs tranches du même SST. Écrit couramment sous forme de chaîne hexadécimale à 6 chiffres, comme `0x000001`. Lorsque le SD est absent, le S-NSSAI s'applique à tous les différenciateurs de ce SST.

Un **NSSAI** est un ensemble de S-NSSAIs. Le système 5G distingue plusieurs types de NSSAI, avec lesquels le NSSF travaille :

Type de NSSAI	Signification	Où OmniNSSF l'utilise
NSSAI demandé	S-NSSAIs que l'UE demande à utiliser, envoyés par l'AMF dans la requête de sélection de tranche	Entrée - extrait de la demande d'informations de tranche d'enregistrement ou de session PDU
NSSAI abonné	S-NSSAIs auxquels l'UE a droit selon son abonnement	Entrée - repli lorsque aucun NSSAI demandé n'est présent dans une requête d'enregistrement
NSSAI autorisé	S-NSSAIs que l'UE peut effectivement utiliser dans le PLMN/zone de service	Sortie - calculé à partir de l'ensemble autorisé par PLMN et filtré par la disponibilité TAI
NSSAI configuré	S-NSSAIs configurés pour l'UE dans un PLMN (utilisés pour l'itinérance/mapping)	Sortie - retourné lorsqu'un mapping NSSAI configuré est défini pour le S-NSSAI

Le NSSF résout une requête comme suit : il fait correspondre le S-NSSAI demandé avec la liste **Network Slice Instance (NSI)** configurée pour trouver le NRF spécifique à la tranche, calcule le NSSAI autorisé pour le PLMN de service (filtré optionnellement par la disponibilité TAI rapportée) et ajoute un ensemble AMF cible et un mapping NSSAI configuré lorsque ceux-ci sont configurés pour la tranche.

Points de terminaison SBI

Tous les points de terminaison SBI sont servis via l'URL de base `{sbi_scheme}://{sbi_addr}:{sbi_port}` (par défaut `http://127.0.0.14:7777`).

Nnssf_NSSelection (TS 29.531)

Méthode	Chemin	Description
GET	`/nnssf-nsselection/v2/network-slice-information`	Résoudre la sélection de tranche pour l'AMF. Retourne un objet `AuthorizedNetworkSliceInfo`

Paramètres de requête obligatoires : `nf-type`, `nf-id`. Un paramètre obligatoire manquant retourne `400` avec la cause `MANDATORY_IE_MISSING`. L'entrée de tranche est fournie via l'un des `slice-info-request-for-pdu-session`, `slice-info-request-for-registration`, ou un paramètre direct `snssai`. Les paramètres optionnels incluent `tai`, `home-plmn-id`, et `supported-features`.

Codes de réponse :

Code	Signification
`200`	`AuthorizedNetworkSliceInfo` (informations NSI, et si applicable NSSAI autorisé, ensemble AMF cible, NSSAI configuré)
`400`	Paramètre obligatoire manquant ou aucune information de tranche fournie
`403`	Aucune NSI configurée pour le S-NSSAI demandé (cause `SNSSAI_NOT_FOUND`)

Nnssf_NSSAIAvailability (TS 29.531)

Méthode	Chemin	D
PUT	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	Stocker la list charge par l'A `AuthorizedN`
PATCH	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	Mettre à jour disponibilité s JSON Patch (R `application/`
DELETE	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	Supprimer la pour l'AMF. Re
POST	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions`	Créer un abo changement Retourne `201` `Location`
PATCH	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions/{subscriptionId}`	Mettre à jour utilisant JSON
DELETE	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions/{subscriptionId}`	Supprimer ur Retourne `204`

Un corps de disponibilité PUT/PATCH doit contenir un tableau `supportedNssaiAvailabilityData` non vide, chaque entrée portant un objet `tai` et une `supportedSnssaiList` non vide. Un PATCH sur la disponibilité stockée (ou sur un abonnement) qui référence un `{nfId}` / `{subscriptionId}` inconnu retourne `404`. Un corps d'abonnement doit contenir `nfNssaiAvailabilityUri`, une `taiList` non vide, et un `event`. Tout chemin SBI non apparié retourne `404`.

Forme des données de disponibilité et correspondance TAI

Chaque entrée `supportedNssaiAvailabilityData` a deux parties requises :

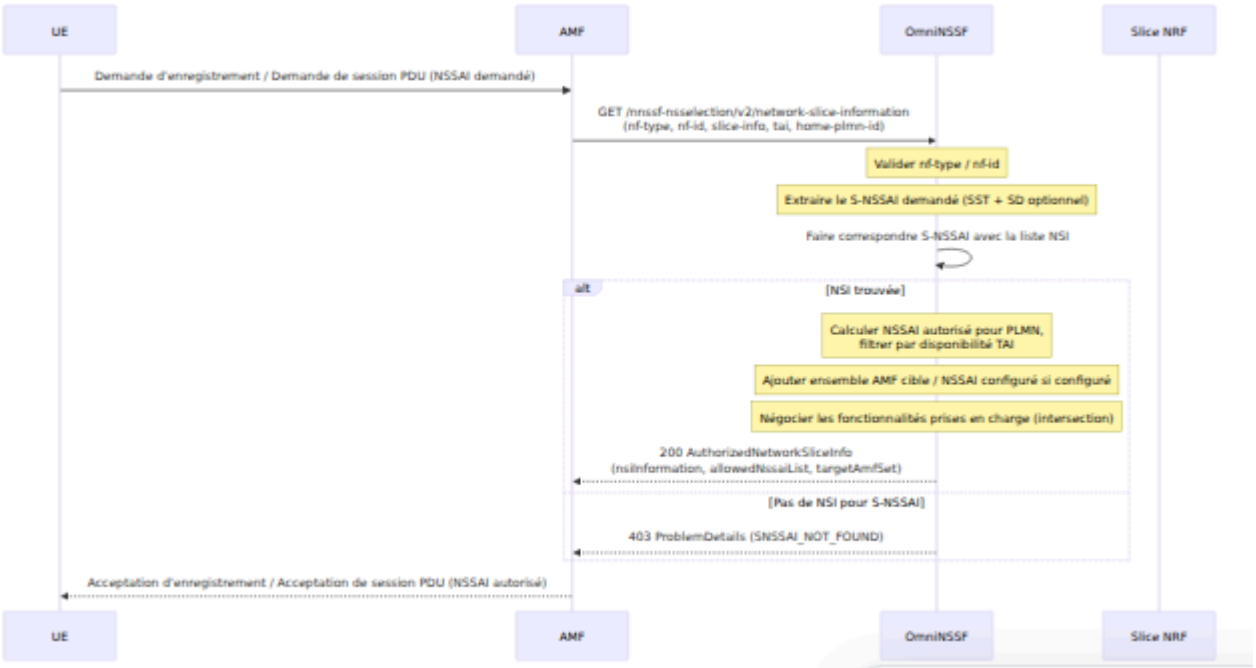
Champ	Forme	Signification
`tai`	`{ "plmnId": { "mcc", "mnc" }, "tac" }`	La zone de suivi à laquelle la disponibilité s'applique
`supportedSnssaiList`	tableau non vide de `{ "sst", "sd"? }`	S-NSSAIs que l'AMF rapportant prend en charge dans ce TAI

Lorsque la sélection de tranche filtre ultérieurement un NSSAI autorisé par un TAI de requête (voir Sélection de tranche de réseau), la correspondance est **exacte sur les trois `mcc`, `mnc`, et `tac`**, et un S-NSSAI est considéré comme disponible uniquement si une entrée rapportée pour ce TAI le liste (SD comparé exactement, ou correspondant à tout SD lorsque la requête omet le SD). Une tranche peut donc apparaître comme indisponible simplement parce que l'AMF a rapporté un TAI dont le `tac` ou le PLMN diffère de celui de la requête de sélection - assurez-vous que l'AMF rapporte le même TAI qu'il encode dans les requêtes ultérieures.

Procédures clés

Sélection de tranche de réseau (Nnssf_NSSelection)

Interrogé par l'AMF lors de l'enregistrement de l'UE et lors de l'établissement de session PDU. Le NSSF résout le S-NSSAI demandé en une réponse de sélection de tranche autorisée.

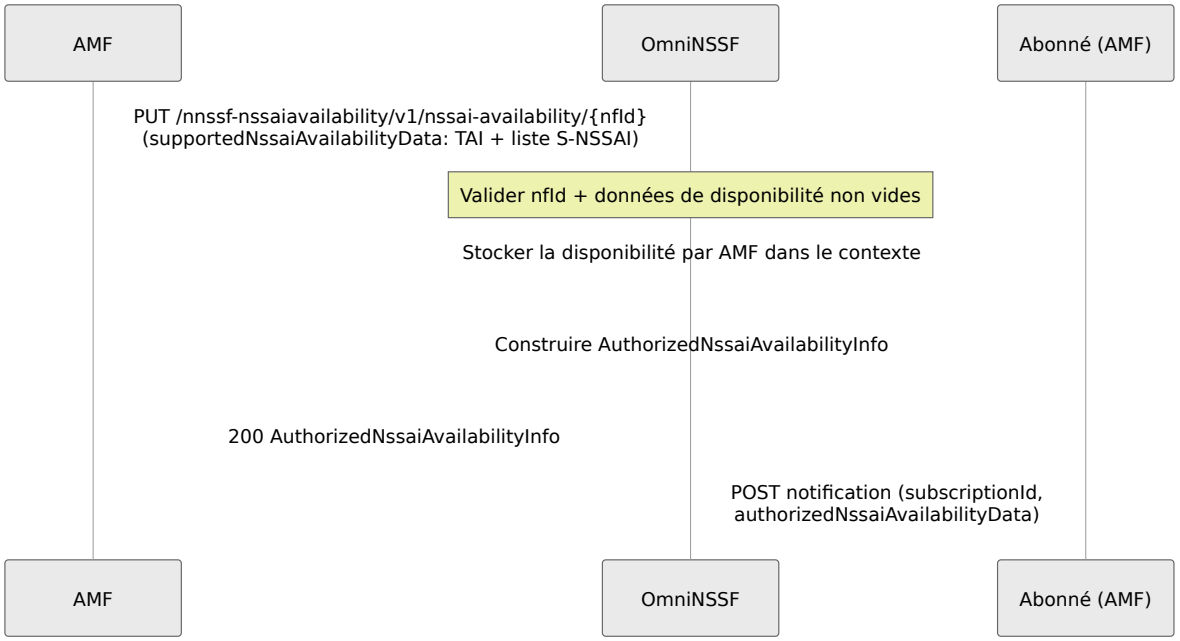


Lorsqu'un TAI est fourni dans la requête et qu'un ou plusieurs AMF ont rapporté la disponibilité, le NSSF vérifie que le S-NSSAI demandé est disponible à ce TAI et filtre le NSSAI autorisé en conséquence. Si aucune disponibilité n'a encore été rapportée, toutes les tranches configurées sont considérées comme disponibles (par défaut permissif ; définissez `strict_availability: true` pour échouer en mode fermé). Lorsque l'AMF fournit une valeur `supported-features`, le NSSF retourne l'intersection négociée par rapport à son propre ensemble de fonctionnalités (actuellement le bit NssaiAvailability).

Le NRF cible retourné en tant que `nsiInformation` provient normalement de l'entrée `nsi_list` correspondante. Lorsque `dynamic_nrf_discovery` est activé, le NSSF résout plutôt le NRF servant la tranche à l'exécution en interrogeant le NRF racine via Nnrf_NFDiscovery pour le S-NSSAI demandé, et revient au NRF de la `nsi_list` configurée lorsque la découverte est désactivée, inaccessible ou ne retourne rien. Voir la Référence de configuration pour la clé de configuration.

Mise à jour de la disponibilité NSSAI (Nnssf_NSSAIAvailability)

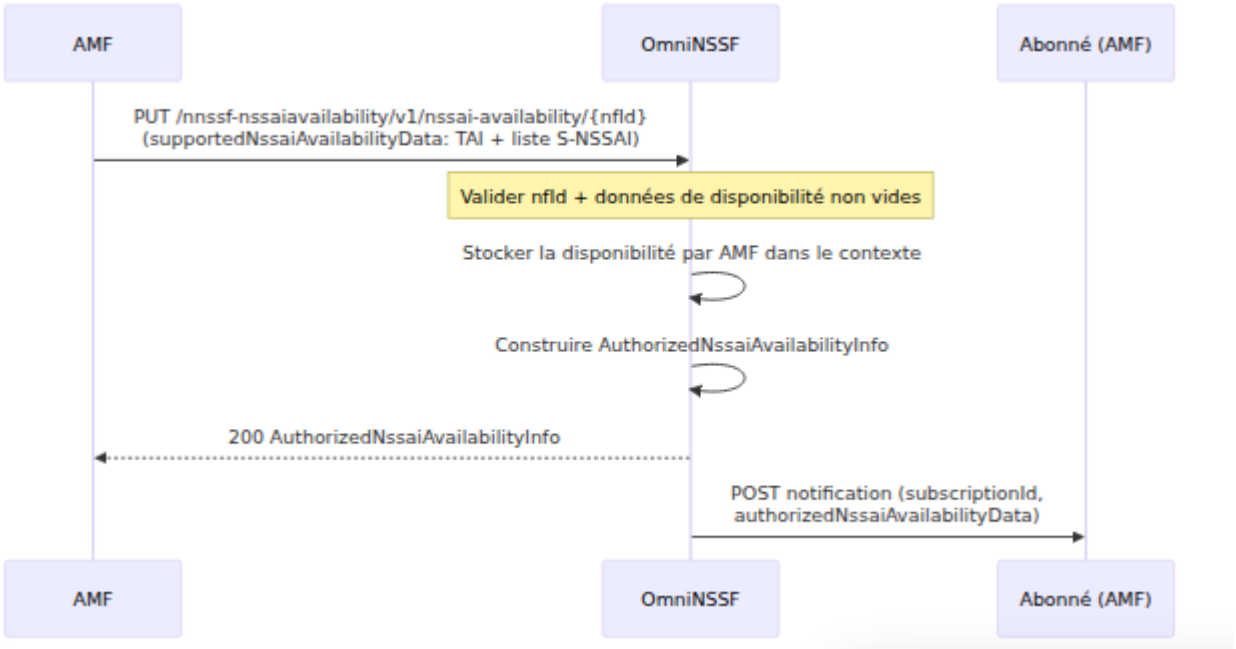
Les AMF poussent les S-NSSAIs qu'ils prennent en charge par TAI. Le NSSF stocke et autorise cette information et notifie tous les abonnés du changement.



Un PATCH sur la disponibilité stockée applique les opérations JSON Patch (RFC 6902) à l'enregistrement par AMF existant ; un PATCH contre un `{nfId}` inconnu retourne `404`. Les PUT et PATCH notifient à nouveau les abonnés avec les données de disponibilité autorisées.

Abonnement et notification de disponibilité NSSAI

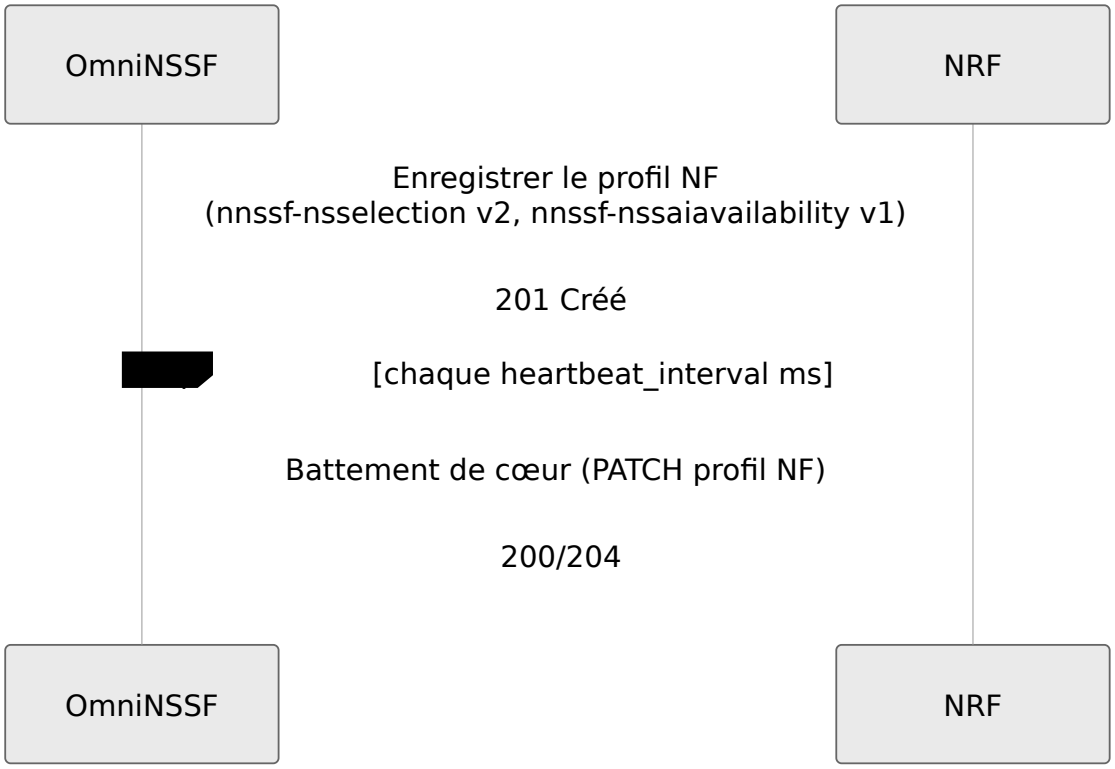
Les abonnés enregistrent une URI de rappel ; le NSSF envoie une notification de changement à chaque abonné chaque fois qu'un PUT ou PATCH met à jour la disponibilité stockée.



Les notifications sont des POST HTTP à l'`nfNssaiAvailabilityUri` de chaque abonné ; la notification contient le `subscriptionId` et les `authorizedNssaiAvailabilityData`. Chaque notification est livrée hors du chemin de demande avec des tentatives limitées : jusqu'à `nssf_notification_max_attempts` tentatives (par défaut `3`) utilisant le calendrier de retour `nssf_notification_backoff_ms` (par défaut `200` ms puis `1000` ms), après quoi la notification est abandonnée et l'échec est enregistré. Comme la livraison se fait hors du chemin de demande, un abonné lent ou inaccessible ne bloque pas la mise à jour de disponibilité qui a déclenché la notification.

Enregistrement et battement de cœur du NRF

Au démarrage, le NSSF enregistre son profil NF auprès du NRF, annonçant les deux services SBI, puis se réenregistre à intervalles réguliers.



Le statut d'enregistrement est reflété dans le compteur `omni_nssf_nrf_registration_status` et peut être forcé à se réenregistrer via l'API OAM (`POST /oam/nrf/reregister`).

← Vue d'ensemble

Dépannage OmniNSSF

Problèmes courants

Symptôme	Cause probable	
La sélection de tranche renvoie `400` `MANDATORY_IE_MISSING`	La requête AMF manque `nf-type` ou `nf-id`	Confirmer que deux paramèt obligatoires
La sélection de tranche renvoie `400` "Aucune info de tranche fournie"	La requête n'a aucune des `slice-info-request-for-pdu-session`, `slice-info-request-for-registration`, ou `snssai`	Vérifier que l'A slice-info pour
La sélection de tranche renvoie `403` `SNSSAI_NOT_FOUND`	Le S-NSSAI demandé n'a aucune entrée NSI correspondante	Ajouter le S-N (config) ou via
La réponse omet `allowedNssaiList`	Aucune entrée `allowed_nssai` pour le PLMN de la requête et aucune correspondance de secours home-PLMN	Ajouter le PLM `mnc`) à `allowe`
La tranche semble indisponible à un TAI	Un rapport de disponibilité AMF conflictuelle existe pour ce TAI sans le S-NSSAI	Inspecter `GET` `/api/nssai_av` invalider les d avec `DELETE` `/oam/nssai_av`
Pas de `targetAmfSet` dans la réponse	Aucune entrée `amf_set_mapping` pour la clé S-NSSAI	Ajouter la clé `amf_set_mapp`
Statut d'enregistrement NRF `0`	NRF injoignable ou mauvais `nrf_uri`	Vérifier `NRF_U` réinscription a `/oam/nrf/rere`

Symptôme	Cause probable	
Rejet de l'Availability PUT `400`	Données `supportedNssaiAvailabilityData` manquantes/vides, ou une entrée manque `tai` ou une `supportedSnssaiList` non vide	Corriger la for disponibilité d
L'Availability PATCH renvoie `404`	JSON Patch envoyé pour un `nfId` sans disponibilité stockée	PUT la disponi cibler le bon `n`
Rejet de la création de souscription `400`	Corps manquant `nfNssaiAvailabilityUri`, `taiList`, ou `event`	Fournir les tro souscription re
L'Subscription PATCH/DELETE renvoie `404`	`subscriptionId` inconnu	Confirmer le `s` partir de l'en-t d'origine
Abonnés non notifiés	Callback `nfNssaiAvailabilityUri` injoignable	Les notificatio avec un backo (`nssf_notifi` tentatives, pa d'être abando joignabilité de journaux NSSF l'avertissemer