← Vue d'ensemble

Référence de configuration OmniBSF

La configuration est lue à partir de la clé d'environnement de l'application `:omnibsf`. Le mécanisme de déploiement standard est `config/runtime.exs`, qui associe chaque valeur d'une variable d'environnement du système d'exploitation avec une valeur par défaut intégrée.

```
config :omnibsf,
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr:   System.get_env("SBI_ADDR", "127.0.0.15"),
  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", "9571")),
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000"))

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

Tableau des paramètres

Paramètre	Type	Requis	Par défaut
`sbi_scheme`	chaîne	Non	`"http"`
`sbi_addr`	chaîne	Non	`"127.0.0.15"`
`sbi_port`	entier	Non	`7777`
`nrf_uri`	chaîne	Non	`"http://127.0.0.1:7777"`

Paramètre	Type	Requis	Par défaut
`mcc`	chaîne	Non	`"999"`
`mnc`	chaîne	Non	`"70"`
`prometheus_metrics_port`	entier	Non	`9571`
`heartbeat_interval`	entier (ms)	Non	`10000`

> Remarque : la valeur par défaut `runtime.exs` pour `nrf_uri` est `http://127.0.0.1:7777`. Lorsque la variable d'environnement n'est pas définie et que l'application revient à la valeur par défaut de l'accessoire compilé, la valeur est `http://127.0.0.10:7777`. Définissez `NRF_URI` explicitement en production pour éviter toute ambiguïté.

Journalisation

Le journaliseur est configuré pour émettre des JSON structurés via `OmniLogger.JsonFormatter` avec des métadonnées complètes. Le niveau de journal actif peut être changé à l'exécution via l'API de gestion sans redémarrage. Voir API de gestion pour les points de terminaison de niveau de journal à l'exécution.

Réglage de la durée de liaison

Le TTL de liaison et l'intervalle de nettoyage sont des valeurs par défaut internes et ne sont pas exposés en tant que variables d'environnement dans `runtime.exs`. Ils peuvent être définis via l'environnement de l'application si un déploiement nécessite des valeurs non par défaut :

Clé d'environnement de l'application	Type	Par défaut	Description
`:binding_ttl_seconds`	entier (s)	`86400` (24 h)	Durée de vie appliquée à une liaison lors de l'enregistrement et rafraîchie à chaque PATCH ; après cela, l'entrée est éligible pour le nettoyage.
`:binding_cleanup_interval_ms`	entier (ms)	`60000` (60 s)	À quelle fréquence le nettoyeur en arrière-plan scanne et supprime les liaisons expirées.

Le TTL est estampillé lors de l'enregistrement et **rafraîchi à chaque PATCH**. Un merge-patch `UpdateIndPCFBinding` (PATCH) recalcule l'expiration de la liaison à partir de `:binding_ttl_seconds` au moment du patch, donc un PATCH agit comme un keep-alive qui prolonge la durée de vie de la liaison. Un inscrit qui touche périodiquement sa liaison (par exemple, lors d'une modification de session PDU) maintient cette liaison active tant que la session est active. Une liaison qui n'est jamais patchée vit au maximum `:binding_ttl_seconds` depuis son enregistrement. Réglez le TTL confortablement au-dessus de l'intervalle entre les touches de keep-alive (ou au-dessus de la durée de vie de session PDU la plus longue attendue lorsqu'aucun keep-alive n'est émis) ; sinon, la liaison d'une session encore active est balayée en milieu de session et les requêtes de découverte ultérieures renvoient `204 No Content` jusqu'à ce que l'inscrit se réenregistre.

L'intervalle de nettoyage détermine combien de temps une liaison expirée persiste avant suppression - une entrée peut survivre jusqu'à un intervalle complet après son expiration. Un intervalle plus court récupère de la mémoire et efface les liaisons obsolètes plus rapidement au prix de scans complets plus fréquents de la table de liaison. Les suppressions sont enregistrées

(`[BindingCleaner] Removed N expired binding(s)`); voir Dépannage pour la croissance des liaisons obsolètes.

Capacité et priorité du profil NRF

Le profil NF qu'OmniBSF enregistre auprès du NRF porte une `capacité` et une `priorité`. Les consommateurs qui découvrent le BSF via le NRF utilisent ces informations pour la sélection et la distribution de charge selon TS 29.510 : parmi les instances candidates, une `priorité` plus basse est préférée, et la `capacité` pèse la part de trafic envoyée aux instances de même priorité. Ils ne limitent pas OmniBSF lui-même - ce sont des indices annoncés pour la logique de sélection du consommateur et ne comptent que lorsque plus d'une instance BSF est enregistrée.

Clé d'environnement de l'application	Type	Par défaut	Description
`:nf_capacity`	entier	`100`	Poids de capacité relative annoncé dans le profil NF (plage typique 0–65535). Des valeurs plus élevées attirent une plus grande part de trafic parmi les instances de même priorité.
`:nf_priority`	entier	`0`	Priorité de sélection annoncée dans le profil NF. Des valeurs plus basses sont préférées par les consommateurs.

Les deux peuvent être modifiés à l'exécution via l'API de gestion, qui renvoie le profil mis à jour au NRF lors du prochain cycle de mise à jour ; voir API de gestion. Les modifications à l'exécution ne sont pas persistées et reviennent à ces valeurs au redémarrage.

Ports de gestion et de métriques

En plus du listener SBI, OmniBSF expose :

- Un point de terminaison de récupération Prometheus sur `prometheus_metrics_port` (par défaut `9571`). Voir Métriques & Surveillance.
- Une API de gestion des opérateurs (OAM) pour visualiser des statistiques, inspecter/effacer des liaisons, ajuster la capacité/priorité du profil NRF et le niveau de journal à l'exécution, et forcer la réinscription au NRF. Voir API de gestion pour la référence complète des points de terminaison.

← Vue d'ensemble

API de gestion OmniBSF

L'API de gestion (OAM) est une interface REST destinée aux opérateurs, distincte du SBI 5G. Elle expose des contrôles de statut, de statistiques, d'inspection/ suppression de liaison et de configuration à l'exécution. Elle est servie sur HTTPS.

Retour au Guide des opérations.

Accès

Propriété	Valeur
Schéma	HTTPS (TLS)
Port	`8443`
Adresse de liaison	`0.0.0.0`

Toutes les réponses sont en JSON, enveloppées dans une enveloppe standard avec un statut `success` / `error`.

Points de terminaison de statut

Méthode	Chemin	Description
`GET`	`/status/api`	Sonde de vivacité. Retourne `"ok"`.
`GET`	`/status/nrf`	Statut d'enregistrement NRF : `nrf_status`, `nf_instance_id`, `nrf_uri`. Retourne `unknown` si le processus d'enregistrement est indisponible.
`GET`	`/status/nf`	Statut du service NF : `nf_type`, service `nbsf-management` annoncé, `sbi_addr`/`sbi_port`, et le nombre de liaisons PCF actives en cours.

Inspection de liaison

Méthode	Chemin	Description
`GET`	`/pcf_bindings`	Liste des liaisons PCF actives (vue résumé : ID de liaison, IP UE, DNN, S-NSSAI, SUPI, FQDN PCF).
`GET`	`/api/bindings`	Liste de toutes les liaisons avec adresse UE, DNN, S-NSSAI, SUPI, GPSI et informations sur l'adresse PCF, plus un total.
`GET`	`/api/bindings/{id}`	Vue détaillée d'une seule liaison par ID, y compris les listes de routes encadrées et `suppFeat` négocié. Retourne 404 si inconnu.
`GET`	`/api/statistics`	Statistiques agrégées : total des liaisons et répartition par DNN.

Gestion des liaisons

Méthode	Chemin	Description
`DELETE`	`/oam/binding/{id}`	Supprimer une seule liaison PCF par ID. Retourne 404 si la liaison n'existe pas.
`POST`	`/oam/binding`	Effacer toutes les liaisons PCF. Retourne le nombre de liaisons effacées. Cela est perturbant - les sessions actives perdent leur liaison jusqu'à ce qu'elles soient réenregistrées.

Configuration à l'exécution

Méthode	Chemin	Description
`GET`	`/oam/config`	Voir la configuration actuelle à l'exécution : `sbi_addr`, `sbi_port`, `nrf_uri`, `mcc`, `mnc`, capacité/ priorité NF, et niveau de journalisation.
`PATCH`	`/oam/config/{id}`	Mettre à jour les valeurs à l'exécution sélectionnées. Clés acceptées : `nrf_uri`, `nf_capacity`, `nf_priority`, `log_level`. Les mises à jour invalides ou vides retournent 400.
`POST`	`/oam/nrf/reregister`	Forcer le réenregistrement NF avec le NRF. Retourne le statut d'enregistrement résultant et l'ID d'instance NF.
`POST`	`/oam/log_level`	Changer le niveau de journalisation à l'exécution. Corps : `{"level": "<level>"}`.

Valeurs configurables

Clé	Type	Remarques
`nrf_uri`	chaîne	URI de base du NRF. Prend effet lors du prochain cycle d'enregistrement/heartbeat.
`nf_capacity`	entier	Capacité NF annoncée dans le profil NRF ; poussée vers le NRF lors de la mise à jour.
`nf_priority`	entier	Priorité NF annoncée dans le profil NRF ; poussée vers le NRF lors de la mise à jour.
`log_level`	chaîne	Un des `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info`, `debug`.

Les modifications apportées via cette API ne sont appliquées qu'à l'instance en cours d'exécution et ne sont pas persistées ; elles reviennent aux valeurs configurées au redémarrage.

← Vue d'ensemble

Métriques et Surveillance d'OmniBSF

Les métriques sont exposées sur `prometheus_metrics_port` (par défaut `9571`).

Métriques BSF

Métrique	Type	Étiquettes	D
`omni_bsf_pcf_binding_register_count`	compteur	`dnn`	Enre de lia étiqu DNN
`omni_bsf_pcf_binding_discover_count`	compteur	`result`	Requ déco étiqu `resu` `not_`
`omni_bsf_pcf_binding_deregister_count`	compteur	`binding_id`	Désir étiqu l'ID d
`omni_bsf_binding_registers_total`	compteur	-	Com d'enr agré
`omni_bsf_binding_discovers_total`	compteur	`result`	Com déco agré `resu`
`omni_bsf_binding_deregisters_total`	compteur	-	Com désir agré
`omni_bsf_active_bindings_count`	jauge	-	Nom liaisc activ direc par l'

Métrique	Type	Étiquettes	D
			de té [:om :bin :act lors d l'enr de li désir de ch balay netto expir liaiso suit l actue activ
`omni_bsf_nrf_registration_status`	jauge	`nf_type`	Statu d'enr NRF enre non)

> Le compteur `omni_bsf_pcf_binding_deregister_count` est étiqueté par `binding_id`. Comme les ID de liaison sont des UUID uniques, cela produit une série temporelle par liaison et a une haute cardinalité ; préférez `omni_bsf_binding_deregisters_total` pour le taux d'agrégation/alerte.

> La jauge `omni_bsf_active_bindings_count` est une jauge en direct alimentée par l'événement de télémétrie `[:omnibsf, :bindings, :active]`, que le BSF émet chaque fois que l'ensemble des liaisons actives change : lors de l'enregistrement de liaison, lors de la désinscription et à chaque balayage de BindingCleaner qui supprime les liaisons expirées. Elle reflète donc le nombre actuel de liaisons actives et est sûre pour les alertes et les tableaux de bord. La même valeur est également disponible via le

> point de terminaison des statistiques OAM (`GET /api/statistics`) et le point de terminaison de statut NF ; voir API de gestion.

Événements uniquement de télémétrie

OmniBSF émet les événements internes `:telemetry` suivants qui ne sont **pas** actuellement exportés en tant que compteurs Prometheus. Ils sont disponibles pour tout gestionnaire de télémétrie attaché mais n'apparaissent pas sur le point de terminaison de récupération des métriques.

Événement	Émis	Description
`[:omnibsf, :pcf_binding, :update]`	Une fois par PATCH (`UpdateIndPCFBinding`)	Déclenché lorsqu'une liaison est mise à jour par merge-patch. Transporte l'`binding_id` affecté.
`[:omnibsf, :binding, :expired]`	Une fois par balayage de nettoyeur qui supprime des entrées	Déclenché par le nettoyeur en arrière-plan avec le nombre de liaisons balayées pour avoir dépassé leur TTL.

Métriques de la VM BEAM

Métrique	Type	Description
`beam_memory_total`	jauge	Mémoire totale BEAM en 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
`beam_memory_atom`	jauge	Mémoire totale des atomes
`beam_memory_atom_used`	jauge	Mémoire d'atome 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 en secondes

Exemple PromQL

Taux d'enregistrement des liaisons (par seconde, par DNN) :

```
sum by (dnn) (rate(omni_bsf_pcf_binding_register_count[5m]))
```

Ratio de non-découverte sur les 5 dernières minutes :

```
sum(rate(omni_bsf_binding_discovers_total{result="not_found"}
[5m]))
  /
sum(rate(omni_bsf_binding_discovers_total[5m]))
```

Alerte lorsque le BSF n'est pas enregistré avec le NRF :

```
omni_bsf_nrf_registration_status{nf_type="bsf"} == 0
```

> Les noms de métriques ci-dessus utilisent la normalisation des points en traits de soulignement de Prometheus. Les valeurs d'étiquettes pour `result` sont rapportées comme `found` / `not_found`.

← Vue d'ensemble

Guide des opérations OmniBSF

Vue d'ensemble

OmniBSF implémente la Fonction de Support de Liaison (BSF) du cœur 5G. Le BSF maintient un registre des liaisons PCF : pour chaque session PDU, il stocke la correspondance entre une identité UE (adresse IPv4, préfixe IPv6, adresse MAC ou SUPI) ainsi que le DNN et le S-NSSAI, et l'instance PCF qui sert la politique de cette session. D'autres fonctions réseau - un AF/NEF, un SMF ou un autre PCF - interrogent le BSF pour découvrir quel PCF est responsable d'une session donnée, afin que tous les signaux de politique pour cette session soient dirigés vers la même instance PCF.

OmniBSF expose le service `Nbsf_Management` (3GPP TS 29.521) sous le chemin de base SBI `/nbsf-management/v1`. Les liaisons sont entièrement conservées en mémoire dans des tables ETS et sont indexées pour une recherche concurrente à faible latence. Les liaisons portent une durée de vie configurable (TTL) ; un nettoyeur en arrière-plan supprime périodiquement les entrées expirées. OmniBSF enregistre son profil NF auprès du NRF au démarrage et envoie des signaux de vie périodiques.

Stockage de données interne

Les liaisons sont conservées dans six tables ETS à l'exécution : la table de liaison principale plus cinq indices de recherche dédiés. Toutes les données sont en mémoire et ne survivent pas à un redémarrage du processus.

Table ETS	Type	Clé	Valeur
`bsf_bindings`	`:set`	binding_id (chaîne UUID)	carte de liaison complète
`bsf_ipv4_index`	`:set`	adresse IPv4 UE (chaîne)	binding_id
`bsf_ipv6_index`	`:set`	préfixe IPv6 UE (chaîne)	binding_id
`bsf_supi_index`	`:bag`	SUPI (chaîne)	binding_id
`bsf_mac_index`	`:set`	adresse MAC UE (chaîne en minuscules)	binding_id
`bsf_dnn_snssai_index`	`:bag`	`{dnn, sst}`	binding_id

Chaque clé de recherche est servie par son propre index, de sorte que la découverte ne parcourt pas la table de liaison principale. Les recherches `macAddr48` se résolvent via `bsf_mac_index` (les clés sont en minuscules, donc la correspondance est insensible à la casse), et les recherches DNN+S-NSSAI se résolvent via `bsf_dnn_snssai_index` indexées par `{dnn, sst}` (le petit seau correspondant est ensuite filtré sur le S-NSSAI `sd` optionnel).

`bsf_supi_index` est un `:bag` : un seul SUPI peut avoir **plusieurs liaisons concurrentes** (par exemple, plusieurs sessions PDU sur différents DNN), donc un SUPI peut correspondre à plusieurs IDs de liaison. Une découverte par `supi` renvoie une seule `PcfBinding` par TS 29.521, donc la première liaison correspondante est renvoyée.

Références de rôle et de spécification 3GPP

Spécification	Pertinence
TS 23.501	Architecture système - rôle de la fonction réseau BSF
TS 23.503	Cadre de contrôle de politique et de facturation - contexte de liaison de session pour l'association de politique
TS 29.521	API du service `Nbsf_Management` (enregistrement / découverte / mise à jour / désenregistrement de liaison PCF), modèle de données `PcfBinding`, négociation de fonctionnalités prises en charge
TS 29.500	Cadre commun SBI 5G (ProblemDetails, sémantique HTTP)
TS 29.510	Enregistrement NF et signal de vie avec le NRF

Le BSF devient pertinent lorsque le PCF est déployé en tant que plusieurs instances (ou en tant qu'ensemble PCF). Son rôle est la liaison de session : enregistrer quel PCF sert chaque session PDU afin que les interactions de politique ultérieures (d'un AF via le NEF, ou d'un SMF) atteignent le bon PCF.

Points de terminaison SBI

Tous les points de terminaison sont servis sous l'URL de base `{sbi_scheme}://{sbi_addr}:{sbi_port}` avec le chemin de base `/nbsf-management/v1`.

Méthode	Chemin	Description	Su
`POST`	`/pcfBindings`	Enregistrer une liaison PCF pour une session PDU. Renvoie un en-tête `Location` pointant vers la liaison créée.	201 Cr
`GET`	`/pcfBindings`	Découvrir la liaison PCF correspondant aux paramètres de requête fournis.	200 Ok `PcfBin` 204 Pa conten aucune corresp
`PATCH`	`/pcfBindings/{bindingId}`	Mettre à jour une liaison existante (`UpdateIndPCFBinding`). Le corps doit être `application/merge-patch+json`.	200 Ok `PcfBin` mis à j
`DELETE`	`/pcfBindings/{bindingId}`	Désenregistrer une liaison PCF par son ID.	204 Pa conten

Tout autre chemin renvoie `404` avec un corps `ProblemDetails`.

Paramètres de requête GET

Le point de terminaison de découverte résout une liaison en utilisant le premier paramètre correspondant, dans cet ordre de priorité :

Priorité	Paramètre	Type	Recherche	Description
1	`ipv4Addr`	chaîne	index	Adresse IPv4 UE (correspondance exacte)
2	`ipv6Prefix`	chaîne	index	Préfixe IPv6 UE (correspondance exacte)
3	`macAddr48`	chaîne	index	Adresse MAC UE (correspondance insensible à la casse via `bsf_mac_index`)
4	`supi`	chaîne	index	Identifiant Permanent d'Abonné (première correspondance si plusieurs liaisons concurrentes partagent le SUPI)
5	`dnn` + `snssai`	chaîne / JSON	index	DNN avec S-NSSAI, résolu via `bsf_dnn_snssai_index` indexé par `{dnn, sst}` (`sst` doit correspondre ; `sd` correspond lorsque fourni)

Lorsqu'ils sont fournis, les paramètres de requête `pcfSetId`, `pcfRegionId` et `bindLevel` affinent davantage le résultat selon TS 29.521 : un candidat dont la valeur stockée ne correspond pas à un filtre fourni n'est pas renvoyé. Les filtres absents de la requête sont ignorés.

Corps de requête POST - Champs

Champ	Type	Requis	Description
`dnn`	chaîne	Oui	Nom de Réseau de Données
`snssai`	objet	Oui	S-NSSAI (`sst` entier, `sd` chaîne optionnelle)
`pcfFqdn`	chaîne	Conditionnel	FQDN PCF - requis si `pcfIpEndPoints` absent
`pcfIpEndPoints`	tableau	Conditionnel	Liste des points de terminaison IP PCF - requise si `pcfFqdn` absent
`ipv4Addr`	chaîne	Non	Adresse IPv4 UE (indexée ; déclenche également la détection des doublons)
`ipv6Prefix`	chaîne	Non	Préfixe IPv6 UE (indexé ; déclenche également la détection des doublons)
`macAddr48`	chaîne	Non	Adresse MAC UE
`supi`	chaîne	Non	Identifiant Permanent d'Abonné (indexé)
`gpsi`	chaîne	Non	Identifiant Public d'Abonnement Générique
`ipv4FrameRouteList`	tableau	Non	Routes encadrées IPv4
`ipv6FrameRouteList`	tableau	Non	Routes encadrées IPv6

Champ	Type	Requis	Description
`pcfSetId`	chaîne	Non	Identifiant de l'ensemble PCF. Stocké sur la liaison et utilisé pour affiner la découverte lorsqu'il est fourni comme filtre de requête.
`pcfRegionId`	chaîne	Non	Identifiant de la région PCF. Stocké sur la liaison et utilisé pour affiner la découverte lorsqu'il est fourni comme filtre de requête.
`bindLevel`	chaîne	Non	Niveau de liaison. Stocké sur la liaison et utilisé pour affiner la découverte lorsqu'il est fourni comme filtre de requête.
`suppFeat`	chaîne (hex)	Non	Bitmap des fonctionnalités prises en charge ; négocié via un ET logique avec la fonctionnalité `0x01` (BindingUpdate)

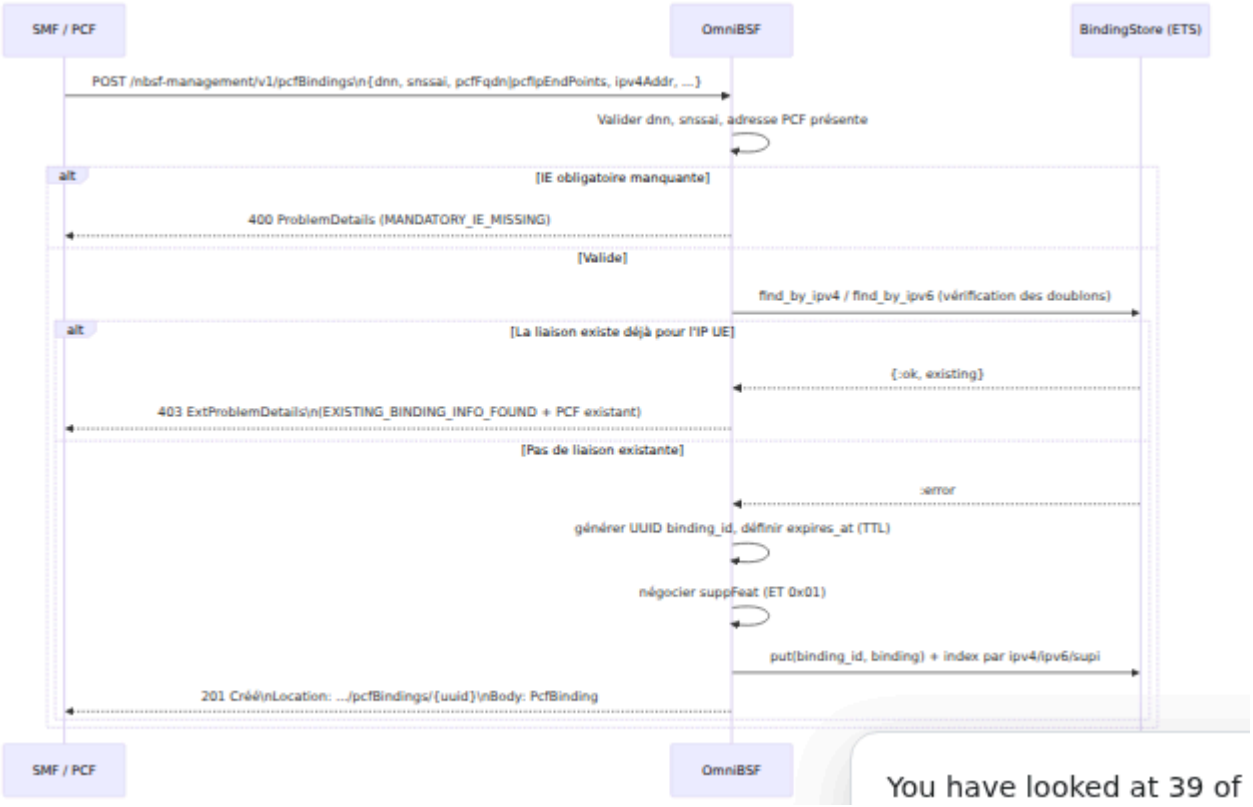
Une réponse `403` à un enregistrement en double contient un corps `ExtProblemDetails` (`cause` = `EXISTING_BINDING_INFO_FOUND`) avec le `pcfSmFqdn` / `pcfSmIpEndPoints` du PCF existant en ligne au niveau supérieur, conformément à TS 29.521.

API de gestion des opérateurs

En plus de l'écouteur SBI, OmniBSF expose une interface REST de gestion des opérateurs (OAM) pour visualiser des statistiques, inspecter/effacer des liaisons, ajuster la capacité/priorité du profil NRF et le niveau de journalisation à l'exécution, et forcer la réinscription du NRF. Voir API de gestion pour la référence complète des points de terminaison.

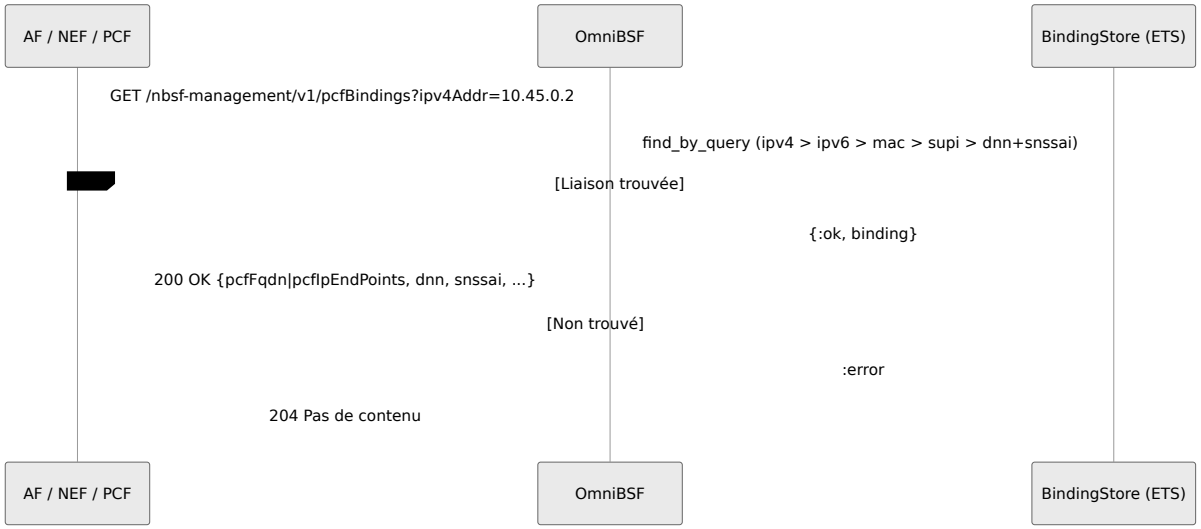
Procédures clés

Enregistrement de liaison PCF (POST)

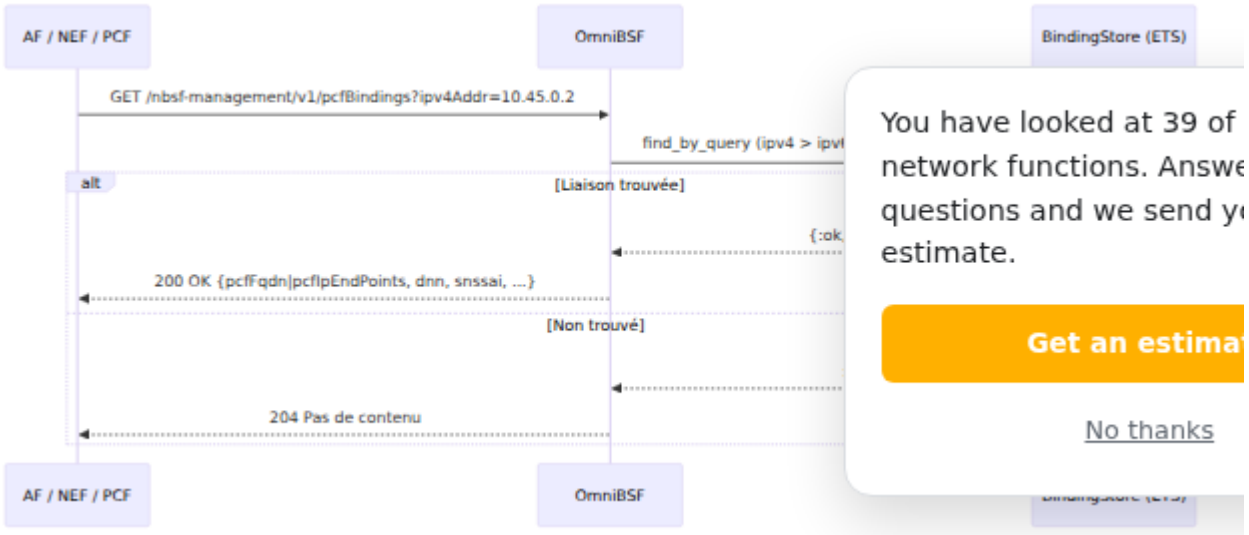


You have looked at 39 of

Découverte de liaison PCF (GET)

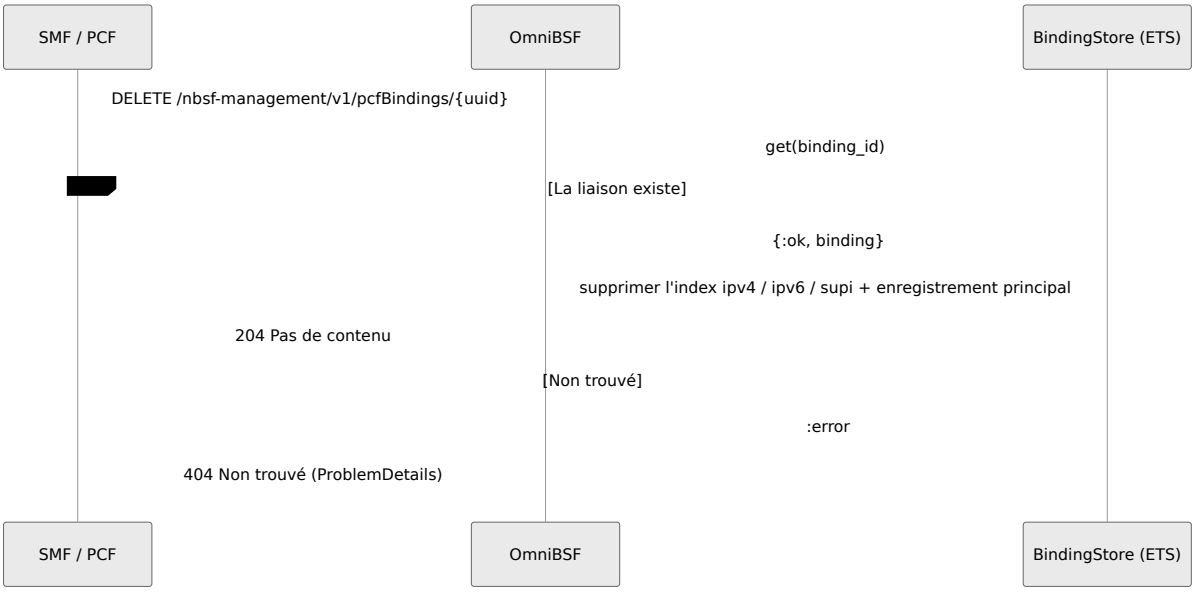


Mise à jour de liaison PCF (PATCH)



Le merge-patch s'applique uniquement aux champs `ipv4Addr`, `ipv6Prefix`, `macAddr48`, `pcfFqdn` et `pcfIpEndPoints` ; parce que les indices IPv4/IPv6/SUPI/MAC et DNN+S-NSSAI se basent sur le contenu de la liaison, l'ancien enregistrement et ses entrées d'index sont supprimés et la liaison mise à jour est réinsérée afin que les indices restent cohérents. Un PATCH rafraîchit également `expires_at` de la liaison à partir de `binding_ttl_seconds`, de sorte qu'il agit comme un keep-alive qui prolonge la durée de vie de la liaison. Voir Ajustement de la durée de vie de la liaison.

Désenregistrement de liaison PCF (DELETE)



Enregistrement NRF et signal de vie

Au démarrage, OmniBSF enregistre un profil NF BSF auprès du NRF (à partir de `nrf_uri`), en annonçant son adresse/port/schéma SBI, le PLMN configuré (`mcc`/`mnc`), et le service `nbsf-management` v1. Il envoie ensuite des signaux de vie NFUpdate périodiques toutes les `heartbeat_interval` millisecondes. L'enregistrement peut être forcé à tout moment via l'API de gestion.

← Vue d'ensemble

Dépannage OmniBSF

204 retourné lorsqu'un binding est attendu

1. Confirmez que la clé de requête correspond à ce qui a été enregistré. Les recherches `ipv4Addr`, `ipv6Prefix` et `supi` sont des correspondances exactes de chaînes - les différences de format (zéros non significatifs, notation de longueur de préfixe) entraînent un échec.
2. Vérifiez la présence de `PCF binding not found for query:` dans les journaux pour voir la requête exacte.
3. Confirmez que le registrant a complété le POST (recherchez `PCF binding registered:` dans les journaux).
4. La découverte DNN+S-NSSAI se résout via l'index `{dnn, sst}` et correspond exactement à `sst` ; un décalage de type entier contre chaîne dans le corps du POST original empêchera une correspondance.
5. Vérifiez si le TTL du binding a expiré et si le nettoyeur l'a supprimé (`[BindingCleaner] Removed N expired binding(s)`).

403 sur POST

Un binding existe déjà pour l'`ipv4Addr` / `ipv6Prefix` fourni. La réponse est un `ExtProblemDetails` avec `cause = EXISTING_BINDING_INFO_FOUND` et le `pcfSmFqdn` / `pcfSmIpEndPoints` du PCF en place. Réutilisez le PCF retourné, ou `DELETE` le binding existant d'abord si un véritable rebinding est prévu.

400 sur POST

Le corps manque d'un IE obligatoire. Le champ `detail` l'identifie : `No DNN`, `No S-NSSAI`, ou `No PCF address information (pcfFqdn or pcfIpEndPoints`

`required)`. Le chemin générique de champ manquant rapporte `MANDATORY_IE_MISSING`.

404 sur DELETE ou PATCH

L'ID de binding ne correspond à aucun binding actif. Soit il a déjà été supprimé, son TTL a expiré, ou OmniBSF a redémarré (tous les bindings sont en mémoire et perdus au redémarrage). Vérifiez la présence de `PCF binding not found for deregister:` / `PCF binding not found for update:`.

415 sur PATCH

Le gestionnaire `PATCH` nécessite `Content-Type: application/merge-patch+json`. Tout autre type de contenu est rejeté avec `415 Unsupported Media Type`.

Mémoire élevée / augmentation du nombre de bindings

Surveillez le nombre de bindings actifs en direct via le compteur `omni_bsf_active_bindings_count`, qui est émis lors de l'enregistrement, de la désinscription et de chaque balayage du nettoyeur ; le point de terminaison des statistiques OAM (`GET /api/statistics`) et le point de terminaison d'état NF rapportent le même chiffre. Les bindings dont les sessions PDU se terminent sans un `DELETE` restent jusqu'à ce que leur TTL expire et que le nettoyeur les supprime. Parce qu'un PATCH rafraîchit le TTL d'un binding, un registrant qui continue à patcher un binding le maintient en vie ; un binding ne vieillit que si ni un DELETE ni un PATCH n'ont été reçus dans les `:binding_ttl_seconds`. Raccourcissez `:binding_ttl_seconds` ou `:binding_cleanup_interval_ms` si des bindings obsolètes s'accumulent plus rapidement que le TTL par défaut de 24 heures ne les efface. L'API de gestion propose également une opération de suppression de tout.

Échec de l'enregistrement NRF

Vérifiez `omni_bsf_nrf_registration_status{nf_type="bsf"}` et les journaux. Vérifiez que `nrf_uri` est accessible, que `mcc`/`mnc` correspondent au PLMN configuré au NRF, et que la connectivité de `sbi_addr` au NRF fonctionne. Une réinscription peut être forcée via l'API de gestion.