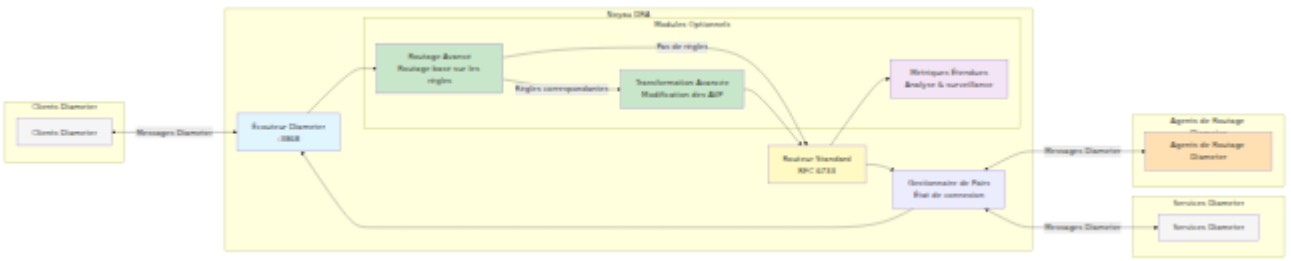
Guide des Opérations DRA

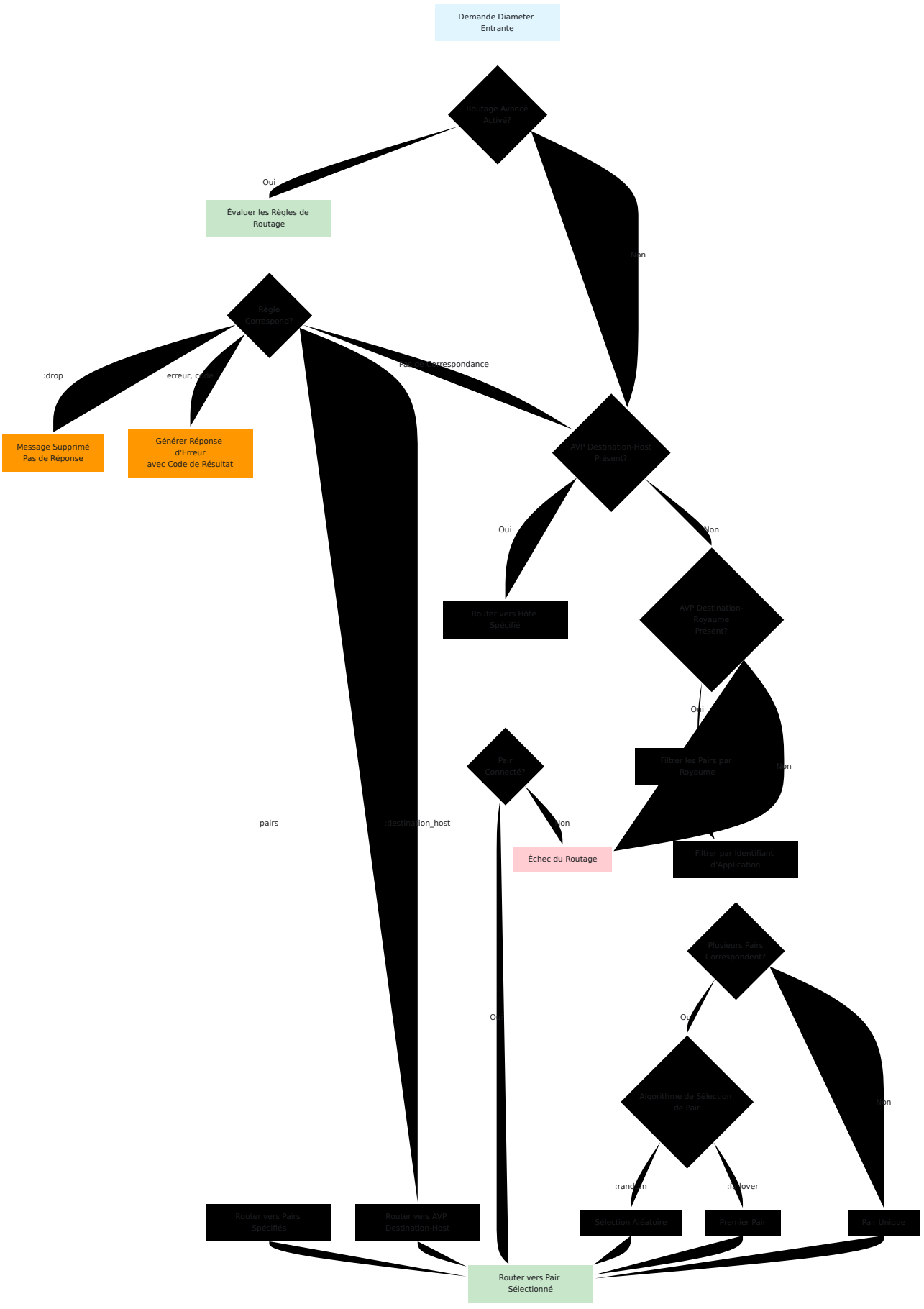
Table des Matières

Vue d'Ensemble de l'Architecture DRA



Routage Diameter Standard

Sans les modules Routage Avancé ou Transformation Avancée, le DRA effectue un routage Diameter standard basé sur le Protocole de Base Diameter (RFC 6733):


Demande Diameter Entrante
Routage Avancé Activé?
Oui
Évaluer les Règles de Routage
Non
Règle Correspond?
:drop
erreur, code
Paire Correspondance
Message Supprimé Pas de Réponse
Générer Réponse d'Erreur avec Code de Résultat
AVP Destination-Host Présent?
Oui
Non
Router vers Hôte Spécifié
AVP Destination-Royaume Présent?
Oui
Non
Filtrer les Pairs par Royaume
Pair Connecté?
pairs
:destination_host
Non
Échec du Routage
Filtrer par Identifiant d'Application
Plusieurs Pairs Correspondent?
Oui
Oui
Algorithme de Sélection de Pair
Non
:random
:failover
Router vers Pairs Spécifiés
Router vers AVP Destination-Host
Sélection Aléatoire
Premier Pair
Pair Unique
Router vers Pair Sélectionné

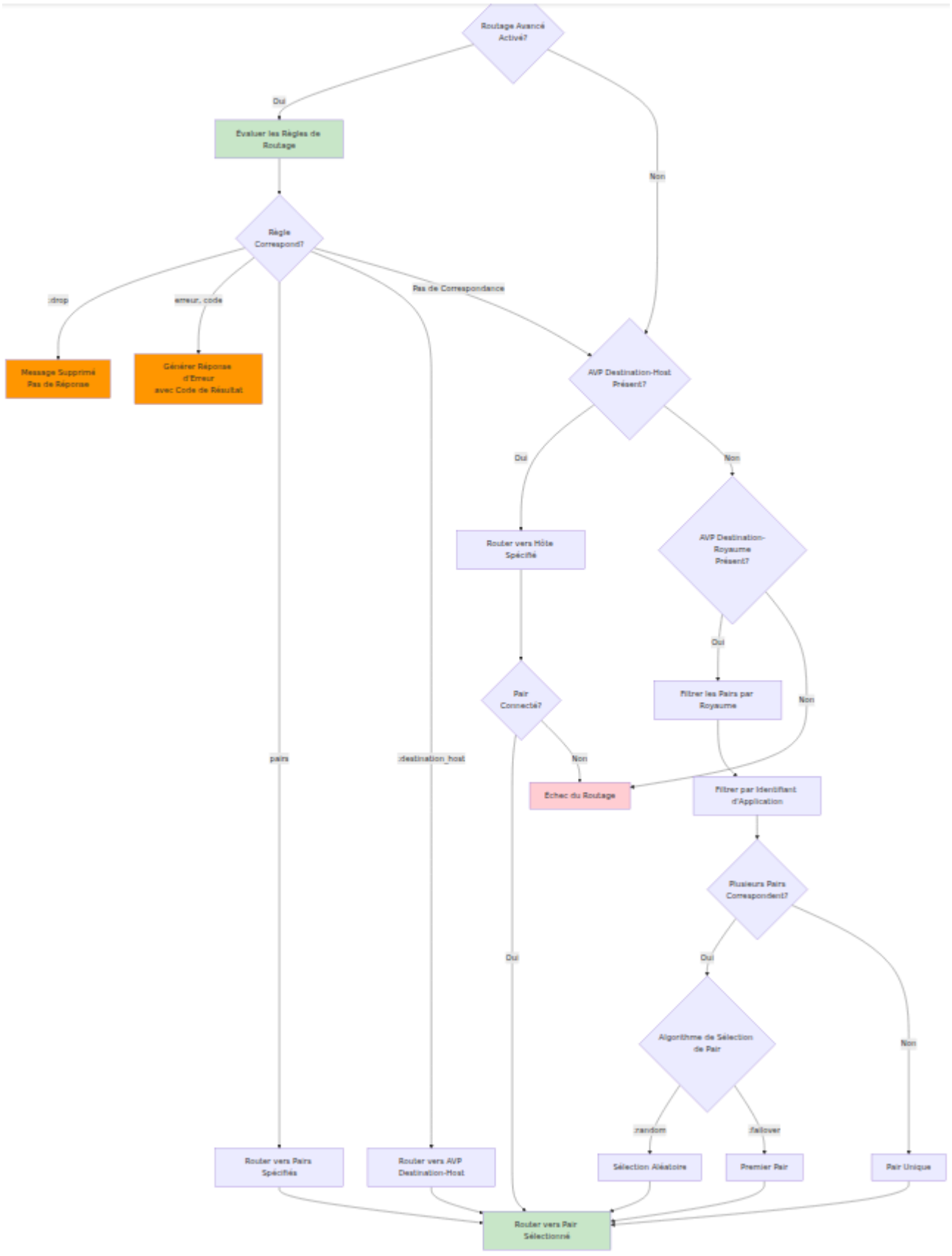
Routage des Demandes

Le DRA route les messages de demande en utilisant un mécanisme basé sur la priorité défini dans RFC 6733 Section 6.1:

1. **AVP Destination-Host (293)** - S'il est présent, le DRA route directement vers le pair spécifié
    - C'est le mécanisme de routage de la plus haute priorité
    - Si le pair n'est pas connecté, le routage échoue
    - Fournit un contrôle explicite du routage au niveau de l'hôte
2. **AVP Destination-Royaume (283)** - Si Destination-Host est absent, route en fonction du royaume
    - Le DRA sélectionne un pair connecté qui annonce le support pour le royaume cible
    - L'équilibrage de charge est appliqué lorsque plusieurs pairs correspondent au royaume
    - Le routage basé sur le royaume permet une flexibilité entre plusieurs hôtes
3. **Identifiant d'Application** - Les pairs sont filtrés par les applications Diameter prises en charge
    - Seuls les pairs annonçant le support pour l'Identifiant d'Application du message sont considérés
    - Basé sur l'Échange de Capacités (CER/CEA) lors de l'établissement de la connexion avec le pair
    - Voir Identifiants d'Application 3GPP Communs pour référence

Routage des Réponses

Les paquets de réponse utilisent un mécanisme de routage fondamentalement différent de celui des demandes :



- **Routage basé sur la session**: Les paquets de réponse suivent toujours le chemin inverse de la demande

- **Préservation de l'ID de bout en bout**: L'Identifiant de bout en bout reste inchangé à travers tous les sauts
- **Routage par saut**: Le DRA utilise l'Identifiant par saut pour maintenir l'état de routage (changements à chaque saut)
- **Aucune évaluation de règle**: Le DRA n'évalue pas les règles de routage ou le contenu des AVP pour les réponses
- **Corrélation avec état**: Les tables de routage internes suivent quel pair a envoyé chaque demande

Pourquoi les réponses ne sont pas routées par des modules avancés :

- Le routage des réponses est déterministe et doit revenir au pair d'origine
- Le protocole Diameter exige que les réponses suivent le chemin de demande établi
- Les décisions de routage pour les réponses sont prises en fonction du contexte de la demande originale, pas du contenu de la réponse
- Cela garantit une gestion correcte des sessions et empêche les boucles de routage

Voir RFC 6733 Section 6.2 pour les détails sur le routage des messages de réponse.

Sélection de Pair

Lorsque plusieurs pairs correspondent aux critères de routage, l'algorithme de `peer_selection_algorithm` configuré détermine la sélection :

- `:random` - Sélectionne aléatoirement parmi les pairs disponibles (par défaut)
- `:failover` - Sélectionne toujours le premier pair de la liste (basé sur la priorité)
- Les pairs doivent être en **état connecté** pour être sélectionnés
- Les pairs déconnectés ou hors service sont automatiquement exclus

Limitations du Routage Standard

- Pas de règles de routage personnalisées basées sur les valeurs AVP (par exemple, motifs IMSI)
- Pas de traduction de royaume ou de modification d'AVP
- Ne peut pas router en fonction du pair d'origine
- Contrôle limité sur la distribution du trafic

Les modules Routage Avancé et Transformation Avancée étendent ce comportement standard avec des capacités de routage basées sur des règles et de manipulation de paquets.

Configuration de Base du DRA

Le DRA nécessite une configuration de base définissant son identité, ses paramètres réseau et ses connexions de pair. Cette configuration établit la base pour toutes les opérations de routage.

Structure de Configuration

```
%{
  host: "dra01.example.com",
  realm: "example.com",
  listen_ip: "192.168.1.10",
  listen_port: 3868,
  service_name: :example_dra,
  product_name: "OmniDRA",
  vendor_id: 10415,
  request_timeout: 5000,
  peer_selection_algorithm: :random,
  allow_undefined_peers_to_connect: false,
  log_unauthorized_peer_connection_attempts: true,
  peers: [
    # Configurations des pairs...
  ]
}
```

Paramètres d'Identité DRA

Paramètre	Type	Description
`host`	Chaîne	L'Identité Diameter du DRA (nom de domaine entièrement qualifié)
`realm`	Chaîne	Le royaume Diameter du DRA
`product_name`	Chaîne	Nom du produit annoncé dans les messages CER/CEA
`vendor_id`	Entier	ID du fournisseur tel que défini dans RFC 6733 Section 5.3.3 (10415 = 3GPP)

Paramètres Réseau

Paramètre	Type	Description
`listen_ip`	Chaîne ou Liste	Adresse(s) IP sur laquelle le DRA écoute. Pour le multihoming SCTP, utilisez une liste de chaînes IP (voir Multihoming SCTP)
`listen_port`	Entier	Port TCP/SCTP pour les connexions Diameter (standard : 3868)
`service_name`	Atome	Identifiant de service interne Erlang
`request_timeout`	Entier	Délai d'attente en millisecondes pour les paires demande/réponse (par défaut : 5000)

Paramètres de Sélection de Pair

Paramètre	Type	Description
`peer_selection_algorithm`	Atome	Algorithme d'équilibrage de charge : `:random` (sélection aléatoire) ou `:failover` (priorité au premier pair)
`allow_undefined_peers_to_connect`	Booléen	Autoriser les connexions des pairs non configurés (par défaut : `false`)
`log_unauthorized_peer_connection_attempts`	Booléen	Journaliser les tentatives de connexion des pairs non autorisés

Configuration des Pairs

Chaque pair dans la liste `peers` définit une connexion Diameter :

```
%{
  host: "mme01.operator.com",
  realm: "operator.com",
  ip: "192.168.1.20",
  port: 3868,
  transport: :diameter_tcp,
  tls: false,
  initiate_connection: false
}
```

Paramètres de Pair

Paramètre	Type	Description
`host`	Chaîne	Identité Diameter du pair (FQDN) - doit correspondre exactement pour le routage
`realm`	Chaîne	Royaume Diameter du pair
`ip`	Chaîne	Adresse IP principale du pair pour la connexion (obligatoire)
`ips`	Liste	Liste d'adresses IP pour le multihoming SCTP (facultatif, voir Multihoming SCTP)
`port`	Entier	Port Diameter du pair (généralement 3868)
`transport`	Atome	Protocole de transport : `:diameter_tcp` ou `:diameter_sctp`
`tls`	Booléen	Activer le chiffrement TLS (si `true`, utiliser généralement le port 3869)
`initiate_connection`	Booléen	`true`: le DRA se connecte au pair, `false`: le DRA attend que le pair se connecte

Modes de Connexion

Initier la Connexion (`initiate_connection: true`)

- Le DRA agit en tant que client Diameter
- Le DRA initie une connexion TCP/SCTP au pair
- Utilisé pour se connecter à HSS, PCRF ou autres systèmes backend
- Le DRA réessaiera les connexions si le pair est injoignable

Accepter la Connexion (`initiate_connection: false`)

- Le DRA agit en tant que serveur Diameter
- Le DRA attend que le pair se connecte
- Utilisé pour les connexions MME, SGSN, P-GW
- Le pair doit être dans la configuration ou `allow_undefined_peers_to_connect: true`

Exemple de Configuration

```
%{
  host: "dra01.mvno.example.com",
  realm: "mvno.example.com",
  listen_ip: "10.100.1.10",
  listen_port: 3868,
  service_name: :mvno_dra,
  product_name: "OmniDRA",
  vendor_id: 10415,
  request_timeout: 5000,
  peer_selection_algorithm: :random,
  allow_undefined_peers_to_connect: false,
  log_unauthorized_peer_connection_attempts: true,
  peers: [
    # MME - attend que le MME se connecte
    %{
      host: "mme01.operator.example.com",
      realm: "operator.example.com",
      ip: "10.100.2.15",
      port: 3868,
      transport: :diameter_sctp,
      tls: false,
      initiate_connection: false
    },
    # HSS - le DRA initie la connexion
    %{
      host: "hss01.mvno.example.com",
      realm: "mvno.example.com",
      ip: "10.100.3.141",
      port: 3868,
      transport: :diameter_tcp,
      tls: false,
      initiate_connection: true
    },
    # PCRF avec TLS - le DRA initie une connexion sécurisée
    %{
      host: "pcrf01.mvno.example.com",
      realm: "mvno.example.com",
      ip: "10.100.3.22",
      port: 3869,
      transport: :diameter_tcp,
      tls: true,
```

```
      initiate_connection: true
    }
  ]
}
```

Remarques Importantes

- **Correspondance de Nom d'Hôte**: Les noms d'hôtes des pairs dans les règles de Routage Avancé doivent correspondre exactement à la valeur `host` configurée ici (sensible à la casse)
- **Échange de Capacités**: Lors de la connexion, les pairs échangent les applications prises en charge via des messages CER/CEA
- **Support d'Application**: Le DRA annonce toutes les applications 3GPP prises en charge (voir Identifiants d'Application 3GPP Communs)
- **Vendor-ID 10415**: Valeur standard pour les applications 3GPP
- **Délai d'Attente des Demandes**: Affecte le TTL des Métriques Étendues (délai d'attente + 5 secondes)
- **Sélection de Pair**: Lorsque plusieurs pairs correspondent aux critères de routage, `peer_selection_algorithm` détermine lequel est choisi

Considérations de Sécurité

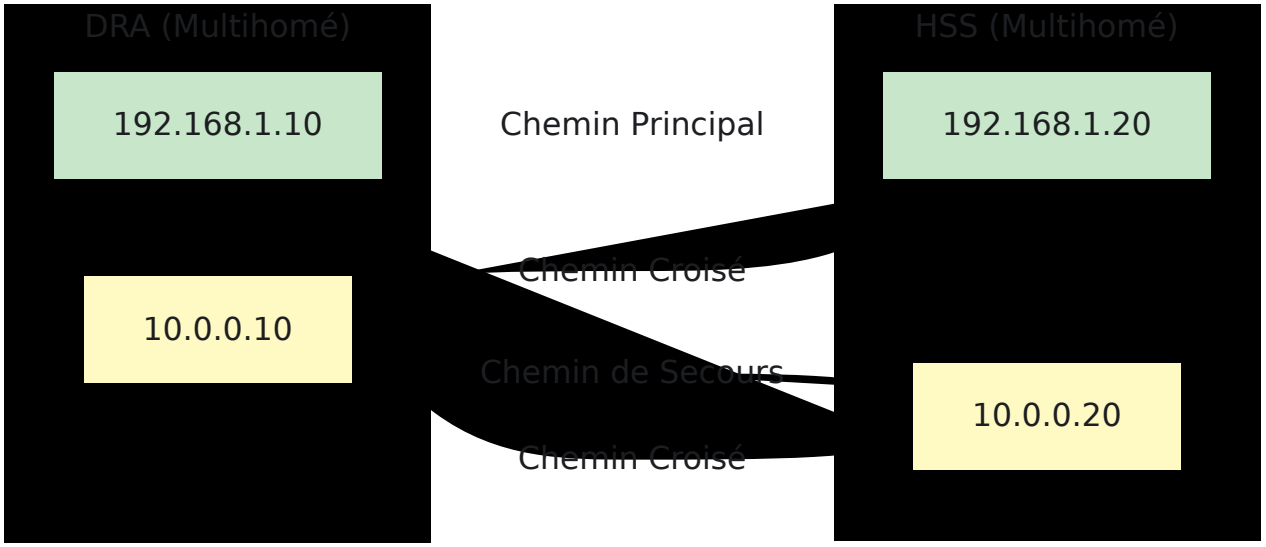
- Définir `allow_undefined_peers_to_connect: false` en production
- Activer `log_unauthorized_peer_connection_attempts: true` pour la surveillance de la sécurité
- S'assurer que les règles de pare-feu correspondent aux paramètres `listen_ip` et `listen_port`
- Valider les certificats des pairs lors de l'utilisation de TLS

Multihoming SCTP

Le multihoming SCTP fournit une redondance réseau en permettant aux points de terminaison de se lier à plusieurs adresses IP. Si le chemin réseau principal

échoue, le SCTP bascule automatiquement vers un chemin alternatif sans perturber la session Diameter.

Comment Ça Fonctionne



- Les cœurs SCTP surveillent tous les chemins réseau
- La bascule automatique se produit si le chemin principal devient injoignable
- Aucune perturbation de la session Diameter pendant le changement de chemin
- Le noyau gère automatiquement la sélection de chemin

Configuration

Adresses d'Écoute DRA

Configurer plusieurs adresses IP locales pour que le DRA s'y lie :

```
%{
  # IP unique (compatible avec les versions antérieures)
  listen_ip: "192.168.1.10",

  # Plusieurs IP pour le multihoming SCTP
  listen_ip: ["192.168.1.10", "10.0.0.10"],

  listen_port: 3868,
  ...
}
```

Remarques :

- Le transport TCP utilise uniquement la première IP de la liste
- Le transport SCTP se lie à toutes les IP spécifiées
- Le format de chaîne IP unique reste entièrement pris en charge

Configuration des Pairs

Configurer plusieurs adresses IP distantes pour les connexions de pairs :

```
peers: [
  %{
    host: "hss01.example.com",
    realm: "example.com",
    ip: "192.168.1.20",                         # IP principale
(requise)
    additional_ips: ["192.168.1.20", "10.0.0.20"],        # Toutes
les IP pour le multihoming
    port: 3868,
    transport: :diameter_sctp,
    tls: false,
    initiate_connection: true
  }
]
```

Remarques :

- Le champ `ip` est requis pour la compatibilité avec les versions antérieures

- Le champ `ips` est facultatif ; s'il est omis, seul `ip` est utilisé
- Pour SCTP, inclure l'IP principale dans la liste `ips`
- Pour TCP, seul `ip` est utilisé (TCP ne prend pas en charge le multihoming)

Exemple Complet

```
config :dra,
  diameter: %{
    service_name: :omnitouch_dra,
    listen_ip: ["192.168.1.10", "10.0.0.10"],  # DRA multihomé
    listen_port: 3868,
    host: "dra01",
    realm: "example.com",
    product_name: "OmniDRA",
    vendor_id: 10415,
    request_timeout: 5000,
    peer_selection_algorithm: :random,
    allow_undefined_peers_to_connect: false,
    peers: [
      # Connexion HSS multihomée
      %{
        host: "hss01.example.com",
        realm: "example.com",
        ip: "192.168.1.20",
        additional_ips: ["192.168.1.20", "10.0.0.20"],
        port: 3868,
        transport: :diameter_sctp,
        tls: false,
        initiate_connection: true
      },
      # MME à domicile unique (compatible avec les versions
antérieures)
      %{
        host: "mme01.example.com",
        realm: "example.com",
        ip: "192.168.1.30",
        port: 3868,
        transport: :diameter_sctp,
        tls: false,
        initiate_connection: false
      }
    ]
  }
```

Exigences

- Le module noyau SCTP doit être chargé (package `lksctp-tools` sur Linux)
- Toutes les adresses IP doivent être routables depuis/vers le pair
- Les règles de pare-feu doivent autoriser le trafic SCTP sur toutes les IP configurées
- Les deux points de terminaison doivent être configurés pour le multihoming pour une redondance complète

Limitations

- Le transport TCP ne prend pas en charge le multihoming (n'utilise que l'IP principale)
- TLS sur le multihoming SCTP peut avoir des limitations de compatibilité
- Le timing de la bascule de chemin dépend des paramètres SCTP du noyau

---

Tables de Référence

Identifiants d'Application 3GPP Communs

Identifiant d'Application	Interface	Description
16777251	S6a/S6d	Authentification MME/SGSN et données d'abonnement vers HSS
16777252	S13/S13'	Vérification de l'identité de l'équipement MME vers EIR
16777238	Gx	PCEF vers PCRF contrôle de politique et de facturation
16777267	S9	Politique de roaming de PCRF domicile vers PCRF visité
16777272	Sy	Liaison de session PCRF vers OCS
16777216	Cx	I-CSCF/S-CSCF vers enregistrement IMS HSS
16777217	Sh	AS vers données utilisateur IMS HSS
16777236	SLg	MME/SGSN vers services de localisation GMLC
16777291	SLh	GMLC vers informations d'abonnement de localisation HSS
16777302	S6m	MTC-IWF vers HSS/HLR pour dispositifs M2M

Identifiant d'Application	Interface	Description
16777308	S6c	SMS-SC/IP-SM-GW vers routage SMS HSS
16777343	S6t	SCEF vers événements de surveillance HSS
16777334	Rx	AF vers autorisation de média PCRF

Codes AVP Communs

Code	Nom AVP	Type	Utilisation
1	User-Name	UTF8String	Identifiant de l'abonné (IMSI dans 3GPP)
264	Origin-Host	DiameterIdentity	Nom d'hôte du pair d'origine
268	Result-Code	Unsigned32	Code de résultat standard
283	Destination-Realm	DiameterIdentity	Royaume cible
293	Destination-Host	DiameterIdentity	Hôte cible (facultatif)
296	Origin-Realm	DiameterIdentity	Royaume source
297	Experimental-Result	Grouped	Code de résultat spécifique au fournisseur

Codes de Commande Communs

Les codes de commande font partie de l'en-tête du message Diameter, pas des AVP :

Code	Nom de Commande	Description
257	CER/CEA	Demande/Réponse d'Échange de Capacités
258	RAR/RAA	Demande/Réponse de Ré-Authentification
274	ASR/ASA	Demande/Réponse d'Interruption de Session
275	STR/STA	Demande/Réponse de Terminaison de Session
280	DWR/DWA	Demande/Réponse de Surveillance de Dispositif
282	DPR/DPA	Demande/Réponse de Déconnexion de Pair
316	ULR/ULA	Demande/Réponse de Mise à Jour de Localisation (S6a)
317	CLR/CLA	Demande/Réponse d'Annulation de Localisation (S6a)
318	AIR/AIA	Demande/Réponse d'Information d'Authentification (S6a)
321	PUR/PUA	Demande/Réponse de Purge d'UE (S6a)

Module de Routage Avancé

Le module de Routage Avancé fournit des capacités de routage de messages flexibles et basées sur des règles avec support pour des conditions de correspondance complexes.

Important : Ce module évalue **uniquement les paquets de demande Diameter entrants** (pas les paquets de réponse). Les paquets de réponse suivent le routage de session établi vers le pair d'origine - voir Routage des Réponses pour plus de détails.

Configuration

Activez le module et définissez les règles de routage dans votre configuration :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: <identifiant_de_règle>
      match: <portée_de_correspondance>
      filters: [<liste_de_filtres>]
      route:
        peers: [<liste_de_pairs>]
```

Paramètres

Paramètre	Description
`enabled`	Définir sur `True` pour activer le module
`rule_name`	Identifiant unique pour la règle de routage
`match`	Comment les filtres sont combinés : `:all` (logique ET - tous les filtres doivent correspondre), `:any` (logique OU - au moins un filtre doit correspondre), `:none` (logique NOR - aucun filtre ne peut correspondre)
`filters`	Liste des conditions de filtre (voir Filtres Disponibles)
`route`	Action de routage (voir Actions de Routage ci-dessous)

Actions de Routage

Le paramètre `route` prend en charge plusieurs actions :

Router vers des Pairs

```
route:
  peers: [pair01.example.com, pair02.example.com]
```

Routage vers les noms d'hôtes de pair sp��cifiés. Les pairs doivent être :

- Définis dans la configuration des pairs Diameter du DRA
- Le nom d'hôte exact tel que configuré (sensible à la casse)
- Actuellement connectés pour que le routage réussisse (les pairs déconnectés sont ignorés)

Router vers AVP Destination-Host

```
route: :destination_host
```

Routage vers le pair spécifié dans l'AVP Destination-Host (293). Si l'AVP Destination-Host est manquant, le routage revient au comportement normal.

Supprimer le Trafic

```
route: :drop
```

Supprime silencieusement le message sans envoyer de réponse. Utilisé pour :

- Filtrage de trafic et blackholing
- Blocage des demandes indésirables
- Limitation de taux en supprimant le trafic excessif

Comportement :

- Le message est supprimé au DRA (non transféré)
- Aucun message de réponse n'est envoyé au pair demandeur
- Implémente le comportement `:discard` de Diameter Erlang
- Métrique : `diameter_advanced_routing_drop_count_total` (voir Métriques Prometheus)

Générer une Réponse d'Erreur

```
route: {:error, 3004}
```

Génère une réponse d'erreur Diameter avec le Code de Résultat spécifié et l'envoie au pair demandeur. Codes de résultat courants :

- `3002` - DIAMETER_UNABLE_TO_DELIVER (routage indisponible)
- `3003` - DIAMETER_REALM_NOT_SERVED (royaume non pris en charge)
- `3004` - DIAMETER_TOO_BUSY (protection contre la surcharge, limitation de taux)
- `5012` - DIAMETER_UNABLE_TO_COMPLY (rejet général)

Comportement :

- Le DRA génère une réponse d'erreur avec le Code de Résultat spécifié
- La réponse inclut Origin-Host, Origin-Realm, Session-Id (auto-rempli par Diameter)
- Le message n'est PAS transféré à aucun pair
- Implémente `{:protocol_error, code}` de Diameter Erlang (équivalent à `{:answer_message, code}`)
- Métrique : `diameter_advanced_routing_error_count_total` (voir Métriques Prometheus)

Filtres Disponibles

Filtres Standards

Disponibles dans les modules Routage Avancé et Transformation Avancée :

- **`:application_id`** - Correspond à l'identifiant d'application Diameter (voir référence Identifiant d'Application)
  - Valeur unique : `{:application_id, 16777251}` (S6a/S6d)
  - Valeurs multiples : `{:application_id, [16777251, 16777252]}` (S6a ou S6b)
- **`:command_code`** - Correspond au code de commande Diameter
  - Valeur unique : `{:command_code, 318}` (demande AIR)
  - Valeurs multiples : `{:command_code, [317, 318]}` (ULR ou AIR)
- **`:avp`** - Correspond à la valeur AVP (voir référence code AVP)
  - Correspondance exacte : `{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}`
  - Correspondance regex : `{:avp, {1, ~r"999001.*"}}`
  - Plusieurs motifs : `{:avp, {1, ["505057001313606", ~r"999001.*", ~r"505057.*"]}}`
  - Toute valeur (vérification de présence) : `{:avp, {264, :any}}`

Filtre Spécifique au Routage

Uniquement disponible dans Routage Avancé :

- `:via_peer` - Correspond au pair d'où la demande a été reçue
  - Pair unique : `{:via_peer, "omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org"}`
  - Pairs multiples : `{:via_peer, ["omnitouch-lab-dra01.epc.mnc001.mcc001.3gppnetwork.org", "omnitouch-lab-dra02.epc.mnc001.mcc001.3gppnetwork.org"]}`
  - Tout pair : `{:via_peer, :any}`

Filtres Spécifiques à la Transformation

Uniquement disponibles dans Transformation Avancée :

- `:to_peer` - Correspond au pair de destination prédéterminé (uniquement pour les paquets de demande)
  - Pair unique : `{:to_peer, "dra01.omnitouch.com.au"}`
  - Pairs multiples : `{:to_peer, ["dra01.omnitouch.com.au", "dra02.omnitouch.com.au"]}`
- `:from_peer` - Correspond au pair qui a envoyé la réponse (uniquement pour les paquets de réponse)
  - Pair unique : `{:from_peer, "hss-01.example.com"}`
  - Pairs multiples : `{:from_peer, ["hss-01.example.com", "hss-02.example.com"]}`
- `:packet_type` - Correspond à la direction du paquet
  - Demande : `{:packet_type, :request}`
  - Réponse : `{:packet_type, :answer}`

Remarques Importantes sur les Filtres

- **Filtres AVP** : Recommandés uniquement pour les AVP simples (User-Name, Origin-Host, Destination-Realm, etc.)

- Les AVP groupés ne sont **pas pris en charge** et ne correspondront pas
- Les valeurs binaires complexes ne sont **pas prises en charge**
- Utiliser le format : `{:avp, {code, value}}`

- **Opérateurs de Liste** : Pris en charge pour toutes les valeurs de filtre sauf `:packet_type`
  - Lorsqu'une liste est utilisée, elle applique une logique **OU** au sein de la liste
  - Exemple : `{:command_code, [317, 318]}` correspond au code de commande 317 **OU** 318

- **Valeurs Spéciales** :
  - `:any` - Correspond à toute valeur (vérifie la présence de l'AVP)
  - Exemple : `{:avp, {264, :any}}` correspond si l'AVP Origin-Host existe avec n'importe quelle valeur

Exemples de Routage

Exemple 1 : Routage Via Pair

Router les messages en fonction du DRA dont ils proviennent :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: temporary_until_cutover_s6a_via_to_local_hss
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:via_peer, ["omnitouch-lab-
dra01.epc.mnc001.mcc001.3gppnetwork.org", "omnitouch-lab-
dra02.epc.mnc001.mcc001.3gppnetwork.org"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
      route:
        peers: [omnitouch-lab-
hss01.epc.mnc001.mcc001.3gppnetwork.org, omnitouch-lab-
hss02.epc.mnc001.mcc001.3gppnetwork.org]
```

Comment ça fonctionne : Route le trafic S6a qui arrive via des pairs DRA spécifiques vers des nœuds HSS locaux.

Exemple 2 : Roaming Entrant avec Correspondance de Motif

Router le trafic de roaming en fonction des motifs IMSI :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: inbound_s6a_roaming_to_dcc
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, ["505571234567", ~r"999001.*"]}}'
      route:
        peers: [dra01.omnitouch.com.au, dra02.omnitouch.com.au]
```

Comment ça fonctionne : Route les messages S6a d'un Royaume d'Origine spécifique avec des motifs IMSI correspondants vers des pairs DRA désignés.

Exemple 3 : Routage Dynamique avec :destination_host

Router vers la valeur de l'AVP Destination-Host dans le message :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: route_to_specified_destination_host
      match: ":all"
      filters:
        - '{:avp, {1, [~r"90199.*"]}}'  # Correspondance de motif
IMSI
      route: :destination_host
```

Comment ça fonctionne :

- Lorsque les filtres correspondent, route vers le pair spécifié dans l'AVP Destination-Host (293)

- Si l'AVP Destination-Host est manquant, la correspondance est considérée comme un échec et revient au routage normal
- Utile pour honorer le routage lorsque l'expéditeur spécifie la destination exacte

Exemple 4 : Supprimer le Trafic Indésirable

Supprimer le trafic des plages IMSI spécifiques :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: drop_test_subscribers
      match: ":all"
      filters:
        - '{:application_id, 16777251}'  # S6a
        - '{:avp, {1, [~r"999999.*"]}}'  # Plage IMSI de test
      route: :drop
```

Comment ça fonctionne :

- Correspond aux messages S6a avec un IMSI commençant par 999999
- Supprime silencieusement le message sans envoyer de réponse
- Utile pour filtrer le trafic de test ou bloquer des plages d'abonnés spécifiques
- Voir Métriques Prometheus pour surveiller le trafic supprimé

Exemple 5 : Limitation de Taux avec Réponses d'Erreur

Retourner DIAMETER_TOO_BUSY pour des motifs de trafic spécifiques :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: rate_limit_high_volume_peer
      match: ":all"
      filters:
        - '{:via_peer, "mme-overloaded-01.example.com"}'
        - '{:application_id, 16777251}'
      route: {:error, 3004}
```

Comment ça fonctionne :

- Correspond au trafic S6a provenant d'un pair surchargé spécifique
- Retourne une réponse d'erreur DIAMETER_TOO_BUSY (3004)
- Le pair demandeur reçoit une erreur et doit se retirer
- Utile pour la protection contre la surcharge et la limitation de taux
- Voir Métriques Prometheus pour surveiller les réponses d'erreur

Exemple 6 : Réponses d'Erreur Conditionnelles par Commande

Bloquer des types de commandes spécifiques avec des codes d'erreur appropriés :

```
dra_module_advanced_routing:
  enabled: True
  rules:
    - rule_name: block_purge_requests
      match: ":all"
      filters:
        - '{:application_id, 16777251}'  # S6a
        - '{:command_code, 321}'         # PUR (Demande de Purge
d'UE)
      route: {:error, 5012}
```

Comment ça fonctionne :

- Correspond aux messages S6a Demandes de Purge d'UE
- Retourne une erreur DIAMETER_UNABLE_TO_COMPLY (5012)

- Bloque des opérations spécifiques sans supprimer le trafic silencieusement
- Utile pour désactiver sélectivement certaines commandes Diameter

---

Module de Transformation Avancée

Le module de Transformation Avancée permet la modification dynamique des AVP de message Diameter en fonction de critères de correspondance. Voir Traitement des Règles pour des détails sur la façon dont les règles sont évaluées.

Configuration

Activez le module et définissez les règles de transformation :

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: <identifiant_de_règle>
      match: <portée_de_correspondance>
      filters: [<liste_de_filtres>]
      transform:
        action: <action_de_transformation>
        avps: [<modifications_avp>]
```

Paramètres

Paramètre	Description
`enabled`	Définir sur `True` pour activer le module
`rule_name`	Identifiant unique pour la règle de transformation
`match`	Comment les filtres sont combinés : `:all` (logique ET), `:any` (logique OU), `:none` (logique NOR) - voir Logique de Filtre
`filters`	Liste des conditions de filtre (voir Filtres Disponibles)
`transform.action`	Type de transformation (`:edit`, `:remove`, ou `:overwrite`)
`transform.avps`	Liste des modifications AVP à appliquer (voir référence code AVP)

Actions de Transformation

Paquets de Demande (Demandes Diameter)

- **`:edit`** - Modifier les valeurs AVP existantes
  - Modifie uniquement les AVP qui existent dans le message
  - Si l'AVP n'existe pas, aucun changement n'est effectué
- **`:remove`** - Supprimer les AVP du message
- **`:overwrite`** - Remplacer des structures AVP entières
  - Nécessite le paramètre `dictionary` spécifiant le dictionnaire Diameter (par exemple, `:diameter_gen_3gpp_s6a`)

Paquets de Réponse (Réponses Diameter)

- **`:remove`** - Supprimer les AVP du message
- **`:overwrite`** - Remplacer des structures AVP entières
  - Nécessite le paramètre `dictionary`

Important : Si aucune règle ne correspond, le paquet est transmis de manière transparente sans aucune transformation.

Syntaxe de Modification AVP

Modification standard :

- `{:avp, {<code>, <new_value>}}` - Définir l'AVP à une nouvelle valeur

Suppression d'AVP :

- `{:avp, {<code>, :any}}` - Supprimer l'AVP par ID (supprime indépendamment de la valeur actuelle)
- Remarque : La suppression basée sur l'ID avp est prise en charge ; la suppression basée sur le contenu de l'AVP n'est pas prise en charge

Remplacement avec dictionnaire :

```
transform: %{
  action: :overwrite,
  dictionary: :diameter_gen_3gpp_s6a,
  avps: [{:avp, {:"s6a_Supported-Features", {:"s6a_Supported-
Features", 10415, 1, 3221225470, []}}}]
}
```

Exemples de Transformation

Exemple 1 : Réécriture du Royaume de Destination Basée sur le Pair

Réécrire le Royaume de Destination en fonction de l'endroit où le message est routé :

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: rewrite_s6a_destination_realm_for_Operator_X
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, [~r"9999999.*"]}}'
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc999.mcc999.3gppnetwork.org"}}'
```

Comment ça fonctionne : Lorsque les demandes S6a sont routées vers des pairs DRA spécifiques et correspondent au motif IMSI, réécrit le Royaume de Destination pour le réseau de l'Opérateur X.

Exemple 2 : Routage de Plusieurs Opérateurs avec Transformations

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name:
rewrite_s6a_destination_realm_for_roaming_partner_ausie
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc057.mcc505.3gppnetwork.org"}}'
        - '{:avp, {1, [~r"50557.*"]}}'
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc030.mcc310.3gppnetwork.org"}}'
```

Comment ça fonctionne : Route différentes plages d'abonnés IMSI vers les royaumes de réseau appropriés en fonction des motifs IMSI. La première règle correspondante l'emporte (voir Ordre d'Exécution).

Exemple 3 : Réécriture du Royaume pour MVNO

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: rewrite_s6a_destination_realm_for_single_sub
      match: ":all"
      filters:
        - '{:to_peer, ["dra01.omnitouch.com.au",
"dra02.omnitouch.com.au"]}'
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'
        - '{:avp, {1, ["505057000003606"]}}'  # Correspondance
exacte IMSI
      transform:
        action: ":edit"
        avps:
          - '{:avp, {283, "epc.mnc001.mcc001.3gppnetwork.org"}}'
```

Comment ça fonctionne : Transforme le Royaume de Destination pour un abonné MVNO spécifique vers leur réseau central hébergé.

Exemple 4 : Transformation Uniquement pour les Demandes avec Filtre de Type de Paquet

Transformer uniquement les paquets de demande (pas les réponses) :

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: Tutorial_Rule_AIR
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
        - '{:command_code, 318}'
        - '{:packet_type, :request}'
        - '{:avp, {1, "999999000000001"}}'
        - '{:avp, {264, :any}}'  # Origin-Host doit exister avec
n'importe quelle valeur
      transform:
        action: ":edit"
        avps:
          - '{:avp, {1, "999999000000002"}}'
```

Comment ça fonctionne :

- Correspond uniquement aux paquets **demande** S6a AIR (pas aux paquets de réponse)
- Vérifie que le User-Name (AVP 1) est égal à "999999000000001"
- Vérifie que l'Origin-Host (AVP 264) existe avec n'importe quelle valeur
- Réécrit le User-Name à "999999000000002"
- Si l'AVP n'existe pas, aucun changement n'est effectué

Exemple 5 : Supprimer un AVP

Supprimer un AVP spécifique des messages :

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: remove_user_name_avp
      match: ":all"
      filters:
        - '{:application_id, 16777251}'
      transform:
        action: ":remove"
        avps:
          - '{:avp, {1, :any}}'  # Supprimer User-Name
indépendamment de la valeur
```

Comment ça fonctionne : Supprime l'AVP User-Name (code 1) de tous les messages S6a, indépendamment de sa valeur actuelle.

Exemple 6 : Écraser un AVP Groupé sur les Paquets de Réponse

Modifier des AVP groupés complexes dans les paquets de réponse en utilisant l'action `:overwrite` avec le support de dictionnaire :

```
dra_module_advanced_transform:
  enabled: True
  rules:
    - rule_name: add_sos_apn_to_ula
      match: ":all"
      filters:
        - '{:application_id, 16777251}'       # S6a/S6d
        - '{:command_code, 316}'              # ULA (Réponse de
Mise à Jour de Localisation)
        - '{:packet_type, :answer}'           # Paquets de réponse
uniquement
        - '{:avp, {296, "epc.mnc001.mcc001.3gppnetwork.org"}}'  #
Royaume d'Origine
      transform:
        action: ":overwrite"
        dictionary: ":diameter_gen_3gpp_s6a"
        avps:
          - '{:avp, {:"s6a_APN-Configuration-Profile",
              {:"s6a_APN-Configuration-Profile", 1, 0, [
                {:"s6a_APN-Configuration", 1, 0, "internet", [],
                  [{:"s6a_EPS-Subscribed-QoS-Profile", 9,
                    {:"s6a_Allocation-Retention-Priority", 1, [0],
[0], []}, []}],
                  [1], [], [], [1], ["0800"],
                  [{:s6a_AMBR, 4200000000, 4200000000, [], [],
[]}],
                  [], [], [], [], [], [], [], [], [], [], [], [],
[], [], []},
                {:"s6a_APN-Configuration", 2, 0, "ims", [],
                  [{:"s6a_EPS-Subscribed-QoS-Profile", 5,
                    {:"s6a_Allocation-Retention-Priority", 1, [0],
[1], []}, []}],
                  [0], [], [], [1], ["0800"],
                  [{:s6a_AMBR, 4200000000, 4200000000, [], [],
[]}],
                  [], [], [], [], [], [], [], [], [], [], [], [],
[], [], []},
                {:"s6a_APN-Configuration", 3, 0, "sos", [],
                  [{:"s6a_EPS-Subscribed-QoS-Profile", 5,
                    {:"s6a_Allocation-Retention-Priority", 1, [0],
[1], []}, []}],
                  [1], [], [], [1], ["0800"],
                  [{:s6a_AMBR, 4200000000, 4200000000, [], [],
```

```
[]}],
                  [], [], [], [], [], [], [], [], [], [], [], [],
[], [], []}
              ], []}
            }}'
```

Comment ça fonctionne :

- Correspond aux paquets S6a de Réponse de Mise à Jour de Localisation (ULA) d'un Royaume d'Origine spécifique
- Utilise l'action `:overwrite` pour remplacer l'ensemble de l'AVP groupé APN-Configuration-Profile
- **Nécessite le paramètre `dictionary`** pour encoder correctement des structures AVP groupées complexes
- Ajoute trois configurations APN : "internet" (contexte 1), "ims" (contexte 2) et "sos" (contexte 3)
- Chaque APN inclut des profils QoS, des limites de bande passante (AMBR) et des paramètres de type PDN
- La transformation garantit que les services d'urgence (SOS) APN sont provisionnés pour tous les abonnés de ce royaume

Quand utiliser `:overwrite` avec dictionnaire :

- Modifier des AVP groupés avec des structures imbriquées (comme APN-Configuration-Profile)
- Ajouter ou restructurer des données d'abonnement complexes 3GPP
- Lorsque l'action `:edit` ne peut pas gérer la complexité de l'AVP
- Le dictionnaire doit correspondre à l'application Diameter (`:diameter_gen_3gpp_s6a` pour S6a, etc.)

Remarques importantes :

- `:overwrite` remplace l'ensemble de l'AVP, pas seulement des champs individuels
- La structure de l'AVP doit correspondre exactement à la définition du dictionnaire

- Une structure incorrecte entraînera des échecs d'encodage et des paquets supprimés
- Il s'agit d'une fonctionnalité avancée - validez soigneusement dans un environnement de test d'abord

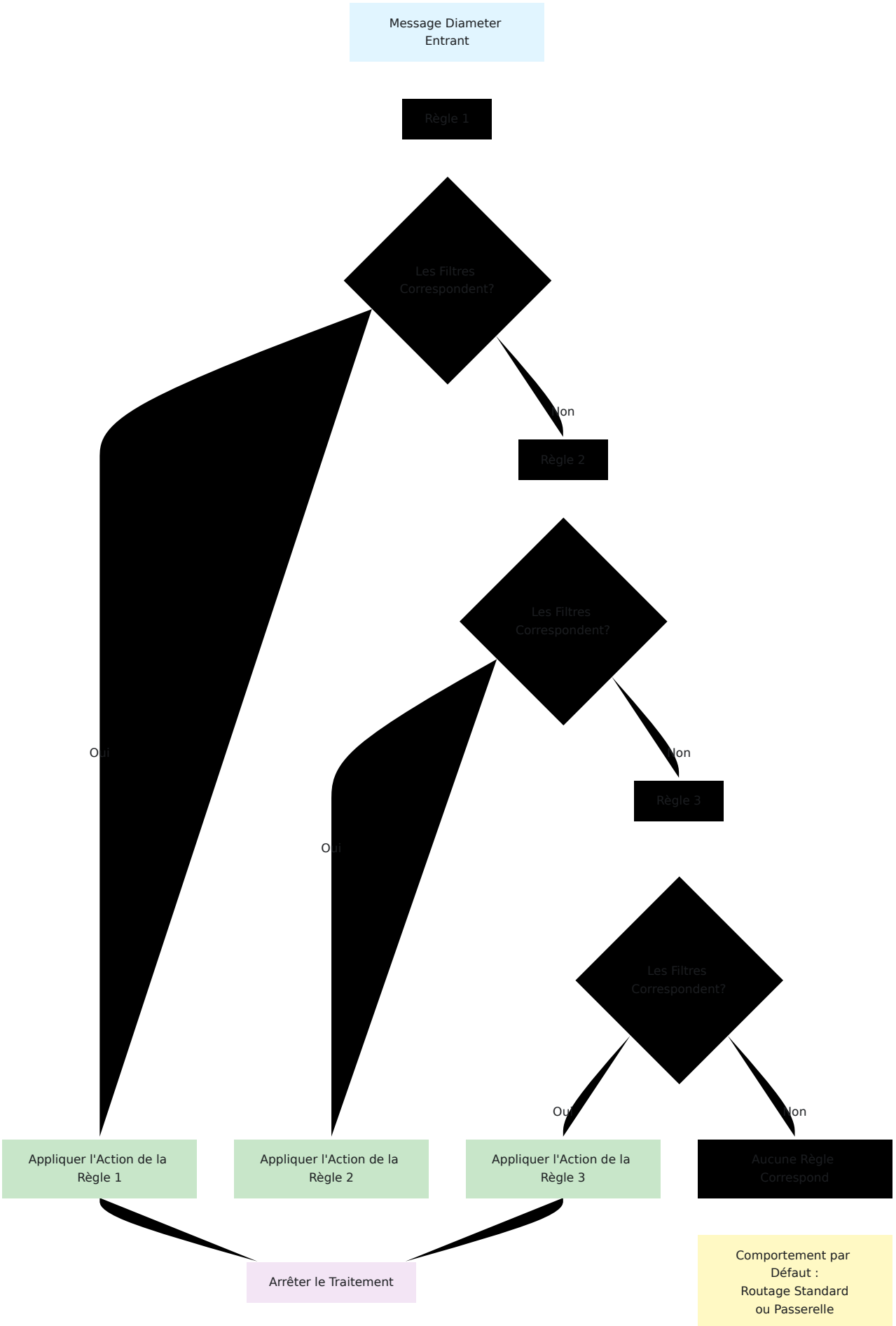
Cas d'Utilisation

- **Support MVNO** : Router le trafic des opérateurs virtuels vers des réseaux centraux hébergés
- **Migration Réseau** : Rediriger progressivement les abonnés vers une nouvelle infrastructure
- **Traduction de Royaume** : Convertir entre différents schémas de nommage pour les partenaires de roaming
- **Multi-location** : Isoler les populations d'abonnés par royaume
- **Routage des Opérateurs** : Diriger le trafic vers les réseaux d'opérateurs corrects en fonction des plages IMSI

Traitement des Règles

S'applique aux modules Routage Avancé et Transformation Avancée.

Ordre d'Exécution


Message Diameter Entrant
Règle 1
Les Filtres Correspondent?
Non
Règle 2
Les Filtres Correspondent?
Oui
Non
Règle 3
Oui
Les Filtres Correspondent?
Oui
Non
Appliquer l'Action de la Règle 1
Appliquer l'Action de la Règle 2
Appliquer l'Action de la Règle 3
Aucune Règle Correspond
Arrêter le Traitement
Comportement par Défaut : Routage Standard ou Passerelle

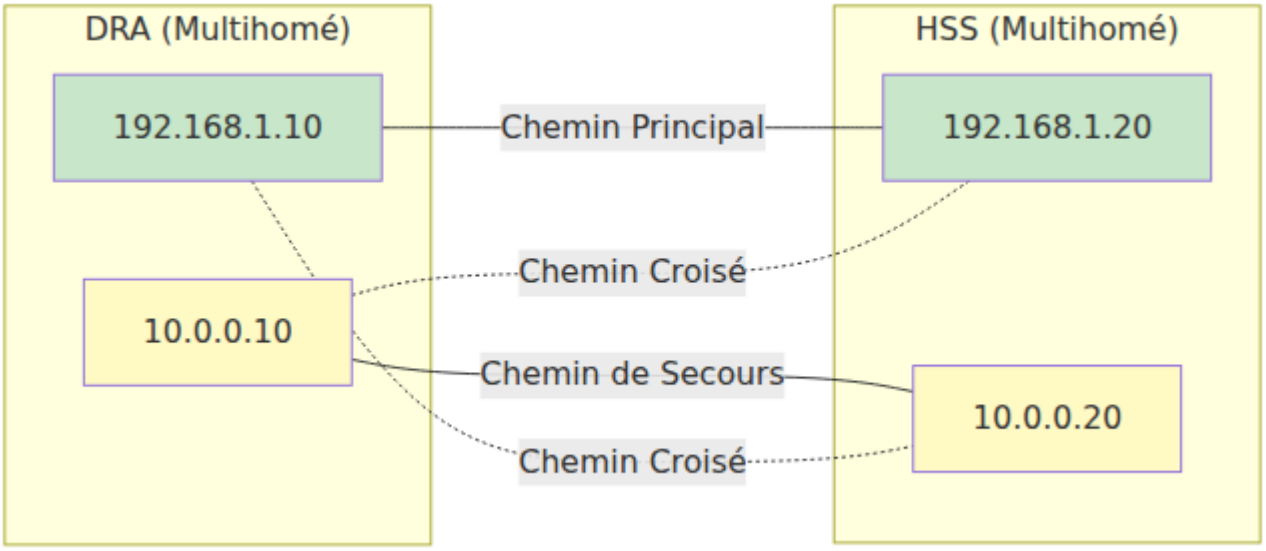
1. Les règles sont évaluées **dans l'ordre de haut en bas** tel que défini dans la configuration
2. Les filtres au sein d'une règle sont évalués en fonction du paramètre `match` (`:all`, `:any`, ou `:none`)
3. **La première règle correspondante l'emporte** - les règles suivantes ne sont pas évaluées
4. Si aucune règle ne correspond, le comportement de routage/passerelle par défaut est utilisé

Logique de Filtre

Le paramètre `match` détermine comment les filtres sont combinés :

match: :all (Logique ET)

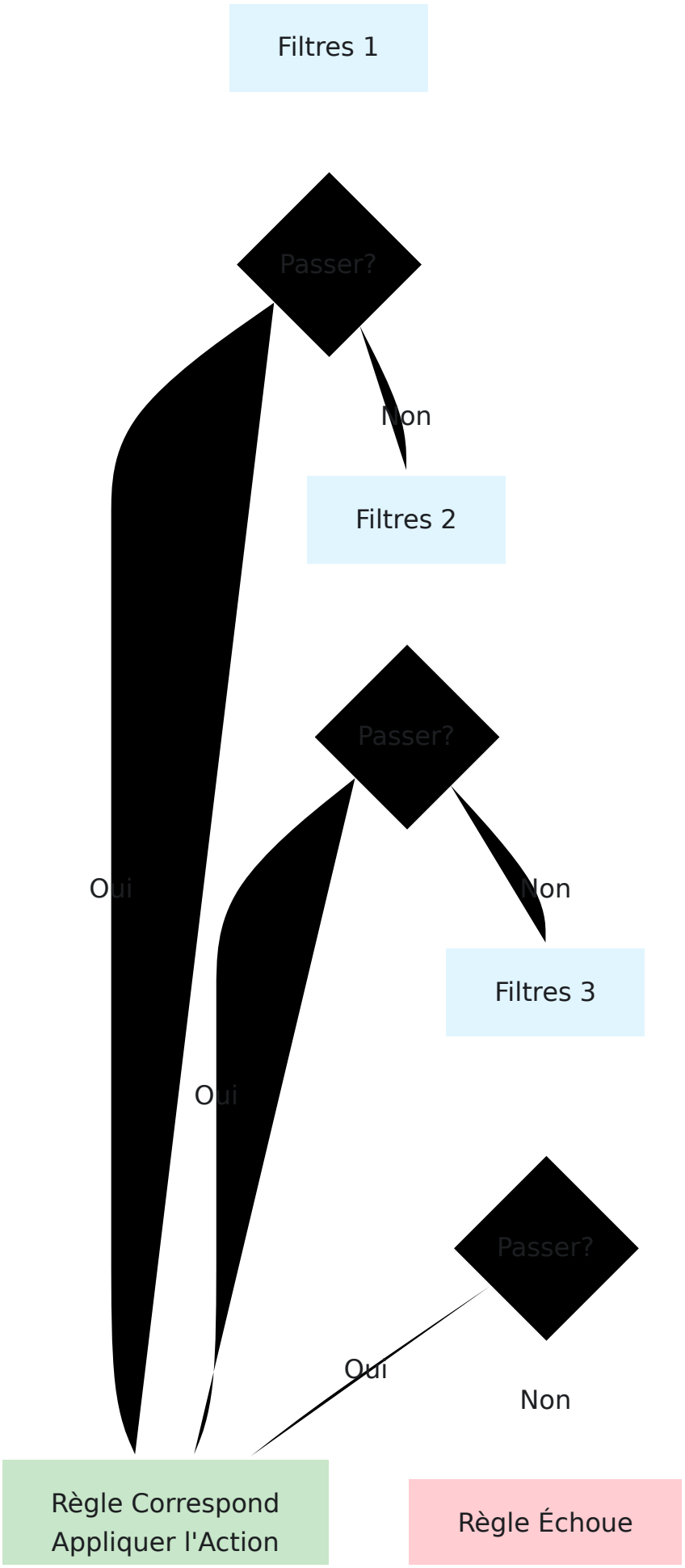
Tous les filtres doivent correspondre pour que la règle réussisse.



Exemple : Avec 3 filtres, `filtre1 ET filtre2 ET filtre3` doivent tous être vrais.

match: :any (Logique OU)

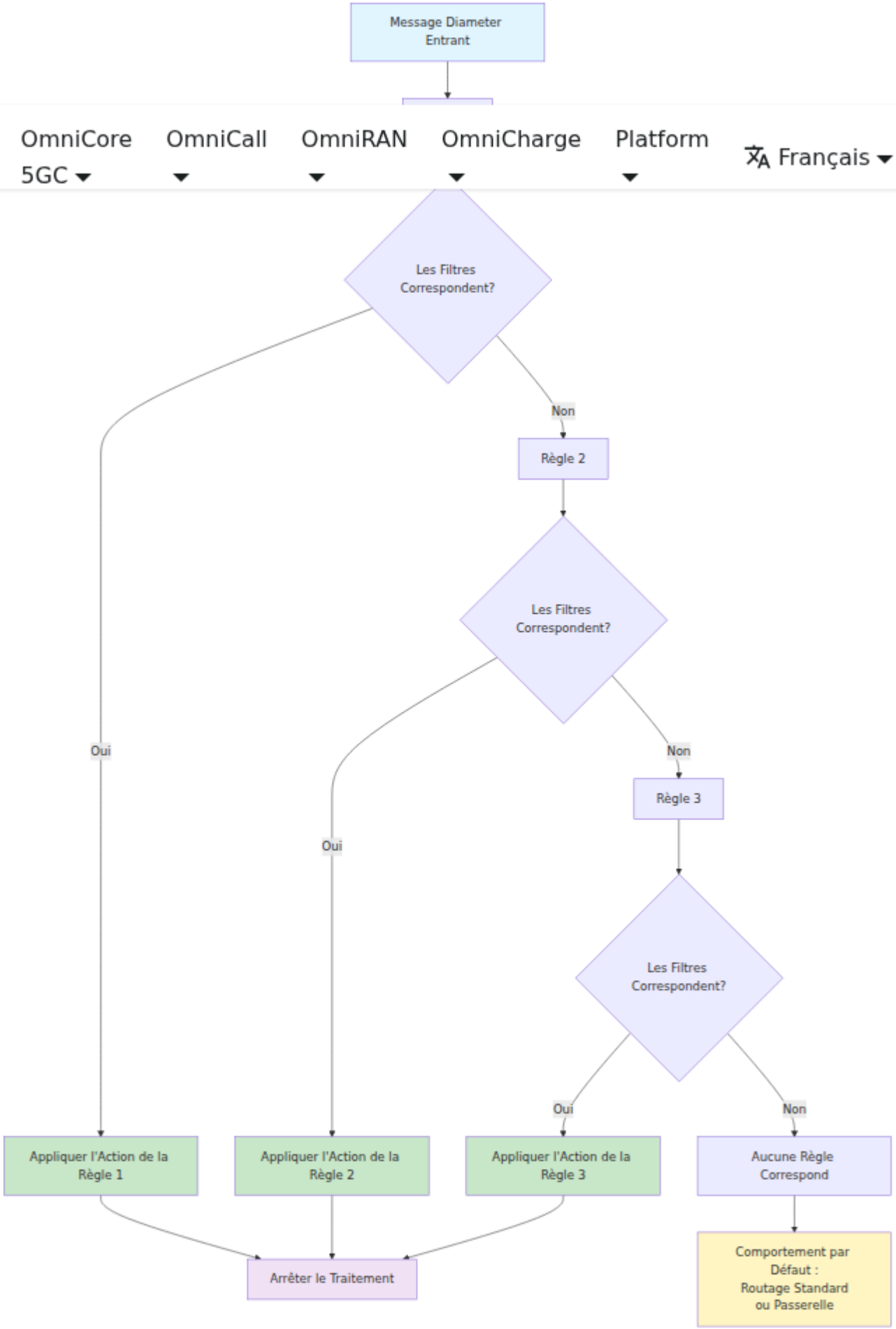
Au moins un filtre doit correspondre pour que la règle réussisse.


Filtres 1
Passer?
Non
Filtres 2
Passer?
Oui
Non
Filtres 3
Oui
Passer?
Oui
Non
Règle Correspond
Appliquer l'Action
Règle Échoue

Exemple : Avec 3 filtres, `filtre1 OU filtre2 OU filtre3` (au moins un passe).

match: :none (Logique NOR)

Aucun filtre ne peut correspondre pour que la règle réussisse (correspondance inverse).


Message Diameter Entrant
Les Filtres Correspondent?
Oui
Non
Règle 2
Les Filtres Correspondent?
Oui
Non
Règle 3
Les Filtres Correspondent?
Oui
Non
Appliquer l'Action de la Règle 1
Appliquer l'Action de la Règle 2
Appliquer l'Action de la Règle 3
Aucune Règle Correspond
Arrêter le Traitement
Comportement par Défaut : Routage Standard ou Passerelle

Exemple : Avec 3 filtres, `PAS filtre1 ET PAS filtre2 ET PAS filtre3` (tous doivent échouer).

Remarques Supplémentaires :

Lors de l'utilisation d'opérateurs de liste au sein d'une valeur de filtre (par exemple, `{:avp, {1, ["value1", "value2"]}}`), les valeurs utilisent la logique **OU** (n'importe lequel peut correspondre).

Modèles d'Expression Régulière

Utilisez la syntaxe `~r"pattern"` pour la correspondance regex :

- `~r"999001.*"` - Correspond aux IMSI commençant par 999001
- `~r"^310[0-9]{3}.*"` - Correspond aux IMSI avec des motifs MNC spécifiques
- `~r".*test$"` - Correspond aux valeurs se terminant par "test"

Meilleures Pratiques

1. **Spécificité** : Ordre des règles de la plus spécifique à la plus générale
2. **Performance** : Placer les correspondances les plus courantes en premier pour réduire la surcharge de traitement
3. **Tests** : Valider les motifs regex avant le déploiement
4. **Documentation** : Utiliser des valeurs `rule_name` descriptives pour la clarté opérationnelle
5. **Surveillance** : Suivre les taux de correspondance des règles pour vérifier le comportement attendu

---

Module de Métriques Étendues

Le module de Métriques Étendues fournit des capacités de télémétrie et d'analyse avancées pour analyser les motifs de trafic Diameter au-delà des métriques standard.

Configuration

Activez le module et configurez des types de métriques spécifiques :

```
module_extended_metrics:
  enabled: true
  attach_attempt_reporting_enabled: true
```

Paramètres

Paramètre	Description
`enabled`	Définir sur `true` pour activer le module de métriques étendues
`attach_attempt_reporting_enabled`	Activer le suivi et le rapport des tentatives de connexion LTE (S6a AIR/AIA)

Métriques Disponibles

Suivi des Tentatives de Connexion

Suit les tentatives de connexion des abonnés LTE en surveillant les paires de messages Demande d'Information d'Authentification (AIR) et Réponse (AIA) :

Parse error on line 36: ... style Metrics fill:#f3e5f5 style E ----------^ Expecting 'SOLID_OPEN_ARROW', 'DOTTED_OPEN_ARROW', 'SOLID_ARROW', 'BIDIRECTIONAL_SOLID_ARROW', 'DOTTED_ARROW', 'BIDIRECTIONAL_DOTTED_ARROW', 'SOLID_CROSS', 'DOTTED_CROSS', 'SOLID_POINT', 'DOTTED_POINT', got 'TXT'

Réessayer

Mesure : `attach_attempt_count`

Champs :

- `imsi` - L'IMSI de l'abonné (depuis l'AVP User-Name - voir codes AVP)

Tags :

- `origin_host` - Le pair qui a originairement fait la demande de connexion
- `result_code` - Le code de résultat Diameter de la réponse HSS

Comment ça fonctionne :

1. Lorsqu'une AIR (code de commande 318, application S6a 16777251 - voir Identifiants d'Application) est reçue, le module extrait :
   - ID de Bout en Bout pour la corrélation demande/réponse
   - IMSI (AVP User-Name code 1)
   - Origin-Host (AVP code 264)
2. Les métadonnées de la demande sont stockées dans ETS avec un TTL
3. Lorsque l'AIA correspondante est reçue, le module :
   - Corrèle en utilisant l'ID de Bout en Bout
   - Extrait le code de résultat (AVP 268 ou AVP de code de résultat expérimental 297)
   - Émet la métrique avec IMSI, origin host, et code de résultat

Cas d'Utilisation

- **Analyse du Taux de Succès de Connexion** - Suivre les tentatives de connexion réussies vs échouées par code de résultat
- **Dépannage au Niveau de l'IMSI** - Identifier les abonnés rencontrant des échecs de connexion
- **Surveillance des Performances Réseau** - Surveiller les motifs de tentatives de connexion par origine (MME/SGSN)
- **Analyse du Roaming** - Analyser les taux de succès de connexion de roaming entrant

Intégration

Les métriques étendues sont exportées via l'intégration InfluxDB :

```
DRA.Metrics.InfluxDB.write(%{
  measurement: "attach_attempt_count",
  fields: %{imsi: "505057000000001"},
  tags: %{origin_host: "mme-01.example.com", result_code: 2001}
})
```

Les codes de résultat sont des codes Diameter standards :

- `2001` - Succès (DIAMETER_SUCCESS)
- `5001` - Échec d'authentification (DIAMETER_AUTHENTICATION_REJECTED)
- `5004` - AVP Diameter non pris en charge
- Voir RFC 6733 pour la liste complète des codes de résultat

Remarques Importantes

- Les métriques des tentatives de connexion ne suivent que les paires AIR/AIA S6a (Identifiant d'Application 16777251, Code de Commande 318)
- Les métadonnées de la demande expirent en fonction du délai d'attente de demande configuré + 5 secondes
- Le traitement des métriques est asynchrone (processus lancé) pour éviter de bloquer le flux de messages
- Le module fonctionne indépendamment des modules de routage et de transformation

---

Modules de Sécurité et de Pilotage

Le DRA est livré avec trois modules autonomes, configurables indépendamment pour la sécurité d'interconnexion et le contrôle du roaming. Chacun a son propre guide d'opérations dédié :

- **Sécurité Diameter** — Protection alignée sur GSMA FS.19/FS.21 : le Pare-feu Diameter (vérifications de format de bas niveau plus filtrage Catégorie 1/2/3), Masquage de Topologie (suppression de Route-Record et réécriture d'Origine), Limitation de Taux par pair, et Assainissement des AVP.

- **Pilotage de Roaming (SoR)** — Oriente les abonnés en roaming vers les VPLMNs préférés en rejetant les Demandes de Mise à Jour de Localisation provenant de réseaux non préférés.
- **Recherche d'Abonnés (SLF)** — Apprend dynamiquement les liaisons abonné-nœud de service à partir du trafic Diameter et route les demandes SLg/SLh vers le bon nœud de service.

Métriques Prometheus

Le DRA expose des métriques Prometheus complètes pour surveiller le trafic Diameter, la santé des pairs et les opérations des modules. Toutes les métriques sont disponibles à l'endpoint `/metrics`.

Métriques de Base Diameter

État des Pairs

Métrique : `diameter_peer_status` **Type** : Gauge **Description** : Si le pair est connecté (1) ou non (0) **Tags** :

- `origin_host` - Identité Diameter du pair
- `ip` - Adresse IP du pair

Exemple :

```
# Vérifier si un pair spécifique est connecté
diameter_peer_status{origin_host="hss01.example.com"}

# Compter les pairs déconnectés
count(diameter_peer_status == 0)
```

Compte de Messages

Métrique : `diameter_peer_message_count_total` **Type** : Counter **Description** : Nombre total de messages Diameter échangés avec les pairs **Tags** :

- `origin_host` - Identité Diameter du pair
- `received_from` - Pair d'où le message a été reçu
- `application_id` - Identifiant d'Application Diameter (voir référence Identifiant d'Application)
- `cmd_code` - Code de Commande Diameter (voir Codes de Commande Communs)
- `application_name` - Nom d'application lisible par l'homme (par exemple, "3GPP_S6a")
- `cmd_name` - Nom de commande lisible par l'homme (par exemple, "AIR")
- `direction` - "request" ou "response"

Exemple :

```
# Taux de demande AIR S6a d'un MME spécifique
rate(diameter_peer_message_count_total{
  cmd_code="318",
  direction="request",
  origin_host="mme01.example.com"
}[5m])

# Taux total de messages par application
sum by (application_name)
(rate(diameter_peer_message_count_total[5m]))
```

Codes de Résultat des Réponses

Métrique : `diameter_peer_message_result_code_count_total` **Type** : Counter **Description** : Nombre total de réponses Diameter par code de résultat **Tags** :

- `origin_host` - Demandeur d'origine
- `routed_to` - Pair qui a envoyé la réponse
- `application_id` - Identifiant d'Application Diameter
- `cmd_code` - Code de Commande Diameter
- `application_name` - Nom de l'application
- `cmd_name` - Nom de la commande
- `result_code` - Code de Résultat Diameter ou Code de Résultat Expérimental

Exemple :

```
# Taux de succès pour les demandes AIR S6a
rate(diameter_peer_message_result_code_count_total{
  cmd_code="318",
  result_code="2001"
}[5m])

# Taux d'erreur par code de résultat
sum by (result_code) (
  rate(diameter_peer_message_result_code_count_total{
    result_code!="2001"
  }[5m])
)
```

Codes de Résultat Communs :

- `2001` - DI

Direction de Roaming (SoR)

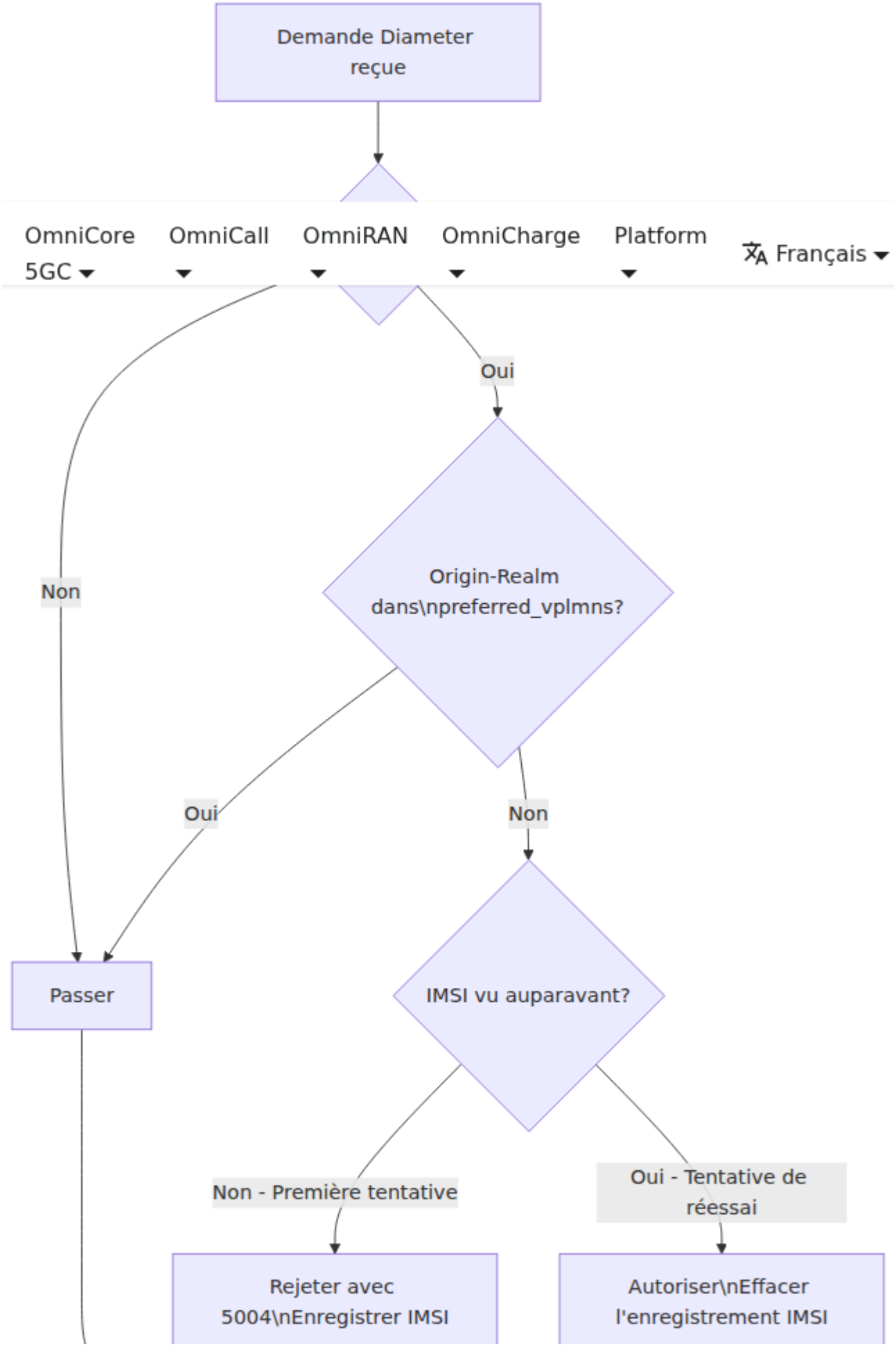
La direction de roaming (SoR) permet au HPLMN d'influencer le VPLMN auquel un abonné en roaming se connecte. Lorsque un abonné tente de s'enregistrer via un VPLMN non préféré, le DRA intercepte la demande S6a Update-Location-Request (ULR) et la rejette avec **DIAMETER_ERROR_ROAMING_NOT_ALLOWED (5004)**. Cela provoque le détachement de l'UE et une nouvelle tentative de connexion, idéalement en sélectionnant un VPLMN préféré.

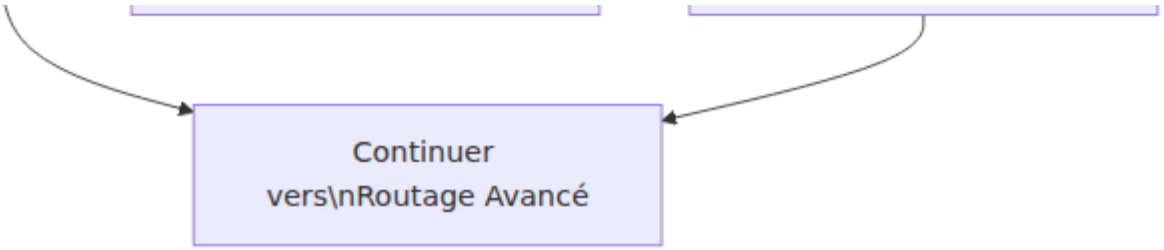
Si la couverture préférée n'est pas disponible et que l'abonné revient au même VPLMN non préféré, le module permet la demande lors de la tentative suivante.

Le SoR est défini dans 3GPP TS 29.272 Section 5.2.1.1 et 3GPP TS 23.122.

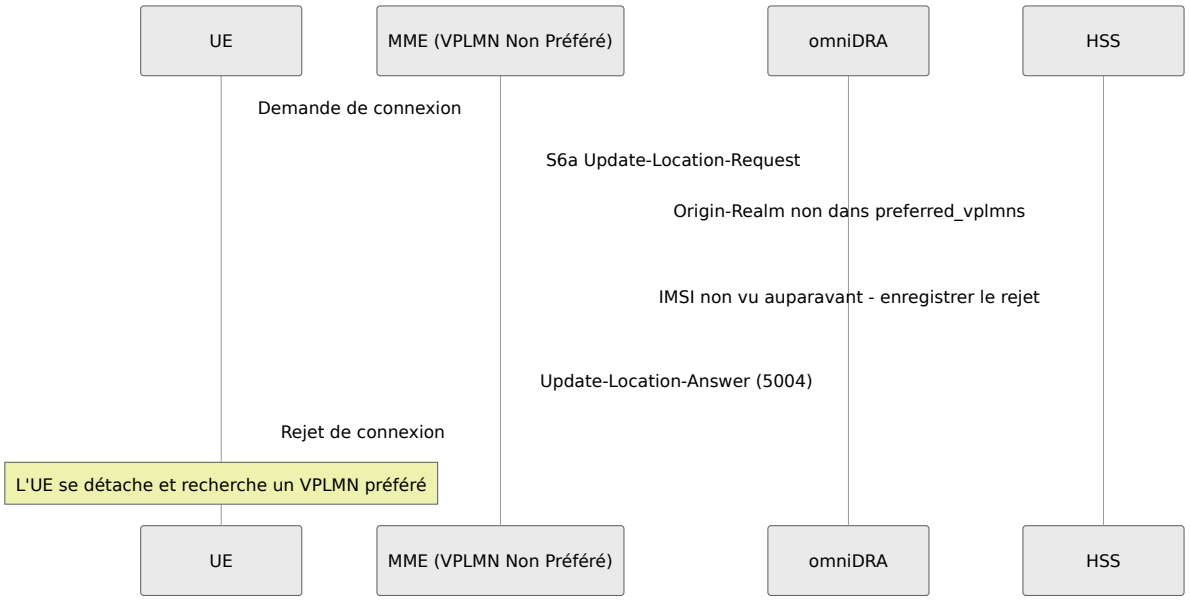
Comment ça fonctionne

Le module fonctionne dans le pipeline de traitement des demandes du DRA, s'exécutant **avant** le routage avancé. Seules les demandes S6a Update-Location-Requests (Application-Id `16777251`, Command-Code `316`) sont évaluées. Tous les autres messages Diameter passent sans modification.

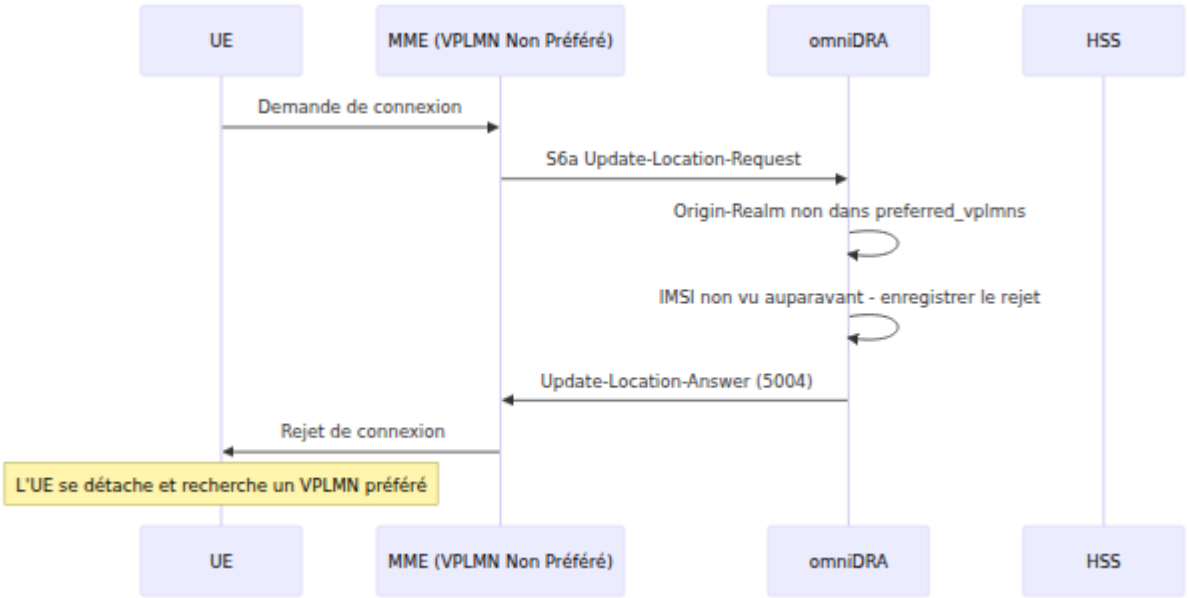

Demande Diameter reçue
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Français
Oui
Non
Origin-Realm dans\npreferred_vplmns?
Oui
Non
Passer
IMSI vu auparavant?
Non - Première tentative
Oui - Tentative de réessai
Rejeter avec 5004\nEnregistrer IMSI
Autoriser\nEffacer l'enregistrement IMSI



Séquence : Première Tentative (Rejetée)



Séquence : Deuxième Tentative (Autorisée)



Position dans le Pipeline

La direction de roaming s'exécute **avant** le routage avancé dans le pipeline de traitement des demandes. Si le SoR rejette une demande, le routage avancé n'est jamais atteint pour ce message.

Suivi des Abonnés

Le module suit chaque abonné rejeté par IMSI. Chaque enregistrement comprend un compteur de rejets et un horodatage. Les enregistrements sont automatiquement nettoyés après l'expiration du TTL configuré.

Lorsqu'un IMSI suivi a atteint le seuil de `max_rejections` et envoie un autre ULR, la demande est autorisée et l'enregistrement de suivi est supprimé.

Si aucun autre ULR n'arrive dans la fenêtre TTL, l'enregistrement expire et est supprimé lors du prochain cycle de nettoyage. Un ULR ultérieur du même IMSI sera traité comme une première tentative et rejeté à nouveau.

Configuration

Le module est configuré sous `module_roaming_steering` dans `config/runtime.exs`.

```
module_roaming_steering: %{
  # Activer ou désactiver le module
  enabled: false,

  # Nombre de rejets avant d'autoriser
  max_rejections: 1,

  # Temps en secondes avant qu'un enregistrement IMSI suivi
n'expire
  rejection_ttl_seconds: 300,

  # Code de résultat Diameter renvoyé dans la réponse de rejet
  rejection_result_code: 5004,

  # Liste des valeurs Origin-Realm considérées comme VPLMNs
préférés
  # Les ULR de ces royaumes sont toujours autorisés
  preferred_vplmns: [
    "epc.mnc001.mcc001.3gppnetwork.org",
    "epc.mnc002.mcc001.3gppnetwork.org"
  ]
}
```

Paramètres

Paramètre	Type	Requis	Par défaut	De
`enabled`	Boolean	Oui	`false`	Activer ou désactiv de roaming. Lorsqu demandes passent
`max_rejections`	Integer	Non	`1`	Nombre de fois à r d'autoriser. Avec la le premier ULR est autorisé.
`rejection_ttl_seconds`	Integer	Non	`300`	Temps en seconde enregistrement IM nettoyé. Si l'abonn cette fenêtre, sa p traitée comme une Contrôle également nettoyage.
`rejection_result_code`	Integer	Non	`5004`	Code de résultat D réponse de rejet. `5` DIAMETER_ERROR_ selon 3GPP TS 29.2
`preferred_vplmns`	Liste	Non	`[]`	Liste de chaînes O VPLMNs préférés. I pairs avec un Orig sont toujours auto

Exemples de Configuration

Rejeter Une Fois, Puis Autoriser

Le comportement par défaut. Les abonnés se connectant à un VPLMN non préféré sont rejetés une fois. S'ils reviennent, l'ULR est transféré au HSS.

```
module_roaming_steering: %{
  enabled: true,
  max_rejections: 1,
  rejection_ttl_seconds: 300,
  rejection_result_code: 5004,
  preferred_vplmns: [
    "epc.mnc001.mcc310.3gppnetwork.org",
    "epc.mnc002.mcc310.3gppnetwork.org"
  ]
}
```

Comment ça fonctionne : Un ULR arrivant d'un MME dans `epc.mnc003.mcc310.3gppnetwork.org` (non dans la liste préférée) est rejeté avec le Code de Résultat 5004. L'UE se détache et tente de trouver un réseau préféré. S'il revient au même VPLMN non préféré dans les 300 secondes, l'ULR est transféré normalement.

Cas d'utilisation : Direction de roaming standard où l'opérateur a des accords de roaming préférés avec des réseaux partenaires spécifiques.

Rejets Multiples Avant Autorisation

Pour des scénarios où l'opérateur souhaite que l'UE effectue plusieurs tentatives avant de revenir à un VPLMN non préféré.

```
module_roaming_steering: %{
  enabled: true,
  max_rejections: 3,
  rejection_ttl_seconds: 600,
  rejection_result_code: 5004,
  preferred_vplmns: [
    "epc.mnc001.mcc310.3gppnetwork.org"
  ]
}
```

Comment ça fonctionne : L'abonné est rejeté trois fois avant d'être autorisé lors de la quatrième tentative. Le TTL est prolongé à 600 secondes pour tenir compte des cycles de réessai supplémentaires.

Cas d'utilisation : Zones urbaines denses où une couverture préférée existe mais peut nécessiter plusieurs tentatives de connexion pour acquérir.

Métriques

Rejets

Métrique: `diameter.roaming_steering.reject.count` **Type**: Compteur **Description**: Incrémenté chaque fois qu'un ULR est rejeté par le module SoR. **Étiquettes**:

- `origin_realm` - Origin-Realm du VPLMN non préféré
- `result_code` - Code de résultat Diameter renvoyé dans le rejet
- `imsi` - IMSI de l'abonné rejeté

Autorisé (VPLMN Préféré)

Métrique: `diameter.roaming_steering.allow.count` **Type**: Compteur **Description**: Incrémenté lorsqu'un ULR est autorisé. **Étiquettes**:

- `origin_realm` - Origin-Realm du VPLMN d'origine
- `reason` - Pourquoi la demande a été autorisée : `preferred_vplmn` ou `max_rejections_reached`
- `imsi` - IMSI de l'abonné (présent uniquement lorsque la raison est `max_rejections_reached`)

Réponses d'Erreur

Métrique: `diameter.roaming_steering.error.count` **Type**: Compteur **Description**: Incrémenté lorsque le processeur renvoie un message de réponse d'erreur en raison d'un rejet SoR. **Étiquettes**:

- `result_code` - Code de résultat Diameter
- `application_id` - Identifiant d'application Diameter (numérique)
- `cmd_code` - Code de commande Diameter (numérique)
- `application_name` - Nom d'application lisible par l'homme
- `cmd_name` - Nom de commande lisible par l'homme

Dépannage

Abonnés Toujours Rejetés (Jamais Autorisés)

Symptômes: Les abonnés sur des VPLMNs non préférés sont systématiquement rejetés et ne parviennent jamais à se connecter avec succès.

Causes possibles:

- `rejection_ttl_seconds` est trop court, ce qui entraîne l'expiration de l'enregistrement de suivi avant que l'abonné ne réessaie
- `max_rejections` est défini plus haut que le nombre de réessais que l'UE effectue avant d'abandonner

Résolution:

1. Augmenter `rejection_ttl_seconds` pour tenir compte du timing de réessai de l'UE
2. Vérifier que `max_rejections` est défini à une valeur que l'UE peut raisonnablement atteindre dans son cycle de réessai

Abonnés sur VPLMNs Préférés Rejetés

Symptômes: Les ULR des réseaux partenaires préférés sont rejetés avec 5004.

Causes possibles:

- L'Origin-Realm du VPLMN préféré n'est pas répertorié dans `preferred_vplmns`

- La chaîne Origin-Realm ne correspond pas exactement (sensible à la casse)

Résolution:

1. Vérifier l'Origin-Realm du pair dans les journaux DRA au moment de la connexion
2. S'assurer que la chaîne Origin-Realm exacte est incluse dans la liste `preferred_vplmns`

Module Ne Prend Pas Effet

Symptômes: Les ULR des VPLMNs non préférés sont transférés sans rejet.

Causes possibles:

- `enabled` est défini sur `false`
- Le processus du module n'est pas en cours d'exécution (vérifier le superviseur)

Résolution:

1. Vérifier `enabled: true` dans la configuration
2. Confirmer que le DRA a été redémarré après le changement de configuration

Référence

Codes Diameter

Code	Nom	Description	Référ
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	Abonné non autorisé à roaming dans ce VPLMN	3GPP 29.27 Sectic 7.4.4

Commandes S6a

Command-Code	Nom	Description	Référence
316	Update-Location-Request/Answer (ULR/ULA)	Envoyé par le MME au HSS lors de la connexion pour mettre à jour le nœud de service de l'abonné	3GPP TS 29.272 Section 7.2.3

Identifiants d'Application

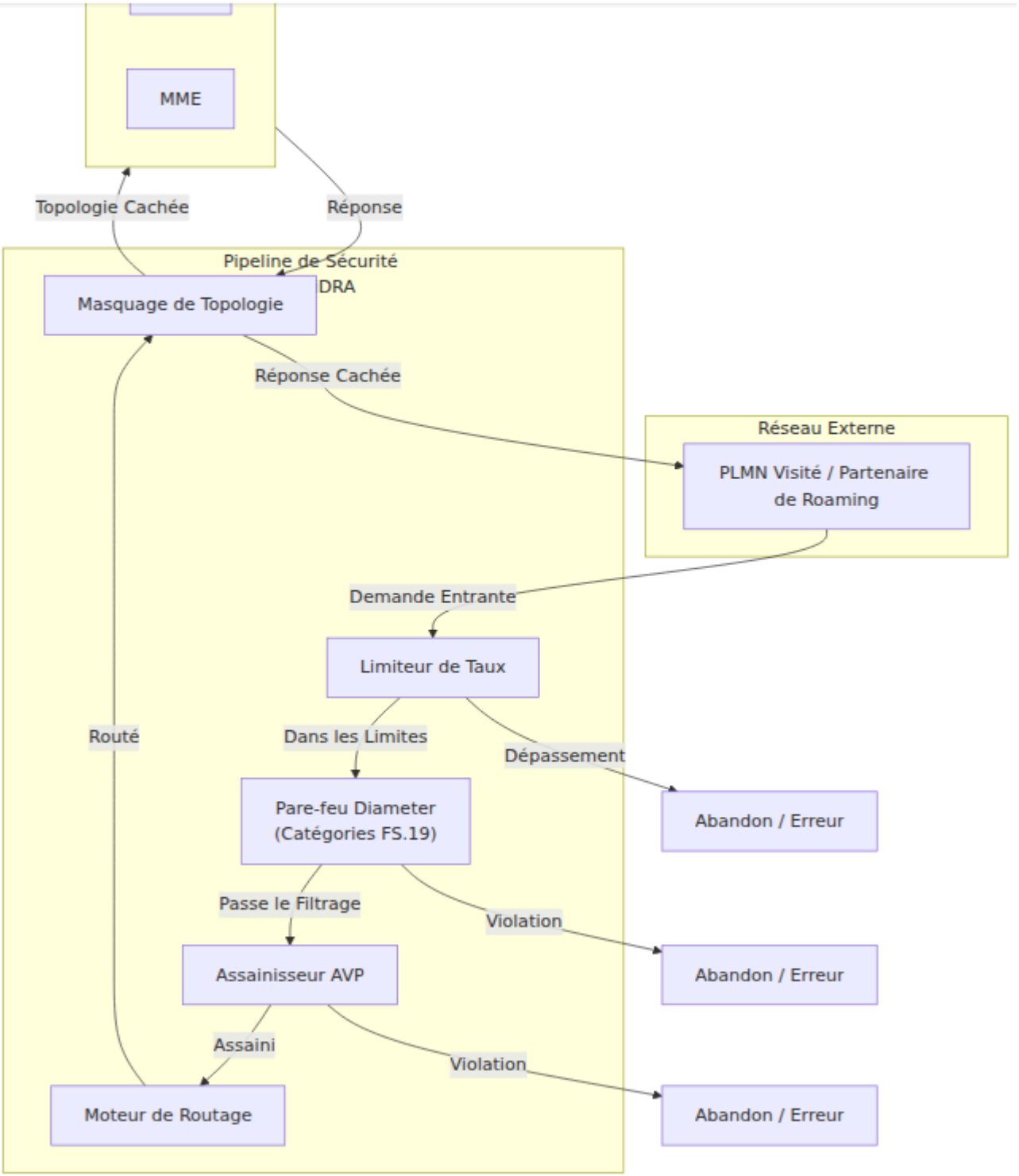
ID	Interface	Description	Référence
16777251	S6a/S6d	Authentification MME/SGSN et gestion des abonnements HSS	3GPP TS 29.272

Sécurité Diameter

OmniDRA fournit une suite complète de modules de sécurité Diameter alignés avec **GSMA FS.19** (Sécurité d'interconnexion Diameter, v10.0) et **GSMA FS.21** (Recommandations de sécurité pour le signalement d'interconnexion, v12.0). Ces modules protègent le réseau contre les attaques au niveau du signal sur les interfaces d'interconnexion Diameter.

Chaque module est configurable de manière indépendante, peut être activé ou désactivé sans affecter les autres, et émet ses propres événements de télémétrie pour la surveillance et l'alerte.

Vue d'ensemble de l'architecture



Ordre de Traitement

Les demandes Diameter entrantes passent par les modules de sécurité dans l'ordre suivant. Un message rejeté à n'importe quelle étape n'est jamais transmis aux étapes suivantes.

Ordre	Module	Direction	Objectif
1	Limiteur de Taux	Entrant	Protection contre les inondations volumétriques
2	Pare-feu Diameter	Entrant	Filtrage de protocole et de contenu FS.19
3	Assainisseur AVP	Entrant	Validation et suppression des AVP
4	Moteur de Routage	-	Routage Diameter standard
5	Masquage de Topologie	Sortant	Dissimulation de la topologie du réseau

Actions

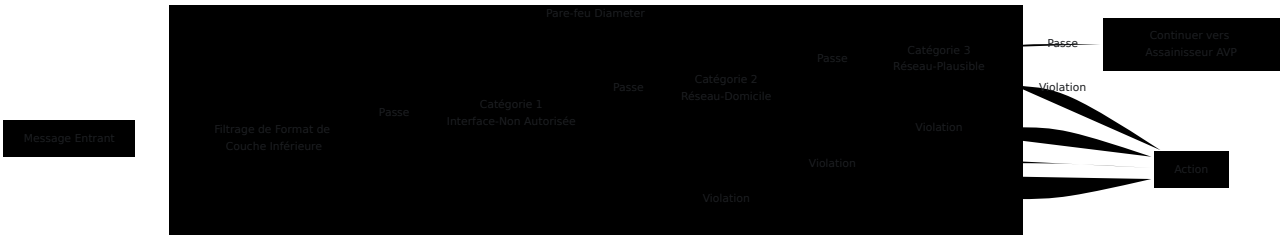
Tous les modules de sécurité prennent en charge des actions configurables lorsqu'une violation est détectée. Les actions sont configurées par module (et par catégorie pour le Pare-feu Diameter).

Action	Comportement	Résultat Diameter
`{:error, 3002}`	Répondre avec une réponse d'erreur Diameter	DIAMETER_UNABLE_TO_DELIVER
`{:error, 3004}`	Répondre avec une réponse d'erreur Diameter	DIAMETER_TOO_BUSY
`{:error, 3007}`	Répondre avec une réponse d'erreur Diameter	DIAMETER_APPLICATION_UNSUPPORTED
`:drop`	Jeter silencieusement le message	Aucune réponse envoyée
`:log_only`	Journaliser la violation mais permettre le passage du message	Le message continue

---

Pare-feu Diameter

Le Pare-feu Diameter implémente les quatre couches de filtrage définies dans **GSMA FS.19 Section 3.3** et la catégorisation des paquets indépendante du protocole de **FS.21 Section 7**. Les messages sont évalués à travers chaque couche dans l'ordre ; une violation à n'importe quelle couche arrête le traitement ultérieur.



Filtrage de Format de Couche Inférieure

Référence GSMA : FS.19 Section 3.3.4, FS.21 Section 7.3.1

Le filtrage de couche inférieure détecte les violations au niveau du protocole et les tentatives de spoofing de base sans avoir besoin de comprendre la sémantique des applications de couche supérieure. Cette couche attrape les messages mal formés avant qu'ils n'atteignent une inspection plus approfondie.

Contrôles effectués :

Contrôle	Référence FS.19	Description
Duplication AVP	Section 3.3.4, 4.8.1	Détecte les instances dupliquées d'AVP qui doivent apparaître au maximum une fois (par exemple, Origin-Host, Origin-Realm). Empêche les attaques d'évasion par doublement d'AVP.
Ordre de Session-Id	Section 3.3.4	Valide que le Session-Id (AVP 263) est le premier AVP dans le message, conformément à RFC 6733 Section 8.8.
Destination-Host dans les Réponses	Section 3.3.4	Rejette les messages de réponse contenant un AVP Destination-Host, ce qui constitue une violation de protocole pouvant indiquer une évasion de filtre.
Validation des AVP Obligatoires	Section 3.3.5	Valide que les AVP requis sont présents pour chaque code de commande (par exemple, ULR doit contenir User-Name et Visited-PLMN-Id conformément à 3GPP TS 29.272 Section 7.2.3).

```
low_layer: %{
  enabled: true,
  action: {:error, 3002},
  # Codes AVP qui ne doivent pas apparaître plus d'une fois (FS.19
Section 3.3.4)
  # 264 = Origin-Host, 296 = Origin-Realm, 283 = Destination-
Realm, 293 = Destination-Host
  single_instance_avps: [264, 296, 283, 293],
  # Session-Id doit être le premier AVP (RFC 6733 Section 8.8)
  enforce_session_id_first: true,
  # Les messages de réponse ne doivent pas contenir Destination-
Host (FS.19 Section 3.3.4)
  reject_destination_host_in_answers: true,
  # AVP obligatoires par code de commande (3GPP TS 29.272)
  # 1 = User-Name (IMSI), 1407 = Visited-PLMN-Id, 264 = Origin-
Host, 296 = Origin-Realm
  mandatory_avps: %{
    316 => [1, 1407, 264, 296],
    318 => [1, 1407, 264, 296]
  }
}
```

Catégorie 1 — Filtrage de Paquets Interface-Non Autorisée

Référence GSMA : FS.19 Section 3.3.5, FS.21 Section 7.2.1

Le filtrage de la catégorie 1 garantit que seuls les ID d'application Diameter et les codes de commande autorisés sont acceptés sur chaque interface d'interconnexion. Cela empêche l'accès externe aux interfaces réservées à l'interne (par exemple, bloquer les commandes Sh sur une interface S6a) et impose que les partenaires de roaming n'envoient que des types de messages couverts par leurs accords de roaming.

L'approche de liste blanche suit la recommandation FS.19 : **bloquer tous les messages Diameter sauf ceux explicitement requis pour une interface donnée**.

Les listes blanches peuvent être configurées globalement (s'appliquant à tous les pairs) ou par pair, permettant à différents partenaires de roaming d'avoir

différents ensembles de messages autorisés.

```
category_1: %{
  enabled: true,
  action: {:error, 3007},
  whitelists: %{
    # Liste blanche par pair – restreindre ce partenaire
uniquement à ULR et AIR (FS.19 Section 3.3.5)
    "restricted-partner.roaming.com" => %{
      16_777_251 => [316, 318]
    },
    # Liste blanche par défaut – commandes S6a standard autorisées
pour tous les autres pairs
    all: %{
      # S6a/S6d (FS.19 Section 3.3.5, Tableau 2)
      16_777_251 => [316, 317, 318, 319, 320, 321, 323],
      # S13 – Vérification d'Identité ME (FS.19 Section 3.3.5)
      16_777_252 => [324],
      # S6c – SMS via HSS (FS.19 Section 3.3.5.1)
      16_777_312 => [8388647, 8388648],
      # SGd – SMS via MME (FS.19 Section 3.3.5.2)
      16_777_313 => [8388645, 8388646]
    }
  }
}
```

Liste Blanche S6a/S6d Commune :

Code de Commande	Nom	Direction	Référence
316	Demande/Réponse de Mise à Jour de Localisation	MME → HSS	3GPP TS 29.272 §7.2.3
317	Demande/Réponse d'Annulation de Localisation	HSS → MME	3GPP TS 29.272 §7.2.7
318	Demande/Réponse d'Information d'Authentification	MME → HSS	3GPP TS 29.272 §7.2.5
319	Demande/Réponse de Données d'Abonné	HSS → MME	3GPP TS 29.272 §7.2.9
320	Demande/Réponse de Suppression de Données d'Abonné	HSS → MME	3GPP TS 29.272 §7.2.11
321	Demande/Réponse de Purge-UE	MME → HSS	3GPP TS 29.272 §7.2.13
323	Demande/Réponse de Notification	MME → HSS	3GPP TS 29.272 §7.2.15

IDs d'Application Communs pour l'interconnexion de roaming (FS.19 Section 3.3.5) :

ID d'Application	Interface	Référence
16777251	S6a/S6d	3GPP TS 29.272
16777252	S13	3GPP TS 29.272
16777312	S6c	3GPP TS 29.338
16777313	SGd	3GPP TS 29.338
16777255	SLg	3GPP TS 29.172
16777267	S9	3GPP TS 29.215

Catégorie 2 — Filtrage de Paquets Réseau-Domicile

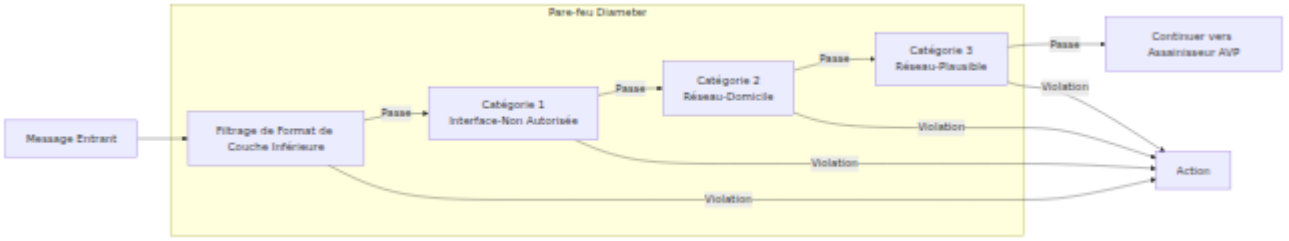
Référence GSMA : FS.19 Section 3.3.6, FS.21 Section 7.2.2

Le filtrage de la catégorie 2 protège les abonnés domicile contre les messages provenant de l'interconnexion. Les messages sur les codes de commande protégés sont inspectés pour l'identité de l'abonné (IMSI dans l'AVP User-Name, MSISDN dans l'AVP 701). Si l'identité correspond à un préfixe d'abonné domicile, le message est rejeté — le trafic légitime pour les abonnés domicile doit provenir de l'intérieur du réseau domicile, et non de pairs externes.

Cela empêche les attaques où une entité externe envoie des mises à jour de localisation, des demandes de données d'abonné ou des requêtes d'authentification ciblant les propres abonnés de l'opérateur.

```
# Niveau supérieur : définir les préfixes d'abonné domicile
home_imsi_prefixes: ["31338", "31339"],
home_msisdn_prefixes: ["+1313"],

category_2: %{
  enabled: true,
  action: {:error, 3002},
  # Codes de commande S6a qui portent l'identité de l'abonné
(FS.19 Section 3.3.6, Tableau 12)
  protected_command_codes: [316, 317, 318, 319, 320, 321, 323]
}
```



Catégorie 3 — Filtrage de Paquets Réseau-Plausible

Référence GSMA : FS.19 Section 3.3.7, FS.21 Section 7.2.3

Le filtrage de la catégorie 3 détecte les changements de localisation peu plausibles en suivant le dernier réseau vu pour chaque abonné (IMSI). Lorsqu'une Demande de Mise à Jour de Localisation arrive d'un réseau visité différent de celui précédent, le temps écoulé est comparé à un seuil configurable. Un changement de domaine se produisant plus rapidement que physiquement possible indique une attaque potentielle (par exemple, des mises à jour de localisation falsifiées).

Ce module maintient un suivi d'état par IMSI en utilisant une table ETS avec nettoyage automatique des entrées obsolètes (les entrées de plus de 24 heures sont supprimées périodiquement).

Contrôles effectués :

Contrôle	Référence FS.19	Description
Vérification de l'Emplacement Précédent	Section 3.3.7.1	Compare l'Origin-Realm de l'ULR actuel avec le dernier Origin-Realm vu pour cet IMSI
Vérification de Vitesse/Temps	Section 3.3.7.2	Signale les changements de domaine qui se produisent dans une fenêtre de temps minimale configurable

```
category_3: %{
  enabled: true,
  # Commencer avec log_only pour observer les modèles avant
d'appliquer (FS.19 Section 3.3.7)
  action: :log_only,
  # Vitesse maximale plausible en km/h (FS.19 Section 3.3.7.2)
  max_velocity_kmh: 1200,
  # Temps minimum entre les ULR de différents domaines pour le
même IMSI
  min_time_between_updates_seconds: 2
}
```

Configuration

Le Pare-feu Diameter est activé au niveau supérieur, chaque couche de filtrage étant configurée indépendamment. Les extraits de configuration montrés ci-dessus dans chaque section de catégorie sont combinés sous une seule clé `module_diameter_firewall` :

```
config :dra,
  module_diameter_firewall: %{
    enabled: true,
    home_imsi_prefixes: ["31338", "31339"],
    home_msisdn_prefixes: ["+1313"],
    low_layer: %{ ... },      # Voir Filtrage de Format de Couche
Inférieure ci-dessus
    category_1: %{ ... },     # Voir Catégorie 1 ci-dessus
    category_2: %{ ... },     # Voir Catégorie 2 ci-dessus
    category_3: %{ ... }      # Voir Catégorie 3 ci-dessus
  }
```

Paramètres de Niveau Supérieur

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Booléen	Oui	`false`	Activer ou désactiver l'ensemble du module de Pare-feu Diameter. Lorsque `false`, tous les messages passent sans inspection.
`home_imsi_prefixes`	Liste	Non	`[]`	Liste de chaînes de préfixes IMSI identifiant les abonnés domicile. Utilisé par le filtrage de la Catégorie 2. Exemple : `["31338", "31339"]`.
`home_msisdn_prefixes`	Liste	Non	`[]`	Liste de chaînes de préfixes MSISDN identifiant les abonnés domicile. Utilisé par le filtrage de la Catégorie 2. Exemple : `["+1313"]`.

Paramètres de Couche Inférieure

Paramètre	Type	Requis	Par Défaut	D
`enabled`	Booléen	Non	`true`	A d fi fc c in
`action`	Action	Non	`{:error, 3002}`	A e lo vi c in es d
`single_instance_avps`	Liste	Non	`[264, 296, 283, 293]`	C q d a pl fc m va d c O (2 O (2 D R

Paramètre	Type	Requis	Par Défaut	D
				et D H
`enforce_session_id_first`	Booléen	Non	`true`	R m ou S n' pr
`reject_destination_host_in_answers`	Booléen	Non	`true`	R m d c u D H
`mandatory_avps`	Carte	Non	`%{}`	C c c lis c re E {3 14 29 U V Pl O et R

Paramètre	Type	Requis	Par Défaut	D
				le m U

Paramètres de Catégorie 1

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Booléen	Non	`true`	Activer ou désactiver le filtra Catégorie 1.
`action`	Action	Non	`{:error, 3007}`	Action à entreprendre lorsqu' violation de la Catégorie 1 es La valeur par défaut répond DIAMETER_APPLICATION_UNS
`whitelists`	Carte	Oui	-	Carte de nom d'hôte de pair vers les combinaisons ID d'ap Code de commande autorisé `:all` fournit une liste blanch défaut. Les entrées spécifiqu pairs ont la priorité.

Paramètres de Catégorie 2

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Booléen	Non	`true`	Activer ou désactiver le filtrage de la Catégorie 2.
`action`	Action	Non	`{:error, 3002}`	Action à entreprendre lorsqu'un abonné domic est ciblé depu l'interconnexio
`protected_command_codes`	Liste	Non	`[]`	Codes de commande qu déclenchent d vérifications d'identité d'abonné domicile. Typiquement l codes de commande S6 qui portent l'identité de l'abonné.

Paramètres de Catégorie 3

Paramètre	Type	Requis	Par Défaut	
`enabled`	Booléen	Non	`false`	Ac dé filt Ca Né pr Lo so d'e (d au pa su lor ac
`action`	Action	Non	`:log_only`	Ac en lor co pla loc éc Re co `:l` aju av d'a
`max_velocity_kmh`	Entier	Non	`1200`	Vit pla

Paramètre	Type	Requis	Par Défaut	
				km po ca dis gé
`min_time_between_updates_seconds`	Entier	Non	`2`	Te en en jou loc dif do mé ch do ra so

Événements de Télémétrie

Événement	Description
`[:diameter, :firewall, :low_layer, :block]`	Violation de format de couche inférieure détectée et bloquée
`[:diameter, :firewall, :category_1, :block]`	Violation de la Catégorie 1 détectée et bloquée
`[:diameter, :firewall, :category_2, :block]`	Violation de la Catégorie 2 détectée et bloquée
`[:diameter, :firewall, :category_3, :block]`	Violation de la Catégorie 3 détectée et bloquée
`[:diameter, :firewall, :pass, :count]`	Message passé tous les contrôles du pare-feu

Tous les événements incluent des métadonnées : `origin_host`, `application_id`, `command_code`, `application_name`, `command_name`, et `reason`.

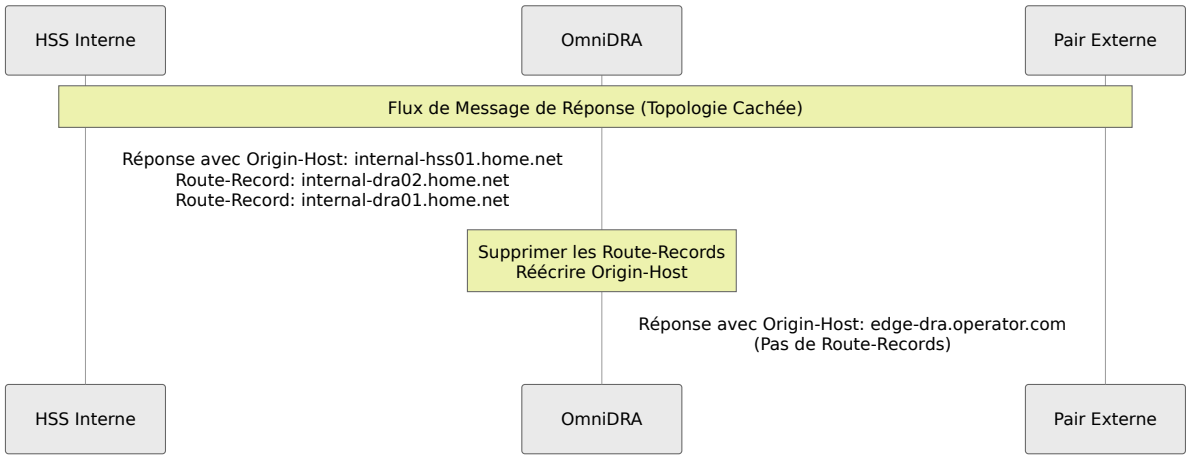
Masquage de Topologie

Référence GSMA : FS.19 Section 2.4, Section 3.4 ; FS.21 Section 3.6

Le module de Masquage de Topologie empêche l'exposition de la topologie interne du réseau aux pairs externes. Lorsque les messages Diameter traversent plusieurs nœuds internes, ils accumulent des AVP Route-Record et portent des valeurs Origin-Host/Origin-Realm qui révèlent des noms d'hôtes internes, la structure du réseau et le nombre de nœuds. Ces informations peuvent être utilisées par des attaquants pour cartographier le réseau interne pour des attaques ciblées.

Le masquage de topologie opère sur le **chemin sortant** — après que les décisions de routage ont été prises, avant que les messages ne soient transmis

au pair de destination.



Fonctionnalités

Fonctionnalité	Référence FS.19	Description
Suppression de Route-Record	Section 2.4	Supprime tous les AVP Route-Record (282) qui révèlent les chemins de routage internes et les noms d'hôtes des nœuds
Réécriture d'Origin-Host	Section 3.4	Remplace l'AVP Origin-Host par l'identité du DRA (ou une valeur personnalisée) sur les messages de réponse
Réécriture d'Origin-Realm	Section 3.4	Remplace optionnellement l'AVP Origin-Realm pour cacher la structure de domaine interne
Contrôle par Pair	-	Appliquer le masquage de topologie de manière sélective — uniquement aux pairs externes, ou à tous les pairs

```
module_topology_hiding: %{
  enabled: true,
  # Supprimer les AVP Route-Record révélant des chemins internes
(FS.19 Section 2.4)
  strip_route_records: true,
  # Réécrire Origin-Host sur les réponses pour cacher les noms de
nœuds internes (FS.19 Section 3.4)
  rewrite_origin_host: %{enabled: true, replacement: :self},
  # Optionnellement cacher la structure de domaine interne
  rewrite_origin_realm: %{enabled: false, replacement: :self},
  # Appliquer à tous les pairs externes, ou lister des noms
d'hôtes spécifiques
  external_peers: :all
}
```

Paramètres

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Booléen	Oui	`false`	Activer ou désactiver le module de Masquage de Topologie.
`strip_route_records`	Booléen	Non	`true`	Supprimer tous les AVP Route-Record (code 282) des messages avant de les transmettre aux pairs externes.
`rewrite_origin_host`	Carte	Non	Voir ci-dessous	Contrôle la réécriture d'Origin-Host sur les messages de réponse.
`rewrite_origin_realm`	Carte	Non	Voir ci-dessous	Contrôle la réécriture d'Origin-Realm sur les messages de réponse.
`external_peers`	`:all` ou Liste	Non	`:all`	Pairs considérés comme externes. Le masquage de topologie n'est

Paramètre	Type	Requis	Par Défaut	Description
				appliqué qu'aux messages destinés à ces pairs. Utilisez `:all` pour les déploiements d'interconnexion. Utilisez une liste de noms d'hôtes pour une application sélective.

Paramètres de Réécriture (s'appliquent à la fois à `rewrite_origin_host` et `rewrite_origin_realm`) :

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Booléen	Non	varie	Activer la réécriture pour cet AVP. La valeur par défaut est `true` pour Origin-Host, `false` pour Origin-Realm.
`replacement`	`:self` ou Chaîne	Non	`:self`	Valeur de remplacement. `:self` utilise l'identité propre du DRA à partir de la configuration `diameter` (`host.realm`). Une valeur de chaîne est utilisée telle quelle.

Événements de Télémétrie

Événement	Description
`[:diameter, :topology_hiding, :route_record, :stripped]`	AVP Route-Record supprimés. La mesure inclut le `count` des AVP supprimés.
`[:diameter, :topology_hiding, :origin_host, :rewritten]`	AVP Origin-Host réécrit
`[:diameter, :topology_hiding, :origin_realm, :rewritten]`	AVP Origin-Realm réécrit

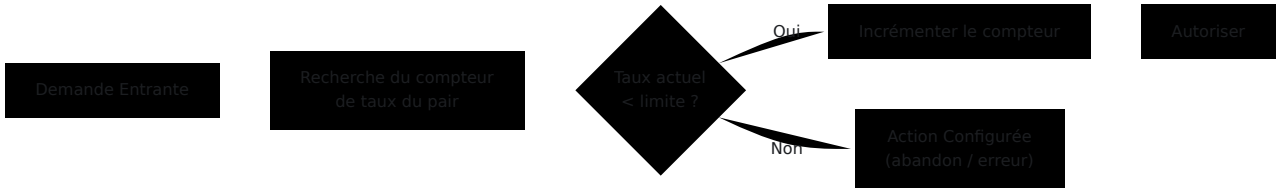
Limiteur de Taux

Référence GSMA : FS.19 Section 3.4 (Disponibilité / Protection contre DoS)

Le Limiteur de Taux impose des limites de taux de message par pair pour protéger contre les attaques volumétriques et les inondations de messages. Il fonctionne comme le premier contrôle de sécurité dans le pipeline — avant tout parsing de message ou inspection de contenu — pour réduire la charge excédentaire le plus tôt possible.

Les limites de taux sont suivies par pair en utilisant un compteur à fenêtre glissante. Chaque pair a un compteur indépendant qui se réinitialise chaque seconde.

```
module_rate_limiter: %{
  enabled: true,
  # Limite par défaut pour tous les pairs (FS.19 Section 3.4)
  default_max_requests_per_second: 1000,
  default_action: {:error, 3004},
  # Remplacements par pair
  peer_limits: %{
    "high-volume-partner.roaming.com" => %
{max_requests_per_second: 5000, action: {:error, 3004}},
    "restricted-peer.roaming.com" => %{max_requests_per_second:
100, action: :drop}
  }
}
```



Paramètres

Paramètre	Type	Requis	Par Défaut	
`enabled`	Booléen	Oui	`false`	Activ le mo Taux.
`default_max_requests_per_second`	Entier	Non	`1000`	Nomb dema secon parti quel
`default_action`	Action	Non	`{:error, 3004}`	Actio lorsq dépa taux. `3004` DIAM
`peer_limits`	Carte	Non	`%{}`	Carte de pa remp confi pairs utilis par d

Paramètres de Remplacement par Pair :

Paramètre	Type	Requis	Par Défaut	Description
`max_requests_per_second`	Entier	Non	Hérite du défaut	Nombre maximum de demandes par seconde pour ce pair spécifique.
`action`	Action	Non	Hérite du défaut	Action lorsque ce pair dépasse sa limite de taux.

Événements de Télémétrie

Événement	Description
`[:diameter, :rate_limiter, :throttled]`	Un message a été limité par le taux. Les mesures incluent `current_rate` et `limit`.
`[:diameter, :rate_limiter, :allowed]`	Un message était dans les limites de taux.

Assainisseur AVP

Référence GSMA : FS.19 Section 3.3.4, 4.8.1, 4.8.2 ; FS.21 Section 3.6

L'Assainisseur AVP valide et assainit les AVP Diameter à la frontière de l'interconnexion. Il répond aux recommandations de la Section 3.6 de FS.21 pour traiter les attaques de manipulation au niveau du protocole qui opèrent en dessous du niveau des catégories de filtrage FS.19.

Fonctionnalités

Fonctionnalité	Référence GSMA	Description
Suppression d'AVP de Vendeur Inconnu	FS.21 Section 3.6	Supprime les AVP provenant de vendeurs non listés. Empêche l'injection d'AVP propriétaires qui pourraient déclencher un comportement inattendu dans les nœuds backend.
Profondeur de Nesting d'AVP Groupés	FS.21 Section 3.6	Imposer une profondeur de nesting maximale pour les AVP groupés. Empêche les attaques de débordement de pile utilisant des structures AVP profondément imbriquées.

```
module_avp_sanitizer: %{
  enabled: true,
  action: {:error, 3002},
  # N'autoriser que les AVP de vendeur standard sur
l'interconnexion (FS.21 Section 3.6)
  # 0 = IETF, 10415 = 3GPP, 13019 = ETSI, 5535 = 3GPP2
  allowed_vendor_ids: [0, 10415, 13019, 5535],
  strip_unknown_vendor_avps: true,
  # Prévenir le débordement de pile via des AVP groupés
profondément imbriqués (FS.21 Section 3.6)
  max_avp_nesting_depth: 10
}
```

Paramètres

Paramètre	Type	Requis	Par Défaut	Descri
`enabled`	Booléen	Oui	`false`	Activer ou d le module Assainisseu
`action`	Action	Non	`{:error, 3002}`	Action à entreprendr lorsqu'une v de profonde nesting est La suppress vendeur élir silencieuser AVP offensa bloquer le m
`allowed_vendor_ids`	Liste	Non	`[0, 10415, 13019, 5535]`	Liste des ID vendeur aut sur l'interco Les AVP d'au vendeurs sc supprimés.
`strip_unknown_vendor_avps`	Booléen	Non	`true`	Activer la suppression provenant d vendeurs nc figurant dar `allowed_ve`
`max_avp_nesting_depth`	Entier	Non	`10`	Profondeur autorisée de d'AVP group

Paramètre	Type	Requis	Par Défaut	Descri
				messages d cela sont so l'action conf

IDs de Vendeur Communs :

ID de Vendeur	Organisation	Remarques
0	IETF	AVP Diameter standard définis dans les RFC
10415	3GPP	Tous les AVP définis par la 3GPP (S6a, Gx, Rx, etc.)
13019	ETSI	AVP définis par l'ETSI
5535	3GPP2	AVP définis par la 3GPP2 (interopérabilité CDMA)

Événements de Télémétrie

Événement	Description
`[:diameter, :avp_sanitizer, :unknown_vendor, :stripped]`	Un ou plusieurs AVP de vendeurs inconnus ont été supprimés
`[:diameter, :avp_sanitizer, :nesting_depth, :violation]`	Un message a dépassé la profondeur maximale de nesting d'AVP
`[:diameter, :avp_sanitizer, :pass, :count]`	Le message a passé tous les contrôles d'assainissement

Recommandations de Déploiement

Ordre de Activation Recommandé

Lors du déploiement des modules de sécurité pour la première fois, activez-les progressivement pour éviter de perturber le trafic en direct :

1. **Limiteur de Taux** — Commencez avec des limites généreuses et surveillez les modèles de trafic. Resserrez les limites une fois que les taux de base sont compris.
2. **Assainisseur AVP** — Risque faible de faux positifs. Activez la suppression et les vérifications de nesting.
3. **Pare-feu Diameter (Catégorie 1)** — Définissez des listes blanches basées sur des accords de roaming. Commencez avec des combinaisons ID d'application / Code de commande connues et valides.
4. **Pare-feu Diameter (Couche Inférieure)** — Activez les vérifications de conformité au protocole.
5. **Pare-feu Diameter (Catégorie 2)** — Configurez les préfixes IMSI/MSISDN domicile.
6. **Masquage de Topologie** — Activez d'abord la suppression des Route-Records, puis la réécriture d'Origin-Host.
7. **Pare-feu Diameter (Catégorie 3)** — Activez avec l'action `:log_only` pour observer les modèles de localisation avant d'appliquer.

Défense en Profondeur

Ces modules mettent en œuvre le principe de **défense en profondeur** décrit dans la Section 3.4 de FS.19. Chaque couche aborde une classe différente d'attaque :

Classe d'Attaque	Défense Principale	Défense Secondaire
DoS Volumétrique	Limiteur de Taux	Pare-feu Diameter (toutes les catégories)
Abus d'Interface	Catégorie 1	Assainisseur AVP
Ciblage d'Abonné Domicile	Catégorie 2	Catégorie 1 (restriction d'interface)
Spoofing de Localisation	Catégorie 3	Catégorie 2 (vérification d'abonné domicile)
Découverte de Topologie	Masquage de Topologie	-
Injection d'AVP	Assainisseur AVP	Filtrage de Couche Inférieure
Évasion de Protocole (Doublement d'AVP)	Filtrage de Couche Inférieure	Assainisseur AVP

Référence Croisée des Documents GSMA

Module / Fonctionnalité	FS.19 v10.0	FS.21 v12.0
Filtrage de Format de Couche Inférieure	Section 3.3.4	Section 7.3.1
Filtrage de Catégorie 1	Section 3.3.5, Annexe B.3.3	Section 7.2.1
Filtrage de Catégorie 2	Section 3.3.6, Annexe B.3.4	Section 7.2.2, Section 16
Filtrage de Catégorie 3	Section 3.3.7, Annexe B.3.5	Section 7.2.3
Masquage de Topologie	Section 2.4, Section 3.4	Section 3.6
Limitation de Taux	Section 3.4	-
Assainissement d'AVP	Section 4.8.1, 4.8.2	Section 3.6
Défense en Profondeur	Section 3.4	Section 3.15
Catégories de Filtrage (Indépendant du Protocole)	Annexe A	Section 7

Fonction de Recherche d'Abonnés (SLF)

La Fonction de Recherche d'Abonnés (SLF) apprend dynamiquement quel élément de réseau sert chaque abonné en observant le trafic de signalisation Diameter passant par le DRA. Elle utilise ces liaisons apprises pour acheminer les demandes ultérieures — telles que les requêtes de service de localisation — directement vers le nœud de service correct sans nécessiter de règles de routage statiques.

Ceci est particulièrement utile pour les requêtes de service de localisation SLg/SLh, où le GMLC doit atteindre le MME de service pour un abonné donné mais n'a aucune connaissance préalable de quel MME il s'agit.

Le concept de SLF est décrit dans 3GPP TS 29.172 et 3GPP TS 29.173 pour les services de localisation, avec l'enregistrement des abonnés défini dans 3GPP TS 29.272.

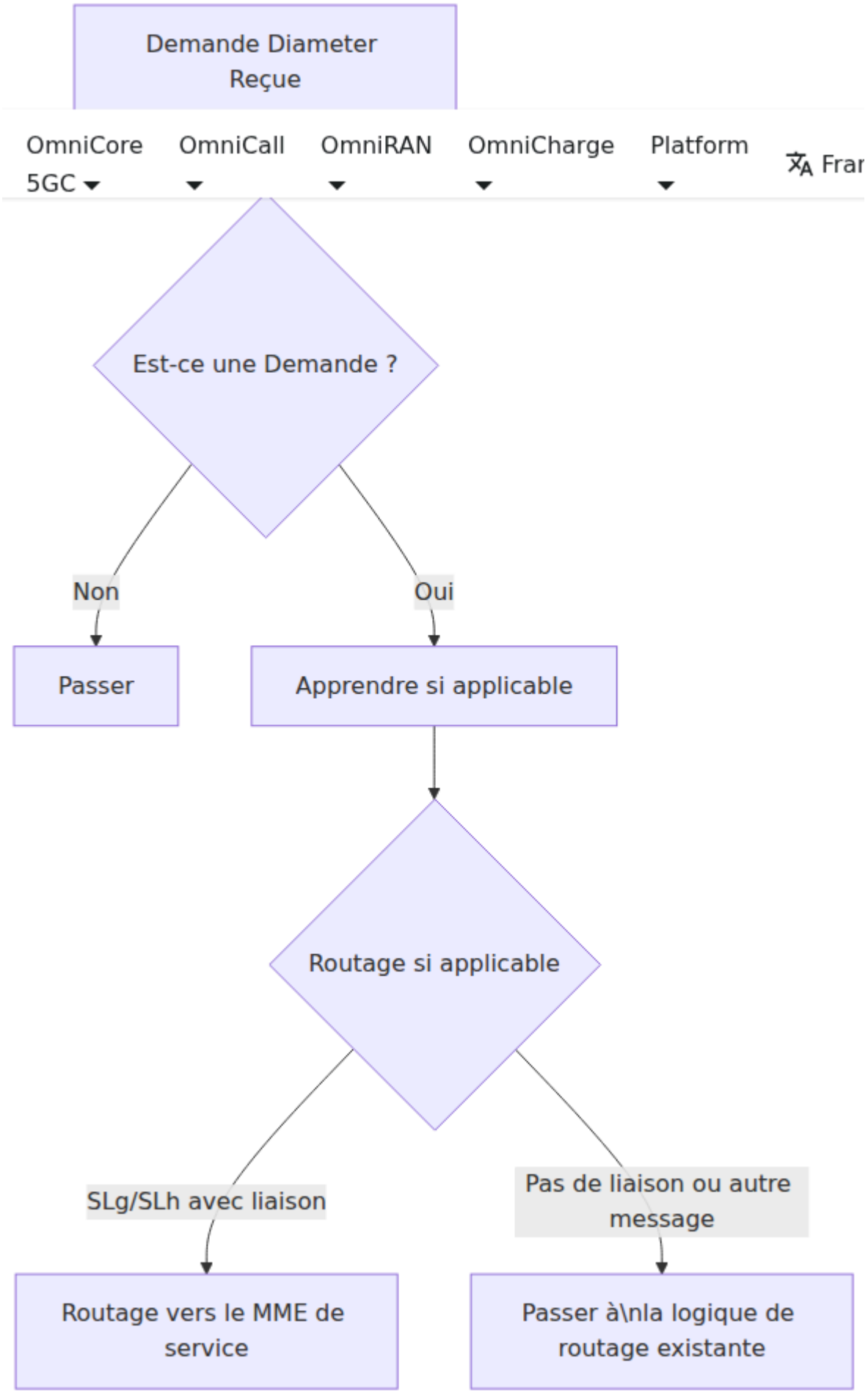
Comment ça fonctionne

Le module fonctionne en deux phases : **apprentissage** et **routage**.

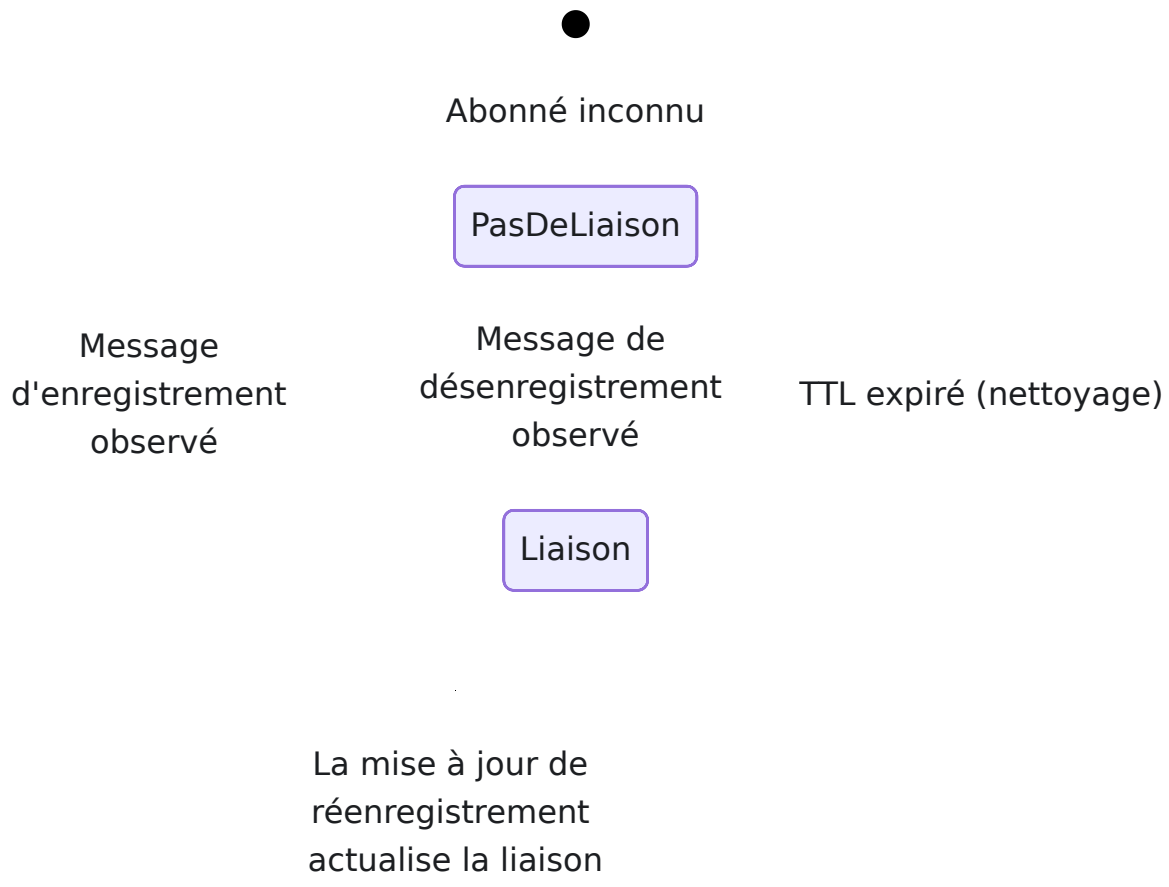
Lors de l'**apprentissage**, le module observe passivement les messages d'enregistrement et de désenregistrement circulant à travers le DRA. Lorsqu'un abonné s'enregistre (par exemple via une S6a Update-Location-Request), le module enregistre quel élément de réseau sert maintenant cet abonné.

Lors du **routage**, lorsqu'une demande arrive qui doit atteindre le nœud de service d'un abonné (par exemple, une SLg Provide-Location-Request), le module recherche la liaison et route directement vers le pair correct.

Si aucune liaison n'existe pour un abonné, la demande passe à la logique de routage existante (y compris le Routage Avancé).

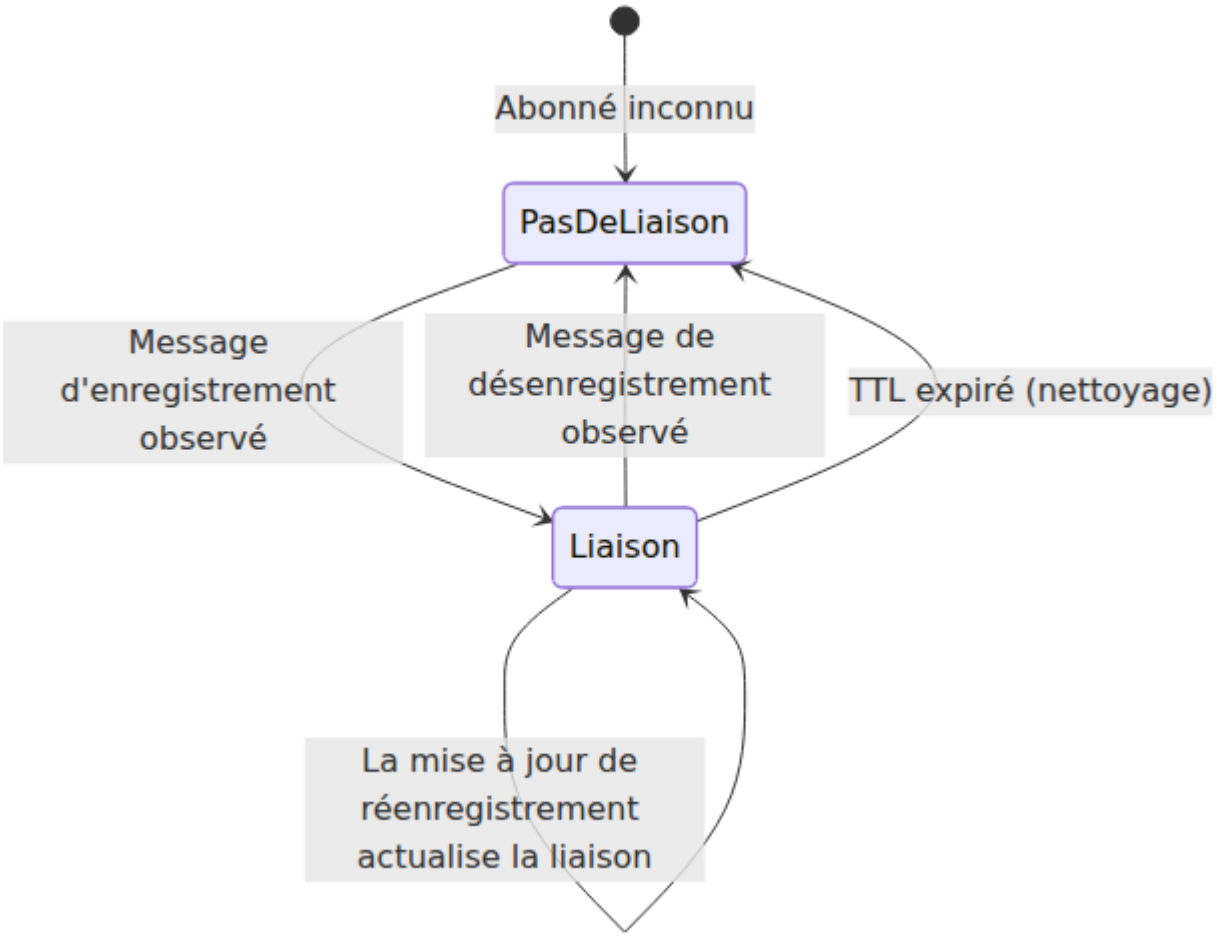

Demande Diameter Reçue
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Frar
Est-ce une Demande ?
Non
Oui
Passer
Apprendre si applicable
Routage si applicable
SLg/SLh avec liaison
Pas de liaison ou autre message
Routage vers le MME de service
Passer à\nla logique de routage existante

Cycle de Vie des Liaisons



Apprentissage : Messages Qui Créent des Liaisons

Le module apprend les liaisons abonné-nœud de service à partir des messages Diameter suivants :



Interface	Message	Code de Commande	Liaison Créée	Source IMSI
S6a	Update-Location-Request (ULR)	316	`serving_mme`	User-Name AVP (1)
Gx	Credit-Control-Request Initial (CCR-I)	272 (CC-Request-Type=1)	`serving_pgw`	Subscription-Id AVP (443)
Cx	Server-Assignment-Request (SAR)	301 (Server-Assignment-Type=1)	`serving_cscf`	Public-Identity AVP (601)

Apprentissage : Messages Qui Suppriment des Liaisons

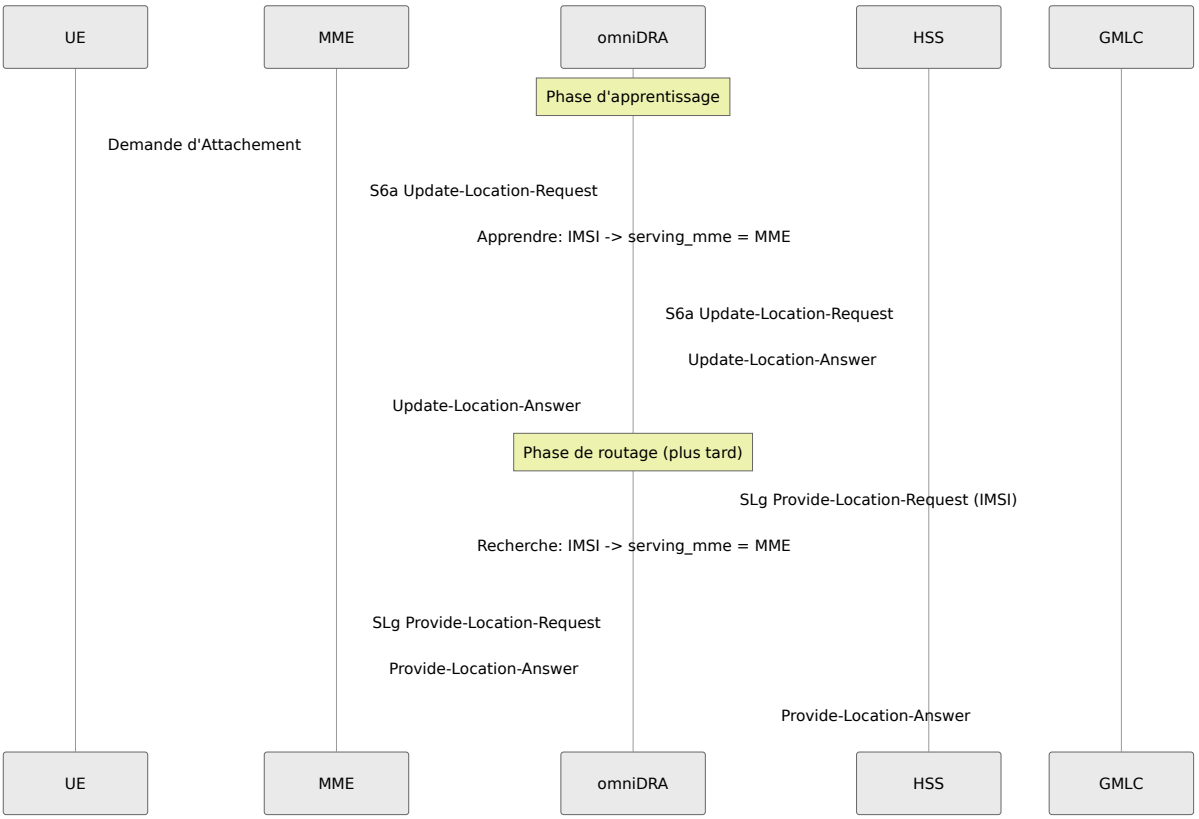
Interface	Message	Code de Commande	Liaison Supprimée
S6a	Cancel-Location-Request (CLR)	317	`serving_mme`
S6a	Purge-UE-Request (PUR)	321	`serving_mme`
Gx	Credit-Control-Request Termination (CCR-T)	272 (CC-Request-Type=3)	`serving_pgw`
Cx	Server-Assignment-Request (SAR)	301 (Server-Assignment-Type=4)	`serving_cscf`

Lorsque la dernière liaison pour un abonné est supprimée, l'ensemble de l'enregistrement de l'abonné est supprimé de la table.

Routage : Messages Qui Utilisent des Liaisons

Interface	Message	Code de Commande	Liaison Utilisée
SLg	Provide-Location-Request (PLR)	8388620	`serving_mme`
SLh	LCS-Routing-Info-Request (RIR)	8388622	`serving_mme`

Flux d'Exemple : Requête de Service de Localisation



Position dans le Pipeline

La recherche d'abonnés s'exécute **avant** le routage avancé dans le pipeline de traitement des demandes. Si le SLF trouve une liaison et remplace la route, le routage avancé s'exécute toujours mais la route SLF prend le pas.

Configuration

Le module est configuré sous `module_subscriber_lookup` dans `config/runtime.exs`.

```
module_subscriber_lookup: %{
  # Activer ou désactiver le module
  enabled: false,

  # Durée de conservation des liaisons avant expiration (secondes)
  binding_ttl_seconds: 86400
}
```

Paramètres

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Boolean	Oui	`false`	Activer ou désactiver le module de recherche d'abonnés. Lorsque `false`, toutes les demandes passent sans modification et aucune liaison n'est apprise.
`binding_ttl_seconds`	Integer	Non	`86400`	Temps en secondes avant qu'une liaison d'abonné n'expire et soit supprimée lors du nettoyage. Par défaut, c'est 24 heures. Le nettoyage s'exécute à la moitié de l'intervalle TTL, avec un minimum de 60 secondes.

Exemples de Configuration

Déploiement Standard

Adapté à la plupart des réseaux où les abonnés se réenregistrent dans les 24 heures.

```
module_subscriber_lookup: %{
  enabled: true,
  binding_ttl_seconds: 86400
}
```

Comment ça fonctionne : Le module apprend les liaisons MME/PGW/CSCF à partir du trafic passant et les conserve pendant 24 heures. Les requêtes SLg et SLh sont automatiquement routées vers le MME de service correct. Les liaisons sont rafraîchies chaque fois qu'un nouveau message d'enregistrement est observé pour le même abonné.

Cas d'utilisation : Réseaux avec des exigences de service de localisation (LCS) où le GMLC doit atteindre le MME de service sans mappages statiques abonné-MMEs.

Environnement à Forte Mobilité

Pour les réseaux avec une mobilité fréquente des abonnés où les liaisons obsolètes doivent expirer rapidement.

```
module_subscriber_lookup: %{
  enabled: true,
  binding_ttl_seconds: 3600
}
```

Comment ça fonctionne : Les liaisons expirent après 1 heure. Ceci est approprié lorsque les abonnés se déplacent fréquemment entre les MMEs et que les liaisons obsolètes pourraient causer des requêtes de localisation mal dirigées. L'intervalle de nettoyage s'exécute toutes les 30 minutes.

Cas d'utilisation : Réseaux urbains denses ou événements avec une forte mobilité des abonnés.

Métriques

Mises à Jour des Liaisons

Métrique : `diameter.subscriber_lookup.binding.update` **Type** : Compteur **Description** : Incrémenté chaque fois qu'une liaison d'abonné est créée ou mise à jour. **Étiquettes** :

- `imsi` - IMSI de l'abonné
- `binding_type` - Type de liaison : `serving_mme`, `serving_pgw`, ou `serving_cscf`
- `serving_host` - Origin-Host de l'élément de réseau de service

Suppressions de Liaisons

Métrique : `diameter.subscriber_lookup.binding.delete` **Type** : Compteur **Description** : Incrémenté chaque fois qu'une liaison d'abonné est explicitement supprimée via un message de désenregistrement. **Étiquettes** :

- `imsi` - IMSI de l'abonné
- `binding_type` - Type de liaison supprimée

Requêtes Routées

Métrique : `diameter.subscriber_lookup.route.count` **Type** : Compteur **Description** : Incrémenté chaque fois qu'une requête SLg/SLh est correctement routée en utilisant une liaison apprise. **Étiquettes** :

- `imsi` - IMSI de l'abonné
- `binding_type` - Type de liaison utilisée pour le routage (actuellement toujours `serving_mme`)
- `serving_host` - Origin-Host du pair vers lequel la demande a été routée

Dépannage

Requêtes de Localisation Non Routées vers le MME de Service

Symptômes : Les SLg Provide-Location-Requests ou SLh LCS-Routing-Info-Requests ne sont pas routées vers le MME attendu, passant à la routage par défaut à la place.

Causes possibles :

- L'abonné ne s'est pas enregistré via ce DRA, donc aucune liaison n'existe
- La liaison a expiré (TTL dépassé)
- Le pair MME de service n'est pas connecté à ce DRA
- `enabled` est réglé sur `false`

Résolution :

1. Vérifiez que le module est activé dans la configuration
2. Vérifiez que le trafic S6a ULR pour l'abonné passe par ce DRA (les liaisons ne sont apprises qu'à partir du trafic observé)
3. Vérifiez que `binding_ttl_seconds` est suffisamment long pour couvrir l'intervalle entre l'enregistrement et la demande de localisation
4. Confirmez que le MME de service est connecté en tant que pair

Liaisons Non Apprises

Symptômes : La table de liaisons reste vide malgré le passage du trafic S6a/Gx/Cx à travers le DRA.

Causes possibles :

- Le module n'est pas activé
- Le processus du module n'est pas en cours d'exécution (vérifiez le superviseur)
- Les messages ne contiennent pas un IMSI valide dans l'AVP attendu

Résolution :

1. Vérifiez `enabled: true` dans la configuration
2. Confirmez que le DRA a été redémarré après le changement de configuration
3. Vérifiez les journaux de débogage du DRA pour les entrées `SubscriberLookup: Learned` pour confirmer l'activité de liaison

Liaisons Obsolètes Causant des Requêtes Mal Routées

Symptômes : Les requêtes de localisation sont routées vers un MME qui ne sert plus l'abonné.

Causes possibles :

- Le Cancel-Location-Request (CLR) ou le Purge-UE-Request (PUR) n'ont pas passé par ce DRA
- `binding_ttl_seconds` est réglé trop haut pour le modèle de mobilité du réseau

Résolution :

1. Réduisez `binding_ttl_seconds` pour correspondre à l'intervalle de réenregistrement attendu des abonnés
2. Assurez-vous que tout le trafic de désenregistrement S6a passe par ce DRA

Référence

Identifiants d'Application

ID	Interface	Description	Référence
16777251	S6a/S6d	Authentification MME/SGSN et gestion des abonnements vers HSS	3GPP TS 29.272
16777238	Gx	PCEF vers PCRF contrôle de politique et de facturation	3GPP TS 29.212
16777216	Cx	I-CSCF/S-CSCF vers HSS enregistrement IMS	3GPP TS 29.229
16777255	SLg	GMLC vers MME services de localisation	3GPP TS 29.172
16777291	SLh	GMLC vers HSS/DRA informations de routage LCS	3GPP TS 29.173

Codes de Commande

Code	Nom	Interface	Description
272	Credit-Control-Request/Answer (CCR/CCA)	Gx	Contrôle de politique et de facturation au niveau de la session
301	Server-Assignment-Request/Answer (SAR/SAA)	Cx	Enregistrement et désenregistrement IMS
316	Update-Location-Request/Answer (ULR/ULA)	S6a	Mise à jour de la localisation de l'abonné au MME
317	Cancel-Location-Request/Answer (CLR/CLA)	S6a	Annulation de localisation initiée par HSS
321	Purge-UE-Request/Answer (PUR/PUA)	S6a	Purge d'UE initiée par MME
8388620	Provide-Location-Request/Answer (PLR/PLA)	SLg	Requête de service de localisation vers le MME de service
8388622	LCS-Routing-Info-Request/Answer (RIR/RIA)	SLh	Recherche d'informations de routage LCS

Types de Liaisons

Type de Liaison	Appris de	Utilisé par	Description
`serving_mme`	S6a ULR	SLg PLR, SLh RIR	Le MME servant actuellement l'abonné
`serving_pgw`	Gx CCR-I	—	Le PGW gérant la session de l'abonné (réservé pour un routage futur)
`serving_cscf`	Cx SAR (Enregistrement)	—	Le S-CSCF servant l'enregistrement IMS de l'abonné (réservé pour un routage futur)