← Resumen

Referencia de Configuración de OmniUDR

OmniUDR lee su configuración de operador de la clave de entorno de la aplicación `:omniudr`, poblada al inicio desde `config/runtime.exs`. La mayoría de las claves de tiempo de ejecución tienen una sobreescritura de variable de entorno y un valor predeterminado. `hss_api_key` no tiene ninguna y es obligatoria. El bloque a continuación es la superficie completa del operador.

Configuración de Tiempo de Ejecución

```
import Config

config :omniudr,
  # Backend de API HSS
  hss_api_base_url: System.get_env("HSS_API_BASE_URL",
"https://127.0.0.1:8443"),
  hss_api_key: "<Clave API de OmniHSS con rol hlr_interworking>",

  # Configuración de SBI
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr: System.get_env("SBI_ADDR", "127.0.0.22"),
  sbi_port: String.to_integer(System.get_env("SBI_PORT", "7777")),

  # Configuración de NRF
  nrf_uri: System.get_env("NRF_URI", "http://127.0.0.1:7777"),

  # Identidad PLMN
  mcc: System.get_env("MCC", "999"),
  mnc: System.get_env("MNC", "70"),

  # Métricas de Prometheus
  prometheus_metrics_port:
String.to_integer(System.get_env("PROMETHEUS_PORT", "9568")),

  # Intervalo de latido (ms)
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000")),

  # Suscriptores de prueba - ayuda para la configuración del
laboratorio. Mapa JSON de IMSI → {ki, opc, amf}.
  test_subscribers: System.get_env("TEST_SUBSCRIBERS")

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

Tabla de Parámetros

Parámetro	Tipo	Requerido	Predeterminado
`hss_api_base_url`	cadena (URL)	Sí	`https://127.0.0.1:8`
`hss_api_key`	cadena	Sí	Ninguno

Parámetro	Tipo	Requerido	Predeterminado
`sbi_scheme`	cadena	No	`http`
`sbi_addr`	cadena (IP)	No	`127.0.0.22`
`sbi_port`	entero	No	`7777`
`nrf_uri`	cadena (URL)	No	`http://127.0.0.1:7`
`mcc`	cadena	No	`999`

Parámetro	Tipo	Requerido	Predeterminado
`mnc`	cadena	No	`70`
`prometheus_metrics_port`	entero	No	`9568`
`heartbeat_interval`	entero (ms)	No	`10000`
`test_subscribers`	cadena (JSON) | nil	No	`nil`

Parámetro	Tipo	Requerido	Predeterminado
`default_am_nssai`	mapa	No	eMBB / URLLC / mIoT slices (ver más abajo
`dnn_snssai_map`	mapa	No	`%{"iot" => %{"sst" 3, "sd" => "000003"`

Parámetro	Tipo	Requerido	Predeterminado
`notification_max_attempts`	entero	No	`4`
`notification_backoff_ms`	entero (ms)	No	`500`

Parámetro	Tipo	Requerido	Predeterminado

> Tanto `default_am_nssai` como `dnn_snssai_map` existen para unir el modelo de datos HSS 4G (sin concepto de slice) a las estructuras Nudr 5G. Alinee estos con el plan de slices real de la implementación y con el NSSAI servido por el AMF/SMF. Los valores predeterminados de AM/SM (eMBB `sst 1`, URLLC `sst 2 / sd 000002`, mIoT `sst 3 / sd 000003`) son valores de laboratorio.

Formas del mapa de síntesis de slices (`default_am_nssai`, `dnn_snssai_map`)

Ambas claves son mapas estructurados de objetos S-NSSAI. Un **S-NSSAI** tiene dos subcampos:

Subcampo	Tipo	Requerido	Significado
`sst`	entero (0–255)	Sí	Tipo de Slice/Servicio. Valores estandarizados: `1` = eMBB, `2` = URLLC, `3` = mIoT (según TS 23.501).
`sd`	cadena (6 dígitos hexadecimales)	No	Diferenciador de Slice. Distingue dos slices con el mismo `sst`. Omitirlo para un slice bare-`sst`; cuando está presente, debe coincidir exactamente con el valor servido por el AMF/SMF.

`default_am_nssai` es el NSSAI de acceso y movilidad, con dos listas:

Lista	Significado
`defaultSingleNssais`	Los S-NSSAIs que el UE obtiene cuando se registra sin solicitar un slice específico - el conjunto de slices predeterminado de la red para el suscriptor.
`singleNssais`	El conjunto completo de S-NSSAIs que el suscriptor puede usar (un superconjunto de `defaultSingleNssais`). Un UE que solicita un S-NSSAI calificado por SD que no se encuentra aquí es rechazado por el AMF con la causa 5GMM #62.

```
config :omniudr, :default_am_nssai, %{
  "defaultSingleNssais" => [%{"sst" => 1}, %{"sst" => 1, "sd" =>
"000001"}],
  "singleNssais" => [
    %{"sst" => 1},
    %{"sst" => 1, "sd" => "000001"},
    %{"sst" => 2, "sd" => "000002"},
    %{"sst" => 3, "sd" => "000003"}
  ]
}
```

`dnn_snssai_map` mapea una **cadena DNN** al S-NSSAI único en el que viven sus sesiones PDU. Cualquier DNN no presente en el mapa se predetermina a eMBB (`%{"sst" => 1, "sd" => "000001"}`). Mantenga cada S-NSSAI referenciado aquí dentro de `default_am_nssai.singleNssais`, o el SMF colocará la sesión en un slice al que el suscriptor no está suscrito.

```
config :omniudr, :dnn_snssai_map, %{
  "iot" => %{"sst" => 3, "sd" => "000003"},
  "internet" => %{"sst" => 1, "sd" => "000001"}
}
```

Acerca de `hss_api_base_url` (el backend REST de HSS)

Esta es la configuración de operador más importante. OmniUDR es una fachada - no tiene base de datos. Cuando un consumidor le pide a OmniUDR la suscripción de autenticación de un suscriptor, datos AM, datos SM o datos de políticas, OmniUDR emite solicitudes REST en vivo a `hss_api_base_url` (principalmente `GET /api/subscriber/imsi/{imsi}` y `GET /api/key_set/{id}`, y para vectores de autenticación / SQN / ubicación los endpoints `/api/hlr/*`), luego mapea la respuesta HSS a la estructura 3GPP. Cada solicitud HSS tiene un tiempo de espera de 5 segundos (3 segundos para la verificación de estado). Si este backend no es accesible, las lecturas correspondientes de Nudr devuelven `404`/`5xx` - vea Solución de Problemas.

Acerca de `hss_api_key` (control de acceso de OmniHSS)

OmniHSS utiliza control de acceso basado en roles en su API REST. Cada cliente debe enviar su clave API en el encabezado `x-omnihss-key`. OmniHSS rechaza una solicitud sin una clave válida con `401`. OmniUDR envía `hss_api_key` en cada solicitud a `hss_api_base_url`.

Otorgue a la clave de OmniUDR el rol `hlr_interworking` de OmniHSS. Este rol permite estos endpoints de OmniHSS:

- `GET /api/subscriber/imsi/{imsi}` y `GET /api/subscriber/msisdn/{msisdn}`
- `GET /api/key_set/{id}`
- `POST /api/hlr/generate_auth_vectors` y `POST /api/hlr/generate_eutran_auth_vectors`
- `PUT`, `PATCH` y `DELETE /api/hlr/circuit_session/{imsi}`
- `GET /api/status/api`

El valor es un secreto. OmniUDR no lo escribe en el registro. Instale esta versión antes de habilitar el control de acceso en OmniHSS, de lo contrario, todas las solicitudes de OmniUDR a OmniHSS fallarán con `401`.

Acerca de `test_subscribers`

`test_subscribers` poblada un conjunto de suscriptores estáticos en el código (dos IMSIs integrados - vectores de prueba Milenage bien conocidos - más cualquier proporcionado a través del mapa JSON) que devuelve datos de autenticación, AM y SM fijos sin ninguna llamada HSS. Es una **ayuda de laboratorio** para la configuración y pruebas de conformidad donde no se justifica la configuración de un HSS completo. En la versión actual, la ruta de datos Nudr_DR siempre lee del backend HSS; el conjunto de suscriptores de prueba no está conectado como un respaldo automático cuando el HSS está inactivo. Trate esta clave solo como una superficie de laboratorio/pruebas y déjela sin configurar (`nil`) en cualquier implementación respaldada por un HSS real. Un valor JSON no válido se predetermina a los dos valores predeterminados integrados.

Identidad PLMN y perfil NRF

`mcc` y `mnc` se publicitan en el perfil NF de NRF como el PLMN que sirve. La capacidad y prioridad del NF publicitadas al NRF se gestionan a través de la configuración compartida de Omni5G NRF y pueden cambiarse en tiempo de ejecución a través de la API OAM (`nf_capacity`, `nf_priority`).

Entrega de notificaciones

`notification_max_attempts` y `notification_backoff_ms` regulan cómo OmniUDR entrega notificaciones DataChangeNotify de Nudr_DR a suscriptores de cambio de datos (el servicio `subs-to-notify`, suscripciones TS 29.504). Cuando un recurso monitoreado cambia, OmniUDR publica una notificación en la URI de callback de cada suscriptor coincidente. Si una entrega falla (la callback no es accesible o devuelve un estado no 2xx), OmniUDR reintenta hasta `notification_max_attempts` veces, esperando `notification_backoff_ms` para el primer reintento y duplicando el retraso en cada reintento subsiguiente. Una vez que se agotan los intentos, la notificación se descarta y el evento se registra. La entrega es de mejor esfuerzo y nunca afecta la respuesta SBI devuelta al consumidor que causó el cambio. Ambas claves son solo de configuración de aplicación (sin sobreescritura de variable de entorno): configúrelas bajo `config :omniudr, ...`.

API de Gestión y OAM

Más allá del listener SBI, OmniUDR expone una API de gestión HTTPS orientada al operador en el puerto **8443** (TLS, `enable_tls: true`). También se sirve una interfaz web del Panel de Control a través de HTTPS en el puerto **7443**.

Superficie	Puerto	Propósito
SBI (Nudr_DR)	`sbi_port` (predeterminado `7777`)	Interfaz de servicio 3GPP
Métricas de Prometheus	`prometheus_metrics_port` (predeterminado `9568`)	Endpoint de recolección de métricas
API de Gestión / OAM	`8443` (HTTPS, montada bajo `/api`)	Estado, salud de HSS, proxy de suscriptores, acciones OAM
Interfaz del Panel de Control	`7443` (HTTPS)	Recursos de aplicación, configuración, licencia y páginas de registro

La API de gestión está montada bajo el prefijo de ruta `/api`. Proporciona estado de solo lectura, una verificación de conectividad HSS, un proxy de suscriptores de solo lectura hacia el HSS y acciones OAM:

Método	Ruta	Propósito
GET	`/api/status/api`	Estado de API/servicio
GET	`/api/status/nrf`	Estado de registro NRF
GET	`/api/status/nf`	Perfil / estado NF
GET	`/api/status/license`	Estado de licencia
GET	`/api/health/hss`	Verificación de conectividad HSS (llama a `GET /api/status/api` en el HSS)
GET	`/api/subscribers`	Lista de suscriptores de solo lectura paginada proxied desde el HSS (`page`, `page_size`)
GET	`/api/subscribers/{imsi}`	Detalle completo del suscriptor proxied desde el HSS
GET	`/api/oam/config`	Ver configuración de tiempo de ejecución en vivo
PATCH	`/api/oam/config/{id}`	Actualizar `hss_api_base_url`, `nrf_uri`, `nf_capacity`, `nf_priority` o `log_level` en tiempo de ejecución
POST	`/api/oam/nrf/reregister`	Forzar re-registro de NRF
DELETE	`/api/oam/subscriber-cache/{id}`	Vaciar el contexto AMF/SMF en caché para un suscriptor (`id=all` vacía todos los suscriptores)
POST	`/api/oam/log_level`	Cambiar el nivel de registro en tiempo de ejecución

Notas sobre la actualización de configuración OAM (`PATCH /api/oam/config/{id}`):

- Solo se aceptan `hss_api_base_url`, `nrf_uri`, `nf_capacity`, `nf_priority` y `log_level`; cualquier otra clave se ignora, y una solicitud sin clave válida devuelve 400.
- `nf_capacity` y `nf_priority` deben ser enteros; `log_level` debe ser uno de `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info`, `debug`.
- Cambiar `nf_capacity` / `nf_priority` persiste el valor y envía un perfil actualizado al NRF.
- El proxy de suscriptores es de solo lectura. Si el HSS no es accesible, `/api/subscribers` y `/api/subscribers/{imsi}` devuelven `502`; un suscriptor deshabilitado o ausente devuelve `404`.

Registro

OmniUDR emite registros JSON estructurados a través de `OmniLogger.JsonFormatter` con metadatos completos. El nivel de registro en tiempo de ejecución puede cambiarse sin reiniciar a través de la API OAM (`POST /api/oam/log_level` o `PATCH /api/oam/config/{id}` con `log_level`). Las acciones OAM (actualizaciones de configuración, vaciado de caché, re-registro) se registran a nivel `info` con un prefijo `[OAM]`; los fallos en solicitudes HSS se registran a nivel `warning`/`error`.

← Resumen

Métricas de OmniUDR

OmniUDR expone métricas de Prometheus en el endpoint de métricas (`prometheus_metrics_port`, por defecto `9568`). Los nombres de las métricas a continuación utilizan la forma de exposición de Prometheus (los puntos se convierten en guiones bajos). Las etiquetas y sus valores están en minúsculas.

Métricas de UDR / HSS

Métrica	Tipo	Etiquetas	Descri
`omni_udr_nrf_registration_status`	gauge	`nf_type`	Estado de de NRF (1 registrado
`omni_udr_hss_health`	gauge	-	Salud de conectivid OmniHSS activo, 0 = inactivo); actualizad cada verif de `/healt`
`omni_udr_hss_requests_total`	counter	`endpoint`, `result`	Conteo de llamadas I backend d etiquetada ruta de so (`endpoint` `result` (`success`/
`omni_udr_hss_request_duration_ms`	distribution	`endpoint`	Latencia d solicitud R HSS en milisegun Cubos: 5, 50, 100, 2 1000, 250

Métricas de BEAM VM

Métrica	Tipo	Descripción
`beam_memory_total`	gauge	Memoria total de BEAM en bytes
`beam_memory_processes`	gauge	Memoria utilizada por procesos de Erlang
`beam_memory_system`	gauge	Memoria del sistema (ETS, átomos, código, binarios)
`beam_processes_count`	gauge	Número de procesos de Erlang
`beam_vm_uptime`	gauge	Tiempo de actividad de la VM en segundos

PromQL Útil

- Disponibilidad del backend de HSS: `omni_udr_hss_health`
- UDR registrado con NRF: `omni_udr_nrf_registration_status`
- Tasa de error de HSS (5 min): `sum(rate(omni_udr_hss_requests_total{result="failure"}[5m])) / sum(rate(omni_udr_hss_requests_total[5m]))`
- Latencia p95 de HSS (5 min): `histogram_quantile(0.95, sum(rate(omni_udr_hss_request_duration_ms_bucket[5m])) by (le, endpoint))`
- Tasa de solicitudes de HSS por endpoint: `sum(rate(omni_udr_hss_requests_total[1m])) by (endpoint)`
- Conteo de procesos de BEAM: `beam_processes_count`

← Resumen

Guía de Operaciones de OmniUDR

OmniUDR es el Repositorio de Datos Unificado (UDR) del núcleo 5G de Omnitouch. Implementa el servicio Nudr_DataRepository (TS 29.504) y expone datos estructurados de suscriptores, políticas y contexto al UDM y PCF a través de la interfaz basada en servicios (SBI).

OmniUDR no posee datos de suscriptores. Es una fachada delgada y conforme a estándares Nudr_DR sobre el **backend de aprovisionamiento REST de OmniHSS**: cada lectura del repositorio de datos se traduce en el momento de la solicitud en una o más llamadas REST de OmniHSS, y la respuesta del HSS se mapea al modelo de datos Nudr de 3GPP antes de ser devuelta al consumidor. OmniUDR es la única función de red del núcleo 5G que se comunica directamente con la API REST de OmniHSS: el UDM (y, a través de él, el AUSF) accede a los datos de suscriptores llamando a OmniUDR a través de Nudr estándar, no llamando al HSS.

Debido a que el HSS es el sistema de registro, OmniUDR casi no mantiene estado persistente. Su único estado local son tres pequeños almacenes en memoria (ETS): un caché de contexto de registro de AMF y SMF (utilizado para decidir si una escritura de contexto es una creación o una actualización), un almacén de suscripciones de cambios de datos que contiene las suscripciones activas Nudr_DR `subs-to-notify`, y un almacén de datos de aplicaciones que contiene las familias de datos `application-data` de Nudr_DR (PFDs, influencia de tráfico, BDT). Ninguno de estos se persiste; todos se borran al reiniciar. El caché de contexto de registro se puede vaciar a través de la API de gestión.

Referencias de Rol y Especificación 3GPP

Especificación	Relevancia
TS 23.501	Arquitectura del sistema 5G - rol del UDR y su relación con UDM, PCF y la capa de datos de UDM
TS 29.504	Servicio Nudr_DataRepository - marco de recursos, datos de suscripción, datos de contexto, datos de políticas
TS 29.505	Estructuras de datos de suscripción - suscripción de autenticación, acceso y movilidad, gestión de sesiones, selección de SMF
TS 29.519	Estructuras de datos de políticas - formas de recursos de datos de políticas AM y SM
TS 29.503	Servicios Nudm - el consumidor UDM cuyas procedimientos UEAU/SDM/UECM impulsan lecturas de UDR y escrituras de contexto
TS 29.571	Tipos de datos comunes - `PatchResult`, `ProblemDetails`, codificaciones PLMN y S-NSSAI

Interfaces

Interfaz	Enlace	Propósito
SBI (Nudr_DR)	`{sbi_scheme}://{sbi_addr}:{sbi_port}` (por defecto `http://127.0.0.22:7777`)	Sirve los recursos `nudr-dr/v2` consumidos por el UDM y PCF
Backend REST de HSS	`hss_api_base_url` (por defecto `https://127.0.0.1:8443`)	Llamadas REST salientes a OmniHSS que satisfacen cada lectura/actualización. Cada llamada envía `hss_api_key` en el encabezado `x-omnihss-key` (rol de OmniHSS `hlr_interworking`)
NRF	`nrf_uri`	Registro de NF y latido periódico
API de Gestión / OAM	`:8443` (HTTPS, montado bajo `/api`)	Estado, salud del HSS, proxy de suscriptores y acciones de OAM - ver Referencia de Configuración
Métricas de Prometheus	`prometheus_metrics_port` (por defecto `9568`)	Punto de finalización de métricas - ver Métricas

---

Conjuntos de Datos Servidos

OmniUDR sirve los siguientes conjuntos de datos Nudr_DR. Cada uno se deriva en el momento de la lectura del registro de suscriptor de OmniHSS; la columna

"fuente HSS" nombra los datos REST subyacentes de los que se mapea el valor.

Conjunto de datos	Familia de recursos Nudr	Acceso	Consumid
Suscripción de autenticación	`subscription-data/.../authentication-data/authentication-subscription`	Leer / actualizar	UDM (Nudm_UEA
Vectores de autenticación	`subscription-data/.../authentication-data/authentication-vectors`	Generar (propietario)	OmniUDM
Datos de acceso y movilidad (AM)	`subscription-data/.../provisioned-data/am-data`	Leer	UDM (Nudm_SDI
Datos de gestión de sesiones (SM)	`subscription-data/.../provisioned-data/sm-data`	Leer	UDM (Nudm_SDI
Datos de selección de SMF	`subscription-data/.../provisioned-data/smf-selection-subscription-data`	Leer	UDM (Nudm_SDI
Contexto de registro de AMF	`subscription-data/.../context-data/amf-3gpp-access`	Escribir	UDM (Nudm_UEC
Contexto de registro de SMF	`subscription-data/.../context-data/smf-registrations/{pduSessionId}`	Escribir	UDM (Nudm_UEC

Conjunto de datos	Familia de recursos Nudr	Acceso	Consumic
Datos de políticas AM	`policy-data/ues/{ueId}/am-data`	Leer	PCF
Datos de políticas SM	`policy-data/ues/{ueId}/sm-data`	Leer	PCF

| Datos de aplicación | `application-data/{family}/{id}` (familias `pfds`, `influenceData`, `bdtData`) | Crear / leer / reemplazar / eliminar | PCF, NEF | Ninguno - mantenido en un almacén en memoria local (ETS), no respaldado por HSS y no persistido |

> Los datos de aplicación son servidos por un almacén genérico en memoria en lugar del HSS. No se persiste: un reinicio lo borra. Ver Puntos finales de SBI → Datos de Aplicación a continuación.

Puntos finales de SBI

Todos los puntos finales de Nudr_DR se sirven a través de HTTP en `{sbi_scheme}://{sbi_addr}:{sbi_port}` (por defecto `http://127.0.0.22:7777`) con `Content-Type: application/json`. Los fallos se devuelven como `ProblemDetails` de TS 29.571.

El segmento de ruta `{ueId}` es un SUPI en forma de `imsi-<dígitos>` (también se acepta un IMSI desnudo). El segmento `{servingPlmnId}` en las lecturas de datos aprovisionados se valida: debe ser un PLMN bien formado (5 o 6 dígitos) y debe coincidir con el PLMN que sirve configurado desde `mcc`/`mnc`, de lo contrario, la lectura es rechazada con `400 MANDATORY_IE_INCORRECT`. Esta construcción sirve un único registro por suscriptor.

Datos de Suscripción - Autenticación (TS 29.504 / 29.505)

Método	Ruta	Descripción	Éxito
GET	`/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-subscription`	Recuperar suscripción de autenticación (método, AMF, SQN)	200
PATCH	`/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-subscription`	Actualizar suscripción de autenticación (SQN)	204, o 200 + `PatchResult` en fallo parcial
POST	`/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-vectors`	Generar vectores de autenticación (ver nota)	200

> **Punto final no estándar.** El POST `authentication-vectors` es una extensión propietaria de Omni 5GC; TS 29.504/29.505 no definen tal recurso. La generación de vectores y la gestión de SQN/resync se delegan al UDR para que se manejen centralmente en el HSS. Solo es consumido por OmniUDM; un consumidor de terceros conforme a estándares nunca lo llamará.

Datos de Suscripción - Aprovisionados (TS 29.505)

Método	Ruta	Descripción	
GET	`/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/am-data`	Recuperar datos de suscripción de acceso y movilidad	2(`M` er m cc
GET	`/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/sm-data`	Recuperar datos de suscripción de gestión de sesiones	2(`M` er m cc
GET	`/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/smf-selection-subscription-data`	Recuperar datos de suscripción de selección de SMF	2(`M` er m cc

Datos de Suscripción - Contexto (TS 29.504)

Método	Ruta	Descripción	Éxito
PUT	`/nudr-dr/v2/subscription-data/{ueId}/context-data/amf-3gpp-access`	Almacenar contexto de registro de acceso 3GPP de AMF	201 + `Location` (nuevo) o 204 (actualización)
PUT	`/nudr-dr/v2/subscription-data/{ueId}/context-data/smf-registrations/{pduSessionId}`	Almacenar contexto de registro de SMF	201 + `Location` (nuevo) o 204 (actualización)

Un cuerpo vacío en cualquiera de los PUT de contexto devuelve 400 `ProblemDetails` (`cause: MANDATORY_IE_MISSING`).

Datos de Políticas (TS 29.519)

Método	Ruta	Descripción	Éxito
GET	`/nudr-dr/v2/policy-data/ues/{ueId}/am-data`	Recuperar datos de políticas AM (`subscCats`, `praInfos`)	200
GET	`/nudr-dr/v2/policy-data/ues/{ueId}/sm-data`	Recuperar datos de políticas SM (`smPolicySnssaiData` por DNN)	200

Datos de Aplicación (TS 29.504)

CRUD genérico sobre las familias de `application-data` de Nudr_DR. El segmento de ruta `{family}` es uno de `pfds`, `influenceData`, o `bdtData`; el segmento `{id}` es el id del recurso de la familia (`appId` / `influenceId` /

`bdtReferenceId`). Los registros se mantienen en un almacén en memoria local (ETS), no en el HSS, y no se persisten a través de un reinicio. Cualquier otro valor de `{family}` cae en el 404 catch-all.

Método	Ruta	Descripción	Éxito
GET	`/nudr-dr/v2/application-data/{family}`	Listar todos los recursos en la familia (colección GET)	200 (array)
GET	`/nudr-dr/v2/application-data/{family}/{id}`	Recuperar un recurso	200, o 404 si es desconocido
PUT	`/nudr-dr/v2/application-data/{family}/{id}`	Crear o reemplazar un recurso	201 + `Location` (nuevo) o 200 (reemplazar)
DELETE	`/nudr-dr/v2/application-data/{family}/{id}`	Eliminar un recurso	204, o 404 si es desconocido

Un cuerpo vacío en el PUT devuelve 400 `ProblemDetails` (`cause: MANDATORY_IE_MISSING`).

Suscripciones de Cambio de Datos (TS 29.504 subscriptions-to-notify)

Método	Ruta	Descripción	Éxito
POST	`/nudr-dr/v2/subscription-data/subs-to-notify`	Crear una suscripción de cambio de datos (el cuerpo nombra los URIs de recursos monitoreados y el URI de callback de notificación)	201 + `Location`, o 400 en un cuerpo incorrecto
DELETE	`/nudr-dr/v2/subscription-data/subs-to-notify/{subsId}`	Eliminar una suscripción de cambio de datos	204, o 404 si la suscripción es desconocida

Ver Suscripciones de Cambio de Datos y DataChangeNotify para el comportamiento de fan-out.

Cualquier ruta que no coincida con los recursos anteriores se responde con un 404 `ProblemDetails`.

Llamadas REST del Backend de HSS

Cada operación de SBI se cumple mediante una o más llamadas al backend REST de OmniHSS en `hss_api_base_url`. La relación es:

Operación Nudr_DR	Llamada(s) REST de HSS	Nota
Lectura de suscripción de autenticación	`GET /api/subscriber/imsi/{imsi}`, luego `GET /api/key_set/{id}` si el material clave se referencia por id	El material clave en el registro de omite la segund
Generación de vectores de autenticación	`GET /api/subscriber/imsi/{imsi}`, luego `POST /api/hlr/generate_eutran_auth_vectors`	Cuerpo `{imsi, num_vectors: 1 resynchronizat resynchronizat` es `rand‖auts` he resync
Actualización de SQN (PATCH)	`POST /api/hlr/update_sqn`	Cuerpo `{imsi,`
Lectura de AM / SM / selección de SMF	`GET /api/subscriber/imsi/{imsi}`	Recuperación de suscriptor, mape localmente
Escritura de contexto de AMF	`PUT /api/hlr/circuit_session/{imsi}`	Actualización de ubicación/sesión también almace localmente
Escritura de contexto de SMF	(ninguna)	Solo caché local
Lectura de datos de políticas (AM / SM)	`GET /api/subscriber/imsi/{imsi}`	Reutiliza el mapa lo reconfigura

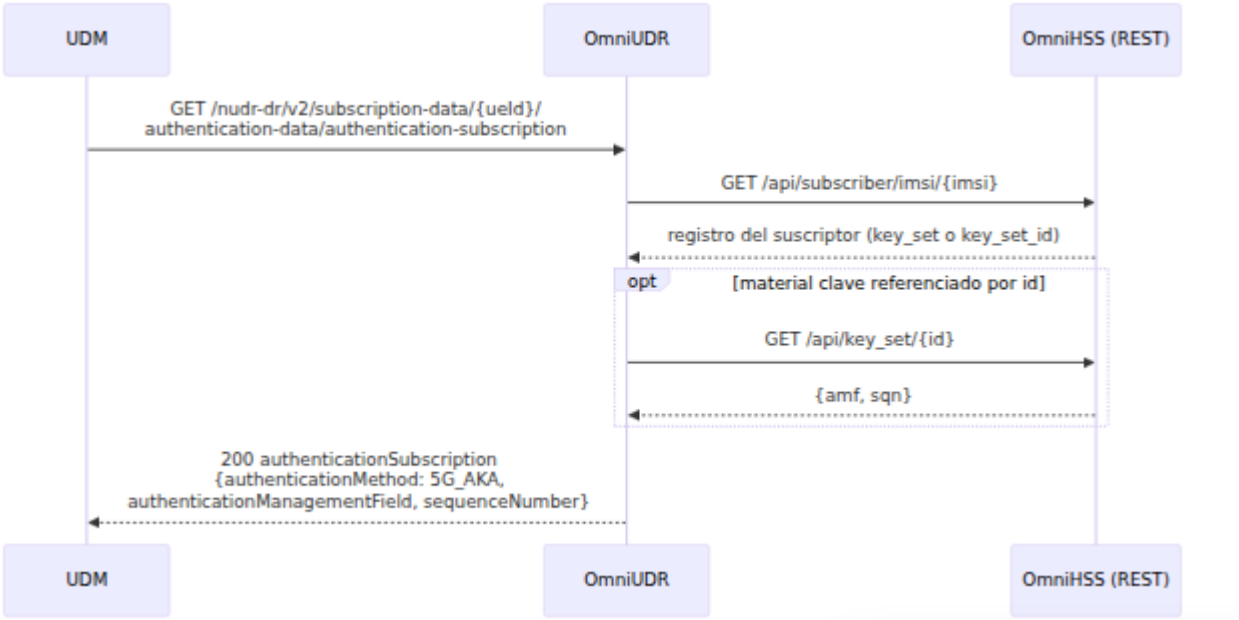
Operación Nudr_DR	Llamada(s) REST de HSS	Nota
Verificación de salud	`GET /api/status/api`	Impulsa el punto gestión `/health` medidor de salu

> Los puntos finales `/api/hlr/*` (`generate_eutran_auth_vectors`, `update_sqn`, `circuit_session/{imsi}`) son la interfaz REST diseñada a medida del HSS para las operaciones de vector de autenticación, SQN y ubicación de OmniUDR, junto con los puntos finales de suscriptor/conjunto de claves (`/api/subscriber/*`, `/api/key_set/*`) y la sonda de salud (`/api/status/api`). Un suscriptor deshabilitado (`enabled: false`) se trata como no encontrado.

Procedimientos Clave

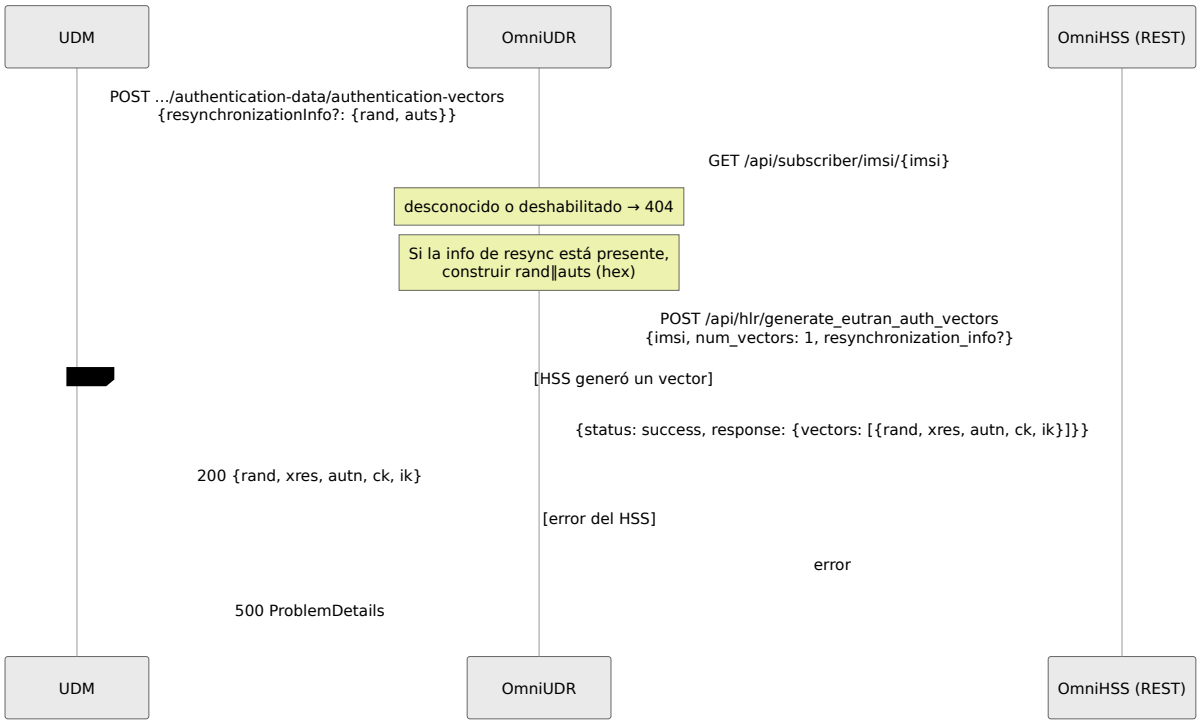
Lectura de Suscripción de Autenticación (UDM → UDR → HSS)

El flujo central de UDR: el UDM (sirviendo al AUSF durante 5G-AKA) pide a OmniUDR la suscripción de autenticación de un suscriptor; OmniUDR lee el suscriptor y el material clave del HSS y lo mapea a la estructura Nudr.

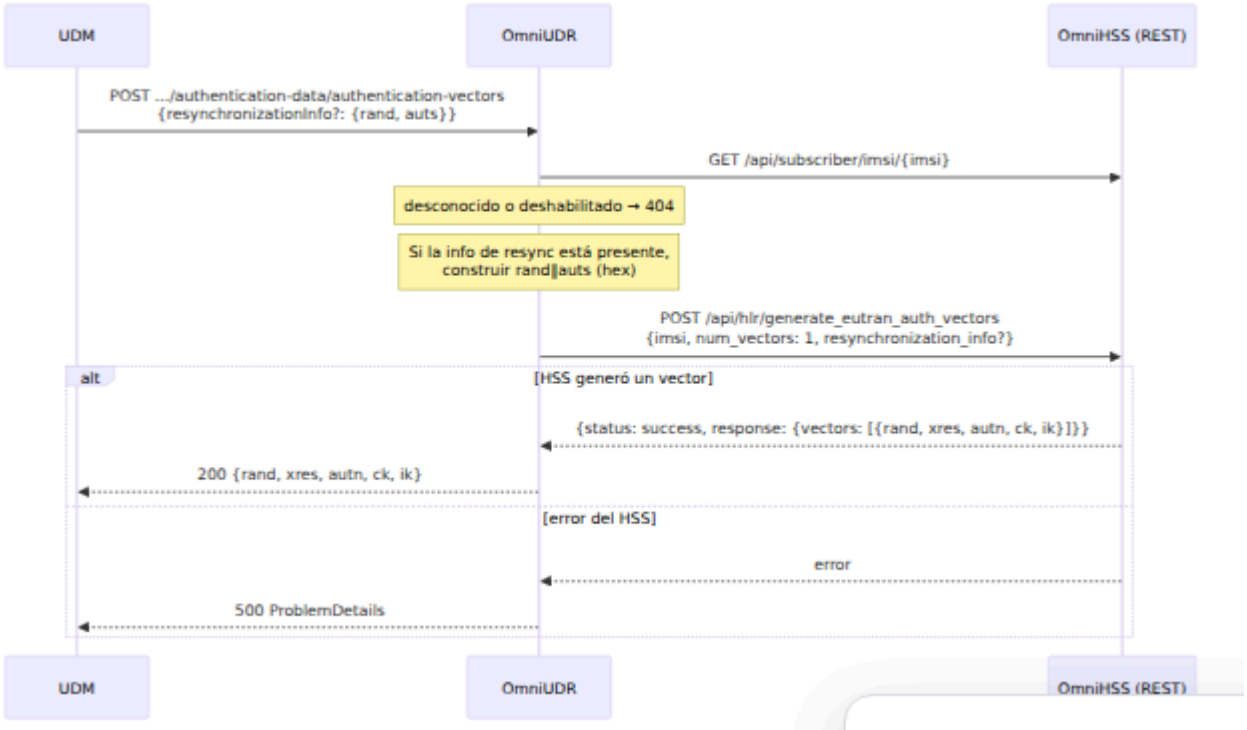


Generación de Vectores de Autenticación (propietario, con resync)

OmniUDM llama al POST no estándar `authentication-vectors` para que la generación de vectores y la gestión de SQN/resync se manejen centralmente en el HSS, y Ki/OPc nunca salgan de él. El HSS devuelve un vector EUTRAN (`rand`, `xres`, `autn`, `ck`, `ik`) cuyo AUTN lleva el AMF aprovisionado del suscriptor; OmniUDM deriva las claves 5G de él. En una resincronización, el UDM pasa `resynchronizationInfo` (`rand` + `auts`), que OmniUDR concatena en el hex `rand‖auts` que el HSS espera.



Lectura de Datos Aprovisionados (AM / SM / Selección de SMF)



Para los datos de AM, el mapeo en su lugar deriva `subscribedUeAmbr` del EPC UE-AMBR y adjunta un NSSAI sintetizado (ver `default_am_nssai`); para los datos de selección de SMF, lista los APNs del suscriptor como información DNN bajo una única clave S-NSSAI suscrita.

Cómo el perfil APN del HSS se convierte en datos de suscripción SM 5G

El registro HSS 4G no tiene estructuras de gestión de sesiones 5G, por lo que OmniUDR sintetiza una entrada `dnnConfigurations` por cada APN aprovisionado (`epc_profile.apn_profiles[]`). Cada campo se deriva de la siguiente manera. Cuando el campo nombrado del HSS está ausente, se sirve el fallback - un fallback en todos los ámbitos significa que los nombres de los campos del perfil QoS del APN no coinciden (ver Solución de Problemas → Los datos de SM sirven un AMBR fijo).

Campo SM 5G	Derivado de (campo HSS)
`singleNssai`	DNN → S-NSSAI a través de `dnn_snssai_`
Clave DNN en `dnnConfigurations`	Nombre APN (`apn_identifier.apn`)
`pduSessionTypes`	APN `apn_identifier.ip_version` (ver ta continuación)
`sscModes`	fijo: `SSC_MODE_1` por defecto, permitido `SSC_MODE_1` + `SSC_MODE_2`
`5gQosProfile.5qi`	`apn_qos_profile.qci`
`5gQosProfile.arp.priorityLevel`	`apn_qos_profile.allocation_retentio`
`5gQosProfile.arp.preemptCap`	`apn_qos_profile.pre_emption_capabil`
`5gQosProfile.arp.preemptVuln`	`apn_qos_profile.pre_emption_vulnera` (bool)
`sessionAmbr.uplink`	`apn_qos_profile.apn_ambr_ul_kbps`
`sessionAmbr.downlink`	`apn_qos_profile.apn_ambr_dl_kbps`
`chargingRules`	APN `pcrf_profile.charging_rules[]` (r precedencia, 5QI/QCI, ARP, GBR/MBR, inf flujo)
`staticIpAddress`	Suscriptor `static_ips[]` cuyo `apn_prof` este perfil APN: `ipv4_static_ip` → `ipv4` `ipv6_static_ip` → `ipv6Addr`

Tipos de sesión PDU. El tipo de sesión PDU 5G es el gemelo 5G del tipo PDN 4G, derivado de la `ip_version` del APN, por lo que un APN IPv6 o de pila dual produce una sesión capaz de IPv6 en lugar de la por defecto solo IPv4. `defaultSessionType` es lo que el SMF utiliza como fallback cuando el UE no solicita un tipo permitido específico.

APN `ip_version`	**`defaultSessionType`**	**`allowedSessionTypes`**
`ipv4`	`IPV4`	`IPV4`
`ipv6`	`IPV6`	`IPV6`
`ipv4v6` / `ipv4_or_ipv6`	`IPV4V6`	`IPV4V6`, `IPV4`, `IPV6`
no establecido / otro	`IPV4`	`IPV4`, `IPV4V6`

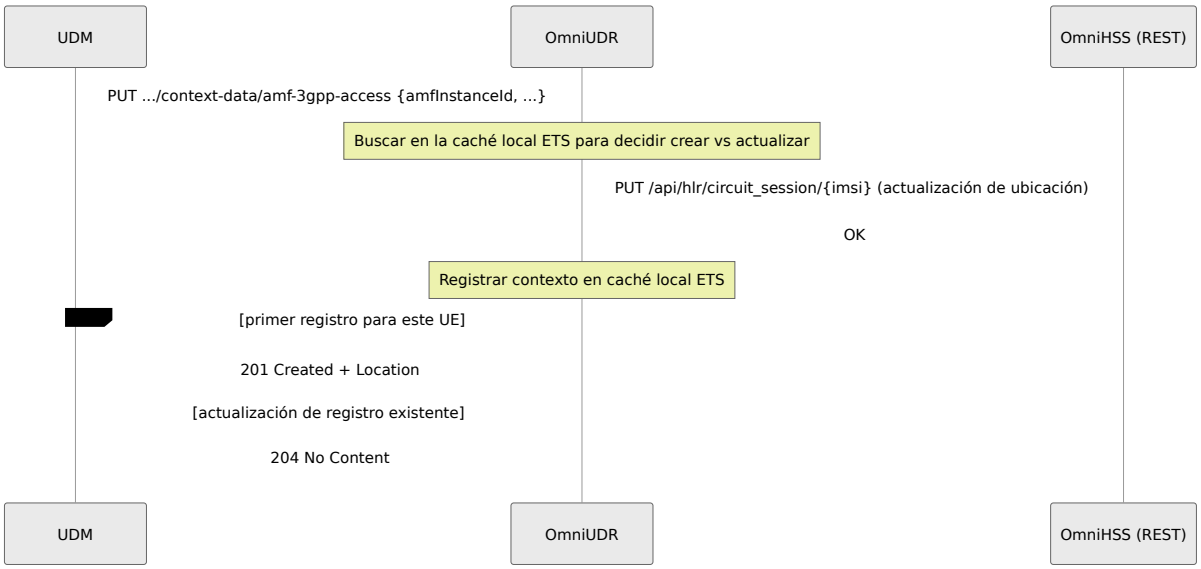
Modos SSC no son aprovisionables por suscriptor: cada DNN se publicita con `SSC_MODE_1` por defecto y `SSC_MODE_1`/`SSC_MODE_2` permitidos. Según TS 29.503 / 29.571 `SscModes.allowedSscModes` tiene `maxItems: 2` (el por defecto más como máximo una alternativa), por lo que el conjunto no puede ampliarse.

Los datos de suscripción de selección de SMF enumeran cada APN aprovisionado como un DNN bajo una única clave S-NSSAI suscrita `01`, independientemente del mapeo por DNN en `dnn_snssai_map`. En un despliegue de múltiples slices donde los DNN viven en diferentes S-NSSAIs, utiliza los datos de suscripción AM/SM (que llevan el S-NSSAI correcto por DNN) para decisiones de slice; la selección de SMF solo publicita la lista de DNN.

Actualización de SQN (PATCH suscripción de autenticación)

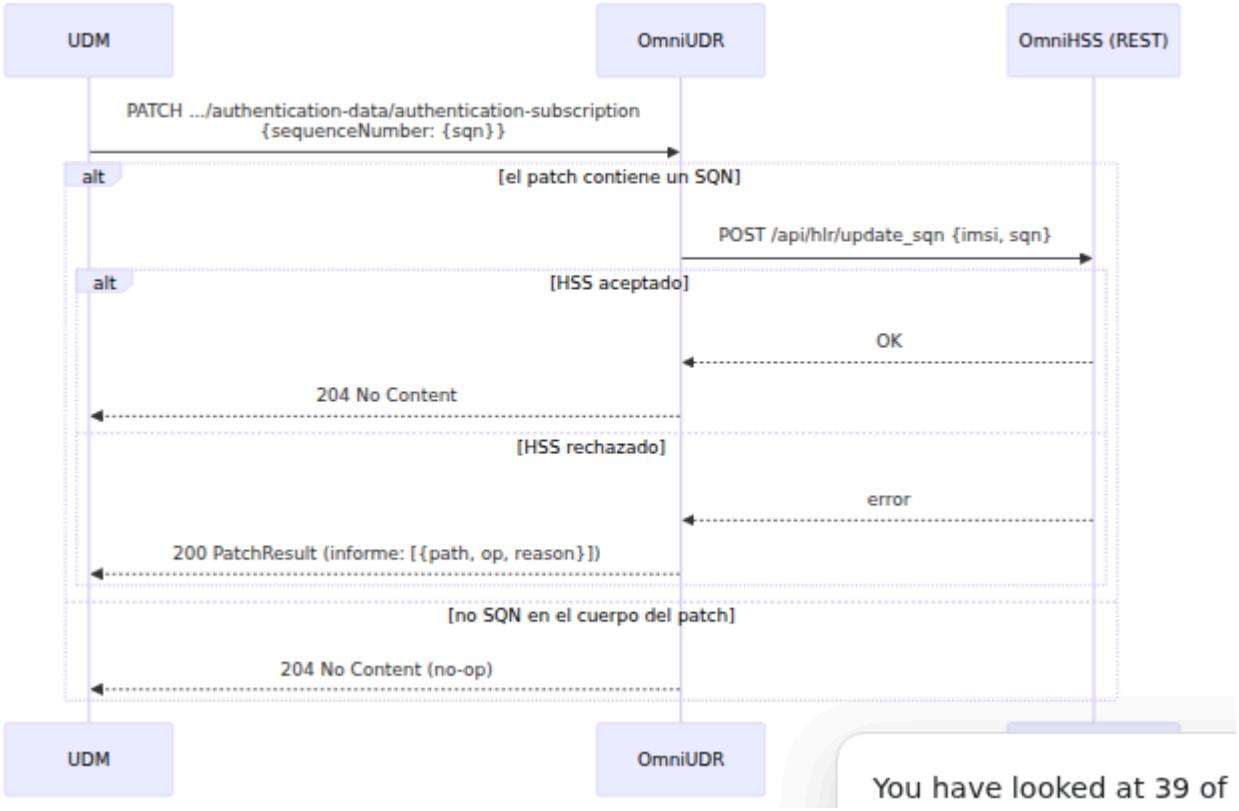


Escritura de Contexto de Registro de AMF



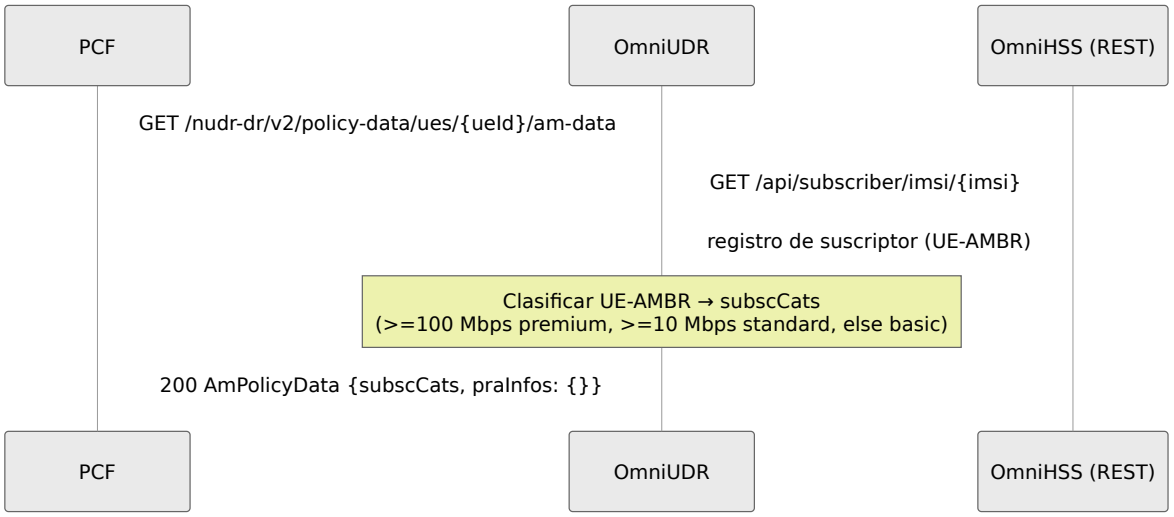
Escritura de Contexto de Registro de SMF (solo caché local)

El contexto de registro de SMF se almacena solo en la caché local ETS, indexado por `{ueId}/{pduSessionId}`; nunca se persiste al HSS y se pierde al reiniciar. La presencia de una entrada de caché existente decide crear vs. actualizar.

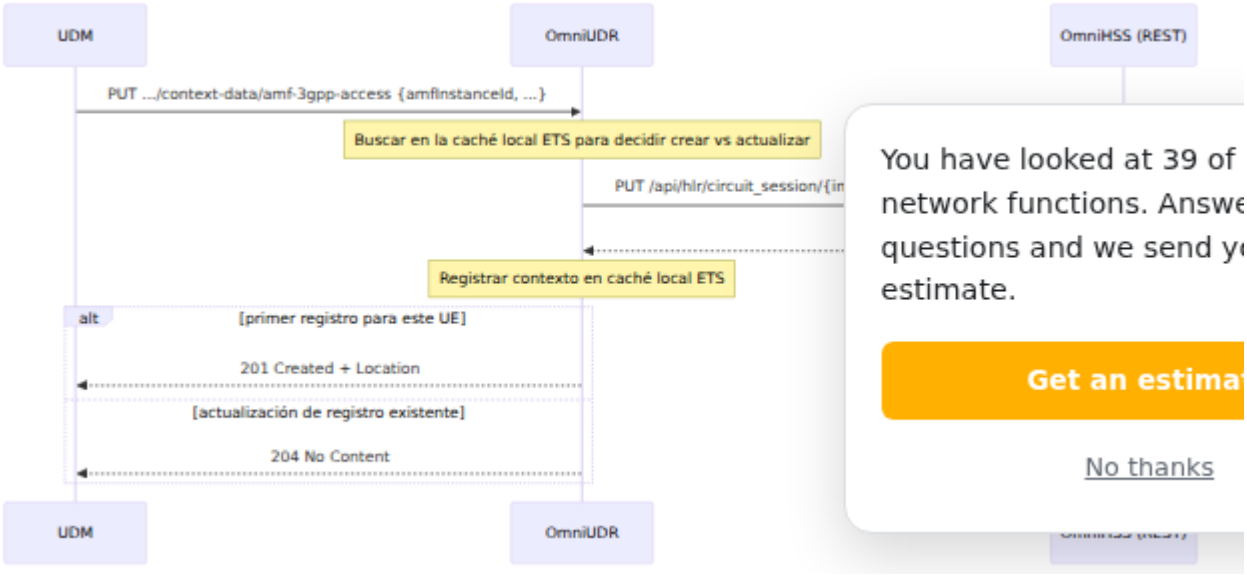


Lectura de Datos de Política AM (PCF → UDR → HSS)

Los datos de política AM son generales: OmniUDR lee el UE-AMBR del suscriptor desde el HSS y lo clasifica en una sola categoría de suscriptor (`premium` / `standard` / `basic`); `praInfos` siempre está vacío.

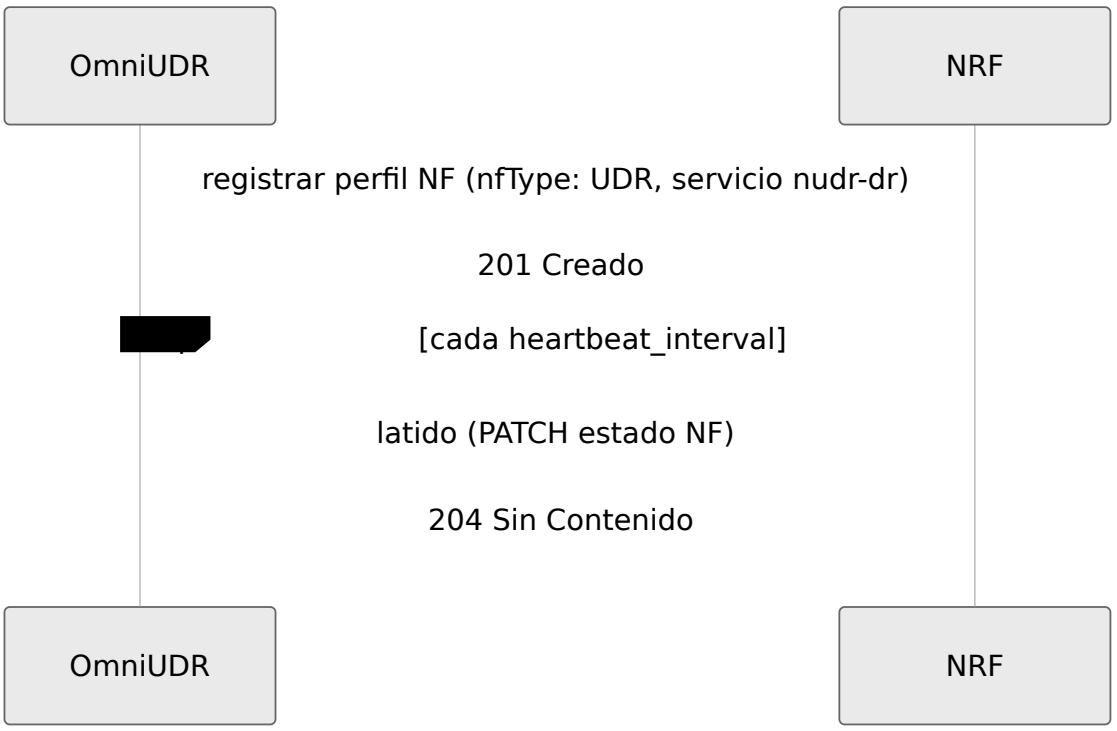


Lectura de Datos de Política SM (PCF → UDR → HSS)



Registro y Latido de NRF

Al iniciar, OmniUDR registra su perfil NF con el NRF y luego envía latidos cada `heartbeat_interval` milisegundos para que el registro siga anunciándolo como activo. El estado de registro se expone como un medidor (ver Métricas), y una acción OAM puede forzar un nuevo registro.



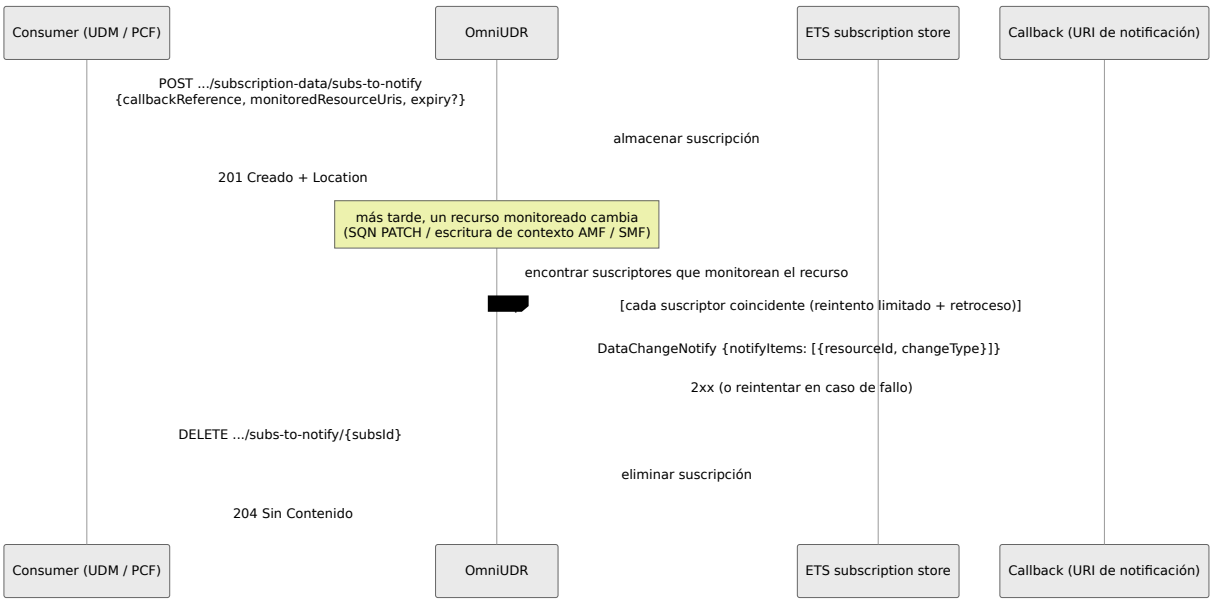
Suscripciones de Cambio de Datos y DataChangeNotify

OmniUDR implementa el servicio de suscripción de cambio de datos Nudr_DR (el recurso `subscription-data/subs-to-notify`, suscripciones TS 29.504 a notificar). Un consumidor registra una suscripción que nombra las URIs de recursos que desea monitorear y una URI de callback de notificación, opcionalmente con una fecha de caducidad. Las suscripciones se mantienen en una tienda de suscripciones en memoria (ETS); una fecha de caducidad, cuando se proporciona, es respetada por la tienda, que elimina la suscripción una vez que ha expirado.

Cuando un recurso monitoreado cambia, OmniUDR emite un `DataChangeNotify` a la URI de callback de cada suscriptor cuyas URIs monitoreadas cubren el recurso cambiado (una coincidencia exacta de URI, o un prefijo de colección para que una suscripción en un recurso padre capture sus recursos hijos). Cada notificación lleva un tipo de cambio: `ADD`, `REPLACE`, o `REMOVE`. En esta versión, las notificaciones se activan en una actualización de SQN (PATCH de la suscripción de autenticación) y en escrituras de contexto

AMF y SMF (un primer registro para un recurso es un `ADD`, una escritura posterior es un `REPLACE`).

La entrega es de mejor esfuerzo y nunca afecta la respuesta SBI que recibe el consumidor. Un callback que falla o es inalcanzable se reintenta con un número limitado de intentos y un retroceso exponencial (ver `notification_max_attempts` y `notification_backoff_ms`); una vez que se agotan los intentos, la notificación se descarta y el evento se registra.



← Resumen

Solución de problemas de OmniUDR

Todas las lecturas de Nudr devuelven 404 / las búsquedas de suscriptores fallan

OmniUDR cumple con cada lectura desde el backend de HSS. Si las lecturas fallan:

1. Confirme que `hss_api_base_url` es accesible desde el host de OmniUDR y que el HSS está activo. Use la verificación incorporada: `GET /api/health/hss` en la API de gestión y observe el medidor `omni_udr_hss_health` (1 = activo, 0 = inactivo).
2. Confirme que el suscriptor existe y está **habilitado** en el HSS. Un suscriptor deshabilitado (`enabled: false`) se trata como no encontrado (404 para el consumidor).
3. Confirme que el `{ueId}` utiliza el formato `imsi-<dígitos>` (también se acepta un IMSI puro).
4. Confirme que `hss_api_key` está configurado y que OmniHSS lo acepta. Un HSS `401` significa que la clave falta o no es conocida por OmniHSS. Un `403` significa que la clave no tiene el rol `hlr_interworking`. Si `hss_api_key` no está configurado, el registro muestra un error que menciona `hss_api_key`.
5. Observe `omni_udr_hss_requests_total{result="failure"}` y `omni_udr_hss_request_duration_ms` para distinguir un HSS caído/lento de un suscriptor genuinamente ausente.

El consumidor recibe 5xx / errores cuando el HSS no es accesible

Cuando el backend de HSS está caído o se agota el tiempo de espera (5 s por solicitud, 3 s para la verificación de salud), el cliente REST de HSS devuelve un error y OmniUDR lo presenta al consumidor (típicamente un `404` en los recursos de lectura, o un `502` en la API de gestión en el proxy de suscriptores). Esto es esperado: OmniUDR no tiene un almacén local de suscriptores al que recurrir. Restaure la accesibilidad del HSS; `omni_udr_hss_health` vuelve a 1 una vez que la siguiente verificación tiene éxito. Tenga en cuenta que `test_subscribers` es una ayuda de laboratorio y **no** es un respaldo automático en caso de caída del HSS en esta versión.

PATCH para la actualización de SQN devuelve 200 en lugar de 204

Un `200` con un cuerpo `PatchResult` significa que el HSS **rechazó** la actualización de SQN; el arreglo `report` nombra la ruta fallida y la razón del HSS. Una actualización exitosa devuelve `204`. Verifique los registros del HSS y la ruta `/api/hlr/update_sqn`. Un cuerpo PATCH sin `sequenceNumber` se trata como un no-op exitoso (`204`).

La generación de vectores de autenticación devuelve 404 / 500

El POST `authentication-vectors` busca al suscriptor y luego delega al endpoint del HSS `/api/hlr/generate_eutran_auth_vectors`. Un `404` significa que el suscriptor es desconocido para el HSS o está deshabilitado; un `500` significa que el HSS devolvió un error o una forma de respuesta inesperada. En una resincronización, asegúrese de que el UDM proporcionó `resynchronizationInfo` con ambos `rand` y `auts` - OmniUDR los concatena en el `rand‖auts` hexadecimal que el HSS espera. Este endpoint es propietario y solo OmniUDM debería llamarlo.

Los datos de política parecen escasos

Los datos de política de AM devuelven solo `subscCats` gruesos derivados del nivel UE-AMBR del suscriptor (`premium` ≥ 100 Mbps, `standard` ≥ 10 Mbps, de lo contrario `basic`), y `praInfos` está vacío. El nivel utiliza el que tenga el UE-AMBR de enlace ascendente o descendente más alto. Cuando el perfil EPC del HSS del suscriptor no lleva UE-AMBR, OmniUDR sirve un valor predeterminado de 1000 Kbps, por lo que un suscriptor no provisionado o mal mapeado cae en `basic` - si un suscriptor que espera que sea `premium`/`standard` muestra `basic`, verifique que el UE-AMBR EPC (`ue_ambr_ul_kbps` / `ue_ambr_dl_kbps`) esté provisionado en el HSS. Los datos de política de SM se construyen a partir del APN/QoS del suscriptor y los perfiles de reglas de carga del PCRF. No hay un almacén de políticas 5G dedicado; esto es intencional en la arquitectura actual.

Desajuste de la porción de datos de AM (S-NSSAI)

El registro HSS 4G no lleva ninguna suscripción de porción 5G, por lo que el NSSAI de datos de AM se sintetiza por OmniUDR y se establece en las porciones de laboratorio estándar (eMBB, URLLC, mIoT). Si un S-NSSAI calificado por SD solicitado por un UE es rechazado por el AMF (causa 5GMM #62), alinee `default_am_nssai` (y, por DNN, `dnn_snssai_map`) con el NSSAI servido por el AMF para la implementación. Consulte la Referencia de Configuración.

Los datos de SM sirven un AMBR fijo para cada suscriptor

Si los datos de suscripción de SM muestran el mismo AMBR de sesión / QoS para todos los suscriptores independientemente del aprovisionamiento, los nombres de campo del perfil de QoS APN del HSS no se están emparejando y OmniUDR está volviendo a los valores predeterminados. El mapeo espera los campos `ApnQosProfile` de OmniHSS (`apn_ambr_dl_kbps`, `apn_ambr_ul_kbps`,

`allocation_retention_priority`, `pre_emption_capability`, `pre_emption_vulnerability`, `qci`). Verifique que el registro del suscriptor en el HSS lleve estos campos.

La caché de contexto parece obsoleta después de cambios fuera de banda en el HSS

La caché de registro AMF/SMF solo rastrea las escrituras de contexto y se utiliza para la decisión de 201 vs 204; no almacena datos de lectura. Si necesita ser limpiada (por ejemplo, después de ciclos de prueba), vacíela a través de `DELETE /api/oam/subscriber-cache/{imsi}` (o `.../all`).