← Visão Geral

Referência de Configuração do OmniUDR

OmniUDR lê sua configuração de operador a partir da chave de ambiente da aplicação `:omniudr`, preenchida na inicialização a partir de `config/runtime.exs`. A maioria das chaves de tempo de execução possui uma sobreposição de variável de ambiente e um padrão. `hss_api_key` não possui nenhuma e é obrigatória. O bloco abaixo é a superfície completa do operador.

Configuração em Tempo de Execução

```
import Config

config :omniudr,
  # Backend da API HSS
  hss_api_base_url: System.get_env("HSS_API_BASE_URL",
"https://127.0.0.1:8443"),
  hss_api_key: "<Chave da API OmniHSS com papel
hlr_interworking>",

  # Configuração do 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")),

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

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

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

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

  # Assinantes de teste - ajuda para configuração de laboratório.
Mapa JSON de IMSI → {ki, opc, amf}.
  test_subscribers: System.get_env("TEST_SUBSCRIBERS")

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

Tabela de Parâmetros

Parâmetro	Tipo	Obrigatório	Padrão
`hss_api_base_url`	string (URL)	Sim	`https://127.0.0.1`
`hss_api_key`	string	Sim	Nenhum

Parâmetro	Tipo	Obrigatório	Padrão
`sbi_scheme`	string	Não	`http`
`sbi_addr`	string (IP)	Não	`127.0.0.22`
`sbi_port`	integer	Não	`7777`
`nrf_uri`	string (URL)	Não	`http://127.0.0.1:`
`mcc`	string	Não	`999`
`mnc`	string	Não	`70`

Parâmetro	Tipo	Obrigatório	Padrão
`prometheus_metrics_port`	integer	Não	`9568`
`heartbeat_interval`	integer (ms)	Não	`10000`
`test_subscribers`	string (JSON) | nil	Não	`nil`

Parâmetro	Tipo	Obrigatório	Padrão
`default_am_nssai`	map	Não	fatias de laboratório eMBB / URLLC / mIo (veja abaixo)
`dnn_snssai_map`	map	Não	`%{"iot" => %{"sst 3, "sd" => "000003`

Parâmetro	Tipo	Obrigatório	Padrão
`notification_max_attempts`	integer	Não	`4`
`notification_backoff_ms`	integer (ms)	Não	`500`

Parâmetro	Tipo	Obrigatório	Padrão

> Tanto `default_am_nssai` quanto `dnn_snssai_map` existem para conectar o modelo de dados HSS 4G (sem conceito de fatia) às estruturas Nudr 5G. Alinhe-os com o plano de fatias real da implantação e com o NSSAI servido pelo AMF/SMF. Os padrões AM/SM (eMBB `sst 1`, URLLC `sst 2 / sd 000002`, mIoT `sst 3 / sd 000003`) são valores de laboratório.

Formas do mapa de síntese de fatia (`default_am_nssai`, `dnn_snssai_map`)

Ambas as chaves são mapas estruturados de objetos S-NSSAI. Um **S-NSSAI** tem dois subcampos:

Subcampo	Tipo	Obrigatório	Significado
`sst`	integer (0–255)	Sim	Tipo de Fatia/Serviço. Valores padronizados: `1` = eMBB, `2` = URLLC, `3` = mIoT (por TS 23.501).
`sd`	string (6 dígitos hexadecimais)	Não	Diferenciador de Fatia. Distingue duas fatias com o mesmo `sst`. Omitir para uma fatia bare-`sst`; quando presente, deve corresponder exatamente ao valor servido pelo AMF/SMF.

`default_am_nssai` é o NSSAI de acesso e mobilidade, com duas listas:

Lista	Significado
`defaultSingleNssais`	Os S-NSSAIs que o UE recebe quando se registra sem solicitar uma fatia específica - o conjunto de fatias padrão da rede para o assinante.
`singleNssais`	O conjunto completo de S-NSSAIs que o assinante está autorizado a usar (um superconjunto de `defaultSingleNssais`). Um UE que solicitar um S-NSSAI qualificado SD não listado aqui é rejeitado pelo AMF com a 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` mapeia uma **string DNN** para o único S-NSSAI em que suas sessões PDU residem. Qualquer DNN não presente no mapa volta para eMBB (`%{"sst" => 1, "sd" => "000001"}`). Mantenha cada S-NSSAI referenciado aqui dentro de `default_am_nssai.singleNssais`, ou o SMF colocará a sessão em uma fatia à qual o assinante não está inscrito.

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

Sobre `hss_api_base_url` (o backend REST do HSS)

Esta é a configuração de operador mais importante. OmniUDR é uma fachada - não possui banco de dados. Quando um consumidor solicita ao OmniUDR a assinatura de autenticação de um assinante, dados AM, dados SM ou dados de política, o OmniUDR emite solicitações REST ao `hss_api_base_url` (principalmente `GET /api/subscriber/imsi/{imsi}` e `GET /api/key_set/{id}`, e para vetores de autenticação / SQN / localização os endpoints `/api/hlr/*`), e então mapeia a resposta do HSS na estrutura 3GPP. Cada solicitação HSS tem um tempo limite de 5 segundos (3 segundos para a verificação de saúde). Se este backend estiver inacessível, as leituras Nudr correspondentes retornam `404`/`5xx` - veja Solução de Problemas.

Sobre `hss_api_key` (controle de acesso do OmniHSS)

OmniHSS usa controle de acesso baseado em papéis em sua API REST. Cada cliente deve enviar sua chave da API no cabeçalho `x-omnihss-key`. O OmniHSS rejeita uma solicitação sem uma chave válida com `401`. O OmniUDR envia `hss_api_key` em cada solicitação para `hss_api_base_url`.

Dê à chave do OmniUDR o papel `hlr_interworking` do OmniHSS. Este papel permite estes endpoints do OmniHSS:

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

O valor é um segredo. O OmniUDR não o escreve no log. Instale esta versão antes de habilitar o controle de acesso no OmniHSS, caso contrário, todas as solicitações do OmniUDR para o OmniHSS falharão com `401`.

Sobre `test_subscribers`

`test_subscribers` preenche um conjunto de assinantes estáticos em código (dois IMSIs internos - vetores de teste Milenage bem conhecidos - além de qualquer um fornecido via o mapa JSON) que retorna dados fixos de autenticação, AM e SM sem nenhuma chamada ao HSS. É uma **ajuda de laboratório** para configuração e testes de conformidade onde a configuração de um HSS completo não é justificada. Na versão atual, o caminho de dados Nudr_DR ao vivo sempre lê do backend HSS; o conjunto de assinantes de teste não está conectado como um fallback automático quando o HSS está fora do ar. Trate esta chave como uma superfície de laboratório/teste apenas e deixe-a não definida (`nil`) em qualquer implantação suportada por um HSS real. Um valor JSON inválido volta para os dois padrões internos.

Identidade PLMN e perfil NRF

`mcc` e `mnc` são anunciados no perfil NF do NRF como o PLMN que serve. A capacidade e prioridade do NF anunciadas ao NRF são gerenciadas através da configuração compartilhada do Omni5G NRF e podem ser alteradas em tempo de execução através da API OAM (`nf_capacity`, `nf_priority`).

Entrega de Notificação

`notification_max_attempts` e `notification_backoff_ms` governam como o OmniUDR entrega notificações Nudr_DR DataChangeNotify aos assinantes de mudança de dados (o serviço `subs-to-notify`, assinaturas TS 29.504). Quando um recurso monitorado muda, o OmniUDR publica uma notificação para cada URI de callback correspondente. Se uma entrega falhar (o callback estiver inacessível ou retornar um status não 2xx), o OmniUDR tenta novamente até `notification_max_attempts` vezes, aguardando `notification_backoff_ms` para a primeira tentativa e dobrando o atraso em cada tentativa subsequente. Uma vez esgotadas as tentativas, a notificação é descartada e o evento é registrado. A entrega é feita com o melhor esforço e nunca afeta a resposta SBI retornada ao consumidor que causou a mudança. Ambas as chaves são apenas configuração da aplicação (sem sobreposição de variável de ambiente): configure-as em `config :omniudr, ...`.

API de Gerenciamento e OAM

Além do listener SBI, o OmniUDR expõe uma API de gerenciamento HTTPS voltada para o operador na porta **8443** (TLS, `enable_tls: true`). Uma interface web do Painel de Controle também é servida sobre HTTPS na porta **7443**.

Superfície	Porta	Propósito
SBI (Nudr_DR)	`sbi_port` (padrão `7777`)	Interface de serviço 3GPP
Métricas do Prometheus	`prometheus_metrics_port` (padrão `9568`)	Endpoint de coleta de métricas
API de Gerenciamento / OAM	`8443` (HTTPS, montada sob `/api`)	Status, saúde do HSS, proxy de assinantes, ações OAM
UI do Painel de Controle	`7443` (HTTPS)	Recursos da aplicação, configuração, licença e páginas de log

A API de gerenciamento é montada sob o prefixo de caminho `/api`. Ela fornece status somente leitura, uma verificação de conectividade HSS, um proxy de assinante somente leitura para o HSS e ações OAM:

Método	Caminho	Propósito
GET	`/api/status/api`	Status da API/serviço
GET	`/api/status/nrf`	Status de registro do NRF
GET	`/api/status/nf`	Perfil / status do NF
GET	`/api/status/license`	Status da licença
GET	`/api/health/hss`	Verificação de conectividade HSS (chama `GET /api/status/api` no HSS)
GET	`/api/subscribers`	Lista de assinantes somente leitura paginada, proxy do HSS (`page`, `page_size`)
GET	`/api/subscribers/{imsi}`	Detalhes completos do assinante, proxy do HSS
GET	`/api/oam/config`	Ver configuração de tempo de execução ao vivo
PATCH	`/api/oam/config/{id}`	Atualizar `hss_api_base_url`, `nrf_uri`, `nf_capacity`, `nf_priority` ou `log_level` em tempo de execução
POST	`/api/oam/nrf/reregister`	Forçar re-registro do NRF
DELETE	`/api/oam/subscriber-cache/{id}`	Limpar o contexto AMF/SMF em cache para um assinante (`id=all` limpa todos os assinantes)

Método	Caminho	Propósito
POST	`/api/oam/log_level`	Alterar o nível de log em tempo de execução

Notas sobre a atualização de configuração OAM (`PATCH /api/oam/config/{id}`):

- Apenas `hss_api_base_url`, `nrf_uri`, `nf_capacity`, `nf_priority` e `log_level` são aceitos; qualquer outra chave é ignorada, e uma solicitação sem chave válida retorna 400.
- `nf_capacity` e `nf_priority` devem ser inteiros; `log_level` deve ser um dos seguintes: `emergency`, `alert`, `critical`, `error`, `warning`, `warn`, `notice`, `info`, `debug`.
- Alterar `nf_capacity` / `nf_priority` persiste o valor e empurra um perfil atualizado para o NRF.
- O proxy de assinante é somente leitura. Se o HSS estiver inacessível, `/api/subscribers` e `/api/subscribers/{imsi}` retornam `502`; um assinante desativado ou ausente retorna `404`.

Registro

OmniUDR emite logs JSON estruturados através de `OmniLogger.JsonFormatter` com metadados completos. O nível de log em tempo de execução pode ser alterado sem reinicialização através da API OAM (`POST /api/oam/log_level` ou `PATCH /api/oam/config/{id}` com `log_level`). Ações OAM (atualizações de configuração, limpezas de cache, re-registro) são registradas como `info` com um prefixo `[OAM]`; falhas de solicitação HSS são registradas como `warning`/`error`.

← Visão Geral

Métricas do OmniUDR

OmniUDR expõe métricas Prometheus no endpoint de métricas (`prometheus_metrics_port`, padrão `9568`). Os nomes das métricas abaixo usam a forma de exposição do Prometheus (pontos se tornam sublinhados). Rótulos e seus valores estão em letras minúsculas.

Métricas UDR / HSS

Métrica	Tipo	Rótulos	Descr
`omni_udr_nrf_registration_status`	gauge	`nf_type`	Status de do NRF (1 registrado não)
`omni_udr_hss_health`	gauge	-	Saúde da conectivid OmniHSS (ativo, 0 = atualizado cada verifi `/health/h`
`omni_udr_hss_requests_total`	counter	`endpoint`, `result`	Contagem chamadas para o bac HSS, etiqu pelo camir solicitação (`endpoint` `result` (`success`/
`omni_udr_hss_request_duration_ms`	distribution	`endpoint`	Latência d solicitação HSS em milissegur Baldes: 5, 50, 100, 2 1000, 250

Métricas da VM BEAM

Métrica	Tipo	Descrição
`beam_memory_total`	gauge	Memória total do BEAM em bytes
`beam_memory_processes`	gauge	Memória usada por processos Erlang
`beam_memory_system`	gauge	Memória do sistema (ETS, átomos, código, binários)
`beam_processes_count`	gauge	Número de processos Erlang
`beam_vm_uptime`	gauge	Tempo de atividade da VM em segundos

PromQL Útil

- Disponibilidade do backend HSS: `omni_udr_hss_health`
- UDR registrado com NRF: `omni_udr_nrf_registration_status`
- Taxa de erro do HSS (5 min): `sum(rate(omni_udr_hss_requests_total{result="failure"}[5m])) / sum(rate(omni_udr_hss_requests_total[5m]))`
- Latência p95 do HSS (5 min): `histogram_quantile(0.95, sum(rate(omni_udr_hss_request_duration_ms_bucket[5m])) by (le, endpoint))`
- Taxa de solicitação do HSS por endpoint: `sum(rate(omni_udr_hss_requests_total[1m])) by (endpoint)`
- Contagem de processos BEAM: `beam_processes_count`

← Visão Geral

Guia de Operações do OmniUDR

OmniUDR é o Repositório de Dados Unificado (UDR) do núcleo 5G da Omnitouch. Ele implementa o serviço Nudr_DataRepository (TS 29.504) e expõe dados estruturados de assinantes, políticas e contexto para o UDM e PCF através da interface baseada em serviços (SBI).

OmniUDR não possui dados de assinantes. É uma fachada Nudr_DR fina e compatível com padrões sobre o **backend de provisionamento REST do OmniHSS**: cada leitura do repositório de dados é traduzida no momento da solicitação em uma ou mais chamadas REST do OmniHSS, e a resposta do HSS é mapeada para o modelo de dados Nudr do 3GPP antes de ser retornada ao consumidor. OmniUDR é a única função da rede núcleo 5G que se comunica diretamente com a API REST do OmniHSS - o UDM (e, através dele, o AUSF) acessam os dados do assinante chamando o OmniUDR via Nudr padrão, e não chamando o HSS.

Como o HSS é o sistema de registro, o OmniUDR quase não mantém estado persistente. Seu único estado local é três pequenos armazenamentos em memória (ETS): um cache de contexto de registro AMF e SMF (usado para decidir se uma gravação de contexto é uma criação ou uma atualização), um armazenamento de assinaturas de mudança de dados que mantém as assinaturas Nudr_DR `subs-to-notify` ativas, e um armazenamento de dados de aplicação que mantém as famílias de `application-data` do Nudr_DR (PFDs, influência de tráfego, BDT). Nenhum desses é persistido; todos são limpos na reinicialização. O cache de contexto de registro pode ser limpo através da API de gerenciamento.

Função e Referências de Especificação 3GPP

Especificação	Relevância
TS 23.501	Arquitetura do sistema 5G - função do UDR e sua relação com UDM, PCF e a camada de dados do UDM
TS 29.504	Serviço Nudr_DataRepository - estrutura de recursos, dados de assinatura, dados de contexto, dados de política
TS 29.505	Estruturas de dados de assinatura - assinatura de autenticação, acesso e mobilidade, gerenciamento de sessão, seleção de SMF
TS 29.519	Estruturas de dados de política - dados de política AM e formas de dados de política SM
TS 29.503	Serviços Nudm - o consumidor UDM cujos procedimentos UEAU/SDM/UECM impulsionam leituras de UDR e gravações de contexto
TS 29.571	Tipos de dados comuns - `PatchResult`, `ProblemDetails`, codificações PLMN e S-NSSAI

Interfaces

Interface	Bind	Propósito
SBI (Nudr_DR)	`{sbi_scheme}://{sbi_addr}:{sbi_port}` (padrão `http://127.0.0.22:7777`)	Serve os recursos `nudr-dr/v2` consumidos pelo UDM e PCF
Backend REST do HSS	`hss_api_base_url` (padrão `https://127.0.0.1:8443`)	Chamadas REST de saída para o OmniHSS que atendem cada leitura/atualização. Cada chamada envia `hss_api_key` no cabeçalho `x-omnihss-key` (papel do OmniHSS `hlr_interworking`)
NRF	`nrf_uri`	Registro de NF e heartbeat periódico
API de Gerenciamento / OAM	`:8443` (HTTPS, montado sob `/api`)	Status, saúde do HSS, proxy de assinante e ações OAM - veja Referência de Configuração
Métricas do Prometheus	`prometheus_metrics_port` (padrão `9568`)	Endpoint de coleta de métricas - veja Métricas

Conjuntos de Dados Servidos

OmniUDR serve os seguintes conjuntos de dados Nudr_DR. Cada um é derivado no momento da leitura do registro do assinante do OmniHSS; a coluna "Fonte HSS" nomeia os dados REST subjacentes dos quais o valor é mapeado.

Conjunto de dados	Família de recursos Nudr	Acesso	Consum
Assinatura de autenticação	`subscription-data/.../authentication-data/authentication-subscription`	Leitura / atualização	UDM (Nudm_U
Vetores de autenticação	`subscription-data/.../authentication-data/authentication-vectors`	Gerar (proprietário)	OmniUDI
Dados de acesso e mobilidade (AM)	`subscription-data/.../provisioned-data/am-data`	Leitura	UDM (Nudm_S
Dados de gerenciamento de sessão (SM)	`subscription-data/.../provisioned-data/sm-data`	Leitura	UDM (Nudm_S
Dados de seleção de SMF	`subscription-data/.../provisioned-data/smf-selection-subscription-data`	Leitura	UDM (Nudm_S
Contexto de registro AMF	`subscription-data/.../context-data/amf-3gpp-access`	Gravação	UDM (Nudm_U
Contexto de registro SMF	`subscription-data/.../context-data/smf-registrations/{pduSessionId}`	Gravação	UDM (Nudm_U

Conjunto de dados	Família de recursos Nudr	Acesso	Consum
Dados de política AM	`policy-data/ues/{ueId}/am-data`	Leitura	PCF
Dados de política SM	`policy-data/ues/{ueId}/sm-data`	Leitura	PCF

| Dados de aplicação | `application-data/{family}/{id}` (famílias `pfds`, `influenceData`, `bdtData`) | Criar / ler / substituir / excluir | PCF, NEF | Nenhum - mantido em um armazenamento local em memória (ETS), não respaldado pelo HSS e não persistido |

> Os dados de aplicação são servidos por um armazenamento genérico em memória em vez do HSS. Não é persistido: uma reinicialização o limpa. Veja Endpoints SBI → Dados de Aplicação abaixo.

Endpoints SBI

Todos os endpoints Nudr_DR são servidos via HTTP em `{sbi_scheme}://{sbi_addr}:{sbi_port}` (padrão `http://127.0.0.22:7777`) com `Content-Type: application/json`. Falhas são retornadas como `ProblemDetails` do TS 29.571.

O segmento de caminho `{ueId}` é um SUPI na forma `imsi-<dígitos>` (um IMSI nu também é aceito). O segmento `{servingPlmnId}` nas leituras de dados provisionados é validado: deve ser um PLMN bem formado (5 ou 6 dígitos) e deve corresponder ao PLMN em serviço configurado a partir de `mcc`/`mnc`, caso contrário, a leitura é rejeitada com `400 MANDATORY_IE_INCORRECT`. Esta versão serve um único registro por assinante.

Dados de Assinatura - Autenticação (TS 29.504 / 29.505)

Método	Caminho	Descrição	Sucesso
GET	`/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-subscription`	Recuperar assinatura de autenticação (método, AMF, SQN)	200
PATCH	`/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-subscription`	Atualizar assinatura de autenticação (SQN)	204, ou 200 + `PatchResult` em falha parcial
POST	`/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-vectors`	Gerar vetores de autenticação (veja a nota)	200

> **Endpoint não padrão.** O POST `authentication-vectors` é uma extensão proprietária do Omni 5GC; TS 29.504/29.505 não definem tal recurso. A geração de vetores e o gerenciamento de SQN/resync são delegados ao UDR, portanto, são tratados centralmente no HSS. É consumido apenas pelo OmniUDM; um consumidor de terceiros compatível com padrões nunca o chamará.

Dados de Assinatura - Provisionados (TS 29.505)

Método	Caminho	Descrição
GET	`/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/am-data`	Recuperar dados de assinatura de acesso e mobilidade
GET	`/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/sm-data`	Recuperar dados de assinatura de gerenciamento de sessão
GET	`/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/smf-selection-subscription-data`	Recuperar dados de assinatura de seleção de SMF

Dados de Assinatura - Contexto (TS 29.504)

Método	Caminho	Descrição	Sucesso
PUT	`/nudr-dr/v2/subscription-data/{ueId}/context-data/amf-3gpp-access`	Armazenar contexto de registro de acesso 3GPP AMF	201 + `Location` (novo) ou 204 (atualização)
PUT	`/nudr-dr/v2/subscription-data/{ueId}/context-data/smf-registrations/{pduSessionId}`	Armazenar contexto de registro SMF	201 + `Location` (novo) ou 204 (atualização)

Um corpo vazio em qualquer PUT de contexto retorna 400 `ProblemDetails` (`cause: MANDATORY_IE_MISSING`).

Dados de Política (TS 29.519)

Método	Caminho	Descrição	Sucesso
GET	`/nudr-dr/v2/policy-data/ues/{ueId}/am-data`	Recuperar dados de política AM (`subscCats`, `praInfos`)	200
GET	`/nudr-dr/v2/policy-data/ues/{ueId}/sm-data`	Recuperar dados de política SM (`smPolicySnssaiData` por DNN)	200

Dados de Aplicação (TS 29.504)

CRUD genérico sobre as famílias de `application-data` do Nudr_DR. O segmento de caminho `{family}` é um dos `pfds`, `influenceData` ou `bdtData`; o

segmento `{id}` é o id do recurso da família (`appId` / `influenceId` / `bdtReferenceId`). Os registros são mantidos em um armazenamento local em memória (ETS), não no HSS, e não são persistidos entre reinicializações. Qualquer outro valor `{family}` cai no 404 catch-all.

Método	Caminho	Descrição	Sucesso
GET	`/nudr-dr/v2/application-data/{family}`	Listar todos os recursos na família (GET de coleção)	200 (array)
GET	`/nudr-dr/v2/application-data/{family}/{id}`	Recuperar um recurso	200, ou 404 se desconhecido
PUT	`/nudr-dr/v2/application-data/{family}/{id}`	Criar ou substituir um recurso	201 + `Location` (novo) ou 200 (substituir)
DELETE	`/nudr-dr/v2/application-data/{family}/{id}`	Excluir um recurso	204, ou 404 se desconhecido

Um corpo vazio no PUT retorna 400 `ProblemDetails` (`cause: MANDATORY_IE_MISSING`).

Assinaturas de Mudança de Dados (TS 29.504 subscriptions-to-notify)

Método	Caminho	Descrição	Sucesso
POST	`/nudr-dr/v2/subscription-data/subs-to-notify`	Criar uma assinatura de mudança de dados (o corpo nomeia os URIs de recurso monitorados e o URI de callback de notificação)	201 + `Location`, ou 400 em um corpo ruim
DELETE	`/nudr-dr/v2/subscription-data/subs-to-notify/{subsId}`	Remover uma assinatura de mudança de dados	204, ou 404 se a assinatura for desconhecida

Veja Assinaturas de Mudança de Dados e DataChangeNotify para o comportamento de fan-out.

Qualquer caminho que não corresponda aos recursos acima é respondido com um 404 `ProblemDetails`.

Chamadas REST do Backend HSS

Cada operação SBI é atendida por uma ou mais chamadas ao backend REST do OmniHSS em `hss_api_base_url`. A relação é:

Operação Nudr_DR	Chamada(s) REST do HSS	Notas
Leitura de assinatura de autenticação	`GET /api/subscriber/imsi/{imsi}`, então `GET /api/key_set/{id}` se o material da chave for referenciado por id	Material da chav registro do assin a segunda cham
Geração de vetor de autenticação	`GET /api/subscriber/imsi/{imsi}`, então `POST /api/hlr/generate_eutran_auth_vectors`	Corpo `{imsi, num_vectors: 1, resynchronizati resynchronizati` `rand‖auts` hex n
Atualização de SQN (PATCH)	`POST /api/hlr/update_sqn`	Corpo `{imsi, sq`
Leitura de AM / SM / Seleção de SMF	`GET /api/subscriber/imsi/{imsi}`	Busca de assinar mapeada localm
Gravação de contexto AMF	`PUT /api/hlr/circuit_session/{imsi}`	Atualização de localização/sessã circuito; também armazenada loca
Gravação de contexto SMF	(nenhuma)	Cache local ETS a
Leitura de dados de política (AM / SM)	`GET /api/subscriber/imsi/{imsi}`	Reutiliza o mape AM/SM e o remo

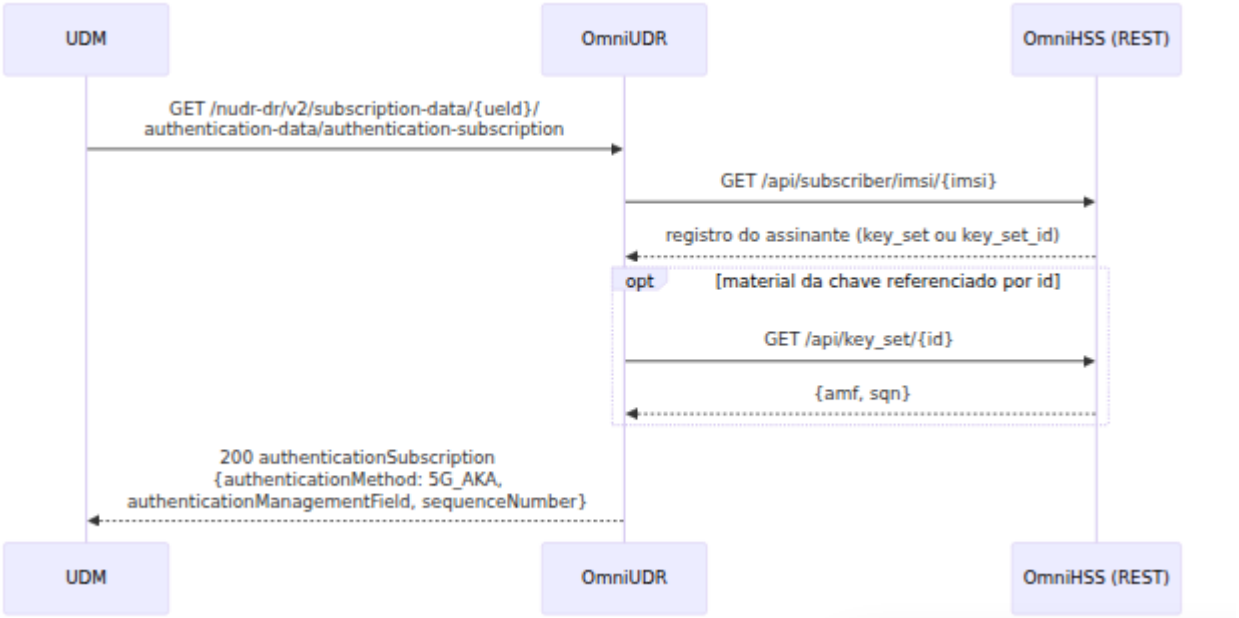
Operação Nudr_DR	Chamada(s) REST do HSS	Notas
Verificação de saúde	`GET /api/status/api`	Impulsiona o end gerenciamento `/health/hss` e o de saúde

> Os endpoints `/api/hlr/*` (`generate_eutran_auth_vectors`, `update_sqn`, `circuit_session/{imsi}`) são a interface REST projetada do HSS para as operações de vetor de autenticação, SQN e localização do OmniUDR, juntamente com os endpoints de assinante/conjunto de chaves (`/api/subscriber/*`, `/api/key_set/*`) e a verificação de saúde (`/api/status/api`). Um assinante desativado (`enabled: false`) é tratado como não encontrado.

Procedimentos Chave

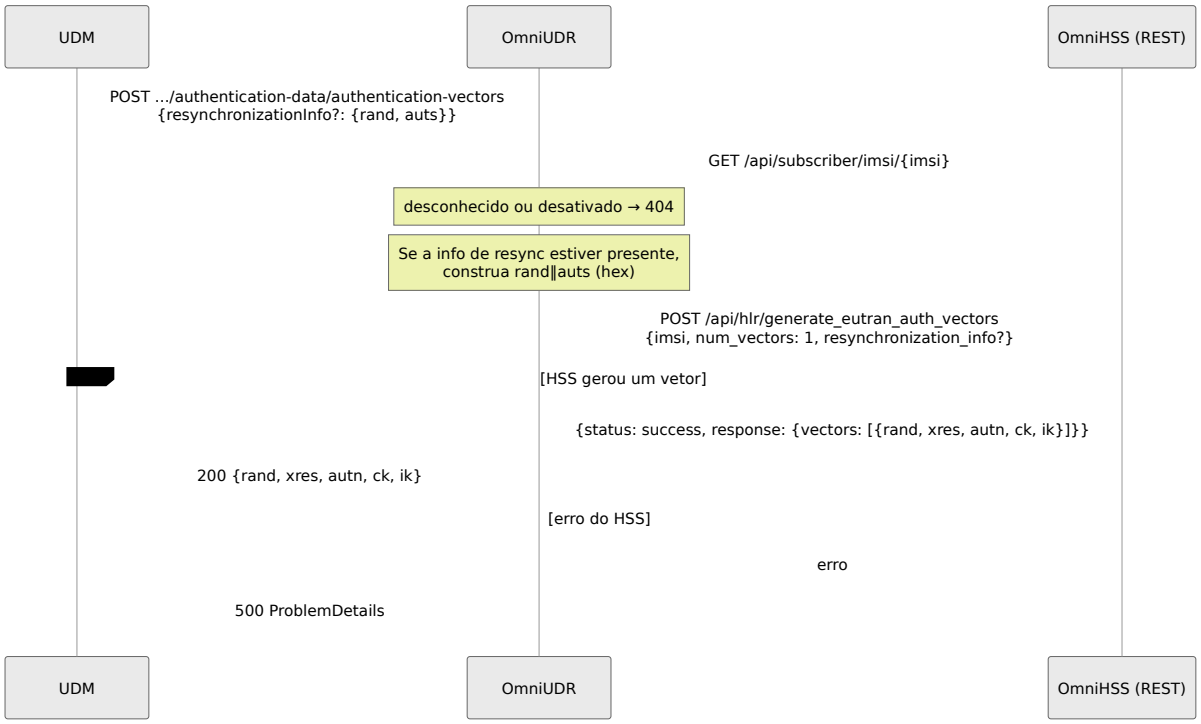
Leitura de Assinatura de Autenticação (UDM → UDR → HSS)

O fluxo central do UDR: o UDM (servindo o AUSF durante 5G-AKA) pede ao OmniUDR a assinatura de autenticação de um assinante; o OmniUDR lê o assinante e o material da chave do HSS e o mapeia para a estrutura Nudr.

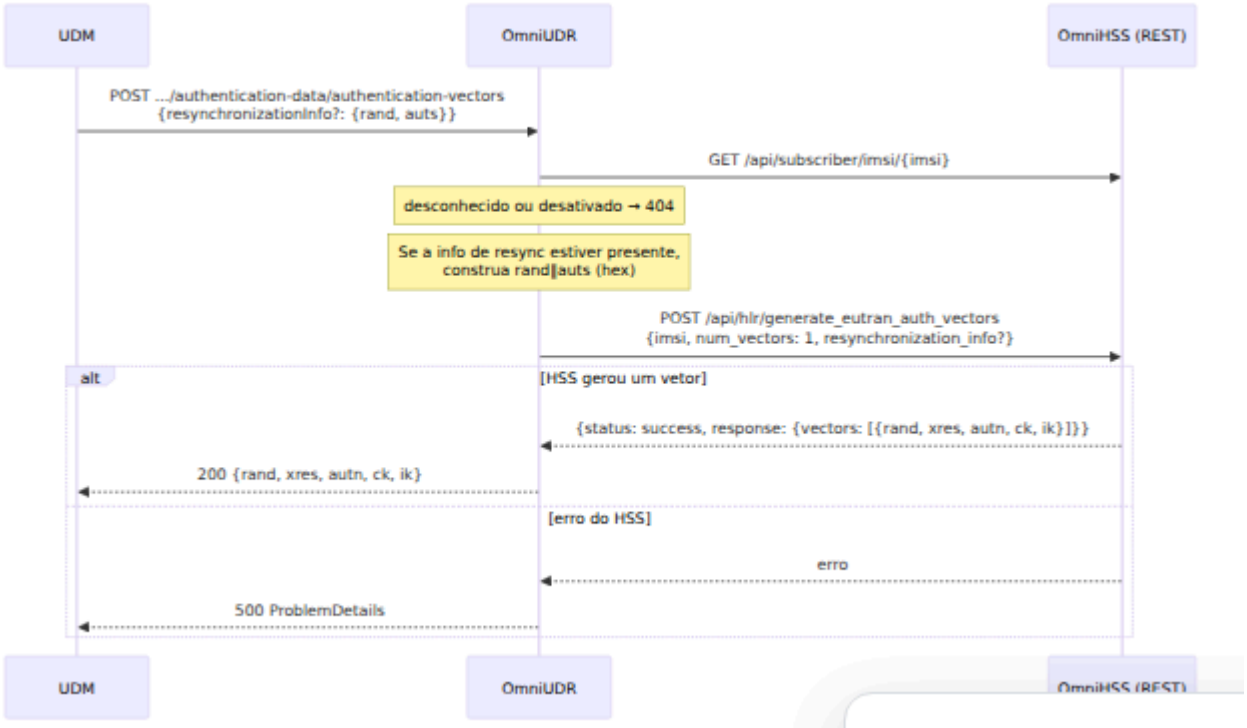


Geração de Vetor de Autenticação (proprietário, com resync)

OmniUDM chama o POST não padrão `authentication-vectors` para que a geração de vetores e SQN/resync sejam tratadas centralmente no HSS, e Ki/OPc nunca saiam dele. O HSS retorna um vetor EUTRAN (`rand`, `xres`, `autn`, `ck`, `ik`) cujo AUTN carrega o AMF provisionado do assinante; o OmniUDM deriva as chaves 5G a partir dele. Em uma ressincronização, o UDM passa `resynchronizationInfo` (`rand` + `auts`), que o OmniUDR concatena no hex `rand‖auts` que o HSS espera.



Leitura de Dados Provisionados (AM / SM / Seleção de SMF)



Para dados AM, o mapeamento em vez disso deriva `subscribedUeAmbr` do EPC UE-AMBR e anexa um NSSAI sintetizado (veja `default_am_nssai`); para dados de seleção de SMF, lista os APNs do assinante como informações DNN sob uma única chave S-NSSAI assinada `01`.

Como o perfil APN do HSS se torna dados de assinatura de SM 5G

O registro do HSS 4G não possui estruturas de gerenciamento de sessão 5G, portanto, o OmniUDR sintetiza uma entrada `dnnConfigurations` por APN provisionado (`epc_profile.apn_profiles[]`). Cada campo é derivado da seguinte forma. Quando o campo nomeado do HSS está ausente, o fallback é servido - um fallback em todo o conjunto significa que os nomes dos campos do perfil de QoS do APN não estão correspondendo (veja Solução de Problemas → Dados de SM servem um AMBR fixo).

Campo SM 5G	Derivado de (campo HSS)
`singleNssai`	DNN → S-NSSAI via `dnn_snssai_map`
Chave DNN em `dnnConfigurations`	Nome do APN (`apn_identifier.apn`)
`pduSessionTypes`	APN `apn_identifier.ip_version` (veja t abaixo)
`sscModes`	fixo: padrão `SSC_MODE_1`, permitido `SSC_` `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`	Regras de cobrança do APN `pcrf_profile.charging_rules[]` (nome precedência, 5QI/QCI, ARP, GBR/MBR, inf fluxo)
`staticIpAddress`	Assinante `static_ips[]` cujo `apn_profi` este perfil de APN: `ipv4_static_ip` → `ip`

Campo SM 5G	Derivado de (campo HSS)
	`ipv6_static_ip` → `ipv6Addr`

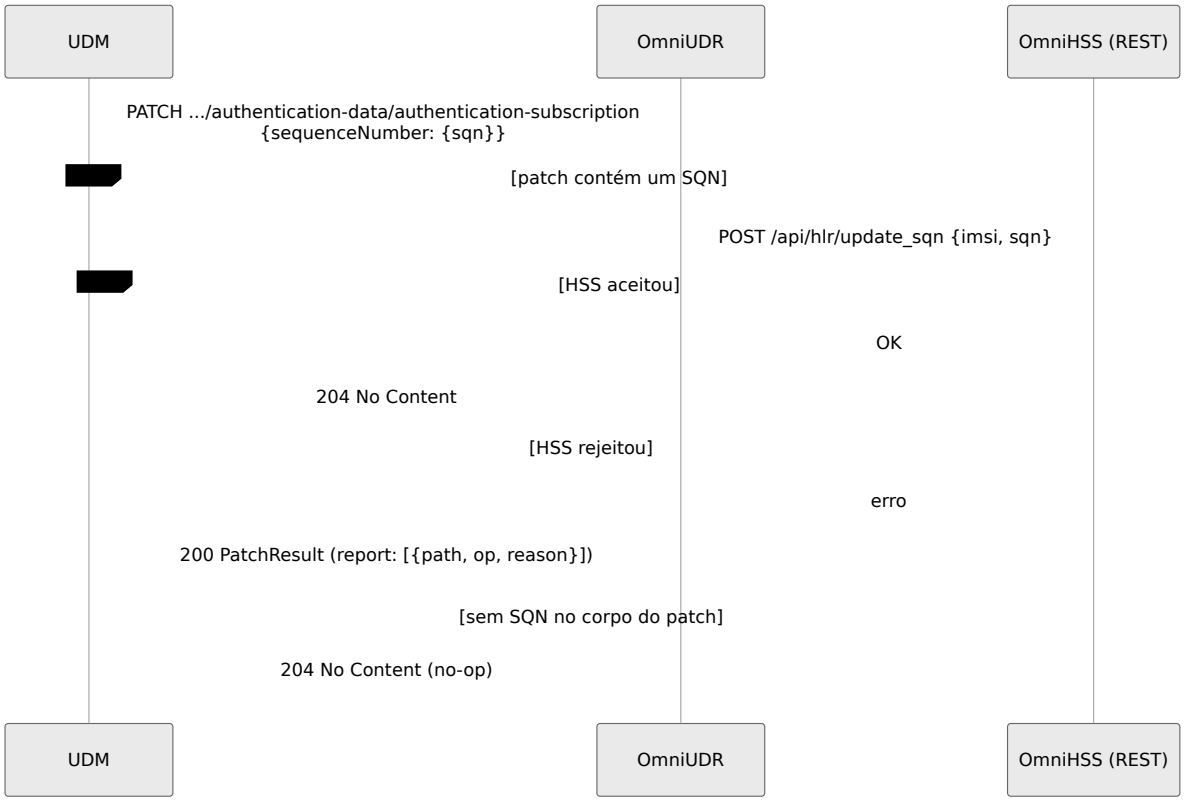
Tipos de sessão PDU. O tipo de sessão PDU 5G é o gêmeo 5G do PDN-Type 4G, derivado do `ip_version` do APN, de modo que um APN IPv6 ou de pilha dupla gera uma sessão capaz de IPv6 em vez do padrão apenas IPv4. `defaultSessionType` é o que o SMF recorre quando o UE não solicita um tipo específico permitido.

APN `ip_version`	`defaultSessionType`	`allowedSessionTypes`
`ipv4`	`IPV4`	`IPV4`
`ipv6`	`IPV6`	`IPV6`
`ipv4v6` / `ipv4_or_ipv6`	`IPV4V6`	`IPV4V6`, `IPV4`, `IPV6`
não definido / outro	`IPV4`	`IPV4`, `IPV4V6`

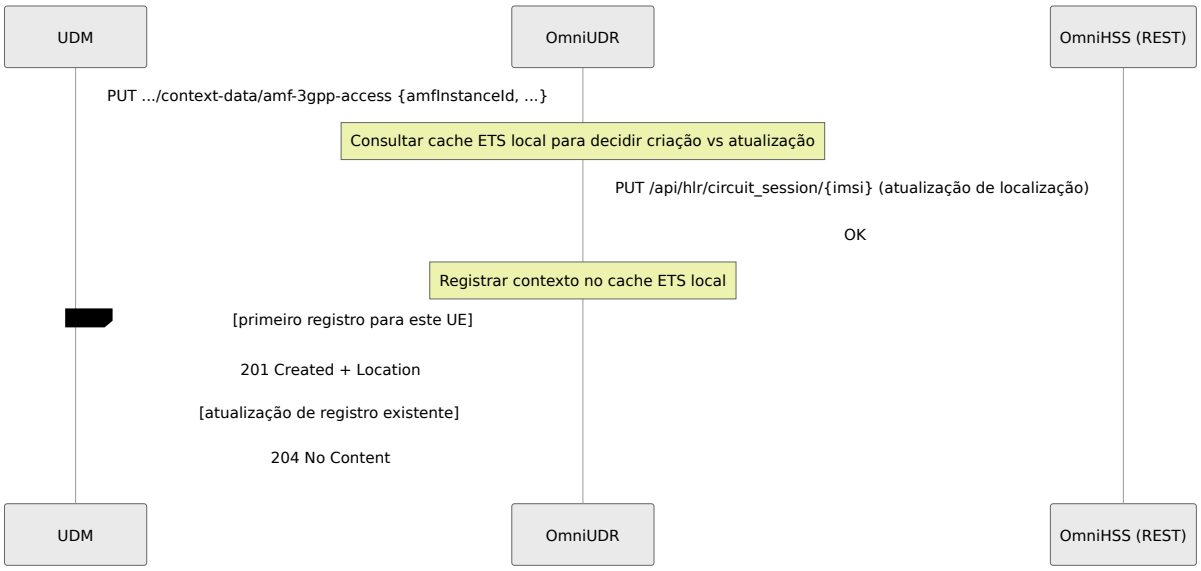
Modos SSC não são provisionáveis por assinante: cada DNN é anunciada com `SSC_MODE_1` padrão e `SSC_MODE_1`/`SSC_MODE_2` permitidos. De acordo com TS 29.503 / 29.571, `SscModes.allowedSscModes` tem `maxItems: 2` (o padrão mais no máximo uma alternativa), portanto, o conjunto não pode ser ampliado.

Dados de assinatura de seleção de SMF lista cada APN provisionado como um DNN sob uma única chave S-NSSAI assinada `01`, independentemente do mapeamento por DNN em `dnn_snssai_map`. Em uma implantação de múltiplas fatias onde DNNs residem em diferentes S-NSSAIs, use os dados de assinatura AM/SM (que carregam o S-NSSAI correto por DNN) para decisões de fatia; a seleção de SMF anuncia apenas a lista de DNNs.

Atualização de SQN (PATCH assinatura de autenticação)

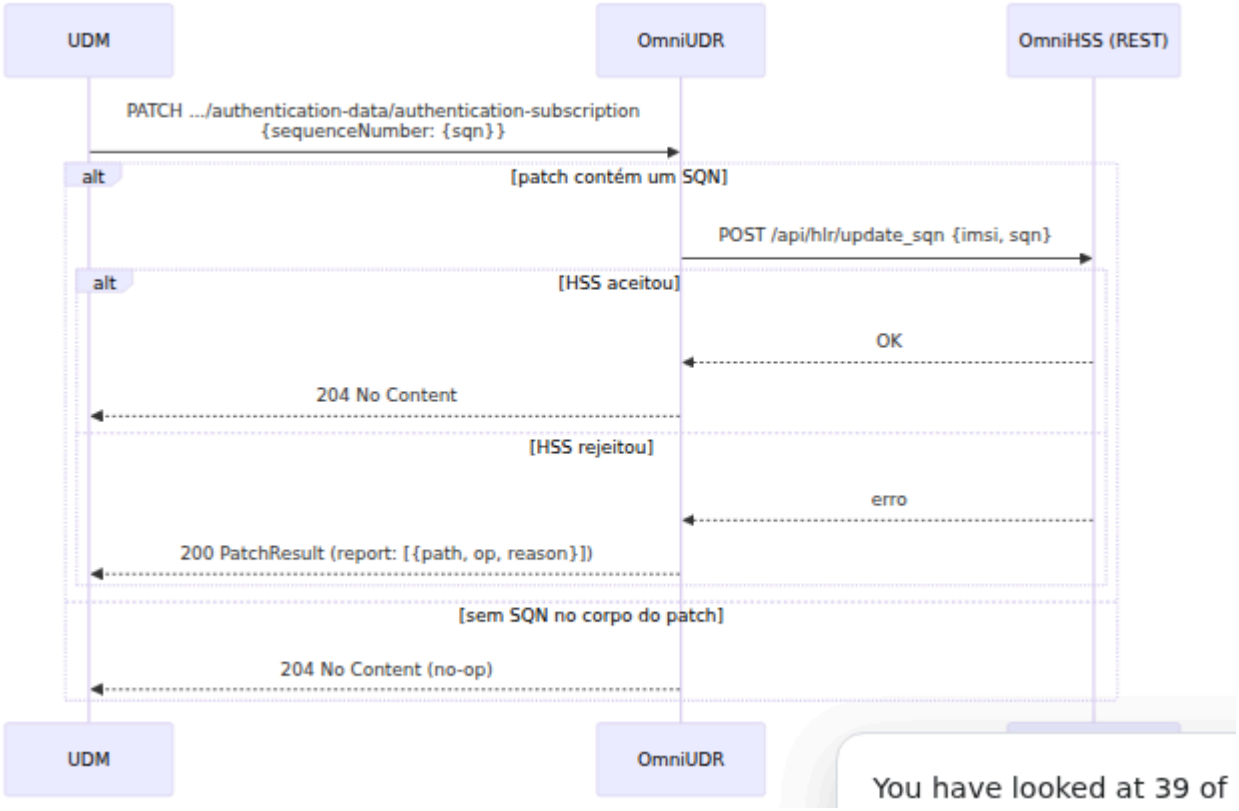


Gravação de Contexto de Registro AMF



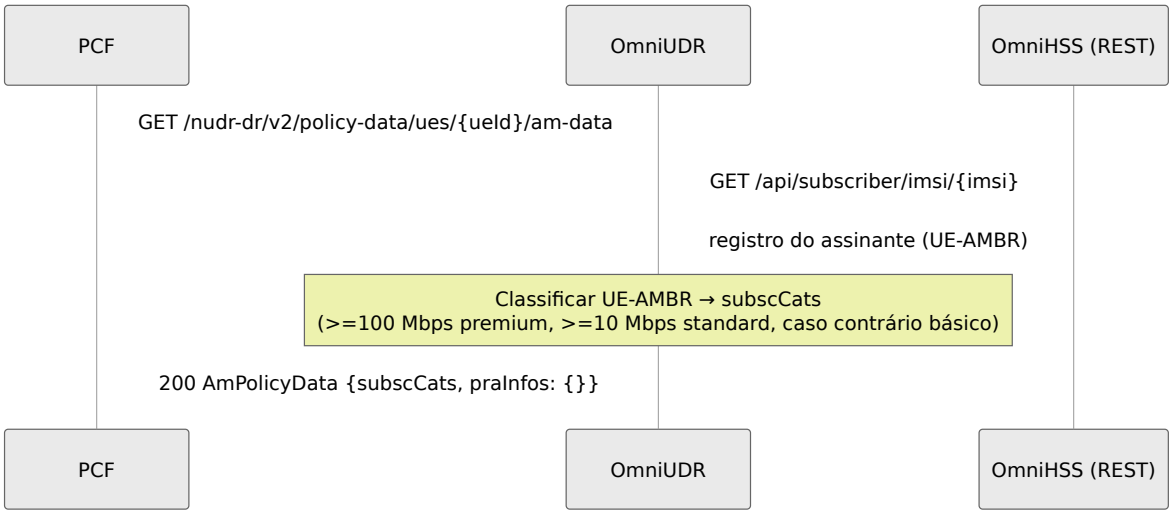
Gravação de Contexto de Registro SMF (cache local apenas)

O contexto de registro SMF é armazenado apenas no cache ETS local, indexado por `{ueId}/{pduSessionId}`; nunca é persistido no HSS e é perdido na reinicialização. A presença de uma entrada de cache existente decide criação vs. atualização.

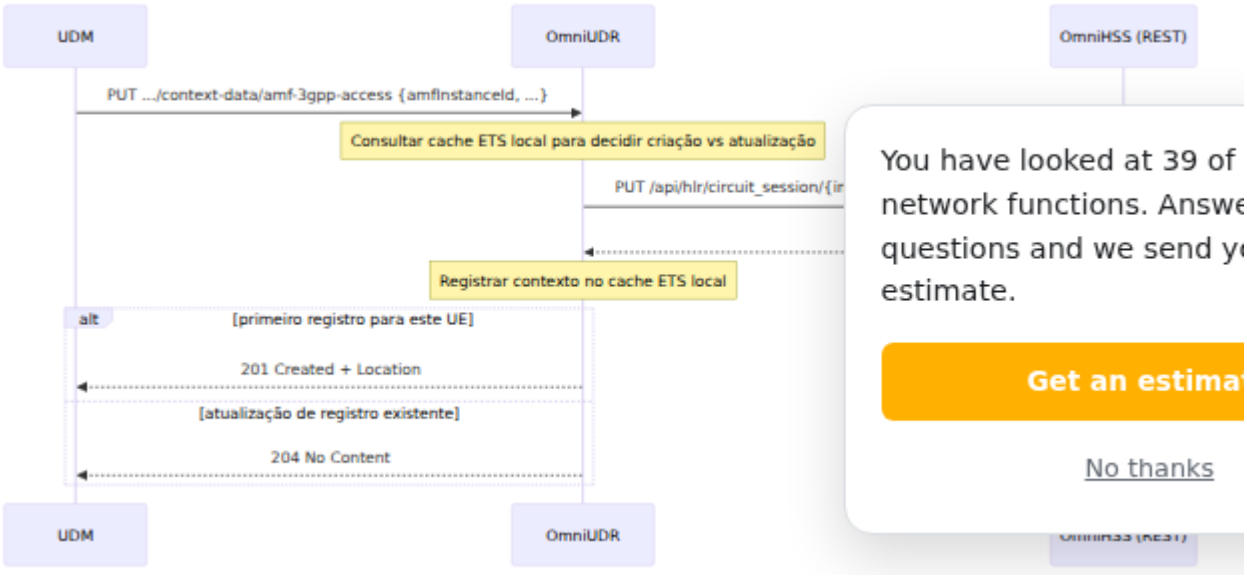


Leitura de Dados de Política AM (PCF → UDR → HSS)

Os dados de política AM são grosseiros: o OmniUDR lê o UE-AMBR do assinante do HSS e o classifica em uma única categoria de assinante (`premium` / `standard` / `basic`); `praInfos` está sempre vazio.

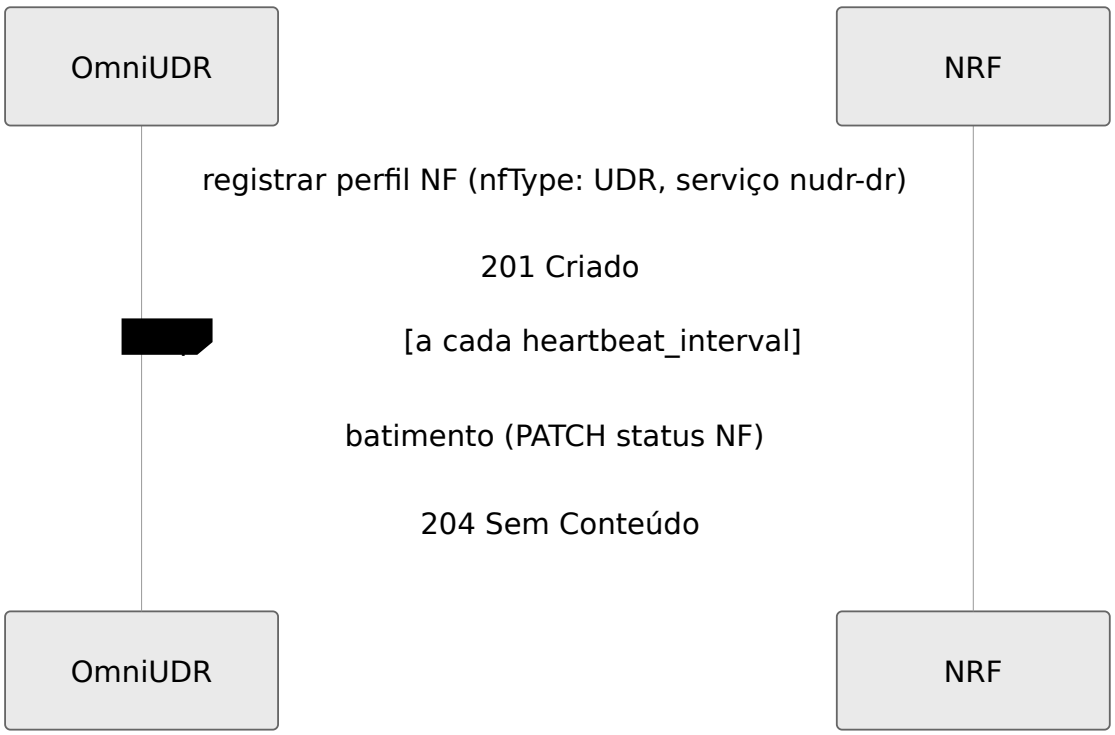


Leitura de Dados de Política SM (PCF → UDR → HSS)



Registro e Batimento do NRF

Na inicialização, o OmniUDR registra seu perfil NF com o NRF e, em seguida, envia batimentos a cada `heartbeat_interval` milissegundos para que o registro continue anunciando-o como ativo. O status de registro é exposto como um gauge (veja Métricas), e uma ação OAM pode forçar o re-registro.



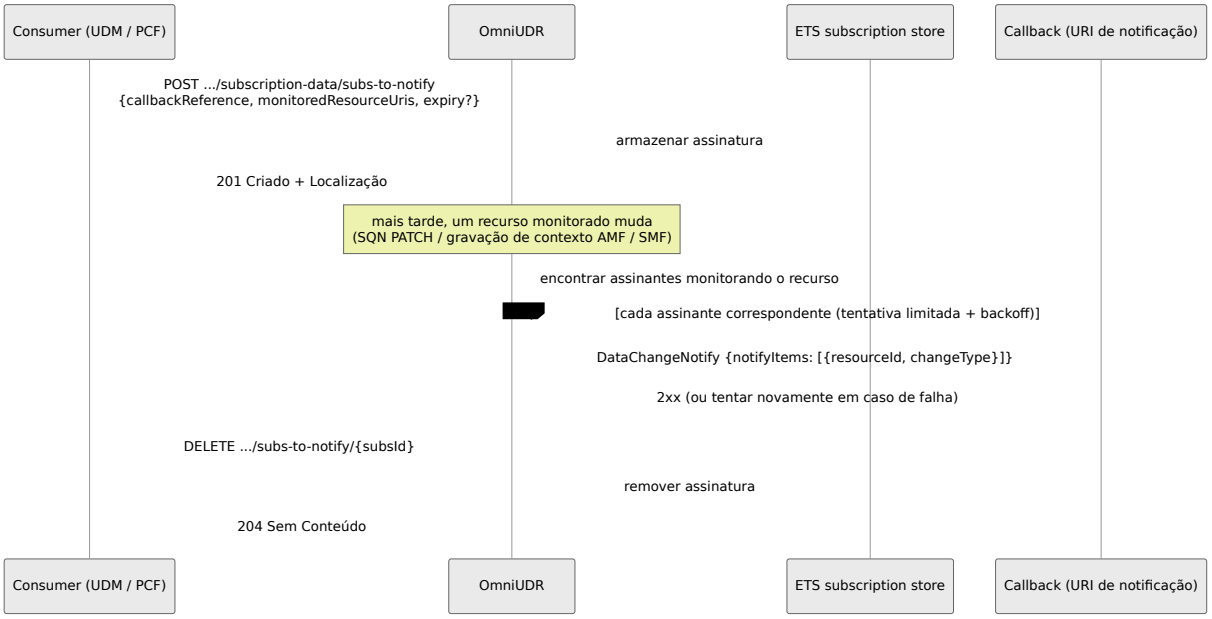
Assinaturas de Mudança de Dados e DataChangeNotify

O OmniUDR implementa o serviço de assinatura de mudança de dados Nudr_DR (o recurso `subscription-data/subs-to-notify`, TS 29.504 subscriptions-to-notify). Um consumidor registra uma assinatura que nomeia os URIs de recurso que deseja monitorar e um URI de callback de notificação, opcionalmente com uma expiração. As assinaturas são mantidas em um armazenamento de assinatura em memória (ETS); uma expiração, quando fornecida, é respeitada pelo armazenamento, que descarta a assinatura uma vez que ela expira.

Quando um recurso monitorado muda, o OmniUDR emite um `DataChangeNotify` para o URI de callback de cada assinante cujos URIs monitorados cobrem o recurso alterado (uma correspondência exata de URI, ou um prefixo de coleção para que uma assinatura em um recurso pai capture seus recursos filhos). Cada notificação carrega um tipo de mudança: `ADD`, `REPLACE` ou `REMOVE`. Nesta versão, as notificações são disparadas em uma atualização de SQN (PATCH da assinatura de autenticação) e em gravações de

contexto AMF e SMF (um primeiro registro para um recurso é um `ADD`, uma gravação subsequente é um `REPLACE`).

A entrega é de melhor esforço e nunca afeta a resposta SBI que o consumidor recebe. Um callback que falha ou está inacessível é refeito com um número limitado de tentativas e um backoff exponencial (veja `notification_max_attempts` e `notification_backoff_ms`); uma vez que as tentativas se esgotam, a notificação é descartada e o evento é registrado.



← Visão Geral

Solução de Problemas do OmniUDR

Todas as leituras do Nudr retornam 404 / buscas de assinantes falham

O OmniUDR atende a cada leitura do backend HSS. Se as leituras falharem:

1. Confirme se `hss_api_base_url` é acessível a partir do host OmniUDR e se o HSS está ativo. Use a verificação embutida: `GET /api/health/hss` na API de gerenciamento e observe o gauge `omni_udr_hss_health` (1 = ativo, 0 = inativo).
2. Confirme se o assinante existe e está **ativado** no HSS. Um assinante desativado (`enabled: false`) é tratado como não encontrado (404 para o consumidor).
3. Confirme se o `{ueId}` usa o formato `imsi-<dígitos>` (um IMSI puro também é aceito).
4. Confirme se `hss_api_key` está definido e se o OmniHSS o aceita. Um HSS `401` significa que a chave está ausente ou não é conhecida pelo OmniHSS. Um `403` significa que a chave não possui o papel `hlr_interworking`. Se `hss_api_key` não estiver definido, o log mostrará um erro que menciona `hss_api_key`.
5. Observe `omni_udr_hss_requests_total{result="failure"}` e `omni_udr_hss_request_duration_ms` para distinguir um HSS inativo/lento de um assinante genuinamente ausente.

O consumidor recebe 5xx / erros quando o HSS está inacessível

Quando o backend HSS está inativo ou expira (5 s por solicitação, 3 s para a verificação de saúde), o cliente REST do HSS retorna um erro e o OmniUDR o apresenta ao consumidor (tipicamente um `404` nos recursos de leitura, ou um `502` na API de gerenciamento no proxy do assinante). Isso é esperado - o OmniUDR não possui um armazenamento local de assinantes para recorrer. Restaure a acessibilidade do HSS; `omni_udr_hss_health` retorna a 1 assim que a próxima verificação for bem-sucedida. Note que `test_subscribers` é uma ferramenta de laboratório e **não** é um fallback automático para HSS inativo nesta versão.

PATCH para atualização de SQN retorna 200 em vez de 204

Um `200` com um corpo `PatchResult` significa que o HSS **rejeitou** a atualização de SQN; o array `report` nomeia o caminho falhado e a razão do HSS. Uma atualização bem-sucedida retorna `204`. Verifique os logs do HSS e o caminho `/api/hlr/update_sqn`. Um corpo PATCH sem `sequenceNumber` é tratado como um no-op bem-sucedido (`204`).

Geração de vetor de autenticação retorna 404 / 500

O POST `authentication-vectors` procura o assinante e, em seguida, delega ao endpoint HSS `/api/hlr/generate_eutran_auth_vectors`. Um `404` significa que o assinante é desconhecido para o HSS ou está desativado; um `500` significa que o HSS retornou um erro ou uma forma de resposta inesperada. Em uma ressincronização, certifique-se de que o UDM forneceu `resynchronizationInfo` com ambos `rand` e `auts` - o OmniUDR os concatena no `rand||auts` hex que o HSS espera. Este endpoint é proprietário e apenas o OmniUDM deve chamá-lo.

Os dados de política parecem escassos

Os dados de política AM retornam apenas `subscCats` grosseiros derivados do nível UE-AMBR do assinante (`premium` ≥ 100 Mbps, `standard` ≥ 10 Mbps, caso contrário `basic`), e `praInfos` está vazio. O nível utiliza o que quer que seja maior entre uplink ou downlink UE-AMBR. Quando o perfil EPC do HSS do assinante não possui UE-AMBR, o OmniUDR serve um padrão de 1000 Kbps, então um assinante não provisionado ou mal mapeado cai em `basic` - se um assinante que você espera que seja `premium`/`standard` mostrar `basic`, verifique se o UE-AMBR EPC (`ue_ambr_ul_kbps` / `ue_ambr_dl_kbps`) está provisionado no HSS. Os dados de política SM são construídos a partir do APN/QoS e dos perfis de regras de cobrança do PCRF do assinante. Não há um armazenamento de política 5G dedicado; isso é intencional na arquitetura atual.

Desvio de dados AM (S-NSSAI) incompatível

O registro HSS 4G não possui assinatura de fatia 5G, então o NSSAI de dados AM é sintetizado pelo OmniUDR e padrão para as fatias de laboratório padrão (eMBB, URLLC, mIoT). Se um S-NSSAI qualificado por SD solicitado por um UE for rejeitado pelo AMF (causa 5GMM #62), alinhe `default_am_nssai` (e, por DNN, `dnn_snssai_map`) com o NSSAI servido pelo AMF para a implantação. Veja a Referência de Configuração.

Os dados SM servem um AMBR fixo para todos os assinantes

Se os dados de assinatura SM mostram o mesmo AMBR de sessão / QoS para todos os assinantes, independentemente do provisionamento, os nomes dos campos do perfil de QoS APN do HSS não estão sendo correspondidos e o OmniUDR está recorrendo aos padrões. O mapeamento espera os campos `ApnQosProfile` do OmniHSS (`apn_ambr_dl_kbps`, `apn_ambr_ul_kbps`,

`allocation_retention_priority`, `pre_emption_capability`, `pre_emption_vulnerability`, `qci`). Verifique se o registro do assinante HSS contém esses campos.

O cache de contexto parece desatualizado após alterações HSS fora de banda

O cache de registro AMF/SMF apenas rastreia gravações de contexto e é usado para a decisão de 201-vs-204; ele não armazena dados de leitura. Se precisar ser limpo (por exemplo, após ciclos de teste), limpe-o via `DELETE /api/oam/subscriber-cache/{imsi}` (ou `.../all`).