← 概述

# OmniUDR 配置参考

OmniUDR 从应用环境键 `:omniudr` 读取其操作员配置，该键在启动时从 `config/runtime.exs` 填充。大多数运行时键都有环境变量覆盖和默认值。`hss_api_key` 既没有环境变量覆盖也没有默认值，并且是必需的。下面的块是完整的操作员表面。

## 运行时□置

```
import Config

config :omniudr,
  # HSS API □端
  hss_api_base_url: System.get_env("HSS_API_BASE_URL",
"https://127.0.0.1:8443"),
  hss_api_key: "<OmniHSS API key with role hlr_interworking>",

  # SBI 配置
  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")),

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

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

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

  # 心跳间隔 (毫秒)
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000")),

  # 测试订阅者 - 实验室启动辅助。IMSI → {ki, opc, amf} 的 JSON 映射。
  test_subscribers: System.get_env("TEST_SUBSCRIBERS")

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

## 参数表

| 参数 | 类型 | 必需 | 默认 | |
|---|---|---|---|---|
| `hss_api_base_url` | 字符串 (URL) | 是 | `https://127.0.0.1:8443` | HSS |
| `hss_api_key` | 字符串 | 是 | 无 | 无 |
| `sbi_scheme` | 字符串 | 否 | `http` | SBI |
| `sbi_addr` | 字符串 (IP) | 否 | `127.0.0.22` | SBI |
| `sbi_port` | 整数 | 否 | `7777` | SBI |

| 参数 | 类型 | 必需 | 默认 | |
|---|---|---|---|---|
| `nrf_uri` | 字符串 (URL) | 否 | `http://127.0.0.1:7777` | `NRF` |
| `mcc` | 字符串 | 否 | `999` | `MCC` |
| `mnc` | 字符串 | 否 | `70` | `MNC` |
| `prometheus_metrics_port` | 整数 | 否 | `9568` | `PRO` |
| `heartbeat_interval` | 整数 (毫秒) | 否 | `10000` | `HEA` |
| `test_subscribers` | 字符串 (JSON) \| nil | 否 | `nil` | `TES` |
| `default_am_nssai` | 映射 | 否 | eMBB / URLLC / mIoT 实验室切片（见下文） | *(□ -* |

| 参数 | 类型 | 必需 | 默认 | |
|---|---|---|---|---|
| | | | | |
| `dnn_snssai_map` | 映射 | 否 | `%{"iot" => %{"sst" => 3, "sd" => "000003"}}` | *(* - |
| `notification_max_attempts` | 整数 | 否 | `4` | *(* - |
| `notification_backoff_ms` | 整数（毫秒） | 否 | `500` | *(* - |

> `default_am_nssai` 和 `dnn_snssai_map` 的存在是为了将 4G HSS 数据模型（没有切片□念）桥接到 5G Nudr 结构。将它们与部署的真实切片计划和 AMF/SMF 提供的 NSSAI 对齐。AM/SM 默认值（eMBB `sst 1`, URLLC `sst 2 / sd 000002`, mIoT `sst 3 / sd 000003`）是实验室值。

## 切片合成映射形状（`default_am_nssai`，`dnn_snssai_map`）

这两个键是 S-NSSAI 对象的结构化映射。一个 **S-NSSAI** 有两个子字段：

| 子字段 | 类型 | 必需 | 意义 |
| --- | --- | --- | --- |
| `sst` | 整数 (0–255) | 是 | 切片/服务类型。标准化值：`1` = eMBB，`2` = URLLC，`3` = mIoT（根据 TS 23.501）。 |
| `sd` | 字符串 (6 个十六进制数字) | 否 | 切片区分符。区分具有相同 `sst` 的两个切片。省略它以获得裸 `sst` 切片；如果存在，它必须与 AMF/SMF 提供的值完全匹配。 |

`default_am_nssai` 是接入和移动 NSSAI，包含两个列表：

| 列表 | 意义 |
| --- | --- |
| `defaultSingleNssais` | UE 在注册时未请求特定切片时获得的 S-NSSAI - 网络为订阅者提供的默认切片集。 |
| `singleNssais` | 订阅者被允许使用的完整 S-NSSAI 集合（`defaultSingleNssais` 的超集）。请求未在此列出的 SD 合格 S-NSSAI 的 UE 将被 AMF 拒绝，原因是 5GMM 原因 #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` 将一个 **DNN** 字符串映射到其 PDU 会话所在的单个 S-NSSAI。任何不在映射中的 DNN 将回退到 eMBB（`%{"sst" => 1, "sd" => "000001"}`）。确保此处引用的每个 S-NSSAI 都在 `default_am_nssai.singleNssais` 内，否则 SMF 将把会话放置在订阅者未订阅的切片上。

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

## 关于 `hss_api_base_url`（HSS REST 后端）

这是最重要的操作员设置。OmniUDR 是一个外观 - 它没有数据库。当消费者请求 OmniUDR 获取订阅者的认证订阅、AM 数据、SM 数据或策略数据时，OmniUDR 会对 `hss_api_base_url` 发出实时 REST 请求（主要是 `GET /api/subscriber/imsi/{imsi}` 和 `GET /api/key_set/{id}`，以及对于认证向量 / SQN / 位置的 `/api/hlr/*` 端点），然后将 HSS 响应映射到 3GPP 结构。每个 HSS 请求都有 5 秒的超时（健康检查为 3 秒）。如果该后端无法访问，相应的 Nudr 读取将返回 `404`/`5xx` - 请参见 故障排除。

## 关于 `hss_api_key`（OmniHSS 访问控制）

OmniHSS 在其 REST API 上使用基于角色的访问控制。每个客户端必须在 `x-omnihss-key` 头中发送其 API 密钥。没有有效密钥的请求将被 OmniHSS 拒绝，返回 `401`。OmniUDR 在每个请求中都会将 `hss_api_key` 发送到 `hss_api_base_url`。

给 OmniUDR 密钥赋予 OmniHSS 角色 `hlr_interworking`。此角色允许访问这些 OmniHSS 端点：

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

该值是一个秘密。OmniUDR 不会将其写入日志。在启用 OmniHSS 的访问控制之前安装此版本，否则所有 OmniUDR 对 OmniHSS 的请求将以 `401` 失败。

## 关于 `test_subscribers`

`test_subscribers` 填充一个静态的、代码内的订阅者集（两个内置 IMSI - 众所周知的 Milenage 测试向量 - 以及通过 JSON 映射提供的任何内容），返回固定的认证、AM 和 SM ���据，而无需任何 HSS 调用。这是一个 实验室辅助工具，用于启

动和一致性自测，在不需要完整 HSS 的情况下。在当前构建中，实时 Nudr_DR 数据路径始终从 HSS 后端读取；测试订阅者集没有在 HSS 停机时作为自动回退接入。将此键视为仅用于实验室/测试的表面，并在任何由真实 HSS 支持的部署中保持未设置（`nil`）。无效的 JSON 值将回退到两个内置默认值。

## PLMN 身份和 NRF 置文件

`mcc` 和 `mnc` 在 NRF NF 配置文件中作为服务 PLMN 进行宣传。向 NRF 宣传的 NF 容量和优先级通过共享的 Omni5G NRF 配置进行管理，并可以通过 OAM API 在运行时进行更改（`nf_capacity`，`nf_priority`）。

### 通知投递

`notification_max_attempts` 和 `notification_backoff_ms` 控制 OmniUDR 如何向数据变化订阅者（`subs-to-notify` 服务，TS 29.504 subscriptions-to-notify）投递 Nudr_DR DataChangeNotify 通知。当监控的资源发生变化时，OmniUDR 会向每个匹配的订阅者的回调 URI 发布通知。如果投递失败（回调不可达或返回非 2xx 状态），OmniUDR 会重试最多 `notification_max_attempts` 次，第一次重试等待 `notification_backoff_ms`，并在每次后续重试时将延迟加倍。一旦尝试耗尽，通知将被丢弃，事件将被记录。投递是尽力而为的，绝不会影响返回给导致变化的消费者的 SBI 响应。这两个键仅用于应用程序配置（没有环境变量覆盖）：在 `config :omniudr, ...` 下进行配置。

---

# 管理和 OAM API

除了 SBI 监听器，OmniUDR 还在 **8443** 端口（TLS，`enable_tls: true`）上公开了一个面向操作员的 HTTPS 管理 API。控制面板 Web UI 也通过 HTTPS 在 **7443** 端口上提供。

| 表面 | 端口 | 目的 |
| --- | --- | --- |
| SBI (Nudr_DR) | `sbi_port` (默认 `7777`) | 3GPP 服务接口 |
| Prometheus 指标 | `prometheus_metrics_port` (默认 `9568`) | 指标抓取端点 |
| 管理 / OAM API | `8443` (HTTPS，挂载在 `/api` 下) | 状态、HSS 健康、订阅者代理、OAM 操作 |
| 控制面板 UI | `7443` (HTTPS) | 应用资源、配置、许可证和日志页面 |

管理 API 挂载在 `/api` 路径前缀下。它提供只读状态、HSS 连接检查、只读订阅者代理到 HSS 和 OAM 操作：

| 方法 | 路径 | 目的 |
| --- | --- | --- |
| GET | `/api/status/api` | API/服务状态 |
| GET | `/api/status/nrf` | NRF 注册状态 |
| GET | `/api/status/nf` | NF 配置文件 / 状态 |
| GET | `/api/status/license` | 许可证状态 |
| GET | `/api/health/hss` | HSS 连接检查（调用 HSS 的 `GET /api/status/api`） |
| GET | `/api/subscribers` | 从 HSS 代理的分页只读订阅者列表（`page`，`page_size`） |
| GET | `/api/subscribers/{imsi}` | 从 HSS 代理的完整订阅者详细信息 |
| GET | `/api/oam/config` | 查看实时运行时配置 |
| PATCH | `/api/oam/config/{id}` | 在运行时更新 `hss_api_base_url`、`nrf_uri`、`nf_capacity`、`nf_priority` 或 `log_level` |
| POST | `/api/oam/nrf/reregister` | 强制 NRF 重新注册 |
| DELETE | `/api/oam/subscriber-cache/{id}` | 刷新订阅者的 AMF/SMF 上下文缓存（`id=all` 刷新每个订阅者） |
| POST | `/api/oam/log_level` | 更改运行时日志级别 |

关于 OAM 配置更新的说明（`PATCH /api/oam/config/{id}`）：

- 仅接受 `hss_api_base_url`、`nrf_uri`、`nf_capacity`、`nf_priority` 和 `log_level`；任何其他键将被忽略，且没有有效键的请求将返回 400。

- `nf_capacity` 和 `nf_priority` 必须是整数；`log_level` 必须是 `emergency`、`alert`、`critical`、`error`、`warning`、`warn`、`notice`、`info`、`debug` 之一。
- 更改 `nf_capacity` / `nf_priority` 将同时持久化该值并向 NRF 推送更新的配置文件。
- 订阅者代理是只读的。如果 HSS 无法访问，`/api/subscribers` 和 `/api/subscribers/{imsi}` 将返回 `502`；禁用或缺失的订阅者将返回 `404`。

---

## 日志记录

OmniUDR 通过 `OmniLogger.JsonFormatter` 以结构化 JSON 日志的形式发出完整的元数据。可以通过 OAM API（`POST /api/oam/log_level` 或 `PATCH /api/oam/config/{id}` 与 `log_level`）在不重启的情况下更改运行时日志级别。OAM 操作（配置更新、缓存刷新、重新注册）以 `info` 级别记录，并带有 `[OAM]` 前缀；HSS 请求失败以 `warning`/`error` 级别记录。

← 概述

# OmniUDR 指标

OmniUDR 在指标端点（`prometheus_metrics_port`，默认 `9568`）上公开 Prometheus 指标。下面的指标名称使用 Prometheus 展示形式（点变为下划线）。标签及其值均为小写。

## UDR / HSS 指标

| 指标 | 类型 | 标签 | 描 |
| --- | --- | --- | --- |
| `omni_udr_nrf_registration_status` | gauge | `nf_type` | NRF 注册状态[ 0 = 未注册） |
| `omni_udr_hss_health` | gauge | - | OmniHSS [ = 正常，0 = [ `/health/h` |
| `omni_udr_hss_requests_total` | counter | `endpoint`，`result` | 对 HSS 后端的 数，按请求路径 （`endpoint` `result` （`success`/ 标记 |
| `omni_udr_hss_request_duration_ms` | distribution | `endpoint` | HSS REST 为单位）。桶：5, 50, 100, 2 1000, 250 |

# BEAM VM 指标

| 指标 | 类型 | 描述 |
| --- | --- | --- |
| `beam_memory_total` | gauge | 总 BEAM 内存（以字节为单位） |
| `beam_memory_processes` | gauge | Erlang 进程使用的内存 |
| `beam_memory_system` | gauge | 系��内存（ETS、原子、代码、二进制） |
| `beam_processes_count` | gauge | Erlang 进程数量 |
| `beam_vm_uptime` | gauge | VM 运行时间（以秒为单位） |

# 有用的 PromQL

- HSS 后端可用性：`omni_udr_hss_health`
- 已与 NRF 注册的 UDR：`omni_udr_nrf_registration_status`
- HSS 错误率（5 分钟）：`sum(rate(omni_udr_hss_requests_total{result="failure"}[5m])) / sum(rate(omni_udr_hss_requests_total[5m]))`
- HSS p95 延迟（5 分钟）：`histogram_quantile(0.95, sum(rate(omni_udr_hss_request_duration_ms_bucket[5m])) by (le, endpoint))`
- 按端点的 HSS 请求速率：`sum(rate(omni_udr_hss_requests_total[1m])) by (endpoint)`
- BEAM 进程数量：`beam_processes_count`

← 概述

# OmniUDR 操作指南

OmniUDR 是 Omnitouch 5G 核心的统一数据存储库 (UDR)。它实现了 Nudr_DataRepository 服务 (TS 29.504)，并通过基于服务的接口 (SBI) 向 UDM 和 PCF 暴露结构化的订阅者、策略和上下文数据。

OmniUDR 不拥有订阅者数据。它是一个符合标准的 Nudr_DR 轻量级外观，位于 **OmniHSS REST** 前置后端 之上：每次数据存储库读取在请求时都会转换为一个或多个 OmniHSS REST 调用，HSS 响应在返回给消费者之前被映射到 3GPP Nudr 数据模型。OmniUDR 是唯一一个直接与 OmniHSS REST API 通信的 5G 核心网络功能 - UDM（通过它，AUSF）通过标准 Nudr 调用 OmniUDR 来访问订阅者数据，而不是直接调用 HSS。

由于 HSS 是记录系统，OmniUDR 几乎不持有持久状态。它唯一的本地状态是三个小的内存中 (ETS) 存储：AMF 和 SMF 注册上下文的缓存（用于决定上下文写入是创建还是更新）、持有活动 Nudr_DR `subs-to-notify` 订阅的数据变更订阅存储，以及持有 Nudr_DR `application-data` 家族（PFD、流量影响、BDT）的应用数据存储。这些都没有持久化；所有在重启时都会被清除。注册上下文缓存可以通过管理 API 刷新。

---

## 3GPP 角色和规范参考

| 规范 | 相关性 |
|---|---|
| TS 23.501 | 5G 系统架构 - UDR 角色及其与 UDM、PCF 和 UDM 数据层的关系 |
| TS 29.504 | Nudr_DataRepository 服务 - 资源框架、订阅数据、上下文数据、策略数据 |
| TS 29.505 | 订阅数据结构 - 认证订阅、接入与移动性、会话管理、SMF 选择 |
| TS 29.519 | 策略数据结构 - AM 策略数据和 SM 策略数据资源形状 |
| TS 29.503 | Nudm 服务 - UDM 消费者，其 UEAU/SDM/UECM 程序驱动 UDR 读取和上下文写入 |
| TS 29.571 | 通用数据类型 - `PatchResult`、`ProblemDetails`、PLMN 和 S-NSSAI 编码 |

---

## 接口

| 接口 | 绑定 | 目的 |
| --- | --- | --- |
| SBI (Nudr_DR) | `{sbi_scheme}://{sbi_addr}:{sbi_port}` (默认 `http://127.0.0.22:7777`) | 提供 UDM 和 PCF 消耗的 `nudr-dr/v2` 资源 |
| HSS REST 后端 | `hss_api_base_url` (默认 `https://127.0.0.1:8443`) | 向 OmniHSS 发出的出站 REST 调用，满足每次读取/更新。每次调用在 `x-omnihss-key` 头中发送 `hss_api_key` (OmniHSS 角色 `hlr_interworking`) |
| NRF | `nrf_uri` | NF 注册和定期心跳 |
| 管理 / OAM API | `:8443` (HTTPS, 挂载在 `/api` 下) | 状态、HSS 健康、订阅者代理和 OAM 操作 - 参见 配置参考 |
| Prometheus 指标 | `prometheus_metrics_port` (默认 `9568`) | 指标抓取端点 - 参见 指标 |

---

## 提供的数据集

OmniUDR 提供以下 Nudr_DR 数据集。每个数据集在读取时从 OmniHSS 订阅者记录中派生；“HSS 源”列命名了值映射的底层 REST 数据。

| 数据集 | **Nudr** 资源家族 | 访问 | 消费者 | |
|---|---|---|---|---|
| 认证订阅 | `subscription-data/.../authentication-data/authentication-subscription` | 读取 / 更新 | UDM (Nudm_UEAU) | 订阅者 `key_set` `key_set_id[` |
| 认证向量 | `subscription-data/.../authentication-data/authentication-vectors` | 生成（专有） | OmniUDM | HSS `generate_eu` （向量 + SQN/[ |
| 接入与移动性 (AM) 数据 | `subscription-data/.../provisioned-data/am-data` | 读取 | UDM (Nudm_SDM) | 订阅者 EPC 配置 U NSSAI |
| 会话管理 (SM) 数据 | `subscription-data/.../provisioned-data/sm-data` | 读取 | UDM (Nudm_SDM) | EPC APN 配置 · AMBR、默认 Qo |
| SMF 选择数据 | `subscription-data/.../provisioned-data/smf-selection-subscription-data` | 读取 | UDM (Nudm_SDM) | EPC APN 列表 · DNN 信息 |
| AMF 注册上下文 | `subscription-data/.../context-data/amf-3gpp-access` | 写入 | UDM (Nudm_UECM) | 映射到 HSS 电路会 |
| SMF 注册上下文 | `subscription-data/.../context-data/smf-registrations/{pduSessionId}` | 写入 | UDM (Nudm_UECM) | 仅存储在本地缓存中（[ |
| AM 策略数据 | `policy-data/ues/{ueId}/am-data` | 读取 | PCF | 从 EPC UE-AM |

| 数据集 | **Nudr** 资源家族 | 访问 | 消费者 | |
|---|---|---|---|---|
| SM 策略数据 | `policy-data/ues/{ueId}/sm-data` | 读取 | PCF | 从 EPC APN/Q 生 |

| 应用数据 | `application-data/{family}/{id}` (家族 `pfds`、`influenceData`、`bdtData`) | 创建 / 读取 / 替换 / 删除 | PCF, NEF | 无 - 存储在本地内存中 (ETS) 存储中，不支持 HSS 并且不持久化 |

> 应用数据由通用内存存储提供，而不是 HSS。它不持久化：重启会清除它。请参见 SBI 端点 → 应用数据 下文。

---

# SBI 端点

所有 Nudr_DR 端点通过 HTTP 提供，地址为 `{sbi_scheme}://{sbi_addr}:{sbi_port}` (默认 `http://127.0.0.22:7777`)，并使用 `Content-Type: application/json`。失败将作为 TS 29.571 `ProblemDetails` 返回。

`{ueId}` 路径段是 `imsi-<digits>` 形式的 SUPI（也接受裸 IMSI）。在配置数据读取中，`{servingPlmnId}` 段经过验证：它必须是格式良好的 PLMN（5 或 6 位数字），并且必须与从 `mcc`/`mnc` 配置的服务 PLMN 匹配，否则读取将被拒绝，返回 `400 MANDATORY_IE_INCORRECT`。此构建为每个订阅者提供单个记录。

## 订阅数据 - 认证 (TS 29.504 / 29.505)

| 方法 | 路径 | 描述 | 成功 |
| --- | --- | --- | --- |
| GET | `/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-subscription` | 检索认证订阅（方法、AMF、SQN） | 200 |
| PATCH | `/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-subscription` | 更新认证订阅（SQN） | 204，或 200 + `PatchResult` 在部分失败时 |
| POST | `/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-vectors` | 生成认证向量（见注释） | 200 |

> 非标准端点。`authentication-vectors` POST 是 Omni 5GC 的专有扩展；TS 29.504/29.505 未定义此类资源。向量生成和 SQN/重新同步管理委托给 UDR，因此它们在 HSS 中集中处理。它仅被 OmniUDM 消耗；符合标准的第三方消费者永远不会调用它。

## 订阅数据 - □置 (TS 29.505)

| 方法 | 路径 | 描述 | 成功 |
| --- | --- | --- | --- |
| GET | `/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/am-data` | 检索接入与移动性订阅数据 | 200, 或 400 `MANDATORY_IE_` 在格式错误或不匹配的 `{servingPlmnI` |
| GET | `/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/sm-data` | 检索会话管理订阅数据 | 200, 或 400 `MANDATORY_IE_` 在格式错误或不匹配的 `{servingPlmnI` |
| GET | `/nudr-dr/v2/subscription-data/{ueId}/{servingPlmnId}/provisioned-data/smf-selection-subscription-data` | 检索 SMF 选择订阅数据 | 200, 或 400 `MANDATORY_IE_` 在格式错误或不匹配的 `{servingPlmnI` |

## 订阅数据 - 上下文 (TS 29.504)

| 方法 | 路径 | 描述 | 成功 |
| --- | --- | --- | --- |
| PUT | `/nudr-dr/v2/subscription-data/{ueId}/context-data/amf-3gpp-access` | 存储 AMF 3GPP 接入注册上下文 | 201 + `Location` (新) 或 204 (更新) |
| PUT | `/nudr-dr/v2/subscription-data/{ueId}/context-data/smf-registrations/{pduSessionId}` | 存储 SMF 注册上下文 | 201 + `Location` (新) 或 204 (更新) |

任一上下文 PUT 的空主体返回 400 `ProblemDetails` (`cause: MANDATORY_IE_MISSING`)。

## 策略数据 (TS 29.519)

| 方法 | 路径 | 描述 | 成功 |
|---|---|---|---|
| GET | `/nudr-dr/v2/policy-data/ues/{ueId}/am-data` | 检索 AM 策略数据 (`subscCats`, `praInfos`) | 200 |
| GET | `/nudr-dr/v2/policy-data/ues/{ueId}/sm-data` | 检索 SM 策略数据 (`smPolicySnssaiData` 每 DNN) | 200 |

## 应用数据 (TS 29.504)

对 Nudr_DR `application-data` 家族进行通用 CRUD。`{family}` 路径段是 `pfds`、`influenceData` 或 `bdtData` 之一；`{id}` 段是该家族的资源 id (`appId` / `influenceId` / `bdtReferenceId`)。记录存储在本地内存 (ETS) 存储中，而不是 HSS，并且在重启时不持久化。任何其他 `{family}` 值将返回 404 捕获。

| 方法 | 路径 | 描述 | 成功 |
|---|---|---|---|
| GET | `/nudr-dr/v2/application-data/{family}` | 列出家族中的每个资源（集合 GET） | 200 (数组) |
| GET | `/nudr-dr/v2/application-data/{family}/{id}` | 检索一个资源 | 200，或 404 如果未知 |
| PUT | `/nudr-dr/v2/application-data/{family}/{id}` | 创建或替换资源 | 201 + `Location` (新) 或 200 (替换) |
| DELETE | `/nudr-dr/v2/application-data/{family}/{id}` | 删除资源 | 204，或 404 如果未知 |

PUT 的空主体返回 400 `ProblemDetails` (`cause: MANDATORY_IE_MISSING`)。

## 数据变更订阅 (TS 29.504 subscriptions-to-notify)

| 方法 | 路径 | 描述 | 成功 |
| --- | --- | --- | --- |
| POST | `/nudr-dr/v2/subscription-data/subs-to-notify` | 创建数据变更订阅（主体命名监控的资源 URI 和通知回调 URI） | 201 + `Location`，或 400 在主体错误时 |
| DELETE | `/nudr-dr/v2/subscription-data/subs-to-notify/{subsId}` | 移除数据变更订阅 | 204，或 404 如果订阅未知 |

请参见 数据变更订阅和 DataChangeNotify 以了解扇出行为。

任何不匹配上述资源的路径都将返回 404 `ProblemDetails`。

---

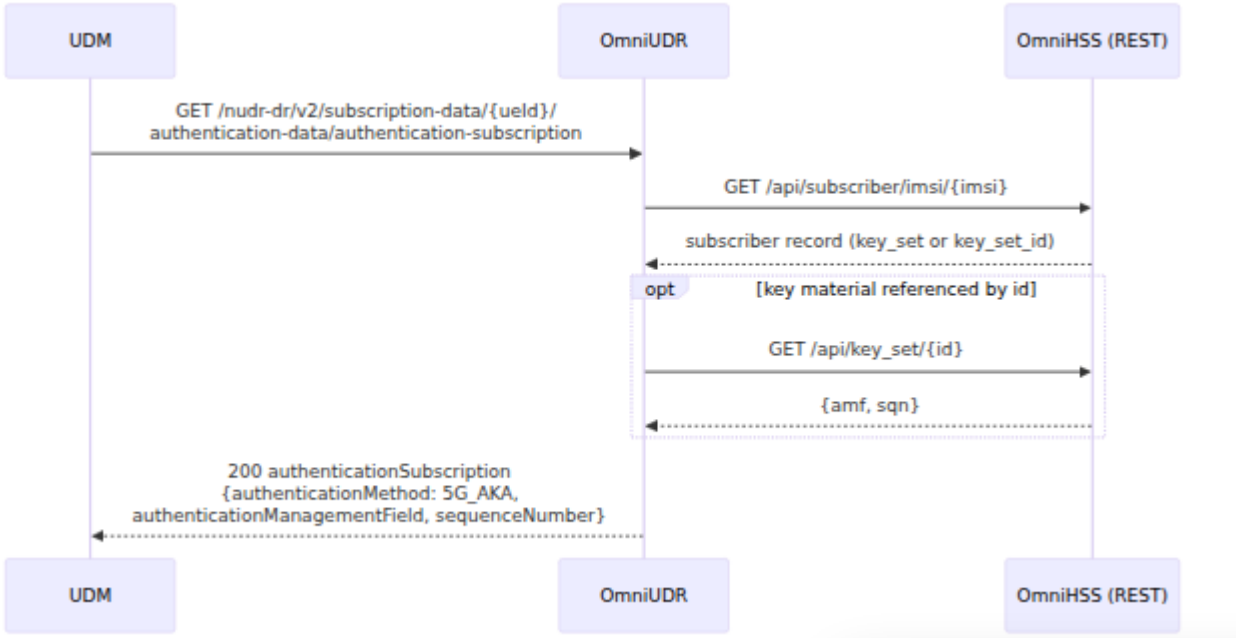
# HSS 后端 REST 调用

每个 SBI 操作由一个或多个调用 OmniHSS REST 后端的 `hss_api_base_url` 来满足。关系如下：

| **Nudr_DR** 操作 | **HSS REST** 调用 | 备注 |
| --- | --- | --- |
| 认证订阅读取 | `GET /api/subscriber/imsi/{imsi}`，然后如果密钥材料通过 id 引用则 `GET /api/key_set/{id}` | 订阅者记录中的密钥材料内联跳过第 |
| 认证向量生成 | `GET /api/subscriber/imsi/{imsi}`，然后 `POST /api/hlr/generate_eutran_auth_vectors` | 主体 `{imsi, num_vect resynchronization_ resynchronization_ rand\|\|auts` hex 在重新 |
| SQN 更新 (PATCH) | `POST /api/hlr/update_sqn` | 主体 `{imsi, sqn}` |
| AM / SM / SMF 选择读取 | `GET /api/subscriber/imsi/{imsi}` | 单个订阅者获取，本地映射 |
| AMF 上下文写入 | `PUT /api/hlr/circuit_session/{imsi}` | 位置/电路会话更新；也在本地缓存 |
| SMF 上下文写入 | (无) | 仅本地 ETS 缓存 |
| 策略数据读取 (AM / SM) | `GET /api/subscriber/imsi/{imsi}` | 重用 AM/SM 映射并重新塑造 |
| 健康检查 | `GET /api/status/api` | 驱动 `/health/hss` 管理 |

> `/api/hlr/*` 端点（`generate_eutran_auth_vectors`、`update_sqn`、`circuit_session/{imsi}`）是 HSS 为 OmniUDR 的认证向量、SQN 和位置操作而专门构建的 REST 接口，连同订阅者/密钥集端点（`/api/subscriber/*`、`/api/key_set/*`）和健康探测（`/api/status/api`）。禁用的订阅者（`enabled: false`）被视为未找到。

---



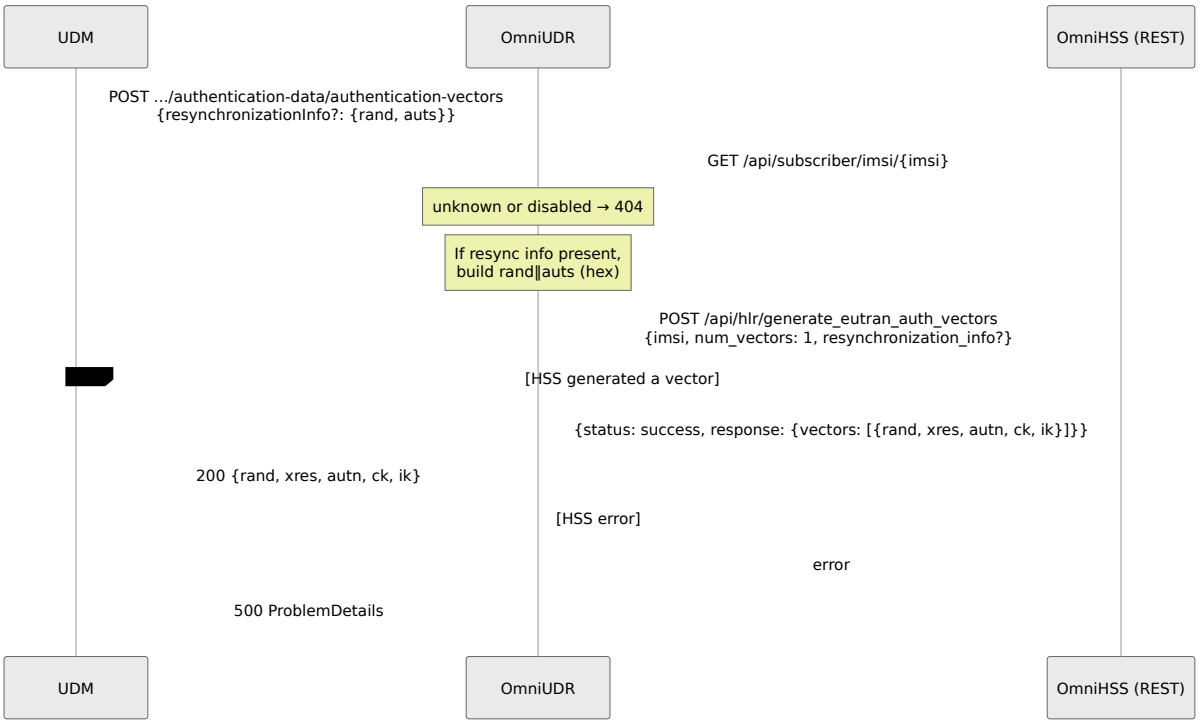
# 关键程序

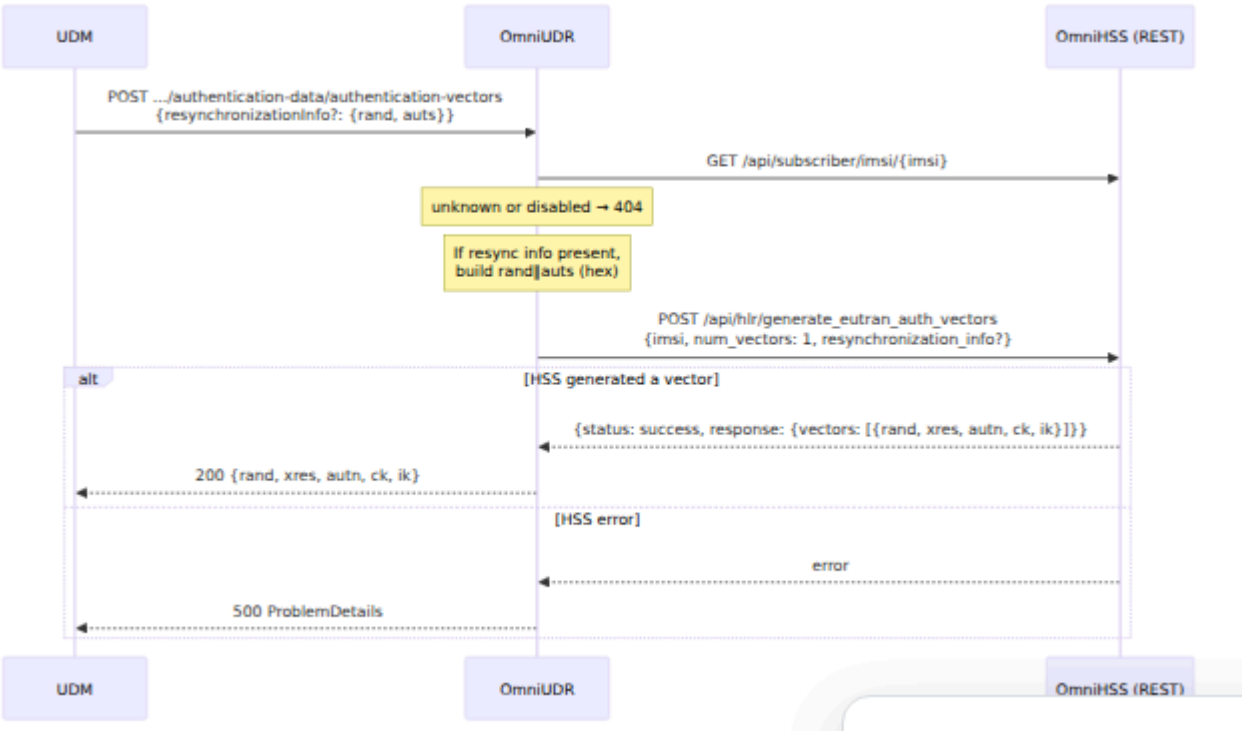
## 认证订阅读取 **(UDM → UDR → HSS)**

核心 UDR 流程: UDM（在 5G-AKA 期间服务 AUSF）请求 OmniUDR 获取订阅者的认证订阅; OmniUDR 从 HSS 读取订阅者和密钥材料，并将其映射到 Nudr 结构。

### 认证向量生成（专有，带重新同步）

OmniUDM 调用非标准的 `authentication-vectors` POST，以便向量生成和 SQN/重新同步在 HSS 中集中处理，Ki/OPc 从未离开 HSS。HSS 返回一个 EUTRAN 向量（`rand`、`xres`、`autn`、`ck`、`ik`），其 AUTN 携带订阅者的配置 AMF; OmniUDM 从中派生 5G 密钥。在重新同步时，UDM 传递 `resynchronizationInfo`（`rand` + `auts`），OmniUDR 将其连接成 HSS 所期望的 hex `rand||auts`。





## □置数据读取 **(AM / SM / SMF** 选择**)**

对于 AM 数据，映射改为从 EPC UE-AMBR 派生 `subscribedUeAmbr` 并附加合成 NSSAI（见 `default_am_nssai`）；对于 SMF 选择数据，它将订阅者的 APN 列表作为 DNN 信息列出，使用单个订阅的 S-NSSAI。

### **HSS APN** □置如何成为 **5G SM** 订阅数据

4G HSS 记录没有 5G 会话管理结构，因此 OmniUDR 为每个配置 APN (`epc_profile.apn_profiles[]`) 合成一个 `dnnConfigurations` 条目。每个字段的派生如下。当命名的 HSS 字段缺失时，将提供后备 - 如果所有后备都缺失，则 APN QoS 配置字段名称不匹配（见 故障排除 → SM 数据提供固定 AMBR）。

| **5G SM** 字段 | **（HSS 字段）** 派生 |
| --- | --- |
| `singleNssai` | DNN → S-NSSAI 通过 `dnn_snssai_map` |
| DNN 键在 `dnnConfigurations` 中 | APN 名称 (`apn_identifier.apn`) |
| `pduSessionTypes` | APN `apn_identifier.ip_version` (见下表) |
| `sscModes` | 固定: 默认 `SSC_MODE_1`，允许 `SSC_MODE_1` + `SS` |
| `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[]` ([ 5QI/QCI、ARP、GBR/MBR、流信息) |
| `staticIpAddress` | 订阅者 `static_ips[]` 其 `apn_profile_id` 是此 `ipv4_static_ip` → `ipv4Addr`，`ipv6_stat` `ipv6Addr` |

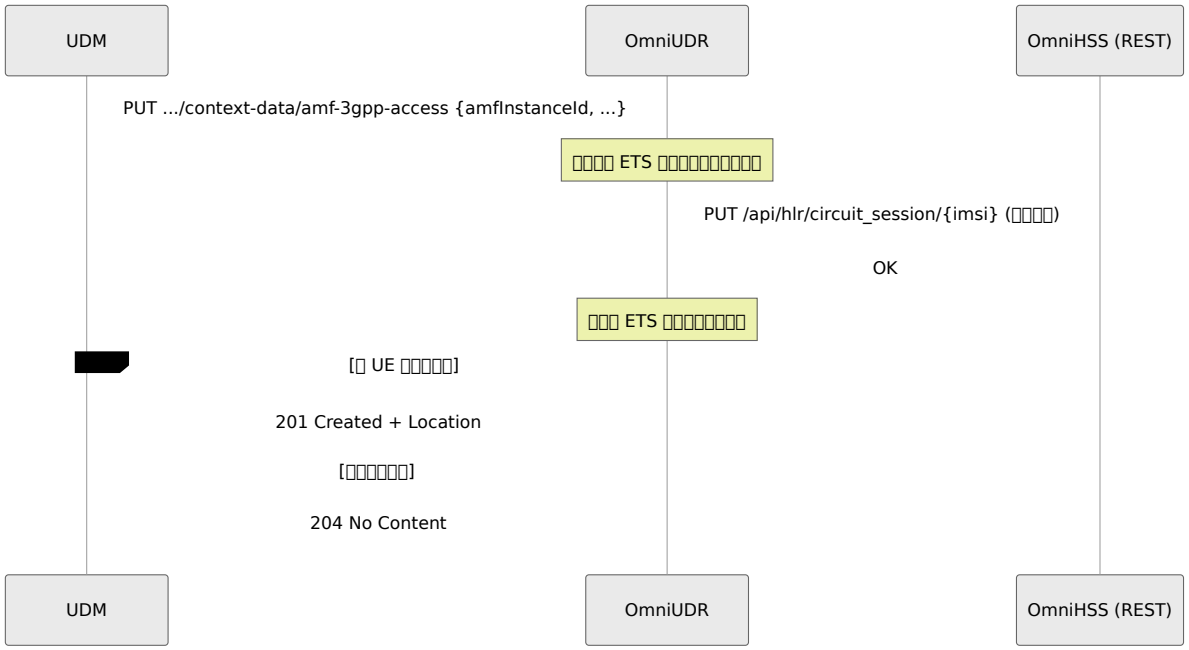
**PDU** 会话类型。 5G PDU 会话类型`` 4G PDN 类型的 5G 对应物，从 APN 的 `ip_version` 派生，因此 IPv6 或双栈 APN 产生一个 IPv6 能力会话，而不是 IPv4 唯一的默认值。`defaultSessionType` 是 SMF 在 UE 不请求特定允许类型时的后备。

| APN `ip_version` | `defaultSessionType` | `allowedSessionTypes` |
|---|---|---|
| `ipv4` | `IPV4` | `IPV4` |
| `ipv6` | `IPV6` | `IPV6` |
| `ipv4v6` / `ipv4_or_ipv6` | `IPV4V6` | `IPV4V6`、`IPV4`、`IPV6` |
| 未设置 / 其他 | `IPV4` | `IPV4`、`IPV4V6` |

**SSC** 模式 不可按订阅者配置：每个 DNN 都以默认 `SSC_MODE_1` 和允许 `SSC_MODE_1`/`SSC_MODE_2` 广告。根据 TS 29.503 / 29.571，`SscModes.allowedSscModes` 的 `maxItems: 2`（默认加最多一个替代），因此集合不能扩大。

**SMF** 选择订阅数据 将每个配置 APN 列出为单个订阅的 S-NSSAI 键 `01` 下的 DNN，无论 `dnn_snssai_map` 中的每个 DNN 映射如何。在多切片部署中，DNN 位于不同的 S-NSSAIs 上，使用 AM/SM 订阅数据（携带正确的每个 DNN S-NSSAI）进行切片决策；SMF 选择仅广告 DNN 列表。



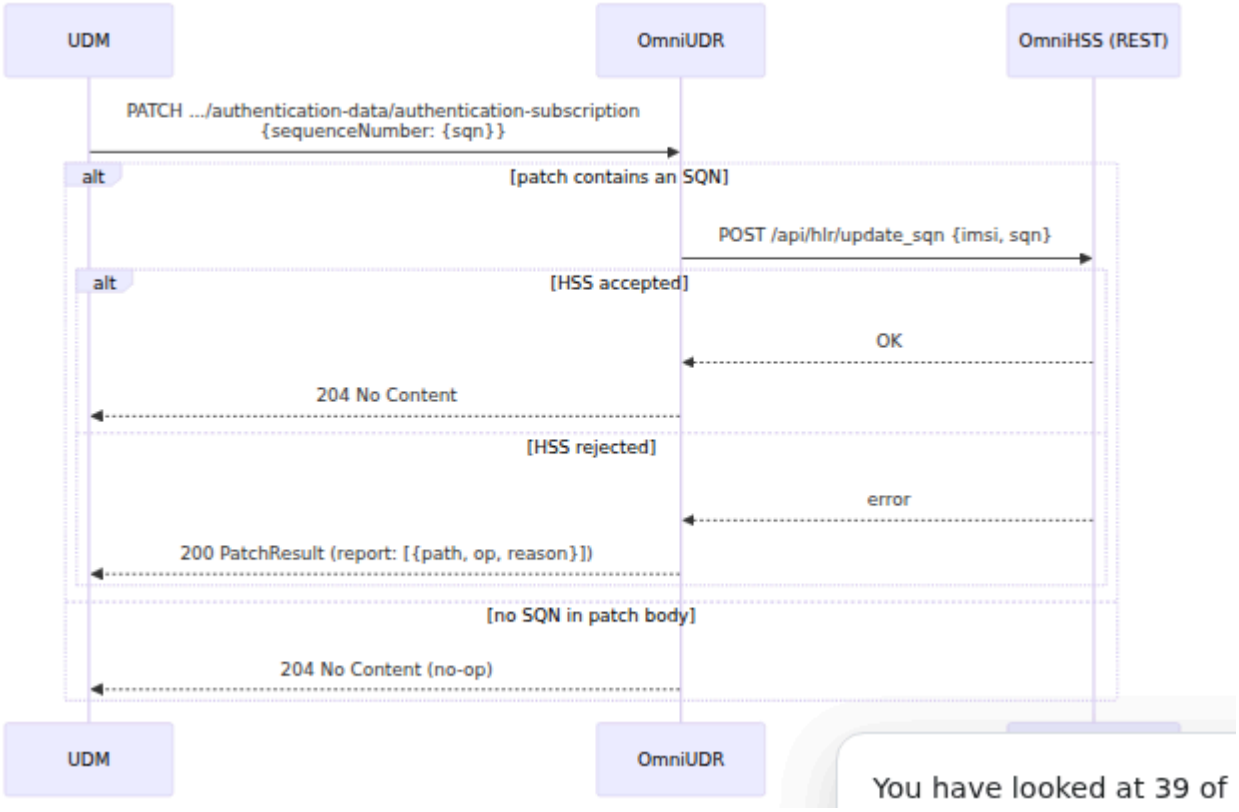


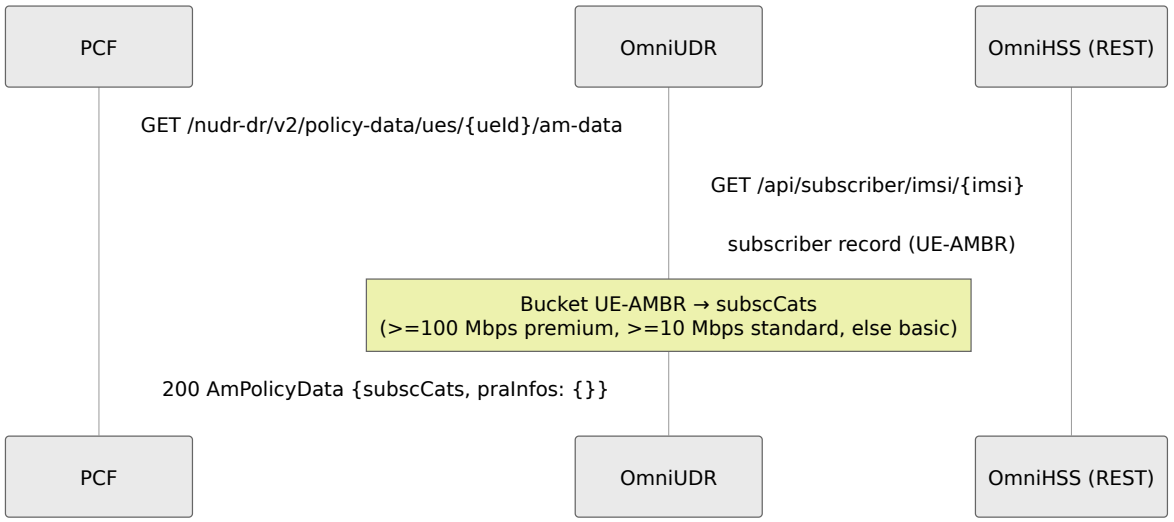
# SQN 更新（PATCH 认证订阅）


UDM
OmniUDR
OmniHSS (REST)
PATCH .../authentication-data/authentication-subscription {sequenceNumber: {sqn}}
[patch contains an SQN]
POST /api/hlr/update_sqn {imsi, sqn}
[HSS accepted]
OK
204 No Content
[HSS rejected]
error
200 PatchResult (report: [{path, op, reason}])
[no SQN in patch body]
204 No Content (no-op)
UDM
OmniUDR
OmniHSS (REST)

# AMF 注册上下文写入


UDM
OmniUDR
OmniHSS (REST)
PUT .../context-data/amf-3gpp-access {amfInstanceId, ...}
查找本地 ETS 缓存以决定创建与更新
PUT /api/hlr/circuit_session/{imsi} (位置更新)
OK
在本地 ETS 缓存中记录上下文
[此 UE 的首次注册]
201 Created + Location
[更新现有注册]
204 No Content
UDM
OmniUDR
OmniHSS (REST)





## SMF 注册上下文写入（仅本地缓存）

SMF 注册上下文仅存储在本地 ETS 缓存中，以 `{ueId}/{pduSessionId}` 为键；它从未持久化到 HSS，并且在重启时丢失。现有缓存条目的存在决定创建与更新。

## AM 策略数据读取（PCF → UDR → HSS）

AM 策略数据是粗略的：OmniUDR 从 HSS 读取订阅者的 UE-AMBR，并将其归类为单一订阅者类别（`premium` / `standard` / `basic`）；`praInfos` 始终为空。

## SM 策略数据读取（PCF → UDR → HSS）

## NRF 注册与心跳

在启动时，OmniUDR 将其 NF 配置文件注册到 NRF，然后每 `heartbeat_interval` 毫秒发送一次心跳，以便注册表保持其为活动状态。注册状态以仪表形式暴露（见 Metrics），并且 OAM 操作可以强制重新注册。

# 数据变更订阅与 DataChangeNotify

OmniUDR 实现了 Nudr_DR 数据变更订阅服务（`subscription-data/subs-to-notify` 资源，TS 29.504 subscriptions-to-notify）。消费者注册一个订阅，命名其希望监控的资源 URI 和通知回调 URI，选项上可以带有过期时间。订阅保存在内存中的（ETS）订阅存储中；当提供过期时间时，存储会遵守该时间，并在过期后删除订阅。

当监控的资源发生变化时，OmniUDR 会向每个其监控的 URI 涉及变化资源的订阅者的回调 URI 发送 `DataChangeNotify`（精确的 URI 匹配，或一个集合前缀，以便父资源上的订阅捕获其子资源）。每个通知携带一个变更类型：`ADD`、`REPLACE` 或 `REMOVE`。在此版本中，通知在 SQN 更新（对认证订阅的 PATCH）和 AMF 与 SMF 上下文写入时触发（资源的首次注册为 `ADD`，后续写入为 `REPLACE`）。

交付是尽力而为的，绝不会影响消费者收到的 SBI 响应。失败或无法访问的回调会在有[illegible]的尝试次数内重试，并采用指数退避（见 `notification_max_attempts` 和 `notification_backoff_ms`）；一旦尝试耗尽，通知将被丢弃并记录事件。

← 概述

# OmniUDR 故障排除

## 所有 Nudr 读取返回 404 / 订阅者查找失败

OmniUDR 从 HSS 后端满足每个读取请求。如果读取失败：

1. 确认 `hss_api_base_url` 从 OmniUDR 主机可访问，并且 HSS 正在运行。使用内置检查：在管理 API 上执行 `GET /api/health/hss`，并观察 `omni_udr_hss_health` 指标（1 = 正常，0 = 异常）。
2. 确认订阅者在 HSS 中存在并且是 启用 状态。禁用的订阅者（`enabled: false`）被视为未找到（对消费者返回 404）。
3. 确认 `{ueId}` 使用 `imsi-<digits>` 格式（也接受裸 IMSI）。
4. 确认 `hss_api_key` 已设置且 OmniHSS 接受它。HSS 返回 `401` 表示密钥缺失或不为 OmniHSS 所知。返回 `403` 表示密钥没有 `hlr_interworking` 角色。如果未设置 `hss_api_key`，日志中会显示一个指名 `hss_api_key` 的错误。
5. 观察 `omni_udr_hss_requests_total{result="failure"}` 和 `omni_udr_hss_request_duration_ms` 来区分 HSS 的故障/延迟与真正缺失的订阅者。

## 当 HSS 无法访问时消费者收到 5xx / 错误

当 HSS 后端宕机或超时（每个请求 5 秒，���康检查 3 秒）时，HSS REST 客户端返回错误，OmniUDR 将其呈现给消费者（通常在读取资源上返回 `404`，或在订阅者代理上返回管理 API `502`）。这是预期的 - OmniUDR 没有本地订阅者存储可供回退。恢复 HSS 可访问性；一旦下一个检查成功，`omni_udr_hss_health` 将返回 1。请注意，`test_subscribers` 是实验室辅助工具，并且在此构建中 不是 自动 HSS 宕机的回退。

## SQN 更新的 PATCH 返回 200 而不是 204

返回 `200` 和 `PatchResult` 主体意味着 HSS 拒绝了 SQN 更新；`report` 数组列出了失败的路径和 HSS 原因。成功的更新返回 `204`。检查 HSS 日志和 `/api/hlr/update_sqn` 路径。没有 `sequenceNumber` 的 PATCH 主体被视为成功的无操作（`204`）。

## 身份验证向量生成返回 404 / 500

`authentication-vectors` POST 查找订阅者，然后委托给 HSS 的 `/api/hlr/generate_eutran_auth_vectors` 端点。返回 `404` 表示订阅者在 HSS 中未知或已禁用；返回 `500` 表示 HSS 返回了错误或意外的响应格式。在重新同步时，确保 UDM 提供了 `resynchronizationInfo`，其中包含 `rand` 和 `auts` - OmniUDR 将它们连接成 HSS 所期望的 `rand||auts` 十六进制格式。此端点是专有的，仅应由 OmniUDM 调用。

## 策略数据看起来稀薄

AM 策略数据仅返回来自订阅者的 UE-AMBR 级别派生的粗略 `subscCats`（`premium` ≥ 100 Mbps，`standard` ≥ 10 Mbps，否则为 `basic`），并且 `praInfos` 为空。该级别使用上行或下行 UE-AMBR 中较高的值。当订阅者的 HSS EPC 配置文件没有 UE-AMBR 时，OmniUDR 提供 1000 Kbps 的默认值，因此未配置或错误映射的订阅者将被归入 `basic` - 如果您期望的订阅者为 `premium`/`standard` 但显示为 `basic`，请验证 HSS 中的 EPC UE-AMBR（`ue_ambr_ul_kbps` / `ue_ambr_dl_kbps`）是否已配置。SM 策略数据是根据订阅者的 APN/QoS 和 PCRF 计费规则配置文件构建的。当前架构中没有专用的 5G 策略存储；这是设计使然。

## AM 数据切片 (S-NSSAI) 不匹□

4G HSS 记录不携带 5G 切片订阅，因此 AM 数据 NSSAI 由 OmniUDR 合成，并默认为标准实验室切片（eMBB、URLLC、mIoT）。如果 UE 请求的 SD 合格 S-NSSAI 被 AMF 拒绝（5GMM 原因 #62），请将 `default_am_nssai`（以及每个 DNN 的 `dnn_snssai_map`）与 AMF 提供的 NSSAI 对齐。请参见 配置参考。

## SM 数据为每个订阅者提供固定的 AMBR

如果 SM 订阅数据对于所有订阅者显示相同的会话 AMBR / QoS，而不考虑配置，则 HSS APN QoS 配置文件字段名称未被匹配，OmniUDR 正在回退到默认值。映射期望 OmniHSS 的 `ApnQosProfile` 字段（`apn_ambr_dl_kbps`、`apn_ambr_ul_kbps`、`allocation_retention_priority`、`pre_emption_capability`、`pre_emption_vulnerability`、`qci`）。验证 HSS 订阅者记录是否携带这些字段。

# 上下文缓存在带外 HSS 更改后似乎过时

AMF/SMF 注册缓存仅跟踪上下文写入，并用于 201 与 204 的决策；它不缓存读取数据。如果需要清除（例如，在测试循环后），请通过 `DELETE /api/oam/subscriber-cache/{imsi}`（或 `.../all`）进行刷新。