← Overview

OmniUDM Configuration

OmniUDM is configured from environment variables at boot via `config/runtime.exs`, under the `:omniudm` application key. Every value has a built-in default, so the process starts with no environment set; production deployments should set at least the SBI, NRF, UDR, PLMN, and `hnet_key_dir` values.

Example Configuration

```
config :omniudm,
  udr_uri: "http://127.0.0.22:7777",
  sbi_scheme: "http",
  sbi_addr: "127.0.0.12",
  sbi_port: 7777,
  nrf_uri: "http://127.0.0.1:7777",
  mcc: "999",
  mnc: "70",
  prometheus_metrics_port: 9568,
  heartbeat_interval: 10_000,
  hnet_key_dir: "/etc/omniudm/hnet"

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

Parameter Reference

Parameter	Type	Required	Default
`udr_uri`	string (URI)	No	`http://127.0.0.22:7777`
`sbi_scheme`	string	No	`http`
`sbi_addr`	string (IP)	No	`127.0.0.12`

Parameter	Type	Required	Default
`sbi_port`	integer	No	`7777`
`nrf_uri`	string (URI)	No	`http://127.0.0.1:7777`
`mcc`	string	No	`999`
`mnc`	string	No	`70`
`prometheus_metrics_port`	integer	No	`9568`

Parameter	Type	Required	Default
`heartbeat_interval`	integer (ms)	No	`10000`
`hnet_key_dir`	string (path)	No	`/etc/omniudm/hnet`

Optional Tuning Keys

These keys are read from application config and are not surfaced as environment variables in `runtime.exs`; set them in config if a deployment needs to override the defaults.

Parameter	Type	Default	Description
`default_nssai`	map	eMBB / URLLC / mIoT lab slices (see below)	Fallback NSSAI subscription returned by `GET /{supi}/nssai` when the UDR `am-data` response carries no explicit `nssai`. Set this to advertise the network's real slices, including SD-qualified S-NSSAIs
`nf_capacity`	integer (0-65535)	Application default	Relative processing capacity of this NF instance advertised in the NF profile registered with the NRF. Consumers weight NF selection by capacity. Adjustable at runtime via the OAM config PATCH endpoint (`PATCH /api/oam/config/{id}`); typically has no dedicated environment variable
`nf_priority`	integer (0-65535)	Application default	Relative selection priority of this NF instance advertised in the NF profile registered with the NRF (lower value is higher priority). Adjustable at runtime via the OAM config PATCH endpoint (`PATCH /api/oam/config/{id}`); typically has no dedicated environment variable

The built-in `default_nssai` fallback advertises `defaultSingleNssais` of `sst 1` (and `sst 1 / sd 000001`) and `singleNssais` covering `sst 1`, `sst 1 / sd 000001`, `sst 2 / sd 000002` (URLLC), and `sst 3 / sd 000003` (mIoT). A subscriber offered a slice the AMF is not configured to serve is rejected with

5GMM cause #62, so this fallback must match what the network actually serves.

The value is an `Nssai` object (TS 29.503 / TS 29.571) with the following sub-fields:

Field	Type	Semantics
`defaultSingleNssais`	list of S-NSSAI	Slices the UE is allowed to use *without* explicitly requesting them - the subscribed default set the AMF applies when the UE requests no NSSAI. Must be a subset of `singleNssais`
`singleNssais`	list of S-NSSAI	The full set of subscribed slices the UE is entitled to on this PLMN

Each S-NSSAI entry is a map:

Field	Type	Semantics
`sst`	integer (0–255)	Slice/Service Type. Standardised values: `1` = eMBB, `2` = URLLC, `3` = mIoT
`sd`	string (6 hex digits)	Optional Slice Differentiator qualifying the `sst`. Omit for an SST-only slice; an SST that the network serves *only* in SD-qualified form must be listed with its `sd` (an unqualified entry will not match)

An S-NSSAI in `defaultSingleNssais` that is not also present in `singleNssais`, or a slice here that no AMF is configured to serve, leads to the AMF rejecting the UE with 5GMM cause #62.

Home Network Keys (`hnet_key_dir`)

`hnet_key_dir` is the directory holding the Home Network private keys used for ECIES Profile A/B SUCI de-concealment. Each SIM is provisioned with a Home Network **public** key and its key identifier (`home_network_public_key_id`); OmniUDM must hold the matching **private** key under that same identifier.

Key files are loaded once at first use and cached in memory. A directory that does not exist is tolerated (a warning is logged and ECIES de-concealment is effectively disabled - Null-scheme SUCIs still work).

File naming. The key identifier and curve are taken from the filename. Two conventions are accepted:

Convention	Profile A (Curve25519)	Profile B (P-256)
Dotted `<id>.<curve>.key`	`1.x25519.key`	`2.secp256r1.key`
Dashed `<curve>-<id>.key`	`curve25519-1.key` or `x25519-1.key`	`secp256r1-2.key`

The numeric `<id>` must match the `home_network_public_key_id` provisioned on the SIM. Files that do not match a recognised pattern, and files without a `.key` extension, are skipped with a warning.

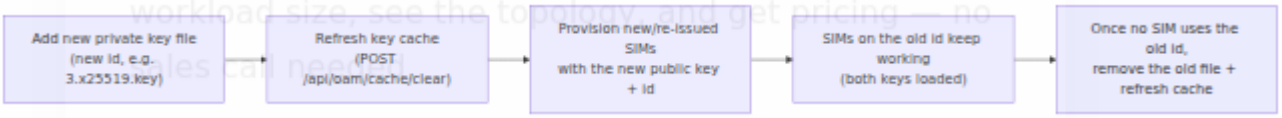
File contents. Each file holds the raw private key in one of three encodings, auto-detected:

- **PEM** - PKCS#8 (`-----BEGIN PRIVATE KEY-----`) or SEC1 (`-----BEGIN EC PRIVATE KEY-----`). The raw private scalar is extracted automatically. This is the recommended field format.
- **Hex** - a single hex-encoded string of the raw private key bytes.
- **Raw** - the raw private key bytes as-is.

A single corrupt or malformed key file is skipped (logged as a warning) without preventing the other keys from loading.

How a key identifier maps to a SIM. A SUCI carries the key identifier the SIM used to conceal its SUPI: `suci-0-<mcc>-<mnc>-<routing-ind>-<scheme>-<home_nw_pub_key_id>-<scheme_output>`. OmniUDM reads `<home_nw_pub_key_id>` from the SUCI and selects the private key file whose numeric `<id>` matches. Several key identifiers can therefore be served at once - one file per identifier in use across the SIM estate - and a SUCI whose identifier has no loaded key fails with `unknown_home_network_key_id`.

Rotating a Home Network key (zero-downtime). Because de-concealment is keyed by identifier and multiple identifiers coexist, a key is rolled by introducing a *new* identifier alongside the old rather than overwriting a file in place:



Overwriting the file for an identifier still in use on live SIMs breaks de-concealment for those SIMs (they present the old public key, MAC verification fails). Always retire an identifier only after every SIM using it has been re-provisioned onto a new one.

Operational care.

- Treat the contents of `hnet_key_dir` as highly sensitive secret material - compromise of a Home Network private key allows any party to de-conceal SUCIs for that key identifier. Restrict directory and file permissions to the OmniUDM service account only.
- The private key here must be the counterpart of the public key provisioned on the SIMs for the same key identifier. A mismatch causes MAC verification to fail and authentication to be rejected.
- Keys are cached after first load. After adding, replacing, or removing key files, the cache must be refreshed (see the OAM cache-clear management action, `POST /api/oam/cache/clear`) or the process restarted for the change to take effect.

Management / OAM API

An HTTPS management/OAM API is served (default port `8443`, TLS enabled) for operational inspection and control. It is separate from the SBI listener.

Method	Path	Purpose
GET	`/api/status/api`	Management API status
GET	`/api/status/license`	License status
GET	`/api/status/nrf`	NRF registration status
GET	`/api/status/nf`	NF status summary
GET	`/api/ue`	List active in-memory UE contexts
GET	`/api/ue/{supi}`	Inspect a single UE context by SUPI
GET	`/api/statistics`	Service statistics, including a live count of active in-memory UE contexts
GET	`/api/oam/config`	Read current effective configuration
PATCH	`/api/oam/config/{id}`	Update a configuration value (PUT is also accepted)
DELETE	`/api/oam/ue_cache/{id}`	Evict a UE context from the in-memory cache
POST	`/api/oam/ue/sdm-notify`	Manually trigger an SDM data-change notification for a SUPI. POSTs a `ModificationNotification` to every active SDM subscription's `callbackReference` (e.g. after out-of-band provisioning changes)

Method	Path	Purpose
DELETE	`/api/oam/ue/amf-registration/{id}`	Force AMF deregistration for a UE. Clears the stored AMF 3GPP registration *and* notifies the serving AMF at its `deregCallbackUri` so it releases the UE (dereg reason `SUBSCRIPTION_WITHDRAWN`); returns 404 if the UE has no AMF registration
POST	`/api/oam/cache/clear`	Clear caches (including the Home Network key cache)
POST	`/api/oam/log_level`	Change the runtime log level (`{"level": ...}`)
GET	`/api/oam/resources`	Resource inventory
GET	`/api/oam/diagnostics`	Diagnostics snapshot
POST	`/api/oam/nrf/reregister`	Force re-registration with the NRF

Logging

Setting	Value	Description
`format`	`{OmniLogger.JsonFormatter, :format}`	Log lines are emitted as structured JSON via the shared OmniLogger formatter, suitable for ingestion by a log aggregator
`metadata`	`:all`	All log metadata (including per-request session context such as SUPI and procedure) is included in each line

The runtime log level can be changed on the fly via the Management/OAM API (`POST /api/oam/log_level`). Key material (CK/IK/KAUSF/SQN⊕AK) is never logged; only non-secret context such as the serving network name and the de-concealed SUPI appears in logs.

← Overview

OmniUDM Metrics

Metrics are exposed in Prometheus format on `prometheus_metrics_port` (default `9568`) at `/metrics`.

UDM Service Metrics

Metric	Type	Labels	Des
`omni_udm_auth_requests_count`	counter	`supi`, `result`	Auth vect requ outc (`suc` `fail`
`omni_udm_auth_vector_generations_total`	counter	`result`	Auth gene atte outc
`omni_udm_auth_result_count`	counter	`supi`, `result`	Auth resu reco `auth` (`res` repo `succ` valu
`omni_udm_uecm_registrations_total`	counter	`access_type`	UEC regis acro acce Both and AMF regis path incre cour tagg

Metric	Type	Labels	Des
			`acce` (`3gp` `non_`
`omni_udm_sdm_queries_total`	counter	`data_type`	SDM quer data (`am-` `data` `sele` `nssa` `cont` `smf-` `data`
`omni_udm_active_ue_contexts_count`	gauge	none	Num activ mem cont Emit UE-c crea dele
`omni_udm_nrf_registration_status`	gauge	`nf_type`	NRF regis statu regis = nc

Useful Queries

```
# Authentication failure rate over 5 minutes
sum(rate(omni_udm_auth_requests_count{result="failure"}[5m]))
  / sum(rate(omni_udm_auth_requests_count[5m]))

# SDM query rate by data type
sum by (data_type) (rate(omni_udm_sdm_queries_total[5m]))

# UECM registration rate split by access type (3gpp / non_3gpp)
sum by (access_type) (rate(omni_udm_uecm_registrations_total[5m]))

# Auth vector generation failure rate
sum(rate(omni_udm_auth_vector_generations_total{result="failure"}
[5m]))
  / sum(rate(omni_udm_auth_vector_generations_total[5m]))

# Alert if UDM is not registered with the NRF
min(omni_udm_nrf_registration_status) < 1
```

`omni_udm_active_ue_contexts_count` tracks the live number of active in-memory UE contexts; it is updated on every UE-context create and delete. The OAM `/api/statistics` and `/api/status/nf` endpoints report the same count on demand.

BEAM VM Metrics

Metric	Type	Description
`beam_memory_total`	gauge	Total BEAM memory (bytes)
`beam_memory_processes`	gauge	Memory allocated to Erlang processes
`beam_memory_processes_used`	gauge	Memory actually used by processes
`beam_memory_system`	gauge	System memory (ETS, atoms, code)
`beam_memory_atom`	gauge	Total atom memory
`beam_memory_atom_used`	gauge	Used atom memory
`beam_memory_binary`	gauge	Binary memory
`beam_memory_code`	gauge	Code memory
`beam_memory_ets`	gauge	ETS table memory
`beam_processes_count`	gauge	Number of Erlang processes
`beam_ports_count`	gauge	Number of Erlang ports
`beam_atom_count`	gauge	Number of atoms
`beam_vm_uptime`	gauge	VM uptime (seconds)

← Overview

OmniUDM Operations

OmniUDM is the Unified Data Management (UDM) function of the Omnitouch 5G Core. It presents subscriber identity, subscription, and context data to the rest of the core through the standard Nudm services, backing all authoritative data onto the Unified Data Repository (OmniUDR) over `nudr-dr`.

3GPP Role and Specification References

Aspect	Reference
UDM functional role in the 5GC	TS 23.501
Nudm services (SDM / UEAU / UECM)	TS 29.503
Nudr_DataRepository (consumed)	TS 29.504
SUCI / SUPI concealment and protection schemes	TS 33.501 §6.12 and Annex C
SUCI / SUPI structure and encoding	TS 23.003 §2.2B
5G-AKA authentication and key hierarchy	TS 33.501 §6.1.3, Annex A
Milenage algorithm set	TS 35.206
Mobile identity (SUCI/MSIN BCD) encoding	TS 24.501

Interfaces and SBI Endpoints

OmniUDM registers the following services with the NRF and serves them on the SBI listener (`sbi_scheme://sbi_addr:sbi_port`):

Service	Version	Purpose
`nudm-sdm`	v2	Subscriber data management
`nudm-ueau`	v1	UE authentication vector generation
`nudm-uecm`	v1	UE context (AMF/SMF registration) management

All authoritative subscriber and context data is read from / written to OmniUDR via `nudr-dr/v2` (base path `.../nudr-dr/v2/subscription-data/{ueId}/...`). All request/response bodies are JSON. Errors are returned as `application/problem+json` ProblemDetails.

Nudm_UEAuthentication (`nudm-ueau/v1`)

Method	Path	Description
POST	`/{supiOrSuci}/security-information/generate-auth-data`	Generate a 5G-AKA authentication vector. Accepts a SUPI or SUCI; de-conceals SUCI first. Requires `servingNetworkName` and `ausfInstanceId`
POST	`/{supi}/auth-events`	Record an authentication result event. Requires `nfInstanceId`, `success`, `timeStamp`, `authType`, `servingNetworkName`
PUT	`/{supi}/auth-events/{authEventId}`	Update / remove an authentication event (AuthEvent body, `authRemovalInd=true` clears it)

Nudm_UECM (`nudm-uecm/v1`)

Method	Path	Description
PUT	`/{supi}/registrations/amf-3gpp-access`	AMF registers 3GPP access for a UE
PATCH	`/{supi}/registrations/amf-3gpp-access`	Update parameters of the stored AMF 3GPP registration (merge-patch)
GET	`/{supi}/registrations/amf-3gpp-access`	Retrieve the stored AMF 3GPP registration
PUT	`/{supi}/registrations/amf-non-3gpp-access`	AMF registers non-3GPP access for a UE
GET	`/{supi}/registrations/amf-non-3gpp-access`	Retrieve the stored AMF non-3GPP registration
PUT	`/{supi}/registrations/smf-registrations/{pduSessionId}`	SMF registers a PDU session context
DELETE	`/{supi}/registrations/smf-registrations/{pduSessionId}`	SMF deregisters a PDU session context

Nudm_SDM (`nudm-sdm/v2`)

Method	Path	Description
GET	`/{supi}`	GetDataSets - retrieve multiple data sets selected by the mandatory `dataset-names` query parameter (comma-separated `AM`, `SM`, `SMF_SEL`, `UEC_SMF`)
GET	`/{supi}/am-data`	Access and mobility subscription data (optional `plmn-id` query)
GET	`/{supi}/sm-data`	Session management subscription data (optional `plmn-id` query)
GET	`/{supi}/smf-select-data`	SMF selection subscription data (optional `plmn-id` query)
GET	`/{supi}/nssai`	Network slice selection subscription data
GET	`/{supi}/ue-context-in-smf-data`	UE context in SMF (active PDU sessions): see note below
POST	`/{supi}/sdm-subscriptions`	Subscribe to data-change notifications. Requires `nfInstanceId`, `callbackReference`, `monitoredResourceUris`

Method	Path	Description
GET	`/{supi}/sdm-subscriptions/{subscriptionId}`	Retrieve a stored SDM subscription
PATCH	`/{supi}/sdm-subscriptions/{subscriptionId}`	Modify an SDM subscription (merge-patch)
DELETE	`/{supi}/sdm-subscriptions/{subscriptionId}`	Remove an SDM subscription

SUCI / SUPI Concealment

To protect the subscriber's permanent identity over the air, the UE sends a **SUCI** (Subscription Concealed Identifier) instead of the **SUPI** (Subscription Permanent Identifier, i.e. `imsi-<IMSI>`). When the AUSF asks OmniUDM to generate an authentication vector for a SUCI, OmniUDM **de-conceals the SUCI to the SUPI** before it can look up subscriber data in UDR.

OmniUDM supports the three protection schemes defined in TS 33.501 §6.12 / Annex C:

Scheme	Name	Curve / Crypto	Requires HNET key?
0	Null scheme	None - MSIN is in cleartext	No
1	ECIES Profile A	Curve25519 (X25519) + AES-128-CTR + HMAC-SHA-256	Yes
2	ECIES Profile B	secp256r1 / P-256 + AES-128-CTR + HMAC-SHA-256	Yes

For the **Null scheme**, the MSIN is carried in the clear; it is **BCD-decoded** from the scheme output to reconstruct the IMSI (per TS 24.501 - the MSIN digits are BCD-encoded, not hex). For **Profile A and Profile B**, de-concealment performs an ephemeral-static ECDH against the operator's **Home Network private key**, derives the encryption/MAC keys (ANSI-X9.63-KDF with SHA-256), verifies the MAC, decrypts the concealed MSIN, and BCD-decodes it to reconstruct the IMSI.

Home Network private keys for Profile A/B are loaded from the `hnet_key_dir` directory. File naming, encodings, and the security and cache-refresh requirements are documented in Configuration → Home Network Keys (`hnet_key_dir`).

Key Procedures

SUCI De-concealment and Authentication Vector Generation

The AUSF requests an authentication vector; OmniUDM resolves the identity (de-concealing a SUCI if needed), fetches key material and a base vector from the HSS via UDR, then re-derives the 5G-specific AUTN and keys. **SQN management and any resynchronisation are handled by the HSS behind UDR** - OmniUDM re-derives the 5G AUTN (separation bit set), `XRES*`, and `KAUSF` from the returned material. A `resynchronizationInfo` supplied by the AUSF is forwarded to the HSS via UDR.

Parse error on line 20: ...compute AUTN with 5G
separation bit; ----------^ Expecting 'NEWLINE', 'AS', ',', 'SOLID_OPEN_ARROW', 'DOTTED_OPEN_ARROW', 'SOLID_ARROW', 'BIDIRECTIONAL_SOLID_ARROW', 'DOTTED_ARROW', 'BIDIRECTIONAL_DOTTED_ARROW', 'SOLID_CROSS', 'DOTTED_CROSS', 'SOLID_POINT', 'DOTTED_POINT', 'TXT', got 'INVALID'

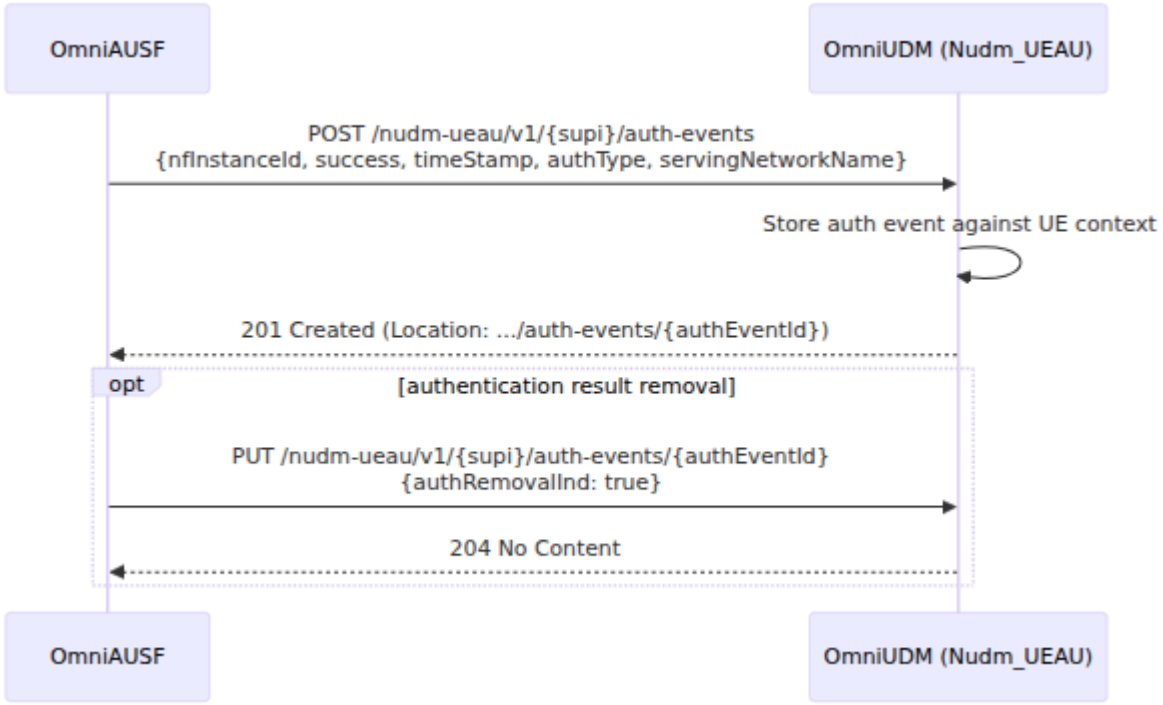
Try again

De-concealment failures return HTTP 403 (`SUCI De-concealment Failed`), an unsupported protection scheme returns HTTP 501, a missing subscriber returns

HTTP 404 (`USER_NOT_FOUND`), and invalid key material or a failed HSS vector generation returns HTTP 500.

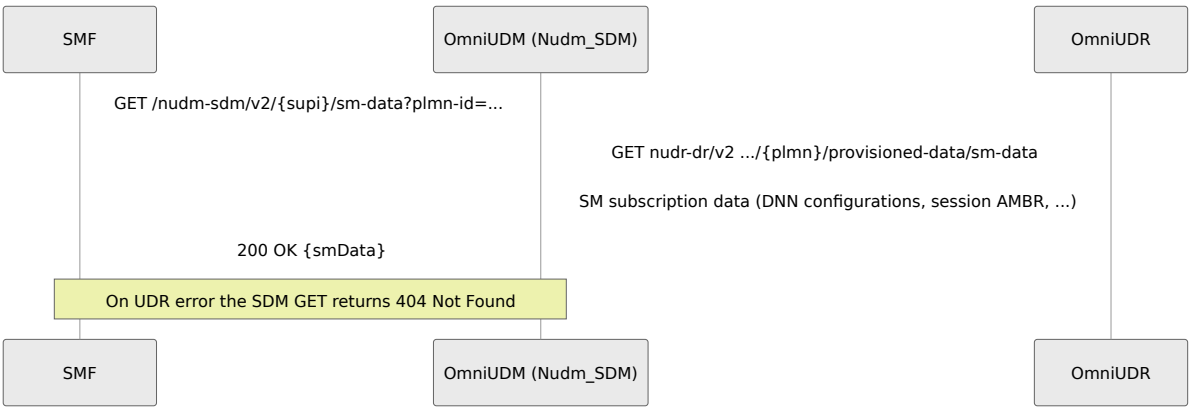
Authentication Result Event Recording

After the AUSF completes authentication, it records the outcome. OmniUDM stores the event against the UE context and returns a `Location` for the created event; the AUSF may later `PUT` an `AuthEvent` with `authRemovalInd=true` to clear it (204 No Content).



SDM Subscriber Data Fetch (`sm-data`)

After registration the AMF/SMF fetch subscription data. OmniUDM proxies the request to UDR, adding the serving PLMN, and returns the UDR response. For `sm-data`, a single UDR object is wrapped in an array per TS 29.503 `SmSubsData`.

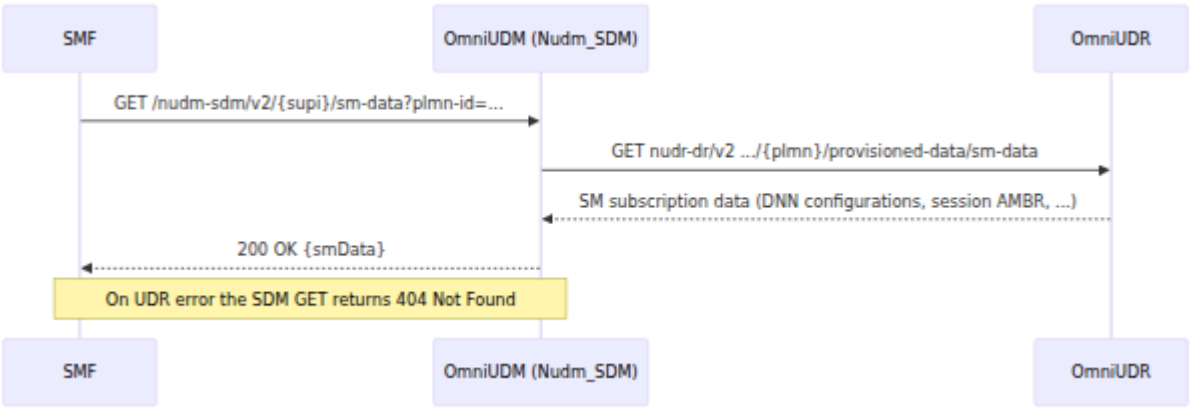


The `am-data`, `smf-select-data`, and GetDataSets (`GET /{supi}`) flows follow the same proxy pattern against UDR. GetDataSets assembles only the requested data sets it can resolve, skipping any that error. `nssai` is derived from the UDR `am-data` response (see below).

> **UE-context-in-SMF returns live PDU session context.** `GET /{supi}/ue-context-in-smf-data` and the `UEC_SMF` data set in GetDataSets return the UE's active PDU session contexts as reported by the SMF via the Nudm_UECM SMF registration procedure. Each stored SMF registration is projected into the `pduSessions` map (keyed by PDU session ID) per TS 29.503. The map is empty (`{"pduSessions": {}}`) only when the UE genuinely holds no SMF registrations.

NSSAI Retrieval with Default Fallback

`GET /{supi}/nssai` fetches `am-data` from UDR and extracts the embedded `nssai`. If UDR returns `am-data` without an `nssai`, or the UDR lookup fails, OmniUDM returns the configurable `default_nssai` fallback so the AMF always receives a slice list.



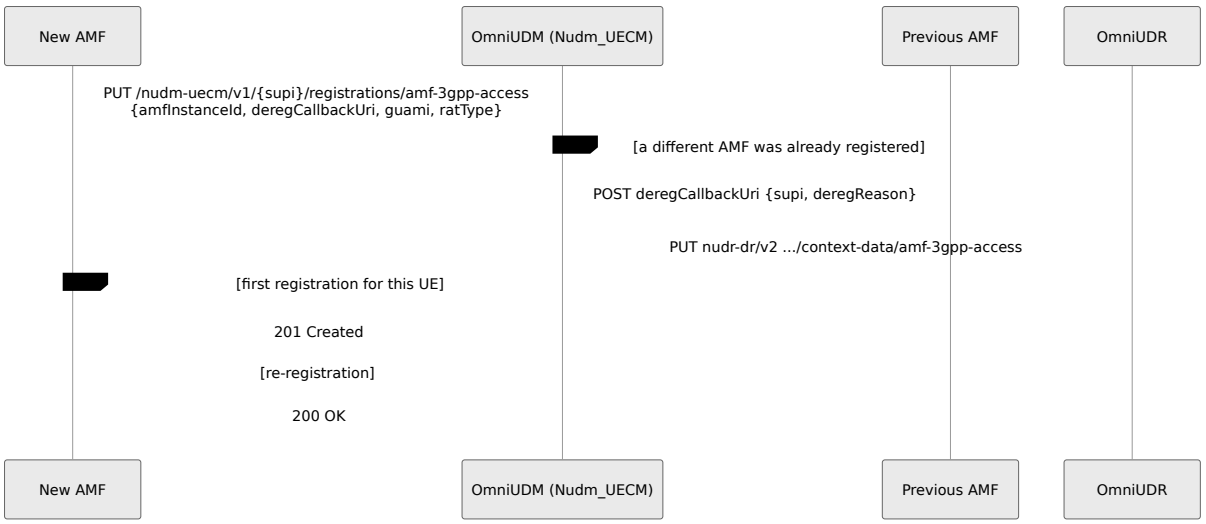
The `default_nssai` fallback must match the slices the network actually serves - a subscriber offered a slice the AMF is not configured to serve is rejected with 5GMM cause #62. See Configuration → Optional Tuning Keys.

SDM Change Notifications

Subscribers with an active SDM subscription are notified of subsequent changes: OmniUDM POSTs a `ModificationNotification` to each subscription's `callbackReference` (asynchronously) when the monitored resources change. A notification can also be triggered manually for a SUPI via the Management/OAM API.

UECM AMF Registration and Mobility

When a UE registers, the AMF registers itself for that UE at OmniUDM. If a different AMF was previously registered, OmniUDM notifies the old AMF's `deregCallbackUri` so it can release the UE, then persists the new registration to UDR.



Mandatory IEs are validated per TS 29.503: AMF 3GPP registration requires `amfInstanceId`, `deregCallbackUri`, `guami`, `ratType`; SMF registration requires `smfInstanceId`, `pduSessionId`, `singleNssai`, `plmnId`, and the body `pduSessionId` must match the path (a mismatch returns `MANDATORY_IE_INCORRECT`). Missing IEs return HTTP 400.

← Overview

OmniUDM Troubleshooting

Authentication fails with 404 User Not Found

OmniUDM resolved the identity but UDR returned no authentication subscription. Confirm `udr_uri` is reachable, OmniUDR can reach the subscriber database, and the IMSI is provisioned. For a SUCI input, confirm de-concealment produced the expected IMSI (check the "De-concealed SUCI to ..." log line).

Authentication fails with 403 SUCI De-concealment Failed

OmniUDM could not de-conceal the SUCI. The detail field names the specific cause:

- `unknown_home_network_key_id` - no private key for the SUCI's key identifier is loaded. Confirm a key file for that `home_network_public_key_id` exists in `hnet_key_dir` and follows a recognised naming convention, and that the key cache has been refreshed since the file was added.
- `ecies_mac_verification_failed` - the loaded private key does not match the public key on the SIM (wrong or swapped key), or the SUCI is corrupted. Verify the key pair provisioned on the SIMs against the deployed private key for that identifier.
- `ecies_profile_a_scheme_output_too_short` / `ecies_profile_b_scheme_output_too_short` - the SUCI scheme output is truncated / malformed.
- `invalid_suci_format` - the SUCI string is not well formed.

Authentication fails with 501 Unsupported Protection Scheme

The SUCI used a protection scheme other than 0 (Null), 1 (Profile A), or 2 (Profile B). Only these three are supported.

Authentication fails with 500 Internal Error

The subscriber's key material from UDR (`encPermanentKey`/`encOpcKey`) or the HSS vector fields (`rand`/`ck`/`ik`/`autn`) were absent, not valid hex, or the wrong length, or HSS vector generation failed. Verify the subscriber's Ki/OPc provisioning in the subscriber database behind UDR and that OmniUDR can reach the HSS.

`am-data` / `sm-data` / `smf-select-data` returns 404 Not Found

The UDR lookup failed for that data set. Confirm the subscriber's provisioned data exists in UDR for the serving PLMN, that `mcc`/`mnc` (or the request's `plmn-id`) match how the data is provisioned, and that `udr_uri` is reachable.

UE rejected by AMF with 5GMM cause #62 (no supported slice)

The NSSAI served to the AMF does not include a slice the AMF is configured to serve. If the subscriber's UDR `am-data` carries no `nssai`, OmniUDM returns the `default_nssai` fallback - set `default_nssai` (or provision `nssai` in UDR) to match the network's real slices, including any SD-qualified S-NSSAIs.

UE context or registration not present after restart

Per-UE runtime state (authentication context, AMF/SMF registrations, SDM subscriptions) is held in an in-memory context store and is rebuilt by re-registration; authoritative subscriber data persists in OmniUDR. After a restart, AMF/SMF re-register on the next periodic registration update, PDU session re-establishment, or UE power cycle. If a specific registration is missing sooner than expected, trigger the consumer NF's registration update rather than waiting for the periodic timer.

SDM data-change notification not received

Confirm the subscription's `callbackReference` URI is reachable from the OmniUDM host. Notifications are sent asynchronously; look for "SDM notification to ... failed" warnings. A notification can be triggered manually for a SUPI via the Management/OAM API (`POST /api/oam/ue/sdm-notify`).

After changing Home Network keys, de-concealment still uses the old key

Home Network keys are cached in memory after first load. Hot-reload them via the OAM management action (`POST /api/oam/hnet/reload`) after adding or replacing key files in `hnet_key_dir`; the response reports the number of keys loaded and their key ids. No process restart is required.

UDM not discoverable by other NFs

Check `omni_udm_nrf_registration_status`. If `0`, confirm `nrf_uri` is reachable and the heartbeat is succeeding; the OAM management API offers an NRF re-

registration action (`POST /api/oam/nrf/reregister`).