← Visão Geral

Configuração do OmniUDM

O OmniUDM é configurado a partir de variáveis de ambiente na inicialização via `config/runtime.exs`, sob a chave de aplicativo `:omniudm`. Cada valor possui um padrão embutido, portanto, o processo começa sem ambiente definido; implantações em produção devem definir pelo menos os valores de SBI, NRF, UDR, PLMN e `hnet_key_dir`.

Exemplo de Configuração

```
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
```

Referência de Parâmetros

Parâmetro	Tipo	Obrigatório	Padrão
`udr_uri`	string (URI)	Não	`http://127.0.0.22`
`sbi_scheme`	string	Não	`http`

Parâmetro	Tipo	Obrigatório	Padrão
`sbi_addr`	string (IP)	Não	`127.0.0.12`
`sbi_port`	inteiro	Não	`7777`
`nrf_uri`	string (URI)	Não	`http://127.0.0.1:`
`mcc`	string	Não	`999`

Parâmetro	Tipo	Obrigatório	Padrão
`mnc`	string	Não	`70`
`prometheus_metrics_port`	inteiro	Não	`9568`
`heartbeat_interval`	inteiro (ms)	Não	`10000`
`hnet_key_dir`	string (caminho)	Não	`/etc/omniudm/hnet`

Chaves de Ajuste Opcionais

Essas chaves são lidas da configuração do aplicativo e não são expostas como variáveis de ambiente em `runtime.exs`; defina-as na configuração se uma implantação precisar substituir os padrões.

Parâmetro	Tipo	Padrão	Descrição
`default_nssai`	mapa	fatias de laboratório eMBB / URLLC / mIoT (veja abaixo)	NSSAI de fallback retornado por `GET /{supi}/nssai` quando a resposta `am-data` do UDR não carrega um `nssai` explícito. Defina isso para anunciar as fatias reais da rede, incluindo S-NSSAIs qualificados por SD
`nf_capacity`	inteiro (0-65535)	Padrão do aplicativo	Capacidade de processamento relativa desta instância NF anunciada no perfil NF registrado com o NRF. Consumidores ponderam a seleção NF pela capacidade. Ajustável em tempo de execução via o endpoint PATCH de configuração OAM (`PATCH /api/oam/config/{id}`); normalmente não possui uma variável de ambiente dedicada
`nf_priority`	inteiro (0-65535)	Padrão do aplicativo	Prioridade de seleção relativa desta instância NF anunciada no perfil NF registrado com o NRF (valor mais baixo é maior prioridade). Ajustável em tempo de execução via o endpoint PATCH de configuração OAM (`PATCH /api/oam/config/{id}`); normalmente não possui uma variável de ambiente dedicada

O fallback embutido `default_nssai` anuncia `defaultSingleNssais` de `sst 1` (e `sst 1 / sd 000001`) e `singleNssais` cobrindo `sst 1`, `sst 1 / sd 000001`, `sst 2 / sd 000002` (URLLC) e `sst 3 / sd 000003` (mIoT). Um assinante

oferecido uma fatia que o AMF não está configurado para servir é rejeitado com a causa 5GMM #62, portanto, esse fallback deve corresponder ao que a rede realmente serve.

O valor é um objeto `Nssai` (TS 29.503 / TS 29.571) com os seguintes subcampos:

Campo	Tipo	Semântica
`defaultSingleNssais`	lista de S-NSSAI	Fatias que o UE pode usar *sem* solicitá-las explicitamente - o conjunto padrão assinado que o AMF aplica quando o UE não solicita NSSAI. Deve ser um subconjunto de `singleNssais`
`singleNssais`	lista de S-NSSAI	O conjunto completo de fatias assinadas às quais o UE tem direito neste PLMN

Cada entrada S-NSSAI é um mapa:

Campo	Tipo	Semântica
`sst`	inteiro (0–255)	Tipo de Fatia/Serviço. Valores padronizados: `1` = eMBB, `2` = URLLC, `3` = mIoT
`sd`	string (6 dígitos hexadecimais)	Diferenciador de Fatia opcional qualificando o `sst`. Omitir para uma fatia apenas SST; uma SST que a rede serve *apenas* em forma qualificada por SD deve ser listada com seu `sd` (uma entrada não qualificada não corresponderá)

Um S-NSSAI em `defaultSingleNssais` que não está também presente em `singleNssais`, ou uma fatia aqui que nenhum AMF está configurado para servir, leva à rejeição do UE pelo AMF com a causa 5GMM #62.

Chaves da Rede Doméstica (`hnet_key_dir`)

`hnet_key_dir` é o diretório que contém as chaves privadas da Rede Doméstica usadas para desvelamento de SUCI do ECIES Profile A/B. Cada SIM é provisionado com uma chave **pública** da Rede Doméstica e seu identificador de chave (`home_network_public_key_id`); o OmniUDM deve manter a chave **privada** correspondente sob esse mesmo identificador.

Os arquivos de chave são carregados uma vez na primeira utilização e armazenados em cache na memória. Um diretório que não existe é tolerado (um aviso é registrado e o desvelamento de ECIES é efetivamente desativado - SUCIs com esquema nulo ainda funcionam).

Nomeação de arquivos. O identificador da chave e a curva são retirados do nome do arquivo. Duas convenções são aceitas:

Convenção	Perfil A (Curve25519)	Perfil B (P-256)
Pontuada `<id>.<curve>.key`	`1.x25519.key`	`2.secp256r1.key`
Hifenizada `<curve>-<id>.key`	`curve25519-1.key` ou `x25519-1.key`	`secp256r1-2.key`

O `<id>` numérico deve corresponder ao `home_network_public_key_id` provisionado no SIM. Arquivos que não correspondem a um padrão reconhecido, e arquivos sem uma extensão `.key`, são ignorados com um aviso.

Conteúdo do arquivo. Cada arquivo contém a chave privada bruta em uma das três codificações, detectadas automaticamente:

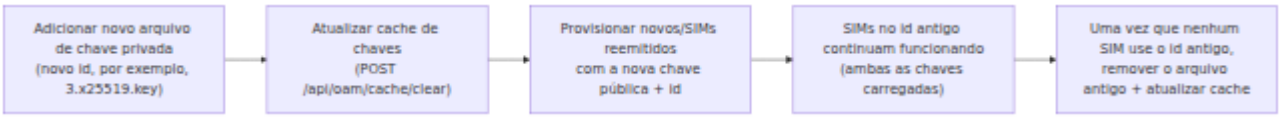
- **PEM** - PKCS#8 (`-----BEGIN PRIVATE KEY-----`) ou SEC1 (`-----BEGIN EC PRIVATE KEY-----`). O escalar privado bruto é extraído automaticamente. Este é o formato de campo recomendado.
- **Hex** - uma única string codificada em hex dos bytes da chave privada bruta.

- **Raw** - os bytes da chave privada bruta como estão.

Um único arquivo de chave corrompido ou malformado é ignorado (registrado como um aviso) sem impedir que as outras chaves sejam carregadas.

Como um identificador de chave se mapeia para um SIM. Uma SUCI carrega o identificador de chave que o SIM usou para ocultar seu SUPI: `suci-0-<mcc>-<mnc>-<routing-ind>-<scheme>-<home_nw_pub_key_id>-<scheme_output>`. O OmniUDM lê `<home_nw_pub_key_id>` da SUCI e seleciona o arquivo de chave privada cujo `<id>` numérico corresponde. Vários identificadores de chave podem, portanto, ser servidos ao mesmo tempo - um arquivo por identificador em uso em toda a propriedade do SIM - e uma SUCI cujo identificador não possui chave carregada falha com `unknown_home_network_key_id`.

Rotacionando uma chave da Rede Doméstica (zero-downtime). Como o desvelamento é acionado por identificador e múltiplos identificadores coexistem, uma chave é trocada introduzindo um *novo* identificador ao lado do antigo, em vez de sobrescrever um arquivo no local:



Sobrescrever o arquivo para um identificador ainda em uso em SIMs ativos quebra o desvelamento para esses SIMs (eles apresentam a chave pública antiga, a verificação de MAC falha). Sempre retire um identificador somente após cada SIM que o utiliza ter sido re-provisionado para um novo.

Cuidado operacional.

- Trate o conteúdo de `hnet_key_dir` como material secreto altamente sensível - a compromissão de uma chave privada da Rede Doméstica permite que qualquer parte desvele SUCIs para aquele identificador de chave. Restrinja as permissões de diretório e arquivo apenas à conta de serviço do OmniUDM.
- A chave privada aqui deve ser a contraparte da chave pública provisionada nos SIMs para o mesmo identificador de chave. Uma incompatibilidade faz com que a verificação de MAC falhe e a autenticação seja rejeitada.

- As chaves são armazenadas em cache após o primeiro carregamento. Após adicionar, substituir ou remover arquivos de chave, o cache deve ser atualizado (veja a ação de gerenciamento de limpeza de cache OAM, `POST /api/oam/cache/clear`) ou o processo reiniciado para que a alteração tenha efeito.

API de Gerenciamento / OAM

Uma API de gerenciamento/OAM HTTPS é servida (porta padrão `8443`, TLS habilitado) para inspeção e controle operacional. É separada do listener SBI.

Método	Caminho	Propósito
GET	`/api/status/api`	Status da API de gerenciamento
GET	`/api/status/license`	Status da licença
GET	`/api/status/nrf`	Status de registro do NRF
GET	`/api/status/nf`	Resumo do status NF
GET	`/api/ue`	Listar contextos UE ativos em memória
GET	`/api/ue/{supi}`	Inspecionar um único contexto UE por SUPI
GET	`/api/statistics`	Estatísticas de serviço, incluindo uma contagem ao vivo de contextos UE ativos em memória
GET	`/api/oam/config`	Ler configuração efetiva atual
PATCH	`/api/oam/config/{id}`	Atualizar um valor de configuração (PUT também é aceito)
DELETE	`/api/oam/ue_cache/{id}`	Evitar um contexto UE do cache em memória
POST	`/api/oam/ue/sdm-notify`	Acionar manualmente uma notificação de mudança de dados SDM para um SUPI. POSTs um `ModificationNotification` para cada `callbackReference` de assinatura SDM ativa (por exemplo, após mudanças de provisionamento fora de banda)

Método	Caminho	Propósito
DELETE	`/api/oam/ue/amf-registration/{id}`	Forçar a desregistrar o AMF para um UE. Limpa o registro de AMF 3GPP armazenado *e* notifica o AMF que está servindo em seu `deregCallbackUri` para que libere o UE (razão de desregistro `SUBSCRIPTION_WITHDRAWN`); retorna 404 se o UE não tiver registro AMF
POST	`/api/oam/cache/clear`	Limpar caches (incluindo o cache de chaves da Rede Doméstica)
POST	`/api/oam/log_level`	Alterar o nível de log em tempo de execução (`{"level": ...}`)
GET	`/api/oam/resources`	Inventário de recursos
GET	`/api/oam/diagnostics`	Instantâneo de diagnósticos
POST	`/api/oam/nrf/reregister`	Forçar re-registro com o NRF

Registro

Configuração	Valor	Descrição
`format`	`{OmniLogger.JsonFormatter, :format}`	As linhas de log são emitidas como JSON estruturado via o formatador compartilhado OmniLogger, adequado para ingestão por um agregador de logs
`metadata`	`:all`	Todos os metadados de log (incluindo contexto de sessão por solicitação, como SUPI e procedimento) estão incluídos em cada linha

O nível de log em tempo de execução pode ser alterado dinamicamente via a API de Gerenciamento/OAM (`POST /api/oam/log_level`). Materiais de chave (CK/IK/KAUSF/SQN⊕AK) nunca são registrados; apenas contextos não secretos, como o nome da rede servidora e o SUPI desvelado, aparecem nos logs.

← Visão Geral

Métricas do OmniUDM

As métricas são expostas no formato Prometheus na `prometheus_metrics_port` (padrão `9568`) em `/metrics`.

Métricas do Serviço UDM

Métrica	Tipo	Rótulos	D
`omni_udm_auth_requests_count`	contador	`supi`, `result`	Cor soli de aut por (`su` `fai`
`omni_udm_auth_vector_generations_total`	contador	`result`	Ten ger vet aut por
`omni_udm_auth_result_count`	contador	`supi`, `result`	Eve res aut reg via `eve` (`re` val `suc` rep
`omni_udm_uecm_registrations_total`	contador	`access_type`	Reg UE em ace Tan car reg 3GI

Métrica	Tipo	Rótulos	D

			não
			incr
			est
			cor
			ma
			`acc`
			(`3g`
			`nor`

`omni_udm_sdm_queries_total`	contador	`data_type`	Cor
			dac
			por
			dac
			`dat`
			`dat`
			`sel`
			`nss`
			`con`
			`smf`
			`dat`

`omni_udm_active_ue_contexts_count`	gauge	nenhum	Núr cor ativ me Em cria exc cor

Métrica	Tipo	Rótulos	D
`omni_udm_nrf_registration_status`	gauge	`nf_type`	Sta reg (`1` reg = r reg

Consultas Úteis

```
# Taxa de falha de autenticação em 5 minutos
sum(rate(omni_udm_auth_requests_count{result="failure"}[5m]))
  / sum(rate(omni_udm_auth_requests_count[5m]))

# Taxa de consulta SDM por tipo de dado
sum by (data_type) (rate(omni_udm_sdm_queries_total[5m]))

# Taxa de registro UECM dividida por tipo de acesso (3gpp /
non_3gpp)
sum by (access_type) (rate(omni_udm_uecm_registrations_total[5m]))

# Taxa de falha na geração de vetor de autenticação
sum(rate(omni_udm_auth_vector_generations_total{result="failure"}
[5m]))
  / sum(rate(omni_udm_auth_vector_generations_total[5m]))

# Alerta se o UDM não estiver registrado com o NRF
min(omni_udm_nrf_registration_status) < 1
```

`omni_udm_active_ue_contexts_count` rastreia o número ativo de contextos UE em memória; é atualizado em cada criação e exclusão de contexto UE. Os endpoints OAM `/api/statistics` e `/api/status/nf` relatam a mesma contagem sob demanda.

Métricas da VM BEAM

Métrica	Tipo	Descrição
`beam_memory_total`	gauge	Memória total do BEAM (bytes)
`beam_memory_processes`	gauge	Memória alocada para processos Erlang
`beam_memory_processes_used`	gauge	Memória realmente utilizada pelos processos
`beam_memory_system`	gauge	Memória do sistema (ETS, átomos, código)
`beam_memory_atom`	gauge	Memória total de átomos
`beam_memory_atom_used`	gauge	Memória de átomos utilizada
`beam_memory_binary`	gauge	Memória binária
`beam_memory_code`	gauge	Memória de código
`beam_memory_ets`	gauge	Memória da tabela ETS
`beam_processes_count`	gauge	Número de processos Erlang
`beam_ports_count`	gauge	Número de portas Erlang
`beam_atom_count`	gauge	Número de átomos
`beam_vm_uptime`	gauge	Tempo de atividade da VM (segundos)

← Visão Geral

Operações do OmniUDM

OmniUDM é a função de Gerenciamento de Dados Unificado (UDM) do núcleo 5G do Omnitouch. Ele apresenta dados de identidade do assinante, assinatura e contexto para o restante do núcleo através dos serviços padrão Nudm, respaldando todos os dados autoritativos no Repositório de Dados Unificado (OmniUDR) via `nudr-dr`.

Papel e Referências de Especificação da 3GPP

Aspecto	Referência
Papel funcional do UDM no 5GC	TS 23.501
Serviços Nudm (SDM / UEAU / UECM)	TS 29.503
Nudr_DataRepository (consumido)	TS 29.504
Esquemas de ocultação e proteção de SUCI / SUPI	TS 33.501 §6.12 e Anexo C
Estrutura e codificação de SUCI / SUPI	TS 23.003 §2.2B
Hierarquia de autenticação e chave 5G-AKA	TS 33.501 §6.1.3, Anexo A
Codificação de identidade móvel (SUCI/MSIN BCD)	TS 24.501

Interfaces e Endpoints SBI

OmniUDM registra os seguintes serviços com o NRF e os disponibiliza no listener SBI (`sbi_scheme://sbi_addr:sbi_port`):

Serviço	Versão	Propósito
`nudm-sdm`	v2	Gerenciamento de dados do assinante
`nudm-ueau`	v1	Geração de vetor de autenticação do UE
`nudm-uecm`	v1	Gerenciamento de contexto do UE (registro AMF/SMF)

Todos os dados autoritativos de assinantes e contexto são lidos de / escritos para o OmniUDR via `nudr-dr/v2` (caminho base `.../nudr-dr/v2/subscription-data/{ueId}/...`). Todos os corpos de requisição/resposta são JSON. Erros são retornados como `application/problem+json` ProblemDetails.

Nudm_UEAuthentication (`nudm-ueau/v1`)

Método	Caminho	Descrição
POST	`/{supiOrSuci}/security-information/generate-auth-data`	Gera um vetor de autenticação 5G-AKA. Aceita um SUPI ou SUCI; de-oculta SUCI primeiro. Requer `servingNetworkName` e `ausfInstanceId`
POST	`/{supi}/auth-events`	Registra um evento de resultado de autenticação. Requer `nfInstanceId`, `success`, `timeStamp`, `authType`, `servingNetworkName`
PUT	`/{supi}/auth-events/{authEventId}`	Atualiza / remove um evento de autenticação (corpo AuthEvent, `authRemovalInd=true` limpa)

Nudm_UECM (`nudm-uecm/v1`)

Método	Caminho	Descrição
PUT	`/{supi}/registrations/amf-3gpp-access`	AMF registra acesso 3GPP para um UE
PATCH	`/{supi}/registrations/amf-3gpp-access`	Atualiza parâmetros do registro 3GPP AMF armazenado (merge-patch)
GET	`/{supi}/registrations/amf-3gpp-access`	Recupera o registro 3GPP AMF armazenado
PUT	`/{supi}/registrations/amf-non-3gpp-access`	AMF registra acesso não-3GPP para um UE
GET	`/{supi}/registrations/amf-non-3gpp-access`	Recupera o registro não-3GPP AMF armazenado
PUT	`/{supi}/registrations/smf-registrations/{pduSessionId}`	SMF registra um contexto de sessão PDU
DELETE	`/{supi}/registrations/smf-registrations/{pduSessionId}`	SMF desregistra um contexto de sessão PDU

Nudm_SDM (`nudm-sdm/v2`)

Método	Caminho	Descrição
GET	`/{supi}`	GetDataSets - recupera múltiplos conjuntos de dados selecionados pelo parâmetro de consulta obrigatório `dataset-names` (separados por vírgula `AM`, `SM`, `SMF_SEL`, `UEC_SMF`)
GET	`/{supi}/am-data`	Dados de assinatura de acesso e mobilidade (consulta opcional `plmn-id`)
GET	`/{supi}/sm-data`	Dados de assinatura de gerenciamento de sessão (consulta opcional `plmn-id`)
GET	`/{supi}/smf-select-data`	Dados de assinatura de seleção de SMF (consulta opcional `plmn-id`)
GET	`/{supi}/nssai`	Dados de assinatura de seleção de fatias de rede
GET	`/{supi}/ue-context-in-smf-data`	Contexto do UE no SMF (sessões PDU ativas): veja a nota abaixo
POST	`/{supi}/sdm-subscriptions`	Inscreve-se em notificações de alteração de dados. Requer `nfInstanceId`,

Método	Caminho	Descrição
		`callbackReference`, `monitoredResourceUris`
GET	`/{supi}/sdm-subscriptions/{subscriptionId}`	Recupera uma assinatura SDM armazenada
PATCH	`/{supi}/sdm-subscriptions/{subscriptionId}`	Modifica uma assinatura SDM (merge-patch)
DELETE	`/{supi}/sdm-subscriptions/{subscriptionId}`	Remove uma assinatura SDM

Ocultação de SUCI / SUPI

Para proteger a identidade permanente do assinante no ar, o UE envia um **SUCI** (Identificador de Assinatura Ocultado) em vez do **SUPI** (Identificador Permanente de Assinatura, ou seja, `imsi-<IMSI>`). Quando o AUSF pede ao OmniUDM para gerar um vetor de autenticação para um SUCI, o OmniUDM **de-oculta o SUCI para o SUPI** antes que ele possa procurar os dados do assinante no UDR.

OmniUDM suporta os três esquemas de proteção definidos na TS 33.501 §6.12 / Anexo C:

Esquema	Nome	Curva / Cripto	Requer chave HNET?
0	Esquema nulo	Nenhum - MSIN está em texto claro	Não
1	Perfil ECIES A	Curve25519 (X25519) + AES-128-CTR + HMAC-SHA-256	Sim
2	Perfil ECIES B	secp256r1 / P-256 + AES-128-CTR + HMAC-SHA-256	Sim

Para o **esquema nulo**, o MSIN é transportado em texto claro; ele é **decodificado em BCD** a partir da saída do esquema para reconstruir o IMSI (conforme TS 24.501 - os dígitos do MSIN são codificados em BCD, não em hex). Para **Perfil A e Perfil B**, a de-ocultação realiza um ECDH efêmero-estático contra a **chave privada da Rede Doméstica** do operador, deriva as chaves de criptografia/MAC (ANSI-X9.63-KDF com SHA-256), verifica o MAC, descriptografa o MSIN oculto e o decodifica em BCD para reconstruir o IMSI.

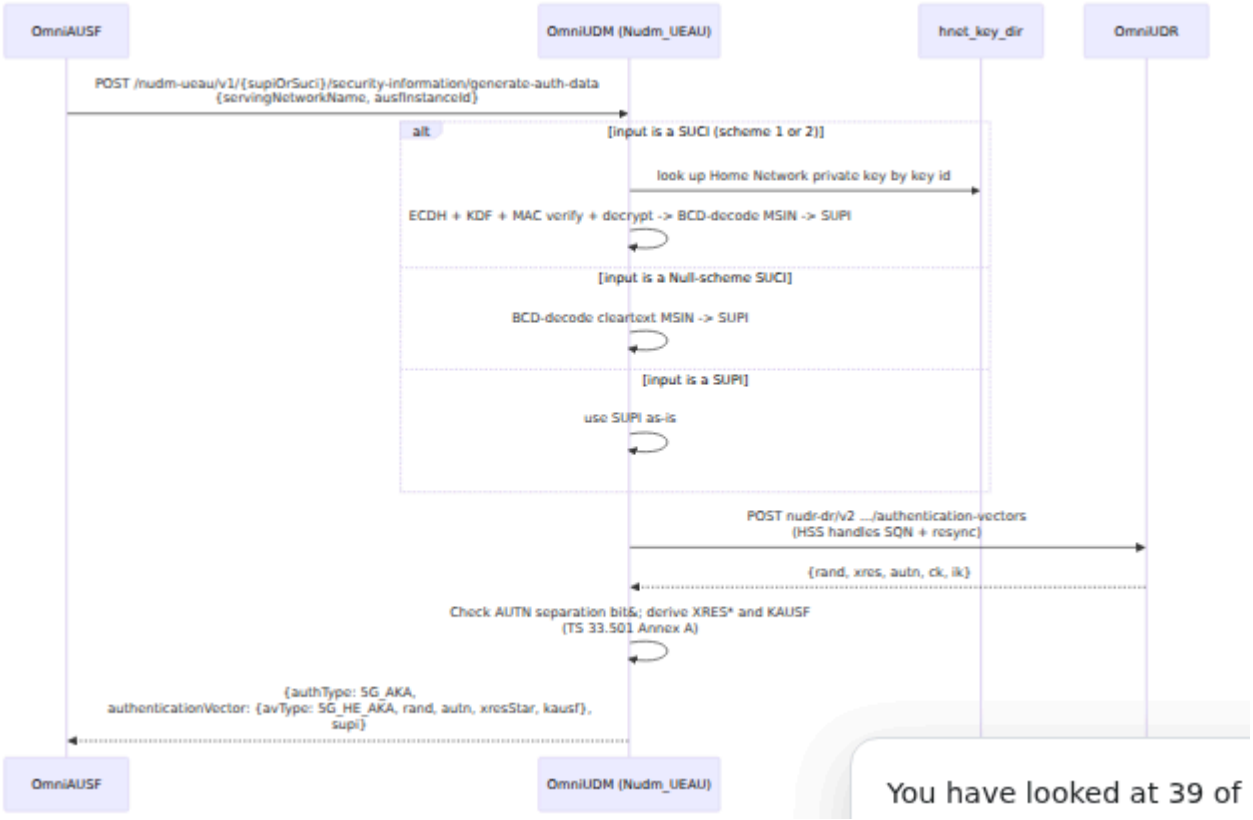
As chaves privadas da Rede Doméstica para os Perfis A/B são carregadas do diretório `hnet_key_dir`. A nomeação de arquivos, codificações e os requisitos de segurança e atualização de cache estão documentados em Configuração → Chaves da Rede Doméstica (`hnet_key_dir`).

Procedimentos Chave

De-ocultação de SUCI e Geração de Vetor de Autenticação

O AUSF solicita um vetor de autenticação; o OmniUDM resolve a identidade (de-ocultando um SUCI se necessário), solicita um vetor do HSS via UDR, e então deriva as chaves 5G a partir dele. **Ki/OPc, gerenciamento de SQN e qualquer re-sincronização permanecem no HSS atrás do UDR** - o OmniUDM verifica se o AUTN do HSS carrega o bit de separação 5G (bit 0 do AMF, TS 33.501 §6.1.3.2, retirado do AMF provisionado do assinante) e deriva

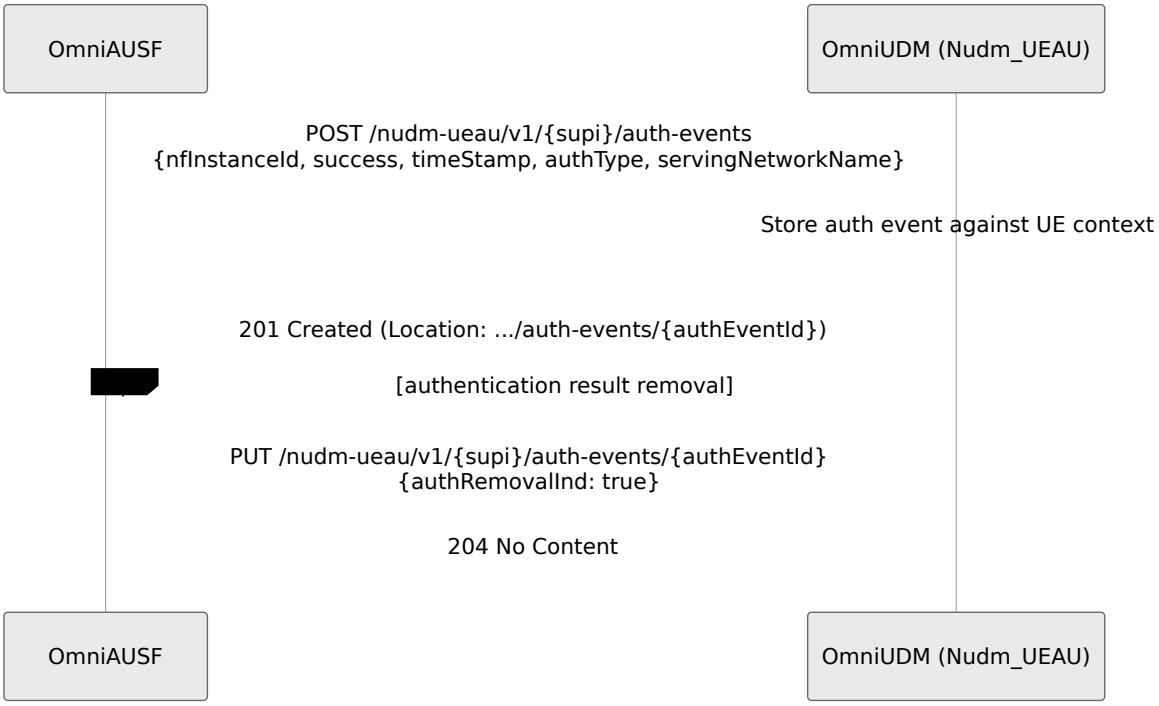
`XRES*` e `KAUSF` a partir do `xres`, `ck` e `ik` retornados. Um `resynchronizationInfo` fornecido pelo AUSF é encaminhado ao HSS via UDR.



Falhas de de-ocultação retornam HTTP 403 (`Falha na De-ocultação de SUCI`), um esquema de proteção não suportado retorna HTTP 501, um assinante ausente ou desativado retorna HTTP 404 (`USUÁRIO_NÃO_ENCONTRADO`), e um vetor HSS inválido (incluindo um AUTN sem o bit de separação) ou uma falha na geração do vetor HSS retorna HTTP 500.

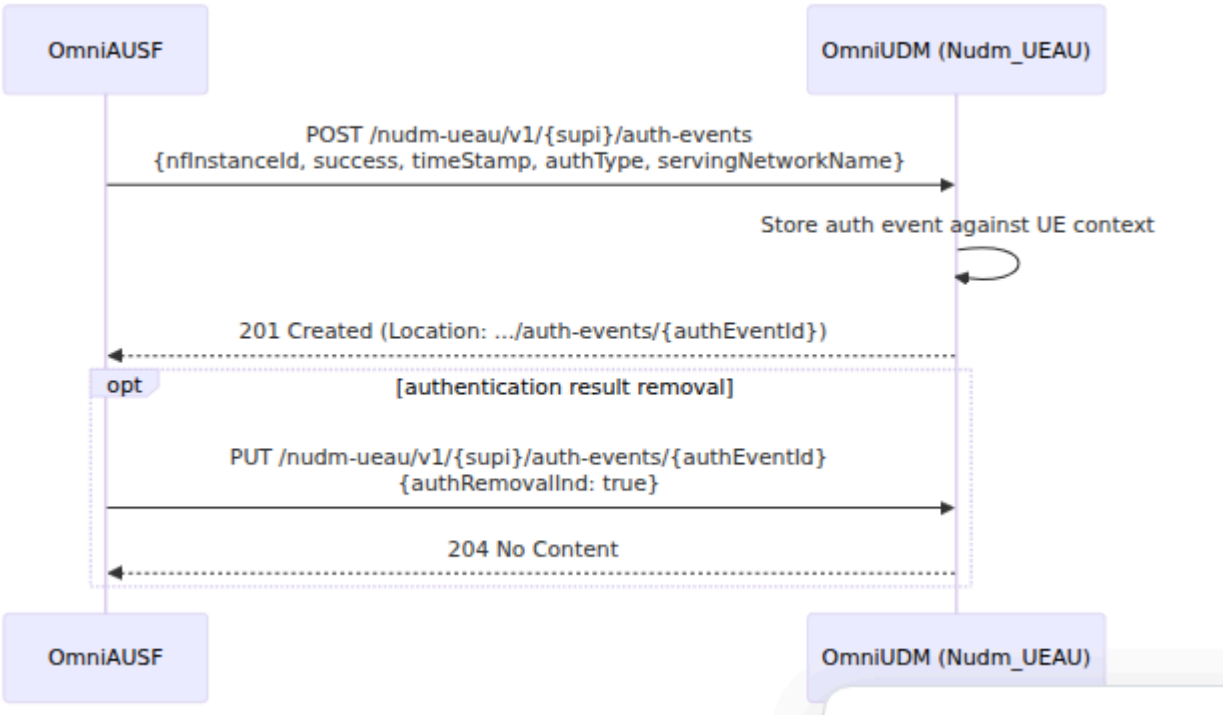
Registro de Evento de Resultado de Autenticação

Após a conclusão da autenticação pelo AUSF, ele registra o resultado. O OmniUDM armazena o evento contra o contexto do UE e retorna uma `Location` para o evento criado; o AUSF pode posteriormente `PUT` um `AuthEvent` com `authRemovalInd=true` para limpá-lo (204 Sem Conteúdo).



Busca de Dados do Assinante SDM (`sm-data`)

Após o registro, o AMF/SMF busca dados de assinatura. O OmniUDM faz proxy da solicitação para o UDR, adicionando o PLMN de serviço, e retorna a resposta do UDR. Para `sm-data`, um único objeto UDR é envolto em um array conforme TS 29.503 `SmSubsData`.

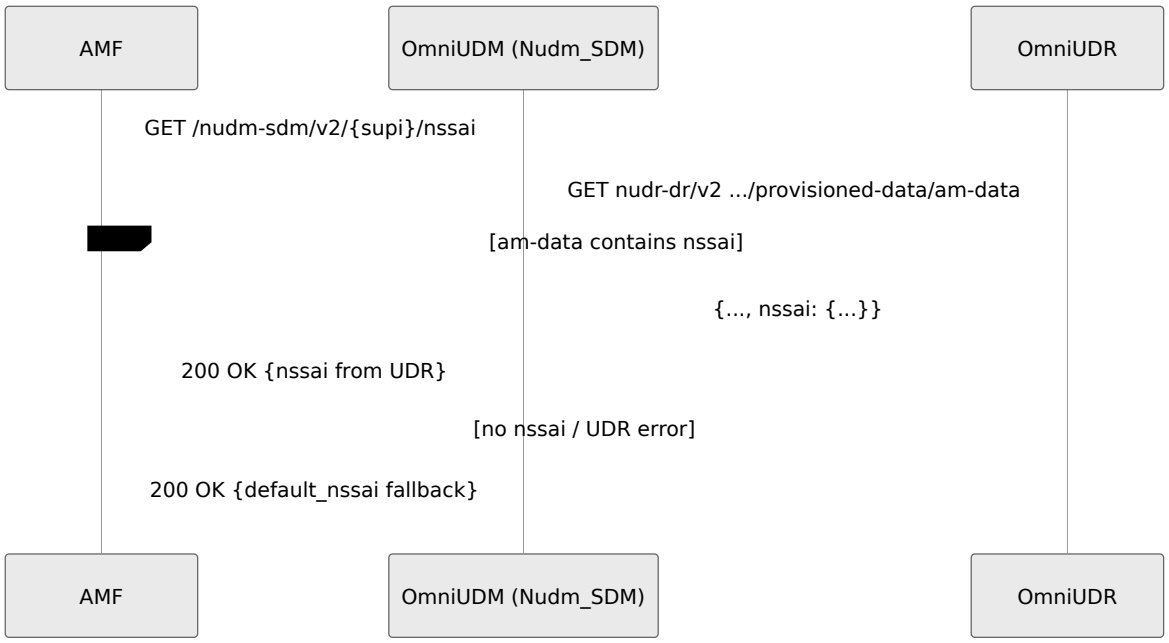


Os fluxos `am-data`, `smf-select-data` e GetDataSets (`GET /{supi}`) seguem o mesmo padrão de proxy contra o UDR. GetDataSets monta apenas os conjuntos de dados solicitados que pode resolver, ignorando quaisquer que apresentem erro. `nssai` é derivado da resposta `am-data` do UDR (veja abaixo).

> **O contexto do UE no SMF retorna o contexto ativo da sessão PDU.** `GET /{supi}/ue-context-in-smf-data` e o conjunto de dados `UEC_SMF` em GetDataSets retornam os contextos ativos de sessão PDU do UE conforme relatado pelo SMF via o procedimento de registro SMF Nudm_UECM. Cada registro SMF armazenado é projetado no mapa `pduSessions` (chaveado pelo ID da sessão PDU) conforme TS 29.503. O mapa está vazio (`{"pduSessions": {}}`) apenas quando o UE realmente não possui registros SMF.

Recuperação de NSSAI com Retorno Padrão

`GET /{supi}/nssai` busca `am-data` do UDR e extrai o `nssai` embutido. Se o UDR retornar `am-data` sem um `nssai`, ou se a busca no UDR falhar, o OmniUDM retorna o retorno configurável `default_nssai` para que o AMF sempre receba uma lista de fatias.



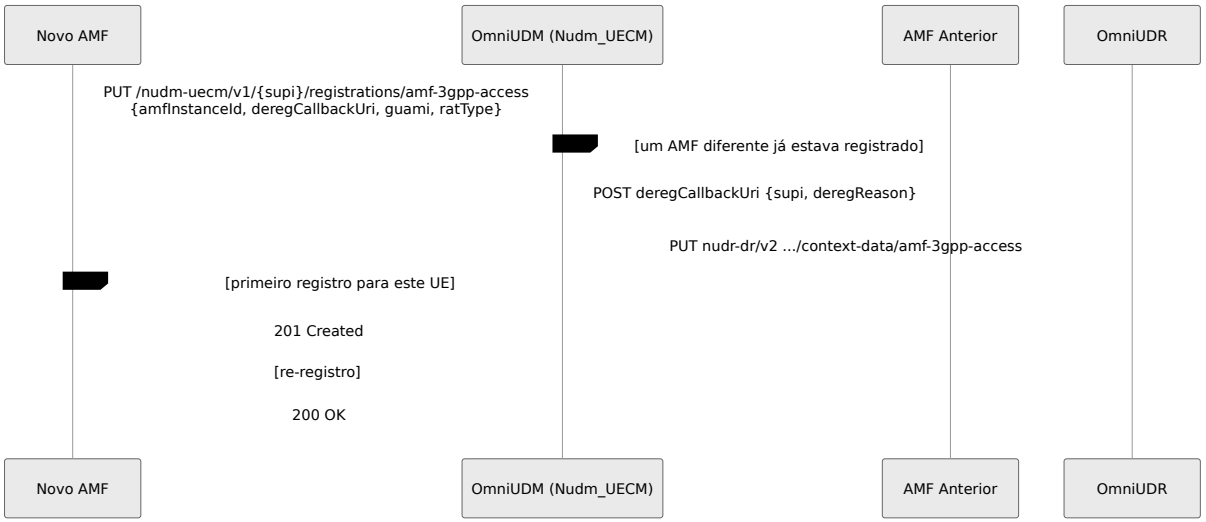
O retorno `default_nssai` deve corresponder às fatias que a rede realmente serve - um assinante oferecido uma fatia que o AMF não está configurado para servir é rejeitado com a causa 5GMM #62. Veja Configuração → Chaves de Ajuste Opcionais.

Notificações de Alteração do SDM

Assinantes com uma assinatura SDM ativa são notificados de alterações subsequentes: o OmniUDM POSTa uma `ModificationNotification` para cada `callbackReference` da assinatura (assíncrono) quando os recursos monitorados mudam. Uma notificação também pode ser acionada manualmente para um SUPI via a API de Gerenciamento/OAM.

Registro AMF UECM e Mobilidade

Quando um UE se registra, o AMF se registra para aquele UE no OmniUDM. Se um AMF diferente foi registrado anteriormente, o OmniUDM notifica o `deregCallbackUri` do antigo AMF para que ele possa liberar o UE, e então persiste o novo registro no UDR.



Os IEs obrigatórios são validados conforme TS 29.503: o registro 3GPP AMF requer `amfInstanceId`, `deregCallbackUri`, `guami`, `ratType`; o registro SMF requer `smfInstanceId`, `pduSessionId`, `singleNssai`, `plmnId`, e o corpo `pduSessionId` deve corresponder ao caminho (uma discrepância retorna `MANDATORY_IE_INCORRECT`). IEs ausentes retornam HTTP 400.

← Visão Geral

Solução de Problemas do OmniUDM

A autenticação falha com 404 Usuário Não Encontrado

OmniUDM resolveu a identidade, mas o UDR não retornou nenhuma assinatura de autenticação. Confirme se `udr_uri` é acessível, se o OmniUDR pode acessar o banco de dados de assinantes e se o IMSI está provisionado. Para uma entrada SUCI, confirme se a desocultação produziu o IMSI esperado (verifique a linha de log "De-concealed SUCI to ...").

A autenticação falha com 403 Falha na Desocultação do SUCI

OmniUDM não conseguiu desocultar o SUCI. O campo de detalhe nomeia a causa específica:

- `unknown_home_network_key_id` - nenhuma chave privada para o identificador de chave do SUCI está carregada. Confirme se um arquivo de chave para aquele `home_network_public_key_id` existe em `hnet_key_dir` e segue uma convenção de nomenclatura reconhecida, e que o cache de chaves foi atualizado desde que o arquivo foi adicionado.
- `ecies_mac_verification_failed` - a chave privada carregada não corresponde à chave pública no SIM (chave errada ou trocada), ou o SUCI está corrompido. Verifique o par de chaves provisionado nos SIMs em relação à chave privada implantada para aquele identificador.
- `ecies_profile_a_scheme_output_too_short` / `ecies_profile_b_scheme_output_too_short` - a saída do esquema SUCI está truncada / malformada.

- `invalid_suci_format` - a string SUCI não está bem formada.

A autenticação falha com 501 Esquema de Proteção Não Suportado

O SUCI usou um esquema de proteção diferente de 0 (Nulo), 1 (Perfil A) ou 2 (Perfil B). Apenas esses três são suportados.

A autenticação falha com 500 Erro Interno

Os campos do vetor HSS (`rand`/`xres`/`ck`/`ik`/`autn`) estavam ausentes, não eram hexadecimais válidos ou tinham o comprimento errado; o AUTN não tinha o bit de separação 5G; ou a geração do vetor HSS falhou. A linha de log `Invalid auth key material ... separation_bit_not_set` significa que o AMF provisionado do assinante no conjunto de chaves HSS não tem seu bit superior definido (use `8000`). Caso contrário, verifique o conjunto de chaves do assinante no HSS e se o OmniUDR pode acessar o HSS.

`am-data` / `sm-data` / `smf-select-data` retorna 404 Não Encontrado

A busca do UDR falhou para aquele conjunto de dados. Confirme se os dados provisionados do assinante existem no UDR para o PLMN de atendimento, que `mcc`/`mnc` (ou o `plmn-id` da solicitação) correspondem à forma como os dados estão provisionados, e que `udr_uri` é acessível.

UE rejeitada pelo AMF com causa 5GMM #62 (nenhum slice

suportado)

O NSSAI servido ao AMF não inclui um slice que o AMF está configurado para servir. Se os `am-data` do UDR do assinante não contiverem `nssai`, o OmniUDM retorna o fallback `default_nssai` - defina `default_nssai` (ou provisione `nssai` no UDR) para corresponder aos slices reais da rede, incluindo quaisquer S-NSSAIs qualificados por SD.

Contexto da UE ou registro não presente após reinício

O estado de execução por UE (contexto de autenticação, registros AMF/SMF, assinaturas SDM) é mantido em um armazenamento de contexto em memória e é reconstruído por meio de re-registro; os dados autoritativos do assinante persistem no OmniUDR. Após um reinício, AMF/SMF se re-registram na próxima atualização de registro periódica, restabelecimento de sessão PDU ou ciclo de energia da UE. Se um registro específico estiver ausente antes do esperado, acione a atualização de registro do NF consumidor em vez de esperar pelo temporizador periódico.

Notificação de alteração de dados SDM não recebida

Confirme se o URI `callbackReference` da assinatura é acess��vel a partir do host OmniUDM. As notificações são enviadas de forma assíncrona; procure avisos "SDM notification to ... failed". Uma notificação pode ser acionada manualmente para um SUPI via a API de Gerenciamento/OAM (`POST /api/oam/ue/sdm-notify`).

Após alterar as chaves da Rede Doméstica, a desocultação ainda usa a chave antiga

As chaves da Rede Doméstica são armazenadas em cache na memória após o primeiro carregamento. Recarregue-as via a ação de gerenciamento OAM (`POST /api/oam/hnet/reload`) após adicionar ou substituir arquivos de chave em `hnet_key_dir`; a resposta informa o número de chaves carregadas e seus IDs de chave. Nenhum reinício de processo é necessário.

UDM não descobrível por outros NFs

Verifique `omni_udm_nrf_registration_status`. Se `0`, confirme se `nrf_uri` é acessível e se o heartbeat está sendo bem-sucedido; a API de gerenciamento OAM oferece uma ação de re-registro NRF (`POST /api/oam/nrf/reregister`).