← Resumen

Configuración de OmniAUSF

La configuración se lee desde la clave del entorno de la aplicación `:omniausf`, poblada desde `config/runtime.exs`. La mayoría de las claves de tiempo de ejecución están respaldadas por una variable de entorno del sistema operativo leída al inicio, como se muestra a continuación; las dos claves del recolector de contexto se leen directamente del entorno de la aplicación `:omniausf` y no están respaldadas por variables de entorno.

```
config :omniausf,
  udm_uri:                "http://127.0.0.12:7777",
  sbi_scheme:             "http",
  sbi_addr:               "127.0.0.19",
  sbi_port:               7777,
  nrf_uri:                "http://127.0.0.1:7777",
  mcc:                    "999",
  mnc:                    "70",
  prometheus_metrics_port: 9568,
  heartbeat_interval:     10_000

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

Referencia de Parámetros

Parámetro	Tipo	Requerido	Predeterminado
`udm_uri`	cadena	No	`"http://127.0.0.12:`
`sbi_scheme`	cadena	No	`"http"`
`sbi_addr`	cadena	No	`"127.0.0.19"`

Parámetro	Tipo	Requerido	Predeterminado
`sbi_port`	entero	No	`7777`
`nrf_uri`	cadena	No	`"http://127.0.0.1:7`
`mcc`	cadena	No	`"999"`
`mnc`	cadena	No	`"70"`
`prometheus_metrics_port`	entero	No	`9568`
`heartbeat_interval`	entero (ms)	No	`10000`

Parámetro	Tipo	Requerido	Predeterminado
`context_ttl_seconds`	entero (s)	No	`3600`
`context_reap_interval_ms`	entero (ms)	No	`60000`

> **Nota sobre el valor predeterminado de `nrf_uri`:** `config/runtime.exs` establece `NRF_URI` en `http://127.0.0.1:7777`. En un despliegue de bucle de retorno estándar, el NRF se alcanza en `http://127.0.0.10:7777`; establezca `NRF_URI` explícitamente para que coincida con la dirección de su NRF.

Registro

Los registros se emiten como JSON (`OmniLogger.JsonFormatter`) con todos los metadatos incluidos. Las líneas de registro en el camino de autenticación llevan contexto de sesión (`supi`, `auth_ctx_id`, `procedure`) para que un único intercambio de autenticación pueda ser rastreado de extremo a extremo por `auth_ctx_id` o SUPI/SUCI. El nivel de registro se puede cambiar en tiempo de ejecución a través de la API de Gestión (ver más abajo).

API de Gestión (Puerto 8443)

Además del oyente SBI y el puerto de métricas, OmniAUSF expone una API de gestión/OAM HTTPS en el puerto `8443` (TLS, nombre del producto `OmniAUSF`). Se utiliza para la inspección operativa y el control en tiempo de ejecución, y no forma parte del SBI de 3GPP. Puntos finales clave:

Método	Ruta	Propósito
GET	`/api/status/api`	Estado del servicio API
GET	`/api/status/nrf`	Estado del registro NRF
GET	`/api/status/nf`	Resumen de NF (tipo `AUSF`, servicio `nausf-auth`, conteo de contexto UE activo)
GET	`/api/ausf_sessions`	Listar sesiones de autenticación activas (id de contexto, SUCI/SUPI, nombre de la red que sirve, tipo/resultados de autenticación)
DELETE	`/api/ausf_sessions/{id}`	Limpiar una sesión de autenticación específica por id de contexto
POST	`/api/ausf_sessions`	Limpiar todas las sesiones de autenticación activas
GET	`/api/statistics`	Contadores de sesiones de autenticación (total / éxito / fallo / pendiente)
GET	`/api/oam/config`	Ver configuración de tiempo de ejecución efectiva (`sbi_addr`, `sbi_port`, `nrf_uri`, `udm_uri`, capacidad/prioridad de NF, nivel de registro)
PATCH	`/api/oam/config/{id}`	Actualizar configuración de tiempo de ejecución: `udm_uri`, `nrf_uri`, `nf_capacity`, `nf_priority`, `log_level`

Método	Ruta	Propósito
POST	`/api/oam/log_level`	Cambiar el nivel de registro en tiempo de ejecución (`emergency`…`debug`)
GET	`/api/oam/resources`	Uso de recursos de la VM BEAM (memoria, conteos de procesos/puertos, versión de OTP/BEAM)
GET	`/api/oam/diagnostics`	Diagnósticos de la VM BEAM (cola de ejecución, I/O, GC, nodos conectados)
POST	`/api/oam/nrf/reregister`	Forzar la re-registro con el NRF

Los cambios en tiempo de ejecución realizados a través de `PATCH /api/oam/config` y `POST /api/oam/log_level` se aplican solo al nodo en ejecución y no se persisten; vuelven a los valores configurados al reiniciar.

Configuraciones solo OAM

Dos valores se exponen a través de `GET`/`PATCH /api/oam/config` pero no son parámetros de inicio en la tabla anterior; viven en el perfil NRF en lugar de la configuración central `:omniausf`:

Campo	Tipo	Significado
`nf_capacity`	entero (0-65535)	Capacidad relativa anunciada en el perfil NRF del AUSF, utilizada por los consumidores como un peso de balanceo de carga cuando varias instancias de AUSF están registradas. Un `PATCH` vuelve a anunciar el nuevo valor al NRF
`nf_priority`	entero (0-65535)	Prioridad de selección relativa anunciada en el perfil NRF; se prefiere un número menor. Un `PATCH` vuelve a anunciar el nuevo valor al NRF

`PATCH /api/oam/config` valida estos antes de aplicarlos: `nf_capacity`/`nf_priority` deben ser enteros (se acepta y analiza una cadena decimal), y `log_level` debe ser un nivel conocido. Un valor no válido rechaza toda la solicitud con `400` sin aplicar nada.

Inspeccionando sesiones de autenticación

`GET /api/ausf_sessions` devuelve una entrada por cada contexto de autenticación de UE en memoria, con estos campos:

Campo	Significado
`ctx_id`	El `authCtxId` (UUID); páselo a `DELETE /api/ausf_sessions/{ctx_id}` para limpiar una sesión
`suci`	El `supiOrSuci` que el AMF presentó al inicio
`supi`	El SUPI plano resuelto por el UDM (nulo hasta que llegue la respuesta del vector)
`serving_network_name`	El nombre de la SN al que están vinculadas las claves de la sesión
`auth_type`	`5G_AKA` una vez que se almacena el vector
`auth_result`	Resultado final, o nulo mientras la sesión aún está en vuelo (pendiente)
`has_kausf`	Si se mantiene `K_AUSF` para esta sesión - indica que se obtuvo el vector y se derivaron las claves. El material de clave en s`[illegible]` nunca se expone

Debido a que los contextos se eliminan en el momento en que se confirma un asentamiento, las entradas en esta lista son normalmente de corta duración; una sesión que persiste es una que el AMF inició pero nunca confirmó o desregistró.

← Resumen

Métricas de OmniAUSF

Las métricas se exponen en `http://{host}:{prometheus_metrics_port}/metrics` (puerto por defecto `9568`).

Métricas de AUSF

Métrica	Tipo	Etiquetas	Descripc
`omni_ausf_auth_count`	contador	`result`	Operacione autenticaci de UE. `resu` es uno de `initiated` (vector obtenido, desafío devuelto), `success` (R coincide co XRES*), `failure` (e de UDM, ve malformado discrepanci RES*), `resy` (se detectó comunicó u re-sincronizaci de AUTS/SC UDM), o `resync_inv` (se descart blob de re-sincronizaci malformado
`omni_ausf_nrf_registration_status`	medidor	`nf_type`	Estado de registro de (1 = registr

Métrica	Tipo	Etiquetas	Descripc
			0 = no registrado)
`omni_ausf_active_contexts_count`	medidor	-	Número de contextos d autenticaci de UE activ Medidor en impulsado p el evento d telemetría `[:omniausf :ue_contex :active]`, emitido en creación y eliminación contexto y cada barrid ContextRea por lo que rastrea el conteo actu de contexto vuelo

> **`omni_ausf_active_contexts_count` es un medidor en vivo.** Está impulsado por el evento de telemetría `[:omniausf, :ue_contexts, :active]`, que la tienda de contexto de AUSF emite cada vez que cambia el conjunto de contextos activos: en la creación de contexto, en la eliminación (confirmación, desregistro o eliminación de resultado de autenticación), y en cada barrido de ContextReaper que recoge contextos inactivos. Por lo tanto, el medidor refleja el conteo actual de contextos de autenticación activos y es seguro para alertar y mostrar en el panel. La API de OAM sigue disponible para verificaciones cruzadas: `GET`

> `/api/status/nf` (incluye un conteo de contexto de UE activo) o `GET /api/statistics` (contadores de sesión total / éxito / fallo / pendiente).

> `omni_ausf_auth_count` cubre el flujo de autenticación principal más el camino de re-sincronización. Las etiquetas `resync` y `resync_invalid` cuentan los intentos de re-sincronización de SQN detectados en la solicitud de iniciación (validados y comunicados, y malformados y descartados, respectivamente). Los caminos de desregistro y eliminación de resultado de autenticación emiten sus propios eventos de telemetría, pero actualmente no están adjuntos a ninguna métrica de Prometheus exportada.

> **Telemetría interna (no es una métrica de Prometheus).** El recolector de contexto emite un evento de telemetría `[:omniausf, :ausf, :context_reap]` que lleva el conteo de contextos recogidos por barrido. Esta es telemetría interna para operadores que conectan sus propios controladores; no está registrada como una métrica de Prometheus exportada.

Ejemplo de PromQL

- Tasa de éxito de autenticación en 5 minutos: `sum(rate(omni_ausf_auth_count{result="success"}[5m])) / sum(rate(omni_ausf_auth_count{result="initiated"}[5m]))`
- Fallos de autenticación por segundo: `sum(rate(omni_ausf_auth_count{result="failure"}[5m]))`
- AUSF actualmente registrado con el NRF: `omni_ausf_nrf_registration_status == 1`
- Intentos de re-sincronización de SQN por segundo (válidos vs malformados): `sum(rate(omni_ausf_auth_count{result="resync"}[5m]))` y `sum(rate(omni_ausf_auth_count{result="resync_invalid"}[5m]))`
- Contextos de autenticación activos en vuelo: `omni_ausf_active_contexts_count`

Para una verificación cruzada independiente del conteo de contextos en vuelo, la API de OAM (`GET /api/status/nf` o `GET /api/statistics`) informa la misma cifra.

Métricas de BEAM VM

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

← Resumen

Operaciones de OmniAUSF

OmniAUSF implementa la Función del Servidor de Autenticación (AUSF) del núcleo 5G, ejecutando la autenticación primaria 5G-AKA entre el AMF (N12) y el UDM (N13). Solicita un vector de autenticación del UDM, transmite el desafío RAND/AUTN (y HXRES*) al AMF, verifica la respuesta del UE, deriva la clave de anclaje K_SEAF y reporta el resultado de la autenticación de vuelta al UDM.

Rol y Referencias de Especificación de 3GPP

Especificación	Relevancia
TS 23.501	Arquitectura del sistema - rol de AUSF en el sistema 5G
TS 33.501	Arquitectura de seguridad - autenticación 5G-AKA (6.1.3), jerarquía de claves K_AUSF/K_SEAF, HXRES*/HRES* (Anexo A.5), derivación de K_SEAF (Anexo A.6)
TS 29.509	API del servicio Nausf_UEAuthentication (HTTP SBI)
TS 29.503	API del servicio Nudm_UEAuthentication - generación de vector de autenticación e informes de resultados de autenticación
TS 29.510	APIs de servicio Nnrf_NFManagement y Nnrf_NFDiscovery - registro de NF/pulso y descubrimiento de UDM
TS 24.501	NAS - formato del nombre de la red de servicio (`5G:mnc<MNC>.mcc<MCC>.3gppnetwork.org`)

Interfaces

Interfaz	Par	Dirección	Transporte	Propósi
N12 (Nausf_UEAuthentication)	AMF	Entrante	HTTP SBI	Inicio de autenticación confirmación AKA
N13 (Nudm_UEAuthentication)	UDM	Saliente	HTTP SBI	Generación d vector de autenticación informes de resultados de autenticación
Nnrf	NRF	Saliente	HTTP SBI	Registro de N pulso y descubrimien NF par (UDM (`Nnrf_NFDis` TS 29.510)

Puntos finales de SBI

Todos los puntos finales de SBI se sirven bajo la URL base `{sbi_scheme}://{sbi_addr}:{sbi_port}` y utilizan `Content-Type: application/json` en la solicitud. La respuesta de creación exitosa utiliza `application/3gppHal+json` según TS 29.509.

Método	Ruta	Servicio	
POST	`/nausf-auth/v1/ue-authentications`	Nausf_UEAuthentication	I U R s s
PUT	`/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation`	Nausf_UEAuthentication	C A
DELETE	`/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation`	Nausf_UEAuthentication	E r d A
POST	`/nausf-auth/v1/ue-authentications/deregister`	Nausf_UEAuthentication	E c s u s

Resumen de Solicitud / Respuesta

Iniciar autenticación UE - el cuerpo de la solicitud lleva `supiOrSuci` (SUPI o SUCI; también se acepta el campo `suci`) y `servingNetworkName`; un objeto opcional `resynchronizationInfo` se pasa al UDM para la resincronización de SQN. El AUSF solicita un vector 5G-AKA del UDM, luego devuelve `201 Created` con un cuerpo `UEAuthenticationCtx`:

- `authType: "5G_AKA"`
- `5gAuthData`: `{ rand, autn, hxresStar }` (cadenas hexadecimales)
- `_links.5g-aka.href`: la URL de confirmación absoluta para este contexto
- `servingNetworkName`

Un encabezado `Location` apunta al recurso de contexto creado (`.../ue-authentications/{authCtxId}`).

Confirmación 5G-AKA - el cuerpo de la solicitud lleva `resStar` (hex). El AUSF compara el `resStar` decodificado con el `XRES*` almacenado (TS 33.501 6.1.3.2; el AMF es responsable de la verificación del hash HRES*/HXRES*). Registra el resultado en el UDM (`auth-events`) y siempre devuelve `200 OK` con un `ConfirmationDataResponse`:

- En caso de éxito: `{ authResult: "AUTHENTICATION_SUCCESS", supi, kseaf }` (K_SEAF como cadena hexadecimal).
- En caso de fallo: `{ authResult: "AUTHENTICATION_FAILURE", supi }`. Tenga en cuenta que un fallo de autenticación se señala en el cuerpo a través de `authResult`, **no** por un estado de error HTTP.

El contexto de autenticación se elimina después de la confirmación independientemente del resultado.

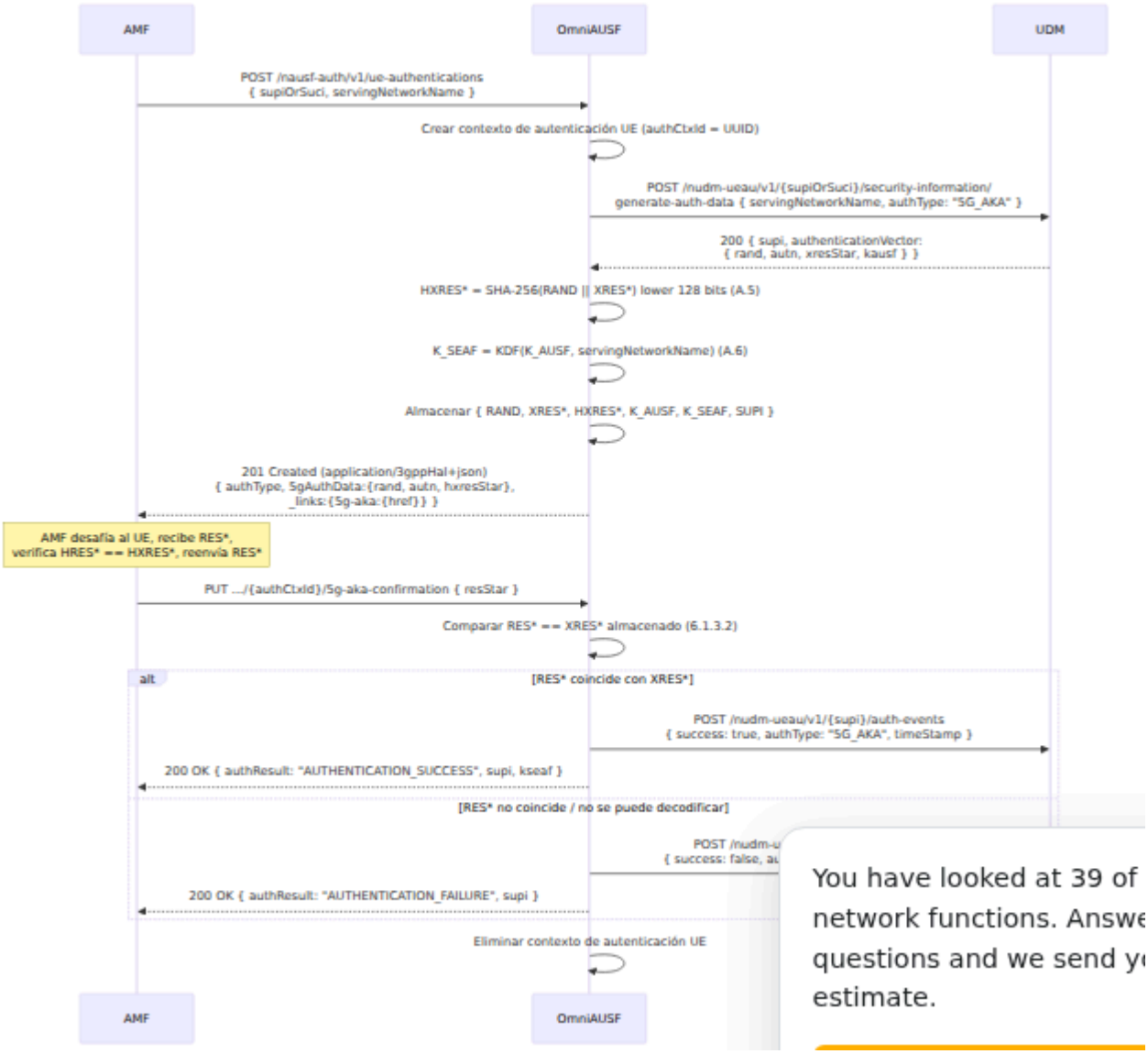
Respuestas de Error

Los errores se devuelven como cuerpos de `ProblemDetails`.

Estado HTTP	Causa	Condición
400	`MANDATORY_IE_MISSING`	Falta un campo requerido (`supiOrSuci`/`servingNetworkName` en iniciar; `resStar` en confirmación; `supi` en deregister)
404	`USER_NOT_FOUND`	El UDM devolvió 404 para el suscriptor durante la generación del vector
404	`CONTEXT_NOT_FOUND`	El `authCtxId` en una solicitud de confirmación no coincide con ningún contexto activo
500	`SYSTEM_FAILURE`	UDM inalcanzable, o el UDM devolvió un vector de autenticación mal formado

Procedimientos Clave

Autenticación 5G-AKA (Sección 6.1.3 de TS 33.501)



Iniciación. En `POST .../ue-authentications` el AUSF crea un contexto, luego solicita un vector 5G-AKA del UDM en `/nudm-ueau/v1/{supi0rSuci}/security-information/generate-auth-data`, incluyendo el id de instancia NF de AUSF y cualquier `resynchronizationInfo` suministrado por el AMF. El UDM devuelve un vector de autenticación (`rand`, `autn`, `xresStar`, `kausf`) y el `supi` resuelto (de-concealado del SUCI). El AUSF calcula `HXRES*` a partir de RAND y XRES*, deriva `K_SEAF` de `K_AUSF` vinculado al nombre de la red de servicio, almacena el material y devuelve el desafío al AMF. Los campos del vector se decodifican

en hex de manera defensiva: un vector faltante o mal formado produce un limpio `500 SYSTEM_FAILURE` en lugar de un fallo.

Confirmación. En `PUT .../{authCtxId}/5g-aka-confirmation` el AUSF busca el contexto, decodifica `resStar` y lo compara directamente con el `XRES*` almacenado. Informa el resultado al UDM a través de `auth-events` (mejor esfuerzo; un informe fallido se registra pero no cambia la respuesta al AMF) y devuelve el resultado en el cuerpo. Luego, el contexto se elimina. Un `authCtxId` desconocido devuelve `404 CONTEXT_NOT_FOUND`.

Nombre de la red de servicio. Construido a partir de `mcc`/`mnc` como `5G:mnc<MNC>.mcc<MCC>.3gppnetwork.org` según TS 24.501, con MNC de dos dígitos rellenados con ceros a tres dígitos. El nombre de la red de servicio vincula `K_SEAF` a la red de servicio, por lo que el AMF y el AUSF deben coincidir en la identidad PLMN.

Resincronización de SQN. Cuando el AMF incluye `resynchronizationInfo` (el `RAND` y `AUTS` producidos por el UE) en la solicitud de inicio, el AUSF detecta la condición de resync, valida que tanto `rand` como `auts` estén bien formados en hex, y reenvía el par validado al UDM/ARPF en la solicitud `generate-auth-data` para que el UDM pueda volver a derivar el número de secuencia (TS 33.501 6.1.3.3). La entrada de resync mal formada se descarta y no se reenvía. El AUSF no calcula el número de secuencia, pero no es un paso ciego: inspecciona y valida el blob de resync antes de reenviarlo.

Jerarquía de Claves y División de Derivación (Anexo A de TS 33.501)

La jerarquía de claves 5G-AKA se divide entre el UDM y el AUSF. El AUSF nunca ve la clave a largo plazo (Ki) o CK/IK; recibe `K_AUSF` ya derivado por el UDM dentro del vector de autenticación y solo deriva la clave de anclaje `K_SEAF` a partir de ella. El AMF (no el AUSF) continúa la cadena hasta `K_AMF` y las claves de acceso.



UDM / ARPF



K_SEAF = KDF(K_AUSF,
SNN)
FC 0x6C (A.6)



K_AMF = KDF(K_SEAF,
SUPI, ABBA)
FC 0x6D (A.7)

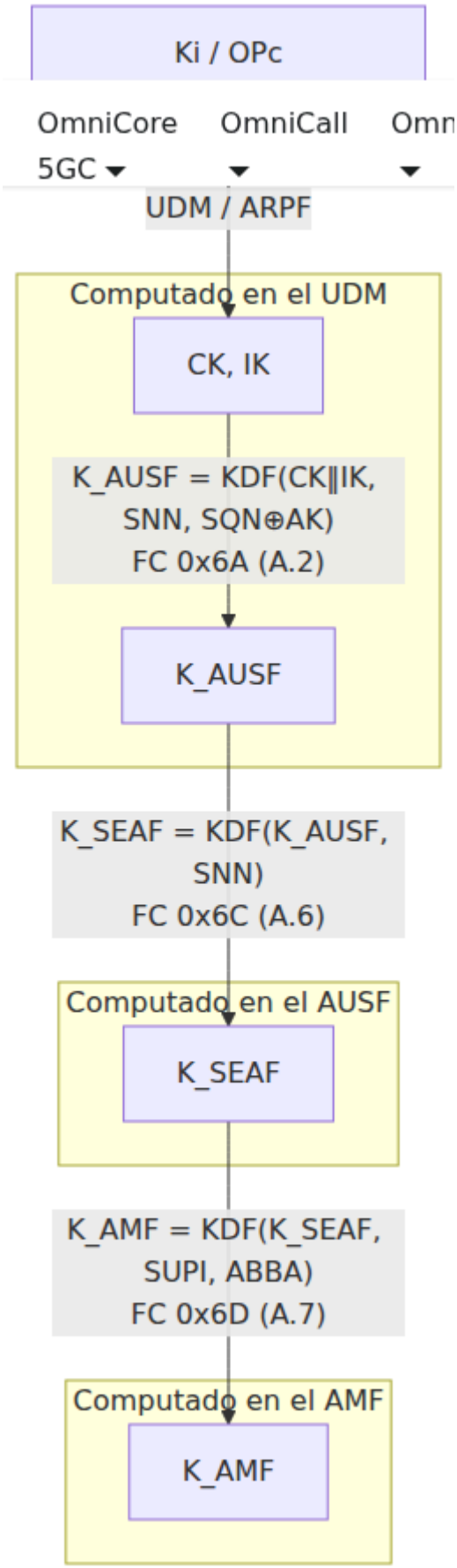


Clave / valor	Donde se deriva	Derivación	Propósito en el AUSF
`K_AUSF`	UDM (ARPF)	`KDF(CK‖IK, servingNetworkName, SQN⊕AK)`, FC `0x6A` (A.2)	Recibido en el vector; la clave de anclaje AUSF de la cual se deriva `K_SEAF`. Se mantiene en contexto, nunca se devuelve al AMF
`K_SEAF`	AUSF	`KDF(K_AUSF, servingNetworkName)`, FC `0x6C` (A.6)	La clave de anclaje de la red de servicio. Se devuelve al AMF solo en `AUTHENTICATION_SUCCESS`; el AMF deriva `K_AMF` a partir de ella
`HXRES*`	AUSF	lower 128 bits de `SHA-256(RAND ‖ XRES*)` (A.5)	Devuelto al AMF en la iniciación para que el AMF pueda hacer coincidir el `HRES*` del UE antes de reenviar `RES*`
`XRES*`	UDM (ARPF)	`KDF(CK‖IK, SNN, RAND, XRES)`, FC `0x6B` (A.4)	Recibido en el vector; la respuesta esperada contra la cual el AUSF compara `RES*` en la confirmación

Debido a que `K_SEAF` está vinculado al `servingNetworkName` (y `K_AUSF` aguas arriba está vinculado al mismo nombre de SN y a `SQN⊕AK`), el AMF, AUSF y UDM deben coincidir en la identidad PLMN para que las claves derivadas coincidan con las del UE. Un desajuste en el nombre de la red de servicio se presenta como un `AUTHENTICATION_FAILURE` en la confirmación, no como un error explícito.

Eliminar Resultado de Autenticación 5G-AKA (TS 29.509)

El AMF puede eliminar explícitamente un resultado/contexto de autenticación sin un intercambio de confirmación. Esto se mapea a `Delete5gAkaAuthenticationResult` en TS 29.509.

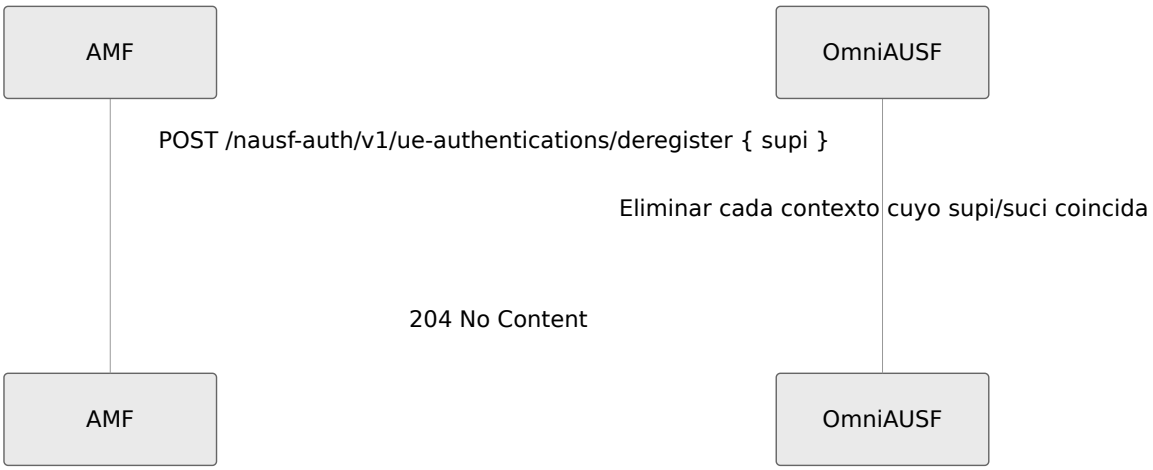


El AUSF elimina el contexto incondicionalmente (un `authCtxId` desconocido es un no-op) y siempre devuelve `204 No Content`; no se realiza ninguna llamada

al UDM.

Deregister (TS 29.509)

Elimina cualquier contexto(s) de seguridad UE que el AUSF tenga para un SUPI dado.



`supi` es obligatorio (un valor faltante/vacío devuelve `400 MANDATORY_IE_MISSING`). El AUSF itera sobre sus contextos en memoria y elimina cualquier contexto cuyo `supi` o `suci` almacenado sea igual al SUPI solicitado; la operación es local y no llama al UDM. Siempre devuelve `204 No Content`, incluso cuando no existe un contexto coincidente.

Contexto de Autenticación UE

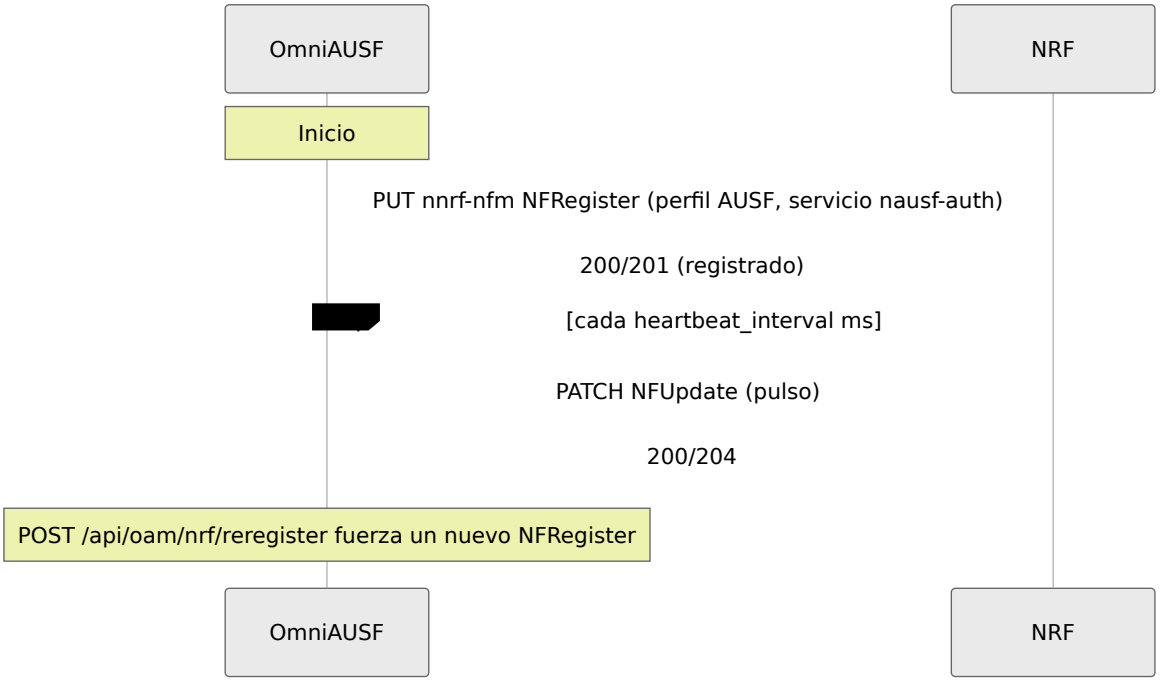
Cada iniciación crea un contexto de autenticación UE en memoria, identificado por un `authCtxId` generado (UUID). El contexto contiene el material necesario para completar el intercambio y es la unidad que los puntos finales de la sesión OAM (`GET /api/ausf_sessions`, `DELETE /api/ausf_sessions/{id}`) enumeran y limpian.

Campo	Establecido en	Significado
`ctx_id`	iniciación	El UUID `authCtxId`; aparece en la URL de confirmación y en la lista de sesiones OAM
`suci`	iniciación	El `supiOrSuci` presentado originalmente por el AMF (SUCI o SUPI)
`supi`	respuesta del vector	El SUPI plano resuelto (de-concealado) por el UDM
`serving_network_name`	iniciación	El nombre de la red SN al que están vinculadas las claves
`rand`, `xres_star`, `hxres_star`, `kausf`, `kseaf`	respuesta del vector	El vector y las claves derivadas mantenidas durante la duración del intercambio
`auth_type`	respuesta del vector	`5G_AKA`
`auth_result`	-	Poblado para una sesión resuelta; un contexto aún en vuelo no tiene resultado y se cuenta como `pending` en `GET /api/statistics`

Un contexto normalmente existe solo entre la iniciación y la resolución. Se elimina en la confirmación (éxito **o** fallo), en un `DELETE .../5g-aka-confirmation` explícito, en `deregister` para su SUPI, y en el reinicio del proceso. Además, los contextos inactivos o abandonados son recolectados por un recolector periódico supervisado: un contexto que el AMF inicia pero nunca confirma se recolecta una vez que excede `context_ttl_seconds` (por defecto `3600` segundos), con el recolector barriendo cada `context_reap_interval_ms`

(por defecto `60000` ms). Esto limita la duración de los contextos abandonados sin intervención del operador. Consulte Crecimiento del conteo de contextos activos para cómo inspeccionar y limpiar contextos manualmente si es necesario.

Registro de NRF



Al iniciar, el AUSF registra un perfil NF `AUSF` con el NRF anunciando el servicio `nausf-auth` (`v1`, `1.0.0`), su dirección/puerto/esquema SBI y el PLMN configurado. Envía pulsos cada `heartbeat_interval` milisegundos. El registro se puede forzar nuevamente en tiempo de ejecución a través de `POST /api/oam/nrf/reregister`. El AUSF también utiliza el NRF para descubrir el UDM: antes de cada llamada Nudm, consulta `Nnrf_NFDiscovery` (TS 29.510) para una instancia `udm` y apunta al UDM descubierto. Solo cuando el descubrimiento no produce ninguna instancia (o el NRF es inalcanzable) se retrocede a la `udm_uri` configurada estáticamente, por lo que las implementaciones de UDM único y estático siguen funcionando.

← Resumen

Solución de problemas de OmniAUSF

La autenticación falla con 404 USER_NOT_FOUND

El UDM devolvió 404 para el suscriptor durante `generate-auth-data`. Confirme:

1. `udm_uri` es correcto y accesible desde el host de OmniAUSF.
2. El suscriptor (IMSI) está provisionado en el backend de UDM/UDR/HSS.
3. El SUCI presentado por el AMF está bien formado y es deconocible por el UDM.

Los registros de AUSF indican `AUSF auth failed for {supiOrSuci}: user not found at UDM`.

La confirmación devuelve AUTHENTICATION_FAILURE (HTTP 200)

El `resStar` decodificado no era igual al `XRES*` almacenado. Esto se señala en el cuerpo (`authResult: "AUTHENTICATION_FAILURE"`), no como un error HTTP. Causas comunes:

- Las credenciales del UE (Ki/OPc) no coinciden con las provisionadas en el backend.
- El RAND/AUTN o RES* fueron alterados en tránsito, o el AMF/AUSF no están de acuerdo sobre el nombre de la red de servicio (lo que produciría un vector desajustado).
- Un `resStar` mal formado/no hexadecimal se trata como un desajuste.

Los registros de AUSF indican `AUSF 5G-AKA authentication FAILED for {supi}: RES* mismatch`.

La confirmación devuelve 404 CONTEXT_NOT_FOUND

El `authCtxId` en la solicitud PUT no coincide con ningún contexto activo. Los contextos se eliminan después de cualquier confirmación (éxito o fallo) y se pierden en el reinicio del AUSF. El AMF debe reiniciar la autenticación desde `POST .../ue-authentications`.

La iniciación devuelve 500 SYSTEM_FAILURE

O bien el UDM es inalcanzable en `udm_uri`, o el UDM devolvió un vector de autenticación mal formado (falta/no hexadecimal `rand`, `xresStar`, o `kausf`, o falta `autn`). Verifique la conectividad del UDM y que la generación de vectores del UDM esté saludable. Los registros de AUSF indican `AUSF auth failed ...` con la razón, o `AUSF auth failed: malformed authentication vector from UDM`.

AUSF no es descubrible / el estado de NRF muestra no registrado

Verifique `GET /api/status/nrf` y la métrica `omni_ausf_nrf_registration_status`. Verifique que `nrf_uri` sea correcto (tenga en cuenta la advertencia de valor predeterminado en tiempo de ejecución en la referencia de configuración) y accesible, luego fuerce la re-registro con `POST /api/oam/nrf/reregister`.

Aumento en el conteo de contextos activos

Los contextos de autenticación en vuelo deberían regresar a cerca de cero en reposo. Los contextos se eliminan en la confirmación, eliminación o desregistro. Si el AMF abandona las autenticaciones sin confirmar, esos contextos son recolectados automáticamente por el recolector de contextos periódico una vez que superan `context_ttl_seconds` (valor predeterminado `3600` segundos); el recolector barre cada `context_reap_interval_ms` (valor predeterminado `60000` ms). Por lo tanto, los contextos obsoletos se eliminan por sí mismos después del TTL sin intervención del operador.

Si necesita inspeccionar o eliminar contextos antes, use la API OAM: `GET /api/status/nf` o `GET /api/statistics` para los conteos en vivo, `GET /api/ausf_sessions` para listarlos, y `DELETE /api/ausf_sessions/{id}` (o `POST /api/ausf_sessions` para eliminar todos) para eliminarlos inmediatamente. La métrica `omni_ausf_active_contexts_count` rastrea este conteo directamente (se emite en la creación, eliminación de contextos y cada barrido del recolector), por lo que puede monitorearlo en Prometheus así como a través de la API OAM. Si el conteo aumenta constantemente más rápido de lo que el TTL puede drenarlo, verifique si el AMF está iniciando repetidamente autenticaciones que nunca confirma.