← Resumen

Referencia de Configuración de OmniCHF

La configuración del operador se lee desde la clave del entorno de la aplicación `:omnichf`, poblada al inicio por `config/runtime.exs`. Cada clave es sobreescribible por una variable de entorno del sistema operativo. El ejemplo a continuación muestra la configuración completa en tiempo de ejecución con los valores aplicados cuando la variable de entorno correspondiente no está establecida.

```
config :omnichf,
  # Escuchador SBI (interfaz basada en servicio)
  sbi_scheme: "http",
  sbi_addr:   "127.0.0.14",
  sbi_port:   7777,

  # Registro / latido del NRF
  nrf_uri:    "http://127.0.0.1:7777",

  # Identidad PLMN de servicio
  mcc: "999",
  mnc: "70",

  # Observabilidad
  prometheus_metrics_port: 9568,

  # Cadencia de latido del NRF
  heartbeat_interval: 10_000,

  # Integración del motor de calificación / saldo de CGRateS
  cgrates_enabled: true,
  cgrates_url:     "http://localhost:2080/jsonrpc",
  cgrates_tenant:  "cgrates.org",
  cgrates_timeout: 5000

# Registro estructurado en JSON
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all
```

Parámetros Principales

Parámetro	Tipo	Requerido	Predeterminado
`sbi_scheme`	String	No	`"http"`
`sbi_addr`	String	No	`"127.0.0.14"`
`sbi_port`	Integer	No	`7777`
`nrf_uri`	String	No	`"http://127.0.0.1:777`

Parámetro	Tipo	Requerido	Predeterminado
`mcc`	String	No	`"999"`
`mnc`	String	No	`"70"`
`prometheus_metrics_port`	Integer	No	`9568`

Parámetro	Tipo	Requerido	Predeterminado
`heartbeat_interval`	Integer (ms)	No	`10000`

Parámetros de CGRateS

Estas claves controlan la integración con el motor de calificación y saldo externo de CGRateS. Ver Integración de CGRateS para detalles de comportamiento.

Parámetro	Tipo	Requerido	Predeterminado
`cgrates_enabled`	Boolean	No	`true` (`"true"`)
`cgrates_url`	String	No	`"http://localhost:2080/jsonr`

Parámetro	Tipo	Requerido	Predeterminado
`cgrates_tenant`	String	No	`"cgrates.org"`
`cgrates_timeout`	Integer (ms)	No	`5000`

> **Predeterminado cuando está deshabilitado en tiempo de construcción:** el predeterminado compilado (utilizado si el entorno de la aplicación nunca se llena por `runtime.exs`) es `cgrates_enabled: false`. En un despliegue normal, `runtime.exs` lo establece, y allí el predeterminado es `true`. El perfil de prueba obliga a `cgrates_enabled: false`.

Claves Avanzadas / Opcionales

Estas claves no son pobladas por el `runtime.exs` predeterminado, pero se leen del entorno `:omnichf` si están presentes. Ajustan la resolución del grupo de calificación, la carga fuera de línea y la generación de ID de carga. No tienen una variable de entorno dedicada y se establecen directamente en el bloque de configuración `:omnichf`.

Parámetro	Tipo	Requerido	Predeterminado
`rating_groups`	Map	No	`%{}`
`default_rating_group`	Integer	No	`1`
`offline_charging_enabled`	Boolean	No	`false`
`cdr_output_dir`	String	No	`"/var/log/omnichf/c`
`node_id`	Integer	No	`1`

Parámetro	Tipo	Requerido	Predeterminado

Grupos de Calificación

`rating_groups` mapea un **DNN** (la clave del mapa, una cadena) al **grupo de calificación** (el valor, un entero) aplicado a las sesiones de ese DNN. El grupo de calificación resuelto aparece en el campo `ratingGroup` de `multipleUnitInformation` devuelto al consumidor y en el campo `rating_group` del CDR. La resolución es:

1. Si el DNN de la sesión es una clave en `rating_groups`, use ese valor.
2. De lo contrario, use `default_rating_group`.

```
config :omnichf,
  rating_groups: %{
    "internet" => 10,
    "ims"      => 20,
    "iot"      => 30
  },
  default_rating_group: 1
```

La coincidencia se realiza en la cadena DNN exacta. Un DNN sin entrada (y sin `default_rating_group`) vuelve al grupo de calificación `1`. Un valor `rating_groups` mal formado (no un mapa) se ignora y se utiliza el predeterminado, por lo que una mala configuración nunca bloquea la carga.

Registro

El registro estructurado en JSON se configura a través del `:logger` `:default_formatter`.

Parámetro	Tipo	Predeterminado	Descripción
`format`	Tuple	`{OmniLogger.JsonFormatter, :format}`	Formateador de línea de registro. Emite JSON estructurado, adecuado para la ingestión por un transportador de registros.
`metadata`	Atom / List	`:all`	Metadatos de registro incluidos en cada línea. `:all` emite todos los metadatos adjuntos (SUPI, DNN, ID de sesión PDU, ID de carga, procedimiento, etc.).

Interfaces de Gestión

OmniCHF expone superficies de gestión orientadas al operador junto al SBI. Estos puertos son relevantes para la supervisión y OAM, pero no son parte del servicio Nchf:

Interfaz	Puerto	Esquema	Propósito
SBI (Nchf)	`sbi_port` (predeterminado `7777`)	HTTP	Tráfico de carga inte 5G.
Métricas de Prometheus	`prometheus_metrics_port` (predeterminado `9568`)	HTTP	Punto final de raspa de métricas.
API de Gestión	`8443`	HTTPS	Lecturas de estado, lecturas de sesión/estadísticas/s de CGRateS, búsque de CDR y acciones d OAM (configuración, vaciado de CDR, niv de registro, aborto d sesión, re-registro d NRF).
Panel de Control	`7443`	HTTPS	Interfaz web para recursos, configurac licencia y registros.

Puntos Finales de la API de Gestión / OAM

La API de Gestión en el puerto `8443` (HTTPS) expone las siguientes superficies para el operador:

Método	Ruta	Propósito
`GET`	`/api/status/api`	Leer estado de API/servicio.
`GET`	`/api/status/license`	Leer estado de licencia.
`GET`	`/api/status/nrf`	Leer estado de registro del NRF.
`GET`	`/api/status/nf`	Leer estado del NF (nodo).
`GET`	`/api/sessions`	Listar sesiones de carga activas.
`GET`	`/api/sessions/{id}`	Inspeccionar una sola sesión de carga por `charging_data_ref`.
`GET`	`/api/statistics`	Leer estadísticas de carga (número de sesiones activas, volumen total cargado, conectividad de CGRateS).
`GET`	`/api/health/cgrates`	Leer salud de conectividad de CGRateS.
`GET`	`/api/oam/config`	Leer configuración en tiempo de ejecución.
`PATCH`	`/api/oam/config/{id}`	Actualizar una clave de configuración en tiempo de ejecución.
`POST`	`/api/oam/charging_session`	Abortar una sesión de carga por `charging_data_ref`.
`GET`	`/api/oam/cdr`	Buscar CDRs completados. Parámetros de consulta: `supi`, `dnn`, `date_from`, `date_to` (todos opcionales).

Método	Ruta	Propósito
`POST`	`/api/oam/cdr/flush`	Forzar el vaciado del uso acumulado como un CDR para una sesión (`{"charging_data_ref": "<ref>"}`) o todas las sesiones rastreadas (`{}` / `"all"`).
`POST`	`/api/oam/log_level`	Cambiar el nivel de registro en tiempo de ejecución.
`POST`	`/api/oam/nrf/reregister`	Activar el re-registro del NRF.

> **Visibilidad del almacenamiento de sesiones:** las superficies de gestión orientadas a sesiones (`/api/sessions`, `/api/statistics`, `/api/oam/charging_session`, `/api/oam/cdr/flush`) presentan cada sesión activa. La ruta de carga SBI activa mantiene cada sesión en un proceso **worker** por sesión, y cada sesión worker se refleja en el **Context store** en memoria al crear y actualizar y se elimina al detener. Las superficies de gestión consultan primero el registro de trabajadores y luego recurren al Context, por lo que una sesión activa siempre es visible y accionable. Ver Almacenamiento de Sesiones y Visibilidad de Gestión para lo que esto significa operativamente. La búsqueda de CDR (`/api/oam/cdr`) lee los archivos CDR fuera de línea y es independiente de ambos almacenes.

← Resumen

Referencia de Métricas de OmniCHF

Las métricas se exponen en `prometheus_metrics_port` (por defecto `9568`) en `/metrics`.

Métricas de Carga

Métrica: `omni_chf_charging_initial_count`
Tipo: Contador
Descripción: Solicitudes de sesión de carga INICIAL (Crear) recibidas.
Etiquetas: `supi`

Métrica: `omni_chf_charging_update_count`
Tipo: Contador
Descripción: Solicitudes de sesión de carga ACTUALIZAR recibidas.
Etiquetas: `ref`

Métrica: `omni_chf_charging_release_count`
Tipo: Contador
Descripción: Solicitudes de sesión de carga LIBERAR recibidas.
Etiquetas: `ref`

Métrica: `omni_chf_charging_creates_total`
Tipo: Contador
Descripción: Resultados de la operación de creación.
Etiquetas: `result` (`success` / `failure`)

Métrica: `omni_chf_charging_updates_total`
Tipo: Contador
Descripción: Resultados de la operación de actualización.
Etiquetas: `result` (`success` / `failure`)

Métrica: `omni_chf_charging_releases_total`
Tipo: Contador
Descripción: Resultados de la operación de liberación. Liberado a través del SBI con un CGRateS terminate exitoso (`success`), liberado a través del SBI donde el `TerminateSession` de CGRateS falló pero el CDR aún se finalizó (`failure`, presentado al consumidor como un `500`), o finalizado por la fuerza a través del punto final de vaciado de CDR de OAM (`oam_flush`).
Etiquetas: `result` (`success` / `failure` / `oam_flush`)

Métrica: `omni_chf_sessions_active_count`
Tipo: Medidor
Descripción: Sesiones de carga activas actualmente mantenidas en memoria. Emitido en cada creación y eliminación de almacenamiento de contexto, que ahora incluye cada sesión de trabajador en vivo reflejada, por lo que el medidor rastrea el verdadero conteo de sesiones activas.
Etiquetas: ninguna

```
# Tasa de creación de sesión de carga
rate(omni_chf_charging_creates_total{result="success"}[5m])

# Tasa de fallos en la creación
sum(rate(omni_chf_charging_creates_total{result="failure"}[5m]))
  / sum(rate(omni_chf_charging_creates_total[5m]))

# Sesiones actualmente activas
omni_chf_sessions_active_count
```

Métricas de CGRateS

Métrica: `omni_chf_cgrates_requests_total`
Tipo: Contador
Descripción: Solicitudes JSON-RPC de CGRateS.
Etiquetas: `operation` (método SessionS), `result` (`success` / `failure`)

Métrica: `omni_chf_cgrates_request_duration_ms`
Tipo: Histograma
Descripción: Duración de la solicitud JSON-RPC de CGRateS en milisegundos.

Etiquetas: `operation`
Cubos: 5, 10, 25, 50, 100, 250, 500, 1000, 2500

Métrica: `omni_chf_cgrates_health`
Tipo: Medidor
Descripción: Salud de conectividad de CGRateS (`1` = alcanzable, `0` = caído/deshabilitado).
Etiquetas: ninguna

```
# Latencia p99 de CGRateS por operación
histogram_quantile(0.99,
  sum by (le, operation)
(rate(omni_chf_cgrates_request_duration_ms_bucket[5m])))

# Tasa de errores de CGRateS
sum(rate(omni_chf_cgrates_requests_total{result="failure"}[5m]))

# Alcance de CGRateS (alerta cuando 0)
omni_chf_cgrates_health
```

Métricas de NRF

Métrica: `omni_chf_nrf_registration_status`
Tipo: Medidor
Descripción: Estado de registro de NRF (`1` = registrado, `0` = no registrado).
Etiquetas: `nf_type`

```
omni_chf_nrf_registration_status
```

Métricas de BEAM VM

Los medidores de salud de tiempo de ejecución estándar también se exportan.

Métrica	Tipo	Descripción
`beam_memory_total`	Medidor	Memoria total de BEAM (bytes)
`beam_memory_processes` / `beam_memory_processes_used`	Medidor	Memoria de proceso asignada / utilizada
`beam_memory_system`	Medidor	Memoria del sistema (no de proceso)
`beam_memory_atom` / `beam_memory_atom_used`	Medidor	Memoria de átomo asignada / utilizada
`beam_memory_binary`	Medidor	Memoria binaria
`beam_memory_code`	Medidor	Memoria de código
`beam_memory_ets`	Medidor	Memoria de tabla ETS
`beam_processes_count`	Medidor	Conteo de procesos Erlang
`beam_ports_count`	Medidor	Conteo de puertos Erlang
`beam_atom_count`	Medidor	Conteo de átomos
`beam_vm_uptime`	Medidor	Tiempo de actividad de la VM (segundos)

← Resumen

Guía de Operaciones de OmniCHF

Tabla de Contenidos

1. Resumen
2. Ciclo de Vida de la Sesión de Carga
3. Referencias de Rol y Especificación 3GPP
4. Puntos Finales de SBI
5. Integración de CGRateS
6. Procedimientos Clave

Resumen

OmniCHF implementa la **Función de Carga (CHF)** del núcleo 5G. El CHF proporciona carga en línea y fuera de línea convergente para sesiones PDU 5G a través del servicio **Nchf_ConvergedCharging** (3GPP TS 32.290 / TS 32.291 / TS 32.255).

Un consumidor de carga, típicamente el **SMF**, crea una sesión de carga para cada sesión PDU, informa el uso durante la vida útil de esa sesión y finalmente la libera. OmniCHF traduce cada solicitud de carga 5G en una llamada JSON-RPC de **CGRateS** SessionS para autorización de crédito en tiempo real (carga en línea / gestión de cuotas) y genera un **Registro de Datos de Carga (CDR)** al liberar la sesión (carga fuera de línea).

Cada sesión de carga es propiedad de un **ChargingWorker** GenServer dedicado bajo un `DynamicSupervisor` (arquitectura de proceso por sesión). Todo el estado de una sesión (SUPI, DNN, S-NSSAI, volumen/duración acumulados, grupo de calificación, estado de sesión de CGRateS, ID de carga)

se mantiene en ese proceso de trabajador. Un fallo en una sesión no afecta a las demás. El estado de la sesión se mantiene en memoria y se reconstruye a medida que los consumidores crean nuevas sesiones de carga después de un reinicio.

Responsabilidades principales:

- Servir el servicio `nchf-convergedcharging/v3` en el SBI.
- Asignar un `chargingDataRef` por sesión y gestionar su ciclo de vida INICIAL → ACTUALIZAR → TERMINAR.
- Conceder unidades de servicio (GSU) en respuesta a Unidades de Servicio Solicitadas (RSU), devolviendo volumen/tiempo concedido y **disparadores** de gestión de cuotas al consumidor.
- Conectar con CGRateS para la gestión de calificación y saldo (carga en línea).
- Emitir CDRs al registro de aplicaciones y, opcionalmente, a archivos de carga fuera de línea por día.
- Registrarse con el NRF y mantener un latido.

Ciclo de Vida de la Sesión de Carga

Cada sesión PDU se asigna a una sesión de carga, rastreada por un `chargingDataRef` (UUID).

Estado	Disparador	Acción
Creado	`POST /chargingdata`	ChargingWorker creado, llamada a CGRateS `InitiateSession`, cuota inicial concedida
Actualizado	`POST /chargingdata/{ref}/update`	Llamada a CGRateS `UpdateSession`, uso acumulado, número de secuencia incrementado, nueva cuota concedida
Liberado	`POST /chargingdata/{ref}/release`	Llamada a CGRateS `TerminateSession`, CDR construido y emitido, trabajador detenido

Almacenamiento de Sesiones y Visibilidad de Gestión

OmniCHF mantiene el estado de sesiones activas en **dos** almacenes en memoria que se mantienen sincronizados, de modo que las superficies de gestión ven y pueden actuar sobre cada sesión activa:

Almacén	Respaldo	Contiene	Rol
Registro de trabajadores	`ChargingWorker` GenServer por sesión bajo un `DynamicSupervisor`, indexado en un `Registry`	Estado completo por sesión (volumen/duración acumulados, número de secuencia, estado de CGRateS)	Fuente de verdad para una sesión activa. Crear genera un trabajador; Actualizar y Liberar resuelven primero el trabajador
Almacén de contexto	Un único mapa `Agent` en memoria indexado por `chargingDataRef`	Un espejo de los mismos atributos de sesión	Modelo de lectura para observabilidad y superficies OAM

En el SBI, **Crear** siempre genera un trabajador, y **Actualizar** y **Liberar** resuelven primero ese trabajador, volviendo al almacén de contexto solo si no se encuentra ningún trabajador (por ejemplo, después de un fallo del supervisor). Cada sesión de trabajador activa se refleja en el almacén de contexto al crear y en cada actualización, y se elimina de él cuando el trabajador se detiene (liberación normal, aborto o fallo). Las superficies de gestión consultan primero el registro de trabajadores y luego vuelven al contexto, por lo que ambos almacenes presentan una vista consistente.

Operativamente, esto significa:

- Una sesión creada normalmente es completamente visible a través de las superficies de gestión: aparece en `GET /api/sessions` y en los totales de estadísticas, y puede ser abortada (`POST /api/oam/charging_session`) y vaciada de CDRs (`POST /api/oam/cdr/flush`) a través de los puntos finales OAM mientras esté activa.

- El medidor de sesiones activas, estadísticas y listados de sesiones reflejan el verdadero conjunto de sesiones activas, no solo un subconjunto de respaldo.
- Ambos almacenes se mantienen en memoria y se repoblan a medida que los consumidores crean nuevas sesiones después de un reinicio. Dado que las sesiones se mantienen en un registro local, un `chargingDataRef` es propiedad de la instancia que lo creó.

El medidor `omni_chf_sessions_active_count` y los contadores `omni_chf_charging_creates_total` / `omni_chf_charging_releases_total` corroboran el conteo de sesiones activas informado por `GET /api/sessions`.

Referencias de Rol y Especificación 3GPP

Especificación	Relevancia
TS 23.501	Arquitectura del sistema 5G: rol del CHF, concepto de carga convergente
TS 32.240	Arquitectura y principios de carga
TS 32.290	Arquitectura de carga del sistema 5G y flujos de mensajes
TS 32.291	Carga del sistema 5G: API del servicio **Nchf_ConvergedCharging** (HTTP/2 SBI), modelo de datos ChargingDataRequest/Response
TS 32.255	Carga del dominio de conectividad de datos 5G (carga de sesión PDU)
TS 32.251	Formato de ID de carga (`chargingId`), cláusula 5.2.1.6 de TS 32.251
TS 32.298	Descripciones y codificación de parámetros CDR
TS 29.500 / 29.501	Marco común y diseño de API de SBI 5G
TS 29.510	Registro NF y latido con el NRF

Referencias de procedimientos Nchf_ConvergedCharging (TS 32.291):

Procedimiento	Cláusula
Crear Datos de Carga (INICIAL)	6.1.3.2.1
Actualizar Datos de Carga (ACTUALIZAR)	6.1.3.2.2
Liberar Datos de Carga (TERMINAR)	6.1.3.2.3
Notificar Carga (REAUTORIZACIÓN / ABORTAR_CARGA)	6.1.6.2

Puntos Finales de SBI

Todos los puntos finales de Nchf se sirven bajo la URL base `{sbi_scheme}://{sbi_addr}:{sbi_port}` y la ruta base `/nchf-convergedcharging/v3`.

Método	Ruta	Descripción
`POST`	`/chargingdata`	Crear una sesión de carga (**INICIAL**). Asigna un `chargingDataRef`, inicia una sesión de CGRateS, concede cuota inicial.
`POST`	`/chargingdata/{chargingDataRef}/update`	Actualizar una sesión de carga (**ACTUALIZAR**). Informa uso y solicita cuota adicional.
`POST`	`/chargingdata/{chargingDataRef}/release`	Liberar una sesión de carga (**TERMINAR**). Informa uso final, genera un CDR, termina la sesión de CGRateS.
`POST`	`/notify`	Disparador de callback de Notificación de Carga. Indica a OmniCHF que envíe una notificación de `REAUTORIZACIÓN` o `ABORTAR_CARGA` a la `notifyUri`

Método	Ruta	Descripción
		almacenada del consumidor.

Al liberar, se devuelve un `500` (causa `FALLO_DEL_SISTEMA`, título "Terminación de Carga Fallida") cuando la `TerminateSession` de CGRateS falla. El CDR aún se finaliza y se escribe, pero el fallo se muestra para que una inconsistencia en el backend de facturación (una sesión de CGRateS dejada abierta) no se oculte detrás de un falso `204`. Ver Liberar devuelve 500.

Validación de Solicitudes

Cada cuerpo de solicitud se valida para los siguientes Elementos de Información obligatorios antes de procesar. Un IE faltante devuelve `400 Solicitud Incorrecta` con causa `MANDATORY_IE_MISSING`:

IE Obligatorio	Tipo
`nfConsumerIdentification`	objeto
`invocationTimeStamp`	cadena (ISO 8601)
`invocationSequenceNumber`	entero

ChargingDataRequest: Campos Clave Consumidos

Campo	Tipo	Usado en	Descri
`subscriberIdentifier`	cadena	Todos	SUPI (por ejemplo `999700000000001`) como la `Account` retrocede a `nfConsumerIdent` si está ausente.
`notifyUri`	cadena	Crear	URI de callback al la posterior Notifi (REAUTORIZACIÓ ABORTAR_CARGA)
`pDUSessionChargingInformation`	objeto	Todos	Detalles de la ses `pduSessionInfor` ID de sesión PDU, tipo de RAT, `qoSI` `networkSlicingI`
`multipleUnitUsage`	array	Actualizar, Liberar	Contenedores de Las entradas de `usedUnitContain` elemento se suma volumen/duración También lleva el `r` solicitado.
`requestType`	cadena	Todos	`INITIAL_REQUEST` `UPDATE_REQUEST` o `TERMINATION_REQ`

ChargingDataResponse: Campos Clave Devueltos

Campo	Tipo	Descripción
`invocationSequenceNumber`	entero	`1` en Crear; incrementado en cada Actualización.
`invocationTimeStamp`	cadena	Marca de tiempo ISO 8601 de la respuesta.
`multipleUnitInformation`	array	Unidades concedidas: una entrada con `resultCode: "SUCCESS"`, `grantedUnit` (`totalVolume`, `time`) y el `ratingGroup` resuelto.
`triggers`	array	Disparadores de gestión de cuotas que indican al consumidor cuándo informar a continuación: un `QUOTA_THRESHOLD` (IMMEDIATE_REPORT a la mitad del volumen concedido) y un `TIME_LIMIT` (DEFERRED_REPORT a 3600 s).

Integración de CGRateS

CGRateS es el motor externo de calificación y saldo. OmniCHF actúa como un **cliente** de SessionS, mapeando cada operación de carga Nchf a un método JSON-RPC de SessionS:

Operación Nchf	Método SessionS de CGRateS
Crear (INICIAL)	`SessionSv1.InitiateSession`
Actualizar (ACTUALIZAR)	`SessionSv1.UpdateSession`
Liberar (TERMINAR)	`SessionSv1.TerminateSession`
Verificación de salud	`SessionSv1.GetActiveSessions` (límite 1)

El valor `MaxUsage` que devuelve CGRateS se pasa directamente de vuelta al consumidor como el `grantedUnit` (tanto `totalVolume` como `time`).

Comportamiento Habilitado vs Deshabilitado

Aspecto	`cgrates_enabled: true`	`cgrates_enabled: false` **(modo de bypass)**
Autorización de crédito	Real: CGRateS califica la cuenta y devuelve `MaxUsage`	Ninguno: se devuelve una concesión fija de **86400** unidades para cada Crear y Actualizar
Impacto en el saldo	El saldo del suscriptor se debita/reserva en CGRateS	No se toca ningún saldo
Terminar	`TerminateSession` finaliza la sesión de CGRateS y el CDR	No-op
Verificación de salud	Sondeo activo contra `cgrates_url`	Siempre informa `reachable: false`, `cgrates_url: "disabled"`
Firma de registro	`Iniciando sesión CGRateS para ...`	`Integración CGRateS deshabilitada, devolviendo autorización por defecto` (advertencia)
Caso de uso	Carga en línea de producción	Pruebas de laboratorio / integración sin un motor de calificación

> En modo de bypass, OmniCHF aún asigna `chargingDataRef`s, rastrea sesiones, incrementa números de secuencia, emite disparadores y genera CDRs; solo el control de crédito está simulado. Esto lo hace seguro para ejercer el flujo completo SMF↔CHF sin un despliegue de CGRateS.

La configuración de la integración de CGRateS (`cgrates_enabled`, `cgrates_url`, `cgrates_tenant`, `cgrates_timeout`) está documentada en la Referencia de Configuración.

Mapeo de Eventos 5G a CGRateS

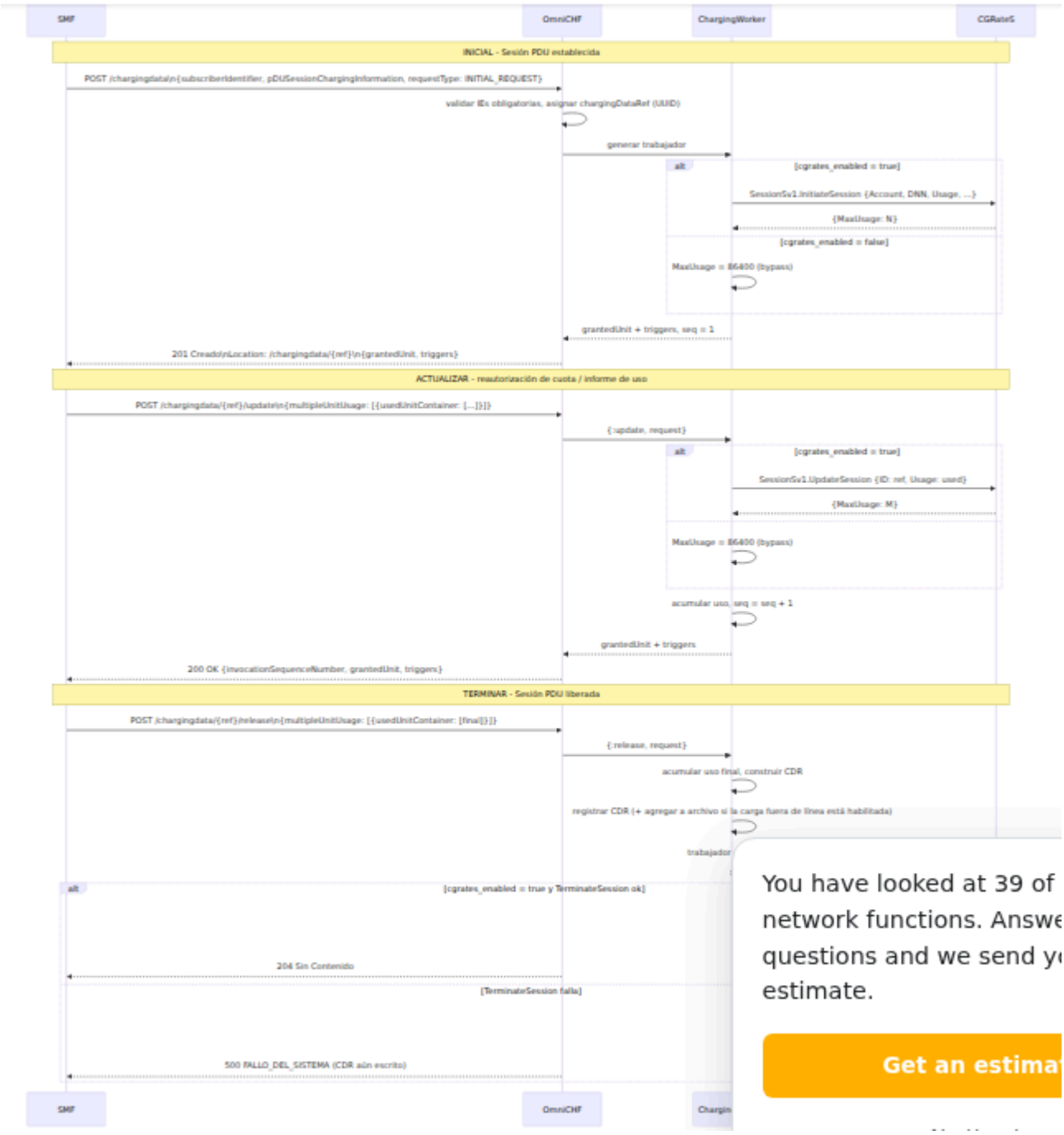
Cada llamada de SessionS lleva un mapa `Event` construido a partir del ChargingDataRequest:

Campo de Evento CGRateS	Fuente	
`Account` / `Subject`	`subscriberIdentifier` (SUPI)	Se retrocede `nfConsumerI` de lo contrar
`Destination` / `DNN`	`pduSessionInformation.dnnId` (o `.dnn`)	Nombre de la
`ToR`	derivado de `pduType`	Siempre `*da` PDU.
`RequestType`	`requestType`	Mapeado a `*`
`Category`	estático	`"data"`.
`Usage`	`usedUnitContainer`	Primer no cer luego `uplink` `downlinkVolu`
`OriginID`	`chargingDataRef`	Único por ses
`OriginHost`	estático	`"OmniCHF"`.
`SUPI`	`subscriberIdentifier`	Campo de ex
`S-NSSAI_SST` / `S-NSSAI_SD`	`pduSessionInformation.sNSSAI`	Valores por d `""`.
`5QI`	`qoSInformation.5qI` (o `fiveQI`)	Por defecto `9`
`RATType`	`pduSessionInformation.ratType`	Por defecto '
`PDUSessionID`	`pduSessionInformation.pduSessionID`	Por defecto `0`
`PDUSessionType`	`pduSessionInformation.pduType`	Por defecto '

Campo de Evento CGRateS	Fuente	
`ChargingOperation`	operación	`Inicial` / `Ac`

Procedimientos Clave

Sesión de Carga Convergente (INICIAL → ACTUALIZAR → TERMINAR)



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima

> Si la `TerminateSession` de CGRateS falla al liberar, OmniCHF **aún** finaliza y escribe el CDR, por lo que el registro fuera de l��nea no se pierde. El consumidor recibe entonces `500` (causa `FALLO_DEL_SISTEMA`) en lugar de un falso `204`, porque la sesión del lado de CGRateS puede quedar abierta y

> el backend de facturación puede necesitar reconciliación. En una terminación exitosa, el consumidor recibe `204 Sin Contenido`.

Gestión de Cuotas (RSU / GSU y Disparadores)

Según TS 32.291, el consumidor envía Unidades de Servicio Solicitadas (RSU) y OmniCHF devuelve Unidades de Servicio Concedidas (GSU) más **disparadores** que indican al consumidor cuándo volver:



El consumidor reautorizza (envía una ACTUALIZACIÓN) cuando alcanza el umbral de volumen o el límite de tiempo. OmniCHF también puede impulsar proactivamente la reautorización o el desmantelamiento a través del punto final `/notify` (`REAUTORIZACIÓN` / `ABORTAR_CARGA`), que publica en la `notifyUri` suministrada en la solicitud INICIAL.

Notificación de Carga (REAUTORIZACIÓN / ABORTAR_CARGA)

Según la cláusula 6.1.6.2 de TS 32.291, OmniCHF envía un `ChargingNotifyRequest` a la `notifyUri` almacenada del consumidor. El disparador es el punto final `POST /nchf-convergedcharging/v3/notify` orientado al operador, cuyo cuerpo lleva el `chargingDataRef` objetivo y el `notificationType`. OmniCHF busca la sesión, resuelve su `notifyUri` y publica la notificación al consumidor.



> La notificación es un esfuerzo de buena fe hacia la superficie del operador: una `notifyUri` rechazada o inalcanzable se registra pero no cambia el resultado de `/notify` una vez que se resuelve la sesión y `notifyUri`. Una sesión sin `notifyUri` almacenada produce un error interno `no_notify_uri`.

Campos CDR

Un CDR se construye al liberar, se registra en INFO (`CHF CDR: %{...}`), y, cuando `offline_charging_enabled` es `true`, se agrega como una línea JSON a `cdr_YYYYMMDD.json`:

Campo	Fuente
`record_type`	Estático `"5G_PDU_SESSION"`
`supi`	Contexto de sesión
`dnn`	Contexto de sesión / `pduSessionInformation`
`snssai`	`{sst, sd}`
`qos_5qi`	`qoSInformation.5qI` (por defecto `9`)
`rat_type`	`pduSessionInformation.ratType` (por defecto `"NR"`)
`pdu_session_id` / `pdu_session_type`	Contexto de sesión / solicitud
`volume_uplink` / `volume_downlink` / `volume_total`	Acumulado de todos los `usedUnitContainer`s
`duration`	Tiempo informado, de lo contrario diferencia de reloj
`start_time` / `end_time`	Tiempos de creación / liberación de sesión
`charging_data_ref`	UUID de sesión
`charging_id`	ID de carga de 32 bits de 3GPP (TS 32.251)
`recording_entity`	URL base de SBI de OmniCHF
`rating_group`	Resuelto a partir del mapeo de DNN

La salida de CDR fuera de línea (`offline_charging_enabled`, `cdr_output_dir`) y la composición del ID de carga (`node_id`) se configuran en la Referencia de Configuración.

Ejecución de Carga Fuera de Línea

Para capturar CDRs en un archivo para mediación/facturación:

1. Establezca `offline_charging_enabled: true` y apunte `cdr_output_dir` a un directorio escribible.
2. Los CDRs se acumulan como un objeto JSON por línea en un archivo por día llamado `cdr_YYYYMMDD.json` (la fecha es la fecha de liberación UTC). Se abre un nuevo archivo cada día UTC; no hay rotación o retención automática más allá de la división del nombre de archivo diario, así que elimine archivos antiguos con su propio mantenimiento.
3. Cada sesión liberada agrega exactamente un CDR. Las sesiones en progreso mantienen su uso en el trabajador/contexto hasta la liberación; no se escriben en el archivo hasta entonces.

Si `offline_charging_enabled` es `false`, los CDRs aún se construyen y se registran en INFO como `CHF CDR: %{...}`; solo se omite la escritura en el archivo. Para capturar CDRs sin salida a archivo, envíe esas líneas de registro con su canal de registro. Ver CDRs faltantes en archivos fuera de línea.

Localización y Búsqueda de CDRs

La API de Gestión expone `GET /api/oam/cdr` para buscar los archivos CDR escritos. Lee cada archivo `cdr_*.json` bajo `cdr_output_dir` y filtra según los parámetros de consulta suministrados (todos opcionales, combinados con AND):

Parámetro de consulta	Coincide	Formato
`supi`	CDR `supi` exacto	p. ej. `imsi-999700000000001`
`dnn`	CDR `dnn` exacto	p. ej. `internet`
`date_from`	Fecha del archivo CDR `>=` esta fecha	`YYYY-MM-DD` (guiones opcionales)
`date_to`	Fecha del archivo CDR `<=` esta fecha	`YYYY-MM-DD` (guiones opcionales)

Los filtros de fecha se aplican a la **fecha del nombre del archivo**, no a las marcas de tiempo de registros individuales, por lo que se incluye o excluye todo el archivo de un día como una unidad. Sin parámetros, el punto final devuelve cada CDR en disco. Debido a que lee archivos, `GET /api/oam/cdr` solo devuelve registros cuando `offline_charging_enabled` estaba activado en el momento de la liberación.

Forzar el Vaciado de un CDR (OAM)

En operación normal, un CDR solo se produce cuando se libera una sesión. `POST /api/oam/cdr/flush` permite a un operador emitir un CDR anticipadamente para cualquier sesión activa (ver Almacenamiento de Sesiones y Visibilidad de Gestión), por ejemplo, para cerrar sesiones que el consumidor nunca liberó:

Cuerpo de solicitud	Efecto
`{"charging_data_ref": "<ref>"}`	Construir y escribir un CDR para esa sesión, luego finalizarlo. `404` si la referencia no se encuentra.
`{}` o `{"charging_data_ref": "all"}`	Vaciar cada sesión actualmente rastreada.

El vaciado construye el CDR a partir del uso acumulado de la sesión más el último `ChargingDataRequest` y reutiliza el mismo escritor de CDR que en la liberación; **no** llama a `TerminateSession` de CGRateS. La respuesta informa `write_result` y `offline_charging_enabled` por sesión. Si la carga fuera de línea está deshabilitada, el CDR se construye y se finaliza la sesión, pero nada se escribe en el archivo (`write_result: "ok"` sin salida a archivo). Cada vaciado incrementa `omni_chf_charging_releases_total{result="oam_flush"}`.

← Resumen

Solución de problemas de OmniCHF

400 Bad Request en cualquier solicitud de carga

Síntomas: El consumidor recibe `400` con causa `MANDATORY_IE_MISSING`.

Causas posibles:

- El cuerpo de la solicitud falta `nfConsumerIdentification`, `invocationTimeStamp` o `invocationSequenceNumber`.

Resolución:

1. Confirme que el consumidor incluya los tres IE obligatorios en cada operación (Crear, Actualizar, Liberar).
2. Verifique el campo `detail` de la respuesta ProblemDetails; enumera los nombres de los IE faltantes.

404 en Actualizar o Liberar

Síntomas: El consumidor recibe `404` con causa `CHARGING_DATA_NOT_FOUND`.

Causas posibles:

- OmniCHF se reinició entre Crear y la operación posterior; el estado de la sesión se mantiene en memoria, por lo que la sesión debe ser recreada después de un reinicio.
- El `chargingDataRef` en la ruta es incorrecto.
- La sesión ya fue liberada (su trabajador se ha detenido).

Resolución:

1. Grep logs para `CHF Update: unknown ref=` / `CHF Release: unknown ref=`.
2. Confirme que OmniCHF no se ha reiniciado desde que se creó la sesión.
3. Confirme que el consumidor está devolviendo el `chargingDataRef` del encabezado `Location` de Crear.

500 en Crear con CGRateS habilitado

Síntomas: El consumidor recibe `500` con causa `CHARGING_FAILED`.

Causas posibles:

- `InitiateSession` de CGRateS falló: motor inalcanzable, inquilino incorrecto o un error del lado de CGRateS.

Resolución:

1. Verifique que `cgrates_url` apunte a un endpoint `/jsonrpc` de CGRateS accesible.
2. Verifique que `cgrates_tenant` coincida con el inquilino de CGRateS.
3. Verifique el medidor `omni_chf_cgrates_health` (`1` = activo).
4. Revise los logs para `CGRateS InitiateSession failed ...`; la razón es una de `{:cgrates_error, msg}`, `{:http_error, status}` o `{:http_error, reason}`.
5. Confirme que el servicio `SessionSv1` esté habilitado en CGRateS.

Liberar devuelve 500

Síntomas: El consumidor recibe `500` con causa `SYSTEM_FAILURE` y título "Charging Termination Failed" en `POST /chargingdata/{ref}/release`.

Causas posibles:

- `TerminateSession` de CGRateS falló durante la Liberación (motor inalcanzable, inquilino incorrecto o un error del lado de CGRateS).

Impacto:

- El CDR todavía se construye y se escribe, por lo que el registro fuera de línea no se pierde.
- La sesión del lado de CGRateS puede quedar abierta, por lo que el backend de facturación puede ser inconsistente y necesita reconciliación.

Resolución:

1. Grep logs para `CGRateS TerminateSession failed` en el `ref` afectado.
2. Confirme la accesibilidad de CGRateS con el medidor `omni_chf_cgrates_health` (`1` = activo).
3. Reconciliar cualquier sesión de CGRateS que haya quedado abierta para ese `ref` contra el CDR escrito.
4. Correlacione con `omni_chf_charging_releases_total{result="failure"}` para ver la tasa de fallos en la terminación.

La verificación de salud de CGRateS falla mientras el demonio está activo

Síntomas: `omni_chf_cgrates_health` lee `0` pero CGRateS está en funcionamiento.

Causas posibles:

- CGRateS es lento: la verificación de salud limita su tiempo de espera a `min(cgrates_timeout, 3000)` ms.
- Desajuste de esquema (HTTP vs HTTPS) en `cgrates_url`.

Resolución:

1. Confirme que el esquema y el puerto de `cgrates_url` sean correctos.

2. Si CGRateS es legítimamente lento, investigue la carga de CGRateS en lugar de aumentar el límite.
3. Busque advertencias de `CGRateS unreachable at <url> ...`.

Alta latencia de CGRateS

Síntomas: Operaciones de carga lentas; `omni_chf_cgrates_request_duration_ms` elevado.

Resolución:

1. Inspeccione el histograma p95/p99 por `operación`.
2. Escale CGRateS o el camino de red entre OmniCHF y CGRateS.
3. Considere reducir `cgrates_timeout` para que las llamadas lentas fallen rápidamente en lugar de bloquear la sesión.

CDRs faltantes en archivos fuera de línea

Síntomas: Las líneas `CHF CDR:` aparecen en el registro, pero no se escribe ningún archivo `cdr_YYYYMMDD.json`.

Causas posibles:

- `offline_charging_enabled` es `false` (por defecto): los CDR se registran solo.
- `cdr_output_dir` no es escribible.

Resolución:

1. Establezca `offline_charging_enabled: true`.
2. Asegúrese de que `cdr_output_dir` exista y sea escribible; esté atento a errores de `[CDR] Failed to write CDR to file`.
3. Para capturar CDRs sin salida de archivo, envíe las líneas de registro `CHF CDR:` con su canal de registro.

El conteo de sesiones activas no disminuye

Síntomas: `omni_chf_sessions_active_count` se mantiene elevado.

Causas posibles:

- El consumidor nunca envía Liberar (las sesiones solo se limpian en la liberación o en un fallo del trabajador).
- Las liberaciones están devolviendo `404` (ver arriba) por lo que el consumidor las considera completadas mientras que OmniCHF puede no hacerlo.

Resolución:

1. Correlacione el conteo activo con las tasas de Crear vs Liberar.
2. Investigue el `404` en Liberar, que generalmente indica un reinicio previo.

Registro de NRF no mantenido

Síntomas: `omni_chf_nrf_registration_status` lee `0`.

Resolución:

1. Verifique que `nrf_uri` sea accesible desde `sbi_addr` de OmniCHF.
2. Confirme que `mcc` / `mnc` coincidan con la configuración PLMN de NRF.
3. Confirme que `sbi_addr` / `sbi_port` en el perfil publicitado sean accesibles por el NRF.
4. Revise los registros de inicio para errores de registro de NRF.