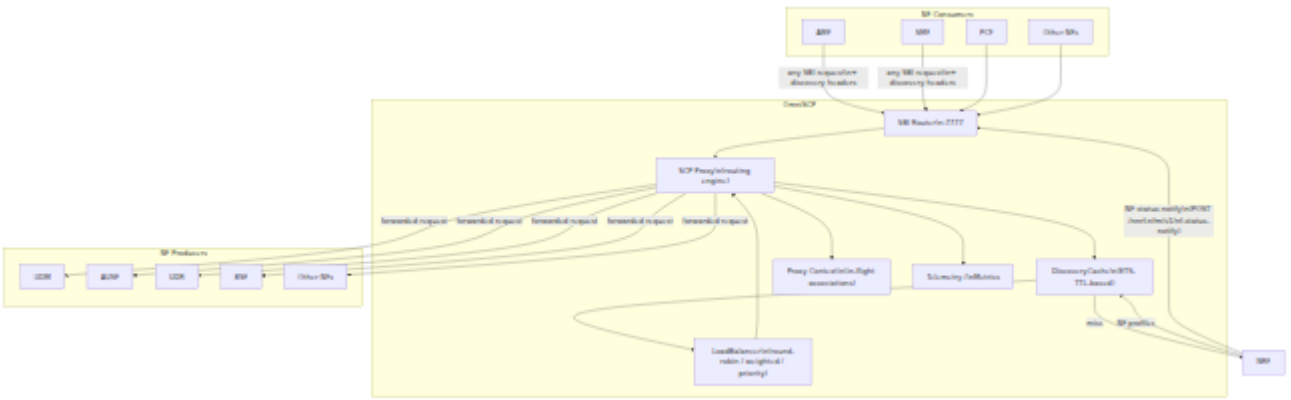
OmniSCP Operations Guide

Table of Contents

1. Component Overview
2. 3GPP Role and Spec References
3. SBI Endpoints
4. Configuration Reference
5. Key Procedures
6. Observability
7. Known Limitations
8. Troubleshooting

Component Overview

OmniSCP implements the Service Communication Proxy (SCP) network function defined in 3GPP TS 29.500 and TS 23.501. The SCP acts as an HTTP reverse proxy between NF consumers and NF producers in the 5G Service Based Architecture (SBA). It provides delegated NRF discovery, load balancing across NF producer instances, retry on failure, and NRF discovery result caching.



Routing Modes

The SCP supports three routing modes, evaluated in priority order on each incoming request:

1. **Direct forwarding** — `3gpp-Sbi-Target-apiRoot` header is present. The request is forwarded directly to the specified base URI without NRF lookup.
2. **Delegated discovery** — `3gpp-Sbi-Discovery-target-nf-type` and `3gpp-Sbi-Discovery-service-names` headers are present. The SCP queries the NRF (or hits the cache) and selects an instance via the configured load balancing strategy.
3. **Path-based inference** — No routing headers present. The SCP infers the target NF type from the path prefix (e.g., `/nudm-` → UDM) and performs delegated discovery.

3GPP Role and Spec References

Item	Reference
SCP NF definition	3GPP TS 23.501 Section 7.3
SCP indirect communication model	3GPP TS 29.500 Section 6.10
3gpp-Sbi-Discovery-* headers	3GPP TS 29.500 Section 6.10.3
3gpp-Sbi-Target-apiRoot header	3GPP TS 29.500 Section 6.10.3.2
3gpp-Sbi-Producer-Id header	3GPP TS 29.500 Section 6.10.3.3
SCP load balancing	3GPP TS 29.500 Section 6.10.4
NF discovery service	3GPP TS 29.510 Section 6.2
NRF NF Status Notification	3GPP TS 29.510 Section 6.3
SBI common framework	3GPP TS 29.500

SBI Endpoints

OmniSCP operates as a transparent proxy. There is only one locally-handled endpoint; all other paths are proxied to the appropriate NF producer.

Method	Path	Handled Locally	Description
`POST`	`/nnrf-nfm/v1/nf-status-notify`	Yes	Receives NRF NF status change notifications. On `NF_DEREGISTERED` or `NF_PROFILE_CHANGED` events, invalidates the entire discovery cache. Returns 204 No Content.
`*`	`/*` (all other paths)	No — proxied	Any method and path not matching the above is proxied to the resolved NF producer according to the active routing mode.

Proxy Error Responses

When the SCP cannot complete a proxy operation, it returns a `ProblemDetails` body per TS 29.500.

HTTP Status	Cause	Condition
400 Bad Request	`MANDATORY_IE_MISSING`	No routing information available: no `3gpp-Sbi-Target-apiRoot`, no discovery headers, and path cannot be mapped to a known service.
502 Bad Gateway	`TARGET_NF_NOT_REACHABLE`	All selected NF producer instances returned 5xx or connection errors after retries, or no SBI URI could be resolved for a discovered instance.
504 Gateway Timeout	`NF_DISCOVERY_FAILURE`	NRF discovery returned zero NF instances for the requested service.
500 Internal Server Error	`SYSTEM_FAILURE`	Unexpected internal error in the SCP proxy.

3gpp-Sbi Headers Consumed

Header	Description
`3gpp-Sbi-Target-apiRoot`	Direct routing target. Stripped before forwarding.
`3gpp-Sbi-Discovery-target-nf-type`	NF type to discover (e.g., `UDM`). Used for delegated discovery. Stripped before forwarding.
`3gpp-Sbi-Discovery-service-names`	Comma-separated list of service names. First value is used as the primary. Stripped before forwarding.
`3gpp-Sbi-Discovery-requester-nf-type`	Requester NF type for NRF query scoping. Stripped before forwarding.
`3gpp-Sbi-Discovery-target-plmn-list`	Target PLMN list. Passed to NRF discovery. Stripped before forwarding.
`3gpp-Sbi-Discovery-requester-snssai-list`	Requester S-NSSAI list. Passed to NRF discovery. Stripped before forwarding.
`3gpp-Sbi-Discovery-nf-set-id`	NF set ID filter for discovery. Stripped before forwarding.
`3gpp-Sbi-Discovery-target-nf-instance-id`	Specific NF instance ID to target. Stripped before forwarding.
`3gpp-Sbi-Discovery-requester-nf-instance-id`	Requester instance ID. Stripped before forwarding.

3gpp-Sbi Headers Produced

Header	Description
`3gpp-Sbi-Producer-Id`	Added to every proxied response. Contains the `nfInstanceId` of the NF producer that handled the request, enabling consumer-to-producer binding per TS 29.500 Section 6.10.3.3.

Configuration Reference

All parameters are set via the application environment (typically `config/runtime.exs`).

```
config :omniscp,
  sbi_scheme:            "http",
  sbi_addr:              "127.0.0.200",
  sbi_port:              7777,
  nrf_uri:               "http://127.0.0.10:7777",
  mcc:                   "999",
  mnc:                   "70",
  heartbeat_interval:    10_000,
  discovery_cache_ttl:   60_000,
  lb_strategy:           :round_robin,
  max_retries:           1,
  upstream_timeout:      5_000
```

Parameter Table

Parameter	Default	Type	Descript
`sbi_scheme`	`"http"`	string	Transport sc for the SBI listener.
`sbi_addr`	`"127.0.0.200"`	string	IP address t HTTP server to. NF consu must route S traffic to thi address.
`sbi_port`	`7777`	integer	TCP port the HTTP server listens on.
`nrf_uri`	`"http://127.0.0.10:7777"`	string	Base URI of NRF. Used fo registration, heartbeat, a discovery qu on behalf of consumers.
`mcc`	`"999"`	string	Mobile Coun Code. Includ the SCP NF registered w the NRF.
`mnc`	`"70"`	string	Mobile Netw Code. Includ the SCP NF

Parameter	Default	Type	Descript
			registered w the NRF.
`heartbeat_interval`	`10_000`	integer (ms)	Interval bet NRF heartbe requests.
`discovery_cache_ttl`	`60_000`	integer (ms)	Time-to-live NRF discove cache entrie keyed by `{target_nf` `service_nar` Expired entr are evicted on lookup a a backgrour cleanup tas every 30 se Increase for deployment decrease wh profiles cha frequently.

Parameter	Default	Type	Descript
`lb_strategy`	`:round_robin`	atom	Load balanc strategy for producer selection. Va values: `:round_rob` `:weighted`, `:priority`. Load Balanc section for semantics.
`max_retries`	`1`	integer	Maximum n of retry atte when an NF producer re 5xx or a connection A value of `1` means one original atte plus one ret to `0` to disa retries.
`upstream_timeout`	`5_000`	integer (ms)	Timeout for upstream H requests to producers (receive tim Requests th exceed this timeout are treated as f

Parameter	Default	Type	Descript
			and may tri retry.

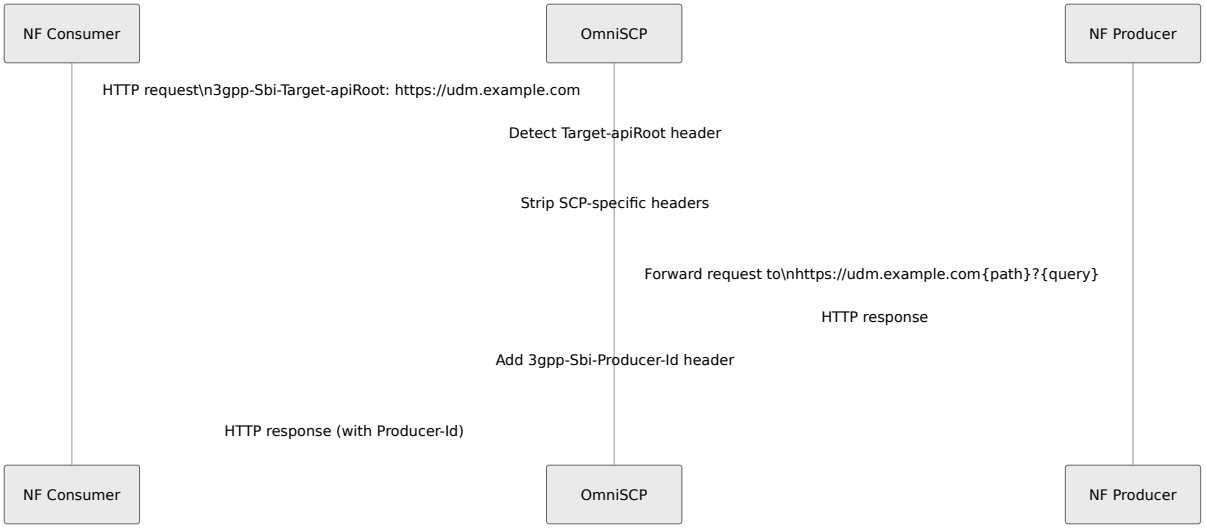
Load Balancing Strategies

Strategy	Description
`:round_robin`	Cycles through healthy instances in order. State is maintained per `{nf_type, service_name}` pair. This is the default and recommended strategy for uniform NF deployments.
`:weighted`	Selects the instance with the lowest `load - capacity` score. Uses the `load` and `capacity` fields from the NRF NF profile. Prefers instances with high capacity and low current load.
`:priority`	Selects the instance with the lowest `priority` value (highest priority). Useful for active/standby deployments.

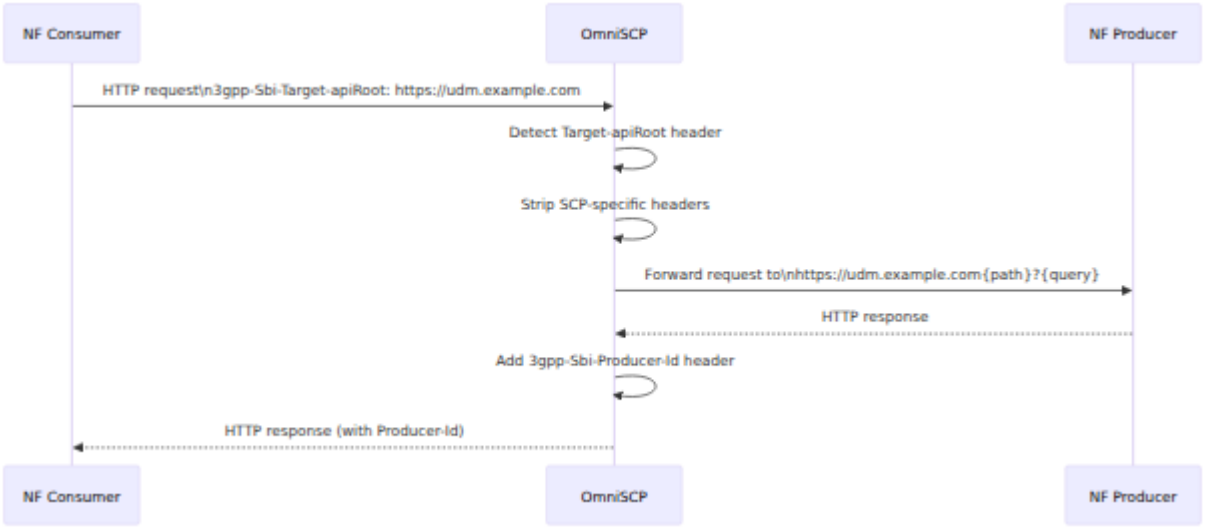
An instance is marked unhealthy after 3 consecutive failures and automatically recovers after a 30-second cooldown. When all instances are unhealthy, the load balancer falls back to the full instance list.

Key Procedures

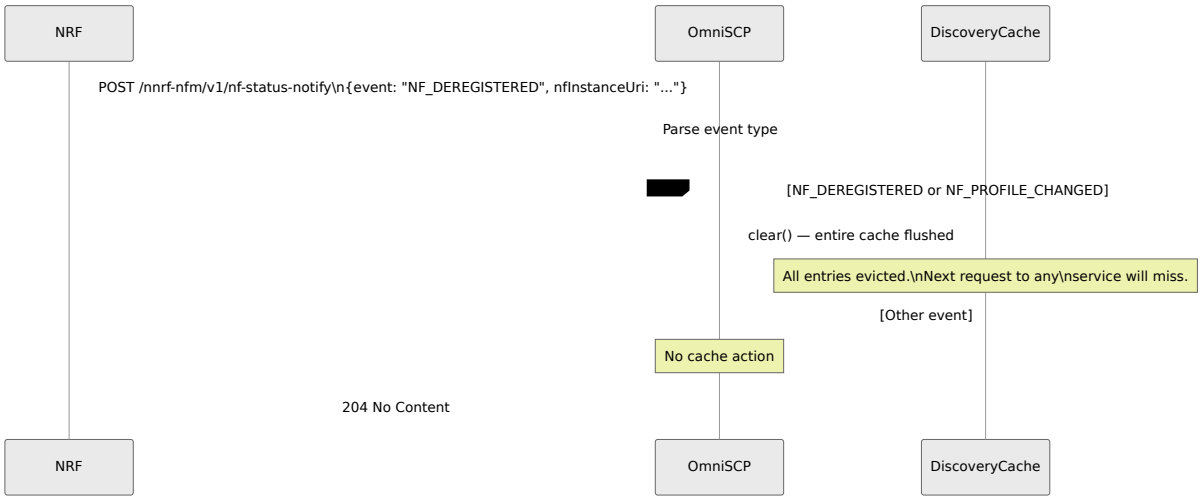
Direct Forwarding (Mode 1)



Delegated Discovery and Forwarding (Mode 2)



NRF Status Notification (Cache Invalidation)



Path-Based Service Inference (Mode 3)

When no routing headers are present, the SCP extracts the service name from the request path prefix and maps it to an NF type using the following built-in table:

Path Prefix	NF Type
`nudm-`	UDM
`nausf-`	AUSF
`namf-`	AMF
`nsmf-`	SMF
`npcf-`	PCF
`nudr-`	UDR
`nnssf-`	NSSF
`nbsf-`	BSF
`nnrf-`	NRF

Note: `nchf-`, `nnef-`, and `naf-` prefixes are not in the built-in map (limitation SCP-L1). Requests to CHF, NEF, or AF services require explicit discovery headers when using mode 3.

Observability

Telemetry Events

Event	Measurements	Tags	Description
`[:omniscp, :proxy, :requests]`	`count`, `duration_ms`	`target_nf_type`, `result`	Per-request proxy outcome
`[:omniscp, :proxy, :result]`	`count`, `duration_ms`	`target_nf_type`, `result`	Same event used for distribution histogram
`[:omniscp, :discovery, :cache]`	`hits`, `misses`	`target_nf_type`, `service_name`	Per-service cache hit/miss
`[:omniscp, :cache, :hit]`	`count`	—	Aggregate cache hit counter
`[:omniscp, :cache, :miss]`	`count`	—	Aggregate cache miss counter
`[:omniscp, :associations, :active]`	`count`	—	Gauge: active proxy associations
`[:omni5g, :nrf, :registration]`	`status`	`nf_type`	NRF registration status (1=registered, 0=not)

Result tag values: `success` (2xx/3xx), `client_error` (4xx), `server_error` (5xx), `error` (connection/timeout).

Prometheus Metrics

SCP Proxy Metrics

Metric	Type	Tags
`omni_scp.proxy.requests.count`	counter	`target_nf_type`, `result`
`omni_scp.proxy.requests.duration_ms`	summary	`target_nf_type`
`omni_scp.proxy_requests.total`	counter	`target_nf_type`, `result`
`omni_scp.proxy_request.duration_ms`	distribution	`target_nf_type`
`omni_scp.active_associations.count`	gauge	--

Cache Metrics

Metric	Type	Tags	Descrip
`omni_scp.discovery.cache.hits`	counter	`target_nf_type`, `service_name`	Per-servi cache hi
`omni_scp.discovery.cache.misses`	counter	`target_nf_type`, `service_name`	Per-servi cache misses
`omni_scp.cache_hits.total`	counter	--	Aggrega cache hi counter
`omni_scp.cache_misses.total`	counter	--	Aggrega cache m counter

NRF Metrics

Metric	Type	Tags	Description
`omni_scp.nrf.registration.status`	gauge	`nf_type`	NRF registration status (1=registered, 0=not)

BEAM VM Metrics

Metric	Type	Description
`beam.memory.total`	gauge	Total BEAM memory in bytes
`beam.memory.processes`	gauge	Memory used by Erlang processes
`beam.memory.processes_used`	gauge	Memory actually used by processes
`beam.memory.system`	gauge	System memory
`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 in seconds

Log Patterns

Level	Pattern	Meaning
`info`	`Received NRF status notification`	NF status notify received
`info`	`NRF notification: event=<E> nf=<URI>`	Parsed notification event
`debug`	`SCP direct forward: <METHOD> <URL>`	Mode 1 forwarding
`debug`	`SCP delegated forward: <METHOD> <URL> (attempt <N>)`	Mode 2/3 forwarding attempt
`warning`	`SCP retrying after <STATUS> from <ID>...`	Retry triggered by 5xx
`warning`	`SCP retrying after error from <ID>...`	Retry triggered by connection error
`warning`	`SCP cannot determine target for <METHOD> <PATH>`	Mode 3 path not in service map
`warning`	`NRF discovery returned no instances for <NF>/<SVC>`	Discovery returned empty list
`warning`	`All NF instances unhealthy, falling back to full list`	LB health fallback
`error`	`NRF discovery failed: ...`	NRF query error
`error`	`SCP proxy error: ...`	Unexpected proxy failure

Level	Pattern	Meaning
`info`	`NF instance &lt;ID> recovered after cooldown`	Instance health restored

Known Limitations

ID	Severity	Description
SCP-H4	High	ECIES SUCI de-concealment is not implemented in the underlying `Omni5gEx` shared library. Requests that require the SCP to de-conceal a SUCI before forwarding (used in some indirect communication models for AUSF/UDM) will be forwarded without de-concealment. This does not affect most SBA deployments where de-concealment is done by the AUSF.
SCP-M1	Medium	Overload control is not implemented. The `3gpp-Sbi-Oci` (Overload Control Information) header is not generated or consumed. In overload scenarios the SCP will continue forwarding requests without shedding load or back-pressuring consumers.
SCP-M2	Medium	Load control indication is not implemented. The `3gpp-Sbi-Lci` (Load Control Information) header is not generated. Consumers cannot use OmniSCP to obtain NF load hints for their own load control decisions.
SCP-L1	Low	The path-based service inference map (Mode 3) is missing `nchf-` (CHF), `nnef-` (NEF), and `naf-` (AF) prefix entries. Requests to these services without explicit discovery headers will receive a 400 Bad Request with cause `MANDATORY_IE_MISSING`. Workaround: configure consumers to send `3gpp-Sbi-Discovery-*` headers for these services.
SCP-L3	Low	NRF status notifications with event `NF_DEREGISTERED` or `NF_PROFILE_CHANGED` clear the entire discovery cache rather than only the affected `{nf_type, service_name}` entry. In deployments with frequent NF profile changes, this causes a burst of NRF re-discovery queries.

Troubleshooting

400 Bad Request — MANDATORY_IE_MISSING

The SCP could not determine a routing target. Check:

1. Is the consumer sending `3gpp-Sbi-Target-apiRoot` or both `3gpp-Sbi-Discovery-target-nf-type` and `3gpp-Sbi-Discovery-service-names`?
2. If relying on path-based inference (Mode 3), does the path prefix appear in the built-in service map? Note that `nchf-`, `nnef-`, and `naf-` are absent (SCP-L1). Add explicit headers for those services.

504 Gateway Timeout — NF_DISCOVERY_FAILURE

NRF returned no NF instances. Check:

1. Is the NRF reachable from OmniSCP? Verify `nrf_uri` and network connectivity.
2. Is the target NF type registered in the NRF? Query the NRF directly: `GET {nrf_uri}/nnrf-disc/v1/nf-instances?target-nf-type=<TYPE>`.
3. Check if an NRF status notification just cleared the cache (`NF_DEREGISTERED` event) and the NF has not re-registered.

502 Bad Gateway — TARGET_NF_NOT_REACHABLE

All NF producer instances failed. Check:

1. Are the NF producers running and reachable on the SBI addresses reported in their NRF profiles?
2. Check `upstream_timeout`. If NF producers are slow to respond, increase this value.
3. Review `max_retries`. If set to `0`, a single failure becomes an immediate 502.

4. Check load balancer health status in logs: look for `NF instance <ID> marked unhealthy after N failures`.

Discovery cache causing stale routing

If NF producers change addresses or restart without proper NRF deregistration, the cache may hold stale SBI URIs until the TTL expires. Options:

1. Reduce `discovery_cache_ttl` to limit the staleness window.
2. Ensure NF producers deregister from the NRF on shutdown; this triggers an NRF status notification that clears the OmniSCP cache.
3. A process restart of OmniSCP clears all cache state.

High proxy latency

1. Check `omni_scp.proxy_request.duration_ms` histogram for latency distribution.
2. Compare cache hit rate (`omni_scp.cache_hits.total` vs `omni_scp.cache_misses.total`). High miss rate means frequent NRF queries. Increase `discovery_cache_ttl`.
3. Check `upstream_timeout` — requests that timeout add the full timeout duration to latency before triggering a retry.

NRF registration not maintained

Check `omni_scp.nrf.registration.status` metric. If it reads 0:

1. Verify `nrf_uri` is correct and the NRF is reachable.
2. Check `mcc` and `mnc` match the NRF PLMN configuration.
3. Look for NRF registration errors in application logs at startup.