


1. `3gpp-Sbi-Target-apiRoot`  URI `3gpp-Sbi-Target-apiRoot` NRF `3gpp-Sbi-Target-apiRoot`
2. `3gpp-Sbi-Discovery-target-nf-type` `3gpp-Sbi-Discovery-service-names` SCP NRF `3gpp-Sbi-Discovery-target-nf-type` `3gpp-Sbi-Discovery-service-names`
3. SCP NF `/nudm-` → UDM

3GPP

Item	Reference
SCP NF	3GPP TS 23.501 7.3
SCP	3GPP TS 29.500 6.10
<code>3gpp-Sbi-Discovery-*</code>	3GPP TS 29.500 6.10.3
<code>3gpp-Sbi-Target-apiRoot</code>	3GPP TS 29.500 6.10.3.2
<code>3gpp-Sbi-Producer-Id</code>	3GPP TS 29.500 6.10.3.3
SCP	3GPP TS 29.500 6.10.4
NF	3GPP TS 29.510 6.2
NRF NF	3GPP TS 29.510 6.3
SBI	3GPP TS 29.500

SBI

OmniSCP `3gpp-Sbi-Target-apiRoot` NRF

Method	Path	Response Code	Description
POST	/nnrf-nfm/v1/nf-status-notify	204	NRF NF status notification. Response codes: NF_DEREGISTERED, NF_PROFILE_CHANGED. Status: 204 OK.
*	/* (Wildcard)	—	Wildcard path for all other endpoints.

Response Codes

SCP ProblemDetails (TS 29.500) ProblemDetails

HTTP Code	ProblemDetails Code	Description
400 Bad Request	MANDATORY_IE_MISSING	Missing mandatory IE in 3gpp-Sbi-Target-apiRoot.
502 Bad Gateway	TARGET_NF_NOT_REACHABLE	Target NF not reachable (5xx) SBI URI.
504 Gateway Timeout	NF_DISCOVERY_FAILURE	NRF discovery failure for NF.
500 Internal Server Error	SYSTEM_FAILURE	SCP system failure.

3gpp-Sbi

Field	Description
3gpp-Sbi-Target-apiRoot	API Root
3gpp-Sbi-Discovery-target-nf-type	Target NF type (e.g., UDM)
3gpp-Sbi-Discovery-service-names	Service names
3gpp-Sbi-Discovery-requester-nf-type	NRF type
3gpp-Sbi-Discovery-target-plmn-list	Target PLMN list
3gpp-Sbi-Discovery-requester-snsai-list	Requester S-NSSAI list
3gpp-Sbi-Discovery-nf-set-id	NF set ID
3gpp-Sbi-Discovery-target-nf-instance-id	Target NF instance ID
3gpp-Sbi-Discovery-requester-nf-instance-id	Requester NF instance ID

3gpp-Sbi

Field	Description
3gpp-Sbi-Producer-Id	Producer ID (e.g., TS 29.500 6.10.3.3)



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
```

□□□

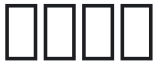
□□	□□□	□□	
<code>sbi_scheme</code>	<code>"http"</code>	□□ □	SBI □□□□□□□□□□
<code>sbi_addr</code>	<code>"127.0.0.200"</code>	□□ □	SBI HTTP □□□□□□□□ □□
<code>sbi_port</code>	<code>7777</code>	□□	SBI HTTP □□□□□□□□
<code>nrf_uri</code>	<code>"http://127.0.0.10:7777"</code>	□□ □	NRF □□□□ URI□□□□
<code>mcc</code>	<code>"999"</code>	□□ □	□□□□□□□□□□ NR
<code>mnc</code>	<code>"70"</code>	□□ □	□□□□□□□□□□ NR
<code>heartbeat_interval</code>	<code>10_000</code>	□□ □□ □□	NRF □□□□□□□□□□
<code>discovery_cache_ttl</code>	<code>60_000</code>	□□ □□ □□	NRF □□□□□□□□□□ <code>service_name</code> } [□ 30 □□□□□□□□□□
<code>lb_strategy</code>	<code>:round_robin</code>	□□	NF □□□□□□□□□□ □□ :round_robin □□□□□□□□□□□□□□
<code>max_retries</code>	<code>1</code>	□□	□ NF □□□□□ 5xx [□ □□□□□□□□□□□ □ 0 [□

項目	設定値	単位	説明
<code>upstream_timeout</code>	<code>5_000</code>	ms	NF への HTTP 接続のタイムアウト時間

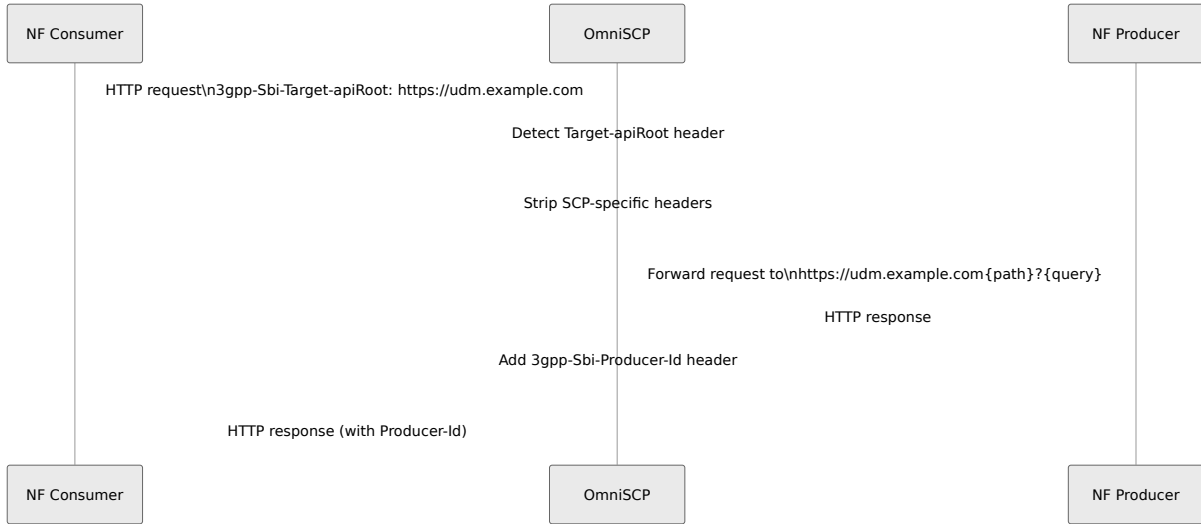
ロードバランシング

項目	説明
<code>:round_robin</code>	ラウンドロビン方式で、 <code>{nf_type, service_name}</code> の各サービスに対して均等にリクエストを振り分ける。N/A
<code>:weighted</code>	各サービスの <code>load - capacity</code> を基準として、NRF 上の NF の <code>load</code> と <code>capacity</code> の比率に基づいてリクエストを振り分ける。
<code>:priority</code>	各サービスの <code>priority</code> を基準として、優先度の高いサービスから順にリクエストを振り分ける。

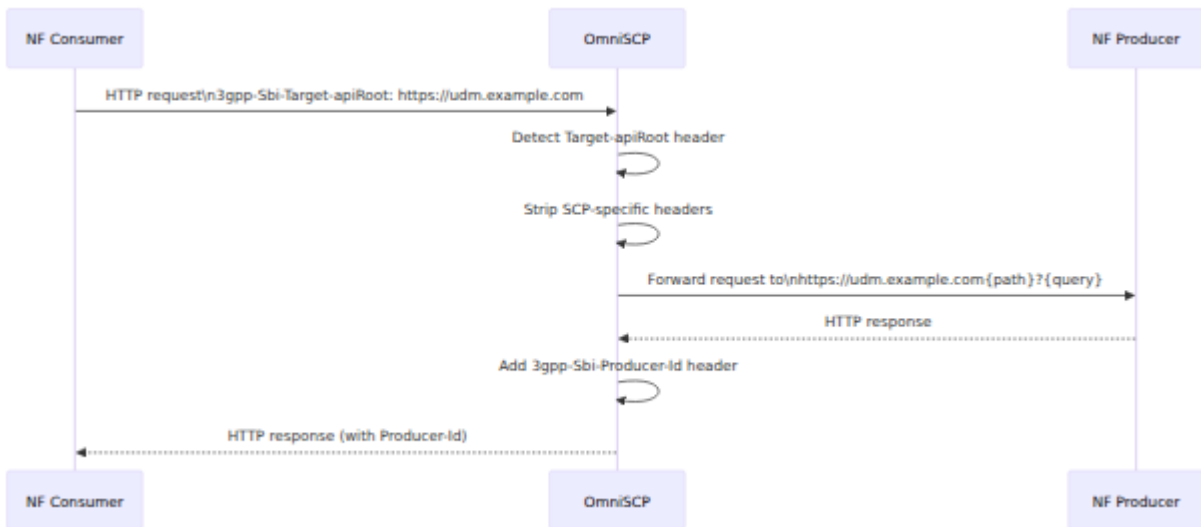
デフォルトは `round_robin` であり、`capacity` の値は 30 である。



□□□□□□□□ 1□



□□□□□□□□□□ 2□



nchf- nnef- naf- SCP-L1 CHF NEF AF 3

□□□□

□□□□

□□	□□	□□	□□
<code>[:omniscp, :proxy, :requests]</code>	<code>count, duration_ms</code>	<code>target_nf_type, result</code>	□□□□□□ □□
<code>[:omniscp, :proxy, :result]</code>	<code>count, duration_ms</code>	<code>target_nf_type, result</code>	□□□□□□ □□□□□□
<code>[:omniscp, :discovery, :cache]</code>	<code>hits, misses</code>	<code>target_nf_type, service_name</code>	□□□□□□ □□/□□□
<code>[:omniscp, :cache, :hit]</code>	<code>count</code>	—	□□□□□□ □□
<code>[:omniscp, :cache, :miss]</code>	<code>count</code>	—	□□□□□□ □□□
<code>[:omniscp, :associations, :active]</code>	<code>count</code>	—	□□□□□□ □□
<code>[:omni5g, :nrf, :registration]</code>	<code>status</code>	<code>nf_type</code>	NRF □□□□ (1=□□□, 0=□□□)

□□□□□ `success` (2xx/3xx), `client_error` (4xx), `server_error` (5xx), `error` (□□/□□)

Prometheus

SCP

メトリック名	タイプ	ラベル
<code>omni_scp.proxy.requests.count</code>	counter	target_nf_type, result
<code>omni_scp.proxy.requests.duration_ms</code>	summary	target_nf_type
<code>omni_scp.proxy_requests.total</code>	counter	target_nf_type, result
<code>omni_scp.proxy_request.duration_ms</code>	distribution	target_nf_type
<code>omni_scp.active_associations.count</code>	gauge	--

完了

メトリック名	タイプ	ラベル	単位
<code>omni_scp.discovery.cache.hits</code>	counter	<code>target_nf_type,</code> <code>service_name</code>	個 個 個 個 個
<code>omni_scp.discovery.cache.misses</code>	counter	<code>target_nf_type,</code> <code>service_name</code>	個 個 個 個 個
<code>omni_scp.cache_hits.total</code>	counter	--	個 個 個 個 個
<code>omni_scp.cache_misses.total</code>	counter	--	個 個 個 個 個

NRF

メトリック名	タイプ	ラベル	説明
<code>omni_scp.nrf.registration.status</code>	gauge	<code>nf_type</code>	NRF ステータス (1=登録済み, 0=未登録)

BEAM VM

名前	タイプ	説明
<code>beam.memory.total</code>	gauge	BEAM の総メモリ使用量
<code>beam.memory.processes</code>	gauge	Erlang のプロセス数
<code>beam.memory.processes_used</code>	gauge	プロセスに使用されているメモリ
<code>beam.memory.system</code>	gauge	システムメモリ
<code>beam.memory.atom</code>	gauge	アトムメモリ
<code>beam.memory.atom_used</code>	gauge	アトムに使用されているメモリ
<code>beam.memory.binary</code>	gauge	バイナリメモリ
<code>beam.memory.code</code>	gauge	コードメモリ
<code>beam.memory.ets</code>	gauge	ETS のメモリ
<code>beam.processes.count</code>	gauge	Erlang のプロセス数
<code>beam.ports.count</code>	gauge	Erlang のポート数
<code>beam.atom.count</code>	gauge	アトムの数
<code>beam.vm.uptime</code>	gauge	VM の稼働時間

□□□□

□□	□□	□□
info	Received NRF status notification	□□ NF □□□□
info	NRF notification: event=<E> nf=<URI>	□□□□□□□□
debug	SCP direct forward: <METHOD> <URL>	□□ 1 □□
debug	SCP delegated forward: <METHOD> <URL> (attempt <N>)	□□ 2/3 □□□□
warning	SCP retrying after <STATUS> from <ID>...	□□ 5xx □ ◆◆◆□□
warning	SCP retrying after error from <ID>...	□□□□□□□□□□
warning	SCP cannot determine target for <METHOD> <PATH>	□□ 3 □□□□□□ □□□
warning	NRF discovery returned no instances for <NF>/<SVC>	□□□□□□□□
warning	All NF instances unhealthy, falling back to full list	LB □□□□
error	NRF discovery failed: ...	NRF □□□□
error	SCP proxy error: ...	□□□□□□□□
info	NF instance <ID> recovered after cooldown	□□□□□□

2. `nchf-` `nnef-` `naf-` SCP-L1

504 Gateway Timeout — NF_DISCOVERY_FAILURE

NRF `nf` `nf`

1. NRF `OmniSCP` `nrf_uri`
2. `GET {nrf_uri}/nnrf-disc/v1/nf-instances?target-nf-type=<TYPE>`
3. NRF `NF_DEREGISTERED` `nf`

502 Bad Gateway — TARGET_NF_NOT_REACHABLE

`nf` `nf`

1. NRF `SBI`
2. `upstream_timeout` `nf`
3. `max_retries` `0` `502`
4. `NF instance <ID> marked unhealthy after N failures`

`nf` `nf`

`nf` `nf` `SBI URI` `TTL`

1. `discovery_cache_ttl`
2. `NF` `NRF` `NRF` `OmniSCP`
3. `OmniSCP`

`nf` `nf`

1. `omni_scp.proxy_request.duration_ms`

2. `omni_scp.cache_hits.total` と `omni_scp.cache_misses.total` の差分を監視し、NRF の `discovery_cache_ttl` を確認する。
3. `upstream_timeout` が設定されているかどうかを確認する。

NRF の監視

`omni_scp.nrf.registration.status` の監視を行う。

1. `nrf_uri` が NRF の URI であるかを確認する。
2. `mcc` と `mnc` が NRF の PLMN であるかを確認する。
3. NRF の登録状態を確認する。