OmniSEPP Configuration Reference

Overview | Operations Guide | Metrics and Monitoring | Troubleshooting

All OmniSEPP configuration lives under the `:omnisepp` application key. A subset of keys can be overridden by environment variables, but that only takes effect in the **production runtime configuration**. The nested maps (`roaming_partners`, `topology_hiding_map`) and a small number of other keys have no environment variable and are set through the application config file only.

Configuration Structure

```
config :omnisepp,
  # SBI/N32 listener identity (advertised to the NRF)
  sbi_scheme: "http",
  sbi_addr: "127.0.0.22",
  sbi_port: 7777,

  # TLS material for the N32 listener (used only when sbi_scheme
is "https")
  sbi_certfile: nil,
  sbi_keyfile: nil,
  sbi_cacertfile: nil,

  # NRF registration
  nrf_uri: "http://127.0.0.10:7777",
  heartbeat_interval: 10_000,

  # Served PLMN (override per deployment)
  mcc: "999",
  mnc: "70",

  # This SEPP's own identity and negotiated capabilities
  sepp_fqdn: "sepp.5gc.mnc70.mcc999.3gppnetwork.org",
  supported_sec_capabilities: ["TLS"],

  # N32 association context persistence (config-file only)
  n32_context_path: "<priv>/data/n32_context.dets",

  # Roaming and topology hiding (config-file only)
  roaming_partners: %{},
  topology_hiding_map: %{}
```

Every parameter above is documented below.

Core Parameters

Parameter	Type	Required	Default
`sbi_scheme`	String	No	`http`
`sbi_addr`	String	No	`127.0.0.22`
`sbi_port`	Integer	No	`7777`
`sbi_certfile`	String (path)	Conditional	(none)

Parameter	Type	Required	Default
`sbi_keyfile`	String (path)	Conditional	(none)
`sbi_cacertfile`	String (path)	No	(none)
`nrf_uri`	String	No	`http://127.0.0.10:`

Parameter	Type	Required	Default
`mcc`	String	No	`999`
`mnc`	String	No	`70`
`heartbeat_interval`	Integer (ms)	No	`10000`
`sepp_fqdn`	String	No	derived (see below)

Parameter	Type	Required	Default
`supported_sec_capabilities`	List of String	No	`["TLS"]`
`n32_context_path`	String (path)	No	derived (see below)

Parameter	Type	Required	Default
`roaming_partners`	Map	No	`%{}`
`topology_hiding_map`	Map	No	`%{}`

sepp_fqdn

When `sepp_fqdn` is not set, it is derived from the served PLMN as:

```
sepp.5gc.mnc<mnc>.mcc<mcc>.3gppnetwork.org
```

For the default PLMN `999-70` this yields `sepp.5gc.mnc70.mcc999.3gppnetwork.org`. This FQDN is sent as the `sender` identity in N32-c handshake requests and responses (TS 29.573). Override it explicitly when the derived form does not match the FQDN your roaming partners expect.

N32 Listener TLS

When `sbi_scheme` is `https`, the N32 listener terminates TLS itself rather than relying on an external terminator. This is N32-f TLS mode (TS 33.501 §13.1), in which two SEPPs protect the interface with TLS between them.

- `sbi_certfile` and `sbi_keyfile` are **both required** when `sbi_scheme` is `https`. If either is missing the SEPP refuses to start rather than silently downgrading to plaintext.
- When `sbi_cacertfile` is also set, the listener requires and verifies the peer SEPP's client certificate against that CA, giving mutually-authenticated TLS. With no CA configured, only server-side TLS is terminated.
- When `sbi_scheme` is `http` (the default) these keys are ignored and the listener serves plaintext.

The advertised scheme in NRF registration follows `sbi_scheme`, so it now reflects the actual transport of the listener.

supported_sec_capabilities

The N32-f security capabilities this SEPP supports, in **local preference order** (most-preferred first). During the N32-c handshake, OmniSEPP selects the first entry in this list that the peer also offers (TS 29.573 §5.2.3.2.2).

Value	Meaning	Reference
`TLS`	Transport-layer protection between SEPPs.	TS 33.501 §13.1
`PRINS`	PRotection with Insertion of iNtermediate Security (JOSE application-layer protection via a roaming hub).	TS 33.501 §13.2

Only these two values are recognised; any other entry is ignored. The default is `["TLS"]`; keep it at `["TLS"]` for interoperable operation.

When overriding via the `SEPP_SEC_CAPABILITIES` environment variable, provide a comma-separated list in preference order, for example `TLS,PRINS`. Whitespace around each entry is trimmed.

N32 Context Persistence

Established N32 associations are persisted to a DETS file at `n32_context_path`, so they survive a process or node restart (TS 29.573). On start-up the store reloads the persisted associations, so a partner does not have to re-run the N32-c handshake after an OmniSEPP restart.

- When unset, the path defaults to `n32_context.dets` under the OTP application's `priv` data directory. The parent directory is created on start-up if it does not exist.
- If the file cannot be opened (for example on corruption), the store logs a warning and starts with an empty in-memory table rather than crashing boot; associations are then re-established by handshake.
- This key has no environment variable and is set through the application config file only.

Roaming Partners

The `roaming_partners` map is the static allow-list of roaming partner PLMNs. It is keyed by the partner PLMN identifier in `"mcc-mnc"` form, and each value is a partner specification.

```
config :omnisepp,
  roaming_partners: %{
    "001-01" => %{
      sepp_uri: "https://sepp.partner-a.example:7777",
      sec_capabilities: ["TLS"]
    },
    "310-260" => %{
      sepp_uri: "https://sepp.partner-b.example:7777"
    }
  }
```

Top-level structure:

Key form	Type	Description
`"mcc-mnc"`	String (map key)	Partner PLMN identifier, for example `"001-01"`.
value	Map	Partner specification (see below).

Partner specification:

Parameter	Type	Required	Default	
`sepp_uri`	String	Yes	-	T (c r
`sec_capabilities`	List of String	No	inherits `supported_sec_capabilities`	N p W i s

An empty `roaming_partners` map (the default) means no roaming partners are configured, and N32-f forwarding refuses all unknown PLMNs. This map has no environment variable and is set through the application config file only.

Topology Hiding Map

The `topology_hiding_map` implements FQDN rewriting for topology hiding (TS 33.501 §13.2). It maps each internal NF FQDN to the external FQDN presented to roaming partners.

```
config :omnisepp,
  topology_hiding_map: %{
    "amf1.5gc.mnc070.mcc999.3gppnetwork.org" =>
      "sepp.5gc.mnc070.mcc999.3gppnetwork.org",
    "udm1.5gc.mnc070.mcc999.3gppnetwork.org" =>
      "sepp.5gc.mnc070.mcc999.3gppnetwork.org"
  }
```

Key form	Type	Description
`internal_fqdn`	String (map key)	The internal NF FQDN to be masked, for example an AMF or UDM FQDN.
`external_fqdn`	String (value)	The external FQDN presented to partners in its place.

The mapping is applied outbound (internal to external) on the `3gpp-Sbi-Target-apiRoot`, `Location`, and producer-id header values, and reversed inbound (external to internal) so that partner-presented apiRoots resolve back to the real internal FQDN. An empty map (the default) means no rewriting. This map has no environment variable and is set through the application config file only.

Environment Variable Summary

Environment variables apply only in the production runtime configuration. Keys not listed here are config-file only.

Environment Variable	Parameter	Notes
`SEPP_SBI_SCHEME`	`sbi_scheme`	`http` or `https`.
`SEPP_SBI_ADDR`	`sbi_addr`	
`SEPP_SBI_PORT`	`sbi_port`	Parsed as an integer.
`SEPP_SBI_CERTFILE`	`sbi_certfile`	Required when scheme is `https`.
`SEPP_SBI_KEYFILE`	`sbi_keyfile`	Required when scheme is `https`.
`SEPP_SBI_CACERTFILE`	`sbi_cacertfile`	Set to enable N32-f mutual TLS (peer client-cert verification).
`SEPP_NRF_URI`	`nrf_uri`	
`SEPP_MCC`	`mcc`	
`SEPP_MNC`	`mnc`	
`SEPP_FQDN`	`sepp_fqdn`	When unset, the FQDN is derived from the PLMN.
`SEPP_SEC_CAPABILITIES`	`supported_sec_capabilities`	Comma-separated,

Environment Variable	Parameter	Notes
		preference order.

Configuration Example

A realistic per-deployment override for a home PLMN `310-260` with a single roaming partner:

```
config :omnisepp,
  sbi_scheme: "https",
  sbi_addr: "10.20.0.5",
  sbi_port: 7777,
  sbi_certfile: "/etc/omnisepp/tls/sepp.crt",
  sbi_keyfile: "/etc/omnisepp/tls/sepp.key",
  sbi_cacertfile: "/etc/omnisepp/tls/partner-ca.crt",
  nrf_uri: "http://10.20.0.1:7777",
  heartbeat_interval: 10_000,
  mcc: "310",
  mnc: "260",
  sepp_fqdn: "sepp.5gc.mnc260.mcc310.3gppnetwork.org",
  supported_sec_capabilities: ["TLS"],
  roaming_partners: %{
    "001-01" => %{
      sepp_uri: "https://sepp.partner-a.example:7777",
      sec_capabilities: ["TLS"]
    }
  },
  topology_hiding_map: %{
    "amf1.5gc.mnc260.mcc310.3gppnetwork.org" =>
      "sepp.5gc.mnc260.mcc310.3gppnetwork.org"
  }
```

How it works: OmniSEPP advertises itself to the NRF at `10.20.0.5:7777` as NFType `SEPP` for PLMN `310-260`, heartbeating every 10 seconds. Because `sbi_scheme` is `https`, the N32 listener terminates TLS with `sepp.crt` / `sepp.key`, and because `sbi_cacertfile` is set it requires and verifies the

partner SEPP's client certificate against `partner-ca.crt` (mutual TLS). It accepts N32-c handshakes as `sepp.5gc.mnc260.mcc310.3gppnetwork.org`, negotiating TLS as the sole N32-f capability, and at start-up initiates the outbound handshake toward partner `001-01`. N32-f traffic is relayed only to and from partner PLMN `001-01`, and the internal AMF FQDN is masked behind the SEPP FQDN toward that partner.

OmniSEPP Metrics and Monitoring

Observability

Observability is provided by **structured application logs** and standard request logging. Operators monitor OmniSEPP through its logs, which capture the handshake, forwarding, and request events described below.

What to Watch in the Logs

The application logs are the primary signal for operating OmniSEPP. The following classes of events are logged and are worth watching.

N32-c Handshake Outcomes

Each capability negotiation logs its outcome, including the remote SEPP identity and the selected N32-f security capability on success. Watch these to confirm that:

- Expected roaming partners are completing the handshake.
- The selected capability matches your intent (for example `TLS`).
- A partner is being rejected because no common capability could be selected.

A peer that selects a capability this SEPP did not offer (on the initiator path) is logged as a warning and rejected.

N32-f Forwarding Decisions and Failures

The N32-f relay logs its forwarding decisions and failures, including:

- Refusals to forward to a PLMN that is not a configured roaming partner. A rising rate of these suggests a missing or misconfigured `roaming_partners` entry.
- Inbound and outbound relay targets (the resolved URL and direction).
- Forward failures, which surface to the partner as `502 TARGET_NF_NOT_REACHABLE` and indicate the addressed internal NF was unreachable.

Request Logging

Standard request logging covers each SBI request handled by the listener, including the handshake endpoints and the N32-f catch-all relay. Use it to observe request volume, methods, and paths, and to correlate partner activity with handshake and forwarding events.

Health Checking

Health is assessed by combining:

- **Liveness:** `GET /n32c-handshake/v1/status` returns `200 {"status":"UP"}` when the SBI listener is serving. See the Operations Guide.
- **NRF registration:** confirm OmniSEPP appears in the NRF as an `SEPP` NF instance and is heartbeating. Registration behaviour is described in the Operations Guide.
- **Log signals:** the handshake and forwarding events described above.

OmniSEPP Operations Guide

Overview | Configuration Reference | Metrics and Monitoring | Troubleshooting

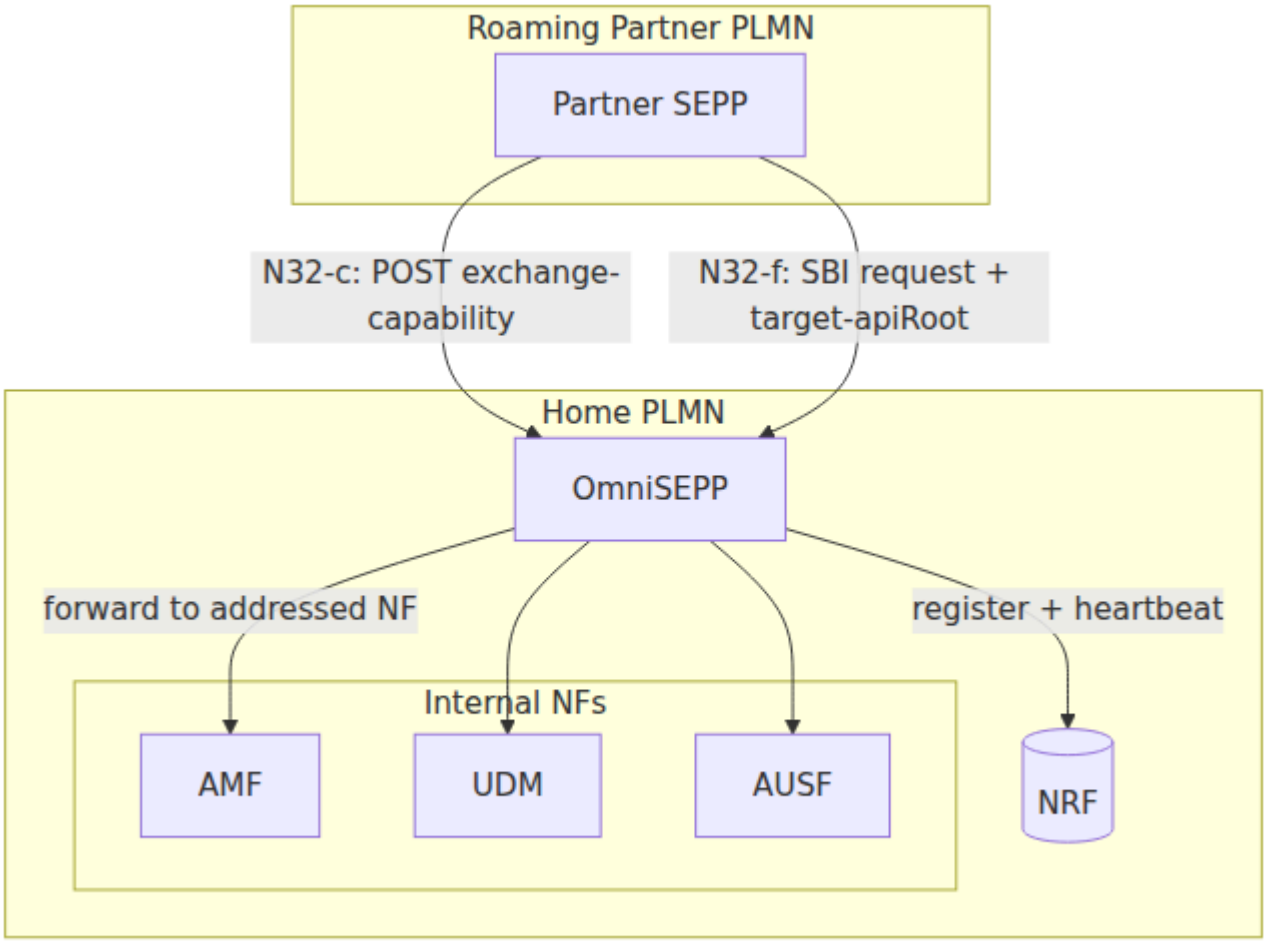
This guide describes how OmniSEPP behaves on the N32 inter-PLMN edge and how to operate it. OmniSEPP is the Security Edge Protection Proxy defined in 3GPP TS 23.501 §6.2.17, with security behaviour from 3GPP TS 33.501 and the N32 interface from 3GPP TS 29.573.

Table of Contents

- Architecture Overview
- The N32 Interface
- N32 Association Lifecycle
- Capability Negotiation
- Roaming-Partner Gating
- Topology Hiding
- Message Flows
- SBI Endpoints
- NRF Registration
- Operational Notes

Architecture Overview

OmniSEPP terminates the N32 interface toward roaming-partner SEPPs and relays traffic to internal network functions in the home PLMN. It registers with the local NRF as NFType `SEPP` so local consumers and the SCP can discover it; the remote peer SEPP and the topology-hiding map are static configuration per roaming agreement (peer SEPPs are not discovered via NRF).



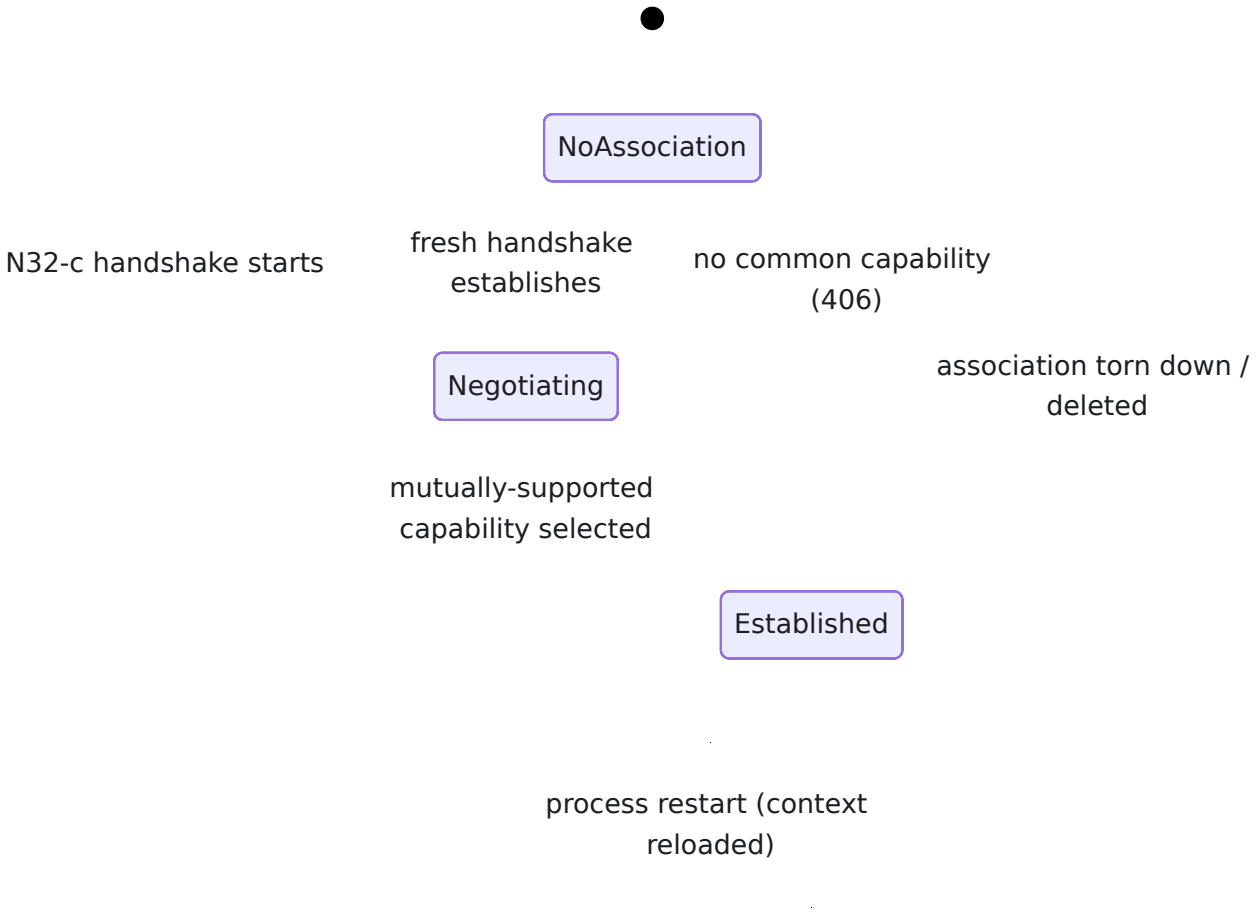
The N32 Interface

N32 has two logical channels (TS 29.573):

- **N32-c (control channel)** - The handshake used to negotiate the N32-f security capability and establish the association between the two SEPPs. OmniSEPP exposes this as the `/n32c-handshake/v1/exchange-capability` endpoint (responder role) and also initiates the handshake toward each configured roaming partner at start-up (initiator role).
- **N32-f (forwarding channel)** - The ongoing exchange of service-based (SBI) messages between the two PLMNs. OmniSEPP relays these in **TLS mode**: the message is forwarded largely verbatim, with hop-by-hop header hygiene and topology hiding applied.

N32 Association Lifecycle

Each successful N32-c negotiation records an association in a **persistent context store** (a DETS table fronted by an in-memory cache), keyed by the remote SEPP FQDN. The stored context includes the selected N32-f security capability and the partner PLMN identity list.



Key operational properties:

- The store is **persistent**: associations are written to the DETS file at `n32_context_path` and reloaded on start-up, so they survive a process or node restart. A partner does not have to re-handshake after an OmniSEPP restart.
- There is **no TTL**: an association persists until it is explicitly torn down (deleted) or replaced by a fresh negotiation.
- If the DETS file cannot be opened (for example on corruption), the store starts empty and logs a warning; associations are then re-established by handshake.

Capability Negotiation

During the N32-c handshake the two SEPPs agree on a single N32-f security capability. Valid capabilities are `TLS` and `PRINS` (TS 33.501 §13.1).

The selection rule (TS 29.573 §5.2.3.2.2) is **local-preference-first**: OmniSEPP walks its own configured `supported_sec_capabilities` list in order and picks the first entry the peer also offers. If the two capability sets are disjoint, the negotiation fails with `INSUFFICIENT_RESOURCES` (HTTP 406) and no association is recorded.

Only well-formed, recognised capabilities (`TLS`, `PRINS`) are advertised. Keep `supported_sec_capabilities` at `["TLS"]` for interoperable operation. See the Configuration Reference for how to set the preference order.

Roaming-Partner Gating

The **roaming-partner registry** is a static allow-list keyed by partner PLMN (`"mcc-mnc"`). It governs N32-f forwarding:

- Only PLMNs present in the registry are relayed to or from.
- An empty registry (the default) means N32-f forwarding refuses all unknown PLMNs.
- Each partner entry carries the partner SEPP API root and, optionally, the N32-f security capabilities permitted with that partner. When the capability list is omitted, the partner inherits this SEPP's `supported_sec_capabilities`.

See Configuration Reference - Roaming Partners for the map structure.

Topology Hiding

Topology hiding (TS 33.501 §13.2) masks internal NF identities from roaming partners by rewriting FQDNs. OmniSEPP applies an operator-configured map of `internal_fqdn => external_fqdn`:

- **Outbound** (toward a partner): internal FQDNs in the `3gpp-Sbi-Target-apiRoot`, `Location`, and producer-id header values are rewritten to their external equivalents.
- **Inbound** (from a partner): the reverse mapping is applied so that the externally-presented apiRoot resolves back to the real internal FQDN before the request is routed to the internal NF.

Rewrites are anchored to **whole FQDN tokens**: a mapped FQDN is rewritten only when it appears as a complete host (bounded by URL delimiters such as `/`, `:`, `@`, or the end of the value), never as a substring of a longer hostname. This prevents a mapping from corrupting an unrelated host that merely contains the mapped name, and stops a value one mapping produced from being re-matched by another mapping entry.

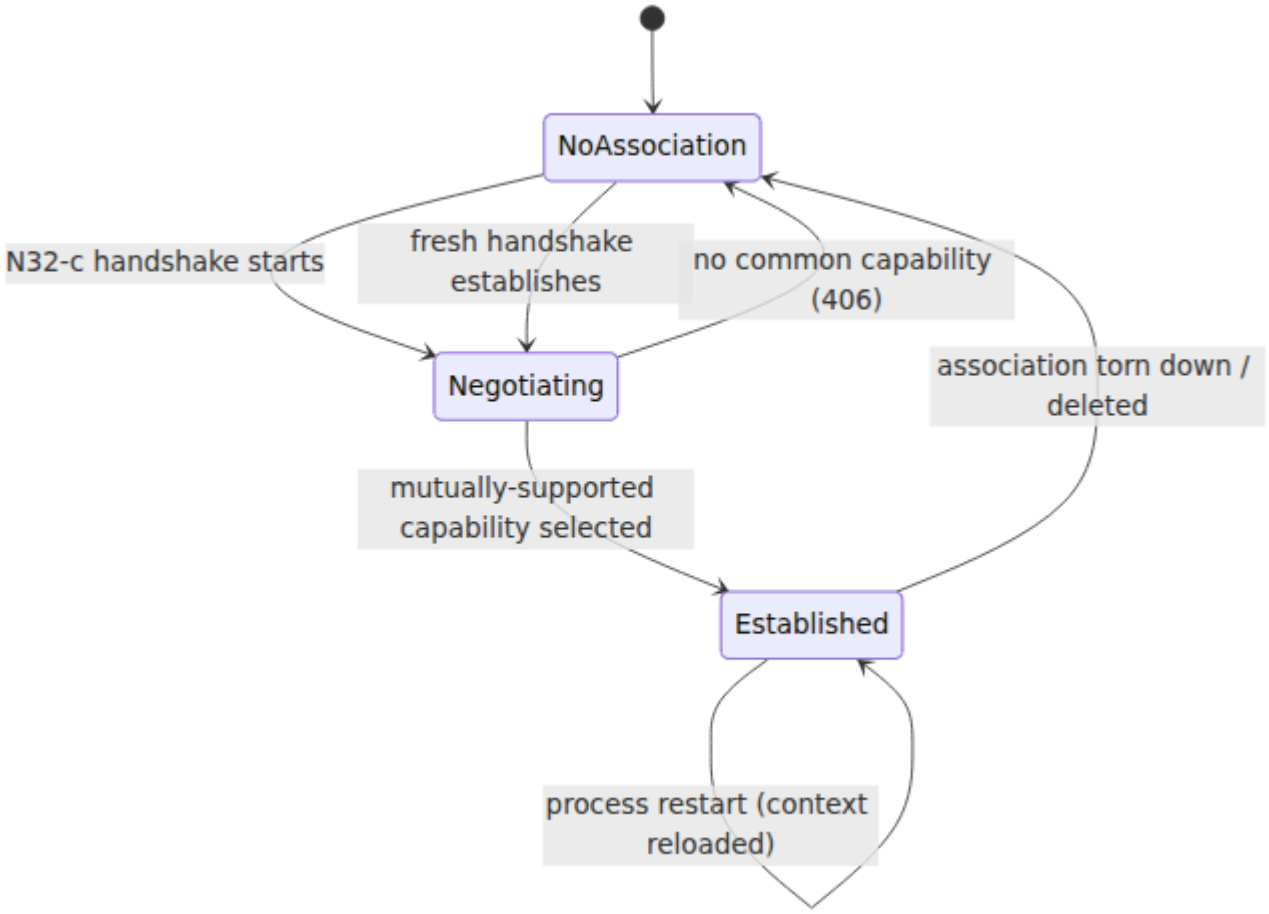
On every relay, hop-by-hop headers (`host`, `content-length`, `connection`) are stripped. On the inbound path the (now-consumed) `3gpp-Sbi-Target-apiRoot` header is also dropped before the request reaches the internal NF.

An empty topology-hiding map (the default) means no rewriting is performed. See Configuration Reference - Topology Hiding Map.

Message Flows

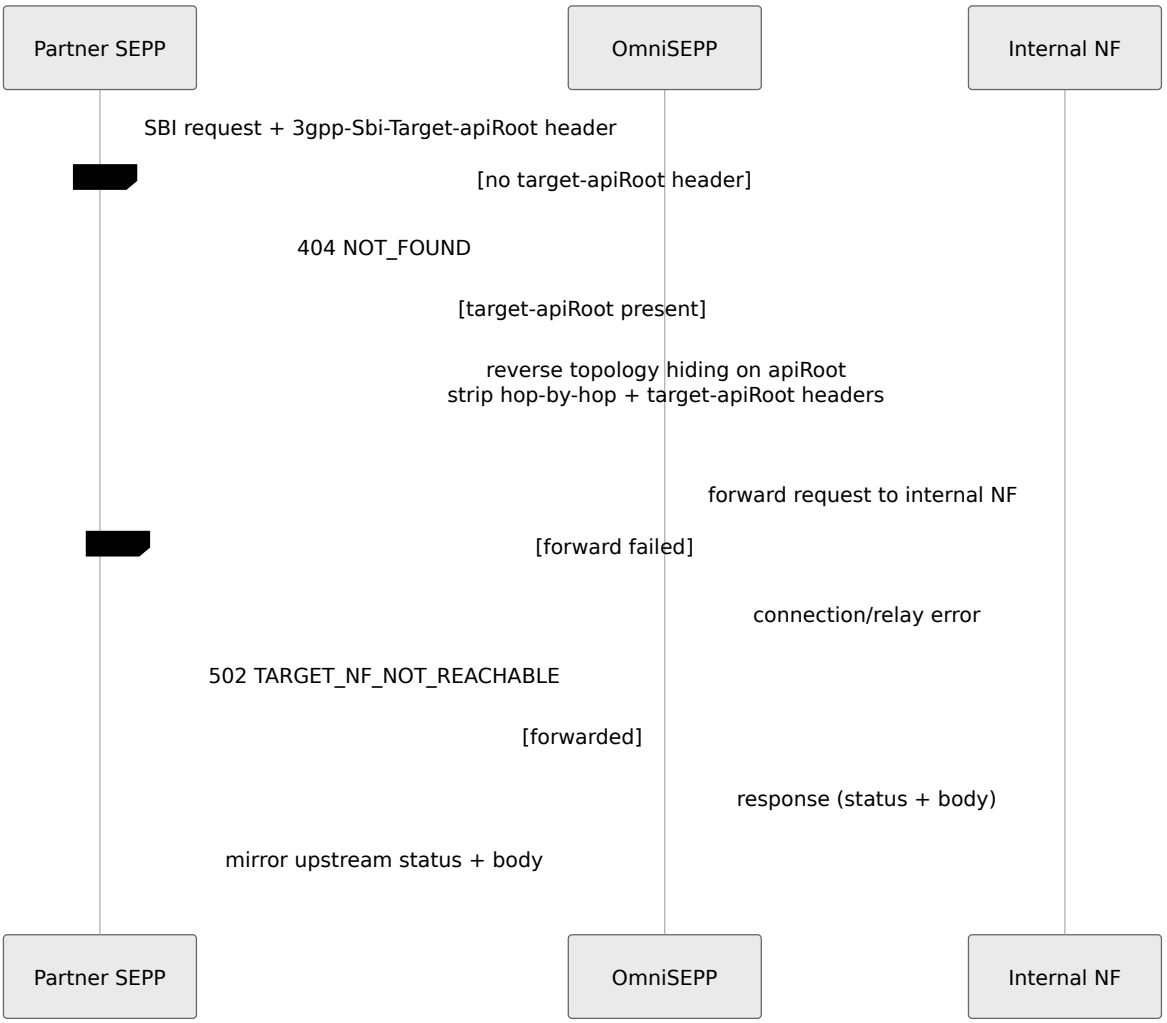
N32-c Handshake (Responder)

A roaming-partner SEPP initiates the handshake by POSTing a capability-negotiation request. OmniSEPP selects a mutually-supported capability, records the association, and replies.



N32-f Inbound Relay

A partner SEPP sends an SBI request carrying a `3gpp-Sbi-Target-apiRoot` header identifying the internal NF. OmniSEPP reverses topology hiding, strips hop-by-hop and routing headers, and forwards the request, mirroring the upstream response.



SBI Endpoints

OmniSEPP exposes the following over its SBI HTTP listener. The N32-c handshake advertises one NRF service, `n32c-handshake` version `v1`.

Method	Path	Purpose
POST	`/n32c-handshake/v1/exchange-capability`	N32-c Security Capability Negotiation (responder).
GET	`/n32c-handshake/v1/status`	Liveness probe.
(any)	any other path	N32-f relay. Routed outbound to the roaming-partner SEPP when the request names a partner PLMN, otherwise inbound to the internal NF named by `3gpp-Sbi-Target-apiRoot`.

POST /n32c-handshake/v1/exchange-capability

Negotiates the N32-f security capability with an initiating partner SEPP.

Status	Cause	Condition
200	-	Negotiation succeeded. Response carries `sender` and `selectedSecCapability`.
400	`MANDATORY_IE_MISSING`	Request is missing `sender`, or `supportedSecCapabilityList` is absent or empty.
406	`INSUFFICIENT_RESOURCES`	No mutually-supported N32-f security capability.

GET /n32c-handshake/v1/status

Returns `200` with body `{"status":"UP"}`. Suitable as a liveness probe.

N32-f Relay (catch-all)

Any request that is not one of the handshake endpoints is treated as N32-f relay traffic. The request is routed **outbound** (to the roaming-partner SEPP) when it names a partner PLMN as its egress selector, otherwise **inbound** (to the internal NF named by `3gpp-Sbi-Target-apiRoot`).

Status	Cause	Condition
(mirrored)	-	Upstream status and body are mirrored back on success (inbound or outbound).
404	`NOT_FOUND`	Inbound: the `3gpp-Sbi-Target-apiRoot` header is absent, so there is no routing target.
403	`TARGET_PLMN_NOT_A_PARTNER`	Outbound: the named target PLMN is not a configured roaming partner.
502	`TARGET_NF_NOT_REACHABLE`	The forward to the internal NF (inbound) or partner SEPP (outbound) failed.

NRF Registration

On start, OmniSEPP registers with the NRF as NFType `SEPP` (TS 23.501 §6.2.17) and heartbeats on the configured interval. The registration advertises the SBI address, port, and scheme, the served PLMN, and the single `n32c-handshake` `v1` service.

Registering with the local NRF lets local consumers and the SCP discover this SEPP (TS 29.573 cl 4.2.1). The **remote** peer SEPP, by contrast, is resolved from the static `roaming_partners` configuration per roaming agreement: this is the

intended model for inter-PLMN peer resolution, since a SEPP in one PLMN cannot look up a SEPP in another PLMN's NRF. Topology hiding is likewise static configuration.

Registration and heartbeat parameters are documented in the Configuration Reference.

Outbound Handshake at Start-up

OmniSEPP acts as the **initiating SEPP** as well as the responder. On boot, once the context store is ready, it initiates an N32-c capability negotiation toward every configured roaming partner (`roaming_partners`) that has a `sepp_uri` (TS 29.573 §5.2.3.2). This is **best-effort**: an unreachable or rejecting partner is logged and skipped rather than crashing start-up, and the task runs once without restart. A partner that was unreachable at boot can still establish the association later by sending its own (responder-side) handshake, or on the next restart.

Operational Notes

- **Restart behaviour:** the N32 association store is persistent (DETS-backed). Associations are reloaded on restart, so partners do not have to re-run the N32-c handshake for relay to resume.
- **Unknown PLMNs:** with an empty roaming-partner registry, N32-f forwarding refuses unknown PLMNs. Populate the registry before expecting partner traffic to be relayed.
- **TLS termination:** when `sbi_scheme` is `https`, the N32 listener terminates TLS itself using the configured certificate and key, and enforces mutual (client-certificate) TLS when a CA is configured. See Configuration Reference - N32 Listener TLS.
- **Default PLMN:** the shipped default served PLMN (`999-70`) is a test value and must be overridden per deployment.
- **Observability:** operate against structured application logs; see Metrics and Monitoring.

OmniSEPP Troubleshooting

This guide covers common operational issues and their resolutions. Diagnosis relies on structured logs; see Metrics and Monitoring.

Common Issues

N32-c Handshake Rejected With 400

Symptoms: A partner SEPP's capability-negotiation request is answered with `400 MANDATORY_IE_MISSING`.

Possible causes:

- The request is missing the `sender` identity.
- The `supportedSecCapabilityList` is absent or empty.

Resolution:

1. Confirm the partner is sending a well-formed `SecNegotiateReqData` with a non-empty `sender`.
2. Confirm the partner includes at least one N32-f security capability in `supportedSecCapabilityList`.

N32-c Handshake Rejected With 406

Symptoms: A partner's handshake is answered with `406 INSUFFICIENT_RESOURCES` and no association is recorded.

Possible causes:

- There is no capability common to this SEPP's `supported_sec_capabilities` and the partner's offered list. For example the partner offers only `PRINS` while this SEPP offers only `TLS`.

Resolution:

1. Compare the partner's offered capabilities against this SEPP's `supported_sec_capabilities`.
2. Ensure both sides share at least one capability. In practice this means both must support `TLS`; keep `supported_sec_capabilities` at `["TLS"]` for interoperable operation.

N32-f Traffic Refused for a PLMN

Symptoms: N32-f relay is refused, and the logs show a refusal to forward to a non-partner PLMN.

Possible causes:

- The partner PLMN is not present in the `roaming_partners` registry.
- The registry is empty (the default), so all unknown PLMNs are refused.

Resolution:

1. Add the partner PLMN to `roaming_partners`, keyed by `"mcc-mnc"`, with the partner SEPP `sepp_uri`.
2. Confirm the PLMN key exactly matches the partner's `mcc-mnc`.

Inbound Relay Returns 404 NOT_FOUND

Symptoms: A partner request is answered with `404 NOT_FOUND` instead of being relayed.

Possible causes:

- The request does not carry a `3gpp-Sbi-Target-apiRoot` header, so OmniSEPP has no routing target.

Resolution:

1. Confirm the partner sets the `3gpp-Sbi-Target-apiRoot` header identifying the internal NF.
2. If topology hiding is in use, confirm the external apiRoot the partner presents has a matching reverse entry in the `topology_hiding_map`.

Inbound Relay Returns 502 TARGET_NF_NOT_REACHABLE

Symptoms: A partner request is answered with `502 TARGET_NF_NOT_REACHABLE`.

Possible causes:

- The addressed internal NF is down or unreachable from OmniSEPP.
- Topology hiding did not resolve the external apiRoot to a reachable internal FQDN.

Resolution:

1. Confirm the internal NF resolved from the (reversed) apiRoot is running and reachable.
2. Check the `topology_hiding_map` entry maps the external FQDN to the correct, resolvable internal FQDN.

Associations Disappear After a Restart

Symptoms: After restarting OmniSEPP, previously established partners are treated as if no association exists, and N32-f relay for them fails until they handshake again.

Possible causes:

- The DETS file at `n32_context_path` could not be opened (for example on corruption or a permissions problem), so the store started empty and logged a warning.
- The configured `n32_context_path` changed between runs, so the previous file is not being read.

Resolution:

1. Associations are normally persistent and reload on restart, so a restart alone should not clear them. If they are missing, check the start-up logs for an N32Context warning about opening DETS.
2. Confirm `n32_context_path` points to a stable, writable location that is preserved across restarts (not a per-run temporary directory).

Partner Cannot Establish Mutually-Authenticated TLS

Symptoms: A partner expects N32 over mutually-authenticated TLS but the connection is refused, is plain HTTP, or the partner's client certificate is not being verified.

Possible causes:

- `sbi_scheme` is `http`, so the listener serves plaintext.
- `sbi_scheme` is `https` but `sbi_cacertfile` is not set, so only server-side TLS is terminated and the peer's client certificate is not required.
- `sbi_scheme` is `https` but `sbi_certfile` / `sbi_keyfile` are not configured, in which case OmniSEPP refuses to start rather than downgrading to plaintext.

Resolution:

1. Set `sbi_scheme` to `https` and configure `sbi_certfile` and `sbi_keyfile` so the listener terminates TLS itself.
2. To require and verify the peer SEPP's client certificate (mutual TLS), set `sbi_cacertfile` to the CA that signs the peer's certificate. See Configuration Reference - N32 Listener TLS.

Reference

Topic	Specification
SEPP network function	3GPP TS 23.501 §6.2.17
N32-f TLS mode and topology hiding	3GPP TS 33.501 §13.1, §13.2
N32 handshake and N32-f forwarding	3GPP TS 29.573 §5.2.3, §5.3, §6.1.5.2