

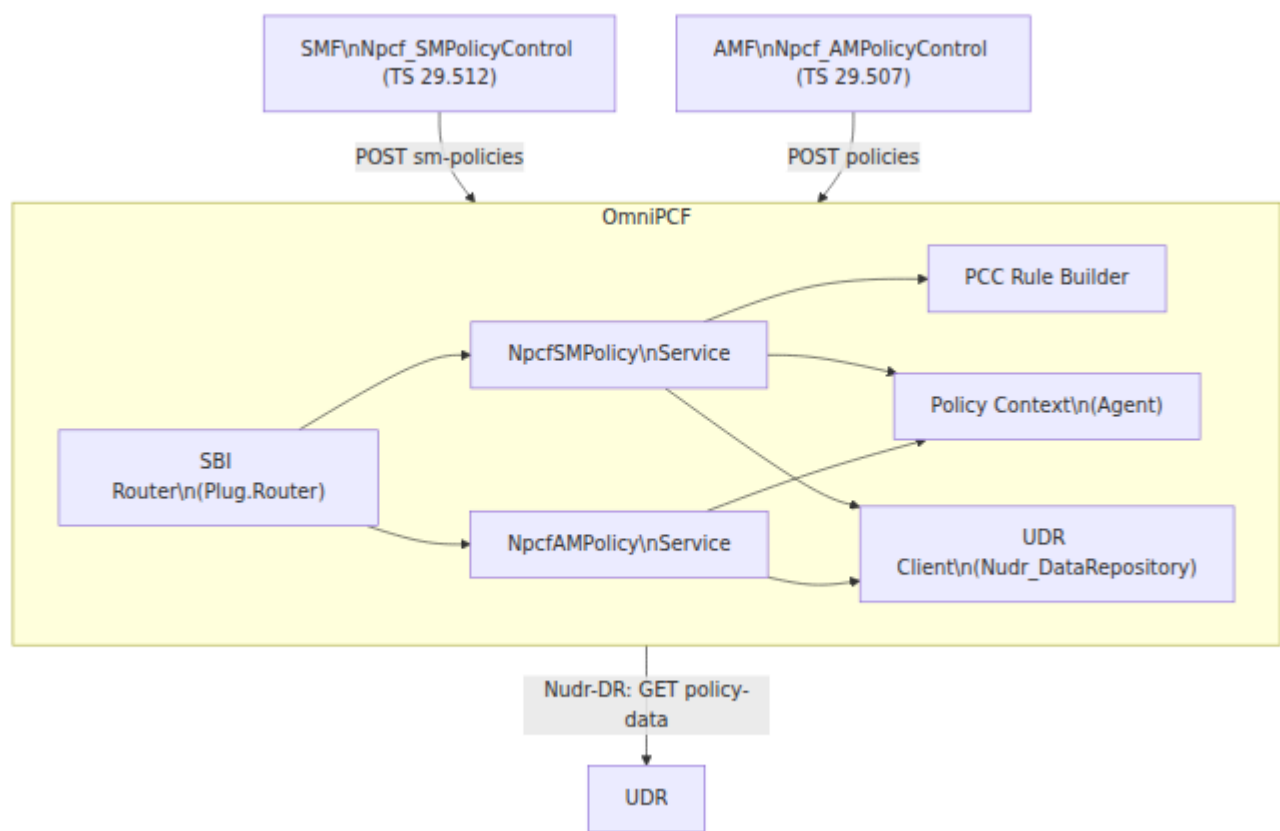
# OmniPCF Operations Guide

## Overview

OmniPCF implements the Policy Control Function (PCF) of the 5G Core. It provides two SBI services: Npcf\_SMPolicyControl (TS 29.512) for session management policy associations, and Npcf\_AMPolicyControl (TS 29.507) for access and mobility policy associations. The PCF optionally queries the UDR for per-subscriber policy data (subscription QoS, UE-AMBR) and builds Policy and Charging Control (PCC) decisions which are returned to the SMF and AMF respectively.

Policy context is held in-process. All PCC rules generated are permit-all default rules; dynamic, subscriber-specific, or GBR flow rules are not supported in the current release.

# Architecture



# 3GPP Role and Specification References

| Specification | Relevance                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------|
| TS 23.501     | System architecture — PCF role, PCC framework                                                            |
| TS 23.503     | Policy and charging control architecture                                                                 |
| TS 29.507     | Npcf_AMPolicyControl API                                                                                 |
| TS 29.512     | Npcf_SMPolicyControl API                                                                                 |
| TS 29.519     | Nudr_DataRepository — policy data resource (queried by PCF)                                              |
| TS 23.502     | Procedures — references to policy association creation during PDU session establishment and registration |

---

## SBI Endpoints

All endpoints are served under the base URL `{sbi_scheme}://{sbi_addr}:{sbi_port}`.

| Method | Path                                                     | Service              | Description                  |
|--------|----------------------------------------------------------|----------------------|------------------------------|
| POST   | /npcf-smpolicycontrol/v1/sm-policies                     | Npcf_SMPolicyControl | Create SM Policy Association |
| POST   | /npcf-smpolicycontrol/v1/sm-policies/{smPolicyId}/update | Npcf_SMPolicyControl | Update SM Policy Association |
| POST   | /npcf-smpolicycontrol/v1/sm-policies/{smPolicyId}/delete | Npcf_SMPolicyControl | Delete SM Policy Association |
| POST   | /npcf-am-policy-control/v1/policies                      | Npcf_AMPolicyControl | Create AM Policy Association |
| DELETE | /npcf-am-policy-control/v1/policies/{polAssoId}          | Npcf_AMPolicyControl | Delete AM Policy Association |

## Request / Response Summary

**Create SM Policy Association** — mandatory request fields: `supi`, `pduSessionId`, `pduSessionType`, `dnn`, `notificationUri`, `sliceInfo`, and at least one of `ipv4Address` or `ipv6AddressPrefix`. Returns `201 Created` with `SmPolicyDecision` and `Location` header.

**Update SM Policy Association** — the SMF sends `SmPolicyUpdateContextData`. The PCF updates the stored context with any provided `ipv4Address`, `ipv6AddressPrefix`, or `ratType` fields and returns a refreshed `SmPolicyDecision`. Rule report processing is not performed (see PCF-M3).

**Delete SM Policy Association** — body is `SmPolicyDeleteData` (ignored beyond logging). Returns `204 No Content`.

**Create AM Policy Association** — mandatory request fields: `supi`, `notificationUri`. Optional fields include `guami`, `accessType`, `ratType`, `allowedSnssais`, `ueAmbr`, `suppFeat`. Returns `201 Created` with `PolicyAssociation` and `Location` header.

**Delete AM Policy Association** — no request body required. Returns `204 No Content`.

---

## Configuration Reference

Configuration is read from the application environment key `:omnipcf`.

```
config :omnipcf,  
  sbi_scheme: "http",  
  sbi_addr:   "127.0.0.13",  
  sbi_port:   7777,  
  nrf_uri:    "http://127.0.0.10:7777",  
  udr_uri:    "http://127.0.0.12:7777",  
  mcc: "999",  
  mnc: "70",  
  heartbeat_interval: 10_000
```

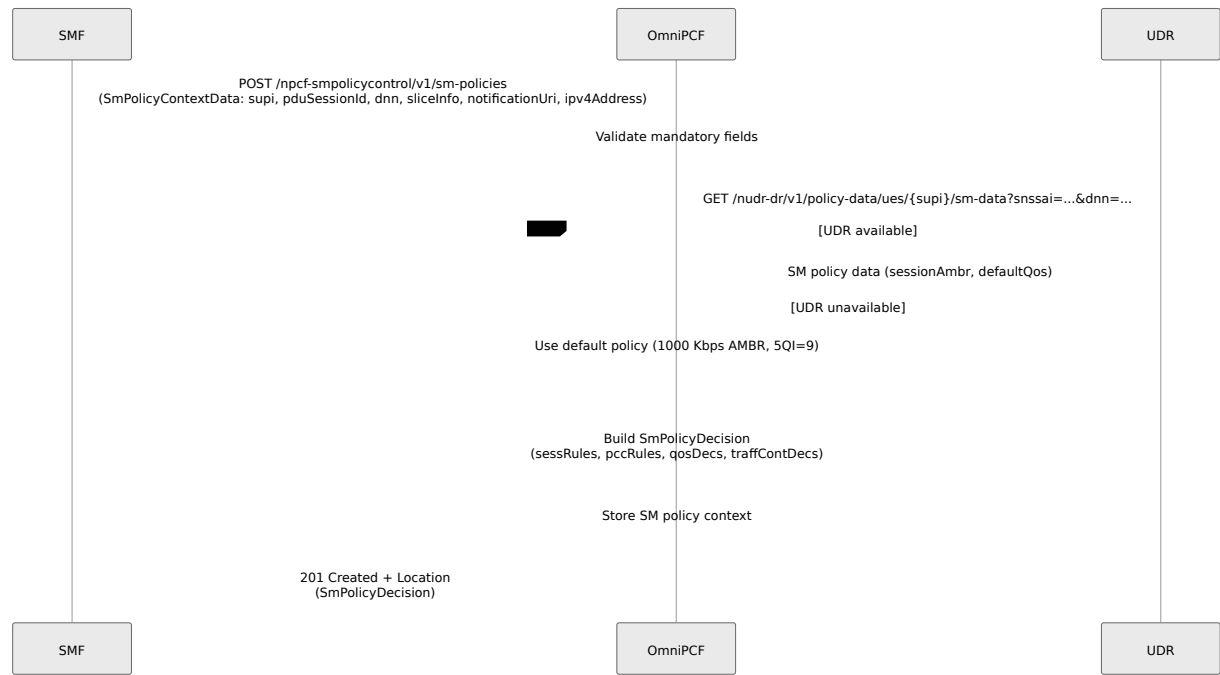
# Parameter Table

| Parameter  | Type    | Default                  | Descript                                                                                                                                                                         |
|------------|---------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sbi_scheme | string  | "http"                   | HTTP scheme for SBI listening (http or https)                                                                                                                                    |
| sbi_addr   | string  | "127.0.0.13"             | IP address the HTTP server listens on                                                                                                                                            |
| sbi_port   | integer | 7777                     | TCP port the HTTP server listens on                                                                                                                                              |
| nrf_uri    | string  | "http://127.0.0.10:7777" | Base URI of the NRF. Used for NF registration and heartbeat on                                                                                                                   |
| udr_uri    | string  | "http://127.0.0.12:7777" | Base URI of the UDR. Queried for A data (/nudr-v1/policies/data/ues/{ues-id}) and SM data (/nudr-v1/policies/data/ues/{ues-id}). If unavailable, the PCF proceeds with default p |
| mcc        | string  | "999"                    | Mobile Country Code of the serving                                                                                                                                               |

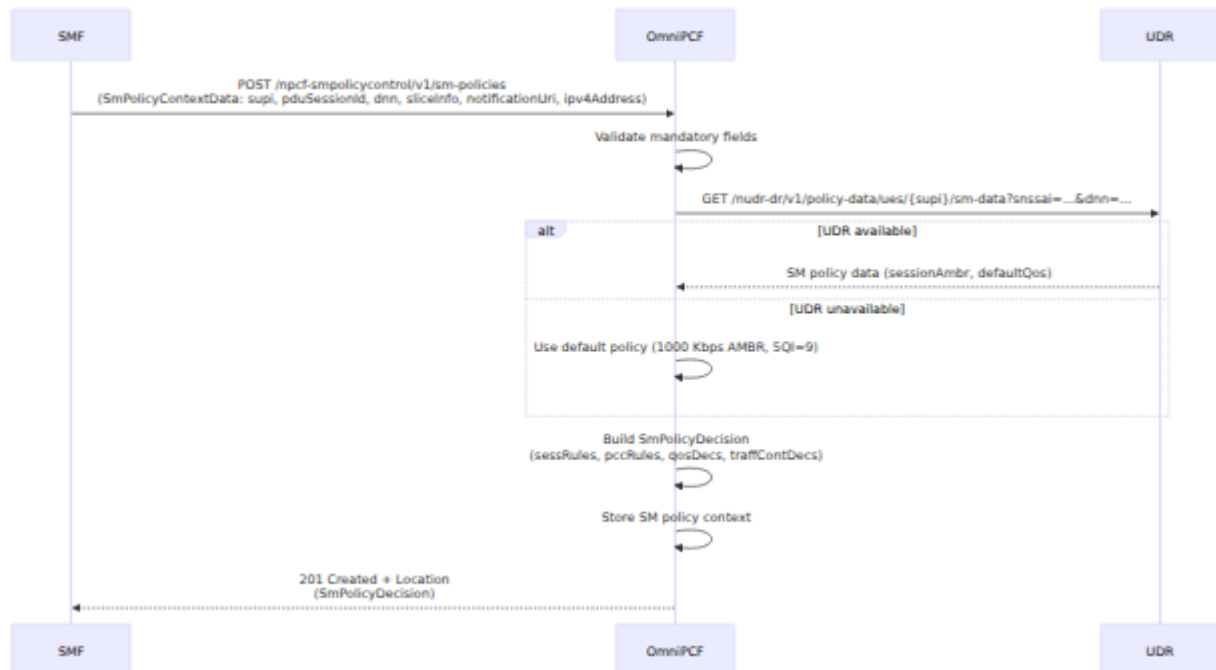
| Parameter          | Type         | Default                  | Descript                                                         |
|--------------------|--------------|--------------------------|------------------------------------------------------------------|
| mnc                | string       | "70"                     | Mobile Network of the serving                                    |
| heartbeat_interval | integer (ms) | 10000                    | Interval in milliseconds NRF heartbeat requests                  |
| hss_api_base_url   | string       | "https://127.0.0.1:8443" | Base URL of HSS API. Pres configuration used by activ procedures |

# Key Procedures

## SM Policy Association Creation (TS 29.512 Section 4.2.2)



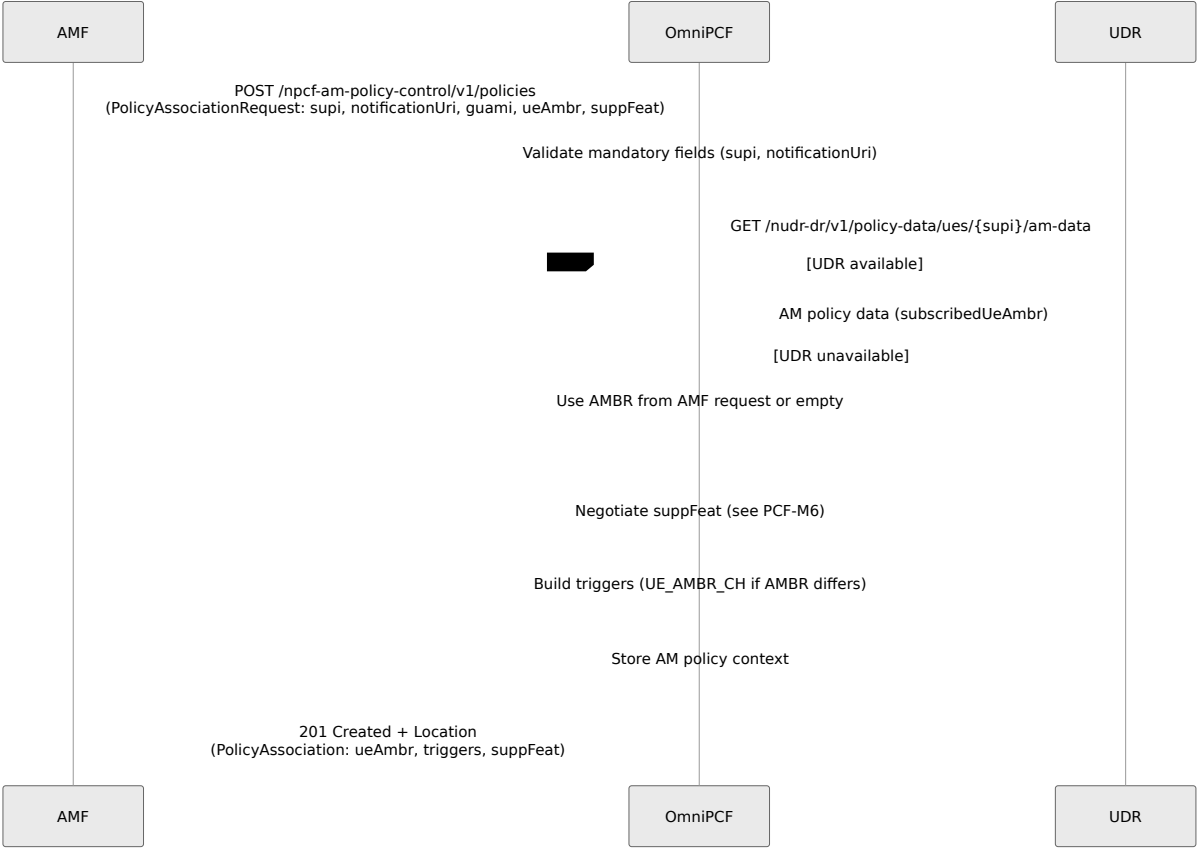
## SM Policy Association Update (TS 29.512 Section 4.2.3)



Note: Rule reports within `SmPolicyUpdateContextData` are not processed (PCF-M3). Trigger-based policy updates are not performed (PCF-M4). The PCF does not proactively notify the SMF of policy changes (PCF-M7).



# AM Policy Association Creation (TS 29.507 Section 4.2.2)



## SmPolicyDecision Structure

The PCF always returns a default permit-all decision. The structure returned per TS 29.512 Table 5.6.2.3-1:

```
SmPolicyDecision
  sessRules:
    SessRule-{dnn}:
      sessRuleId: "SessRule-{dnn}"
      authSessAmbr: {uplink, downlink} <- from UDR or default
1000 Kbps
      authDefQos: {5qi: 9, arp: {...}} <- from UDR or 5QI=9 ARP-
priority=8

    pccRules:
      PccRule-{dnn}-default:
        pccRuleId: "PccRule-{dnn}-default"
        precedence: 255
        flowInfos: [permit out ip from any to {ueIp},
                    permit out ip from {ueIp} to any]
        refQosData: ["QosData-{dnn}"]
        refTcData: ["TcData-{dnn}"]

    qosDecs:
      QosData-{dnn}:
        5qi: 9
        arp: {priorityLevel: 8, preemptCap: NOT_PREEMPT,
preemptVuln: PREEMPTABLE}
        maxbrUl / maxbrDl: session AMBR values

    traffContDecs:
      TcData-{dnn}:
        flowStatus: ENABLED

    suppFeat: echoed from request (see PCF-M6)
```

---

# Prometheus Metrics

## SM Policy Metrics

| Metric                                         | Type    | Tags                                                           | Description                  |
|------------------------------------------------|---------|----------------------------------------------------------------|------------------------------|
| <code>omni_pcf.sm_policy.create.count</code>   | counter | <code>supi,</code><br><code>dnn,</code><br><code>result</code> | SM policy create operations  |
| <code>omni_pcf.sm_policy.delete.count</code>   | counter | <code>supi,</code><br><code>dnn</code>                         | SM policy delete operations  |
| <code>omni_pcf.sm_policy_creates.total</code>  | counter | <code>result</code>                                            | Total SM policy creates      |
| <code>omni_pcf.active_sm_policies.count</code> | gauge   | --                                                             | Number of active SM policies |

# AM Policy Metrics

| Metric                                         | Type    | Tags                                       | Description                  |
|------------------------------------------------|---------|--------------------------------------------|------------------------------|
| <code>omni_pcf.am_policy.create.count</code>   | counter | <code>supi</code> ,<br><code>result</code> | AM policy create operations  |
| <code>omni_pcf.am_policy.delete.count</code>   | counter | <code>supi</code>                          | AM policy delete operations  |
| <code>omni_pcf.am_policy_creates.total</code>  | counter | <code>result</code>                        | Total AM policy creates      |
| <code>omni_pcf.active_am_policies.count</code> | gauge   | --                                         | Number of active AM policies |

# PCC Rule Metrics

| Metric                                        | Type    | Tags | Description                  |
|-----------------------------------------------|---------|------|------------------------------|
| <code>omni_pcf.pcc_rule_installs.total</code> | counter | --   | Total PCC rule installations |

# NRF Metrics

| Metric                           | Type  | Tags    | Description                                   |
|----------------------------------|-------|---------|-----------------------------------------------|
| omni_pcf.nrf.registration.status | gauge | nf_type | NRF registration status (1=registered, 0=not) |

# BEAM VM Metrics

| Metric                                  | Type  | Description                       |
|-----------------------------------------|-------|-----------------------------------|
| <code>beam.memory.total</code>          | gauge | Total BEAM memory in bytes        |
| <code>beam.memory.processes</code>      | gauge | Memory used by Erlang processes   |
| <code>beam.memory.processes_used</code> | gauge | Memory actually used by processes |
| <code>beam.memory.system</code>         | gauge | System memory                     |
| <code>beam.memory.atom</code>           | gauge | Total atom memory                 |
| <code>beam.memory.atom_used</code>      | gauge | Used atom memory                  |
| <code>beam.memory.binary</code>         | gauge | Binary memory                     |
| <code>beam.memory.code</code>           | gauge | Code memory                       |
| <code>beam.memory.ets</code>            | gauge | ETS table memory                  |
| <code>beam.processes.count</code>       | gauge | Number of Erlang processes        |
| <code>beam.ports.count</code>           | gauge | Number of Erlang ports            |
| <code>beam.atom.count</code>            | gauge | Number of atoms                   |
| <code>beam.vm.uptime</code>             | gauge | VM uptime in seconds              |

# Known Limitations

| ID     | Area                   | Description                                                                                                                                                                                                         |
|--------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PCF-M1 | AM Policy Update       | There is no <code>POST /npcf-am-policy-control/v1/policies/{polAssoId}/update</code> endpoint. The AMF cannot request policy updates for an existing AM policy association. Only create and delete are supported.   |
| PCF-M2 | AM Policy Notification | The PCF does not send policy update notifications to the AMF callback URI. AMF-side policy changes (e.g., UE-AMBR modification) are never pushed by the PCF.                                                        |
| PCF-M3 | Rule Report Processing | <code>RuleReport</code> entries within <code>SmPolicyUpdateContextData</code> are not inspected or processed. The PCF cannot react to rule enforcement failures or changes reported by the SMF.                     |
| PCF-M4 | Trigger-based Updates  | <code>PolicyControlRequestTrigger</code> values are not evaluated. The PCF does not detect or respond to events such as PLMN change, RAT change, or access type change that would normally trigger a policy update. |
| PCF-M5 | Dynamic PCC Rules      | Only a single default permit-all PCC rule is generated per session. No service-data-flow-specific PCC rules, application detection rules, or subscriber-specific policy from UDR are applied.                       |

| ID      | Area                    | Description                                                                                                                                                                                                                                                             |
|---------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PCF-M6  | Feature Negotiation     | <code>suppFeat</code> in SM and AM policy responses is set to the value echoed from the request rather than computed as a bitwise AND of the PCF's supported features and the consumer's requested features (TS 29.512 Section 6.1).                                    |
| PCF-M7  | SM Policy Notification  | The PCF does not invoke the SMF's <code>notificationUri</code> to push policy decisions or updates. All PCF-initiated SM policy updates (e.g., in response to application function requests) are absent.                                                                |
| PCF-L1  | GBR Flows               | Guaranteed Bitrate (GBR) QoS flows are not supported. The QER and PCC rules only express MBR-based constraints.                                                                                                                                                         |
| PCF-L3  | UDR Policy Subscription | The PCF queries the UDR once at policy association creation. There is no subscription to UDR data change notifications ( <code>Nudr_DataRepository_Subscribe</code> ), so changes to subscriber policy data in the UDR are not reflected in active policy associations. |
| CROSS-2 | UDR Endpoint Mismatch   | The PCF queries <code>/nudr-dr/v1/policy-data/ues/{ueId}/am-data</code> and <code>.../sm-data</code> on the UDR. These endpoints are not exposed by OmniUDM in the current release. UDR queries will return 404 and the PCF will silently fall back to default policy.  |



# Troubleshooting

## SM Policy Association Returns 400 ERROR\_INITIAL\_PARAMETERS

The `SmPolicyContextData` from the SMF is missing a mandatory field. The PCF requires: `supi`, `pduSessionId`, `pduSessionType`, `dnn`, `notificationUri`, `sliceInfo`, and at least one of `ipv4Address` or `ipv6AddressPrefix`. Check that the SMF is sending the UE IP address in the `create_sm_policy` call — the SMF populates `ipv4Address` from the context's `ue_ipv4` field, which is set by the PFCP handler after session establishment. If PFCP fails before IP allocation, the field will be absent.

## SM Policy Association Returns 403 POLICY\_CONTEXT\_DENIED

Returned only when a specific denial is encountered in policy validation (not currently triggered by default policy logic). If seen unexpectedly, check the PCF log for `[SM Policy] Created association ... failed` entries.

## UDR Queries Always Fail

See limitation CROSS-2. OmniUDM does not expose the `/policy-data/` resource tree. The PCF logs a warning `[UDR Client] SM policy query failed for {supi}` and proceeds with default policy. This is expected in the current deployment. No corrective action is needed unless a UDR that exposes TS 29.519 policy data resources is deployed.

## Policy Decision Contains Default AMBR Instead of Subscribed Values

This occurs when the UDR query fails (CROSS-2) or the UDR returns an empty body. The default session AMBR applied is `1000 Kbps` uplink and downlink. To override, ensure the UDR returns `sessionAmbr` in the SM policy data response, or configure subscriber data in a UDR that implements the TS 29.519 policy data resource.

## Feature Negotiation Mismatch with SMF

Due to PCF-M6, the `suppFeat` in the SM policy response echoes the value from the SMF request rather than performing a proper bitwise AND. If the SMF or AMF relies on specific feature bits being masked off, unexpected behavior may result. As a workaround, configure the SMF to send only the features it actually needs in `suppFeat`.

## AM Policy Associations Accumulate Without Bound

AM policy associations are created at UE registration and deleted at deregistration. If the AMF does not send the DELETE when deregistering (e.g., due to AMF restart), associations will persist indefinitely. There is no expiry mechanism. Monitor the `/statistics` endpoint for association count. To manually clear stale associations, restart OmniPCF (all state is in-process and non-persistent).

## Log Correlation

SM policy log lines are prefixed `[SM Policy]` and include the association ID and SUPI. AM policy log lines are prefixed `[AM Policy]`. UDR client calls are prefixed `[UDR Client]`. All entries include the SUPI, enabling per-subscriber tracing across the PCF log.