OmniEPDG Architecture & Call Flows

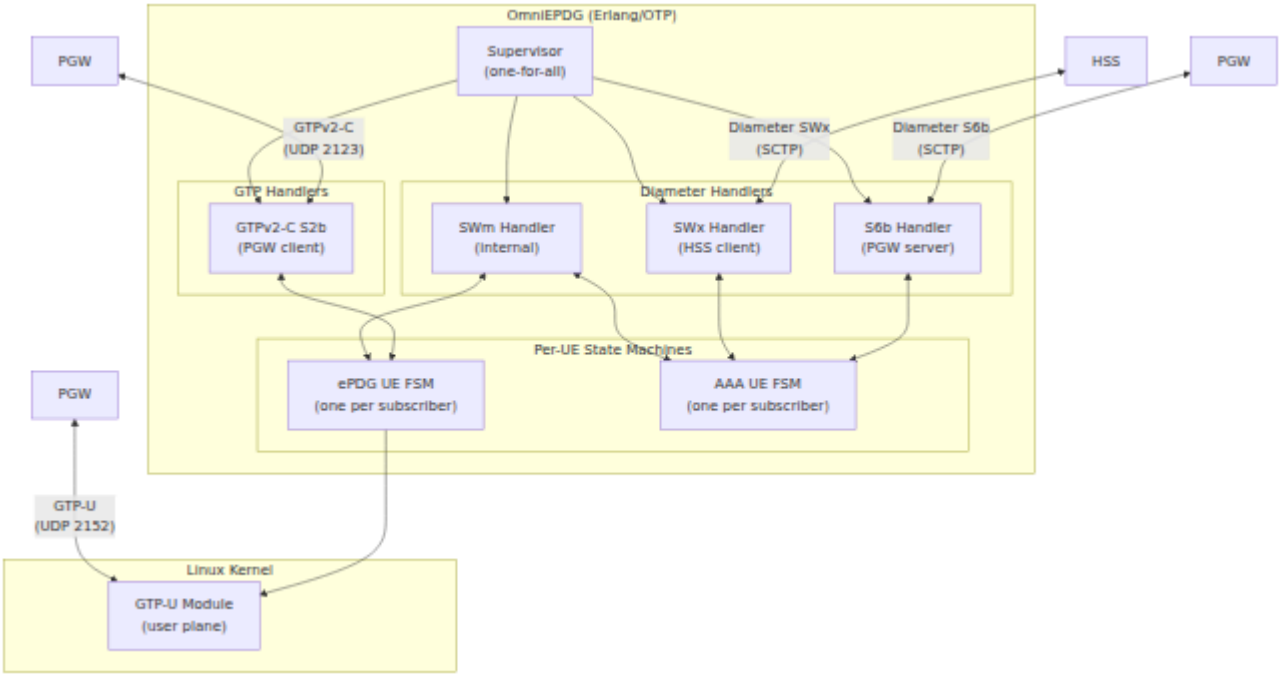
This document describes the internal architecture of OmniEPDG, its protocol interfaces, UE state machines, and detailed message sequence diagrams for key procedures. OmniEPDG supports two operational modes: **GTP mode** (full 3GPP tunneling via PGW) and **Simple VPN mode** (local breakout with TUN interface). See the Operations Guide for a high-level comparison.

System Architecture

OmniEPDG is built on Erlang/OTP and implements the 3GPP ePDG (evolved Packet Data Gateway) network function. It bridges untrusted WiFi access to the mobile core network, allowing UEs to make and receive VoWiFi calls.

GTP Mode Architecture

In GTP mode, subscriber traffic is tunneled through a PGW using GTPv2-C for session control and the Linux kernel GTP-U module for the user plane.



Simple VPN Mode Architecture

In Simple VPN mode, subscriber traffic is routed locally through a Linux TUN interface. No PGW or GTP infrastructure is required. The Diameter S6b, GTPv2-C, and GTP-U components are replaced by the Simple VPN subsystem.

Supervisor Tree

OmniEPDG uses a **one-for-all** supervisor strategy, meaning if any child process crashes, all children are restarted. The supervisor conditionally starts different child processes depending on the operational mode.

Processes started in both modes:

Process	Role	Description
`aaa_diameter_swx`	Diameter SWx Client	Connects to the HSS for authentication and subscriber profile operations
`aaa_diameter_swm`	Diameter SWm (Internal)	Routes Diameter EAP and session messages between the ePDG and AAA FSMs
`epdg_diameter_swm`	SWm ePDG Handler	Handles the ePDG side of internal SWm Diameter signaling

GTP mode additional processes:

Process	Role	Description
`aaa_diameter_s6b`	Diameter S6b Server	Accepts connections from the PGW for session authorization
`epdg_gtpc_s2b`	GTPv2-C Client	Sends Create/Delete Session requests to the PGW over S2b
`gtp_u_kmod`	GTP-U Kernel Handler	Manages GTP-U PDP contexts in the Linux kernel module

Simple VPN mode additional processes:

Process	Role	Description
`simple_vpn_supervisor`	VPN Subsystem Supervisor	Supervises the IP pool and route manager processes
`simple_vpn_pool`	IP Pool Manager	Allocates and releases IPv4 addresses from the configured CIDR pool using ETS
`simple_vpn_route`	Route Manager	Creates the `omniepdg0` TUN interface and manages per-subscriber host routes

Per-UE State Machines

For each active subscriber (identified by IMSI), OmniEPDG creates two state machine instances:

- **ePDG UE FSM** (`epdg_ue_fsm`) - Manages the subscriber's session lifecycle from the ePDG perspective: authentication, GTP tunnel creation, and teardown coordination
- **AAA UE FSM** (`aaa_ue_fsm`) - Manages the AAA-side signaling: Diameter SWx exchanges with the HSS and S6b exchanges with the PGW

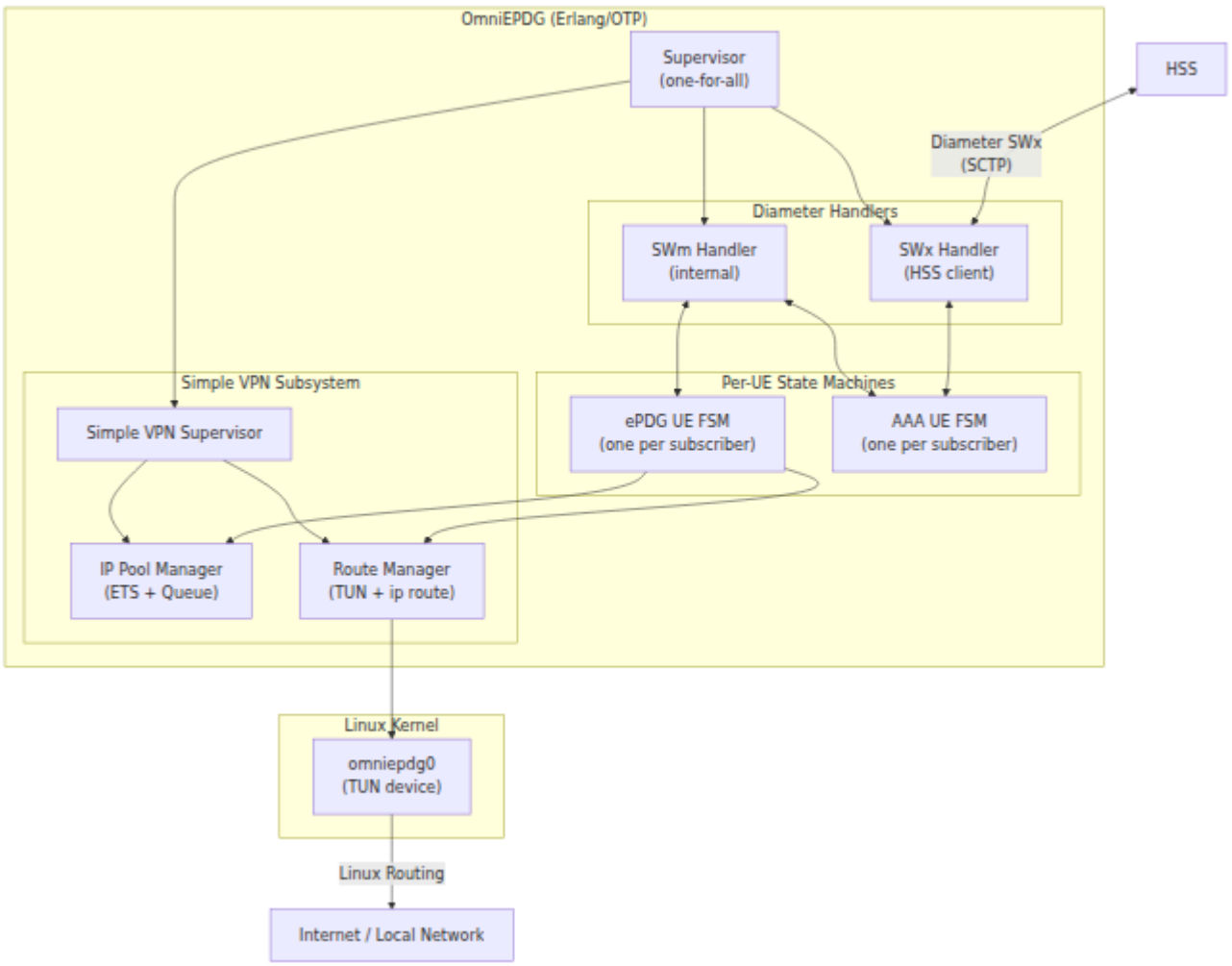
Both FSMs are implemented as Erlang `gen_statem` processes with state function callback mode.

ePDG UE FSM States

The ePDG UE FSM tracks a subscriber's session from initial authentication request through active tunnel state to teardown. The FSM behavior diverges at the `authenticated` state based on the operational mode.

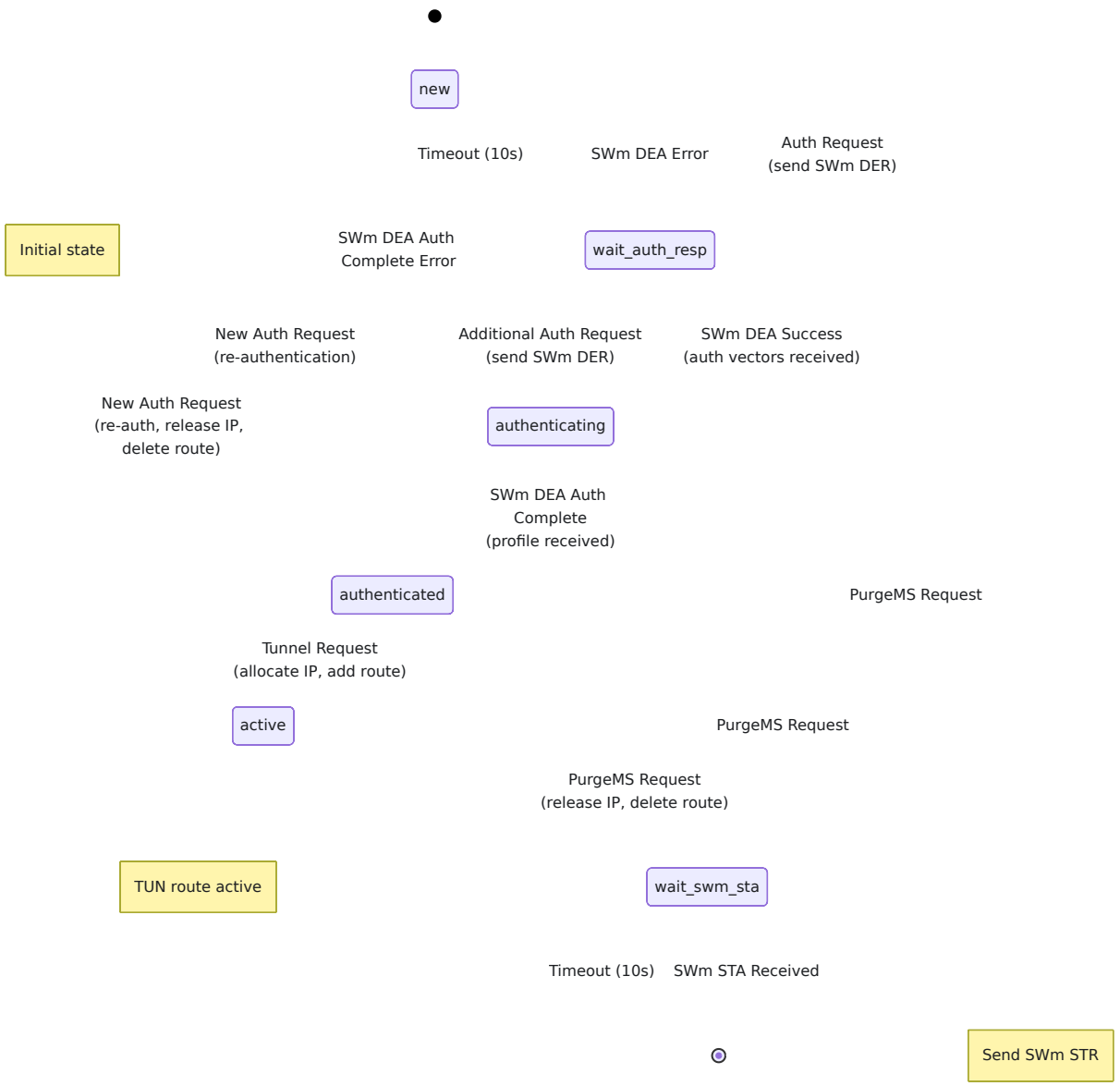
GTP Mode FSM

In GTP mode, tunnel establishment goes through GTPv2-C Create Session to the PGW, and teardown involves GTPv2-C Delete Session, PGW-initiated Delete Bearer, and HSS-initiated deregistration flows.



Simple VPN Mode FSM

In Simple VPN mode, the FSM takes a shortcut at the `authenticated` state. Instead of sending a GTPv2-C Create Session Request, the FSM allocates an IP address from the local pool, creates a host route on the TUN interface, and transitions directly to `active`. The GTP-specific teardown states (`wait_create_session_resp`, `wait_delete_session_resp`, `dereg_pgw_wait_cancel`, `dereg_net_wait_s2b_delete`) are not used.


new
Timeout (10s)
SWm DEA Error
Auth Request
(send SWm DER)
Initial state
SWm DEA Auth
Complete Error
wait_auth_resp
New Auth Request
(re-authentication)
Additional Auth Request
(send SWm DER)
SWm DEA Success
(auth vectors received)
New Auth Request
(re-auth, release IP,
delete route)
authenticating
SWm DEA Auth
Complete
(profile received)
authenticated
PurgeMS Request
Tunnel Request
(allocate IP, add route)
active
PurgeMS Request
PurgeMS Request
(release IP, delete route)
TUN route active
wait_swm_sta
Timeout (10s)
SWm STA Received
Send SWm STR

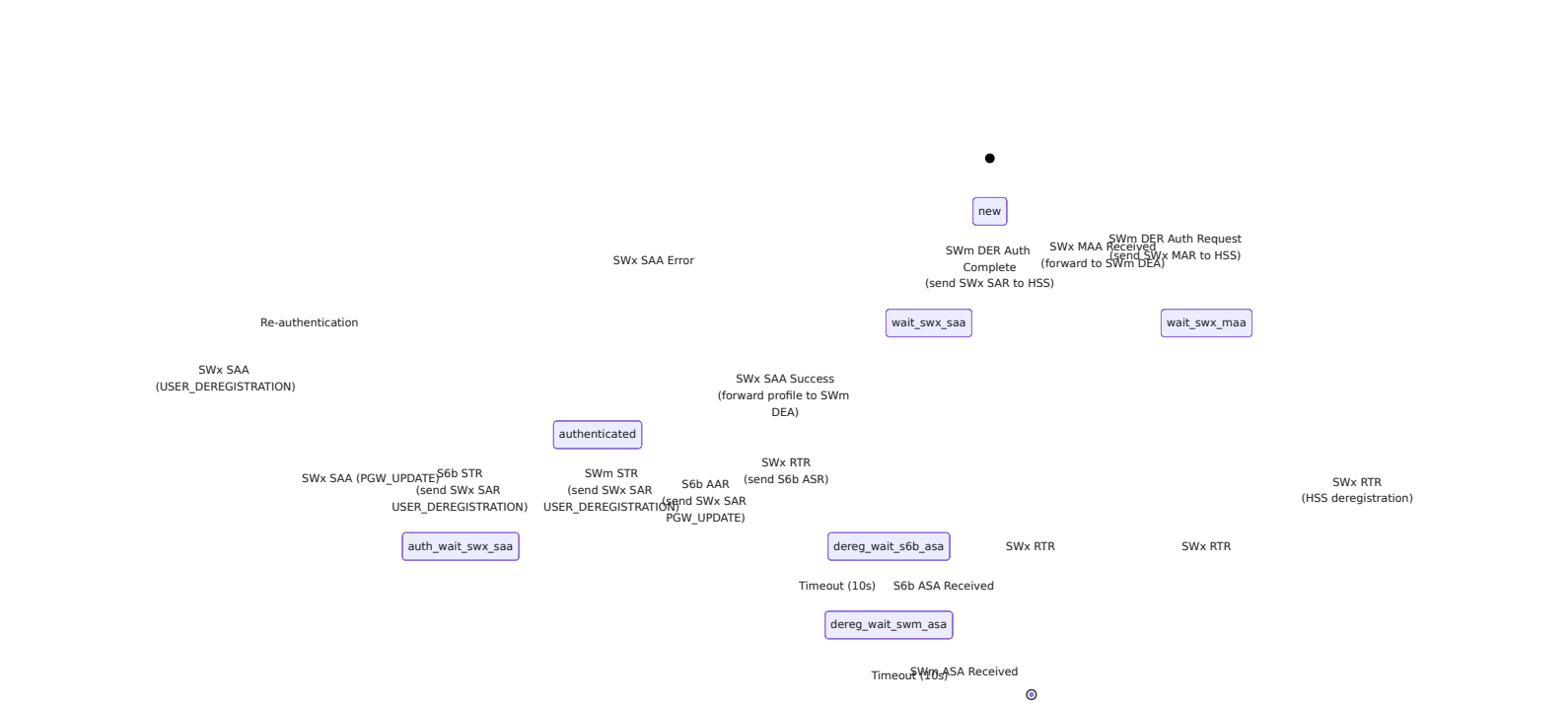
ePDG UE FSM State Reference

State	Mode	Description	Waiting For
`new`	Both	Initial state. No active session.	Auth request from UE
`wait_auth_resp`	Both	Authentication request sent via SWm DER.	SWm DEA with auth vectors or error
`authenticating`	Both	Auth vectors received, EAP exchange in progress.	Location update / auth completion
`authenticated`	Both	Authentication complete, subscriber profile downloaded.	Tunnel request from UE
`wait_create_session_resp`	GTP	GTPv2-C Create Session Request sent to PGW.	Create Session Response from PGW
`active`	Both	Tunnel/route operational. Subscriber traffic is flowing.	Teardown trigger
`wait_delete_session_resp`	GTP	GTPv2-C Delete Session Request sent to PGW (UE-initiated teardown).	Delete Session Response from PGW

State	Mode	Description	Waiting For
`wait_swm_sta`	Both	SWm Session Termination Request sent.	SWm STA from AAA
`dereg_pgw_wait_cancel`	GTP	PGW-initiated deregistration. Cancel Location sent to UE.	Cancel Location Result
`dereg_net_wait_cancel`	GTP	Network/HSS-initiated deregistration. Cancel Location sent to UE.	Cancel Location Result
`dereg_net_wait_s2b_delete`	GTP	Network-initiated deregistration. S2b Delete Session sent to PGW.	Delete Session Response

AAA UE FSM States

The AAA UE FSM manages the Diameter signaling toward the HSS (SWx) and PGW (S6b) on behalf of each subscriber.


new
SWm DER Auth
Complete
(send SWx SAR to HSS)
SWx MAA Received
(forward to SWm DEA)
SWm DER Auth Request
(send SWx MAR to HSS)
SWx SAA Error
Re-authentication
wait_swx_saa
wait_swx_maa
SWx SAA
(USER_DEREGISTRATION)
SWx SAA Success
(forward profile to SWm
DEA)
authenticated
SWx SAA (PGW_UPDATE)
S6b STR
(send SWx SAR
USER_DEREGISTRATION)
SWm STR
(send SWx SAR
USER_DEREGISTRATION)
S6b AAR
(send SWx SAR
PGW_UPDATE)
SWx RTR
(send S6b ASR)
SWx RTR
(HSS deregistration)
auth_wait_swx_saa
dereg_wait_s6b_asa
SWx RTR
SWx RTR
Timeout (10s)
S6b ASA Received
dereg_wait_swm_asa
Timeout (10s)
SWm ASA Received

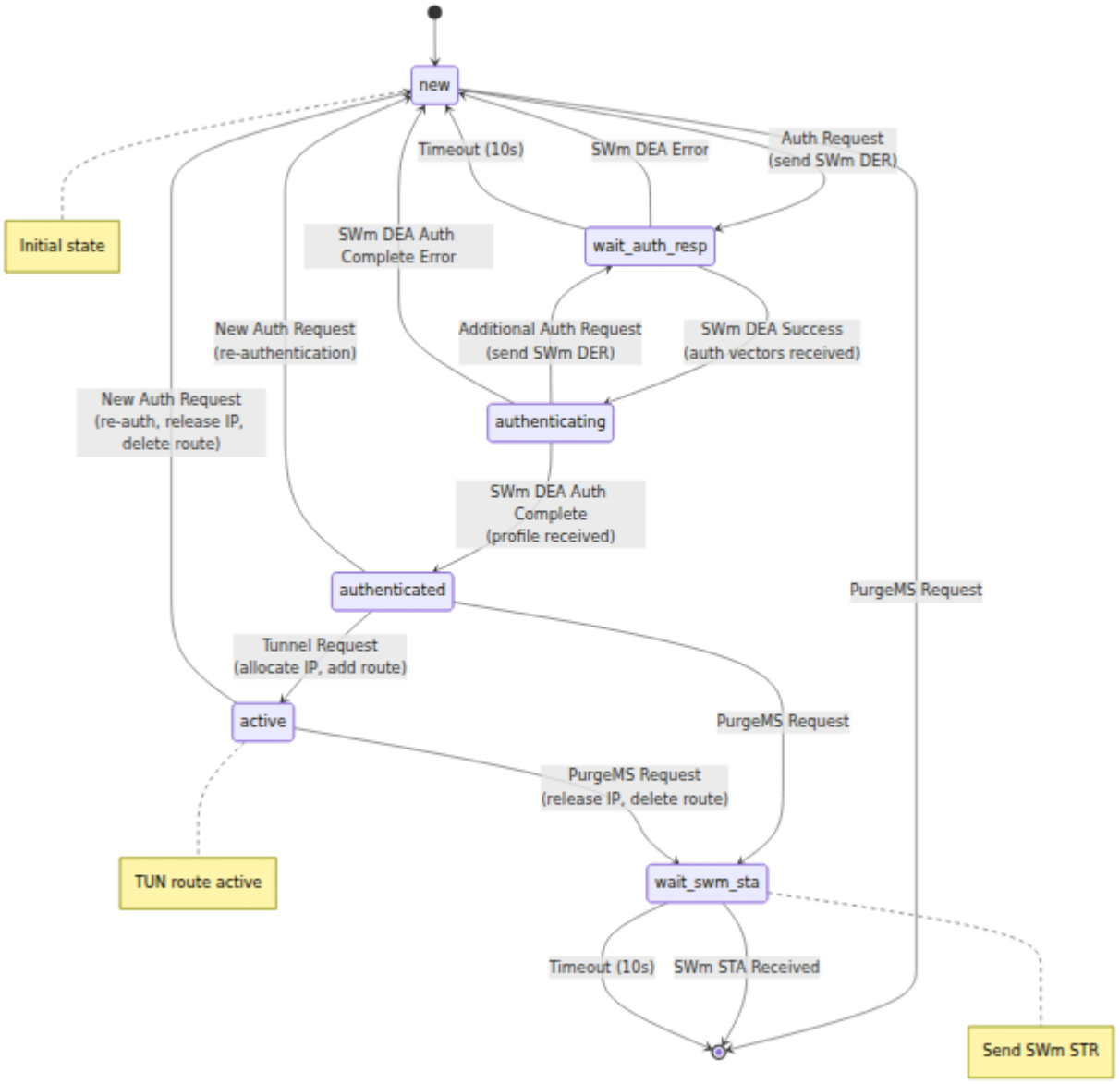
AAA UE FSM State Reference

State	Description	Waiting For
`new`	Initial state. No active AAA session.	Diameter auth request
`wait_swx_maa`	SWx MAR sent to HSS for EAP-AKA vectors.	SWx MAA from HSS
`wait_swx_saa`	SWx SAR sent to HSS for server assignment.	SWx SAA from HSS
`authenticated`	Both ePDG and PGW sessions may be active. Tracks dual session state.	Session events
`auth_wait_swx_saa`	SWx SAR sent for PGW update or user deregistration.	SWx SAA from HSS
`dereg_net_wait_s6b_asa`	HSS-initiated deregistration. S6b ASR sent to PGW.	S6b ASA from PGW
`dereg_net_wait_swm_asa`	S6b teardown complete. SWm ASR sent to ePDG.	SWm ASA from ePDG

Call Flows

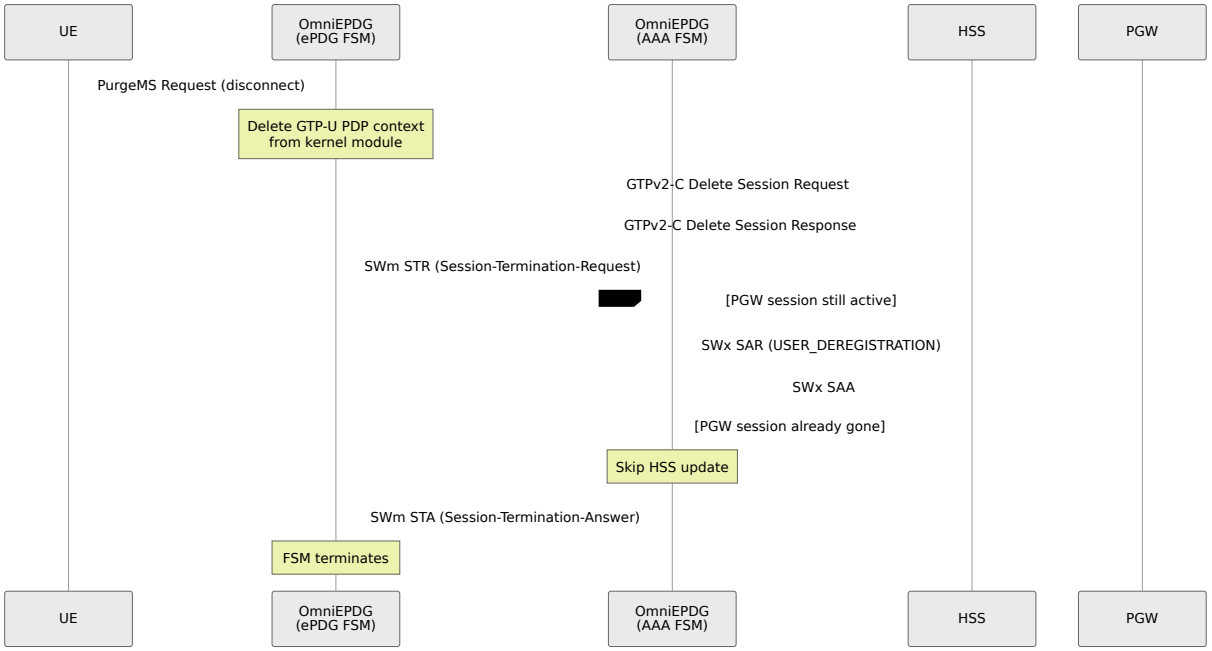
GTP Mode: Successful Session Establishment

This sequence shows a complete successful session from EAP-AKA authentication through to an active GTP tunnel.



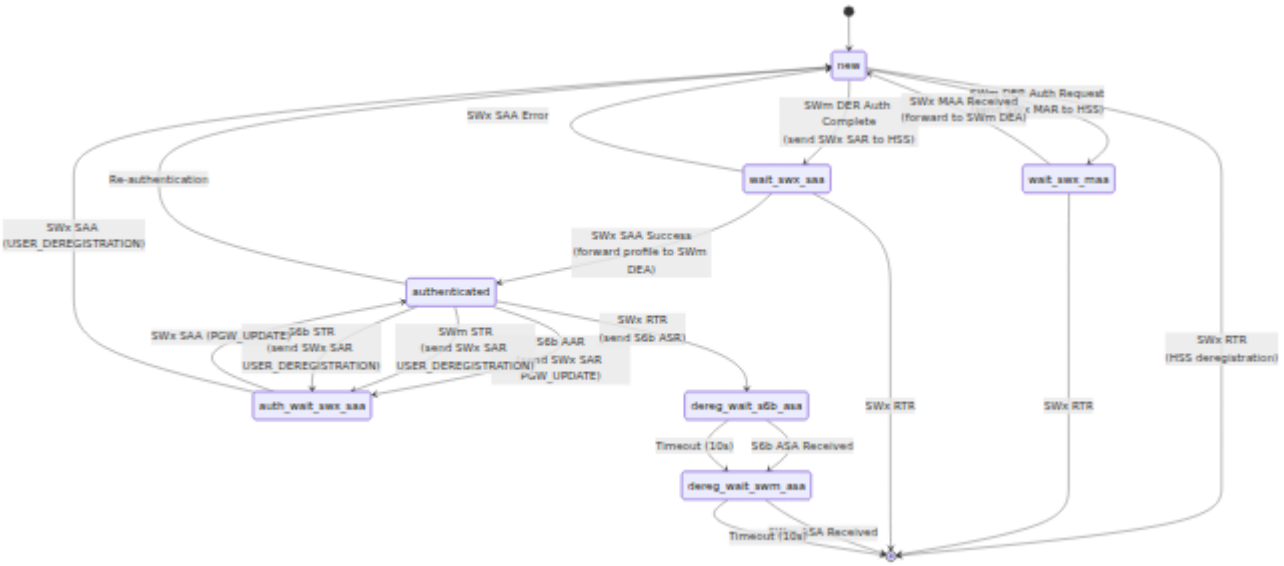
GTP Mode: UE-Initiated Session Teardown

When the UE disconnects (e.g., switches from WiFi to cellular or user hangs up).



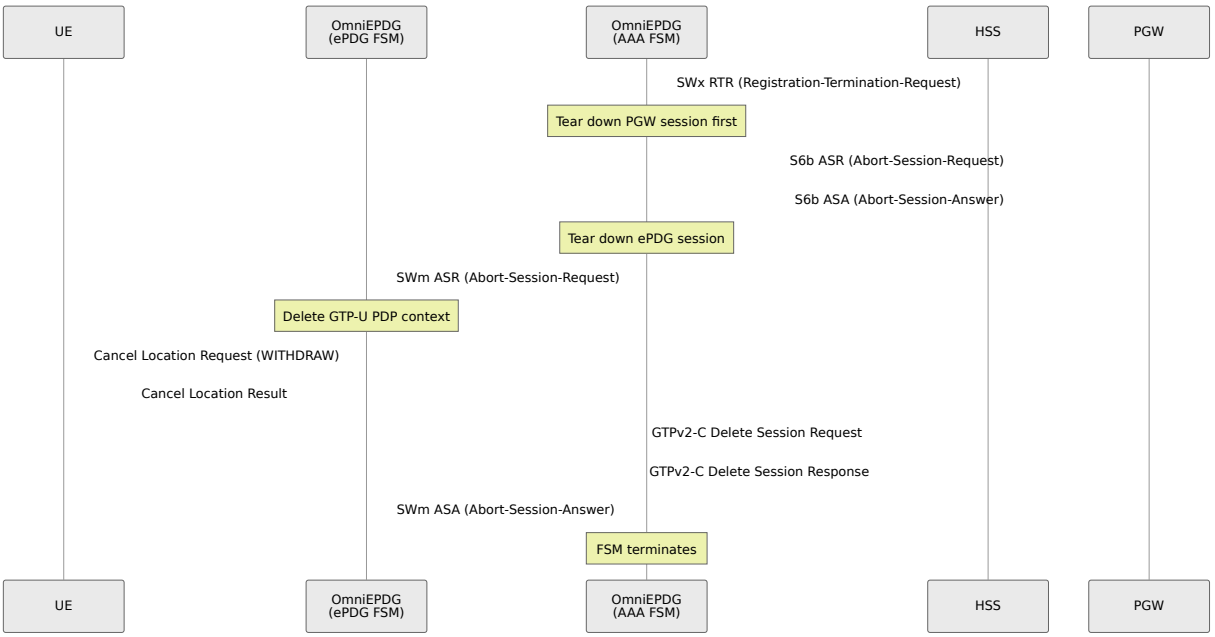
GTP Mode: PGW-Initiated Session Teardown

When the PGW terminates the session (e.g., policy violation, timeout, or administrative action).



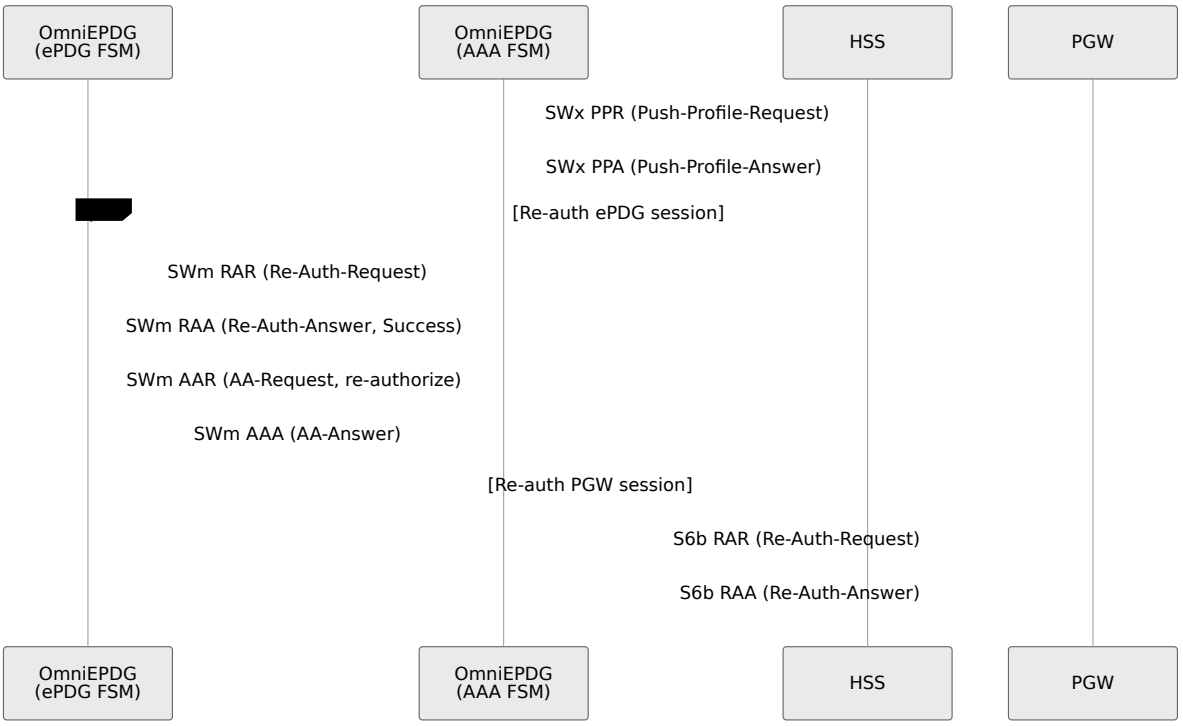
GTP Mode: Network (HSS) Initiated Deregistration

When the HSS revokes a subscriber's registration (e.g., subscription change, fraud detection, or administrative action).



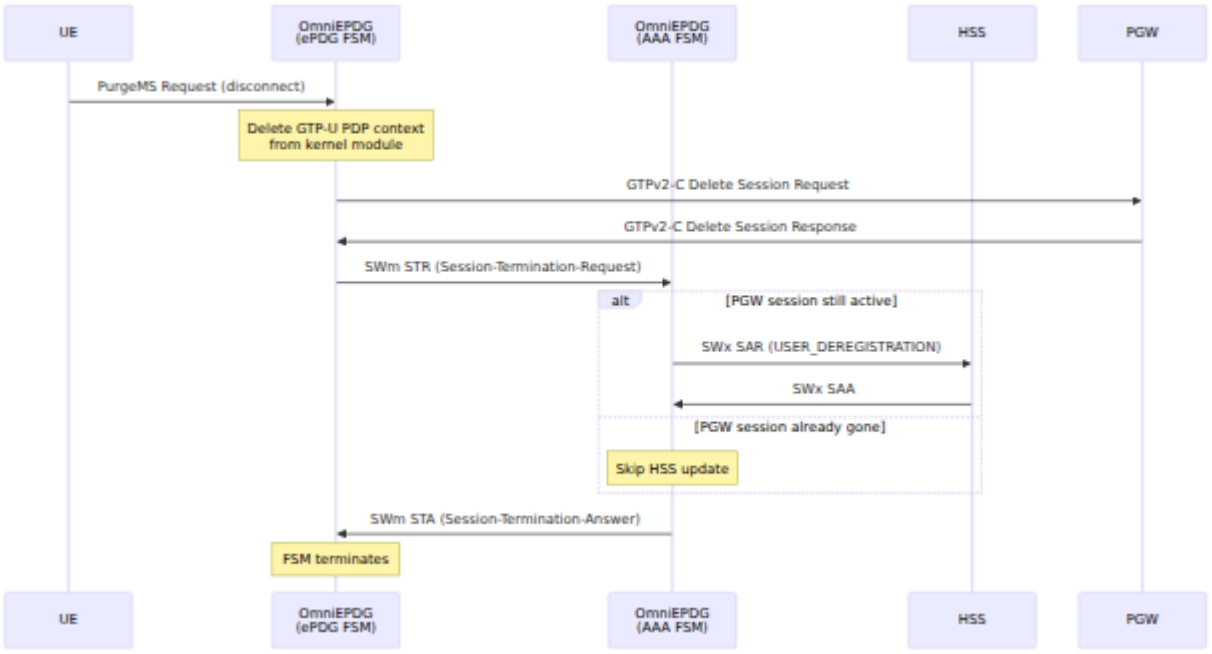
GTP Mode: HSS Profile Push and Re-Authentication

When the HSS pushes an updated subscriber profile, OmniEPDG triggers re-authentication on both the ePDG (SWm) and PGW (S6b) sessions per 3GPP TS 29.273 Section 8.1.2.3.3.



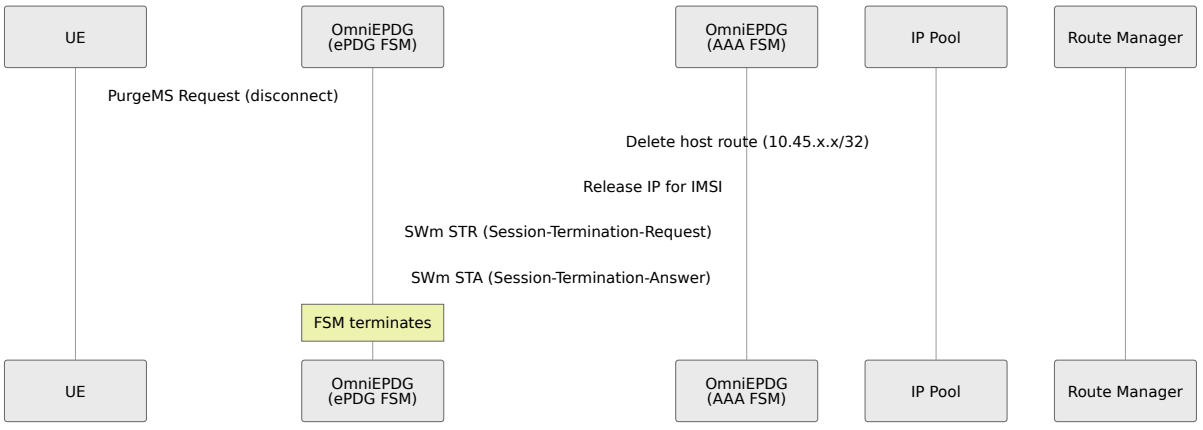
Simple VPN Mode: Successful Session Establishment

In Simple VPN mode, the session establishment is shorter. After EAP-AKA authentication, the ePDG FSM allocates an IP from the local pool and sets up a host route on the TUN interface, bypassing all PGW interaction. If `skip_sar` is enabled, the SAR/SAA exchange with the HSS is also skipped.



Simple VPN Mode: UE-Initiated Session Teardown

When the UE disconnects in Simple VPN mode, the FSM releases the allocated IP address and removes the host route.



Diameter Application IDs

Application ID	Interface	Vendor ID	Description	Reference
16777265	SWx	10415 (3GPP)	ePDG ↔ HSS authentication and subscriber management	3GPP TS 29.273
16777272	S6b	10415 (3GPP)	AAA ↔ PGW session authorization	3GPP TS 29.273

Diameter Result Codes

OmniEPDG maps Diameter result codes to internal cause values for cross-protocol error propagation.

Standard Result Codes

Result Code	Name	Meaning
2001	DIAMETER_SUCCESS	Operation completed successfully
2002	DIAMETER_LIMITED_SUCCESS	Operation partially successful

3GPP Experimental Result Codes

Result Code	Name	Meaning
4181	DIAMETER_AUTHENTICATION_DATA_UNAVAILABLE	HSS temporarily cannot provide auth data
5001	DIAMETER_ERROR_USER_UNKNOWN	Subscriber IMSI not found in HSS
5002	DIAMETER_UNKNOWN_SESSION_ID	Session not found (used for stale STR/AAR)
5003	DIAMETER_AUTHORIZATION_REJECTED	Subscriber not authorized for service
5004	DIAMETER_ERROR_ROAMING_NOT_ALLOWED	Roaming restrictions apply
5005	DIAMETER_MISSING_AVP	Required AVP missing from message
5012	DIAMETER_UNABLE_TO_COMPLY	Generic processing

Result Code	Name	Meaning
		failure
5420	DIAMETER_ERROR_UNKNOWN_EPS_SUBSCRIPTION	No EPS subscription found
5421	DIAMETER_ERROR_RAT_NOT_ALLOWED	Access technology not permitted
5422	DIAMETER_ERROR_EQUIPMENT_UNKNOWN	Device IMEI not recognized

GTPv2-C Cause Codes (GTP Mode Only)

OmniEPDG handles the following GTPv2-C cause codes in Create/Delete Session responses from the PGW. Codes 1-15 are informational, 16-63 indicate success, and 64+ indicate errors. See 3GPP TS 29.274 Section 8.4.

Success Causes

Code	Name	Description
16	Request Accepted	Operation completed successfully
17	Request Accepted Partially	Partial success
18	New PDN Type (Network Preference)	PDN type changed due to network preference
19	New PDN Type (Single Address Bearer)	PDN type changed due to single address bearer restriction

Error Causes (Selected)

Code	Name	Description
64	Context Not Found	Session context not found on PGW
73	No Resources Available	PGW resource exhaustion
78	Missing or Unknown APN	Requested APN not configured on PGW
82	Denied in RAT	Access technology not permitted
84	All Dynamic Addresses Occupied	IP address pool exhausted on PGW
92	User Authentication Failed	Authentication failure at PGW
93	APN Access Denied	Subscriber not authorized for APN
96	IMSI/IMEI Not Known	Subscriber identity not recognized
109	Invalid Peer	Peer validation failed
113	APN Congestion	APN overloaded
120	GTP-C Entity Congestion	PGW control plane overloaded

NAI Format

OmniEPDG identifies subscribers using the Network Access Identifier (NAI) format defined in 3GPP TS 23.003 Section 19:

```
<prefix><IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

Identity Prefix and Authentication Type

The NAI prefix determines the EAP authentication method per 3GPP TS 23.003:

Prefix	Authentication Type	Description
`0`	EAP-AKA	Standard AKA authentication (most common for WiFi calling)
`6`	EAP-AKA'	Enhanced AKA authentication with network binding

OmniEPDG automatically selects the authentication method based on the UE's identity prefix. Most UEs use prefix `0` (EAP-AKA) for WiFi calling.

OmniEPDG extracts the IMSI from the NAI by parsing everything between the prefix and the `@` symbol. The IMSI is then used as the primary key for all per-subscriber state machines and signaling operations.

Cryptographic Algorithms

OmniEPDG implements cryptographic algorithms per 3GPP TS 33.402 and RFC 7296 (IKEv2).

IKEv2 Encryption Algorithms

Algorithm	ID	Key Size	Status	Reference
AES-CBC	12	128, 192, 256 bits	Supported (default: 256)	RFC 3602
AES-GCM-16	20	128, 192, 256 bits	Supported	RFC 5282
AES-GCM-12	19	128, 192, 256 bits	Supported	RFC 5282
AES-GCM-8	18	128, 192, 256 bits	Supported	RFC 5282
3DES	3	192 bits	Supported (legacy)	RFC 2451

IKEv2 Integrity Algorithms

Algorithm	ID	Key Size	ICV Size	Status	Reference
HMAC-SHA2-256-128	12	256 bits	128 bits	Supported (default)	RFC 4868
HMAC-SHA2-384-192	13	384 bits	192 bits	Supported	RFC 4868
HMAC-SHA2-512-256	14	512 bits	256 bits	Supported	RFC 4868
HMAC-SHA1-96	2	160 bits	96 bits	Supported (legacy)	RFC 2404
HMAC-MD5-96	1	128 bits	96 bits	Supported (legacy)	RFC 2403

IKEv2 PRF Algorithms

Algorithm	ID	Output Size	Status	Reference
PRF-HMAC-SHA2-256	5	256 bits	Supported (default)	RFC 4868
PRF-HMAC-SHA2-384	6	384 bits	Supported	RFC 4868
PRF-HMAC-SHA2-512	7	512 bits	Supported	RFC 4868
PRF-HMAC-SHA1	2	160 bits	Supported (legacy)	RFC 2104
PRF-HMAC-MD5	1	128 bits	Supported (legacy)	RFC 2104

IKEv2 Diffie-Hellman Groups

Group	ID	Size	Status	Reference
MODP-2048	14	2048 bits	Supported (default)	RFC 3526
MODP-1024	2	1024 bits	Supported (legacy)	RFC 2409
MODP-1536	5	1536 bits	Supported	RFC 3526
MODP-3072	15	3072 bits	Supported	RFC 3526
MODP-4096	16	4096 bits	Supported	RFC 3526
ECP-256	19	256 bits	Supported	RFC 5903
ECP-384	20	384 bits	Supported	RFC 5903
ECP-521	21	521 bits	Supported	RFC 5903
Curve25519	31	256 bits	Supported	RFC 8031
Curve448	32	448 bits	Supported	RFC 8031

ESP (Child SA) Algorithms

The ESP tunnel uses the same encryption and integrity algorithms negotiated during IKEv2 CREATE_CHILD_SA.

Default ESP configuration:

- Encryption: AES-CBC-256 (32-byte key, 16-byte IV)
- Integrity: HMAC-SHA2-256-128 (32-byte key, 16-byte ICV)

EAP-AKA Cryptographic Functions

Function	Algorithm	Reference
MK derivation	SHA-1	RFC 4187 Section 7
PRF+ key expansion	FIPS 186-2 PRF (SHA-1)	RFC 4187 Appendix D
AT_MAC	HMAC-SHA1-128	RFC 4187 Section 10.15
Milenage (f1-f5)	AES-128	3GPP TS 35.206

EAP-AKA' Cryptographic Functions

Function	Algorithm	Reference
CK'/IK' derivation	HMAC-SHA-256	RFC 5448 Section 3.3
MK derivation	SHA-256	RFC 5448 Section 3.4
AT_MAC	HMAC-SHA256-128	RFC 5448 Section 3.1

3GPP Compliance

OmniEPDG implements all mandatory cryptographic algorithms specified in 3GPP TS 33.402 Section 8:

Requirement	Algorithm	Status
IKEv2 Encryption (mandatory)	AES-CBC-128	✓ Supported
IKEv2 Integrity (mandatory)	HMAC-SHA2-256-128	✓ Supported (default)
IKEv2 PRF (mandatory)	PRF-HMAC-SHA-256	✓ Supported (default)
IKEv2 DH (mandatory)	Group 14 (MODP-2048)	✓ Supported (default)
ESP Encryption (mandatory)	AES-CBC-128/256	✓ Supported
ESP Integrity (mandatory)	HMAC-SHA2-256-128	✓ Supported (default)
EAP-AKA	RFC 4187	✓ Implemented
EAP-AKA'	RFC 5448	✓ Implemented

PDP Address Types (GTP Mode Only)

OmniEPDG supports the following PDP address types as defined in 3GPP TS 29.274 Section 8.14. In Simple VPN mode, only IPv4 addresses are allocated from the local pool.

Type	Description	GTPv2-C PAA Format
IPv4	IPv4-only bearer	4-byte IPv4 address
IPv6	IPv6-only bearer	1-byte prefix length + 16-byte IPv6 address
IPv4v6	Dual-stack bearer	1-byte prefix length + 16-byte IPv6 + 4-byte IPv4

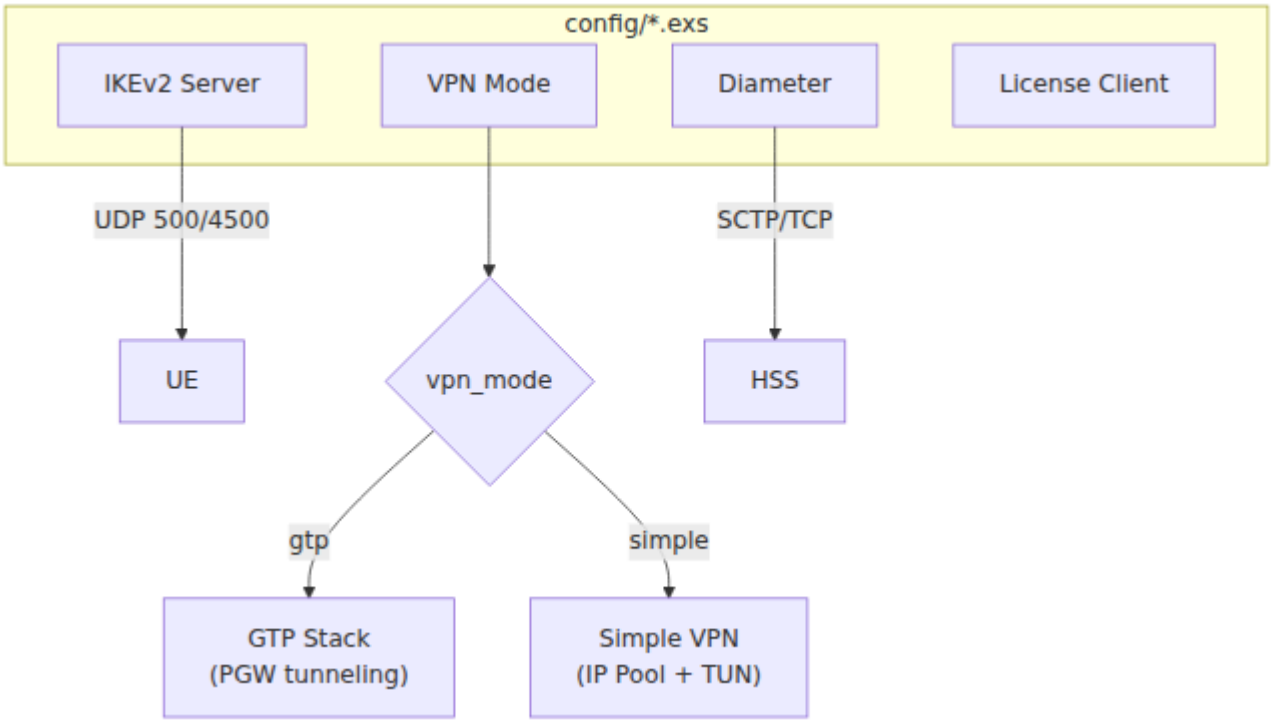
OmniEPDG Configuration Reference

OmniEPDG is configured via `config/runtime.exs` and environment variables. All customer-facing configuration is performed at runtime - compile-time defaults are built into the release and not exposed.

For containerized deployments, use environment variables as documented in the Environment Variables Reference section.

Table of Contents

Configuration Structure



Configuration File

All configuration is done in `config/runtime.exs`. This file is read when OmniEPDG starts and supports environment variable substitution for containerized deployments.

Configuration Example

```
# config/runtime.exs
config :omniepdg,
  # IKEv2 server settings
  listen_ip: {0, 0, 0, 0},
  port_500: 500,
  port_4500: 4500,

  # VPN mode: :simple (local breakout) or :gtp (PGW via GTP-C)
  vpn_mode: :simple,

  # Simple VPN mode settings
  simple_vpn: [
    pool_ipv4: "10.45.0.0/16",
    pool_ipv6: "2001:db8::/32",
    dns_servers_ipv4: ["8.8.8.8", "8.8.4.4"],
    dns_servers_ipv6: ["2001:4860:4860::8888",
"2001:4860:4860::8844"]
  ]

# Control panel configuration
config :control_panel,
  parent_application: :omniepdg,
  parent_application_readable_name: "OmniEPDG",
  use_additional_pages: [
    {OmniEpdg.Web.DashboardLive, "/", "Dashboard"},
    {OmniEpdg.Web.SessionsLive, "/sessions", "Sessions"},
    {OmniEpdg.Web.DiameterLive, "/diameter", "Diameter"}
  ]

# Diameter configuration (runtime.exs)
config :diameter_ex,
  diameter: %{
    service_name: :omniepdg,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    host: "epdg",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    product_name: "OmniEPDG",
    vendor_id: 10415,
    applications: [
      %{application_name: :swx, application_id: 16_777_265,
```

```
vendor_id: 10415},
      %{application_name: :s6b, application_id: 16_777_272,
vendor_id: 10415}
    ],
    peers: [
      %{host: "hss", ip: "127.0.0.1", port: 3868, transport: :tcp}
    ]
  }

# License client configuration (runtime.exs)
config :license_client,
  server_url: "https://license.example.com/api",
  product: "omniepdg"
```

IKEv2 Server Parameters

The IKEv2 server handles the SWu interface between UEs and OmniEPDG. It terminates IPSec tunnels and performs EAP-AKA authentication.

Parameter	Type	Required	Defau
`listen_ip`	Tuple	No	`{0, 0, 0, 0}`
`port_500`	Integer	No	`500`
`port_4500`	Integer	No	`4500`
`cert_file`	String	Yes	`/etc/omniepdg/ce`
`key_file`	String	Yes	`/etc/omniepdg/ce`

Parameter	Type	Required	Defau
`session_inactivity_timeout_ms`	Integer	No	`300000`

Authentication Security Parameters

OmniEPDG includes built-in protection against brute-force attacks and geographic access control. See the Security Guide for detailed documentation.

Rate Limiting Parameters

```
config :omniepdg,
  # Per-IP rate limiting
  auth_rate_limit_per_ip: 10,
  auth_rate_limit_ip_window_ms: 60_000,
  auth_rate_limit_ip_block_ms: 300_000,

  # Per-IMSI rate limiting
  auth_rate_limit_per_imsi: 5,
  auth_rate_limit_imsi_window_ms: 60_000,
  auth_rate_limit_imsi_block_ms: 600_000
```

Parameter	Type	Required	Default	Desc
`auth_rate_limit_per_ip`	Integer	No	`10`	Maxin failed attem IP bef blocki
`auth_rate_limit_ip_window_ms`	Integer	No	`60000`	Slidin windo count failure (millis
`auth_rate_limit_ip_block_ms`	Integer	No	`300000`	Block for IPs excee thresh minut
`auth_rate_limit_per_imsi`	Integer	No	`5`	Maxin failed attem IMSI b blocki
`auth_rate_limit_imsi_window_ms`	Integer	No	`60000`	Slidin windo count failure (millis
`auth_rate_limit_imsi_block_ms`	Integer	No	`600000`	Block for IM excee thresh minut

GeoIP Parameters

```
config :omniepdg,
  geoip_enabled: false,
  geoip_database_path: "/etc/omniepdg/GeoLite2-Country.mmdb",
  geoip_mode: :whitelist,
  geoip_countries: ["AU", "NZ"],
  geoip_allow_unknown: false,
  geoip_fail_open: true
```

Parameter	Type	Required	Default
`geoip_enabled`	Boolean	No	`false`
`geoip_database_path`	String	No	`"/etc/omniepdg/GeoLite2-Country.mmdb"`
`geoip_mode`	Atom	No	`:whitelist`
`geoip_countries`	List	No	`[]`
`geoip_allow_unknown`	Boolean	No	mode-dependent

Parameter	Type	Required	Default
`geoip_fail_open`	Boolean	No	`true`

VPN Mode Selection

Parameter	Type	Required	Default	Env Var	Descriptio
`vpn_mode`	Atom	No	`:simple`	`EPDG_VPN_MODE`	Operational mode: `:simple` for local IP breakout via TUN interface, `:gtp` for tunneling through a PGW via GTPv2-C/GTP-U. See the Operations Guide for a detailed comparison

Simple VPN Parameters

The `simple_vpn` configuration block controls IP allocation and DNS for Simple VPN mode. OmniEPDG supports both IPv4 and IPv6 address pools.

```
config :omniepdg,
  simple_vpn: [
    pool_ipv4: "10.45.0.0/16",
    pool_ipv6: "2001:db8::/32",
    dns_servers_ipv4: ["8.8.8.8", "8.8.4.4"],
    dns_servers_ipv6: ["2001:4860:4860::8888",
"2001:4860:4860::8844"],
    p_cscf_ipv4: ["10.4.12.165"],
    p_cscf_ipv6: [],
    mtu: 1400,
    nat_enabled: true
  ]
```

Parameter	Type	Required	Default	
`pool_ipv4`	String	Yes	`"10.45.0.0/16"`	IF p n IF p
`pool_ipv6`	String	No	`"2001:db8::/32"`	IF p n d s
`dns_servers_ipv4`	List	No	`["8.8.8.8", "8.8.4.4"]`	IF s p U (C O
`dns_servers_ipv6`	List	No	`["2001:4860:4860::8888", "2001:4860:4860::8844"]`	IF s p U
`p_cscf_ipv4`	List	No	`[]`	IF (a V P U IM re

Parameter	Type	Required	Default	
`p_cscf_ipv6`	List	No	`[]`	IF (I a V
`mtu`	Integer	No	`1400`	M th ir L re fr
`nat_enabled`	Boolean	No	`true`	E s tr tr m ru a o tr

Diameter Parameters

The Diameter configuration controls the SWx (HSS) and S6b (PGW) interfaces. When `diameter_enabled` is `true`, OmniEPDG starts the Diameter stack and connects to configured peers.

```
config :diameter_ex,
  diameter: %{
    service_name: :omniepdg,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    host: "epdg",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    product_name: "OmniEPDG",
    vendor_id: 10415,
    applications: [
      %{application_name: :swx, application_id: 16_777_265,
vendor_id: 10415},
      %{application_name: :s6b, application_id: 16_777_272,
vendor_id: 10415}
    ],
    peers: [
      %{host: "hss", ip: "10.74.0.21", port: 3868, transport:
:tcp}
    ]
  }
```

Service Parameters

Parameter	Type	Required	Default
`service_name`	Atom	No	`:omniepdg`
`listen_ip`	String	No	`"0.0.0.0"`
`listen_port`	Integer	No	`3868`
`host`	String	Yes	`"epdg"`
`realm`	String	Yes	`"epc.mnc001.mcc001.3gppnetwork.or`

Parameter	Type	Required	Default
`product_name`	String	No	`"OmniEPDG"`
`vendor_id`	Integer	No	`10415`

Peer Parameters

Each entry in the `peers` list defines a Diameter peer connection (typically to the HSS).

Parameter	Type	Required	Default	Env Var	Description
`host`	String	Yes	-	`HSS_HOST`	Peer's Diameter identity (Origin-Host). Must match the peer's configured identity.
`ip`	String	Yes	-	`HSS_IP`	Peer's IP address for TCP/SCTP connection.
`port`	Integer	No	`3868`	`HSS_PORT`	Peer's Diameter port.
`transport`	Atom	No	`:tcp`	-	Transport protocol: `:tcp` or `:sctp`.

Application IDs

Application	ID	Vendor ID	Interface	Reference
SWx	16777265	10415	ePDG ↔ HSS	3GPP TS 29.273
S6b	16777272	10415	AAA ↔ PGW	3GPP TS 29.273

License Client Configuration

The license client validates OmniEPDG against a license server.

```
config :license_client,
  server_url: "https://license.example.com/api",
  product: "omniepdg"
```

Parameter	Type	Required	Default	Env Var	D
`server_url`	String	Yes	-	`LICENSE_SERVER_URL`	U lic se er
`product`	String	No	`"omniepdg"`	-	Pr id lic va

Control Panel Configuration

The web control panel provides monitoring and management capabilities.

```
config :control_panel,
  port: 4000
```

Parameter	Type	Required	Default	Env Var	Des
`port`	Integer	No	`4000`	`CONTROL_PANEL_PORT`	HTTP for t cont pane inter

Prometheus Metrics Configuration

OmniEPDG exposes Prometheus metrics via HTTP for monitoring and alerting.

```
config :omniepdg,
  prometheus: %{
    port: 9568
  }
```

Parameter	Type	Required	Default	Env Var	Descrip
`port`	Integer	No	`9568`	`PROMETHEUS_PORT`	HTTP pc for Prometh metrics endpoin (`/metri`

Exposed Metrics

Counter Metrics (Event-driven):

- `epdg_ikev2_session_initiated_count` - IKE_SA_INIT exchanges started
- `epdg_ikev2_session_established_count` - IKE SAs successfully established
- `epdg_ikev2_session_failed_count` - IKE SA establishment failures (by reason)
- `epdg_eap_identity_count` - EAP Identity requests
- `epdg_eap_aka_challenge_count` - EAP-AKA Challenges sent
- `epdg_eap_aka_success_count` - EAP-AKA authentications succeeded
- `epdg_eap_aka_failure_count` - EAP-AKA authentications failed (by reason)
- `epdg_eap_aka_sync_failure_count` - EAP-AKA SQN sync failures
- `epdg_diameter_swx_mar_count` - Multimedia-Auth-Requests (by result)
- `epdg_diameter_swx_sar_count` - Server-Assignment-Requests (by result)
- `epdg_diameter_s6b_aar_count` - AA-Requests handled (by result)
- `epdg_diameter_s6b_str_count` - Session-Termination-Requests
- `epdg_session_created_count` - Sessions created (by vpn_mode)
- `epdg_session_terminated_count` - Sessions terminated (by reason)
- `epdg_esp_packets_in_count` - ESP packets decrypted

- `epdg_esp_packets_out_count` - ESP packets encrypted
- `epdg_ip_allocated_count` - IP addresses allocated (by type)
- `epdg_ip_released_count` - IP addresses released (by type)

Gauge Metrics (Polled every 5s):

- `epdg_sessions_active_count` - Total active sessions
- `epdg_sessions_by_state_count` - Sessions by FSM state
- `epdg_ip_pool_allocated_count` - IPs currently allocated
- `epdg_ip_pool_available_count` - IPs available in pool
- `epdg_ip_pool_utilization_ratio` - Pool utilization (0.0-1.0)
- `epdg_diameter_swx_pending_count` - Pending SWx requests
- `epdg_diameter_s6b_active_sessions_count` - Active S6b sessions

Histogram Metrics (Latency tracking):

- `epdg_auth_duration_ms` - Full auth flow duration
- `epdg_diameter_swx_mar_latency_ms` - MAR response time
- `epdg_diameter_swx_sar_latency_ms` - SAR response time
- `epdg_session_duration_seconds` - Session lifetime

VM Metrics:

- `vm_memory_total` - Total VM memory
- `vm_memory_processes` - Process memory
- `vm_memory_binary` - Binary memory
- `vm_memory_ets` - ETS table memory
- `vm_system_info_process_count` - Running processes
- `vm_system_info_port_count` - Open ports
- `vm_statistics_run_queue` - Scheduler run queue

Prometheus Scrape Configuration

```
scrape_configs:
  - job_name: 'omniepdg'
    static_configs:
      - targets: ['localhost:9568']
```

Timeout Reference

All internal FSM timeouts are hardcoded. These govern how long state machines wait for responses before considering them failed.

Timeout	Value	Mode	Context	Description
GTP answer	10,000 ms	GTP	ePDG UE FSM	Maximum wait for GTPv2-C Create/Delete Session Response from PGW.
SWm answer	10,000 ms	Both	ePDG UE FSM	Maximum wait for internal Diameter SWm response (DER/DEA, STR/STA).
S6b answer	10,000 ms	GTP	AAA UE FSM	Maximum wait for Diameter S6b response (ASR/ASA).

Environment Variables Reference

Environment variables are read in `config/runtime.exs` and override compile-time defaults.

IKEv2 Server

Variable	Default	Description
`EPDG_LISTEN_IP`	`"0.0.0.0"`	IKEv2 server bind address (dotted decimal format, e.g., `"10.0.0.1"`).
`EPDG_PORT_500`	`"500"`	IKE protocol port.
`EPDG_PORT_4500`	`"4500"`	IKE NAT-Traversal port.
`EPDG_CERT_FILE`	`"/etc/omniepdg/certs/epdg.crt"`	Path to IKEv2 server certificate (PEM).
`EPDG_KEY_FILE`	`"/etc/omniepdg/certs/epdg.key"`	Path to IKEv2 server private key (PEM).
`EPDG_SESSION_TIMEOUT`	`"300000"`	Session inactivity timeout in milliseconds.

VPN Mode

Variable	Default	Description
`EPDG_VPN_MODE`	`"simple"`	VPN mode: `"simple"` or `"gtp"`.

Diameter

Variable	Default	Description
`DIA_LISTEN_IP`	`"0.0.0.0"`	Diameter listener bind address.
`DIA_LISTEN_PORT`	`"3868"`	Diameter listener port.
`DIA_HOST`	`"epdg"`	Diameter Origin-Host (without realm).
`DIA_REALM`	`"epc.mnc001.mcc001.3gppnetwork.org"`	Diameter Origin-Realm.

HSS Peer

Variable	Default	Description
`HSS_HOST`	`"hss"`	HSS Diameter identity (Origin-Host).
`HSS_IP`	`"127.0.0.1"`	HSS IP address.
`HSS_PORT`	`"3868"`	HSS Diameter port.

License, Control Panel, and Metrics

Variable	Default	Description
`LICENSE_SERVER_URL`	-	License server API URL (required).
`CONTROL_PANEL_PORT`	`"4000"`	Control panel HTTP port.
`PROMETHEUS_PORT`	`"9568"`	Prometheus metrics HTTP port (`/metrics` endpoint).

Example: Docker Compose

```yaml
services:
  omniepdg:
    image: omniepdg:latest
    environment:
      # IKEv2
      EPDG_LISTEN_IP: "0.0.0.0"
      EPDG_CERT_FILE: "/certs/epdg.crt"
      EPDG_KEY_FILE: "/certs/epdg.key"

      # VPN mode
      EPDG_VPN_MODE: "simple"

      # Diameter
      DIA_HOST: "epdg"
      DIA_REALM: "epc.mnc001.mcc001.3gppnetwork.org"
      HSS_HOST: "hss"
      HSS_IP: "10.74.0.21"
      HSS_PORT: "3868"

      # License
      LICENSE_SERVER_URL: "https://license.example.com/api"

      # Control panel
      CONTROL_PANEL_PORT: "4000"

      # Prometheus metrics
      PROMETHEUS_PORT: "9568"
    ports:
      - "500:500/udp"
      - "4500:4500/udp"
      - "4000:4000"
      - "9568:9568"
    volumes:
      - ./certs:/certs:ro
    cap_add:
      - NET_ADMIN
```

OmniEPDG Control Panel

OmniEPDG includes a web-based control panel for real-time monitoring of sessions, Diameter peers, and system logs. The control panel provides live-updating views without page refreshes.

Table of Contents

Accessing the Control Panel

The control panel is served on the configured HTTP port (default 4000):

```
http://<host>:4000/dashboard
```

Navigation

The control panel provides a sidebar with links to all views:

Path	View	Description
`/dashboard`	Dashboard	System overview and quick links
`/sessions`	Sessions	Active UE session list
`/diameter`	Diameter Peers	Diameter peer connection status
`/logs`	Logs	Real-time log streaming
`/docs`	Docs	Integrated documentation browser
`/resources`	Resources	BEAM VM and application information
`/configuration`	Configuration	System configuration viewer

Dashboard

The Dashboard provides a high-level overview of OmniEPDG status with key metrics and quick navigation.

Statistics Cards

The dashboard displays four primary statistics:

Statistic	Description
Active Sessions	Current number of established UE sessions
Data Received (UL)	Total bytes received from UEs (uplink direction)
Data Sent (DL)	Total bytes sent to UEs (downlink direction)
Diameter Peers	Connected peers / total configured peers

Data values automatically scale to appropriate units (B, KB, MB, GB).

Quick Links

Direct navigation to detailed views:

- **View Sessions** - Navigate to the Sessions view for detailed UE information
- **Diameter Peers** - Navigate to the Diameter Peers view for connectivity status
- **System Logs** - Navigate to the Logs view for real-time log streaming
- **Configuration** - Navigate to the Configuration view for system settings

System Information

Displays current operational configuration:

Field	Description
VPN Mode	Current mode: `GTP` or `SIMPLE`
IKEv2 Ports	Standard ports: 500 (IKE), 4500 (NAT-T)
Diameter Status	Whether Diameter signaling is enabled
IP Pool (IPv4)	Configured IP pool CIDR (Simple VPN mode only)

Auto-Refresh

The dashboard automatically refreshes every second to display current statistics.

Sessions View

The Sessions view displays all active UE sessions with detailed information for each subscriber.

The Sessions view shows active UE connections with real-time traffic statistics and session duration.

Session List

Each session row displays:

Column	Description
IMSI	Subscriber's International Mobile Subscriber Identity
UE IP	Assigned IPv4/IPv6 address
SOURCE	UE's external IP and port (NAT address)
APN	Access Point Name for the connection
STATUS	Current session state (Active/Inactive)
DURATION	Time since session establishment
TRAFFIC	Bytes received / sent (UL/DL)

Status Indicators

Sessions display status with color-coded badges:

Status	Color	Description
Active	Green	Session fully established and operational
Connecting	Yellow	Session establishment in progress
Inactive	Red	Session terminated or failed

Session Details

Click any session row to expand detailed information:

Expanded session view showing IMSI, NAI, network configuration, and traffic statistics.

Session Section

Field	Description
IMSI	Full IMSI value
NAI	Network Access Identifier (3GPP format)
UE IP	Assigned IPv4/IPv6 address
Source	UE's external IP and port (NAT address)
APN	Access Point Name for the PDN connection
Child SA SPI	IPSec Child SA Security Parameter Index

Network & Timing Section

Field	Description
DNS	DNS servers provided to the UE
P-CSCF	Proxy-CSCF servers for IMS signaling
Connected	Timestamp when session was established
Last Activity	Timestamp of most recent packet activity
Duration	Time since session establishment

Traffic Section

Field	Description
Bytes In (UL)	Total bytes received from UE (uplink)
Bytes Out (DL)	Total bytes sent to UE (downlink)
Packets In	Total packets received from UE
Packets Out	Total packets sent to UE

Empty State

When no sessions are active, the view displays:

- "No active sessions" message
- Indicates whether UE connections should be attempted

Auto-Refresh

The session list automatically refreshes every second.

Diameter Peers View

The Diameter Peers view displays the status of all configured Diameter peers (HSS for SWx, PGW for S6b).

Peer List

Each peer row displays:

Column	Description
Peer	Diameter Identity (Origin-Host)
Realm	Diameter Realm (Origin-Realm)
IP Address	Transport address in format `protocol://ip:port`
Status	Connection status

Status Indicators

Status	Color	Description
Connected	Green	Diameter peer connection established
Disconnected	Red	Peer not connected
Unknown	Gray	Status cannot be determined

Peer Count Summary

The header displays aggregate counts:

- **X Connected** - Number of peers with active connections
- **Y Disconnected** - Number of peers without connections

Peer Details

Click any peer row to expand detailed information:

Field	Description
Connection Initiation	Who initiates: `local` (we connect to peer) or `remote` (peer connects to us)
Transport	Protocol: `tcp` or `sctp`
Product Name	Peer's advertised product name from CER/CEA
Advertised Applications	Diameter Application IDs supported by the peer

Empty State

When no peers are configured, the view displays:

- "No Diameter Peers configured" if Diameter is enabled
- "Diameter is disabled" with configuration hint if disabled

Auto-Refresh

The peer list automatically refreshes every second.

Logs View

The Logs view provides real-time streaming of system logs with filtering and search capabilities.

Log Display

Logs appear in a scrolling container with newest entries at the bottom. Each log entry displays:

Element	Description
Timestamp	When the log entry was generated
Level	Severity level with color coding
Message	Log message content

Log Levels

Logs are color-coded by severity:

Level	Color	Description
debug	Gray	Detailed diagnostic information
info	Blue	General informational messages
warning	Yellow	Warning conditions
error	Red	Error conditions

Level Filtering

Filter logs by minimum severity level using the dropdown:

Filter	Shows
All Levels	debug, info, warning, error
Info+	info, warning, error
Warning+	warning, error
Error Only	error

Search

The search box filters logs in real-time:

- Enter any text to filter log messages
- Matching is case-insensitive
- Clears when the search box is emptied

Controls

Control	Description
Pause/Resume	Toggle log streaming on/off
Clear	Remove all displayed logs
Auto-scroll	Toggle automatic scrolling to newest entries

Log Buffer

- Maximum 1000 log entries are retained
- Oldest entries are removed when the limit is reached
- Clearing logs removes all entries from the display

Empty State

When no logs match the current filters:

- "No logs to display" message
- Verify filter settings if logs are expected

Auto-Refresh

New logs appear automatically as they are generated (when not paused).

Docs View

The Docs view provides an integrated documentation browser, allowing operators to access all OmniEPDG documentation directly from the control panel.

Document Selection

Select from available documentation files using the button bar:

Document	Description
OPERATIONS.md	Operations guide with quick start and procedures
README.md	Project overview and setup instructions
architecture.md	System architecture and call flows
configuration.md	Complete configuration reference
control-panel.md	This control panel guide
metrics.md	Prometheus metrics reference
troubleshooting.md	Common issues and resolution steps

Markdown Rendering

Documentation is rendered with full Markdown support including:

- Headers and text formatting
- Code blocks with syntax highlighting
- Tables
- Links (internal and external)
- Lists and blockquotes

Resources View

The Resources view displays BEAM VM statistics and running OTP applications.

System Metrics

Metric	Description
Memory Usage	Total memory used by the BEAM VM
BEAM Processes	Number of Erlang/Elixir processes running
Uptime	Time since the application started

Running Applications

Lists all loaded OTP applications grouped by category:

Category	Description
Main	The OmniEPDG application
System	Core Erlang/OTP and Elixir applications

Click an application to view its details including version, description, and dependencies.

Configuration View

The Configuration view displays runtime configuration and loaded applications.

Environment Information

Field	Description
Environment	Current Mix environment (Development/Production)
Elixir Version	Running Elixir version

Application List

Displays all loaded OTP applications with their versions. Select an application to view:

- Application description
- Version information
- Dependencies
- Configuration parameters

Control Panel Configuration

HTTP Port

Configure the control panel port in `config/runtime.exs`:

```
config :omniepdg, OmniEpdg.Web.Endpoint,
  http: [port: 4000]
```

Parameter	Type	Default	Description
`port`	Integer	4000	HTTP port for the control panel

Disabling the Control Panel

The control panel can be disabled by not starting the web endpoint in production if not needed. Contact your system integrator for deployment-specific configuration.

OmniEPDG Metrics Reference

OmniEPDG exposes Prometheus metrics for monitoring authentication flows, session lifecycle, Diameter signaling, and system health. Metrics are served via HTTP for Prometheus scraping.

Table of Contents

Metrics Endpoint

OmniEPDG exposes metrics at:

```
http://<host>:9568/metrics
```

The endpoint returns metrics in Prometheus exposition format, compatible with Prometheus, Grafana, and other monitoring tools.

Configuration

Configure the metrics endpoint in `config/runtime.exs`:

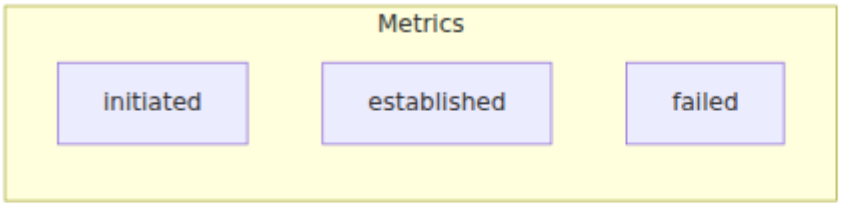
```
config :omniepdg,
  prometheus: %{
    port: 9568
  }
```

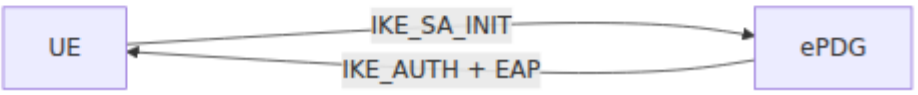
Parameter	Type	Default	Env Var	Description
`port`	Integer	`9568`	`PROMETHEUS_PORT`	HTTP port for the `/metrics` endpoint

Metric Categories

IKEv2 Session Metrics

Metrics tracking IKEv2 tunnel establishment on the SWu interface.





Metric: `epdg_ikev2_session_initiated_count` **Type**: Counter **Description**: Total IKE_SA_INIT exchanges started. Increments when a UE initiates tunnel establishment.

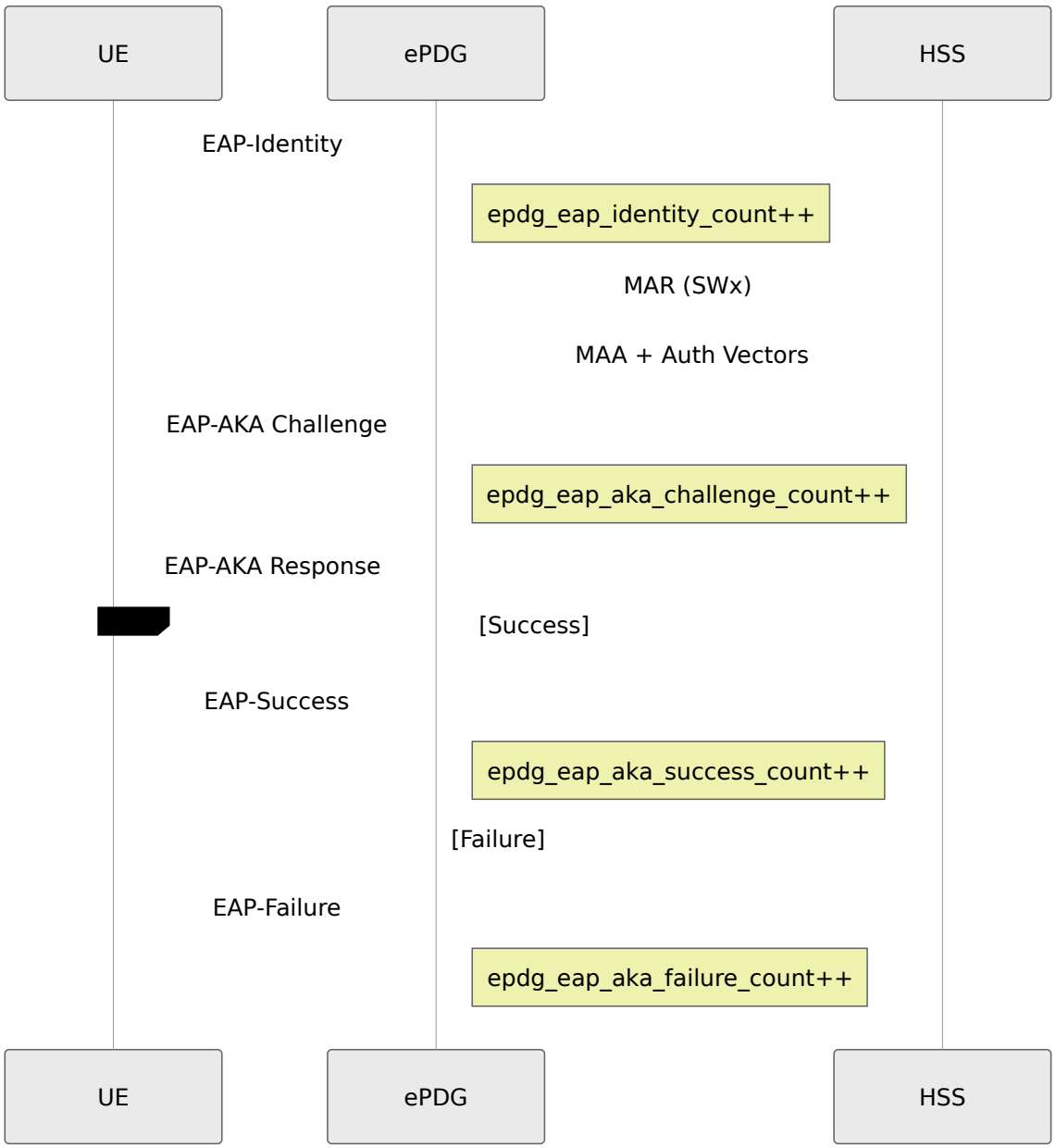
Metric: `epdg_ikev2_session_established_count` **Type**: Counter **Description**: Total IKE SAs successfully established. Increments after successful EAP-AKA authentication and Child SA creation.

Metric: `epdg_ikev2_session_failed_count` **Type**: Counter **Description**: Total IKE SA establishment failures **Labels**:

- `reason` - Failure reason (e.g., `auth_failed`, `timeout`, `invalid_proposal`)

EAP Authentication Metrics

Metrics tracking EAP-AKA authentication flows per RFC 4187. OmniEPDG also supports EAP-AKA' per RFC 5448, with the authentication type automatically selected based on the UE's NAI identity prefix.



Metric: `epdg_eap_identity_count` **Type**: Counter **Description**: Total EAP-Identity requests received from UEs

---

Metric: `epdg_eap_aka_challenge_count` **Type**: Counter **Description**: Total EAP-AKA challenges sent to UEs

---

Metric: `epdg_eap_aka_success_count` **Type**: Counter **Description**: Total successful EAP-AKA authentications

---

Metric: `epdg_eap_aka_failure_count` **Type**: Counter **Description**: Total failed EAP-AKA authentications **Labels**:

- `reason` - Failure reason (e.g., `res_mismatch`, `invalid_identity`, `authentication_rejected`)

Metric: `epdg_eap_aka_sync_failure_count` **Type**: Counter **Description**: Total EAP-AKA sequence number (SQN) synchronization failures. Indicates USIM/HSS sequence number mismatch requiring resynchronization.

Authentication Security Metrics

Metrics for the authentication security layer. See the Security Guide for configuration details.

Metric: `epdg_auth_verification_failed_count` **Type**: Counter **Description**: Total AUTH payload verification failures. Indicates potential man-in-the-middle attacks or implementation bugs.

Metric: `epdg_auth_rate_limited_count` **Type**: Counter **Description**: Total authentication attempts blocked by rate limiting **Labels**:

- `type` - Blocking reason: `ip` (per-IP threshold exceeded) or `imsi` (per-IMSI threshold exceeded)

Example queries:

```
# Rate limited attempts per minute
rate(epdg_auth_rate_limited_count[1m])

# Rate limited by type
sum by (type) (rate(epdg_auth_rate_limited_count[5m]))
```

Metric: `epdg_auth_geoip_blocked_count` **Type**: Counter **Description**: Total authentication attempts blocked by GeoIP country filtering **Labels**:

- `country` - ISO 3166-1 alpha-2 country code (e.g., `CN`, `RU`), or `UNKNOWN` for IPs that could not be geolocated

Example queries:

```
# GeoIP blocks per minute
rate(epdg_auth_geoip_blocked_count[1m])

# Top blocked countries
topk(10, sum by (country) (epdg_auth_geoip_blocked_count))
```

Metric: `epdg_esp_replay_detected_count` **Type**: Counter **Description**: Total ESP packets rejected due to replay detection (per RFC 4303). Indicates potential replay attacks or network issues causing packet reordering.

Diameter SWx Metrics

Metrics for the SWx interface between ePDG and HSS per 3GPP TS 29.273.

Metric: `epdg_diameter_swx_mar_count` **Type**: Counter **Description**: Total Multimedia-Auth-Requests sent to HSS for authentication vector retrieval **Labels**:

- `result` - Request outcome: `success` or `failure`

Metric: `epdg_diameter_swx_sar_count` **Type**: Counter **Description**: Total Server-Assignment-Requests sent to HSS for registration/deregistration **Labels**:

- `result` - Request outcome: `success` or `failure`

Metric: `epdg_diameter_swx_mar_latency_ms` **Type**: Histogram **Description**: MAR response time in milliseconds **Buckets**: 50, 100, 250, 500, 1000, 2500 ms

---

Metric: `epdg_diameter_swx_sar_latency_ms` **Type**: Histogram **Description**: SAR response time in milliseconds **Buckets**: 50, 100, 250, 500, 1000, 2500 ms

---

Metric: `epdg_diameter_swx_pending_count` **Type**: Gauge **Description**: Current number of pending SWx requests awaiting response. High values indicate HSS congestion or connectivity issues.

Diameter S6b Metrics

Metrics for the S6b interface between AAA server and PGW per 3GPP TS 29.273. Only applicable in GTP mode.

Metric: `epdg_diameter_s6b_aar_count` **Type**: Counter **Description**: Total AA-Requests handled for session authorization **Labels**:

- `result` - Request outcome: `success` or `failure`

Metric: `epdg_diameter_s6b_str_count` **Type**: Counter **Description**: Total Session-Termination-Requests processed

---

Metric: `epdg_diameter_s6b_active_sessions_count` **Type**: Gauge **Description**: Current number of active S6b sessions

Session Lifecycle Metrics

Metrics tracking PDN session creation and termination.

Metric: `epdg_session_created_count` **Type**: Counter **Description**: Total sessions created **Labels**:

- `vpn_mode` - VPN mode: `simple` or `gtp`

Metric: `epdg_session_terminated_count` **Type**: Counter **Description**: Total sessions terminated **Labels**:

- `reason` - Termination reason: `user_request`, `timeout`, `error`, `admin`

Metric: `epdg_sessions_active_count` **Type**: Gauge **Description**: Current number of active sessions. Polled every 5 seconds.

Metric: `epdg_sessions_by_state_count` **Type**: Gauge **Description**: Sessions grouped by FSM state **Labels**:

- `state` - Session state (e.g., `init`, `eap_identity`, `eap_aka_challenge`, `authenticated`, `established`)

Metric: `epdg_auth_duration_ms` **Type**: Histogram **Description**: Full authentication flow duration from IKE_SA_INIT to session established **Buckets**: 100, 250, 500, 1000, 2500, 5000, 10000 ms

---

Metric: `epdg_session_duration_seconds` **Type**: Histogram **Description**: Session lifetime from establishment to termination **Buckets**: 60, 300, 900, 1800, 3600, 7200, 14400 seconds (1 min to 4 hours)

ESP Data Plane Metrics

Metrics for ESP packet processing per RFC 4303.

Metric: `epdg_esp_packets_in_count` **Type**: Counter **Description**: Total ESP packets successfully decrypted (UE → network direction)

Metric: `epdg_esp_packets_out_count` **Type**: Counter **Description**: Total ESP packets encrypted (network → UE direction)

Metric: `epdg_esp_bytes_in_total` **Type**: Gauge **Description**: Total bytes decrypted from ESP packets

---

Metric: `epdg_esp_bytes_out_total` **Type**: Gauge **Description**: Total bytes encrypted into ESP packets

IP Pool Metrics

Metrics for IP address pool management in Simple VPN mode.

Metric: `epdg_ip_allocated_count` **Type**: Counter **Description**: Total IP addresses allocated **Labels**:

- `type` - Address type: `ipv4` or `ipv6`

Metric: `epdg_ip_released_count` **Type**: Counter **Description**: Total IP addresses released **Labels**:

- `type` - Address type: `ipv4` or `ipv6`

Metric: `epdg_ip_pool_allocated_count` **Type**: Gauge **Description**: Current number of allocated IP addresses

---

Metric: `epdg_ip_pool_available_count` **Type**: Gauge **Description**: Current number of available IP addresses in the pool

---

Metric: `epdg_ip_pool_utilization_ratio` **Type**: Gauge **Description**: IP pool utilization as a ratio from 0.0 to 1.0. Values approaching 1.0 indicate pool exhaustion risk.

VM Metrics

Erlang/BEAM virtual machine metrics for system health monitoring.

Metric: `vm_memory_total` **Type**: Gauge **Unit**: Bytes **Description**: Total memory allocated by the VM

Metric: `vm_memory_processes` **Type**: Gauge **Unit**: Bytes **Description**: Memory used by Erlang processes

---

Metric: `vm_memory_binary` **Type**: Gauge **Unit**: Bytes **Description**: Memory used for binaries (including packet buffers)

Metric: `vm_memory_ets` **Type**: Gauge **Unit**: Bytes **Description**: Memory used by ETS tables (session state, registries)

Metric: `vm_system_info_process_count` **Type**: Gauge **Description**: Number of running Erlang processes

---

Metric: `vm_system_info_port_count` **Type**: Gauge **Description**: Number of open ports (sockets, file handles)

Metric: `vm_statistics_run_queue` **Type**: Gauge **Description**: Total length of scheduler run queues. High values indicate CPU saturation.

Prometheus Integration

Scrape Configuration

Add OmniEPDG to your Prometheus `prometheus.yml`:

```yaml
scrape_configs:
  - job_name: 'omniepdg'
    scrape_interval: 15s
    static_configs:
      - targets: ['epdg-host:9568']
        labels:
          instance: 'epdg-01'
          environment: 'production'
```

Service Discovery

For Kubernetes deployments, use service discovery:

```
scrape_configs:
  - job_name: 'omniepdg'
    kubernetes_sd_configs:
      - role: pod
    relabel_configs:
      - source_labels: [__meta_kubernetes_pod_label_app]
        action: keep
        regex: omniepdg
      - source_labels:
[__meta_kubernetes_pod_annotation_prometheus_io_port]
        action: replace
        target_label: __address__
        regex: (.+)
        replacement: ${1}:9568
```

Example Queries

Authentication Success Rate

```
# Success rate over 5 minutes
sum(rate(epdg_eap_aka_success_count[5m]))
/
(sum(rate(epdg_eap_aka_success_count[5m])) +
sum(rate(epdg_eap_aka_failure_count[5m])))
```

Session Establishment Rate

```
# Sessions established per second
rate(epdg_ikev2_session_established_count[5m])
```

Authentication Latency (p95)

```
histogram_quantile(0.95,
sum(rate(epdg_auth_duration_ms_bucket[5m])) by (le))
```

HSS Latency (p99)

```
histogram_quantile(0.99,
sum(rate(epdg_diameter_swx_mar_latency_ms_bucket[5m])) by (le))
```

Active Sessions

```
epdg_sessions_active_count
```

IP Pool Utilization

```
epdg_ip_pool_utilization_ratio * 100
```

ESP Throughput

```
# Bytes per second (inbound)
rate(epdg_esp_bytes_in_total[5m])

# Packets per second (both directions)
rate(epdg_esp_packets_in_count[5m]) +
rate(epdg_esp_packets_out_count[5m])
```

Failure Breakdown by Reason

```
# EAP failures by reason
sum by (reason) (rate(epdg_eap_aka_failure_count[5m]))

# Session terminations by reason
sum by (reason) (rate(epdg_session_terminated_count[5m]))
```

Alerting Rules

Example Prometheus alerting rules for OmniEPDG:

```yaml
groups:
  - name: omniepdg
    rules:
      # High authentication failure rate
      - alert: OmniEPDGHighAuthFailureRate
        expr: |
          sum(rate(epdg_eap_aka_failure_count[5m]))
          /
          (sum(rate(epdg_eap_aka_success_count[5m])) +
sum(rate(epdg_eap_aka_failure_count[5m])))
          > 0.1
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "High EAP-AKA authentication failure rate"
          description: "Authentication failure rate is {{ $value |
humanizePercentage }} over the last 5 minutes"

      # IP pool near exhaustion
      - alert: OmniEPDGIPPoolLow
        expr: epdg_ip_pool_utilization_ratio > 0.9
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "IP pool utilization above 90%"
          description: "IP pool is {{ $value | humanizePercentage
}} utilized"

      # IP pool exhausted
      - alert: OmniEPDGIPPoolExhausted
        expr: epdg_ip_pool_available_count == 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "IP pool exhausted"
          description: "No IP addresses available for new
sessions"

      # HSS latency high
      - alert: OmniEPDGHSSLatencyHigh
```

```
        expr: |
          histogram_quantile(0.95,
sum(rate(epdg_diameter_swx_mar_latency_ms_bucket[5m])) by (le))
          > 1000
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "High HSS (SWx) latency"
          description: "95th percentile MAR latency is {{ $value
}}ms"

      # Pending SWx requests backlog
      - alert: OmniEPDGSWxBacklog
        expr: epdg_diameter_swx_pending_count > 100
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: "SWx request backlog building"
          description: "{{ $value }} pending SWx requests"

      # VM memory high
      - alert: OmniEPDGMemoryHigh
        expr: vm_memory_total > 2147483648
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "OmniEPDG memory usage high"
          description: "VM memory usage is {{ $value |
humanize1024 }}"

      # Scheduler overload
      - alert: OmniEPDGSchedulerOverload
        expr: vm_statistics_run_queue > 10
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Erlang scheduler run queue high"
          description: "Run queue length is {{ $value }},
indicating CPU saturation"
```

```
      # No sessions (potential service issue)
      - alert: OmniEPDGNoSessions
        expr: epdg_sessions_active_count == 0 and
epdg_ikev2_session_initiated_count > 0
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "No active sessions despite connection
attempts"
          description: "Sessions are being initiated but none are
active"

      # High rate limiting activity (potential attack)
      - alert: OmniEPDGHighRateLimiting
        expr: rate(epdg_auth_rate_limited_count[5m]) > 10
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "High rate of blocked authentication attempts"
          description: "{{ $value | printf \"%.1f\" }} auth
attempts blocked per second"

      # GeoIP blocking spike (potential attack from specific
region)
      - alert: OmniEPDGGeoIPBlockingSpike
        expr: |
          rate(epdg_auth_geoip_blocked_count[5m]) > 5
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Elevated GeoIP blocking activity"
          description: "{{ $value | printf \"%.1f\" }} connections
blocked per second by GeoIP"

      # ESP replay attacks detected
      - alert: OmniEPDGReplayAttack
        expr: rate(epdg_esp_replay_detected_count[5m]) > 0
        for: 2m
        labels:
          severity: warning
        annotations:
```

```
          summary: "ESP replay attacks detected"
          description: "{{ $value | printf \"%.1f\" }} replay
attempts per second"

      # AUTH verification failures (potential MITM)
      - alert: OmniEPDGAuthVerificationFailures
        expr: rate(epdg_auth_verification_failed_count[5m]) > 0
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: "AUTH payload verification failures detected"
          description: "Potential man-in-the-middle attack or
implementation bug"
```

Network Requirements

This document covers the firewall ports and DNS entries required to deploy OmniEPDG for WiFi calling.

Firewall Ports

Internet-Facing Ports (UE ↔ ePDG)

These ports must be open to the internet for mobile devices to establish WiFi calling connections.

Port	Protocol	Direction	Purpose
500	UDP	Inbound	IKEv2 initial exchange (IKE_SA_INIT, IKE_AUTH)
4500	UDP	Inbound	IKEv2 NAT-Traversal and ESP-in-UDP encapsulation

Port 500/UDP handles the initial IKEv2 negotiation. If NAT is detected between the UE and ePDG (which is almost always the case for WiFi calling), the connection automatically migrates to port 4500.

Port 4500/UDP carries both IKEv2 signaling and ESP-encrypted user data when NAT-T is active. This is the primary data path for all WiFi calling traffic.

Internal Network Ports (ePDG ↔ Core)

These ports are used for communication between OmniEPDG and the mobile core network. They should be accessible from the ePDG but not exposed to the internet.

Port	Protocol	Direction	Purpose	Peer
3868	TCP	Bidirectional	Diameter SWx (authentication)	HSS / DRA
3868	TCP	Bidirectional	Diameter S6b (session authorization)	PGW / AAA
2123	UDP	Bidirectional	GTPv2-C control plane (S2b)	PGW
2152	UDP	Bidirectional	GTP-U user plane (S2b-U)	PGW

Management Ports

These ports are for operational monitoring and should be restricted to management networks.

Port	Protocol	Purpose
4000	TCP	Control panel web interface (HTTP)
443	TCP	Control panel web interface (HTTPS, production)
9568	TCP	Prometheus metrics endpoint

DNS Requirements

ePDG Discovery DNS Record

Mobile devices discover the ePDG using a standardized DNS naming convention defined in 3GPP TS 23.003. The FQDN format is:

```
epdg.epc.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org
```

Where:

- `<MCC>` is the 3-digit Mobile Country Code (e.g., `505` for Australia)
- `<MNC>` is the 2 or 3-digit Mobile Network Code, zero-padded to 3 digits (e.g., `001`)

The DNS record should be an **A record** (or AAAA for IPv6) pointing to the public IP address of OmniEPDG.

```
epdg.epc.mnc999.mcc999.pub.3gppnetwork.org.  IN  A  203.0.113.10
```

Certificate Common Name

The ePDG's TLS certificate must have a Subject Alternative Name (SAN) matching the ePDG FQDN that devices will connect to. This is validated by the UE during IKEv2 authentication.

Certificate requirements:

- SAN must include the ePDG discovery FQDN (e.g., `epdg.epc.mnc001.mcc001.pub.3gppnetwork.org`)
- Certificate must be signed by a trusted CA (devices validate the chain)
- RSA 2048-bit or ECDSA P-256 minimum

Diameter Realm DNS

For proper Diameter routing, the realm should resolve via DNS NAPTR/SRV records per RFC 6408. The realm format is:

```
epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

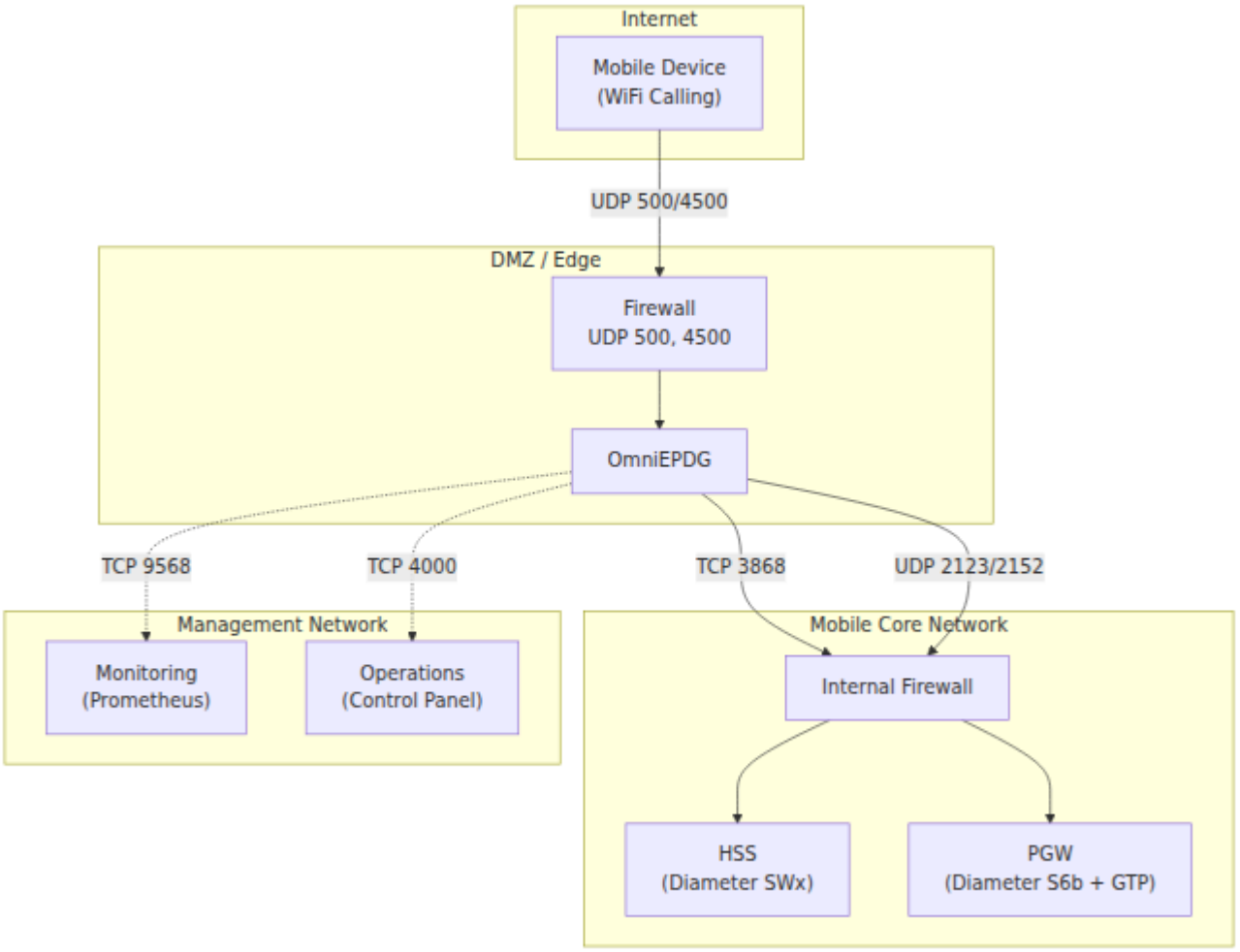
Example:

```
epc.mnc001.mcc001.3gppnetwork.org
```

This realm is used in:

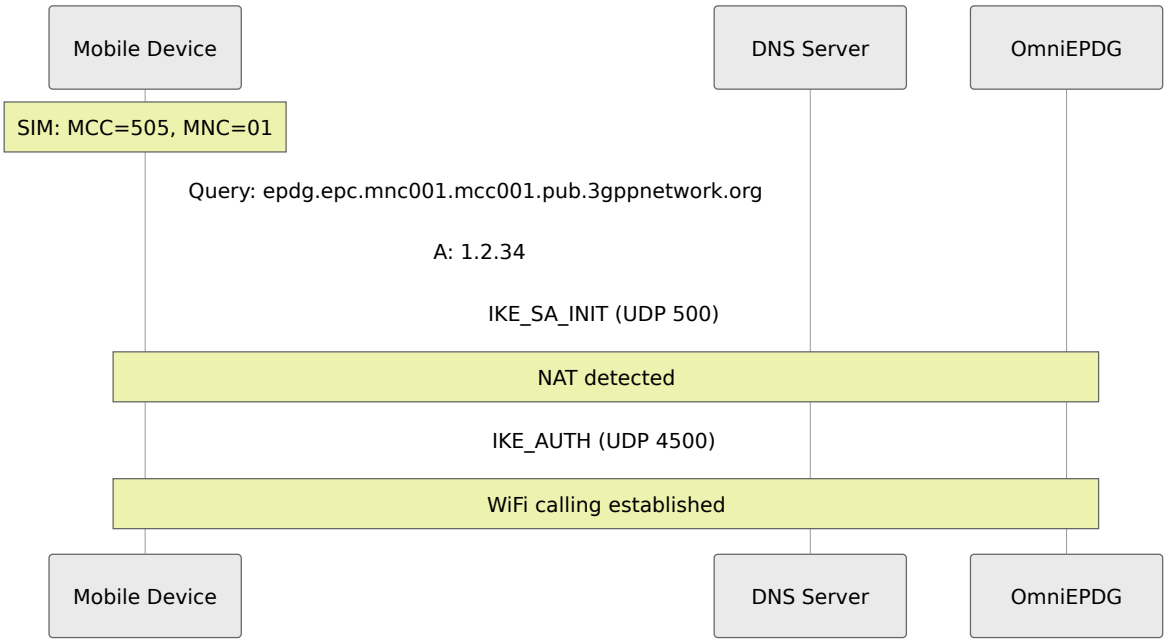
- NAI (Network Access Identifier) sent by the UE during authentication
- Diameter Origin-Realm and Destination-Realm AVPs
- Diameter routing decisions

Network Topology



DNS Lookup Flow

When a mobile device initiates WiFi calling, the following DNS resolution occurs:



Checklist

Internet-Facing Requirements

- [ ] UDP 500 open inbound to OmniEPDG
- [ ] UDP 4500 open inbound to OmniEPDG
- [ ] DNS A record: `epdg.epc.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org` → ePDG public IP
- [ ] TLS certificate with matching SAN installed

Core Network Requirements

- [ ] TCP 3868 open between OmniEPDG and HSS/DRA
- [ ] TCP 3868 open between OmniEPDG and PGW/AAA (GTP mode only)
- [ ] UDP 2123 open between OmniEPDG and PGW (GTP mode only)
- [ ] UDP 2152 open between OmniEPDG and PGW (GTP mode only)

Management Requirements

- [ ] TCP 4000/443 accessible from operations network
- [ ] TCP 9568 accessible from monitoring infrastructure

References

- 3GPP TS 23.003 - Numbering, addressing and identification (ePDG FQDN format)
- 3GPP TS 23.402 - Architecture enhancements for non-3GPP accesses
- RFC 7296 - IKEv2 Protocol
- RFC 3948 - UDP Encapsulation of IPsec ESP Packets (NAT-T)
- RFC 6733 - Diameter Base Protocol

OmniEPDG Operations Guide

OmniEPDG is a 3GPP-compliant ePDG (evolved Packet Data Gateway) that enables WiFi calling by bridging untrusted WiFi access to the mobile core network. It supports two operational modes: **GTP mode** for PGW tunneling and **Simple VPN mode** for local IP breakout.

Quick Links

Configuration & Deployment

- **Network Requirements** - Firewall ports and DNS entries required for deployment
- **Configuration Reference** - Complete parameter documentation for IKEv2, Diameter, VPN modes, and all runtime settings
- **Architecture Overview** - System architecture, call flows, state machines, and protocol references

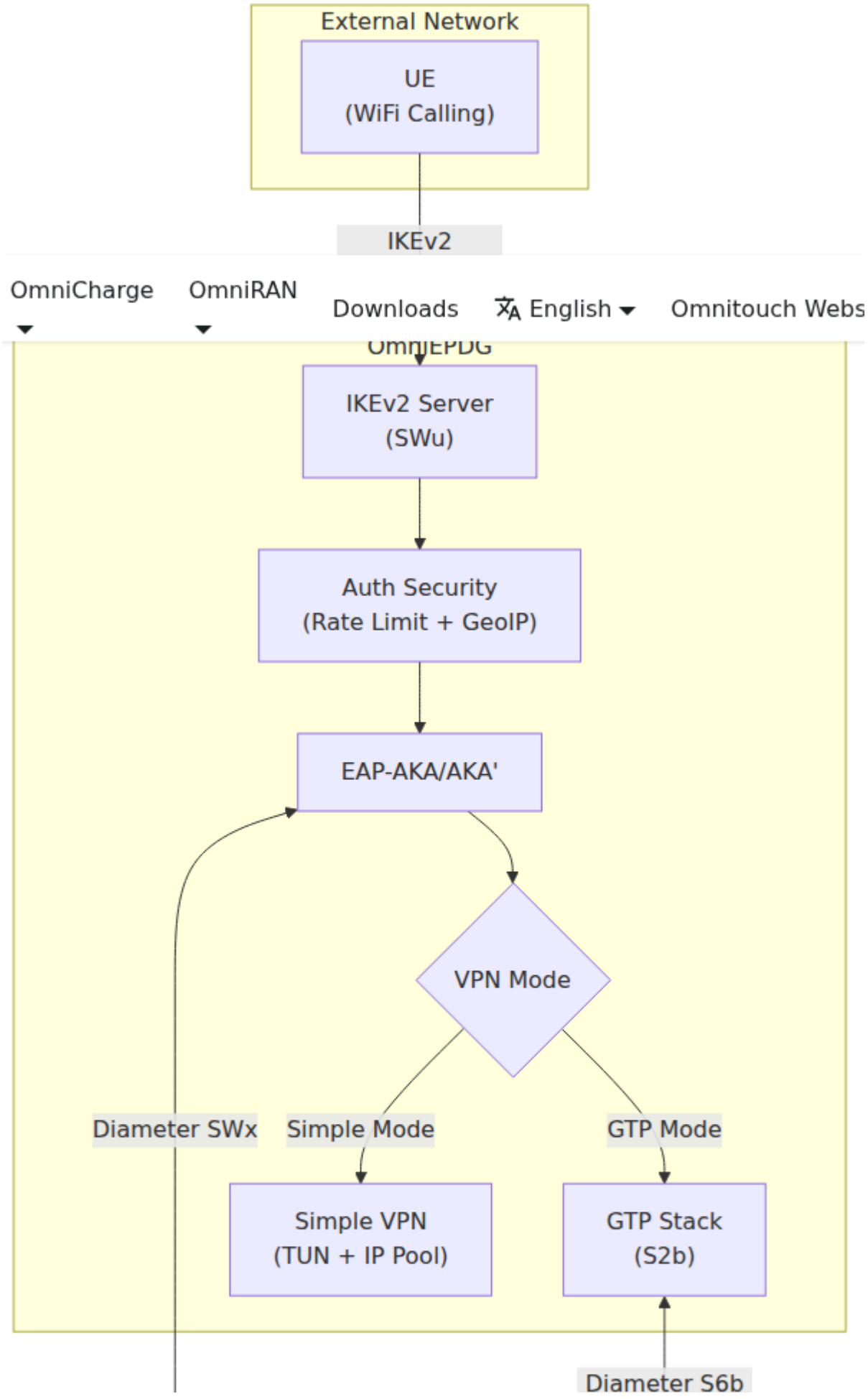
Operations & Monitoring

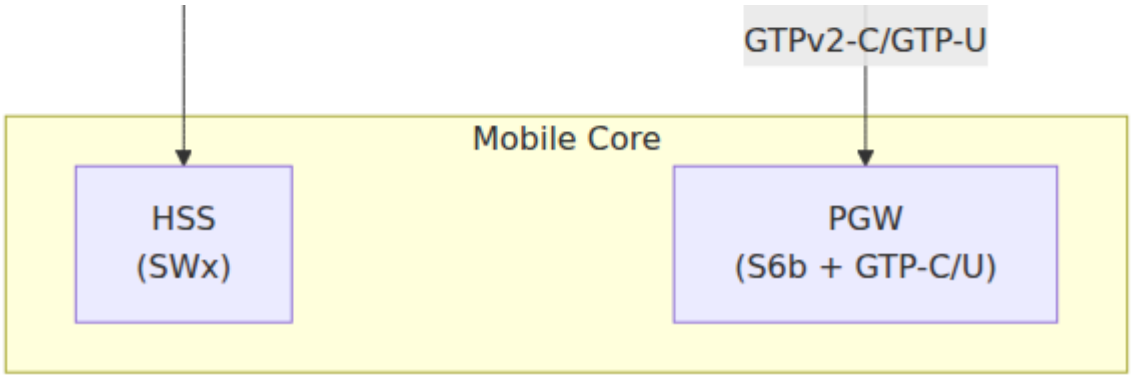
- **Control Panel** - Web-based monitoring interface for sessions, Diameter peers, and logs
- **Metrics Reference** - Prometheus metrics, example queries, and alerting rules
- **Troubleshooting** - Common issues, diagnostic procedures, and resolution steps

Security

- **Authentication Security** - Rate limiting and GeoIP country blocking

Architecture Overview


External Network
UE
(WiFi Calling)
IKEv2
OmniCharge
OmniRAN
Downloads
English
Omnitouch Webs
OmniEPDG
IKEv2 Server
(SWu)
Auth Security
(Rate Limit + GeoIP)
EAP-AKA/AKA'
VPN Mode
Diameter SWx
Simple Mode
GTP Mode
Simple VPN
(TUN + IP Pool)
GTP Stack
(S2b)
Diameter S6b



Operational Modes

OmniEPDG supports two operational modes selected via the `vpn_mode` configuration parameter.

GTP Mode

Full 3GPP tunneling through a PGW. Subscriber traffic is encapsulated in GTP-U and routed through the mobile core.

Use when:

- Integration with existing mobile core infrastructure
- Policy and charging through PCRF/PCEF
- Roaming and inter-operator handoff required

Components:

- Diameter S6b for PGW session authorization
- GTPv2-C for session management (Create/Delete/Modify Session)
- Linux kernel GTP-U module for user plane

Simple VPN Mode

Local IP breakout via TUN interface. Subscriber traffic is routed directly through the OmniEPDG host without PGW involvement.

Use when:

- Standalone deployment without PGW

- Testing and development
- Local breakout scenarios

Components:

- Local IP pool management (IPv4/IPv6)
- TUN interface (`omniepdg0`) with per-subscriber host routes
- Optional NAT/masquerade for internet access

Authentication Security

OmniEPDG includes built-in security features to protect against brute-force attacks and restrict access by geographic location. See the Authentication Security guide for details.

Rate Limiting

Protects against brute-force attacks by tracking failed authentication attempts:

- **Per-IP limiting** - Blocks IPs after 10 failures in 1 minute (5 minute block)
- **Per-IMSI limiting** - Blocks IMSIs after 5 failures in 1 minute (10 minute block)
- Sliding window algorithm with automatic expiration
- Successful authentication clears failure history

GeoIP Country Blocking

Optional geographic access control using MaxMind GeoLite2 database:

- **Whitelist mode** - Only allow connections from specified countries
- **Blacklist mode** - Block connections from specified countries
- Configurable handling of unknown/private IPs
- Fail-open or fail-closed behavior when database unavailable

Key Configuration

Minimal Configuration (Simple VPN Mode)

```
config :omniepdg,
  vpn_mode: :simple,
  simple_vpn: [
    pool_ipv4: "10.45.0.0/16",
    dns_servers_ipv4: ["8.8.8.8", "8.8.4.4"]
  ]

config :diameter_ex,
  diameter: %{
    host: "epdg",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    peers: [
      %{host: "hss", ip: "10.74.0.21", port: 3868, transport:
:tcp}
    ]
  }
```

Enable Authentication Security

```
config :omniepdg,
  # Rate limiting (enabled by default with these values)
  auth_rate_limit_per_ip: 10,
  auth_rate_limit_ip_block_ms: 300_000,
  auth_rate_limit_per_imsi: 5,
  auth_rate_limit_imsi_block_ms: 600_000,

  # GeoIP blocking (disabled by default)
  geoip_enabled: true,
  geoip_mode: :whitelist,
  geoip_countries: ["AU", "NZ"]
```

See the Configuration Reference for complete parameter documentation.

Monitoring

Control Panel

Access the web control panel at `http://<host>:4000/dashboard` for:

- Real-time session monitoring
- Diameter peer status
- Live log streaming
- System configuration

See the Control Panel guide for details.

Prometheus Metrics

Scrape metrics from `http://<host>:9568/metrics` for:

- Authentication success/failure rates
- Session lifecycle events
- Diameter signaling latency
- Security events (rate limiting, GeoIP blocks)
- IP pool utilization
- ESP data plane statistics

See the Metrics Reference for queries and alerting rules.

Troubleshooting

Common issues and resolution steps:

Issue	Quick Check	Guide Section
Authentication failures	Check SWx MAR/MAA in logs	Authentication Failures
Diameter connection issues	Check peer status in control panel	Diameter Connectivity
GTP tunnel failures	Check GTPv2-C cause codes	GTP Tunnel Failures
Simple VPN issues	Check TUN interface and routes	Simple VPN Failures
Rate limiting false positives	Adjust thresholds	Rate Limiting Issues
GeoIP blocking issues	Check database and country codes	GeoIP Issues

See the Troubleshooting guide for detailed diagnostic procedures.

Documentation Index

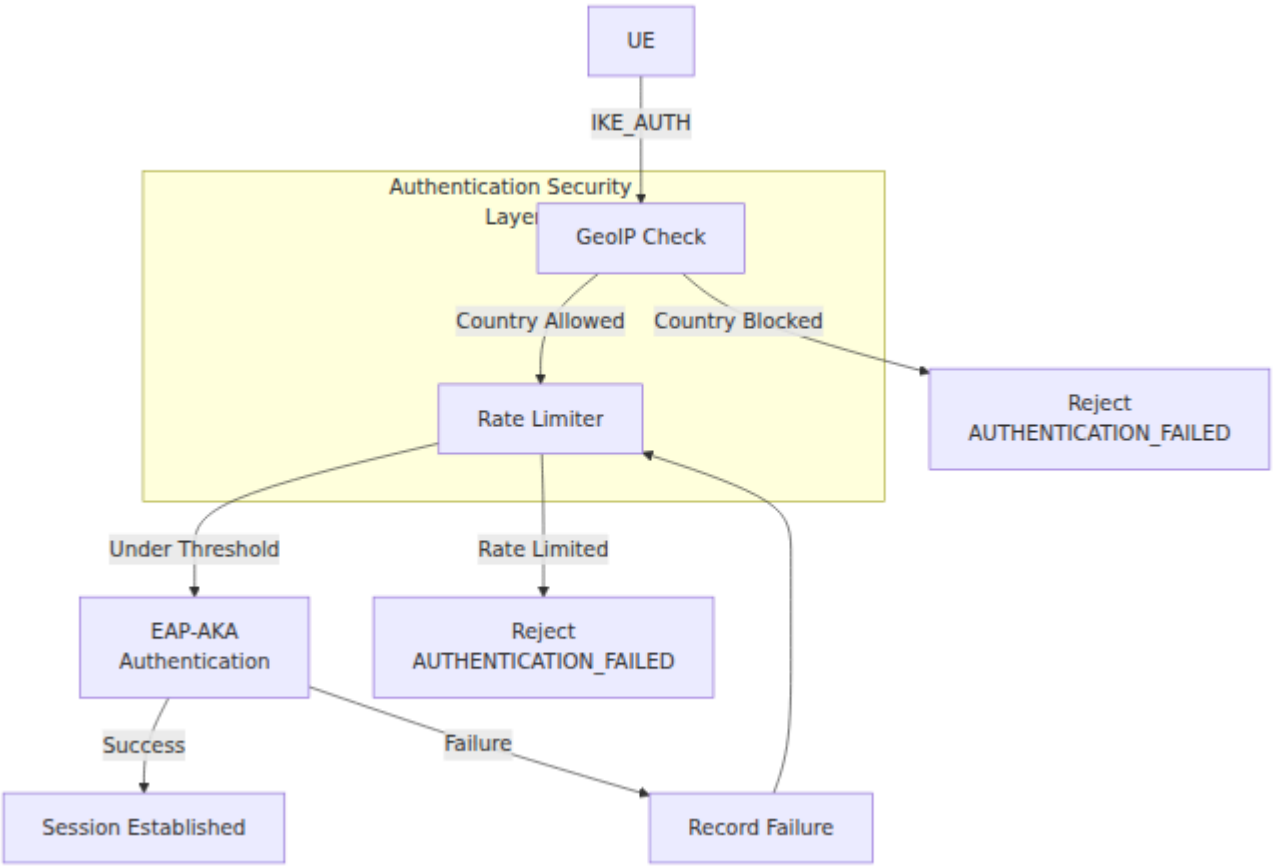
Document	Description
Architecture	System design, state machines, call flows, protocol references
Configuration	Complete configuration reference with examples
Control Panel	Web interface guide with screenshots
Metrics	Prometheus metrics, queries, and alerting
Network Requirements	Firewall ports and DNS entries for deployment
Security	Rate limiting and GeoIP country blocking
Troubleshooting	Common issues and diagnostic procedures

OmniEPDG Authentication Security

OmniEPDG implements multiple layers of authentication security to protect against brute-force attacks, credential stuffing, and unauthorized access from restricted regions.

Table of Contents

Overview



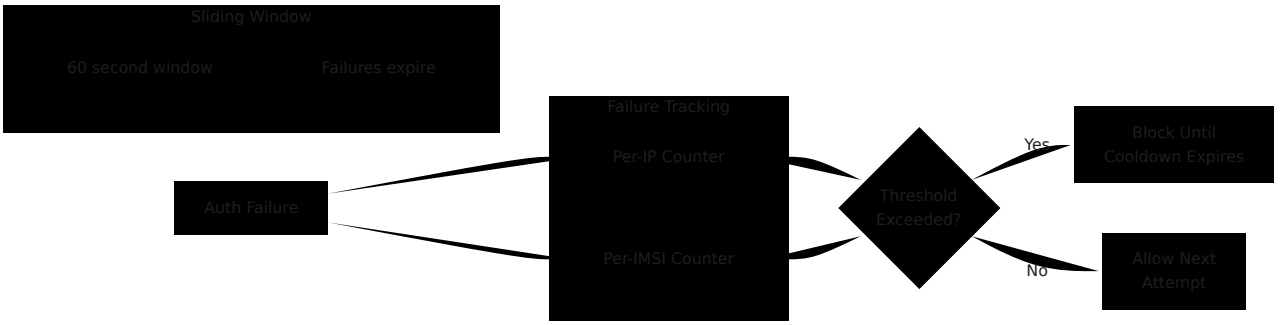
OmniEPDG performs security checks at the start of the IKE_AUTH exchange, before expensive cryptographic operations:

1. **GeoIP Check** (optional) - Verifies the source IP is from an allowed country
2. **Rate Limit Check** - Ensures the IP/IMSI hasn't exceeded failure thresholds
3. **EAP-AKA Authentication** - Standard 3GPP authentication proceeds if checks pass

Authentication Rate Limiting

Rate limiting protects against brute-force attacks by tracking failed authentication attempts per source IP and per IMSI. When thresholds are exceeded, further attempts are temporarily blocked.

How It Works



The rate limiter uses a **sliding window algorithm**:

- Each failed attempt is recorded with a timestamp
- Attempts older than the configured window are automatically expired
- When failures in the window exceed the threshold, the source is blocked
- Blocks expire after the configured cooldown period

Dual Tracking

Two independent limits are enforced simultaneously:

Tracking Type	Purpose	Default Threshold	Default Block
Per-IP	Catches port scanners and distributed attacks from single sources	10 failures / minute	5 minutes
Per-IMSI	Catches targeted attacks on specific subscribers	5 failures / minute	10 minutes

Both checks must pass for an authentication attempt to proceed. If either threshold is exceeded, the attempt is rejected.

Configuration

```
config :omniepdg,
  # Per-IP rate limiting
  auth_rate_limit_per_ip: 10,            # Max failures before
blocking
  auth_rate_limit_ip_window_ms: 60_000,  # Window size (1 minute)
  auth_rate_limit_ip_block_ms: 300_000,  # Block duration (5
minutes)

  # Per-IMSI rate limiting
  auth_rate_limit_per_imsi: 5,             # Max failures before
blocking
  auth_rate_limit_imsi_window_ms: 60_000,  # Window size (1
minute)
  auth_rate_limit_imsi_block_ms: 600_000   # Block duration (10
minutes)
```

Per-IP Parameters

Parameter	Type	Required	Default	Descri
`auth_rate_limit_per_ip`	Integer	No	`10`	Maximu failed authent attempt allowed a single address within th window period b blocking
`auth_rate_limit_ip_window_ms`	Integer	No	`60000`	Sliding window in millisec for coun IP failure Failures than thi not cou
`auth_rate_limit_ip_block_ms`	Integer	No	`300000`	Duratio millisec to block after exceedi threshol Default minutes

Per-IMSI Parameters

Parameter	Type	Required	Default	Desc
`auth_rate_limit_per_imsi`	Integer	No	`5`	Maxin failed authe attem allowe single withir windc perioc blocki Lower per-IP target attack
`auth_rate_limit_imsi_window_ms`	Integer	No	`60000`	Slidin windc in millise for co IMSI f
`auth_rate_limit_imsi_block_ms`	Integer	No	`600000`	Durat millise to blo IMSI a excee thresh Defau minut (longe IP blo prote

Parameter	Type	Required	Default	Desc
				specif subsc

Behavior on Success

When authentication succeeds, the rate limiter clears all failure history for that IP/IMSI pair. This allows legitimate users who experienced transient failures (e.g., network issues) to recover without being permanently penalized.

Example Configurations

High-Security Environment

Strict limits for environments with low tolerance for failed attempts:

```
config :omniepdg,
  auth_rate_limit_per_ip: 5,
  auth_rate_limit_ip_window_ms: 120_000,   # 2 minute window
  auth_rate_limit_ip_block_ms: 900_000,    # 15 minute block

  auth_rate_limit_per_imsi: 3,
  auth_rate_limit_imsi_window_ms: 120_000,
  auth_rate_limit_imsi_block_ms: 1_800_000  # 30 minute block
```

How it works: Only 5 failures per IP or 3 failures per IMSI are allowed within a 2-minute window. Blocks last 15-30 minutes respectively.

Use case: Enterprise deployments, high-value subscriber bases, or networks under active attack.

Relaxed Environment

More permissive limits for development or testing:

```
config :omniepdg,
  auth_rate_limit_per_ip: 50,
  auth_rate_limit_ip_window_ms: 60_000,
  auth_rate_limit_ip_block_ms: 60_000,     # 1 minute block

  auth_rate_limit_per_imsi: 20,
  auth_rate_limit_imsi_window_ms: 60_000,
  auth_rate_limit_imsi_block_ms: 120_000   # 2 minute block
```

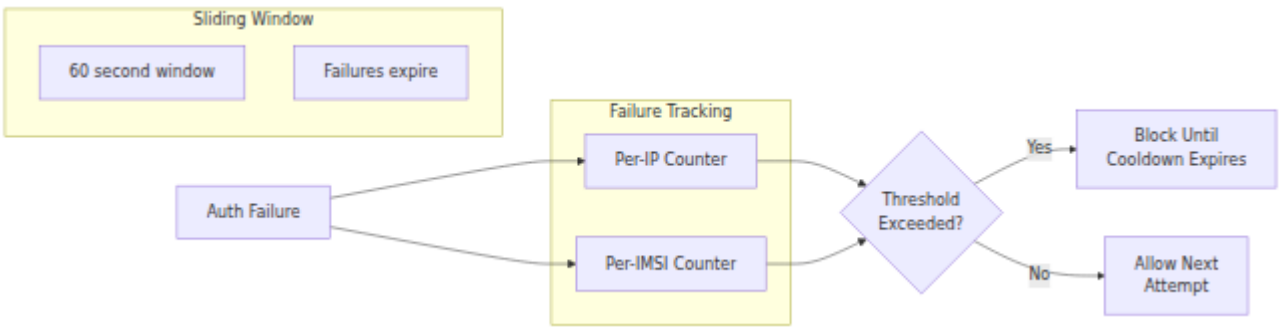
How it works: Higher thresholds and shorter blocks allow more testing flexibility.

Use case: Development environments, integration testing.

GeoIP Country Blocking

GeoIP blocking restricts WiFi calling access based on the geographic location of the connecting IP address. This is useful for operators who need to limit service to specific countries for regulatory or business reasons.

Overview



MaxMind GeoLite2 Database

GeoIP lookups use the MaxMind GeoLite2 Country database, a free IP geolocation database with weekly updates.

To enable GeoIP blocking:

1. Register for a free account at MaxMind GeoLite2 Signup
2. Download the `GeoLite2-Country.mmdb` database file
3. Place the file at the configured path (default: `/etc/omniepdg/GeoLite2-Country.mmdb`)
4. Enable GeoIP in configuration

Configuration

```
config :omniepdg,
  # Enable GeoIP blocking
  geoip_enabled: true,

  # Path to MaxMind database
  geoip_database_path: "/etc/omniepdg/GeoLite2-Country.mmdb",

  # Access control mode
  geoip_mode: :whitelist,

  # Country list (ISO 3166-1 alpha-2 codes)
  geoip_countries: ["AU", "NZ"],

  # Handle unknown IPs
  geoip_allow_unknown: false,

  # Behavior when database unavailable
  geoip_fail_open: true
```

Parameters

Parameter	Type	Required	Default
`geoip_enabled`	Boolean	No	`false`
`geoip_database_path`	String	No	`"/etc/omniepdg/GeoLite2-Country.mmdb"`
`geoip_mode`	Atom	No	`:whitelist`
`geoip_countries`	List	No	`[]`

Parameter	Type	Required	Default
`geoip_allow_unknown`	Boolean	No	See below
`geoip_fail_open`	Boolean	No	`true`

Access Control Modes

Whitelist Mode (Recommended for WiFi Calling)

Only allow connections from specified countries. All other countries are blocked.

```
config :omniepdg,
  geoip_enabled: true,
  geoip_mode: :whitelist,
  geoip_countries: ["AU", "NZ", "FJ"]  # Australia, New Zealand,
Fiji
```

How it works: Only UEs connecting from Australian, New Zealand, or Fijian IP addresses can authenticate. All other countries are rejected.

Use case: Operators who want to restrict WiFi calling to their licensed service areas.

Blacklist Mode

Block connections from specified countries. All other countries are allowed.

```
config :omniepdg,
  geoip_enabled: true,
  geoip_mode: :blacklist,
  geoip_countries: ["CN", "RU", "KP", "IR"]  # China, Russia,
North Korea, Iran
```

How it works: UEs connecting from the listed countries are rejected. All other countries can authenticate.

Use case: Blocking high-risk regions while allowing global roaming.

Handling Unknown Countries

Some IP addresses cannot be geolocated:

- Private IP ranges (10.x.x.x, 192.168.x.x, etc.)

- Newly allocated IP blocks not yet in the database
- Tor exit nodes and some VPNs

The `geoip_allow_unknown` parameter controls behavior:

Mode	`geoip_allow_unknown` Default	Behavior
Whitelist	`false`	Unknown = not in whitelist = blocked
Blacklist	`true`	Unknown = not in blacklist = allowed

To override the default:

```
config :omniepdg,
  geoip_mode: :whitelist,
  geoip_allow_unknown: true  # Allow unknown IPs even in whitelist
mode
```

Database Updates

MaxMind updates the GeoLite2 database weekly. To update:

1. Download the new `GeoLite2-Country.mmdb` file
2. Replace the existing file at the configured path
3. The database is automatically reloaded on the next lookup (no restart required)

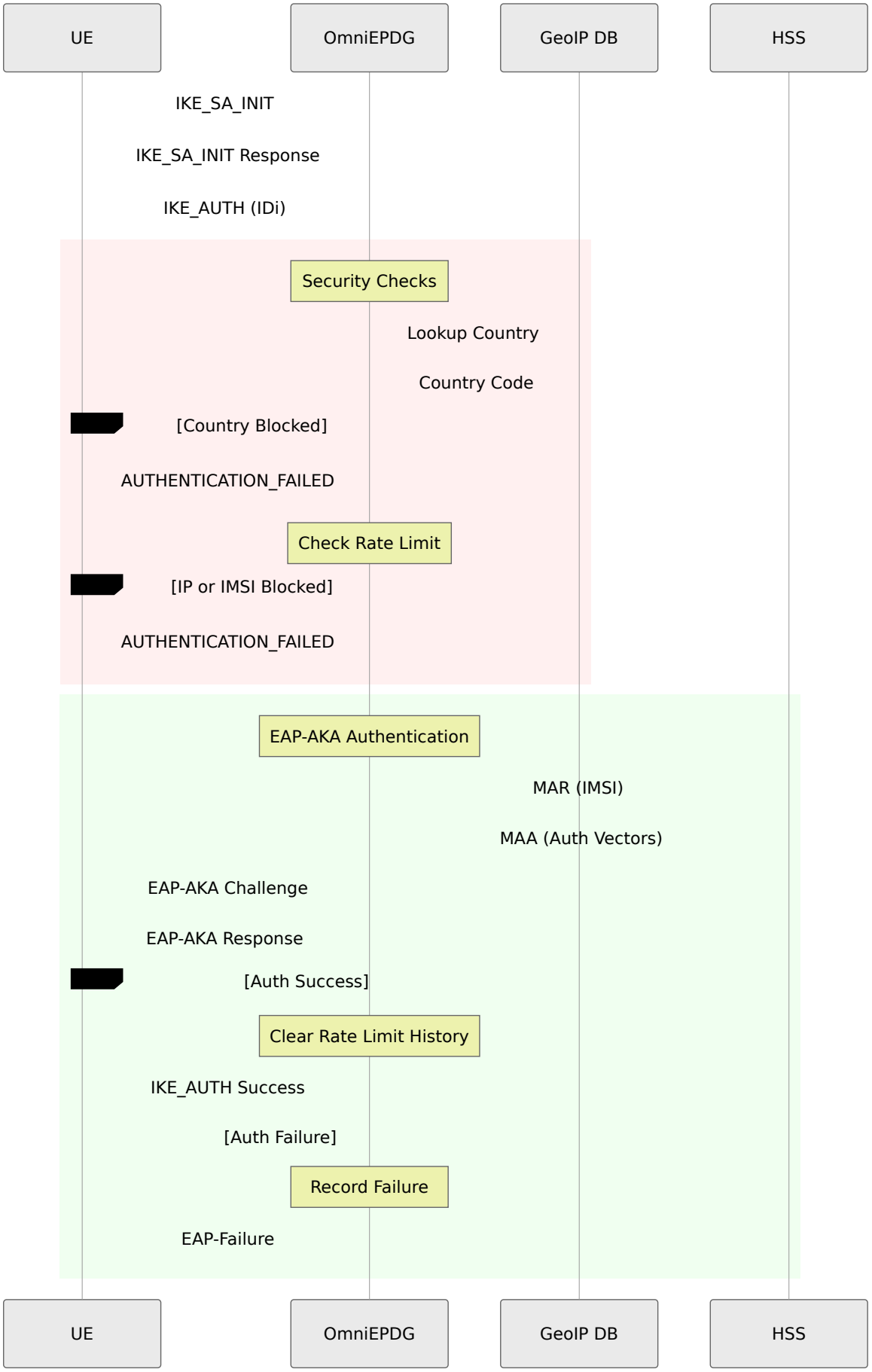
Common Country Codes

Code	Country	Code	Country
AU	Australia	US	United States
NZ	New Zealand	GB	United Kingdom
CA	Canada	DE	Germany
FR	France	JP	Japan
SG	Singapore	HK	Hong Kong
IN	India	CN	China

Full list: ISO 3166-1 alpha-2

Security Flow

The complete authentication security flow:


UE
OmniEPDG
GeoIP DB
HSS
IKE_SA_INIT
IKE_SA_INIT Response
IKE_AUTH (IDi)
Security Checks
Lookup Country
Country Code
[Country Blocked]
AUTHENTICATION_FAILED
Check Rate Limit
[IP or IMSI Blocked]
AUTHENTICATION_FAILED
EAP-AKA Authentication
MAR (IMSI)
MAA (Auth Vectors)
EAP-AKA Challenge
EAP-AKA Response
[Auth Success]
Clear Rate Limit History
IKE_AUTH Success
[Auth Failure]
Record Failure
EAP-Failure
UE
OmniEPDG
GeoIP DB
HSS

Metrics

Rate Limiting Metrics

Metric: `epdg_auth_rate_limited_count` **Type**: Counter **Description**: Number of authentication attempts blocked by rate limiting **Labels**:

- `type` - Blocking reason: `ip` (IP threshold exceeded) or `imsi` (IMSI threshold exceeded)

Example queries:

```
# Rate limited attempts per minute
rate(epdg_auth_rate_limited_count[1m])

# Rate limited by type
sum by (type) (rate(epdg_auth_rate_limited_count[5m]))

# Alert: High rate limiting activity
rate(epdg_auth_rate_limited_count[5m]) > 10
```

GeoIP Metrics

Metric: `epdg_auth_geoip_blocked_count` **Type**: Counter **Description**: Number of authentication attempts blocked by GeoIP **Labels**:

- `country` - ISO 3166-1 alpha-2 country code, or `UNKNOWN` for unresolvable IPs

Example queries:

```
# GeoIP blocks per minute
rate(epdg_auth_geoip_blocked_count[1m])

# Top blocked countries
topk(10, sum by (country) (epdg_auth_geoip_blocked_count))

# Alert: Unusual country attempting access
increase(epdg_auth_geoip_blocked_count{country="XX"}[1h]) > 100
```

Troubleshooting

Rate Limiting Issues

Legitimate Users Being Blocked

Symptoms: Users report being unable to connect after failed attempts

Possible causes:

- User entered wrong credentials multiple times
- Network issues caused authentication timeouts counted as failures
- Thresholds set too low for the environment

Resolution:

1. Check metrics for the affected IP/IMSI
2. Consider increasing thresholds if false positives are common
3. After fixing root cause, the block will expire automatically

High Rate of Blocked Attempts

Symptoms: `epdg_auth_rate_limited_count` increasing rapidly

Possible causes:

- Brute-force attack in progress
- Misconfigured UE repeatedly failing authentication

- Credential stuffing attack

Resolution:

1. Review source IPs in logs for patterns
2. Consider implementing IP-level firewall rules for persistent attackers
3. Verify HSS connectivity if legitimate users are affected

GeoIP Issues

All Connections Being Blocked

Symptoms: No UEs can connect after enabling GeoIP

Possible causes:

- Database file not found or corrupt
- Wrong country codes in configuration
- `geoip_allow_unknown: false` blocking private IPs in lab environment

Resolution:

1. Verify database file exists at configured path
2. Check country codes are correct (uppercase, 2 letters)
3. For lab/development, set `geoip_allow_unknown: true`
4. Check logs for GeoIP-related warnings

GeoIP Database Not Loading

Symptoms: Warning in logs: "GeoIP database not found"

Possible causes:

- File path incorrect
- File permissions prevent reading
- File is not valid MMDB format

Resolution:

1. Verify file exists: `ls -la /etc/omniepdg/GeoLite2-Country.mmdb`
2. Check permissions: `chmod 644 /etc/omniepdg/GeoLite2-Country.mmdb`
3. Verify file integrity by downloading a fresh copy from MaxMind

Unexpected Country Blocks

Symptoms: Users from allowed countries being blocked

Possible causes:

- VPN/proxy making IP appear from different country
- Outdated GeoIP database
- Corporate network egress in unexpected location

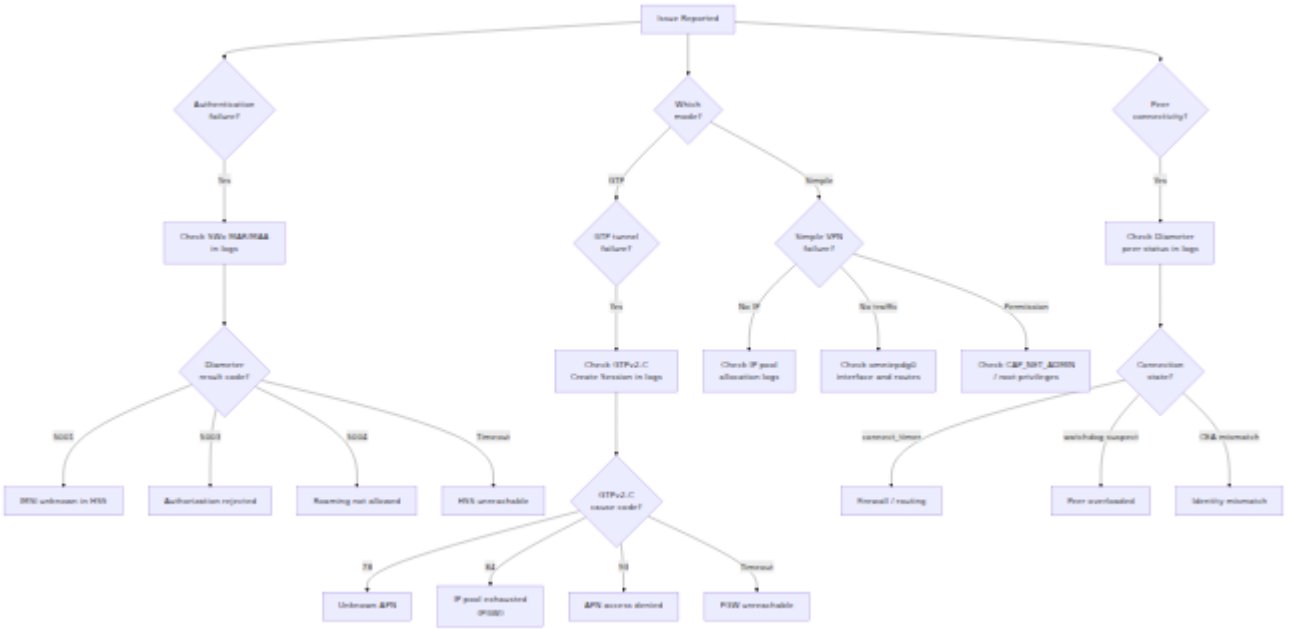
Resolution:

1. Update GeoIP database to latest version
2. Check user's actual egress IP vs expected country
3. Consider adding additional countries if users roam through corporate networks

OmniEPDG Troubleshooting

This guide covers common operational issues, diagnostic procedures, and resolution steps for OmniEPDG.

Diagnostic Overview



Log Files

OmniEPDG writes logs to the `log/` directory relative to the application working directory. See the Configuration Reference for log configuration details.

File	Purpose	When to Check
`log/console.log`	All application messages at debug level	First point of investigation for any issue
`log/error.log`	Errors only	Quick scan for active problems
`log/crash.log`	OTP process crashes	When processes are restarting unexpectedly
`log/erlang.log`	Erlang kernel logger	Low-level Erlang/OTP issues

Key Log Patterns

Diameter peer connection events:

- `peer_up` - Diameter peer connected and capabilities exchanged
- `peer_down` - Diameter peer disconnected

UE FSM state transitions:

- `ue_fsm state_<name> event=<event>` - FSM processing an event in a given state
- `ue_fsm init(&lt;IMSI>)` - New FSM instance created for subscriber
- `terminating epdg_ue_fsm with reason <reason>` - FSM shutting down

Timeout events:

- `Timeout swm_der_timeout` - SWm DER response timed out
- `Timeout create_session_timeout` - GTPv2-C Create Session response timed out
- `Timeout s2b_delete_session_timeout` - GTPv2-C Delete Session response timed out
- `Timeout cancel_location_timeout` - Cancel Location response timed out

Diameter Connectivity Issues

HSS Connection Failure (SWx)

Symptoms: No subscribers can authenticate. Logs show repeated connection attempts to the HSS.

Possible causes:

- Firewall blocking SCTP port 3868 between OmniEPDG and HSS
- Incorrect `dia_swx_remote_ip` or `dia_swx_remote_port` in configuration
- HSS not running or not accepting Diameter connections
- SCTP not enabled on the network path (some firewalls block SCTP by default)
- Origin-Host or Origin-Realm mismatch causing CEA rejection

Resolution:

1. Verify network connectivity to the HSS IP and port
2. Confirm `dia_swx_remote_ip` and `dia_swx_remote_port` match the HSS configuration
3. Check that SCTP traffic is permitted through all firewalls. If SCTP is blocked, set `dia_swx_proto` to `tcp` as a fallback
4. Verify `dia_swx_origin_host` is a resolvable FQDN and matches what the HSS expects
5. Check HSS logs for Diameter CER/CEA negotiation failures

PGW Connection Failure (S6b)

Symptoms: Authentication succeeds but GTP tunnel creation fails or S6b AAR never arrives from PGW. Logs show no S6b peer_up event.

Possible causes:

- PGW not configured to connect to OmniEPDG's S6b listener
- Firewall blocking SCTP port 3868 on OmniEPDG's S6b bind address

- `dia_s6b_local_ip` not reachable from the PGW
- Origin-Host or Origin-Realm mismatch

Resolution:

1. Confirm the PGW is configured to connect to OmniEPDG at `dia_s6b_local_ip`:`dia_s6b_local_port`
2. Verify the S6b bind address is reachable from the PGW network
3. Check firewall rules allow inbound SCTP on port 3868 at the S6b address
4. Verify `dia_s6b_origin_host` and `dia_s6b_origin_realm` match what the PGW expects

Diameter Watchdog Failures

Symptoms: Established Diameter connections intermittently drop. Logs show watchdog transitions to SUSPECT or DOWN state.

Possible causes:

- Network path instability or packet loss
- Peer overloaded and not responding to DWR within `dia_swx_watchdog_timer`
- Aggressive watchdog configuration (too few retries before declaring suspect)

Resolution:

1. Check network path quality (packet loss, latency) between OmniEPDG and the peer
2. If packet loss is expected, increase `dia_swx_watchdog_config` / `dia_s6b_watchdog_config` thresholds (e.g., `[{okay, 5}, {suspect, 3}]`)
3. Check peer system health (CPU, memory, connection count)

Authentication Failures

IMSI Unknown (Diameter 5001)

Symptoms: Specific subscribers fail EAP-AKA authentication. Logs show SWx MAA with result code 5001 (DIAMETER_ERROR_USER_UNKNOWN).

Possible causes:

- Subscriber not provisioned in the HSS
- IMSI mismatch between UE SIM and HSS database
- NAI format incorrect, causing IMSI extraction to fail

Resolution:

1. Verify the subscriber IMSI exists in the HSS database
2. Check the NAI format in logs matches the expected pattern: `0<IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org`
3. Confirm the SIM card IMSI matches the HSS provisioned value

Authorization Rejected (Diameter 5003)

Symptoms: Subscriber authenticates but is rejected during server assignment. Logs show SWx SAA with result code 5003.

Possible causes:

- Subscriber not authorized for WiFi calling service
- APN not permitted for this subscriber
- Subscription profile restrictions

Resolution:

1. Check the subscriber's service profile in the HSS
2. Verify WiFi calling / ePDG access is enabled for the subscriber
3. Confirm the requested APN is in the subscriber's allowed APN list

Roaming Not Allowed (Diameter 5004)

Symptoms: Roaming subscribers fail authentication. Logs show SWx MAA or SAA with result code 5004.

Possible causes:

- HSS roaming policy rejects the subscriber's current location
- WiFi calling not permitted for roaming subscribers

Resolution:

1. Review HSS roaming policies for the subscriber's HPLMN/VPLMN combination
2. Check if WiFi calling is permitted under roaming agreements

Authentication Timeout

Symptoms: Authentication hangs and then fails after 10 seconds. Logs show `Timeout swm_der_timeout` in `state_wait_auth_resp`.

Possible causes:

- HSS not responding to SWx MAR within 10 seconds
- SWx Diameter connection dropped during the request
- HSS overloaded

Resolution:

1. Check HSS responsiveness and load
2. Verify SWx Diameter peer is in OKAY state (not SUSPECT or DOWN)
3. Check `dia_swx_transmit_timer` is adequate for the network latency to HSS

EAP-AKA Type Mismatch

Symptoms: Authentication fails with "type_mismatch" error in logs. The UE identity prefix doesn't match the EAP method used.

Possible causes:

- UE sends identity with prefix `0` (EAP-AKA) but network expects EAP-AKA', or vice versa
- HSS returns authentication vectors for wrong EAP type

Background: Per 3GPP TS 23.003, the NAI identity prefix indicates the expected authentication type:

- Prefix `0` indicates EAP-AKA
- Prefix `6` indicates EAP-AKA'

OmniEPDG automatically selects the authentication method based on the UE's identity prefix. Most WiFi calling UEs use prefix `0` (EAP-AKA).

Resolution:

1. Check the UE's NAI identity in logs to verify the prefix
2. Ensure the HSS is configured to return appropriate authentication vectors
3. Verify the SIM card is provisioned correctly for the expected authentication type

EAP-AKA RES Mismatch

Symptoms: Authentication fails after challenge/response. Logs show "RES mismatch" or "res_mismatch" error.

Possible causes:

- SIM card authentication failure
- Key derivation mismatch between UE and network
- Corrupted authentication vectors from HSS

Resolution:

1. Verify the SIM card is valid and not damaged
2. Check that the HSS returned valid authentication vectors (RAND, AUTN, XRES, CK, IK)

3. Enable debug logging to compare the expected XRES with received RES
4. If using test SIMs, verify the Ki and OP/OPc values match between SIM and HSS

GTP Tunnel Failures (GTP Mode Only)

Create Session Rejected by PGW

Symptoms: Authentication succeeds but tunnel creation fails. Logs show GTPv2-C Create Session Response with error cause.

Common cause codes and actions:

Cause Code	Name	Action
78	Missing or Unknown APN	Verify APN is configured on the PGW and matches subscriber profile
82	Denied in RAT	Check PGW policy allows WiFi (non-3GPP) access type
84	All Dynamic Addresses Occupied	PGW IP pool exhausted; expand pool or investigate leaks
92	User Authentication Failed	PGW-side auth failure; check S6b session authorization
93	APN Access Denied	Subscriber not authorized for APN on PGW
96	IMSI/IMEI Not Known	Subscriber unknown to PGW; verify S6b session was authorized
113	APN Congestion	APN overloaded; retry or investigate PGW capacity
120	GTP-C Entity Congestion	PGW control plane overloaded

Create Session Timeout

Symptoms: Tunnel creation hangs for 10 seconds then fails. Logs show `Timeout create_session_timeout` in `state_wait_create_session_resp`.

Possible causes:

- PGW not reachable at `gtpc_remote_ip`:`gtpc_remote_port`
- Firewall blocking UDP port 2123 between OmniEPDG and PGW
- PGW overloaded and not responding to GTPv2-C requests

Resolution:

1. Verify network connectivity to PGW on UDP port 2123
2. Check firewall rules allow UDP 2123 between OmniEPDG and PGW
3. Check PGW health and GTPv2-C processing capacity

GTP-U Tunnel Not Passing Traffic

Symptoms: Tunnel is established (Create Session succeeds) but subscriber traffic does not flow.

Possible causes:

- GTP-U kernel module not loaded
- `gtp_u_kmod` socket IP does not match the GTP-U tunnel endpoint address signaled to PGW
- Routing not configured for the GTP tunnel device
- Firewall blocking UDP port 2152 (GTP-U)

Resolution:

1. Verify the Linux kernel GTP module is loaded (`lsmod | grep gtp`)
2. Confirm the GTP tunnel device exists (`ip link show gtp0`)
3. Verify `gtp_u_kmod` `ip` matches `gtpc_local_ip` or the address signaled in Create Session Request
4. Check routing table includes routes through the GTP tunnel device
5. Verify firewall allows UDP port 2152 between OmniEPDG and PGW

Simple VPN Failures (Simple VPN Mode Only)

TUN Interface Not Created

Symptoms: OmniEPDG starts but no `omniepdg0` interface appears. Sessions fail at tunnel setup. Logs may show errors from `simple_vpn_route` during

startup.

Possible causes:

- OmniEPDG process lacks `CAP_NET_ADMIN` capability or is not running as root
- TUN/TAP kernel module not loaded
- Another process has already created an interface named `omniepdg0`

Resolution:

1. Verify the TUN kernel module is available (`lsmod | grep tun`)
2. Confirm OmniEPDG is running with sufficient privileges to create TUN interfaces
3. Check if `omniepdg0` already exists from a previous instance (`ip link show omniepdg0`)
4. Check `log/crash.log` for errors from the route manager process

IP Pool Exhausted

Symptoms: Authentication succeeds but tunnel setup fails. Logs show IP allocation failure from `simple_vpn_pool`.

Possible causes:

- All addresses in the configured CIDR pool are allocated to active sessions
- IP addresses not being released after session teardown (leak)
- Pool size too small for the number of concurrent subscribers

Resolution:

1. Check the number of active `epdg_ue_fsm` processes against the pool size
2. Verify sessions are being torn down properly (check for `terminating` log messages)
3. If the pool is genuinely full, expand it by using a larger CIDR prefix in `simple_vpn_pool_ipv4` (requires restart)
4. Check for FSM crashes during teardown in `log/crash.log` that may have prevented IP release

Subscriber Traffic Not Flowing

Symptoms: Session is established and the UE receives an IP address, but traffic does not flow through the TUN interface.

Possible causes:

- Host route not added for the subscriber IP on `omniepdg0`
- IP forwarding not enabled on the OmniEPDG host
- Firewall rules blocking traffic on the `omniepdg0` interface
- Missing NAT/masquerade rules for outbound traffic from the subscriber IP range

Resolution:

1. Verify the host route exists (`ip route show` and look for the subscriber's /32 route via `omniepdg0`)
2. Confirm IP forwarding is enabled (`sysctl net.ipv4.ip_forward`)
3. Check iptables/nftables rules allow forwarding through `omniepdg0`
4. If subscribers need internet access, verify NAT/masquerade is configured for the subscriber IP range (e.g., `iptables -t nat -A POSTROUTING -s 10.45.0.0/16 -o <wan-interface> -j MASQUERADE`)

Stale Routes After Crash

Symptoms: Host routes for subscriber IPs remain in the routing table after OmniEPDG restarts or after sessions terminate abnormally.

Possible causes:

- FSM crashed before the route could be removed
- OmniEPDG process was killed without graceful shutdown

Resolution:

1. Check `log/crash.log` for process crashes during teardown
2. Manually remove stale routes (`ip route del <subscriber-ip>/32 dev omniepdg0`)

3. Restarting OmniEPDG will recreate the `omniepdg0` interface, which removes all associated routes

Session Teardown Issues

Teardown Hangs During Deregistration

Symptoms: Session teardown does not complete. UE FSM stuck in a `dereg_*` or `wait_*` state.

Possible causes:

- PGW not responding to Delete Session Request
- Diameter peer not responding to STR or ASR
- Timeout cascading not completing due to multiple timeouts stacking

Resolution:

1. Check logs for timeout messages in the relevant state
2. Verify PGW and HSS connectivity
3. After 10 seconds, the FSM should time out and proceed to the next teardown step or terminate. If it does not, check for unexpected events logged as `Unexpected call event`

Orphaned GTP-U PDP Contexts

Symptoms: GTP-U tunnel entries remain in the kernel after sessions terminate. `ip link show gtp0` shows the device still has active PDP contexts.

Possible causes:

- FSM terminated abnormally before deleting the PDP context
- Crash during teardown sequence

Resolution:

1. Check `log/crash.log` for process crashes during teardown

2. The FSM's `terminate/3` callback attempts to clean up the PDP context. If the FSM was killed (e.g., supervisor restart), the cleanup may have been skipped
3. Restarting OmniEPDG will recreate the GTP-U socket and clear stale contexts

Process and System Issues

Supervisor Restart Loops

Symptoms: OmniEPDG processes restart repeatedly. Logs show supervisor restart messages and crash reports.

Possible causes:

- Persistent configuration error causing a handler to crash on startup
- External dependency unavailable (e.g., gen_socket library not found)
- Diameter peer sending malformed messages causing handler crashes

Resolution:

1. Check `log/crash.log` for the root cause of the crash
2. Verify `gen_socket` `libdir` path is correct and the library files exist
3. Check all required configuration parameters are present in `config/runtime.exs`
4. Look for malformed Diameter messages in the crash report

High Memory Usage

Symptoms: Erlang VM memory consumption grows over time.

Possible causes:

- UE FSM processes not being cleaned up after session teardown
- Log message accumulation in mailboxes
- Large number of concurrent sessions

Resolution:

1. Check the number of running `epdg_ue_fsm` and `aaa_ue_fsm` processes (these should match active subscriber count)
2. Verify FSMs are terminating properly after session teardown (check for `terminating` log messages)
3. Review log rotation settings to ensure log files are being rotated

OmniEPDG Operations Guide

OmniEPDG is an evolved Packet Data Gateway (ePDG) that enables Voice over WiFi (VoWiFi) calling. It authenticates mobile subscribers over untrusted WiFi networks using EAP-AKA, and connects them to the mobile core network through Diameter signaling to the HSS and GTP tunnels to a Packet Gateway (PGW).

The OmniEPDG control panel showing an active subscriber session with real-time traffic statistics.

OmniEPDG supports two operational modes:

- **GTP Mode** (default) - Full 3GPP-compliant tunneling through a PGW via GTPv2-C and GTP-U
- **Simple VPN Mode** - Local breakout with a built-in IP pool and Linux TUN interface, no PGW required

Documentation

Configuration & Operations

- **Architecture & Call Flows** - System architecture, protocol interfaces, UE state machines, and message sequence diagrams for both modes
- **Configuration Reference** - Complete parameter documentation for Diameter, GTPv2-C, GTP-U, Simple VPN, and logging
- **Control Panel** - Web-based monitoring UI for sessions, Diameter peers, and logs

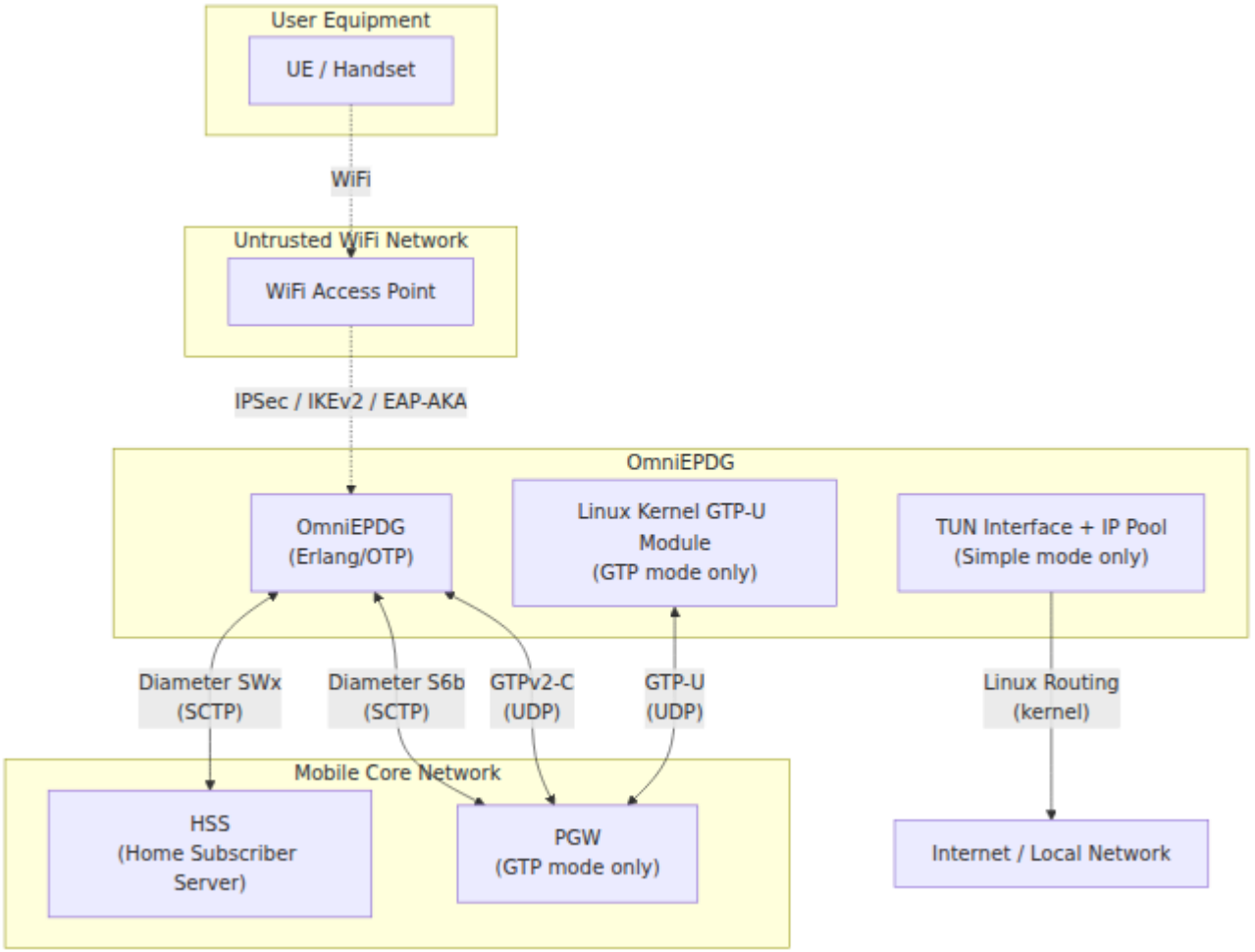
Security

- **Security Guide** - Authentication rate limiting and GeoIP country blocking

Monitoring & Troubleshooting

- **Metrics Reference** - Prometheus metrics for monitoring authentication, sessions, Diameter signaling, and system health
- **Troubleshooting** - Common issues, diagnostic procedures, and resolution steps

Operational Modes



GTP Mode

The default mode. OmniEPDG tunnels all subscriber traffic through a PGW using GTPv2-C for session control and GTP-U (via Linux kernel module) for the user plane. This is fully 3GPP compliant and suitable for carrier deployments with existing EPC infrastructure.

Traffic path: UE → IPSec → OmniEPDG → GTPv2-C Create Session → GTP-U tunnel → PGW → Internet

Required infrastructure: HSS, PGW

Simple VPN Mode

OmniEPDG allocates IP addresses from a local pool and routes subscriber traffic directly through a Linux TUN interface (`tun_epdg`) using standard kernel routing. No PGW or GTP infrastructure is needed. Authentication still happens via Diameter SWx to the HSS.

Traffic path: UE → IPSec → OmniEPDG → Local IP allocation → TUN interface → Linux routing → Internet

Required infrastructure: HSS only (PGW not needed)

Optional optimization: The `skip_sar` flag bypasses HSS Server-Assignment-Request/Answer, reducing connection setup time. This means the HSS will not track which ePDG serves the subscriber and HSS-initiated procedures (deregistration, profile push) will not function. Suitable for private deployments without roaming requirements.

Mode Comparison

Capability	GTP Mode	Simple VPN Mode
3GPP compliant	Yes	No (with `skip_sar`), Partial (without)
PGW required	Yes	No
HSS required	Yes	Yes (authentication only)
IP allocation	From PGW	Local pool (CIDR)
User plane	GTP-U kernel module	Linux TUN + routing
HSS profile push	Yes (PPR/PPA)	No
HSS deregistration	Yes (RTR/RTA)	No (with `skip_sar`)
PGW-initiated teardown	Yes	N/A
Roaming support	Yes	No
IPv6 / Dual-stack	Yes	IPv4 only

Protocol Interfaces

Interface	Protocol	Transport	Mode	Peer	Purpose	Ref
SWu	IKEv2 / IPSec	UDP	Both	UE	Secure tunnel and EAP-AKA authentication	3GF 33.4
SWx	Diameter	SCTP	Both	HSS	Authentication vectors and server assignment	3GF 29.2 Sec
S6b	Diameter	SCTP	GTP only	PGW	Session authorization and policy	3GF 29.2 Sec
S2b	GTPv2-C / GTP-U	UDP	GTP only	PGW	Control and user plane tunnel	3GF 29.2

Features

Core Functionality

- **EAP-AKA Authentication** - Full 3GPP EAP-AKA subscriber authentication via HSS
- **IPSec Tunnel Management** - IKEv2-based secure tunnel between UE and ePDG
- **Dual Operational Modes** - GTP tunneling to PGW or local breakout with Simple VPN
- **Per-UE State Machines** - Independent Erlang FSM per subscriber for session lifecycle management

- **Dual-Stack Support** - IPv4, IPv6, and IPv4v6 PDP address types (GTP mode)

GTP Mode Features

- **GTP Tunnel Establishment** - GTPv2-C session creation and GTP-U user plane via Linux kernel module
- **PGW-Initiated Teardown** - PGW sends Delete Bearer Request, ePDG cascades teardown to UE
- **Network-Initiated Teardown** - HSS triggers deregistration via SWx RTR, ePDG tears down all sessions
- **Re-Authentication** - HSS-triggered profile push and re-authorization per 3GPP TS 29.273 Section 7.1.2.5.1

Simple VPN Mode Features

- **Local IP Pool** - CIDR-based IPv4 address allocation with per-IMSI tracking
- **TUN Interface Routing** - Standard Linux TUN device (`tun_epdg`) with per-UE host routes
- **DNS Configuration** - Configurable DNS servers provided to UEs via PCO
- **Optional SAR Skip** - Bypass HSS registration for faster connection setup

Security Features

- **Authentication Rate Limiting** - Per-IP and per-IMSI brute-force protection with configurable thresholds
- **GeoIP Country Blocking** - Whitelist or blacklist based country access control using MaxMind GeoLite2
- **Dead Peer Detection** - Active liveness monitoring with configurable probes
- **ESP Anti-Replay Protection** - RFC 4303 compliant 64-bit sliding window

HSS Integration (SWx Diameter)

- **Multimedia-Auth-Request/Answer (MAR/MAA)** - Retrieve EAP-AKA authentication vectors (both modes)

- **Server-Assignment-Request/Answer (SAR/SAA)** - Download subscriber profile and APN configuration (skippable in Simple mode)
- **Push-Profile-Request/Answer (PPR/PPA)** - Receive updated subscriber profiles from HSS (GTP mode)
- **Registration-Termination-Request/Answer (RTR/RTA)** - HSS-initiated subscriber deregistration (GTP mode)

PGW Integration (GTP Mode Only)

S6b Diameter:

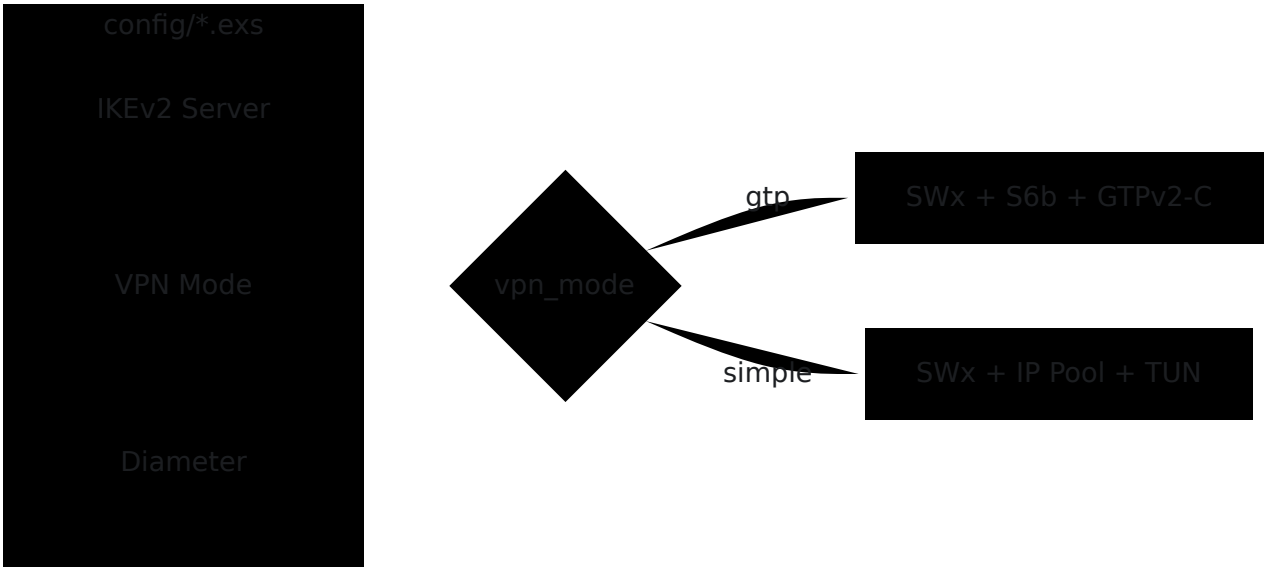
- **AA-Request/Answer (AAR/AAA)** - Authorize PGW sessions
- **Session-Termination-Request/Answer (STR/STA)** - Terminate PGW sessions
- **Re-Auth-Request/Answer (RAR/RAA)** - Re-authorize active sessions
- **Abort-Session-Request/Answer (ASR/ASA)** - Forcefully terminate sessions

S2b GTPv2-C:

- **Create Session Request/Response** - Establish GTP tunnels with TEID allocation
- **Delete Session Request/Response** - Tear down GTP tunnels
- **Delete Bearer Request/Response** - PGW-initiated bearer management

Quick Start

Configuration Structure



Configuration is done in `config/runtime.exs` or via environment variables. The `vpn_mode` parameter selects between GTP and Simple VPN modes. See the Configuration Reference for complete parameter documentation.

Typical Network Addressing (GTP Mode)

Component	IP Address	Port	Notes
OmniEPDG (GTP-U)	10.74.0.11	-	GTP-U tunnel endpoint
OmniEPDG (Diameter S6b)	10.74.0.12	3868	S6b Diameter listener
HSS	10.74.0.21	3868	SWx Diameter peer
PGW	10.74.0.23	2123	GTPv2-C and S6b peer

Typical Network Addressing (Simple VPN Mode)

Component	IP Address	Notes
OmniEPDG (TUN gateway)	10.44.0.1	Gateway IP on the `tun_epdg` interface
UE IP pool	10.45.0.0/16	Configurable CIDR pool for subscriber IPs
HSS	10.74.0.21:3868	SWx Diameter peer (authentication only)

3GPP Specifications

Specification	Title	Relevance
TS 29.273	EPS AAA interfaces (SWx, S6b, SWm)	Primary specification for ePDG Diameter interfaces
TS 29.274	GTPv2-C and GTP-U	S2b tunnel control and user plane (GTP mode)
TS 33.402	Security for non-3GPP accesses	EAP-AKA authentication for untrusted WiFi
TS 23.402	Architecture enhancements for non-3GPP accesses	Overall ePDG architecture and procedures
TS 23.003	Numbering, addressing and identification	NAI format, IMSI structure
TS 29.229	Cx/Dx Diameter (common definitions)	Server-Assignment-Type values used by SWx
RFC 6733	Diameter Base Protocol	Diameter transport, peer management, watchdog
RFC 4187	EAP-AKA	Authentication method used over IKEv2

Documentation By Role

Network Operators:

1. Start with the Architecture & Call Flows to understand the system and both operational modes
2. Review the Configuration Reference for deployment parameters

3. Review the Security Guide to configure rate limiting and GeoIP blocking
4. Set up monitoring using the Metrics Reference for Prometheus integration
5. Keep the Troubleshooting guide available for operations

System Integrators:

1. Review the Architecture & Call Flows for interface details and state machines
2. Use the Configuration Reference for peer connectivity setup
3. Configure alerting using the Metrics Reference
4. Reference the 3GPP specifications table above for protocol compliance