

OmniHSS

概要

OmniHSS は 4G LTE (EPC) と IMS (IP) の HSS を統合し、

OmniHSS は Elixir と Erlang VM を採用しています。

機能

HSS と LTE と IMS の統合機能

- 統合機能 - 統合機能
- 統合機能 - 統合機能
- 統合機能 - 統合機能
- 統合機能 - 統合機能
- 統合機能 - 統合機能
- 統合機能 - 統合機能 (EIR) 統合機能

接続

接続

- **S6a** 接続 - LTE/EPC 統合機能
- **Cx** 接続 - IMS 統合機能
- **Sh** 接続 - IMS 統合機能
- **S13** 接続 - 統合機能 (OmniHSS と EIR)
- **Gx** 接続 - 統合機能 (OmniHSS と PCRF)
- **Rx** 接続 - IMS 統合機能 (OmniHSS と PCRF)

- **PLMN** - PLMN IMS
- **MSISDN** - MSISDN
- **RESTful API** - RESTful API (OmniHLR)
- **Web** - Web

OmniHSS

OmniHSS

- **MME** (MME) - LTE MME
- **P-GW** (PDN GW) - OmniHSS (PCRF)
- **P-CSCF** (P-CSCF) - IMS P-CSCF
- **I-CSCF** (I-CSCF) - IMS I-CSCF
- **S-CSCF** (S-CSCF) - IMS S-CSCF
- **AS** (AS) - IMS AS
- **OmniHLR** - OmniHLR API OmniHSS HLR

OmniHLR

OmniHLR

API

- **API** - Diameter
- **API** - Diameter
- **API** - Diameter

API

- **API** - Diameter
- **API** - Diameter
- **API** - Diameter
- **API** - Diameter
- **Webhooks** - Webhooks

□□□□

- **□□□□** - EPC□IMS□APN □□□□□
- **□□□□** - □□□□□□
- **□□□□** - Diameter □□□□□□□□
- **PCRF** - □□□□□□□□ (Gx/Rx □□□QoS□VoLTE)
- **EIR** - □□□□□□ (S13 □□□IMEI □□)
- **□ MSISDN □□ IMSI □□** - □□□□□□□□□□ IMSI

□□□□

- **Galera □□□□□** - □□ HA □□□□□□□□

□□□□□□□

□□□□

□□□□ (Web □□)

URL: https://[hostname]:7443

□□□□□□□□□□ Diameter □□□□□□□□□□

API □□

URL: https://[hostname]:8443

RESTful API □□□□□□□□□□

□□□□□□

- `config/runtime.exs` - □□□□□ (□□□□Diameter□□□□□□)
- `priv/cert/` - HTTPS □ Diameter □ TLS □□

□□□□

1. □□□□□□ - □□□□□□□□□□
2. □□ **Diameter** □□□ - □□□□□□ Diameter □□
3. □□□□ - □□ API □□ `/api/subscriber/imsi/:imsi`
4. □□□□□□ - □□□□□□□□□□□□ SQL □□□

□□□□□□□□

□□□□

□□□□□□□□ stdout/stderr□□□□□□□□□□□□ (systemd□supervisord □) □□□

□□□□

- **Diameter** □□□ - □□ Diameter □□□□□□□□□□
- □□□□□□ - □□ runtime.exs □□□□□□□□
- □□□□□□ - □□□□□□□□□□□□□□

□□□□

- **API** □□□□ - `GET /api/status`
- □□□□ - □□□□□□□□□□□□
- □□□ - □□□□ SQL □□□□□□□□□□

□□□□

- □□ **TLS** - API □□□□□□□□□□ HTTPS
- □□□□ - `priv/cert/` □□□□□□□□□□
- □□□□□□ - □□ runtime.exs □□□□□□□□□□
- □□□□□□ - Diameter □□□□□□□□□□□□
- **API** □□□ - □□□□□□□□□□□□□□

EIR

00

HSS EIR EIR IMEI

0000

- **S13** Diameter
- **IMEI** IMEI/IMEISV
- IMEI/IMEISV IMSI
-
-
- **REST API** CRUD EIR

00

Diameter

	ID		
S13	16,777,252	MME/SGSN	

00000000

EIR

EIR_RULE		
int	id	PK
string	action	
string	regex	
timestamp	inserted_at	
timestamp	updated_at	

□□□□

- `whitelist` - □□□□
- `blacklist` - □□□□
- `greylist` - □□□□

□□□□□□□□□□ IMEI□IMEISV □ IMSI □□

□□□□□

□□	□□	□□	□□□□
□□□	0	□□□□□	□□□□□□
□□□	1	□□□□/□□□	□□□□□□
□□□	2	□□□□□□	□□□□□

S13 □□

□□□□□

□□□□□□□□□□ **ECR**□/□□□□□□□□□□ **ECA**□

□□□ MME/SGSN → HSS□EIR□

MMME

AVPs

- Session-Id
- Origin-Host, Origin-Realm
- Destination-Realm
- Auth-Session-State
- Terminal-Information
 - IMEI15
 - Software-Version2
- User-NameIMSI
- Vendor-Specific-Application-Id

EIR

1. IMEISoftware-Version IMSI
2. IMSI
 -
 -
3.
 - **IMEISV** IMEI + Software-Version
 - **IMEI** IMEI
 - **IMSI**
 -
- 4.

AVPs

- Session-Id
- Result-Code: 2001
- Equipment-Status: 0/ 1/ 2

- Experimental-Result: 5422/
- Experimental-Result: 5012

□□□□□□

□□□□

EIR □□□□□□□□□□□□□□□□

1. IMEISV□IMEI + Software-Version□
↓ □□□□□□□□
2. □ IMEI
↓ □□□□□□□□
3. IMSI□□□□□□□□□□□□
↓ □□□□□□□□
4. □□□□□□

□□□□

□□ **1**□IMEISV □□

- □□ IMEI + Software-Version□ "35979139461611" + "08" =
"3597913946161108"
- □□□□ EIR □□□□□□□□□□
- □□□□□□□□□□□□□□ "whitelist"□"blacklist"□"greylist"□

□□ **2**□IMEI □□□□□□

- □□□ IMEI□ "35979139461611"
- □□□□ EIR □□□□□□□□□□
- □□□□□□□□□□□□□□

□□ **3**□IMSI □□□□□□□□ IMSI □□□□

- □□□□□□ IMSI□ "999999876543210"
- □□□□ EIR □□❓❓❓□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□ **4**□□□□□□□□□□□□□□

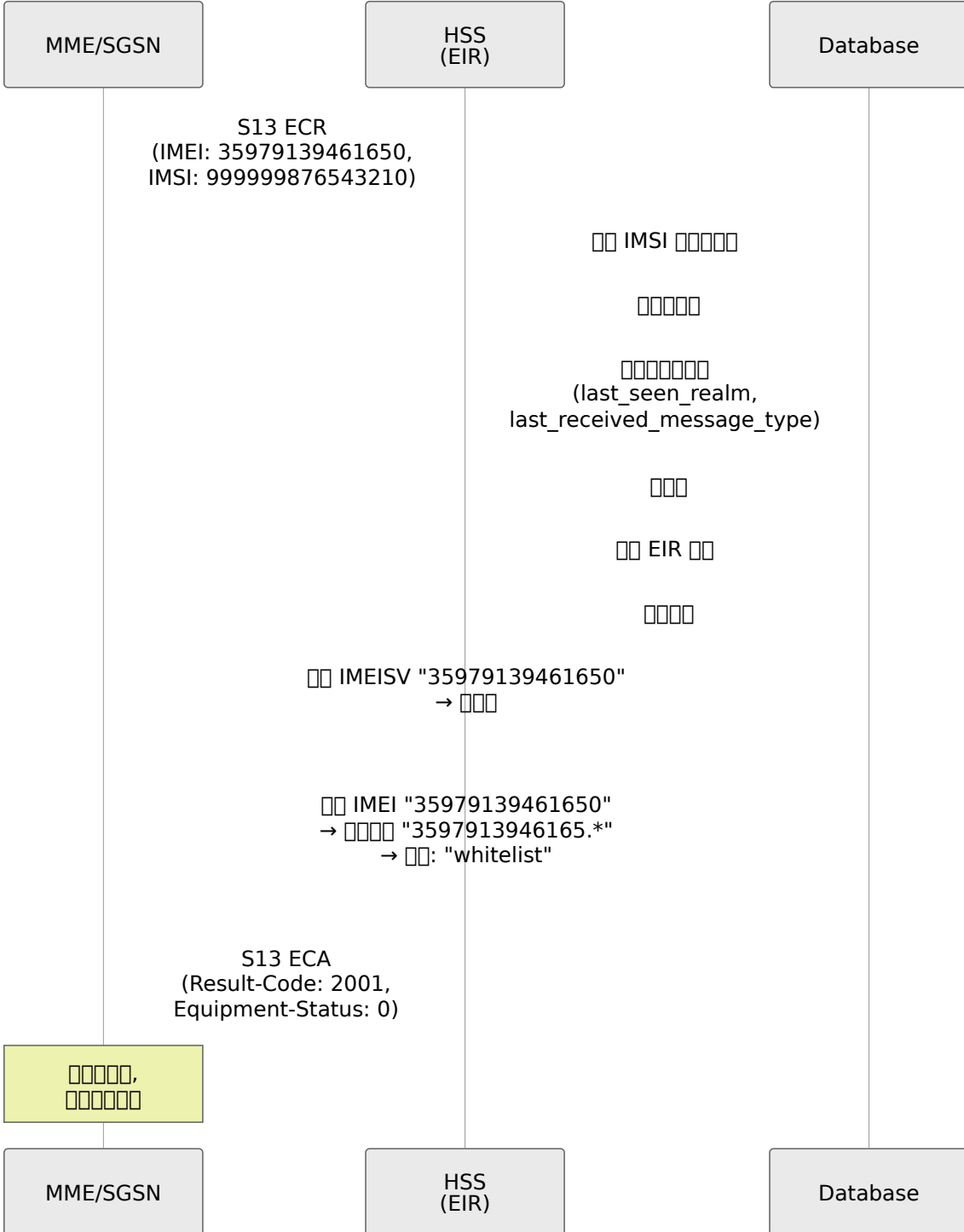
- `eir_unknown_equipment_behaviour`
- `:`
 - `:whitelist` - `0000000000`
 - `:blacklist` - `0000000000`
 - `:greylist` - `0000000000`
 - `:reject_unknown_equipment` - `0000 54220000`

`0000000000`

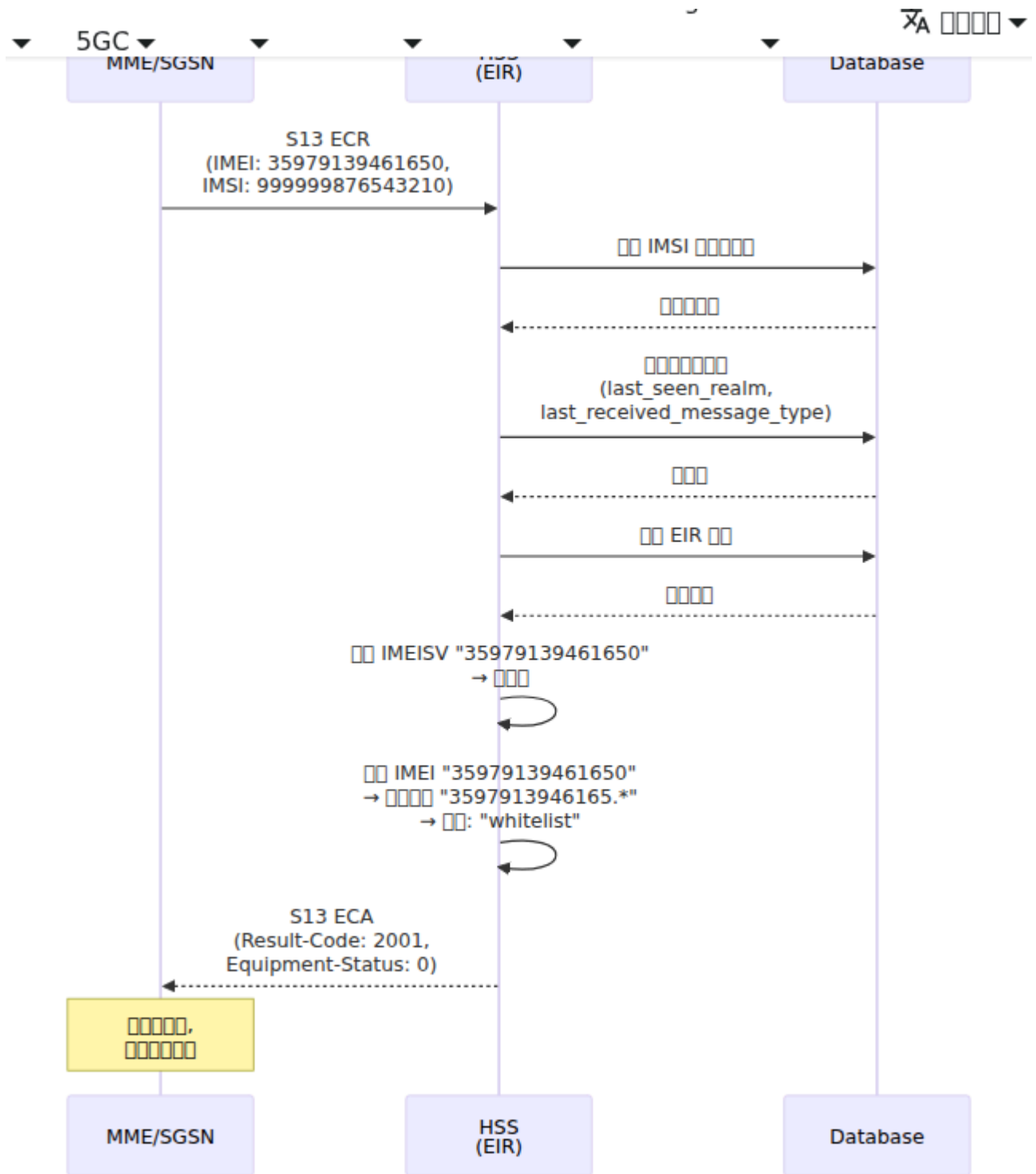
<code>00</code>	<code>00</code>	<code>00</code>
<code>"35979139461650"</code>	<code>00 IMEI</code>	<code>00000000/0000</code>
<code>"3597913946165.*"</code>	<code>IMEI 000000</code>	<code>0000/0000</code>
<code>"3597913946161108"</code>	<code>00 IMEISV</code>	<code>0000000000</code>
<code>"999999876543210"</code>	<code>IMSI</code>	<code>000000000000</code>
<code>"359791.*"</code>	<code>TAC 0000</code>	<code>0000000000</code>

□□□□□□

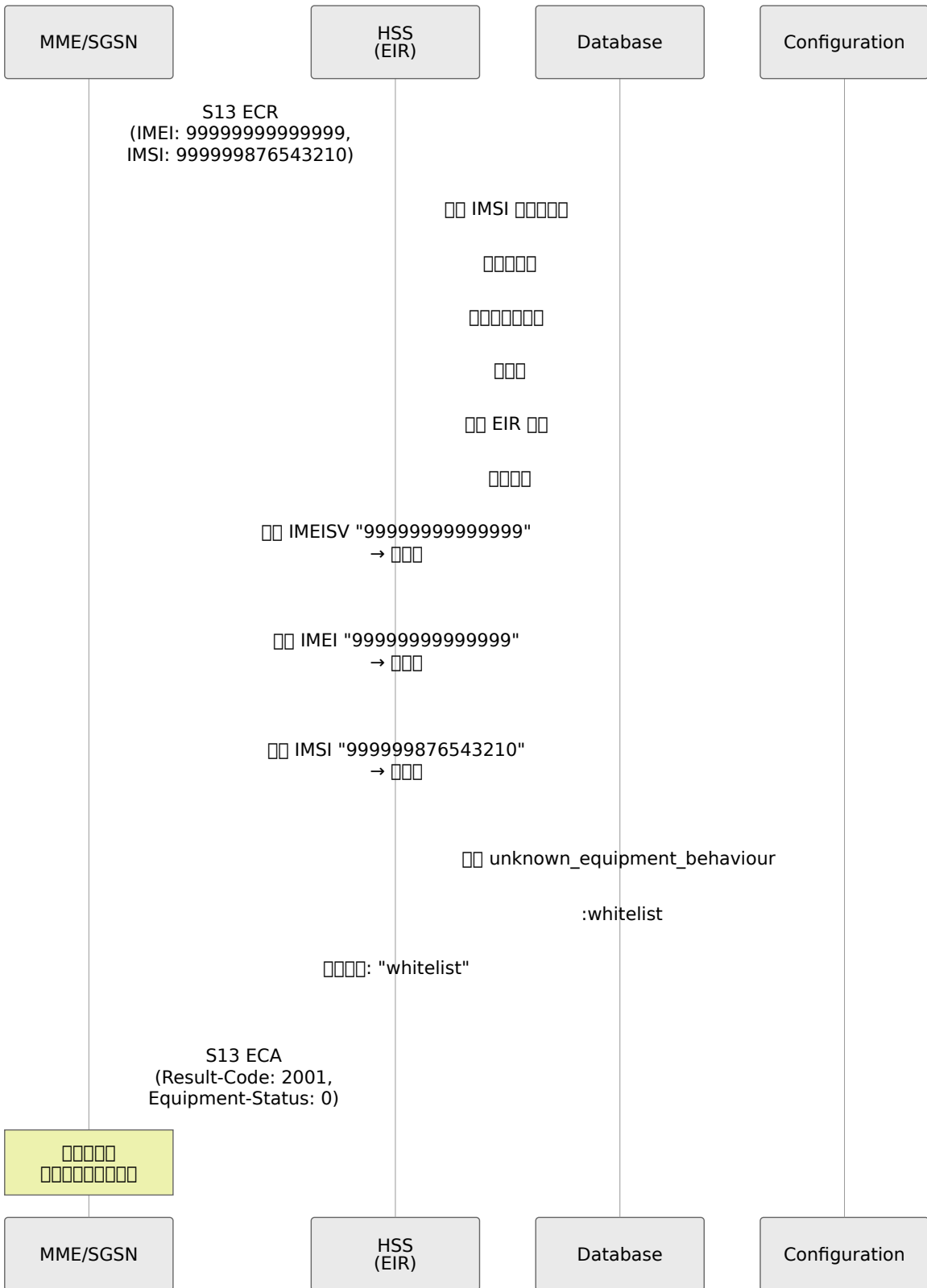
□□ 1□□□□□ - □□□□□ IMEI



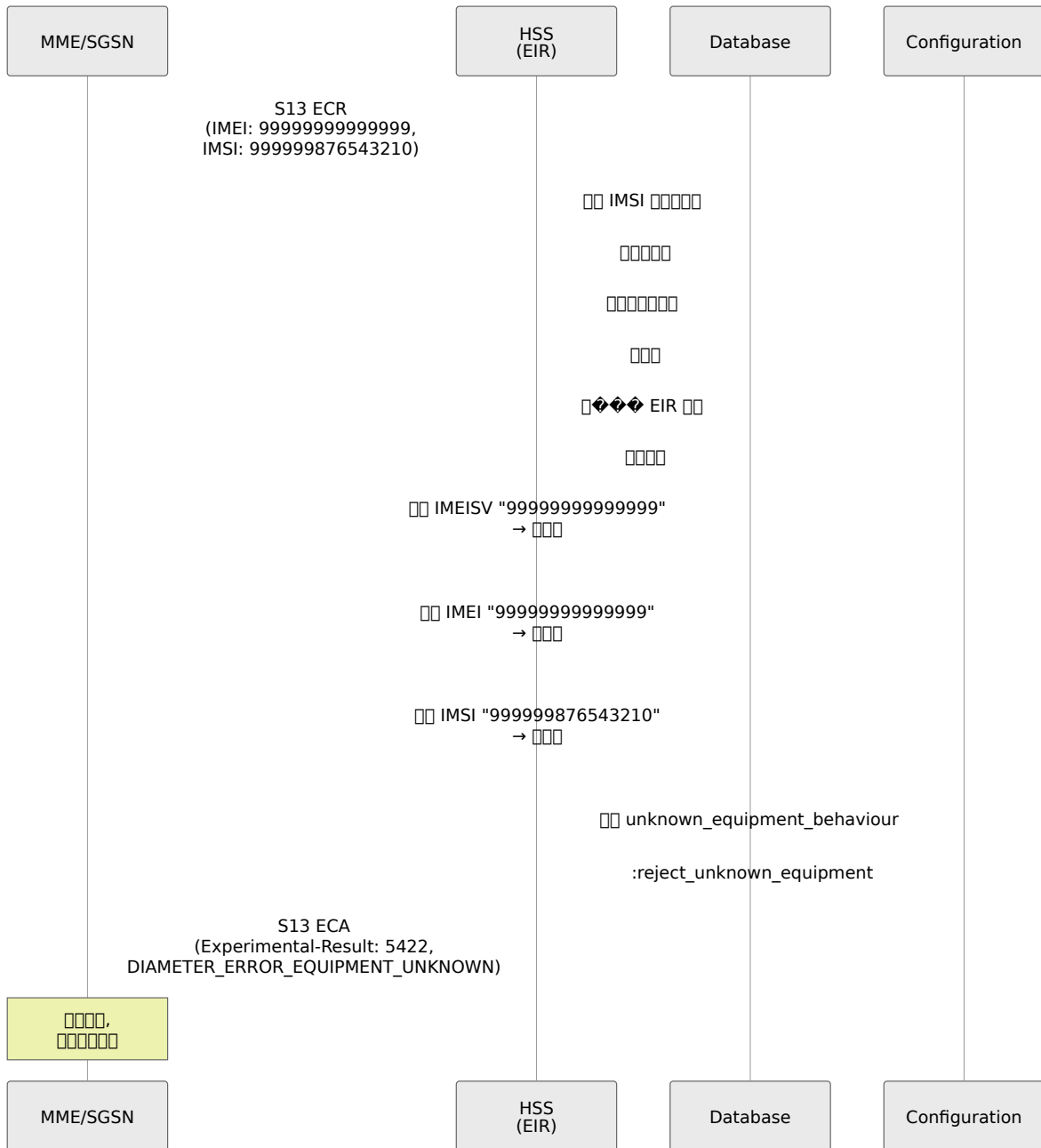
2 - IMEI



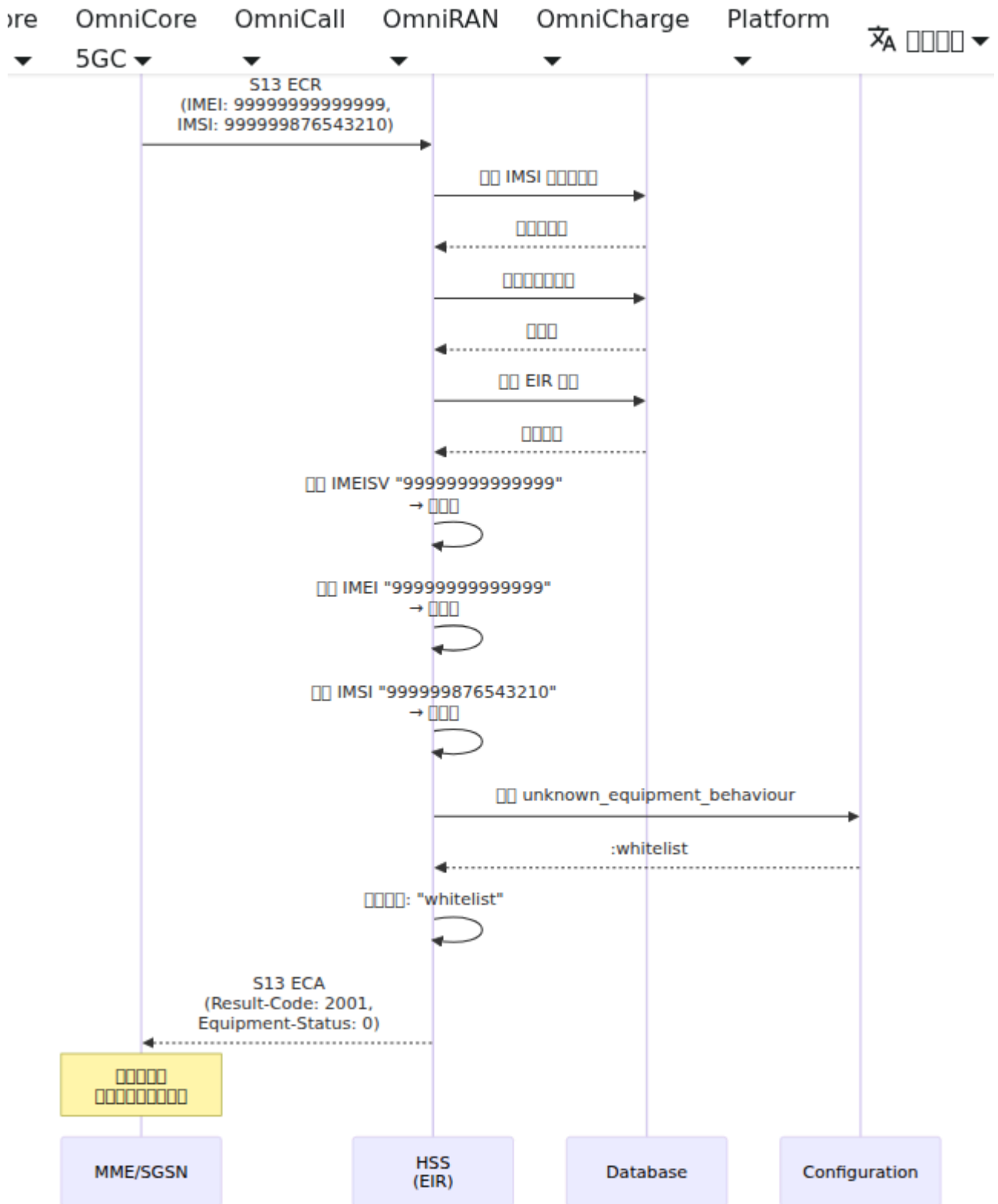
3 -



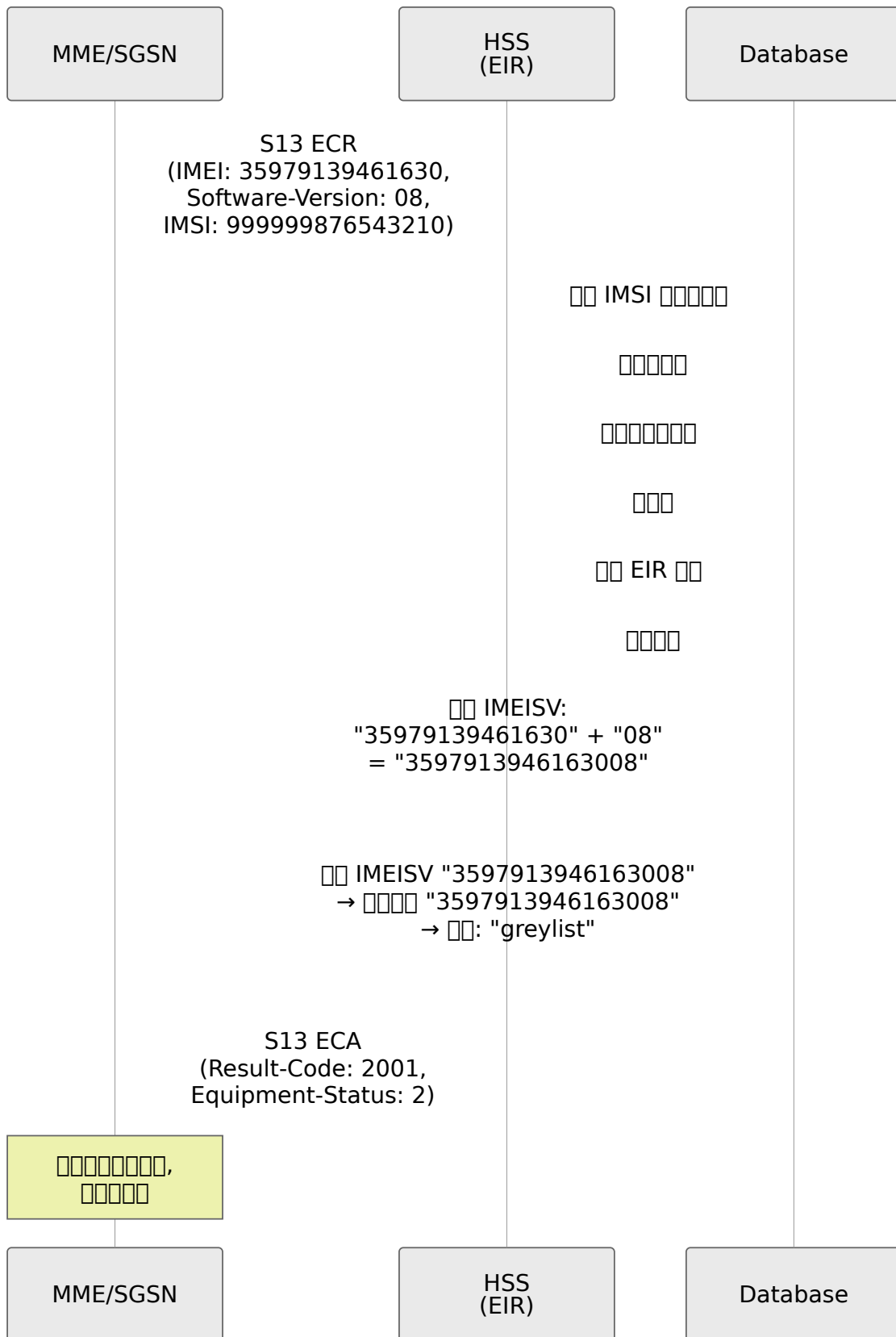
4 -



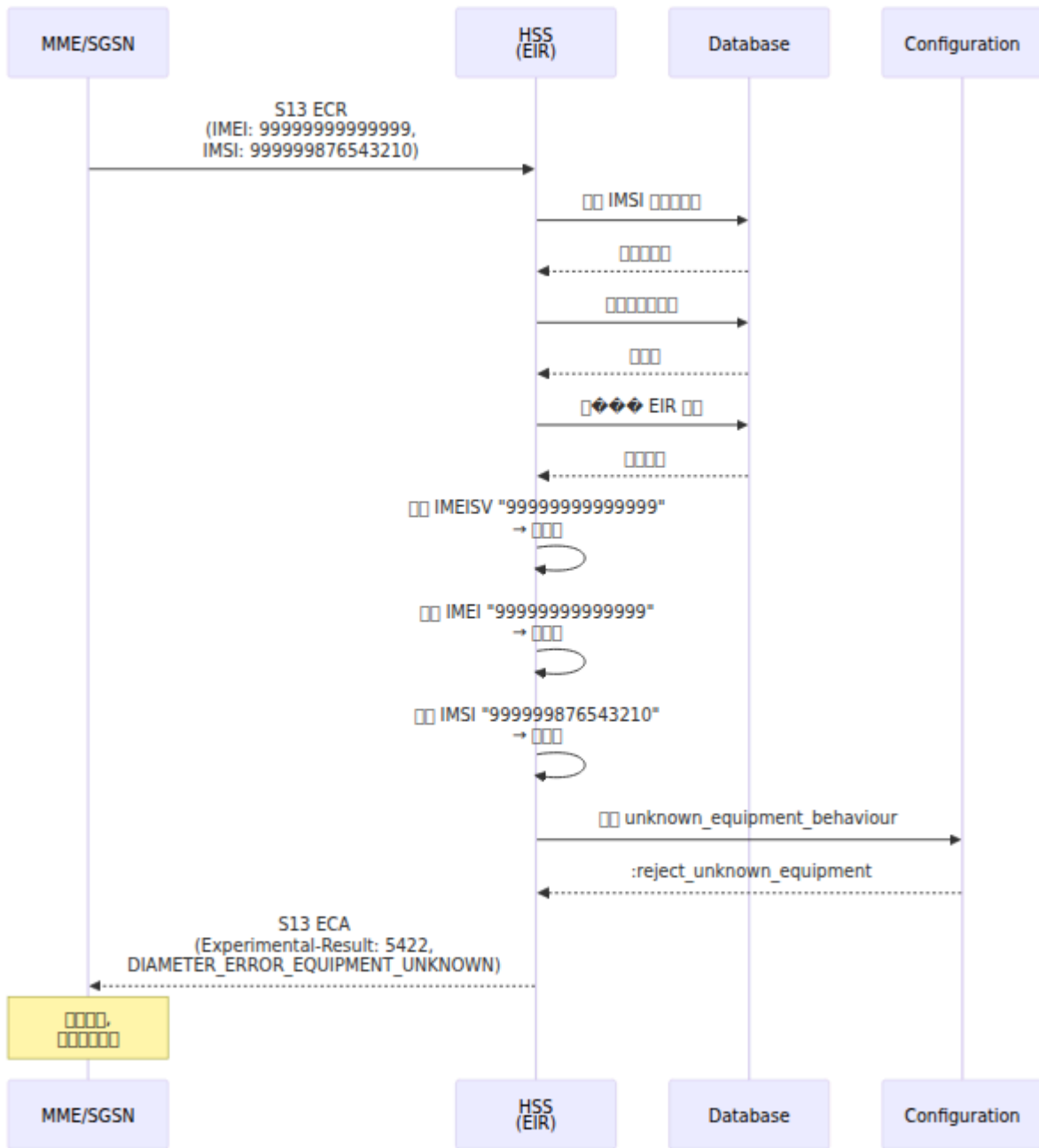
5G - 5G



6 - IMEISV



7 - IMSI



REST API

EIR

`/api/eir/rule`

EIR

□□□

```
GET /api/eir/rule
```

□□□HTTP 200□□

```
{
  "data": [
    {
      "id": 1,
      "action": "whitelist",
      "regex": "3597913946165.*",
      "inserted_at": "2025-01-15T10:30:00Z",
      "updated_at": "2025-01-15T10:30:00Z"
    },
    {
      "id": 2,
      "action": "blacklist",
      "regex": "35979139461640",
      "inserted_at": "2025-01-16T14:20:00Z",
      "updated_at": "2025-01-16T14:20:00Z"
    }
  ]
}
```

□□□□ **EIR** □□

□□□

```
GET /api/eir/rule/{id}
```

□□□HTTP 200□□


```
{
  "errors": {
    "regex": [""]
  }
}
```

⌨ **EIR** ⌨⌨⌨⌨

⌨⌨

```
PATCH /api/eir/rule/{id}
Content-Type: application/json
```

```
{
  "action": "greylist"
}
```

⌨⌨HTTP 200⌨⌨

```
{
  "data": {
    "id": 3,
    "action": "greylist",
    "regex": "35979139461640"
  }
}
```

⌨ **EIR** ⌨

⌨⌨

```
PUT /api/eir/rule/{id}
Content-Type: application/json
```

```
{
  "action": "whitelist",
  "regex": "359791394616.*"
}
```

HTTP 200

```
{
  "data": {
    "id": 3,
    "action": "whitelist",
    "regex": "359791394616.*"
  }
}
```

EIR

```
DELETE /api/eir/rule/{id}
```

HTTP 204

Diameter

s13 config/runtime.exs

```
%{
  application_name: :s13,
  application_dictionary: :diameter_gen_3gpp_s13,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_252}
  ]
}
```

config/runtime.exs

□□□

```
config :hss, :eir,
  unknown_equipment_behaviour: :whitelist
```

□□□□

- `:whitelist` - □□□□□□□□□□□□□□
- `:blacklist` - □□□□□□□□□□
- `:greylist` - □□□□□□□□□□
- `:reject_unknown_equipment` - □□ Diameter □□ 5422□□□□

□□□

- □□/□□□ `:whitelist` - □□□□□□□
- □□□□□□□□ `:whitelist` - □□□□□□□□□□
- □□□□□□□□ `:greylist` - □□□□□□□□□□
- □□□□□□□□ `:reject_unknown_equipment` - □□□□□□□□

□□□□□

□□□□	□□	□□	□□
2001	□□	DIAMETER_SUCCESS	□□□□□□□
5422	□□	DIAMETER_ERROR_EQUIPMENT_UNKNOWN	□□□□□□□□□□□□□□
5012	□□	DIAMETER_ERROR_UNKNOWN	□□□□

□□

1. □□□□□□□

□□□□□□□□□□

□□□

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "35979139461640" # □□ IMEI
}
```

□□□□□□□□□□□□□□□□□□

2. □□□□□□□

□□□□□□□□□□□□□□□□

□□□

```
POST /api/eir/rule
{
  "action": "whitelist",
  "regex": "359791394.*" # □□□□/□□□ TAC
}
```

□□□□□ TAC □□□□□□□□□□

3. □□□□□□□

□□□□□□□□□□□□□□□□SIM □□

□□□

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "999999876543210" # IMSI
}
```

□□□□□□□□ SIM □□□□□□□□

4. 黑名单

黑名单

POST

```
POST /api/eir/rule
{
  "action": "greylist",
  "regex": "35979139.*" # 黑名单 TAC 00
}
```

黑名单

5. 黑名单

黑名单

POST

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "359791394616.*05" # IMEI 00 + 黑名单 05
}
```

黑名单 IMEI 黑名单 "05" 黑名单

黑名单

黑名单

EIR 黑名单

- **S13** 黑名单 - ECR/ECA 黑名单
- 黑名单 - 黑名单 IMEI/IMEISV/IMSI 黑名单

- **EIR** 查詢 - 查詢設備
- **REST API** 查詢 - 查詢設備

查詢設備

查詢設備

1. **IMEISV** 查詢設備 IMEI + 查詢
2. **IMEI** 查詢設備 IMEI
3. **IMSI** 查詢設備 IMSI
4. 查詢設備

查詢設備

- `whitelist` - 查詢設備
- `blacklist` - 查詢設備
- `greylist` - 查詢設備
- `reject_unknown_equipment` - 查詢設備

查詢設備

IMEI 查詢

IMEI 查詢設備EIR

- 查詢設備 IMEI
- 查詢設備
- 查詢 API 查詢設備

查詢設備

EIR 查詢設備 ID

```

查詢 1: 查詢 "359791.*" 查詢 "whitelist" (查詢)
查詢 2: 查詢 "35979139461640" 查詢 "blacklist" (查詢)

```

IMEI 15 Digit Breakdown

Introduction

IMEI is a unique identifier for mobile devices.

- IMEI structure: S13 ECR 8
- REST API EIR 4
- IMEI 15

Structure

- Diameter 4 - S13 8
- API 4 - REST API 4
- HSS 4
- IMEI 15

IMEI Breakdown

IMEI 15 Digit Breakdown

```
35 9791 394616 1
|  |  |          |  |  |  |  |  |  |  |  |  |  |
|  |  |          |  |  |  |  |  |  |  |  |  |  |
|  |  |          |  |  |  |  |  |  |  |  |  |  |
|  |  |          |  |  |  |  |  |  |  |  |  |  |
└─ TAC 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
      └─ RBI 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
            └─ 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
```

IMEISV 16

35 9791 394616 1 08

| | | | | 2

IMEI15

IMEI/IMEISV		
359791394616108	3597913946161.*	TAC+FAC+ 359791394616*
359791394616140	35979139461614.	359791394616141-9
35979139461640	35979139461640	IMEI
3597913946163008	3597913946163008	IMEISV=IMEI + SV

PCRF

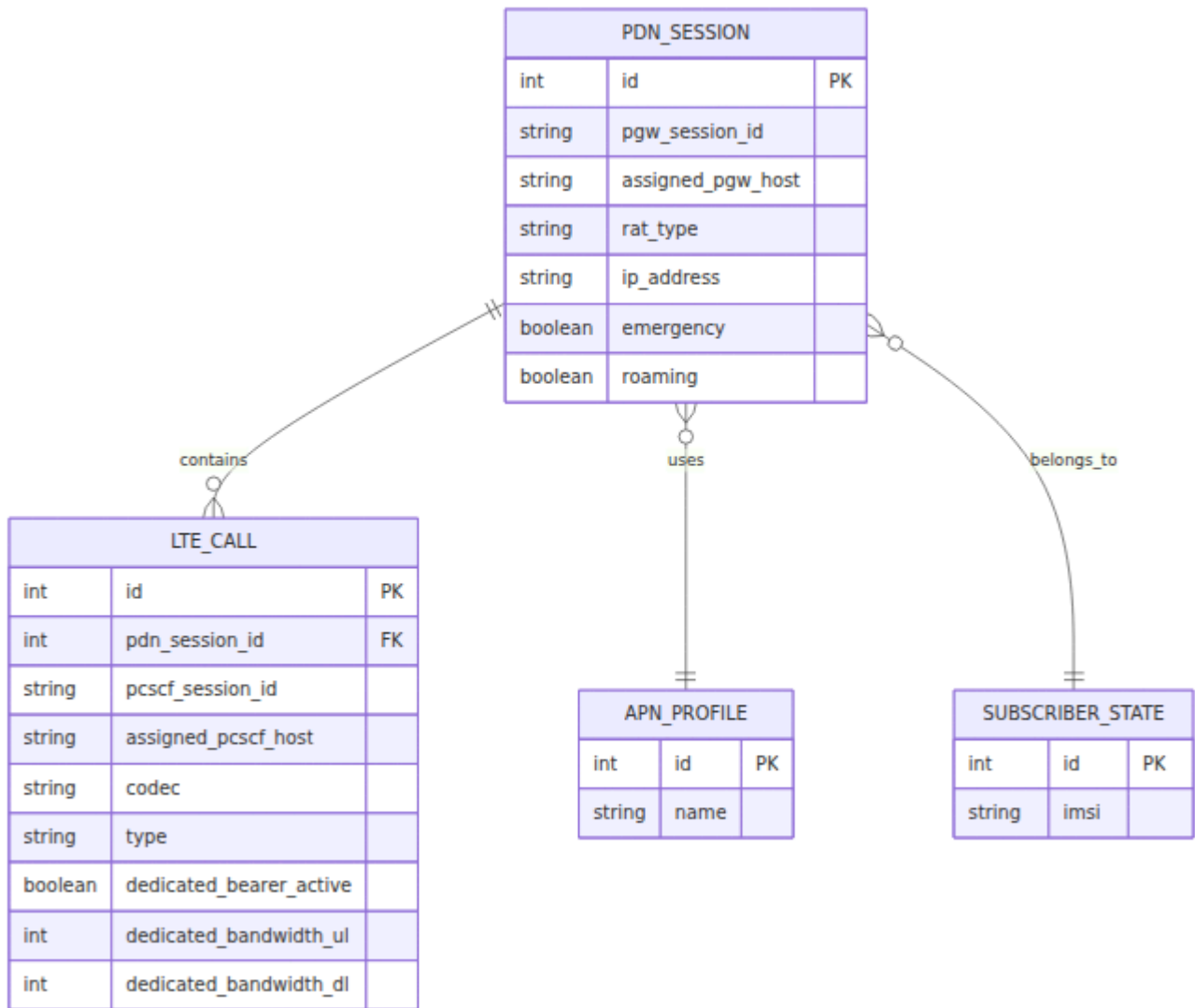
HSS PCRF PCRF LTE QoS

- **Gx** PGW/PCEF
- **Rx** IMS IP QoS
- RAR
- **VoLTE** QoS
- TFT
- **REST API**

Diameter

	ID		
Gx	16,777,238	PGW (PCEF)	PDN QoS
Rx	16,777,236	P-CSCF (AF)	IMS

PCRF PDN VoLTE



Gx

□□□□□

1. □□□□□ - □□□CCR-I□

□□□PGW □□□□□□□□ PDN □□

□□ **AVP**□

- Session-Id
- Origin-Host, Origin-Realm
- Subscription-Id□□□ IMSI□
- Called-Station-Id□APN □□□

- IP-CAN-Type IP
- RAT-Type
- Framed-IP-Address UE IP

PCRF

1. IMSI
2. APN QoS
- 3.
4. APN QoS

AVP

- Result-Code: 2001 DIAMETER_SUCCESS
- QoS-Information APN
- Default-EPS-Bearer-QoS QCI ARP
- Bearer-Control-Mode

2. CCR-U

PGW RAT

PCRF

1. ID
2. RAT
- 3.

Result-Code 2001

3. CCR-T

PGW PDN

PCRF

1. ID
- 2.
- 3.

Result-Code 2001

4. RAR

PCRF → PGW HSS

- IMS Rx AAR Gx RAR
- IMS Rx STR Gx RAR
- REST API

RAR AVP

- Session-Id PGW ID
- Auth-Application-Id: 16,777,238
- Re-Auth-Request-Type 0 =
- Charging-Rule-Install/Remove
- QoS-Information

PGW / /

PCRF TFT

- -
- -
- - QoS

- Gx RAR
-
- 5 / IP / TFT

- - Spotify WhatsApp Facebook

- **QoS** - **Quality of Service**
- **QoS** - **Quality of Service**
- **QoS** - **Quality of Service**
- **QoS** - **Quality of Service**
- **SLA** - **Service Level Agreement** QoS

QoS **QoS**

QoS APN **QoS**

```
{
  "QoS-Class-Identifider": 9,           // QCI9 = QoS
  "APN-Aggregate-Max-Bitrate-UL": 50000, // kbps
  "APN-Aggregate-Max-Bitrate-DL": 100000, // kbps
  "Allocation-Retention-Priority": {
    "Priority-Level": 8,
    "Pre-emption-Capability": 1,       // QoS
    "Pre-emption-Vulnerability": 1    // QoS
  }
}
```

QoS VoLTE

```
{
  "QoS-Class-Identifider": 1,         // QCI 1 = QoS
  "Max-Requested-Bandwidth-UL": 128000, // bps
  "Max-Requested-Bandwidth-DL": 128000, // bps
  "Guaranteed-Bitrate-UL": 128000,
  "Guaranteed-Bitrate-DL": 128000
}
```

Rx **QoS**

QoS

1. AA **AAR**/ **AA** **AAA**

IP-CSCF IMS VoLTE

AVP

- Session-Id IP-CSCF
- Subscription-Id IMS SIP URI
- Media-Component-Description
 - Media-Type
 - Max-Requested-Bandwidth-UL/DL
 - Codec-Data
 - Flow-Description 5
- AF-Application-Identifier

PCRF

1. IMSI SIP URI
2. IMS
- 3.
- 4.
5. Gx RAR PGW
6. Gx RAA
7. Rx AAA

AVP

- Result-Code: 2001 5063

2. STR/ STA

IP-CSCF IMS

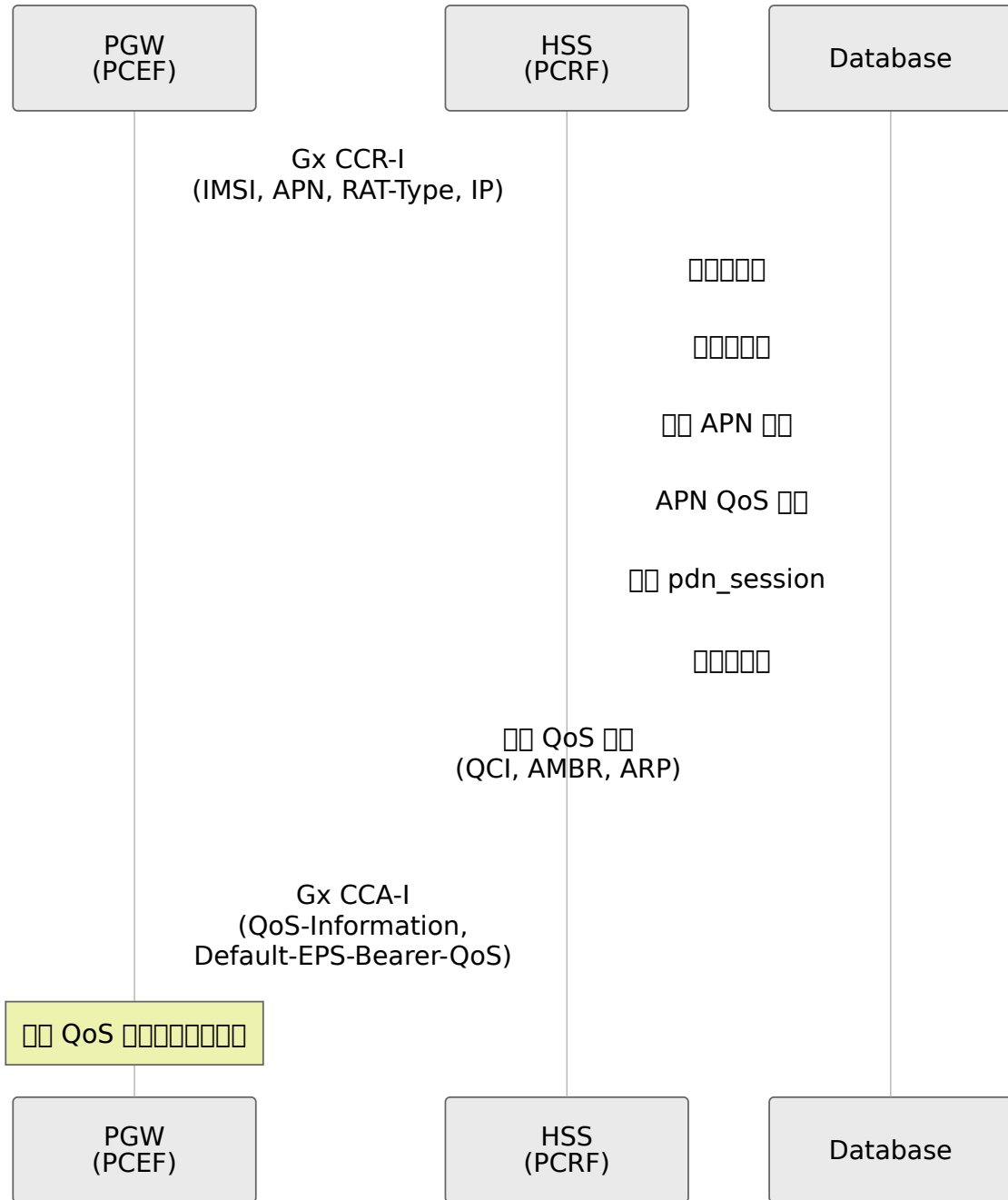
PCRF

1. P-CSCF ID
2. Gx RAR PGW
- 3.
4. STA

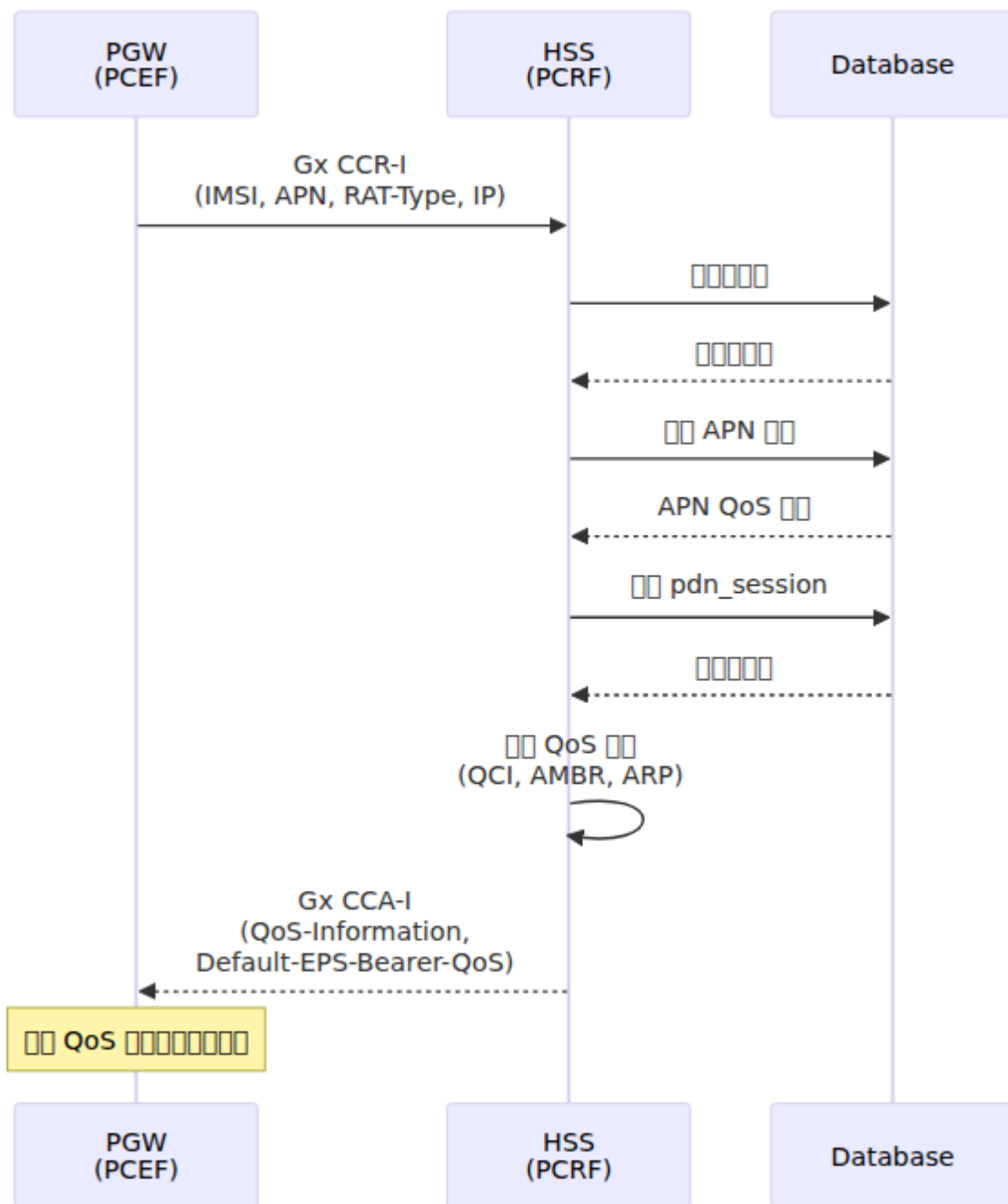
Result-Code 2001

□□□□□

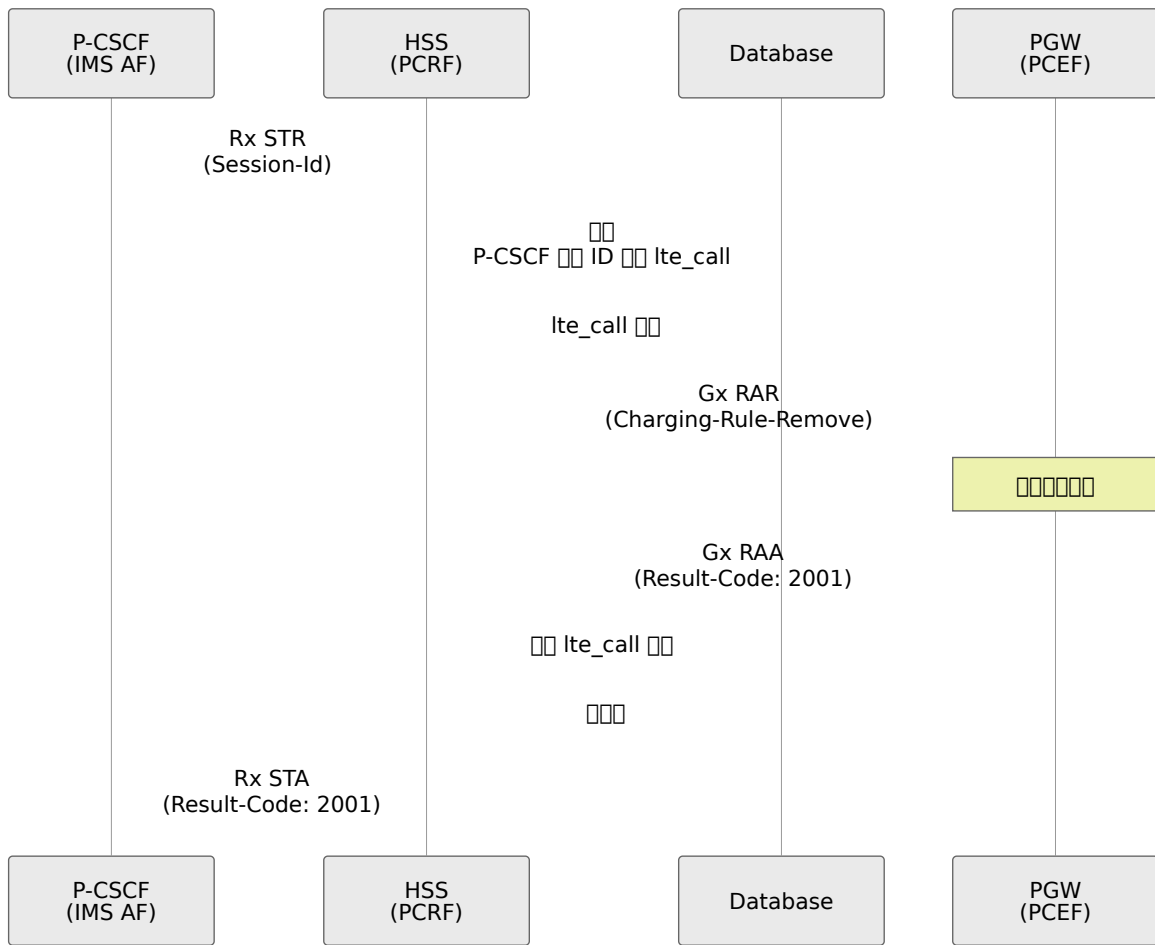
1 PDN □□□□



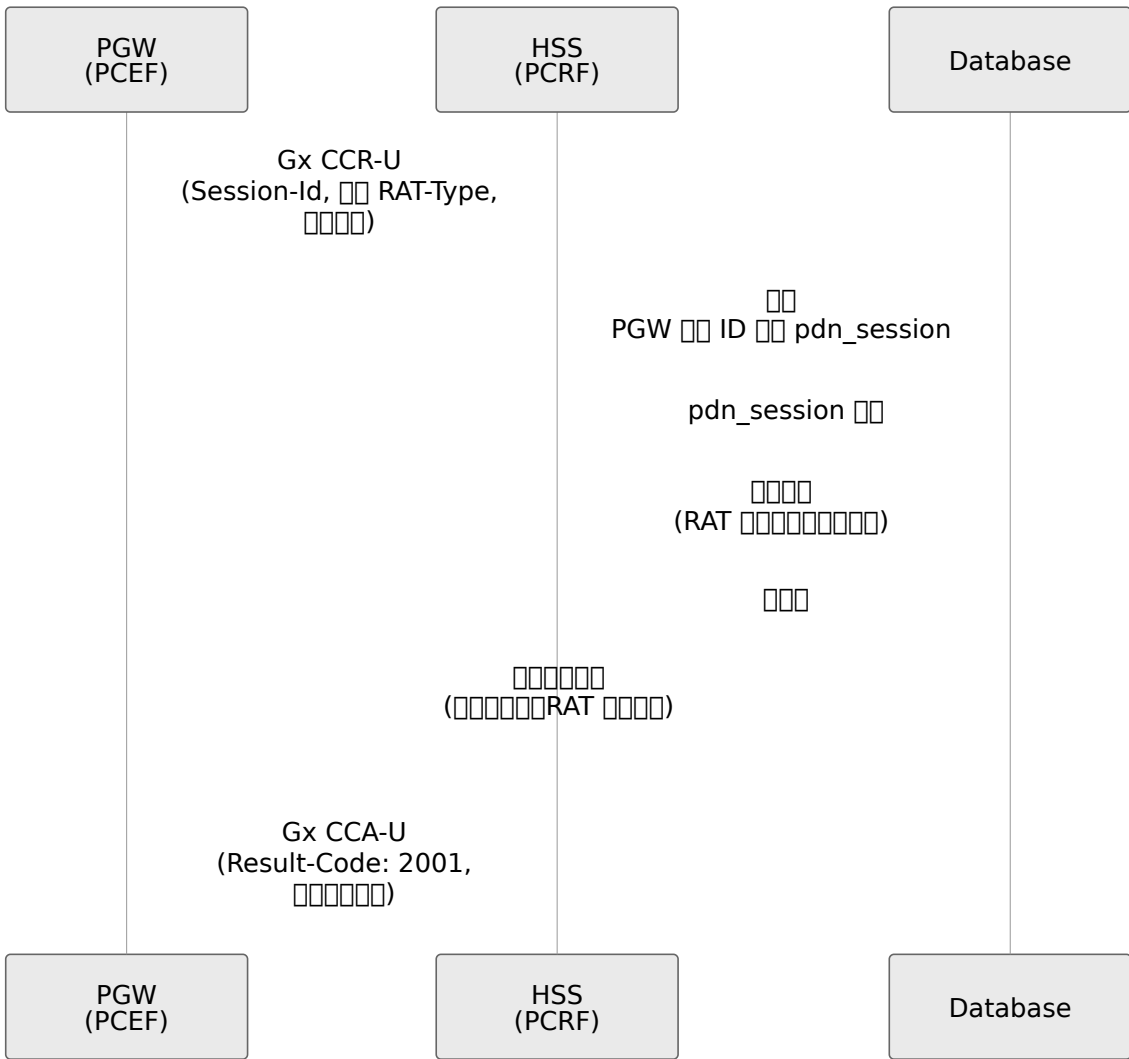
2 VoLTE Rx AAR → Gx RAR



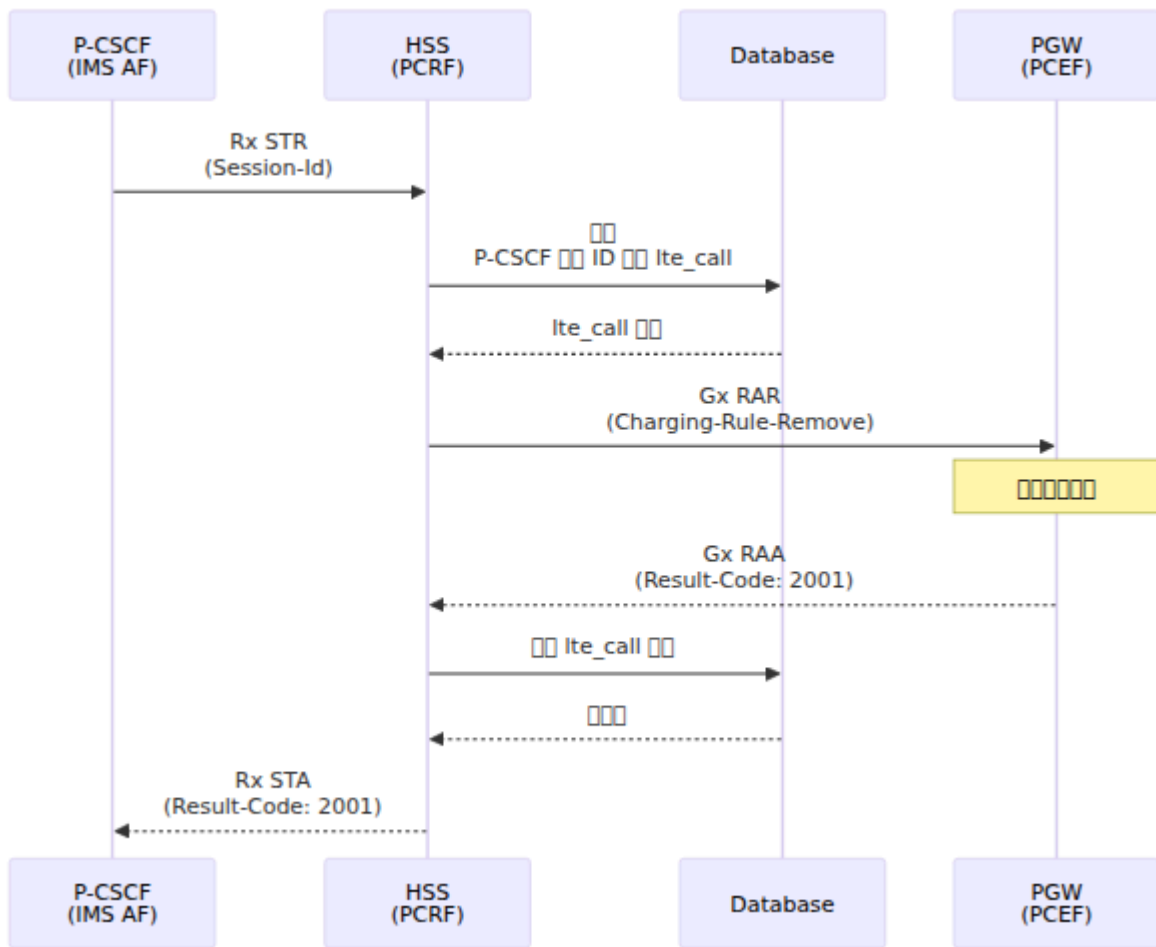
3 VoLTE Rx STR → Gx RAR



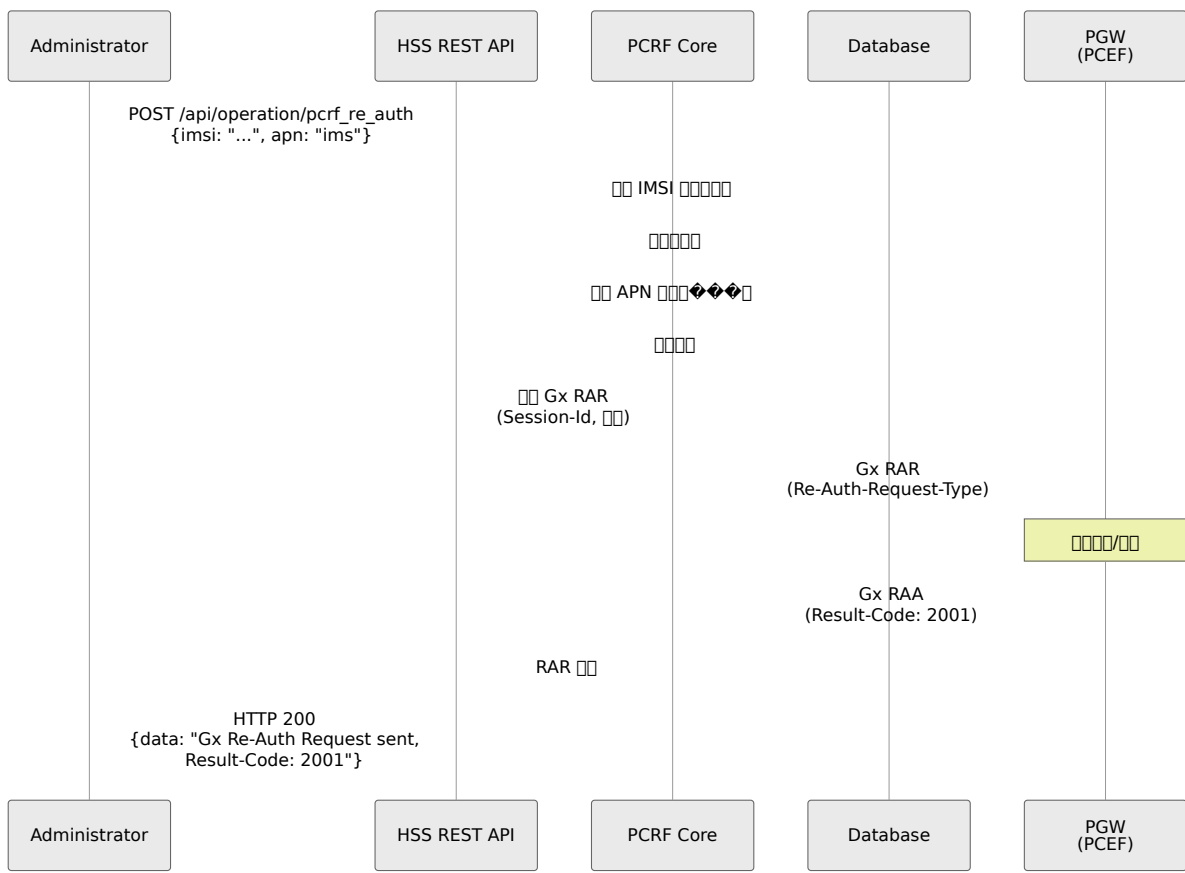
4G PDN



5 PDN



6 REST API



REST API

PCRF

POST /api/operation/pcrf_re_auth

Gx

APN QoS

```
{
  "imsi": "999999876543210",
  "apn": "ims"
}
```

HTTP 200

```
{
  "data": "Gx Re-Auth Request for 999999876543210 sent to
pgw.epc.mnc999.mcc999.3gppnetwork.org, Result-Code: 2001"
}
```

HTTP 400

```
{
  "error": "Unable to send Re-Auth Request for 999999876543210 on
APN ims, no active PDN Session found"
}
```

API

PCRF APN QoS REST API

APN QoS QCI APN PDN PGW Gx RAR

```
APN → APN QoS → APN
  ↓           ↓           ↓
  "internet"  QCI, AMBR, ARP  
```

1. APN

APN IP

POST /api/apn/identifier

Request

```
{
  "apn_identifier": {
    "apn": "internet",
    "ip_version": "ipv4v6"
  }
}
```

IP Options

- "ipv4" - IPv4
- "ipv6" - IPv6
- "ipv4v6" - IPv4 or IPv6
- "ipv4_or_ipv6" - IPv4 or IPv6

Response HTTP 201

```
{
  "data": {
    "id": 1,
    "apn": "internet",
    "ip_version": "ipv4v6"
  }
}
```

Notes

- apn 1-254
- ip_version

APN GET /api/apn/identifier

2. APN QoS

QoS QCI

POST /api/apn/qos_profile

{} {}

```
{
  "apn_qos_profile": {
    "name": "Best Effort Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

QoS {} {}

名前	型	範囲	説明
name	string	1-254 文字	名前
qci	integer	1-254	QoS 値 1-4 = GBR 5-9 = Non-GBR
allocation_retention_priority	integer	1-15	ARP 値 1 = 最高優先度
apn_ambr_dl_kbps	integer	1-4,294,967,293	APN 最大ダウンロード速度 (kbps)
apn_ambr_ul_kbps	integer	1-4,294,967,293	APN 最大アップロード速度 (kbps)
pre_emption_capability	boolean	true/false	優先度引き上げ能力
pre_emption_vulnerability	boolean	true/false	優先度引き下げ脆弱性

QCI 値

- 1 - VoLTE - GBR 100ms 遅延
- 2 - GBR 150ms 遅延
- 5 - IMS 音声 - Non-GBR 100ms 遅延
- 9 - 非 QoS 制御 - Non-GBR 300ms 遅延

HTTP 201 OK

```
{
  "data": {
    "id": 1,
    "name": "Best Effort Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

GET /api/apn/qos_profile

3. APN

APN QoS

POST /api/apn/profile

```
{
  "apn_profile": {
    "name": "Internet APN",
    "apn_identifier_id": 1,
    "apn_qos_profile_id": 1
  }
}
```

- name
- apn_identifier_id APN ID
- apn_qos_profile_id APN QoS ID

HTTP 201

```
{
  "data": {
    "id": 1,
    "name": "Internet APN",
    "apn_identifier_id": 1,
    "apn_qos_profile_id": 1
  }
}
```

□□□

- apn_identifier_id □ apn_qos_profile_id □□□□□□□□
- □□ APN □□□□ QoS □□□□□□□□□□

□□ **APN** □□□ GET /api/apn/profile

□□□□□□□□□□

□□ **1**□□□ **IMS APN** □□□ **VoLTE**□

```

# 1. APN
curl -X POST https://hss.example.com:8443/api/apn/identifier \
-H "Content-Type: application/json" \
-d '{
  "apn_identifier": {
    "apn": "ims",
    "ip_version": "ipv4v6"
  }
}'
# [{"data": {"id": 2, ...}}]

# 2. QoS IMS
curl -X POST https://hss.example.com:8443/api/apn/qos_profile \
-H "Content-Type: application/json" \
-d '{
  "apn_qos_profile": {
    "name": "IMS Signaling QoS",
    "qci": 5,
    "allocation_retention_priority": 2,
    "apn_ambr_dl_kbps": 5000,
    "apn_ambr_ul_kbps": 5000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}'
# [{"data": {"id": 2, ...}}]

# 3. APN
curl -X POST https://hss.example.com:8443/api/apn/profile \
-H "Content-Type: application/json" \
-d '{
  "apn_profile": {
    "name": "IMS APN",
    "apn_identifier_id": 2,
    "apn_qos_profile_id": 2
  }
}'
# [{"data": {"id": 2, ...}}]

```

2

APN EPC API APN

□□□□

- □□□□ APN □□□□ APN □□□□ QoS □□
- □□□□□□□□□□□□ APN □□

□□□□

□□□□□□□□ **100 Mbps** / □□ **50 Mbps**□□

```
{
  "apn_qos_profile": {
    "name": "High Speed Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

□□□□□□□□ **500 Mbps** / □□ **100 Mbps**□□

```
{
  "apn_qos_profile": {
    "name": "Premium Internet",
    "qci": 8,
    "allocation_retention_priority": 5,
    "apn_ambr_dl_kbps": 500000,
    "apn_ambr_ul_kbps": 100000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}
```

□□□/□□□□□□□□

```
{
  "apn_qos_profile": {
    "name": "IoT M2M",
    "qci": 9,
    "allocation_retention_priority": 10,
    "apn_ambr_dl_kbps": 1024,
    "apn_ambr_ul_kbps": 512,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

□□□□□□□□□□□□

```
{
  "apn_qos_profile": {
    "name": "Emergency APN",
    "qci": 5,
    "allocation_retention_priority": 1,
    "apn_ambr_dl_kbps": 10000,
    "apn_ambr_ul_kbps": 10000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}
```

□□

Diameter □□□□

Gx □□□ config/runtime.exs □□

```
%{
  application_name: :gx,
  application_dictionary: :diameter_gen_3gpp_gx,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_238}
  ]
}
```

Rx `config/runtime.exs`

```
%{
  application_name: :rx,
  application_dictionary: :diameter_gen_3gpp_rx,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_236}
  ]
}
```

QoS

QoS

- APN
 - `apn_qos_profile.qci` QoS
 - `apn_qos_profile.apn_ambr_ul_kbps`
 - `apn_qos_profile.apn_ambr_dl_kbps`
 - `apn_qos_profile.priority_level`
- Rx AAR
 - QCI 1
 - Max-Requested-Bandwidth AVP
 - Flow-Description AVP

□□□□

- Diameter □□ - □□□□□□
- API □□ - □□□ API □□
- □□ - □□ HSS □□
- □□□□ - □□□□ Diameter AVP □□□

API 错误

← API 错误

简介

- 错误类型
 - 错误原因
-

400 错误

400 错误

```
{  
  "error": "Invalid JSON format"  
}
```

原因

- JSON 格式错误
- 请求体损坏
- 请求体为空

404 错误

```
{  
  "error": "Resource not found"  
}
```

原因

- 000/0000/00000
- URL 00 ID 000

422 00000000

```
{  
  "errors": {  
    "imsi": ["has already been taken"],  
    "key_set_id": ["does not exist"]  
  }  
}
```

000

- 0000
- 000000000
- 00000000

500 00000000

```
{  
  "error": "Internal server error"  
}
```

000

- 00000000
 - 0000000000
-



API Request

Core OmniCore OmniCall OmniRAN OmniCharge Platform 文A □□□□ ▼
▼ 5GC ▼ ▼ ▼ ▼ ▼

Invalid JSON

400 Bad Request

Valid

Authorized?

No

401 Unauthorized

Yes

Resource Exists?

No

404 Not Found

Yes

Data Valid?

No

422 Validation Error

Yes

Process Request

Database OK?

Error

Success

500 Server Error

200/201 Success

← API →

API 教程

← API 教程

简介

- 什么是 API
 - API 的 IP 地址
-

安装 jq

jq 是一个轻量级的 JSON 处理器，可以在 Linux 和 macOS 上使用。

在 Linux 上使用 `apt-get install jq` 或在 macOS 上使用 `brew install jq` 安装。

安装后：

- 验证安装
- APN 配置
- EPC 配置
- 其他配置

```
# 1. 키셋 생성
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
-H "Content-Type: application/json" \
-d '{
  "ki": "0123456789ABCDEF0123456789ABCDEF",
  "opc": "FEDCBA9876543210FEDCBA9876543210",
  "authentication_algorithm": "milenage",
  "amf": "8000",
  "sqn": 0
}' | jq -r '.response.id')

# 2. APN QoS 프로파일 생성
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
-H "Content-Type: application/json" \
-d '{
  "name": "인터넷 QoS",
  "allocation_retention_priority": 8,
  "apn_ambr_dl_kbps": 50000,
  "apn_ambr_ul_kbps": 25000,
  "pre_emption_capability": true,
  "pre_emption_vulnerability": true,
  "qci": 9
}' | jq -r '.response.id')

# 3. APN 식별자 생성
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
-H "Content-Type: application/json" \
-d '{
  "apn": "internet",
  "ip_version": "ipv4v6"
}' | jq -r '.response.id')

# 4. APN 프로파일 생성
APN_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/profile \
-H "Content-Type: application/json" \
-d '{
  "apn_identifier_id": $APN_ID,
  "apn_qos_profile_id": $APN_QOS_ID,
  "name": "인터넷 APN"
}
```

```
}" | jq -r '.response.id')
```

```
# 5. 创建 EPC 配置
```

```
EPC_PROFILE_ID=$(curl -k -X POST  
https://hss.example.com:8443/api/epc/profile \  
-H "Content-Type: application/json" \  
-d "{  
  \"apn_profiles\": [\"$APN_PROFILE_ID\"],  
  \"name\": \"配置名称\",  
  \"network_access_mode\": \"packet_only\",  
  \"tracking_area_update_interval_seconds\": 600,  
  \"ue_ambr_dl_kbps\": 100000,  
  \"ue_ambr_ul_kbps\": 50000  
}" | jq -r '.response.id')
```

```
# 6. 创建用户
```

```
SUBSCRIBER_ID=$(curl -k -X POST  
https://hss.example.com:8443/api/subscriber \  
-H "Content-Type: application/json" \  
-d "{  
  \"imsi\": \"001001123456789\",  
  \"key_set_id\": $KEY_SET_ID,  
  \"epc_profile_id\": $EPC_PROFILE_ID  
}" | jq -r '.response.id')
```

```
echo "创建用户ID: $SUBSCRIBER_ID"
```

配置

配置参数

1. 配置 (配置) - 配置
2. 配置 (EPC 配置) - 配置
3. **APN** 配置 (APN 配置) - 配置 QoS 配置
4. 配置 (配置) - 配置

配置

- 配置 MSISDN 配置
- 配置 IMS 配置
- 配置 配置

- SIM SIM

- MSISDN -
- -

IP

IP

“” APN IPv4 IoT

```
# 安装 jq (apt-get install jq || brew install jq)

# 1. 创建密钥集
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
-H "Content-Type: application/json" \
-d '{
  "ki": "0123456789ABCDEF0123456789ABCDEF",
  "opc": "FEDCBA9876543210FEDCBA9876543210",
  "authentication_algorithm": "milenage",
  "amf": "8000",
  "sqn": 0
}' | jq -r '.response.id')

# 2. 创建 APN QoS 配置文件
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
-H "Content-Type: application/json" \
-d '{
  "name": "IoT 配置文件",
  "allocation_retention_priority": 8,
  "apn_ambr_dl_kbps": 10000,
  "apn_ambr_ul_kbps": 5000,
  "pre_emption_capability": false,
  "pre_emption_vulnerability": false,
  "qci": 9
}' | jq -r '.response.id')

# 3. 创建 APN 标识符
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
-H "Content-Type: application/json" \
-d '{
  "apn": "internet",
  "ip_version": "ipv4"
}' | jq -r '.response.id')

# 4. 创建 APN 配置文件
APN_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/profile \
-H "Content-Type: application/json" \
-d "{
  \"apn_identifier_id\": $APN_ID,
```

```
\ "apn_qos_profile_id\ ": $APN_QOS_ID,  
\ "name\ ": \ "IoT \ \ APN\  
}" | jq -r '.response.id')
```

5. \ APN \ \ IP

```
STATIC_IP_ID=$(curl -k -X POST  
https://hss.example.com:8443/api/epc/static_ip \  
-H "Content-Type: application/json" \  
-d "{  
  \ "apn_profile_id\ ": $APN_PROFILE_ID,  
  \ "ipv4_static_ip\ ": \ "100.64.1.100\  
}" | jq -r '.response.id')
```

6. \ EPC \ \

```
EPC_PROFILE_ID=$(curl -k -X POST  
https://hss.example.com:8443/api/epc/profile \  
-H "Content-Type: application/json" \  
-d "{  
  \ "apn_profiles\ ": [$APN_PROFILE_ID],  
  \ "name\ ": \ "IoT \ \ \  
  \ "network_access_mode\ ": \ "packet_only\  
  \ "tracking_area_update_interval_seconds\ ": 600,  
  \ "ue_ambr_dl_kbps\ ": 10000,  
  \ "ue_ambr_ul_kbps\ ": 5000  
}" | jq -r '.response.id')
```

7. \ MSISDN \ \ \ \

```
MSISDN_ID=$(curl -k -X POST  
https://hss.example.com:8443/api/msisdn \  
-H "Content-Type: application/json" \  
-d '{  
  "msisdn": "14155551000"  
}' | jq -r '.response.id')
```

8. \ \ \ \ IP \ \

```
SUBSCRIBER_ID=$(curl -k -X POST  
https://hss.example.com:8443/api/subscriber \  
-H "Content-Type: application/json" \  
-d "{  
  \ "imsi\ ": \ "001001999999999\  
  \ "key_set_id\ ": $KEY_SET_ID,  
  \ "epc_profile_id\ ": $EPC_PROFILE_ID,  
  \ "msisdns\ ": [$MSISDN_ID],  
  \ "static_ips\ ": [$STATIC_IP_ID]
```

```
} | jq -r '.response.id')
```

```
echo "IoT 物联网"  
echo "  ID: $SUBSCRIBER_ID"  
echo "  IMSI: 001001999999999"  
echo "  MSISDN: 14155551000"  
echo "  IPv4: 100.64.1.100  'internet' APN  "
```

物联网

物联网物联网物联网 IoT 物联网

1. 物联网 (物联网) - 物联网
2. **APN** 物联网 (APN 物联网) - “物联网”物联网
3. 物联网 IP 物联网 (物联网 IP) - 物联网 IPv4 物联网 100.64.1.100
4. 物联网物联网 (EPC 物联网) - 物联网 IoT 物联网物联网
5. 物联网 (MSISDN) - 物联网物联网
6. 物联网 (物联网) - 物联网物联网

物联网

物联网物联网物联网“物联网” APN 物联网物联网 IP 物联网 100.64.1.100 物联网 DHCP 物联网

物联网

- 物联网 APN 物联网 IP物联网 APN 物联网 2-5
- 物联网物联网物联网 IMS 物联网
- 物联网物联网物联网 物联网物联网
- 物联网 SIM物联网物联网 SIM

物联网

- 物联网 IP 物联网 - 物联网 IP 物联网
- 物联网物联网 - 物联网 IP 物联网
- 物联网 MSISDN 物联网 - 物联网物联网

← 物联网 API 物联网

OmniHSS API

←

- API
-
-
- MSISDN
- SIM
-
-
- IP
-
- EIR
-
-
- API

API

URL

https://[hostname]:8443/api

- **Content-Type:** application/json

- 例: 例 HTTPS
- 例: 8443 例

例: 例 API 例 "例" JSON 例

例:

```
{  
  "name": "value",  
  "field": "value"  
}
```

例:

```
{  
  "subscriber": {  
    "name": "value",  
    "field": "value"  
  }  
}
```

例:

```
# ✓ 例  
curl -X POST https://hss.example.com:8443/api/ims/profile \  
-H "Content-Type: application/json" \  
-d '{"name": "default", "ifc_template": "...}'  
  
# x 例  
curl -X POST https://hss.example.com:8443/api/ims/profile \  
-H "Content-Type: application/json" \  
-d '{"ims_profile": {"name": "default", "ifc_template": "...}}'
```

例

例 JSON 例

例:

```
{
  "status": "success",
  "response": { ... }
}
```

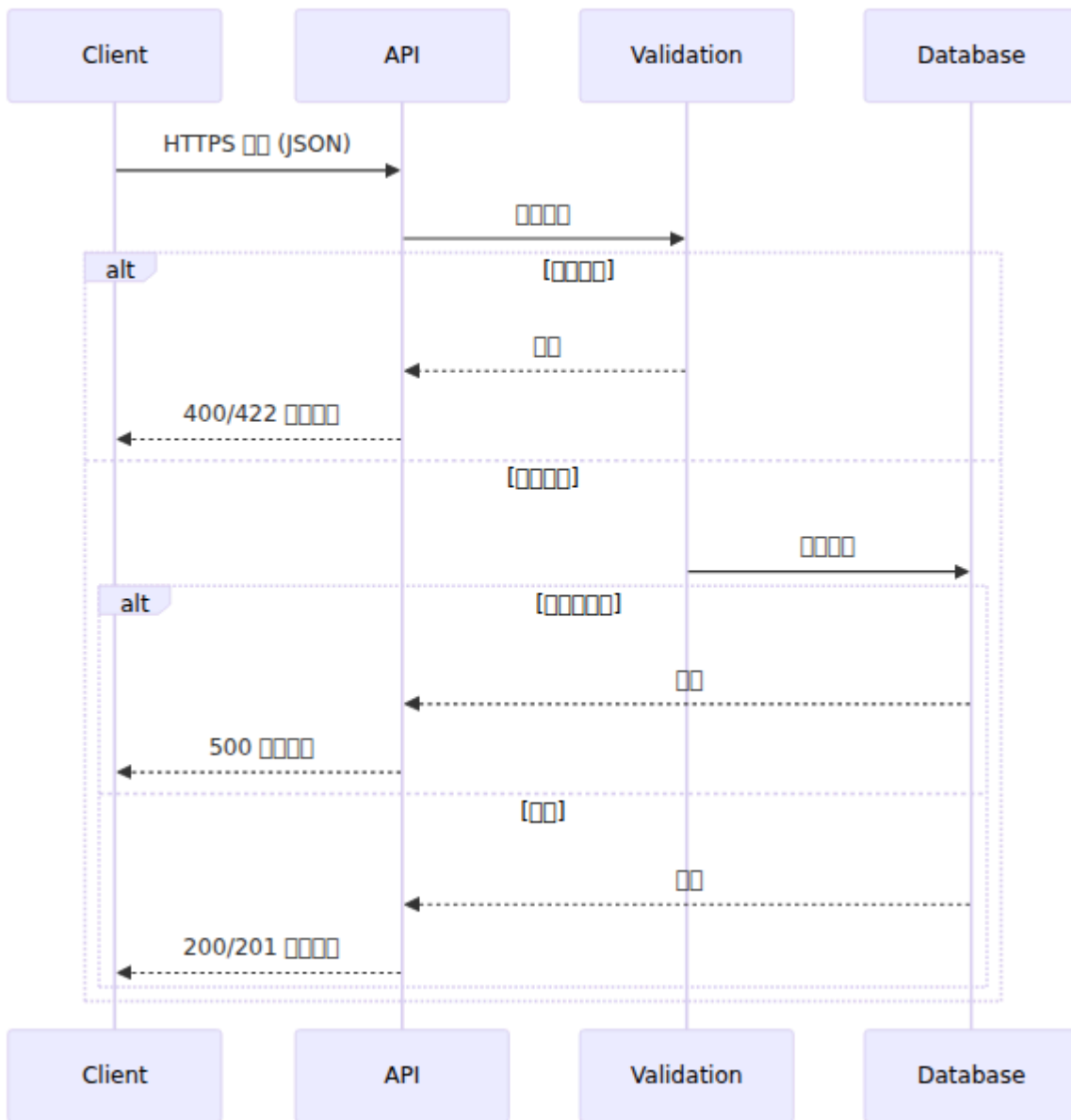
예외:

```
{
  "status": "error",
  "response": {
    "invalid_fields": {
      "field_name": "error message"
    }
  }
}
```

HTTP 코드

코드	상태	설명
200	OK	GET, PUT, DELETE
201	생성	POST
400	잘못된 요청	잘못된 요청
404	찾지 못함	잘못된 요청
422	잘못된 요청	잘못된 요청
500	서버 오류	서버 오류

API 交互



请求

响应

请求头

请求: GET /api/subscriber

响应:

名前	型	説明
enabled	boolean	有効/無効
ims_enabled	boolean	IMS 有効/無効

実行:

```
curl -k https://hss.example.com:8443/api/subscriber
```

返答:

```
{
  "data": [
    {
      "id": 1,
      "imsi": "001001123456789",
      "enabled": true,
      "ims_enabled": true,
      "sim_id": 1,
      "key_set_id": 1,
      "epc_profile_id": 1,
      "ims_profile_id": 1,
      "roaming_profile_id": 1,
      "custom_attributes": {},
      "inserted_at": "2025-10-15T10:30:00Z",
      "updated_at": "2025-10-15T10:30:00Z"
    }
  ]
}
```

名前 ID 説明

名前 ID 説明

リクエスト: GET /api/subscriber/:id

返答:

Field	Type	Description
id	integer	Subscriber ID

Request:

```
curl -k https://hss.example.com:8443/api/subscriber/1
```

GET IMSI

Request IMSI

Request: `GET /api/subscriber/imsi/:imsi`

Response:

Field	Type	Description	Length
imsi	string	Subscriber IMSI	14-15 digits

Request:

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/001001123456789
```

Request: `GET /api/subscriber/imsi/:imsi`

GET MSISDN

Request MSISDN

Request: `GET /api/subscriber/msisdn/:msisdn`

Response:

名前	型	説明	制約
msisdn	string	ISDN 番号	1-15 桁 (E.164)

curl:

```
curl -k
https://hss.example.com:8443/api/subscriber/msisdn/14155551234
```

レスポンス: 200 OK

ヘッダ:

Content-Type: application/json

メソッド: POST /api/subscriber

ボディ:

```
{
  "subscriber": {
    "imsi": "001001123456789",
    "enabled": true,
    "ims_enabled": true,
    "sim_id": 1,
    "key_set_id": 1,
    "epc_profile_id": 1,
    "ims_profile_id": 1,
    "roaming_profile_id": 1,
    "custom_attributes": {
      "note": "test"
    }
  }
}
```

パラメータ:

- imsi - 14-15 桁の IMSI

- `key_set_id` - 唯一标识符 ID
- `epc_profile_id` - 唯一标识符 EPC 配置

请求体:

- `enabled` - 是否启用: true
- `ims_enabled` - 是否启用 IMS: true
- `sim_id` - SIM 卡 ID
- `ims_profile_id` - IMS 配置 ID
- `roaming_profile_id` - 漫游配置 ID
- `msisdns` - MSISDN ID
- `static_ips` - IP 地址 APN 列表
- `custom_attributes` - 自定义属性

响应:

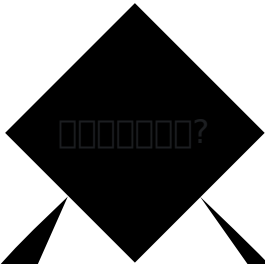
- 返回 ID
- `MSISDN` 列表
- `IP` 列表 - IP 地址 APN

请求:

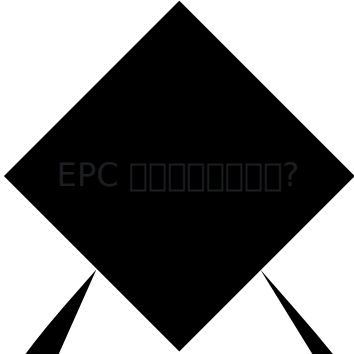
```
curl -k -X POST https://hss.example.com:8443/api/subscriber \
-H "Content-Type: application/json" \
-d '{
  "subscriber": {
    "imsi": "001001123456789",
    "key_set_id": 1,
    "epc_profile_id": 1
  }
}'
```

响应:

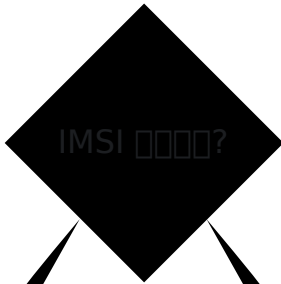
□□□□



□□: □□□□□□



□□: □□□ EPC □□□□



□□: IMSI □□□

□□□□

□□□□□□□□

0000

00000000

00: PUT /api/subscriber/:id

0000:

00	00	00
id	integer	000000 ID

000:

```
{
  "subscriber": {
    "enabled": false,
    "ims_enabled": false,
    "epc_profile_id": 2,
    "custom_attributes": {
      "note": "0000"
    }
  }
}
```

00000:

- enabled - 00/00000000
- ims_enabled - 00/00 IMS 00
- sim_id - 00 SIM 0 00
- key_set_id - 00 0000000000
- epc_profile_id - 00 0000000000
- ims_profile_id - 00 0000000000
- roaming_profile_id - 00 0000

- `msisdns` - 電話番号 0000
- `static_ips` - APN 0 IP
- `custom_attributes` - 属性

headers:

- `imsi` - IMSI

body:

- 0000 - 属性

curl:

```
curl -k -X PUT https://hss.example.com:8443/api/subscriber/1 \
-H "Content-Type: application/json" \
-d '{
  "subscriber": {
    "enabled": false
  }
}'
```

headers:

- `headers`: {"enabled": false}
- `headers`: {"ims_enabled": false}
- `headers`: {"epc_profile_id": 2} (EPC 0000)
- `headers`: {"roaming_profile_id": 3} (0000)

headers

headers

curl: DELETE /api/subscriber/:id

headers:

Field	Type	Description
id	integer	Subscriber ID

Request:

```
curl -k -X DELETE https://hss.example.com:8443/api/subscriber/1
```

Request Body: PDN IMSI

Request Body:

- IMSI - IMSI
- SIM - SIM
- MSISDN - MSISDN
- MSISDNs - MSISDNs

Request Body:

Request Body: CLR MME

Request: POST /api/subscriber/cancel_location

Request Body:

```
{
  "imsi": "001001123456789"
}
```

Request Body:

Field	Type	Length	Description
imsi	string	14-15	IMSI 14-15

Request Body:

```
curl -k -X POST
https://hss.example.com:8443/api/subscriber/cancel_location \
-H "Content-Type: application/json" \
-d '{"imsi": "001001123456789"}'
```

成功 (200 OK):

```
{
  "data": {
    "message": "成功",
    "imsi": "001001123456789",
    "destination_host": "mme01.operator.com",
    "destination_realm": "epc.operator.com"
  }
}
```

失敗 (404 未見):

```
{
  "error": "MME 未見"
}
```

原因:

- MME が S6a CLR (subscriber_state.last_seen_mme)
- Cancellation-Type: subscription_withdrawal
- CLR-Flags: {s6a_indicator: 1, reattach_required: 1} UE
- last_seen_mme が null 404
- IMSI MSISDNs/SIM

解決:

- 確認: 設定
- 確認: 設定
- 確認: MME
- 確認: 設定

- IMSI: 001001123456789

IMSI CLR:

CLR MSISDN

1. IMSI CLR:

```
// IMSI 001001123456789 MSISDNs ["+1234567890", "+9876543210"]
POST /api/subscriber/cancel_location
{"imsi": "001001123456789"}

// IMSI: 001001123456789 CLR MSISDN
```

2. IMSIs:

```
// MSISDN IMSIs
// A: IMSI 001001111111111, MSISDN "+1234567890"
// B: IMSI 001001222222222, MSISDN "+1234567890"

POST /api/subscriber/cancel_location
{"imsi": "001001111111111"}

// IMSI A B
```

Flow:

- IMSI CLR IMSI MSISDN
- IMSI CLR MME
- MME: MME CLR HSS
- IMSI

Flow:

- IMSI
- IMSI
- S6a

MSISDN

MSISDN [MSISDN](#)

MSISDNs

GET /api/msisdn

```
curl -k https://hss.example.com:8443/api/msisdn
```

MSISDN

GET /api/msisdn/:id

```
curl -k https://hss.example.com:8443/api/msisdn/1
```

MSISDN

POST /api/msisdn

```
{
  "msisdn": {
    "msisdn": "14155551234"
  }
}
```

注意:

- 号码 1-15 位数
- 国际号码
- 国际号码 E.164 国际号码 + 号码

示例:

```
curl -k -X POST https://hss.example.com:8443/api/msisdn \
-H "Content-Type: application/json" \
-d '{
  "msisdn": {
    "msisdn": "14155551234"
  }
}'
```

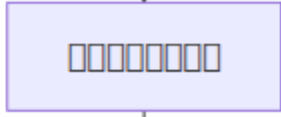
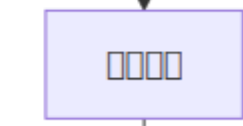
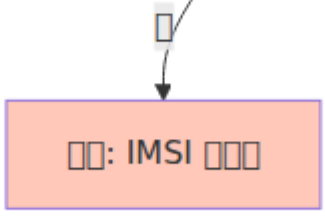
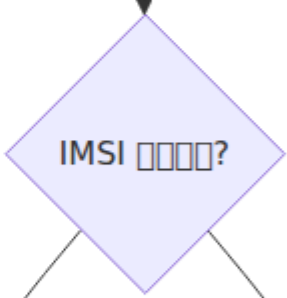
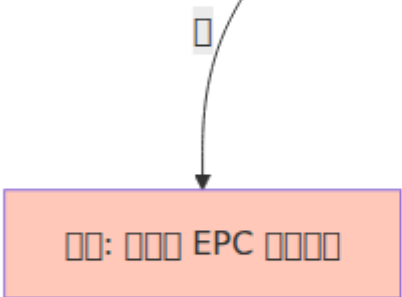
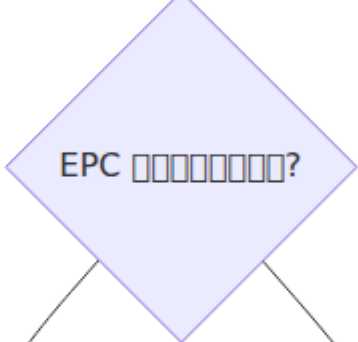
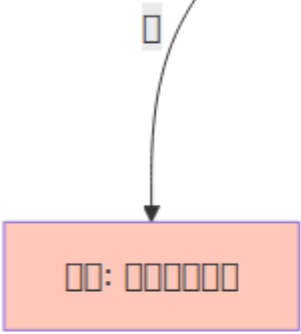
MSISDN 号码

国际号码 E.164 国际号码 + 号码

MSISDN 格式:



ore OmniCore OmniCall OmniRAN OmniCharge Platform
▼ 5GC ▼ ▼ ▼ ▼ ▼



↓
201 OK

MSISDN IMSI

MSISDN

DELETE /api/msisdn/:id

```
curl -k -X DELETE https://hss.example.com:8443/api/msisdn/1
```

SIM

SIM ICCID PIN/PUK OTA SIM

- IMSI - SIM

SIMs

SIM

GET /api/sim

```
curl -k https://hss.example.com:8443/api/sim
```

GET /api/sim/:id

Retrieve SIM information

Request: GET /api/sim/:id

Response:

```
curl -k https://hss.example.com:8443/api/sim/1
```

POST /api/sim

Create SIM information

Request: POST /api/sim

Response:

```
{
  "sim": {
    "iccid": "8991101200003204510",
    "sim_vendor": "Gemalto",
    "batch_name": "2025-Q1-Batch-01",
    "is_esim": false,
    "pin1": "1234",
    "pin2": "5678",
    "puk1": "12345678",
    "puk2": "87654321",
    "adm1": "admin-code-1",
    "kic": "0123456789ABCDEF0123456789ABCDEF",
    "kid": "FEDCBA9876543210FEDCBA9876543210"
  }
}
```

Response:

- `iccid` - 19-20 digit identifier

Response:

- `sim_vendor` - 运营商
- `batch_name` - 批次
- `is_esim` - eSIM 标识
- `pin1`, `pin2` - 运营商 PIN 码
- `puk1`, `puk2` - PIN 解锁码
- `adm1-adm10` - 运营商参数
- `kic`, `kid` - OTA 密钥

请求:

```
curl -k -X POST https://hss.example.com:8443/api/sim \
-H "Content-Type: application/json" \
-d '{
  "sim": {
    "iccid": "8991101200003204510",
    "sim_vendor": "Gemalto"
  }
}'
```

更新 SIM

更新 SIM 信息

请求: `PUT /api/sim/:id`

请求:

```
curl -k -X PUT https://hss.example.com:8443/api/sim/1 \
-H "Content-Type: application/json" \
-d '{
  "sim": {
    "batch_name": "XXXXXXXXXX"
  }
}'
```

API SIM

API SIM API

API: DELETE /api/sim/:id

API: API API SIM

API

API Milenage API Ki OPC/OP AMF SQN API API

API:

- API - API

API

API

API: GET /api/key_set

API:

```
curl -k https://hss.example.com:8443/api/key_set
```

API

API

API: GET /api/key_set/:id

API:

```
curl -k https://hss.example.com:8443/api/key_set/1
```

□□□□:

```
{
  "data": {
    "id": 1,
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "op": null,
    "amf": "8000",
    "sqn": 0,
    "authentication_algorithm": "milenage",
    "ota_counter": 0
  }
}
```

□□□□□

□□□□□□□□□□□□

□□: POST /api/key_set

□□□:

```
{
  "key_set": {
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "amf": "8000",
    "sqn": 0,
    "authentication_algorithm": "milenage"
  }
}
```

□□□□:

- **ki** - 128 □□□□32 □□□□□□□□

- `opc` - `opc` OPC `opc` OP `opc`
- `authentication_algorithm` - `authentication_algorithm` "milenage"

Request:

- `amf` - `amf`: "8000"
- `sqn` - `sqn`: 0
- `ota_counter` - `ota_counter`: 0

Response:

- `authentication_algorithm`
- `ki` OPC OP: 32 `authentication_algorithm` 128 `authentication_algorithm`
- `AMF`: 4 `authentication_algorithm` 16 `authentication_algorithm`

Example:

```
curl -k -X POST https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "key_set": {
      "ki": "0123456789ABCDEF0123456789ABCDEF",
      "opc": "FEDCBA9876543210FEDCBA9876543210",
      "authentication_algorithm": "milenage"
    }
  }'
```

Request: `authentication_algorithm` API `authentication_algorithm`

Request:

Request:

Request: `PUT /api/key_set/:id`

Request: `authentication_algorithm` `authentication_algorithm`

Request: `authentication_algorithm` `authentication_algorithm`

□□□□□

□□□□□□□□

□□: DELETE /api/key_set/:id

□□: □□□□□□□□ □□ □□□□□□□□□□□□□□□□□□

□□□□□

EPC □□□□

EPC□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ □□□□□

□□ **EPC** □□□□

□□: GET /api/epc/profile

□□ **EPC** □□□□

□□: GET /api/epc/profile/:id

□□ **EPC** □□□□

□□: POST /api/epc/profile

□□□:

```
{
  "apn_profiles": [],
  "name": "□□□□□□",
  "network_access_mode": "□□□□",
  "tracking_area_update_interval_seconds": 600,
  "ue_ambr_dl_kbps": 100000,
  "ue_ambr_ul_kbps": 50000
}
```

□□:

필드명	타입	단위	범위
name	문자열	없음	문자열
ue_ambr_dl_kbps	정수	Kbps	10000-1000000
ue_ambr_ul_kbps	정수	Kbps	5000-500000
network_access_mode	문자열	없음	"LTE" 또는 "LTE-M"
tracking_area_update_interval_seconds	정수	없음	600~1800
apn_profiles	리스트	없음	[] 또는 [1, 2, 3]

예시:

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
-H "Content-Type: application/json" \
-d '{
  "apn_profiles": [],
  "name": "LTE 100Mbps",
  "network_access_mode": "LTE",
  "tracking_area_update_interval_seconds": 600,
  "ue_ambr_dl_kbps": 100000,
  "ue_ambr_ul_kbps": 50000
}'
```

참고:

- [UE AMBR](#) - 사용자 장비 최대 속도
- [EPC](#) - EPC 프로파일

GET EPC Profile

Request: GET /api/epc/profile/:id

Response: EPC profile information

PUT EPC Profile

Request: PUT /api/epc/profile/:id

Response: Updated EPC profile information

IMS Profile

IMS IP address and IFC information

GET IMS Profile

Request: GET /api/ims/profile

POST IMS Profile

Request: POST /api/ims/profile

Response:

```
{
  "name": "IMS VoLTE",
  "ifc_template": "<IMS-XML-template>"
}
```

Response:

- name - IMS profile name
- ifc_template - IFC XML template Liquid

IFC Profile:

IFC profile Liquid

変数	説明	値
<code>{{ imsi }}</code>	IMSI	001001123456789
<code>{{ msisdns }}</code>	MSISDN 番号リスト	["14155551234", "14155555678"]
<code>{{ mcc }}</code>	移動体通信ネットワークの国コード	001
<code>{{ mnc }}</code>	移動体通信ネットワークのオペレータコード	001

例:

IFC を使用して **Liquid** テンプレートエンジンで Jinja2 形式の IMS 番号を生成する

1. 変数: IMS 番号生成テンプレートに `{{ imsi }}` と `{% for msisdn in msisdns %}`
2. 変数: API 呼び出しに XML 形式
3. テンプレート: IMS 番号 MAA/SAA から HSS
 - IMS 番号
 - テンプレート変数
 - `{{ imsi }}` → IMSI
 - `{{ msisdns }}` → MSISDN
 - `{{ mcc }}` → MCC
 - `{{ mnc }}` → MNC
 - Cx/Diameter 形式 XML から S-CSCF

例:

```
<!-- 電話番号 -->
{{ imsi }}

<!-- 電話番号 -->
{% for msisdn in msisdns %}
  <MSISDN>{{ msisdn }}</MSISDN>
{% endfor %}

<!-- 電話番号 -->
{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc }}.3gppnetwork.org
```

IFC 電話番号:

```

<?xml version="1.0" encoding="UTF-8"?>
<IMSSubscription>
<PrivateID>{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</PrivateID>
<ServiceProfile>
{% for msisdn in msisdns %}
<PublicIdentity>
<Identity>sip:{{ msisdn }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</Identity>
<Extension>
<IdentityType>0</IdentityType>
</Extension>
</PublicIdentity>
<PublicIdentity>
<Identity>tel:{{ msisdn }}</Identity>
<Extension>
<IdentityType>0</IdentityType>
</Extension>
</PublicIdentity>
{% endfor %}
<InitialFilterCriteria>
<Priority>10</Priority>
<TriggerPoint>
<ConditionTypeCNF>0</ConditionTypeCNF>
<SPT>
<ConditionNegated>0</ConditionNegated>
<Group>0</Group>
<Method>REGISTER</Method>
</SPT>
</TriggerPoint>
<ApplicationServer>
<ServerName>sip:as.ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</ServerName>
<DefaultHandling>0</DefaultHandling>
</ApplicationServer>
</InitialFilterCriteria>
</ServiceProfile>
</IMSSubscription>

```

□□□□ (curl):

```
curl -k -X POST https://hss.example.com:8443/api/ims/profile \
-H "Content-Type: application/json" \
-d '{
  "name": "default",
  "ifc_template": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<IMSSubscription><ServiceProfile>...</ServiceProfile>
</IMSSubscription>"
}'
```

Python:

```
import requests

response = requests.post(
    "https://hss.example.com:8443/api/ims/profile",
    json={
        "name": "default",
        "ifc_template": ifc_template_string
    },
    verify=False # skip SSL
)
```

Response (201 OK):

```
{
  "status": "success",
  "response": {
    "id": 1,
    "name": "default",
    "ifc_template": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>...\"
  }
}
```

Notes:

- API uses IFC to return XML
- ...
- name ...

API:

- **API** - IFC **API**
- **API** - IMS **API**
- **API IFC API** - **API**

APN API

APN API

1. **APN API** - API APN API IP API
2. **APN QoS API** - **API**
3. **APN API** - **API** QoS **API** **EPC API**

API **QoS** **API** **PCRF API** **API** **API** **API** **API** **API**

API **APN** API

API: GET /api/apn/identifier

API **APN** API

API: POST /api/apn/identifier

API:

```
{
  "apn": "internet",
  "ip_version": "ipv4v6"
}
```

IP API:

- "ipv4" - IPv4
- "ipv6" - IPv6
- "ipv4v6" - IPv4v6 **API**
- "ipv4_or_ipv6" - IPv4 **API** IPv6 **API**

API APN QoS

Request: GET /api/apn/qos_profile

API APN QoS

Request: POST /api/apn/qos_profile

Response:

```
{
  "name": "APN QoS",
  "allocation_retention_priority": 8,
  "apn_ambr_dl_kbps": 50000,
  "apn_ambr_ul_kbps": 25000,
  "pre_emption_capability": false,
  "pre_emption_vulnerability": true,
  "qci": 9
}
```

API APN

Request: GET /api/apn/profile

API APN

Request: POST /api/apn/profile

Response:

```
{
  "apn_identifier_id": 1,
  "apn_qos_profile_id": 1,
  "name": "APN"
}
```

Response:

- `apn_identifier_id` - APN ID
- `apn_qos_profile_id` - APN QoS ID

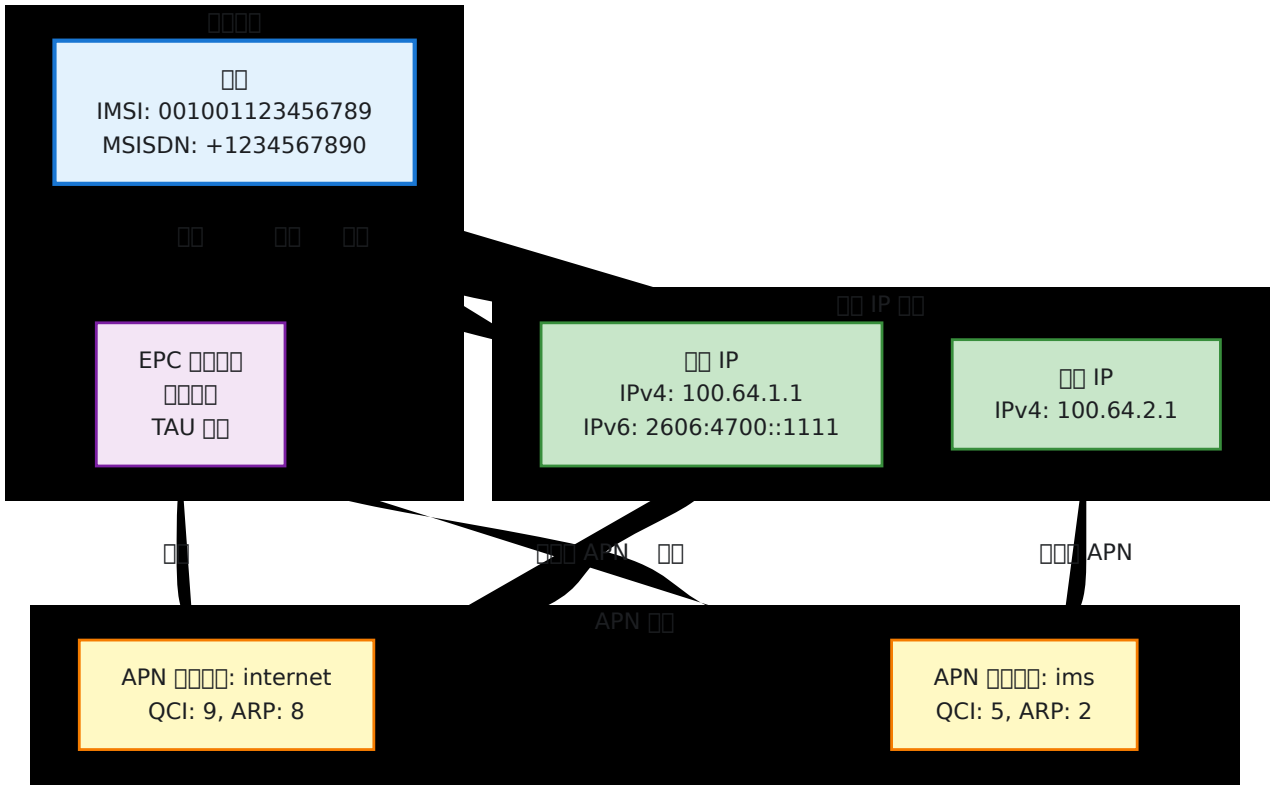
☐☐:

- ☐☐☐☐☐ - ☐☐ APN ☐☐☐☐☐☐
- EPC ☐☐☐☐☐ - APN ☐☐☐☐☐☐ EPC ☐☐☐☐

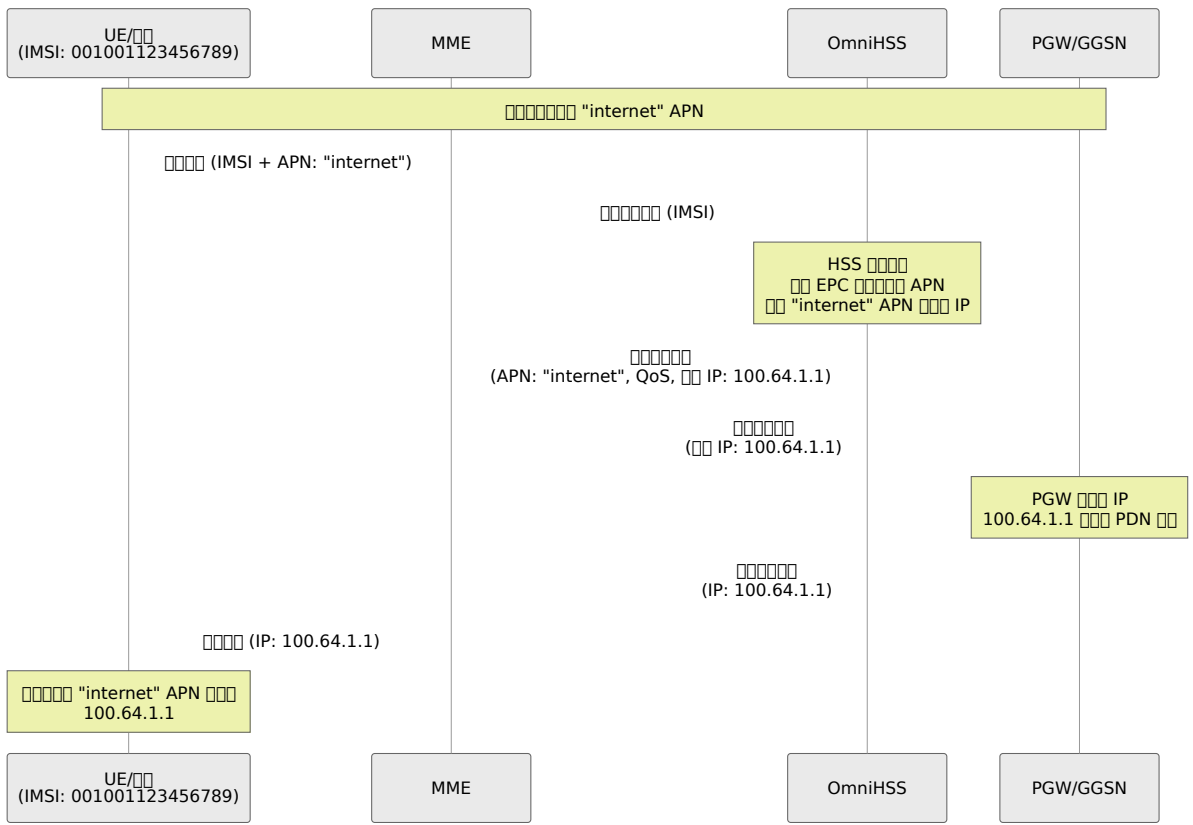
☐☐ IP ☐☐

☐☐ IP ☐☐☐☐☐☐☐☐ APN☐☐☐☐☐☐☐☐☐☐☐☐ APN ☐☐☐☐☐☐ IPv4 ☐/☐ IPv6 ☐☐☐☐☐☐☐ DHCP ☐☐☐☐☐☐☐☐☐☐

☐☐:

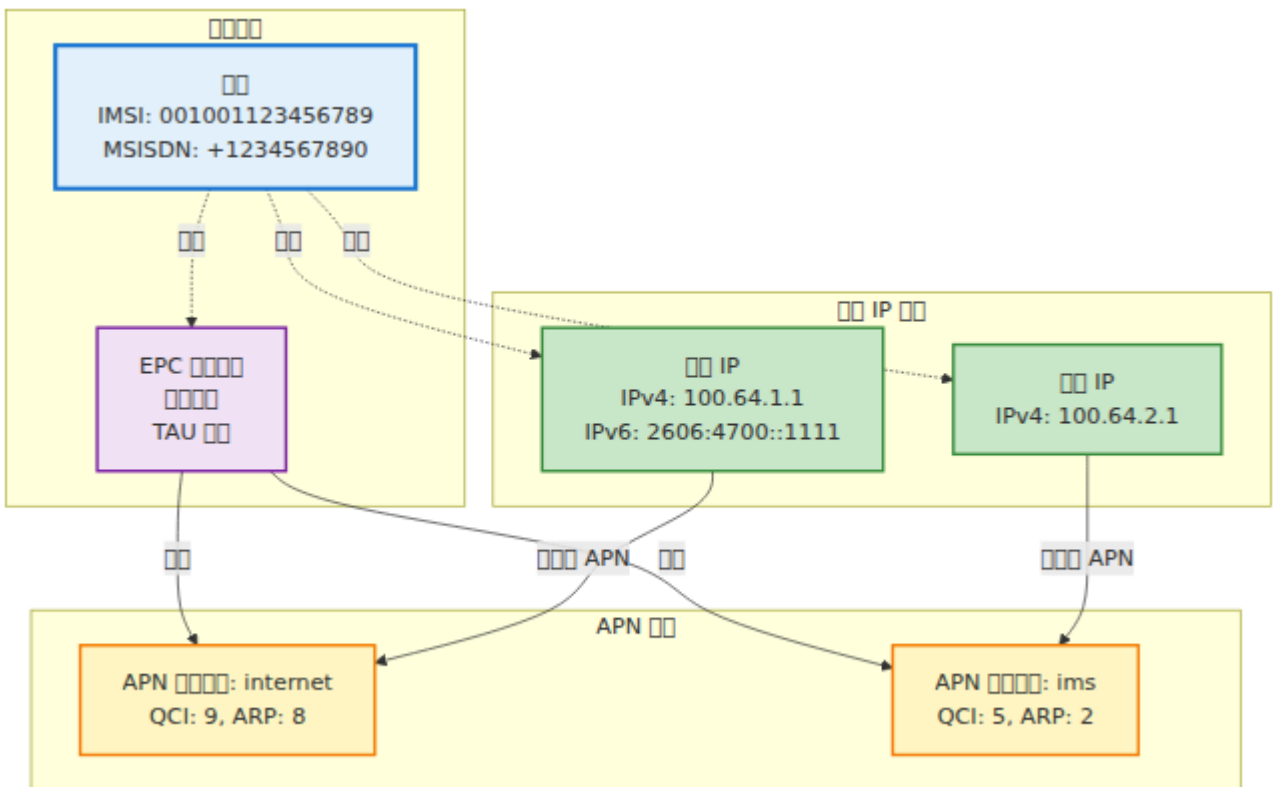


☐☐☐☐☐☐☐☐☐☐:



APN - APN parameters:

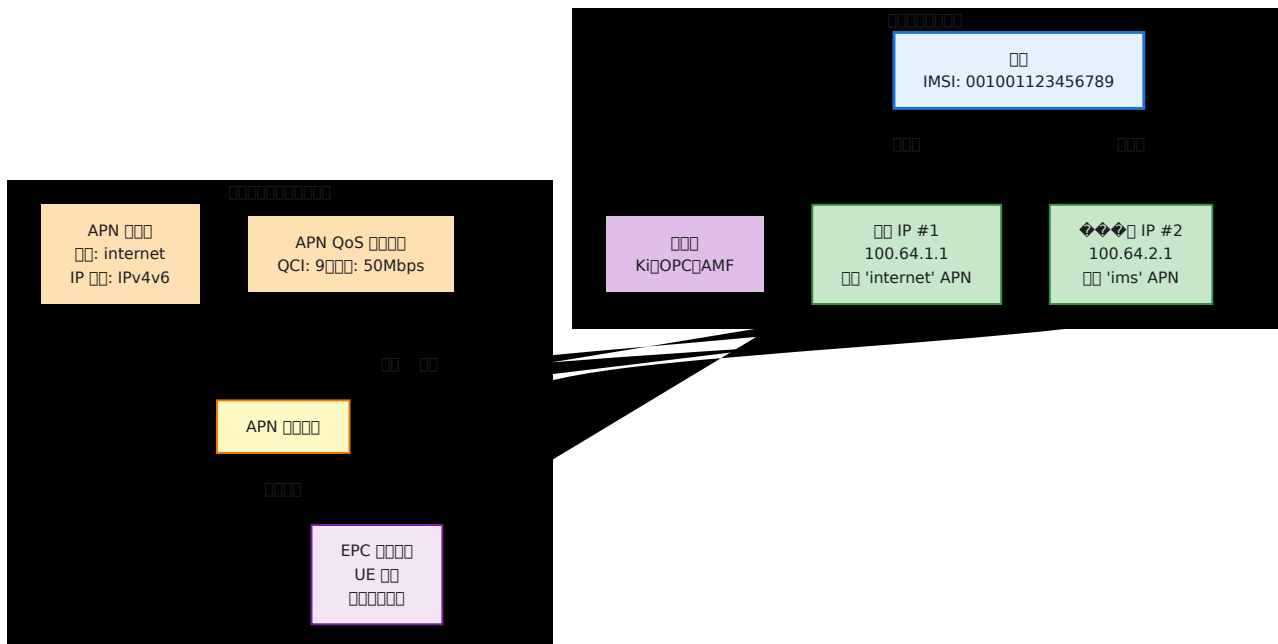
S6a APN- AVP parameters



Parameters:

1. **APN ID:** APN ID 00102...
2. **APN Name:** `apn_identifier.apn` "internet" "ims"
3. **PDN Type:** `apn_identifier.ip_version` ipv4=0 ipv6=1 ipv4v6=2 ipv4_or_ipv6=3
4. **QoS Profile:** `apn_qos_profile`
5. **AMBR:** 1000 kbps → bps
6. **Static IP:** subscriber.static_ips → `apn_profile_id` → IP
 - `subscriber.static_ips` → `apn_profile_id` → IP
 - IP version `apn_identifier.ip_version`
7. **VPLMN-APN:** 0 - IP

Diagram:



Notes:

- **APN Name:** IP version **APN Name**
- **APN Name IP:** APN Name IP
- **IPv4 & IPv6:** IP version IPv4 IPv6
- **Static IP:** IP version IP
 - IPv4 & IPv6 IP version APN Name
 - IP version
 - `ipv4_static_ip` `ipv6_static_ip`
- **Static IP:** IP version

📄:

- IoT 📡 IP 📡
- 📡📡📡📡📡📡📡📡 IP 📡📡📡📡
- 📡📡 IP 📡📡📡📡📡
- 📡📡 IP 📡📡📡📡
- 📡📡 IP 📡📡📡📡

📡📡📡 IPs

📡📡📡📡 IP 📡📡

📄: GET /api/epc/static_ip

📡📡📡:

```
curl -k https://hss.example.com:8443/api/epc/static_ip
```

📡📡📡:

```
{
  "data": [
    {
      "id": 1,
      "apn_profile_id": 5,
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111",
      "apn_profile": {
        "id": 5,
        "name": "☐☐☐ APN",
        "apn_identifier": {
          "apn": "internet",
          "ip_version": "ipv4v6"
        }
      },
      "inserted_at": "2025-11-15T10:30:00Z",
      "updated_at": "2025-11-15T10:30:00Z"
    }
  ]
}
```

☐☐☐☐ IP

☐☐☐☐☐☐ IP ☐☐☐

☐☐: GET /api/epc/static_ip/:id

☐☐☐☐:

☐☐	☐☐	☐☐
id	integer	☐☐ IP ☐☐☐ ID

☐☐☐☐:

```
curl -k https://hss.example.com:8443/api/epc/static_ip/1
```

Static IP

APN Static IP

Request: `POST /api/epc/static_ip`

Response:

```
{
  "static_ip": {
    "apn_profile_id": 5,
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111"
  }
}
```

Parameters:

- `apn_profile_id` - APN ID
- `ipv4_static_ip` & `ipv6_static_ip`

Response:

- `ipv4_static_ip` - IPv4 address
- `ipv6_static_ip` - IPv6 address

IP Address:

- IPv4: `100.64.1.1`
- IPv6: `2606:4700:4700::1111`
- IPv4 & IPv6 IP address
 - IP address
 - IP address associated with APN
 - IP address

Response:

☐☐	IPv4	IPv6	☐☐
☐ IPv4	✓	-	<code>{"ipv4_static_ip": "100.64.1.1"}</code>
☐ IPv6	-	✓	<code>{"ipv6_static_ip": "2606:4700:4700::1111"}</code>
☐☐	✓	✓	☐☐☐☐☐☐☐☐

☐☐☐☐:

☐ IPv4 ☐☐ IP:

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "apn_profile_id": 5,
    "ipv4_static_ip": "100.64.1.1"
  }
}'
```

☐ IPv6 ☐☐ IP:

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "apn_profile_id": 6,
    "ipv6_static_ip": "2606:4700:4700::1111"
  }
}'
```

☐☐☐☐ IP:

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "apn_profile_id": 5,
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111"
  }
}'
```

응답 (201 응답):

```
{
  "data": {
    "id": 1,
    "apn_profile_id": 5,
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111",
    "inserted_at": "2025-11-15T10:30:00Z",
    "updated_at": "2025-11-15T10:30:00Z"
  }
}
```

참고:


- IP 주소 - 100.64.1.1
- APN 이름 - APN ID

응답 IP

응답 IP 주소

참고: PUT /api/epc/static_ip/:id

참고:

		
id	integer	IP ID

Example:

```
{
  "static_ip": {
    "ipv4_static_ip": "100.64.1.2",
    "ipv6_static_ip": "2606:4700:4700::1112"
  }
}
```

Fields:

- `ipv4_static_ip` - IPv4 address
- `ipv6_static_ip` - IPv6 address
- `apn_profile_id` - APN profile ID

Fields:

- `id` - ID

Note: This endpoint is used to create a static IP address for a PDN. The PDN ID and IP address are required.

Example:

```
curl -k -X PUT https://hss.example.com:8443/api/epc/static_ip/1 \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "ipv4_static_ip": "100.64.1.2"
  }
}'
```

Static IP

Static IP

DELETE /api/epc/static_ip/:id

Response:

Field	Type	Description
id	integer	Static IP ID

Request:

```
curl -k -X DELETE https://hss.example.com:8443/api/epc/static_ip/1
```

Notes:

- Static IP ID
- APN (Access Point Name) is required
- Static IP is associated with a specific IP address
- Static IP is associated with a specific APN

Response: Static IP ID, APN, and IP address are returned in the response body.

Static IP Management

Static IP is associated with a specific APN and IP address.

Steps:

1. Static IP is associated with a specific APN
2. Static IP is associated with a specific IP address

Static IP ID:

```
# 1: "internet" APN IP
STATIC_IP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "apn_profile_id": 5,
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111"
  }
}' | jq -r '.data.id')
```

```
# 2: IP
curl -k -X POST https://hss.example.com:8443/api/subscriber \
-H "Content-Type: application/json" \
-d "{
  \"subscriber\": {
    \"imsi\": \"001001123456789\",
    \"key_set_id\": 1,
    \"epc_profile_id\": 1,
    \"static_ips\": [ $STATIC_IP_ID ]
  }
}"
```

IP:

```
curl -k -X PUT https://hss.example.com:8443/api/subscriber/1 \
-H "Content-Type: application/json" \
-d '{
  "subscriber": {
    "static_ips": [1, 2]
  }
}'
```

IP APNs:

IP IP APN

```




# [] "internet" APN [] IP
INTERNET_IP=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "apn_profile_id": 5,
    "ipv4_static_ip": "100.64.1.1"
  }
}' | jq -r '.data.id')

# [] "ims" APN [] IP
IMS_IP=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
-H "Content-Type: application/json" \
-d '{
  "static_ip": {
    "apn_profile_id": 6,
    "ipv4_static_ip": "100.64.2.1"
  }
}' | jq -r '.data.id')

# [] IP []
curl -k -X POST https://hss.example.com:8443/api/subscriber \
-H "Content-Type: application/json" \
-d "{
  \"subscriber\": {
    \"imsi\": \"001001123456789\",
    \"key_set_id\": 1,
    \"epc_profile_id\": 1,
    \"static_ips\": [\$INTERNET_IP, \$IMS_IP]
  }
}"

```

□□□□:

- ✓  : □□□□ APNs □□□□ IP
- ✗ : □□□□ APN □□□□ IP

□□□□ - □□ **APN:**

```
# 创建 IP 地址 APN
curl -k -X POST https://hss.example.com:8443/api/subscriber \
-H "Content-Type: application/json" \
-d '{
  "subscriber": {
    "imsi": "001001123456789",
    "static_ips": [1, 2]
  }
}'

# 响应:
{
  "errors": {
    "static_ips": [
      "指定的 APN 没有 IP 地址。指定的 IP 100.64.1.1 不在
internet 的 100.64.1.2 的 internet"
    ]
  }
}
```

错误:

- 指定的 IP 地址 - 不在 internet
- 指定的 IP 地址 - 不在 internet
- 指定的 IP 地址 - 不在 internet

漫游

漫游配置文件用于 IMS 注册。漫游配置文件包含 MCC/MNC 信息。

漫游配置文件

GET /api/roaming/profile

漫游配置文件

POST /api/roaming/profile

请求:

```
{
  "roaming_profile": {
    "name": "漫游规则",
    "data_action_if_no_rules_match": "deny",
    "ims_action_if_no_rules_match": "deny",
    "roaming_rules": []
  }
}
```

请求:

- "allow" - 允许
- "deny" - 拒绝

请求:

- data_action_if_no_rules_match - 数据漫游规则
- ims_action_if_no_rules_match - IMS 漫游规则

请求

请求: GET /api/roaming/rule

请求

请求: POST /api/roaming/rule

请求:

```
{
  "roaming_rule": {
    "name": "AT&T",
    "mcc": "310",
    "mnc": "410",
    "data_action": "allow",
    "ims_action": "allow"
  }
}
```

□□:

- `mcc` - □□□□□□3 □□□□
- `mnc` - □□□□□□2-3 □□□□
- `data_action` - "allow" □ "deny" □□□□
- `ims_action` - "allow" □ "deny" IMS/□□□□

□□:

- □□□□ - □□□□□□□□
- □□□□ - □□□□□ Diameter □□□□□□□□

EIR □□

OmniHSS □□ S13 Diameter □□□□□□□□□□EIR□□□□EIR □□□□ IMEI □□□□□□□□□□

□□□□□□□□□□S13 □□□□ IMEI □□□□□□□□□□□□ EIR □□

□□ EIR □□

□□: GET /api/eir/rule

□□ EIR □□

□□: POST /api/eir/rule

配置:

```
{
  "eir_rule": {
    "name": "iPhone 6",
    "imei_regex": "^35[0-9]{6}0[0-9]{7}$",
    "action": 1
  }
}
```

参数:

- `name` - 设备名称
- `imei_regex` - IMEI 正则表达式
- `action` - 0: 阻止, 1: 警告, 2: 记录

动作:

- 0 - 阻止
- 1 - 警告
- 2 - 记录

注意:

- 仅支持 IMEI
- 不支持 TAC
- 不支持其他设备类型

来源:

- S13 EIR
- OmniHSS EIR

相关链接

更多配置选项

- **API** - API
- **API** - API
- **API** - API

← API | API: API →

API

← API

API

GET /api/status

```
curl -k https://hss.example.com:8443/api/status
```

```
{  
  "status": "ok"  
}
```

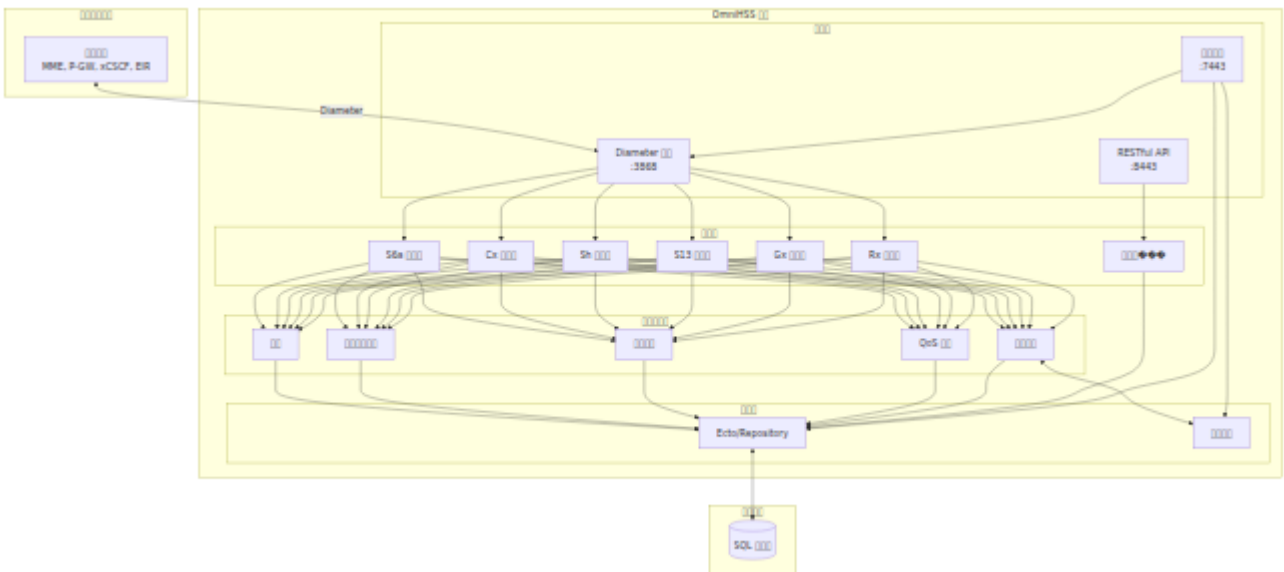
← API

OmniHSS

←

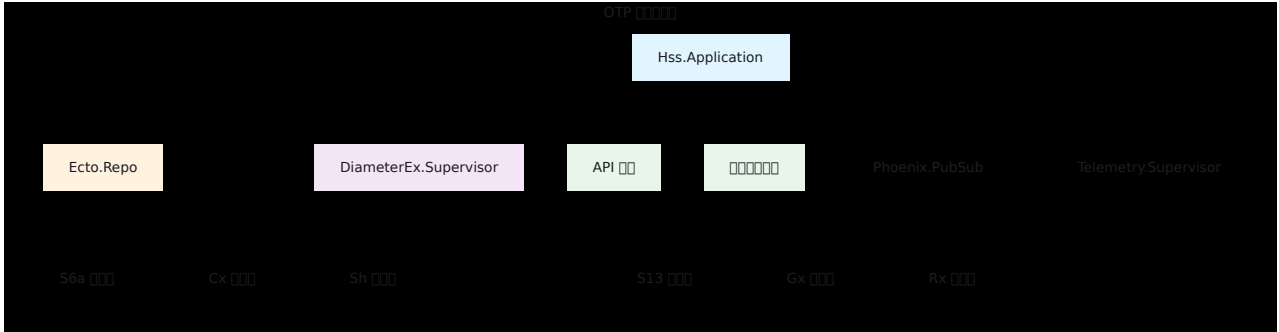
-
-
- Diameter
-
-
-
-

OmniHSS Elixir Erlang/OTP



□□□□

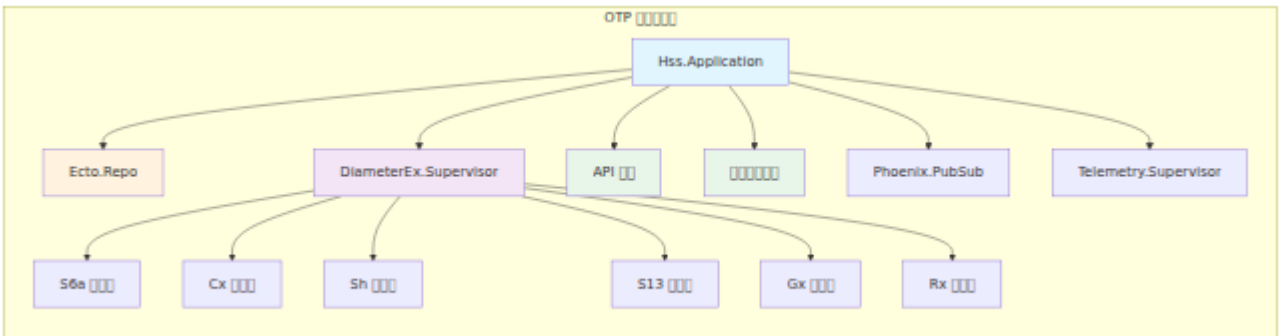
□□□□



Diameter □□□□

□□ Diameter □□□S6a□Cx□Sh□S13□Gx□Rx□□□□□□□□ DiameterEx □□□□□□□□□□

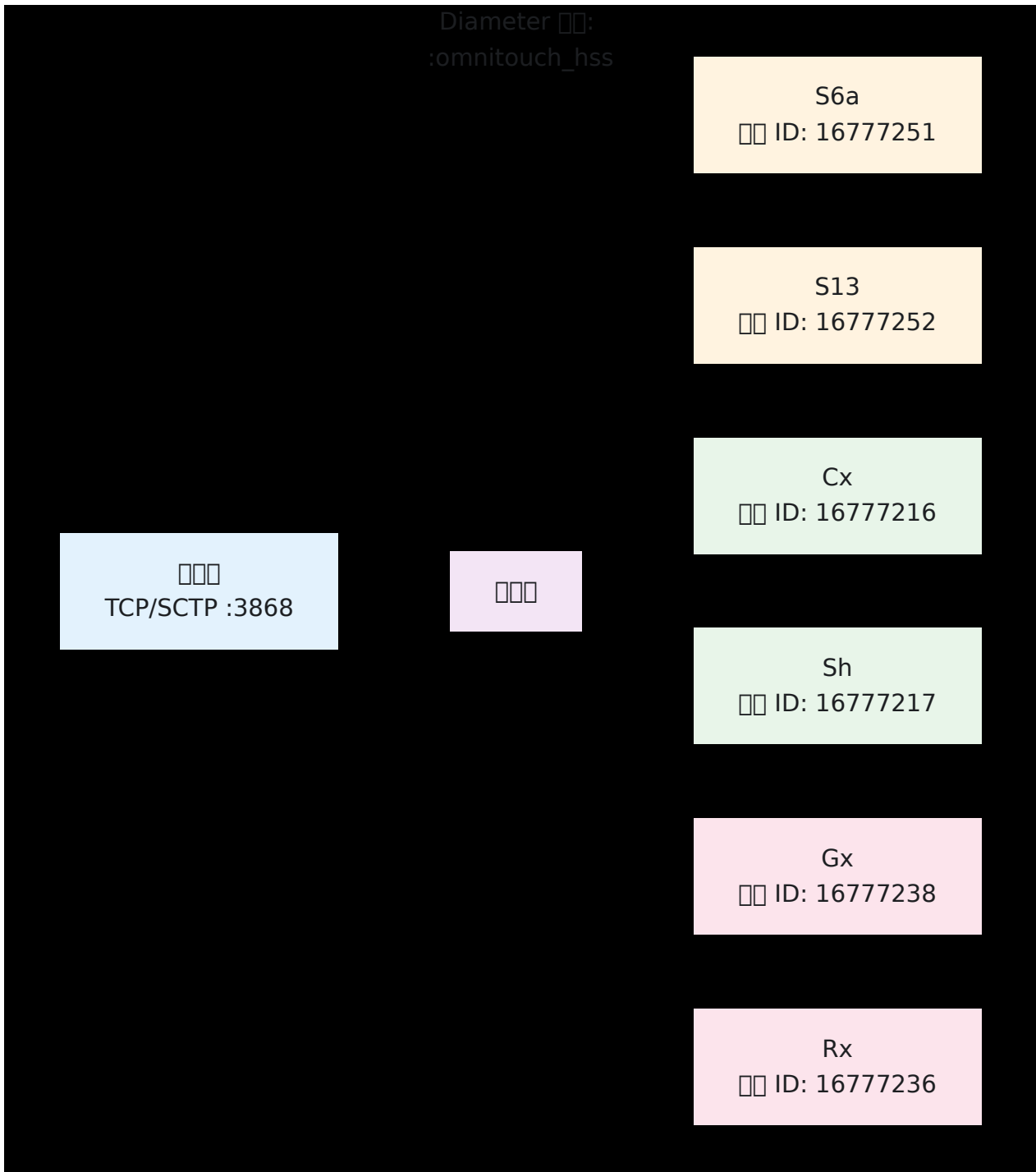
1. □ **DiameterEx** □□ - □□□□□ Diameter □□ ID
2. □□□□ - □□ AVP□□□□□□□□□□
3. □□□□□□ - □□□□□□□□□□□□
4. □□□□ - □□ AVP □□ Diameter □□□□□
5. □□□□ - □□□□□□ Diameter □□□□□



Diameter 00

Diameter 0000

OmniHSS 0000000000000000 Diameter 000



□□□□□□



□□□□□□

Configured

□□□□

Connecting

□□□□

Connected

□□□□

□□□□□□

□□□□□□

□□□□

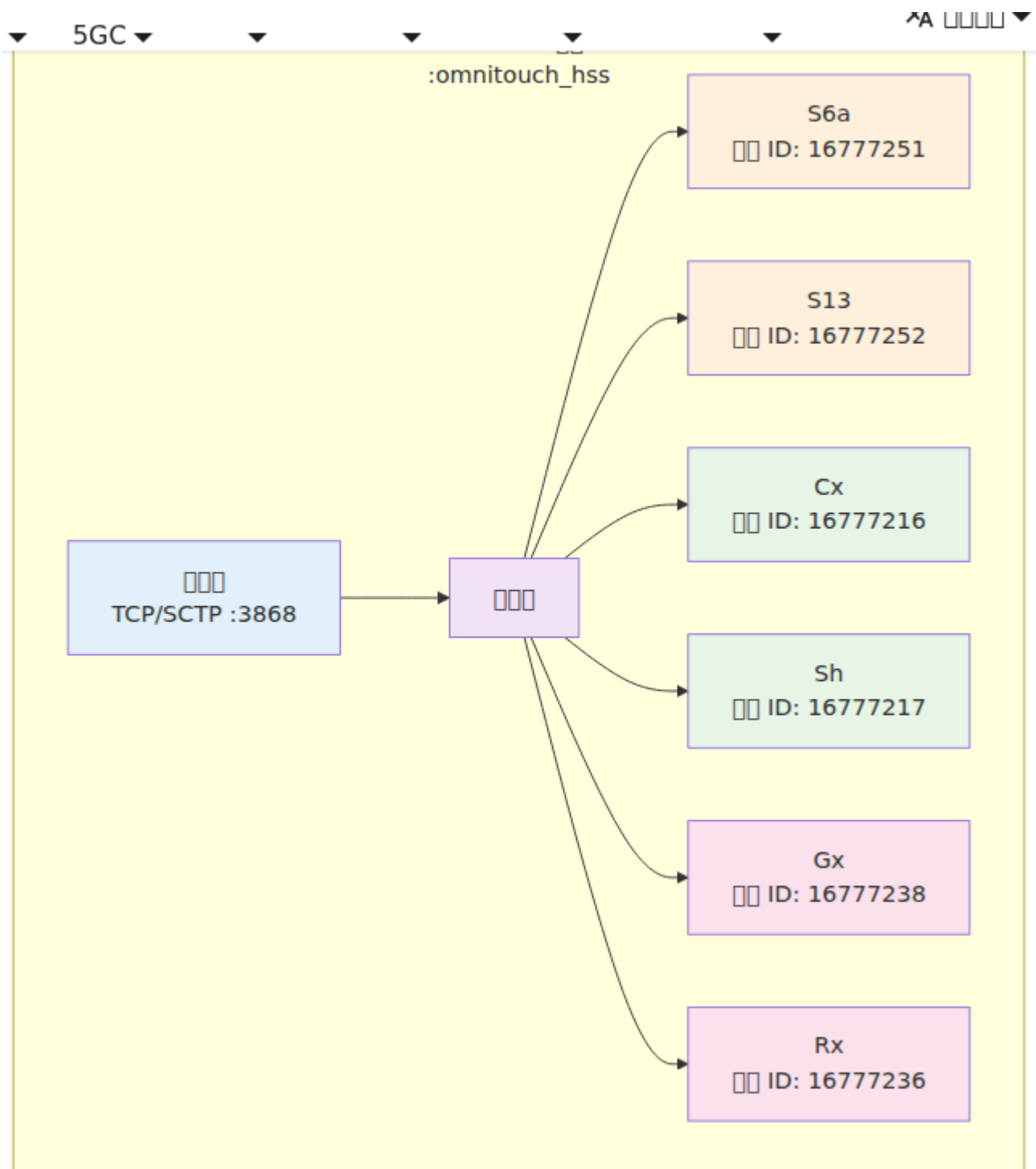
□□□□ Diameter □□



Down

□□□□
□□□□

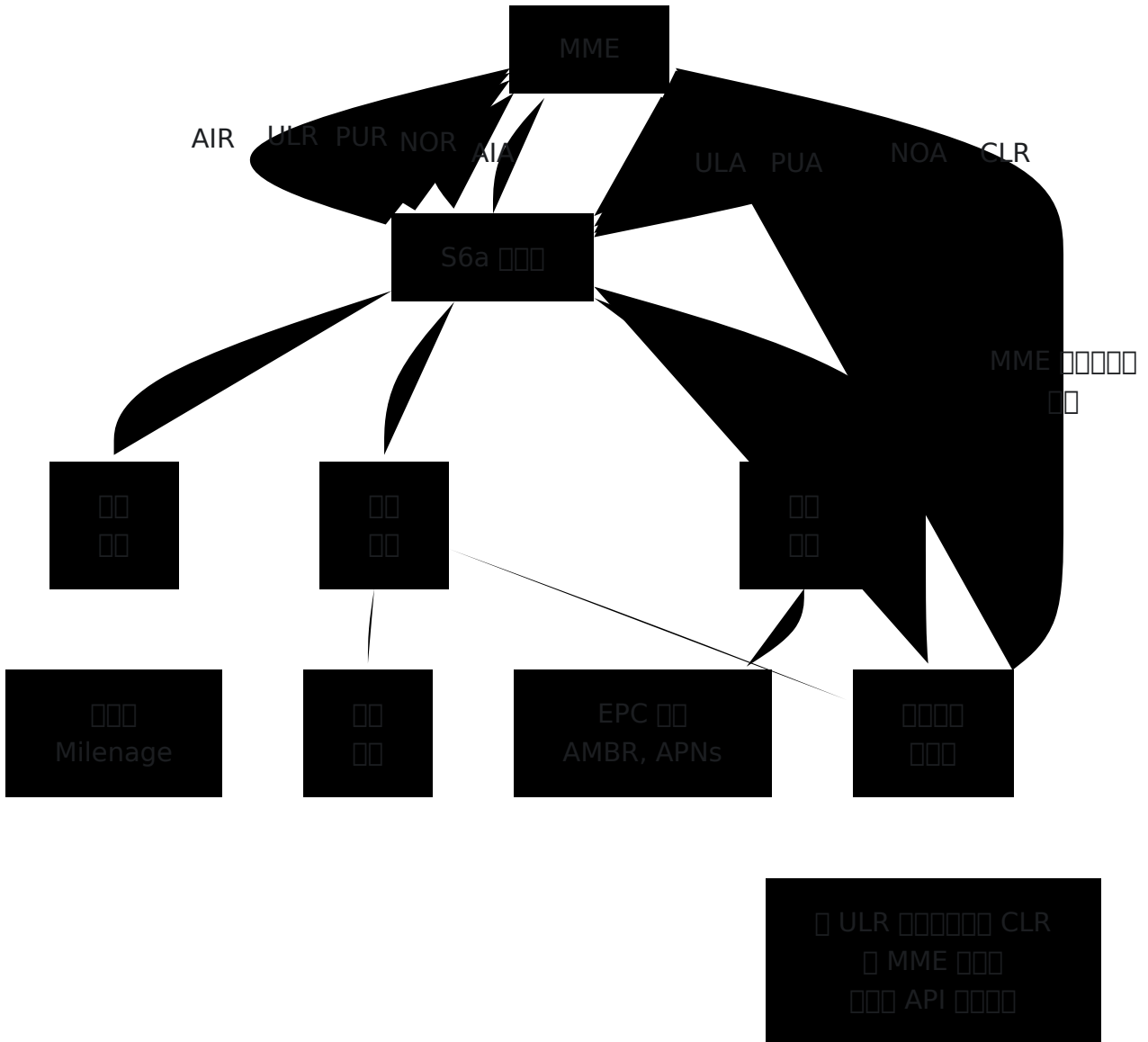
Diameter



□□□

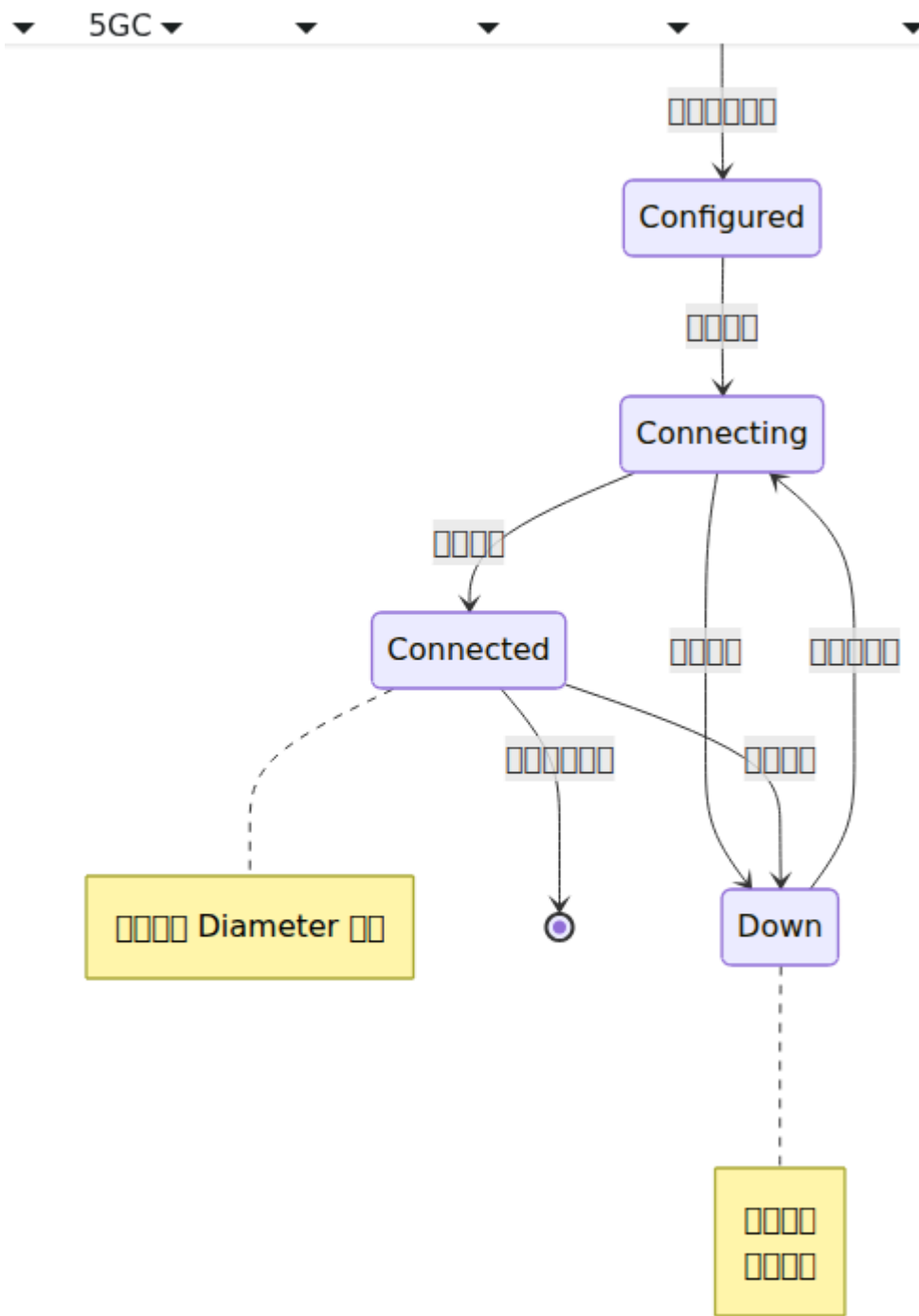
S6a □□ (LTE/EPC)

□□ LTE □□□□□□□□□□



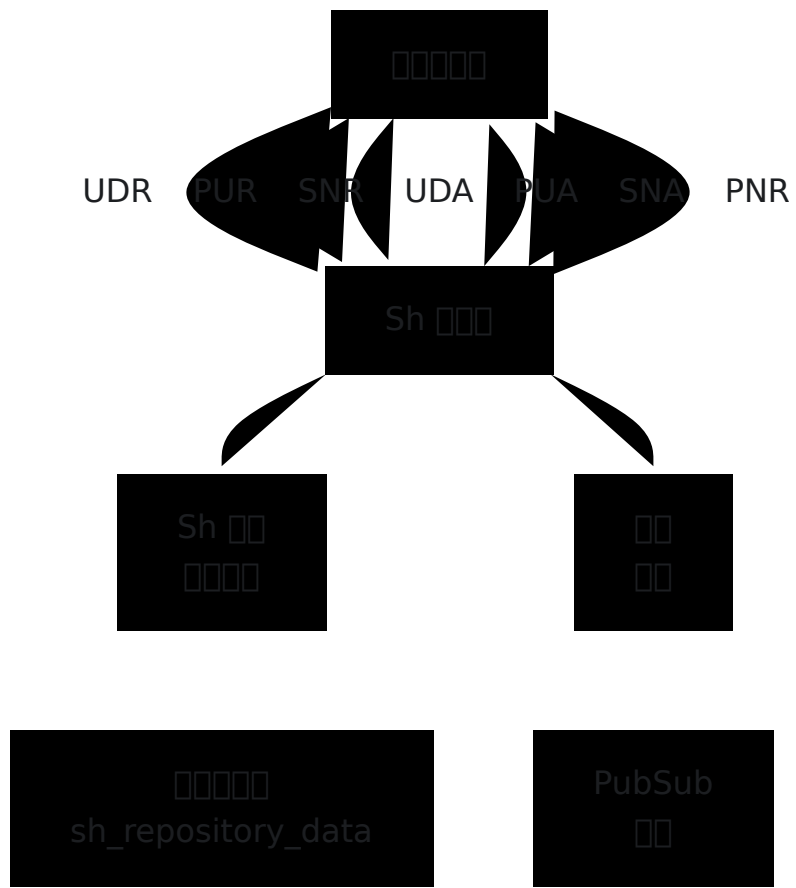
Cx □□ (IMS)

□□ IMS □□□□□□



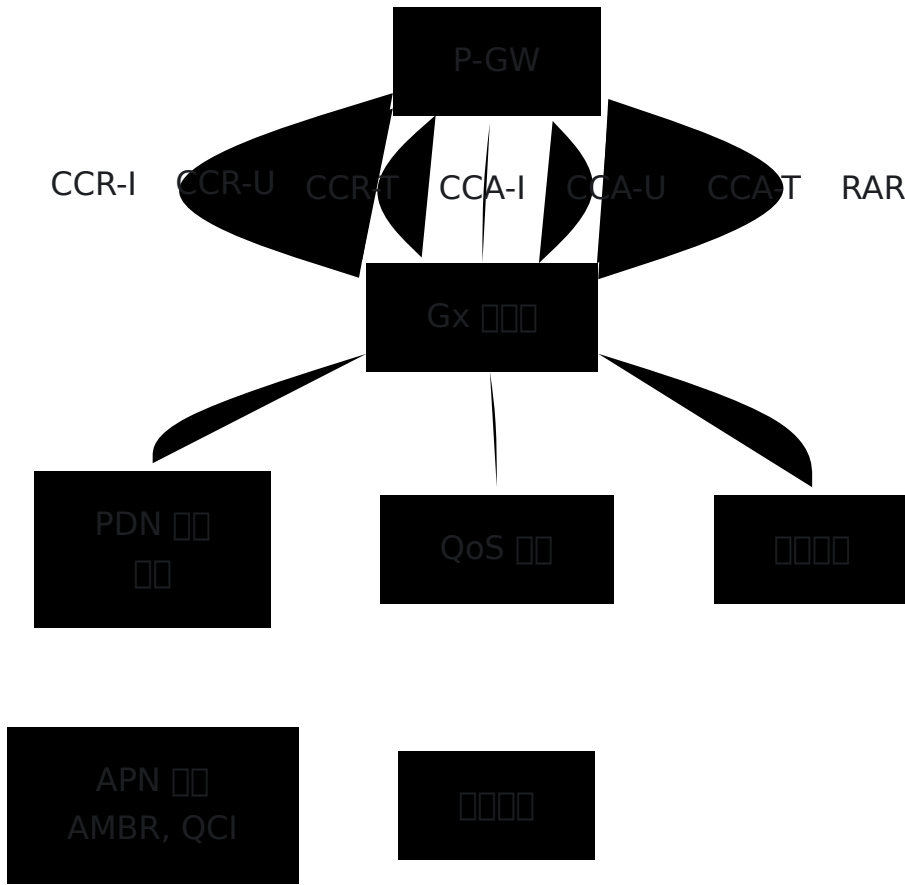
Sh (IMS)

IMS



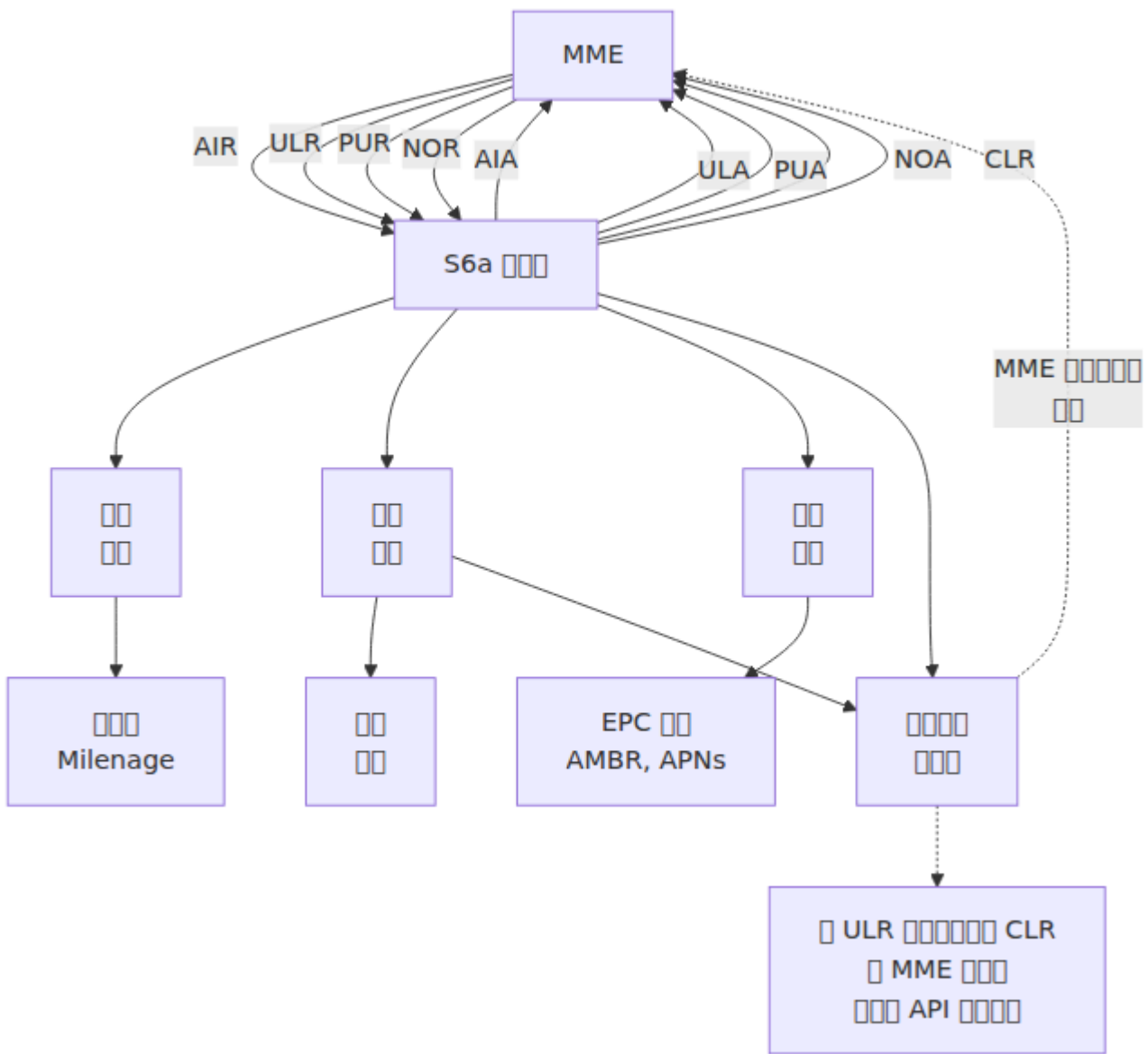
Gx ()

PCRF



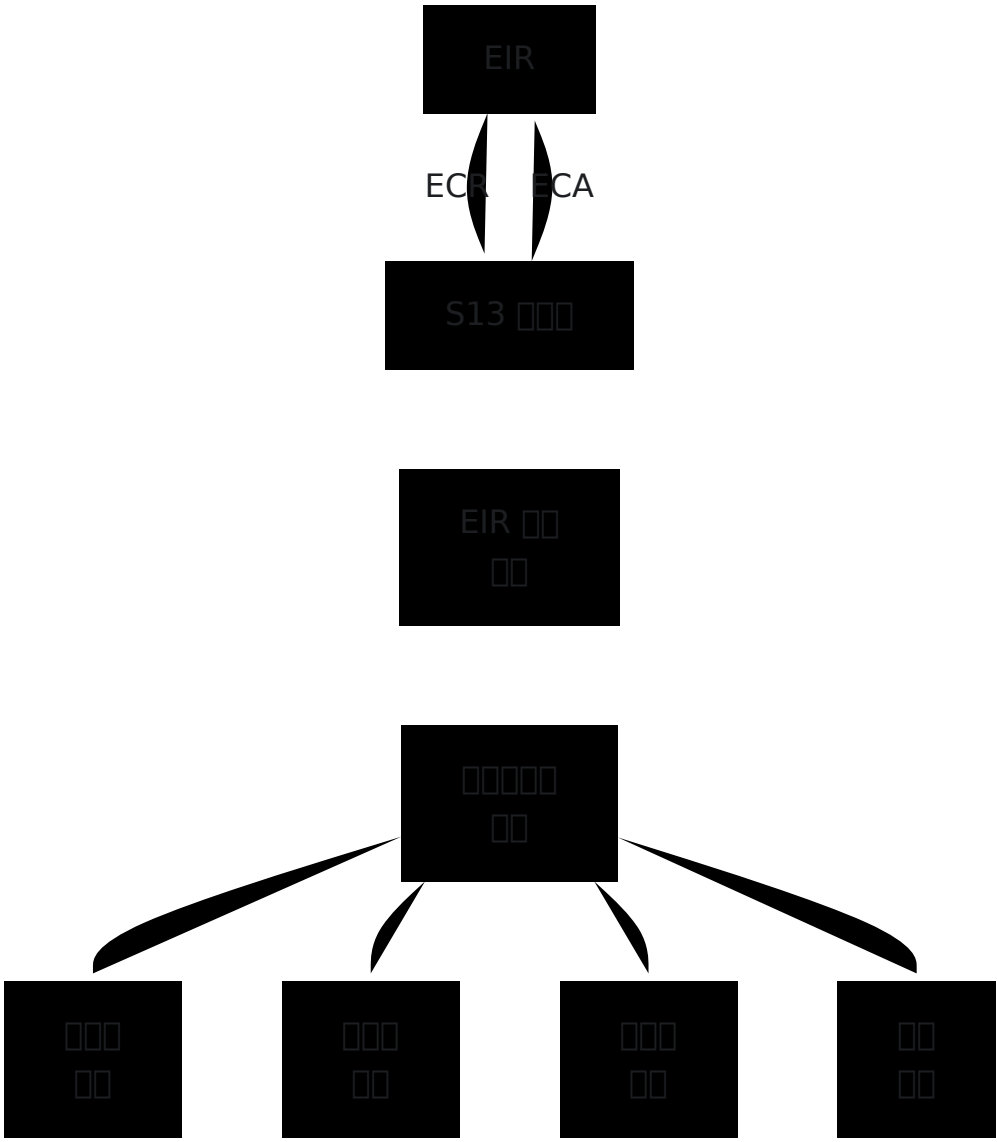
Rx (IMS)

IMS VoLTE PCRF



S13 (EIR)

IMEI EIR



□□□

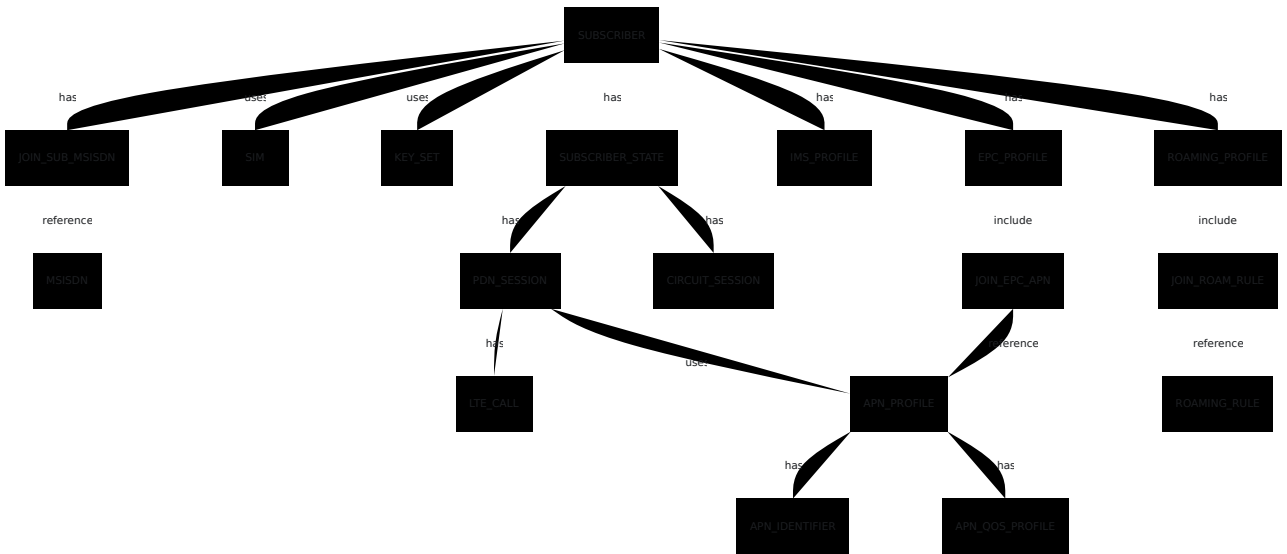
□□□□□

OmniHSS □□ **Ecto** □□□□□□□□□□ Ecto □□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □
 Galera □□□□□□□□□□

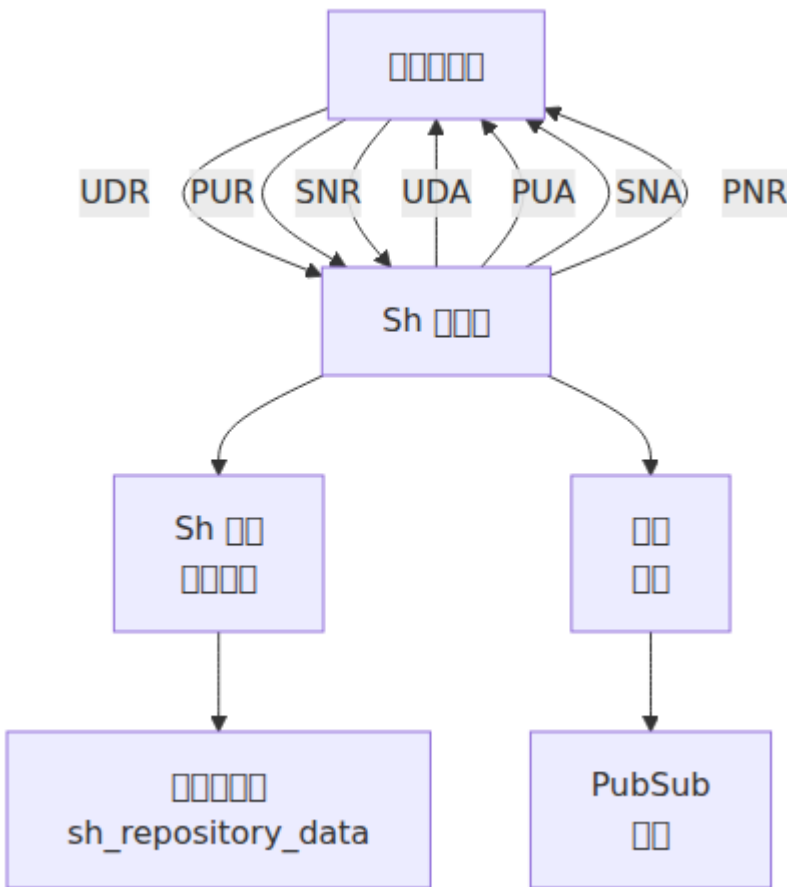
□□□□□□□□□□□□□□□□ □□ □□□□□□□□□□□□□□□□□□□□□□ □□□□□□

□□ Galera □□□□□□□□ □□ Galera □□□□□□ □

□□□□□□□□

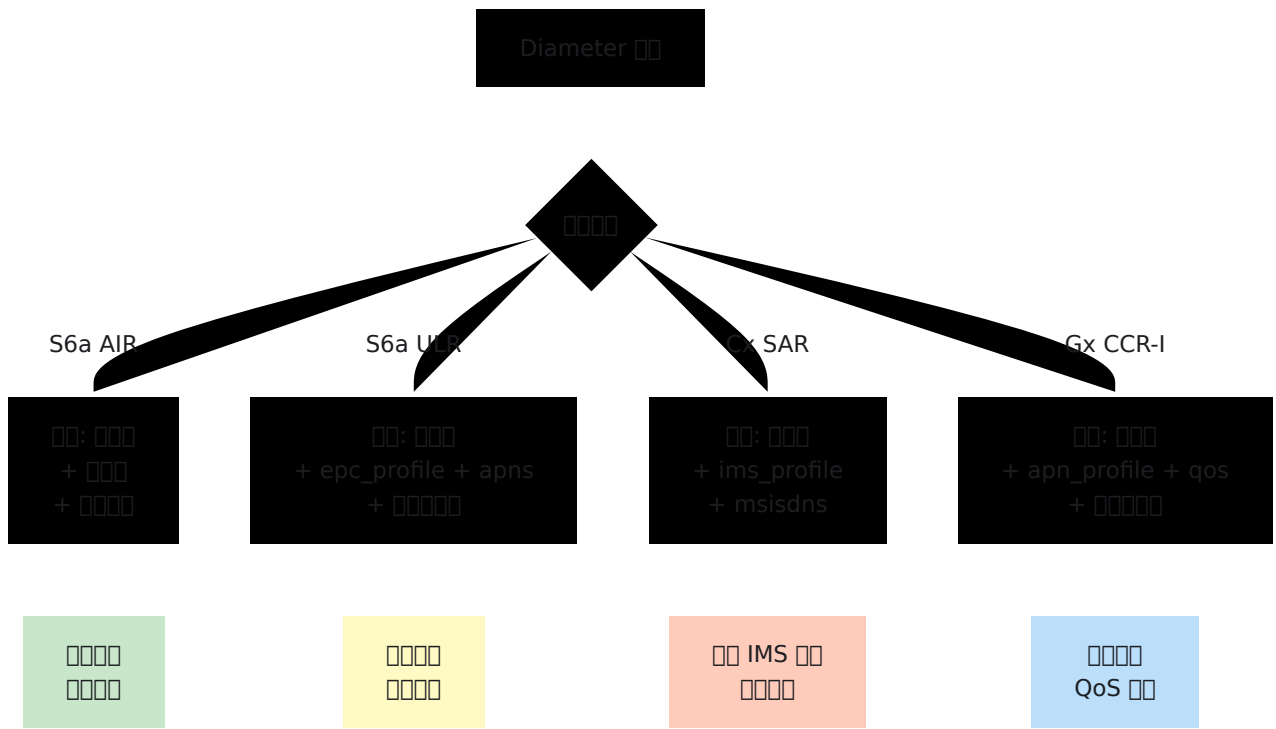


Ecto □□□□



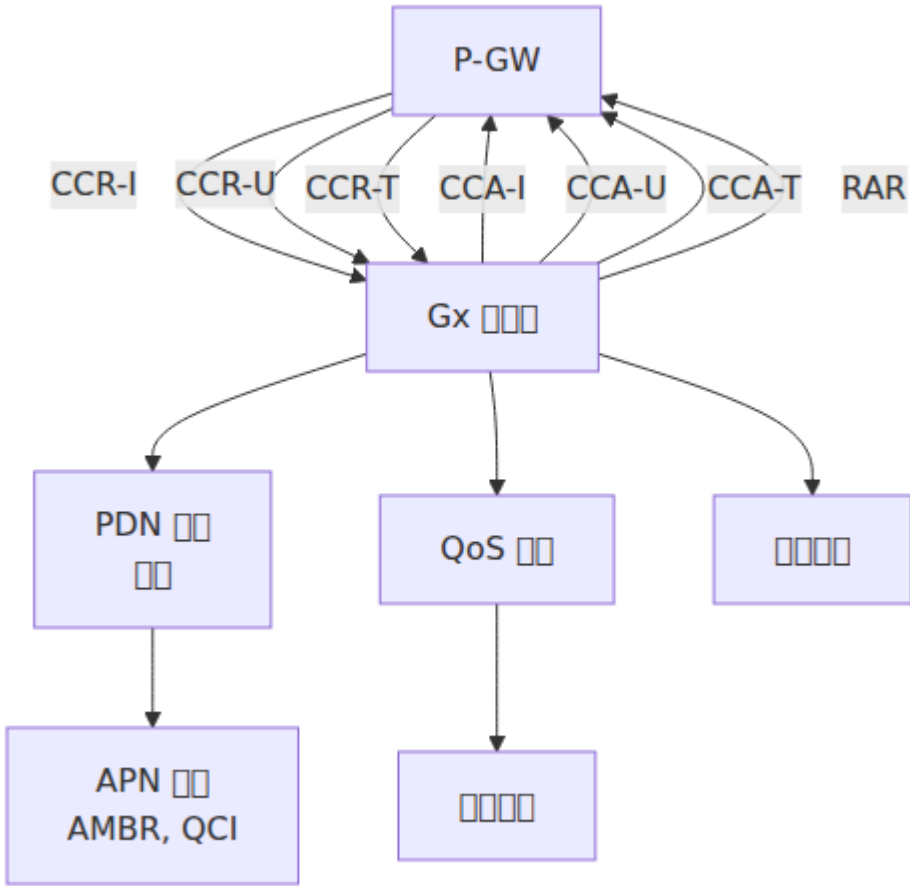
□□□□□□

□□ Diameter □□□□□□□□□□□□□□□□□□

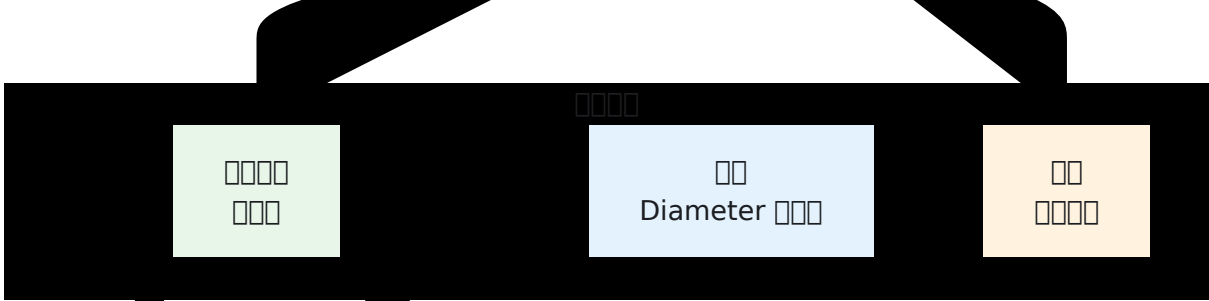
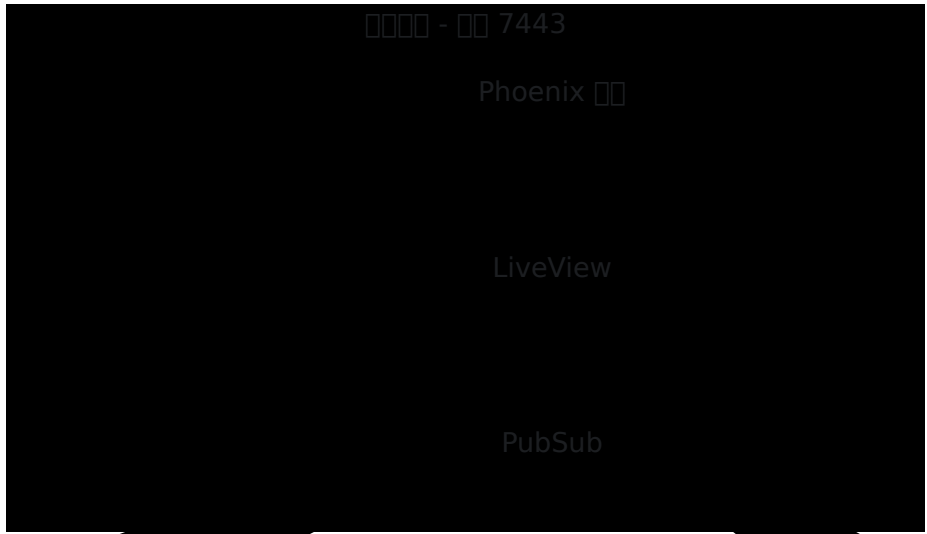


□□□□

API □□

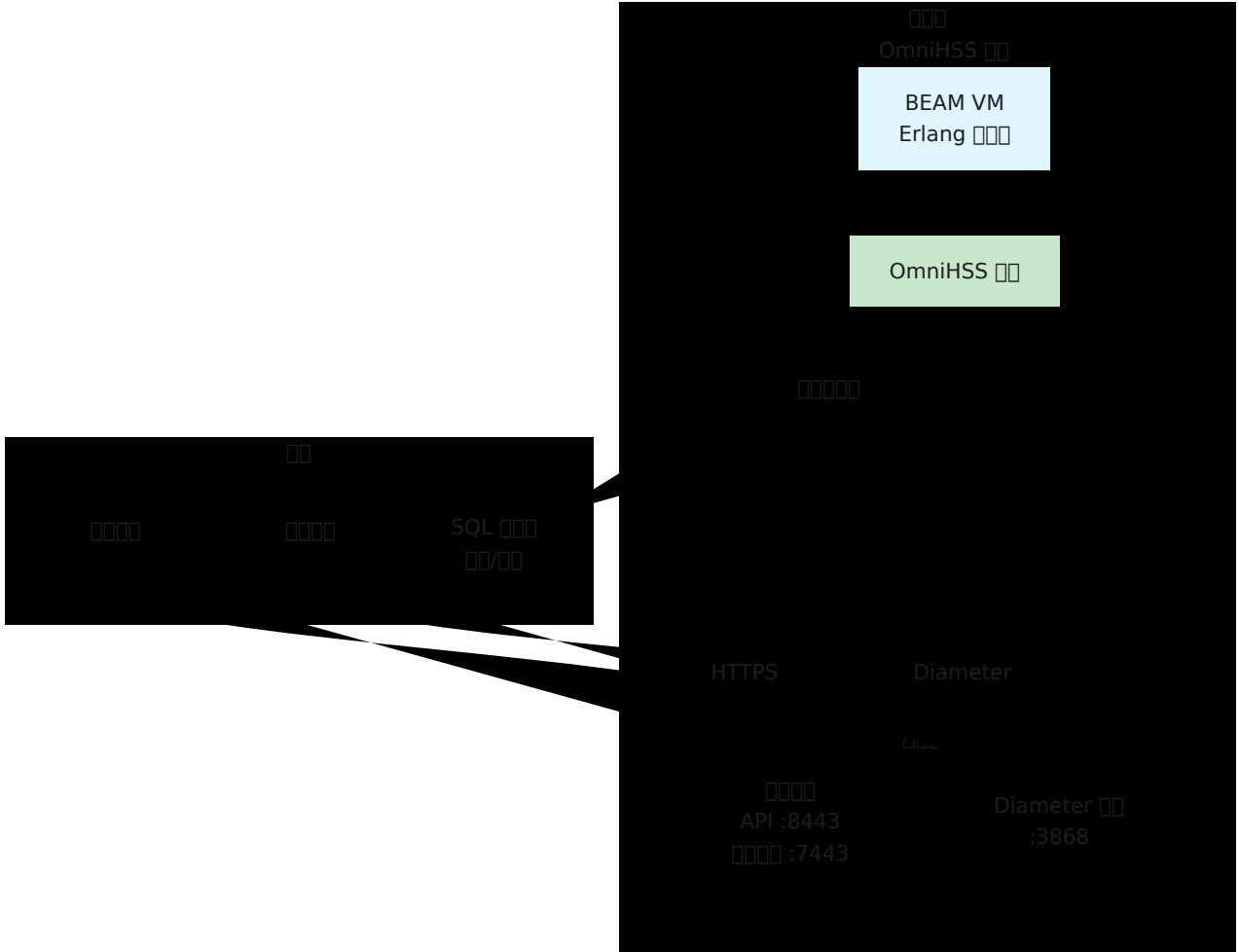


□□□□□□



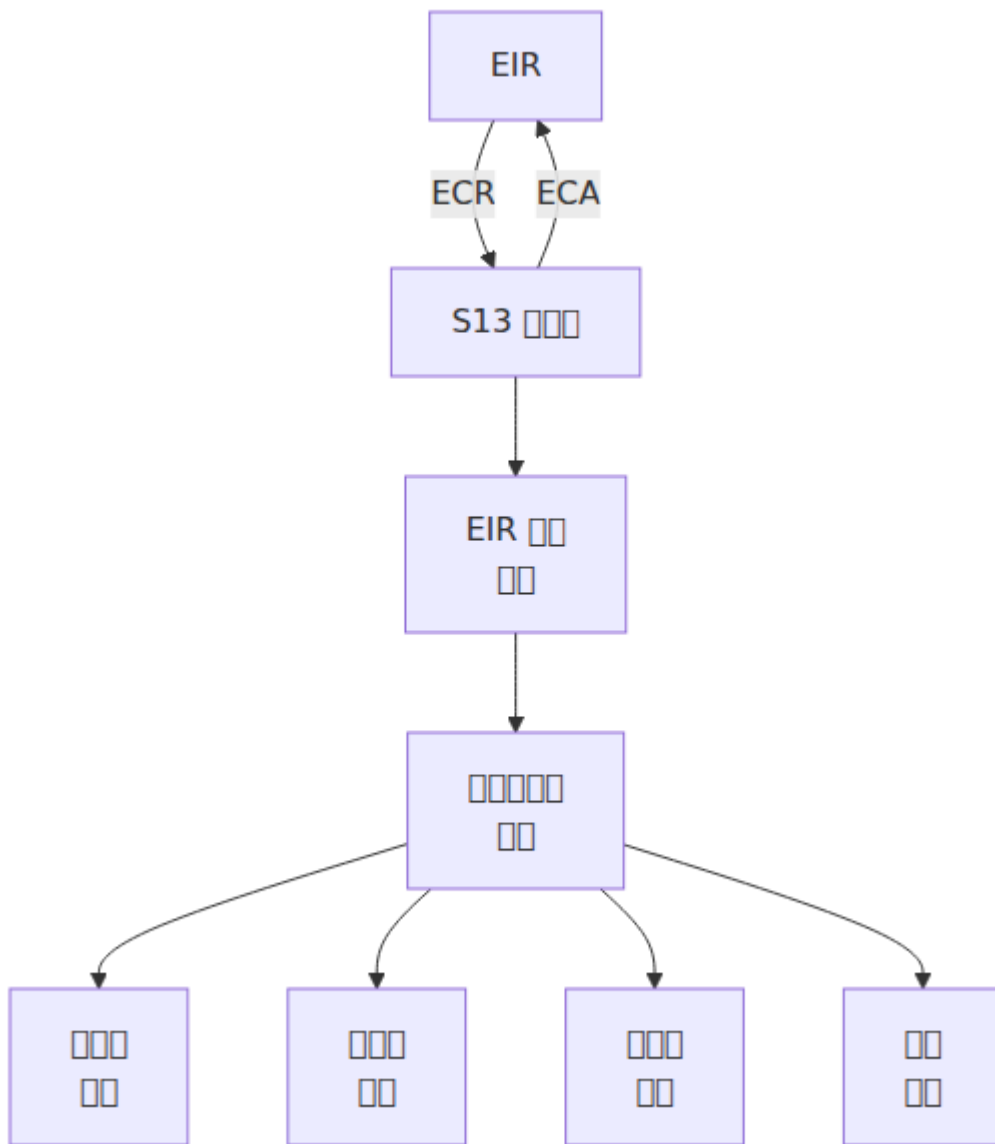
□□□□

□□□□□



□□□□□□□□ (Galera □□)

□□□□□□□□ OmniHSS □□ MariaDB Galera □□□□□□□□□□□□□□



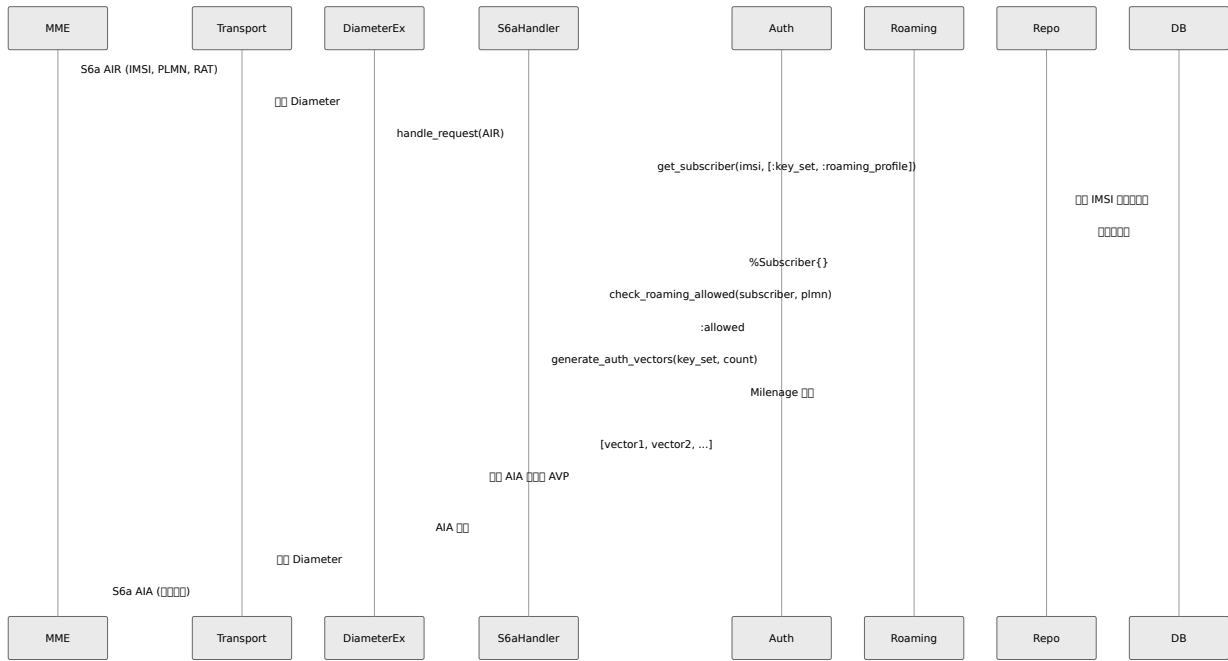
□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□□□□□□□□ **Galera** □□□□□

□□□□□□□□

□□□□□□□□□□□□□□



□□□□□□

1. □□□

- Erlang/OTP □□□□□□□□□□□□
- □□□ Diameter □□□□□□□□□□
- □□□□□□□□□□□□□□□□

2. □□□

- □□ Diameter □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

3. □□□

- □□ Diameter □□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□

4. 00

- 00000000000000000000
- 0000000000000000
- 00000000

5. 0000

- 0000000000000000
- 0000000000000000
- Diameter 00000000
- 00000000000000

OmniHSS 架构图

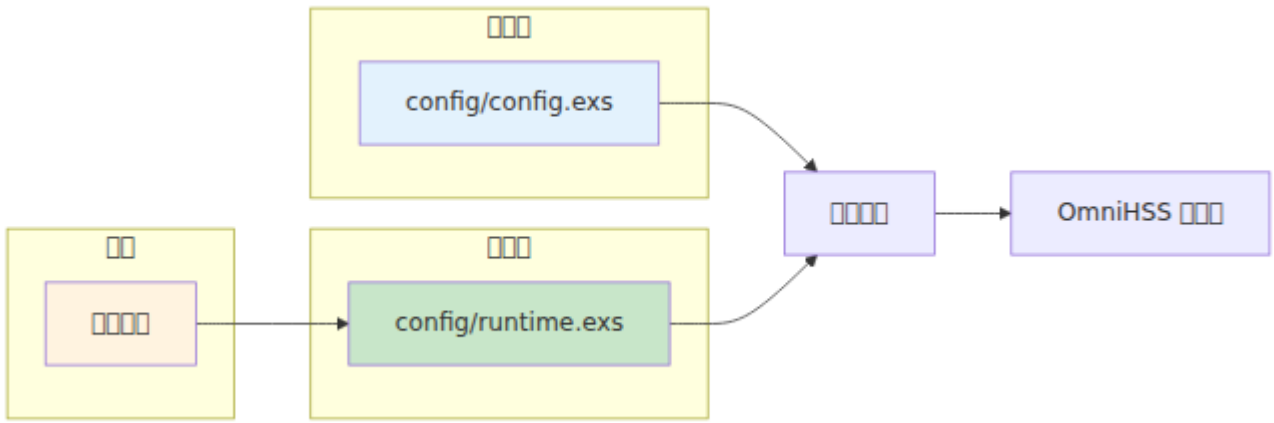
← 架构图

架构

- 核心网
 - 接入网
 - 终端
 - 网络切片
 - Diameter 接口
 - 鉴权
 - 3GPP PLMN 鉴权
 - HSS 鉴权
 - IMS 核心网
 - EIR 鉴权
 - API 鉴权
 - 鉴权
-

鉴权

OmniHSS 鉴权架构图



config/config.exs (HSS)

Configuration parameters

- HSS parameters
- API parameters
- HSS

config/runtime.exs (HSS)

Configuration parameters

- HSS parameters
- Diameter parameters
- PLMN
- IMS S-CSCF
- HSS

Configuration parameters

Configuration parameters HSS

```
# config/runtime.exs

config :license_client,
  # 许可证 API 端点 URL 列表
  license_server_api_urls:
  ["https://license.example.com:8443/api"],

  # 许可证持有者
  licensee: "许可证持有者",

  # 产品名称
  product_name: "omnihss"
```

配置项

配置项	数据类型	是否必填	默认值
license_server_api_urls	许可证 URL 列表	否	["https://10.0.0.1:8443/api"]
licensee	许可证持有者名称	否	"ACME Telecom"
product_name	产品名称	否	"omnihss"

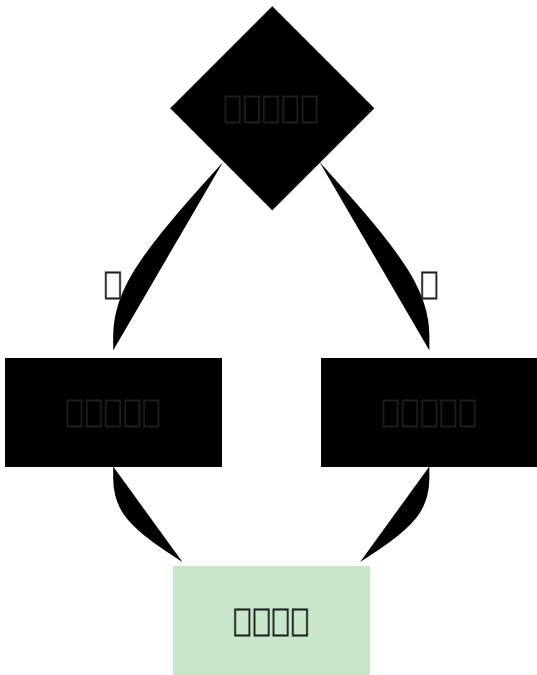
注意事项

- 许可证持有者名称 HSS 必填
- 许可证 URL 列表必填
- 许可证 URL 列表必填
- 许可证持有者名称必填

□□□□□

□□□□□

□□□□



□□□□□□

OmniHSS □□□□□□□□

- □□□□□□□□□□□□□□
 - □□□□ runtime.exs □□□
 - □□□□□□□□□□□□□□□□
-

環境変数

データベース接続

```
# config/runtime.exs

config :hss, Hss.Repo,
  # データベース接続
  username: System.get_env("DATABASE_USERNAME", "root"),
  password: System.get_env("DATABASE_PASSWORD", "password"),
  hostname: System.get_env("DATABASE_HOSTNAME", "localhost"),
  database: System.get_env("DATABASE_NAME", "omnihss"),

  # プールサイズ
  pool_size:
    String.to_integer(System.get_env("DATABASE_POOL_SIZE", "20")),

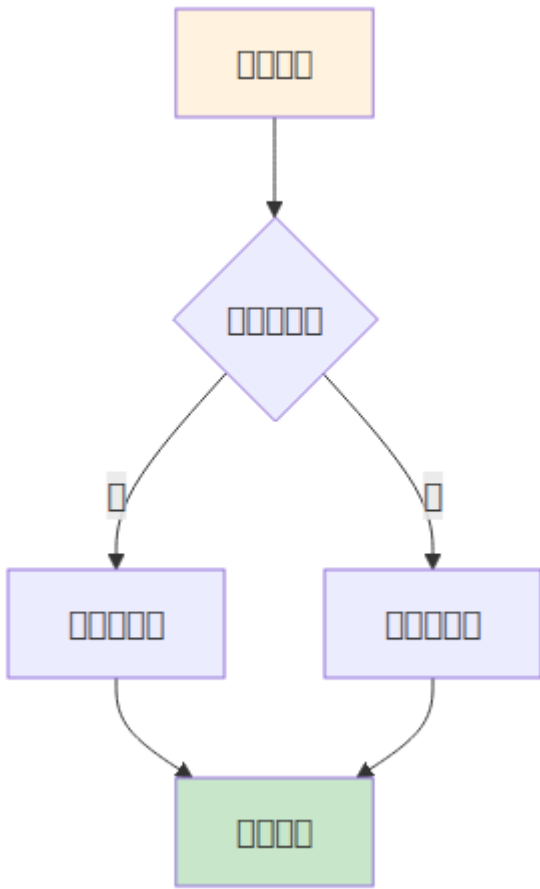
  # タイムアウト
  timeout: 15_000,
  connect_timeout: 15_000,

  # 接続エラー時の挙動
  show_sensitive_data_on_connection_error: false
```

環境変数の設定

変数名	説明	デフォルト値	推奨値
username	SQL データベースユーザ名	"root"	データベースユーザ名
password	SQL データベースパスワード	"password"	データベースパスワード
hostname	SQL データベースホスト名	"localhost"	データベース FQDN または IP
database	データベース名	"omnihss"	データベース名
pool_size	接続プールサイズ	20	データベースに10-50接続を許可

□□□□□



□□□□□

- □ 20 □□□□□
- □□“□□□□□”□□
- □□□□□□□□□□□□□□□□ 10
- □□□□□□□ 4MB □□□
- □□□□□□□□□ SQL □□□□□

□□□□□□□□□□

```
# config/runtime.exs - □□□□

config :hss, Hss.Repo,
  username: System.fetch_env!("DATABASE_USERNAME"),      # □□□□□
  password: System.fetch_env!("DATABASE_PASSWORD"),      # □□□□□
  hostname: System.get_env("DATABASE_HOSTNAME",
"db.internal.example.com"),
  database: System.get_env("DATABASE_NAME", "omnihss"),
  port: String.to_integer(System.get_env("DATABASE_PORT",
"3306")),
  pool_size:
String.to_integer(System.get_env("DATABASE_POOL_SIZE", "30")),
  ssl: true,
  ssl_opts: [
    cacertfile: "/etc/ssl/certs/mysql-ca.pem",
    verify: :verify_peer
  ]
]
```

Diameter ☐☐

Diameter ☐☐☐☐

```
# config/runtime.exs

diameter_config = %{
  service_name: :omnitouch_hss,

  # ☐☐☐☐
  listen_ip: System.get_env("DIAMETER_LISTEN_IP", "10.7.25.186"),
  listen_port:
String.to_integer(System.get_env("DIAMETER_LISTEN_PORT", "3868")),

  # Diameter ☐☐
  host: System.get_env("DIAMETER_HOST", "omnihss"),
  realm: System.get_env("DIAMETER_REALM",
"epc.mnc001.mcc001.3gppnetwork.org"),

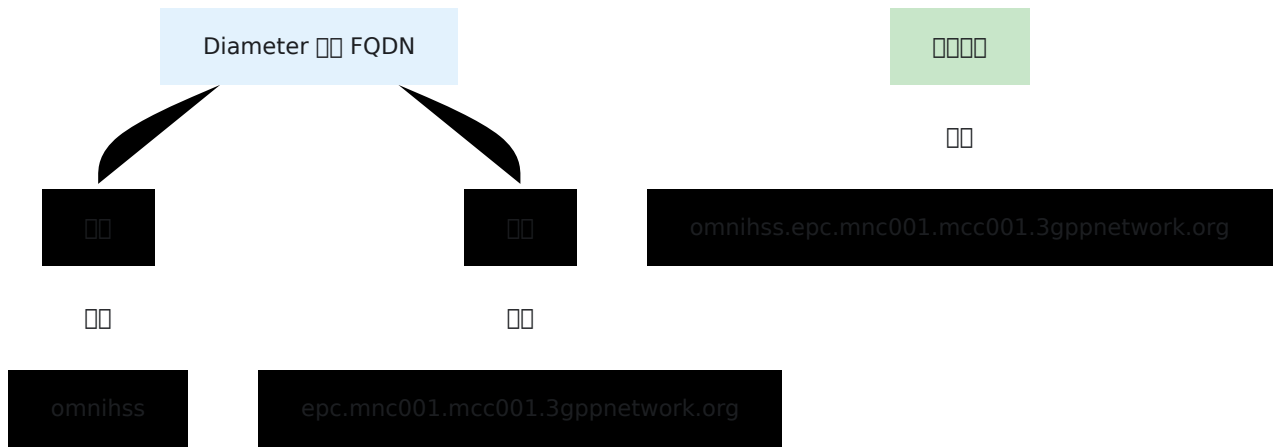
  # ☐☐☐☐
  product_name: "OmniHSS",
  vendor_id: 10415, # 3GPP
  supported_vendor_ids: [5535, 10415],

  # ☐☐☐☐
  request_timeout: 5000,

  # ☐☐☐☐☐
  peers: [
    # ☐☐☐☐☐☐☐☐☐☐
  ]
}

config :hss, :diameter, diameter_config
```

Diameter FQDN



FQDN

- HSS FQDN "omnihss" "hss01"
- PLMN FQDN "epc.mnc001.mcc001.3gppnetwork.org"
- FQDN format: `{host}.{realm}`

Diameter FQDN

FQDN

```

# config/runtime.exs

peers: [
  # MME
  %{
    host: "mme01.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.100",
    port: 3868,
    transport: :sctp, # :tcp
    applications: [:s6a]
  },

  # P-GW
  %{
    host: "pgw01.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.101",
    port: 3868,
    transport: :sctp,
    applications: [:gx]
  },

  # I-CSCF
  %{
    host: "icscf01.ims.mnc001.mcc001.3gppnetwork.org",
    realm: "ims.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.102",
    port: 3868,
    transport: :tcp,
    applications: [:cx]
  }
]

```

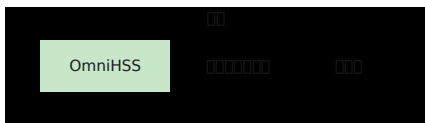
□□□□

□□□□□□◆◆□□□ HSS □□□□

```
# config/runtime.exs

diameter_config = %{
  # ...
  peers: [] # -
}
```

Diameter



SCTP			Diameter
TCP			SCTP

PLMN

PLMN

```
# config/runtime.exs

config :hss, :home_plmn, %{
  mcc: System.get_env("HOME_PLMN_MCC", "001"), #
  mnc: System.get_env("HOME_PLMN_MNC", "001") #
}
```

HSS 配置

配置 HSS 配置文件

```
# config/runtime.exs

config :hss,
  # 配置 Ecto 仓库
  ecto_repos: [Hss.Repo],

  # MME 配置 CLR 选项
  send_clr_on_mme_change: true,

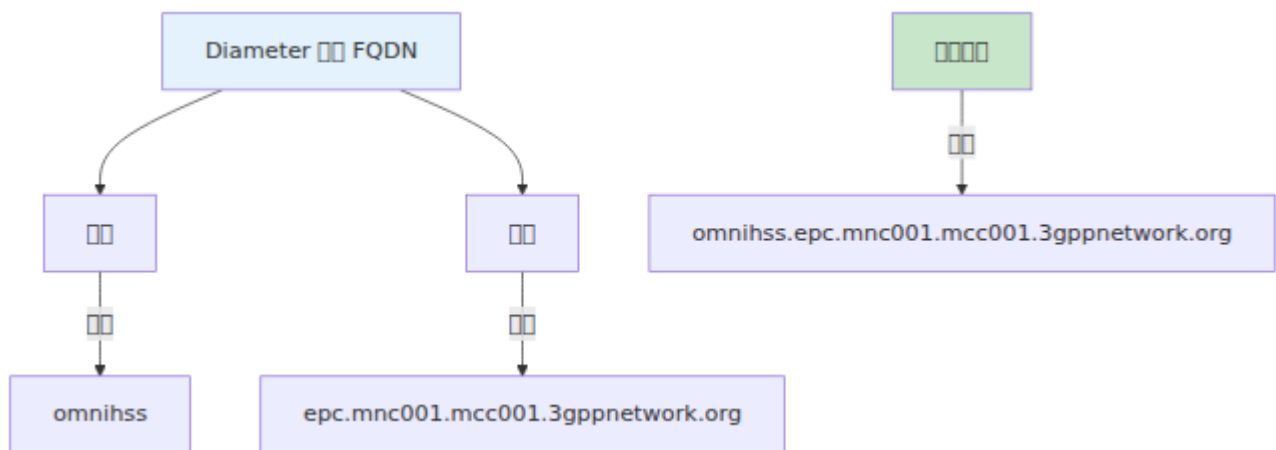
  # 配置 Diameter 选项
  stop_diameter_on_database_failure: true,

  # 配置许可证
  license_enforced: true,
  license_module: LicenseClient
```

HSS 配置

名前	説明	デフォルト値	カテゴリ
ecto_repos	データベースリポジトリ Ecto リポジトリ	[Hss.Repo]	データベース リポジトリ
send_clr_on_mme_change	MME 状態変更 通知	true	通知 フラグ
stop_diameter_on_database_failure	データベースエラー発生時 Diameter サーバを停止	true	エラー 処理
license_enforced	ライセンス強制	true	ライセンス フラグ
license_module	ライセンスモジュール 名	LicenseClient	ライセンス モジュール

PLMN



PLMN

- AT&T MCC=310 MNC=410
- Verizon MCC=311 MNC=480
- Vodafone MCC=234 MNC=15
- MCC=001 MNC=01

□□□□□□

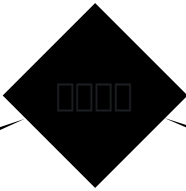
```
# config/runtime.exs

# Diameter □□
listen_ip: System.get_env("DIAMETER_LISTEN_IP", "0.0.0.0"), # □□□
□
# □□□□□□
# listen_ip: "10.7.25.186",

# API □□
config :hss, HssWeb.Api.Endpoint,
  http: [
    ip: {0, 0, 0, 0}, # □□□□
    port: 8443
  ]

# □□□□□□
config :hss, HssWeb.ControlPanel.Endpoint,
  http: [
    ip: {0, 0, 0, 0}, # □□□□
    port: 7443
  ]
```

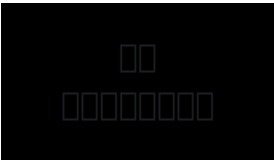
□□□□□□



0.0.0.0
(□□□□)

□□ IP
(□□□192.168.1.10)

127.0.0.1
(□□□□□)



IMS

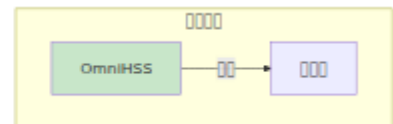
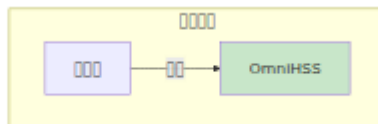
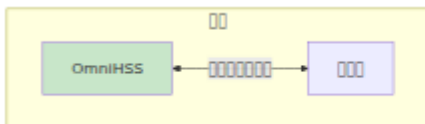
S-CSCF

```
# config/runtime.exs

config :hss, :ims, %{
  scscf: %{
    # :random_peer | :round_robin
    selection_method: :random_peer,

    # S-CSCF
    peers: [
      %{
        host:
        "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060",
        capabilities: [] #
      },
      %{
        host:
        "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060",
        capabilities: []
      }
    ]
  }
}
```

S-CSCF



S-CSCF

Peer	Service	Priority
:random_peer	S-CSCF	1
:round_robin	S-CSCF	2

IMS

IMS EPC

```
# EPC
"epc.mnc001.mcc001.3gppnetwork.org"

# IMS
"ims.mnc001.mcc001.3gppnetwork.org"
```

EIR

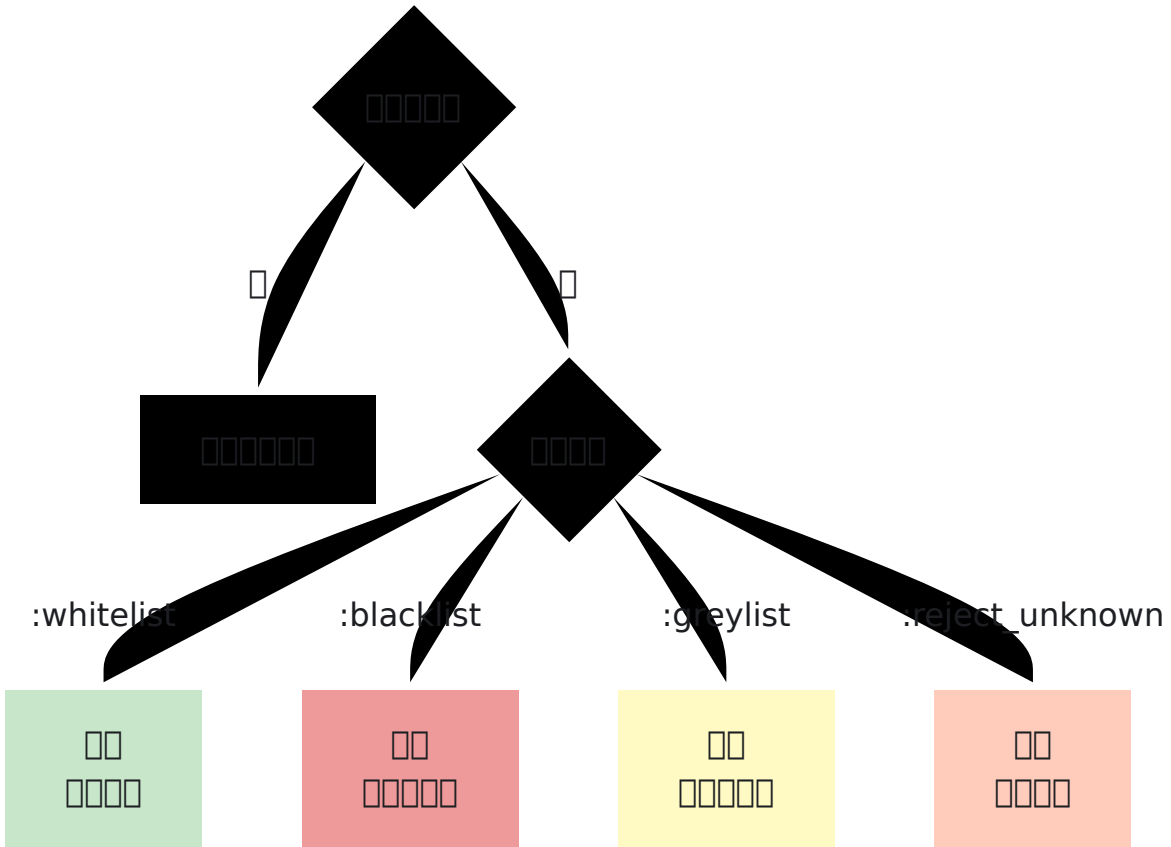
EIR

```
# config/runtime.exs

config :hss, :eir, %{
  #
  unknown_equipment_behaviour: :whitelist
  #
  # :whitelist -
  # :blacklist -
  # :greylist -
  # :reject_unknown_equipment -
}
```

□□□□□□

IMEI □□□□



□□□□□

□□	□□	□□
:whitelist	□□□□□□ IMEI	□□□□□□□□
:blacklist	□□□□□□ IMEI	□□□□□□
:greylist	□□□□□□□□ IMEI	□□□□□□
:reject_unknown_equipment	□□□□□□□□□□	□□□□□□

□□□ □□□□□□ :whitelist □□□□□□□□□□ :greylist□□□□ :blacklist □□□□□□□□□□

API 〇〇〇〇〇〇〇〇

API 〇〇〇〇

```
# config/config.exs

config :hss, HssWeb.Api.Endpoint,
  url: [host: "localhost"],
  render_errors: [view: HssWeb.ErrorView, accepts: ~w(json)],
  pubsub_server: Hss.PubSub,

# HTTPS 〇〇
https: [
  port: 8443,
  cipher_suite: :strong,
  certfile: "priv/cert/omnitouch.crt",
  keyfile: "priv/cert/omnitouch.pem"
]
```

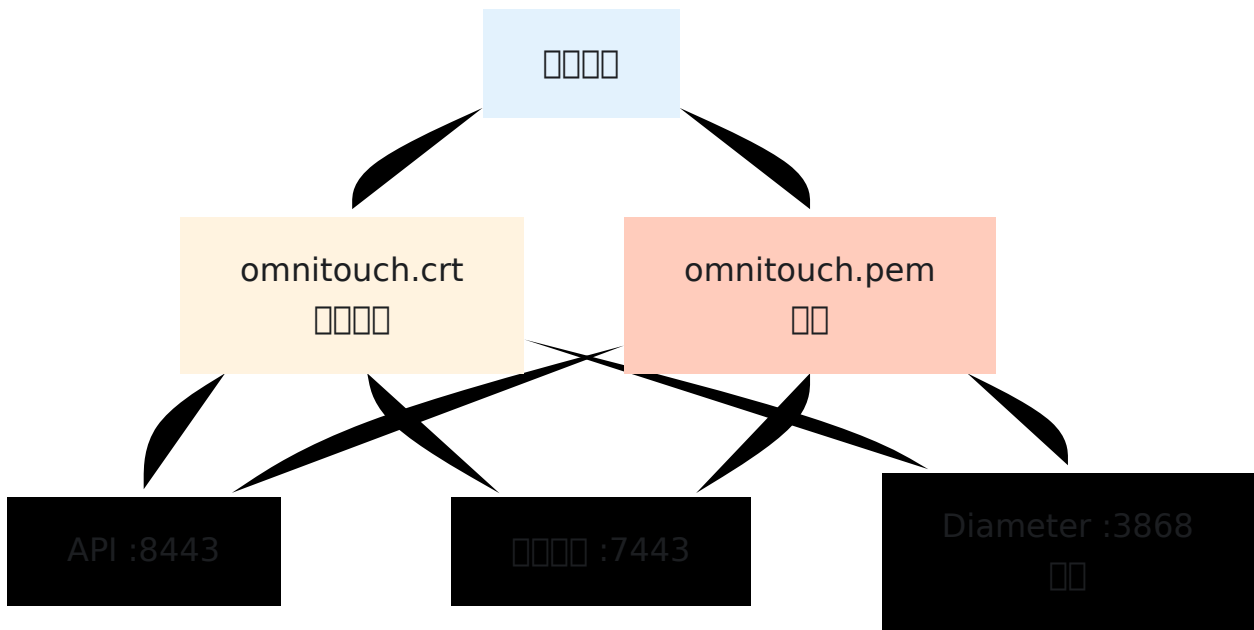
〇〇〇〇〇〇〇

```
# config/config.exs

config :hss, HssWeb.ControlPanel.Endpoint,
  url: [host: "localhost"],
  render_errors: [view: HssWeb.ErrorView, accepts: ~w(html json)],
  pubsub_server: Hss.PubSub,
  live_view: [signing_salt: "some-secret"],

# HTTPS 〇〇
https: [
  port: 7443,
  cipher_suite: :strong,
  certfile: "priv/cert/omnitouch.crt",
  keyfile: "priv/cert/omnitouch.pem"
]
```

TLS 証明書



証明書

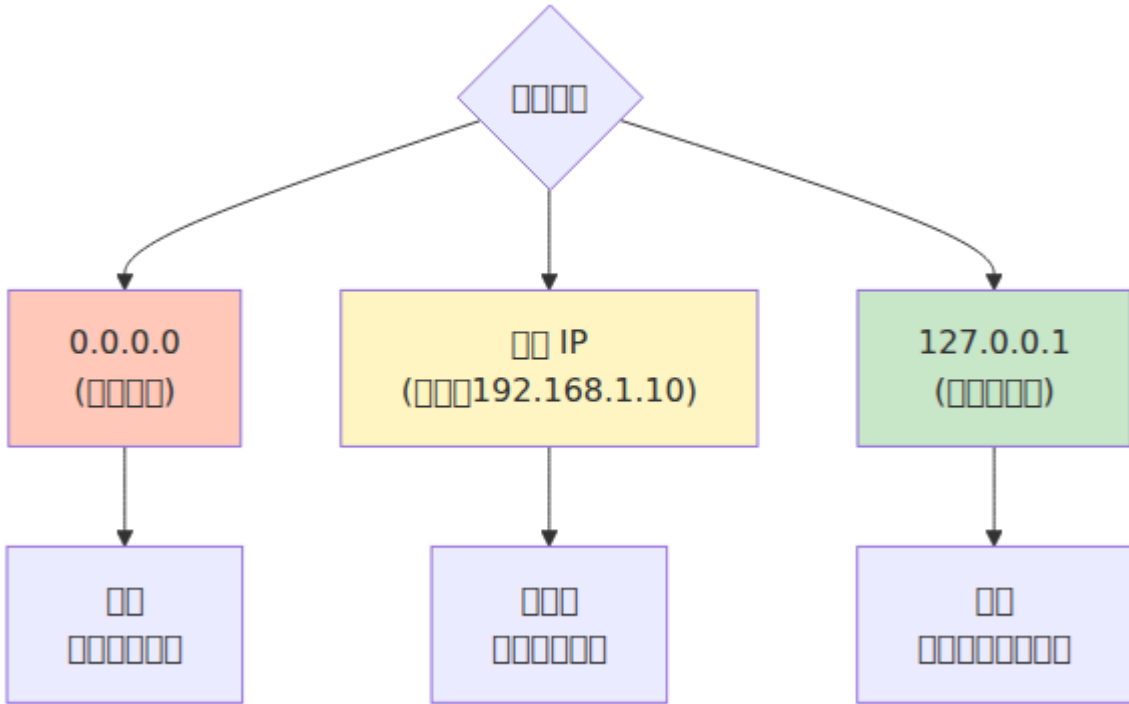
- 証明書 X.509 形式
- 証明書
- 証明書
- CN と SAN 証明書

証明書

```
https: [  
  port: 8443,  
  cipher_suite: :strong,  
  certfile: System.get_env("TLS_CERT_FILE",  
    "/etc/ssl/certs/omnihss.crt"),  
  keyfile: System.get_env("TLS_KEY_FILE",  
    "/etc/ssl/private/omnihss.key"),  
  cacertfile: System.get_env("TLS_CA_FILE", "/etc/ssl/certs/ca-  
bundle.crt")  
]
```

□□□□□□

□□□□□□



□□□□□□

□□□□

-
- PLMN□MCC□MNC□
- Diameter □□□□□
- Diameter □□ IP □□□
- API □□□□□□ TLS □□
- URL□□□□□◆◆□□□□□□□□
- HSS □□□□□send_clr_on_mme_change□stop_diameter_on_database_failure□

□□□□□□

- Diameter □□□□□□□□□□□□□□
- Diameter □□□□□□ 3868□
- HTTPS □□□□□□ 7443□8443□

Diameter DNS

IMS IMS

S-CSCF

S-CSCF

IMS

EIR

1.

2.

```
 https://[hostname]:7443  

```

3. **API**

```
curl -k https://[hostname]:8443/api/status
```

4. **Diameter**

```
 Diameter   

```

5. 数据库

数据库系统概论
数据库 SQL 语言



```
# config/runtime.exs - 配置数据库

import Config

#
=====
# 数据库配置
#
=====
config :hss, Hss.Repo,
  username: System.fetch_env!("DATABASE_USERNAME"),
  password: System.fetch_env!("DATABASE_PASSWORD"),
  hostname: System.get_env("DATABASE_HOSTNAME", "db.omnihss.internal"),
  database: System.get_env("DATABASE_NAME", "omnihss"),
  port: String.to_integer(System.get_env("DATABASE_PORT", "3306")),
  pool_size: String.to_integer(System.get_env("DATABASE_POOL_SIZE", "10")),
  timeout: 15_000,
  connect_timeout: 15_000,
  ssl: true,
  ssl_opts: [
    cacertfile: "/etc/ssl/certs/mysql-ca.pem",
    verify: :verify_peer
  ]

#
=====
# 许可证配置
#
=====
config :license_client,
  license_server_api_urls: [System.get_env("LICENSE_SERVER_URL",
"https://license.example.com:8443/api")],
  licensee: System.get_env("LICENSE_ORGANIZATION", "PLMN"),
  product_name: "omnihss"

#
=====
# 许可证配置
#
=====
```

```

config :hss,
  ecto_repos: [Hss.Repo],
  home_plmn: %{
    mcc: System.get_env("HOME_PLMN_MCC", "001"),
    mnc: System.get_env("HOME_PLMN_MNC", "001")
  },
  send_clr_on_mme_change: true,
  stop_diameter_on_database_failure: true,
  license_enforced: true,
  license_module: LicenseClient

#
=====
# Diameter []
#
=====
diameter_config = %{
  service_name: :omnitouch_hss,
  listen_ip: System.get_env("DIAMETER_LISTEN_IP", "10.7.25.186"),
  listen_port: String.to_integer(System.get_env("DIAMETER_LISTEN_PORT",
"3868")),
  host: System.get_env("DIAMETER_HOST", "omnihss01"),
  realm: System.get_env("DIAMETER_REALM",
"epc.mnc001.mcc001.3gppnetwork.org"),
  product_name: "OmniHSS",
  vendor_id: 10415,
  supported_vendor_ids: [5535, 10415],
  request_timeout: 5000,
  peers: [
    %{
      host: "mme01.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.7.25.100",
      port: 3868,
      transport: :sctp,
      applications: [:s6a]
    }
  ]
}

config :hss, :diameter, diameter_config

#
=====

```

```

# IMS []
#
=====
config :hss, :ims, %{
  scscf: %{
    selection_method: :random_peer,
    peers: [
      %{host: "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060"},
      %{host: "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060"}
    ]
  }
}

#
=====
# EIR []
#
=====
config :hss, :eir, %{
  unknown_equipment_behaviour: :whitelist
}

#
=====
# API []
#
=====
config :hss, HssWeb.Api.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8443],
  https: [
    port: 8443,
    cipher_suite: :strong,
    certfile: System.get_env("TLS_CERT_FILE", "/etc/ssl/certs/omnihss"),
    keyfile: System.get_env("TLS_KEY_FILE", "/etc/ssl/private/omnihss"),
  ],
  url: [host: System.get_env("API_HOST", "api.omnihss.internal"), port: 8443]

#
=====
# []
#
=====
config :hss, HssWeb.ControlPanel.Endpoint,

```

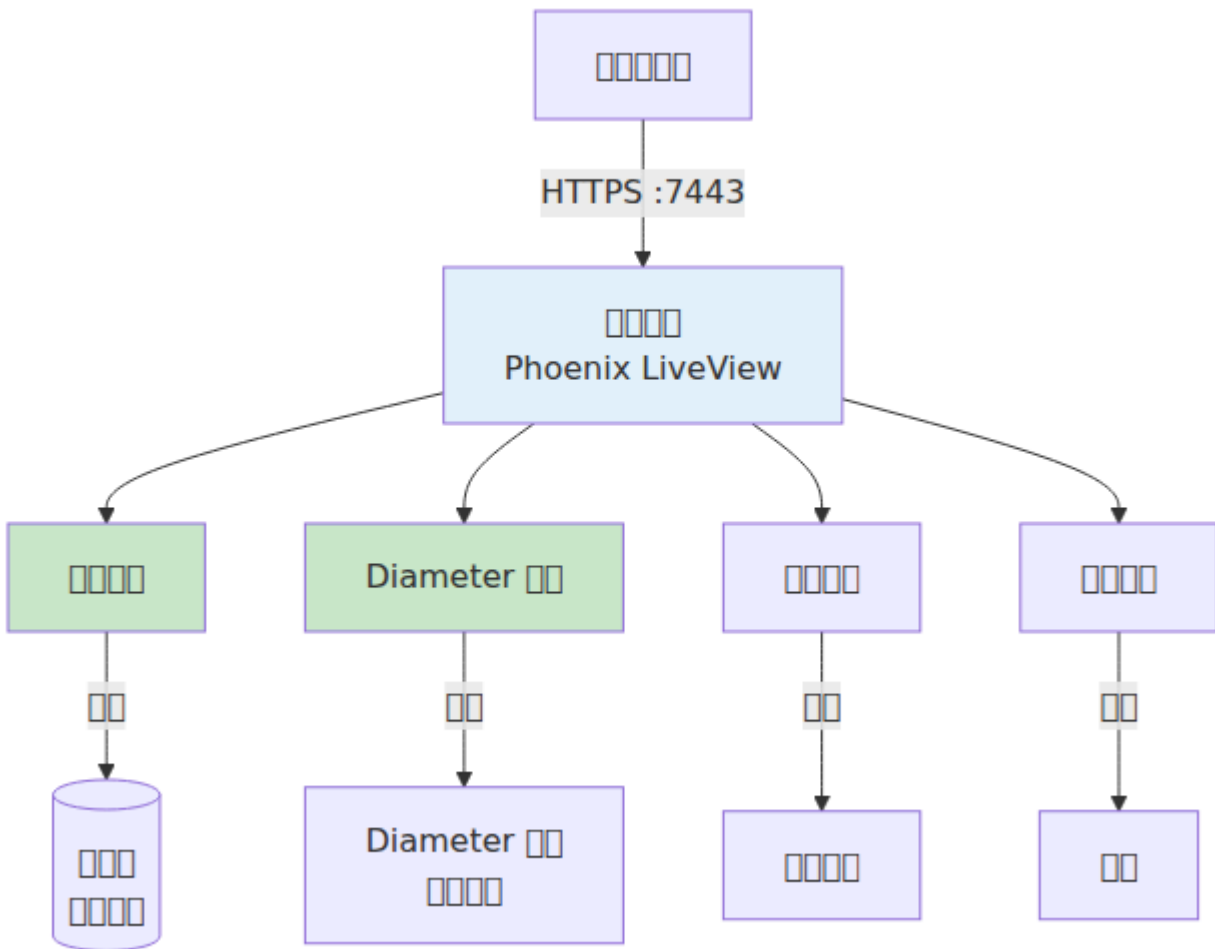
```
http: [ip: {0, 0, 0, 0}, port: 7443],
https: [
  port: 7443,
  cipher_suite: :strong,
  certfile: System.get_env("TLS_CERT_FILE", "/etc/ssl/certs/omnihss"),
  keyfile: System.get_env("TLS_KEY_FILE", "/etc/ssl/private/omnihss"),
],
url: [host: System.get_env("CP_HOST", "hss.omnihss.internal"), port
```

← □□□□□□ | □□□□□□□□ →

□□□□

URL: https://[hostname]:7443
Protocol: □□ HTTPS
Port: 7443□□□□□
Certificate: □ config/config.exs □□□

□□□□□□



□□□□□□

□□□□

1. □□□□□□

2. `https://[hostname]:7443`
3. TLS 証明書
4. 証明書

TLS 証明書

証明書

証明書

証明書

- **7443** 証明書
- **HTTPS** 証明書 - HTTP
- 証明書 7443

証明書

証明書 LiveView, WebSockets

- Chrome/Chromium
- Firefox
- Safari
- Edge

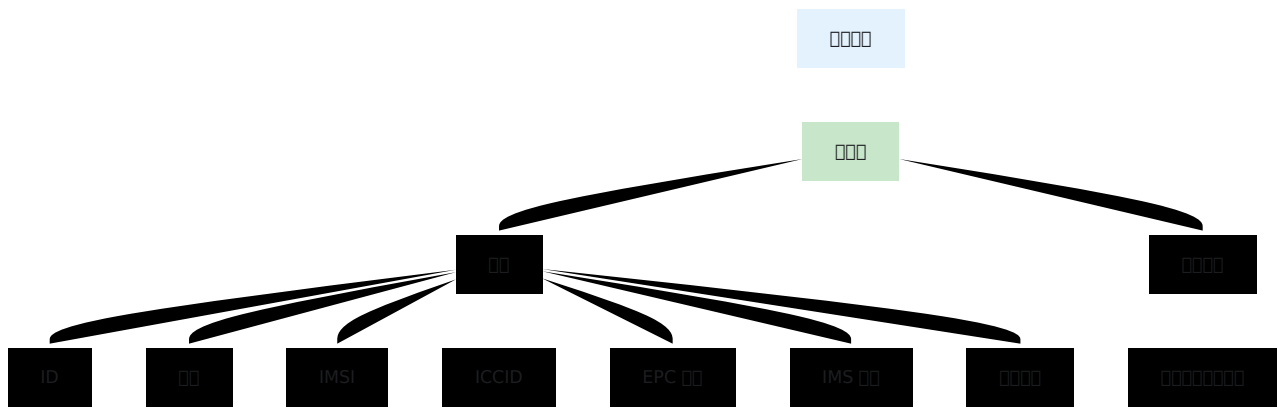
証明書 Internet Explorer

証明書

URL: `https://[hostname]:7443/overview`

証明書

□□□□



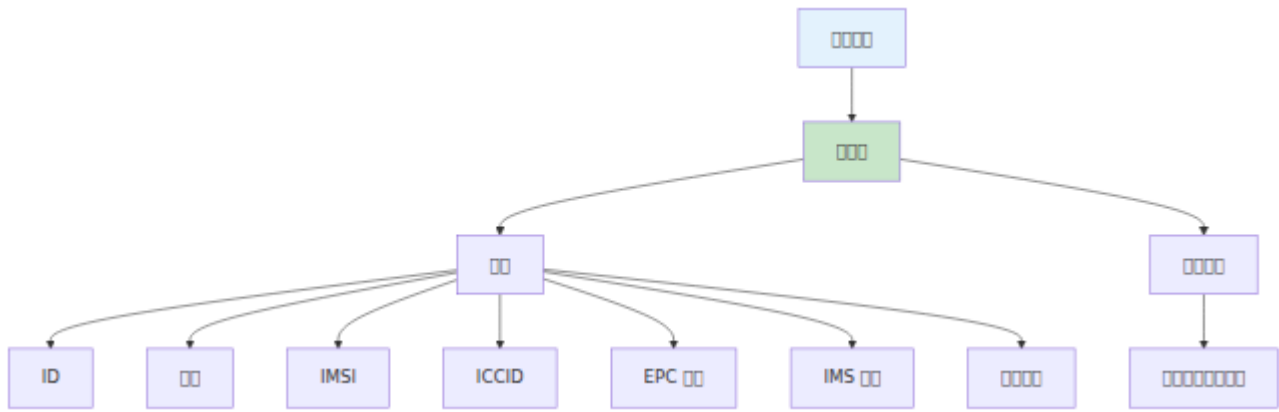
□□

□	□□	□
ID	□□□□□ ID	□□
□□	□□□□	✓□□□□/ X□□□□
IMSI	□□□□□□□□	14-15 □□□
ICCID	SIM □ ID	19-20 □□□□ "N/A"
EPC □□	□□□□□□□□	□□□□□ ID
IMS □□	□□□□□□□□	□□□□□ID □ "N/A"
□□□□	□□□□□□	□□□□□ID □ "N/A"

□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□



IMSI

- **MCC** - 3 digit
- **MNC** - 2-3 digit
- **TAC** - 3 digit
- **ID** - 5 digit
- **eNodeB ID** - 5 digit
- **ECI** - E-UTRAN 5 digit

MME

IMS

- **MME** - MME ID
- **MME ID** - MME ID Diameter ID
- **RAT** ID - "E-UTRAN" LTE
- **MME ID** - Diameter ID

IMS

IMS

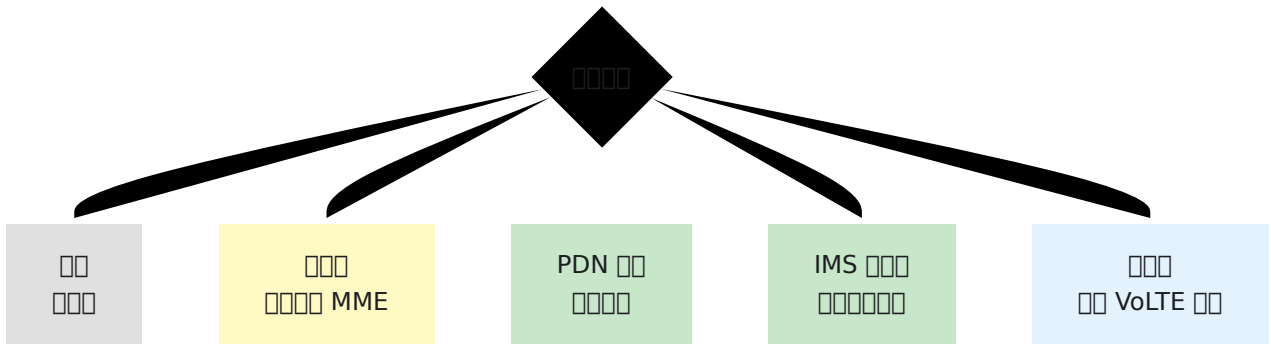
- **S-CSCF** - S-CSCF SIP URI
- **IMS** ID - SIP URI <sip:+14155551234@ims.example.com>
- **P-CSCF** - HSS ID P-CSCF
- **I-CSCF** - HSS ID I-CSCF

IMS

□□□

- **PDN** □□ - □□□□□□□□□□
- □□□□ - □□ VoLTE □□□□□

□□□□□□



□□□□□□□□

- □□□ □□□□□□□□□□ MME
- □□□□ □□□□□ MME □□□□□□□□□□
- **PDN** □□□□ PDN □□□□□ > 0
- **IMS** □□□□ □□□□ S-CSCF □□□
- □□□□ □□□□□□□ > 0

□□□□□

□□□□□ **1** □ □□□□□□□□□□□□□□□□□

□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□

□□□□□

1. □□□□□□□□

- □□□□□□□□□□

- 网络架构
- IMS 网络

2. 网络

- 网络架构
- 网络架构
- 网络架构
- 网络架构

3. 网络

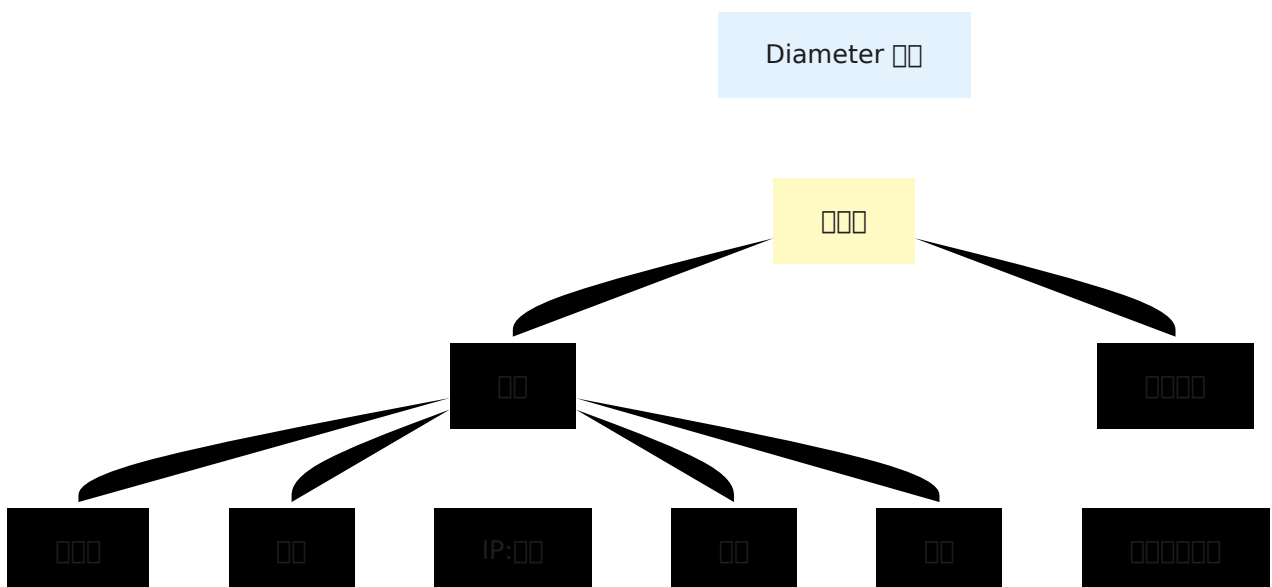
- 网络架构
- PDN 网络
- VoLTE 网络

Diameter 网络

URL: `https://[hostname]:7443/diameter`

Diameter 网络 Diameter 网络

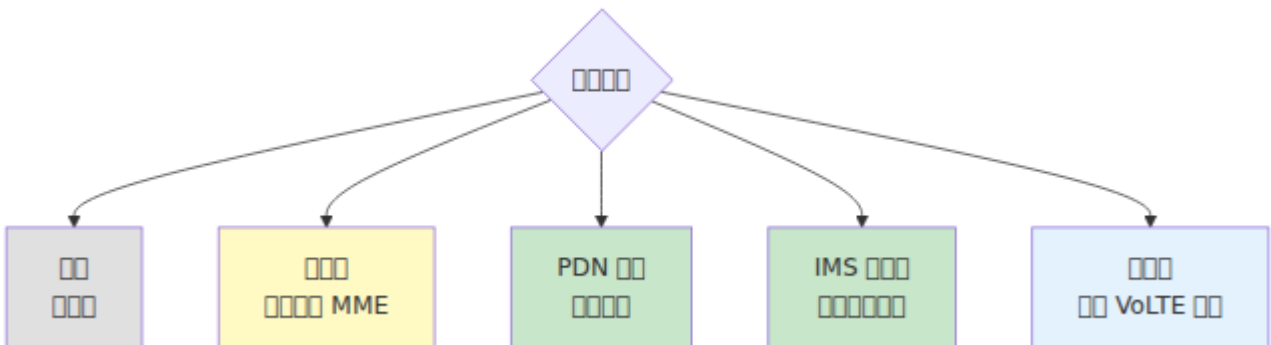
网络



□□

□	□□	□
□□□	Diameter □□□□□	FQDN
□□	Diameter □□	□□
IP:□□	□□□□	IP □□□□□
□□	□□□□	TCP □ SCTP
□□	□□□□	□□□ / □□□

□□□□



□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□

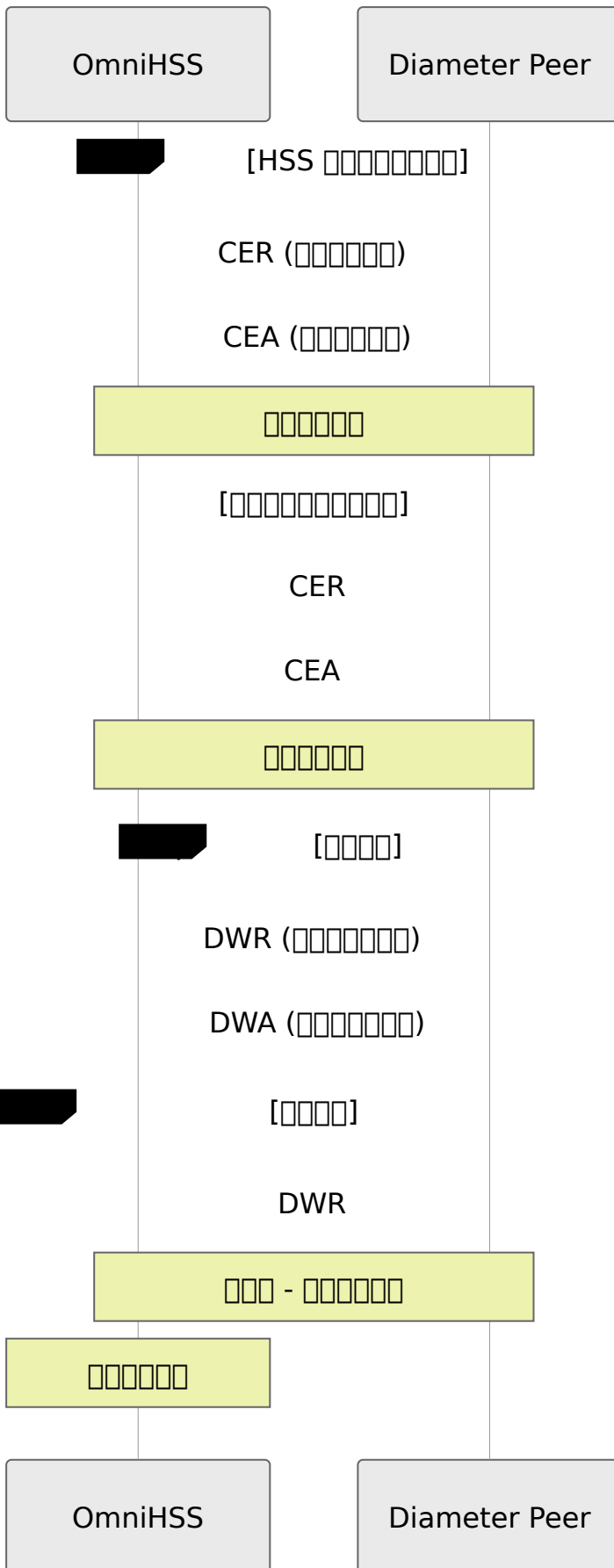
- □□□□ - □ HSS □□□□□□
- □□□□ - □□□□□□□□□□
- □□ **ID** - □□□□ Diameter □□

□□ **ID** □□□

- 16777251 - S6a (MME)

- 16777238 - Gx (P-GW)
- 16777216 - Cx (I-CSCF, S-CSCF)
- 16777217 - Sh (□□□□□)
- 16777236 - Rx (P-CSCF)
- 16777252 - S13 (EIR □□□□□□□□)

□□□□□□



□□□□

Diameter □□□ **1** □ □□□□□

□□□□

1. □□□□□

- □□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□

2. □□□□

- □□□□□□□□
- □□□□□□□□ TCP □ SCTP□
- □□□□ ID □□□□□□
- □□□□□□□□□□□

3. □□□□

- □□□□□□□□□
- □□□□□□□
- □□□□□□□□□

□□□□

□□□□□□□

□□□□□

1. □□□□□□

2. □□□□□□□□□□□

3. □□□□□□□

4. Diameter □□□□□

5. □□□□□□□□□ TLS□

□□□□□□□

1. ping [peer-ip]
2. telnet [peer-ip] 3868
- 3.
4. HSS
5. Diameter HSS

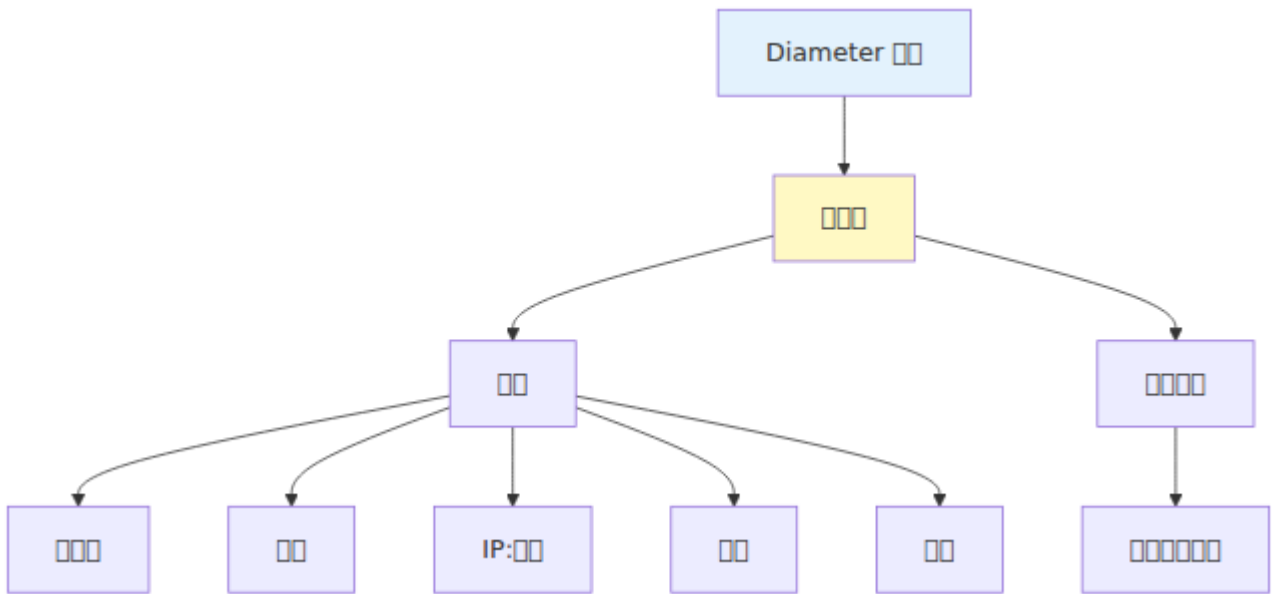
- 1.
- 2.
- 3.
4. Diameter

- 1.
- 2.
- 3.
4. ID

URL: https://[hostname]:7443/application

- Erlang VM
-
- OmniHSS
- **Erlang VM**

□□□□



□□□□

1. □□□□

- □□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□

2. □□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

3. □□□□

- □□□□□□□
 - □□□□□□□□
 - □□ Erlang VM □□
-

□□□□

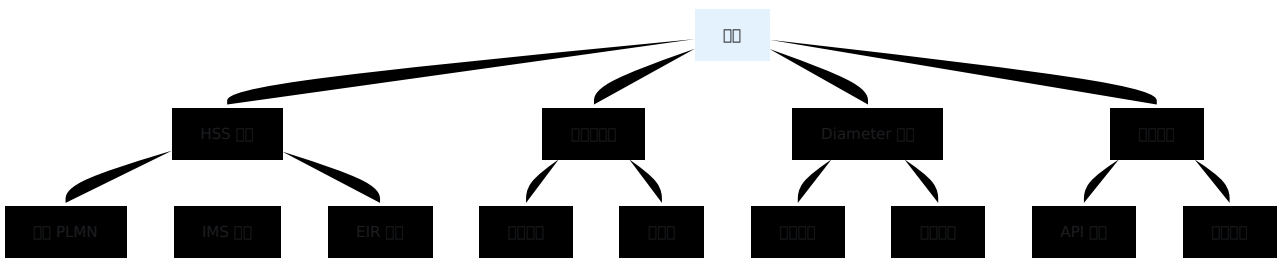
URL: `https://[hostname]:7443/configuration`

□□□□□ OmniHSS □□□□□□□□□□

□□

- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□

□□□□□



□□□□□

1. □□□□

- □□ runtime.exs □□□□□□□□□□
- □□□□□□□□□□□□
- □□ Diameter □□□□□

2. □□□□

- □□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

3. □□

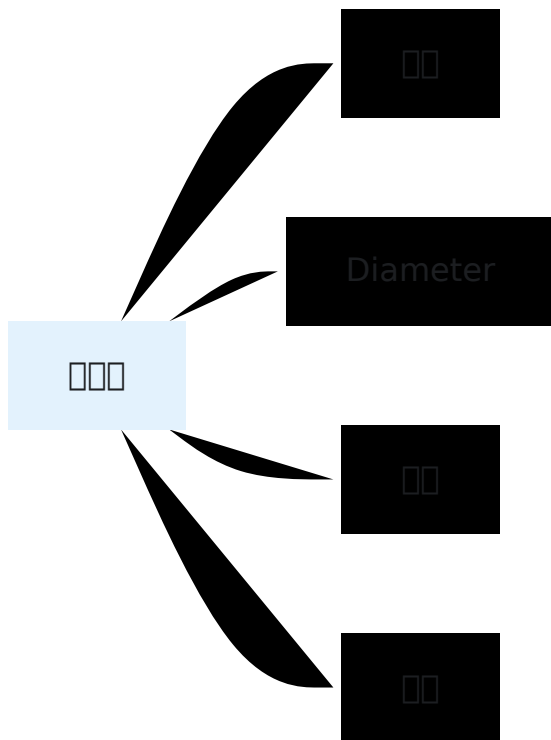
- □□□□□□□□□□□□□□□□

- 設定ファイル

設定ファイルの読み込みと保存の方法について説明します。

設定ファイル

設定ファイル



設定ファイルの読み込みと保存の方法について説明します。

設定ファイル

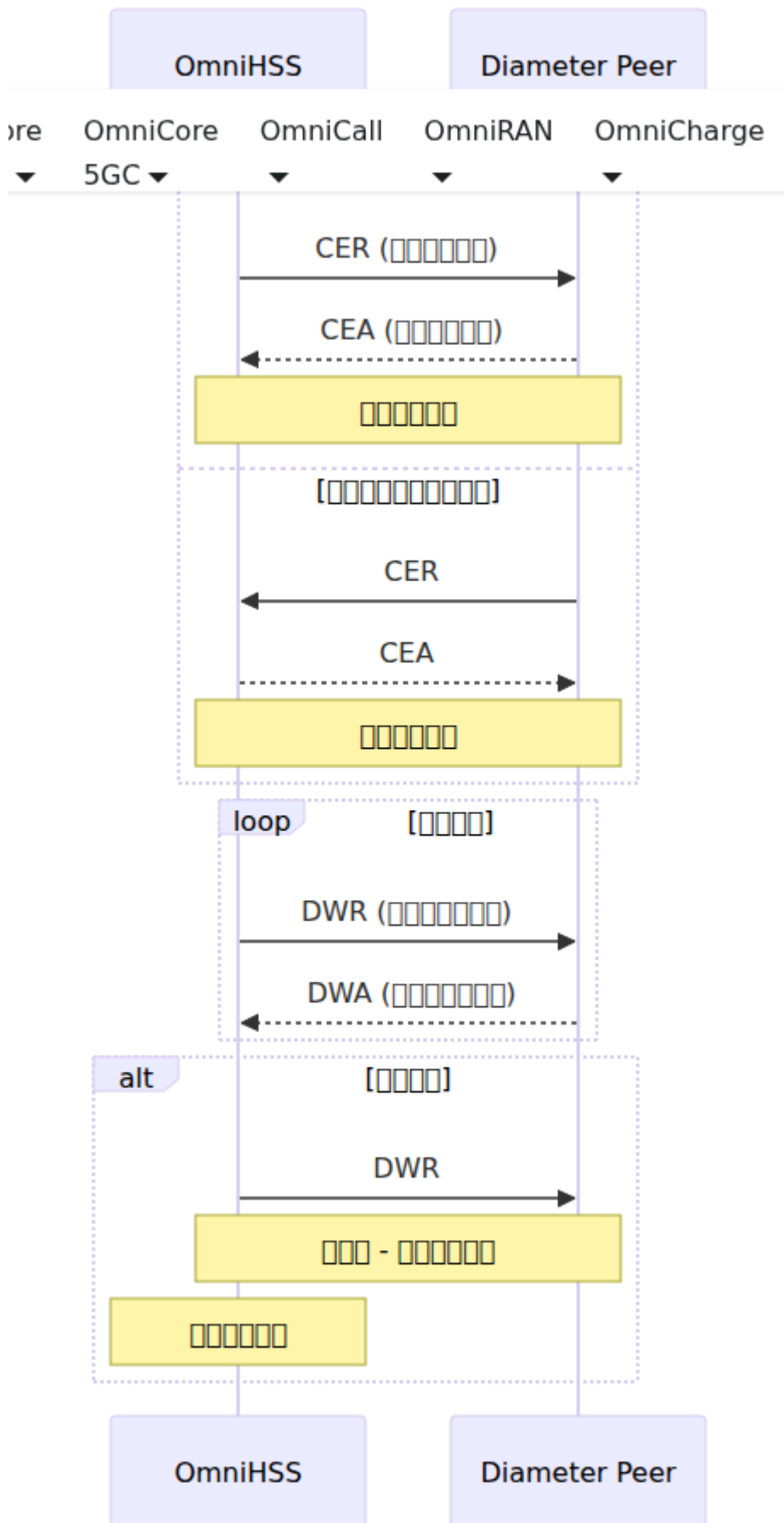
設定ファイルの読み込みと保存の方法について説明します。

- **Ctrl+R / F5** - 設定ファイルの読み込み
- **Ctrl+F** - 設定ファイルの検索
- **Ctrl+T** - 設定ファイルの保存

- □□□□□□□□□□

- □□□□□

□□□□□□□□



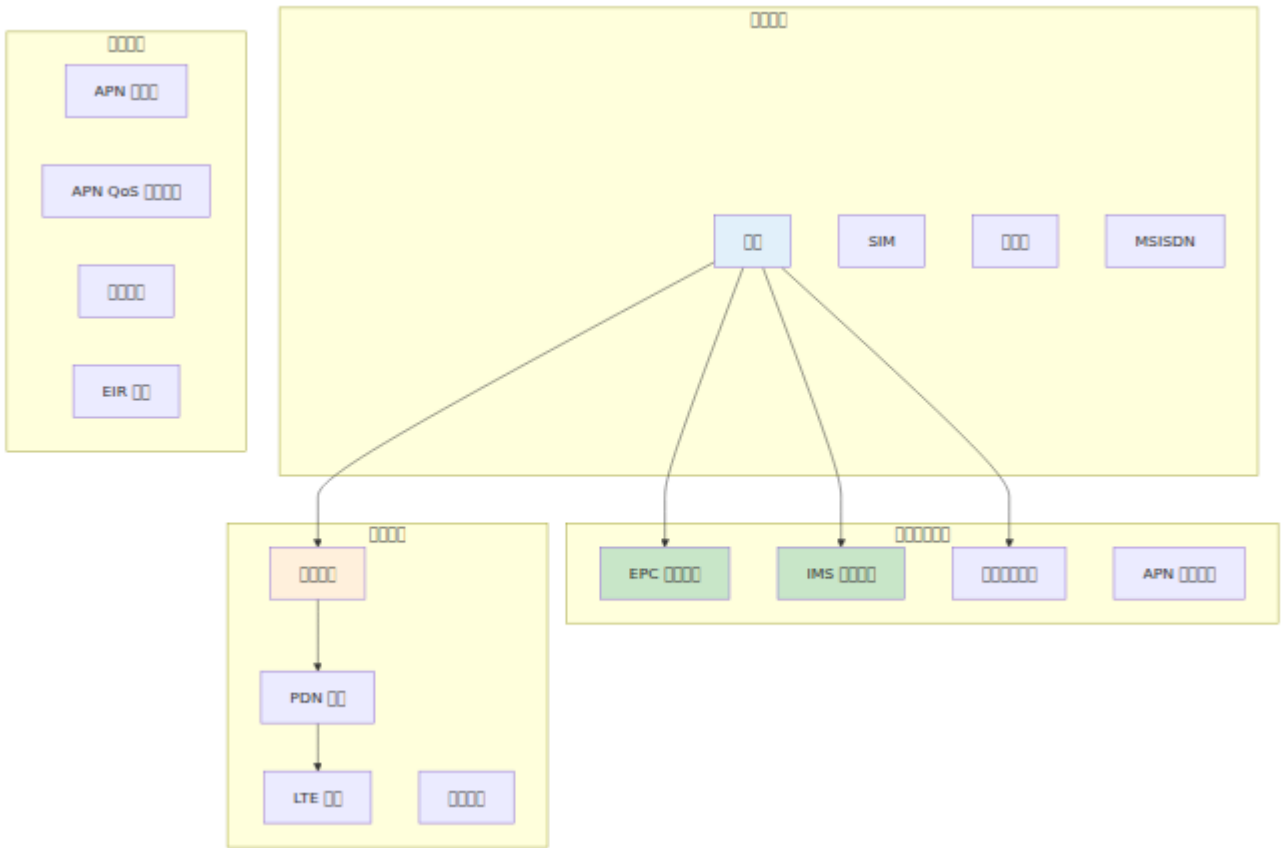
□□□□

□□□□□□□□□□□□□□

□□	□□	□□
□□□□ Diameter □□	1 □□	2+ □□□□□□
□□□□	> 80%	> 90%
□□□□□□	> 5%	> 10%
□□□□	> 80% □□□	> 95% □□□

← □□□□□□ | □□□□□□□□□□ →

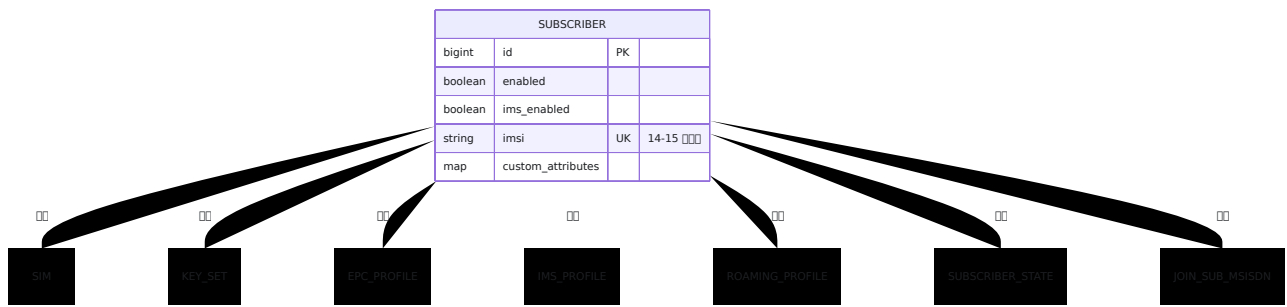
□□□□



□□□□

□□

□□□□□□□□□□□□□□



□□:

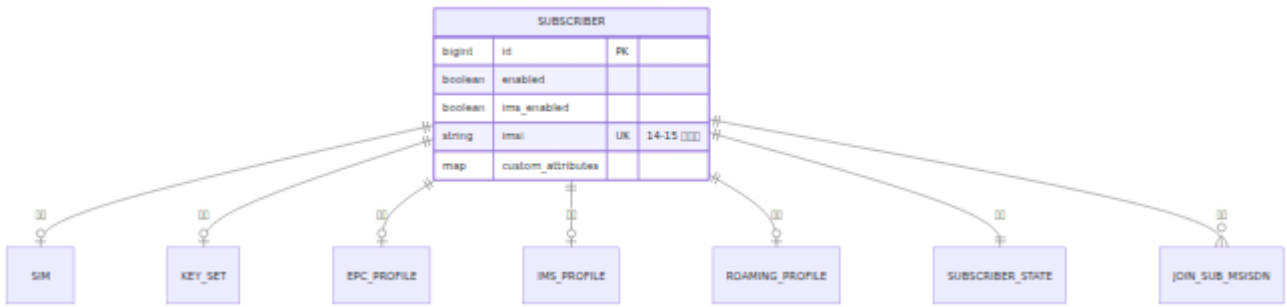
欄名	型別	説明	デフォルト値
id	bigint	ID	
enabled	boolean	有効/無効	enabled: true
ims_enabled	boolean	IMS 有効/無効	ims_enabled: true
imsi	string	国際移動番号	14-15 桁の数字
custom_attributes	map	カスタム属性	
sim_id	bigint	SIM の ID	
key_set_id	bigint	キーセット ID	
epc_profile_id	bigint	EPC プロファイル ID	
ims_profile_id	bigint	IMS プロファイル ID	
roaming_profile_id	bigint	ローミングプロファイル ID	
subscriber_state_id	bigint	サブスクリバースタート ID	

注:

- 国際移動番号 (IMSI)
- IMSI の 14-15 桁は国際移動番号 (MSISDN) と一致する
- 有効/無効 (enabled) は IMS が有効かどうかを示す
- IMS 有効/無効 (ims_enabled) は IMS が有効かどうかを示す

SIM

SIM は SIM カードの ID



📄:

📄	📄	📄	📄📄📄📄
iccid	string	📄📄📄📄 ID	📄
sim_vendor	string	SIM 📄📄	📄
batch_name	string	📄📄📄	📄
is_esim	boolean	📄📄 SIM 📄	📄
pin1, pin2	string	PIN 📄	📄
puk1, puk2	string	PUK 📄	📄
adm1 - adm10	string	📄📄📄	📄📄📄
kic, kid	binary	OTA 📄📄📄	📄📄📄

📄📄:

- ICCID 📄📄📄 SIM 📄
- 📄 SIM 📄📄📄📄📄📄📄📄
- PIN/PUK 📄📄📄📄📄 SIM 📄
- ADM 📄📄📄📄 SIM 📄
- KIC/KID 📄 SIM OTA📄📄📄📄📄

□□□

□□□□□□□□□□□□□□□□

KEY_SET			
bigint	id	PK	
binary	ki		128 □
binary	opc		128 □
binary	op		128 □
binary	amf		16 □
bigint	sqn		48 □□□
string	authentication_algorithm		

□□□



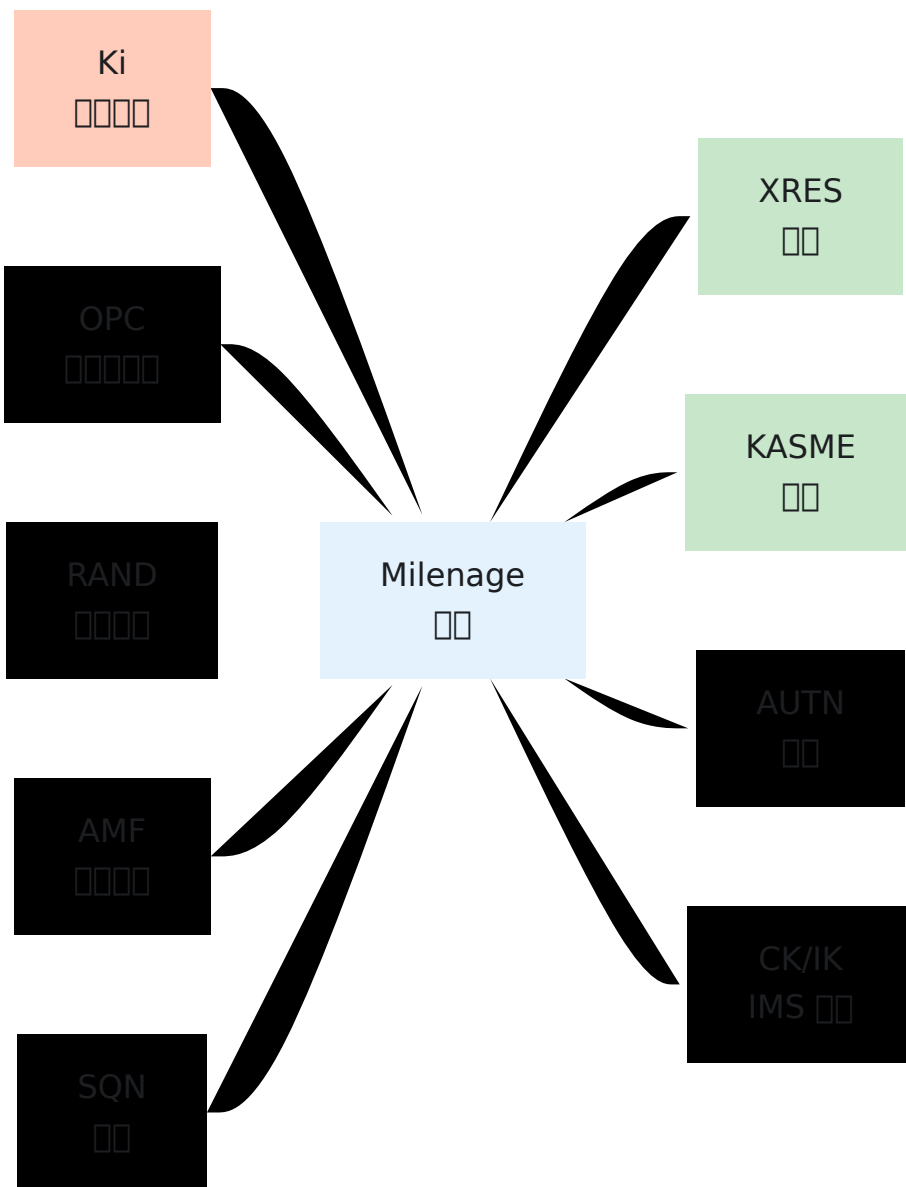
□□:

Field	Type	Description	Length
ki	binary	Key	128 bits
opc	binary	OPC	128 bits
op	binary	OPC	128 bits
amf	binary	Authentication Method	16 bits
sqn	bigint	Sequence Number	48 bits
authentication_algorithm	string	Authentication Algorithm	"milenage"
ota_counter	bigint	OTA Counter	16 bits

Fields:

- Key
- Ki from SIM
- OPC
- SQN
- Milenage

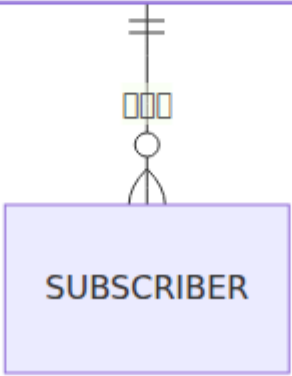
Fields:



MSISDN

MSISDN □□□□□□□□□□

KEY_SET			
bigint	id	PK	
binary	ki		128 []
binary	opc		128 []
binary	op		128 []
binary	amf		16 []
bigint	sqn		48 [] []
string	authentication_algorithm		



□□:

□□	□□	□□	□□
msisdn	string	□□□ ISDN □□	1-15 □□□□E.164 □□

□□□:

- MSISDN □□□□□□□□□□
- □□ MSISDN □□□□□□□□□□
- □□ MSISDN □□□□□□□□□□
- □□□□□□□ + □□□□□□□□ — □□ + □□□□□□ 1415551234 □□□ +1415551234 □
- □ Diameter Cx/Rx □□□□S-CSCF □ P-CSCF □□□□□□□ + □ SIP URI □□+ □□□□□□□□

□ **MSISDN** □□:

☐☐
IMSI: 001001123456789

MSISDN: 14155551001
☐☐

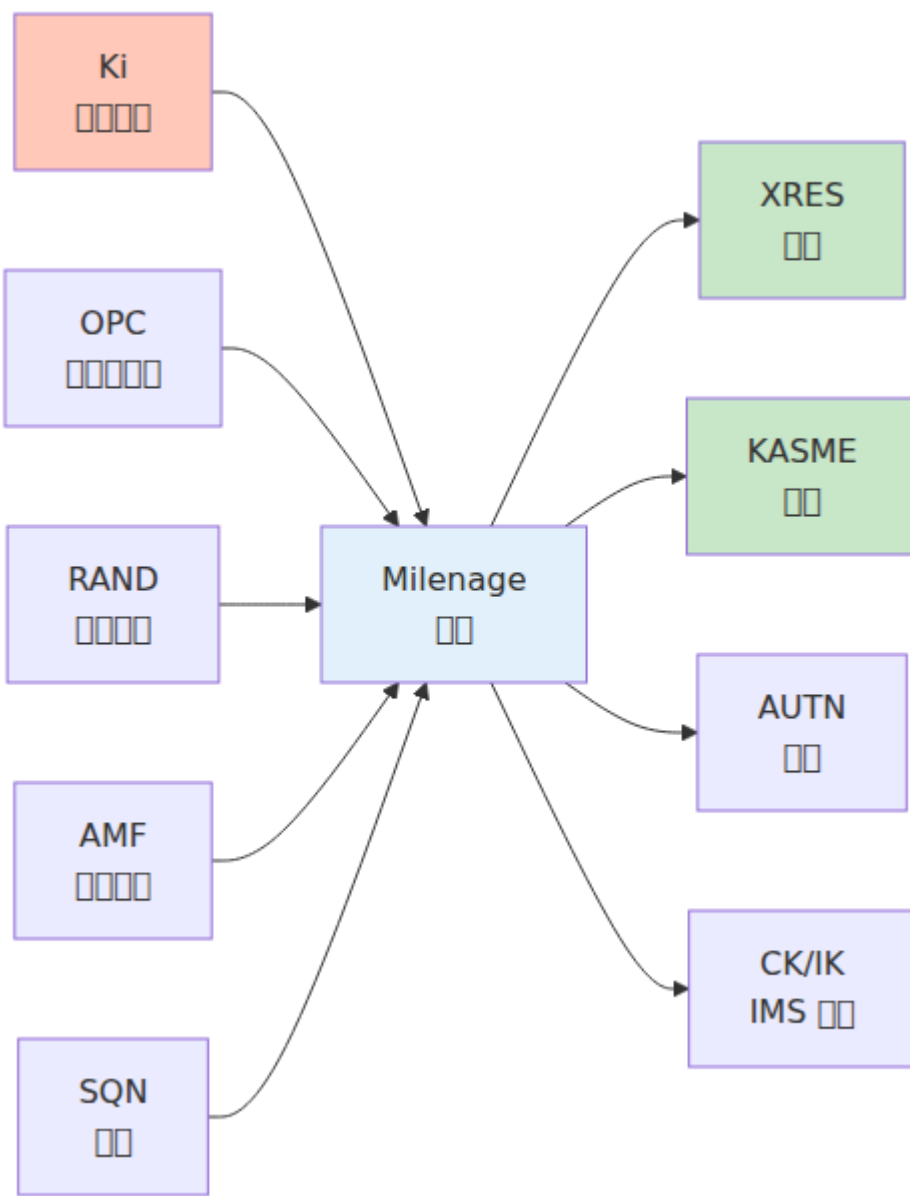
MSISDN: 14155551002
☐☐

MSISDN: 14155551003
☐☐

☐☐☐☐☐☐

EPC ☐☐☐☐

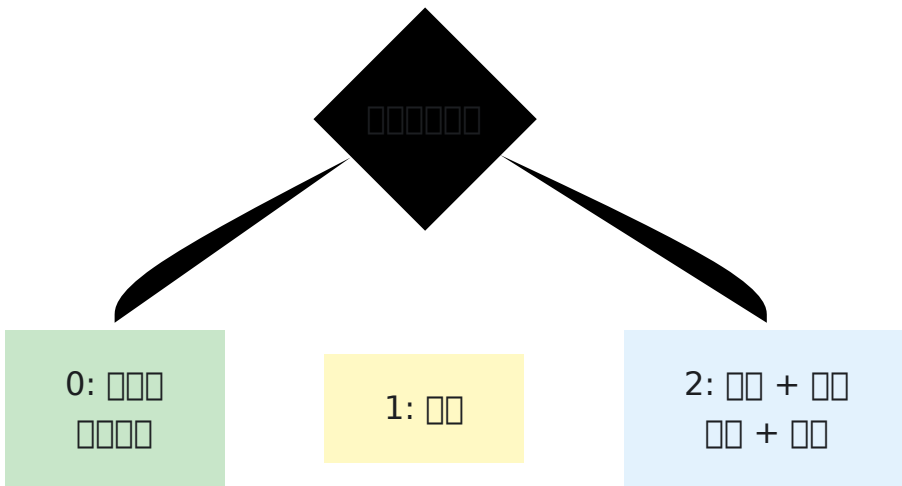
EPC ☐☐☐☐☐☐ LTE ☐☐☐☐☐☐☐☐



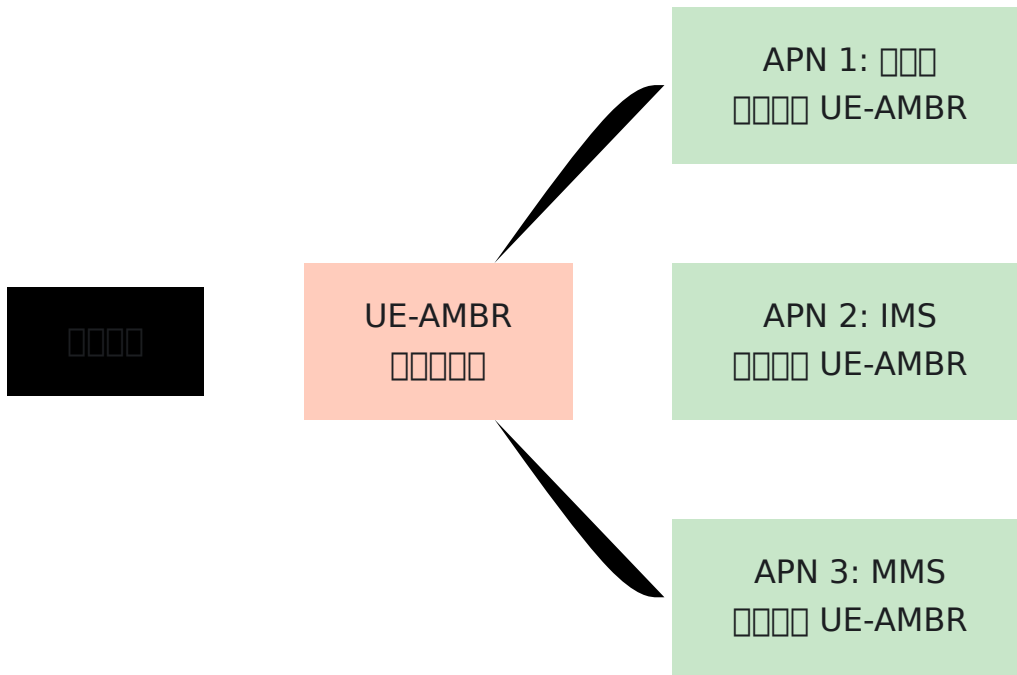
□□:

属性名	属性種別	属性値	属性単位
name	string	文字列 文字	文字
ue_ambr_dl_kbps	integer	整数 文字	Kbps
ue_ambr_ul_kbps	integer	整数 文字	Kbps
network_access_mode	string	文字列	"文字" または "文字 文字"
tracking_area_update_interval_seconds	integer	TAU 文字 文字	文字

属性値:

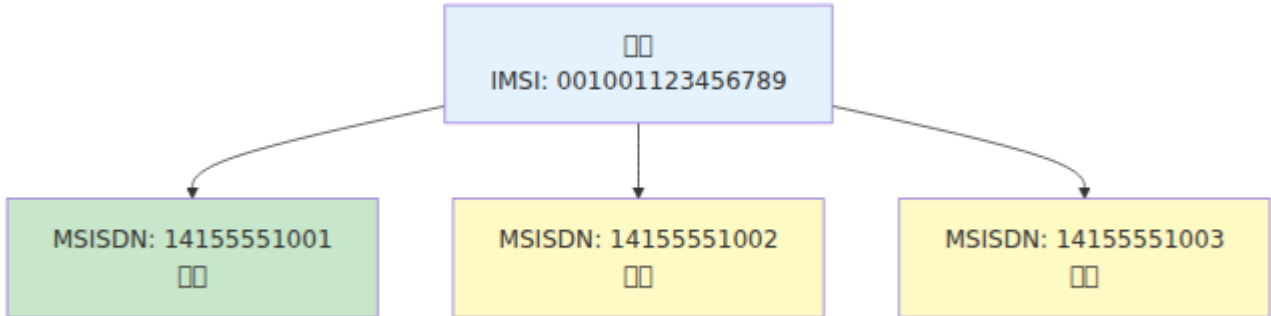


AMBR属性値:



IMS

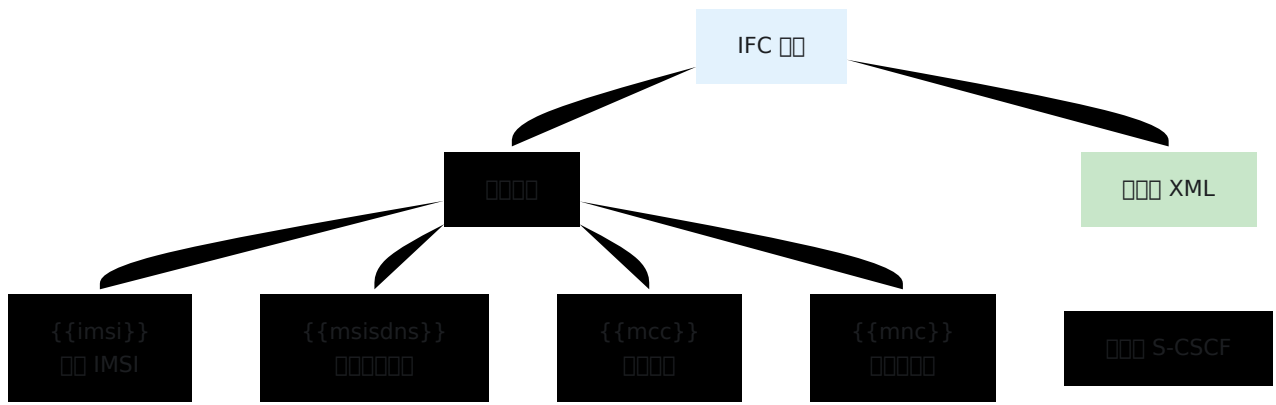
IMS



:

name	string		
ifc_template	text	XML	XML

IFC

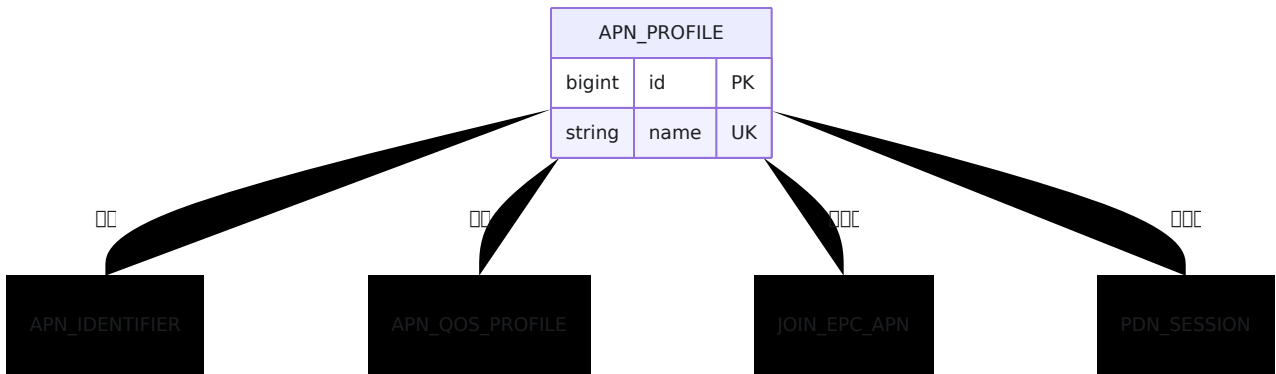


IMS:

- IFC IMS
- IMS
- IMS
- IMS S-CSCF

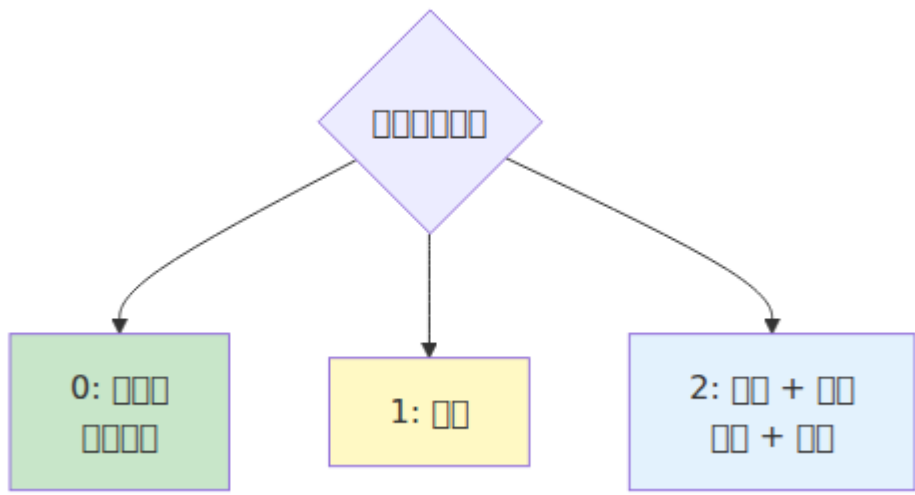
APN

APN



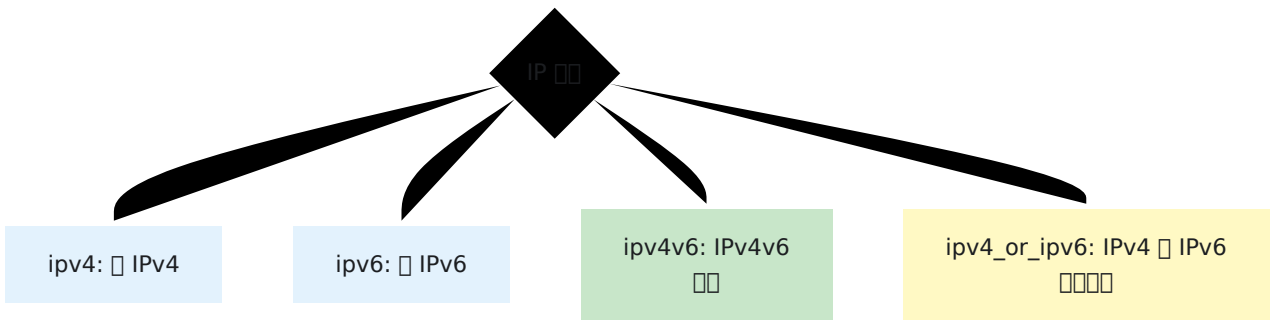
IMS:

APN

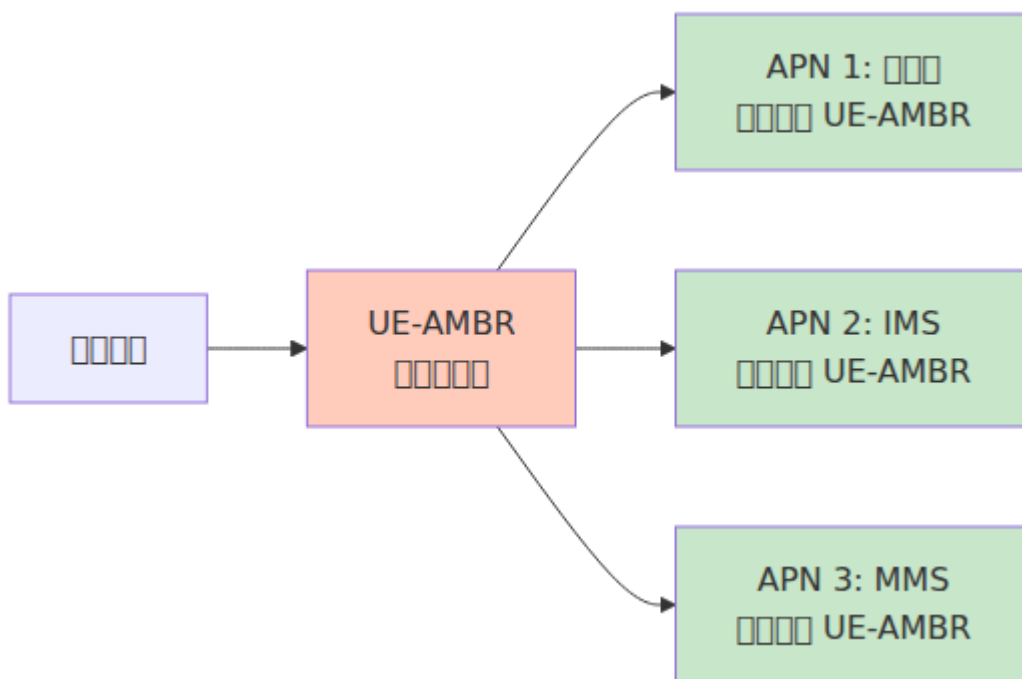


00	00	00	00
apn	string	APN 00	"internet", "ims", "mms"
ip_version	string	IP 0000	000

IP 0000:



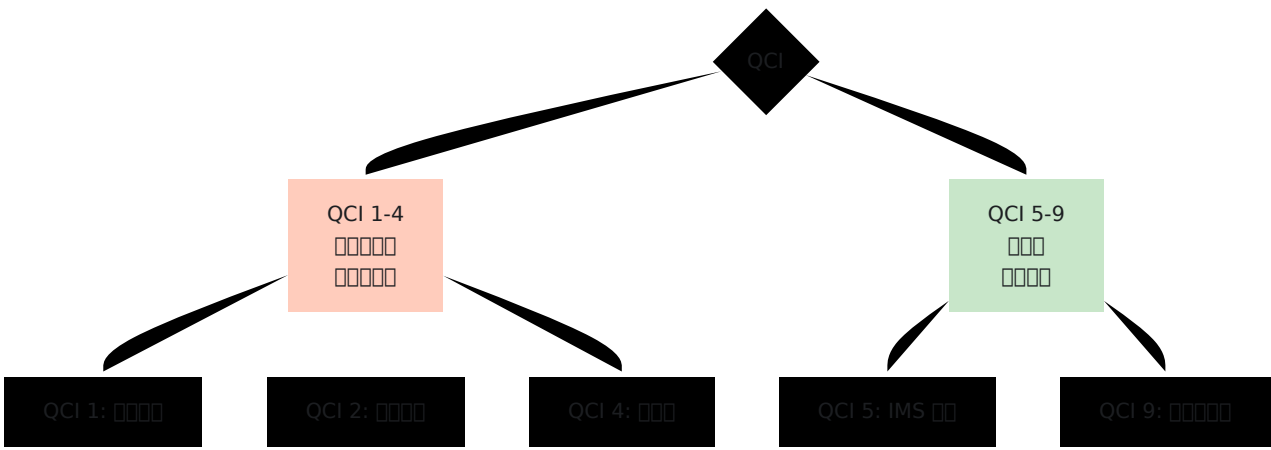
APN QoS 0000



QoS

QoS	QoS	QoS	QoS
qci	QoS	1-9	QCI 9
allocation_retention_priority	ARP	1-15	8
apn_ambr_dl_kbps	APN	0+	
apn_ambr_ul_kbps	APN	0+	
pre_emption_capability		true/false	false
pre_emption_vulnerability		true/false	true

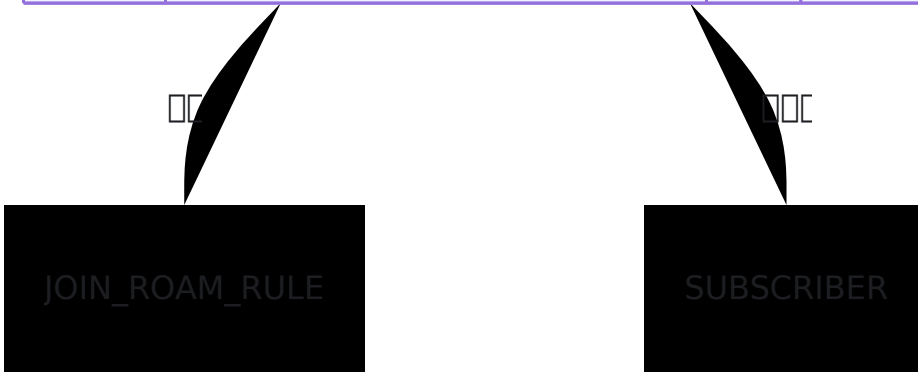
QCI



[][][][]

[][][][][][][][][][][][]

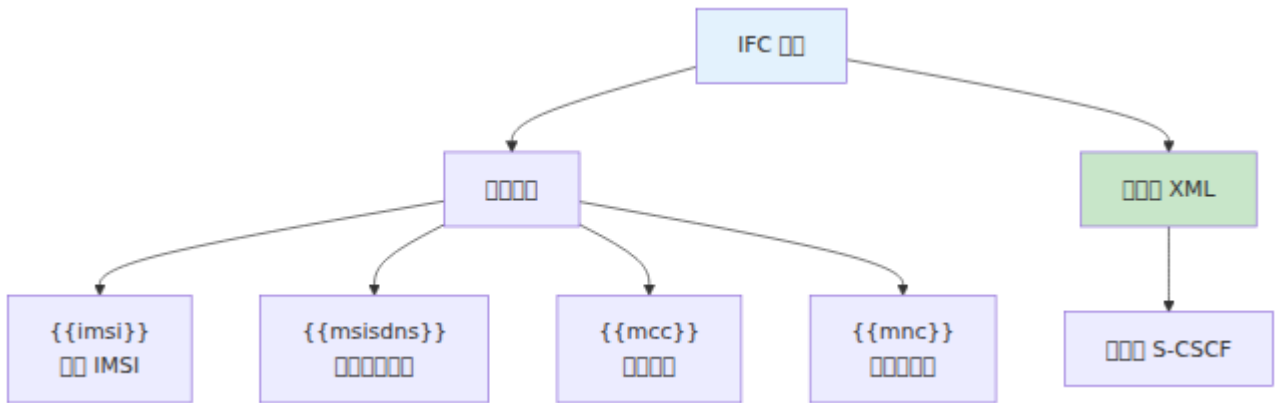
ROAMING_PROFILE			
bigint	id	PK	
string	name	UK	
string	data_action_if_no_rules_match		[][][]
string	ims_action_if_no_rules_match		[][][]



[][]

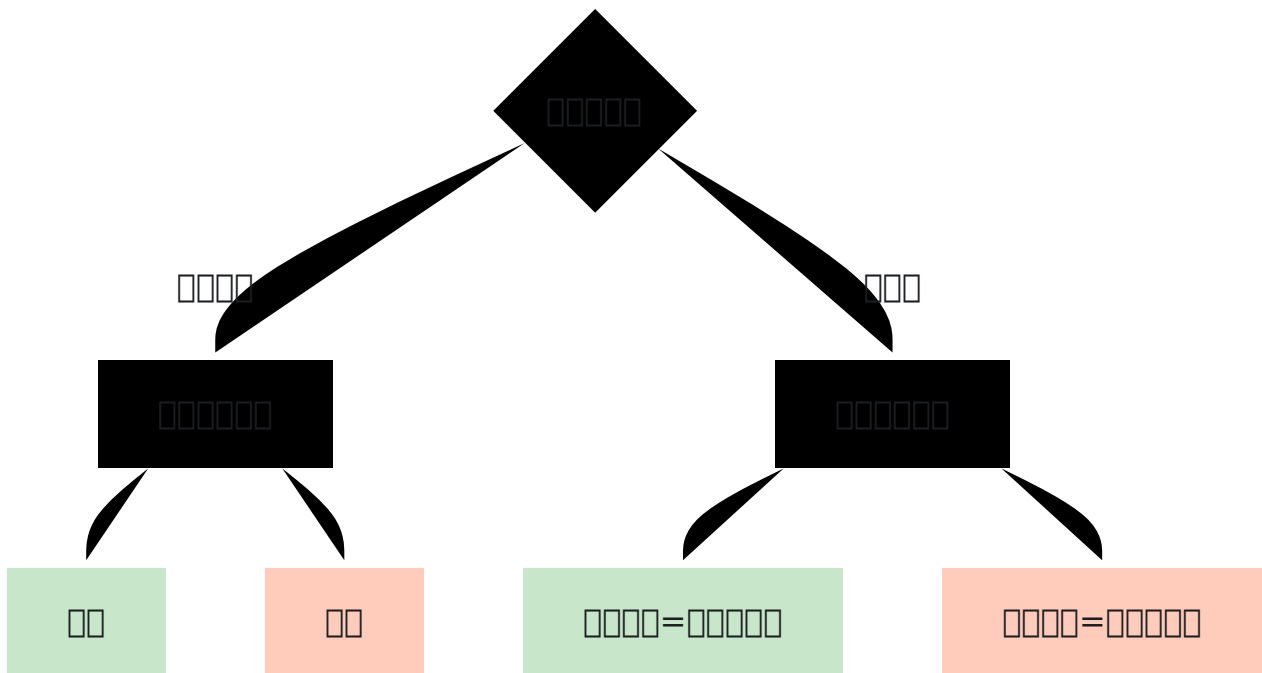
ROAMING_RULE

□□□□:



□□□□:

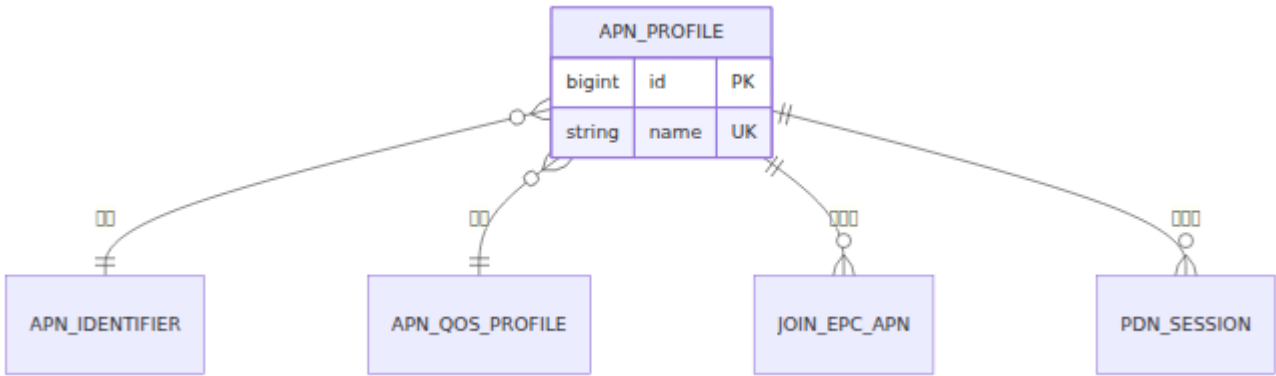
□□□□□□
MCC: 310, MNC: 410



□□□□

□□□□

□□□□□□□□□□□□



□□□□:

□□□□:

- last_seen_mcc, last_seen_mnc - □□□□□
- last_seen_tac - □□□□□□
- last_seen_cell_id - □□ ID
- last_seen_enodeb_id - eNodeB ID
- last_seen_eci - E-UTRAN □□□□□

□□□□:

- last_seen_mme - □□□□□□□ MME
- last_seen_realm - MME □ Diameter □□
- last_seen_rat_type - □□□□□□□□LTE□5G □□

IMS □□:

- assigned_scscf - □□□□□□□ S-CSCF
- ims_public_identity - SIP URI□□□□□
sip:+14155551234@ims.example.com□
- sh_repository_data - □□□ IMS □□□□□□□

□□□:

- last_seen_at - Diameter
- last*_at

PDN

PDN

PDN_SESSION		
bigint	id	PK
string	pgw_session_id	
integer	rat_type	
string	ip_address	
string	assigned_pgw_host	
boolean	emergency	
boolean	roaming	
datetime	created_at	



PDN



Gx CCR-I

Creating

Gx CCA-I ()

Active

Gx CCR-U Gx CCA-U

Modified

XXXXXXXXXX

Gx CCR-T

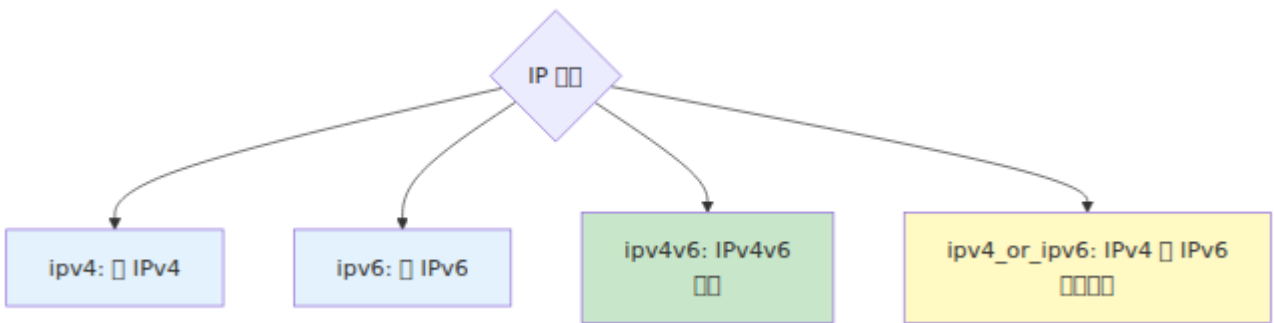
Terminating

Gx CCA-T

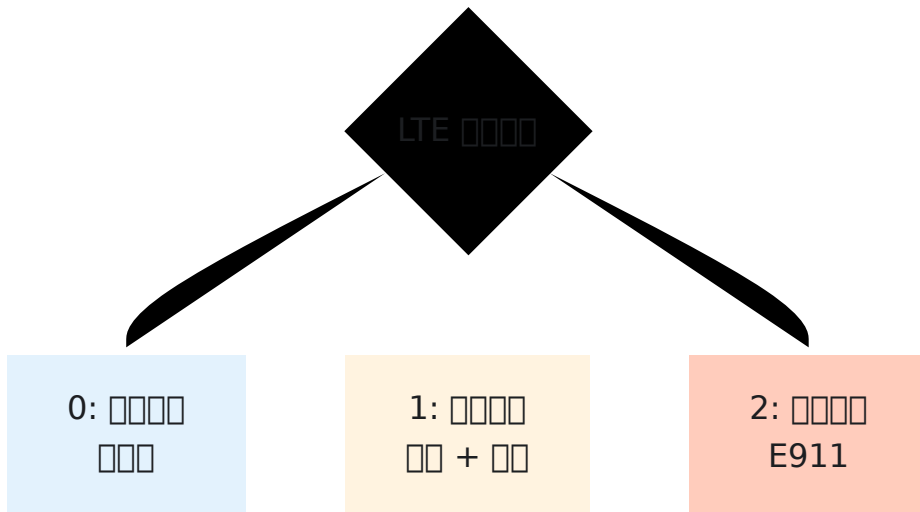


LTE

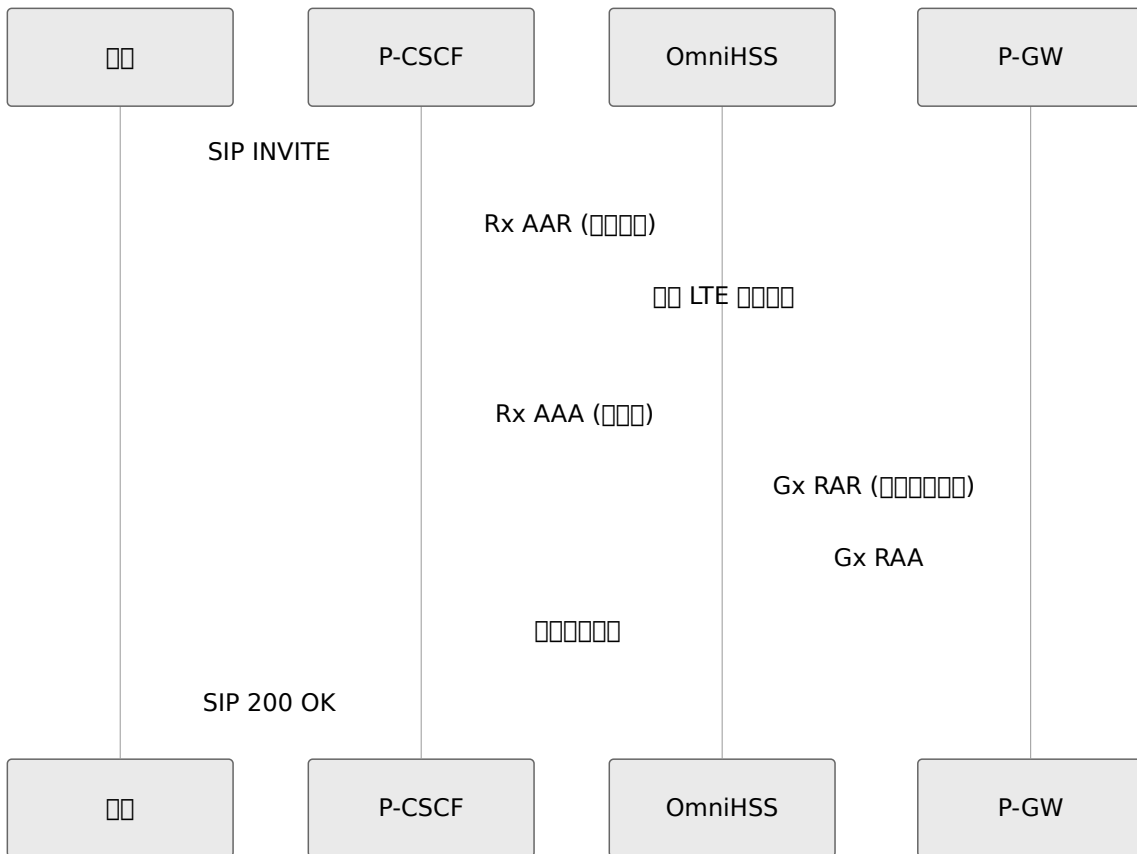
LTE VoLTE



XXXX:

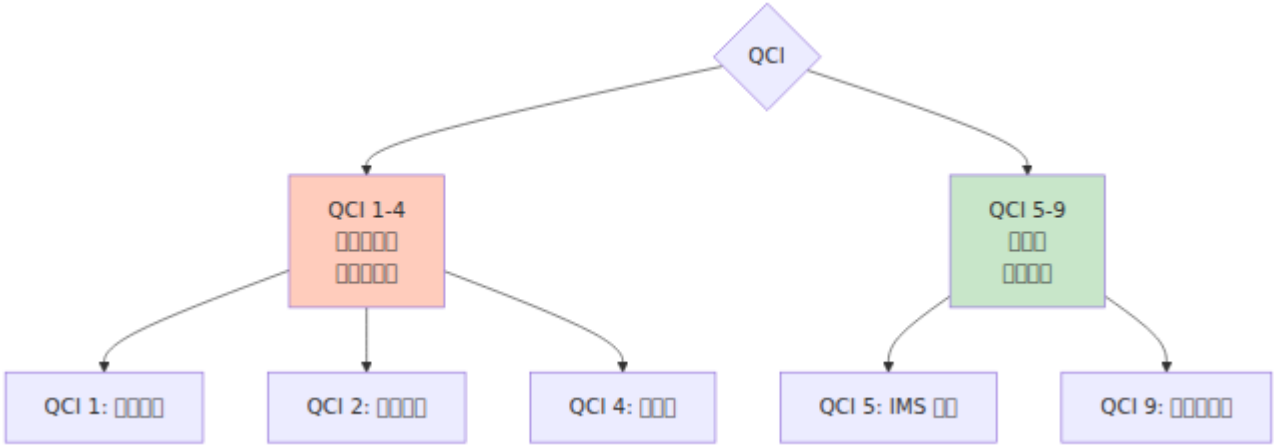


VoLTE 网络:



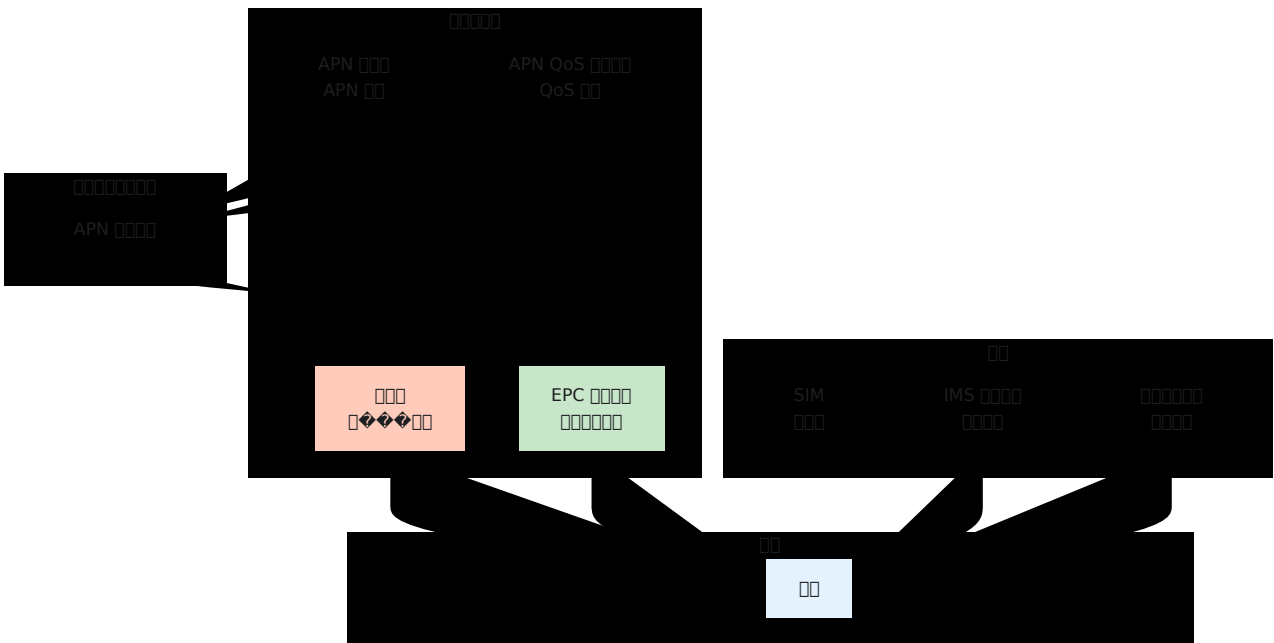
□□□□□

□□□□□□

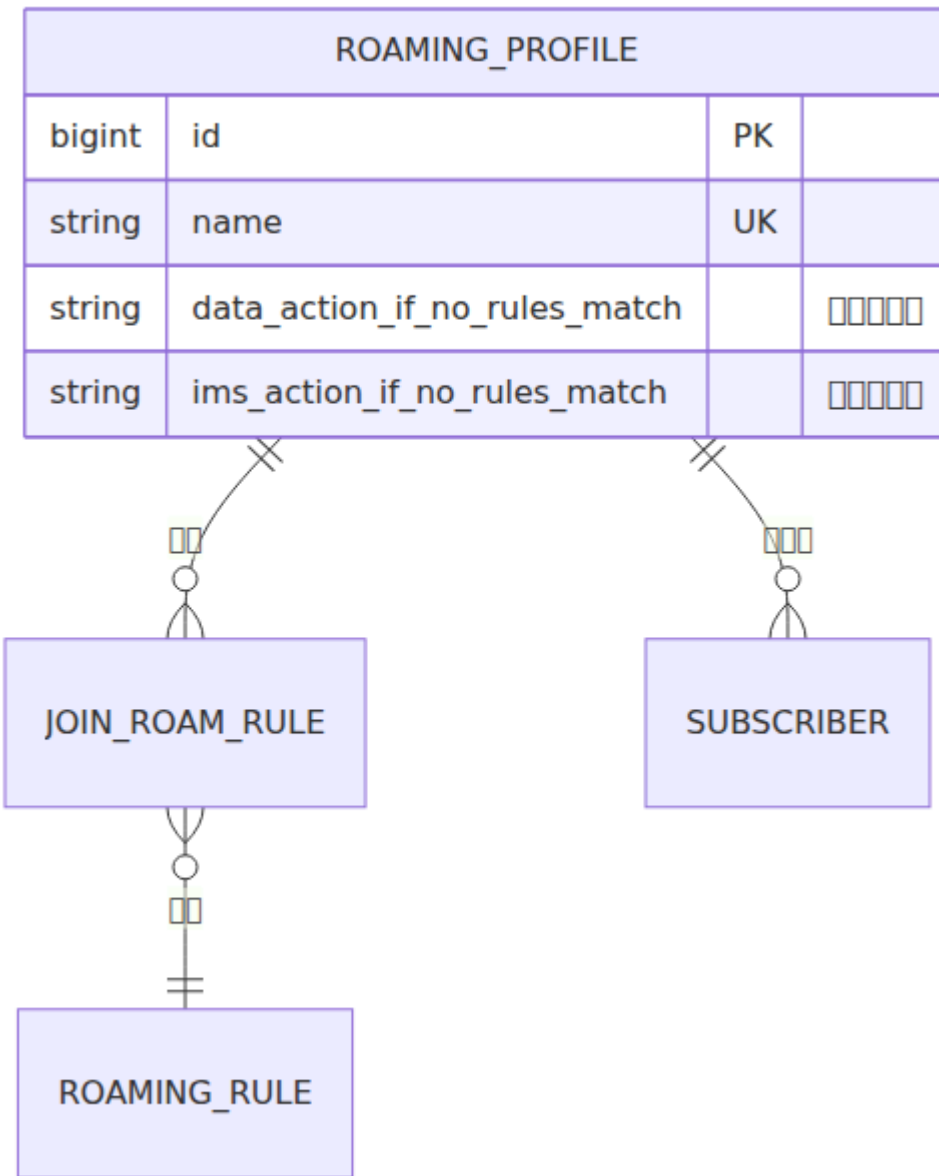


□□□□

□□□□□□□□□□□□□□□□



□□□□□□



□□□□□□

□□□□□□□□



□□□□□□

Create_Prerequisites

□□□□□□□□

□□□□□□
 □□ EPC □□□□□□ APN□
 □□ IMS □□□□□□□□
 □□□□□□□□□□□□
 □□ SIM□□□□□

Create_Subscriber

enabled=false

enabled=true

Disabled

□□ enabled=true

□□ enabled=false

Enabled

□□□□

□□□□□□□□

□□□□

□□□□

Active



IMS □□ IMS □□

IMS_Registered

VoLTE □□□□ □□□□

In_Call

□□□□□□



□□□□

No_Sessions

□□□□□□ □□□□□□

PDN_Active

VoLTE □□□□

VoLTE □□□□

PDN □□□□□□
□□□□□□

PDN_And_Call

□□□□□□ □□□□□□

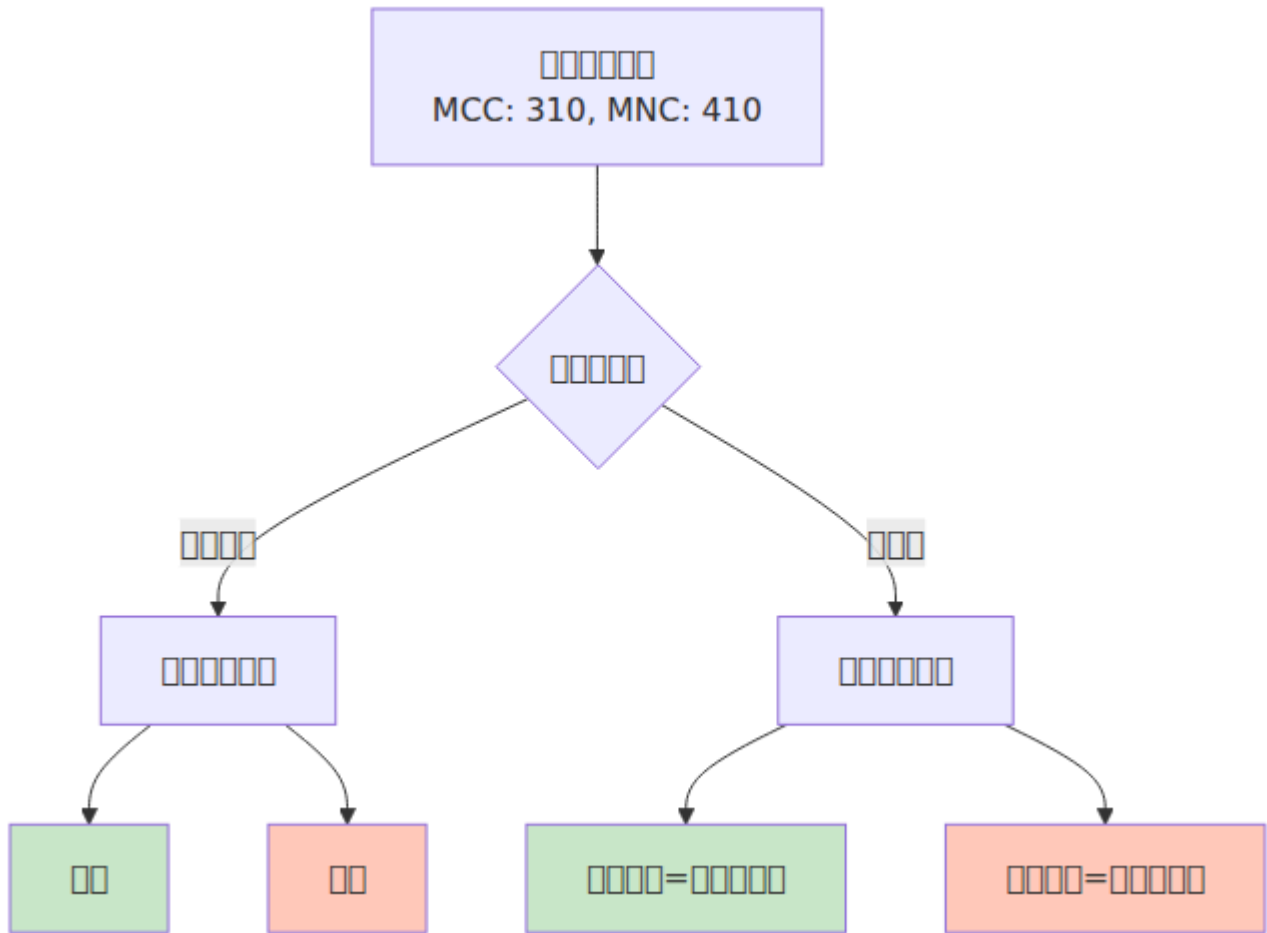
Multiple_Calls

PDN □□ + LTE □□
□□□□



□□□□□

□□□□□



□□□□□

S6a ULR □□

□□ IMSI □□□□

□□ EPC □□□□
+ APN □□□□

□□□□□□
□□□MME □

□□□□□□
AMBR□APN□QoS

S6a ULA □□

IMS □□□

Cx SAR □□

□□ IMSI/MSISDN □□□□

□□ IMS □□□□
+ MSISDN

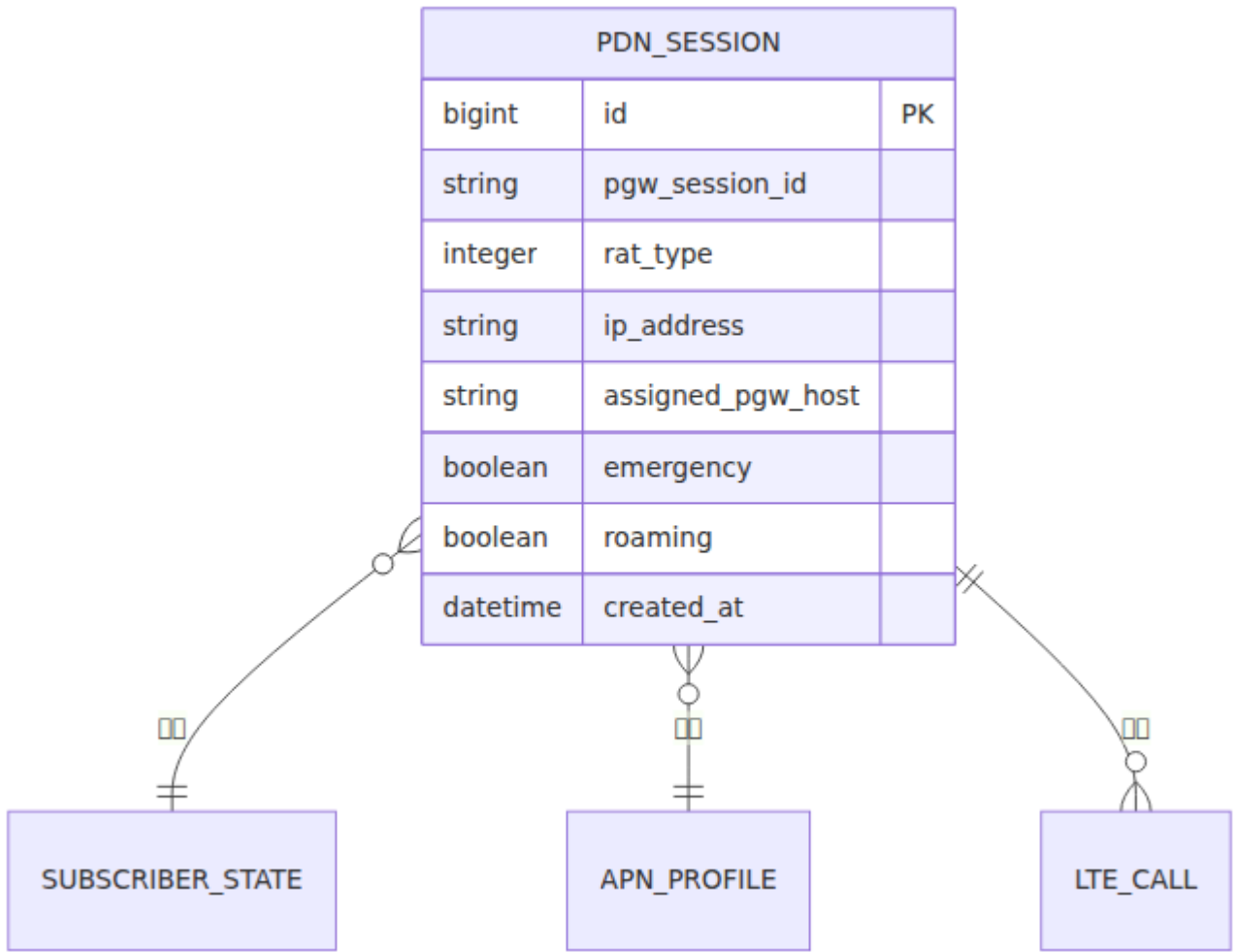
□□ S-CSCF
□□/□□

□□ IFC □□
□□□

□□□□□□
S-CSCF □□

Cx SAA □□

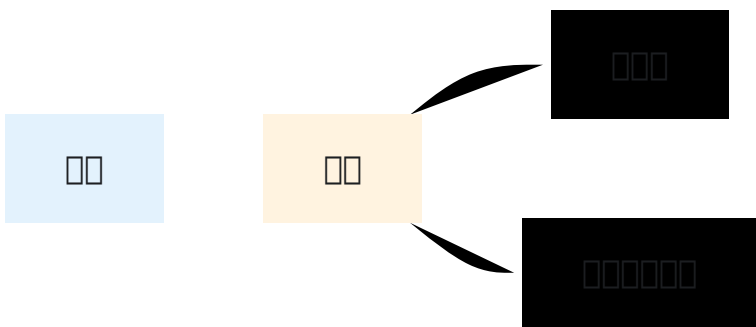
□□□□□



□□□□□□

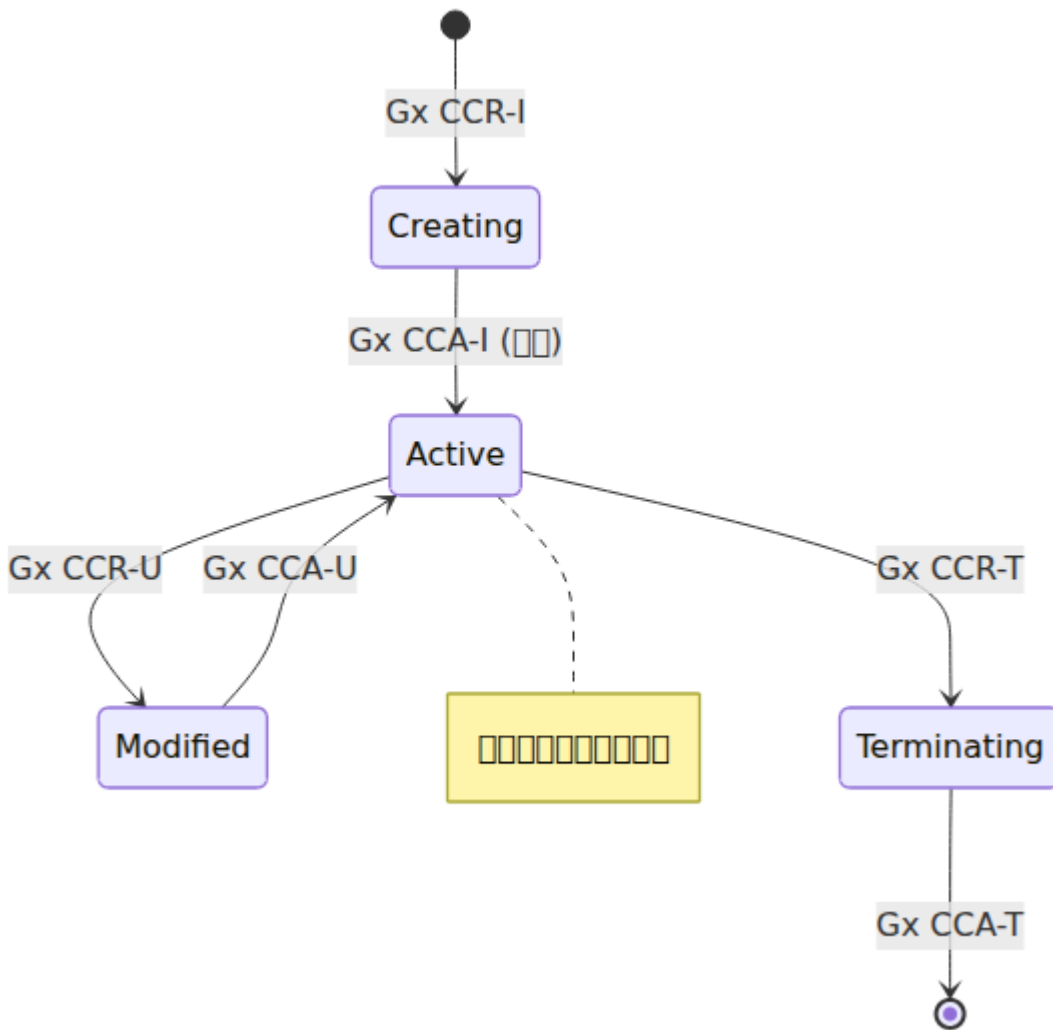
OmniHSS □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□



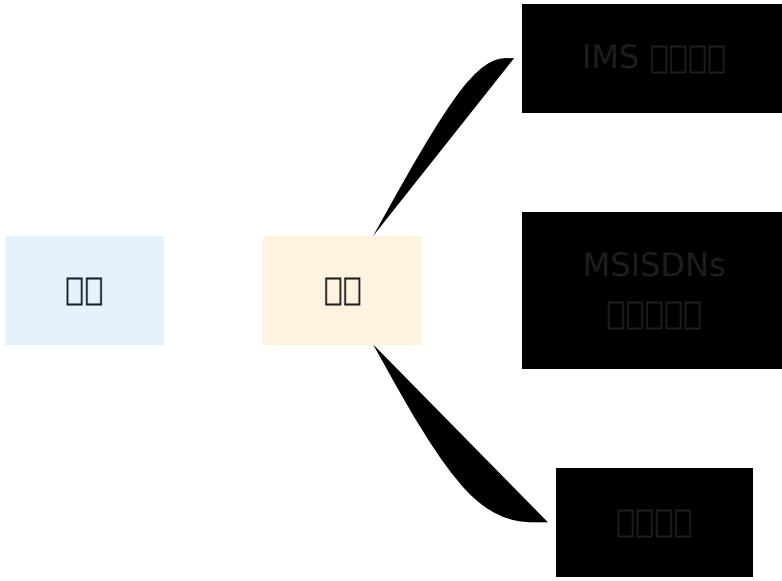
□□: S6a AIR - □□□□□□□□□□

□□□□□□□□□□



□□: S6a ULR - □□□□□ EPC □□□□□□

IMS 数据



数据: Cx SAR - 数据 IMS 数据

← 数据 | 数据: API 数据 →

Galera 高可用性

← 高可用性

高可用性

- 高可用性
- 高可用性
- Galera 高可用性
- 高可用性
- 高可用性
- 高可用性
- 高可用性
- 高可用性
- 高可用性

高可用性

OmniHSS 高可用性 Elixir 高可用性 **Ecto** 高可用性 Ecto 高可用性
MariaDB 高可用性 Galera 高可用性

高可用性 **Omnitouch** 高可用性 (ONS) 高可用性

MariaDB 高可用性 Galera

高可用性	高可用性
MariaDB 10.6+	Galera 高可用性

□□□□□□

□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□
- HSS □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

□□ ONS □□□□□□□□□□

- □□□□□□□□□□□□□□
 - □□□□□□□□□□□□
 - □□□□□□□□□□□□
 - □□□□□□□□□□□□
-

□□

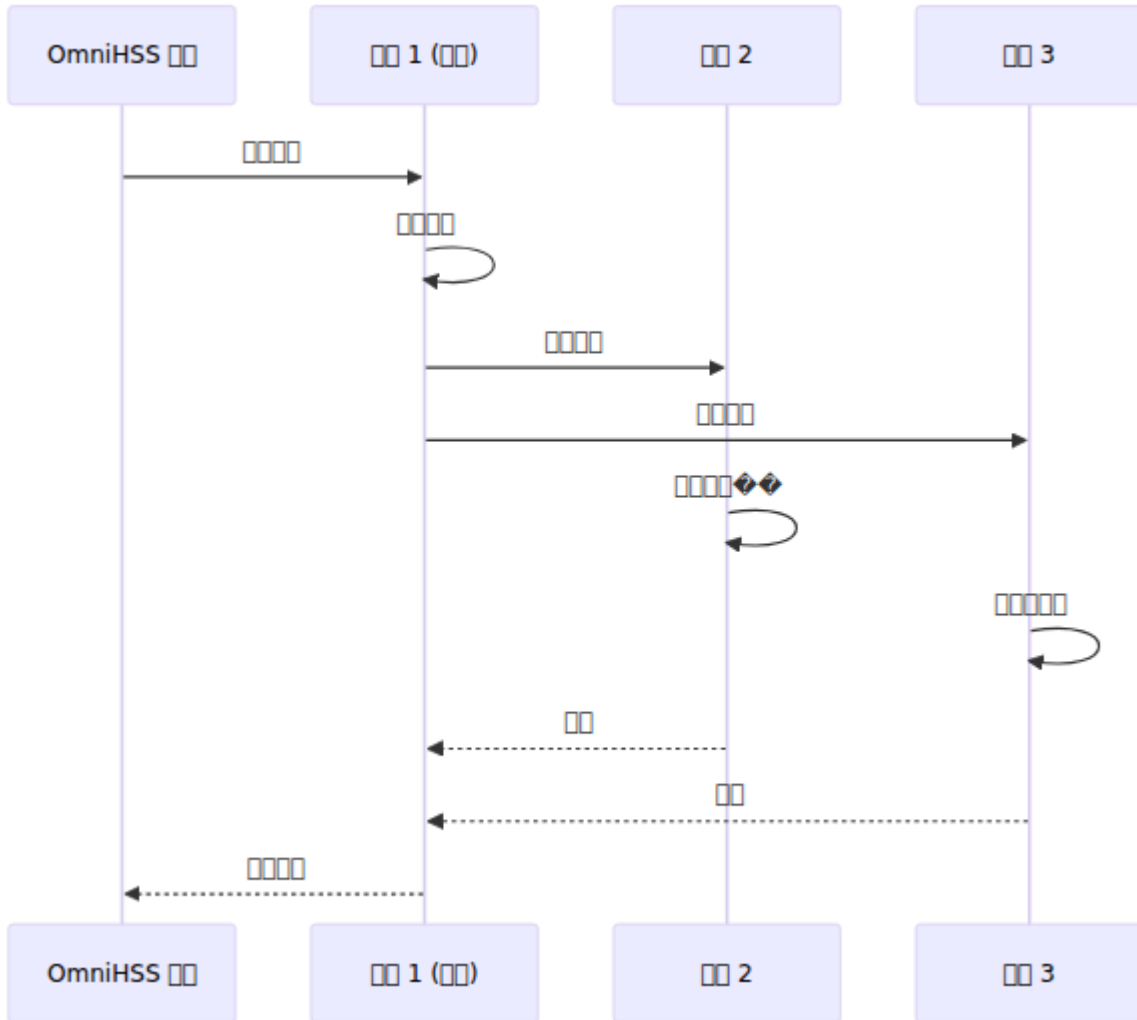
□□□□ **MariaDB Galera** □□□□□□□□ OmniHSS □□□□□□□□□□ Galera □□□□□□□□□□
□ HSS □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□
-

Galera 复制

复制过程



复制 (WSREP)

复制过程

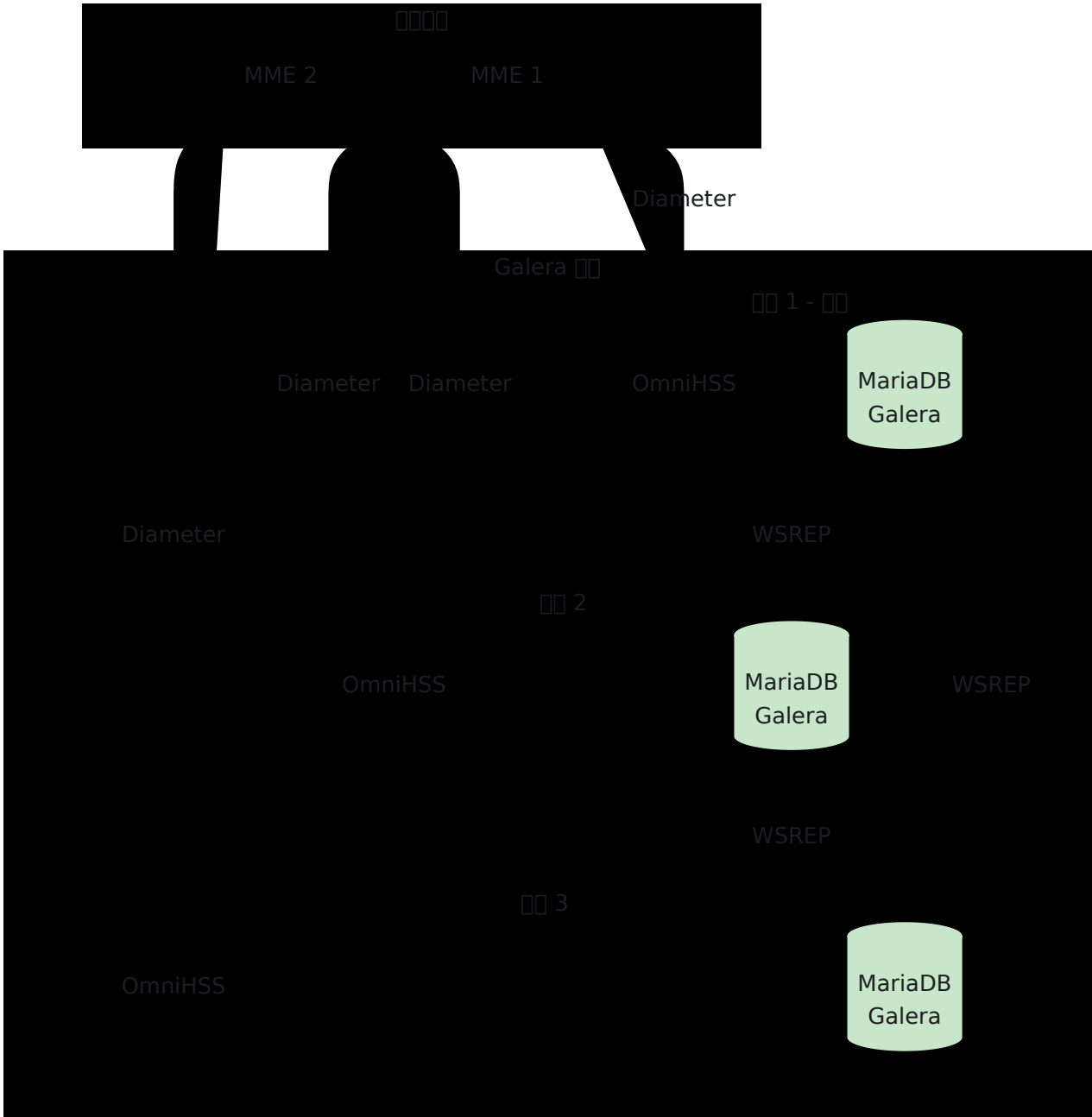
1. 复制 SQL
2. 复制“”
3. 复制
4. 复制
5. 复制

rsync

項目	値	説明
IST (項目)	rsync	rsyncを使用したSST
SST (項目)	rsync	rsyncを使用したSST

OmniHSS 項目 rsync 項目 SST

```
wsrep_sst_method=rsync
```



Ports

Port	Protocol	Service
3306	TCP	MySQL
4567	TCP/UDP	Galera
4568	TCP	Galera (IST)
4444	TCP	Galera (SST)

Firewall

```
# Allow Galera traffic
ufw allow from <node2_ip> to any port 3306,4567,4568,4444 proto tcp
ufw allow from <node2_ip> to any port 4567 proto udp
ufw allow from <node3_ip> to any port 3306,4567,4568,4444 proto tcp
ufw allow from <node3_ip> to any port 4567 proto udp
```

Ansible

Ansible Playbook

group_vars for Galera

```
omnihss:
  database_host: "localhost"
  database_username: "hss"
  database_password: "secure_password"
  mysql:
    replication_mode: "galera"           # Galera
    bootstrap_host: "hss01"            #
    run_bootstrap: false                # true
    reinstall: false                    # true MariaDB
```

Galera

Galera `/etc/mysql/my.cnf`

```
[mysqld]
#
pid-file           = /var/run/mysqld/mysqld.pid
socket             = /var/run/mysqld/mysqld.sock
datadir            = /var/lib/mysql
log-error          = /var/log/mysql/error.log

# Galera
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

#
wsrep_cluster_name="omnihss_galera"
wsrep_cluster_address="gcomm://10.4.10.140,10.4.10.141,10.4.10.142"

#
wsrep_sst_method=rsync

#
wsrep_node_address="10.4.10.140"
wsrep_node_name="hss01"
```

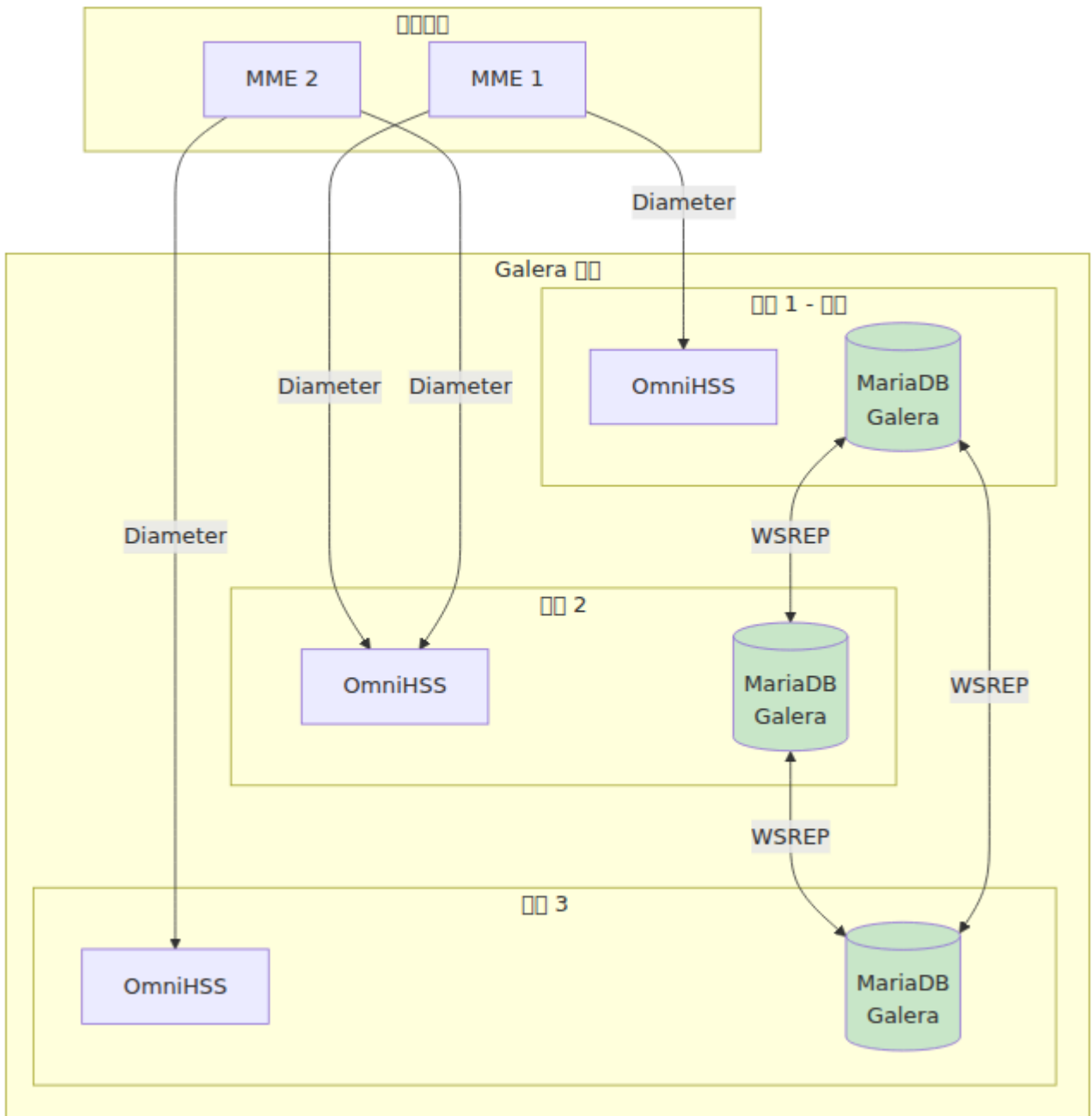
WSREP

WSREP	WSREP	WSREP
<code>binlog_format</code>	<code>ROW</code>	WSREP - ROW
<code>innodb_autoinc_lock_mode</code>	<code>2</code>	WSREP - innodb
<code>wsrep_on</code>	<code>ON</code>	WSREP ON
<code>wsrep_provider</code>	<code>libgalera</code>	Galera
<code>wsrep_cluster_name</code>	<code>"omnihss_galera"</code>	WSREP_CLUSTER_NAME
<code>wsrep_cluster_address</code>	<code>gcomm://ip1,ip2,ip3</code>	WSREP_CLUSTER_ADDRESS IP
<code>wsrep_sst_method</code>	<code>rsync</code>	WSREP_SST_METHOD
<code>wsrep_node_address</code>	<code>IP</code>	WSREP_NODE_ADDRESS IP
<code>wsrep_node_name</code>	<code>WSREP_NODE_NAME</code>	WSREP_NODE_NAME

WSREP

WSREP

WSREP Galera



□□□□

1. □□□□□□

```

omnihss:
  mysql:
    replication_mode: "galera"
    bootstrap_host: "hss01"
    run_bootstrap: true
  
```

2. Ansible

```
ansible-playbook -i hosts/your_site/inventory.ini
services/omnihss.yml
```

3. Galera

- AppArmor Galera
- Galera
- `/var/lib/mysql/grastate.dat` `safe_to_bootstrap=1`
- `mysqld_bootstrap`
- MariaDB `gcomm://`
-

4. Galera

```
omnihss:
  mysql:
    run_bootstrap: false #
```

Grastate

`/var/lib/mysql/grastate.dat`

```
# GALERA
version: 2.1
uuid: abc12345-6789-def0-1234-567890abcdef
seqno: 1234567
safe_to_bootstrap: 0
```

- `uuid`:
 - `seqno`:
 - `safe_to_bootstrap`: 1
-

□□

□□□□□

1. □□□□□ `hss` □□□□□
2. □□ `wsrep_cluster_address` □□□□□□□
3. □□ OmniHSS □□ - □□□□□□
 - □□ Galera □□□
 - □□□□□□□□□□□
 - □□ SST □□□□

□□□□

1. □□□□□□□□□□ OmniHSS □ MariaDB
2. □□□□□□□□□□
3. □□□□□□□□ `wsrep_cluster_address`
4. □□□□□□□□□□ MariaDB

□□□□

□□□□□□□□□□□□□

```
# □□□□□□□□□□□
systemctl stop omnihss
systemctl stop mysql
# □□□□
systemctl start mysql
systemctl start omnihss
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□

1. □□□□□□□□

```
# seqno
cat /var/lib/mysql/grastate.dat
```

2. seqno

```
# seqno
sed -i "/safe_to_bootstrap/s/0/1/" /var/lib/mysql/grastate.dat
mysqld_bootstrap
```

3. mysql

```
# mysql
systemctl start mysql
```

##

###

#####

```
-- #####
SHOW STATUS LIKE 'wsrep_cluster_size';

-- ###
SHOW STATUS LIKE 'wsrep_cluster_status';

-- ###
SHOW STATUS LIKE 'wsrep_local_state_comment';

-- WSREP
SHOW STATUS LIKE 'wsrep_%';
```

Galera

Variable	Value	Unit
<code>wsrep_cluster_size</code>	4	Nodes
<code>wsrep_cluster_status</code>	Primary	Status
<code>wsrep_local_state</code>	4	State
<code>wsrep_local_state_comment</code>	Synced	Comment
<code>wsrep_ready</code>	ON	Boolean
<code>wsrep_connected</code>	ON	Boolean

Galera

Step	Order	Description
1	1	Initial State
2	2	State Set (SST)
3	3	State Transfer
4	4	Final State

Prometheus

Galera MariaDB Galera OmniHSS Galera Galera

□□□□

□□□□□□□□

□□□□□□□□□□□□

□□□

```
# □□ MariaDB □□□□  
tail -f /var/log/mysql/error.log  
  
# □□ wsrep □□□□□□  
mysql -e "SHOW STATUS LIKE 'wsrep_on';"
```




□□□□□

- □□□□□□□ 4567□4568□4444
- wsrep_cluster_address □□ IP □□
- AppArmor □□□□
- □□ UUID □□□

□□□

```
# □□ AppArmor □□□  
systemctl status apparmor  
# □□□□□□□□□□systemctl stop apparmor && systemctl disable apparmor  
  
# □□□□□□□□□□  
ss -tlnp | grep -E '4567|4568|4444|3306'
```

□□ / □□□□

□ wsrep_cluster_status □□ non-Primary

□□□□□

- □□□□□□□□□□□□□□□□

- 設定項目

設定

```
-- 設定項目  
SET GLOBAL wsrep_provider_options='pc.bootstrap=YES';
```

SST 設定

設定項目

設定

```
# 設定項目  
df -h /var/lib/mysql  
  
# rsync 設定  
ps aux | grep rsync
```

設定

- 設定項目
- rsync 設定
- 設定項目 4444

設定項目

設定 wsrep_local_state_comment 設定 Donor/Desynced

設定 SST 設定項目

設定

```
# 確認 rsync
ps aux | grep rsync
# 強制終了
pkill rsync
systemctl restart mysql
```

Grastate 確認

MySQL MariaDB の Grastate 確認

確認

```
# Grastate 削除
rm /var/lib/mysql/grastate.dat

# MySQL を再起動して SST
systemctl start mysql
```

確認

MySQL の wsrep_local_send_queue

確認

```
SHOW STATUS LIKE 'wsrep_local_send_queue%';
SHOW STATUS LIKE 'wsrep_flow_control%';
```

確認

- wsrep_local_send_queue
- wsrep_flow_control
- wsrep_flow_control_bytes

確認

- wsrep_flow_control_bytes
- wsrep_flow_control_bytes_max

- □□□□□□□□
-

Diameter

←

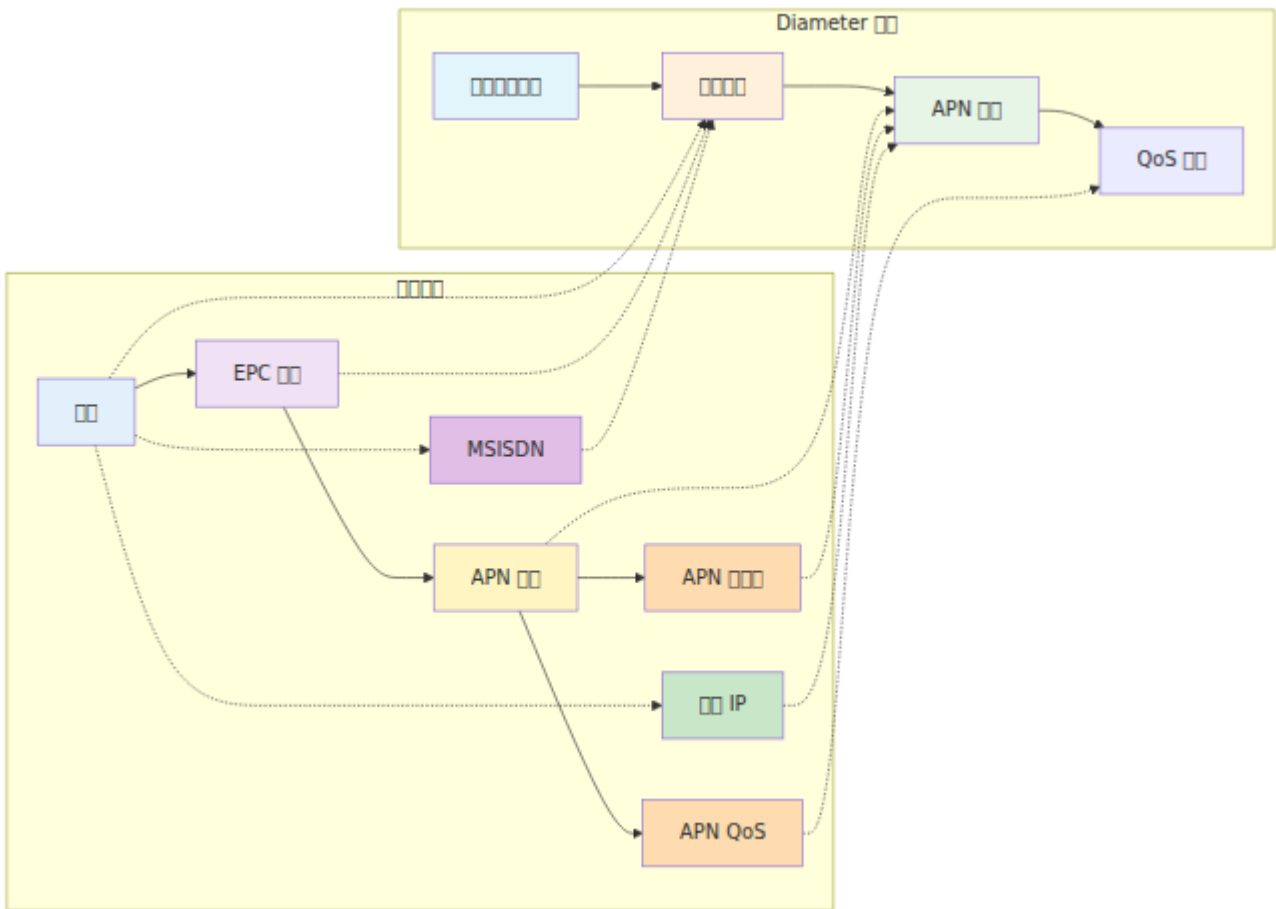
mermaid Diameter OmniHSS

- (S6a ULA)
- (S6a AIA)
- (Cx SAA)
- (Gx CCA)
- (Sh UDA)
- ME (S13 ECA)

(S6a ULA)

HSS LTE MME Diameter AVP

□□□□□



□□□□□□

□□□□□	□□	D
subscriber.enabled	true/false	Su St
msisdn.msisdn	'14155551234'	MS
epc_profile.ue_ambr_ul_kbps	50000	Ma Re Ba UL
epc_profile.ue_ambr_dl_kbps	100000	Ma Re Ba DL
epc_profile.network_access_mode	'packet_only'	Ne Ac Mo
apn_identifier.apn	'internet'	Se Se
apn_identifier.ip_version	'ipv4v6'	PE
apn_qos_profile.qci	9	Qc Id
apn_qos_profile.allocation_retention_priority	8	Pr Le

Property Name	Value	Description
apn_qos_profile.pre_emption_capability	false	Pre-emption Capability
apn_qos_profile.pre_emption_vulnerability	true	Pre-emption Vulnerability
apn_qos_profile.apn_ambr_ul_kbps	25000	APN UL AMBR
apn_qos_profile.apn_ambr_dl_kbps	50000	APN DL AMBR
static_ip.ipv4_static_ip	'100.64.1.1'	Static IPv4 IP Address (If enabled)
static_ip.ipv6_static_ip	'2606:4700::1111'	Static IPv6 IP Address (If enabled)

Configuration

- AMBR** (kbps) Diameter (bps) 1000
- IP** (0=IPv4, 1=IPv6, 2=IPv4v6, 3=IPv4_or_IPv6)
- enabled**: true → 0 (SERVICE_GRANTED), false → 1 (OPERATOR_DETERMINED_BARRING)
- APN (0, 1, 2...)
- IP** (static_ips)

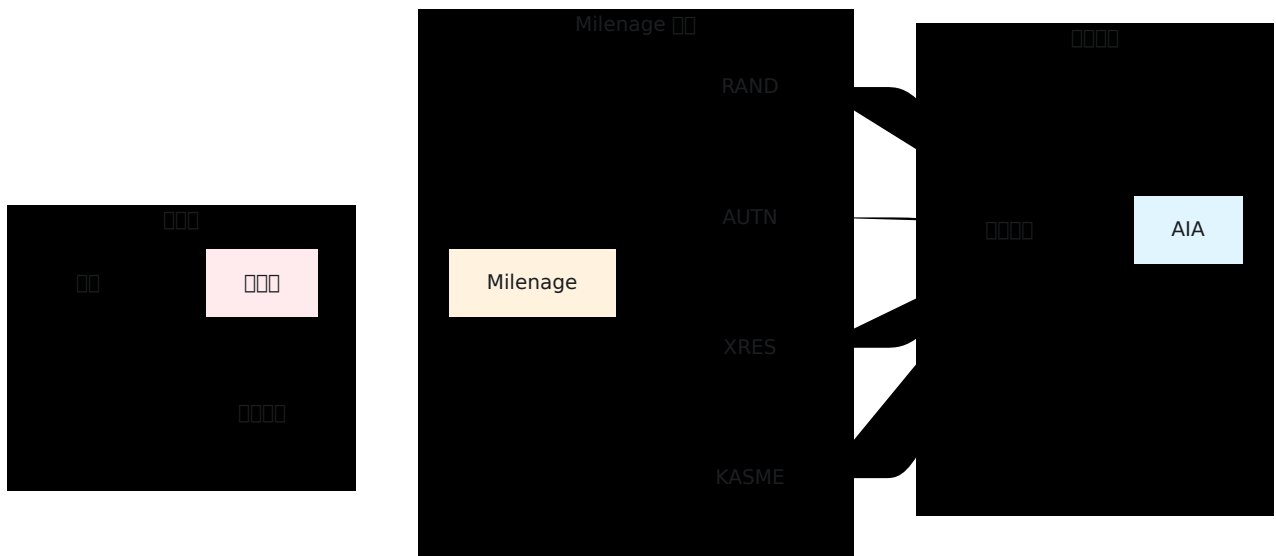
Configuration

- `roaming_profile.roaming_rules`
- `subscriber.enabled == true`
- APN IMS

Authentication (S6a AIA)

LTE/EPC Authentication

Authentication



Authentication

1. `key_set`
2. **SQL**
3. **Milenage** 3GPP TS 35.206 -
4. **KASME** CK||IK KDF TS 33.401

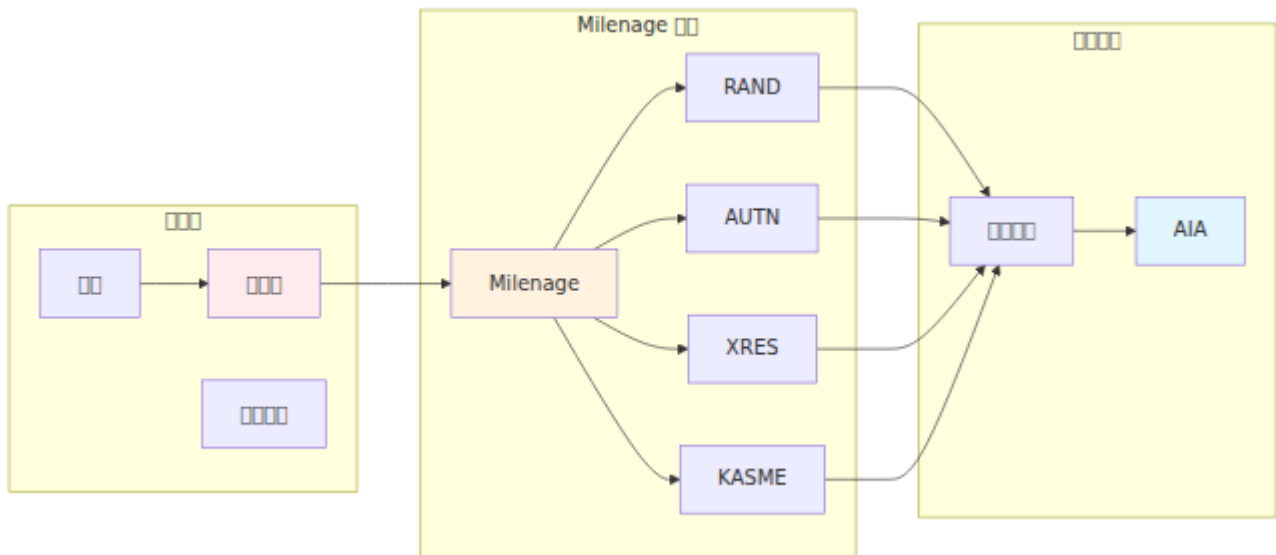
Authentication

- SQL
- Ki/OPc HSS
- AUTN SQL AMF
- Milenage UE

Authentication (Cx SAA)

Authentication HSS IMS Authentication S-CSCF

Authentication



Authentication

1. **IFC** XML template `ims_profile.ifc_template`
2. Authentication parameters `{{msisdn}}` `{{imsi}}` `{{impu}}`
3. **S-CSCF** parameter `subscriber_state.assigned_scscf` S-CSCF
4. **IMS** parameters `sip:+{{msisdn}}@{{ims_domain}}` `tel:+{{msisdn}}`

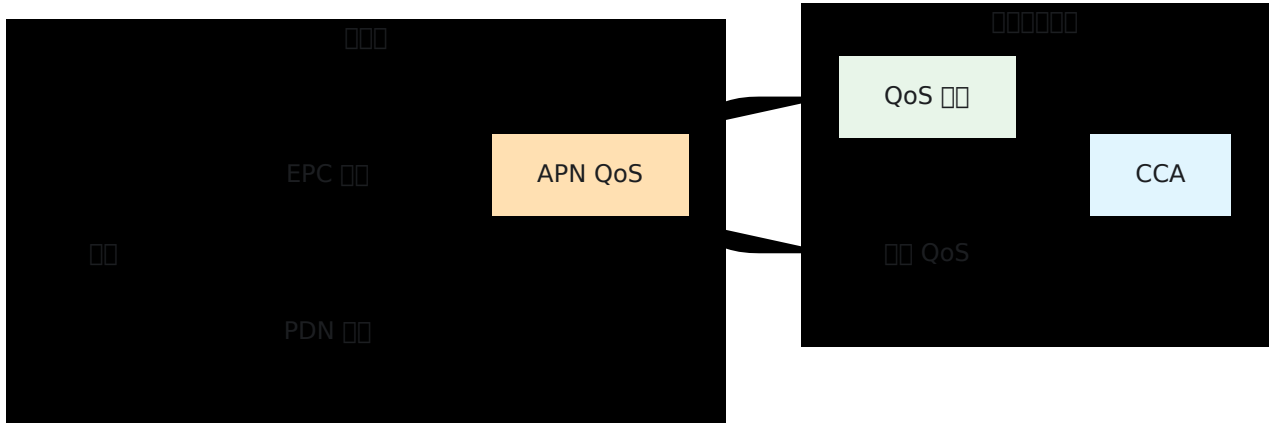
IFC parameters

- `{{msisdn}}` - MSISDN
- `{{imsi}}` - IMSI
- `{{impu}}` - IMS subscriber_state
- `{{impi}}` - IMS IMSI@realm

Authentication (Gx CCA)

Authentication PCRF Authentication PGW

□□□□□



□□□□□

1. □□□□□□□□□□/□□ `pdn_session` □□
2. **QoS** □□□□□□□□□□ APN QoS □□□□ QCI □□□□□□
3. □□□□□□□□□□□□□□□□□□□□
4. **CC-**□□□□□□□□ INITIAL (1)□UPDATE (2)□TERMINATION (3)

□□□□□□□

- `INITIAL_REQUEST` □□□□□□ PDN □□□□□
- `UPDATE_REQUEST` □□□□□□ PDN □□
- `TERMINATION_REQUEST` □□□□ PDN □□□□□

□□□□□□ (Sh UDA)

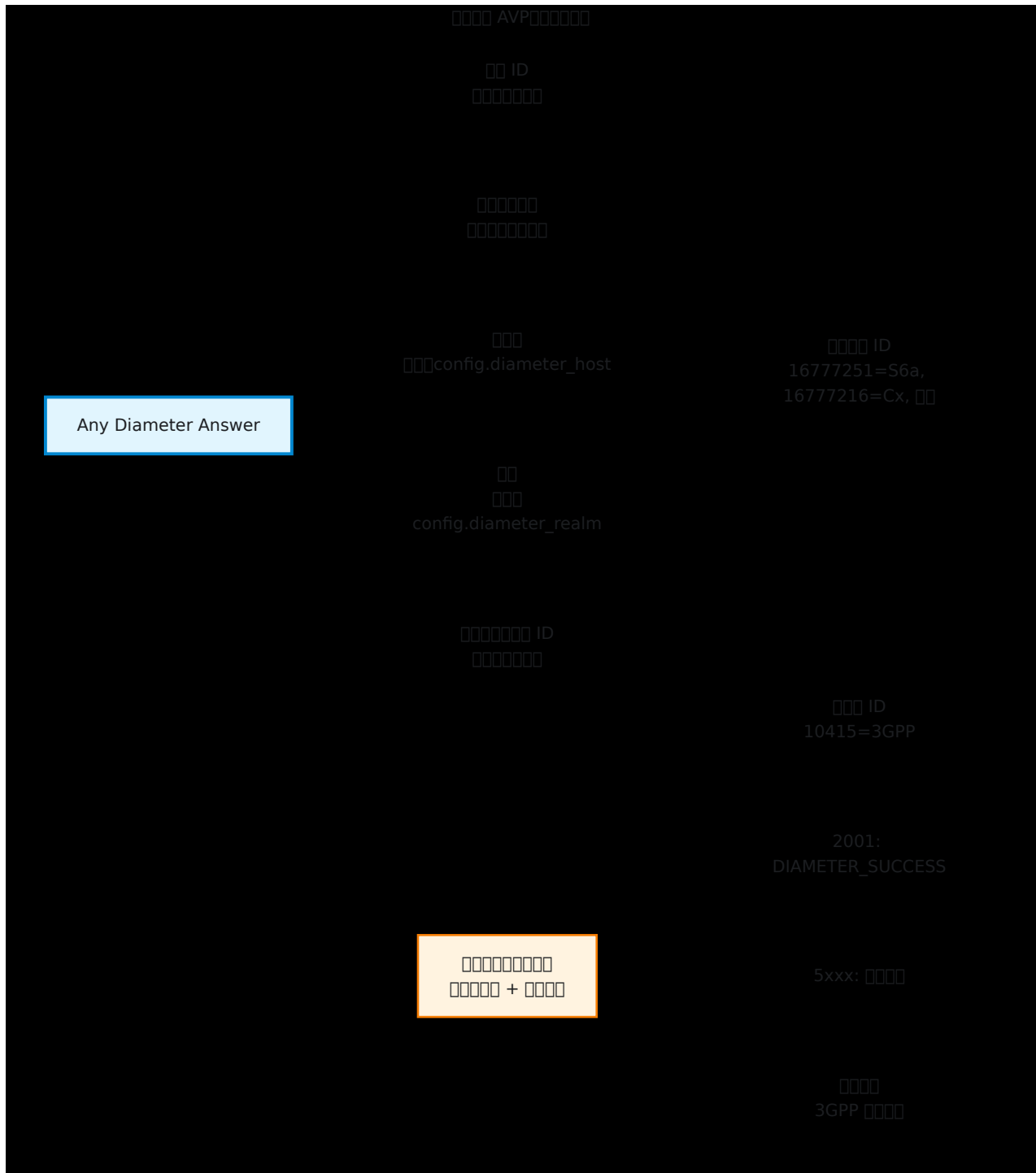
□□□□□□□ HSS □□ Sh □□□□□□ AS□□□□□□□□□□

□□□□□□



- IMEI
 - TAC
 -
 -
-

Diameter AVP

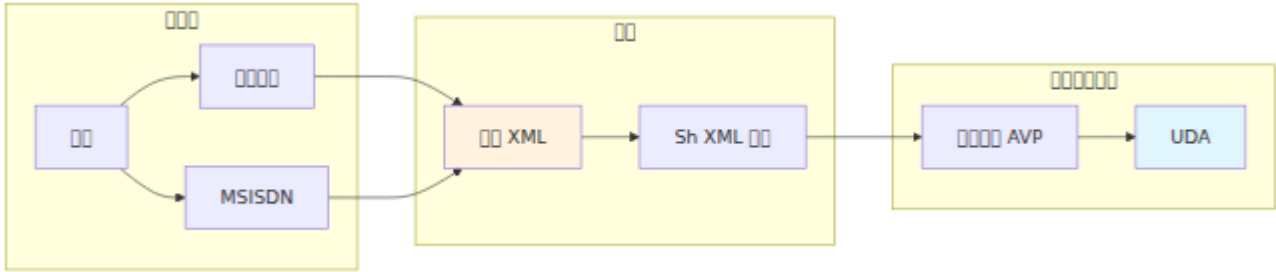


config

```
config :diameter_ex,
  diameter_host: "hss",
  diameter_realm: "example.com",
  diameter_service_name: "OmniHSS"
```

□ ◆ ◆ □ □ □

□ □ □ □ □ □



□ □ □ □

□ □ □ □ □

□ □ □ □ □ Diameter □ □ □ □ □ □

- **S6a** - LTE/MME □ □ □ □ □ □ □ □ □ □
- **Cx** - IMS/CSCF □ □ □ □ IMS □ □ □ □ □ □ □ □
- **Sh** - IMS/AS □ □ □ □ □ □ □ □ □ □
- **Gx** - PCRF □ □ □ □ □ □ □ □ □ □
- **Rx** - IMS/AF □ □ □ □ □ □ □ □
- **S13** - EIR □ □ □ □ IMEI □ ◆ ◆ ◆
- **SWx** - WiFi/IMS □ □ □ □ 3GPP □ □ □ □

□ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □

- □ □ - □ □ □ □ □ □ □ □ IMSI
- □ □ □ - □ □ □ □ □ □ □ □ □ □
- **EPC** □ □ - LTE □ □ □ □
- **APN** □ □ - □ □ □ □ □
- **IMS** □ □ - □ □ IFC □ □ □ □ IMS □ □ □ □

- **IMEI** - **IMEI** **IMEI**
- **IMEI** - **IMEI** **IMEI**
- **PDN** **IMEI** - **IMEI** **IMEI**
- **IP** **IMEI** - **IP** **IMEI**
- **EIR** **IMEI** - **IMEI** **IMEI**

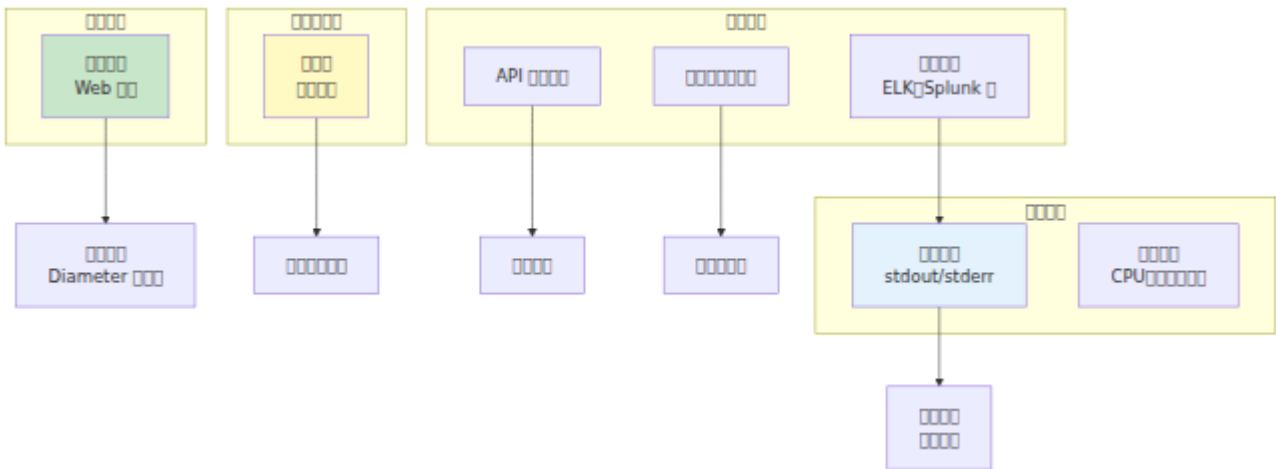
← **IMEI** | **API** **IMEI** → | **IMEI** →

OmniHSS

←

-
-
-
-
-
-
-

OmniHSS



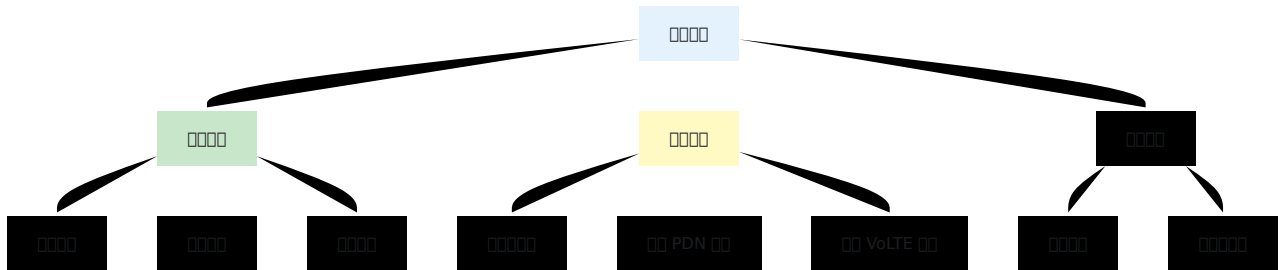
概要

概要

概要

URL: `https://[hostname]:7443/overview`

概要



概要

項目	概要	概要
概要	概要	概要
概要	MME 概要	概要
PDN 概要	PDN 概要 > 0	概要
IMS 概要	S-CSCF 概要	概要
概要	概要 > 0	VoLTE 概要

概要

概要

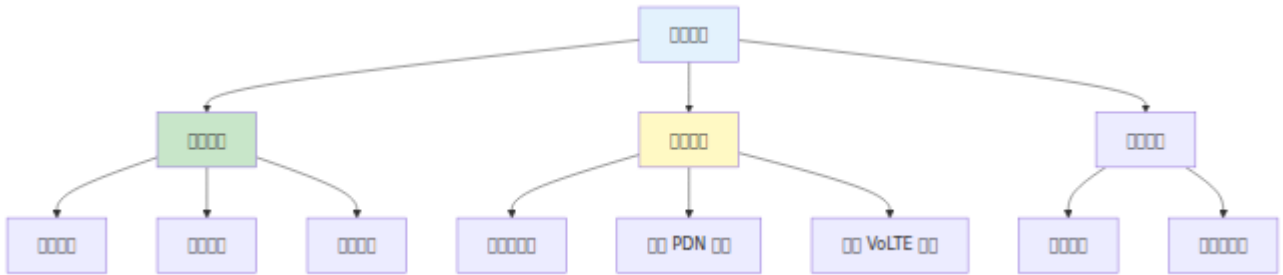
- 概要
- 概要
- 概要

4. Diameter 프로토콜

Diameter 프로토콜

URL: `https://[hostname]:7443/diameter`

프로토콜



프로토콜

프로토콜

프로토콜	주소	프로토콜
MME	1	프로토콜 LTE
P-GW	1	프로토콜
S-CSCF	1	IMS
P-CSCF	1	VoLTE
I-CSCF	1	IMS
AS	1-1	프로토콜

프로토콜

URL: `https://[hostname]:7443/application`

프로토콜

項目	項目	項目	項目
項目	項目 Erlang 項目	項目	> 90% 項目
項目	項目	< 80%	> 90%
項目	項目	N/A	項目

項目

項目

項目 SQL 項目

項目

項目

- 項目
- 項目
- IMS 項目

項目

項目

- 項目 PDN 項目
- 項目 VoLTE 項目
- 項目 APN 項目 PDN 項目

項目

項目

- 項目 MCC-MNC 項目
- 項目 PLMN 001-001 項目

- 0000000000000000

0000

0000000000

- 0000000000000000
- 000 MME 00000000
- 0000000000000000

00000000

000000000000

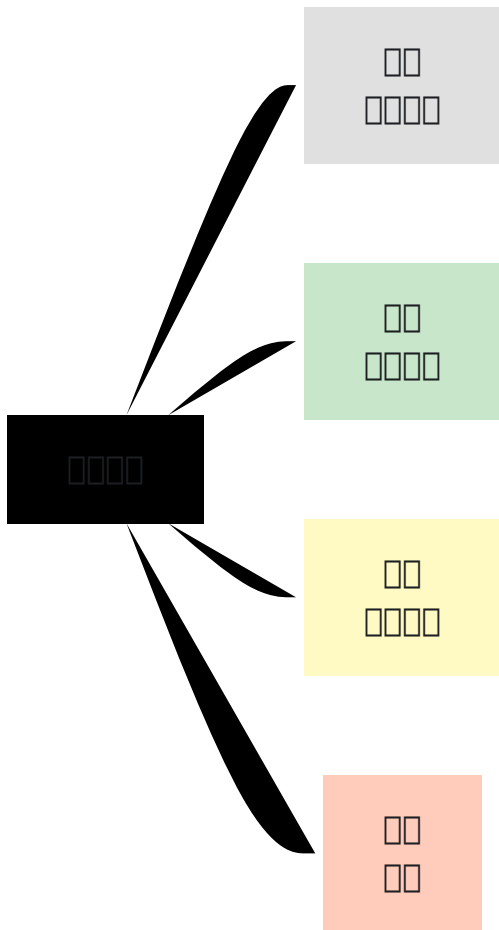
- 000000000000
- 000 00000000
- 00000000
- 000000000000

0000

0000

OmniHSS 000000 **stdout/stderr**0000000000000000

0000



□□□□□□□□□□

Diameter □□□□:

```
[info] Diameter peer connected: mme01.epc.example.com
[warn] Diameter peer disconnected: pgw01.epc.example.com
[error] Diameter peer connection failed: timeout
```

□□□□:

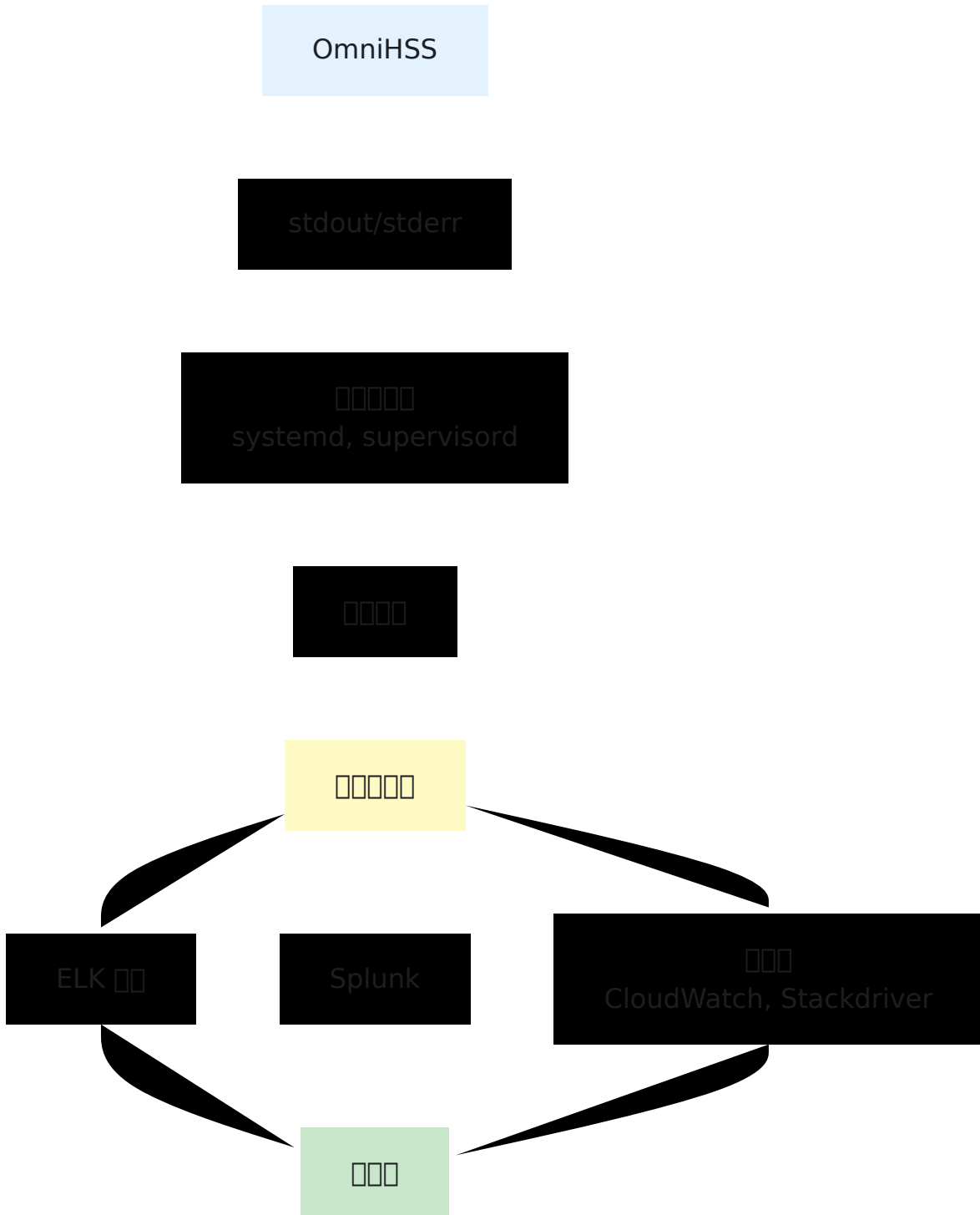
```
[info] Database connection established
[error] Database connection lost: timeout
[error] Database query failed: deadlock detected
```

□□□□:

```
[info] Authentication successful: IMSI 001001123456789
[warn] Authentication failed: IMSI 001001123456789, invalid vector
[error] Roaming denied: IMSI 001001123456789, MCC 310 MNC 410
```

□□□□

□□□□□□□□□□□□□□□□



□□□□□□

□□□□□□

API □□□□: GET /api/status

```
curl -k https://hss.example.com:8443/api/status
```

□□□□:

```
{"status": "ok"}
```

HTTP □□: 200 OK

□□□□□□

Nagios/Icinga □□

```
#!/bin/bash
# check_omnihss.sh

API_URL="https://hss.example.com:8443/api/status"

response=$(curl -k -s -o /dev/null -w "%{http_code}" "$API_URL" --
max-time 5)

if [ "$response" = "200" ]; then
    echo "OK - OmniHSS API responding"
    exit 0
else
    echo "CRITICAL - OmniHSS API not responding (HTTP $response)"
    exit 2
fi
```

Prometheus □□

OmniHSS Prometheus API

SNMP

SNMP SNMP API SNMP OID

KPI

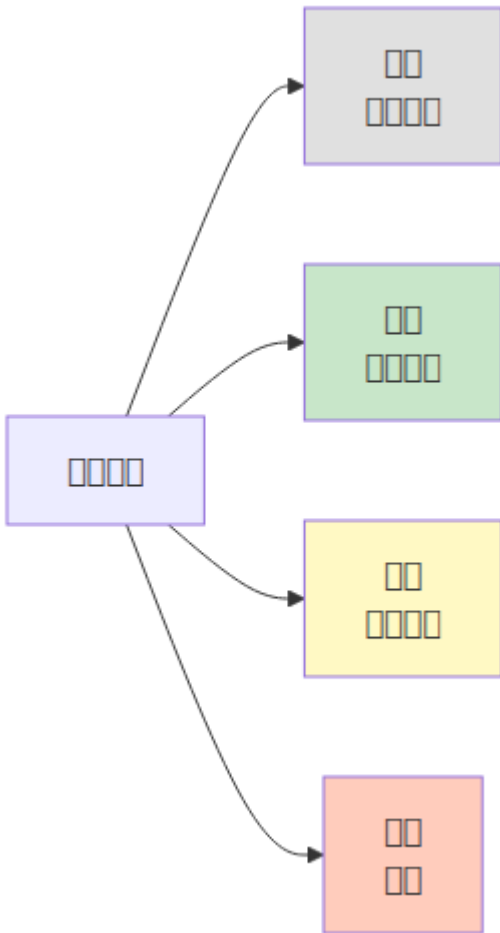


Table 1: KPI

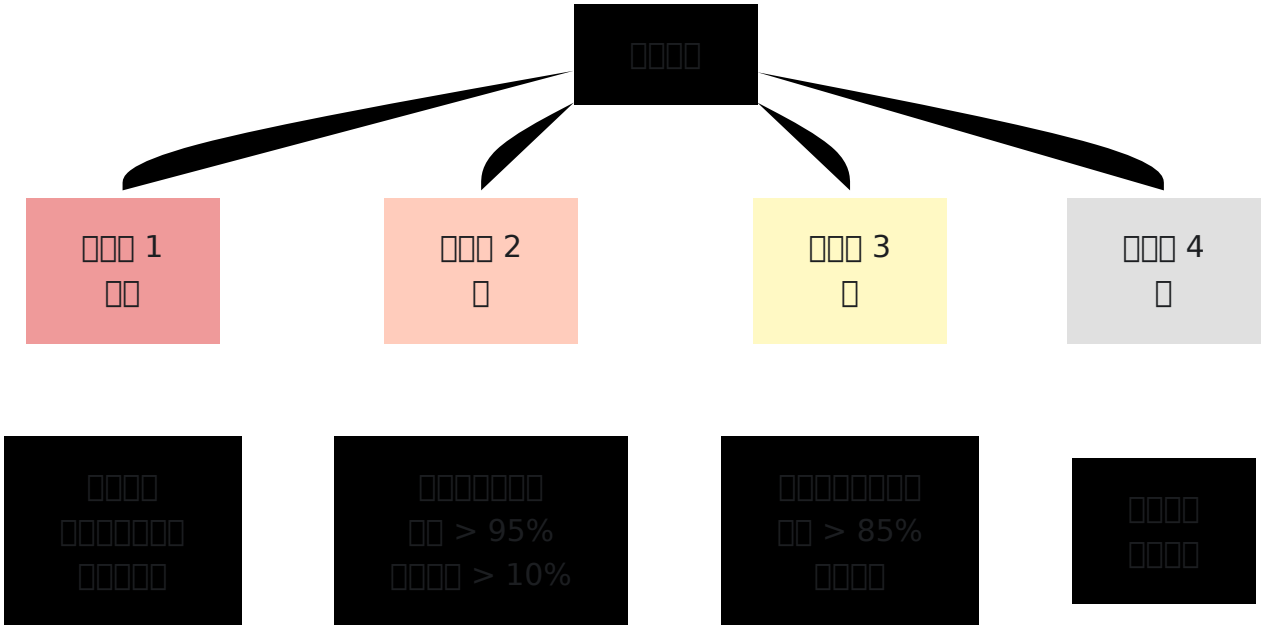
KPI	Target	Warning	Critical
Availability	99.99%	< 99.95%	< 99.9%
Diameter Availability	99.9%	< 99.5%	< 99%
Throughput	> 99%	< 99%	< 95%
Diameter Latency	< 100ms	> 200ms	> 500ms
Latency	< 50ms	> 100ms	> 500ms
Errors	< 0.1%	> 0.5%	> 1%

Table 2: KPI

Target	Warning	Critical
Throughput	Throughput	80% Throughput
PDN Throughput	Throughput	70% Throughput
Throughput	Throughput MB	80% Throughput
Throughput	Throughput	80% Throughput

□□□□

□□□□□



□□□□

□□□□ (P1)

□□□□□:

- API □□□□□□□
- □□□□□□□□□
- □□□□□□□□
- □□: □□□□□□□□

□□ Diameter □□□□□:

- □□□□□□□
- □□: □□□□□□□□□□□

□□□□□:

- □□□□□□ SQL □□□

- 時間: 10分

問題 (P2)

問題 Diameter 問題:

- MME 問題
- P-GW 問題
- S-CSCF 問題
- 時間: 15分

問題:

- 時間 > 95%
- 時間: 10分

問題:

- 10% 問題
- 時間: 10分

問題 (P3)

問題:

- 問題
- 問題
- 時間: 1分

問題:

- 時間 > 85%
- 時間: 10分

問題:

- 時間 > 1%
- 時間: 10分

□□□□□ (P4)

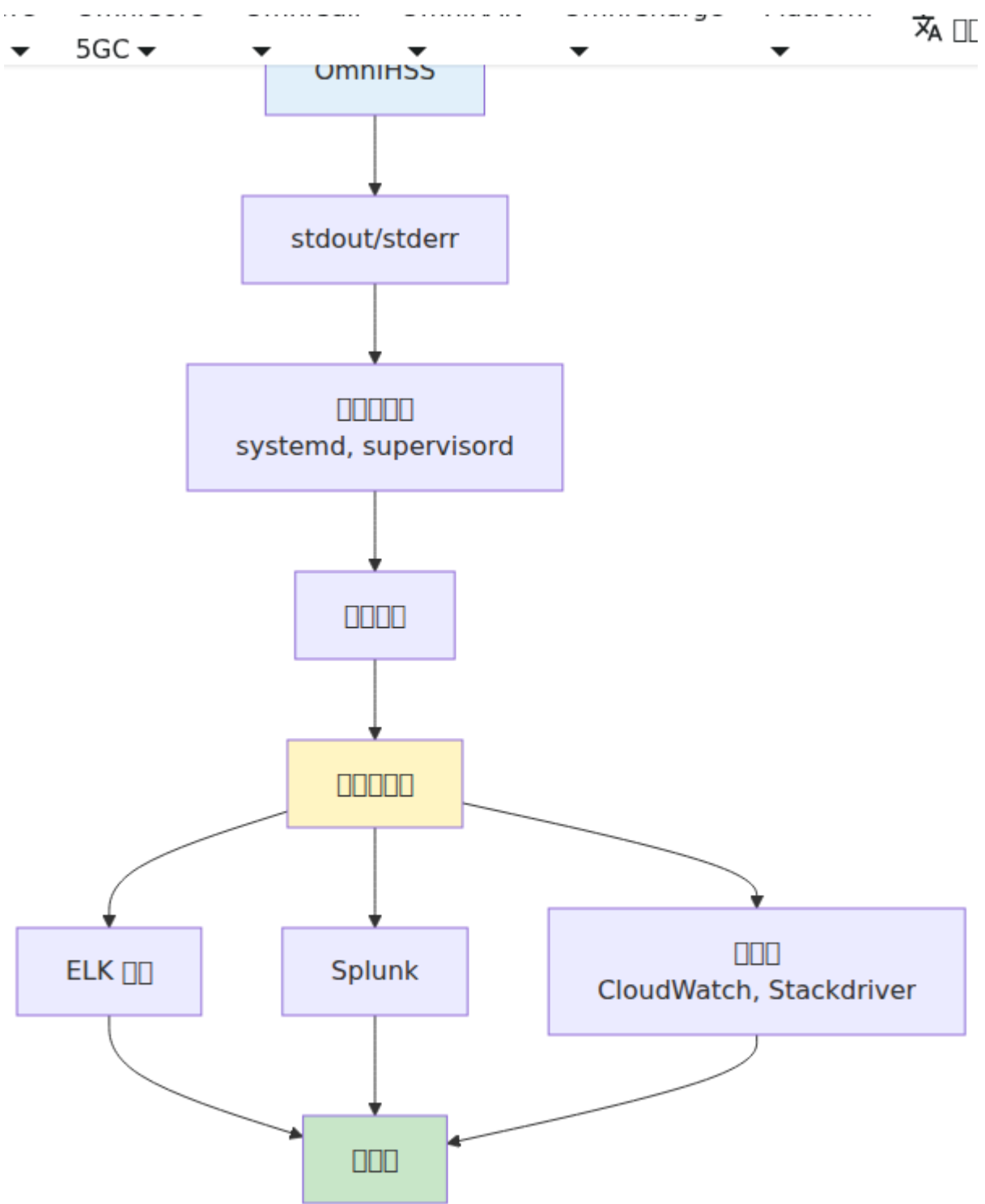
□□□□:

- □□ > 80% □□□
- □□□ > 80% □□□□□
- □□: □□□□□□

□□□□:

- □□□□□□□□□□
- □□: □□□□□□□

□□□□□□



□□□□□□

□□□□

- - □□□□□□
- Diameter □□ - □□□□◆◆□□□□
- - □□□□□□□□□□
- - □□ 24 □□□□□□□□
-

□□□□

- - □□□□
- - □□□□
- - □□□□□□
- - □□□□
- - □□□□□□

□□□□

- - □□ 6 □□□
- - □□□□□□
- - □□□□□□□□
- - □□□□□□
- - □□□□□□□□

OmniHSS MSISDN IMSI

←

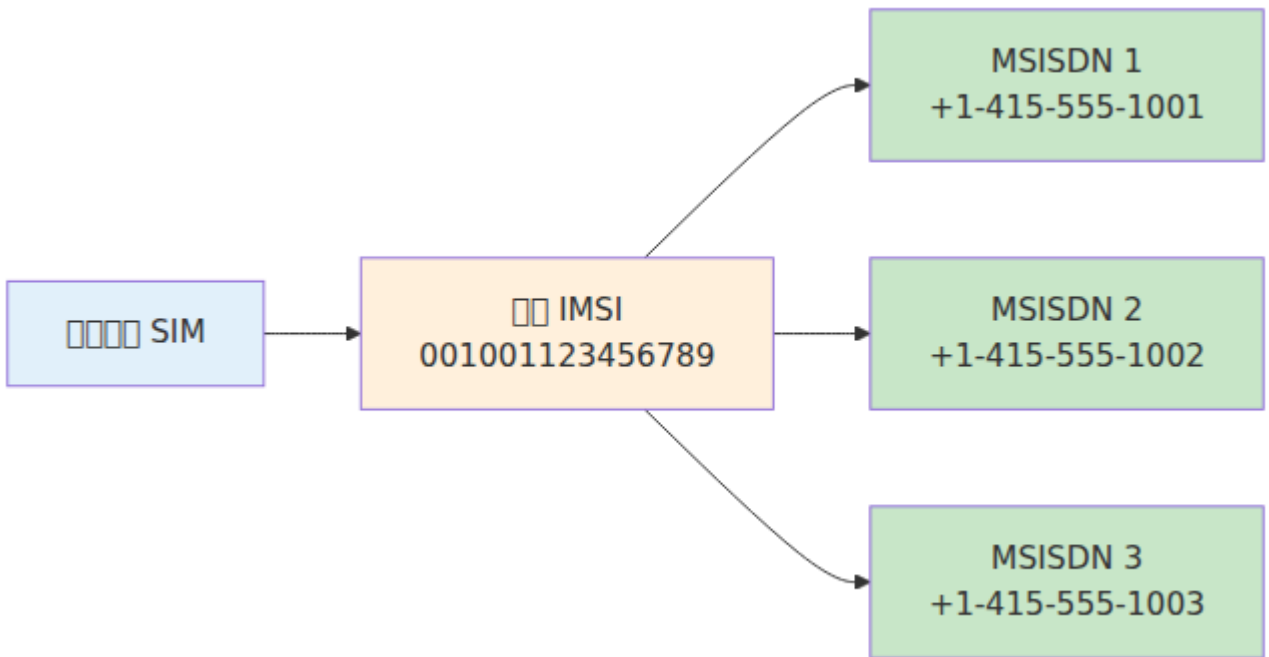
-
- MSISDN
- IMSI SIM
-
-
-

OmniHSS

MSISDN

IMSI →

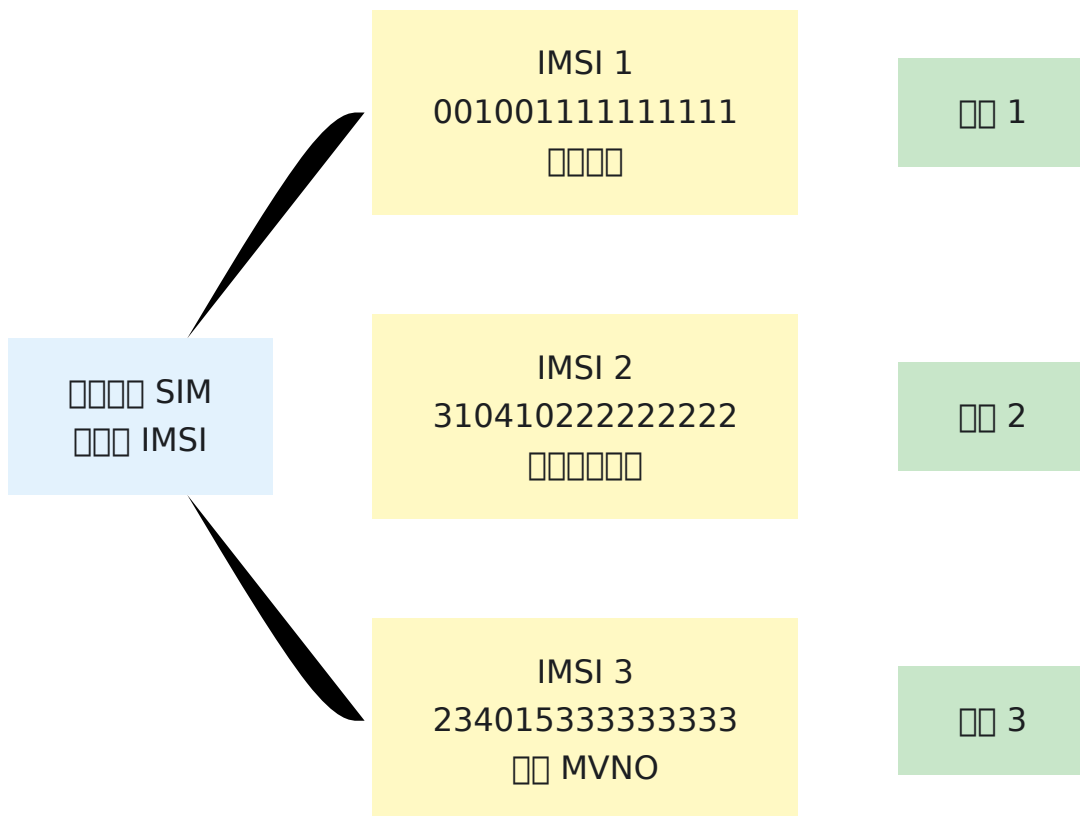
IMSI MSISDN



IMSI SIM

SIM → IMSI

SIM IMSI MVNO



MSISDN

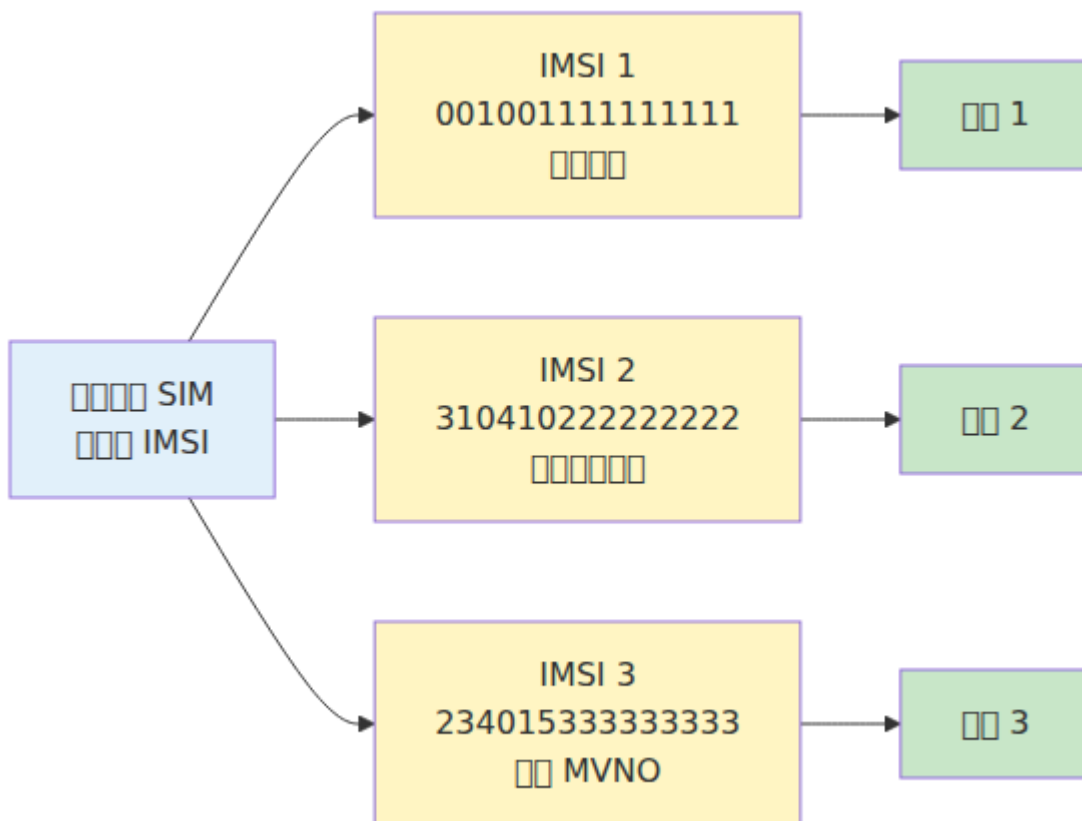
MSISDN

HSS MSISDN IMS MSISDN IMS

MSISDN

- IMSI - SIM IMSI
- MSISDN - MSISDN
- IMS - MSISDN IMS
- - EPC IMS

MSISDN

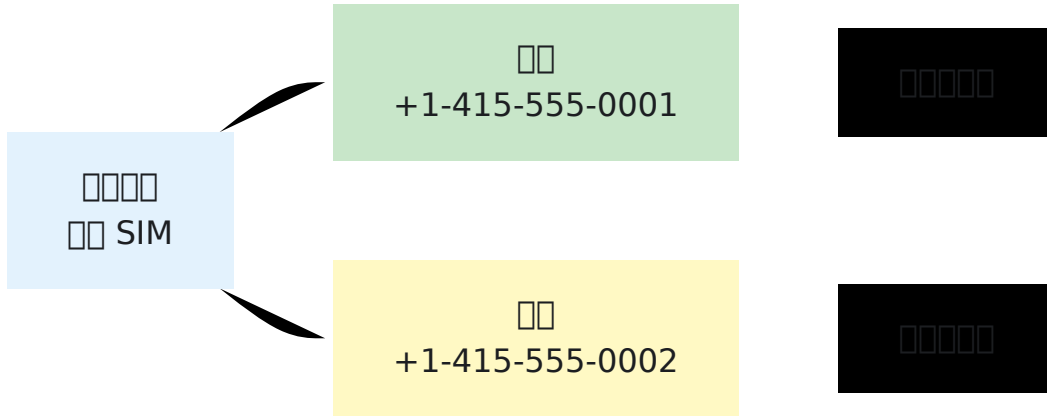


MSISDN MSISDN

□□□□

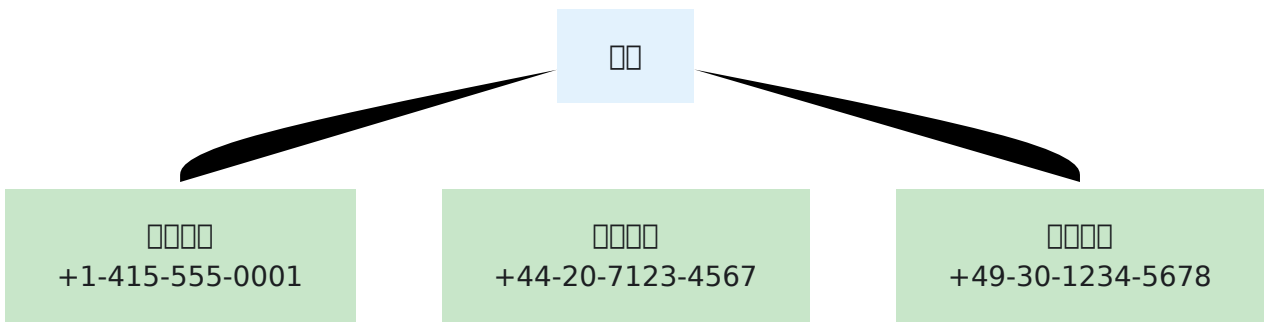
1. □□□□□□

□□□□□□□□□□□□□□□□□□□□



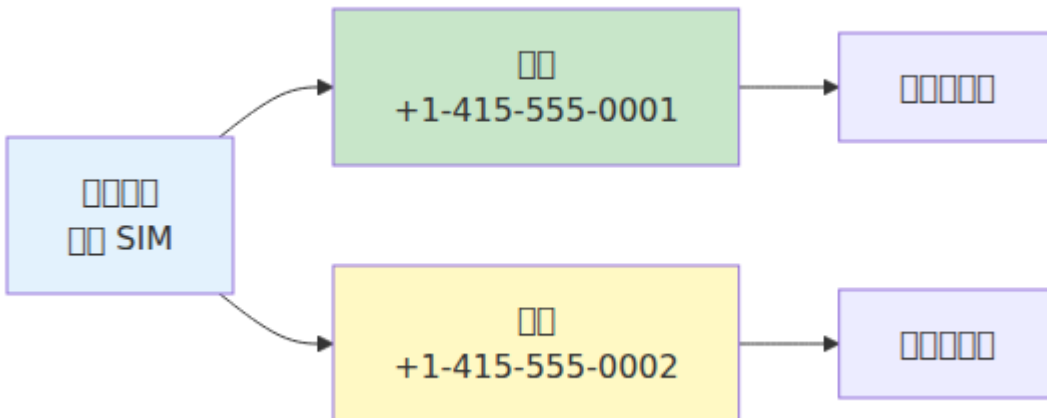
2. □□□□

□□□□□□□□□□□□□□□□□□



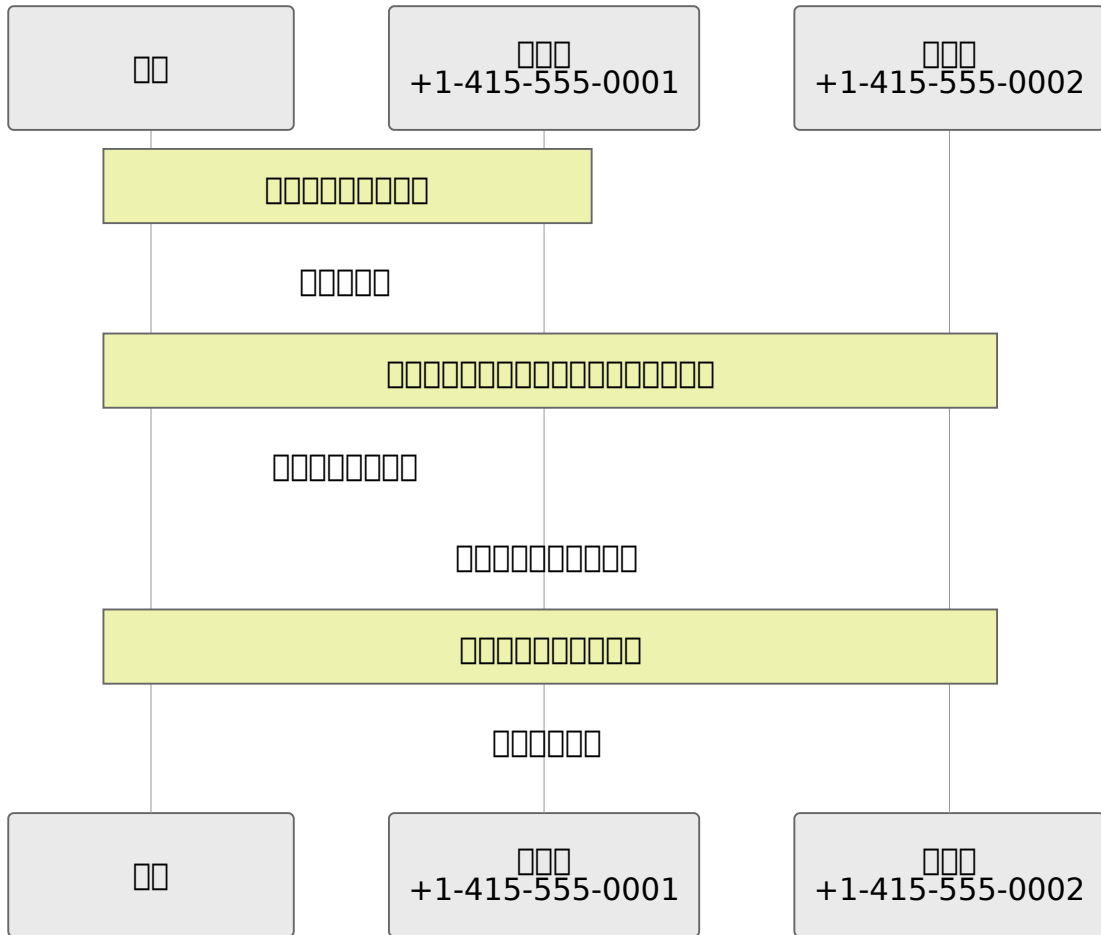
3. □□□□

□□□□□□□□□□□□□□□□



OmniHSS SIM/IMSI MSISDN

4.



MSISDN

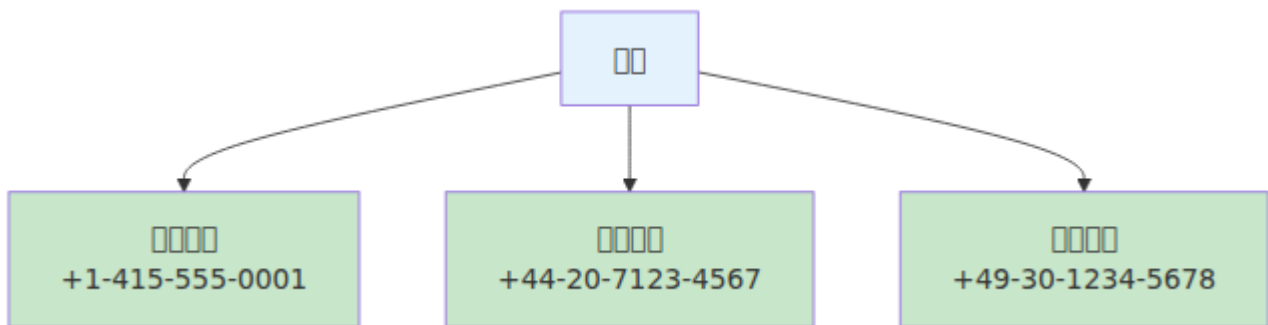
MSISDN

```
# MSISDN
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{"msisdn": {"msisdn": "14155551001"}}'
```

```
# MSISDN
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{"msisdn": {"msisdn": "14155551002"}}'
```

MSISDN

1. IMSI ID
2. MSISDN ID
3. subscriber_id msisdn_id



MSISDN

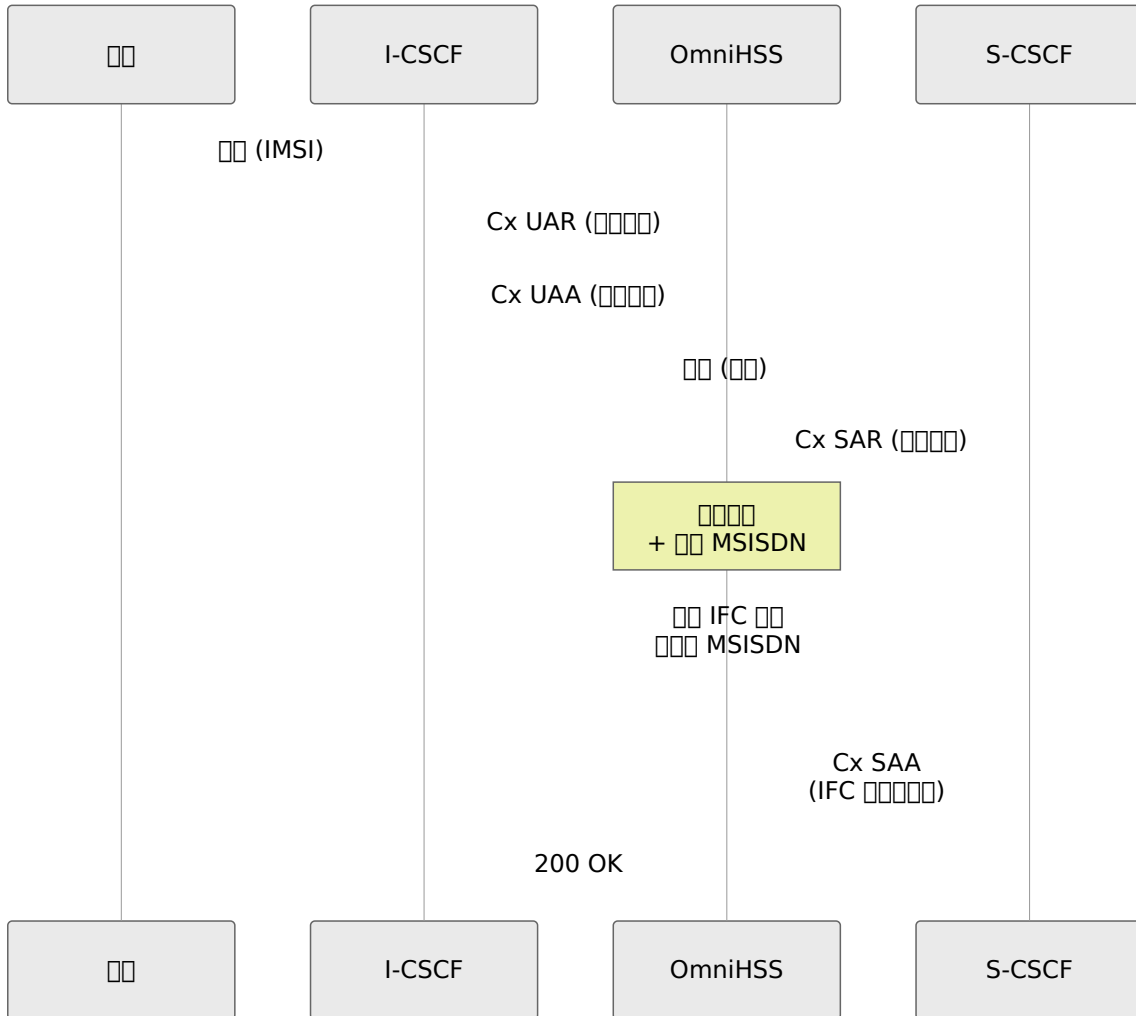
-
- msisdn
-

□□□□□ ID□IMSI □□□□□ MSISDN □□□

IMS □□

IMS □□

□□□□□ IMS □□□□□□ **MSISDN** □□□□□□ S-CSCF □ IMS □□□□□□



IFC □□□□

IMS IFC □□□□□ `{{msisdns}}` □□□□□ MSISDN□

□□ **IFC** □□□

```

<ServiceProfile>
  <PublicIdentity>
    <Identity>sip:
{{imsi}}@ims.mnc{{mnc}}.mcc{{mcc}}.3gppnetwork.org</Identity>
  </PublicIdentity>
  <!-- MSISDN -->
  <PublicIdentity>
    <Identity>sip:+14155551001@ims.example.com</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>tel:+14155551001</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>sip:+14155551002@ims.example.com</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>tel:+14155551002</Identity>
  </PublicIdentity>
  <!-- ... -->
</ServiceProfile>

```

□□□□

- `{{msisdns}}` - □□□□□□□□ MSISDN □□

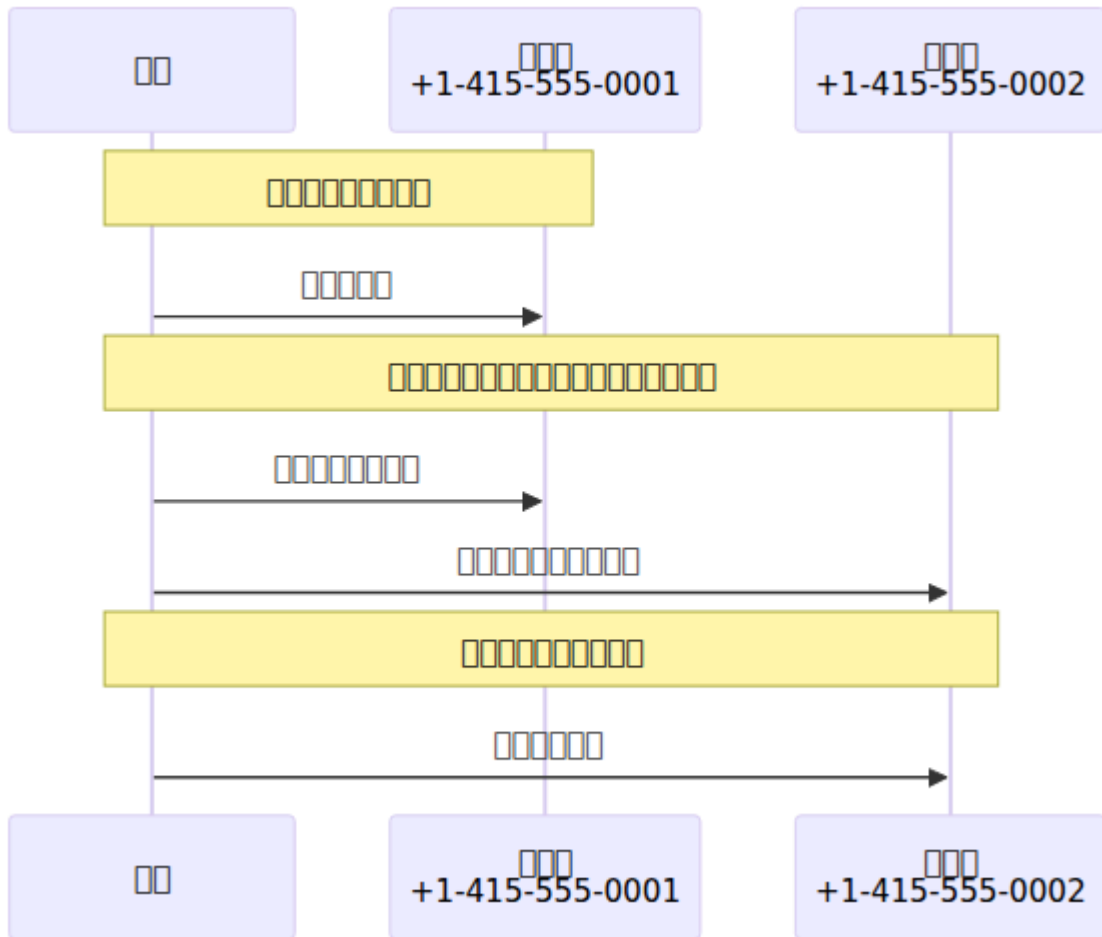
□□□□

□□ MSISDN □□□□□□□□ IMS □□□□□□



□□□□□□

□□□□□□□□□□□□ IMS □□□□□□□□□□ SIP URI□



INVITE

200 OK

SIP INVITE

```
INVITE sip:+15105551234@ims.example.com SIP/2.0
From: "+14155551002" <sip:+14155551002@ims.example.com>;tag=123
To: <sip:+15105551234@ims.example.com>
P-Asserted-Identity: <sip:+14155551002@ims.example.com>
```

From P-Asserted-Identity

MSISDN

MSISDN IMS

- S-CSCF 認證
- 認證

認證

1. 認證 **MSISDN** 認證

- 認證 IMSI 認證 MSISDN
- 認證

2. 認證 **IMS** 認證

- 認證 `{{msisdns}}` 認證
- 認證 XML

3. 認證 **HSS** 認證

- 認證 IMS 認證 Cx SAR 認證
- 認證 MSISDN 認證

4. 認證 **IMS** 認證

- 認證
- 認證 S-CSCF 認證

認證 **MSISDN** 認證

認證

- 認證
- 認證“認證”“認證”

認證

1. **MSISDN** 認證

- 認證 MSISDN 認證
- 認證 認證

2. **MSISDN** 認證

- 電話番号 MSISDN 番号
- 番号 API 電話番号 MSISDN 番号

電話番号

番号

- 番号
- 電話番号

番号

1. IMS 番号

- S-CSCF 番号
- SIP URI 番号

2. IMS 番号

- IFC 番号
- 番号

3. 番号

```
# SIP 番号
sip:+14155551001@ims.example.com # 番号
sip:+14155551002@ims.example.com # 番号
```

番号 MSISDN 番号 API 番号

番号

- API 番号 `/api/subscriber/msisdn/:msisdn` 番号

番号

番号 MSISDN 番号  番号

□□□□

□□□□

1. □□□□□ MSISDN
2. □□□□
3. □ MSISDN □□□□□
4. □□□□□□□□□□

MSISDN □□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□

IMS □□

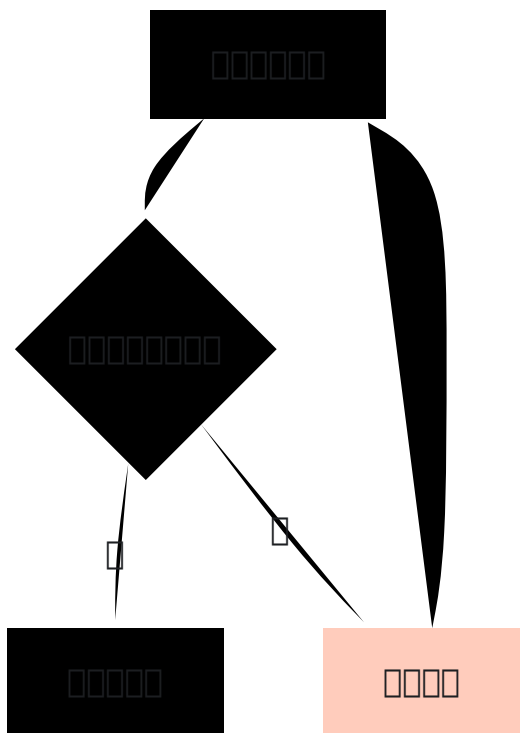
- □□ IFC □□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□

□□

□□□ MSISDN □□□□ MSISDN □□

□□□ 1 □ MSISDN

□□□□□ MSISDN



□□

□ IMSI SIM□□□□□□□□

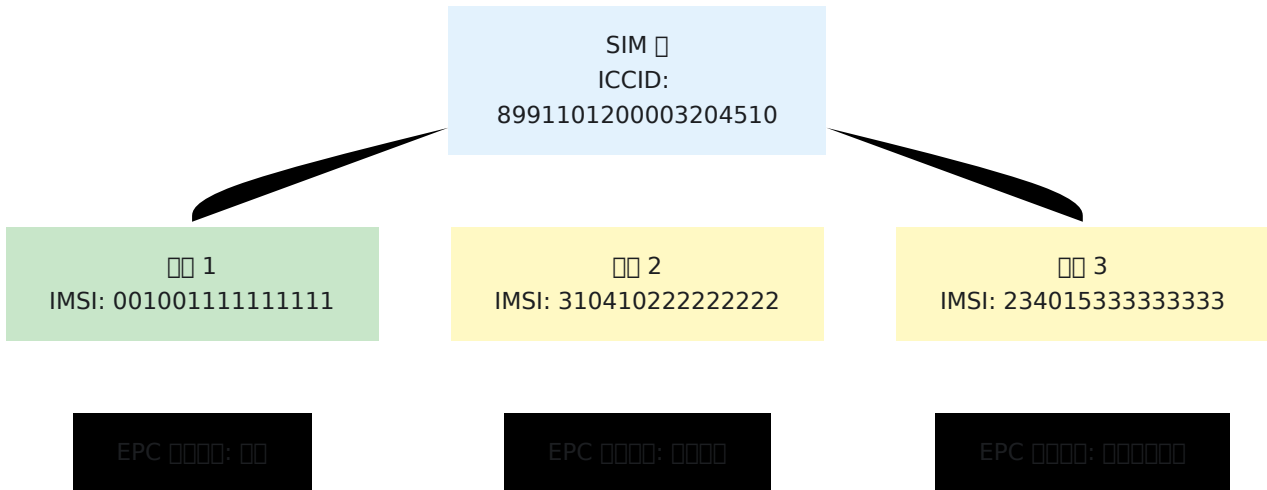
□□□□□

□ IMSI SIM □□□□□□□□□□□□□□□□□□□□□□ IMSI□□□□□□□□□□□□ IMSI □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ □□□□□□ □□□□ IMSI□□◆◆□□□□ SIM □□□□□□□□ IMSI □□HSS □□□□□□□□□□
IMSI□

OmniHSS □□

□ OmniHSS □□□ IMSI SIM □□□□ IMSI □□□□ □□□□□□□□□□□□ □□□ SIM □□



□□□□

1. □□□□□□

- □□ IMSI: 001-001□□□□□□□□
- □□□□ IMSI: 310-410□□□□□□□□
- □□□□ IMSI: 234-015□□□□□□□□
- □□□□□□□□ IMSI

2. MVNO □□

- □□ IMSI: MVNO □□□□□□□□
- □□ IMSI: □□□□□□□□
- □□ MVNO □□□□□□□□□□□□

3. IoT/M2M □□□

- IMSI 1: □□□□□
- IMSI 2: □□□□□□□□□

- IMSI 3: 00/000000
- 0000000000

4. 0000

- 0000 IMSI 0000000000
- 000000000000
- 0000000000000000

0 IMSI 00

0000

- 00 IMSI 000000 KiOPC 0000
- 00 IMSI 00000000
- 00000000000000

00000000

- 0000 EPC 00000000APN
- 0000 IMS 000000000000
- 00 IMSI 00000000

000000

- 00 IMSI 000000 SIM0000 sim_id
- 00000000000000 ICCID
- 00 SIM 00000000

0000

- 0000 SIM 00000000 IMSI
- 00000000000000
- HSS 0000000000 IMSI



```
# 1. SIM ID를 IMSI로
SIM_ID=$(curl -k -X POST https://hss.example.com:8443/api/sim \
  -d '{"sim": {"iccid": "8991101200003204510", "is_esim": false}}' \
  | jq -r '.data.id')

# 2. IMSI 1에 키셋을 할당
KEYSET1=$(curl -k -X POST https://hss.example.com:8443/api/key_set \
  -d '{"key_set": {"ki": "0123456789ABCDEF...", "opc": \
  "FEDCBA9876..."}}' \
  | jq -r '.data.id')

# 3. IMSI 1에 구독자 생성
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -d '{"subscriber": {
    "imsi": "\"001001111111111\"",
    "sim_id": "$SIM_ID",
    "key_set_id": "$KEYSET1",
    "epc_profile_id": 1
  }}'

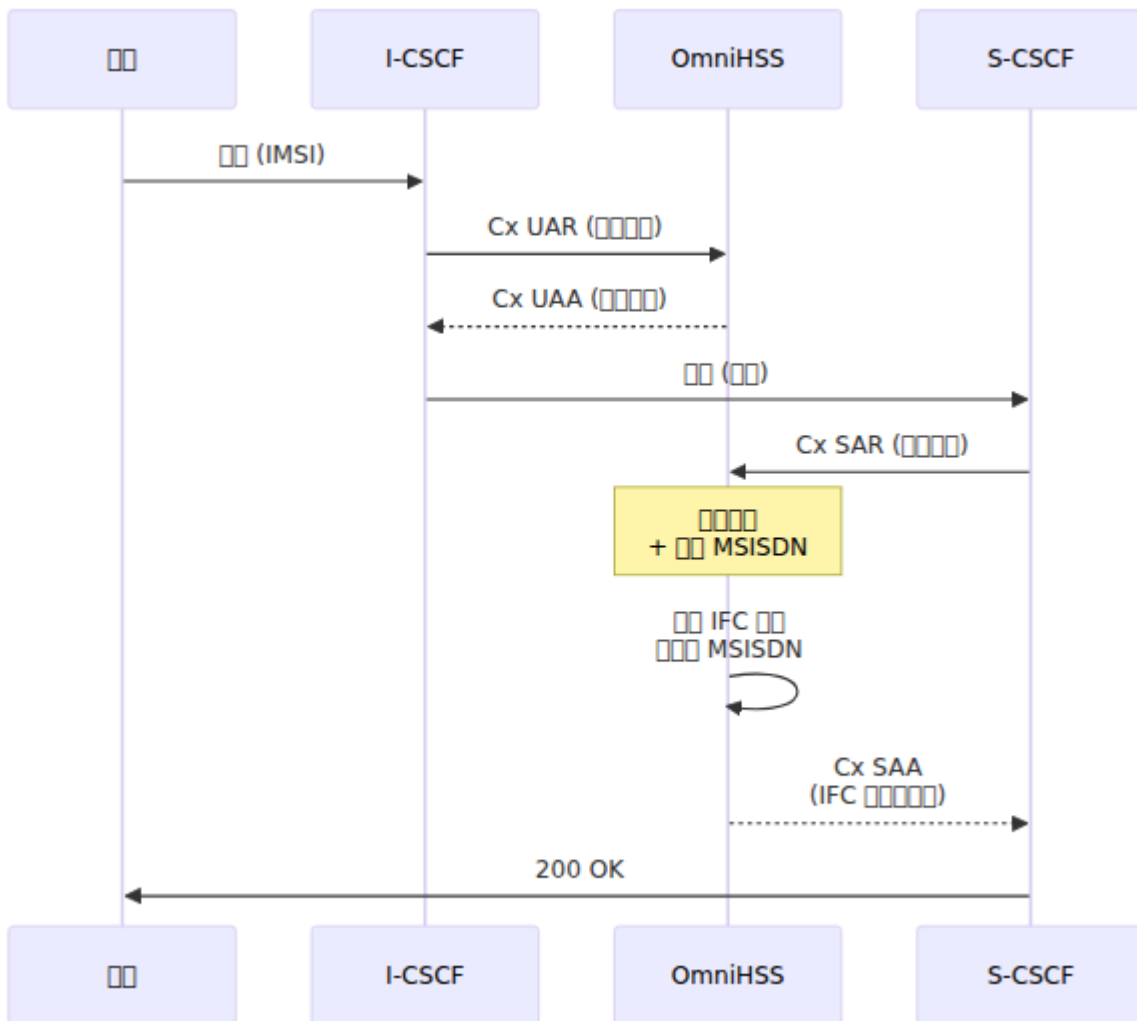
# 4. IMSI 2에 키셋을 할당
KEYSET2=$(curl -k -X POST https://hss.example.com:8443/api/key_set \
  -d '{"key_set": {"ki": "1111111111111111...", "opc": \
  "2222222222..."}}' \
  | jq -r '.data.id')

# 5. IMSI 2에 구독자 생성
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -d '{"subscriber": {
    "imsi": "\"310410222222222\"",
    "sim_id": "$SIM_ID",
    "key_set_id": "$KEYSET2",
    "epc_profile_id": 2
  }}'

# 6. SIM ID를 IMSI로...
```

□□□□

□□ IMSI □□□□□□



HSS □□□□□□□□□□ IMSI SIM——□□□□□□□□□□ IMSI□

IMSI □□□□□□□□

□□ IMSI SIM □□□ IMSI □□□□□□ IMSI □□□□□□□□□□ IMSI□ OmniHSS □□□ HSS □□ □□□□ □□□ CLR □□□□□□□□□□□□□□□□ IMSI□

□□□□ IMSI □□

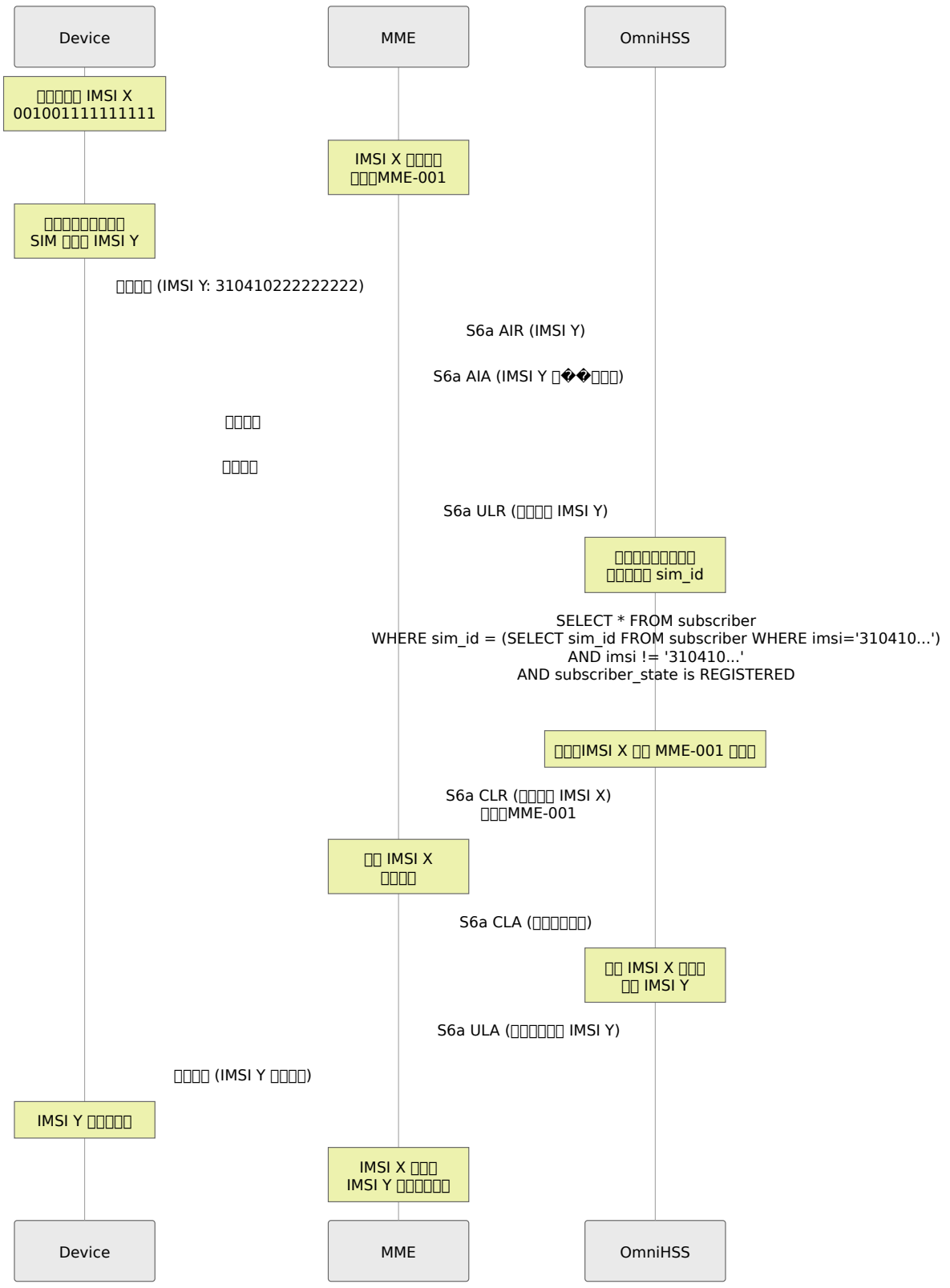
□□□□□ □□ SIM □□□□□□□□□□ IMSI□□□□□□□□□□□□□□□□

- □□□□□ MME □□□□ IMSI X
- □□ HSS □□□□ IMSI Y □□□□□□□□□□ IMSI X □□□□ SIM □□

- HSS 00000 0000000 000 IMSI X

0000 IMSI 00000000000000000000

IMSI 0000



□□□□□□

□□□□□□

- □□□□□□ SIM □□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□

□□□□□□

- □□□□□□□□ IMSI □□□□□□□□□□
- IMSI □□□□□□□□□□□□
- □□□□ CDR□□□□□□□□□□□□

□□□□□□

- □ IMSI □ MME □□□□□□
- PDP □□□□□□□□□□□□
- □□□□□□□□□□□□

IMSI □□□□□□

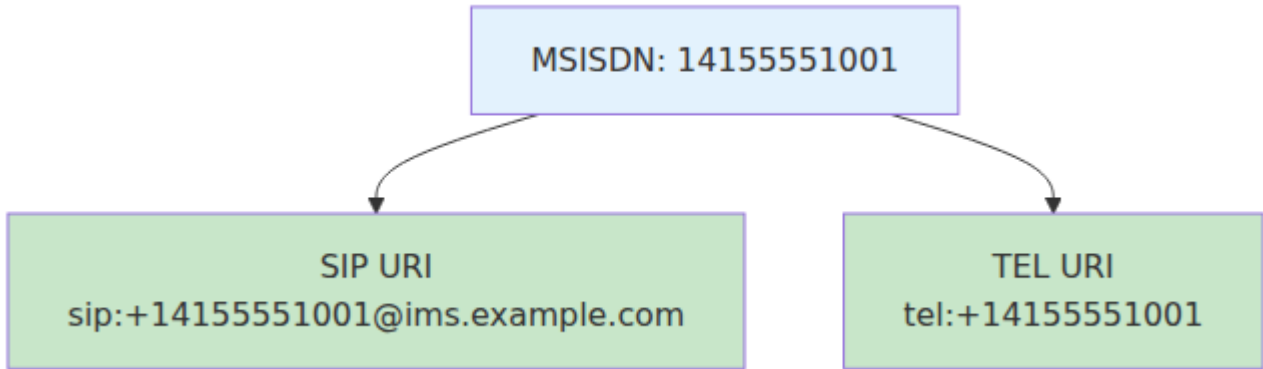
□□/□□□□□□ IMSI□□□□□□

1. □□□□□□
 - □□ IMSI □□□□□□
 - □□□□□□□□□□ IMSI
2. □□□□□□
 - □□□□□□□□□□
 - SIM □□□□□□□ IMSI
3. □□□□□□
 - SIM □□□□□□□□□□□□□□□□□□ IMSI□
 - □□ MCC/MNC □□□□□□
4. □□□□□□

- IMSI
- IMSI

IMS

IMS



IMS

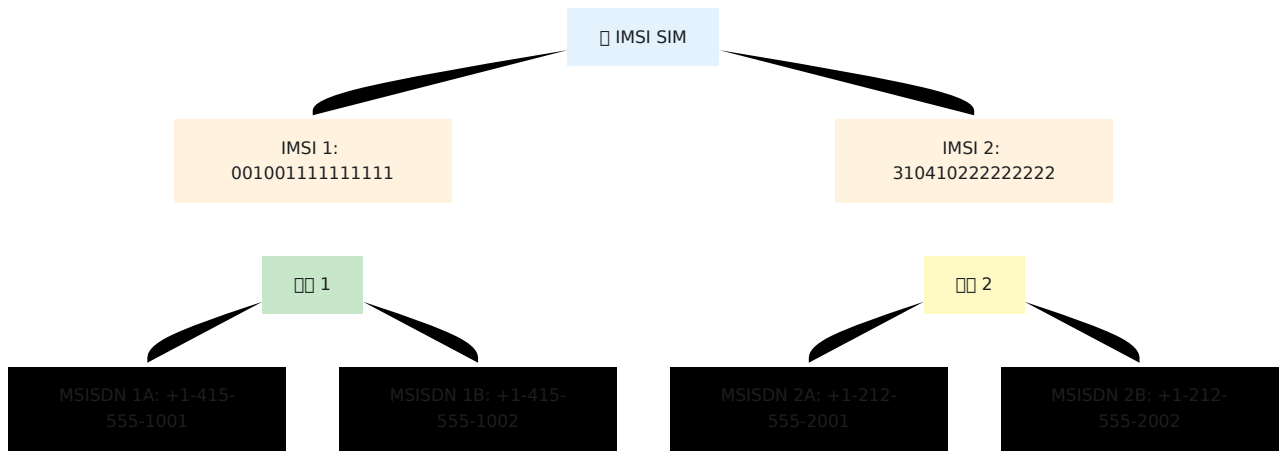
IMS

1. IMSI HSS “”
2. SIM IMSI SIM `sim_id` “”
3. `subscriber_state` IMSI MME/SGSN IMSI
4.
 - IMSI
 - IMSI
 - SIM IMSI

IMS

IMSI + MSISDN

IMS IMSI MSISDN



□□□□

- **???** □□□□ **IMSI 1** □□
 - □□□□+1-415-555-1001
 - □□□□+1-415-555-1002
- □□□□□□ **IMSI 2** □□
 - □□□□+1-212-555-2001
 - □□□□+1-212-555-2002

□□□□□□□□ IMSI 1 □□ MSISDN□□□□□□□□□□ IMSI 2 □□□□□□□□□□ MSISDN□

□□□□

□□□ **MSISDN** □□

□□□□□□ MSISDN□

```

□□ API □□□ GET /api/subscriber/imsi/:imsi
  
```

□□□□□□□□ MSISDN□

IMS

IMS

- IMS
- IMS key_set
- EPC
-

IMS

- /SIM HSS
- HSS IMSI
- IMSI

MSISDN

- MSISDN
- IMS `{{msisdns}}`
- IMS
- S-CSCF

- - HSS
 - HSS
-

□□□□

□ MSISDN □□

- ✓ □□ SIM□□□□□□□□
- ✓ □□□□□□□□□□
- ✓ □□□□□□
- ✓ □□□□□□
- ✓ □□□□□□□□□□□□□□
- ✓ □ IMSI □□□□

□ IMSI SIM □□

- ✓ □□□□□□
- ✓ □□□□□□
- ✓ □□□□□□□□
- ✓ □□□□□□
- ✓ □□□
- ✓ □□□□□□□□□□

□□□□

- ✓ □□□□□□
- ✓ □□□□□□□□□□
- ✓ □□□□□□□□
- ✓ □□□□□□□□
- ✓ □□□□□□□□

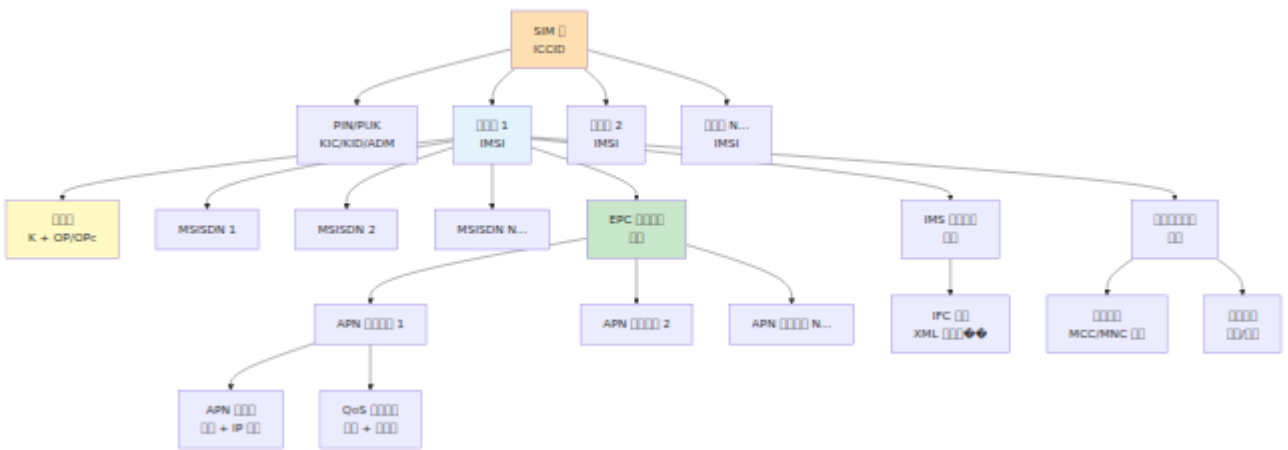
← □□□□□□

OmniHSS

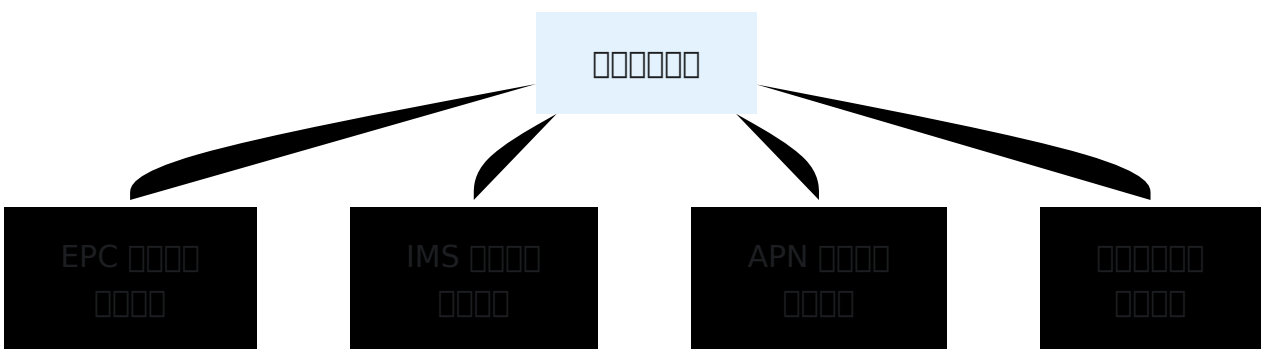
←

OmniHSS

OmniHSS is a central database for storing and managing subscriber information, including IMSI, APN, IMS, and other network-related data.



OmniHSS



EPC

EPC is a core network element that provides IP-based services to mobile devices.

속성

속성명	유형	범위
<code>ue_ambr_dl_kbps</code>	정수 정수	10,000 - 1,000,000 Kbps
<code>ue_ambr_ul_kbps</code>	정수 정수	5,000 - 500,000 Kbps
<code>network_access_mode</code>	정수 문자열	"packet_only" 또는 "packet_and_circuit"
<code>tracking_area_update_interval_seconds</code>	TAU 정수	54 초

예시 EPC 생성

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
-H "Content-Type: application/json" \
-d '{
  "apn_profiles": [],
  "name": "Premium 100Mbps",
  "network_access_mode": "packet_only",
  "tracking_area_update_interval_seconds": 600,
  "ue_ambr_dl_kbps": 100000,
  "ue_ambr_ul_kbps": 50000
}'
```

예시 EPC 속성

속성:

- 속성: 10 Mbps (10,000 Kbps)
- 속성: 5 Mbps (5,000 Kbps)

속성:

- 5G: 50 Mbps (50,000 Kbps)
- 4G: 25 Mbps (25,000 Kbps)

3G:

- 3G: 100 Mbps (100,000 Kbps)
- 2G: 50 Mbps (50,000 Kbps)

2G:

- 2G: 1 Gbps (1,000,000 Kbps)
- 1G: 500 Mbps (500,000 Kbps)

IMS 架构

IMS 架构由 IFC 和 S-CSCF 组成

IFC 参数

IFC 参数由 S-CSCF 通过 XML 消息

参数:

- `{{imsi}}` - IMSI
- `{{msisdns}}` - 域名
- `{{mcc}}` - MCC
- `{{mnc}}` - MNC

IMS

```
curl -k -X POST https://hss.example.com:8443/api/ims/profile \  
-H "Content-Type: application/json" \  
-d '{  
  "ims_profile": {  
    "name": "Standard VoLTE",  
    "ifc_template": "<InitialFilterCriteria>...  
</InitialFilterCriteria>"  
  }  
'
```

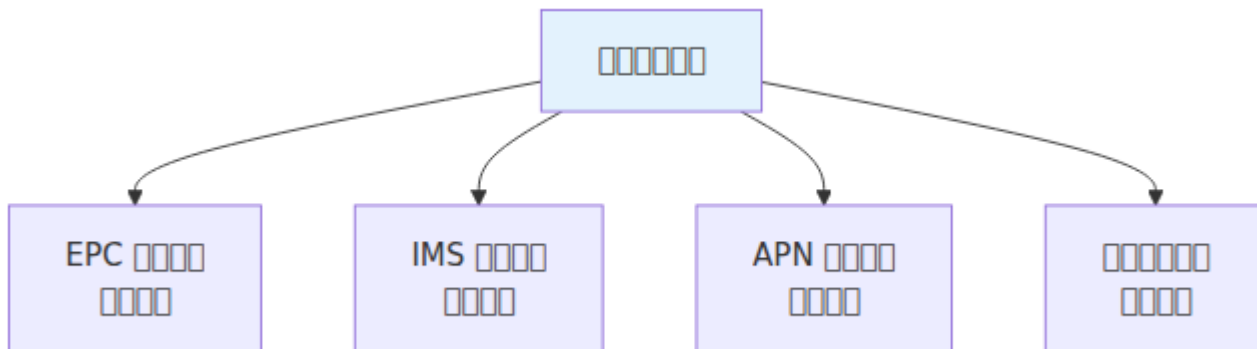
IFC

```
<ServiceProfile>  
  <PublicIdentity>  
    <Identity>sip:  
{{imsi}}@ims.mnc{{mnc}}.mcc{{mcc}}.3gppnetwork.org</Identity>  
  </PublicIdentity>  
  <InitialFilterCriteria>  
    <Priority>0</Priority>  
    <TriggerPoint>  
      <ConditionTypeCNF>0</ConditionTypeCNF>  
      <SPT>  
        <ConditionNegated>0</ConditionNegated>  
        <Group>0</Group>  
        <Method>INVITE</Method>  
      </SPT>  
    </TriggerPoint>  
    <ApplicationServer>  
      <ServerName>sip:as.ims.example.com</ServerName>  
      <DefaultHandling>0</DefaultHandling>  
    </ApplicationServer>  
  </InitialFilterCriteria>  
</ServiceProfile>
```

APN 网络

APN网络结构图

APN 网络



APN 网络

APN 网络 IP 地址

APN:

- internet - 互联网
- ims - IMS/VoLTE
- mms - 彩信
- vzwadmin - 运营商

IP 地址:

- "ipv4": IPv4
- "ipv6": IPv6
- "ipv4v6": IPv4v6
- "ipv4_or_ipv6": IPv4 或 IPv6

APN QoS 网络

网络 QoS

QCI QoS :

QCI	QoS	QoS	QoS
1	GBR	GBR	GBR
2	GBR	GBR	GBR
4	GBR	GBR	GBR
5	GBR	IMS	GBR
9	GBR	GBR	GBR

📡 APN 📡

```
# 1. 📡 APN 📡
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{"apn": "internet", "ip_version": "ipv4v6"}' \
  | jq -r '.response.id')

# 2. 📡 APN QoS 📡
QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Best Effort",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 50000,
    "apn_ambr_ul_kbps": 25000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "qci": 9
  }' | jq -r '.response.id')

# 3. 📡 APN 📡
curl -k -X POST https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_identifier_id\": $APN_ID,
    \"apn_qos_profile_id\": $QOS_ID,
    \"name\": \"Internet APN\"
  }"
```

📡 APN 📡 EPC 📡

APN 📡 `join_epc_profile_to_apn_profile` 📡 EPC 📡

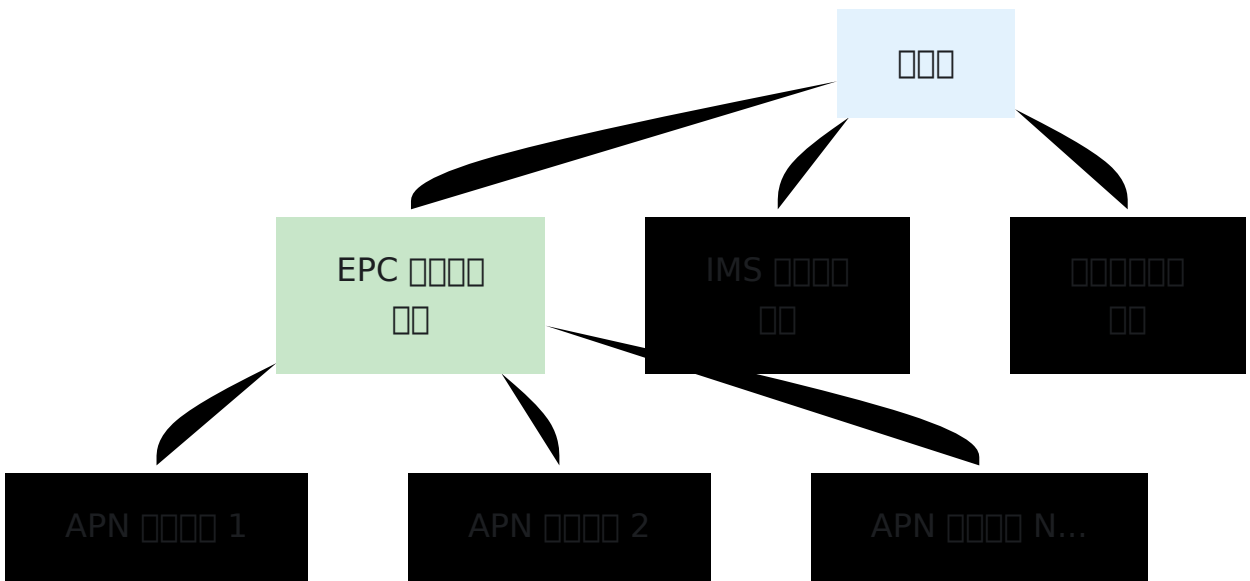
📡 APN 📡 ID 📡 EPC 📡 ID 📡 APN 📡 EPC 📡

□□□□□□

□□□ □□□□□ □□□□□□

□□□□□□

□□□□□□□□



POST /api/subscriber

```
# Create subscriber EPC & IMS profile
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "key_set_id": 1,
      "epc_profile_id": 1,
      "ims_profile_id": 1,
      "roaming_profile_id": 1
    }
  }'
```

```
# Update subscriber EPC profile
curl -k -X PUT https://hss.example.com:8443/api/subscriber/1 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "epc_profile_id": 2
    }
  }'
```

POST /api/subscriber

Steps

1. Create subscriber - IMS profile
2. Create subscriber - EPC profile
3. Update subscriber - EPC profile
4. Create subscriber - IMS profile

Network Architecture

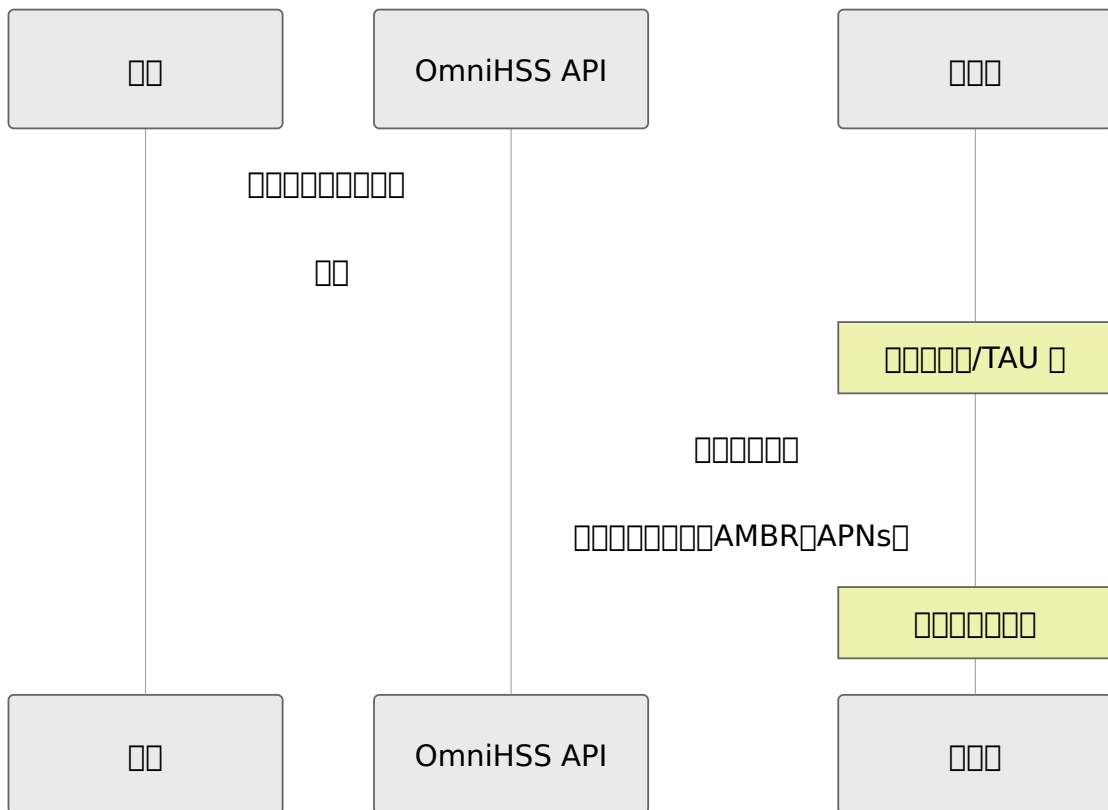
[Network] - [Core] - [Access]

Network:

- "Basic-10Mbps-Internet"
- "Premium-100Mbps-VoLTE"
- "Enterprise-1Gbps-MultiAPN"

Core Network

Network Architecture



Network: Network/TAU

- Network/TAU
- Network
- IMS Network IMS Network

□□□□□□□□□□

□□□□□□□□□□:

1. □□□□□ EPC □□□□ AMBR □
2. □□ APN QoS □□□□ AMBR □
3. □□ MME/P-GW □□□□□□□ QoS
4. □□□□□□

IMS □□□□:

1. □□□□□□□ IMS □□□□
2. □□ IFC □□ XML □□□□
3. □□ S-CSCF □□□□ IFC □□□□
4. □□ S-CSCF □□□□

APN □□□:

1. □□ APN □□□□□□□□□□ EPC □□□□
2. □□ APN □□□□□□□□□□□□
3. □□ UE □ PDN □□□□

OmniHSS □□□□

← □□□□□□

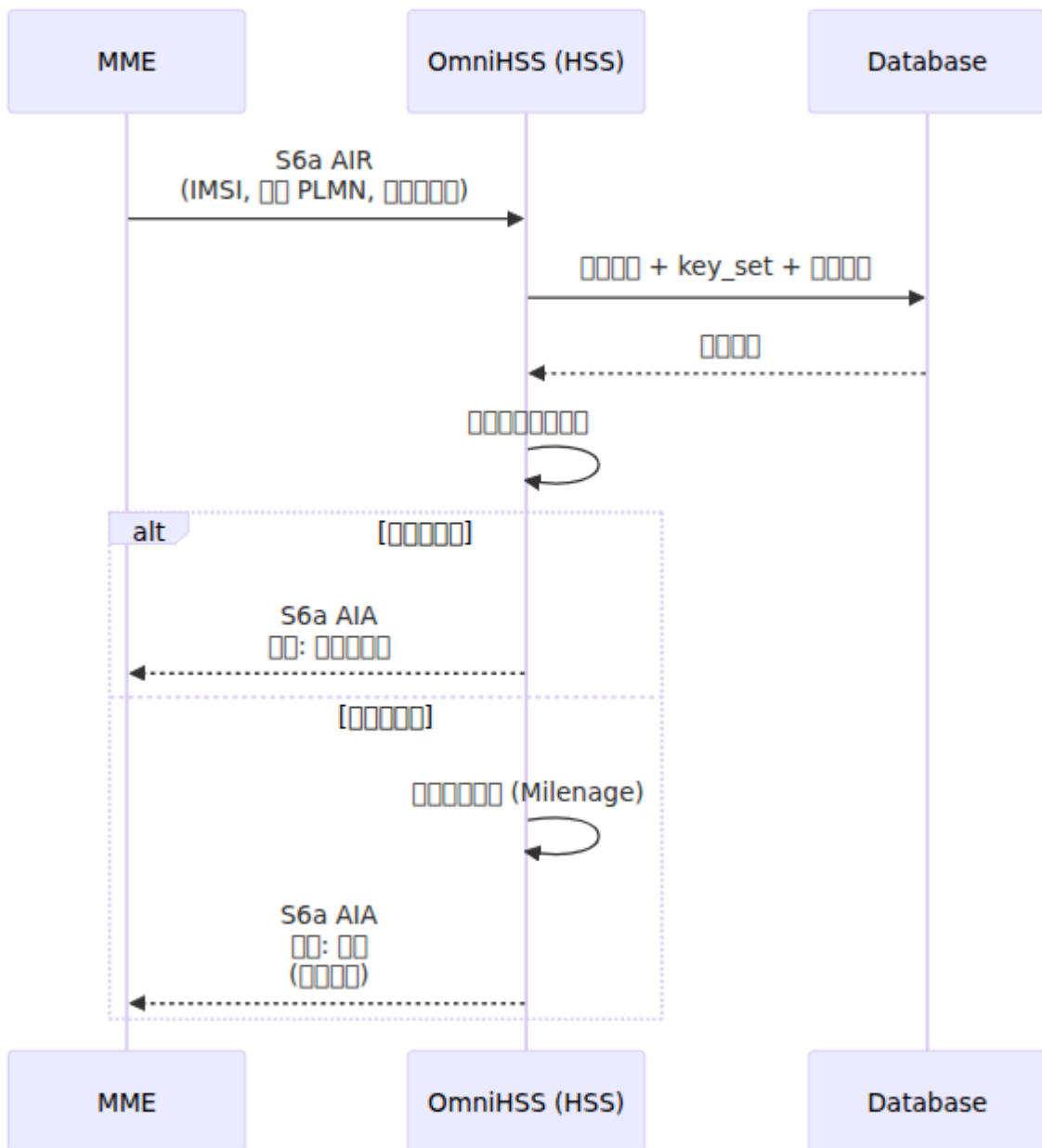
□□

□□□□□□□□ OmniHSS □□□ Diameter □□□□□□□□□□□□□□□□□□□□□□□□□□□□

S6a □□ (LTE/EPC)

□□□□□□ (AIR/AIA)

MME □□□□□□□□□□

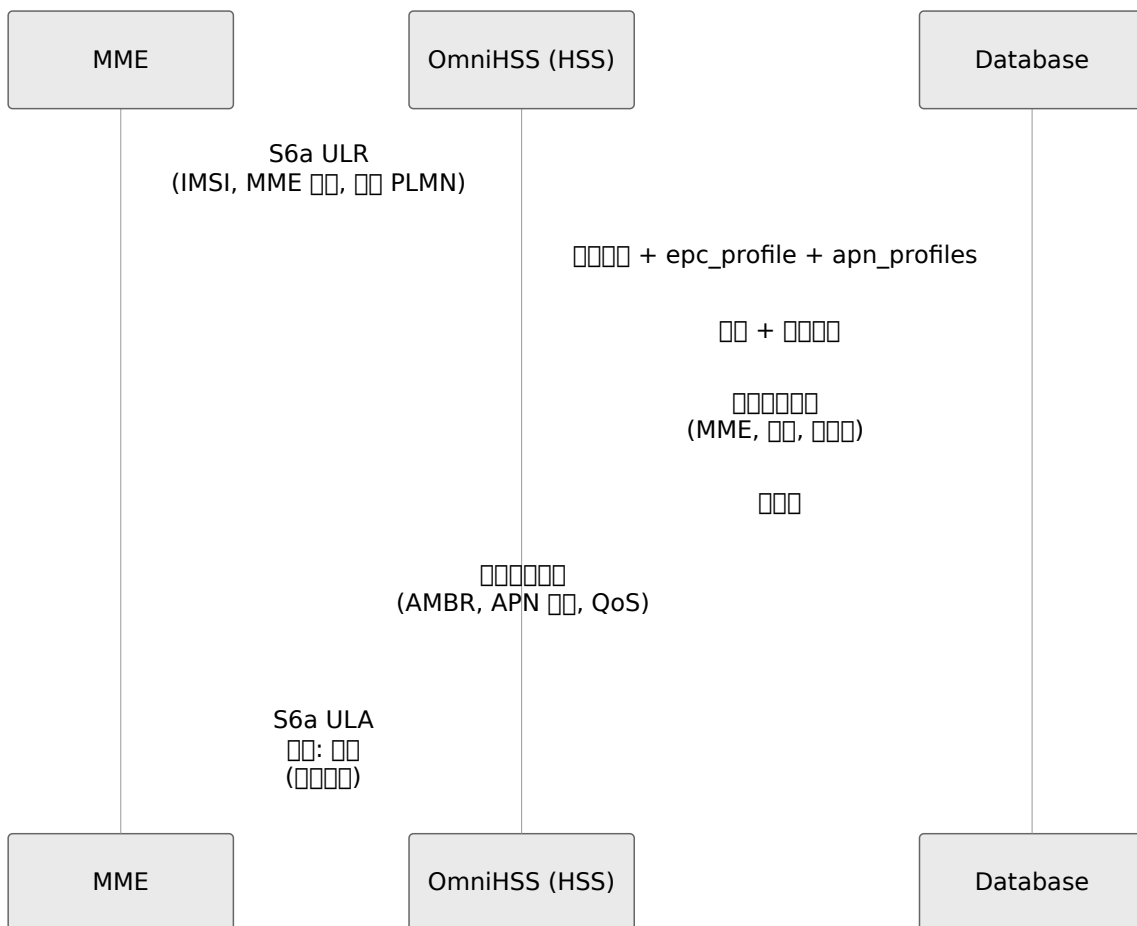


AVP:

- IMSI, PLMN-Id, ...
- RAND, AUTN, XRES, KASME

(ULR/ULA)

MME HSS

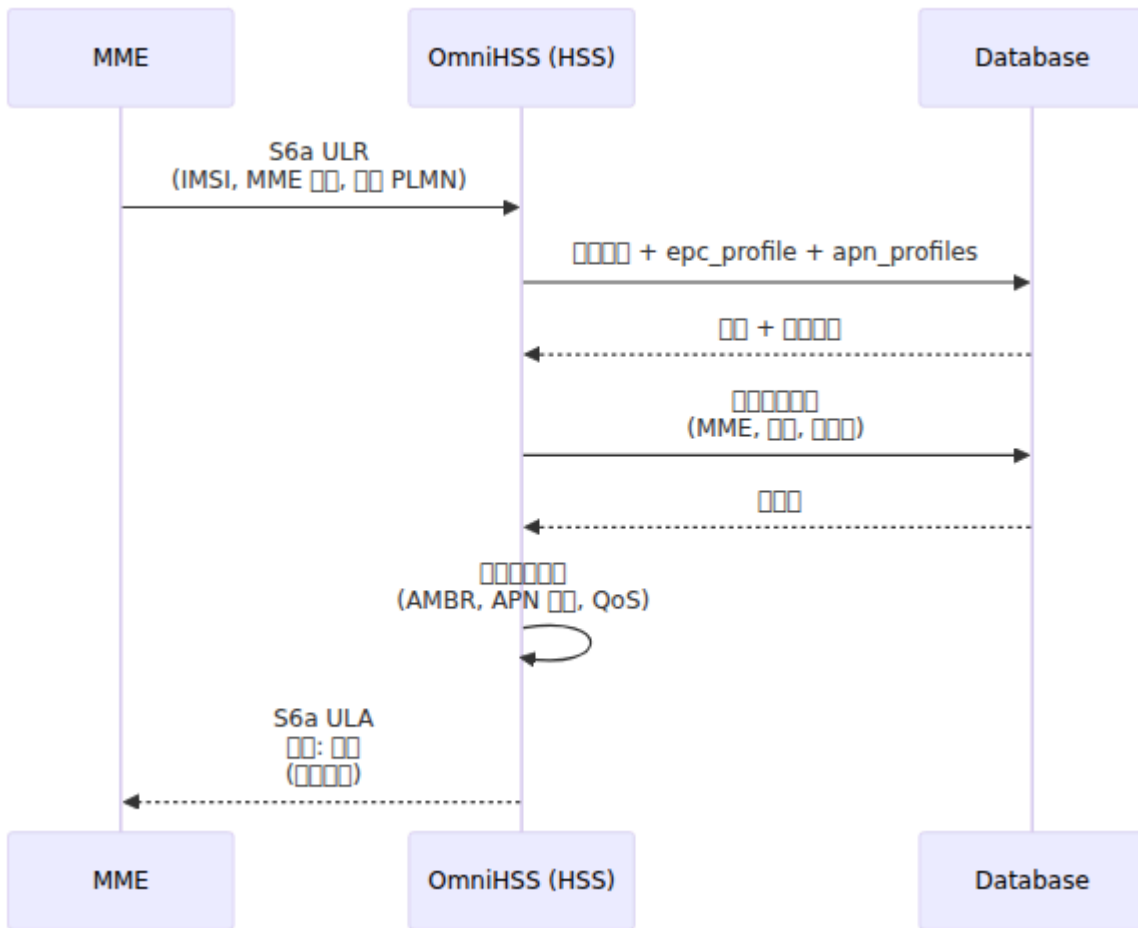


AVP:

- ID: ID (IMSI), RAT ID, ULR ID, ID-PLMN-Id, UE-SRVCC ID
- ID: ID (AMBR, APN ID, ID)

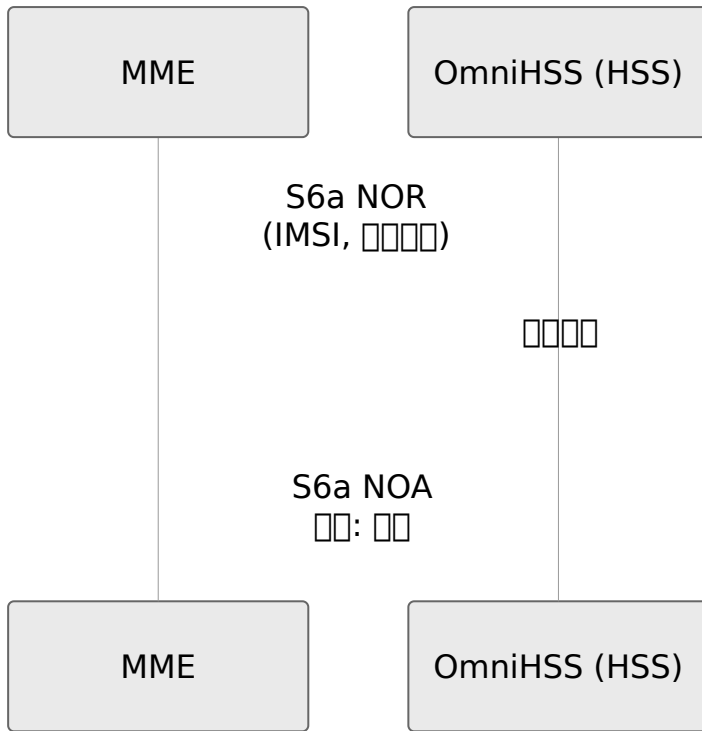
UE ID (PUR/PUA)

MME ID HSS ID



Sequence (NOR/NOA)

MME ID HSS ID

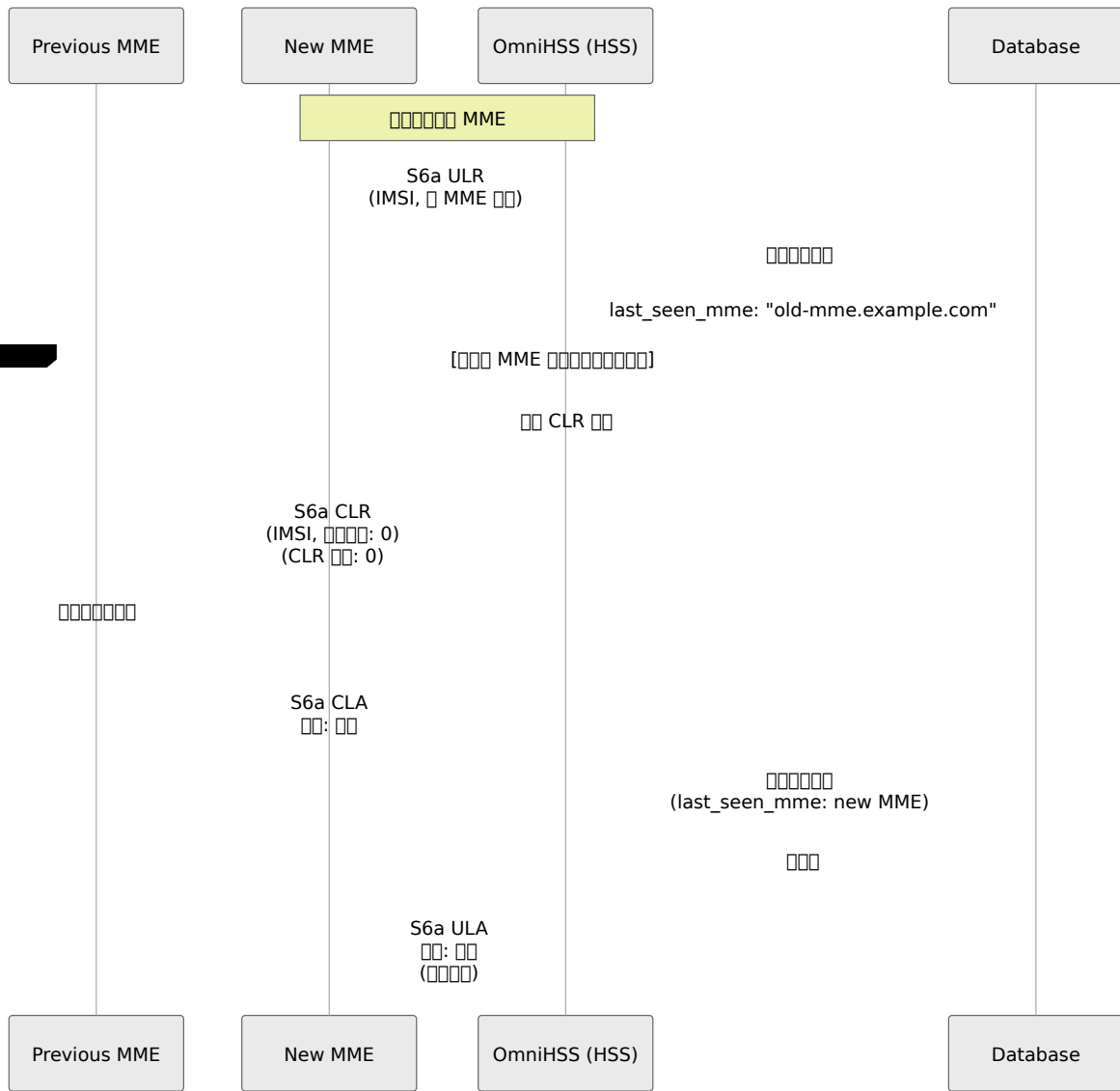


□□□□□□ (CLR/CLA)

HSS □□□□□□□□ MME □□□□□□□□ OmniHSS □□□□□□□□ CLR □□□

□□ CLR (MME □□)

□□□□□□ MME □□□□□□□□□□ OmniHSS □□□~~□□□~~□□□ MME □□ CLR □□□□□□□□□□

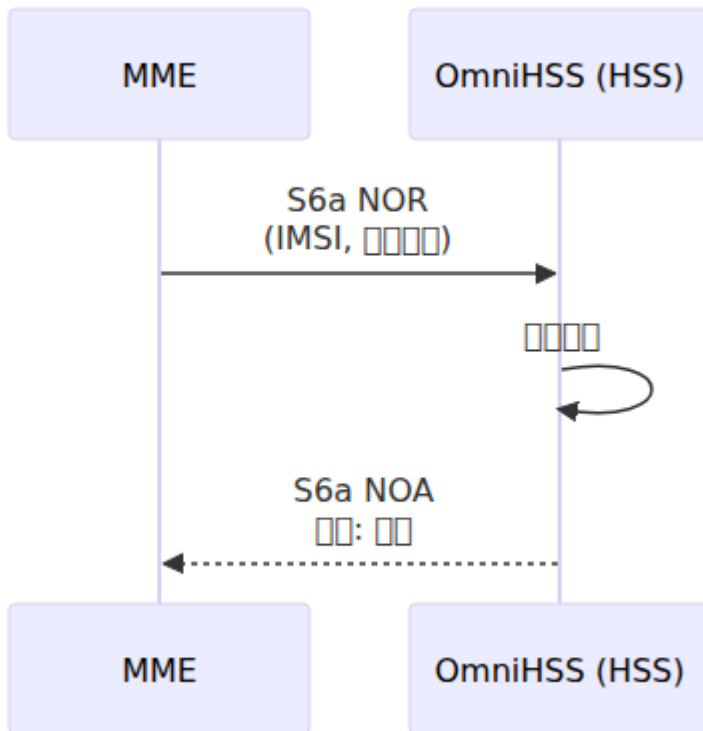


AVP (CLR):

- IMSI: IMSI
- MME ID: MME ID
- MME ID: MME ID
- MME ID: 0 (MME ID)
- CLR ID: 0
- MME ID: MME ID

CLR (API)

API CLR



□□ AVP (□□□ CLR):

- □□□: □□□ IMSI
- □□□□: □□□□□ MME □□□
- □□□□: □□□□□ MME □
- □□□□: `:subscription_withdrawal` (□ 3GPP TS 29.272 □□□□□)
- CLR □□:
 - s6a_indicator: 1 (□□□□ S6a □□)
 - reattach_required: 1 (UE □□□□□□□□□□)

□□□□

OmniHSS □□□ 3GPP TS 29.272 □□□□□□□□□□

消息	消息 ID	消息名称	消息内容
MME 消息	0	MME 消息	MME 消息 ULR
SGSN 消息	1	SGSN 消息	3G/2G 消息
消息	2	消息	API 消息
消息 IWF	3	消息	消息
消息	4	消息	消息

CLR 消息

CLR 消息 AVP 消息

消息	消息 ID	消息名称
S6a/S6d 消息	0	1 = S6a 消息
消息	1	1 = UE 消息

消息 CLR 消息:

```
clr_flags: %{
  s6a_indicator: 1,      # S6a 消息
  reattach_required: 1  # 消息
}
```

IMSI 消息

OmniHSS 消息 (IMSI) 消息 MME 消息 MSISDN 消息 IMSI 消息 CLR 消息

消息 1: 消息 MSISDN 消息 IMSI

- IMSI 000 MME 00000

000000

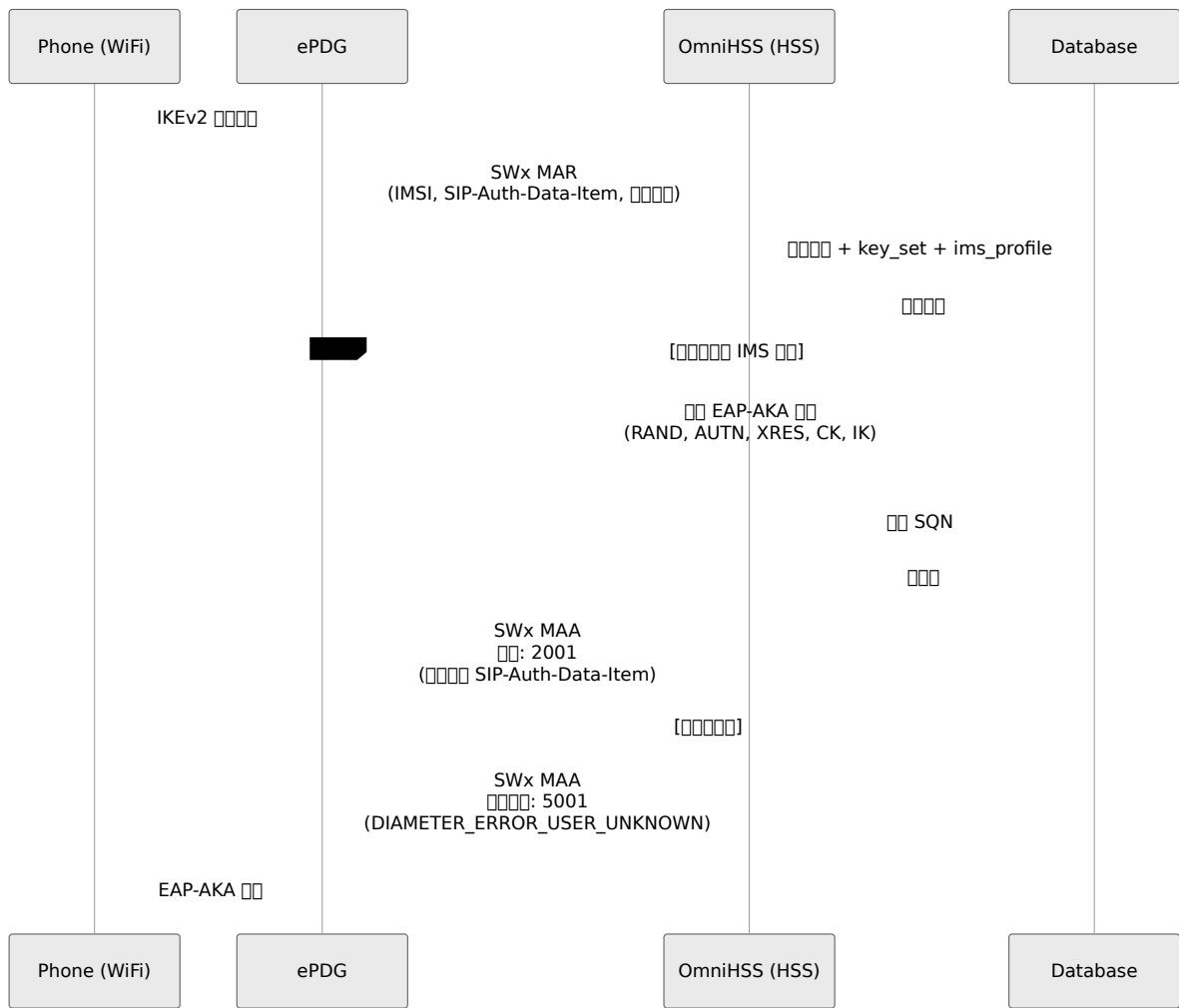
1. **IMSI** 000: CLR 00000 0 **IMSI**00000 MSISDN subscriber_state 0000 (IMSI) 00 last_seen_mme 0
2. 0000: 000000000000 MME 00000 CLR 0000000000000000
3. 0000000 **MME** 0000 **CLR:** 00 last_seen_mme 0 nil 000000000000 ULR 00000 CLR 0
4. 0000000: 00 CLR 00 ULR 0000000000 Subscription-Data AVP 00000 MME 00000 000
5. 00: CLR 0000000000000000 MME 0 ULA 00000 00000 MME 0 CLA 0
6. **CLA** 00: OmniHSS 00 CLA 0000000000000000 398 00 :discard 000000000000 0000 HSS 000

SWx 00 (0 3GPP 00 / WiFi 00)

SWx 00000 ePDG 0000000000000000 HSS 00000 WiFi 000000000000

00000000 (MAR/MAA)

ePDG 00 EAP-AKA 00000000 WiFi 00000



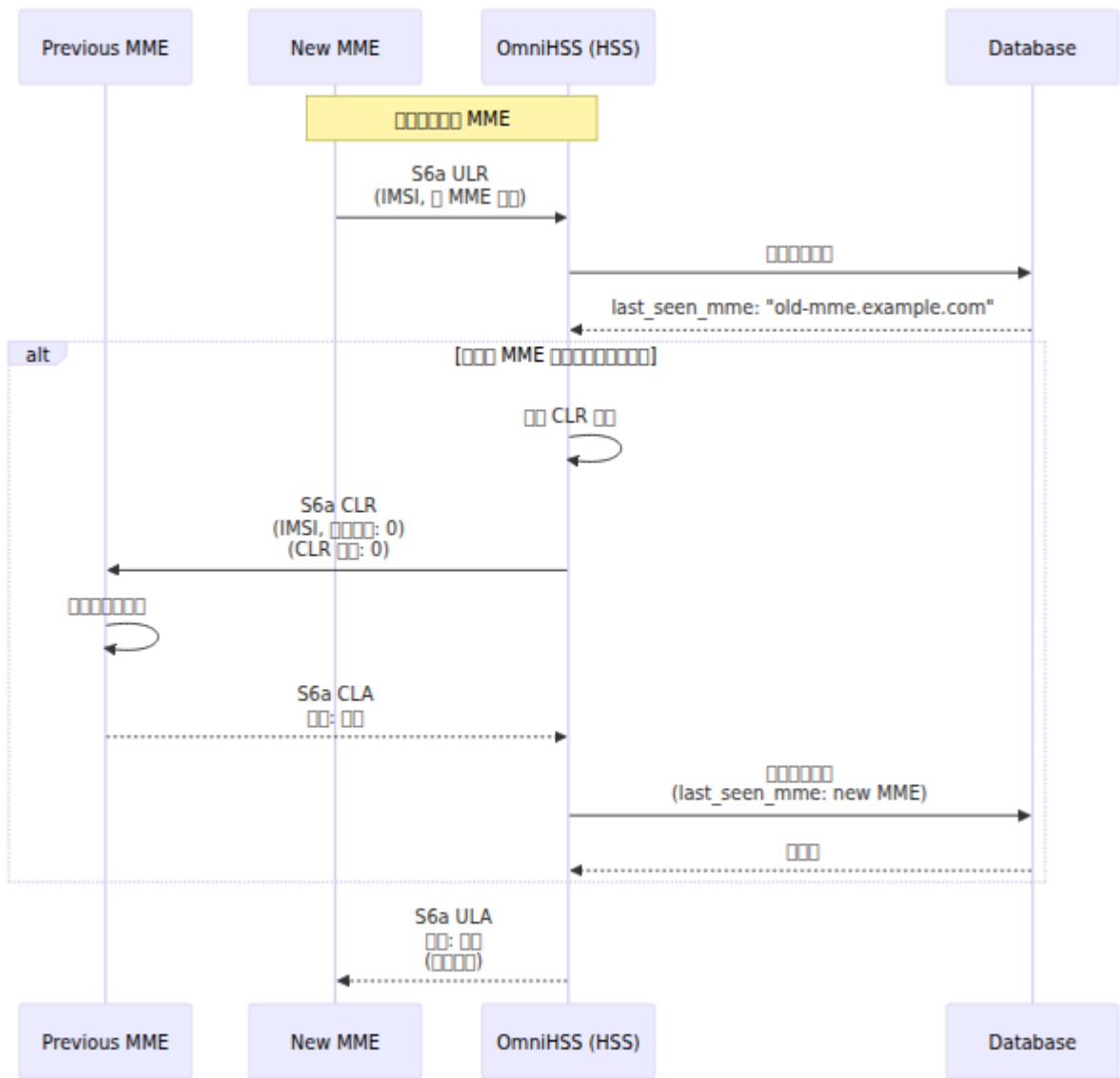
AVP:

- IMSI: IMSI (IMSI @ IMSI@realm), SIP-Auth-Data-Item (SIP-Number-Auth-Items)
- SIP: SIP-Auth-Data-Item (SIP-Authenticate, SIP-Authorization, SIP-Auth-Data-Item, SIP-Auth-Data-Item)

SQN (Sequence Number): UE sends SQN to ePDG for SIP-Authorization AVP. ePDG sends SQN to OmniHSS (HSS) via AUTS (3GPP TS 33.102) and IND (SQN).

(SAR/SAA)

ePDG sends SAR/SAA to WiFi.



[]: []:

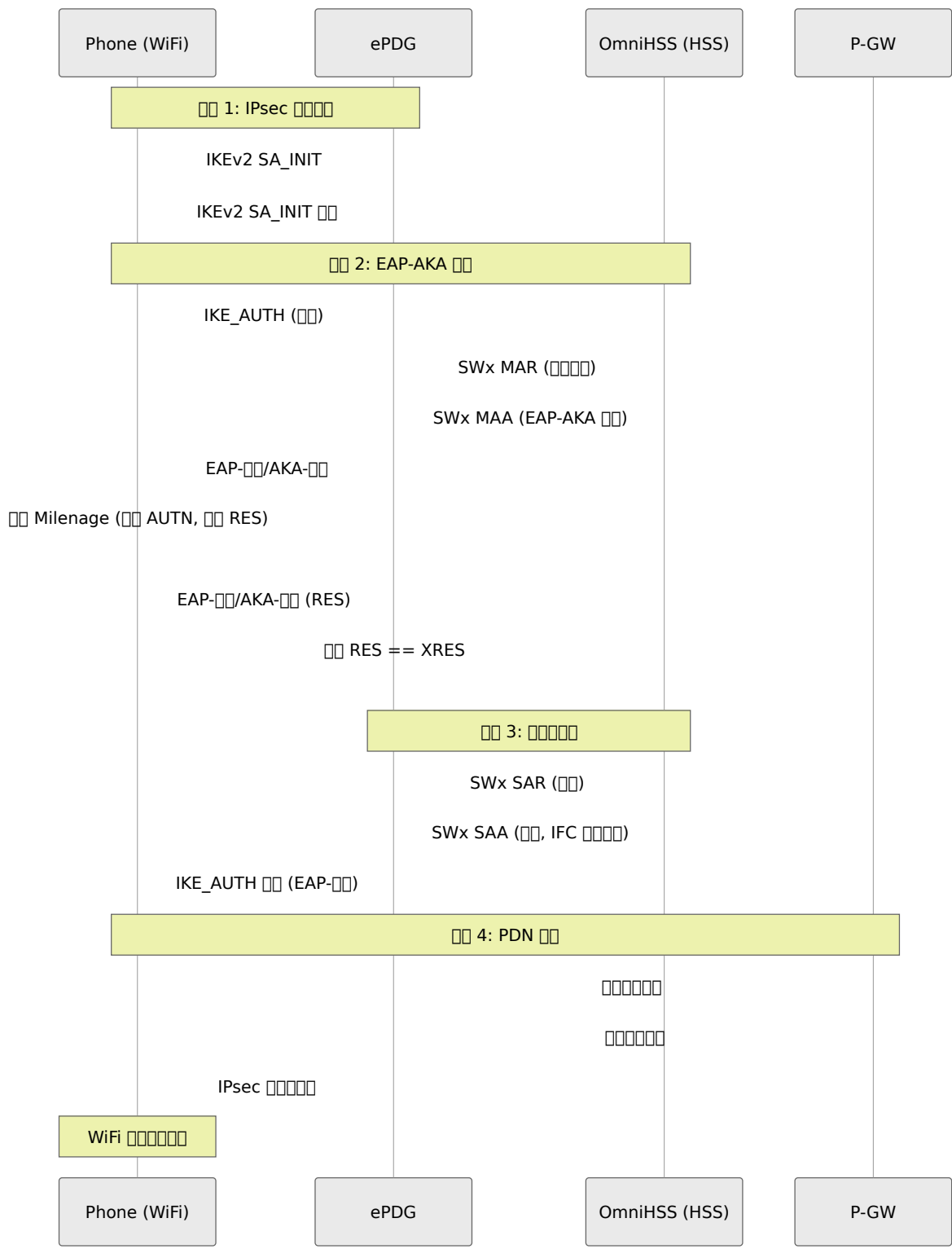
코드	번호	설명
NO_ASSIGNMENT	0	할당되지 않음
REGISTRATION	1	WiFi 등록
RE_REGISTRATION	2	재등록
UNREGISTERED_USER	3	등록되지 않은 사용자
TIMEOUT_DEREGISTRATION	4	시간 초과 탈퇴
USER_DEREGISTRATION	5	사용자 탈퇴
AUTHENTICATION_FAILURE	7	인증 실패
ADMINISTRATIVE_DEREGISTRATION	8	관리자 탈퇴

AVP:

- 코드: IMSI, ICCID, ePDG 코드, APN
- 코드: IFC XML, 3GPP 코드

WiFi 등록

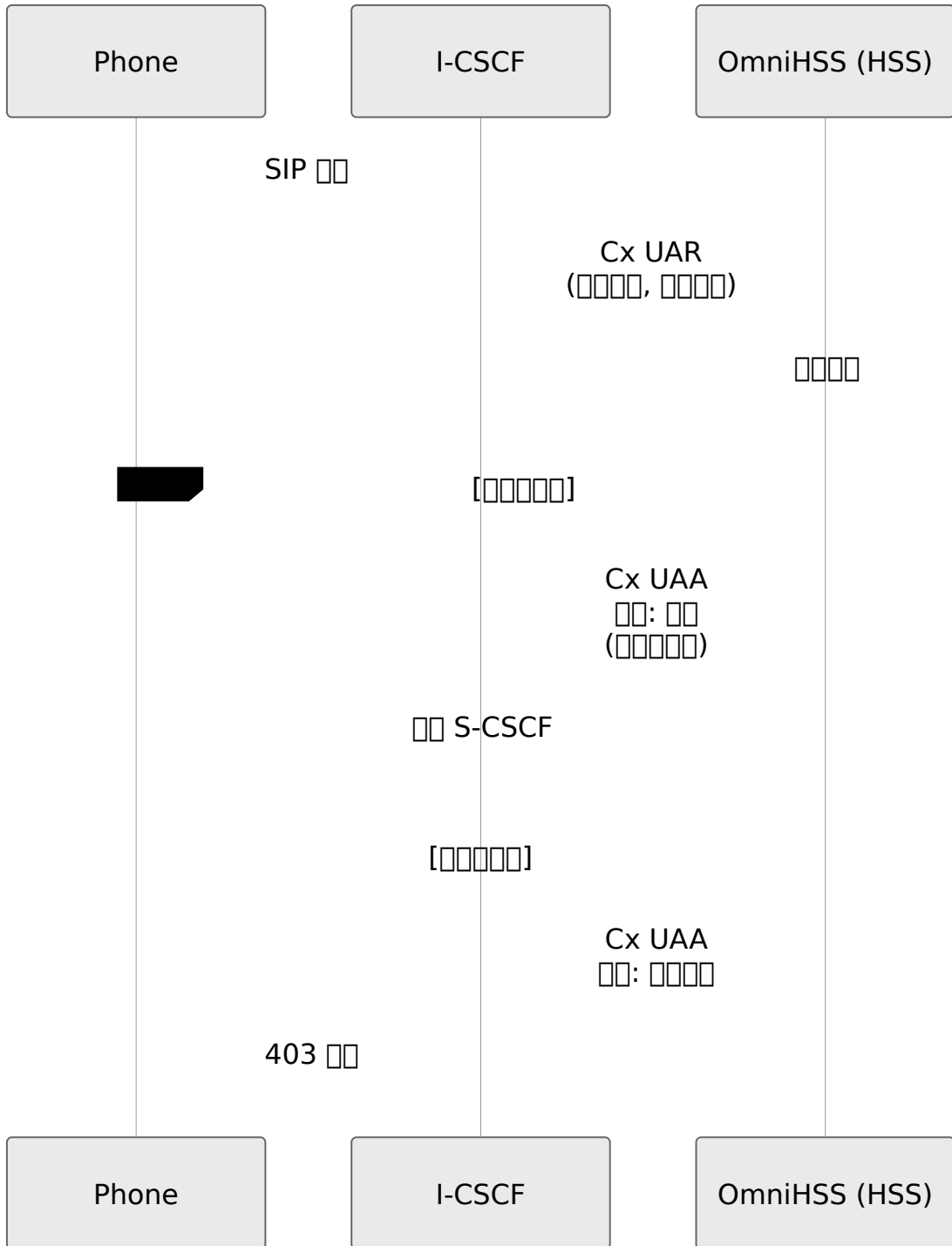
WiFi 등록



Cx (IMS)

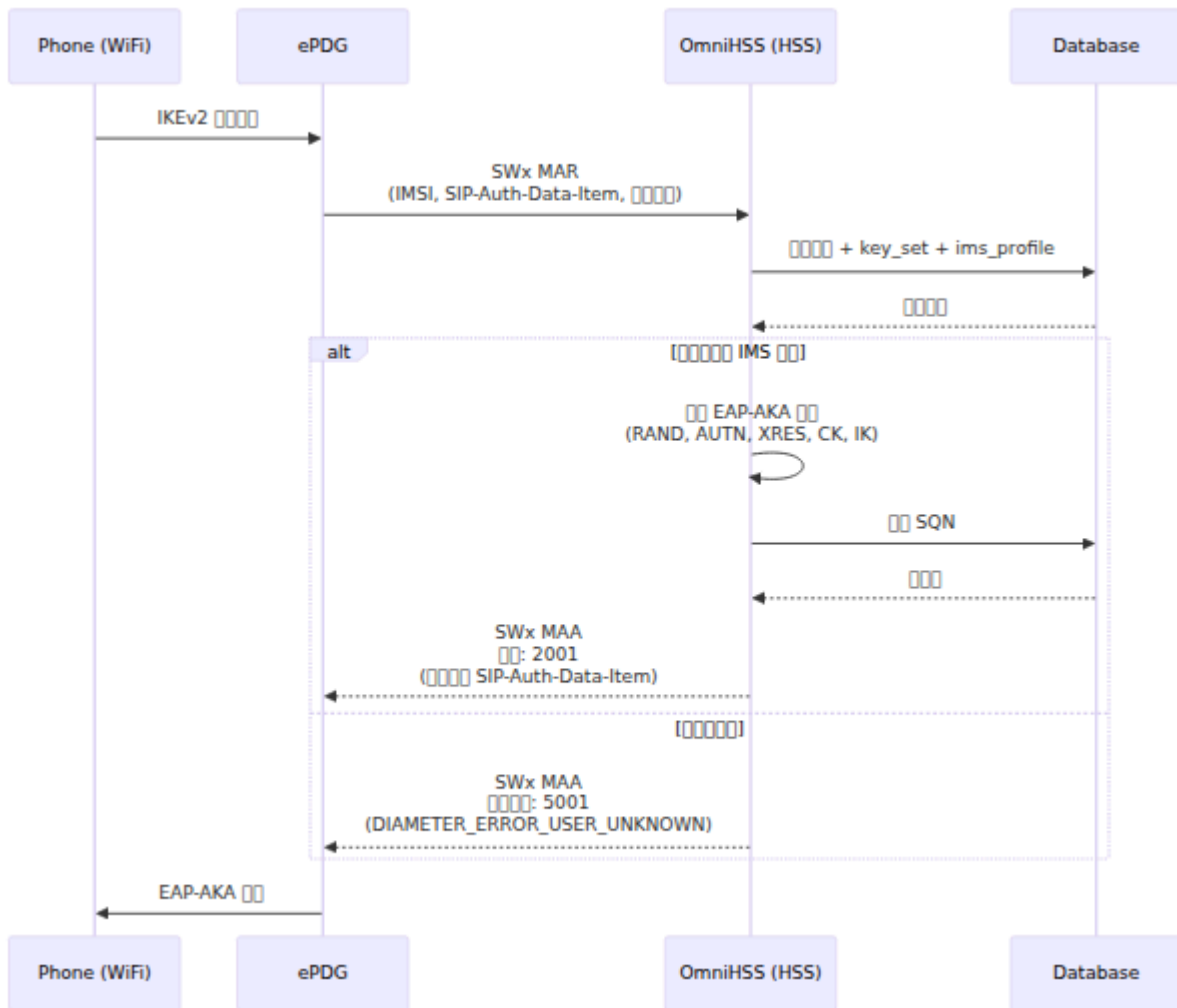
UAR/UAAs

I-CSCF



Authentication (SAR/SAA)

S-CSCF IMS Authentication

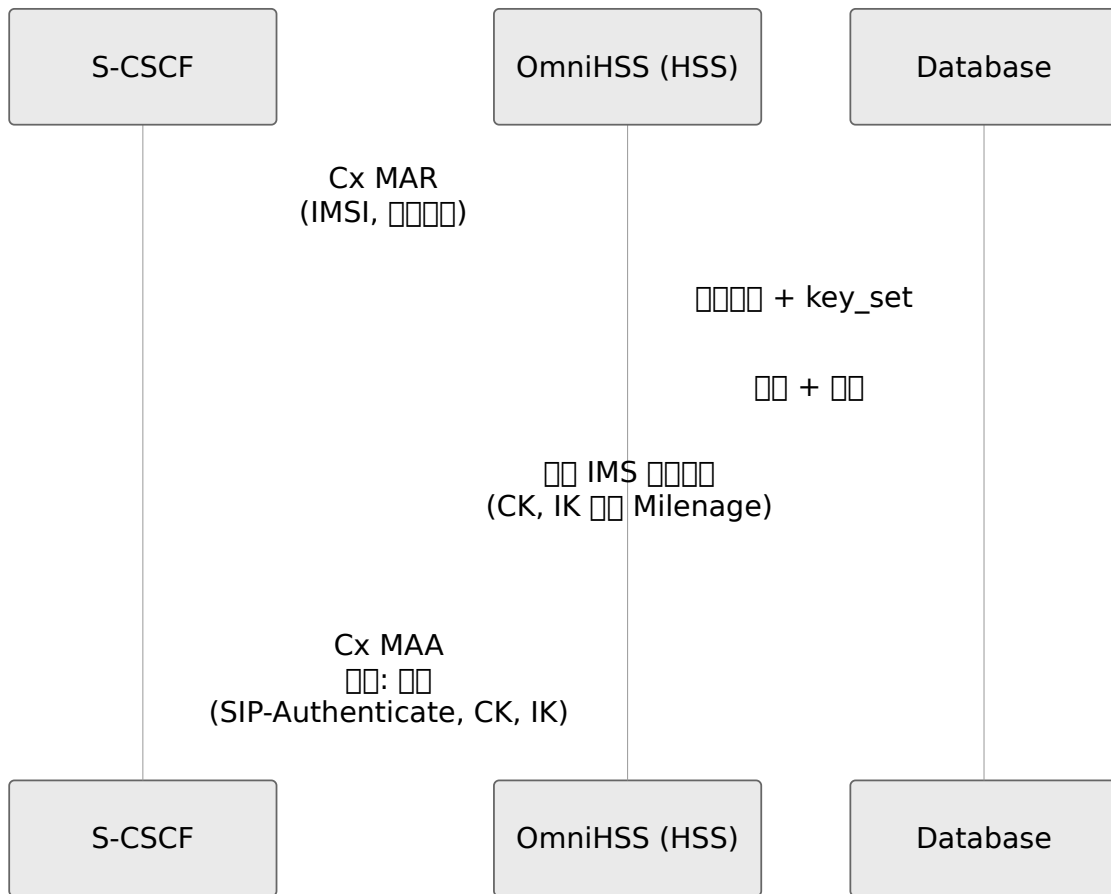


IFC parameters:

- `{{imsi}}` → IMSI
- `{{msisdns}}` → ...
- `{{mcc}}`, `{{mnc}}` → PLMN

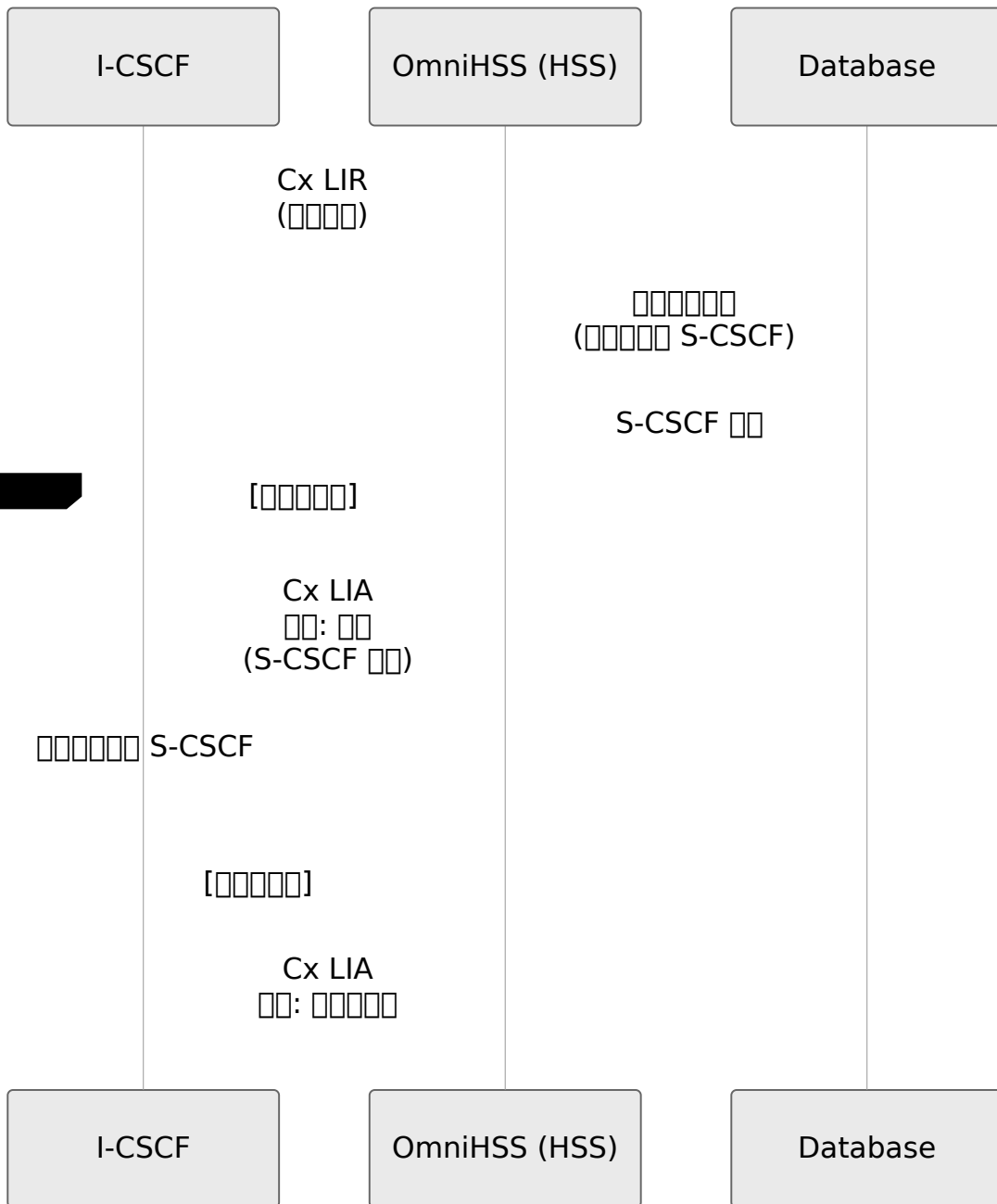
Authentication (MAR/MAA)

S-CSCF IMS Authentication



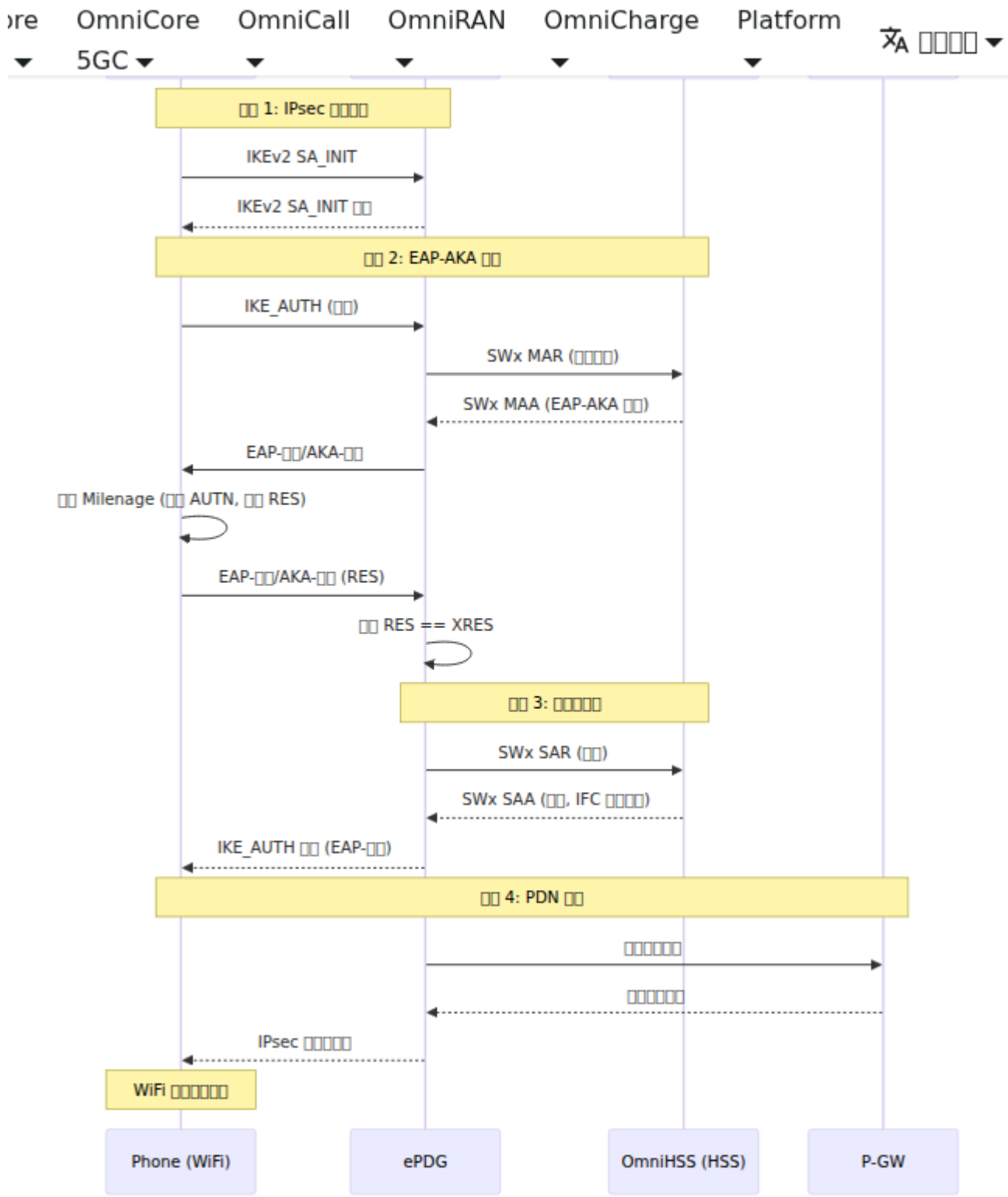
IMS (LIR/LIA)

I-CSCF sends S-CSCF IMSI



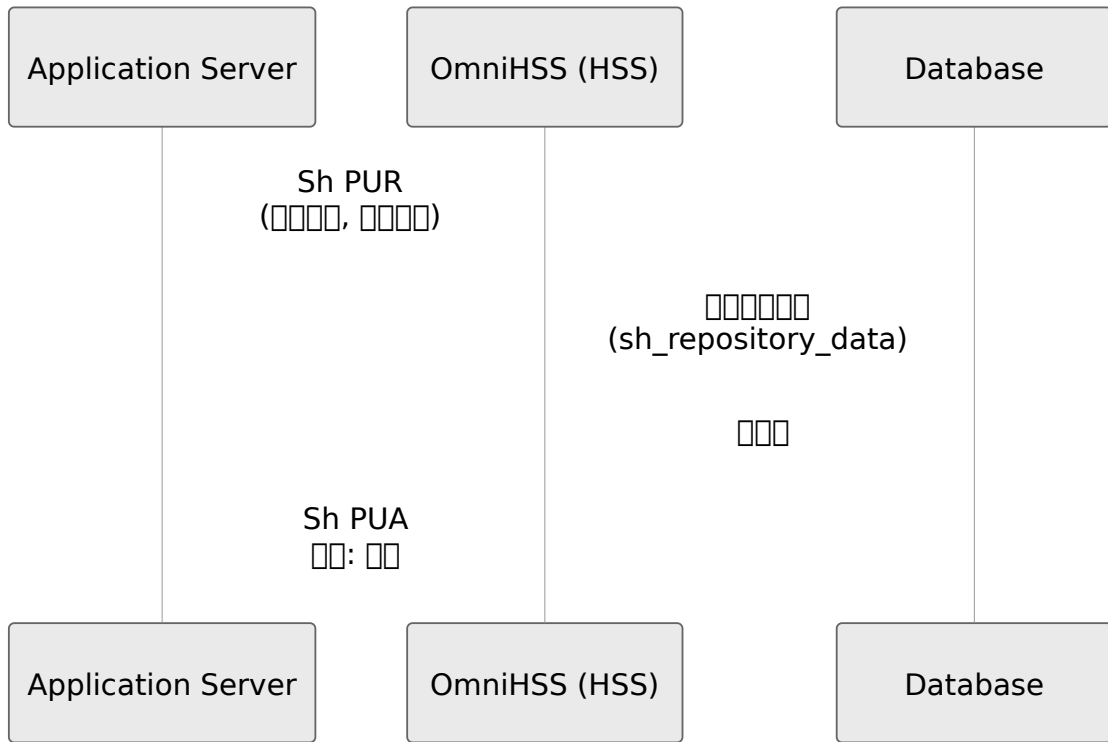
Sh (IMS)

(UDR/UDA)

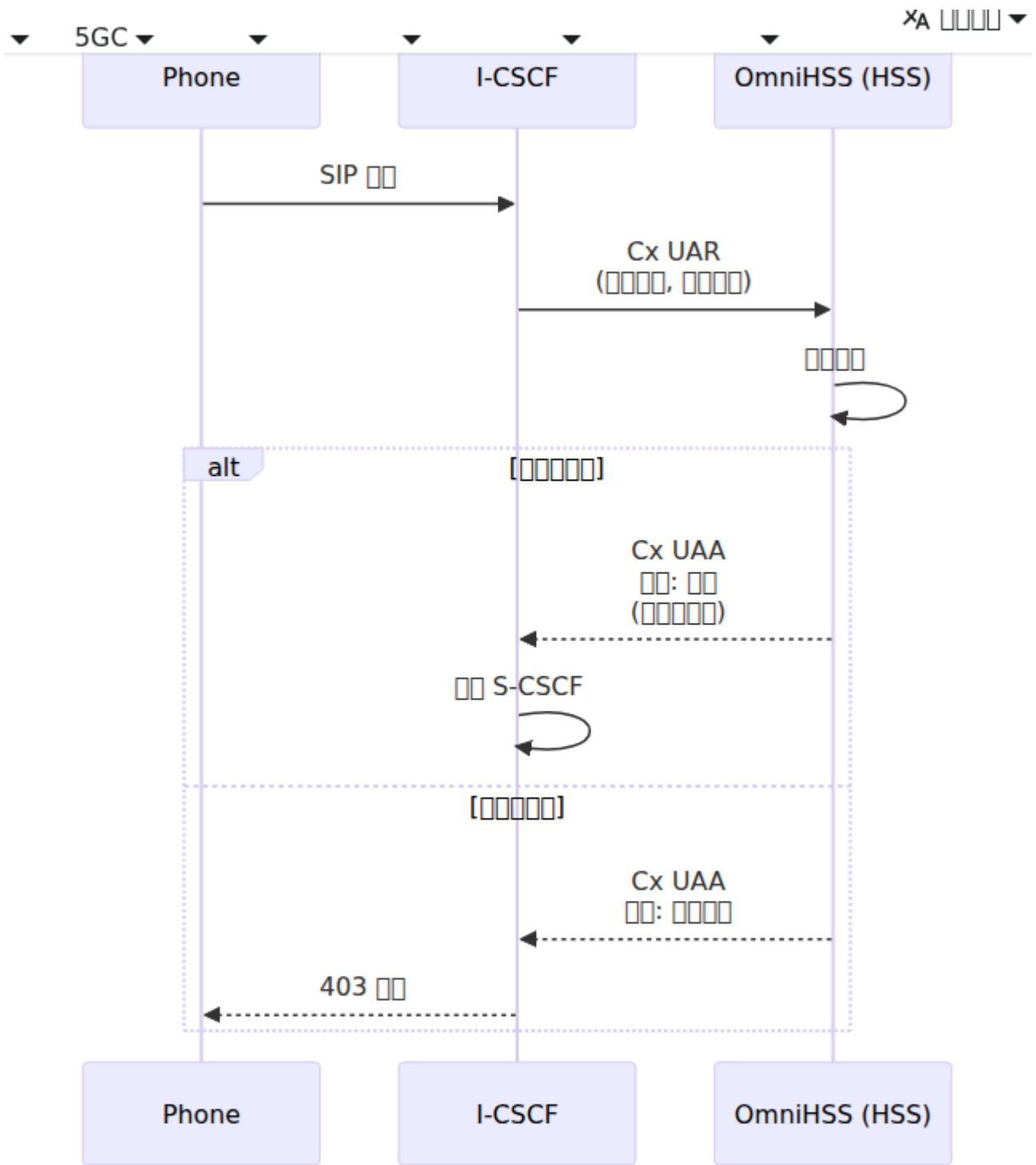


IPsec (PUR/PUA)

IPsec (PUR/PUA)



(SNR/SNA)



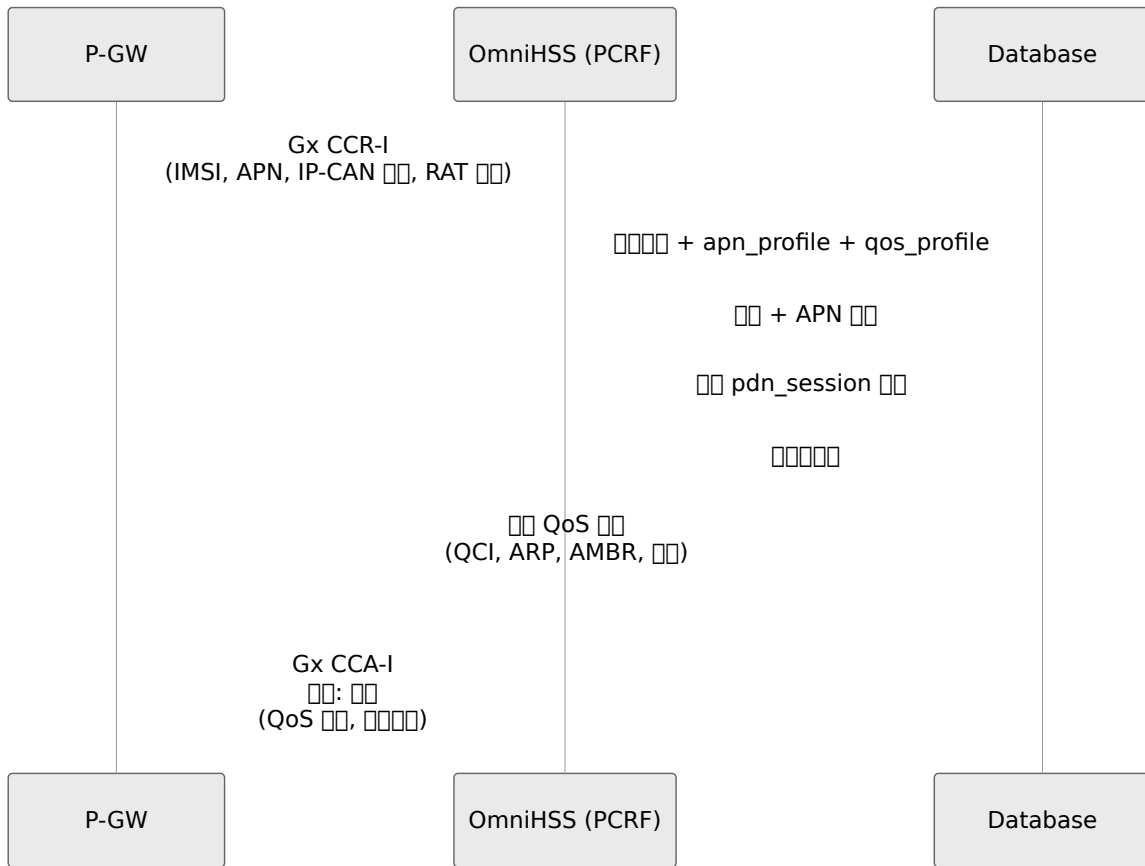
Gx [] ([])

OmniHSS [] Gx [] PCRF []

[] **PCRF** [] [] **QoS** []

CCR-I/CCA-I - P-GW

P-GW PDN

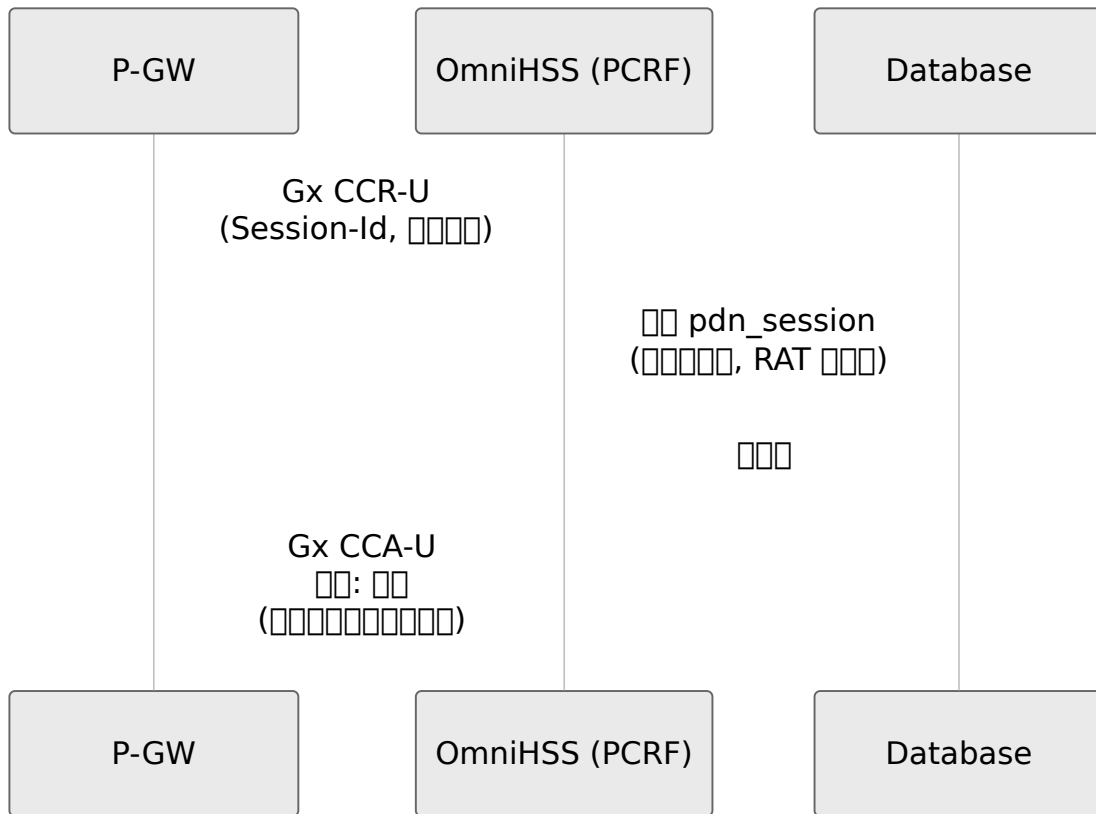


AVP:

- : ID (IMSI), ID (APN), RAT, IP-CAN
- : QoS (QCI, ARP, AMBR),

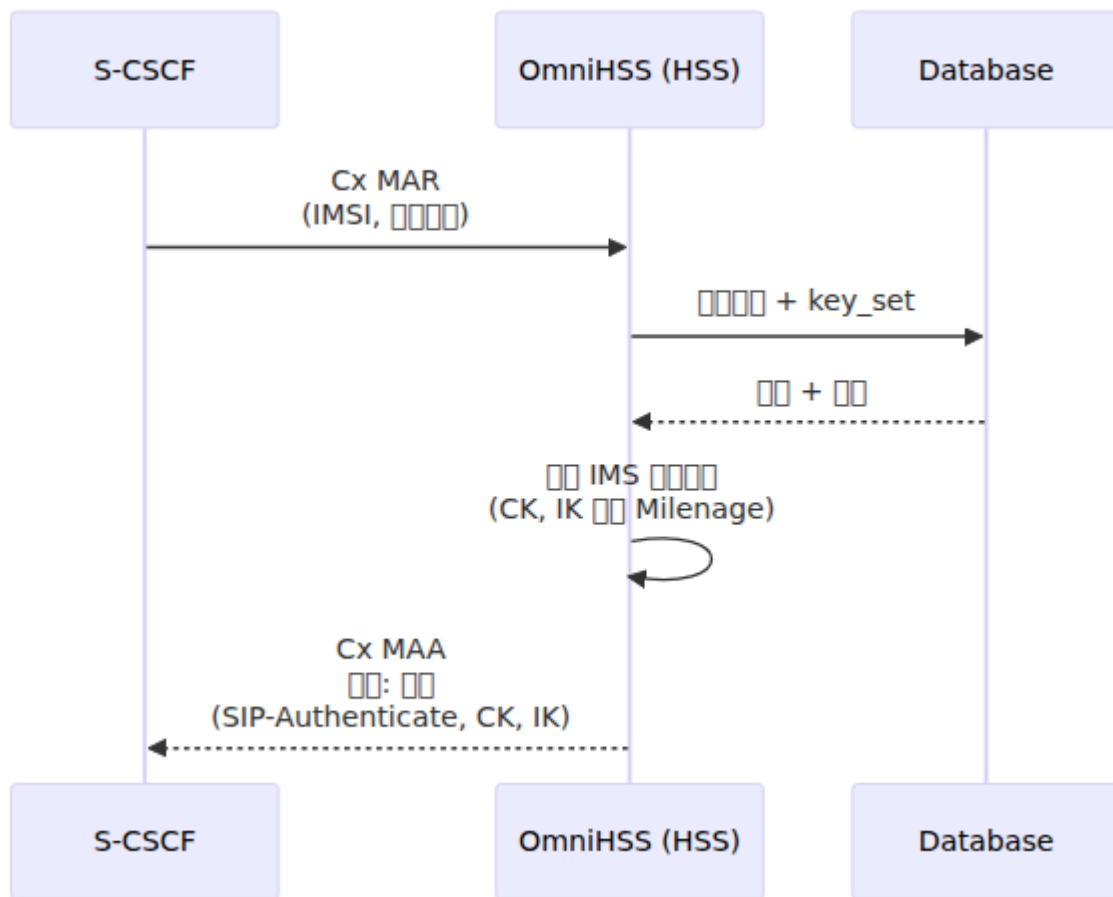
CCR-U/CCA-U - P-GW

P-GW



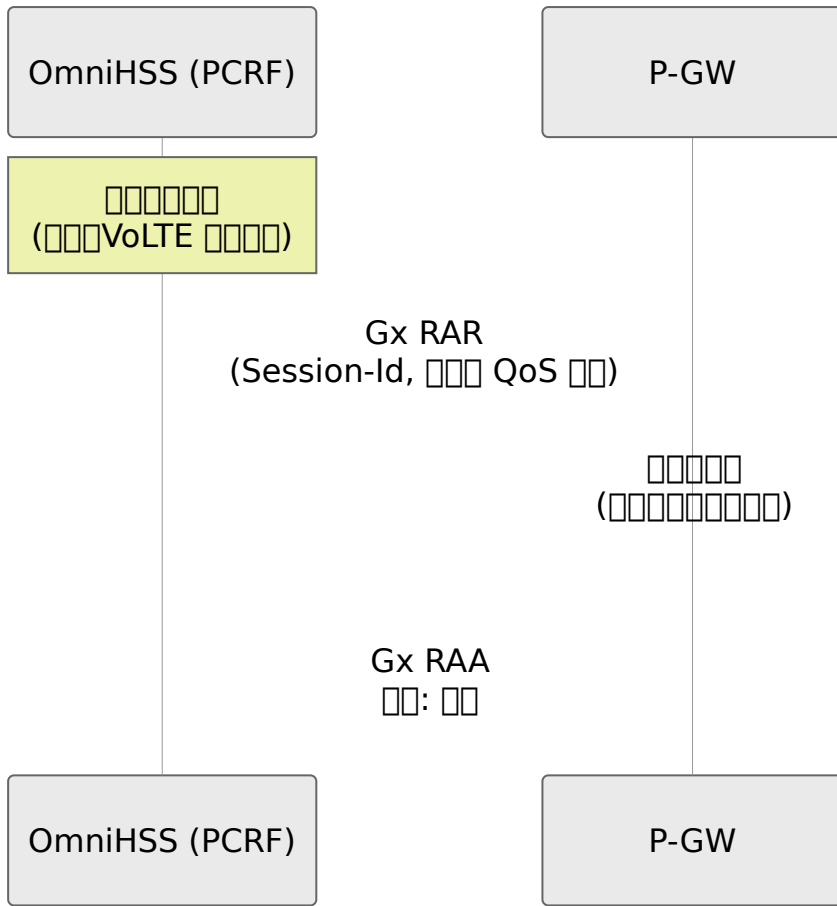
PDN Selection - P-GW (CCR-T/CCA-T)

P-GW PDN Selection



IMS Authentication (RAR/RAA)

OmniHSS (PCRF) ↔ P-GW ↔ IMS



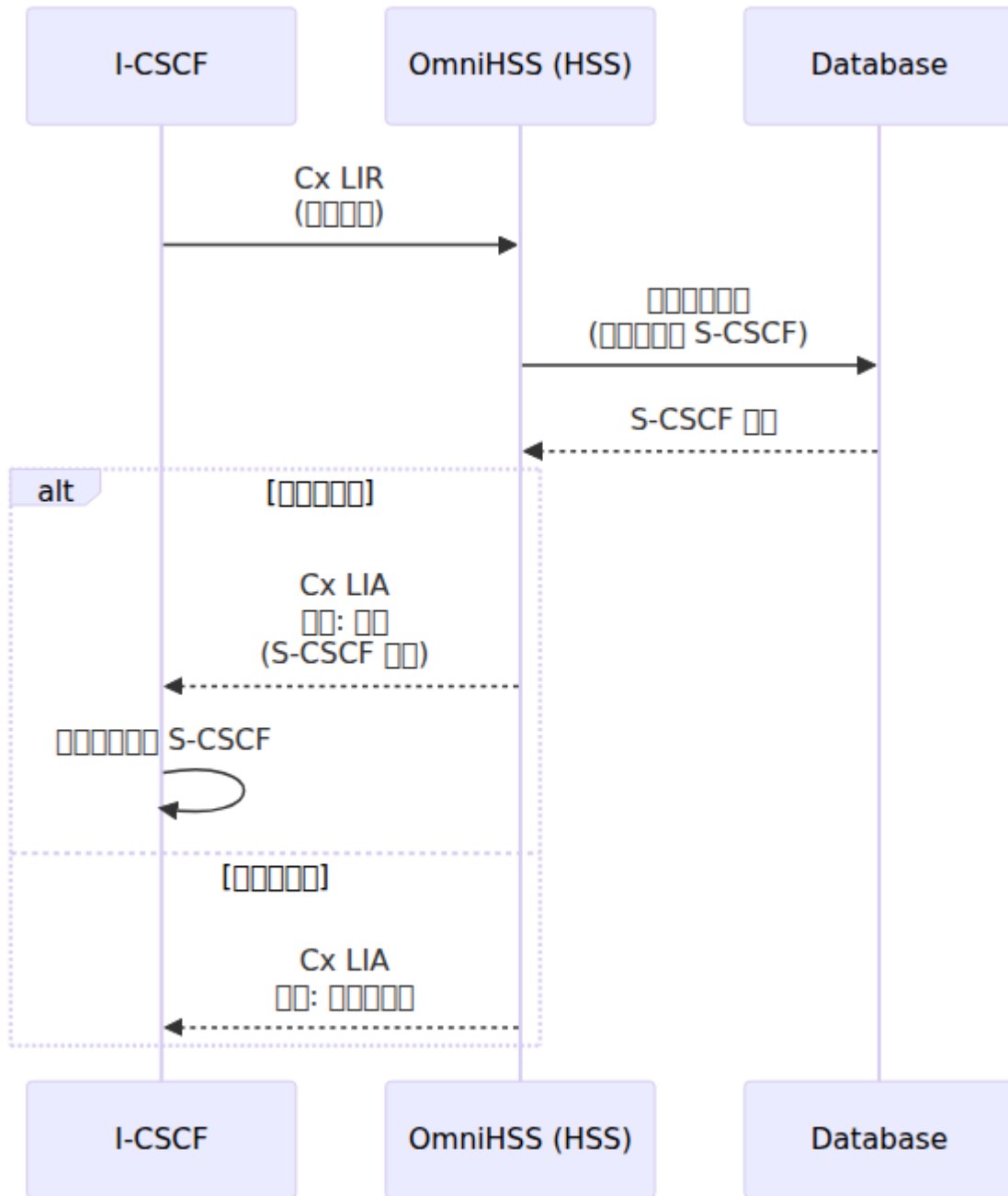
Rx (IMS)

OmniHSS Rx PCRF IMS

PCRF VoLTE

AA (AAR/AAA)

P-CSCF IMS

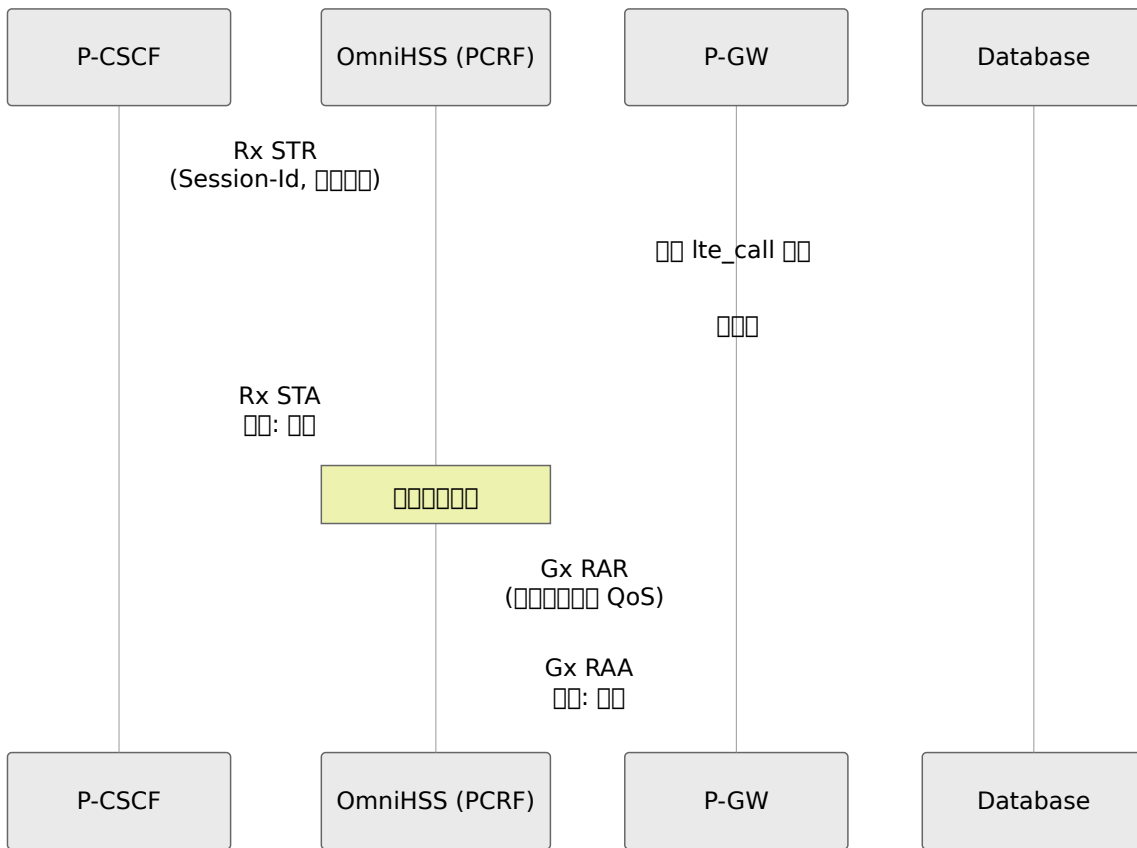


□□□□:

- □□ SDP □□□□□□□□□□
- □□□□□□ (□□/□□)
- □□□□□□ SDF □□□
- □□ Gx RAR □□□□□□

□□□□□□ (STR/STA)

P-CSCF □□ IMS □□□□□□



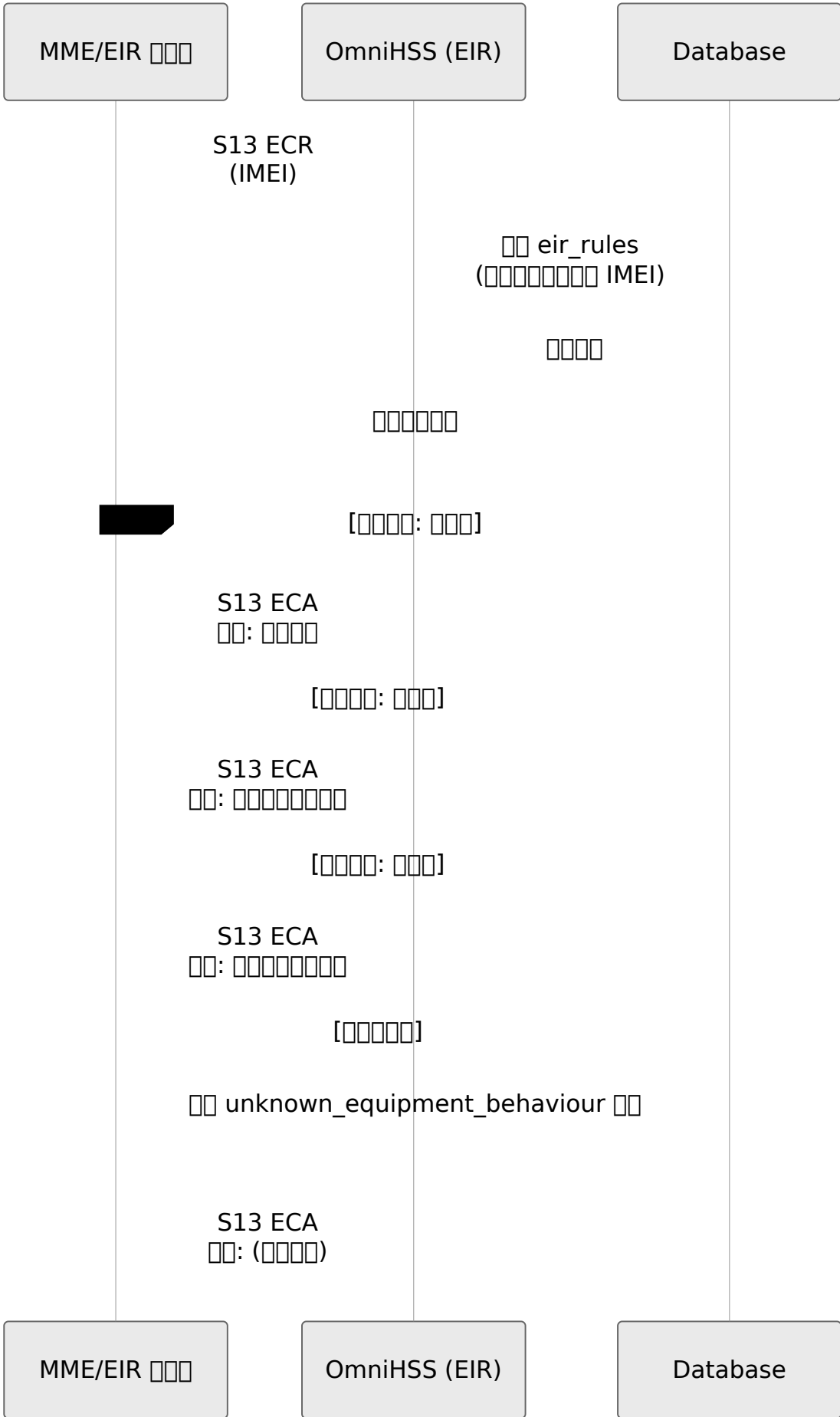
S13 (EIR)

OmniHSS S13 EIR

EIR IMEI

ME (ECR/ECA)

EIR MME

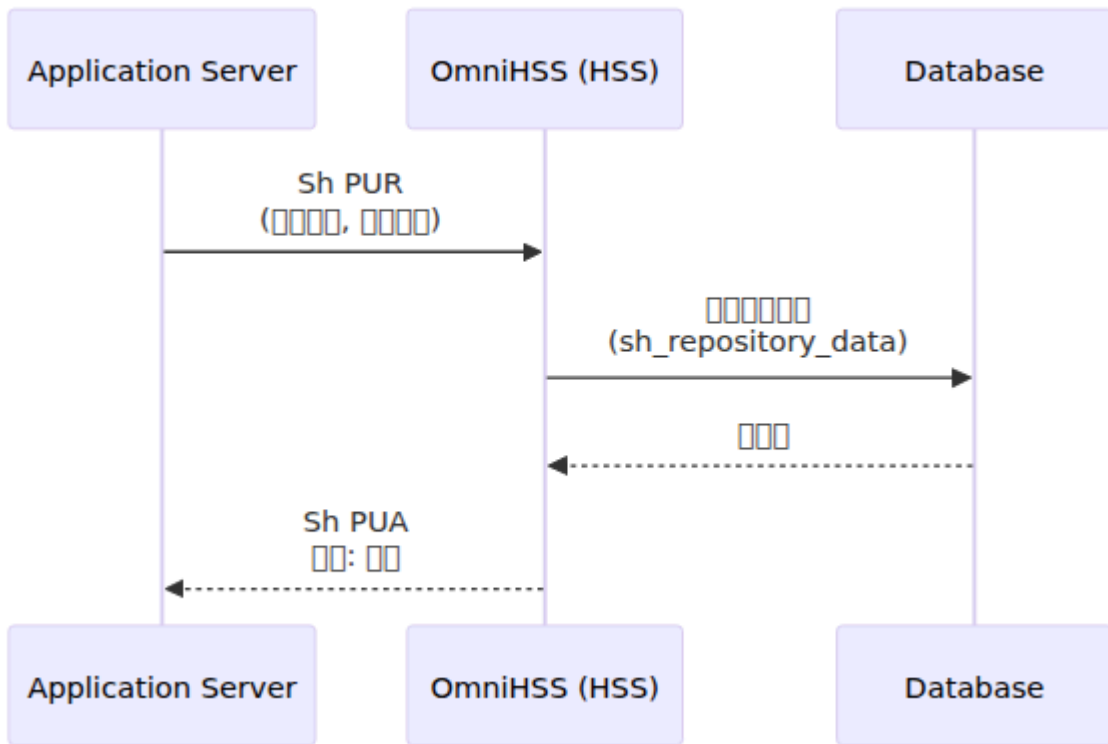


□□□□:

- □□□□ (0) - □□□□□□□□
- □□□□□□□□ (1) - □□□□□
- □□□□□□□□ (2) - □□□□□□□□

□□□□□□□: VoLTE □□

□□□ VoLTE □□□□□□□□□□□□

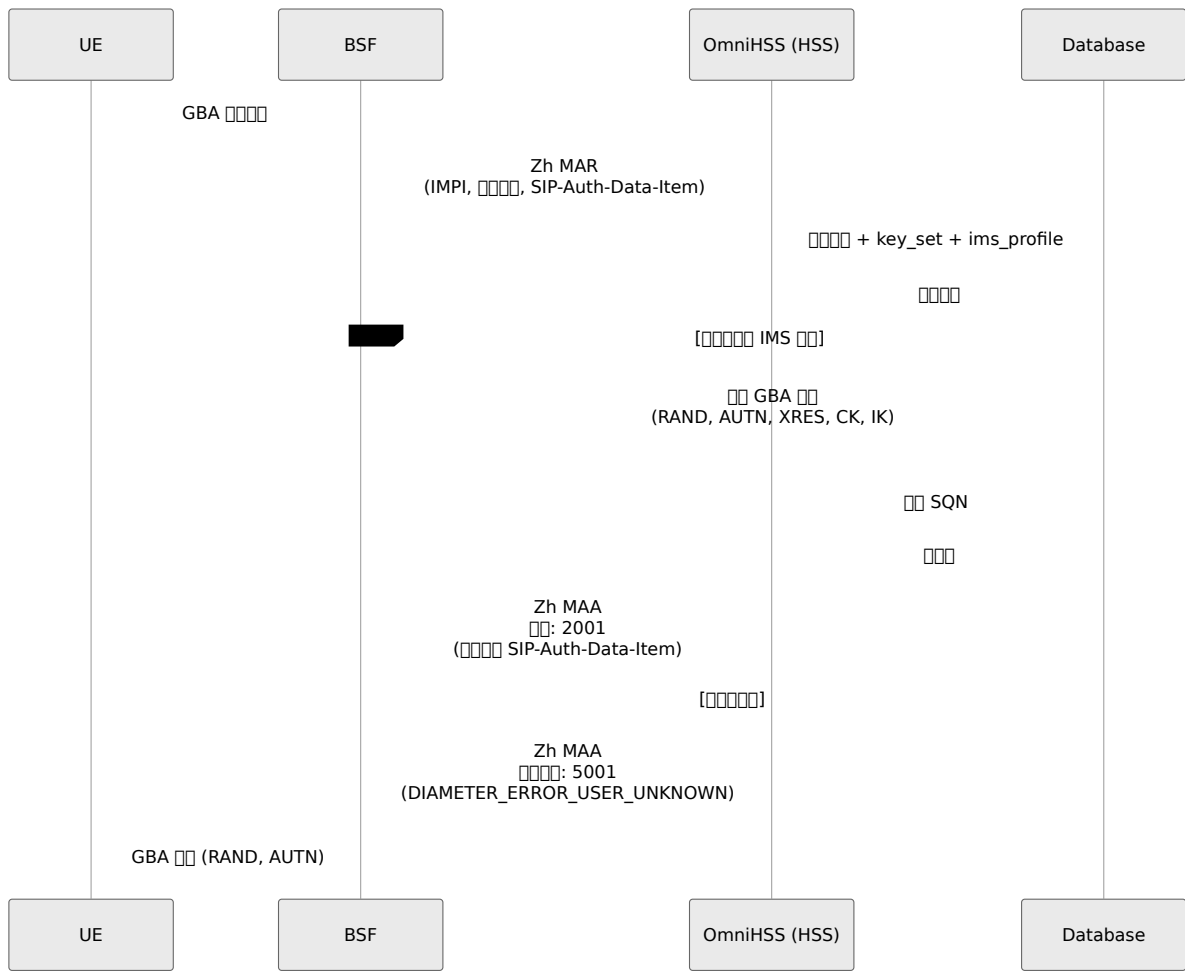


Zh □□ (□□□□□□□)

Zh □□□□□□□□□□□□ (BSF) □□□ HSS□□□□□□□□□□□□ (GBA) □□□□□□□□□□□□

□□□□□□□ (MAR/MAA)

BSF □□ GBA □□□□□□□□□□□□



AVP:

- AVP: IMSI (IMPI), IMSI, SIP-Auth-Data-Item (IMSI), IMSI
- AVP: SIP-Auth-Data-Item (SIP-Authenticate, SIP-Authorization, IMSI, IMSI), GBA-UserSecSettings

GBA AVP:

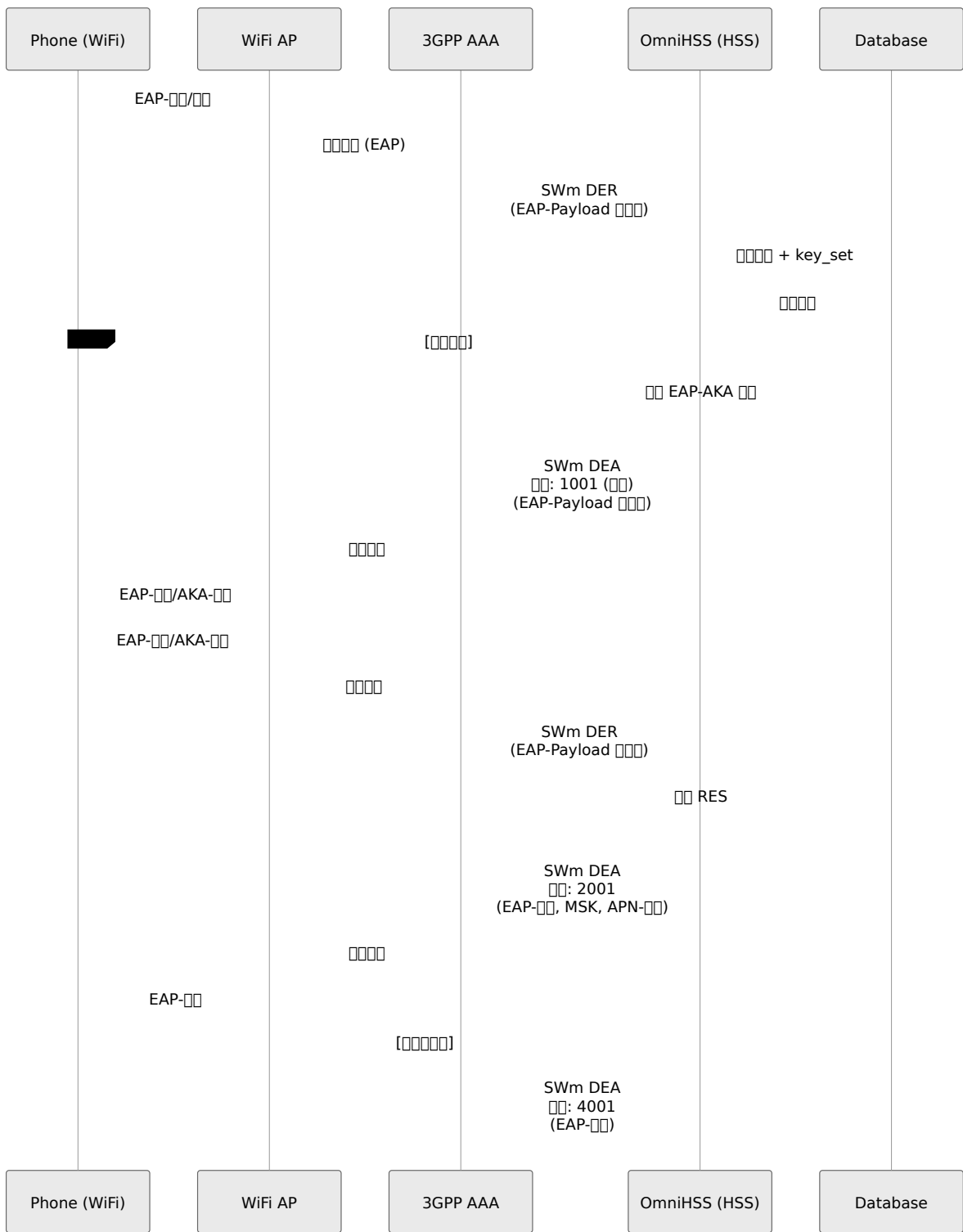
AVP	AVP
MBMS	IMSI
HTTPS	AVP GBA IMSI
NAF	IMSI
Presence	IMSI

SWm (3GPP)

SWm 3GPP AAA HSS 3GPP ePDG
WiFi EAP-AKA

Diameter-EAP- (DER/DEA)

EAP WiFi

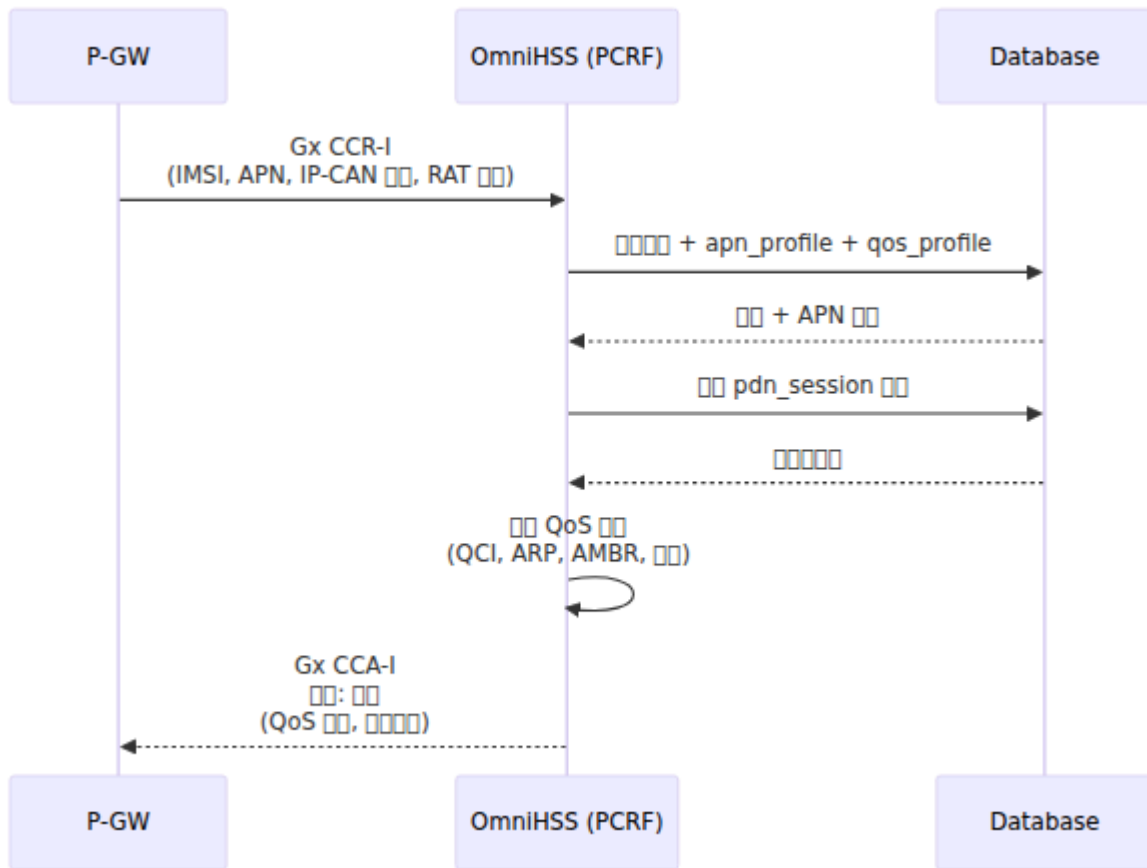


AVP:

- ID, ID, EAP-Payload, , ()
- : EAP-Payload, EAP (MSK), APN , ,

AA-AAA (AAR/AAA)

AAA EAP authentication

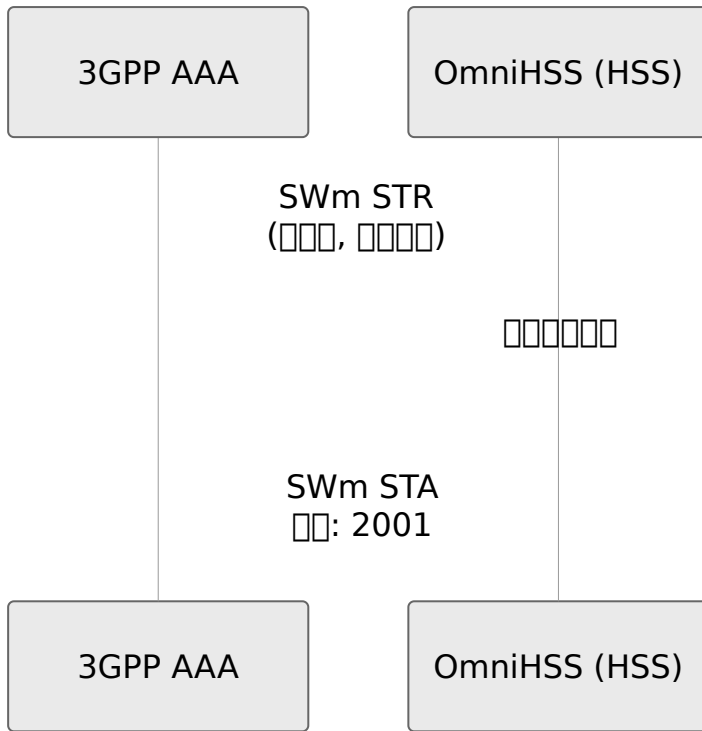


AAA AVP:

- IMSI: IMSI ID, APN ID, IMSI, IMSI, AAR IMSI
- APN: APN IMSI, 3GPP IMSI, IMSI

AAA STR/STA

WiFi authentication



0000:

00	0	00
DIAMETER_LOGOUT	1	0000
DIAMETER_SERVICE_NOT_PROVIDED	2	00000
DIAMETER_BAD_ANSWER	3	0000
DIAMETER_ADMINISTRATIVE	4	00000
DIAMETER_LINK_BROKEN	5	0000
DIAMETER_AUTH_EXPIRED	6	0000
DIAMETER_USER_MOVED	7	0000000000
DIAMETER_SESSION_TIMEOUT	8	0000

□□□□□□□□

□□□□ (S6a AIR)

□□:

- (Ki, OPC, AMF)
- SQN □□ (□□□□□□)
-

□□□□□□ (S6a ULR)

□□:

- EPC □□□□□□□□□□ APN
-
- MME □□□□□□

IMS □□□□ (Cx SAR)

□□:

- IMS □□□□□□□□□□
- IFC □□□□ XML
- S-CSCF □□□□□□
- MSISDN

PDN □□□□ (Gx CCR-I)

□□:

- APN □ EPC □□□□□□ APN □□□□□□
 - APN QoS □□□□□□□□
 - PDN □□□□□□□□□□□□
-

← □□□□□

OmniHSS

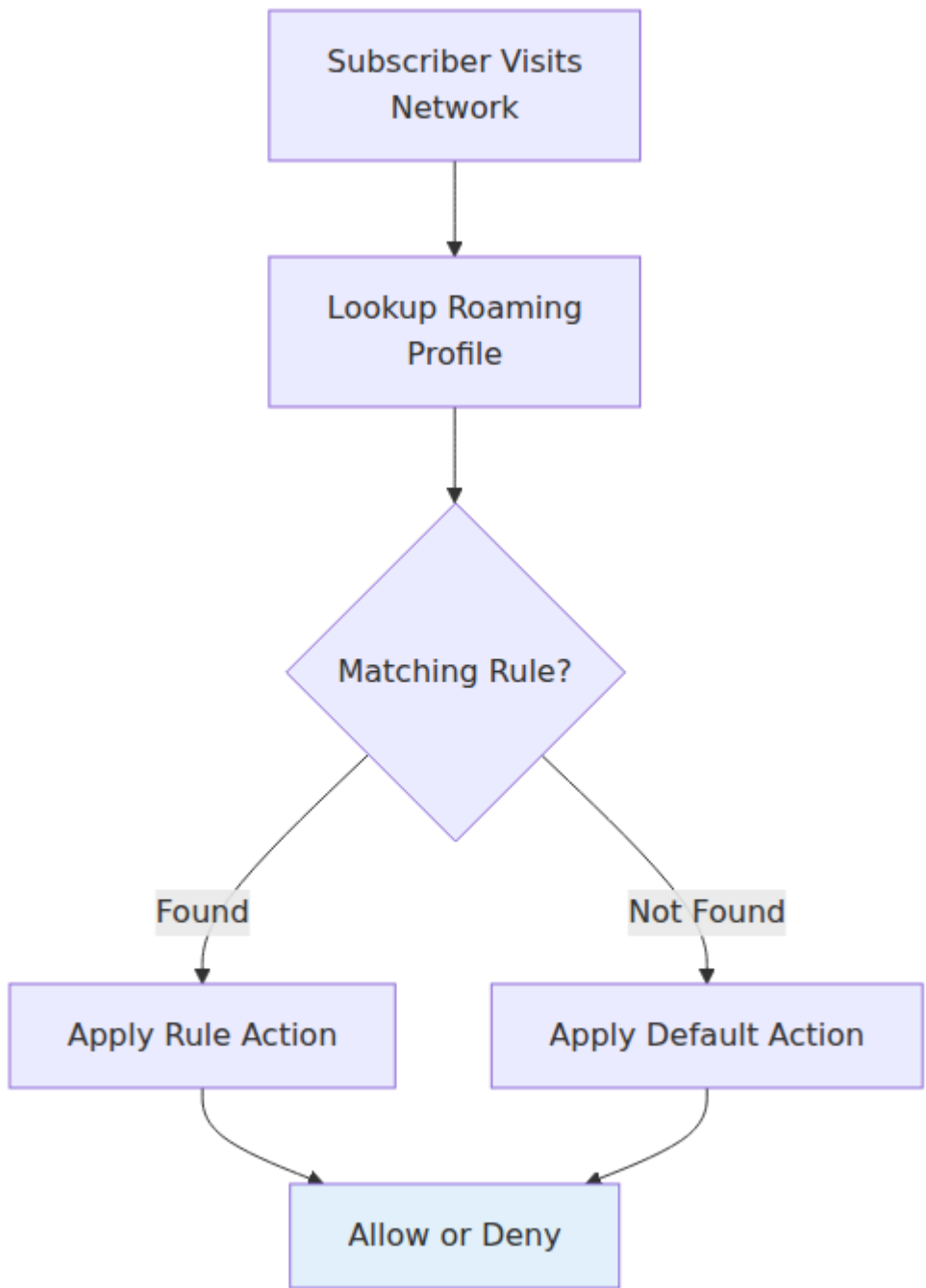


←



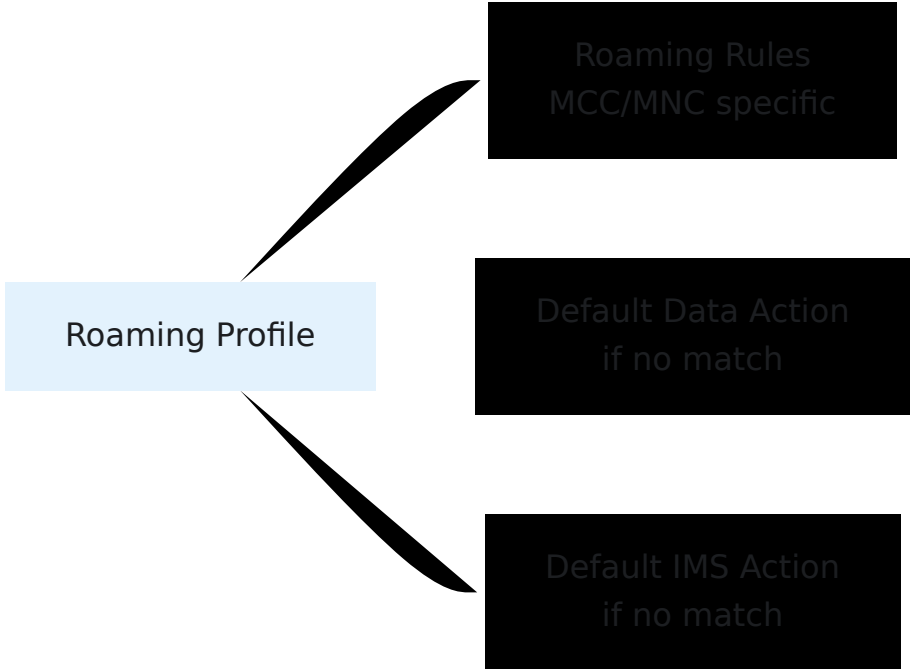
OmniHSS IMS

□□□□□□



□□□□□□

□□



□□□□

□□□□□□□□□□MCC/MNC □□□□□□□□

□□□

- name - □□□□□
- mcc - □□□□□□□3 □□□□
- mnc - □□□□□□◆◆◆2-3 □□□□
- data_action - "allow" □ "deny"
- ims_action - "allow" □ "deny"

□□□□

□□□□□□□□□□□□□□

□□□

- name - 允许
 - data_action_if_no_rules_match - "allow" 或 "deny"
 - ims_action_if_no_rules_match - "allow" 或 "deny"
-

允许

不允许

```
# 允许
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
-H "Content-Type: application/json" \
-d '{
  "roaming_profile": {
    "name": "Allow All",
    "data_action_if_no_rules_match": "allow",
    "ims_action_if_no_rules_match": "allow",
    "roaming_rules": []
  }
}'
```

不允许

```
# 不允许
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
-H "Content-Type: application/json" \
-d '{
  "roaming_profile": {
    "name": "No Roaming",
    "data_action_if_no_rules_match": "deny",
    "ims_action_if_no_rules_match": "deny",
    "roaming_rules": []
  }
}'
```

■■■■■■■■■■■■■■■■■■

■■ AT&T ■■

```
RULE1=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
-H "Content-Type: application/json" \
-d '{
  "roaming_rule": {
    "name": "Allow AT&T",
    "mcc": "310",
    "mnc": "410",
    "data_action": "allow",
    "ims_action": "allow"
  }
}' | jq -r '.response.id')
```

■■ Verizon ■■

```
RULE2=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
-H "Content-Type: application/json" \
-d '{
  "roaming_rule": {
    "name": "Allow Verizon",
    "mcc": "311",
    "mnc": "480",
    "data_action": "allow",
    "ims_action": "allow"
  }
}' | jq -r '.response.id')
```

■■■■■■■■■■■■■■■■■■■■■■

```
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
-H "Content-Type: application/json" \
-d "{
  \"roaming_profile\": {
    \"name\": \"US Carriers Only\",
    \"data_action_if_no_rules_match\": \"deny\",
    \"ims_action_if_no_rules_match\": \"deny\",
    \"roaming_rules\": [$RULE1, $RULE2]
  }
}"
```

□□□□□□□□

```
# □□□□□□□□ IMS □□□
curl -k -X POST https://hss.example.com:8443/api/roaming/rule \
-H "Content-Type: application/json" \
-d '{
  "roaming_rule": {
    "name": "Data Only - T-Mobile",
    "mcc": "310",
    "mnc": "260",
    "data_action": "allow",
    "ims_action": "deny"
  }
}'
```

□□□□□□□□□□

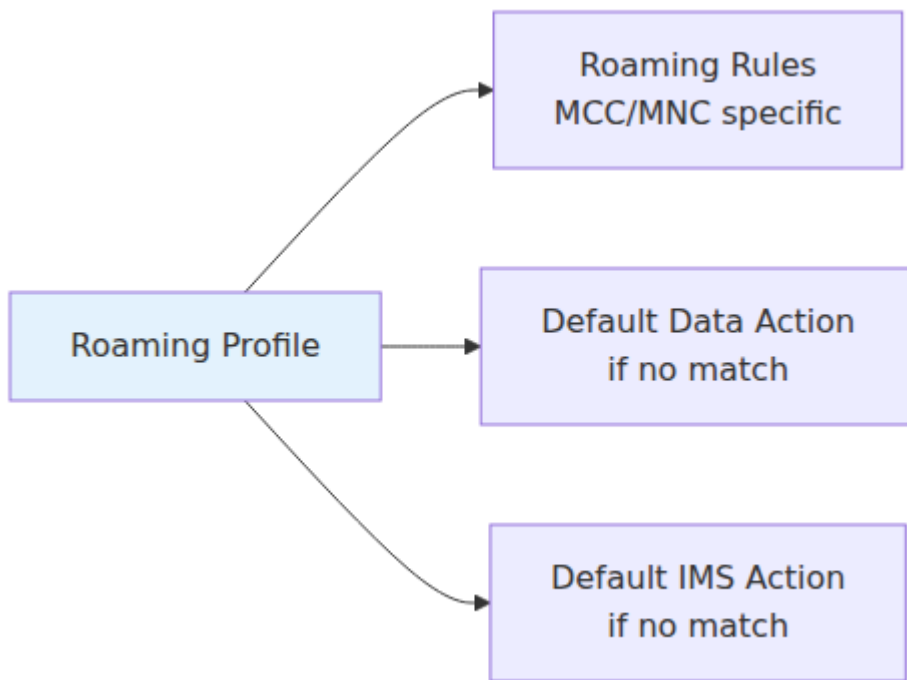
```
# □□□□□□□□□□
RULE=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
-H "Content-Type: application/json" \
-d '{
  "roaming_rule": {
    "name": "Block Expensive Network",
    "mcc": "206",
    "mnc": "01",
    "data_action": "deny",
    "ims_action": "deny"
  }
}' | jq -r '.response.id')

# □□□□□□□□□□
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
-H "Content-Type: application/json" \
-d "{
  \"roaming_profile\": {
    \"name\": \"Block Expensive Networks\",
    \"data_action_if_no_rules_match\": \"allow\",
    \"ims_action_if_no_rules_match\": \"allow\",
    \"roaming_rules\": [$RULE]
  }
}"
```

□□□□□□□

□□ **1**□□□□□□□

□□□□□□□□□□□□□□□□□□□□



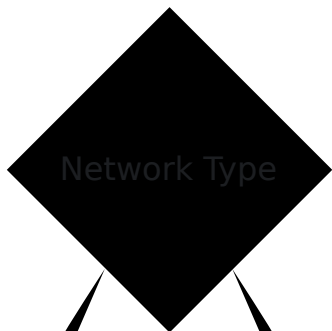
□□□

- □□□□□□□□
- □□□□□□□□ MCC □□□310□311□312□313□314□315□316□

□□ **2**□□□□□□□□□□

□□□□□□□□□□□□□□□□□□

Subscriber



Partner Network

Non-Partner

Allow

Deny

□□□

- □□□□□□□□
- □□□□□□□□□□□□□□□□ MCC/MNC□

□□ **3**□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ WiFi □□□□□□□□

Subscriber Abroad

Data Request

Voice Request

Allow Data

Deny IMS

□□□

- `data_action: "allow"` `ims_action: "deny"`

4

MME/OmniHSS

MCC/MNC

MCC

MCC		
310-316		AT&T, Verizon, T-Mobile
302		Rogers, Bell, Telus
234-235		Vodafone, O2, EE
262		Deutsche Telekom, Vodafone
208		Orange, SFR, Bouygues
222		TIM, Vodafone, Wind
214		Movistar, Vodafone

☐☐☐☐☐☐☐☐ **MCC 310-316** ☐

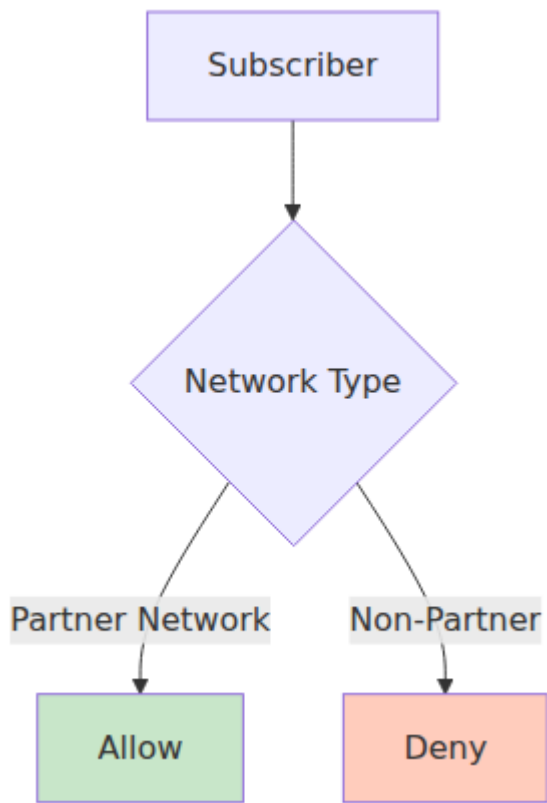
MCC	MNC	☐☐☐
310	410	AT&T
311	480	Verizon
310	260	T-Mobile
310	120	Sprint
313	380	☐☐☐☐☐☐☐☐

☐☐☐☐☐ ☐☐☐ **ITU-T E.212** ☐ **MCC/MNC** ☐☐☐

☐☐☐☐☐☐

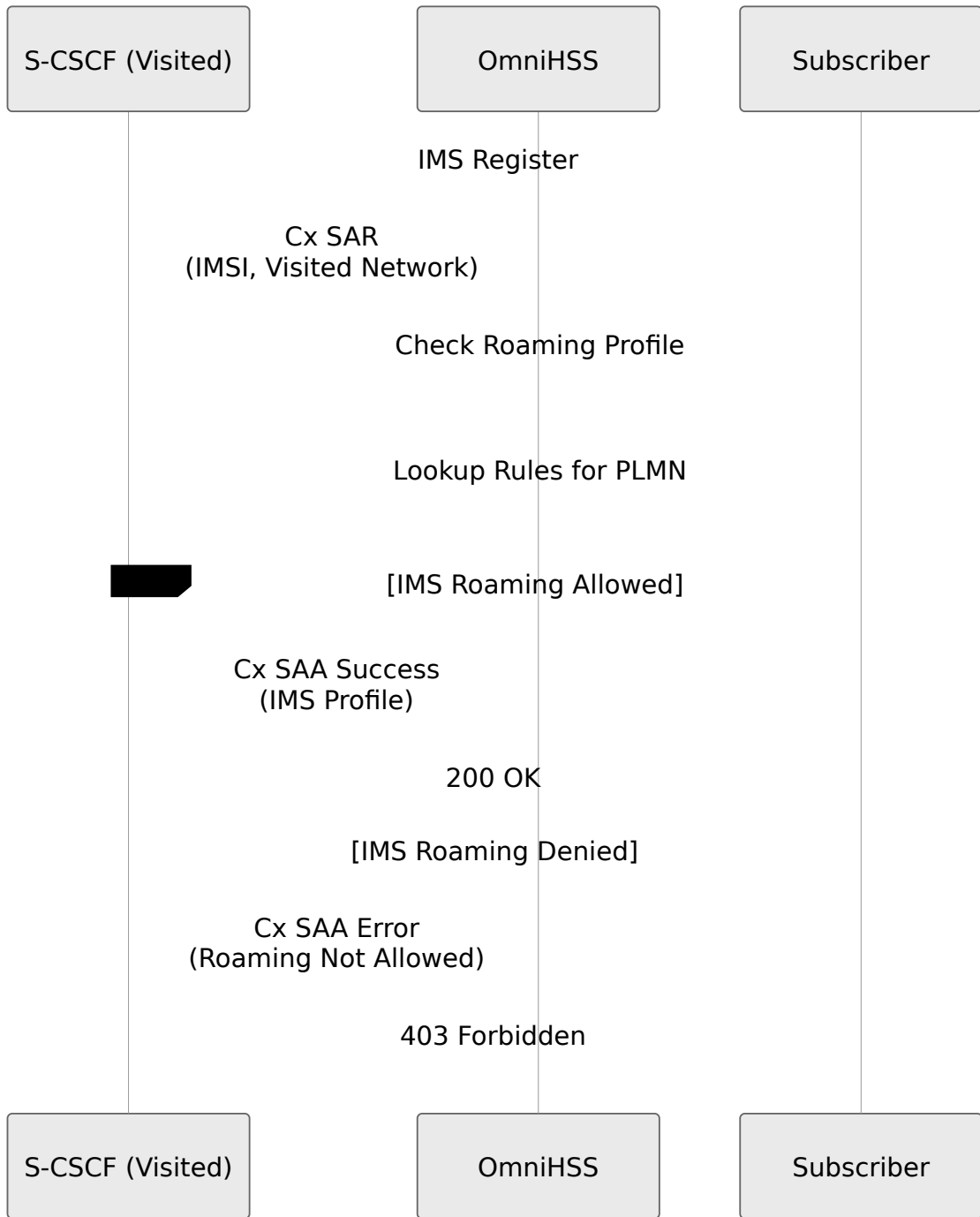
S6a ☐☐☐☐☐☐

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐



Cx IMS

IMS



□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□

- □□□□□□□□□□□□□□□□

- IMSI 001001123456789

Roaming check

- MCC/MNC 310-410
- data_action allow
- IMS action allow

IMS

IMS

- IMSI 001001123456789
- data_action allow, ims_action allow
- IMS

Roaming check

Roaming rule

```
[info] Roaming check: IMSI 001001123456789, Visited PLMN 310-410  
[info] Roaming rule matched: "Allow AT&T"  
[info] Data action: allow, IMS action: allow
```

Roaming check

Roaming rule

1. IMSI - 001001123456789
2. MCC/MNC - 310-410
3. data_action - allow
4. IMS action - allow

□□□□

1. □□□□□□ - "Allow-ATT-Data-Only" □□□ "Rule1"
2. □□ **MCC/MNC** - □□□□□□□□□□□□
3. □□□□□□ - □□□□□□□ IMS
4. □□□□□□ - □□□□□□□□□□□□

□□□□

1. □□□□ - □□□□□□/□□□□□□□□
2. □□□□ - □□□□□□□□□□□□□□
3. □□ - □□□□□□□□□□□□□□
4. □□□□ - □□□□□□□□□□

OmniHSS □□□□□□

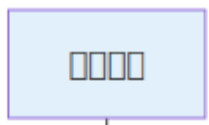
← □□□□□□

□□

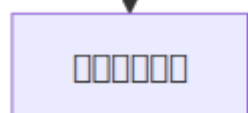
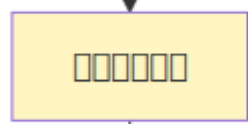
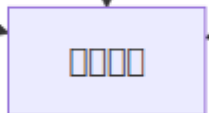
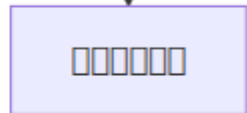
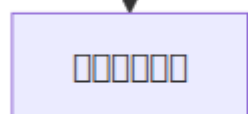
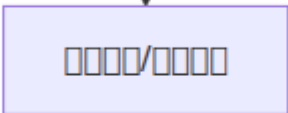
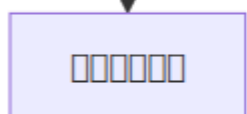
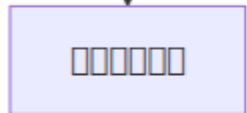
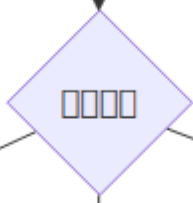
- □□□□□□
 - □□□□
 - Diameter □□□□
 - □□□□□
 - EPC □□□□
 - IMS □□□□
 - VoLTE □□□□
 - □□□□
 - EIR □□
 - □□□□
 - □□□□□□
 - API □□
 - □□□□□□□□
-

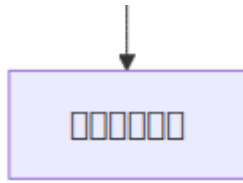
□□□□□□

□□□□□□□□



ore OmniCore OmniCall OmniRAN OmniCharge Platform 文A
▼ 5GC ▼ ▼ ▼ ▼ ▼





□□□□

□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□

- IMSI
- MSISDN□□□□□□
- □□□□□□
- □□□□□□□□

2. □□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

3. □□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□/□□□□□□□□

4. □□□□

- □□ □□□□ □□□□□□□□
 - □□ Diameter □□□□□□
 - □□□□□□□□
-

□□□□

□□

- □□□□□□□□
- “□□□□”□□
- □□□□□□

□□□□□□□□

□□ **1**□□□□□□

□□□

- □□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□

1. □□□□□□ key_set_id□

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/[IMSI]
```

2. □□□□□□□□□□□□□□□□

```
curl -k https://hss.example.com:8443/api/key_set/[KEY_SET_ID]
```

3. □ Ki □ OPC □□ SIM □□□□□□□□

□□□□□

- □□□□□ □□□ □□□□
- □□□□□□□SIM □□□□□□

□□ **2**□SQN□□□

□□□

- 查詢所有 SQN
- 查詢“SQN 範圍”
- 查詢

查詢

1. 查詢所有 SQN
2. 查詢 SQN 範圍
3. 查詢 SQN

查詢

- 查詢 AUTS 的 SQN 範圍
- 查詢所有 SQN 的 0 範圍

查詢 SQN 範圍

查詢 3

查詢

- 查詢
- 查詢

查詢

1. 查詢

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/[IMSI]
```

2. 查詢 `enabled` 查詢 `true`

查詢

- 查詢

```
curl -k -X PUT https://hss.example.com:8443/api/subscriber/[ID] \
  -H "Content-Type: application/json" \
  -d '{"subscriber": {"enabled": true}}'
```

4 EPC

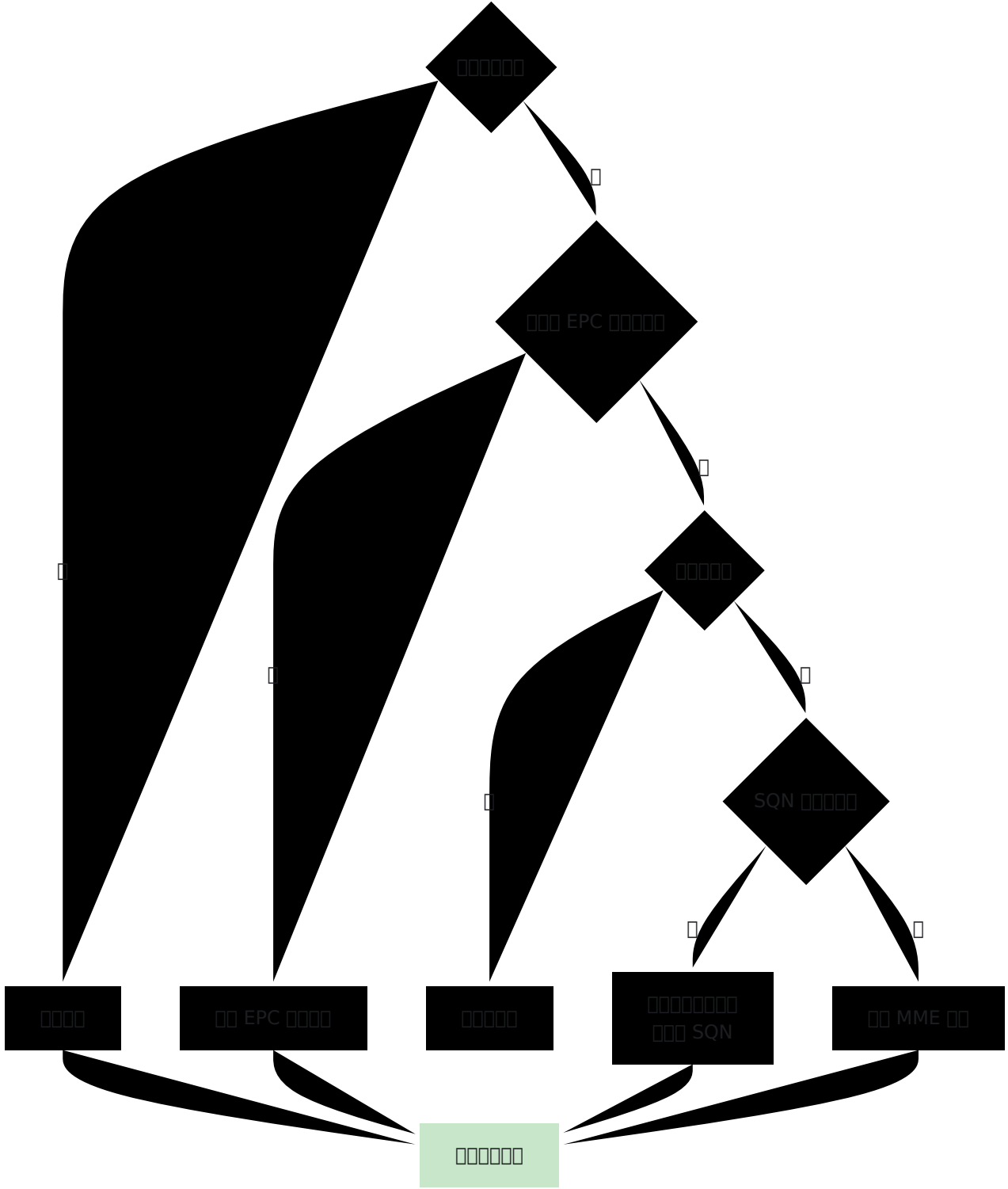
-
- “ EPC ”

1. `epc_profile_id`
2. EPC

```
curl -k
https://hss.example.com:8443/api/epc/profile/[PROFILE_ID]
```

- EPC

□□□□



Diameter 测试

测试

- 使用 telnet 测试 Diameter 服务器
- “telnet [PEER_IP] 3868”
- 测试成功

测试步骤

步骤 1: 测试 IP 连通性

命令

- telnet [PEER_IP]
- telnet [PEER_IP]
- Ping [PEER_IP]

预期结果

1. 成功连接到 OmniHSS 服务器

```
ping [PEER_IP]
```

2. 使用 telnet 测试 Diameter 服务器

```
telnet [PEER_IP] 3868
```

3. 成功连接到 Diameter 服务器 3868

测试失败

- telnet [PEER_IP] 失败
- telnet [PEER_IP] 失败
- telnet [PEER_IP] 失败

2 Diameter

-
- CER/CEA
-

1. runtime.exs Diameter
 - origin_host
 - origin_realm
 - IP
2. CER/CEA
3. OmniHSS origin_host

- Diameter runtime.exs
- OmniHSS
-

3 TLS Diameter

- TLS
-
- “”

1. priv/cert/
- 2.

```
openssl x509 -in priv/cert/diameter.crt -noout -dates
```

3. 認證

4. 認證與 TLS

認證

- 認證
- 認證
- 認證 OmniHSS

4 認證

認證

- 認證
- 認證
- “認證”

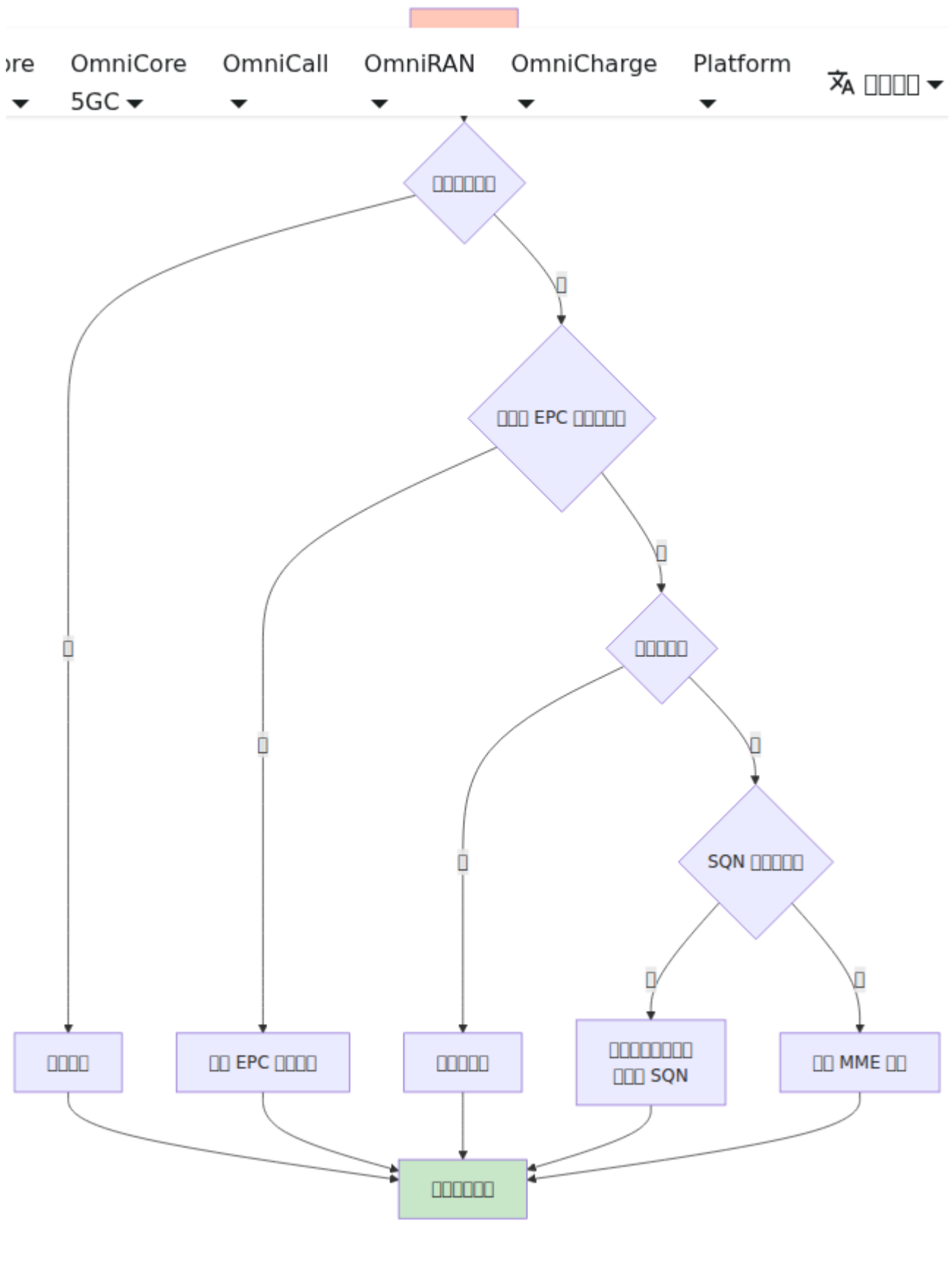
認證

1. 認證 Diameter 認證
2. 認證 S6a Cx Sh 認證
3. 認證 CER/CEA 認證

認證

- 認證 Diameter 認證
- 認證
 - MME 認證 S6a 16777251
 - S-CSCF 認證 Cx 16777216
 - P-GW 認證 Gx 16777238

Diameter 网络架构图



API 測試

API

- API 測試 500 測試
- 測試數據
- “測試數據”測試
- 測試數據

測試數據

API 1 測試數據

測試

- API 測試
- 測試數據
- “測試數據”測試

測試

1. 測試數據

```
# 測試 PostgreSQL  
psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME]  
  
# 測試 MySQL  
mysql -h [DB_HOST] -u [DB_USER] -p [DB_NAME]
```

2. 測試數據

3. 測試數據

測試

- 測試數據
- 測試數據

- 0000000000000000

00 2000000000

000

- “0000”00
- OmniHSS 00000000

00000

1. 00 runtime.exs 00000000
2. 0000000000000000
3. 0000000000

00000

- 00 runtime.exs 00 00000
- 00000000000000
- 00000000 OmniHSS

00 3000000

000

- 000 500 00
- “000000”00
- 0000000000

00000

1. 0000000000000000
2. 00 runtime.exs 00000000
3. 0000000000000000

00000

- 0 runtime.exs 00000000
- 0000000000000000

- 00000000000000000000

00 **4**0000

000

- API 00000
- 000000
- 000 CPU 0

00000

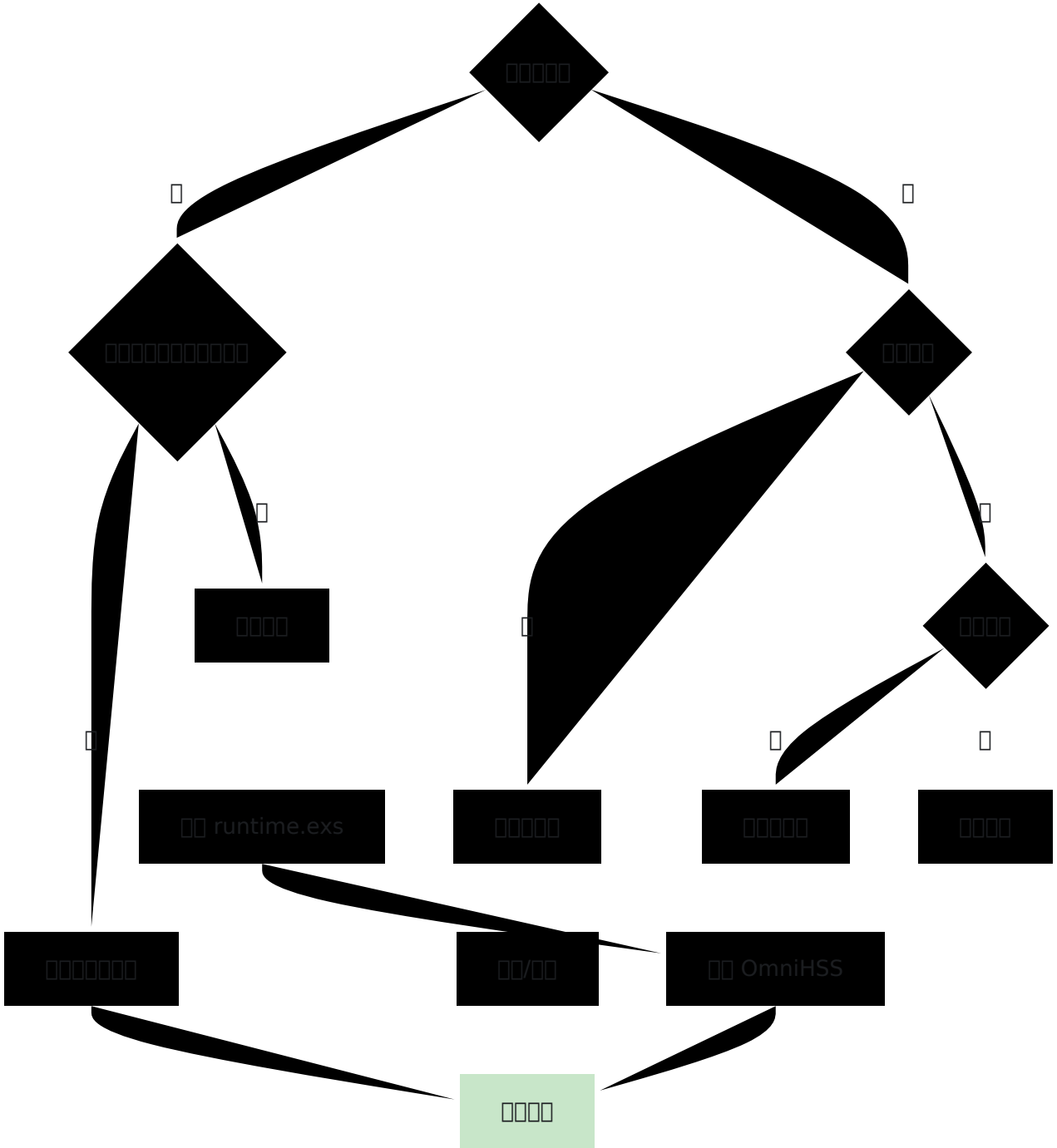
1. 000000000000
2. 000000000
3. 000000000
4. 00000000000

00000

- 00000
- 00000000
- 0000000000
- 000000000000

□□□□□□□□ ? ? ? □□

□□□□



EPC 配置

配置

- 配置 LTE 网络
- MME 配置
- 配置 PDN 网络

配置项

配置 1

配置

- 配置 MME 地址
- “配置”项
- 配置 PDN 地址

配置

1. 配置 roaming_profile_id
2. 配置 MME 地址
3. 配置 MCC/MNC
4. 配置 APN 地址

配置

- 配置 MCC/MNC 地址
- 配置 MME 地址
- 配置 APN 地址

配置 2 APN

配置

- 配置 PDN 地址
- MME 配置“APN”项

- 認證與授權

認證

1. 網路 EPC 認證與授權 APN 認證
2. 網路 APN 認證與授權
3. 網路 APN 認證

認證

- 網路 APN 認證與授權 EPC 認證
- 網路 APN 認證與授權
- 網路 APN QoS 認證與授權

網路 3 MME 認證

認證

- 認證與授權
- 網路 MME 認證
- Diameter 認證

認證

1. 網路 Diameter 認證
2. 網路 MME 認證“認證”
3. 網路 MME 認證 S6a 認證

認證

- 網路 Diameter 認證
- 網路 MME 認證
- 網路 MME 認證

網路 4 認證

認證

- 認證與授權

- □□□□□□
- □□◆◆◆□□□□□

□□□□

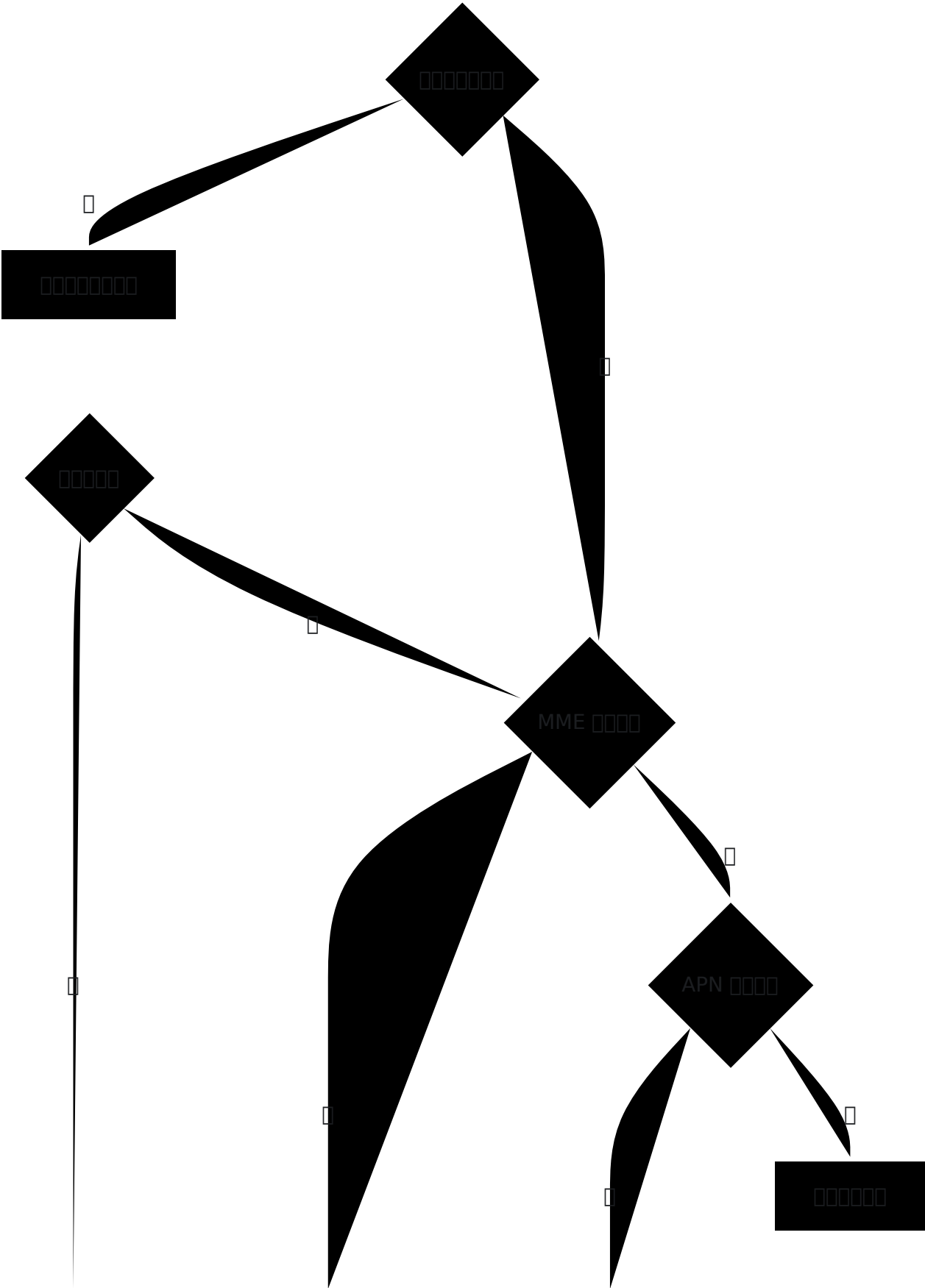
1. □□□□□□□□
2. □□□□□□□□ MME □□
3. □□□□□□□□

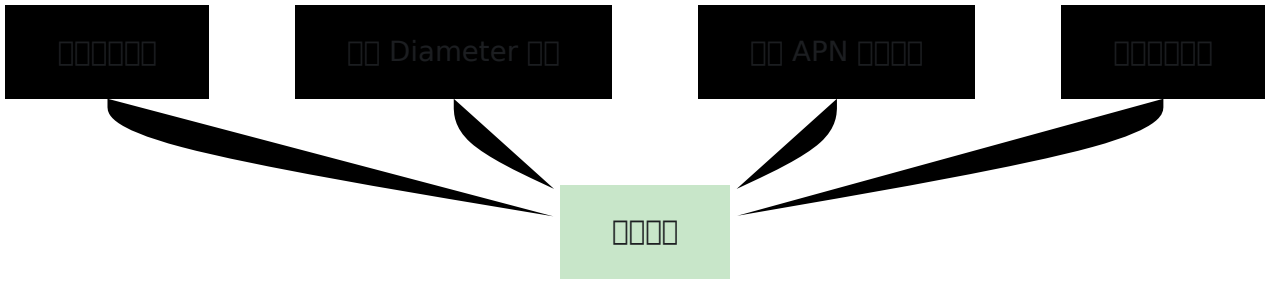
□□□□

- □□□□□□□□□□
- □□□□□□□□ MME
- □□□□□□□□

EPC □□□□□□□□

EPC □□□□





IMS

IMS

- IMS VoLTE
- IMS “IMS Profile”
- IMS

IMS

1 IMS

IMS

- IMS
- IMS

IMS

1. IMS `ims_enabled`
2. IMS `ims_profile_id`

IMS

- IMS
- IMS

2 S-CSCF

IMS

- IMS 網路
- IMS 網路 Diameter 網路

網路

1. IMS 網路 Diameter 網路
2. IMS S-CSCF 網路
3. IMS S-CSCF 網路 Cx 網路

網路

- IMS Diameter 網路 S-CSCF
- IMS S-CSCF 網路

IMS 3 網路 IFC 網路

網路

- IMS 網路
- IMS IFC 網路

網路

1. IMS 網路
2. IMS IFC 網路
3. IMS IFC XML 網路

網路

- IMS IFC 網路 IMS 網路
- IMS 網路 IMS IFC 網路

IMS 4 IMS 網路

網路

- IMS 網路
- IMS 網路
- IMS 網路 IMS 網路

□□□□

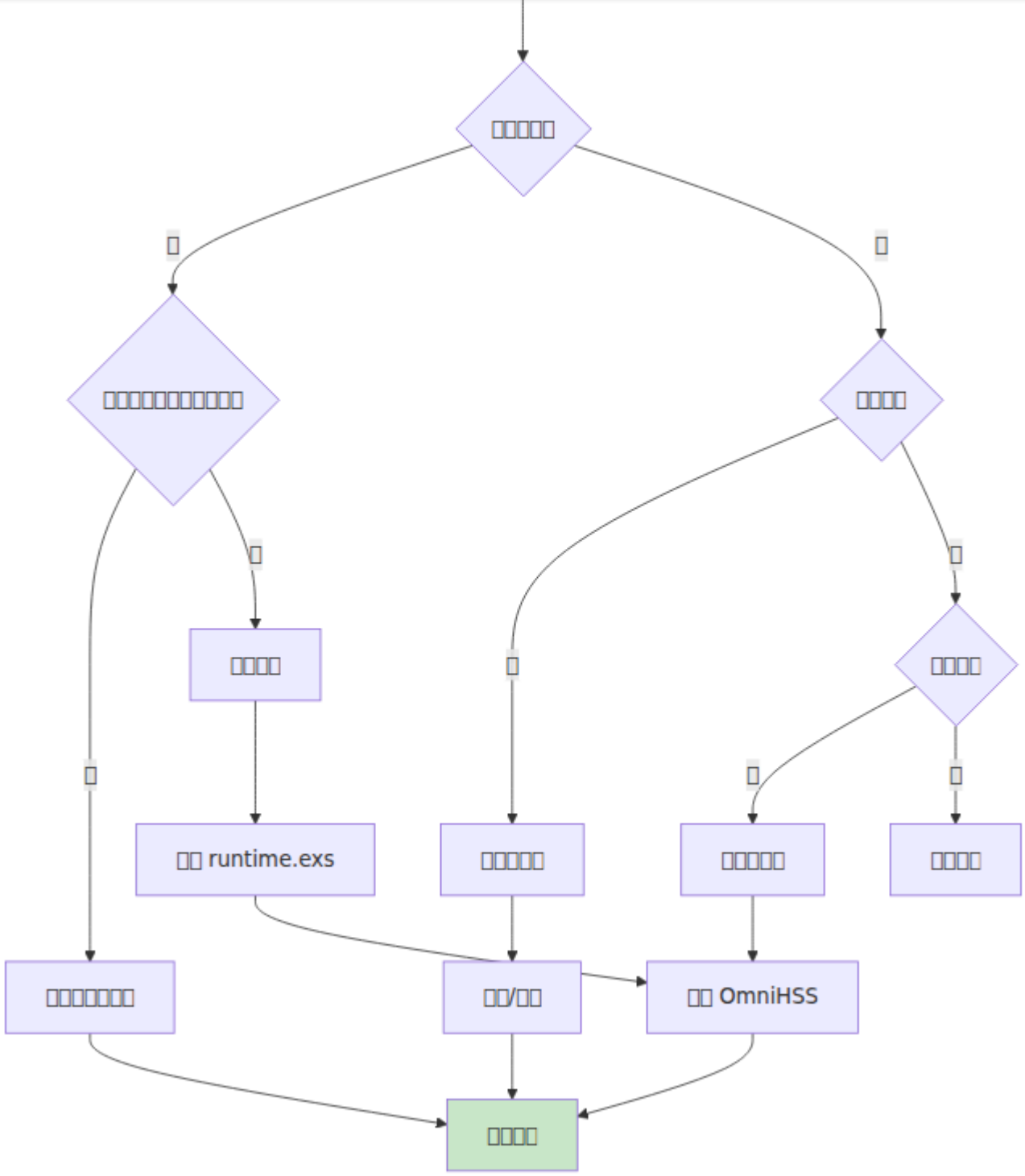
1. □□□□□□□□ IMS □□
2. □□□□□□□□□□□□□□ `ims_action`

□□□□

- □□ □□□□ □□□ IMS
- □□□□□□□□□□□□ IMS □□

IMS 部署架构图

Core 5GC OmniCore OmniCall OmniRAN OmniCharge Platform



VoLTE 网络

网络

- IMS 网络
- 网络
- 网络
- 网络“网络”

网络

1 P-CSCF 网络

网络

- 网络
- 网络

网络

1. 网络 Diameter 网络
2. 网络 P-CSCF 网络
3. 网络 P-CSCF 网络 Rx 网络 OmniHSS PCRF 网络

网络

- 网络 Diameter 网络 P-CSCF
- 网络 P-CSCF 网络 OmniHSS 网络 Rx

2 网络

网络

- 网络
- AAR/AAA 网络
- Rx 网络

网络

1. 設定 Rx Diameter
2. AAR-AA-設定
3. AAA-AA-設定

設定

- P-CSCF 設定 AAR 設定
- OmniHSS Rx 設定
- IMS 設定

3 QoS/設定

設定

- 設定
- 設定
- 設定

設定

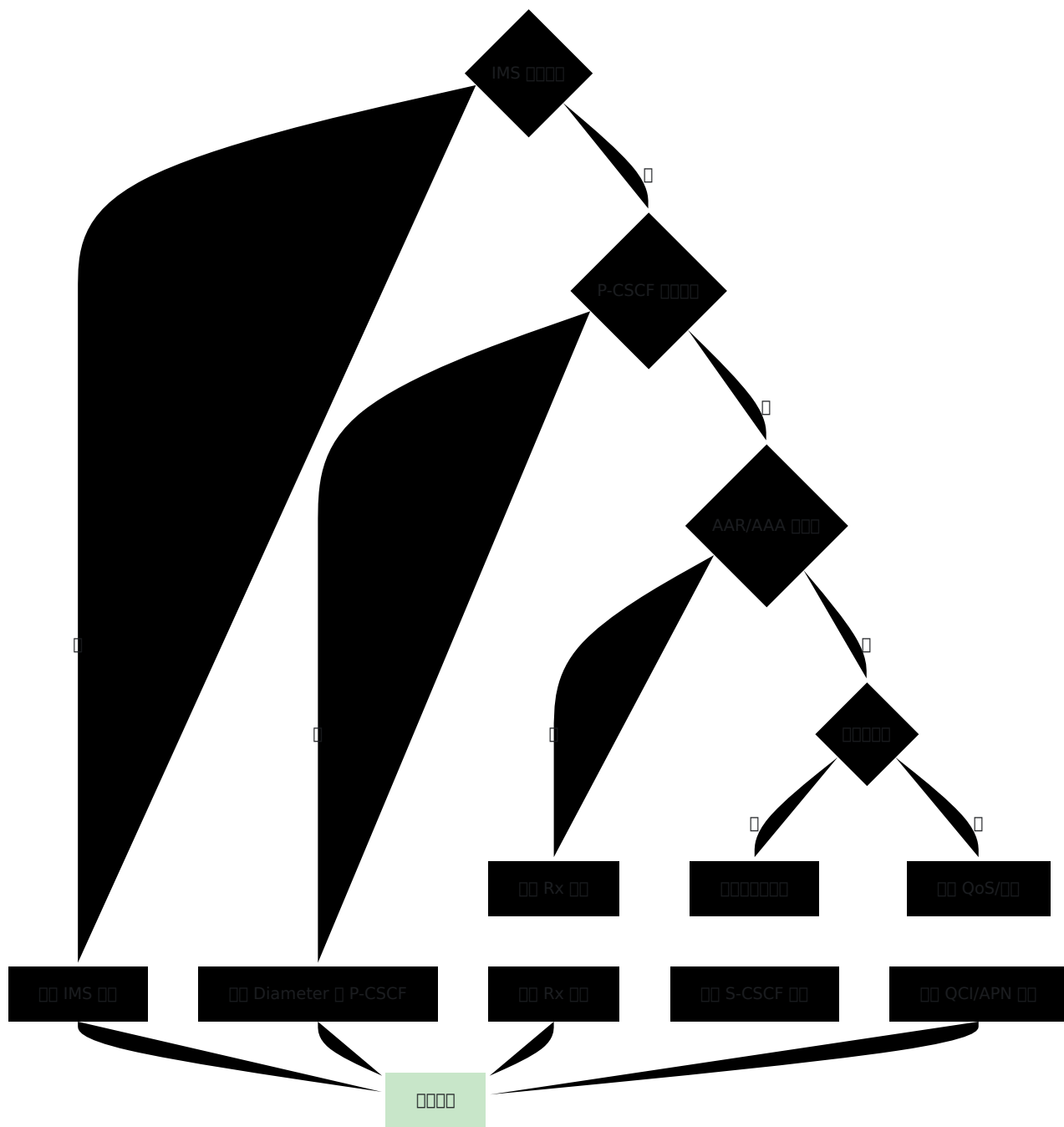
1. APN APN QoS 設定
2. QCI 設定 QCI 1
3. P-GW 設定 Gx-PCRF 設定

設定

- APN QoS 設定 IMS APN
- QCI 1
- Diameter 設定 P-GW

VoLTE 网络架构

VoLTE 网络



□□□□

□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□□

□□ 1□□□□□□□□□□

□□□

- □□□□□□
- □□□□□□□□

□□□□□

1. □□□□□ `roaming_profile_id`
2. □□□□□□□□

□□□□□

- □ □□□□□□□ □□□□□

□□ 2□□□□□□□

□□□

- □□□□□□□□□□□□
- □□□□□□□□

□□□□□

1. □□□□□□ MME □□□□□□ MCC/MNC
2. □□□□□□□□□□□□
3. □□□ MCC/MNC □□□□□□□□

4. 設定する

設定

- [設定](#) 設定

```
curl -k -X POST https://hss.example.com:8443/api/roaming/rule \  
-H "Content-Type: application/json" \  
-d '{  
  "roaming_rule": {  
    "name": "設定",  
    "mcc": "310",  
    "mnc": "410",  
    "data_action": "allow",  
    "ims_action": "allow"  
  }  
'
```

設定 IMS 設定

設定

- 設定
- IMS 設定
- 設定

設定

1. 設定
2. `data_action` `ims_action`
3. 設定

設定

- 設定 IMS
 - `ims_action: "allow"`
- 設定 `ims_action_if_no_rules_match` `"allow"`

設定 [設定](#) 設定

EIR

-
-
- EIR

1 IMEI

-
-

1. EIR
- 2.
3. IMEI
- 4.

- EIR
-
-

2 MME S13

- EIR
-

##

1. MME S13
2. MME Diameter
3. S13
4. MME

- MME S13 EIR
- Diameter S13 16777252
- MME

3

-

1. EIR
- 2.
- 3.

- .* IMEI
-
-

- API
- Diameter
- CPU

- 00000000

0000000000

00 **1**00000000

000

- 000000
- 000 CPU 0
- 0000

00000

1. 0000000000000000
2. 00000
3. 00000000
4. 000000

00000

- 00000
- 00000000
- 00000000
- 00000000
- 000 00000

00 **2**00000000

000

- 00000000
- 0000000000
- 00000000

00000

1. 00000000

2. 000000
3. 0000000000
4. 0000000000

000000

- 00000000
- 000000000000
- 000000000000
- 000000000000

00 3 Diameter 000000

000

- Diameter 00000
- 0000000000
- 000000000000

000000

1. 00 0000 Diameter 00
2. 00000000
3. 000000000000
4. 000000000000

000000

- 000000000000
- 000000000000
- 000000000000
- 00000000 Diameter 00

00 4 000000

000

- OmniHSS 000000

- 1000000
- 10000000

100000

1. 10000000 OmniHSS 10000000
2. 10000000
3. 100000000000
4. 10 Erlang VM 10

100000

- 10 OmniHSS 10000000
 - 1000000000000000000
 - 1 runtime.exs 10 Erlang VM 10000
 - 100000000000000
-

10000000

10

- 1000000000000000000
- 10000000
- 10000000
- 10000000

10000000000

10 **10MME** 10/10

1000

- 10000000 MME 10000000
- 1000 MME 10000000
- 10000

□□□□

1. □□□□□□□□ MME
2. □□ MME □□□□
3. □□ MME □□□□□□

□□□□

- □□□□□□□□□□□□
- □□□□□□□□
- MME □□□□□□□□□□

□□ **2**□□□□□□□□

□□□

- □□□□□□□□□□□□
- PDN □□□□□□□□
- □□□□□

□□□□

1. □□□□ last_seen □□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□

□□□□

- □□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□ **3**□□□□□□

□□□

- □□□□□□□□
- □□□□
- □□□□□

□□□□

1. □□□□□□□□□□
2. □□□□□□
3. □□□□□□

□□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

API □□

□□

- API □□□□
- API □□□□
- □□□□/□□□□
- 500 □□

□□□□□□□□□□

□□ **1**□□□□□□□□

□□□

- 400 □ 422 □□
- □□□□□□
- □□□□□□

□□□□

1. □□□□□□□□□□□□□□
2. □□ API □□□□

3. 认证鉴权

4. 鉴权

鉴权

- 鉴权 API
- 鉴权
- 鉴权 ID

第 2 步

鉴权

- 鉴权
- 鉴权“key_set_id 鉴权”
- 鉴权

鉴权

1. 鉴权
2. 鉴权
 - key_set_id → 鉴权
 - epc_profile_id → EPC 鉴权
 - ims_profile_id → IMS 鉴权

鉴权

- 鉴权
- 鉴权 ID
- 鉴权

第 3 步

鉴权

- 500 鉴权
- 鉴权 API
- 鉴权

□□□□

- □□□ □□□□
-

□□□□□□□□

□□□□□□□□

1. □□□□

- URL: `https://[hostname]:7443/overview`
- □□□□□□□□□□□□□□□□

2. **Diameter** □□

- URL: `https://[hostname]:7443/diameter`
- □□□□□□□□□□□□

3. □□□□

- URL: `https://[hostname]:7443/application`
- □□□□□□□□□□□□□□□□❓❓

API □□□□

□□□□□□□□

```
curl -k https://hss.example.com:8443/api/status
```

□□□□□

```
# IMSI
curl -k https://hss.example.com:8443/api/subscriber/imsi/001001123456789

# MSISDN
curl -k https://hss.example.com:8443/api/subscriber/msisdn/14155551234

# ID
curl -k https://hss.example.com:8443/api/subscriber/1
```

□□□□□□

```
curl -k https://hss.example.com:8443/api/subscriber
```

□□□□□□□□

```
# EPC □□□□
curl -k https://hss.example.com:8443/api/epc/profile/1

# IMS □□□□
curl -k https://hss.example.com:8443/api/ims/profile/1

# □□□□□□
curl -k https://hss.example.com:8443/api/roaming/profile/1
```

□□□□□□

□□ **Diameter** □□□□□□

```
telnet [PEER_IP] 3868
```

□□ **TLS** □□□

```
openssl s_client -connect [hostname]:8443 -showcerts
```

□□□□□□□□

```
# PostgreSQL
psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME] -c "SELECT COUNT(*)
FROM subscriber;"

# MySQL
mysql -h [DB_HOST] -u [DB_USER] -p -e "SELECT COUNT(*) FROM
subscriber;" [DB_NAME]
```

□□□□

□□□□□□□□ **IMSI**□

```
grep "001001123456789" /var/log/omnihss/omnihss.log
```

□□□□□□

```
grep "authentication.*fail" /var/log/omnihss/omnihss.log
```

□□ **Diameter** □□□□□□

```
grep "Diameter peer" /var/log/omnihss/omnihss.log
```

□□□□□□□□

```
grep -i "database.*error" /var/log/omnihss/omnihss.log
```

□□□□

□□□□

□□□□□□□□□□/□□□□□□

1. □□□□□□□□□□□□□□□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□
4. □□□□□□□□□□□□
5. □□□□□□□□□□□□
6. □□□□□□□□□□

□□□□□□

□□□□□□□□

1. □□□□ - □□□□□□□□□□
2. □□□□□□ - □□□□□□□□□□□□
3. □□ - □□□□□□□□□□□□
4. □□ - runtime.exe □□□□□□□□□□□□□□
5. □□ - OmniHSS □□□□□□□□□□□□□□□□
6. □□ - □□□□□□□□□□□□□□
7. □□□□ - □□□□□□□□ IMSI

□□□□□□□□

□□□□□□□□□□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□
- □□□□□□

□□□□□□□□□□□□□□□□□□□□

- 0000000000
- 00000000
- 0000
- 0000

0000000000

0000

0000	00	0000
"00000000"	0000000000	00 00000
"SQN 0000"	SQN0000	0000000
"000000"	000 IMSI	00 IMSI000000
"000000"	enabled=false	0000

Diameter 00

0000	00	0000
"Diameter 00000000"	0000	0000000
"CER/CEA 0000"	000000	00 Diameter 00
"0000000000"	0000000000000000	0000000000
"TLS 0000"	0000	0000

□□□□□

□□□□	□□	□□□□
"□□□□□"	□□□□□	□□□□□
"□□□□"	□□□□	□□□□
"□□□□□□"	□□□	□□□□□
"□□□□"	□□□	□□□□

API □□

□□□□	□□	□□□□
"key_set_id □□□"	□□□□□	□□□□□□□
"IMSI □□□□"	□□ IMSI	□□□□□ IMSI □□□□□□
"□□□□"	□□□□	□□□□□□□□□□

OmniHSS Webhook

←

-
- Webhook
- Webhook
- Webhook
-
-
-
-

OmniHSS **webhooks** IMS
OmniHSS webhook HTTP POST

Webhooks

Webhooks HTTP OmniHSS HSS
API

IMS

Event	Interface	Description
ims_registration	Cx SAR	IMS/VoLTE registration
ims_deregistration	Cx SAR (de-reg)	IMS deregistration
ims_profile_request	Sh UDR	IMS profile request

PCRF (PCRF)

Event	Interface	Description
policy_request	Gx CCR	P-GW policy request
media_authorization	Rx AAR	P-CSCF IMS media authorization

IMSI

Event	Description	Details
imsi_switch	ULR for different IMSI on same SIM	IMSI SIM switch

Webhook

Webhook

OmniHSS provides webhook URL via HTTP POST

```
POST /your-webhook-endpoint HTTP/1.1
Host: your-server.com
Content-Type: application/json
X-OmniHSS-Event: update_location_request
X-OmniHSS-Event-ID: 550e8400-e29b-41d4-a716-446655440000
X-OmniHSS-Timestamp: 2025-01-15T14:30:00Z
```

```
{
  "event": "update_location_request",
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "timestamp": "2025-01-15T14:30:00Z",
  "subscriber": {
    "id": 1234,
    "imsi": "001001123456789",
    "enabled": true,
    "ims_enabled": true,
    "msisdns": [
      {"id": 1, "msisdn": "14155551001"},
      {"id": 2, "msisdn": "14155551002"}
    ],
    "sim": {
      "id": 5678,
      "iccid": "8991101200003204510",
      "is_esim": false
    },
    "key_set": {
      "id": 100,
      "amf": "8000"
    },
    "epc_profile": {
      "id": 1,
      "name": "Premium 100Mbps",
      "ue_ambr_dl_kbps": 100000,
      "ue_ambr_ul_kbps": 50000
    },
    "ims_profile": {
      "id": 1,
      "name": "Standard VoLTE"
    },
    "roaming_profile": {
      "id": 1,
      "name": "International Roaming Allowed"
    },
  },
}
```

```

"subscriber_state": {
  "mme_host": "mme-01.example.com",
  "mme_realm": "epc.mnc001.mcc001.3gppnetwork.org",
  "visited_plmn": "001001",
  "last_update": "2025-01-15T14:30:00Z"
},
"custom_attributes": {
  "account_type": "premium",
  "billing_plan": "unlimited"
}
},
"event_context": {
  "visited_plmn": "310410",
  "mme_host": "mme-roaming.example.com",
  "location_update_type": "initial_attach"
}
}

```

□□□□

□□	□□	□□
event	string	□□□□□□□□ update_location_request □
event_id	string	□ webhook □□□□□ UUID
timestamp	string	□□□□□□ ISO 8601 □□□
subscriber	object	□□□□□□□□□□□□ GET /api/subscriber/:id □□□
event_context	object	□□□□□□□□□□□□

□□□□□□□□

event_context □□□□□□◆◆□□□□□□

□□ update_location_request □

```
{
  "visited_plmn": "310410",
  "mme_host": "mme-roaming.example.com",
  "mme_realm": "epc.mnc410.mcc310.3gppnetwork.org",
  "location_update_type": "initial_attach"
}
```

imsi_switch

```
{
  "previous_imsi": "001001111111111",
  "new_imsi": "310410222222222",
  "sim_id": 5678,
  "previous_mme_host": "mme-home.example.com",
  "new_mme_host": "mme-roaming.example.com"
}
```

ims_registration

```
{
  "scscf_host": "scscf-01.ims.example.com",
  "public_identities": [
    "sip:001001123456789@ims.mnc001.mcc001.3gppnetwork.org",
    "sip:+14155551001@ims.example.com",
    "tel:+14155551001"
  ]
}
```

HTTP

Header	Value	Value
Content-Type	application/json	application/json
X-OmniHSS-Event		update_location_request
X-OmniHSS-Event-ID		UUID
X-OmniHSS-Timestamp		ISO 8601
User-Agent	OmniHSS	OmniHSS/1.0

Webhooks

Webhooks OmniHSS API

Webhook

```
curl -k -X POST https://hss.example.com:8443/api/webhook \
-H "Content-Type: application/json" \
-d '{
  "webhook": {
    "url": "https://your-server.com/omnihss-webhook",
    "events": [
      "update_location_request",
      "ims_registration",
      "imsi_switch"
    ],
    "enabled": true,
    "description": "omnihss webhook"
  }
}'
```

□□□

```
{
  "data": {
    "id": 1,
    "url": "https://your-server.com/omnihss-webhook",
    "events": [
      "update_location_request",
      "ims_registration",
      "imsi_switch"
    ],
    "enabled": true,
    "description": "omnihss webhook",
    "created_at": "2025-01-15T14:00:00Z"
  }
}
```

□□ Webhooks

```
curl -k https://hss.example.com:8443/api/webhook
```

□□ Webhook

```
curl -k -X PUT https://hss.example.com:8443/api/webhook/1 \  
-H "Content-Type: application/json" \  
-d '{  
  "webhook": {  
    "enabled": false  
  }  
'
```

Webhook

```
curl -k -X DELETE https://hss.example.com:8443/api/webhook/1
```

Webhook

webhook

1. **POST** `Content-Type: application/json`
2. - 5 HTTP 200-299
3. -
4. **HTTPS** - TLS/SSL
5. - OmniHSS

Webhook **Node.js/Express**

```

const express = require('express');
const app = express();

app.post('/omnihss-webhook', express.json(), (req, res) => {
  const { event, subscriber, event_context } = req.body;

  console.log(`Received event: ${event}`);
  console.log(`Subscriber IMSI: ${subscriber.imsi}`);

  // TODO
  // ... TODO ...

  // TODO
  res.status(200).json({ received: true });

  // TODO
  processWebhook(req.body).catch(console.error);
});

async function processWebhook(payload) {
  // TODO
  // TODO
}

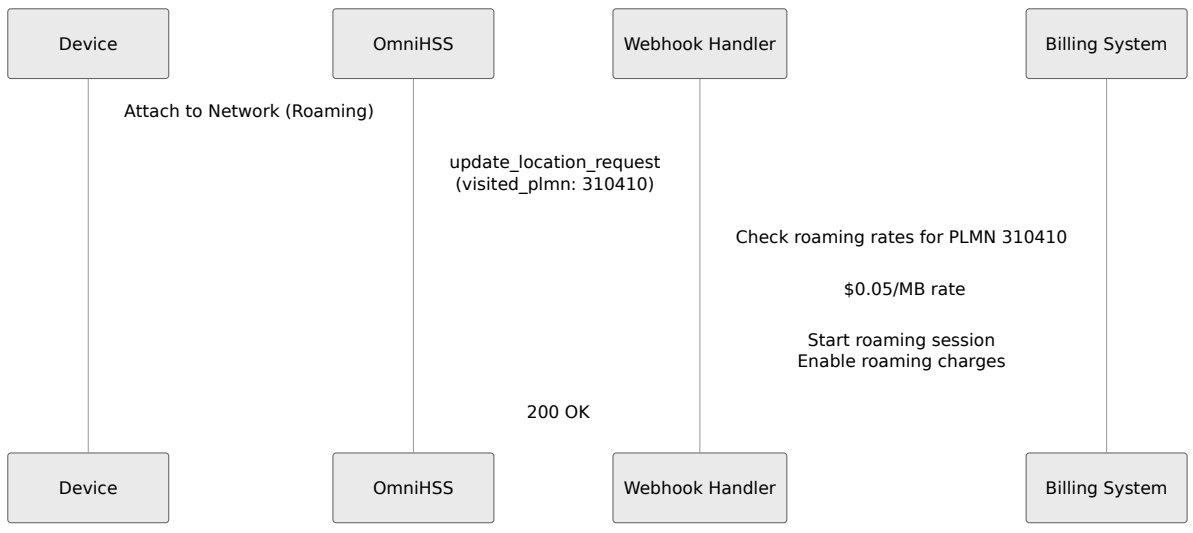
app.listen(3000);

```

□□

1. □□□□□□□□

□□□□□□□□□□□□□□□□□□□□



□□□

- □□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□/□□□□
- □□□□□□□□□□□□

2. □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ □□□□□□□□□□

```
// Webhook
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber, event_context } = req.body;

  if (event === 'update_location_request') {
    await analytics.track({
      event: 'subscriber_location_update',
      imsi: subscriber.imsi,
      visited_plmn: event_context.visited_plmn,
      timestamp: req.body.timestamp,
      profile: subscriber.epc_profile.name
    });
  }

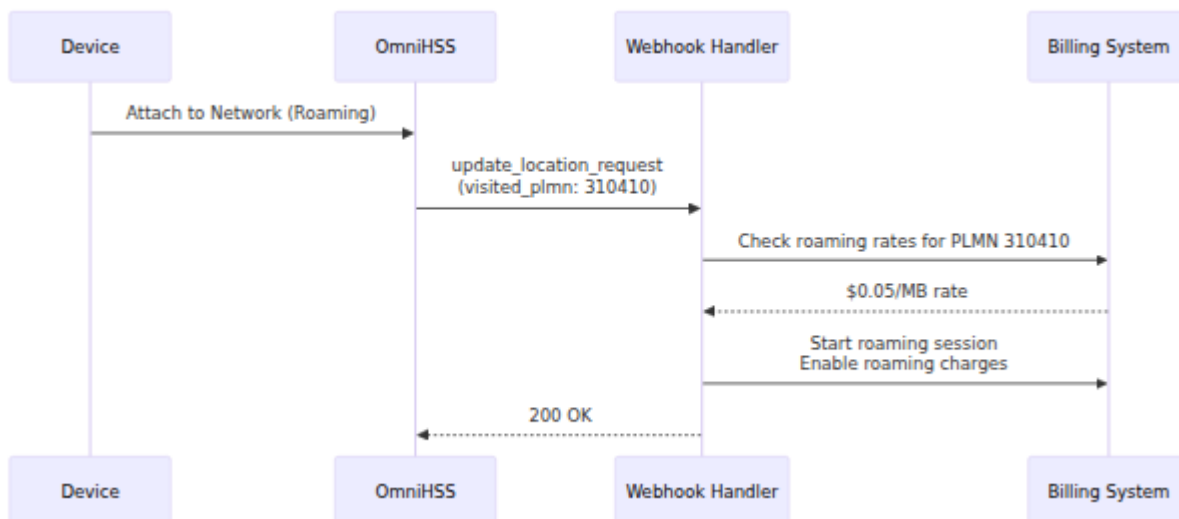
  res.status(200).send();
});
```

□□□□□□

- □□ MME □□□□□□
- □□□□□□□□□□
- □□□□□□
- IMS □□□□□□

3. □□□□□□□□

□□□□□□□□□□□□□□□□□□□□



□□□□□□

1. □□□□□□

- □□□□ A □□□
- 30 □□□□ B □□□□□□□□□□
- □□□□□□□□□□□□□□□□

2. **IMSI** □□□□

- □□ SIM □□□□□ IMSI □□
- □□□ SIM □□□□□□□□ IMSI □□
- □□□□□ SIM □□□□ IMSI□□□□□□□□

3. □□□□□□□

- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□

```

@app.route('/omnihss-webhook', methods=['POST'])
def webhook_handler():
    data = request.json
    subscriber = data['subscriber']
    event_context = data.get('event_context', {})

    if data['event'] == 'update_location_request':
        visited_plmn = event_context.get('visited_plmn')

        #  roaming to blocked
        if visited_plmn in BLOCKED_PLMNS:
            disable_subscriber(subscriber['imsi'])
            alert_security_team(subscriber, 'Roaming to blocked
PLMN')

        #  impossible travel
        if is_impossible_travel(subscriber['imsi'], visited_plmn):
            flag_for_review(subscriber['imsi'])
            alert_fraud_team(subscriber, 'Impossible travel
detected')

    return jsonify({'status': 'ok'}), 200

```

4. IMS

IMS (IP Multimedia Subsystem)

IMS is a key component of 4G LTE networks, enabling VoLTE (Voice over LTE) and IMS services.

```

app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber } = req.body;

  if (event === 'ims_registration' && !subscriber.ims_enabled) {
    // IMS ON - IMS ON
    await omnihss.updateSubscriber(subscriber.id, {
      ims_enabled: true,
      custom_attributes: {
        ...subscriber.custom_attributes,
        volte_activated_at: new Date().toISOString()
      }
    });

    // CRM
    await crm.updateCustomer(subscriber.imsi, {
      features: ['volte']
    });
  }

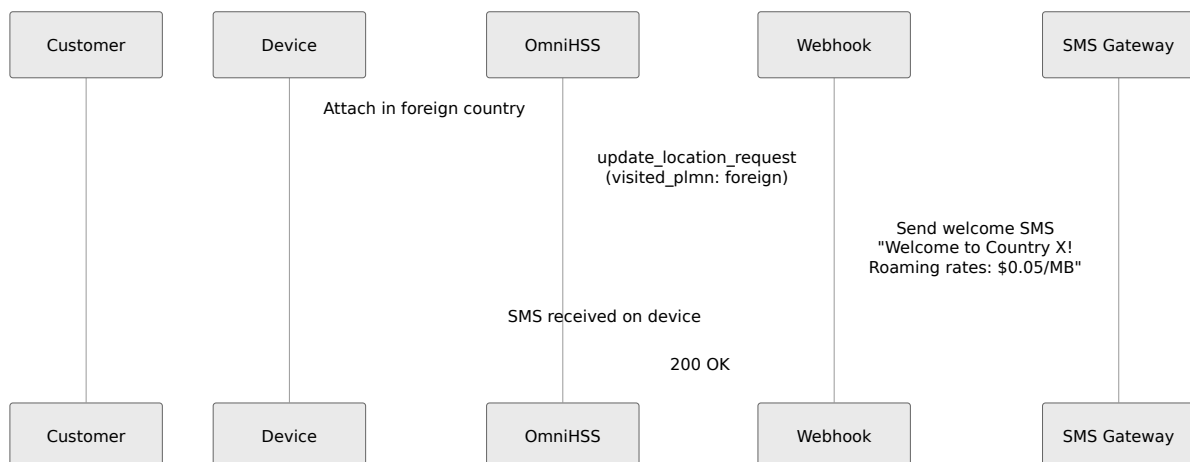
  res.status(200).send();
});

```

5. IMS ON

Scenario 1: IMS ON

Scenario 2: IMS ON



Scenario 3: IMS ON

- "IMEI [IMEI] SIM [SIM ID]"
- "IMEI 80% SIM [SIM ID]"
- "IMEI [IMEI] VoLTE [VoLTE]"
- "IMEI [IMEI] SIM [SIM ID]"

6. IMSI SIM

IMSI SIM IMSI

```

app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber, event_context } = req.body;

  if (event === 'imsi_switch') {
    const { previous_imsi, new_imsi, sim_id } = event_context;

    // IMSI
    await db.logImsiSwitch({
      sim_id,
      from_imsi: previous_imsi,
      to_imsi: new_imsi,
      timestamp: req.body.timestamp
    });

    //
    await billing.endSession(previous_imsi);
    await billing.startSession(new_imsi);

    //
    const switchCount = await db.getSwitchCount(sim_id, '24h');
    if (switchCount > 10) {
      await alertFraudTeam(`Excessive IMSI switching: SIM
${sim_id}`);
    }
  }

  res.status(200).send();
});

```

7. 設定

OmniHSS 設定

設定

- CRM 設定 - 設定
- 設定 - 設定
- 設定 - 設定
- 設定 - 設定
- 設定 - 設定

設定

Webhook 設定

設定 webhooks 設定 OmniHSS 設定

```
# 設定 webhook
curl -k -X POST https://hss.example.com:8443/api/webhook \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "url": "https://your-server.com/omnihss-webhook",
      "events": ["update_location_request"],
      "secret": "your-secret-key-here"
    }
  }'
```

OmniHSS 設定 `X-OmniHSS-Signature` 設定

```
X-OmniHSS-Signature:
sha256=5d7a8f9b2c1e3a4d6f7e8b9c0a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a
```

設定

□□□□

Webhook □□□□ □□□□□□□□□□

- IMSI□□□□□□□□
- MSISDN□□□□□□□□
- □□□□□□□□ PLMN□MME□
- □□□□□□□□

□□□□□□

- **GDPR** - □□ webhook □□□□□□□□ GDPR
 - □□□□ - □□□□□□□□□□□□
 - □□□□ - □□ webhook □□□□
 - □□ - □□ TLS □□ webhook □□
 - □□□□ - □□□□ webhook □□□□□□□□
-

□□□□

Webhook □□□□

□□□□

- □□□□□□ webhook □□□□
- Webhook □□□□□□□□□□

□□□□□□□□

1. □□ **webhook** □□□□□□

```
curl -k https://hss.example.com:8443/api/webhook  
# □□ "enabled": true
```

2. □□ **webhook** □□□□□□

- 設定する webhook の `events` を
- `ims_registration` に設定する

3. HSS の設定

- webhook の設定
- 設定
- DNS の設定

4. テスト

```
curl -X POST https://your-server.com/omnihss-webhook \
  -H "Content-Type: application/json" \
  -d '{"test": true}'
```

Webhook の設定

概要

- HSS の webhook の設定
- Webhook の設定 HSS の設定

手順

1. 設定

- 5 分間隔 HTTP 200
- 設定

2. テスト


```

const processedEvents = new Set();

app.post('/omnihss-webhook', (req, res) => {
  const eventId = req.body.event_id;

  if (processedEvents.has(eventId)) {
    // 중복됨
    return res.status(200).json({ status: 'duplicate' });
  }

  processedEvents.add(eventId);

  // [] webhook...
  processWebhook(req.body);

  res.status(200).json({ status: 'processed' });
});

```

Webhook [] [] [] []

[] [] []

- [] [] [] HTTP 4xx ~ 5xx
- HSS [] [] webhook [] [] [] []

[] [] [] []

1. **401 Unauthorized** - [] [] [] [] []

- [] [] webhook [] [] [] [] [] [] [] [] [] []
- [] [] [] [] [] [] [] [] [] []

2. **400 Bad Request** - [] [] [] []

- [] [] webhook [] [] [] []
- [] [] [] [] Content-Type [] []

3. **500 Internal Server Error** - [] [] [] []

- [] [] [] [] [] [] [] [] [] []

- 000000000000

00000

000000000000

```
app.post('/omnihss-webhook', async (req, res) => {
  try {
    // 0000
    if (!verifyWebhook(req)) {
      return res.status(401).json({ error: 'Invalid signature' });
    }

    // 0000
    if (!req.body.event || !req.body.subscriber) {
      return res.status(400).json({ error: 'Invalid payload' });
    }

    // 00 webhook
    await processWebhook(req.body);

    res.status(200).json({ status: 'ok' });

  } catch (error) {
    console.error('Webhook processing error:', error);
    // 00 200 0000000000000000
    res.status(200).json({ status: 'error', message: error.message
  });
  }
});
```

00000000

000

- 00 webhook 0000000000
- 00000 null 000

00000

1. 00000000 - 0000000000000000IMS0000

2. 接收器 - 接收器 webhook 接收器

接收器

接收器

```
const { subscriber } = req.body;  
  
// 接收器  
const imsProfile = subscriber.ims_profile || { name: 'No IMS' };  
const roamingProfile = subscriber.roaming_profile || { name: 'No  
Roaming' };  
  
// 接收器 MSISDN  
const msisdns = subscriber.msisdns || [];
```

接收器

Webhook 接收器

接收器 webhook 接收器

接收器

- Webhook 接收器
- Webhook 接收器
- 接收器
- 接收器
- 接收器

接收器 **Prometheus/Grafana** 接收器

```
# Webhook 成功率
rate(omnihss_webhook_success_total[5m]) /
rate(omnihss_webhook_attempts_total[5m])

# Webhook 延迟
histogram_quantile(0.95, omnihss_webhook_duration_seconds)
```

Webhook 事件

Webhook 事件是系统记录的事件

示例

```
{
  "timestamp": "2025-01-15T14:30:00Z",
  "level": "info",
  "component": "webhook",
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "webhook_id": 1,
  "event_type": "update_location_request",
  "subscriber_imsi": "001001123456789",
  "endpoint": "https://your-server.com/omnihss-webhook",
  "http_status": 200,
  "duration_ms": 145,
  "error": null
}
```

[← 返回](#) | [API 文档](#) →