

OmniHSS 概要

概要

OmniHSS は 4G LTE (EPC) の IMS (IP 多メディアサブシステム) の認証・課金・政策決定 (HSS) 機能を統合的に提供するためのシステムです。

OmniHSS は Elixir と Erlang VM を利用して構築されています。

機能概要

HSS は LTE の IMS 認証・課金・政策決定を担当します。

- 認証 - ユーザーの認証
- 課金 - 課金情報の管理
- 政策決定 - 政策決定
- 認証 - ユーザーの認証
- 課金 - 課金情報の管理
- 政策決定 - 政策決定
- 課金 - 課金情報の管理 (EIR) の管理

インターフェース

外部インターフェース

- S6a** インターフェース - LTE/EPC の MME との接続
- Cx** インターフェース - IMS の AS との接続
- Sh** インターフェース - IMS の AS との接続 (EIR)
- S13** インターフェース - 課金情報 (OmniHSS の EIR)
- Gx** インターフェース - 課金情報 (OmniHSS の PCRF)
- Rx** インターフェース - IMS の AS との接続 (OmniHSS の PCRF)

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

OmniHSS

OmniHSS

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

OmniHSS

OmniHSS

OmniHSS

- **Diameter** - Diameter
- **HTTP** - HTTP
- **Web** - Web

OmniHSS

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

기능

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

기능

- **Galera** - HA HA 정보

기능

기능

기능 (Web)

URL: https://[hostname]:7443

기능 Diameter 정보

API

URL: https://[hostname]:8443

RESTful API 정보

기능

- **config/runtime.exs** - 정보 (Diameter 정보)
- **priv/cert/** - HTTPS Diameter TLS 정보

목차

1. 프로젝트 - 프로젝트 소개
2. **Diameter** 프로토콜 - Diameter 프로토콜 소개
3. 프로토콜 - API 엔드포인트 `/api/subscriber/imsi/:imsi`
4. 데이터베이스 - PostgreSQL 데이터베이스 SQL 쿼리

프로젝트 소개

목차

프로젝트 소개 stdout/stderr 로그 출력 (systemd/supervisord) 구성

목차

- **Diameter** 프로토콜 - Diameter 프로토콜 소개
- 프로토콜 - runtime.exs 프로토콜
- 데이터베이스 - PostgreSQL 데이터베이스

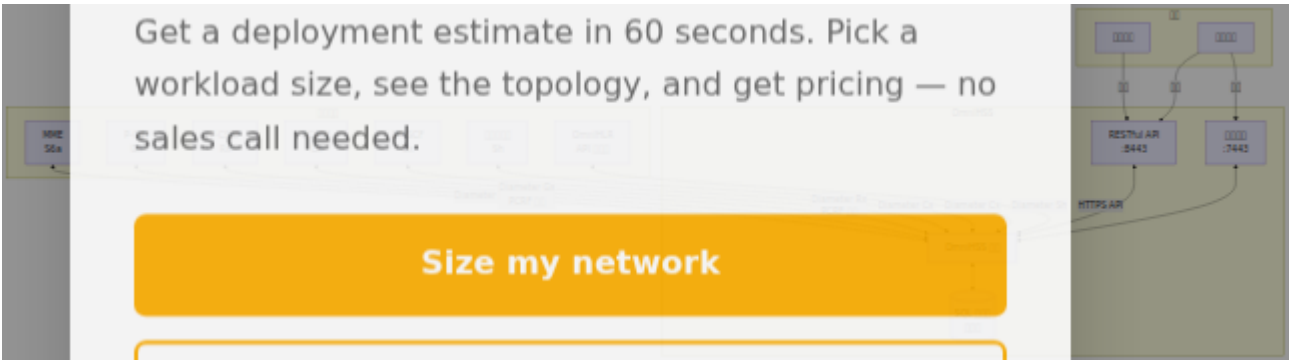
목차

- **API** 엔드포인트 - GET `/api/status`
- 프로토콜 - 프로토콜 소개
- 데이터베이스 - PostgreSQL 데이터베이스

프로젝트 소개

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

网络



网络

网络部署和配置工具

- 网络部署和配置工具
- 网络部署和配置工具
- 网络部署和配置工具
- 网络 **API** 网络部署和配置工具

网络 1.0

网络 Omnitouch 网络

EIR

HSS EIR EIR IMEI

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

Diameter

	ID		
S13	16,777,252	MME/SGSN	

EIR

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

00000

- whitelist - 00000
- blacklist - 00000
- greylist - 00000

0000000000 IMEIIMEISV 0 IMSI 00

000000

00	00	00	0000
000	0	000000	0000000
000	1	00000/000	0000000
000	2	0000000	000000

S13 00

000000

0000000000ECR0/0000000000ECA0

000MME/SGSN → HSS0EIR0

MMME 消息格式

AVPs

- Session-Id
- Origin-Host, Origin-Realm
- Destination-Realm
- Auth-Session-State
- Terminal-Information
 - IMEI 15 字节
 - Software-Version 2 字节
- User-Name IMSI 16 字节
- Vendor-Specific-Application-Id

EIR

1. IMEI Software-Version IMSI
2. IMSI
 - 16 字节
 - 16 字节
3. **IMEISV** IMEI + Software-Version
 - **IMEI** 15 字节
 - **IMSI** 16 字节
 - 16 字节
4. 16 字节

AVPs

- Session-Id 16 字节
- Result-Code: 2001 1 字节
- Equipment-Status: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255

消息

- Experimental-Result: 5422 1 字节
- Experimental-Result: 5012 1 字节

IMEI

IMEI

EIR

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

IMEI

1. IMEISV

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

2. IMEI

- IMEI "35979139461611"
- EIR
-

3. IMSI

- IMSI "999999876543210"
- EIR
-
-

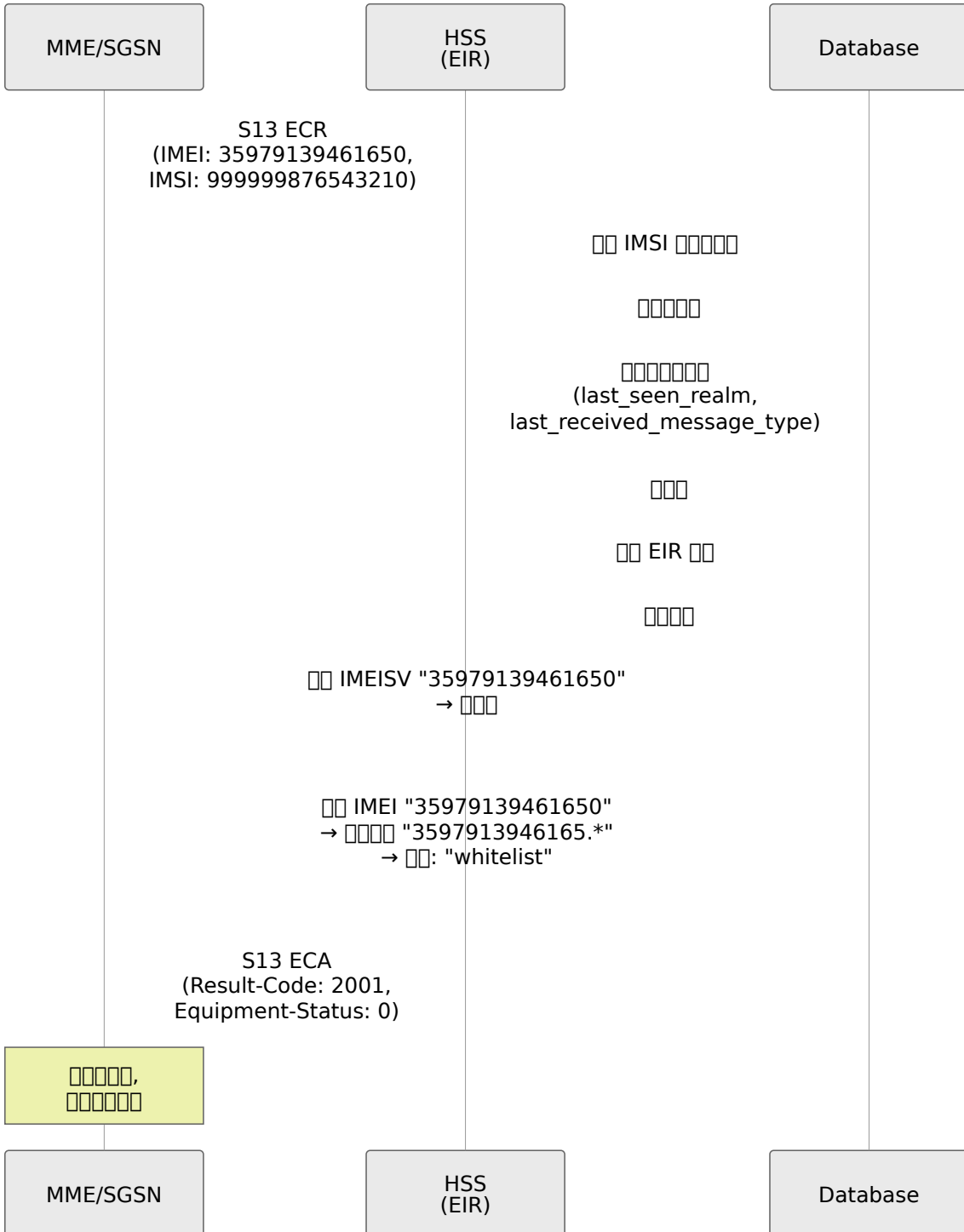
4.

- `eir_unknown_equipment_behaviour`
- `:`
 - `:whitelist` -
 - `:blacklist` -
 - `:greylist` -
 - `:reject_unknown_equipment` - 5422

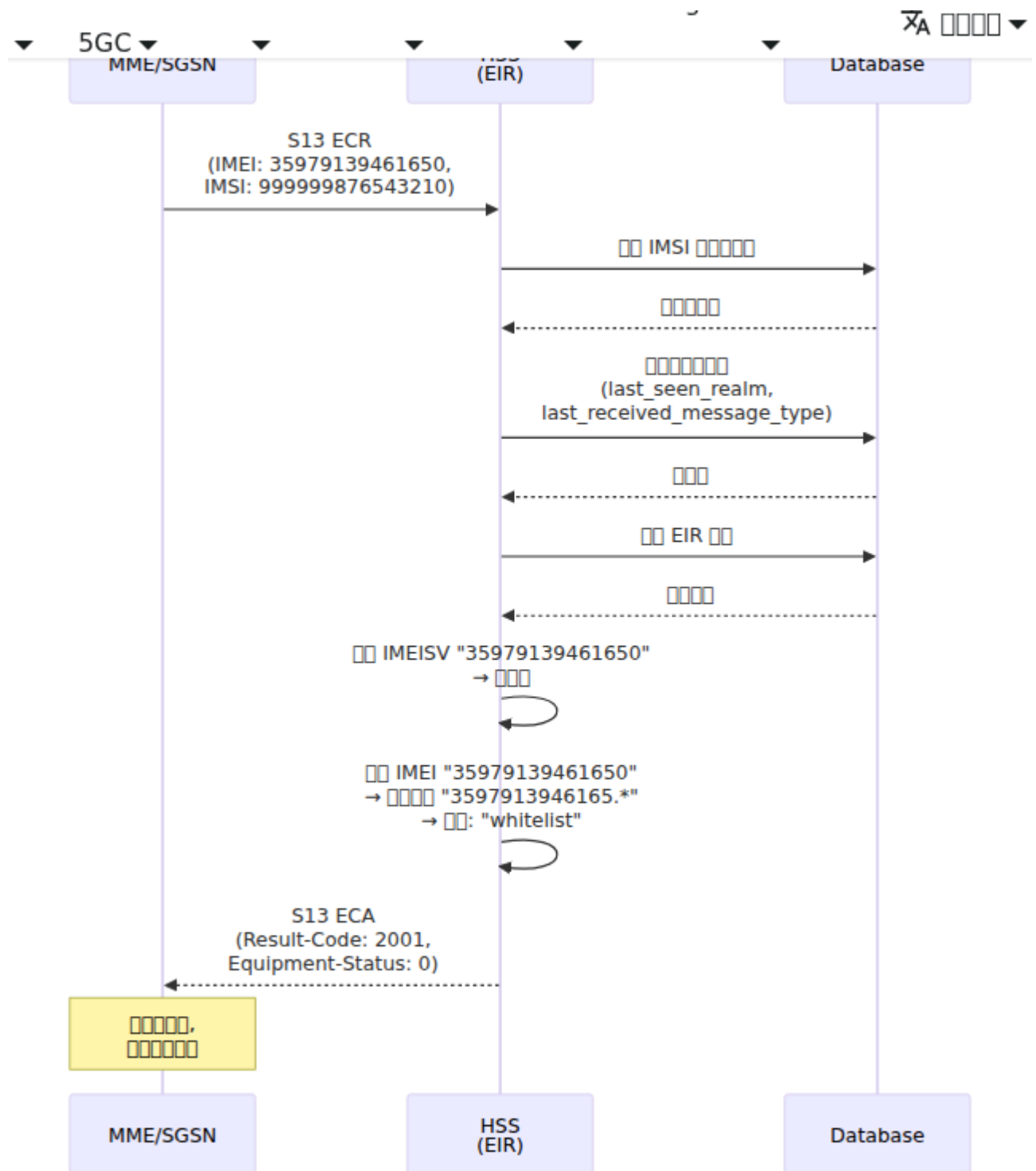
<code>"35979139461650"</code>	IMEI	/
<code>"3597913946165.*"</code>	IMEI	/
<code>"3597913946161108"</code>	IMEISV	
<code>"999999876543210"</code>	IMSI	
<code>"359791.*"</code>	TAC	

□□□□□□

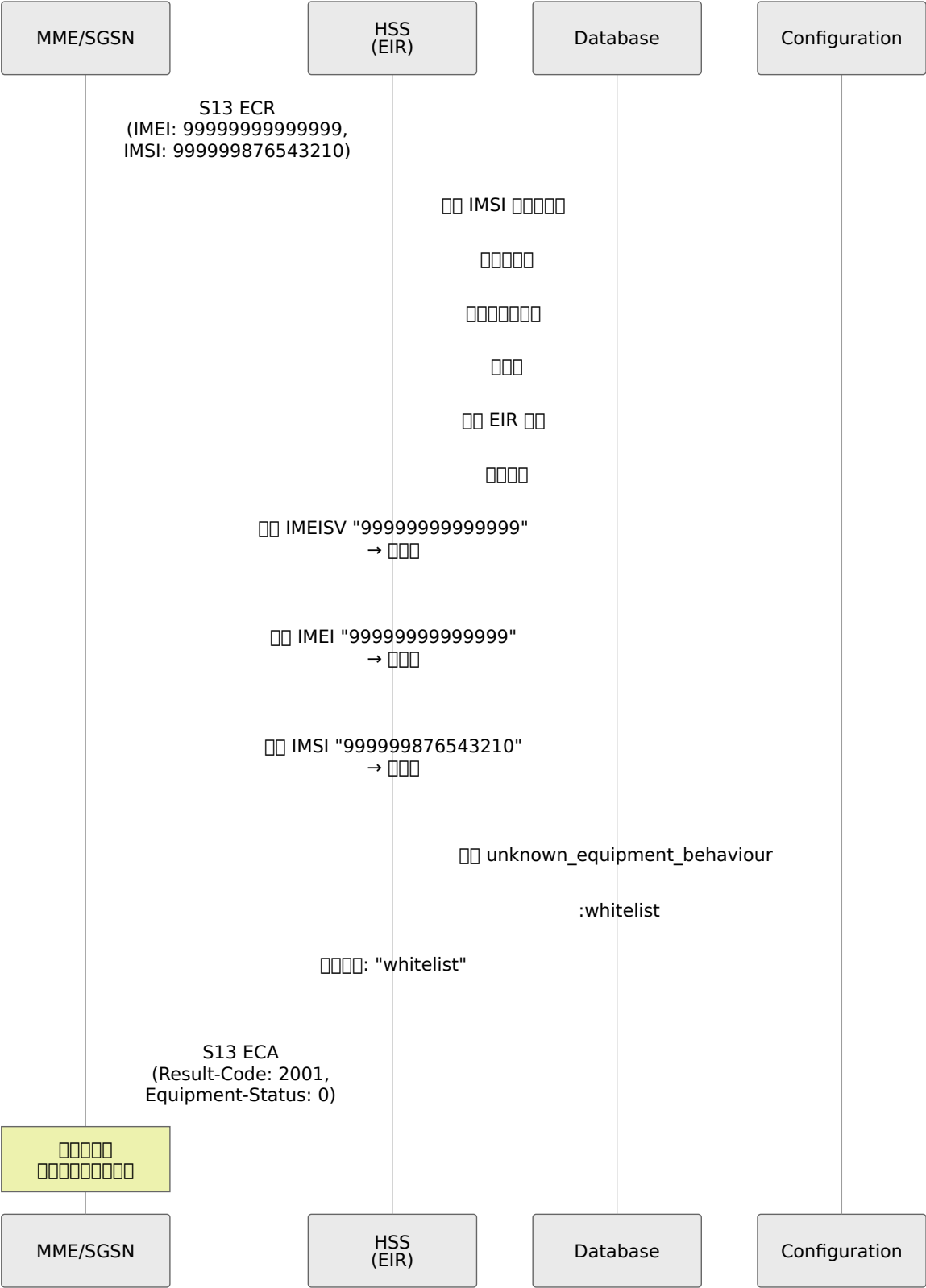
□□ 1□□□□□ - □□□□□ IMEI



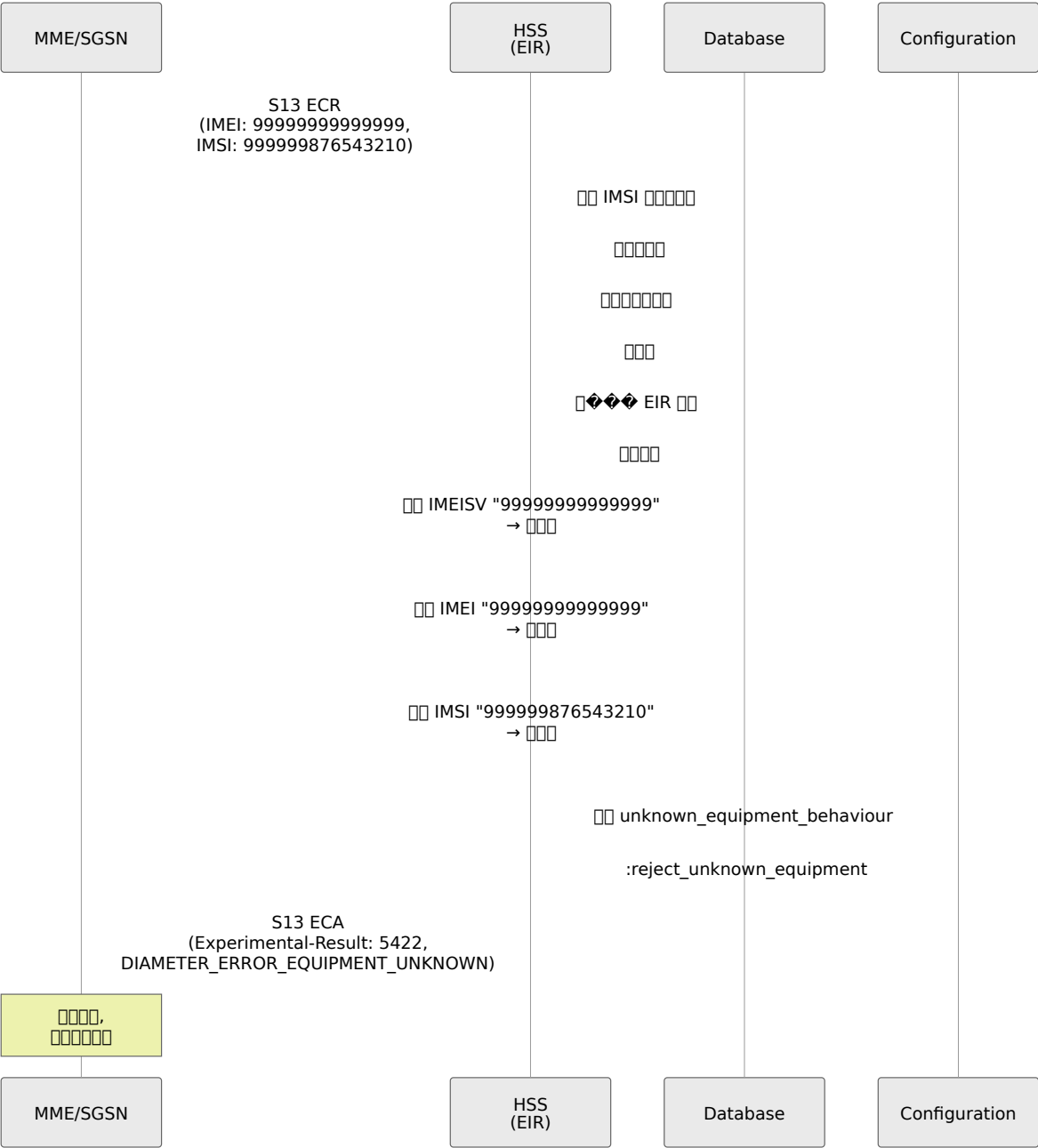
2. IMEI Whitelist



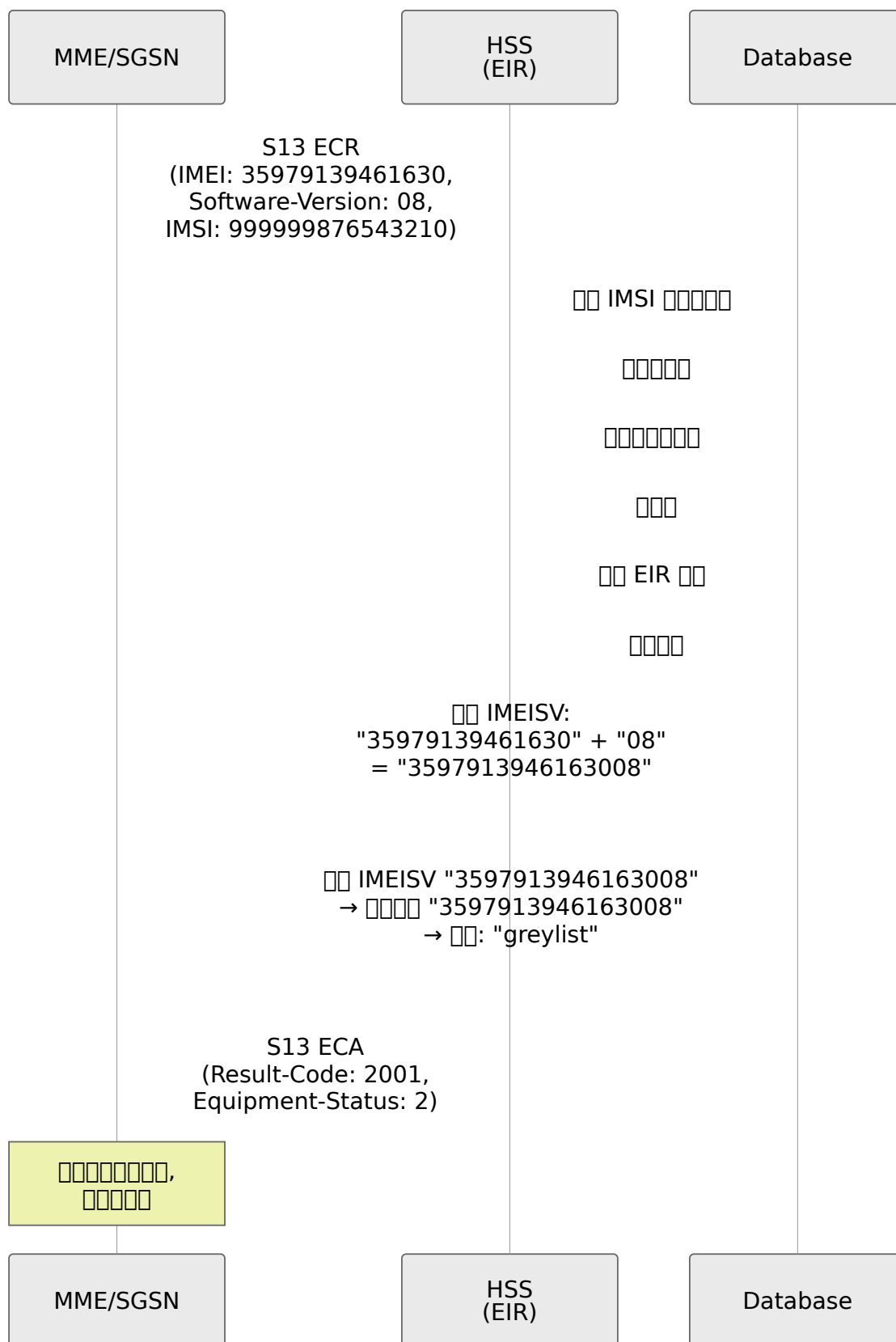
3 -



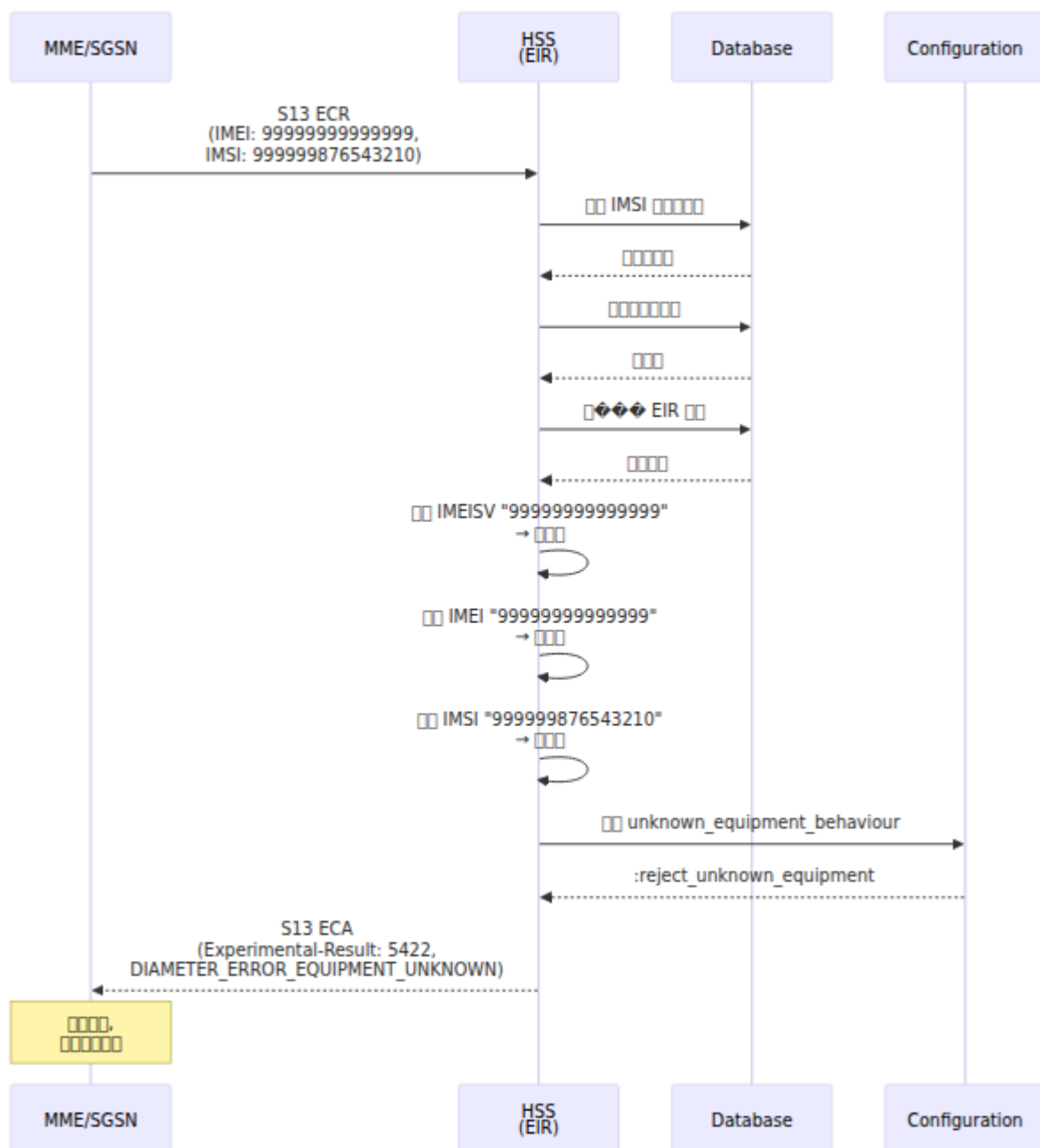
4 -



6 - IMEISV



7 - IMSI



REST API

EIR

/api/eir/rule

EIR

GET

```
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

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

HTTP 200

```
{
  "data": {
    "id": 1,
    "action": "whitelist",
    "regex": "3597913946165.*"
  }
}
```

EIR

```
POST /api/eir/rule
Content-Type: application/json

{
  "action": "blacklist",
  "regex": "35979139461640"
}
```

- `action` `"whitelist"` `"blacklist"` `"greylist"`
- `regex`

HTTP 201

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

HTTP 400

```
{
  "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.*"
}
```


返回HTTP 200

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

删除 EIR 规则

DELETE

DELETE /api/eir/rule/{id}

返回HTTP 204 无响应

配置

Diameter 配置

S13 配置 `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}
  ]
}
```

配置

在 `config/runtime.exs` 中配置

000

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

0000

- :whitelist - 0000000000000000
- :blacklist - 000000000000
- :greylist - 000000000000
- :reject_unknown_equipment - 00 Diameter 00 54220000

000

- 00/000 :whitelist - 0000000
- 00000000 :whitelist - 0000000000
- 00000000 :greylist - 000000000000
- 00000000 :reject_unknown_equipment - 00000000

00000

00000	00	00	00
2001	00	DIAMETER_SUCCESS	0000000
5422	00	DIAMETER_ERROR_EQUIPMENT_UNKNOWN	000000000000000000
5012	00	DIAMETER_ERROR_UNKNOWN	00000

00

1. 00000000

000000000000

POST

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

IMEI 35979139461640

2. 黑名单

黑名单规则

POST

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

TAC 359791394

3. 黑名单

黑名单规则SIM卡

POST

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

IMSI 999999876543210

4. 灰色清單

API 測試工具

URL

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

API 測試工具

5. 黑名單

API 測試工具

URL

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

API 測試工具 IMEI 359791394616 "05" 00

API 測試工具

API 測試工具

EIR 測試工具

- **S13** 測試工具 - ECR/ECA 測試
- 測試工具 - 測試工具 IMEI/IMEISV/IMSI 測試

- **EIR** -
- **REST API** -

1. **IMEISV** IMEI +
2. **IMEI** IMEI
3. **IMSI** IMSI
4.

- -
- -
- -
- -

IMEI

IMEI EIR

- IMEI
-
- API

EIR ID

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

□ □ □ □

- 0000000000 S13 ECR 00
- REST API EIR 0000
- IMEI 000000000000

- Diameter 100 - S13 100mm
- API 100 - 100mm API 100
- 100 - 100 HSS 100
- 100mm - 100mm

IMEI

IMEI 000015 0000

```

35 9791 394616 1
| | | | | L 0000Luhn 000
| | | | | L 00006 0000
| | | | | L FAC0000000004 0000
L TAC0000000000 8 000000 RBI0
| | | | | L RBI0000000002 0000
| | | | | L 000/0006 0000

```

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	IMEISVIMEI + SV

PCRF

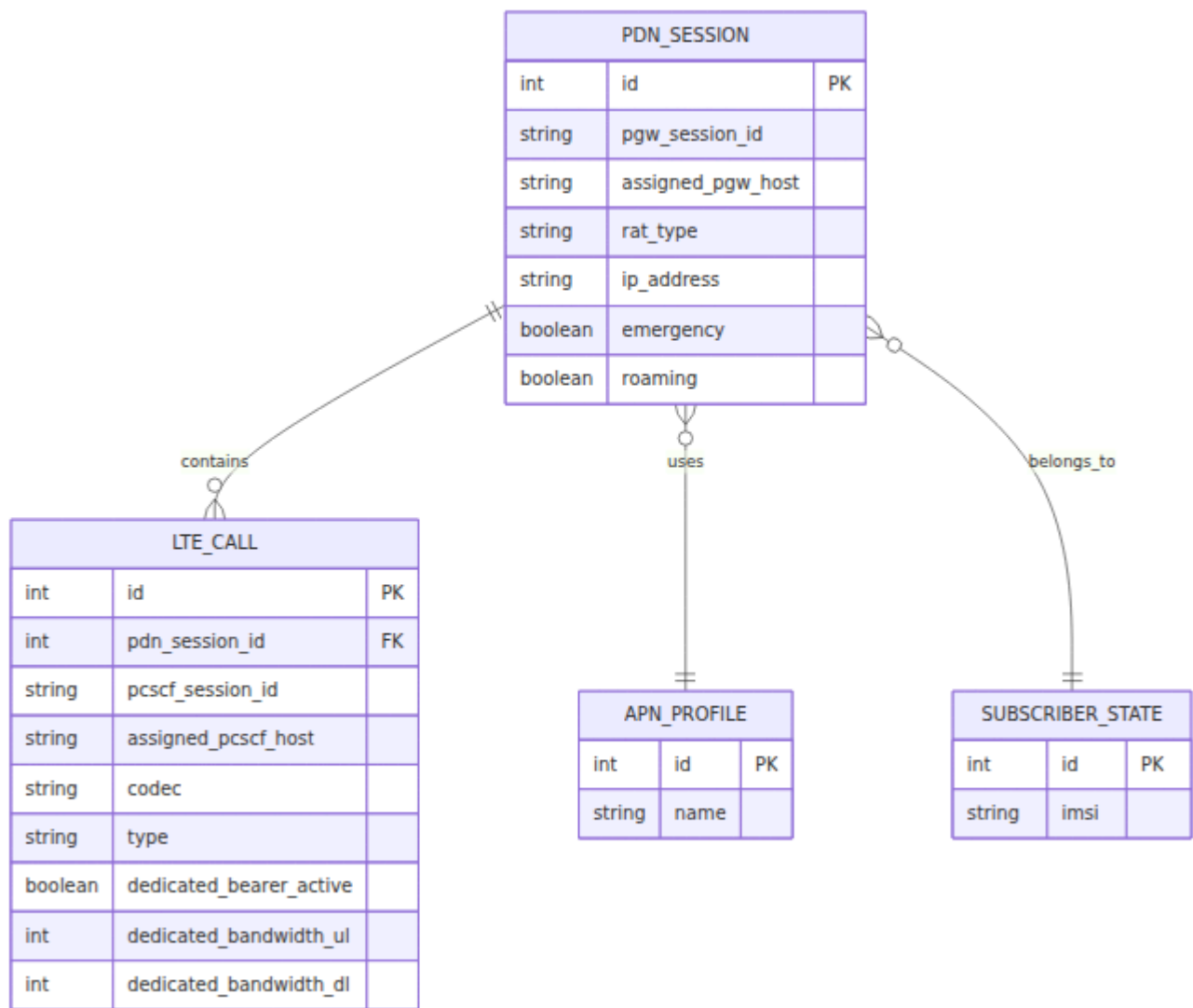
HSS PCRF PCRF LTE QoS

- **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

PGW

1. PGW - CCR-I

PGW sends PDN

AVP

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

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

PCRF 設定

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

AVP

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

2. 設定 - CCR-U

PGW 設定 RAT 設定

PCRF 設定

1. ID 設定
2. RAT 設定
3. 設定

Result-Code 2001 設定

3. 設定 - CCR-T

PGW PDN 設定

PCRF 設定

1. ID 設定
2. 設定
3. 設定

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

- **IPsec** - **IP** **Security** Protocol
- **VPN** - **V**irtual **P**rivacy **N**etwork
- **SSL** - **S**ecure **S**ockets **L**ayer
- **SSH** - **S**ecure **S**hell
- **SLA** - **S**ervice **L**evel **A**greement QoS

QoS □□□□

QoS APN

```
{
  "QoS-Class-Identifier": 9,           // QCI9 = 0000
  "APN-Aggregate-Max-Bitrates-UL": 50000, // kbps
  "APN-Aggregate-Max-Bitrates-DL": 100000, // kbps
  "Allocation-Retention-Priority": {
    "Priority-Level": 8,
    "Pre-emption-Capability": 1,       // 0000
    "Pre-emption-Vulnerability": 1     // 0000
  }
}
```

QoS VoLTE

```
{
  "QoS-Class-Identifier": 1,           // QCI 1 = 0001
  "Max-Requested-Bandwidth-UL": 128000, // bps
  "Max-Requested-Bandwidth-DL": 128000, // bps
  "Guaranteed-Bitrate-UL": 128000,
  "Guaranteed-Bitrate-DL": 128000
}
```

Rx ☐ ☐

□ □ □ □ □

1. AA □□□AAR□/ AA □□□AAA□

IMS P-CSCF IMS VoLTE

□□ **AVP** □

- Session-Id P-CSCF
- Subscription-Id IMSI 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

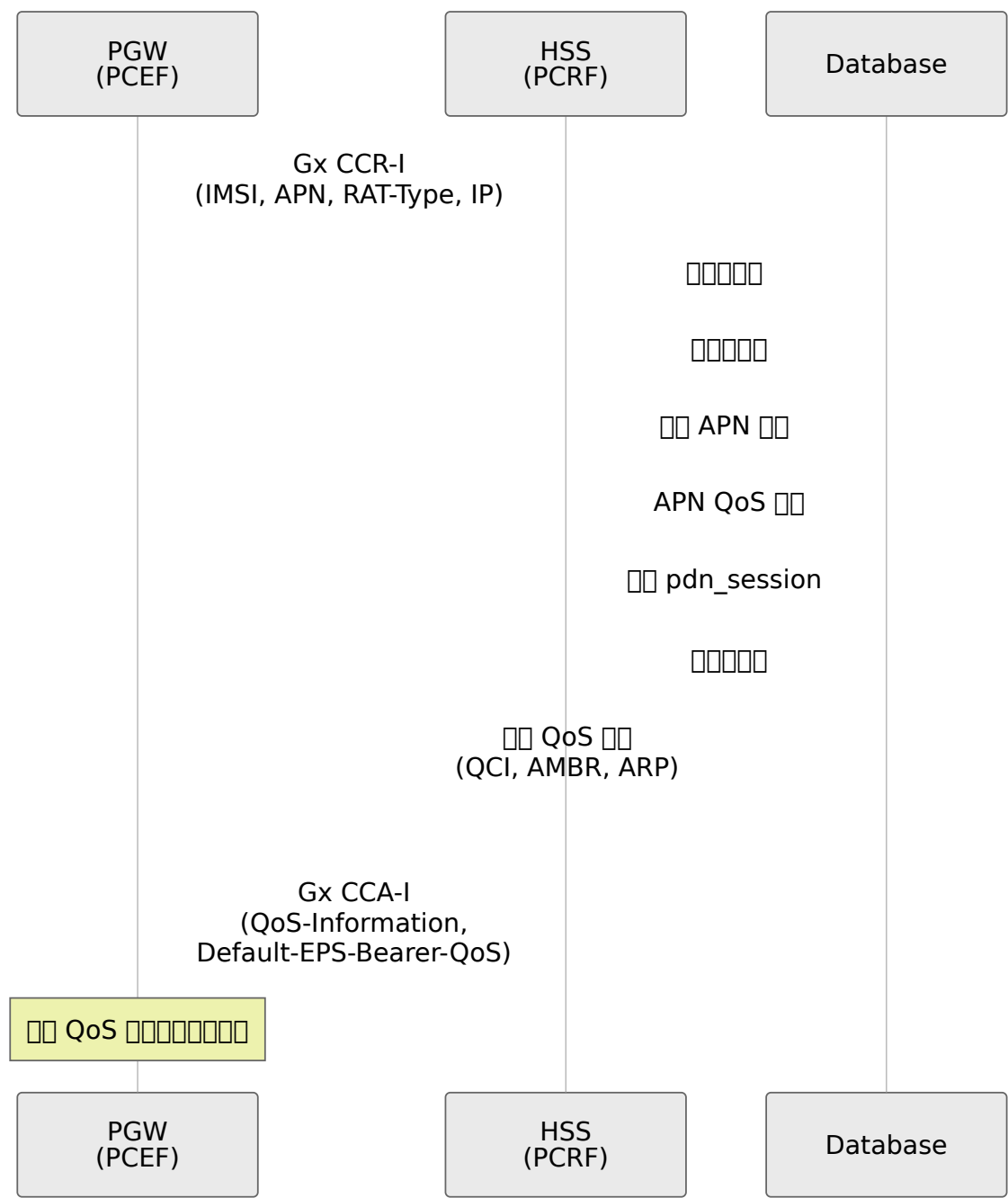
□□□P-CSCF □□ IMS □□□□□□□□

PCRf ☐ ☐ ☐

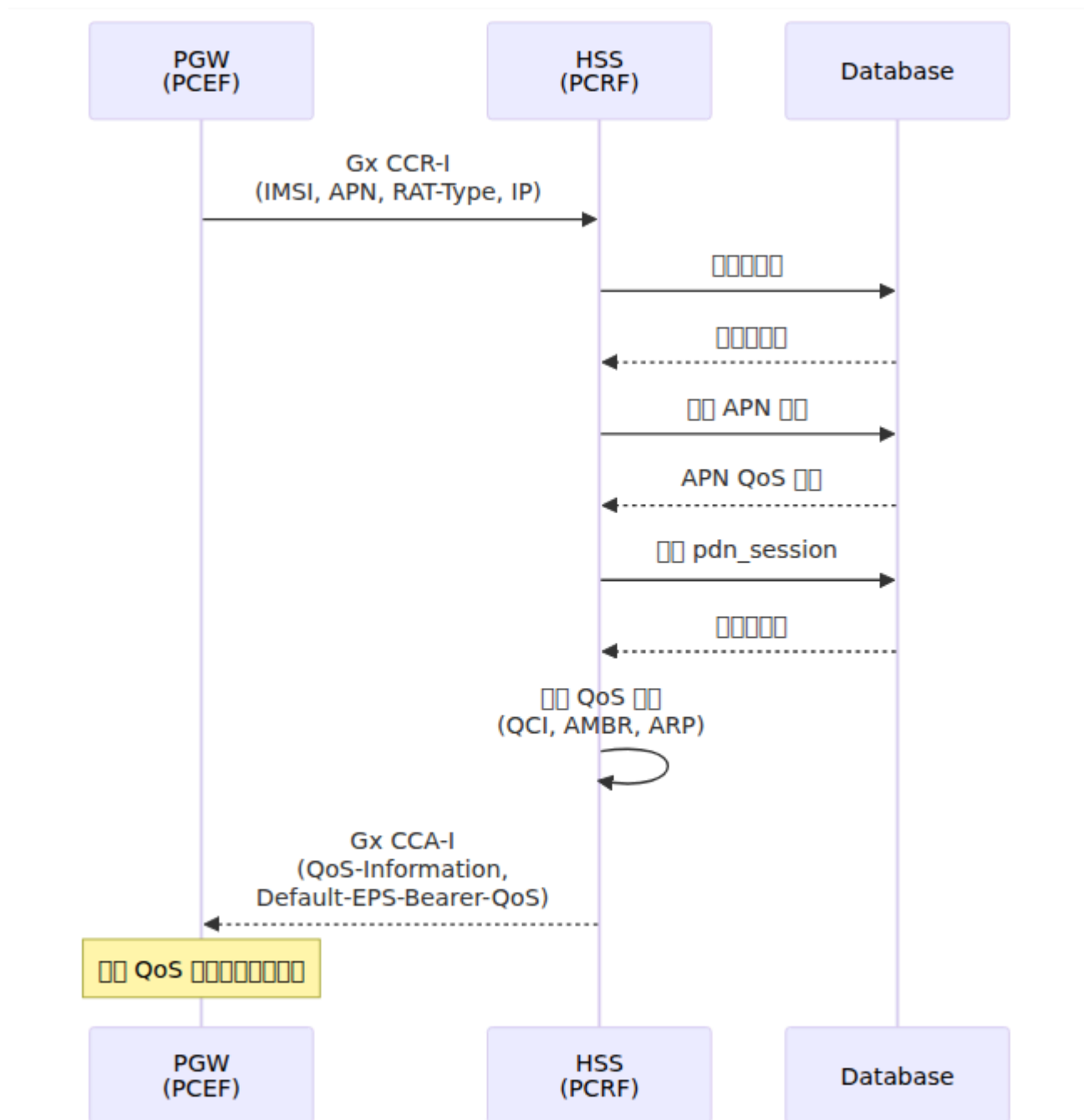
1. 网络 P-CSCF 网络 ID 网络名称
2. 网络 **Gx RAR** 网络 **PGW** 网络名称
3. 网络名称
4. 网络 STA 网络

Result-Code 2001

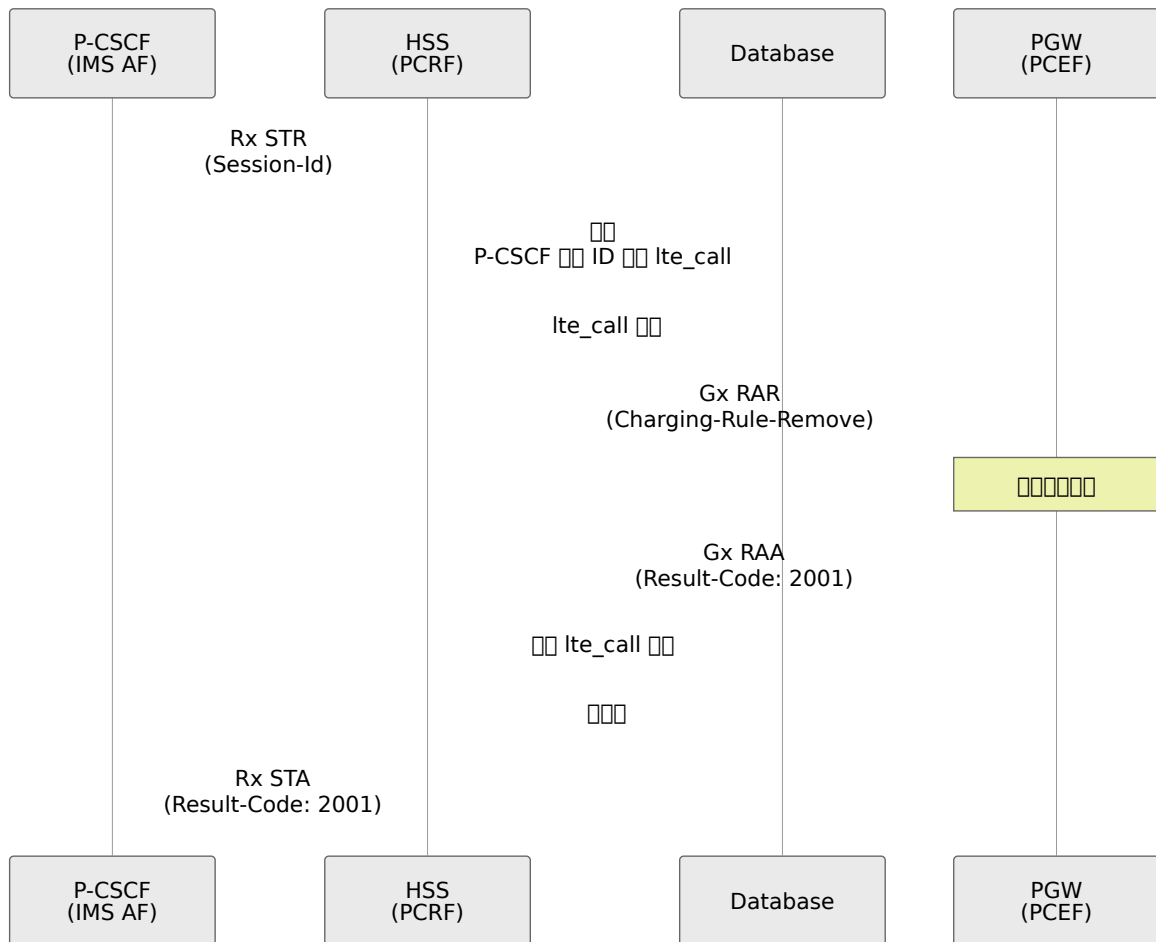
1 PDN



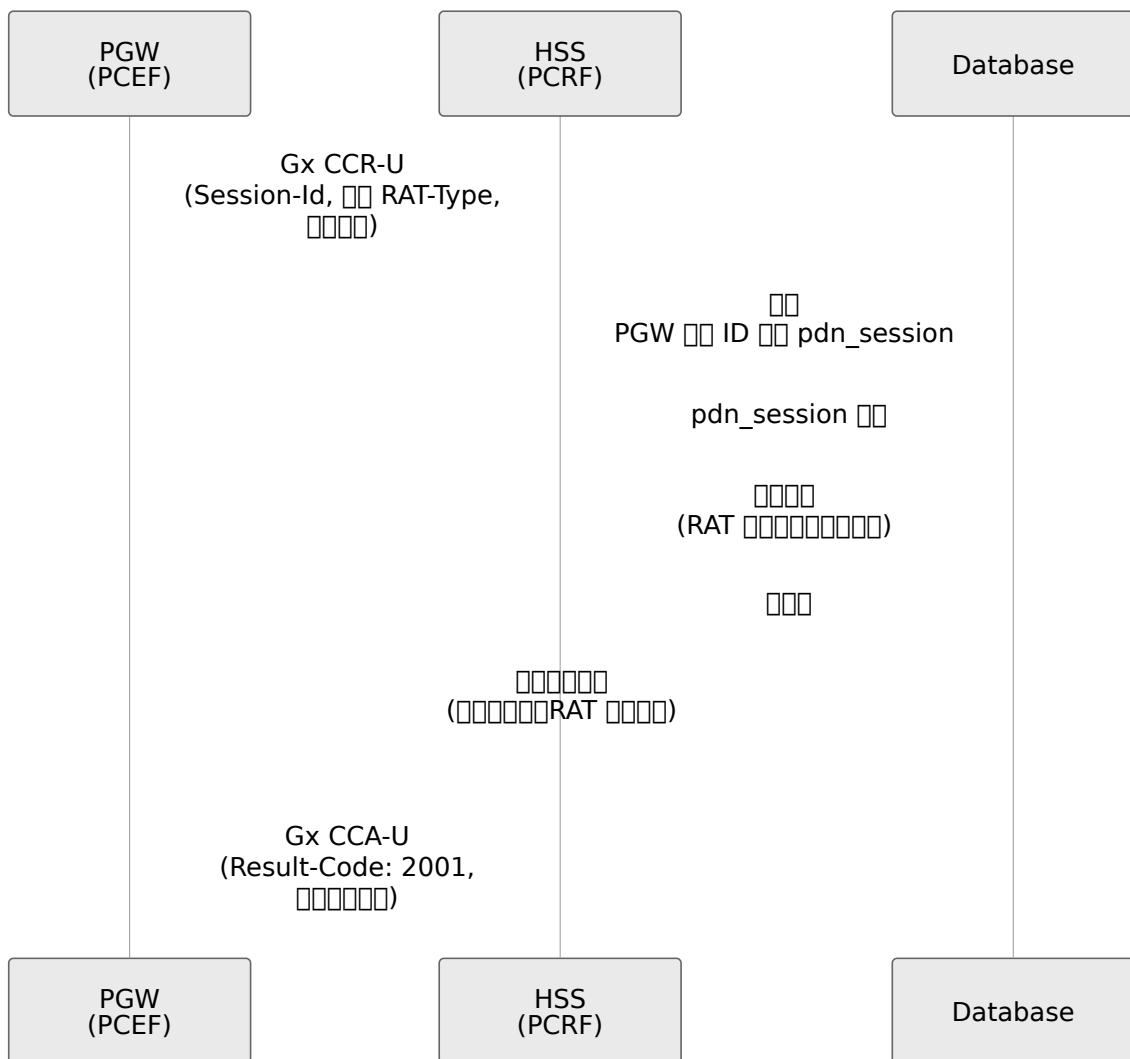
2 VoLTE Rx AAR → Gx RAR



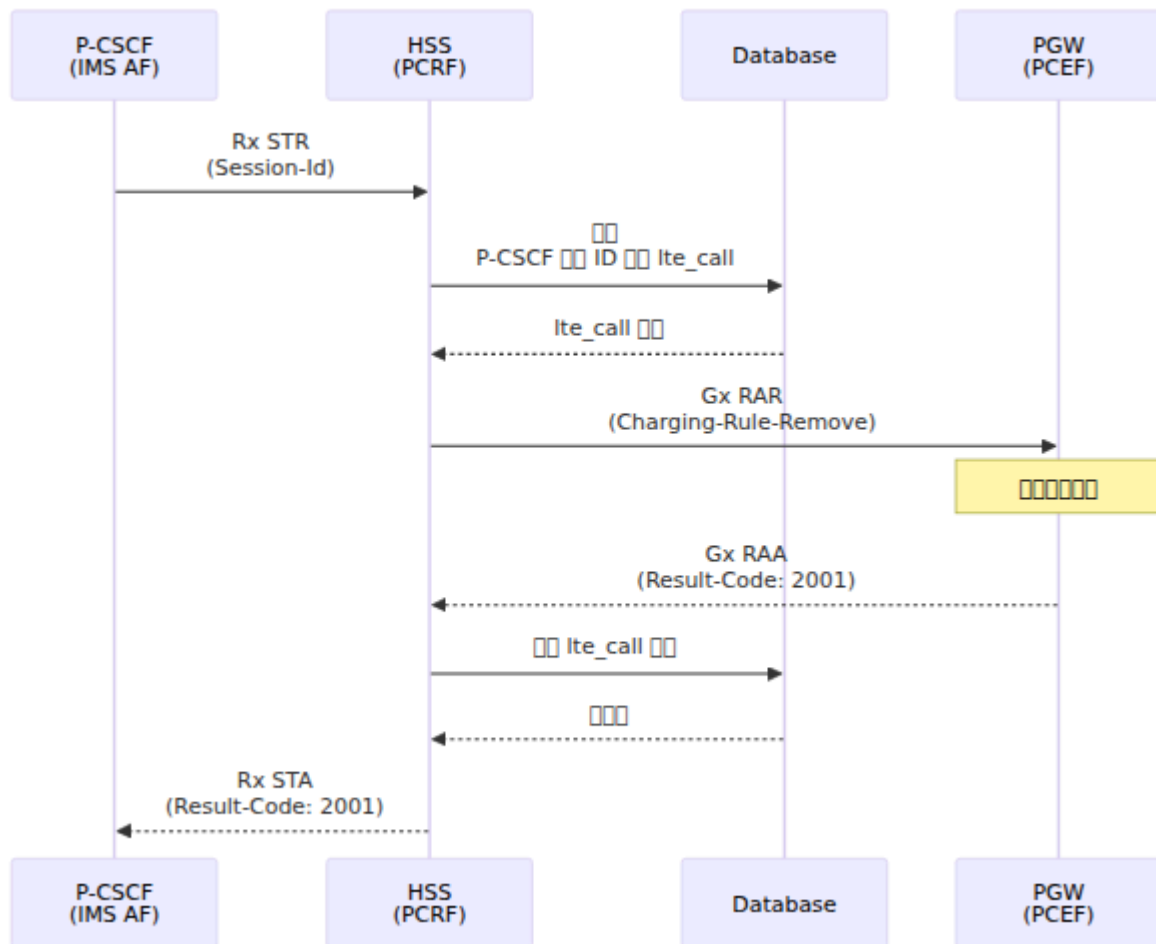
3 VoLTE Rx STR → Gx RAR



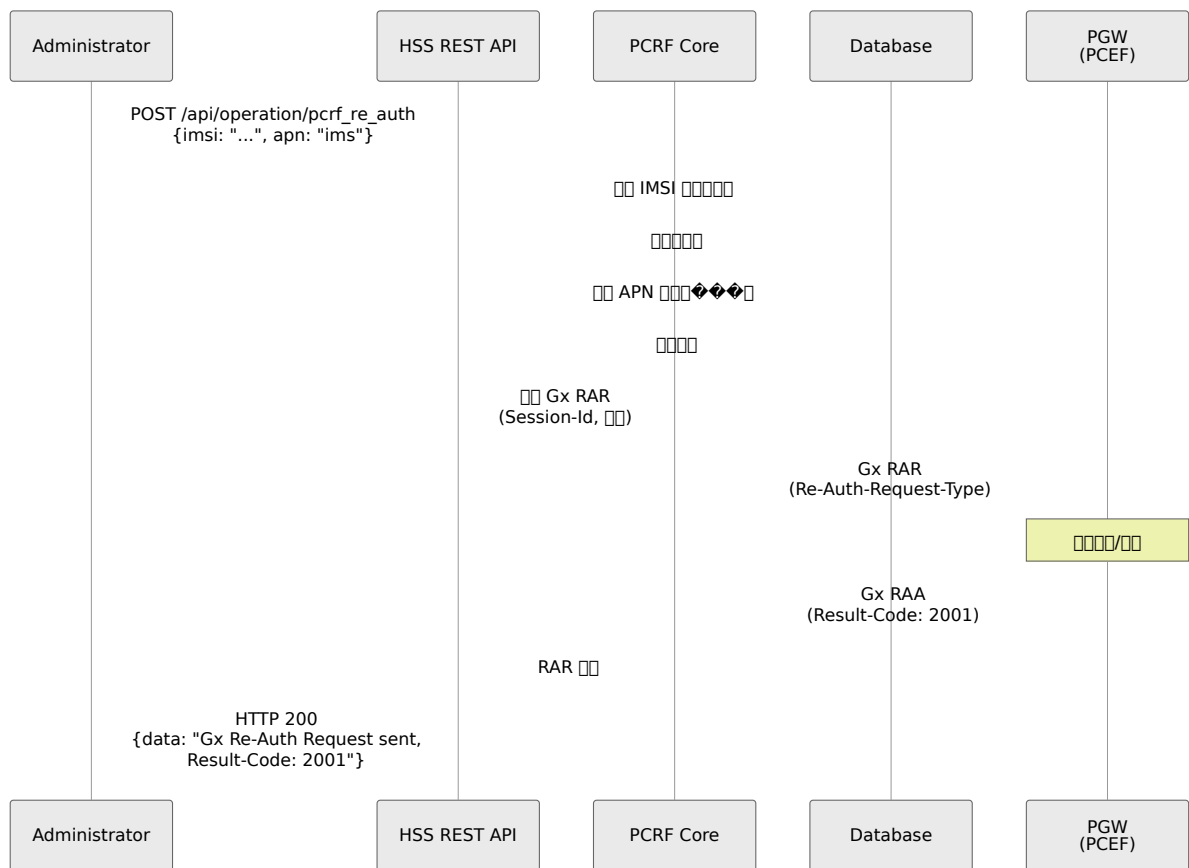
4 PDN



5 PDN



6 REST API



REST API

PCRF

POST /api/operation/pcrf_re_auth

Gx

APN QoS

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

成功HTTP 200

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

失败HTTP 400

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

API

PCRF 通过 APN 和 QoS 参数 REST API 接口

通过 APN QoS 参数 QCI 参数 APN 和 PDN 参数 PGW 和 Gx 参数 RAR 参数

接口

接口

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

1. APN 参数

APN 参数 IP 地址

POST /api/apn/identifier

Request

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

IP version

- "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"
  }
}
```

Response

- 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

項目	型別	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 = 優先度最高
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 - 背景トラフィック - Non-GBR 300ms 遅延

201 HTTP

```
{
  "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
  }
}
```

QoS 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

QoS

□□ = □□□□□□□□

□ □ □ □ □ □ □ □

1. QoS 設定
2. 設定された QoS を APN 毎に PDN 毎に
3. 設定された QoS を PGW 毎に Gx RAR
4. PGW 毎に QoS を設定
5. 設定された API を利用して

APN QoS

□ □ □ □ □

```
DELETE /api/apn/identifrier/{id}
DELETE /api/apn/qos_profile/{id}
DELETE /api/apn/profile/{id}
```

APN

- APN APN QoS
- APN

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
  }
}
```

/M2M

```
{
  "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_qos_profile` 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
 - QCI1
 - Max-Requested-Bandwidth AVP
 - Flow-Description AVP

項目	項目	項目	項目
2001	項目	DIAMETER_SUCCESS	項目項目
5001	項目	項目項目	IMSI 項目項目項目項目
5002	項目	項目項目	PDN 項目項目項目項目/項目
5063	項目	項目項目	IMS 項目項目項目

1111

□ □ □ □

PCRFB 0000

- **PDN** - APN
- **VoLTE** - IMS
- **QoS** - APN
- -

□ □ □ □ □ □

PCRF

- 最大 Gx 100000000/秒
- 最大 TFT 100000000/秒
- 最大 100000000/秒
- 最大 100000000/秒

☐ TFT ☐

□□□□

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

API 入門

← API 入門

API

- API 入門
- API 入門

API 入門

400 入門

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

API

- JSON 入門
- API 入門
- API 入門

404 入門

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

API

- `imei/imsi/number`
- URL `id` `key_set_id`

422 未处理的数据

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

原因

- 数据重复
- 数据不存在
- 数据格式错误

500 服务器内部错误

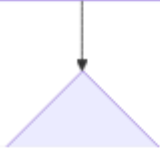
```
{
  "error": "Internal server error"
}
```

原因

- 服务器内部错误
 - 数据格式错误
-



API Request



ore 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

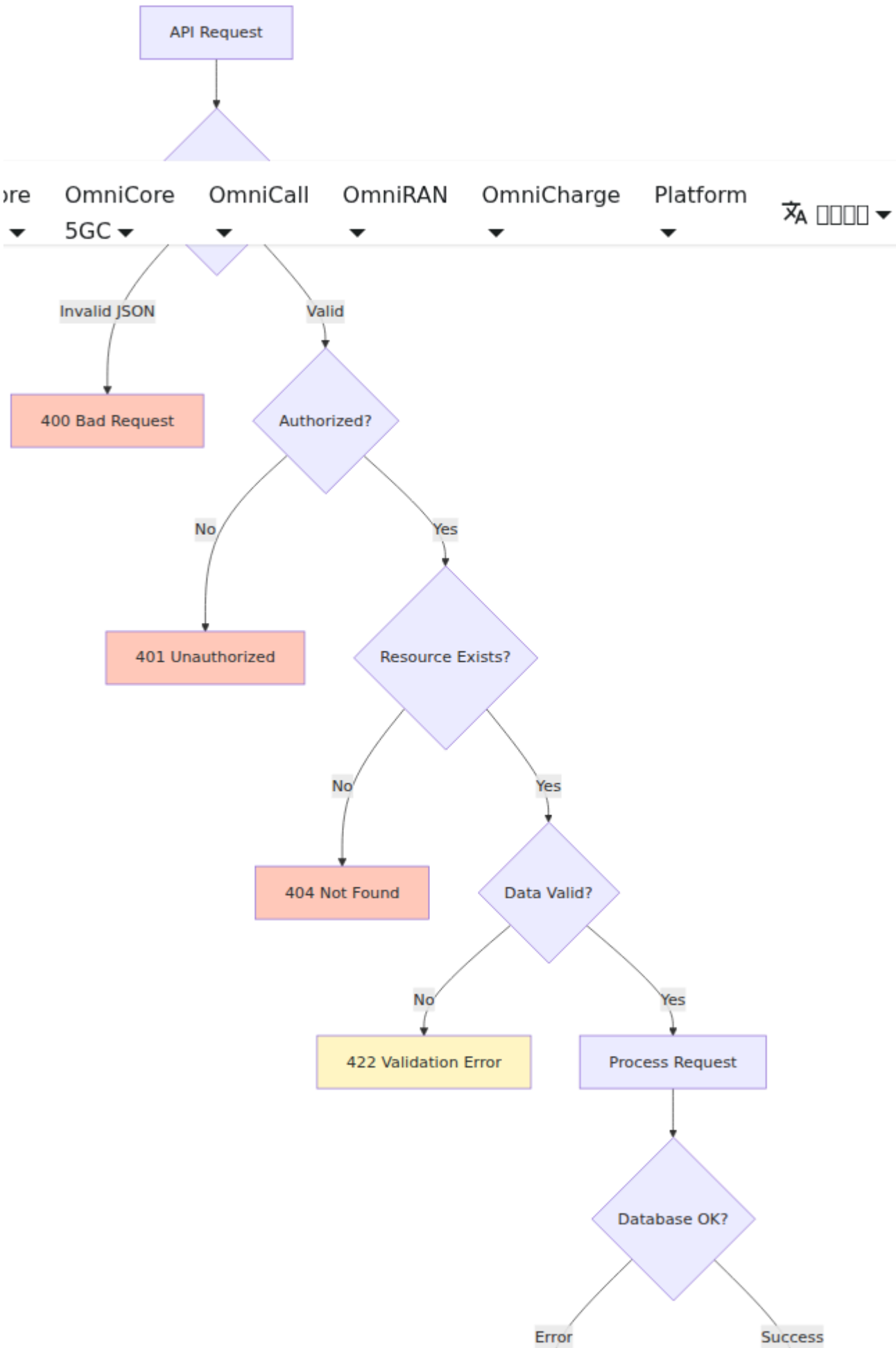
Yes

Process Request

Database OK?

Error

Success





← API

API 開箱

← API 介紹

簡介

- 開箱 API
- 開箱 IP 地址

API 介紹

API 介紹 API 介紹 API 介紹 API 介紹 API 介紹 API 介紹 API 介紹 API 介紹 API 介紹 API 介紹

API 介紹 API 介紹 jq 是 JSON 處理器 apt-get install jq 或 brew install jq 安裝

API 介紹

- API 介紹
- APN 介紹
- EPC 介紹
- API 介紹

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 생성

```
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
-H "Content-Type: application/json" \
-d '{
  "name": "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 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": "APN"
}
```

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

5. APN EPC 등록

```
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\": \"APN000000\",
  \"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. IMSI 등록

```
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 "IMSI 등록 ID: $SUBSCRIBER_ID"
```

결과

APN 등록 성공

1. APN (APN) - EPC 등록
2. EPC 등록 (EPC) - EPC 등록
3. APN (APN) - QoS 등록
4. IMSI (IMSI) - IMS 등록

결과

- IMSI 등록 성공 MSISDN
- IMS 등록 성공 IMS
- IMSI 등록 성공 IMSI

- 普通 SIM 専用 SIM

特徴

- MSISDN あり - 電話番号あり
- 電話番号あり - 電話番号あり

普通 IP あり

電話番号あり IP あり

普通 専用“専用” APN あり IPv4 あり IoT あり

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

```
# 1. 创建key
```

```
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
```

```
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",
  "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
```

```
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
```

```
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\ ": \ "0010019999999999\  
  \ "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
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” 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 SIM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

- 方法: 使用 HTTPS
- 端口: 8443

URL: 使用 API 端点 "/api/ims/profile" 以 JSON 格式返回数据

请求体:

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

响应体:

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

命令:

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

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

返回

返回的 JSON 数据

响应:

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

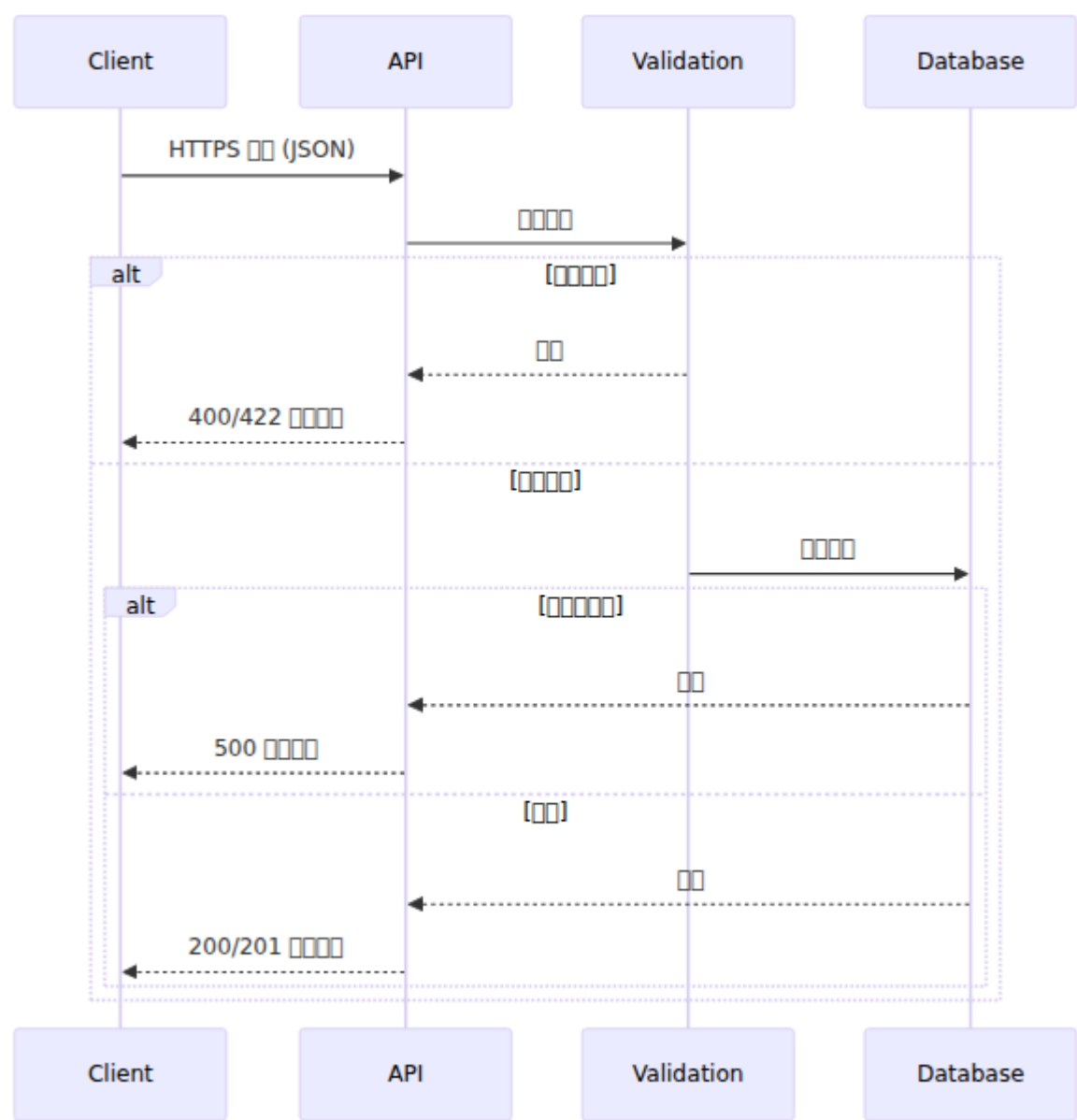
예외:

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

HTTP 코드

코드	상태	설명
200	OK	성공 GET, PUT, DELETE
201	생성	성공 POST
400	잘못된 요청	잘못된 요청
404	찾을 수 없음	잘못된 URL
422	잘못된 요청	잘못된 요청
500	서버 내부 오류	서버 내부 오류

API 流程图



请求

响应

数据库返回数据

请求: GET /api/subscriber

响应:

項目	型別	説明
enabled	boolean	有効/無効
ims_enabled	boolean	IMS 有効/無効

リクエスト:

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

レスポンス:

```
{
  "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"
    }
  ]
}
```

サブスクリバ ID 取得

サブスクリバ ID 一覧取得

リクエスト: GET /api/subscriber/:id

レスポンス:

Field	Type	Description
id	integer	Subscriber ID

Example:

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

Get IMSI

Get IMSI for a subscriber

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

Example:

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

Example:

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

Response: IMSI 001001123456789

Get MSISDN

Get MSISDN for a subscriber

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

Example:

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

リクエスト:

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

レスポンス: 成功した場合に返されるデータ

レスポンス

レスポンス

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": "テスト"
    }
  }
}
```

レスポンス:

- imsi - 14-15 桁の IMSI

- `key_set_id` - 00000000 000
- `epc_profile_id` - 00000000 EPC 0000

0000:

- `enabled` - 00: true
- `ims_enabled` - 00: true
- `sim_id` - 00 SIM 0
- `ims_profile_id` - 00 IMS 0000 IMS 00000
- `roaming_profile_id` - 00 0000000000000000
- `msisdns` - MSISDN ID 00000000
- `static_ips` - 00 IP ID 00000 APN 00
- `custom_attributes` - 000000

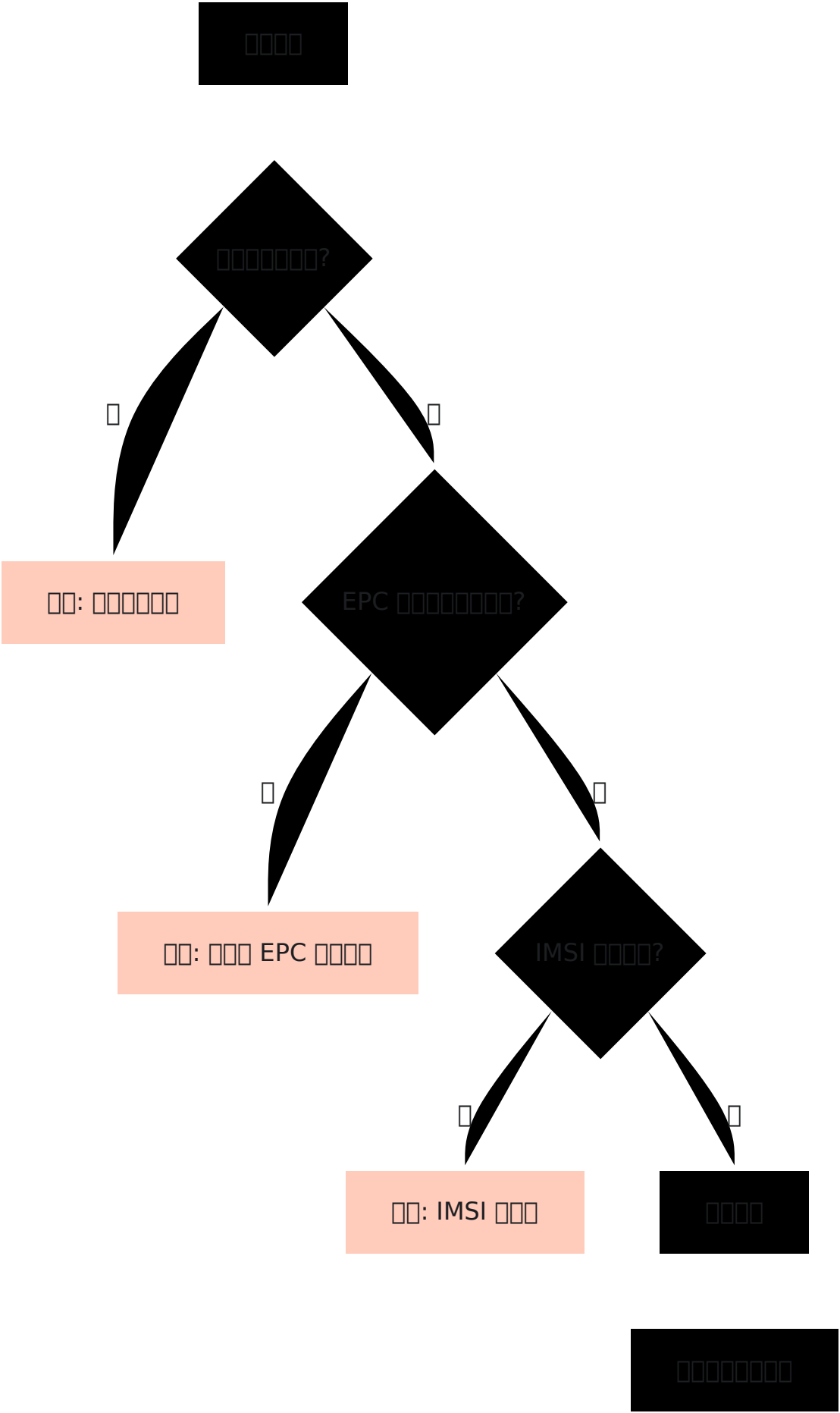
00:

- 00000000 - 000000
- 0 MSISDN 00 - 0000000000
- 00 IP 00 - 000 IP 000 APN

0000:

```
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
  }
}'
```

0000:



20000

20000000

PUT /api/subscriber/:id

20000:

id	integer	20000 ID
id	integer	20000 ID

20000:

```
{
  "subscriber": {
    "enabled": false,
    "ims_enabled": false,
    "epc_profile_id": 2,
    "custom_attributes": {
      "note": "20000"
    }
  }
}
```

20000:

- enabled - 200/20000000
- ims_enabled - 200/200 IMS 200
- sim_id - 200 SIM 200
- key_set_id - 200 20000000000
- epc_profile_id - 200 2000000000
- ims_profile_id - 200 2000000000
- roaming_profile_id - 200 20000

- `msisdns` - 電話番号 0000
- `static_ips` - 静的 APN 0 00 IP
- `custom_attributes` - カスタム属性

レスポンス:

- `imsi` - IMSI 番号

エラー:

- 0000 - エラーコード

リクエスト:

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

レスポンス:

- 0000000: {"enabled": false}
- 0000000: {"ims_enabled": false}
- 00000000: {"epc_profile_id": 2} (EPC 0000)
- 000000: {"roaming_profile_id": 3} (0000)

リクエスト

レスポンス

リクエスト: DELETE /api/subscriber/:id

レスポンス:

項目	型	説明
id	integer	サブスクリプション ID

リクエスト:

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

レスポンス: 成功した場合、PDN 接続と IMSI

レスポンス: 成功した場合

- **id** - サブスクリプション ID
- **SIM** - SIM カード
- **pdn** - PDN 接続
- **MSISDNs** - MSISDNs

リクエスト:

リクエスト: CLR 接続と MME 接続

レスポンス: POST /api/subscriber/cancel_location

レスポンス:

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

レスポンス:

項目	型	説明	項目
imsi	string	IMSI	IMSI 14-15 桁

レスポンス:


```
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"
}
```

注:

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

注:

- 成功: 成功
- 成功: 成功
- 成功: 成功 MME へ通知
- 成功: 成功

- □□ : □□□□□□□□

IMSI

□□□ CLR □□□ MSISDN □□□□

1. MSISDN IMSI:

```
// 0000 IMSI 001001123456789 0 MSISDNs ["+1234567890",
"+9876543210"]
POST /api/subscriber/cancel_location
{"imsi": "001001123456789"}

// 00: 0000 CLR000 MSISDN 0000000000
```

2. IMSIs

```
// 00000000 MSISDN 0000 IMSI 00000000
// 00 A: IMSI 001001111111111, MSISDN "+1234567890"
// 00 B: IMSI 001001222222222, MSISDN "+1234567890"

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

// 00: 000 A 0000000 B 0000
```

□□□□ ■

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

□□□□

- 000000000000
- IMSI 00
- S6a 0000

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:

- 最多 1-15 个
- 最多 15 个
- 最多 E.164 号码 + 国家代码

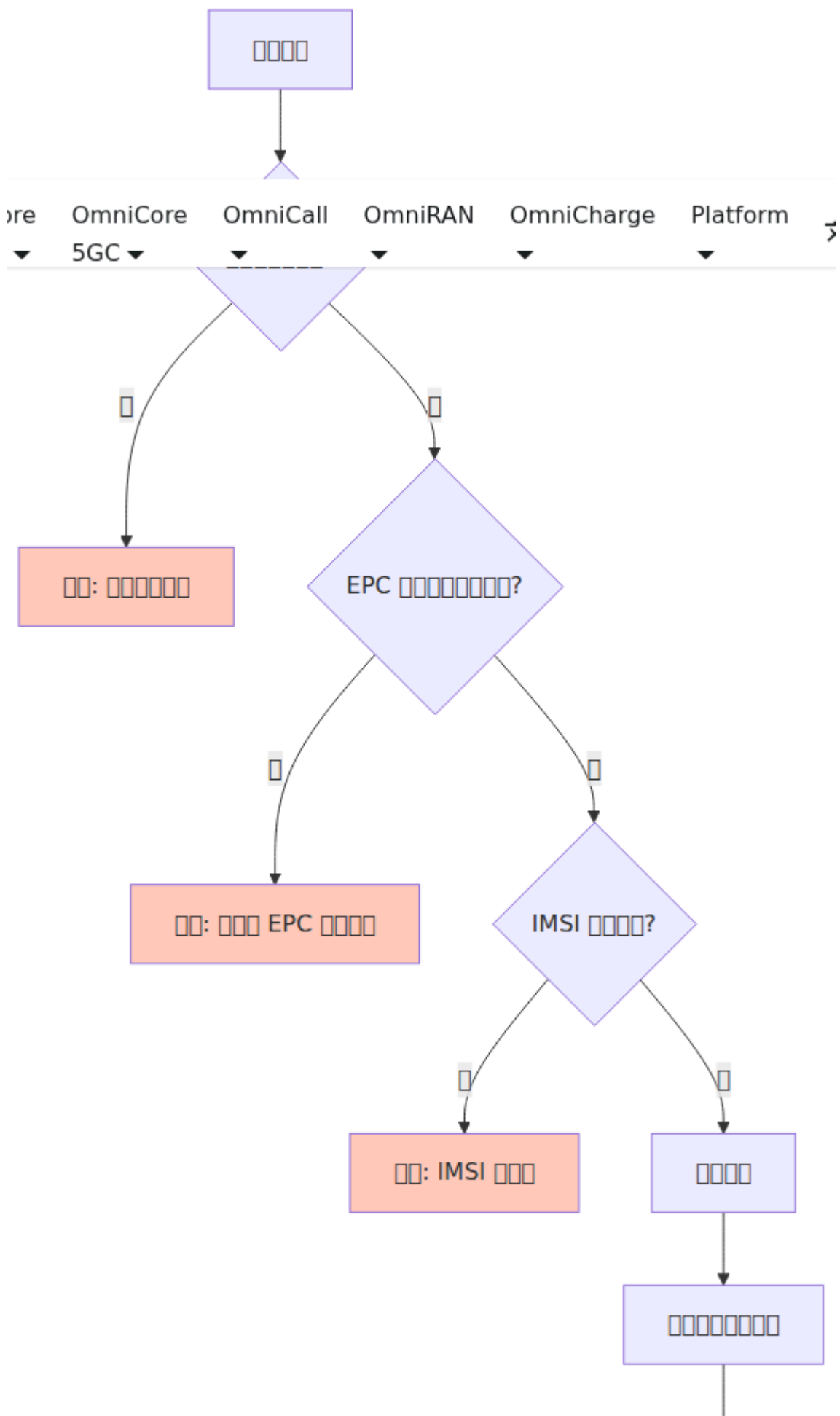
示例:

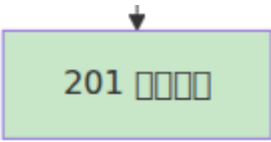
```
curl -k -X POST https://hss.example.com:8443/api/msisdn \
-H "Content-Type: application/json" \
-d '{
  "msisdn": {
    "msisdn": "14155551234"
  }
}'
```

MSISDN 号码

MSISDN 号码是用于标识移动设备的唯一标识符。

MSISDN 格式:





API 使用 **MSISDN** 或 **IMSI** 删除 SIM 卡

删除 MSISDN

删除 SIM 卡

请求: `DELETE /api/msisdn/:id`

响应:

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

SIM 卡

SIM 卡管理 API 支持通过 ICCID 或 PIN/PUK 删除 SIM 卡。OTA 删除 SIM 卡需要 **MSISDN**。

请求:

- IMSI** - 删除 SIM 卡

获取 SIMs

获取 SIM 卡

请求: `GET /api/sim`

响应:

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

GET SIM

Retrieve SIM information

Request: GET /api/sim/:id

Response:

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

POST SIM

Create new SIM card

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 number

Additional info:

- `sim_vendor` - 运营商
- `batch_name` - 批次
- `is_esim` - eSIM 是否
- `pin1`, `pin2` - 运营商 PIN 码
- `puk1`, `puk2` - PIN 解锁码
- `adm1-adm10` - 运营商
- `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"
  }
}'
```

SIM

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": "SIM卡"
  }
}'
```


API SIM

API SIM API

API: DELETE /api/sim/:id

API: API API API API API SIM

API API

API API Milenage API API API API Ki OPC/OP AMF SQN API API API API

API:

- API - 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 API API API

API: GET /api/key_set/:id

API:

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

响应:

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

请求头

请求体

方法: POST /api/key_set

响应:

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

响应:

- `ki` - 128 位 32 字节

□□□□□

□□□□□□□□

□□: 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
}
```

□□:

필드	타입	단위	범위
name	문자열	문자	문자열
ue_ambr_dl_kbps	정수	Kbps	10000-1000000
ue_ambr_ul_kbps	정수	Kbps	5000-500000
network_access_mode	정수	문자	"제한" 또는 "제한 없음"
tracking_area_update_interval_seconds	TAU 간격 문자	문자	600문자
apn_profiles	APN 문자 ID 문자	문자	[] 또는 [1, 2, 3]

예제:

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

참고:

- 제한 - 제한된 대역폭
- 제한 없음 - 제한 없이 EPC 대역폭

GET EPC profile

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

Response: EPC profile information is returned in JSON format.

DELETE EPC profile

Request: `DELETE /api/epc/profile/:id`

Response: Status 200 OK is returned.

IMS profile

IMS IP address is returned in JSON format. IFC profile information is also returned in JSON format.

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 profile XML template Liquid template

IFC profile:

IFC profile Liquid template

変数	変数名	値
<code>{{ imsi }}</code>	IMSI	001001123456789
<code>{{ msisdns }}</code>	MSISDN 番号リスト	["14155551234", "14155555678"]
<code>{{ mcc }}</code>	国番号	001
<code>{{ mnc }}</code>	ネットワーク番号	001

動作概要:

IFC 側から **Liquid** 側で Jinja2 による IMS 番号のループ

- ループ: IMS 番号ループを Jinja2 の `{{ imsi }}` と `{% for msisdn in msisdns %}`
- ループ: API 側で XML 生成
- ループ: IMS 側 MAA/SAA/HSS
 - IMS 側
 - ループループ
 - `{{ imsi }}` → IMSI
 - `{{ msisdns }}` → MSISDN
 - `{{ mcc }}` → 国番号
 - `{{ mnc }}` → ネットワーク番号
 - Cx/Diameter 側 XML 生成 S-CSCF

動作:

```
<!-- 電話番号 -->
{{ imsi }}

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

<!-- 携帯キャリア -->
{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc }}.3gppnetwork.org
```

IFC 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 (Python):

```
import requests

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

レスポンス (201 OK):

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

注意:

- API は IFC を XML で受け取る
- 送信する XML は正しい形式でなければならない
- name は一意である必要がある

API:

- **API** - IFC API
- **API** - IMS API
- **API IFC** - 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 APN 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
- **"ipv4_or_ipv6"** - IPv4 or IPv6

📡 APN QoS 📡📡

📡: GET /api/apn/qos_profile

📡 APN QoS 📡📡

📡: POST /api/apn/qos_profile

📡📡:

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

📡 APN 📡📡

📡: GET /api/apn/profile

📡 APN 📡📡

📡: POST /api/apn/profile

📡📡:

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

📡📡📡:

- apn_identifier_id - 📡📡📡📡📡 APN 📡📡
- apn_qos_profile_id - 📡📡📡📡📡 APN QoS 📡📡

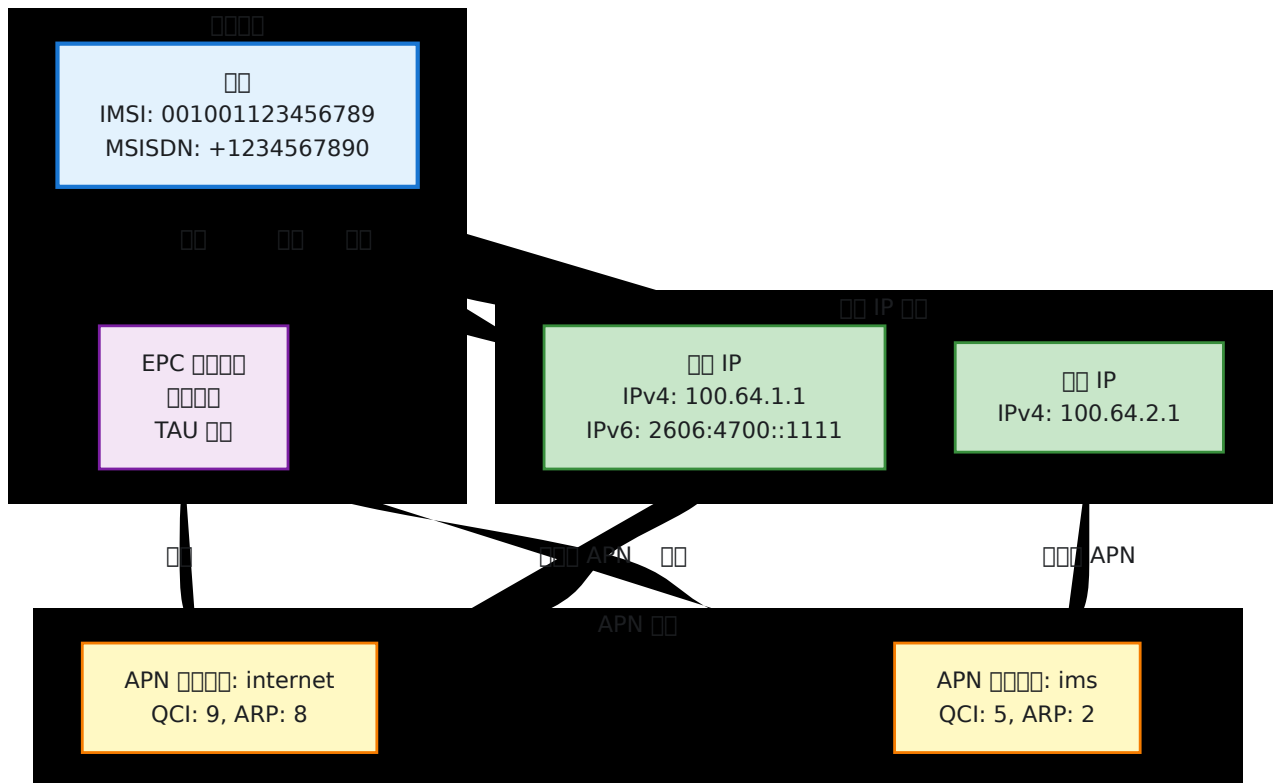
11

- **Orange** - APN **Orange**
- **EPC** **Orange** - APN **Orange** EPC **Orange**

IP

☐ IP ☐ APN ☐ APN ☐ IPv4 ☐ IPv6 ☐ DHCP

11



□□□□□□□□□□

11

1. **APN identifier**: APN identifier 012...
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**: `apn_qos_profile`
5. **AMBR**: 1000 kbps → bps
6. **IP**: `subscriber.static_ips` → `apn_profile_id` → IP
 - `subscriber.static_ips` → `apn_profile_id` → IP
 - IP `apn_identifier.ip_version`
7. **VPLMN**-**APN**: 0 - IP

Diagram:

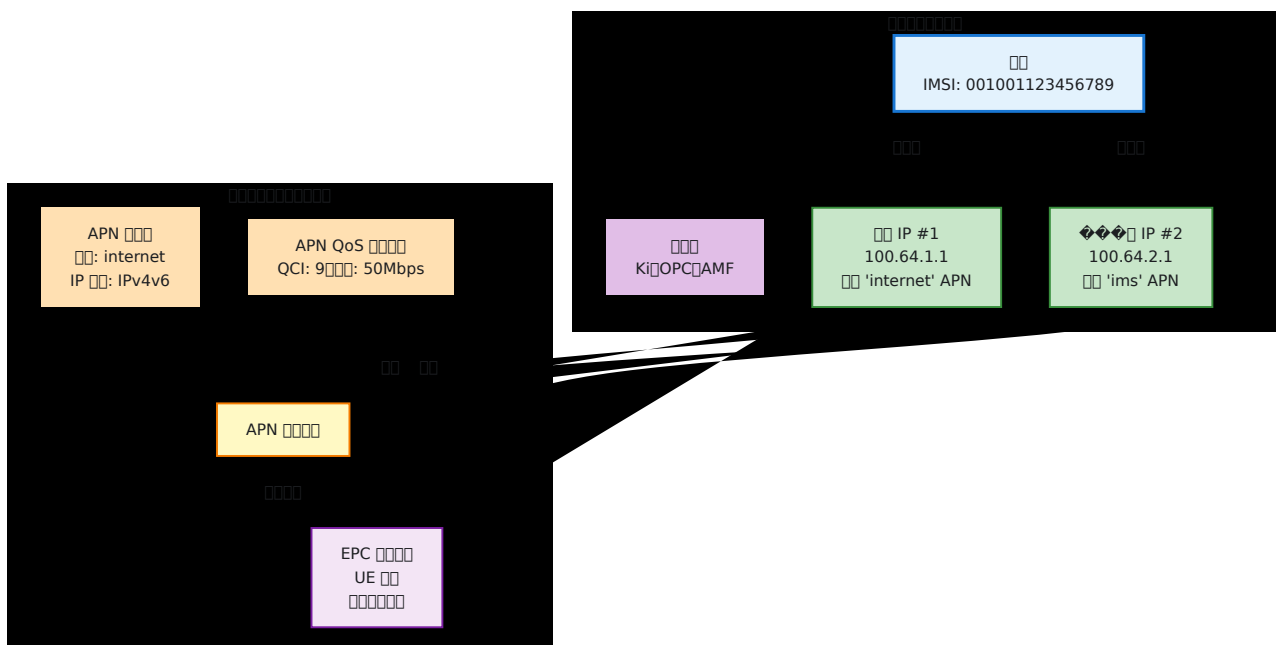


Diagram:

- **APN**: IP address **APN**
- **APN** **IP**: APN IP address
- **IPv4** **IPv6**: IP address IPv4 IPv6
- **IP**: IP address IP address IP address
 - IP IPv4 IPv6 IP address APN
 - IP address IP address
 - IP address `ipv4_static_ip` `ipv6_static_ip` IP address
- **IP**: IP address IP address

📌:

- IoT 設備が IP へ
- 静的 IP アドレスを割り当てる
- 静的 IP アドレスを割り当てる
- 静的 IP アドレスを割り当てる
- 静的 IP アドレスを割り当てる

静的 IPs

静的 IP アドレス

📌: 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
```

靜態 IP

APN 靜態 IP 設定

URL: `POST /api/epc/static_ip`

Body:

```
{
  "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` 或 `ipv6_static_ip`

Options:

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

IP Address:

- IPv4: `100.64.1.1`
- IPv6: `2606:4700:4700::1111`
- IPv4 或 IPv6 IP address
◦ IP address
- IP address APN
- IP address

Response:

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

例:

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"
  }
}'
```

応答 (201 OK):

```
{
  "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"
  }
}
```

注:

- IP アドレス - 一意に決まります
- APN アドレス - 1つのAPNに1つ

更新 IP

更新する IP アドレス

リクエスト: PUT /api/epc/static_ip/:id

応答:

???		
id	integer	IP ID

{}:

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

{}:

- ipv4_static_ip - IPv4
- ipv6_static_ip - IPv6
- apn_profile_id - APN

{}:

- id -

: IP PDN PDN IP

{}:

```
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"
  }
}'
```

IP

IP

요청: DELETE /api/epc/static_ip/:id

응답:

필드	타입	설명
id	integer	정적 IP 주소 ID

예시:

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

참고:

- 정적 IP 주소
- 정적 APN (APN) 주소
- 정적 IP 주소 (???)
- 정적 IP 주소

참고: 정적 IP 주소는 PDN 주소인 정적 IP 주소 (???)

정적 IP 주소

정적 IP 주소는 정적 IP 주소 (???)

참고:

- 정적 IP 주소 (???)
- 정적 static_ips 주소

정적 IP 주소:

```
# 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

```
# Create "internet" APN and assign 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')

# Create "ims" APN and assign 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')

# Assign IP addresses
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]
  }
}'
```

Results:

- ✓ : Successfully created APNs and assigned IP
- ✗ : Failed to create APN and assign IP

Summary - **APN:**


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

# 応答:
{
  "errors": {
    "static_ips": [
      "静的 IP アドレス 100.64.1.1 は
internet に 100.64.1.2 は internet"
    ]
  }
}
```

エラー:

- 静的 IP - 静的 IP
- 静的 IP - 静的 IP
- 静的 IP 静的 IP - 静的 IP

静的 IP

静的 IP アドレスは IMS 静的 IP アドレス MCC/MNC 静的 IP アドレス

静的 IP アドレス

静的 IP: GET /api/roaming/profile

静的 IP アドレス

静的 IP: 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": []
  }
}
```

响应:

- "allow" - 允许
- "deny" - 拒绝

请求:

- data_action_if_no_rules_match - 数据 规则 匹配
- ims_action_if_no_rules_match - IMS 规则 匹配

请求:

请求: GET /api/roaming/rule

请求:

请求: POST /api/roaming/rule

响应:

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

필드:

- `mcc` - 3자리 숫자
- `mnc` - 2-3자리 숫자
- `data_action` - "allow" 또는 "deny" 값
- `ims_action` - "allow" 또는 "deny" IMS/VoLTE

참고:

- `allow` - 허용
- `deny` - Diameter 프로토콜

EIR

OmniHSS는 S13 Diameter 프로토콜을 사용하여 EIR에서 IMEI를 조회합니다.

OmniHSS는 S13 프로토콜을 사용하여 EIR에서 IMEI를 조회합니다. **EIR**은  

GET EIR

요청: GET /api/eir/rule

POST EIR

요청: 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: 无操作, 1: 删除设备, 2: 标记设备

配置:

- 0 - 无操作
- 1 - 删除设备
- 2 - 标记设备

参数:

- IMEIs - IMEI 列表
- TAC - TAC 列表
- IMEI_TAC - IMEI 和 TAC 列表

参数:

- S13 - S13 EIR 列表
- OmniHSS - OmniHSS EIR 列表

配置

配置

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

← **API** | **API**: **API** →

API 入門

← API 入門

API

API 入門

GET /api/status

API

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

API

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

API 入門

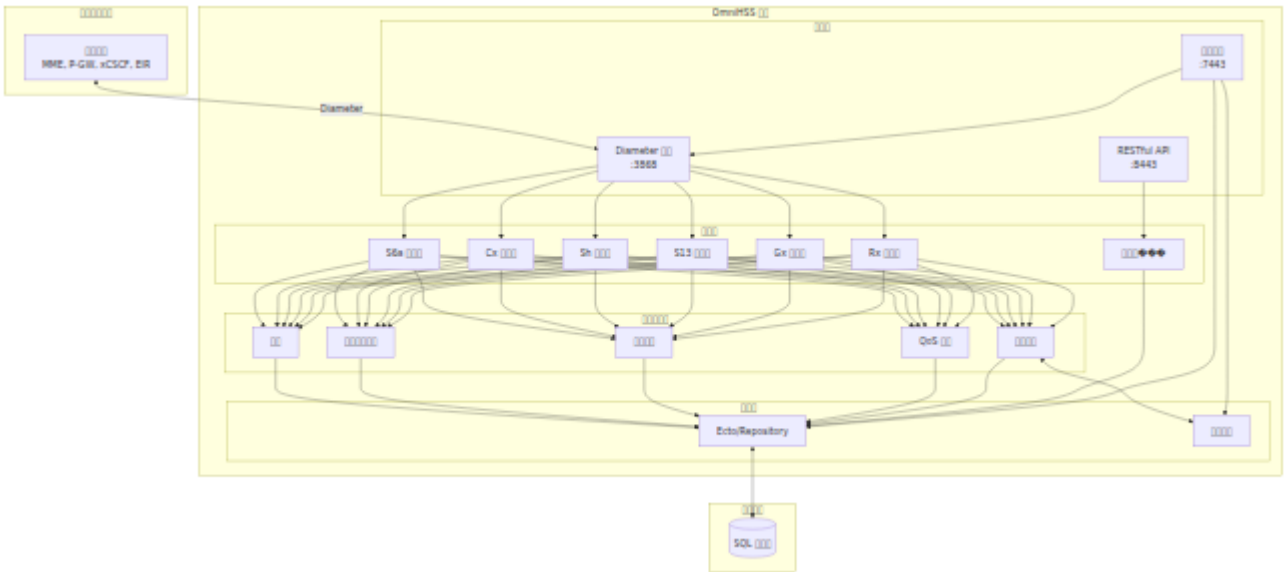
← API 入門

OmniHSS

←

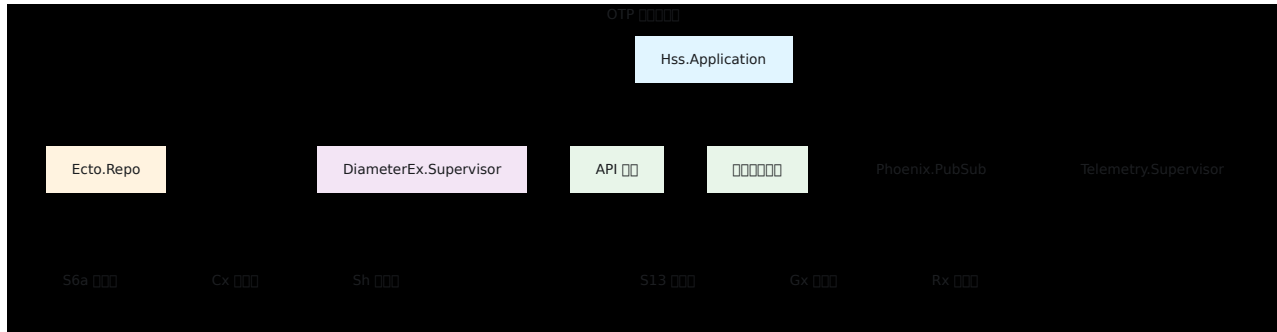
-
-
- Diameter
-
-
-
-

OmniHSS is an Elixir / Erlang/OTP based implementation of a 4G/LTE HSS (Home Subscriber System) and MME (Mobility Management Entity).



□□□□

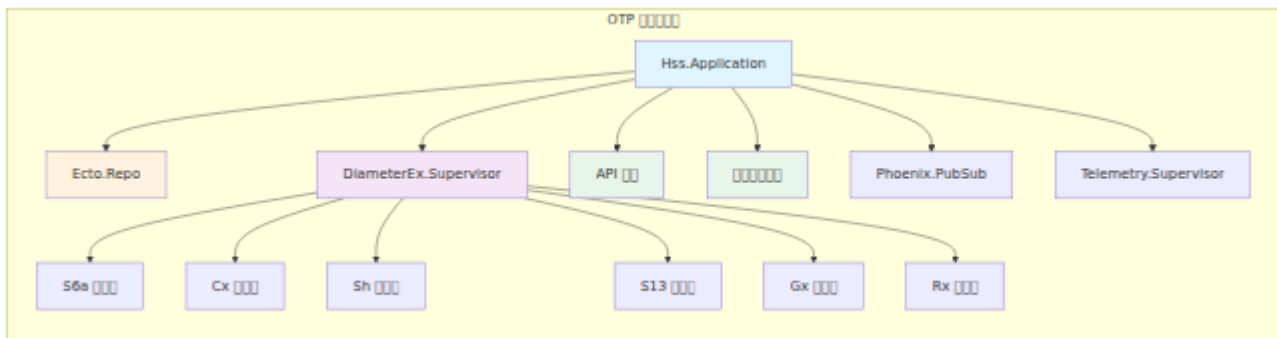
□□□□



Diameter □□□□

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

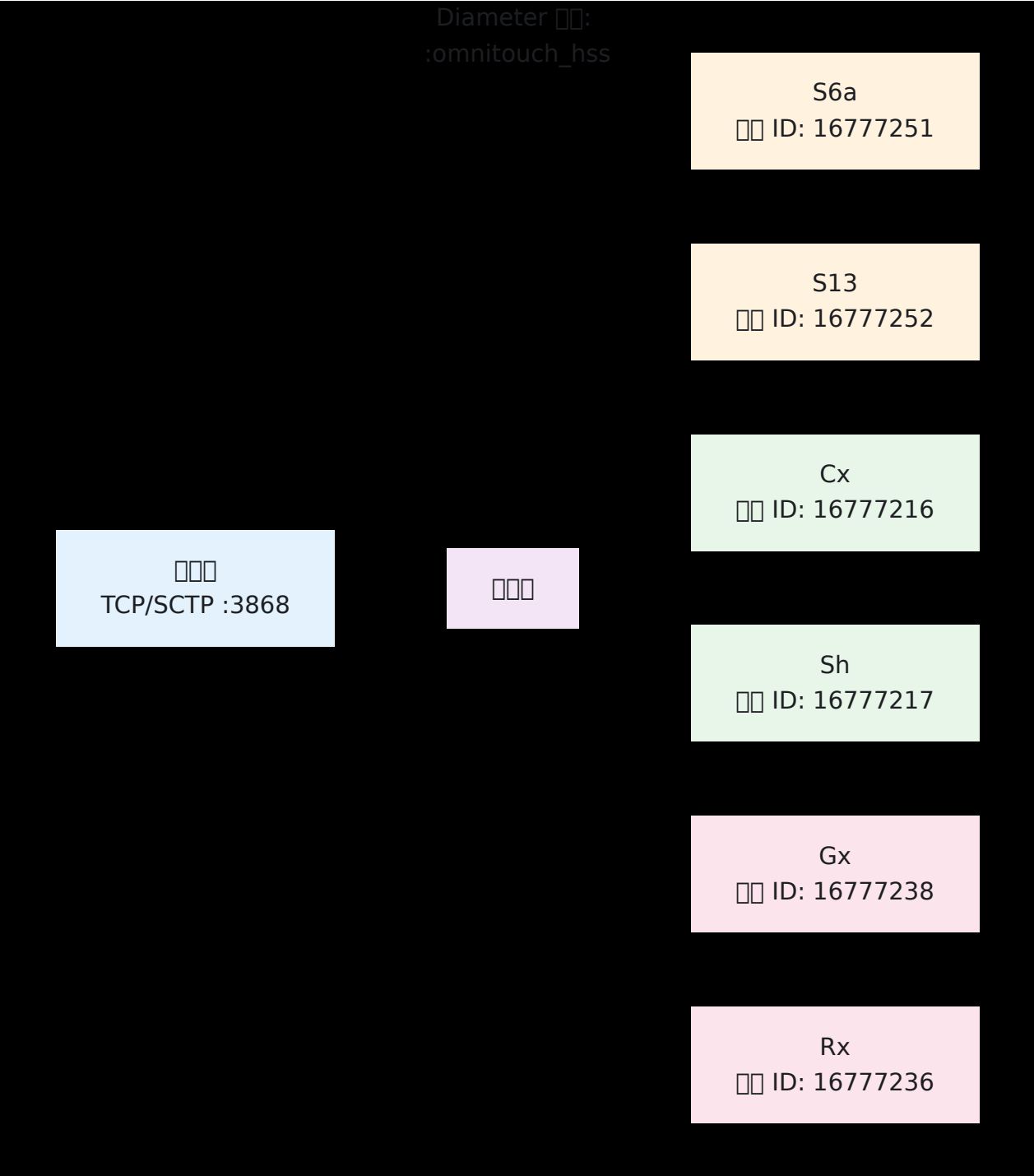
1. □ **DiameterEx** □□ - □□□□□ Diameter □□ ID
2. □□□□ - □□ AVP□□□□□□□□□□
3. □□□□□□ - □□□□□□□□□□□□
4. □□□□ - □□ AVP □□ Diameter □□□□
5. □□□□ - □□□□□□ Diameter □□□□



Diameter 网络

Diameter 网络

OmniHSS 网络部署 Diameter 网络



□□□□□□



□□□□□□

Configured

□□□□

Connecting

□□□□

Connected

□□□□

□□□□□□

□□□□□□

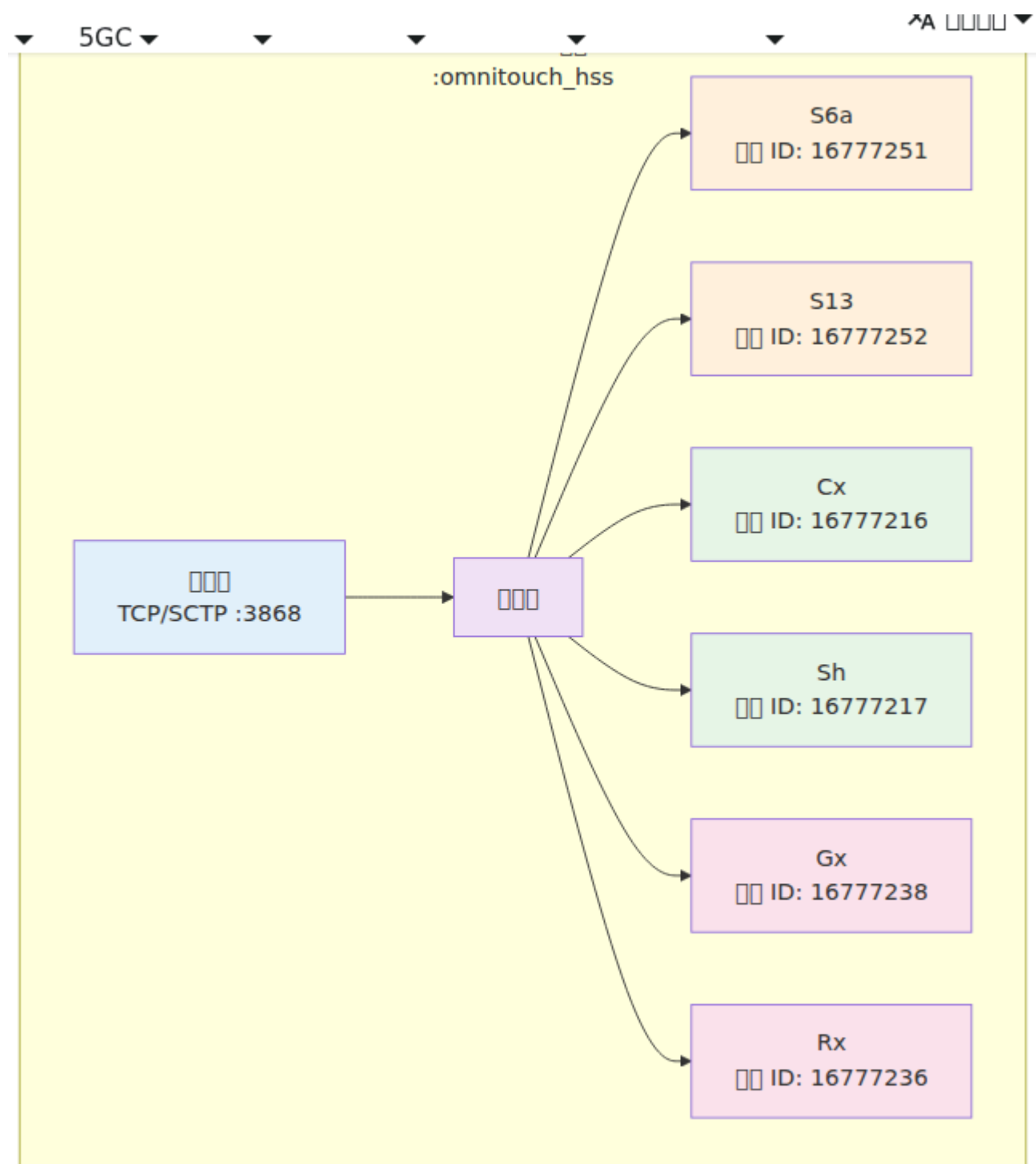
□□□□

□□□□ Diameter □□

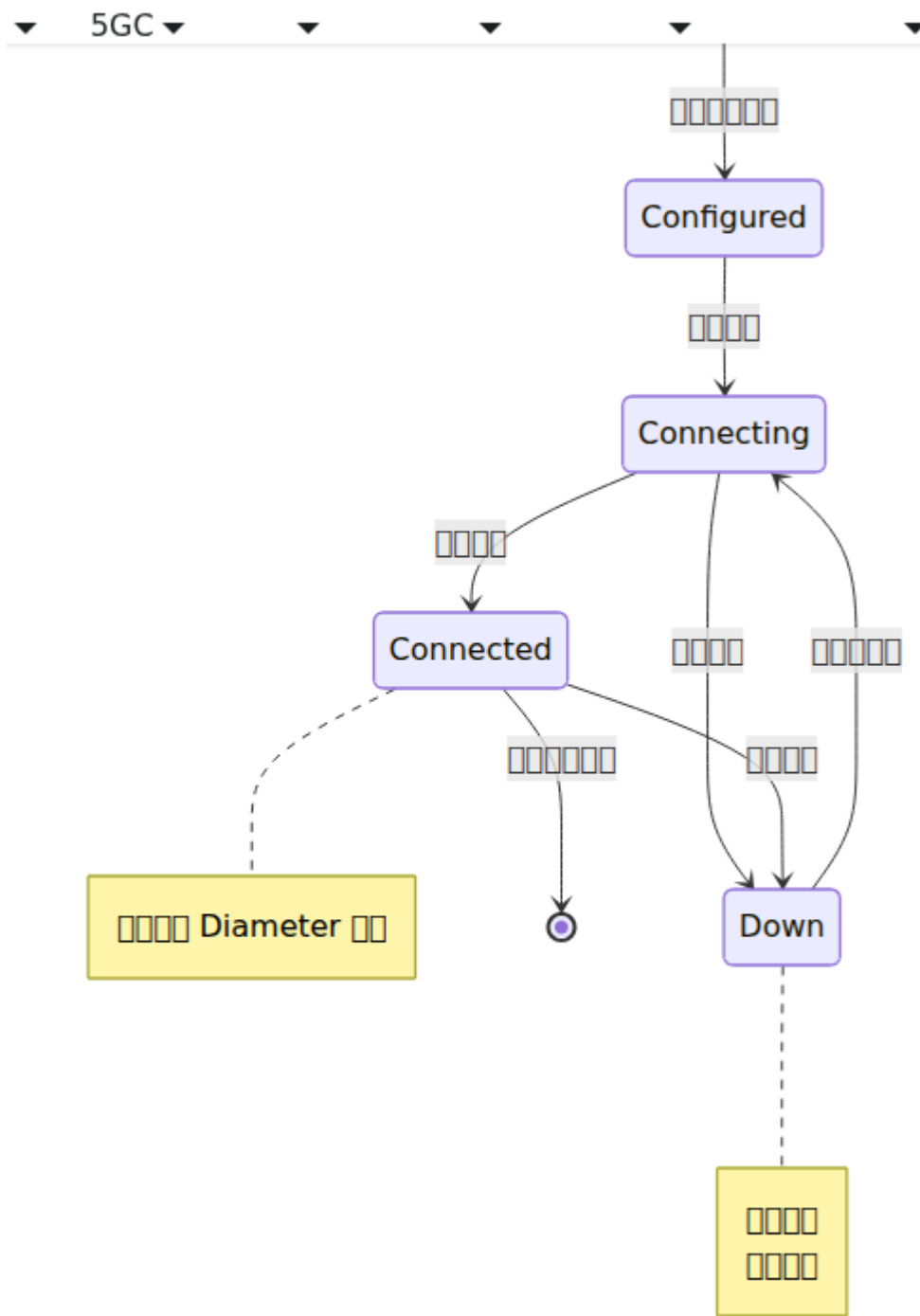


Down

□□□□
□□□□

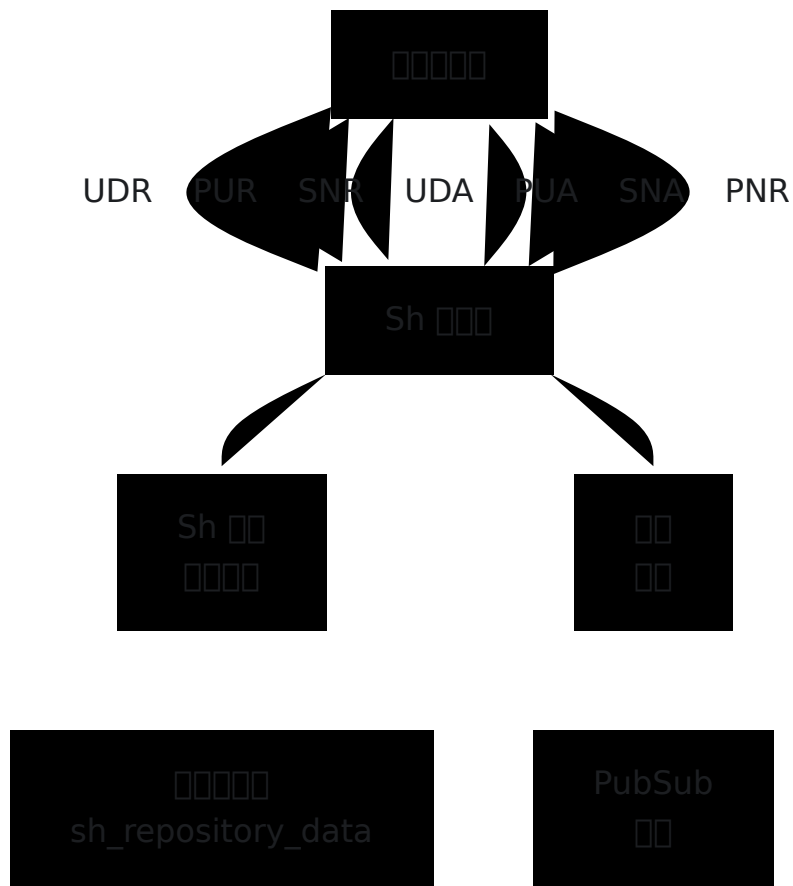
Diameter 

IMS 000000



Sh (IMS)

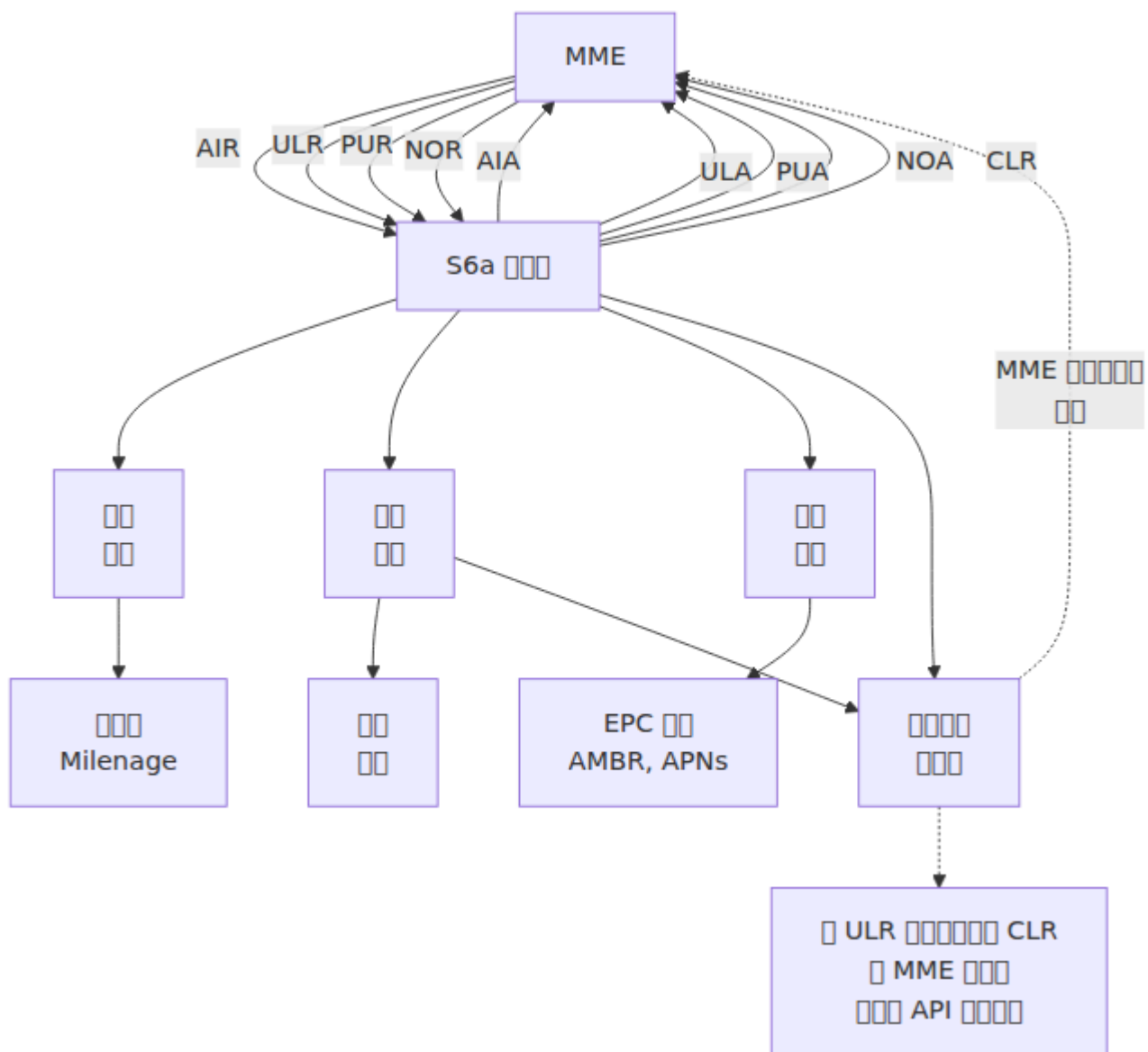
IMS



Gx ()

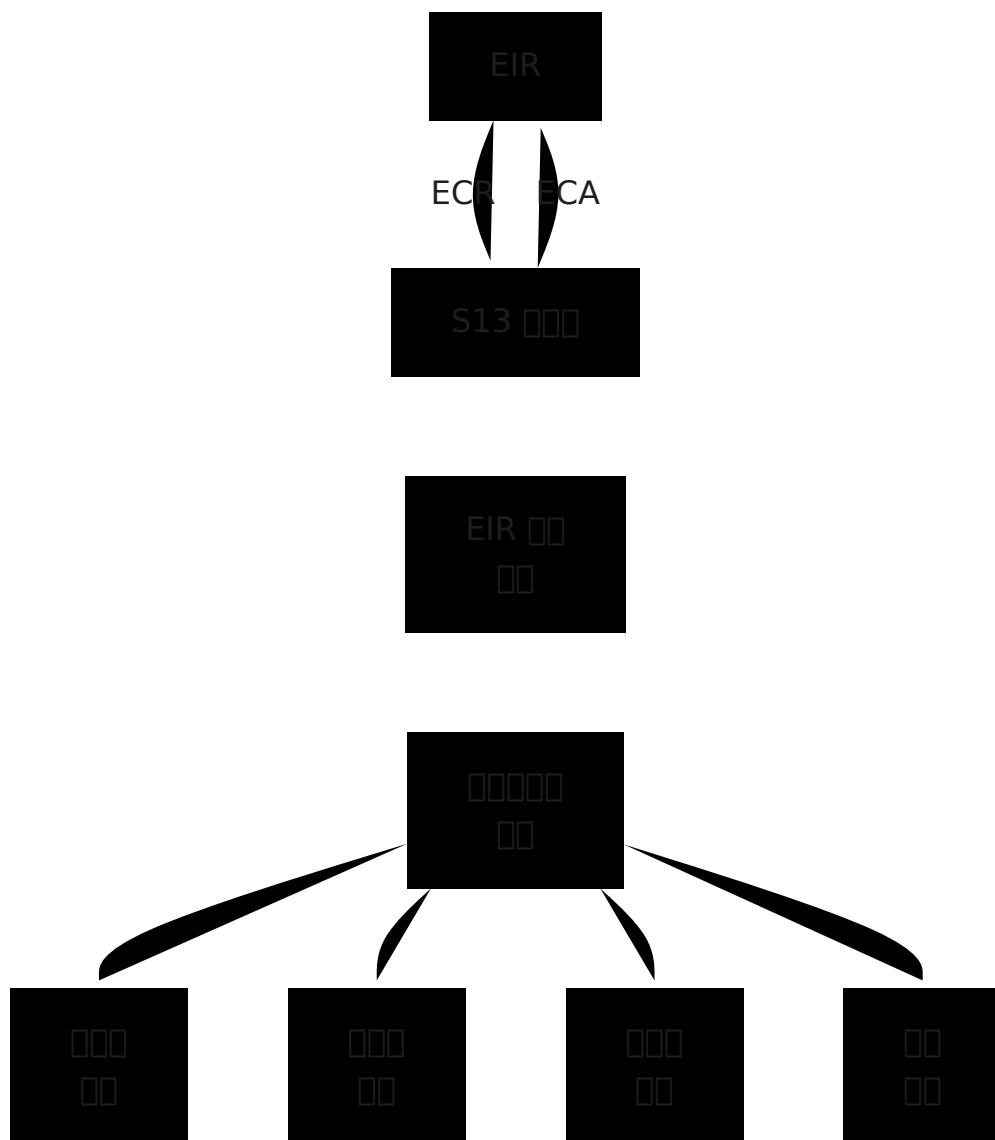
PCRF

IMS VoLTE PCRF



S13 (EIR)

IMEI EIR



□□□

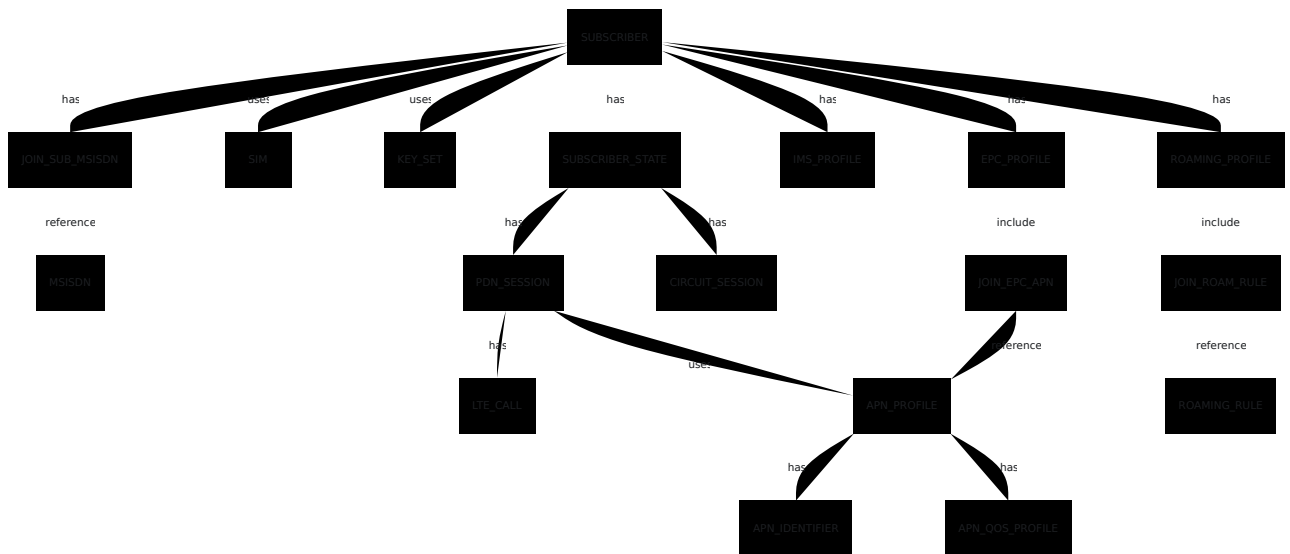
□□□□□

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

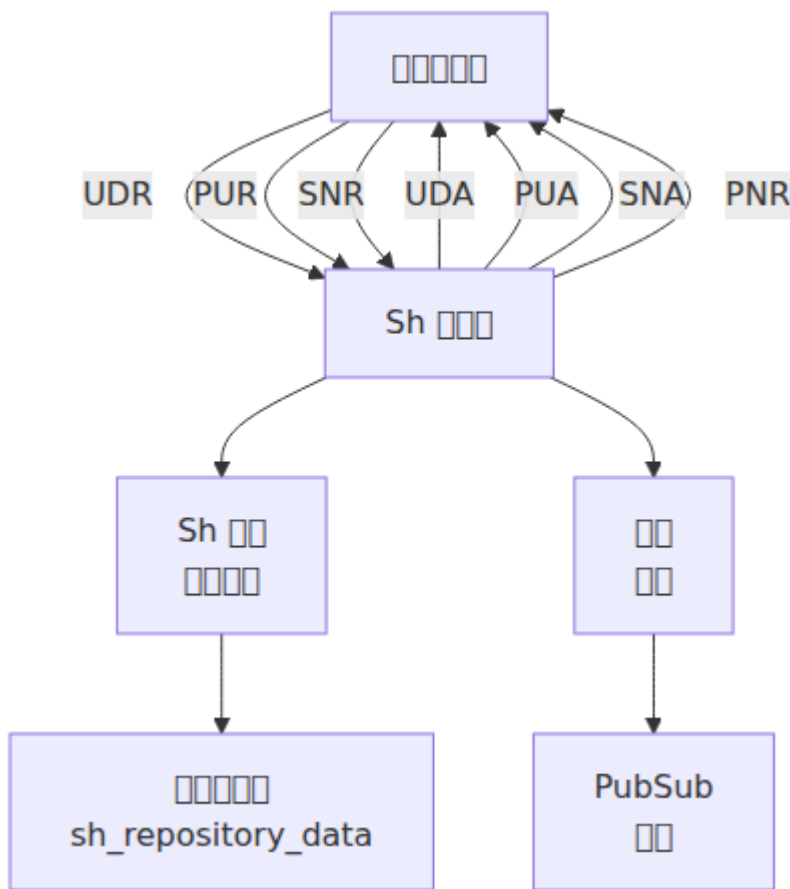
□□□□□□□□□□□□□□□□□□ **ONS** □□□□□□□□□□□□□□□□□□□□

□□ Galera □□□□□□□□□□ **Galera** □□□□□□□

□□□□□□□□

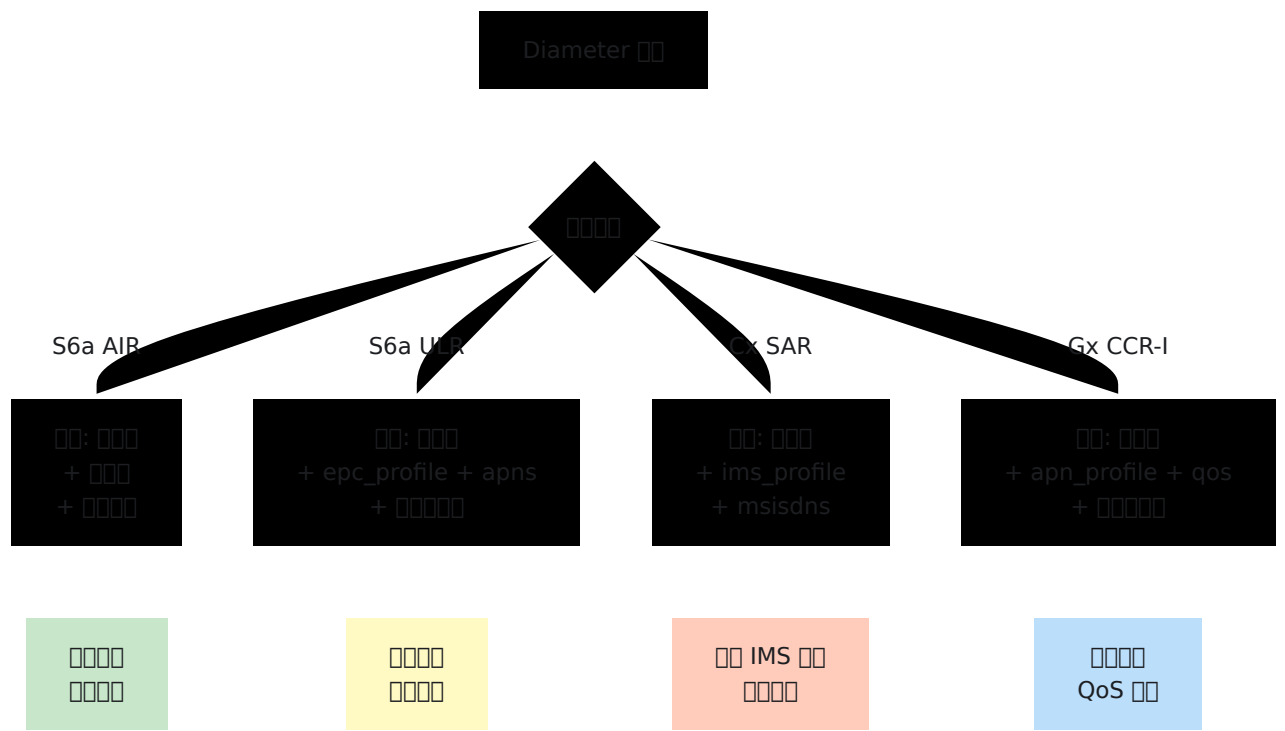


Ecto □□□□



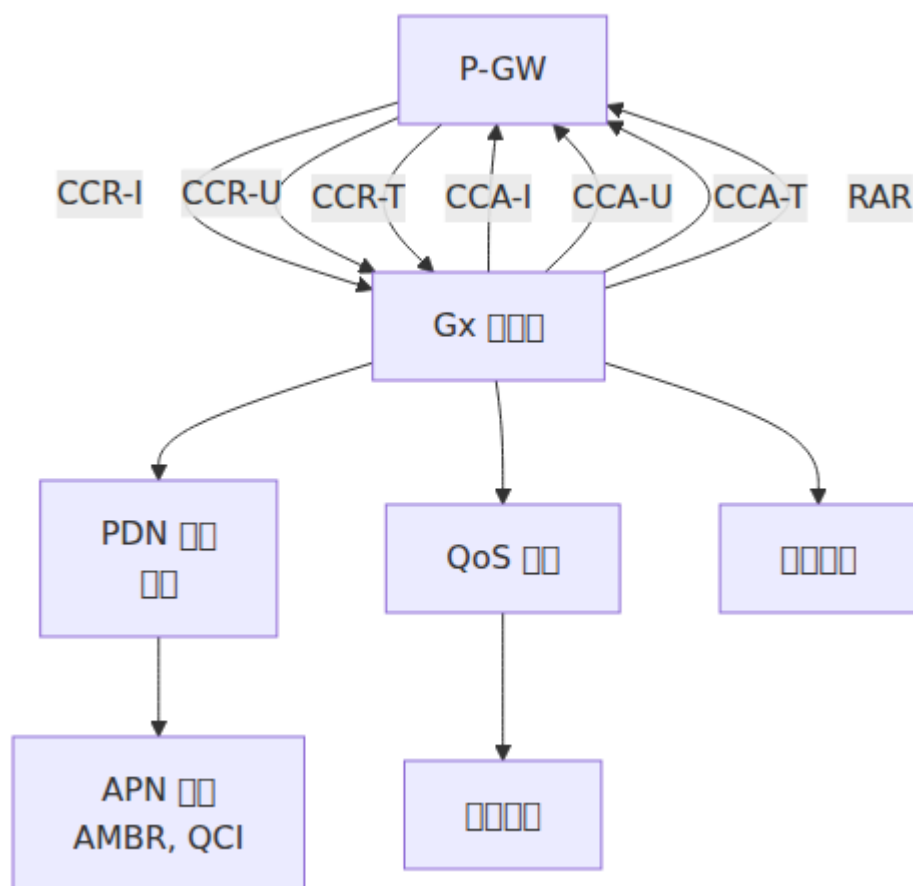
网络架构

网络 Diameter 网络架构

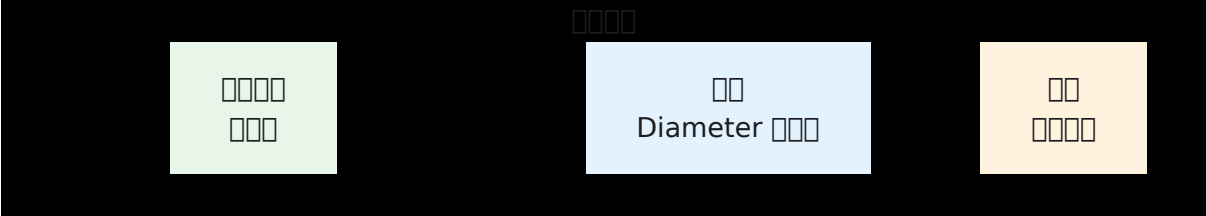
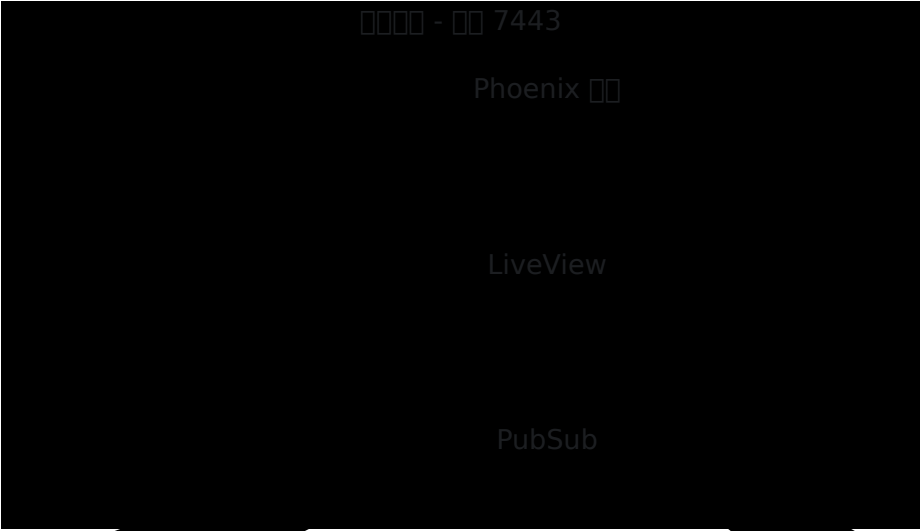


□□□□

API □□

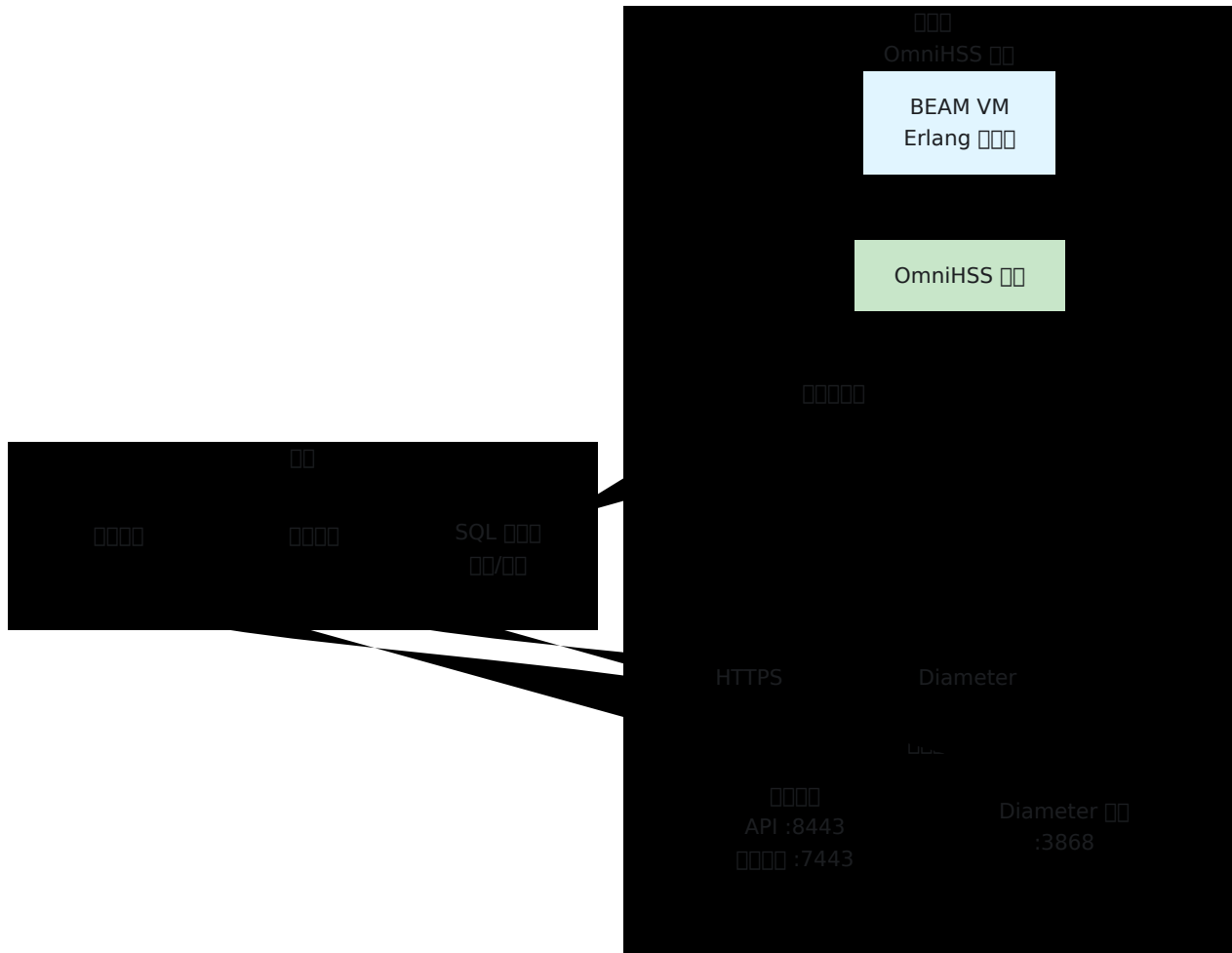


Phoenix 7443



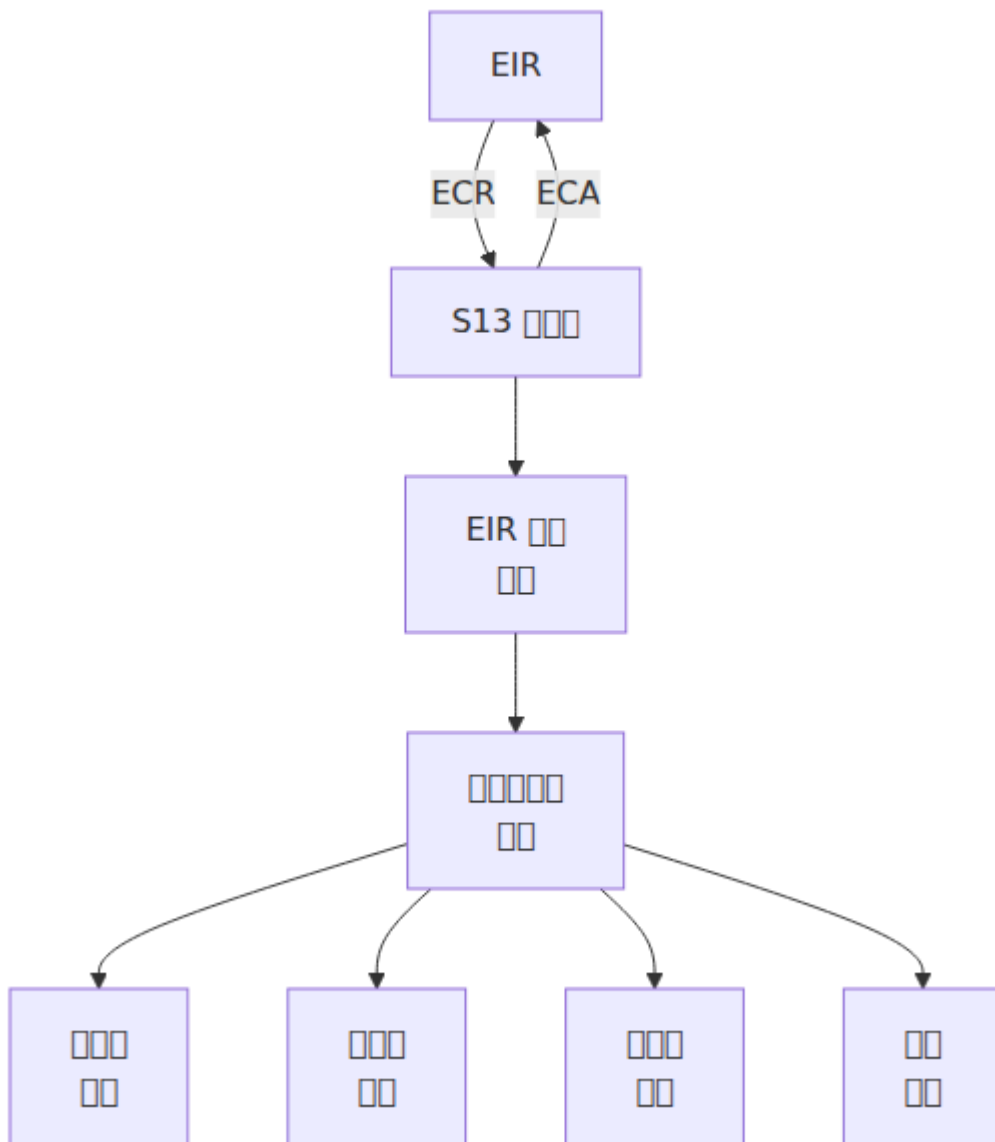
□□□□

□□□□□



数据库集群 (Galera 应用)

数据库集群OmniHSS 应用 MariaDB Galera 应用



□□□□

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

□□□□□□□□□□□□ **Galera** □□□□□

□□□□□□□□

□□□□□□□□□□□□□□



Sequence Diagram

1. Overview

- Erlang/OTP framework
- Diameter protocol
- Authentication and Roaming

2. Components

- Diameter protocol
- Authentication and Roaming
- Authentication and Roaming

3. Flow

- Diameter protocol
- Authentication and Roaming
- Authentication and Roaming

4. 圖

- 圖
- 圖
- 圖

5. 圖

- 圖
- 圖
- Diameter 圖
- 圖

OmniHSS 架构图

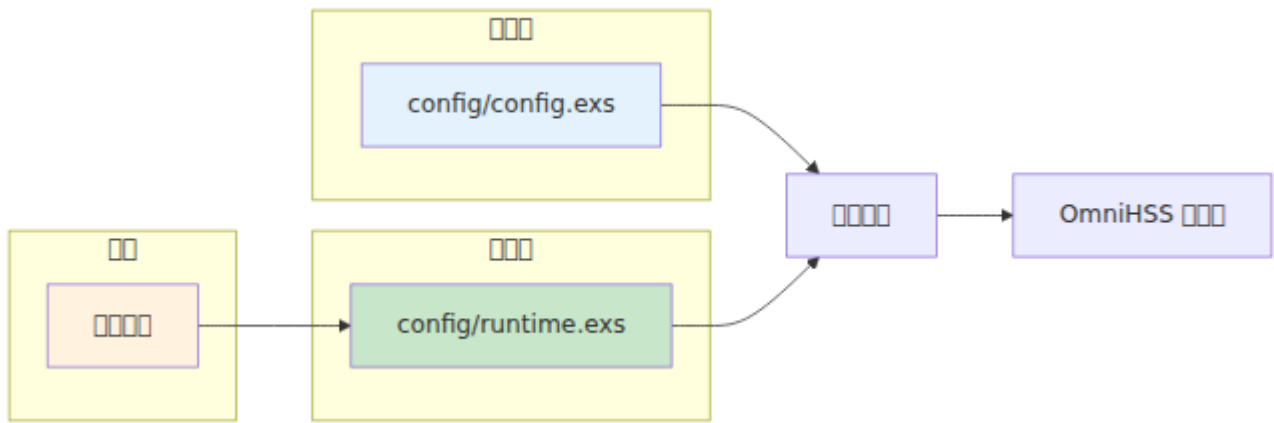
← 架构图

图例

- 数据库
- 网络接口
- 接口
- 接口
- Diameter 接口
- 接口
 - PLMN 接口
 - HSS 接口
- IMS 接口
- EIR 接口
- API 接口
- 接口

架构图

OmniHSS 架构图



config/config.exs (配置)

配置文件中包含以下配置项：

- 数据库配置
- API 配置
- 日志配置

config/runtime.exs (运行时)

配置文件中包含以下配置项：

- 数据库配置
- Diameter 配置
- 网络 PLMN 配置
- IMS S-CSCF 配置
- 其他配置

部署与运行

部署与运行 HSS 模块的步骤如下：

```
# config/runtime.exs

config :license_client,
  # 许可证 API 端点地址
  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"
```

许可证配置

配置项	数据类型	是否必填	默认值
license_server_api_urls	许可证 API 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"

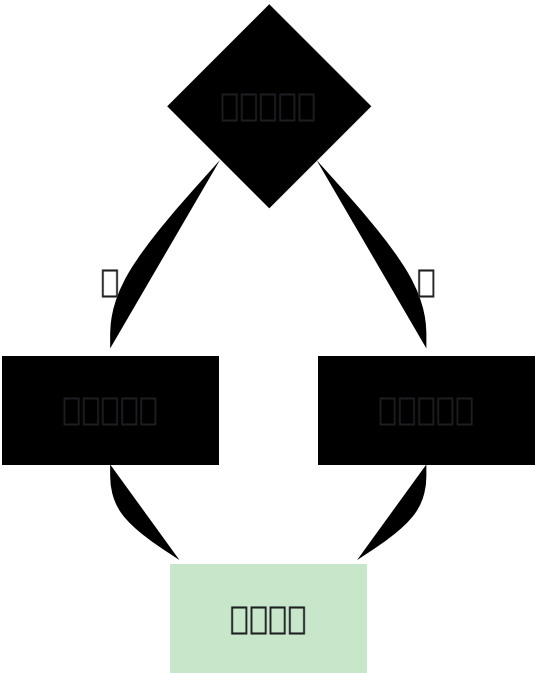
许可证

- 许可证持有者 HSS 名称
- 许可证持有者 HTTPS 端点地址
- 许可证 URL 列表
- 许可证持有者名称

□□□□□

□□□□□

□□□□



□□□□□□

OmniHSS □□□□□□□□

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

環境構築

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")),

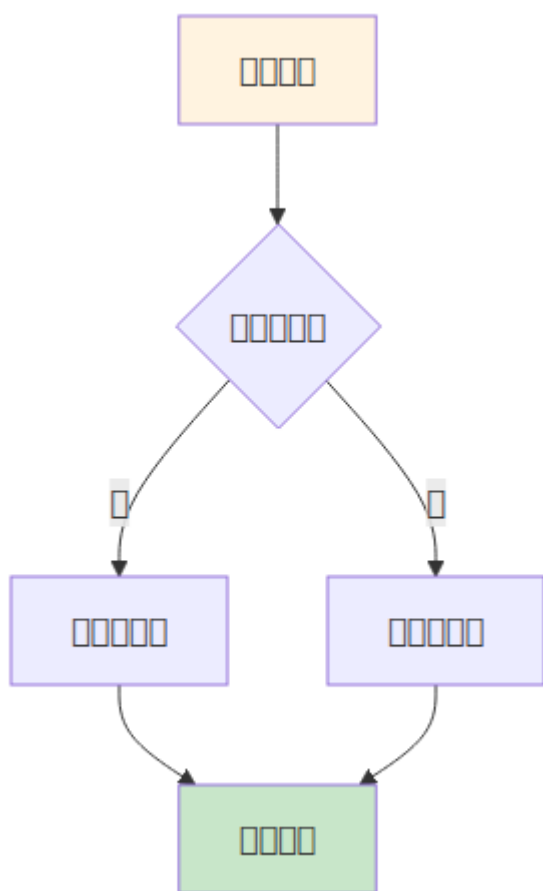
  # タイムアウト
  timeout: 15_000,
  connect_timeout: 15_000,

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

環境変数

変数名	説明	デフォルト値	備考
username	SQL データベースユーザー名	"root"	データベース管理者権限を持つユーザー名
password	SQL データベースパスワード	"password"	データベース管理者パスワード
hostname	SQL データベースホスト名	"localhost"	データベースサーバーの FQDN または IP
database	データベース名	"omnihss"	アプリケーションが使用するデータベース名
pool_size	データベース接続プールサイズ	20	推奨値は10-50程度

□□□□□



□□□□□

- □ 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
  ]
end
```


Diameter 配置

Diameter 配置

```
# config/runtime.exs

diameter_config = %{
  service_name: :omnitouch_hss,

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

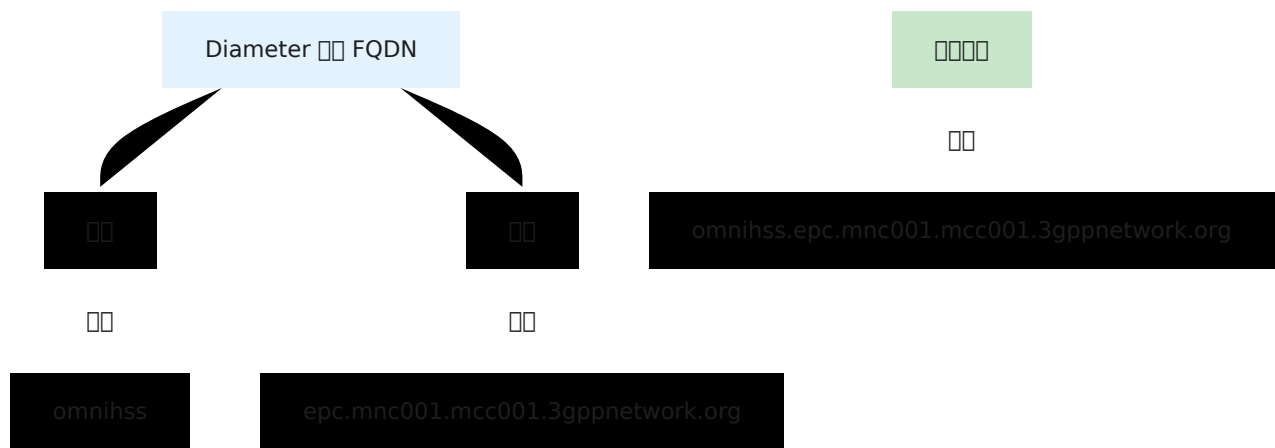
  # 产品信息
  product_name: "OmniHSS",
  vendor_id: 10415, # 3GPP
  supported_vendor_ids: [5535, 10415],

  # 请求超时
  request_timeout: 5000,

  # 对等体
  peers: [
    # 对等体配置
  ]
}

config :hss, :diameter, diameter_config
```

Diameter



□□□□□

- HSS 数据库“omnihss”“hss01”
- 网络 PLMN 的 Diameter 服务器
“epc.mnc001.mcc001.3gppnetwork.org”
- 数据库 名称 {host}.{realm}

□□ Diameter □□□

□ □ □ □ □ □ □ □ □ □ □ □

```
# 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]
```

```
  }
```

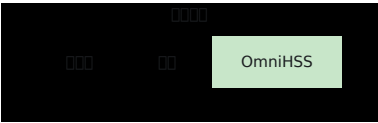
```
]
```

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

```
# config/runtime.exs

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

  # MME 配置 CLR 库
  send_clr_on_mme_change: true,

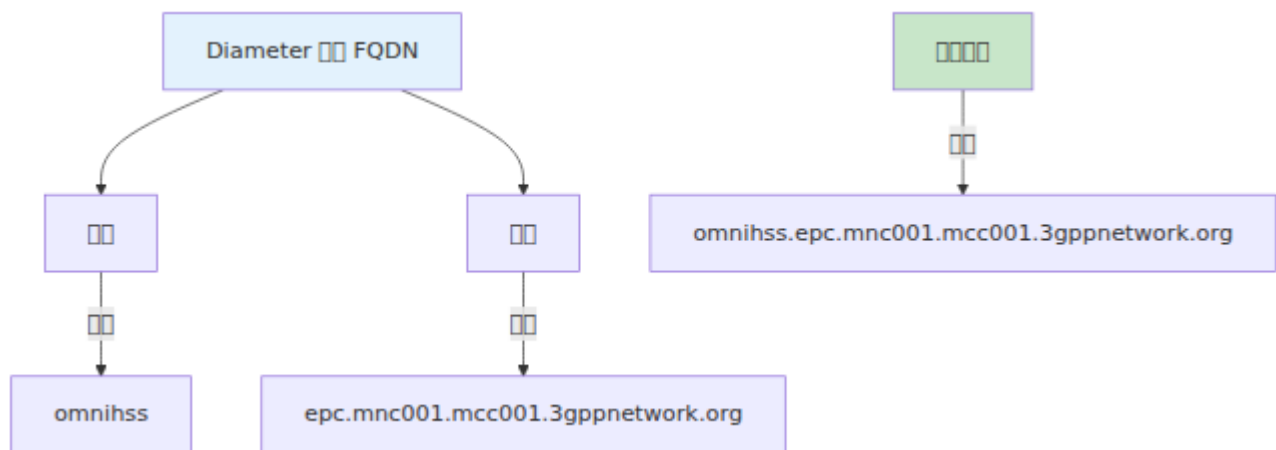
  # 配置 Diameter 库
  stop_diameter_on_database_failure: true,

  # 配置 License
  license_enforced: true,
  license_module: LicenseClient
```

HSS 配置

項目	項目	項目	項目
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 オブジェクト
send_clr_on_mme_change	データベース MME 状態 変更通知	true	データベース リポジトリ オブジェクト
stop_diameter_on_database_failure	データベース Diameter 接続	true	データベース リポジトリ オブジェクト
license_enforced	データベース	true	データベース リポジトリ オブジェクト
license_module	データベース モジュール	LicenseClient	データベース リポジトリ

PLMN



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

□□□□□□

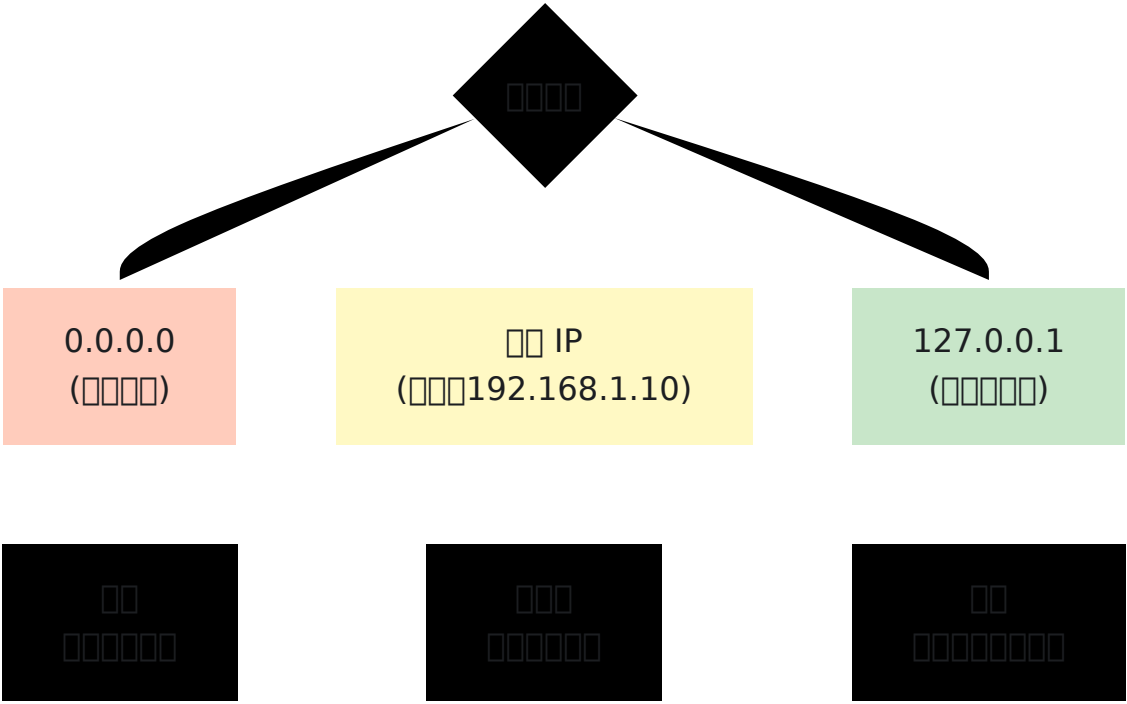
```
# config/runtime.exs

# 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
  ]
```

□□□□□□



IMS

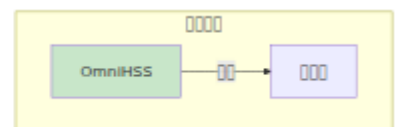
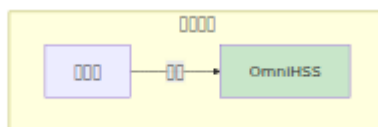
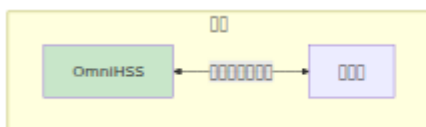
S-CSCF

```
# config/runtime.exs

config :hss, :ims, %{
  scscf: %{
    # 0000:random_peer 0 :round_robin
    selection_method: :random_peer,

    # 00 S-CSCF 00000
    peers: [
      %{
        host:
        "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060",
        capabilities: [] # 0000000
      },
      %{
        host:
        "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060",
        capabilities: []
      }
    ]
  }
}
```

S-CSCF



000000

Peer	Peer	Peer
:random_peer	Random S-CSCF	Random
:round_robin	Round S-CSCF	Round

IMS

IMS EPC

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

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

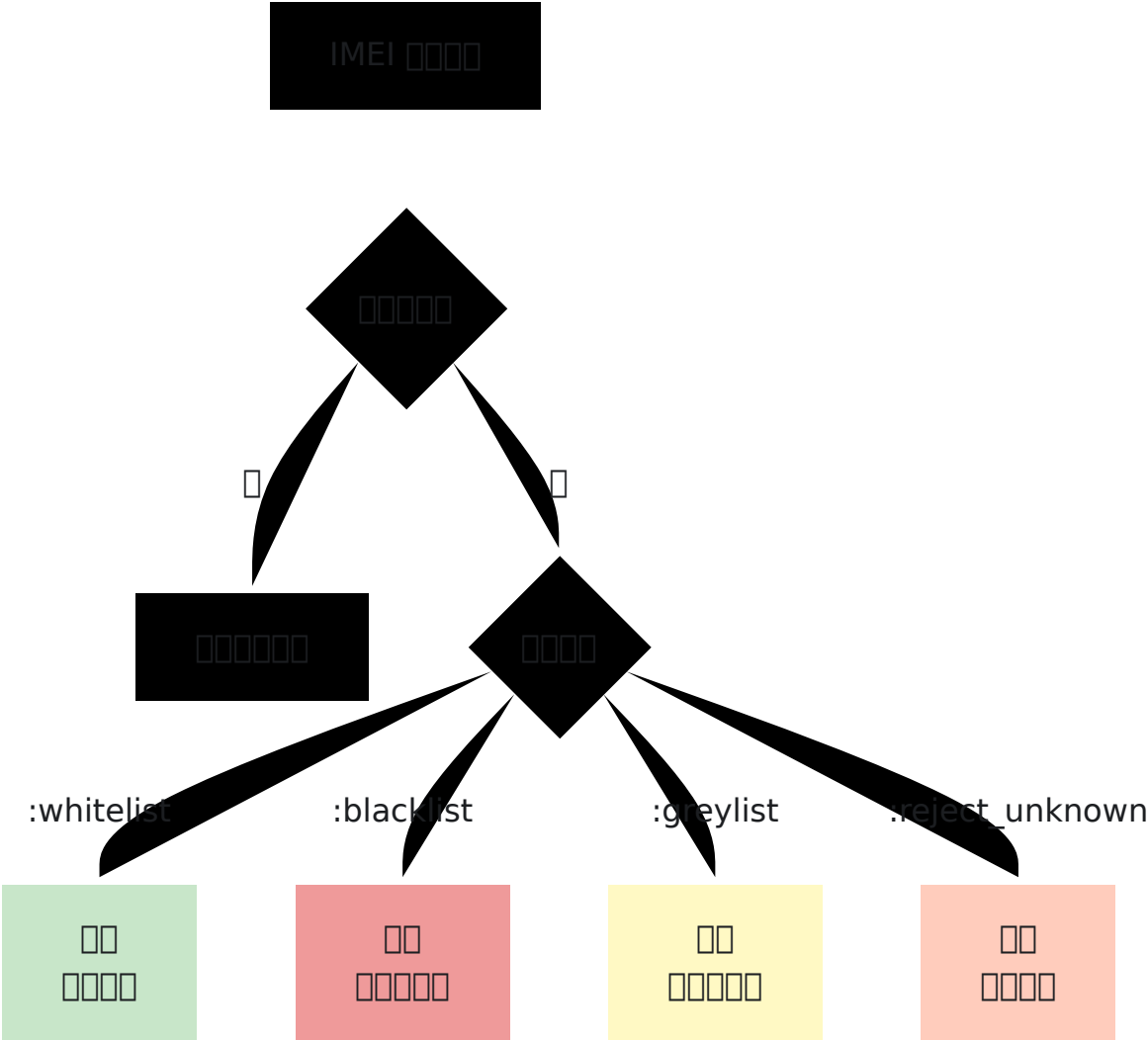
EIR

EIR

```
# config/runtime.exs

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

IMEI 数据库



数据表

表名	表结构	表内容
:whitelist	IMEI 数据库	IMEI 数据库
:blacklist	IMEI 数据库	IMEI 数据库
:greylist	IMEI 数据库	IMEI 数据库
:reject_unknown_equipment	IMEI 数据库	IMEI 数据库

IMEI 数据库 :whitelist 黑名单 :greylist 黑名单

API 環境構築

API 環境

```
# config/config.exs

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

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

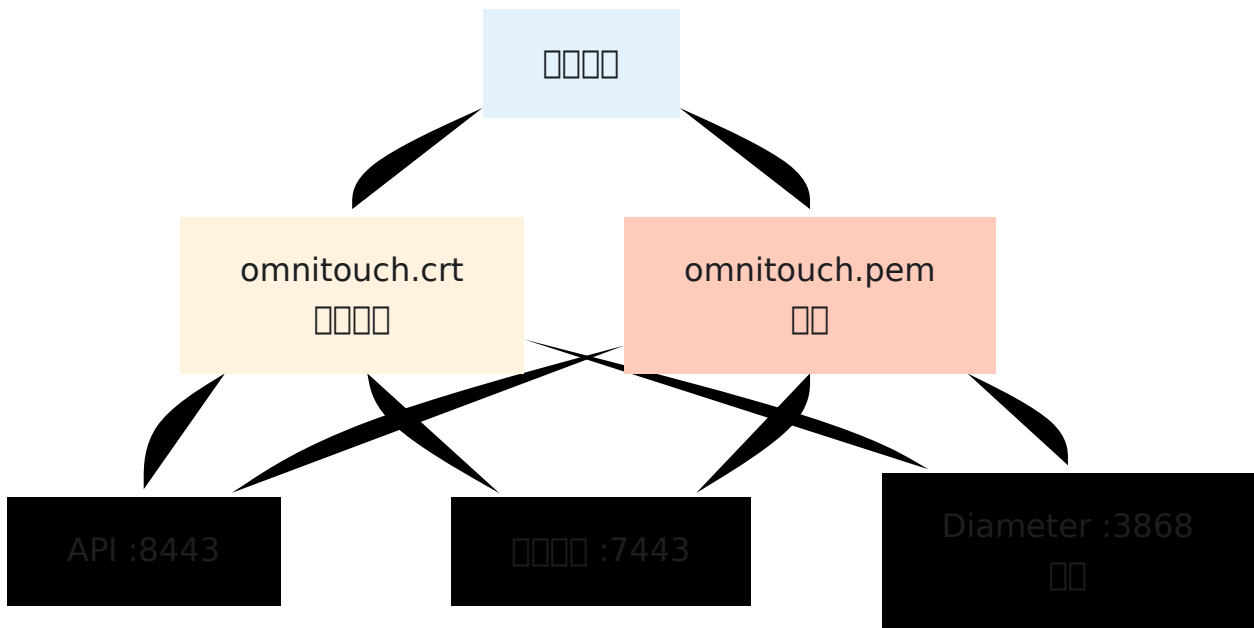
環境構築

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

TLS 証明書



証明書

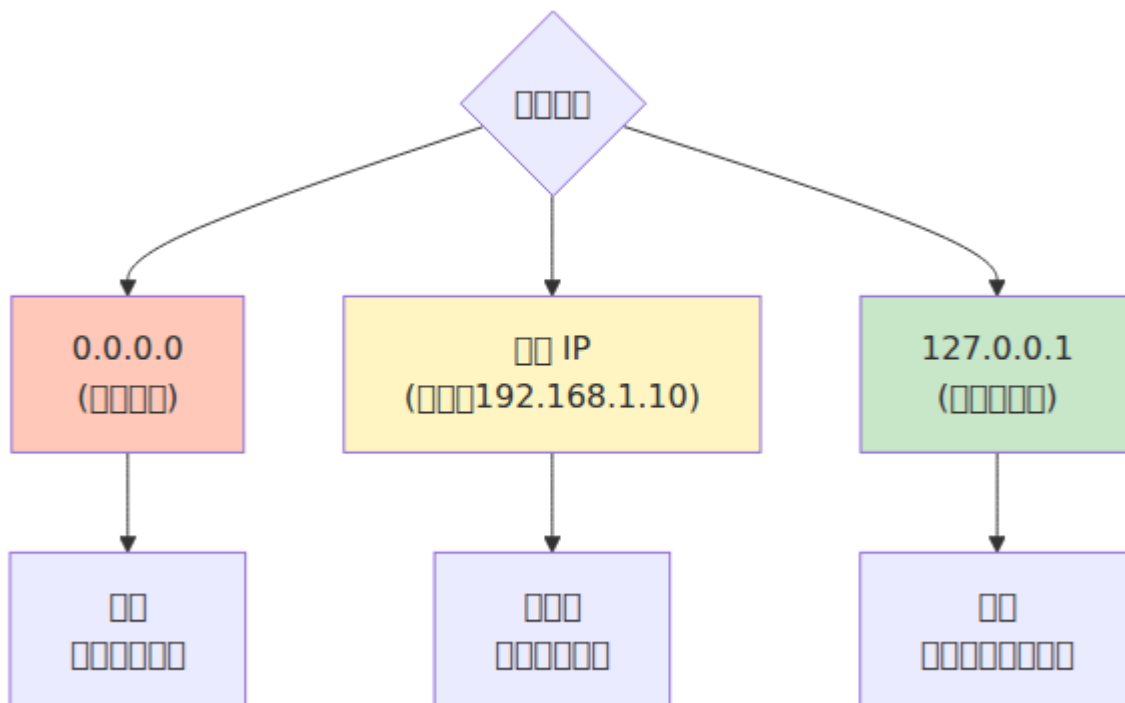
- 証明書 X.509 形式
- 証明書
- 証明書
- CN と SAN 証明書

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** □□□

- ☐ 00 S-CSCF 00000
- ☐ 00 S-CSCF 0000
- ☐ 00 IMS 00

□□□□

- ☐ `00 EIR 00`
 - ☐ `00000000`
 - ☐ `00000000`

□ □ □ □

□□□□□□

1.

□ □ □ □ □ □ □ □ □ □ □ □

2. □□□□□□□

```

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

```

3. API ☐☐☐

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

4. Diameter

□□□□□ Diameter □□
□□□□□

5. 資料庫

資料庫系統
資料庫 SQL 查詢

omnihss

```
# config/runtime.exs - omnihss
```

```
import Config
```

```
#
```

```
=====
```

```
# Database
```

```
#
```

```
=====
```

```
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
```

```
  ]
```

```
#
```

```
=====
```

```
# License
```

```
#
```

```
=====
```

```
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", "omnihss"),
```

```
  product_name: "omnihss"
```

```
#
```

```
=====
```

```
# PLMN HSS
```

```
#
```

```
=====
```

```

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
```

← 000000 | 00000000 →

OmniHSS 開箱開箱開箱

← 開箱開箱

開箱

- 開箱開箱
 - 開箱開箱
 - 開箱
 - Diameter 開箱
 - 開箱
 - 開箱
 - 開箱開箱
-

開箱開箱開箱

OmniHSS 開箱開箱開箱開箱開箱開箱開箱 Diameter 開箱開箱開箱開箱 Phoenix
LiveView 開箱開箱開箱開箱開箱開箱

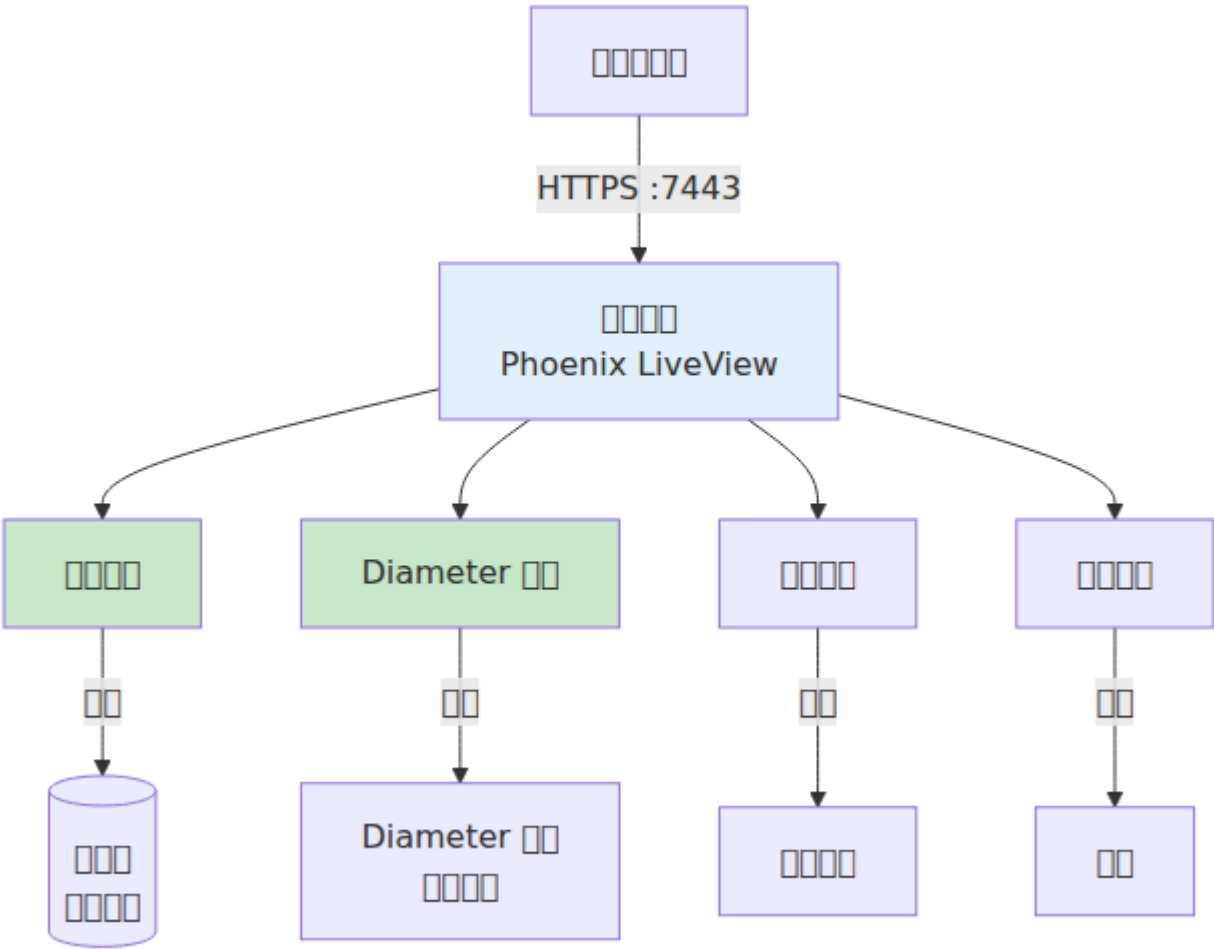
開箱開箱

- 開箱 - 開箱開箱
- 開箱 - 開箱開箱開箱開箱
- **Diameter** 開箱 - 開箱開箱開箱
- 開箱 - 開箱開箱
- 開箱開箱 - 開箱開箱

□□□□

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

□□□□□□



□□□□□□□

□□□□

1. □□□□□□□

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

TLS 証明書

証明書を確認

証明書を確認

証明書

- **7443** 証明書を確認
- **HTTPS** 証明書 - HTTP
- 証明書を確認 7443

証明書

証明書を確認LiveView, WebSockets

- Chrome/Chromium
- Firefox
- Safari
- Edge

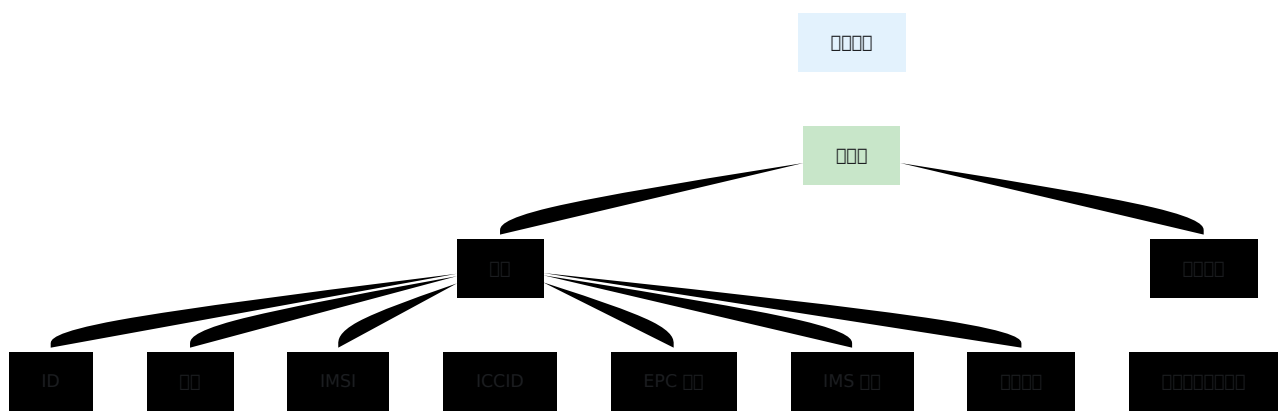
証明書を確認 Internet Explorer

証明書

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

証明書を確認

□□□□



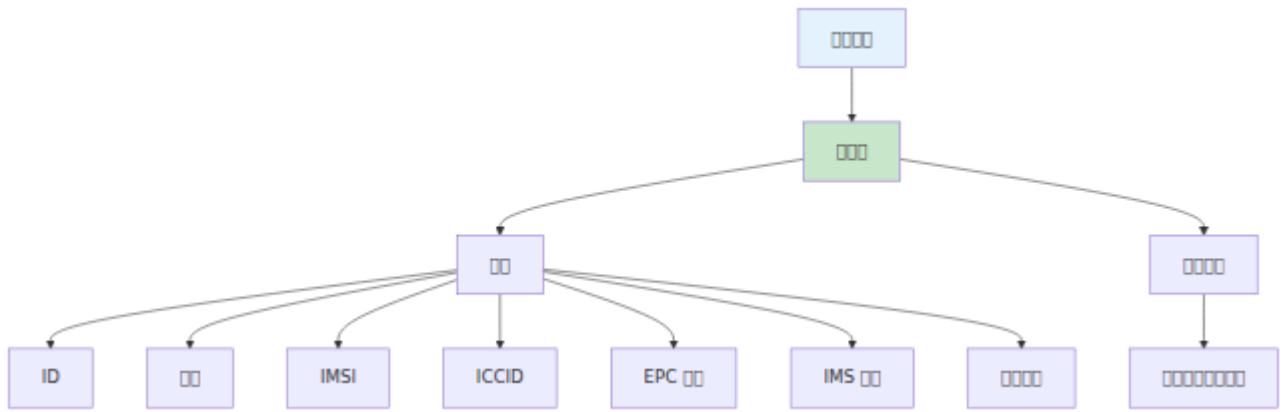
□□

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

□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□



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

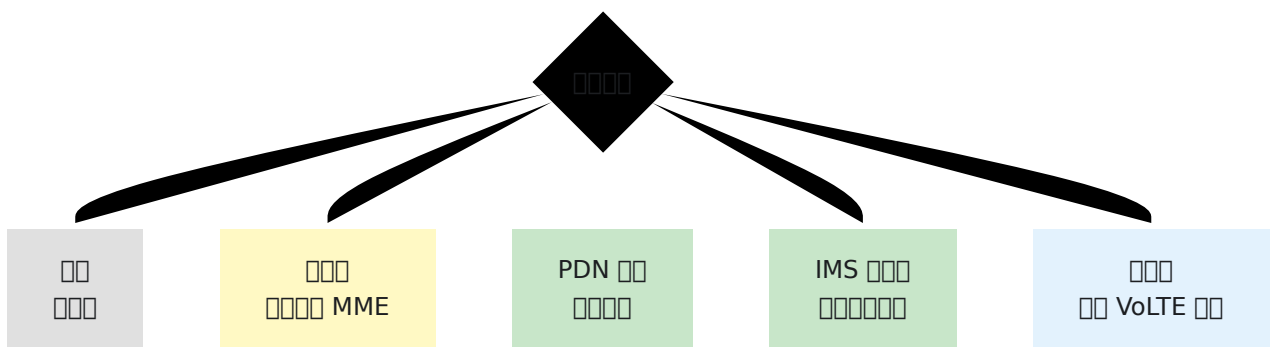
- **MME** - MME
- - MME Diameter
- **RAT** - “E-UTRAN” LTE
- - Diameter

IMS

- **S-CSCF** - S-CSCF SIP URI
- **IMS** - SIP URI sip:[+14155551234@ims.example.com](tel:+14155551234)
- **P-CSCF** - HSS P-CSCF
- **I-CSCF** - HSS I-CSCF

111

- **PDN** - **Packet Data Network**
- **VoLTE** - **Voice over LTE**



□ □ □ □ □ □ □

- **PDN** $\text{PDN\_Delay} > 0$
- **IMS** $\text{IMS\_Delay} > 0$

□ □ □ □

□□□□□ **1** □ □□□□□□□□□□□□□□□□□

5/5

- 
- 
- 

□ □ □ □

1.

- □□□□□□□□

- 网络功能虚拟化
- 网络 IMS 网络

2. 网络

- 网络功能虚拟化
- 网络功能虚拟化网络
- 网络功能
- 网络功能

3. 网络

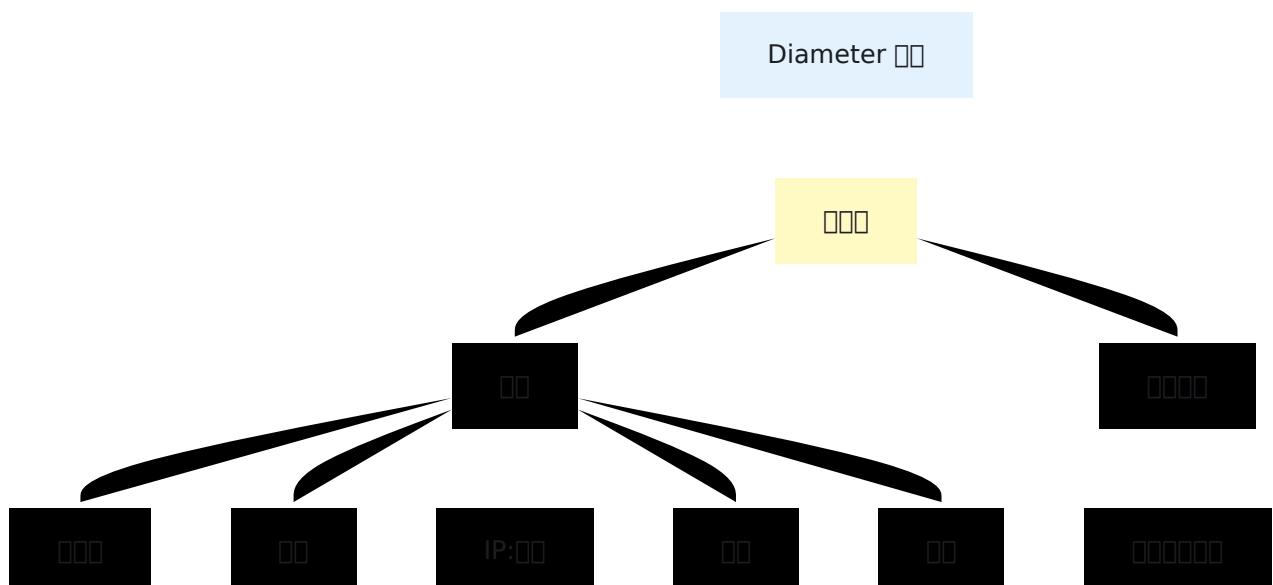
- 网络功能
- 网络 PDN 网络
- 网络 VoLTE 网络

Diameter 网络

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

Diameter 网络 Diameter 网络

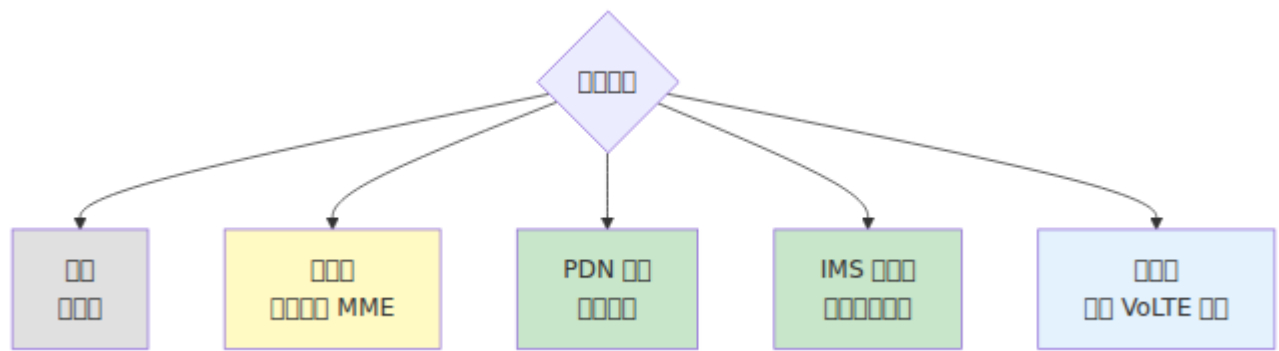
网络



□□

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

□□□□□



□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□

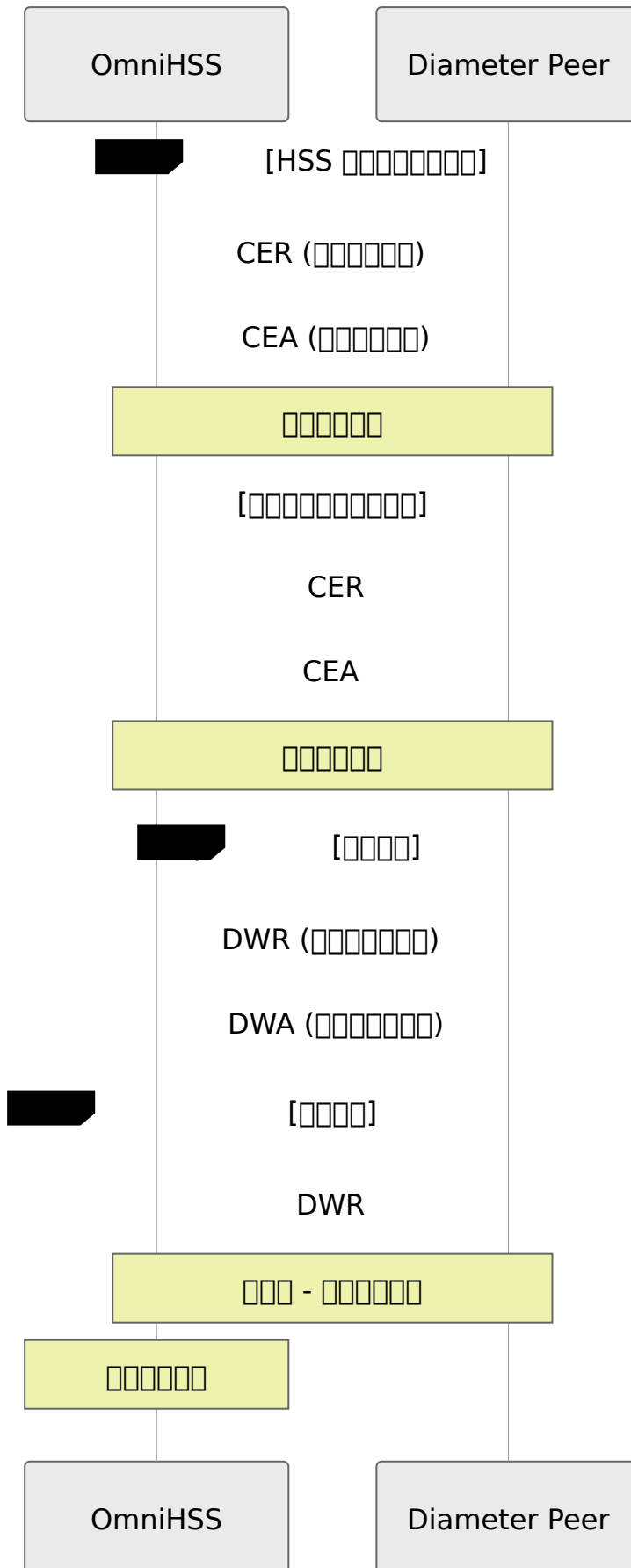
- □□□□ - □ HSS □□□□□□
- □□□□ - □□□□□□□□□□
- □□ **ID** - □□□□ Diameter □□

□□ **ID** □□□

- 16777251 - S6a (MME)

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

□□□□□□



목차

Diameter 프로토콜 1 에 대해서

목차

1. 프로토콜

- 프로토콜의 구조
- 프로토콜의 구성
- 프로토콜의 특징

2. 프로토콜

- 프로토콜의 구조
- 프로토콜의 구성 TCP 와 SCTP
- 프로토콜 ID 프로토콜
- 프로토콜의 특징

3. 프로토콜

- 프로토콜의 구조
- 프로토콜의 구성
- 프로토콜의 특징

목차

프로토콜

프로토콜

1. 프로토콜

2. 프로토콜의 구조

3. 프로토콜의 구성

4. Diameter 프로토콜

5. 프로토콜의 특징 TLS

프로토콜

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

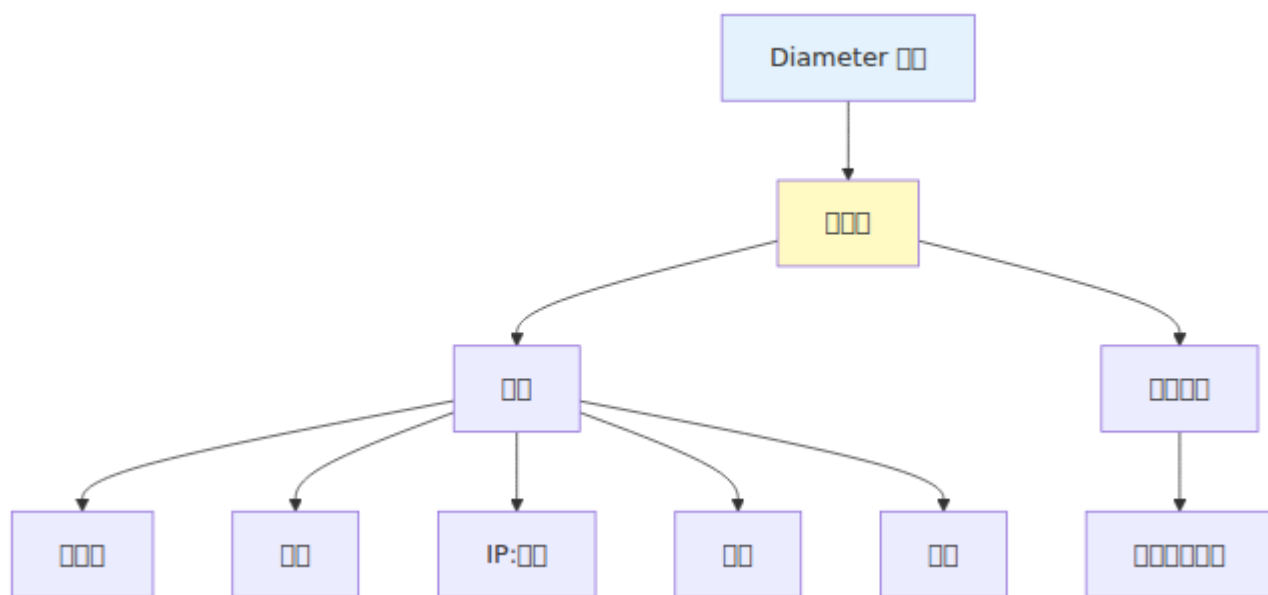
- 1.
- 2.
- 3.
4. Diameter

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

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

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

□□□□



□□□□

1. □□□□

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

2. □□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

3. □□□□

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

□□□□

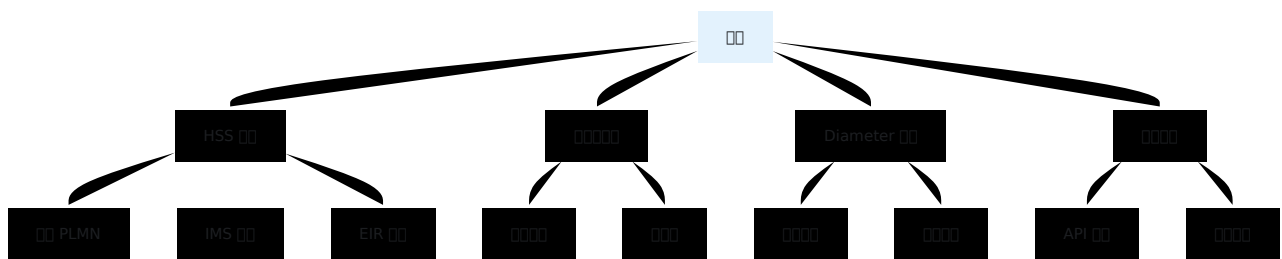
URL: `https://[hostname]:7443/configuration`

□□□□□ OmniHSS □□□□□□□□

□□

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

□□□□



□□□□

1. □□□□

- □□ runtime.exe □□□□□□□□
- □□□□□□□□□□
- □□ Diameter □□□□

2. □□□□

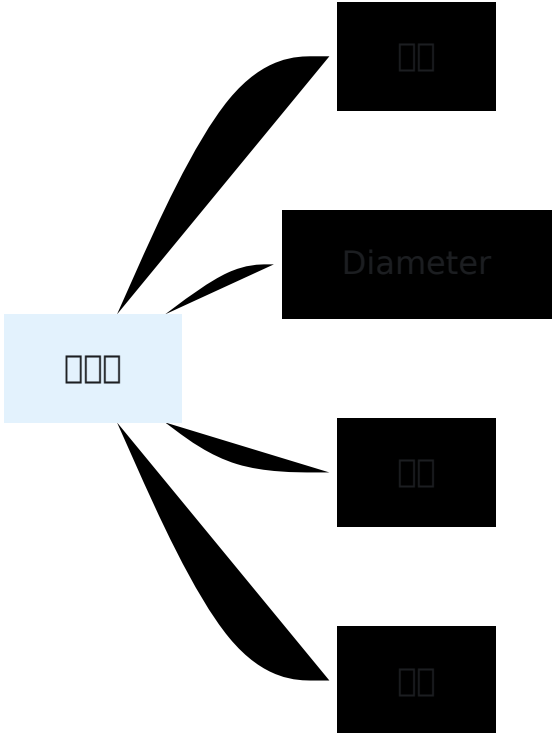
- □□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□

3. □□

- □□□□□□□□□□□□

- □□□□□□□□

[illegible]



□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □

- **Ctrl+R / F5** - 刷新
- **Ctrl+F** - 查找
- **Ctrl+T** - 新建标签

□□□□□□□□

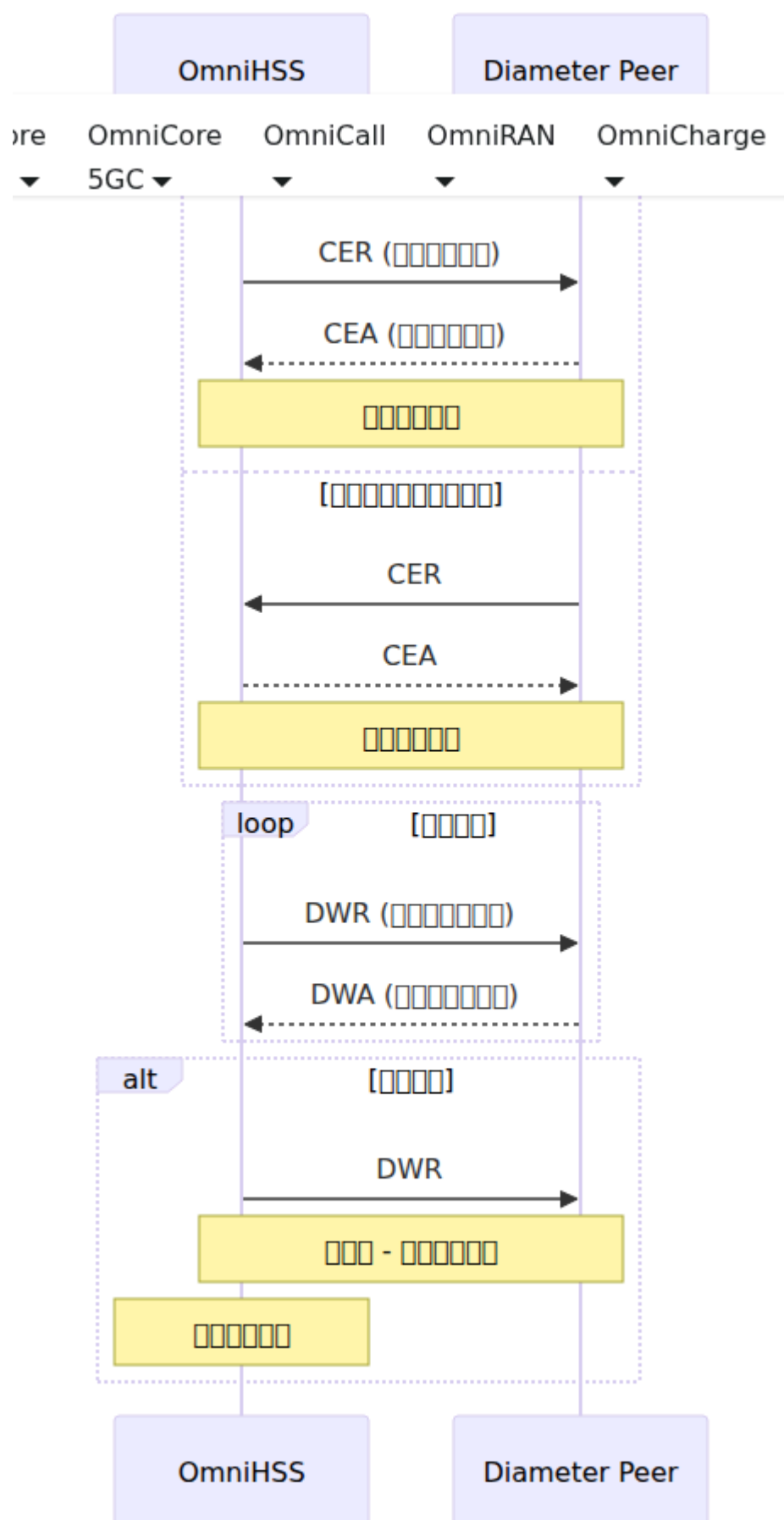


Table 1

Table 1: Summary of Key Performance Indicators (KPIs) for the Project

KPI	Target	Actual
Project Diameter (mm)	1 mm	2+ mm (exceeds target)
Efficiency (%)	> 80%	> 90%
Throughput (units/hr)	> 5%	> 10%
Cost Reduction (%)	> 80% (target)	> 95% (actual)

OmniHSS

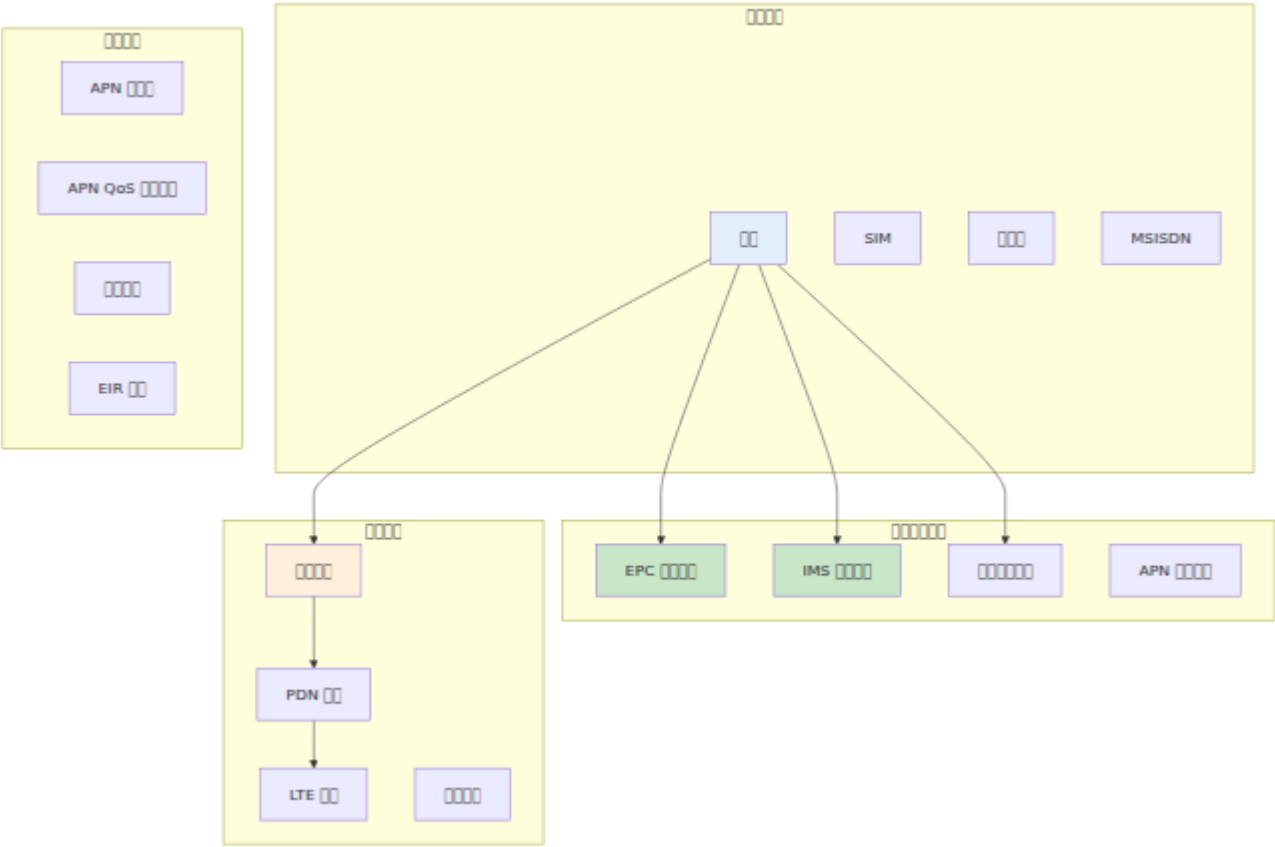
←

11

- 
- 
- 
- 
- 
- 
- 

[illegible]

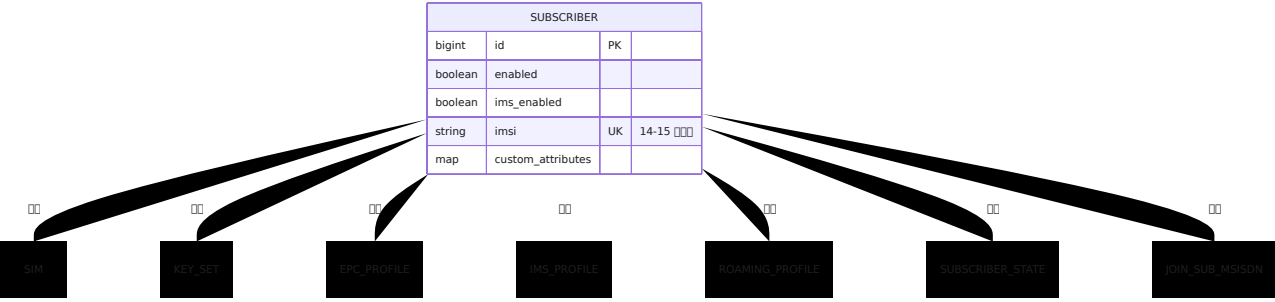
□□□□



□□□□

□□

□□□□□□□□□□□□□□□□



□□:

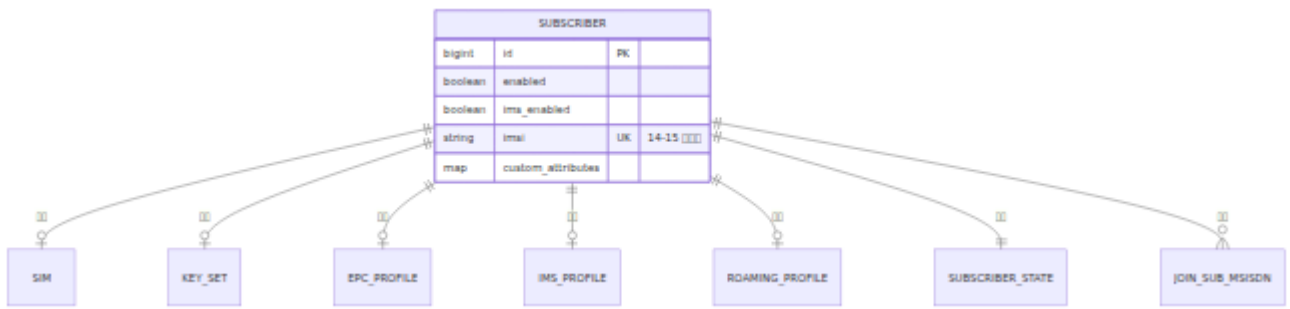
欄名	型	説明	例
id	bigint	ID	00000
enabled	boolean	IMS 有効フラグ	例: true
ims_enabled	boolean	IMS 有効フラグ	例: true
imsi	string	IMSI	14-15 桁の数字
custom_attributes	map	カスタム属性	例
sim_id	bigint	SIM ID	例
key_set_id	bigint	キーセット ID	例
epc_profile_id	bigint	EPC プロファイル ID	例
ims_profile_id	bigint	IMS プロファイル ID	例
roaming_profile_id	bigint	ローミング プロファイル ID	例
subscriber_state_id	bigint	サブスクリプション状態 ID	例

例:

- IMSI
- IMSI 14-15 桁の数字
- MSISDN
-
- enabled IMS
- ims_enabled IMS

SIM

SIM SIM



表结构:

字段名	数据类型	描述	备注
iccid	string	ICCID ID	唯一
sim_vendor	string	SIM 厂商	唯一
batch_name	string	批次名	唯一
is_esim	boolean	是否为 eSIM	默认 false
pin1, pin2	string	PIN 码	唯一
puk1, puk2	string	PUK 码	唯一
adm1 - adm10	string	ADM 码	唯一
kic, kid	binary	OTA 密钥	唯一

说明:

- ICCID 唯一标识 SIM 卡
- SIM 厂商唯一标识
- PIN/PUK 唯一标识 SIM 卡
- ADM 唯一标识 SIM 卡
- KIC/KID 唯一标识 SIM 卡 OTA 密钥

000

00000000000000000000

KEY_SET			
bigint	id	PK	
binary	ki		128 0
binary	opc		128 0
binary	op		128 0
binary	amf		16 0
bigint	sqn		48 000
string	authentication_algorithm		

000

SUBSCRIBER

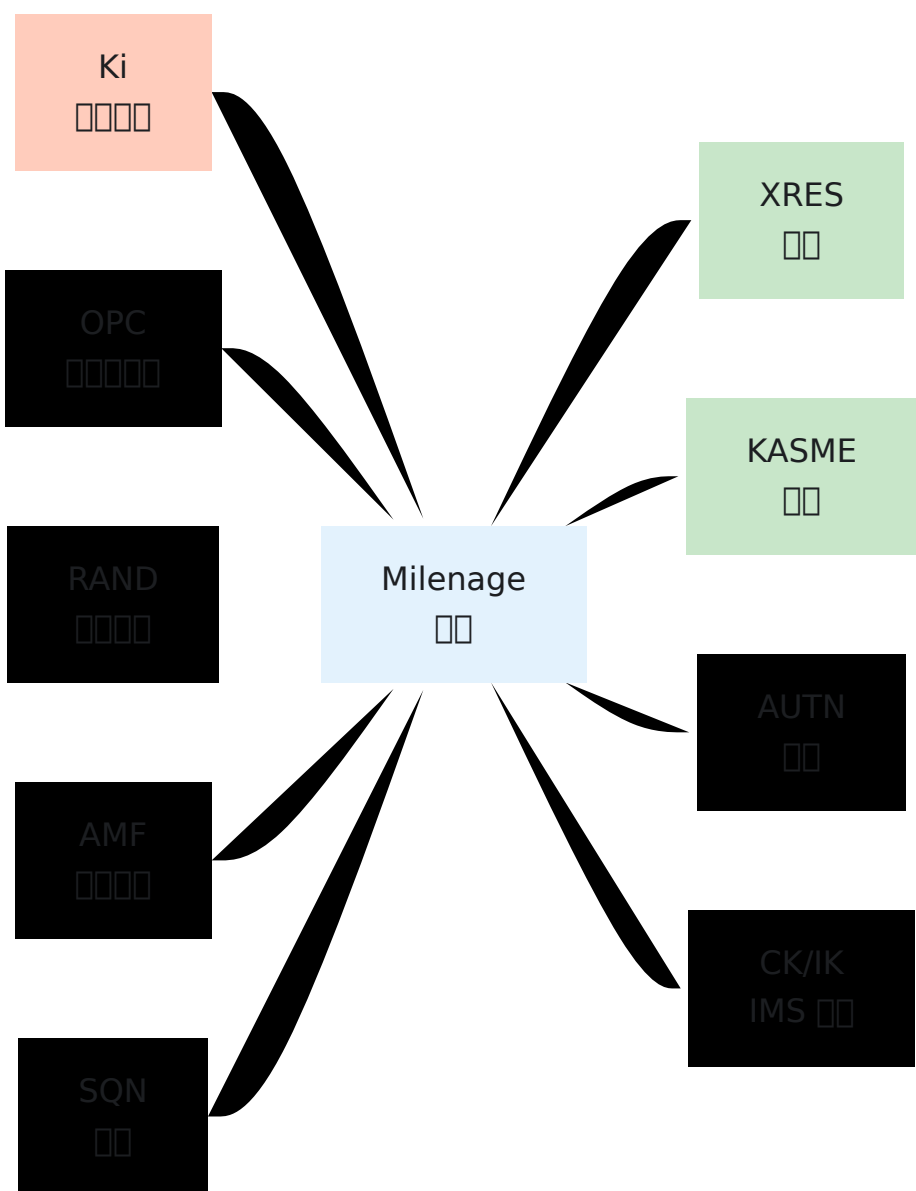
00:

項目	型別	説明	長さ
ki	binary	秘密鍵	128 ビット16 バイト
opc	binary	オペレーションコード	128 ビット
op	binary	オペレーションコード OPC	128 ビット
amf	binary	認証メッセージ	16 ビット2 バイト
sqn	bigint	シーケンス番号	48 ビット
authentication_algorithm	string	認証アルゴリズム	文字列 "milenaage"
ota_counter	bigint	OTA カウンタ	48 ビット

項目:

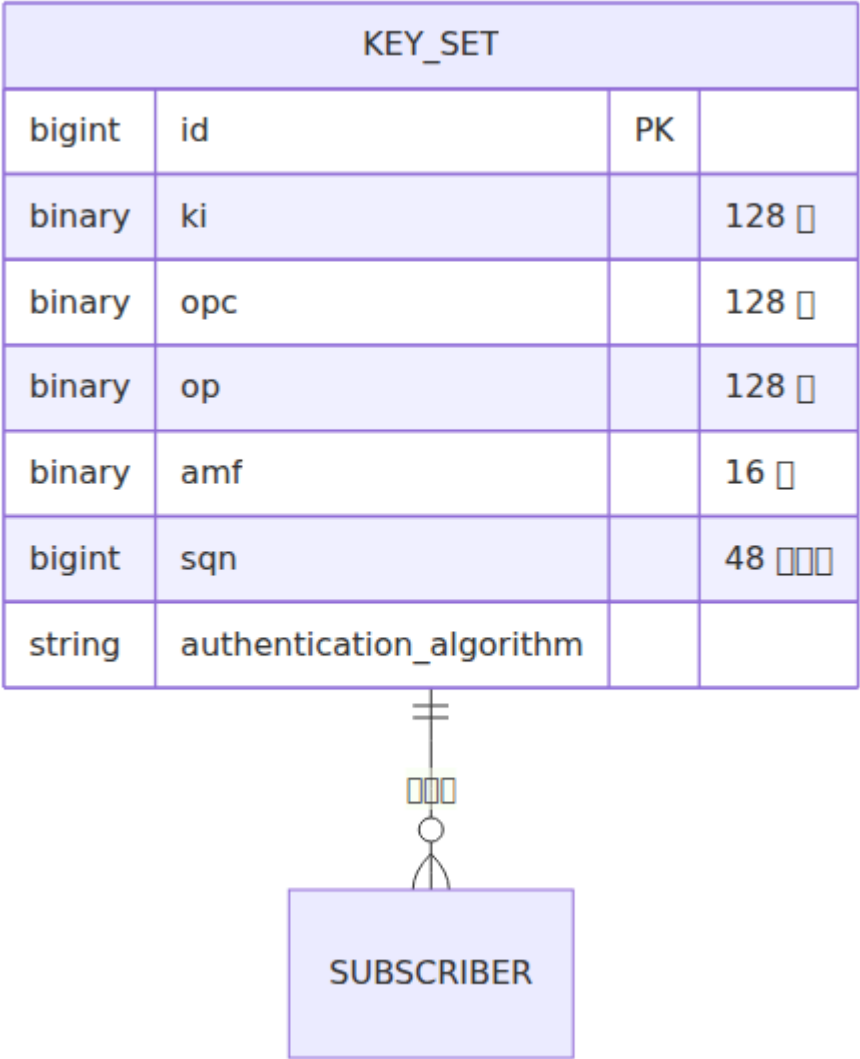
- 秘密鍵
- Ki SIM 秘密鍵
- オペレーションコード OPC OP コード OPC コード OP コード
- SQN シーケンス番号
- Milenage 認証アルゴリズム

項目:



MSISDN

MSISDN 0000000000



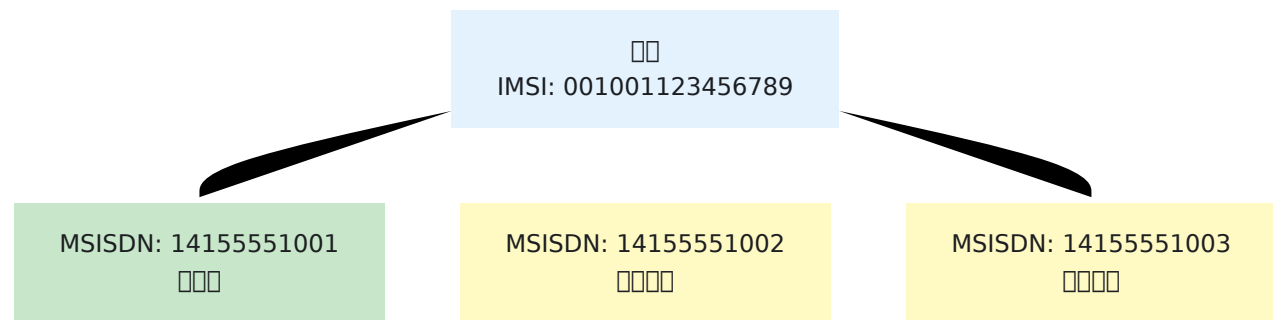
例:

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

例:

- MSISDN 例
- 例 MSISDN 例
- 例 MSISDN 例
- 例: 例 + 例 "+1 415-555-1234" 例 "14155551234"

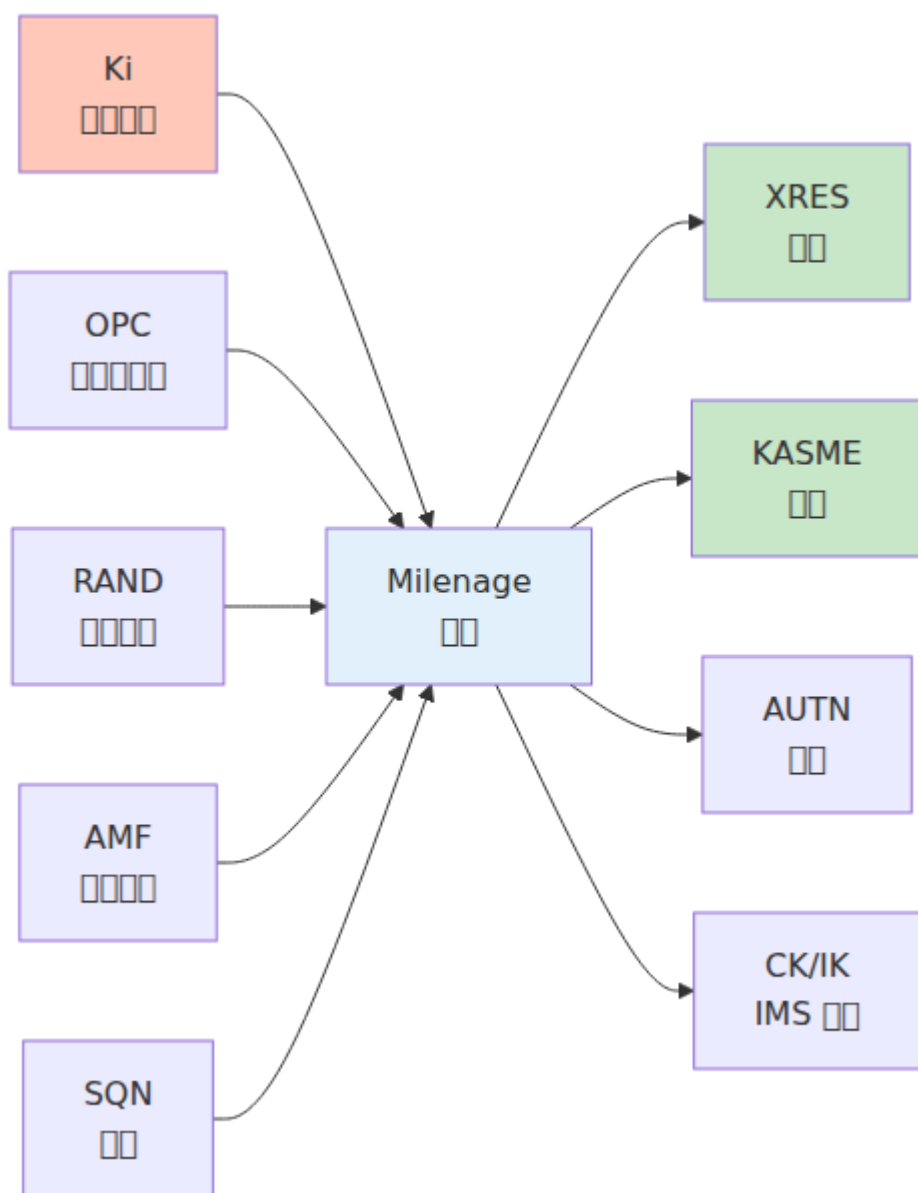
例 **MSISDN** 例:



□□□□□□

EPC □□□□

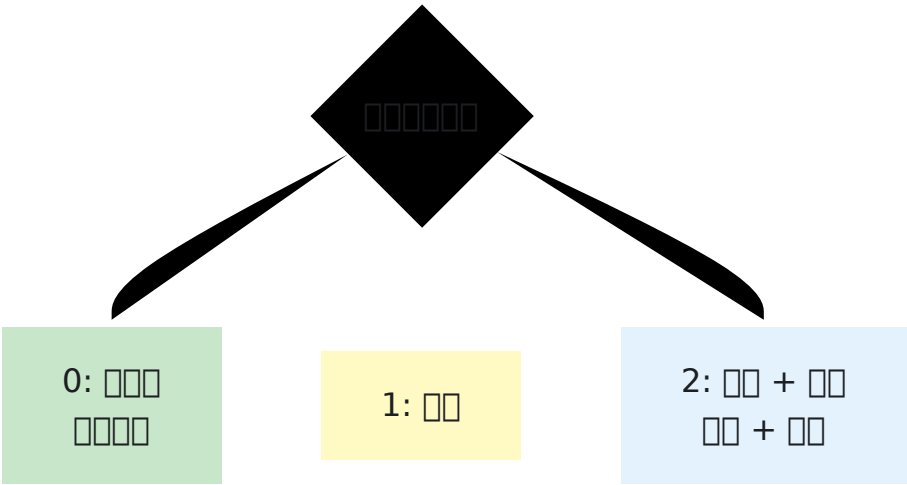
EPC □□□□□□ LTE □□□□□□□□



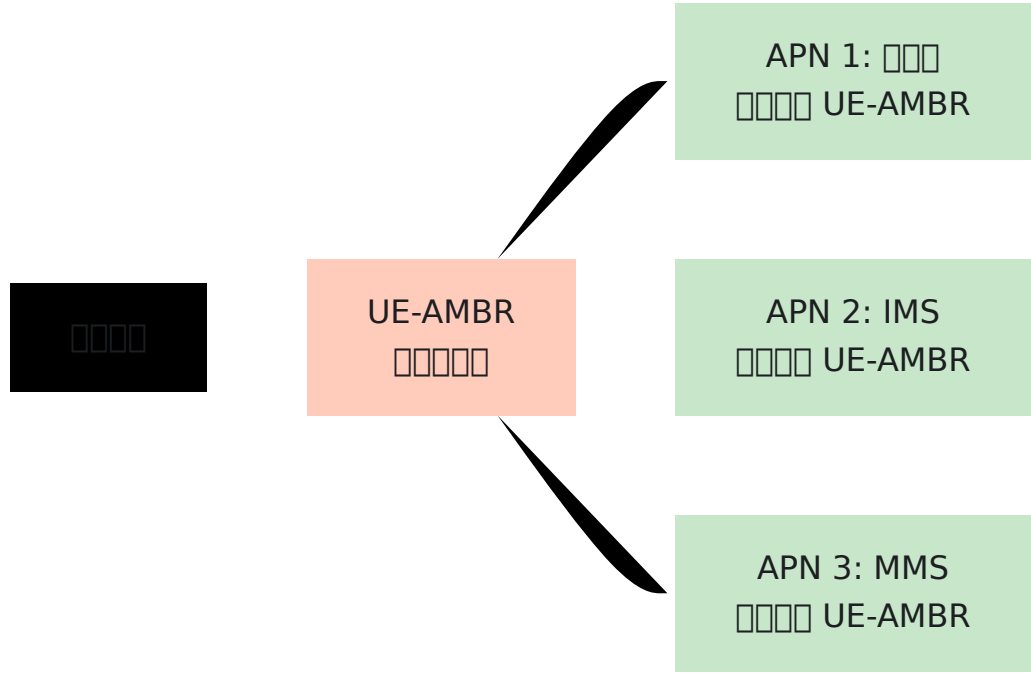
00:

項目	型	範囲	単位
name	string	0~31 文字	
ue_ambr_dl_kbps	integer	0~32767 整数	Kbps
ue_ambr_ul_kbps	integer	0~32767 整数	Kbps
network_access_mode	string	0~31 文字	"0" または "1" 文字
tracking_area_update_interval_seconds	integer	TAU 区間 整数	

AMBRの範囲:

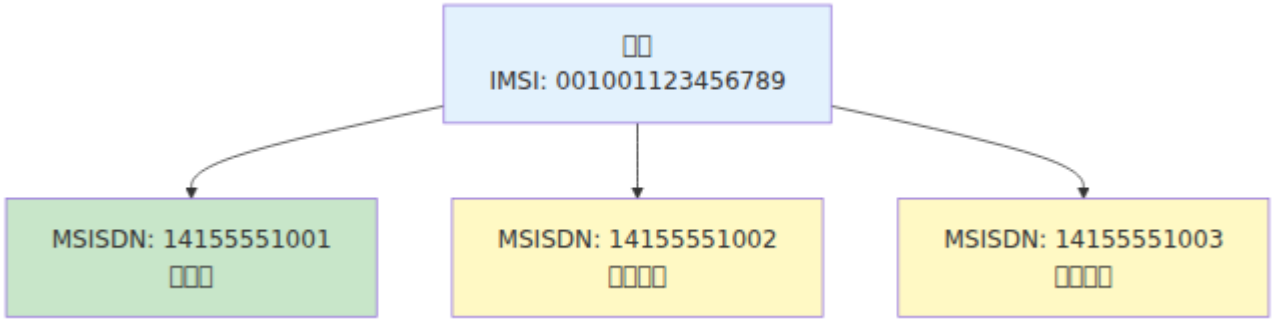


AMBRの範囲:



IMS

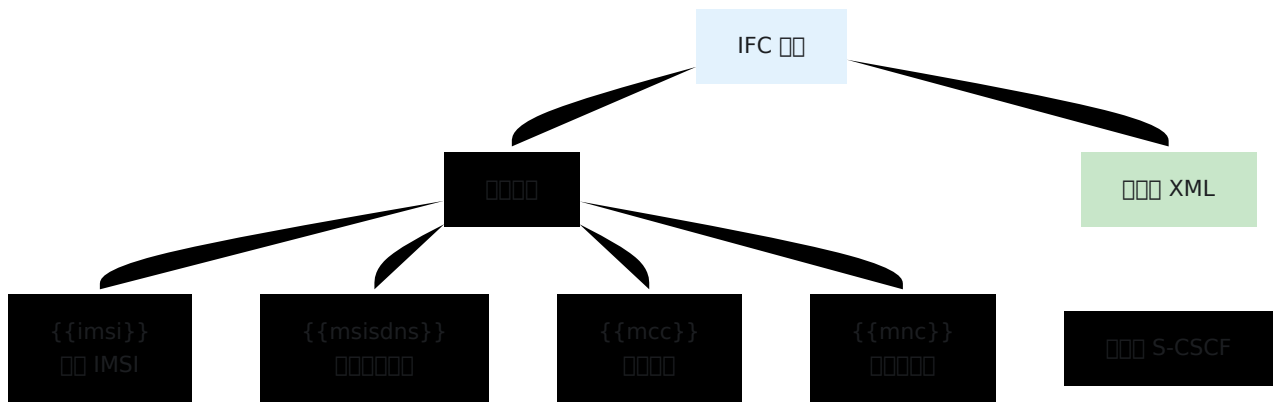
IMS



:

name	string		
ifc_template	text	XML	XML

IFC

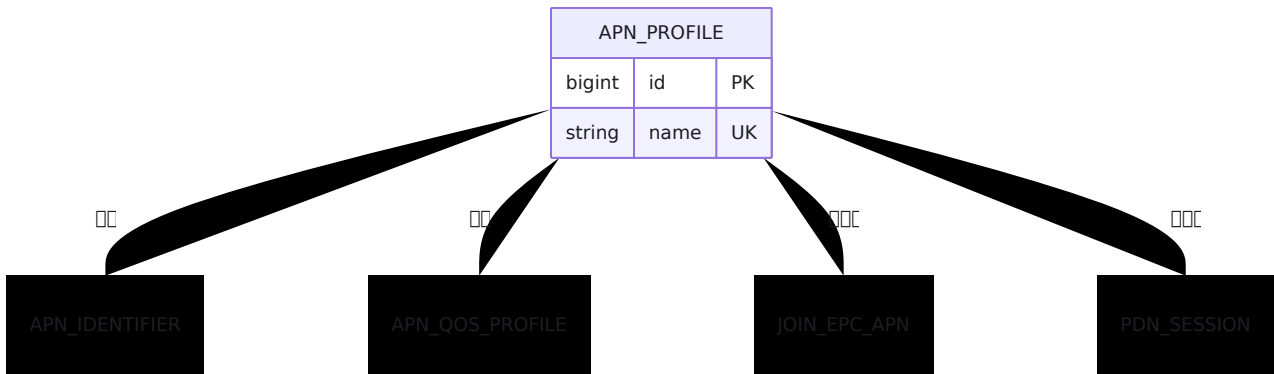


IFC:

- IFC is the IMS FQDN
- It is used to identify the IMS network
- It is used to identify the S-CSCF
- It is used to identify the IMS network

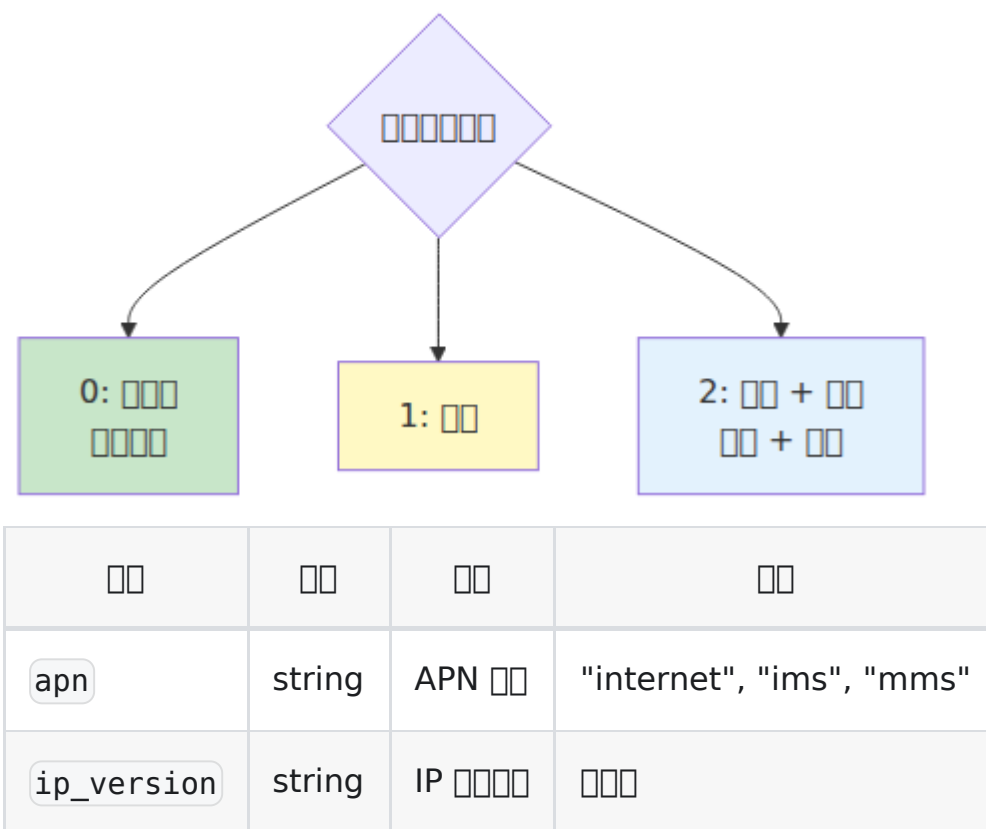
APN

APN is the Access Point Name

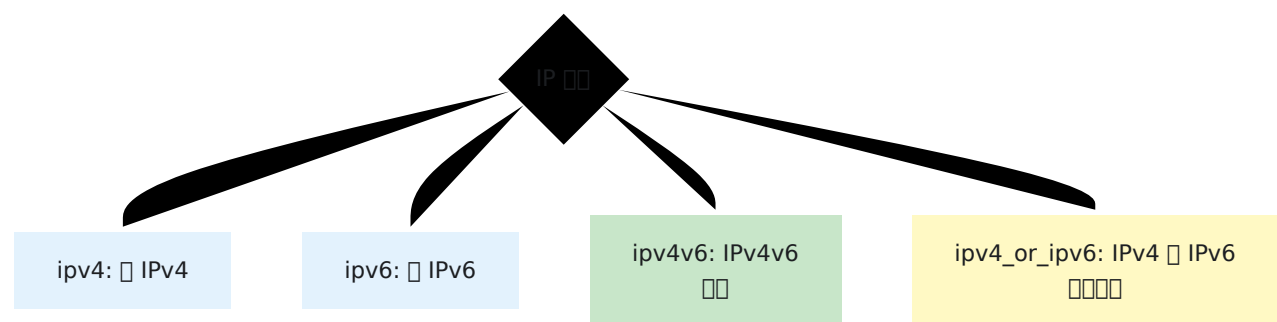


APN:

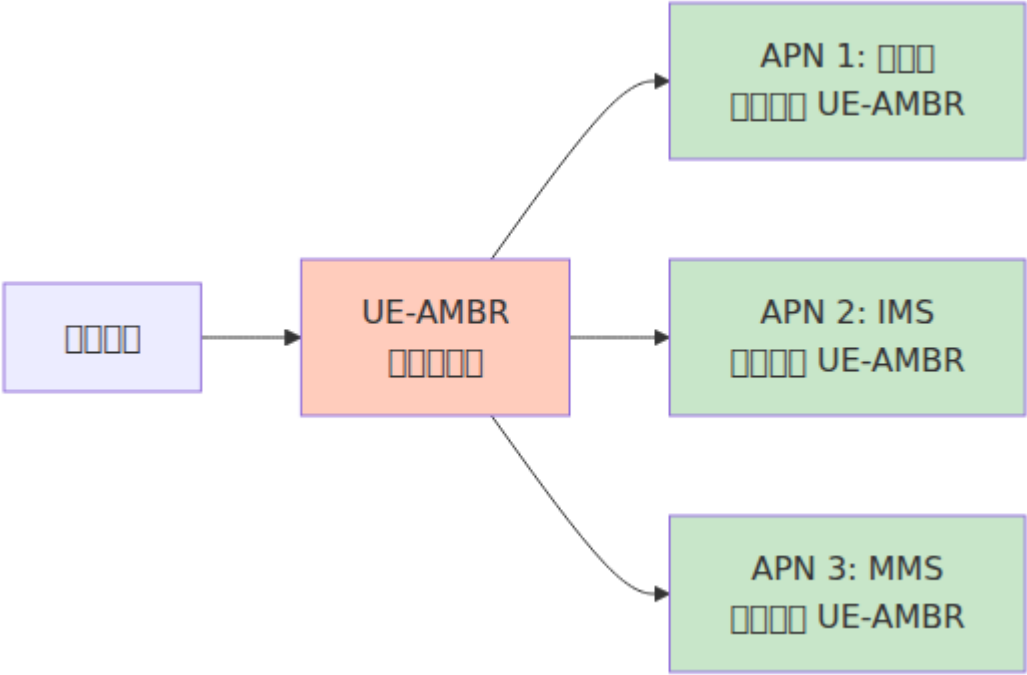
APN is the Access Point Name



IP Selection:



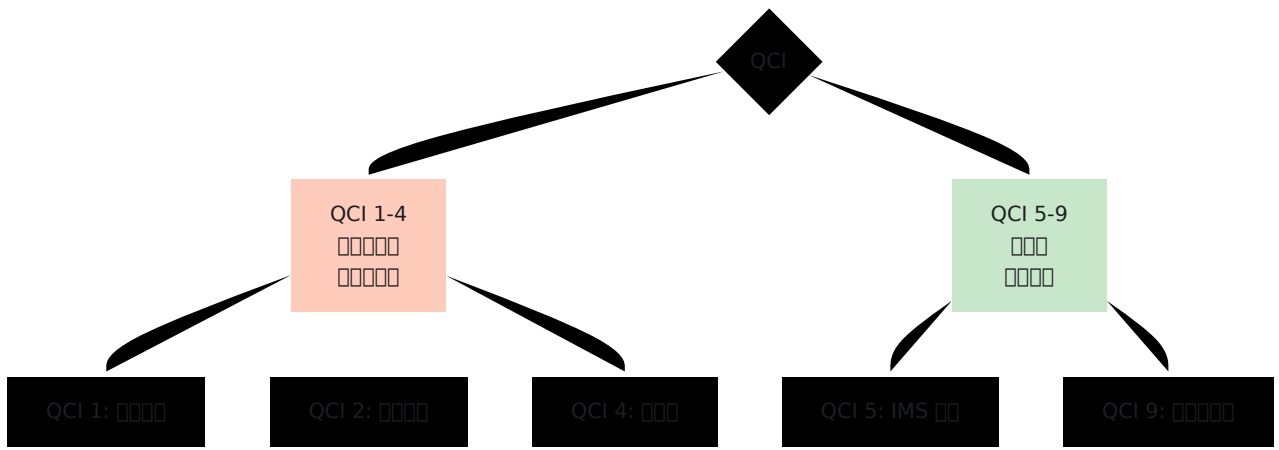
APN QoS Selection



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 :



00000000

00000000000000000000000000000000

ROAMING_PROFILE			
bigint	id	PK	
string	name	UK	
string	data_action_if_no_rules_match		000000
string	ims_action_if_no_rules_match		000000

00

000

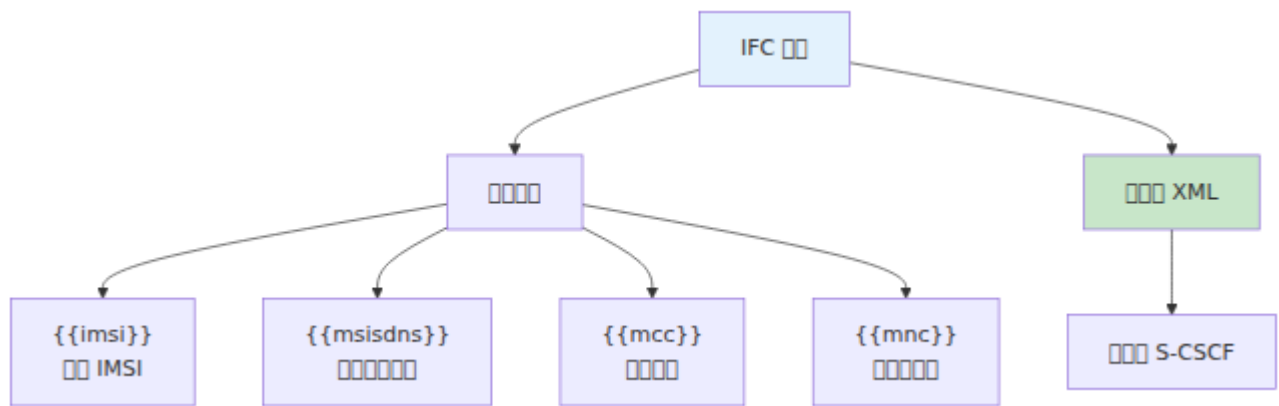
JOIN_ROAM_RULE

SUBSCRIBER

00

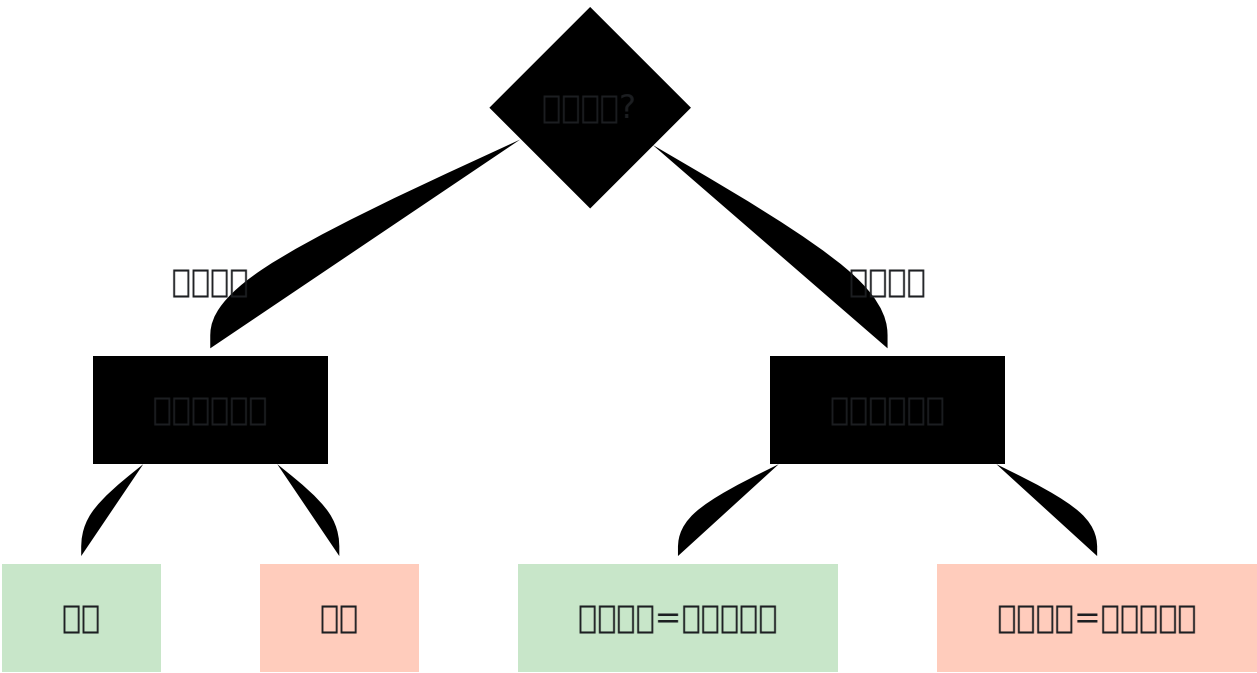
ROAMING_RULE

网络:



网络:

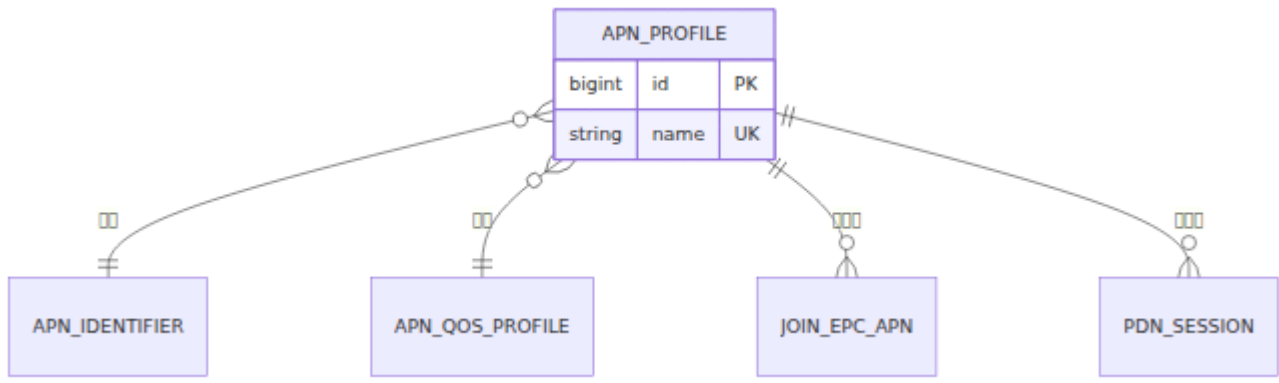
网络
MCC: 310, MNC: 410



数据库

数据库

数据库数据库数据库



数据库:

数据库:

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

数据库:

- last_seen_mme - MME 数据库
- last_seen_realm - MME 数据库
- last_seen_rat_type - LTE/5G 数据库

IMS 数据库:

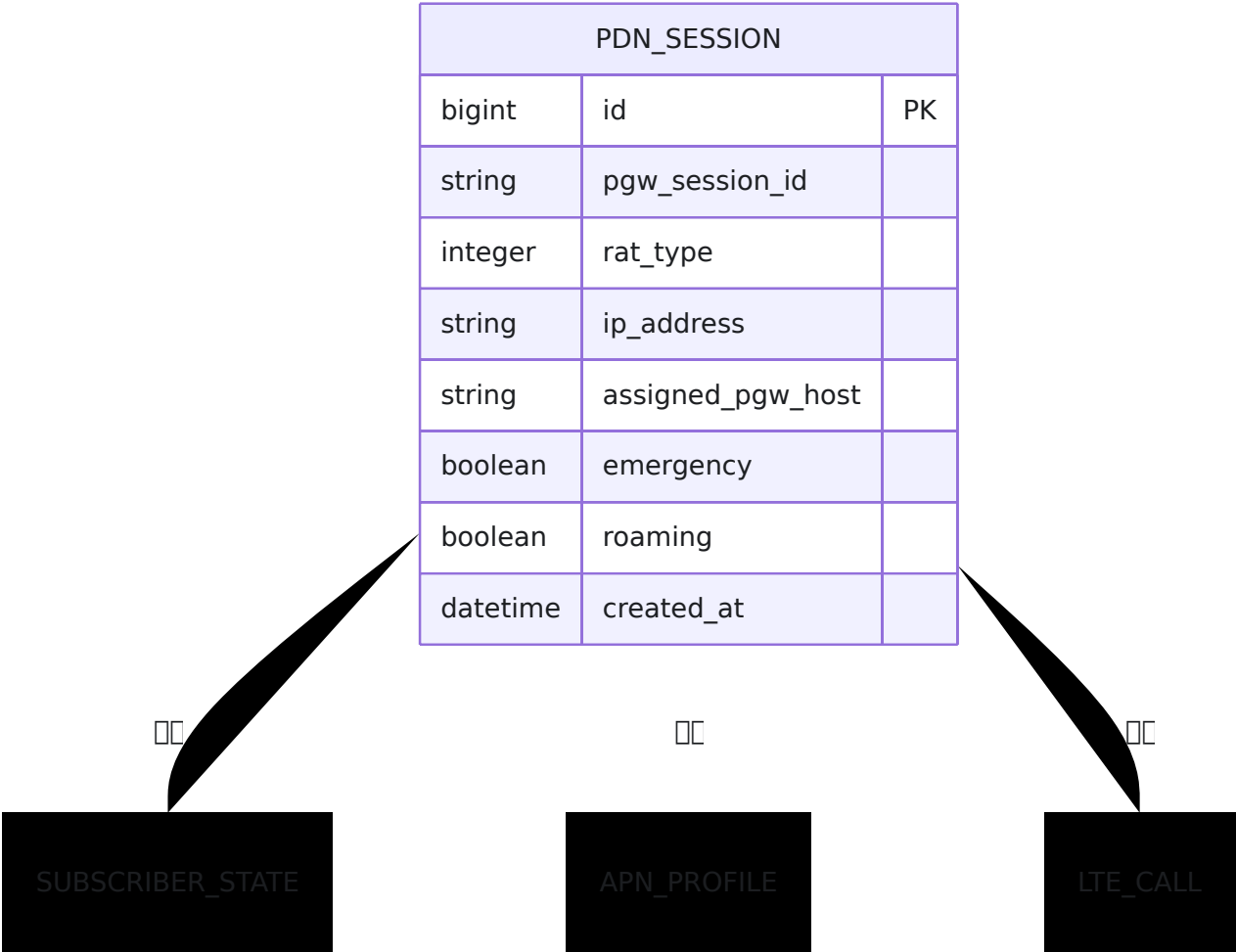
- assigned_scscf - S-CSCF 数据库
- ims_public_identity - SIP URI 数据库
sip:+14155551234@ims.example.com
- sh_repository_data - IMS 数据库

数据库:

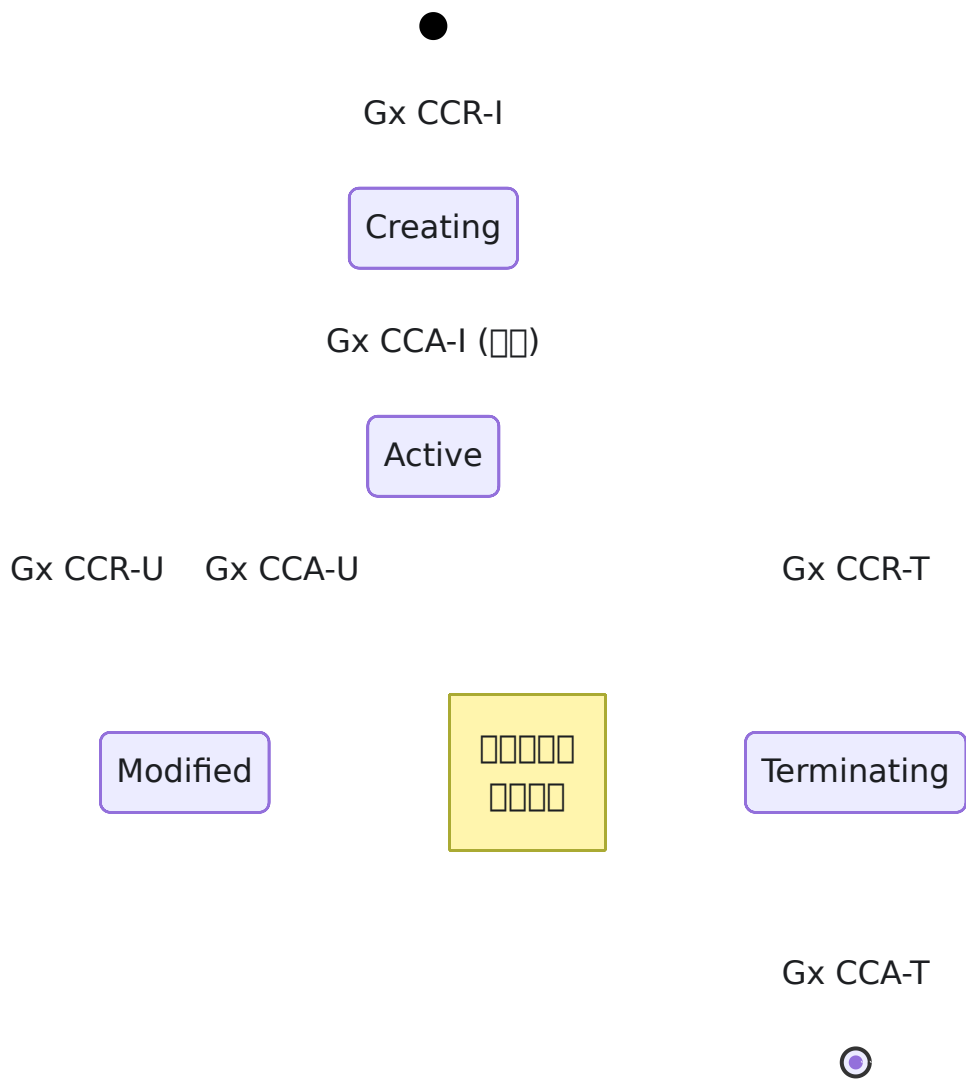
- `last_seen_at` - 000000000000
- 00 `last*_at` 000000000000

PDN 00

PDN 0000000000000000

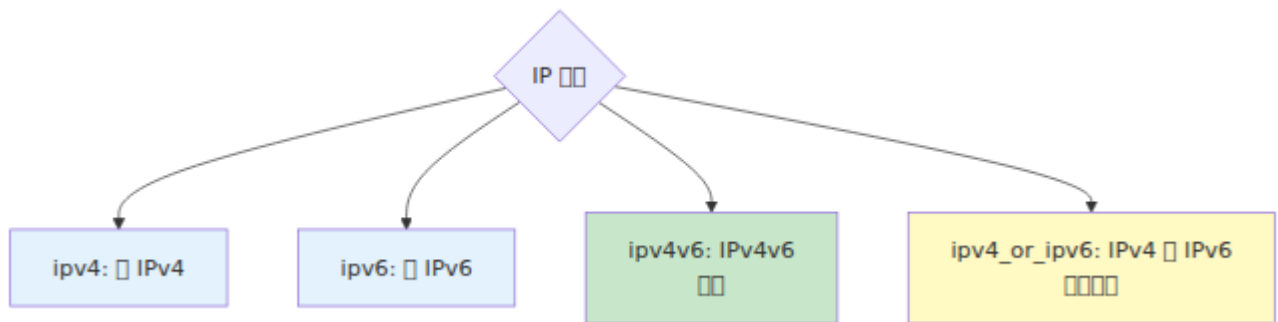


PDN 000000:

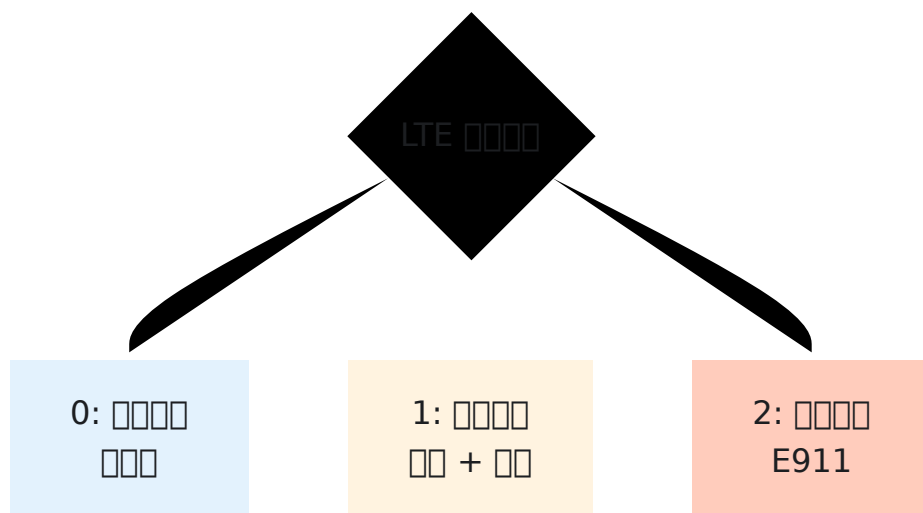


LTE

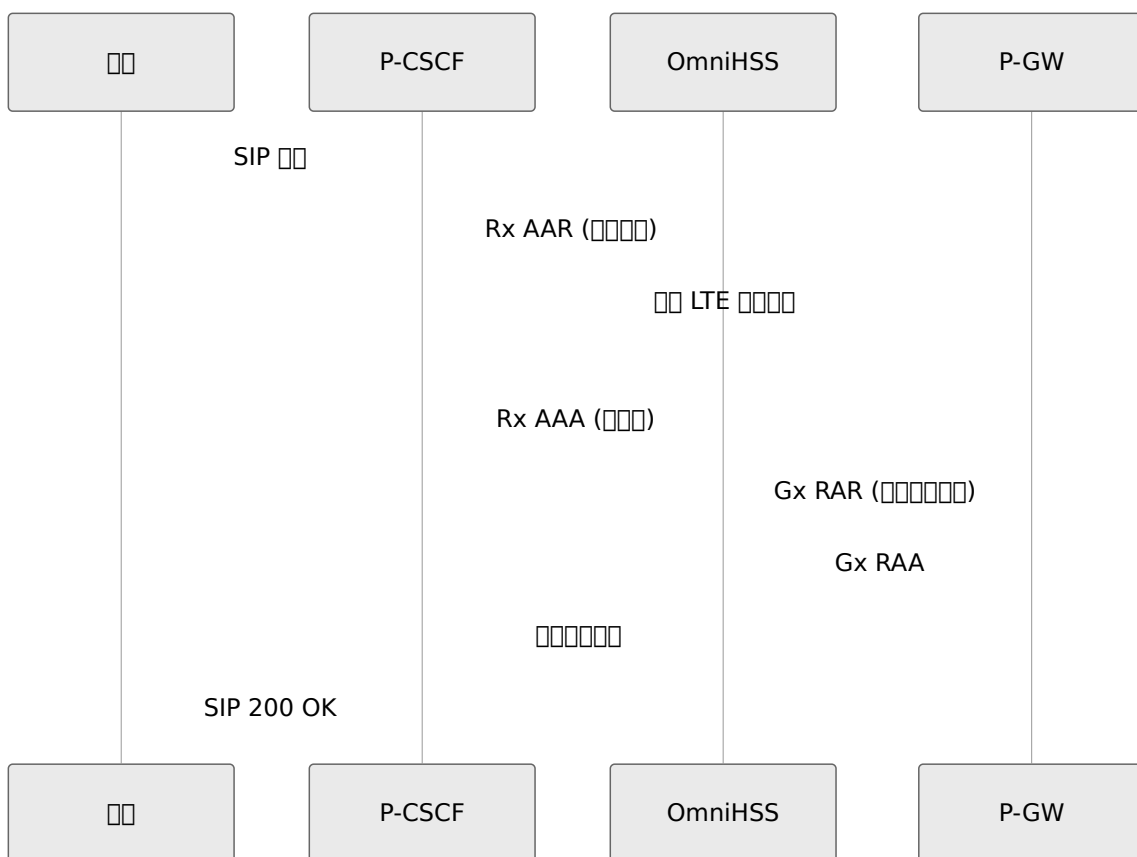
LTE VoLTE



IPv4:

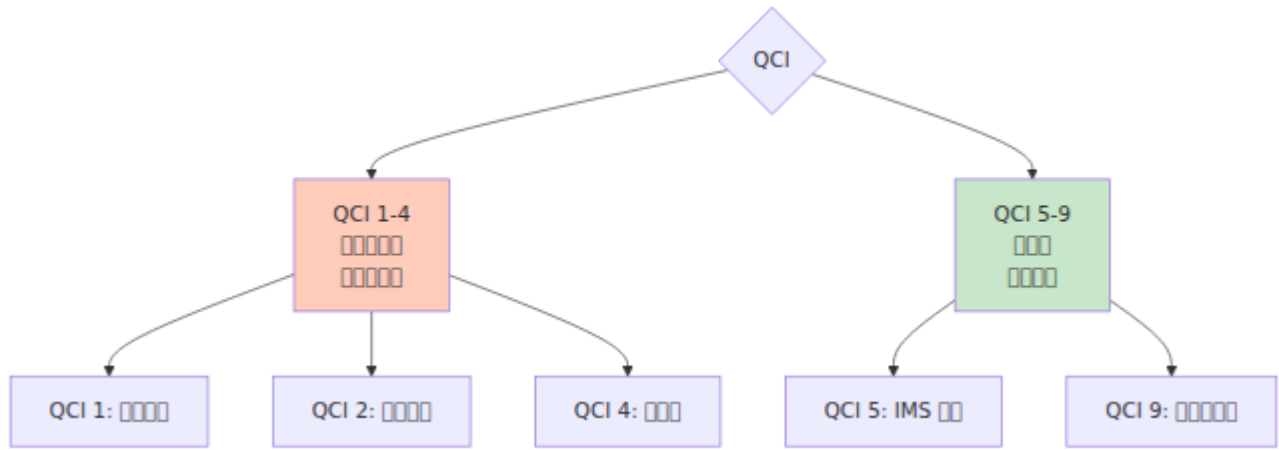


VoLTE



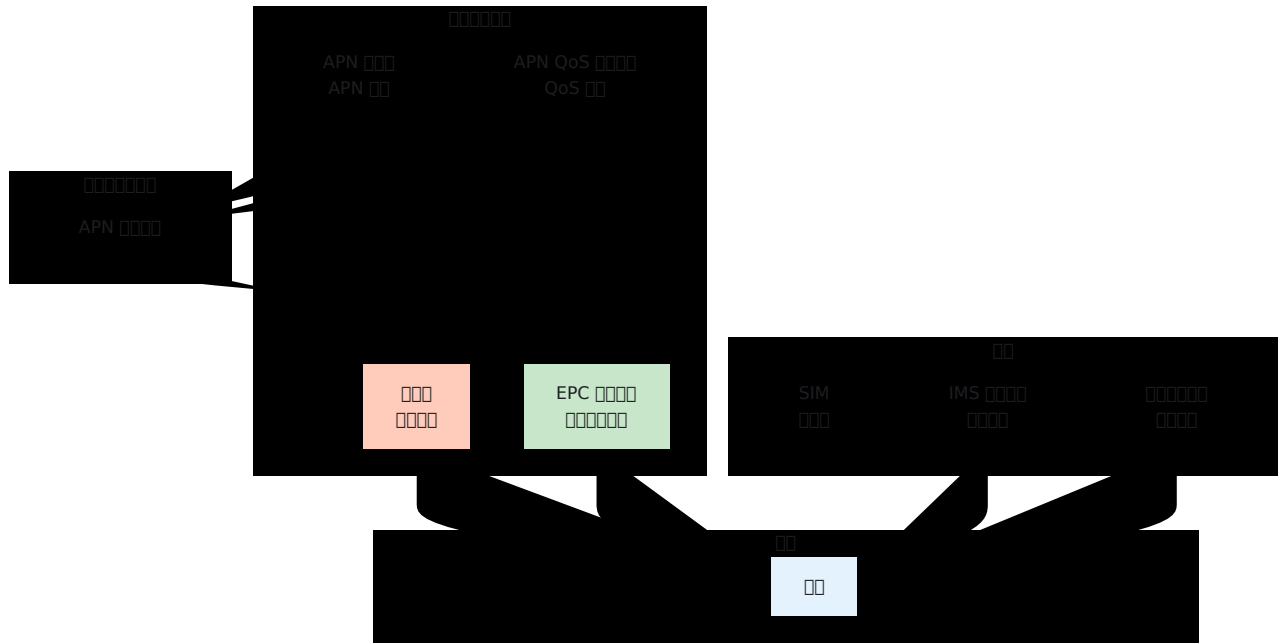
□□□□□

□□□□□□

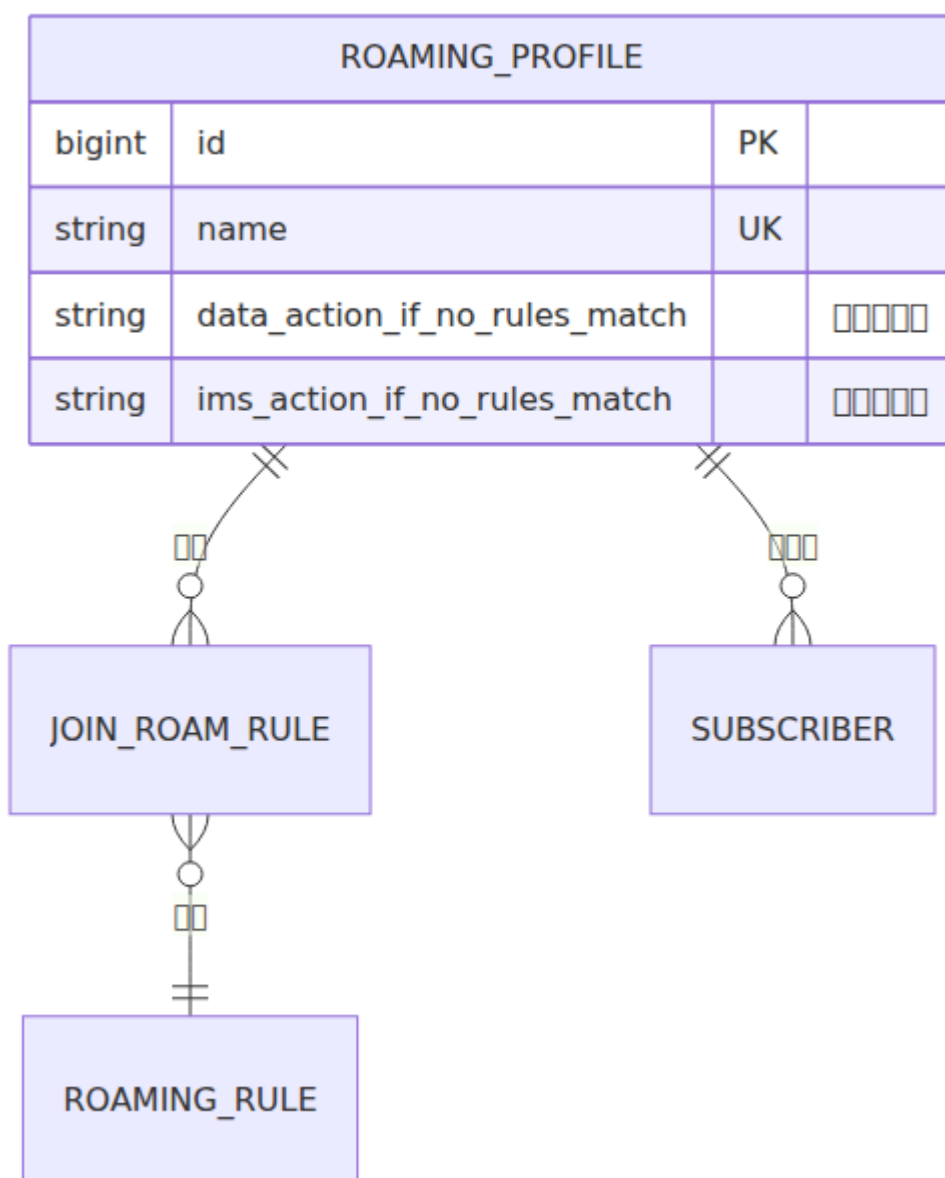


□□□□

□□□□□□□□□□□□□□□□



□□□□□□



□□□□□□

□□□□□□□



□□□□□□

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

□□□□□□ □□□□□□

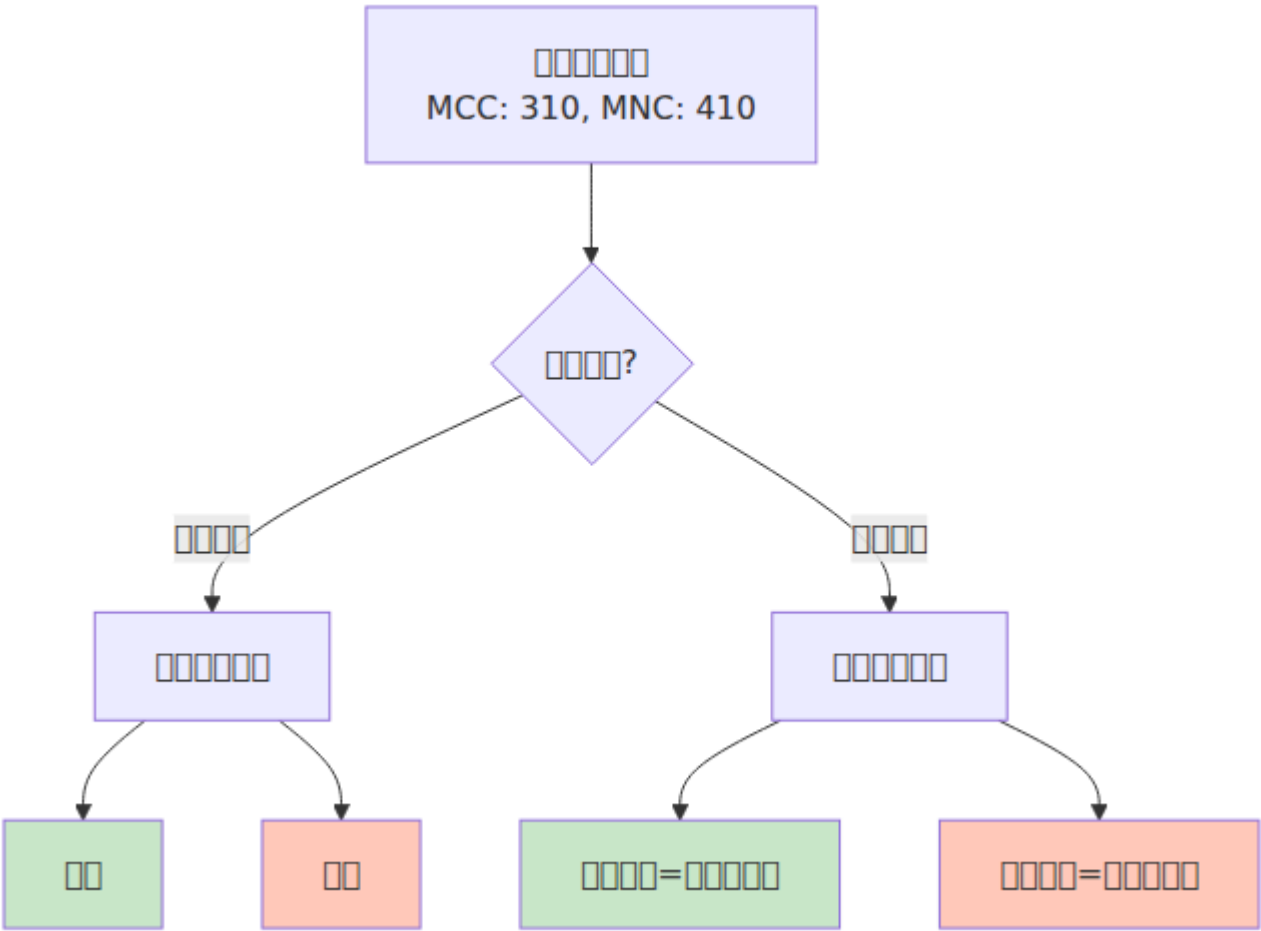
Multiple_Calls

PDN □□ + LTE □□
□□□□



□□□□□

□□□□□□



□□□□□□

S6a ULR □□

□□ IMSI □□□□

□□ EPC □□□□
+ APN □□□□

□□□□□□
□□□MME □

□□□□□□
AMBR□APN□QoS

S6a ULA □□

IMS 网络

Cx SAR 网络

网络 IMSI/MSISDN 网络

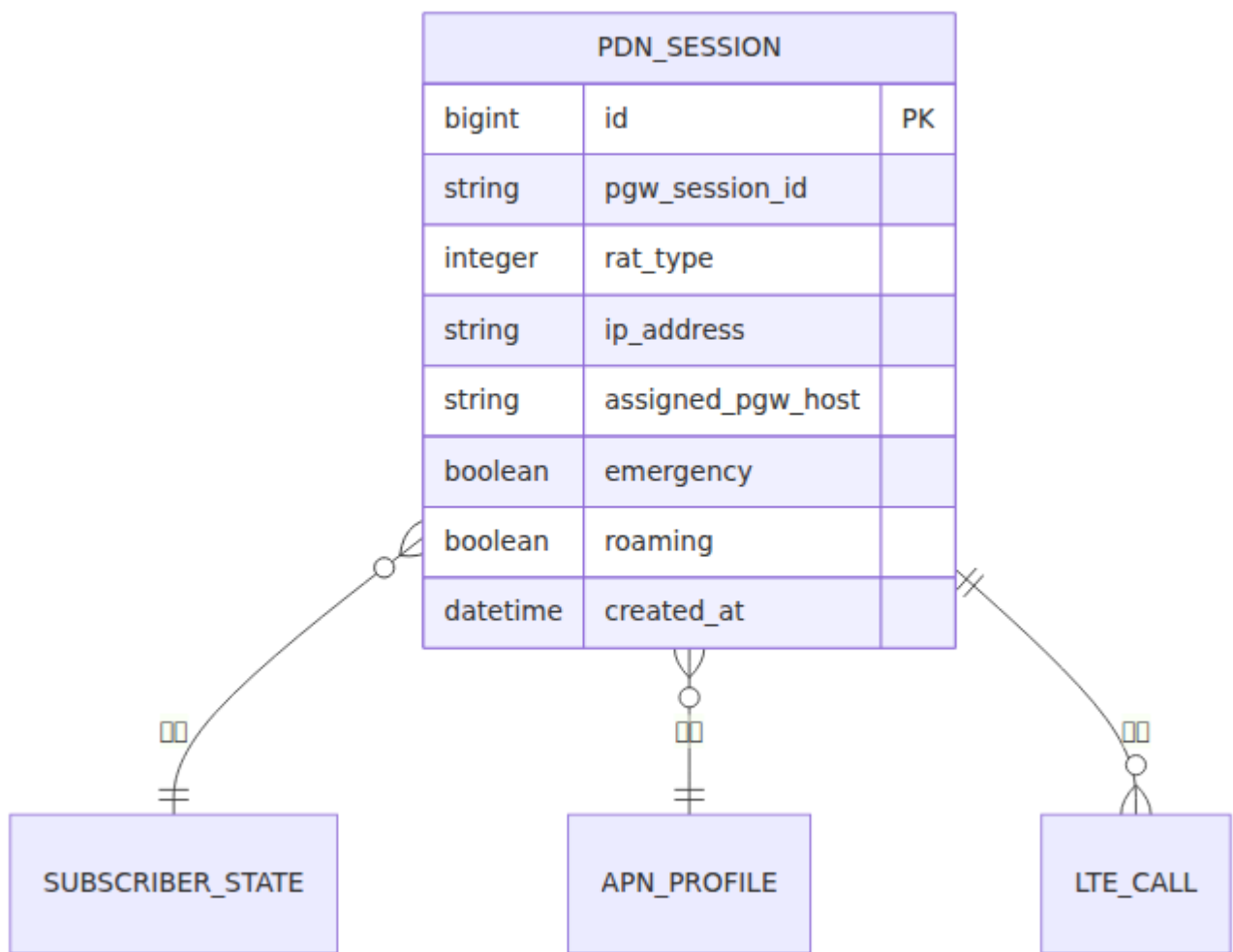
网络 IMS 网络
+ MSISDN

网络 S-CSCF
网络/网络

网络 IFC 网络
网络

网络 S-CSCF 网络

Cx SAA 网络



□ □ □ □ □ □

OmniHSS

□ □ □ □ □ □ □ □ □

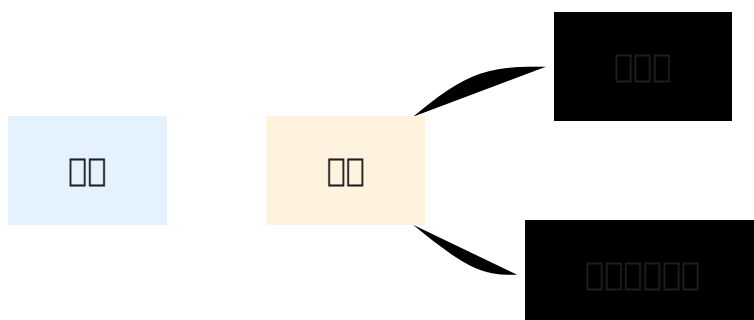


Figure 1: S6a AIR - Initial State

Figure 2: S6a AIR - Initial State

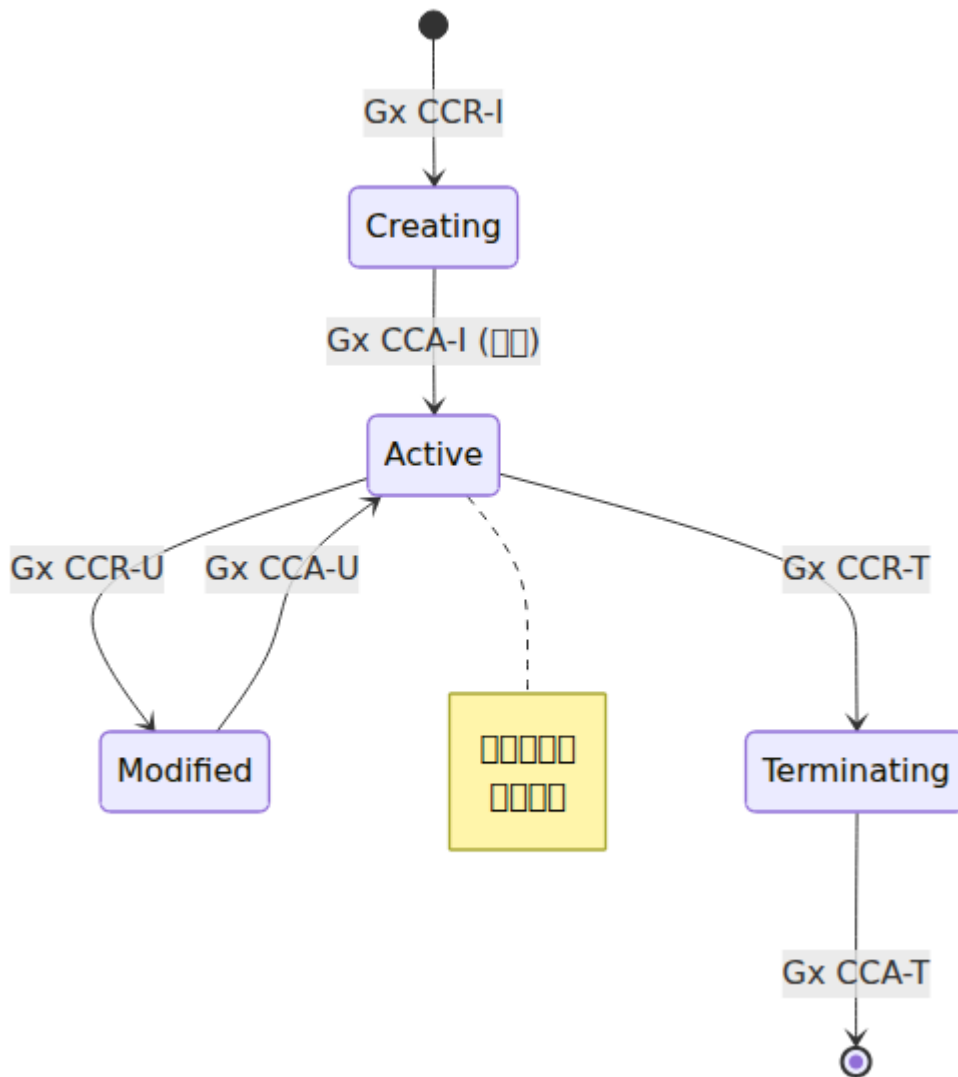
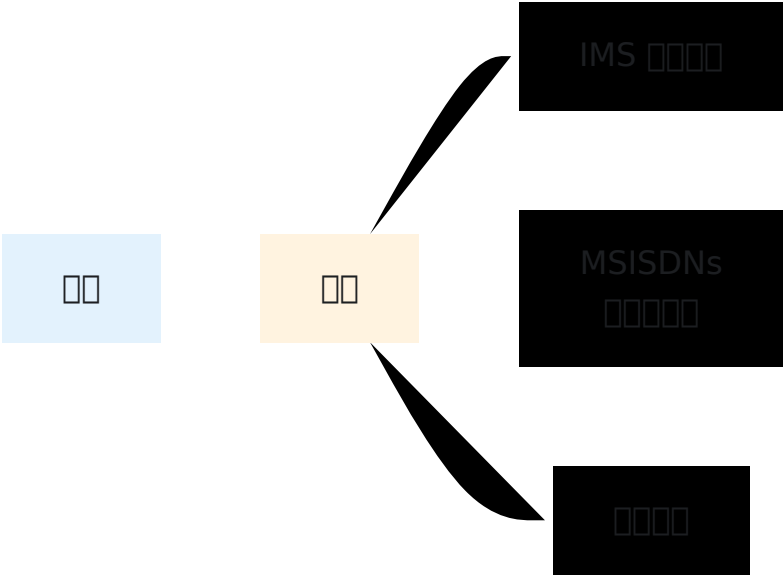


Figure 3: S6a ULR - Initial State EPC Initial State

IMS 数据



数据: Cx SAR - IMS 数据

Galera

←

-
-
- Galera
-
-
-
-
-
-
-

OmniHSS Elixir **Ecto** Ecto MariaDB Galera

Omnitouch (ONS)

MariaDB Galera

MariaDB 10.6+	Galera

データベース

データベースの種類

- データベースの種類
- HSS データベース
- データベースの種類
- データベースの種類

ONS データベース

- データベースの種類
- データベースの種類
- データベースの種類
- データベースの種類

データベース

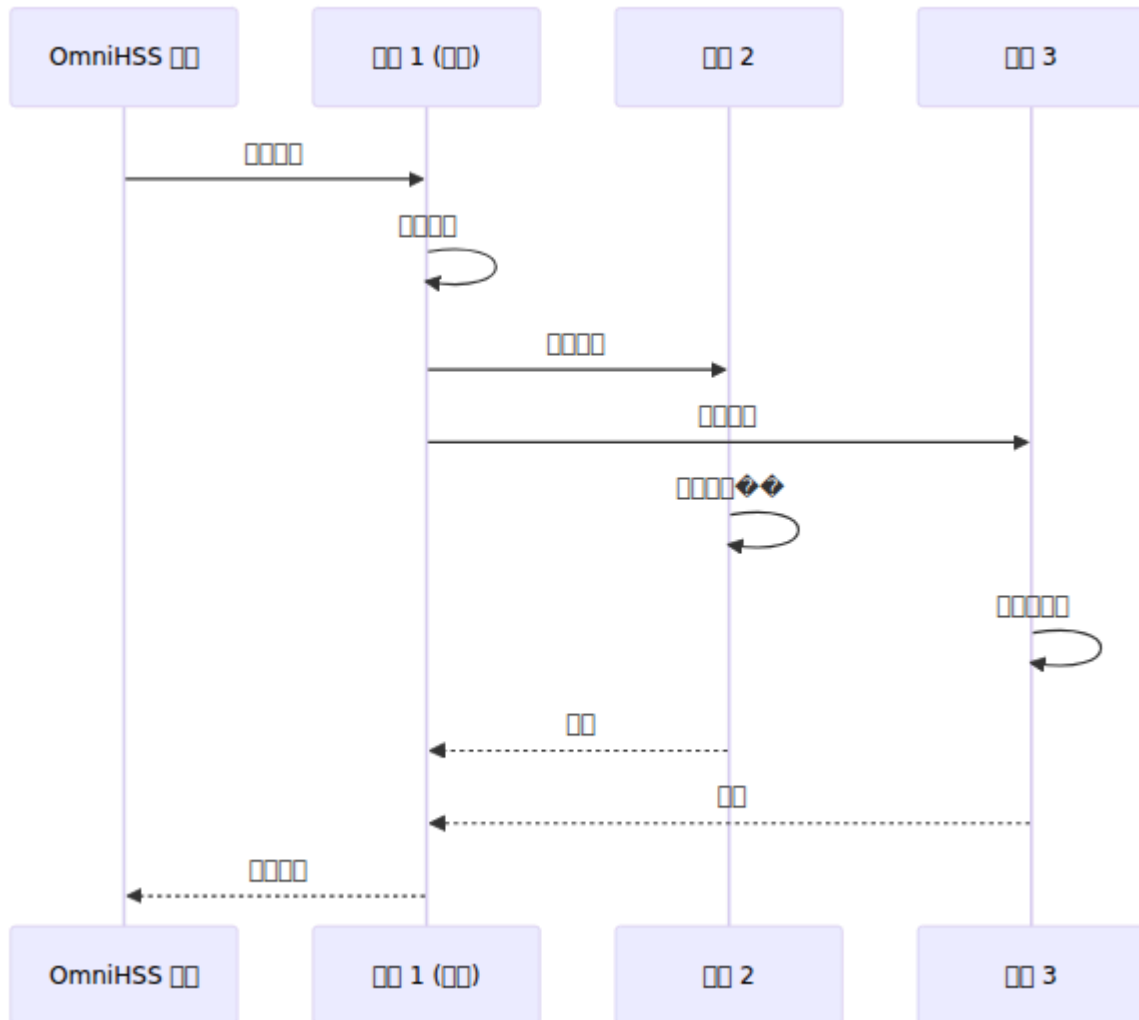
データベース MariaDB Galera データベース OmniHSS データベース Galera データベース
HSS データベース

データベース

- データベースの種類
- データベースの種類
- データベースの種類
- データベースの種類
- データベースの種類

Galera 環境構築

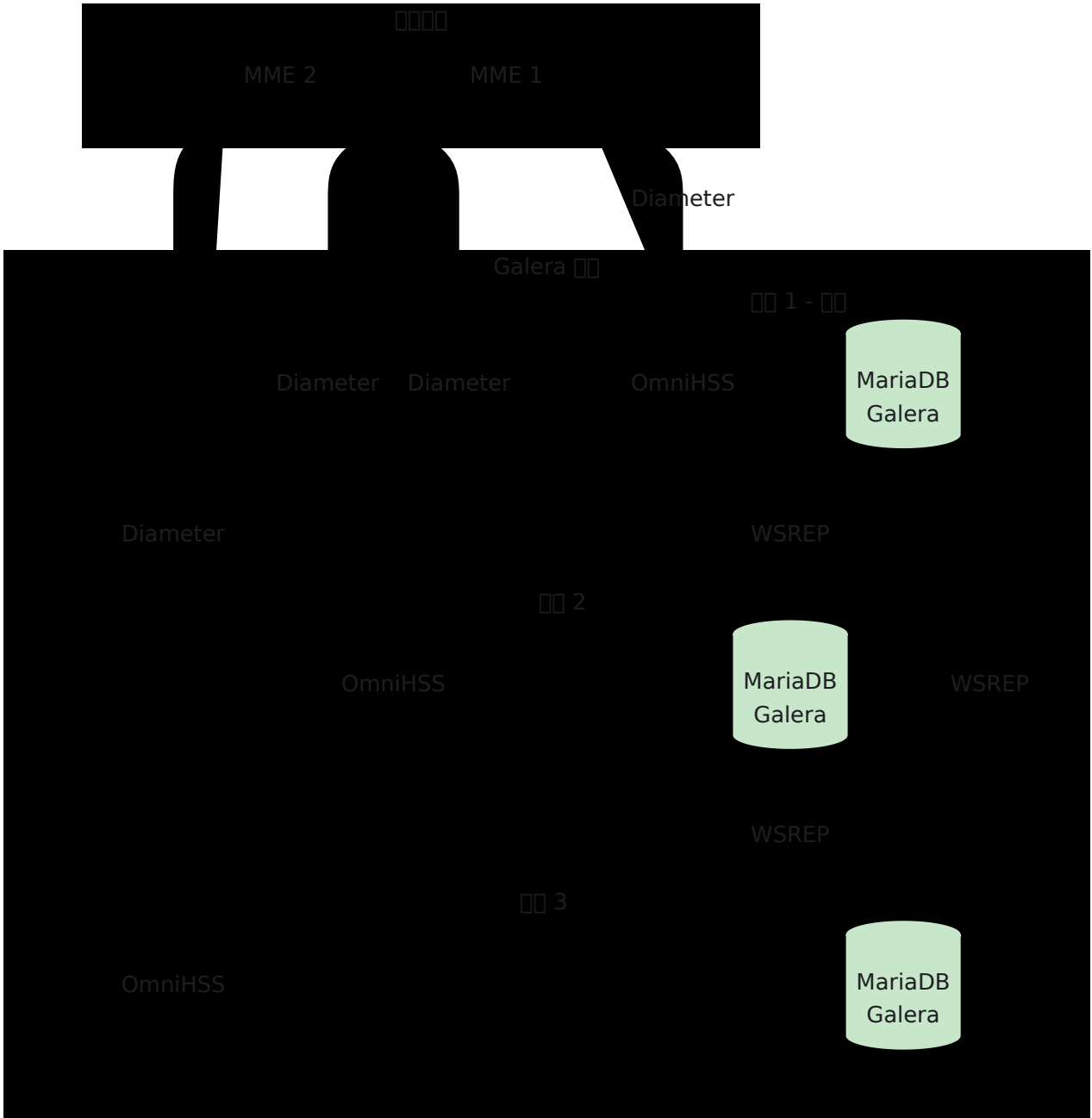
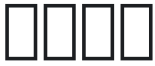
環境構築



環境構築 (WSREP)

環境構築手順

1. 環境構築に必要な SQL
2. 環境構築に必要な“コマンド”
3. 環境構築に必要な“設定”
4. 環境構築に必要な“検証”
5. 環境構築に必要な“確認”



Port List

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

Firewall Rules

```
# Allow Galera Cluster
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"             # bootstrap
    run_bootstrap: false                 # bootstrap true
    reinstall: false                     # true MariaDB
```

Galera

Galera `/etc/mysql/my.cnf`

```
[mysqld]
# 配置
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 配置
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

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

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

# 配置
wsrep_sst_method=rsync

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

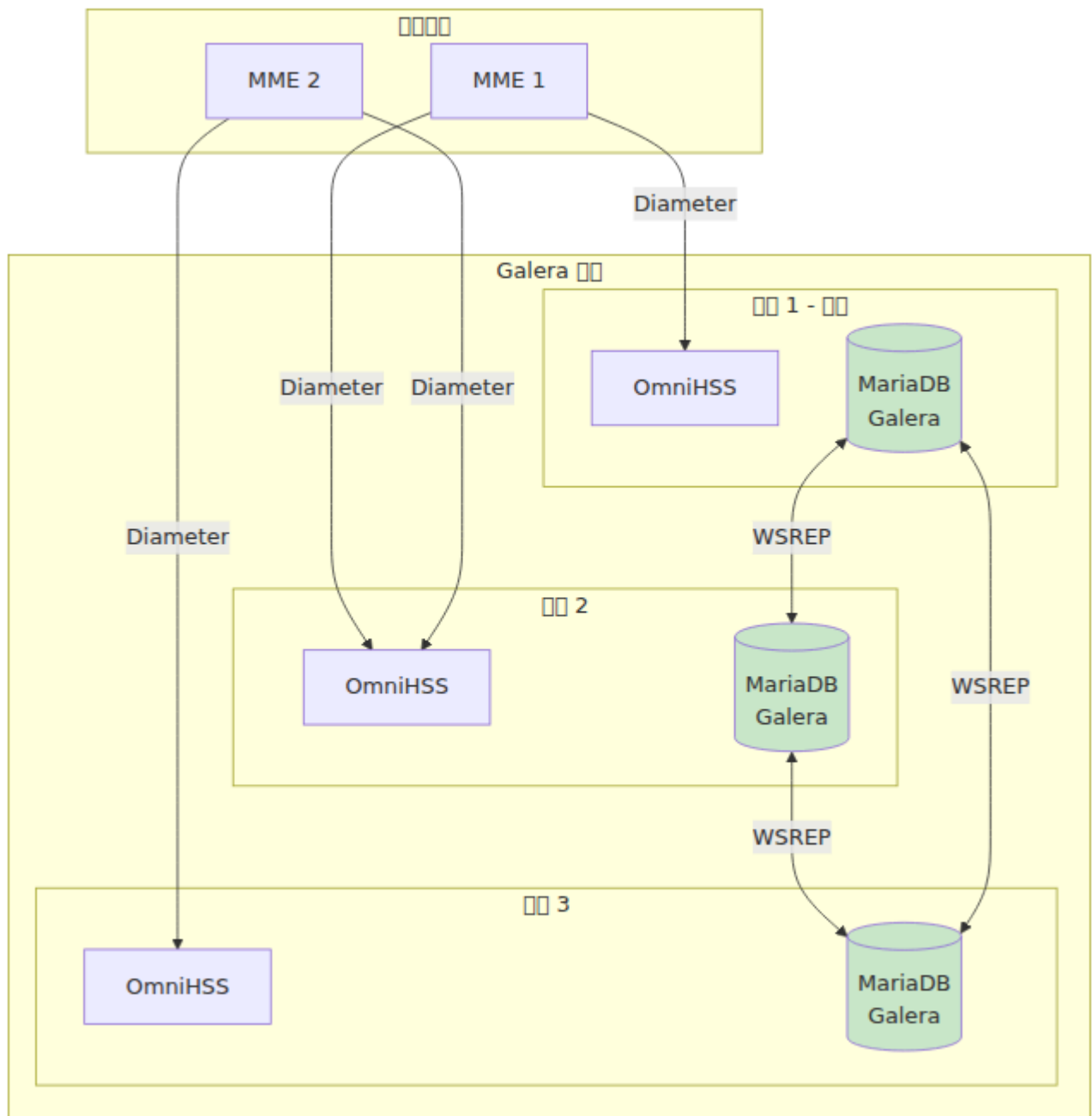

配置項目

項目	値	説明
<code>binlog_format</code>	<code>ROW</code>	ROW - 行単位でログを記録
<code>innodb_autoinc_lock_mode</code>	<code>2</code>	2 - 自動増分ロックモード
<code>wsrep_on</code>	<code>ON</code>	WSREP を有効にする
<code>wsrep_provider</code>	libgalera 3.26.0	Galera 3.26.0
<code>wsrep_cluster_name</code>	"omnihss_galera"	クラスタ名
<code>wsrep_cluster_address</code>	<code>gcomm://ip1,ip2,ip3</code>	クラスタメンバーの IP アドレス
<code>wsrep_sst_method</code>	<code>rsync</code>	rsync を使用して SST を実行
<code>wsrep_node_address</code>	この IP	このノードの IP アドレス
<code>wsrep_node_name</code>	このノード名	このノードの名前

インストール

インストール

Galera 3.26.0 をインストール



1. 1. 1. 1.

1. 1. 1. 1.

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

2. 실행 Ansible 실행

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

3. 확인사항

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

4. 실행

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

Grastate 실행

실행 `/var/lib/mysql/grastate.dat` 실행

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

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


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

2. 確認 seqno

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

3. 確認

```
# 確認
systemctl start mysql
```

確認

確認

確認

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

-- 確認
SHOW STATUS LIKE 'wsrep_cluster_status';

-- 確認
SHOW STATUS LIKE 'wsrep_local_state_comment';

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

状态

变量	值	说明
wsrep_cluster_size	3	集群中的节点数
wsrep_cluster_status	Primary	集群状态
wsrep_local_state	4	本地节点状态
wsrep_local_state_comment	Synced	本地节点状态描述
wsrep_ready	ON	是否准备好
wsrep_connected	ON	是否已连接

节点

节点	ID	状态
节点1	1	Primary
节点2/节点3	2	Secondary SST
节点4	3	Secondary
节点5	4	Secondary

Prometheus 配置

配置 MariaDB Galera 节点OmniHSS 节点和 Galera 节点

環境構築

MySQLのインストール

MySQLのインストール

確認

```
# MySQLのインストールを確認
tail -f /var/log/mysql/error.log

# MySQLのインストールを確認
mysql -e "SHOW STATUS LIKE 'wsrep_on';"
```

確認

- MySQLのインストールを確認 4567 4568 4444
- wsrep_cluster_address IP
- AppArmor
- UUID

確認

```
# AppArmorのインストールを確認
systemctl status apparmor
# AppArmorのインストールを確認
systemctl stop apparmor && systemctl disable apparmor

# AppArmorのインストールを確認
ss -tlnp | grep -E '4567|4568|4444|3306'
```

MySQLのインストール

MySQLのインストールを確認 wsrep_cluster_status non-Primary

確認

- MySQLのインストールを確認

- 確認

```
-- 確認
SET GLOBAL wsrep_provider_options='pc.bootstrap=YES';
```

SST 確認

確認

```
# 確認
df -h /var/lib/mysql

# rsync 確認
ps aux | grep rsync
```

確認

- 確認
- rsync 確認
- 確認 4444

確認

wsrep_local_state_comment Donor/Desynced

SST 確認


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

Grastate 再構築

MySQL MariaDB の Grastate を再構築

手順

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

# MySQL の SST を実行
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

- MySQL の Grastate
- MySQL の I/O
- MySQL の Grastate

確認

- MySQL の Grastate
- MySQL の Grastate

- □□□□□□□□

Diameter

←

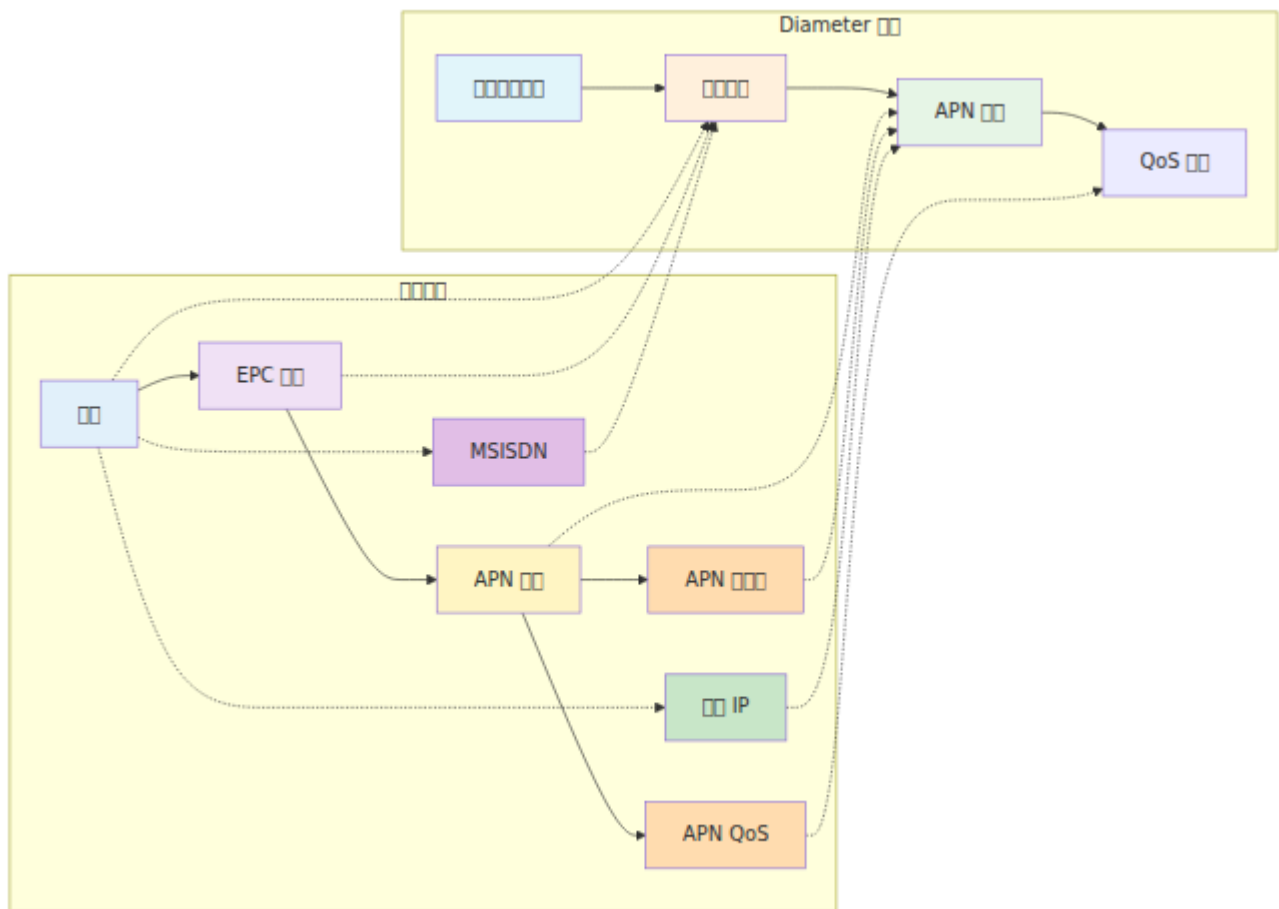
mermaid Diameter OmniHSS

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

(S6a ULA)

HSS LTE MME Diameter AVP

□□□□□



□□□□□□

□□□□□	□□	□
subscriber.enabled	true/false	Sub Sta
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'	PD
apn_qos_profile.qci	9	Qc Id
apn_qos_profile.allocation_retention_priority	8	Pr Le

項目名	値	説明
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 Address (If enabled)
static_ip.ipv6_static_ip	'2606:4700::1111'	Static IPv6 Address (If enabled)

注意事項

1. **AMBR** 単位は kbps であり、Diameter 側は bps 単位で 1000
2. **IP** 0=IPv4, 1=IPv6, 2=IPv4v6, 3=IPv4_or_IPv6
3. **enabled**: true → 0 (SERVICE_GRANTED), enabled: false → 1 (OPERATOR_DETERMINED_BARRING)
4. 静的 IP アドレスは APN ごとに 0, 1, 2...
5. **IP** static_ips 静的 IP アドレス

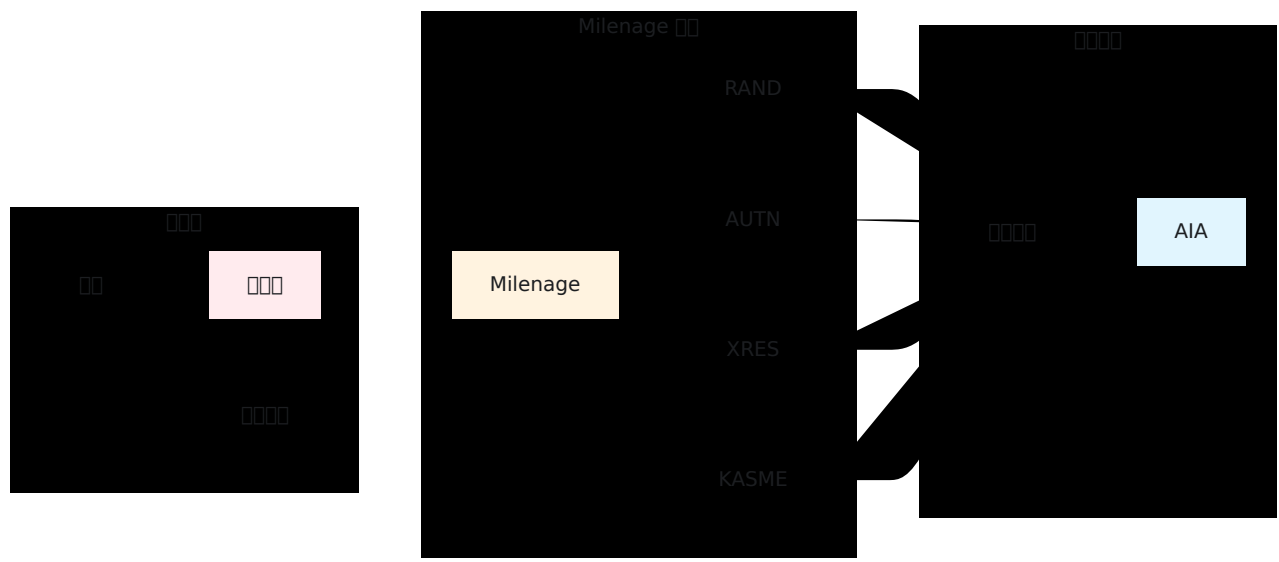
関連項目

- 网络选择规则 PLMN 中 `roaming_profile.roaming_rules`
- 网络选择 `subscriber.enabled == true`
- 网络 APN 网络选择规则 IMS 网络选择 IMS APN

网络选择 (S6a AIA)

网络选择 LTE/EPC 网络选择规则

网络选择



网络选择

1. 网络选择规则 `key_set` 中
2. **SQN** 网络选择规则
3. **Milenage** 网络选择 3GPP TS 35.206 - 网络选择
4. **KASME** 网络选择 CK||IK 网络选择 KDF 网络选择 TS 33.401 网络选择

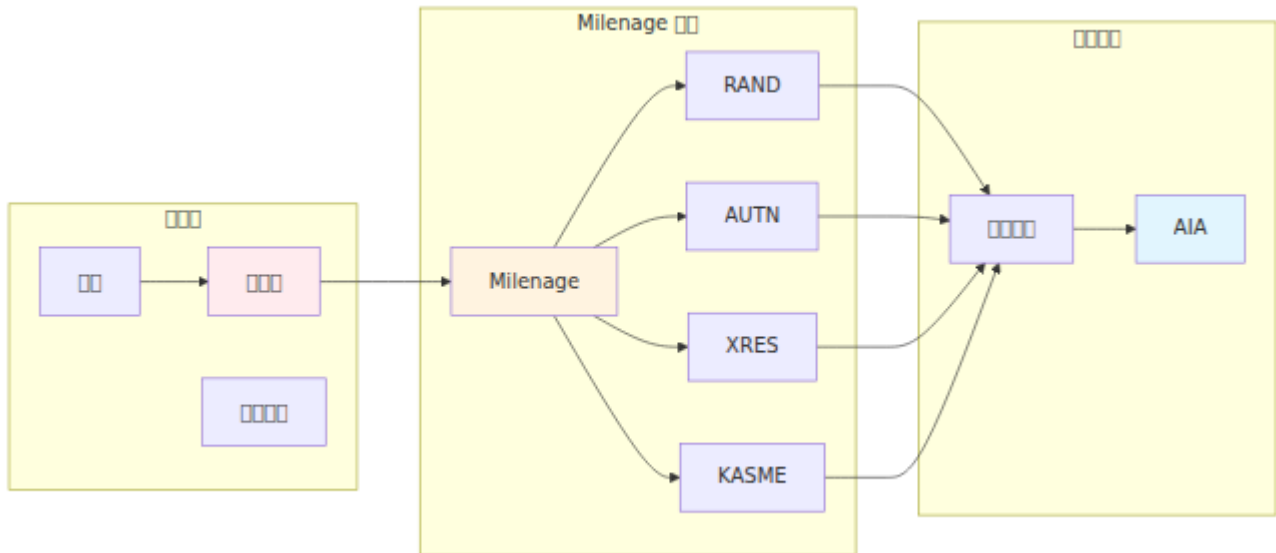
网络选择

- 网络选择 SQN 网络选择
- Ki/OPc 网络选择 HSS 网络选择
- AUTN 网络选择 SQN 网络选择 AMF 网络选择
- Milenage 网络选择 UE 网络选择

AAA (Cx SAA)

AAA HSS IMS AAA S-CSCF

AAA



AAA

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

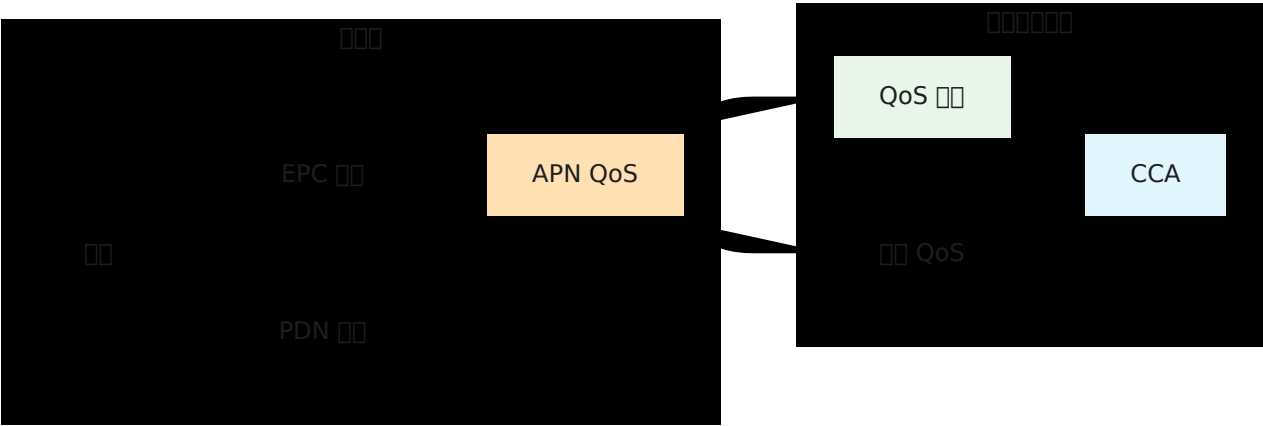
IFC AAA

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

AAA (Gx CCA)

AAA PCRF AAA PGW

概要



手順

1. `pdn_session` 取得
2. **QoS** 取得 (APN QoS から QCI 取得)
3. **QoS** 設定 (QoS 設定)
4. **CC**-**QoS** 設定 (INITIAL (1) UPDATE (2) TERMINATION (3))

メッセージ

- `INITIAL_REQUEST` (PDN 設定)
- `UPDATE_REQUEST` (PDN 更新)
- `TERMINATION_REQUEST` (PDN 削除)

Sh UDA (Sh UDA)

HSS から Sh へ AS へ送信される

概要

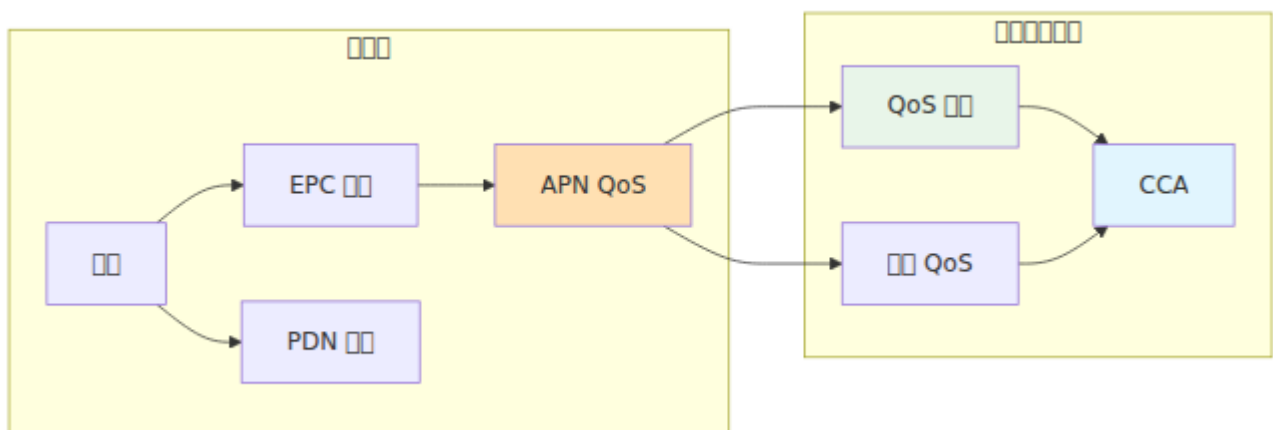


55

1. subscriber_state.sh_repository_data XML
- 2.
3. IMS
- 4.

ME (S13 ECA)

ME 00000000 EIR 000000 MME 00 IMEI 000



□□□□□

1. **IMEI** □□□□□□□□□□□□□□□□□□
2. □□ **TAC** □□□□□□□□□□□□□□□□ 8 □□□□
3. □□□□□□□□ IMEI □□□□□□□□□□
4. □□□□□□
 - 0 = □□□□□□□□□□
 - 1 = □□□□□□/□□□
 - 2 = □□□□□□□□□□
 - 5 = □□□□□□□□□□

111

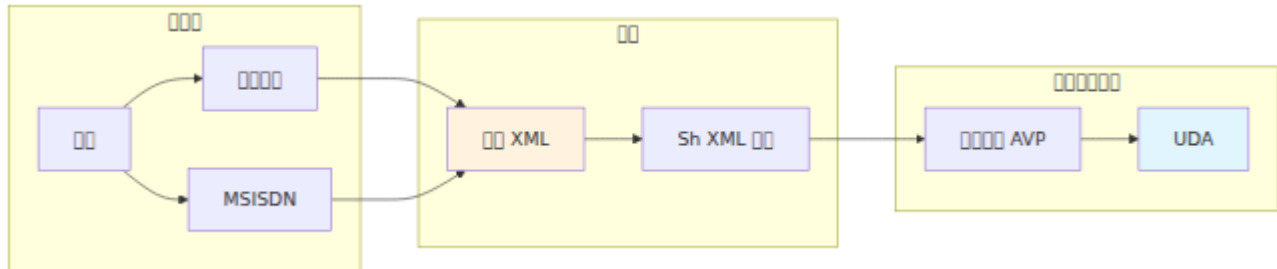
- IMEI
- TAC
-
-

Diameter AVP

```
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** - 國際移動設備識別碼
- **ICCID** - 國際卡識別碼
- **PDN** 類型 - 網路數據網路
- **IP** 地址 - IP 地址
- **EIR** 類型 - IMEI 類型

← 返回首頁 | **API 說明** → | 關於我們 →

OmniHSS 部署架构图

← 部署图

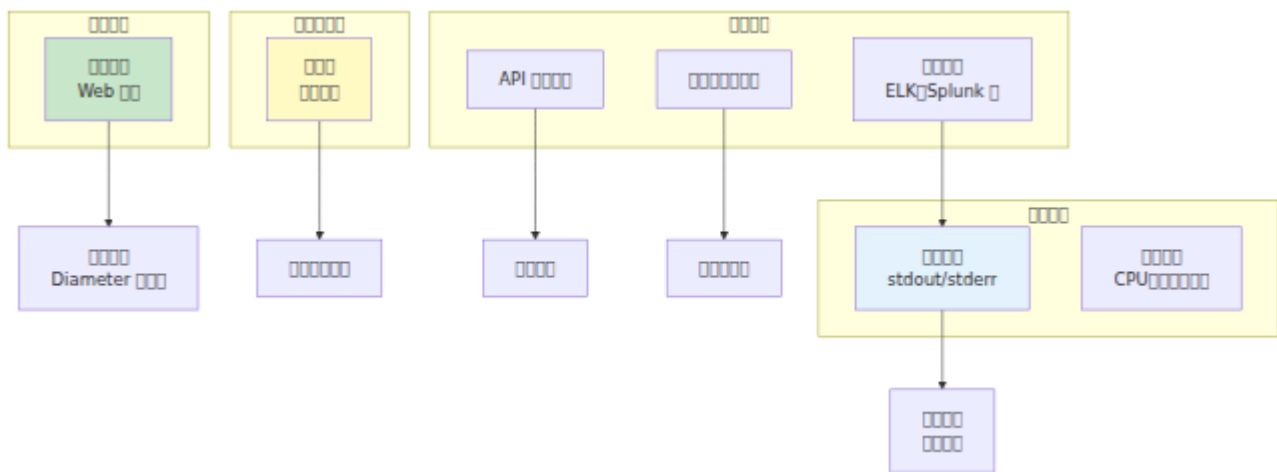
部署

- 部署
- 部署图
- 部署图
- 部署
- 部署图
- 部署图
- 部署

部署图

OmniHSS 部署图展示了 OmniHSS 的部署架构。

部署图



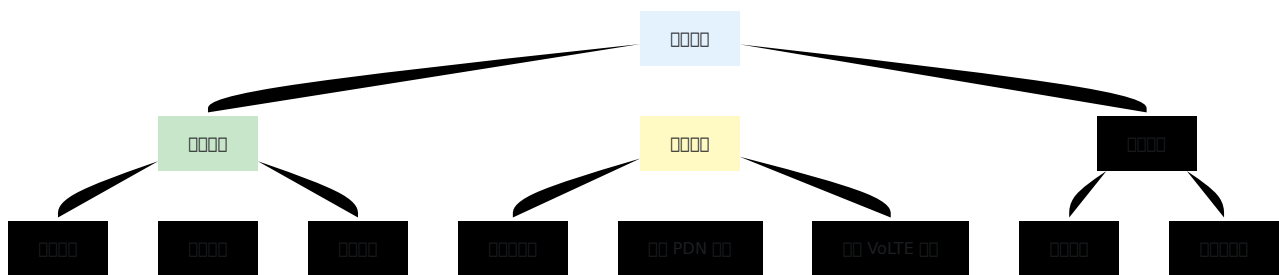
IMS Overview

IMS Overview

IMS Overview

URL: [https://\[hostname\]:7443/overview](https://[hostname]:7443/overview)

IMS Overview



IMS Overview

Item	Item	Item
Item	Item	Item
Item	MME Item	Item
PDN Item	PDN Item > 0	Item
IMS Item	S-CSCF Item	Item
Item	Item > 0	VoLTE Item

IMS Overview

IMS Overview

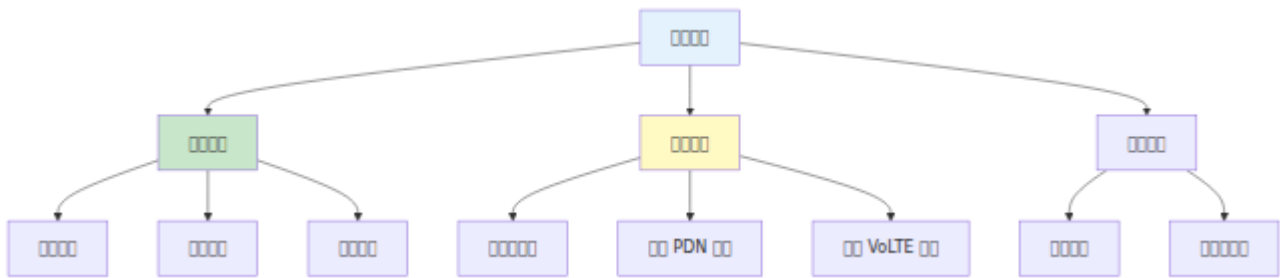
1. Item
2. Item
3. Item

4. Diameter 프로토콜

Diameter 프로토콜

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

개요



구성 요소

구성 요소 설명

구성 요소	기능	설명
MME	1	모바일 네트워크 LTE 망
P-GW	1	패킷 게이트웨이
S-CSCF	1	IMS 망
P-CSCF	1	VoLTE 망
I-CSCF	1	IMS 망
AS	1-1	애플리케이션 서버

구성 요소

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

개요

項目	単位	測定項目	測定結果
通話品質	単位 Erlang	通話品質	> 90% 良好
通話品質	通話品質	< 80%	> 90%
通話品質	通話品質	N/A	通話品質

通話品質

通話品質

通話 SQL 通話品質

通話品質

通話品質

- 通話品質
- 通話品質
- IMS 通話品質

通話品質

通話品質

- 通話 PDN 通話品質
- 通話 VoLTE 通話品質
- 通話 APN 通話品質 PDN 通話

通話品質

通話品質

- 通話品質 MCC-MNC 通話品質
- 通話品質 PLMN 001-001 通話

- 0000000000000000

0000

0000000000

- 0000000000000000
- 000 MME 00000000
- 0000000000000000

0000000000

00000000000000

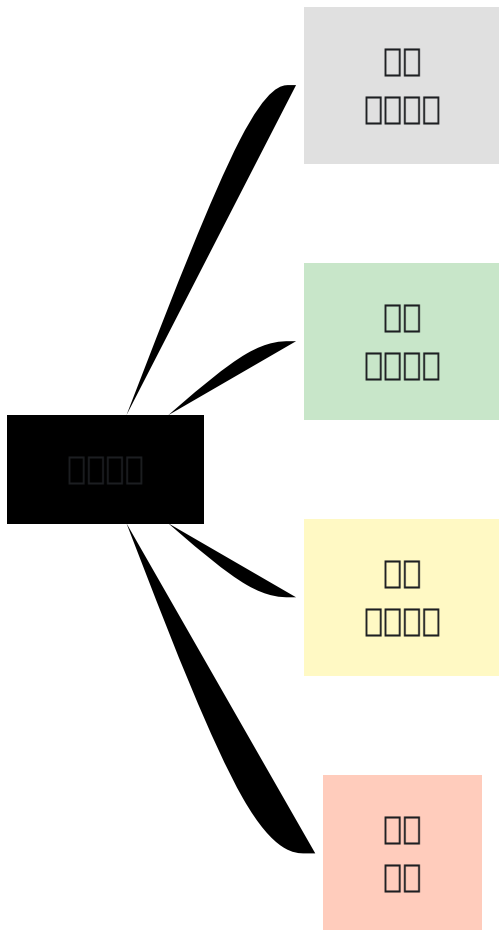
- 00000000000000
- 000 00000000
- 0000000000
- 00000000000000

000000

000000

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
```

□□□□:

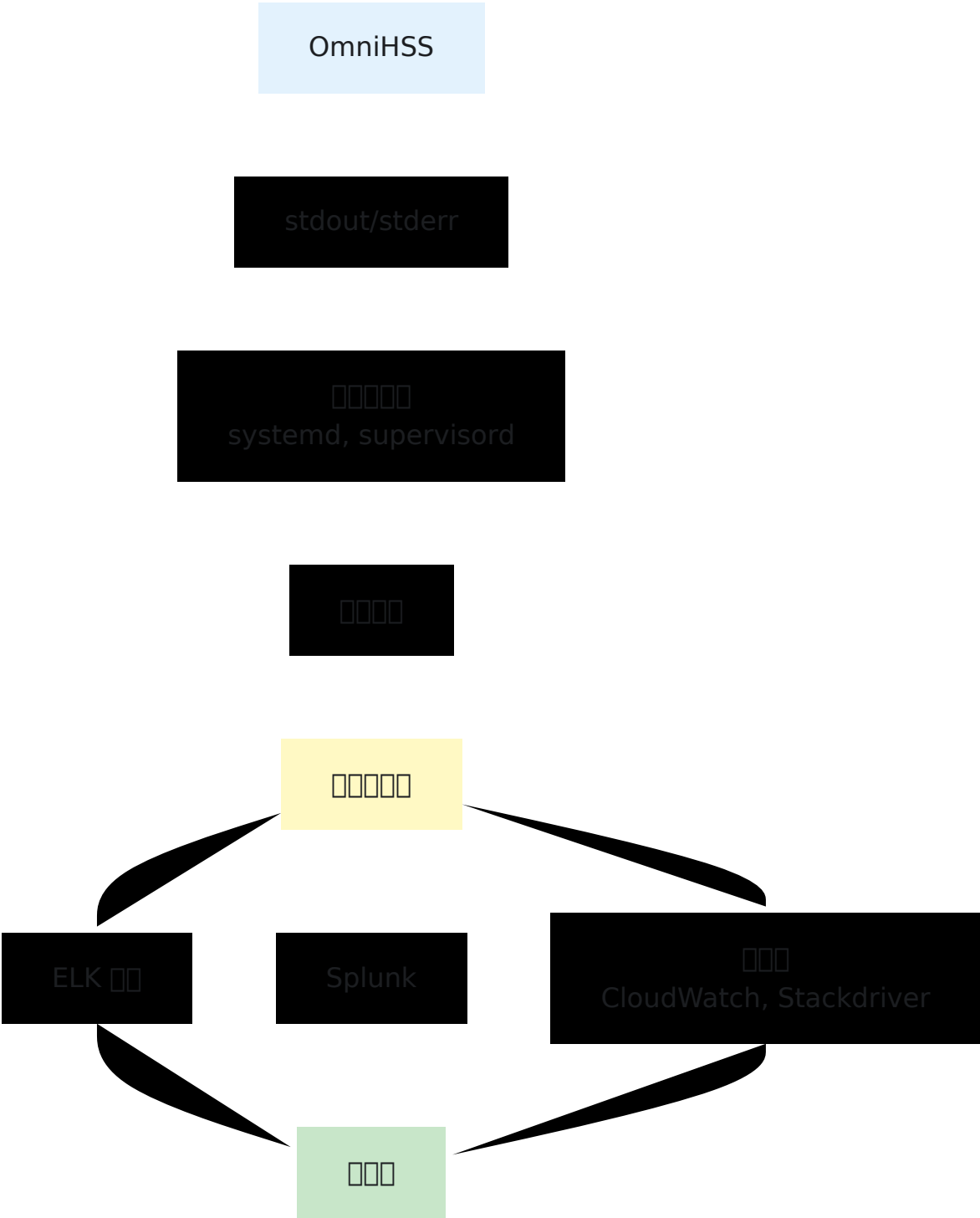
```
[info] Database connection established
[error] Database connection lost: timeout
[error] Database query failed: deadlock detected
```

□□□□:

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

0000

0000000000000000



□□□□□□

□□□□□□

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

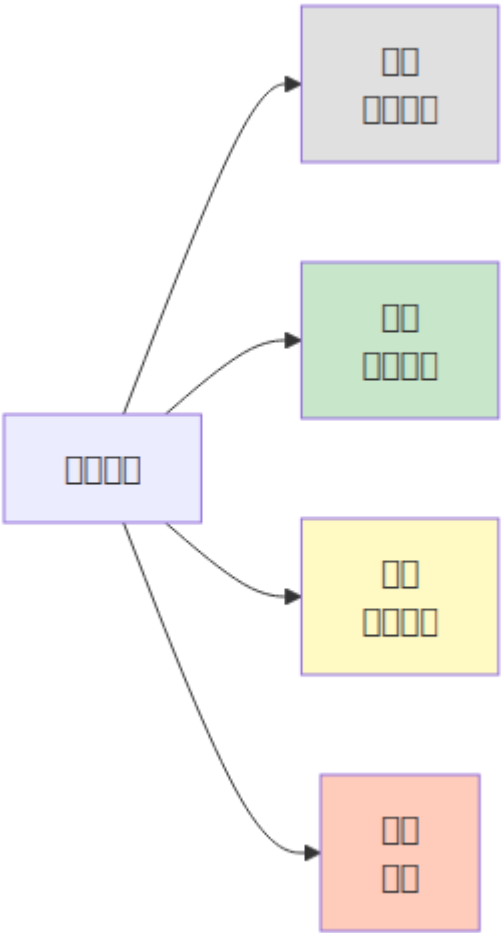


Table 1: KPI

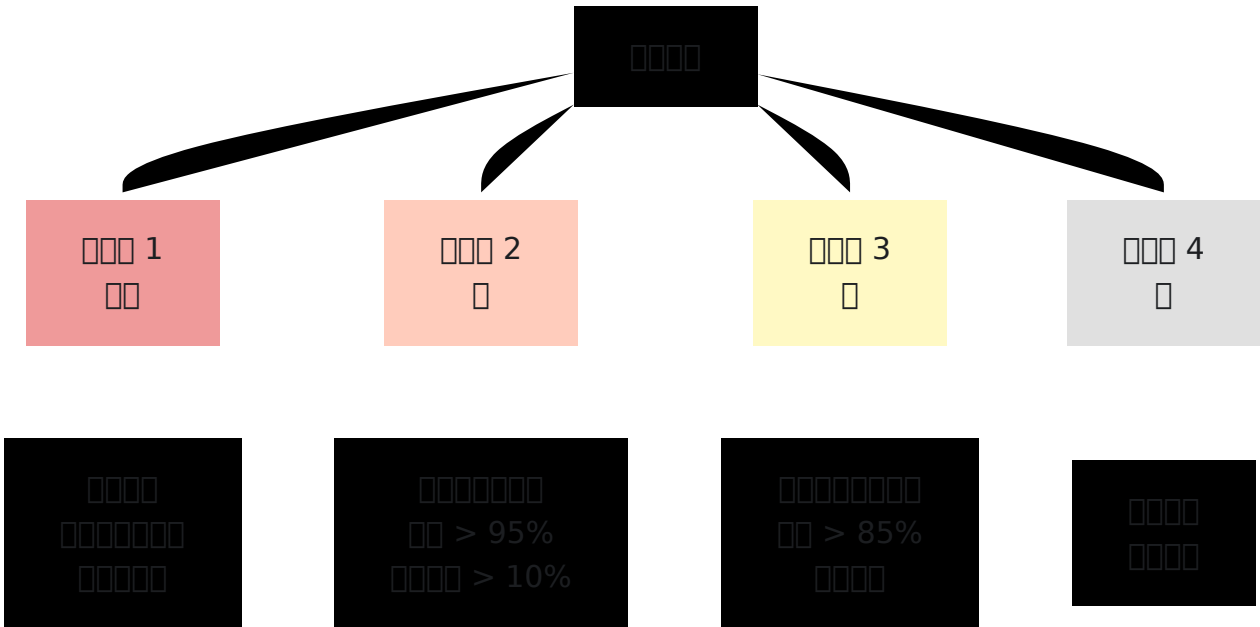
KPI	Target	Actual	Score
Availability	99.99%	< 99.95%	< 99.9%
Diameter Success Rate	99.9%	< 99.5%	< 99%
Latency	> 99%	< 99%	< 95%
Diameter Latency	< 100ms	> 200ms	> 500ms
Throughput	< 50ms	> 100ms	> 500ms
Drop Rate	< 0.1%	> 0.5%	> 1%

Table 2: KPI

Target	Actual	Score
Availability	99.99%	80% Success
PDN Latency	99.9%	70% Success
Throughput	99.9 MB	80% Success
Drop Rate	0.1%	80% Success

□□□□

□□□□□



□□□□

□□□□ (P1)

□□□□□:

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

□□ **Diameter** □□□□□:

- □□□□□□
- □□: □□□□□□□□□□

□□□□□:

- □□□□□ SQL □□□

- 時間: 10分程度

課題 (P2)

課題 Diameter 設定:

- MME 設定
- P-GW 設定
- S-CSCF 設定
- 時間: 15分程度

結果:

- 時間 > 95%
- 時間: 10分程度

結果:

- 10% 程度
- 時間: 10分程度

課題 (P3)

課題:

- 設定
- 設定
- 時間: 1分程度

結果:

- 時間 > 85%
- 時間: 10分程度

結果:

- 時間 > 1%
- 時間: 10分程度

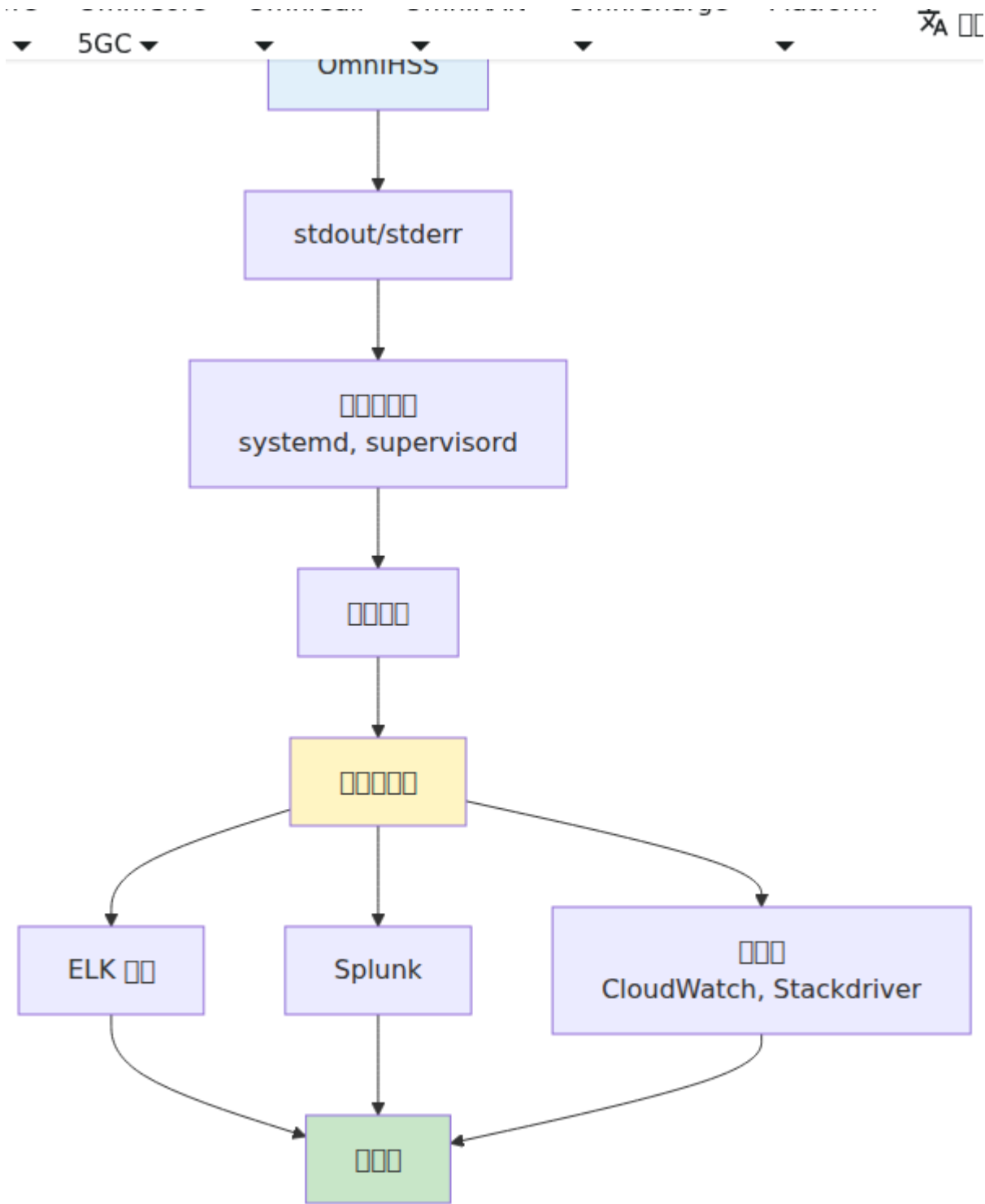
□□□□□ (P4)

□□□□:

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

□□□□:

- □□□□□□□□□□
- □□: □□□□□□□



□□□□□□

□□□□

- ☐ □□□□□□□ - □□□□□□
- ☐ □□ Diameter □□ - □□□□□□□□
- ☐ □□□□□□ - □□□□□□□□□□□□
- ☐ □□□□□□ - □□ 24 □□□□□□□□
- ☐ □□□□□□□□

□□□□

- ☐ □□□□□□ - □□□□
- ☐ □□□□□□ - □□□□
- ☐ □□□□□□□ - □□□□□□□
- ☐ □□□□□ - □□□□
- ☐ □□□□□□ - □□□□□□

□□□□

- ☐ □□□□□□ - □□ 6 □□□
- ☐ □□□□□□ - □□□□□□
- ☐ □□□□ - □□□□□□□□
- ☐ □□□□ - □□□□□□□
- ☐ □□□□□□ - □□□□□□□□

OmniHSS MSISDN IMSI

←

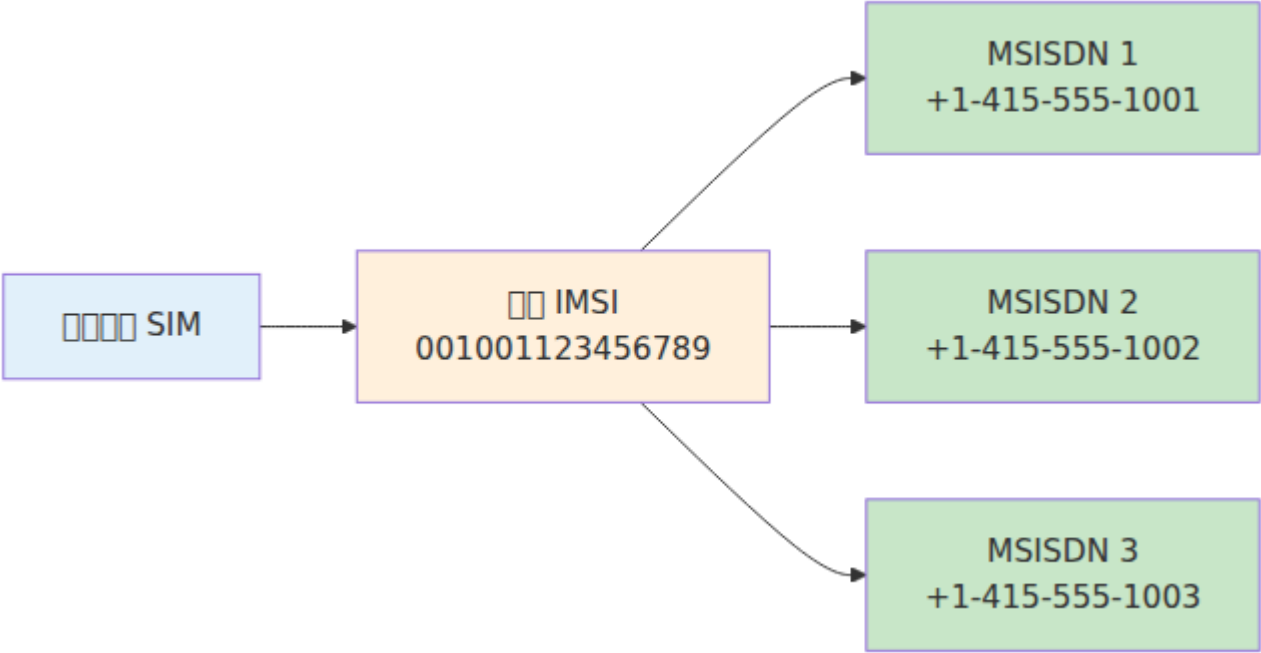
-
- MSISDN
- IMSI SIM
-
-
-

OmniHSS

MSISDN

IMSI →

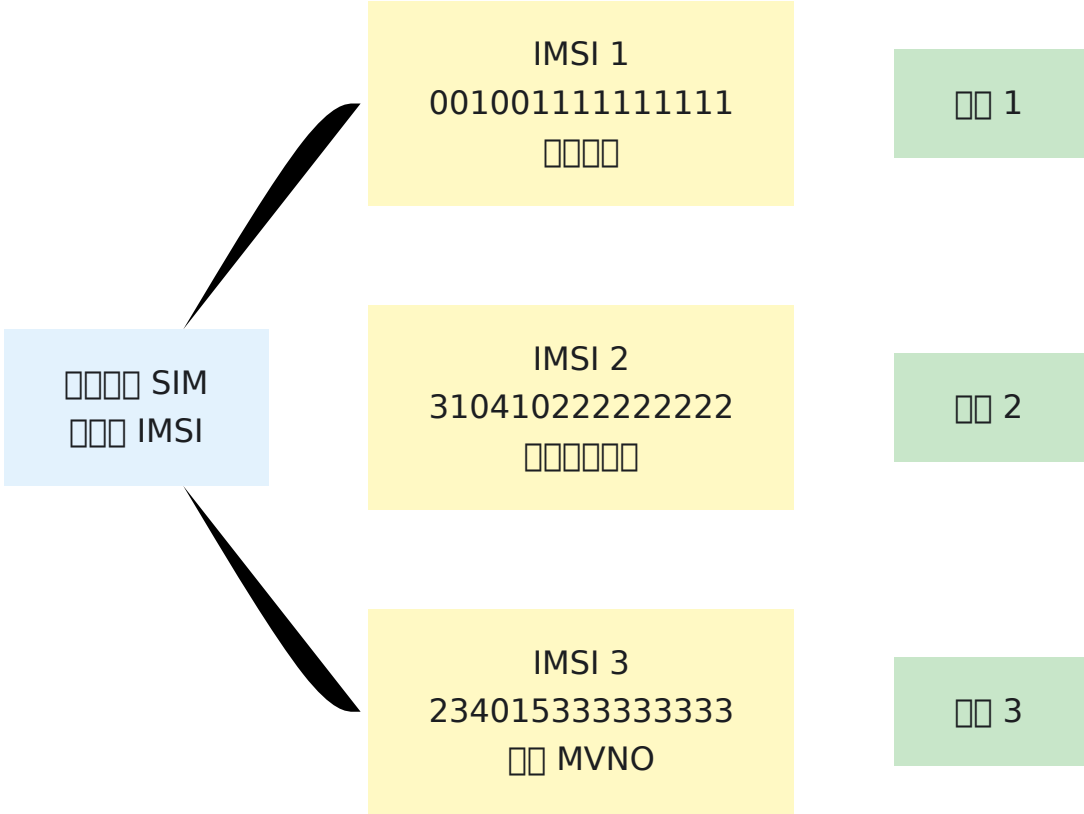
IMSI MSISDN



IMSI SIM

SIM → IMSI

SIM IMSI MVNO



MSISDN

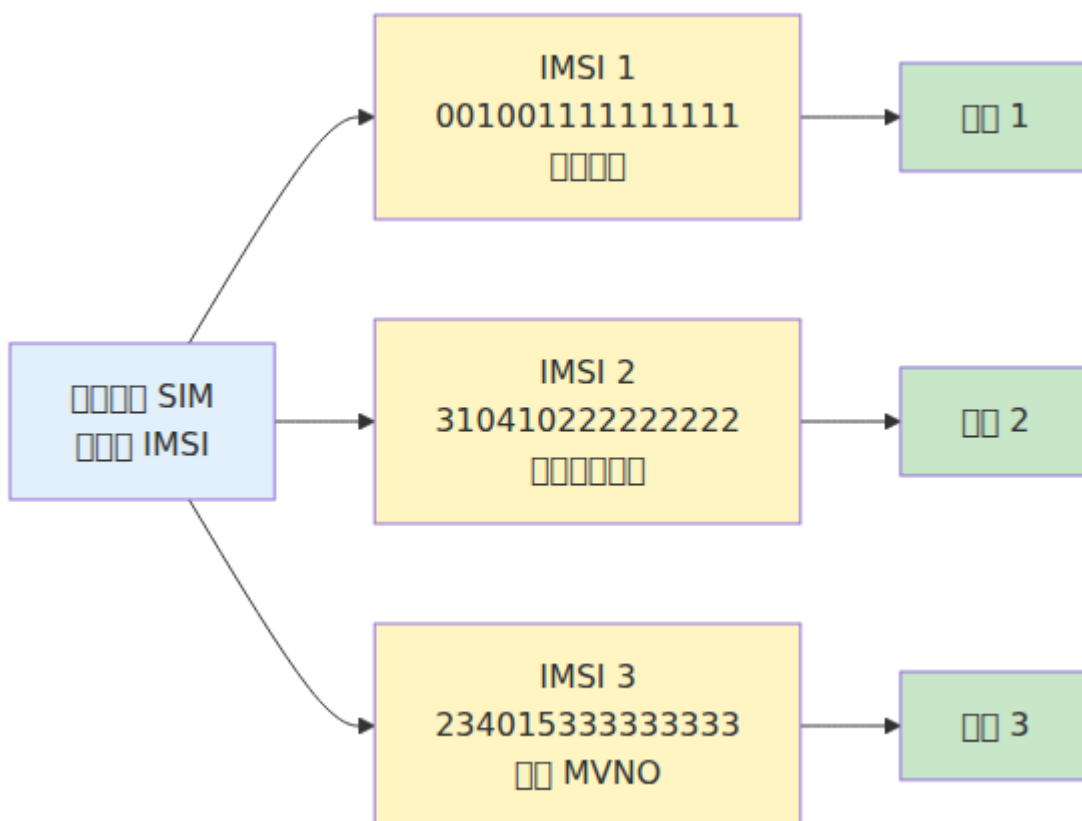
MSISDN

HSS MSISDN IMS MSISDN IMS

MSISDN

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

MSISDN

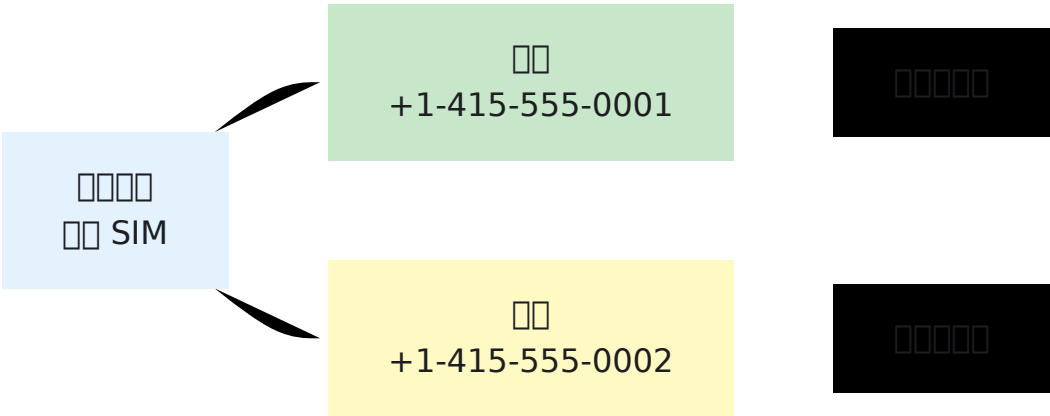


MSISDN MSISDN

□□□□

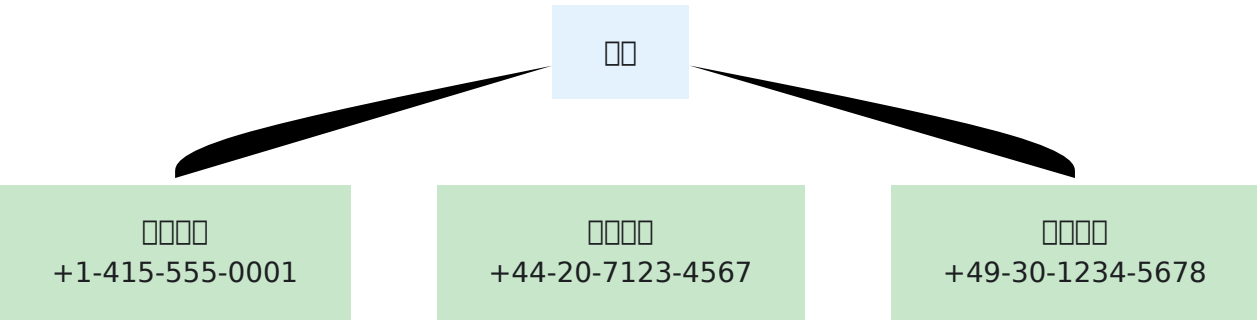
1. □□□□□□

□□□□□□□□□□□□□□□□□□



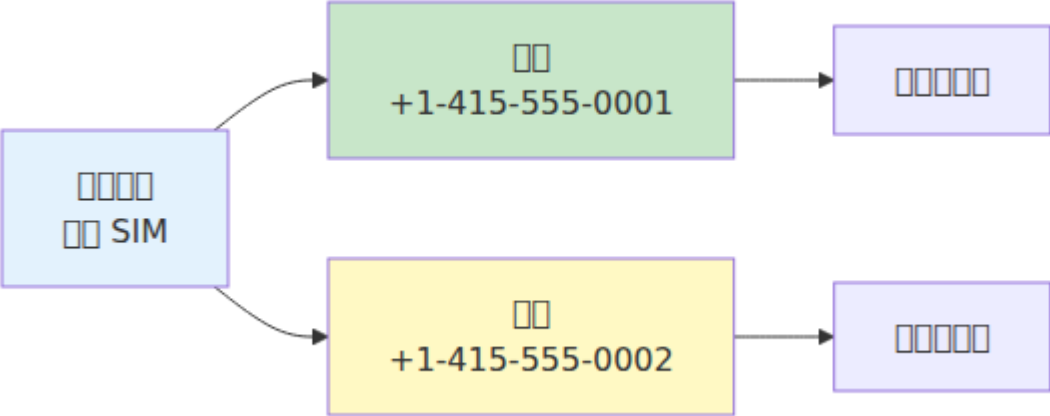
2. □□□□

□□□□□□□□□□□□□□□□



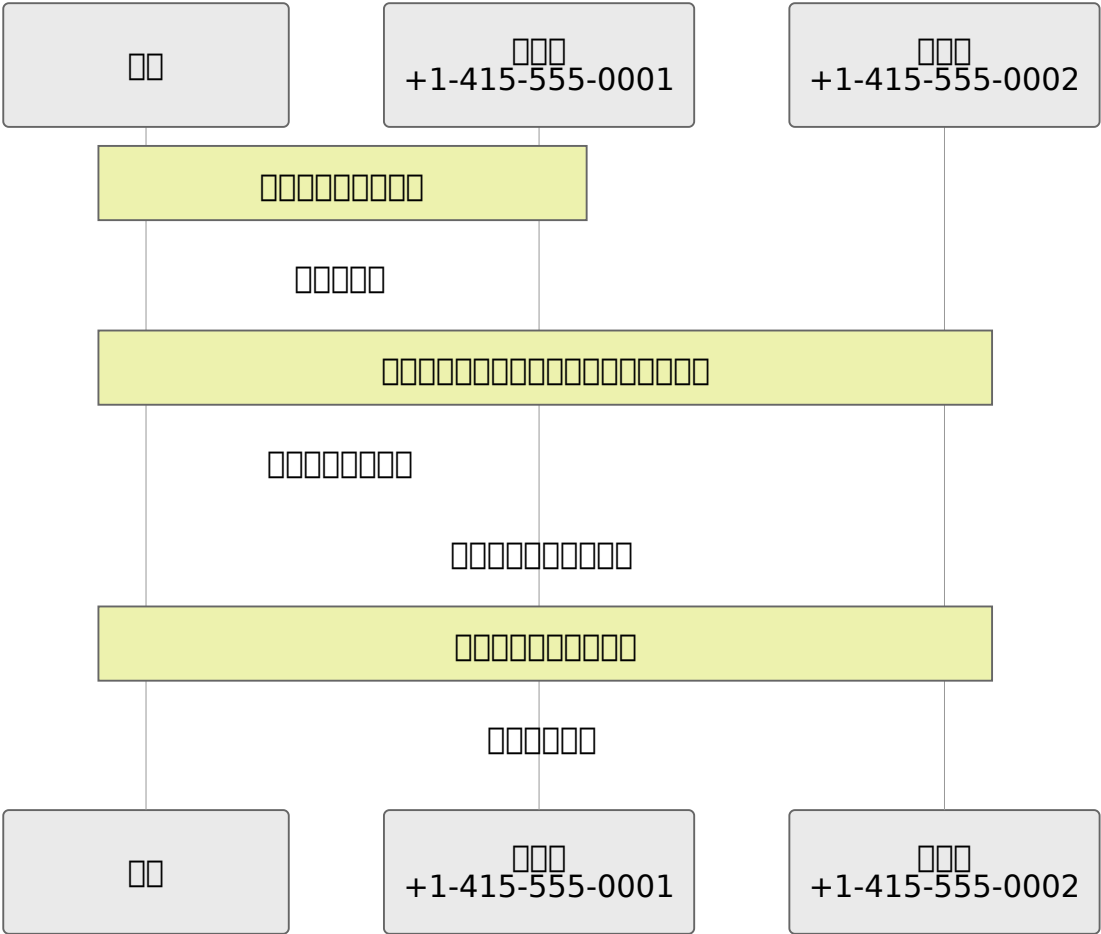
3. □□□□

□□□□□□□□□□□□□□



OmniHSS SIM/IMSI MSISDN

4.



MSISDN

MSISDN

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

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

MSISDN 管理

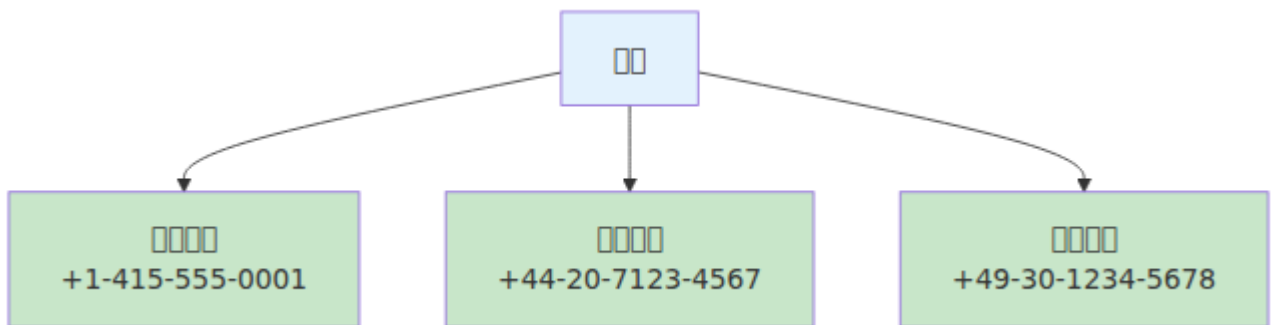
管理 MSISDN 的接口

接口列表

- 创建 MSISDN
- 更新 MSISDN
- 删除 MSISDN

接口参数说明

接口返回



接口说明

创建 MSISDN 接口

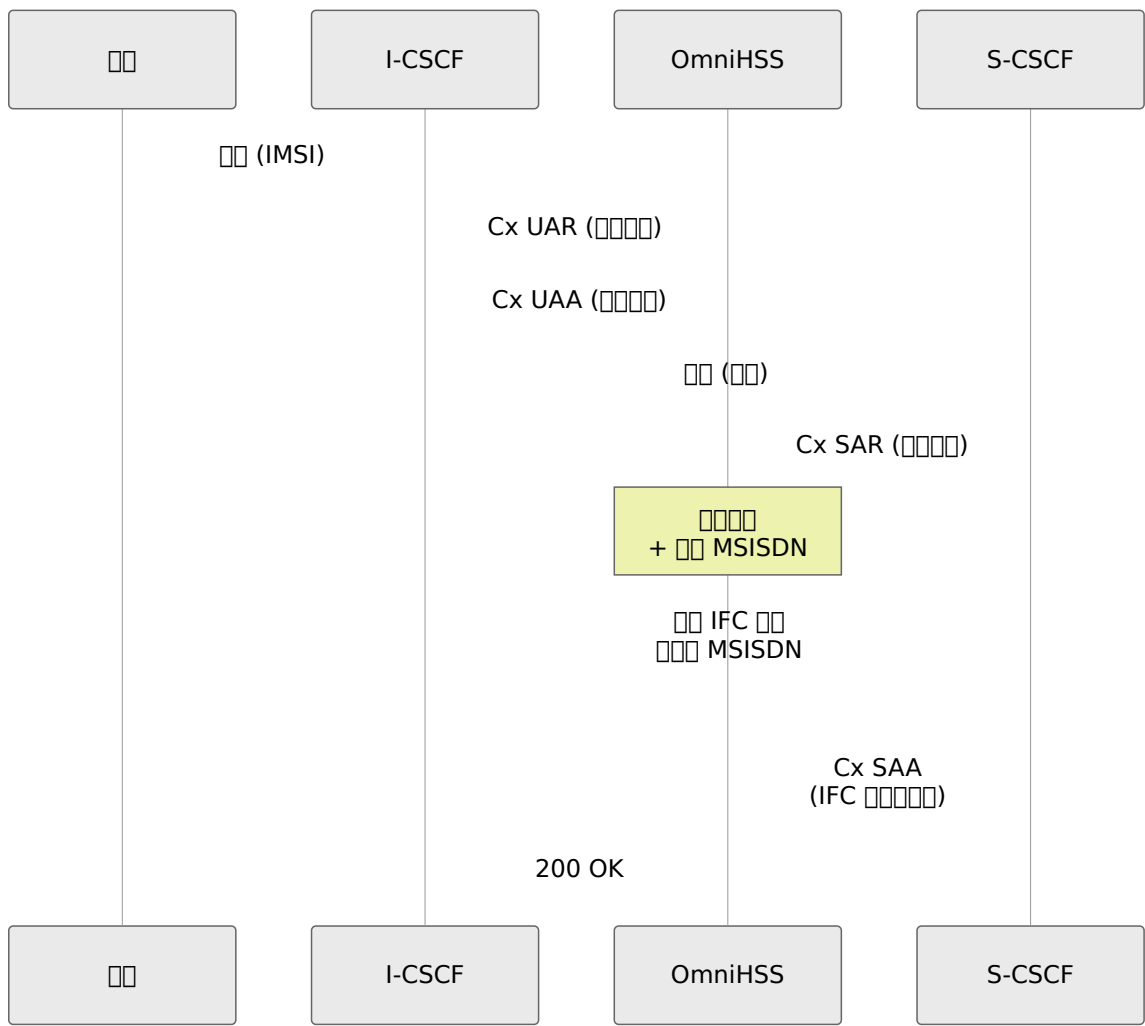
- 接口名称
- 接口地址
- 接口方法

□□□□□ 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 □□

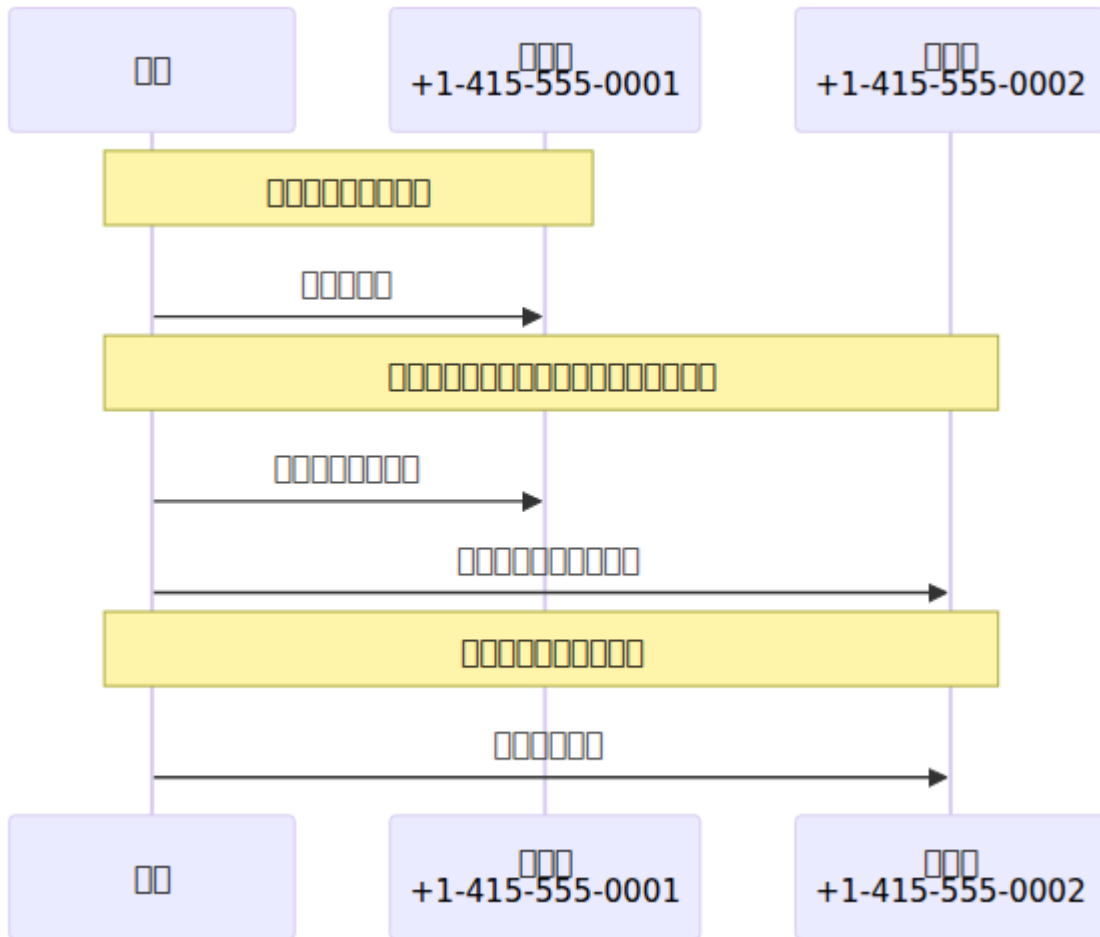
□□□□

□□ MSISDN □□□□□□□□ IMS □□□□□



□□□□□□

□□□□□□□□□□□□IMS □□□□□□□□□□ SIP URI□



SIP INVITE

SIP INVITE

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 IMSI 認證 MSISDN
- 認證

2. 認證 **IMS**

- 認證 `{{msisdns}}`
- 認證 XML

3. 認證 **HSS**

- 認證 IMS 認證 Cx SAR
- 認證 MSISDN 認證

4. 認證 **IMS**

- 認證
- 認證 S-CSCF 認證

認證 **MSISDN**

認證

- 認證
- 認證“認證”“認證”

認證

1. **MSISDN**

- 認證 MSISDN 認證
- 認證 認證

2. **MSISDN**

- MSISDN 查詢
- API 查詢 MSISDN

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

111

- 
- 

□ □ □ □ □ □ □

1. IMS 系统组成

- S-CSCF 注册
- SIP URI

2. IMS 環境

- 11 IFC 1111111111111111
- 1111111111111111

3.

```
# SIP 00000
sip:+14155551001@ims.example.com # 0000
sip:+14155551002@ims.example.com # 0000
```

MSISDN API

000

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

111

MSISDN

□□□□

□□□□

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

MSISDN □□

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

IMS □□

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

□□

□□□ MSISDN □□□□ MSISDN □□

□□□ 1 □ MSISDN

MSISDN

— 333 —

□□□□□



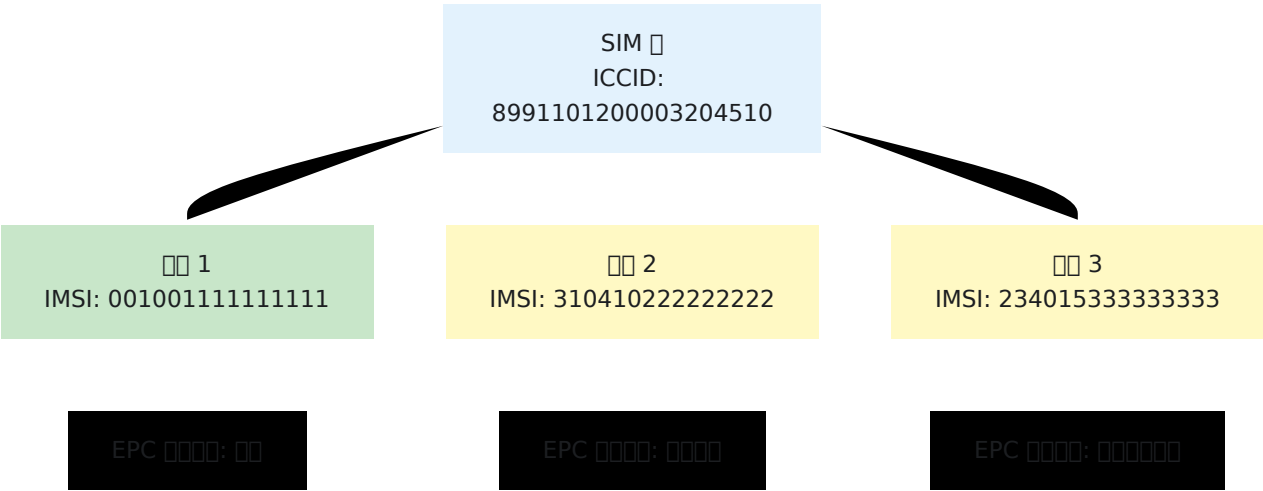
IMSI SIM

IMSI SIM IMSI IMSI
 IMSI

物联网 物联网卡 物联网 **IMSI** 物联网 SIM 物联网 IMSI 物联网 HSS 物联网

OmniHSS 物联网

物联网 OmniHSS 物联网 IMSI SIM 物联网 IMSI 物联网 物联网 物联网 **SIM** 物联网



物联网

1. 物联网

- 物联网 IMSI: 001-001物联网
- 物联网 IMSI: 310-410物联网
- 物联网 IMSI: 234-015物联网
- 物联网 IMSI

2. MVNO 物联网

- 物联网 IMSI: MVNO 物联网
- 物联网 IMSI: 物联网
- 物联网 MVNO 物联网

3. IoT/M2M 物联网

- IMSI 1: 物联网
- IMSI 2: 物联网

- IMSI 3: 00/000000
- 00000000

4. 000

- 000 IMSI 0000000000
- 0000000000
- 00000000000000

IMSI 00

0000

- 00 IMSI 000000 KiOPC 0000
- 00 IMSI 00000000
- 000000000000

00000000

- 000 EPC 00000000APN
- 000 IMS 0000000000
- 00 IMSI 00000000

000000

- 00 IMSI 000000 SIM000 sim_id
- 000000000000 ICCID
- 00 SIM 00000000

0000

- 000 SIM 00000000 IMSI
- 000000000000
- HSS 0000000000 IMSI



```
# 1.  SIM IMSI
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 1 
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.  1 IMSI
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
  }}'

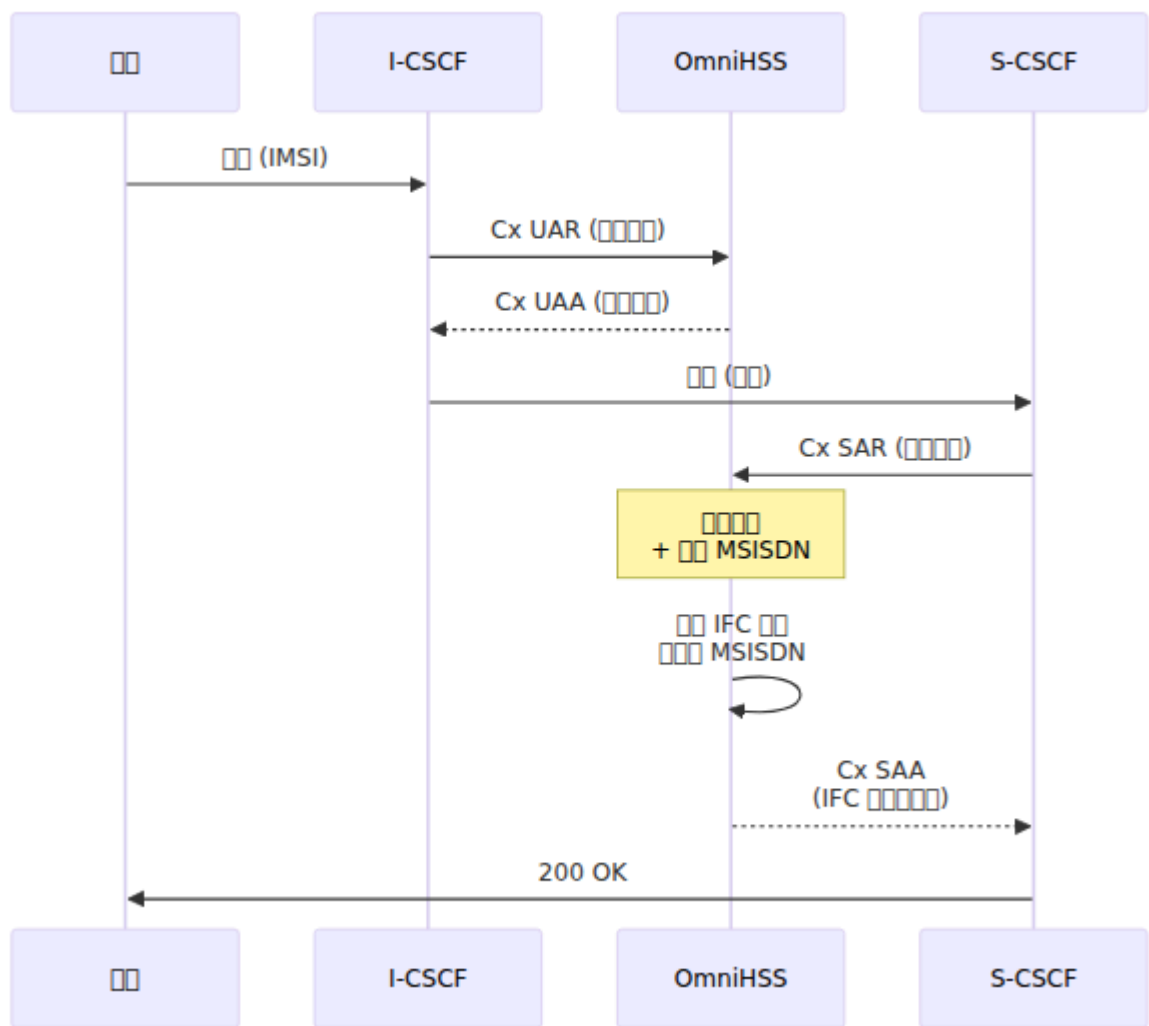
# 4.  IMSI 2 
KEYSET2=$(curl -k -X POST https://hss.example.com:8443/api/key_set \
  -d '{"key_set": {"ki": "111111111111111...", "opc": "2222222222..."}}' \
  | jq -r '.data.id')

# 5.  2 IMSI
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 IMSI ...
```

□□□□

□□ IMSI □□□□□□



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

IMSI □□□□□□□□

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

□□□□ **IMSI** □□

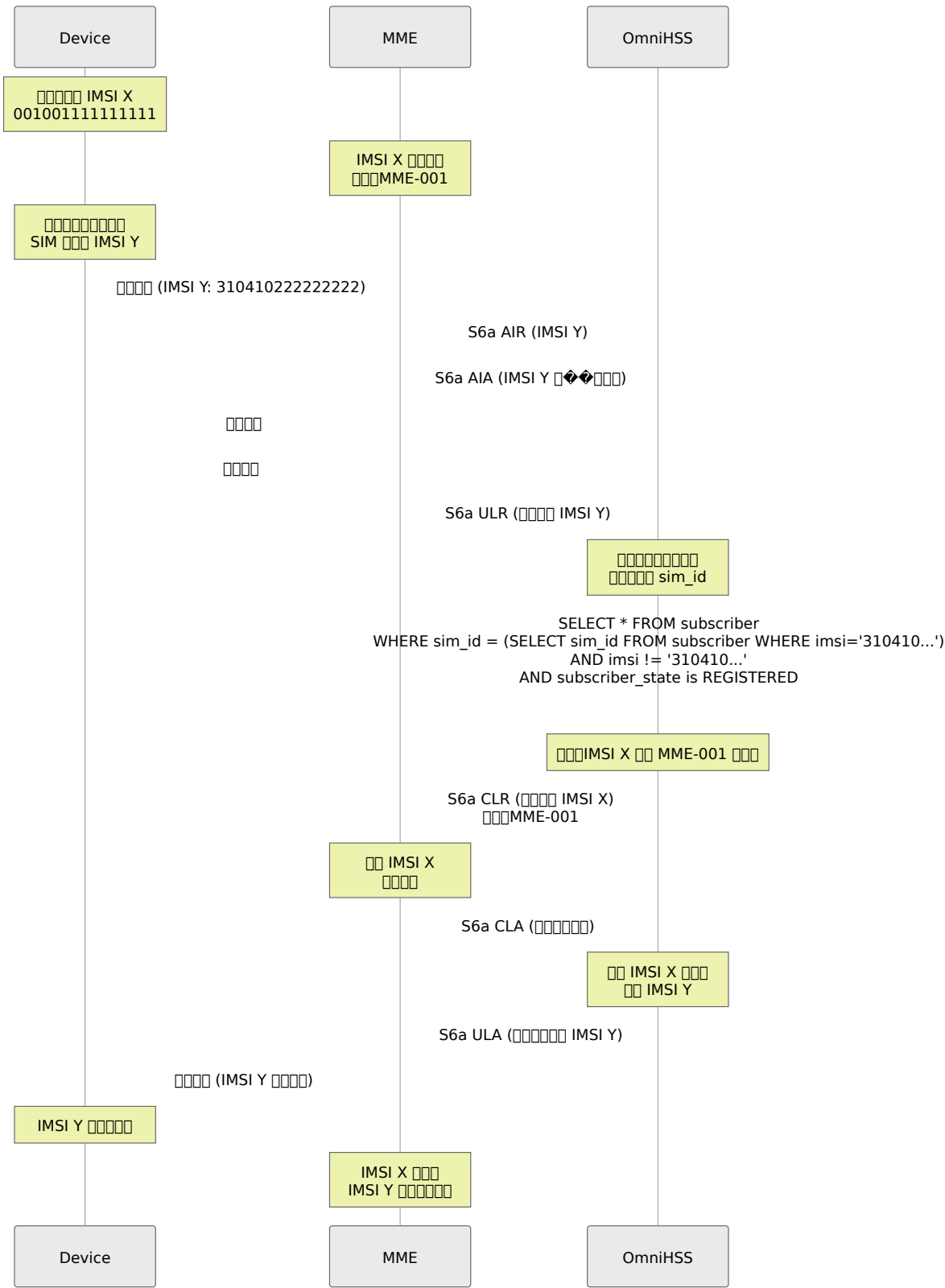
□□□□□ □□ SIM □□□□□□□□□□IMSI□□□□□□□□□□□□□□□□

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

- HSS 数据库 数据库 数据库 IMSI X

数据库 IMSI 数据库数据库数据库数据库数据库数据库

IMSI 数据库



□□□□□□

□□□□□□

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

□□□□□□

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

□□□□□□

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

IMSI □□□□□

□□/SIM □□□□□□ IMSI□□□□□

1. □□□□□

- □□ IMSI □□□□□
- □□□□□□□□□ IMSI

2. □□□□□

- □□□□□□□□□
- SIM □□□□□□ IMSI

3. □□□□□

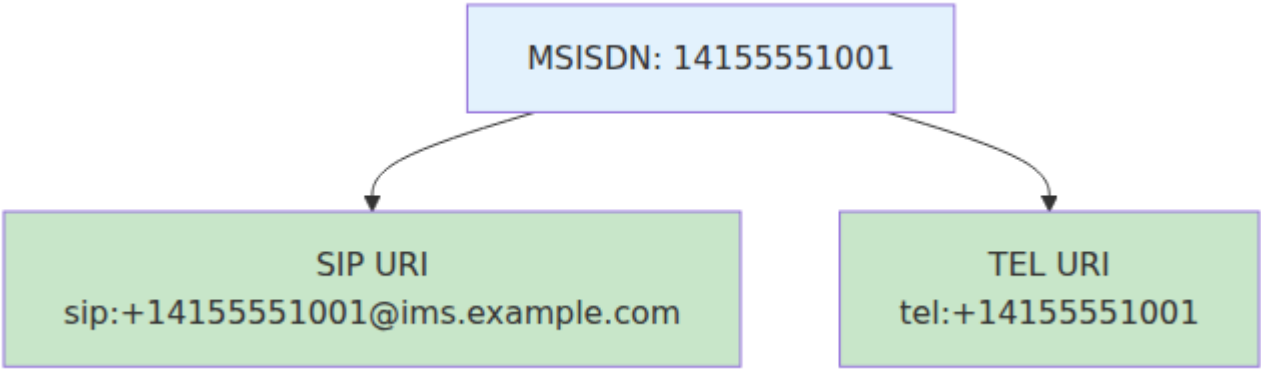
- SIM □□□□□□□□□□□□□□□□ IMSI□
- □□ MCC/MNC □□□□□

4. □□□□□

- 國際移動網際網路 IMSI
- 國際 IMSI 號碼

IMS 網

國際移動網際網路 IMS 網



◆◆◆

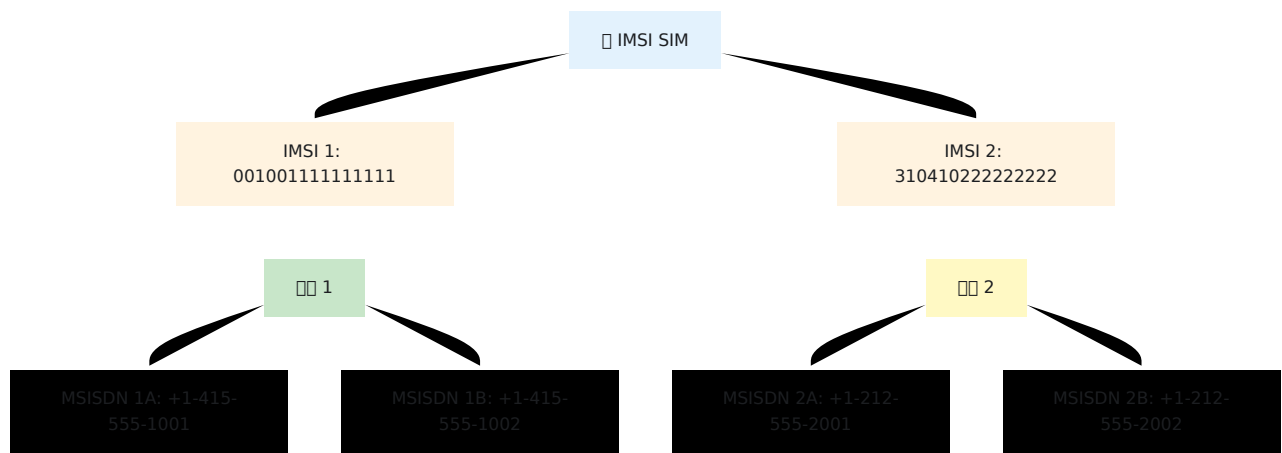
國際網

1. 國際網 IMSI 國際網 IMSI 網 HSS 網“網”網網網網
2. 網 **SIM** 網網網網網 IMSI SIM 網網網網網網網網網 `sim_id` 網網網網網網網網網“網”網網
3. 網網網 `subscriber_state` 網網網 IMSI 網網 MME/SGSN 網 IMSI 網網網網網網網網網
4. 網網網 網網網網網網網
 - 國際網網網 IMSI
 - 國際網網網網網網網 IMSI
 - 網網 SIM 網網 IMSI 網網網網

網網網

網 IMSI + 網 MSISDN

國際網網網網網網網 SIM 網網 IMSI網網 IMSI 網網 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 注册流程

1. 终端 IMS 注册
2. 终端 IMS 向 key_set 注册
3. 终端 EPC 注册
4. 终端注册成功

IMS 鉴权

- 终端/SIM 向 HSS 注册
- HSS 向终端 IMS 注册
- 终端 IMS 注册

MSISDN

MSISDN 注册

1. 终端 MSISDN 注册
2. 终端 IMS 向 `{{msisdns}}` 注册
3. 终端 IMS 注册
4. 终端 S-CSCF 注册

MSISDN 鉴权

- 终端向 HSS 注册
 - 终端向 HSS 注册
 - HSS 向终端 IMS 注册
-

□□□□

□ MSISDN □□

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

□ IMSI SIM □□

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

□□□□

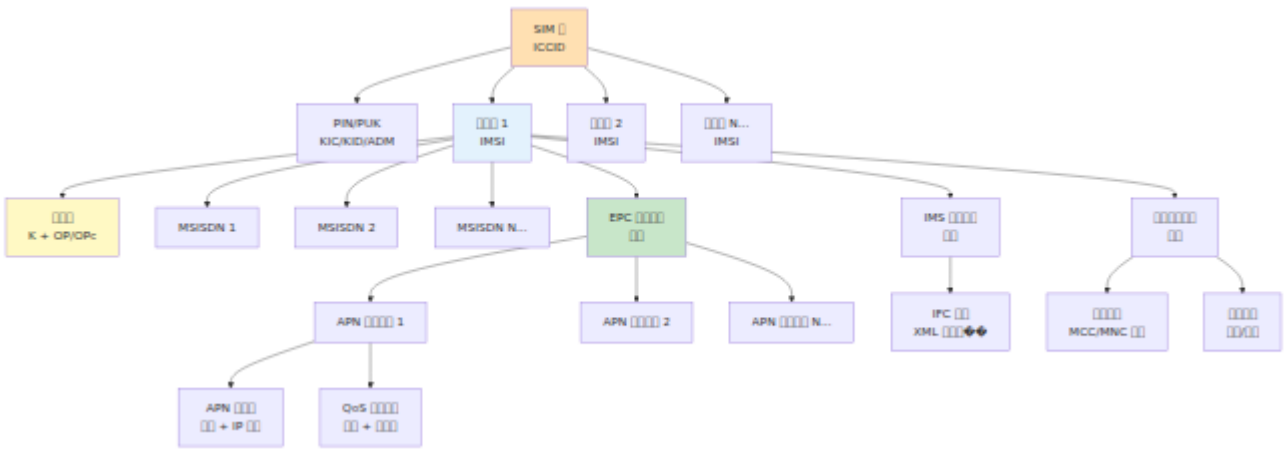
- ✓ □□□□□
- ✓ □□□□□□□□□□
- ✓ □□□□□□□□
- ✓ □□□□□□□
- ✓ □□□□□□□

OmniHSS 数据库结构

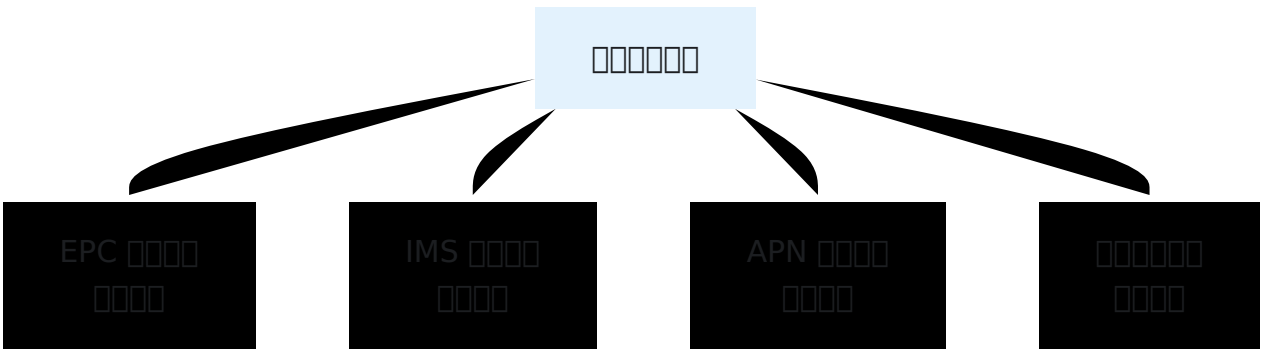
← 数据库

数据库

OmniHSS 数据库结构图如下所示，展示了数据库的层级关系和主要数据表。



数据库



EPC 数据库

EPC 数据库结构图如下所示，展示了数据库的层级关系和主要数据表。

項目

項目	単位	範囲
<code>ue_ambr_dl_kbps</code>	kbps kbps	10,000 - 1,000,000 Kbps
<code>ue_ambr_ul_kbps</code>	kbps kbps	5,000 - 500,000 Kbps
<code>network_access_mode</code>	enum enum	"packet_only" 又は "packet_and_circuit"
<code>tracking_area_update_interval_seconds</code>	TAU seconds	54 秒

API EPC 設定

```
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
}'
```

API EPC 設定

項目:

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

項目:

- 100: 50 Mbps (50,000 Kbps)
- 100: 25 Mbps (25,000 Kbps)

11

- 100: 100 Mbps (100,000 Kbps)
- 50: 50 Mbps (50,000 Kbps)

111

- 100: 1 Gbps (1,000,000 Kbps)
- 50: 500 Mbps (500,000 Kbps)

IMS ☐ ☐ ☐ ☐

IMS IFC

IFC

IFC 00000 S-CSCF 00000000 XML 000

0000

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

IMS 概要

```
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>"
  }
}'
```

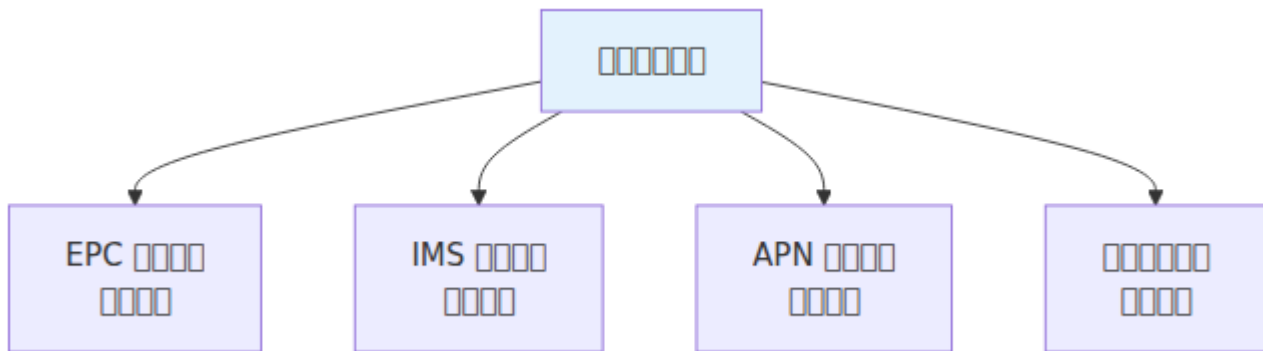
IFC 概要

```
<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 設定

APNは、モバイルネットワークを通じてインターネットに接続するための設定です。

APN 設定



APN 設定

APN 設定は、IP 設定と関連しています。

APN:

- internet - インターネット
- ims - IMS/VoLTE
- mms - MMS
- vzwadmin - VZW Admin

IP 設定:

- ipv4: IPv4
- ipv6: IPv6
- ipv4v6: IPv4v6
- ipv4_or_ipv6: IPv4 または IPv6

APN QoS 設定

QoS 設定は、ネットワークの品質を確保するための設定です。

QCIQoS :

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

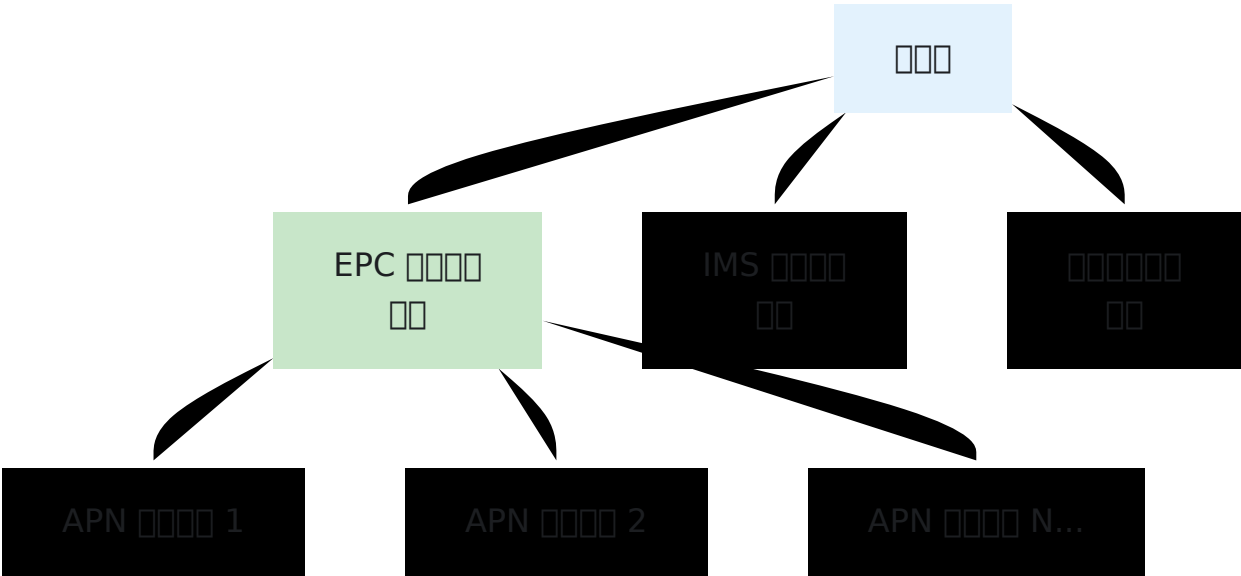
📡 APN 📡 ID 📡 EPC 📡 ID 📡 APN 📡 EPC 📡

□□□□□□

□□□ □□□□□ □□□□□□

□□□□□□

□□□□□□□□



API 호출 예제

```
# IMS 프로파일 EPC IMS 프로파일
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
  }
}'

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

API 호출 결과

결과

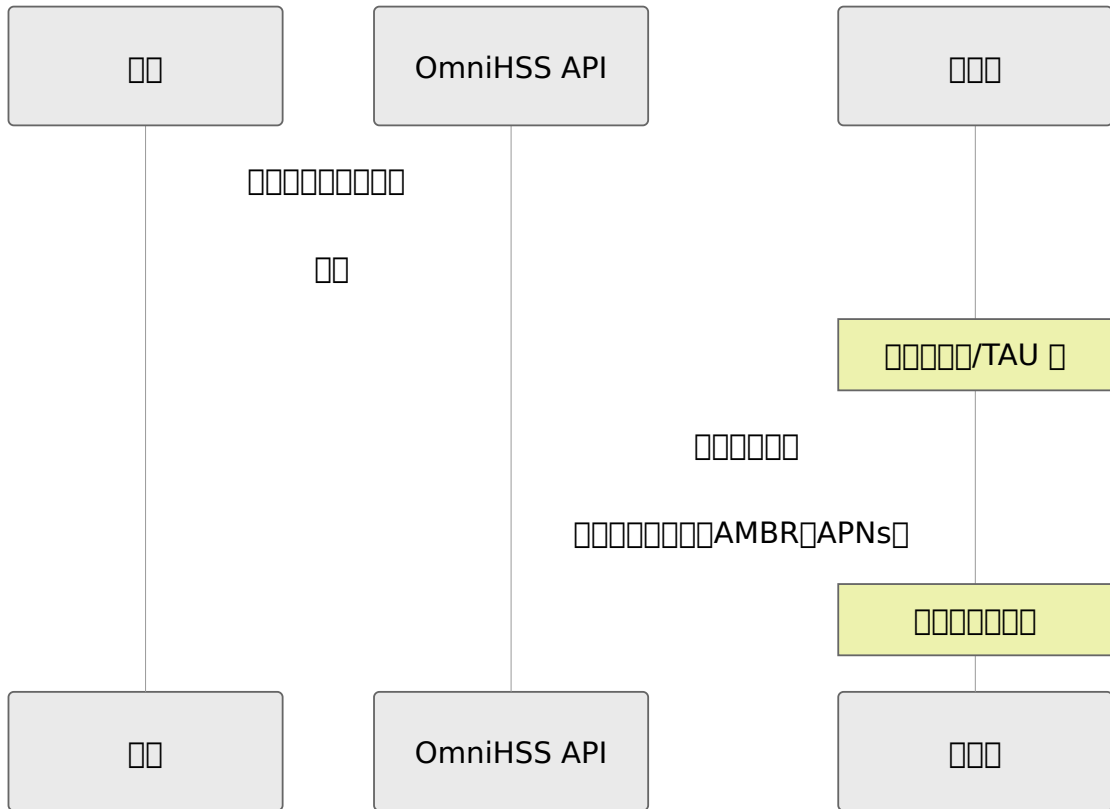
1. IMS 프로파일 - IMS 프로파일 ID와 IMS 프로파일 ID를 반환
2. IMS 프로파일 - IMS 프로파일 ID를 반환
3. IMS 프로파일 - IMS 프로파일 ID를 반환
4. IMS 프로파일 - IMS 프로파일 ID를 반환

[]-[]-[]

□□: □□

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

□ □ □ □ □ □ □ □ □ □ □ □



□□: □□□□□□□□□□□□

- 〇〇〇〇〇〇〇TAU〇
- 〇〇
- IMS 〇〇〇〇〇 IMS 〇〇〇〇〇〇〇〇

配置参数

配置参数如下：

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

IMS 配置：

1. 配置 IMS 参数
2. 配置 IFC 参数 XML 参数
3. 配置 S-CSCF 参数 IFC 参数
4. 配置 S-CSCF 参数

APN 配置：

1. 配置 APN 参数 EPC 参数
2. 配置 APN 参数
3. 配置 UE 参数 PDN 参数

OmniHSS 配置

← 設定

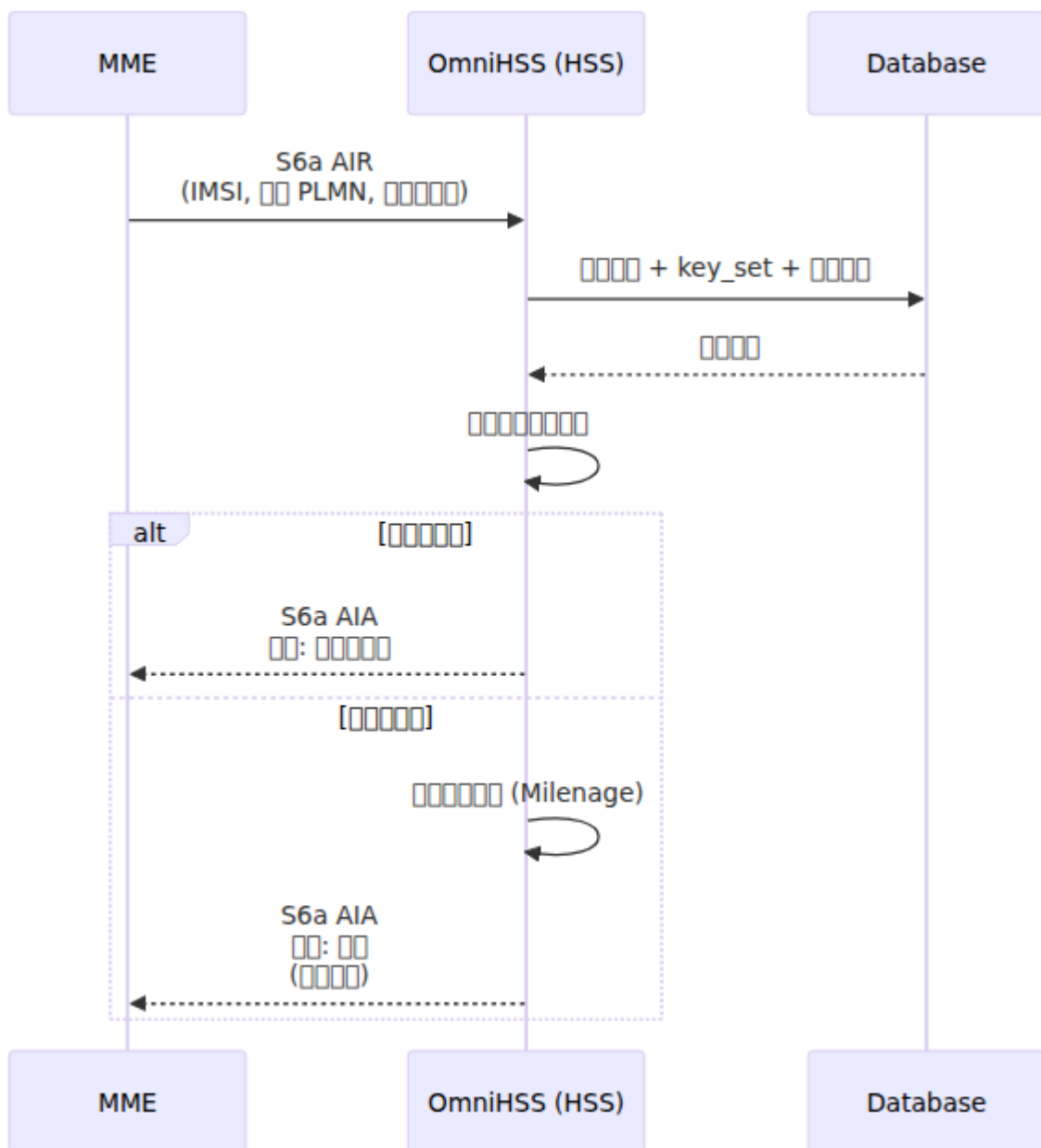
設定

OmniHSS 設定 Diameter 設定

S6a 設定 (LTE/EPC)

設定 (AIR/AIA)

MME 設定

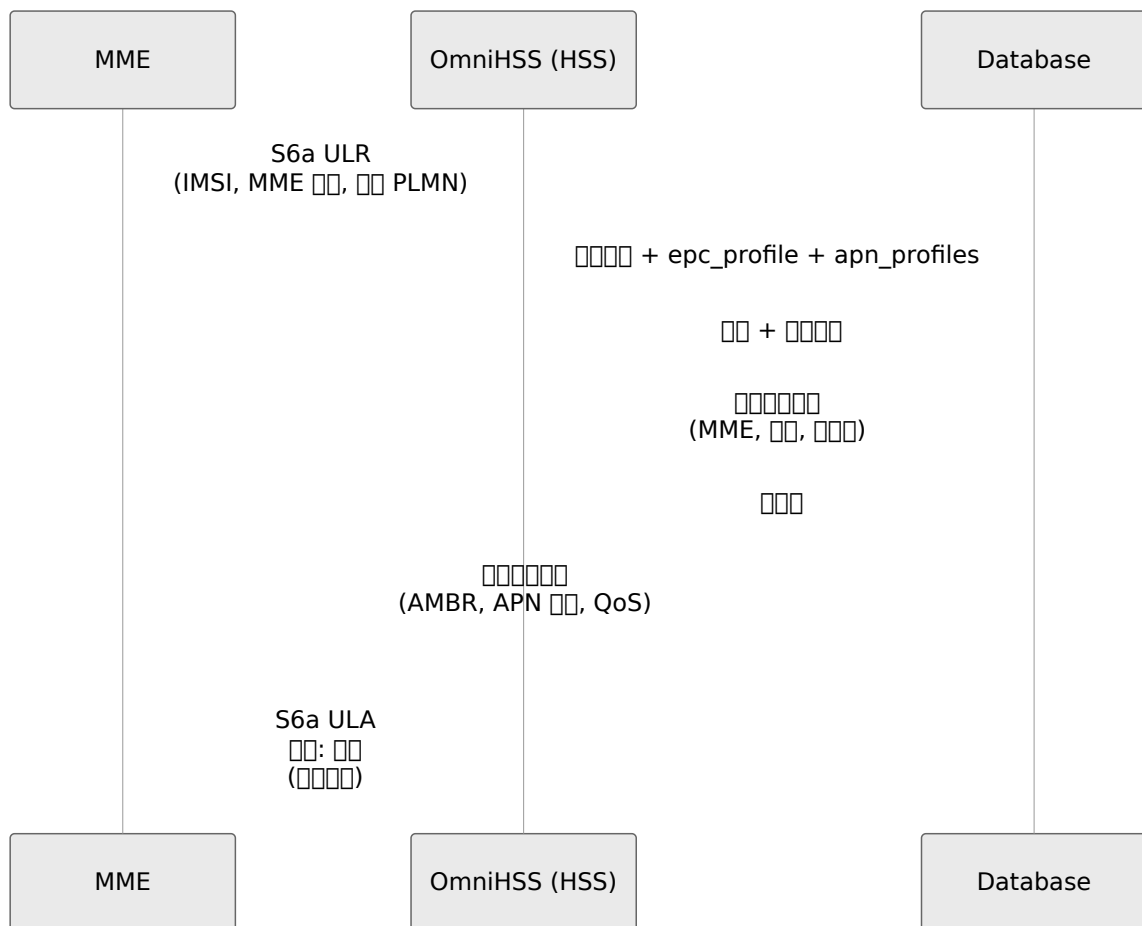


AVP:

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

ULR/ULA

MME HSS

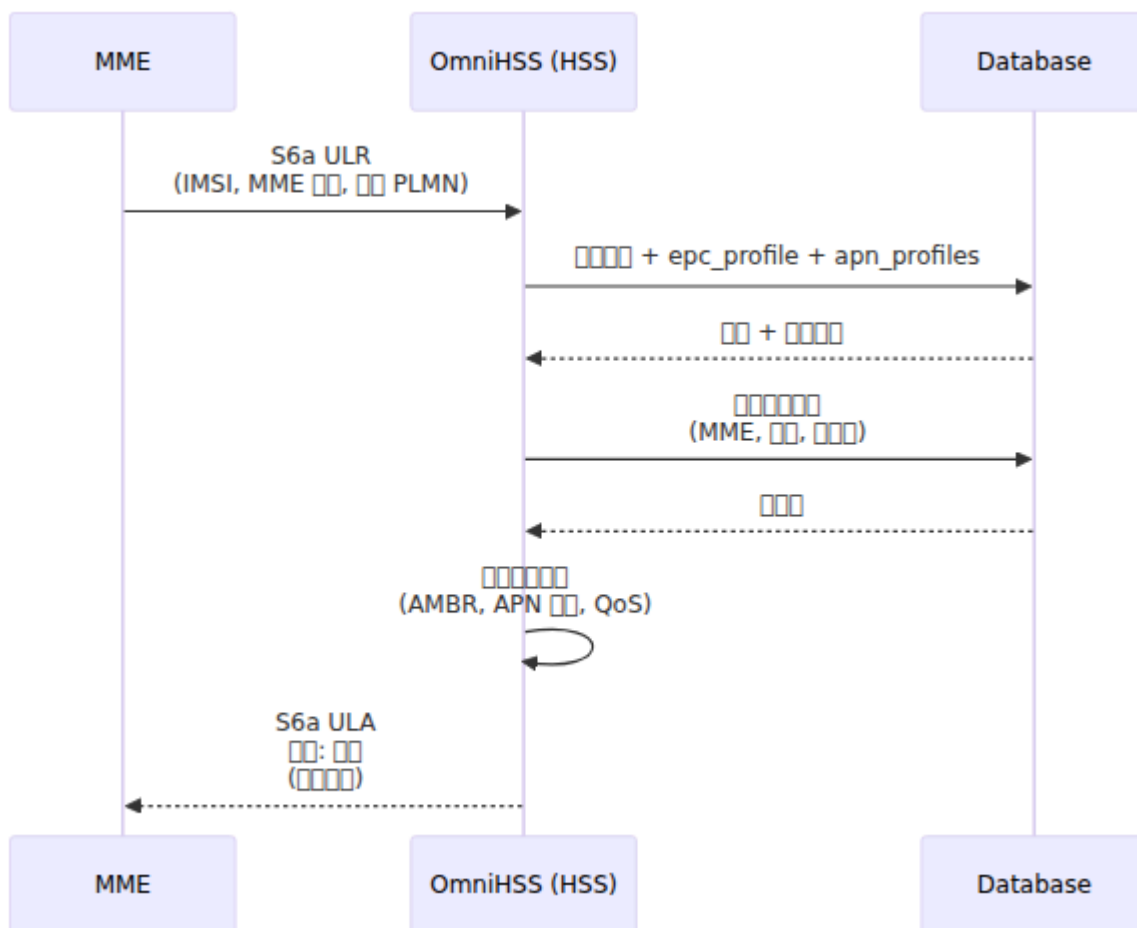


AVP:

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

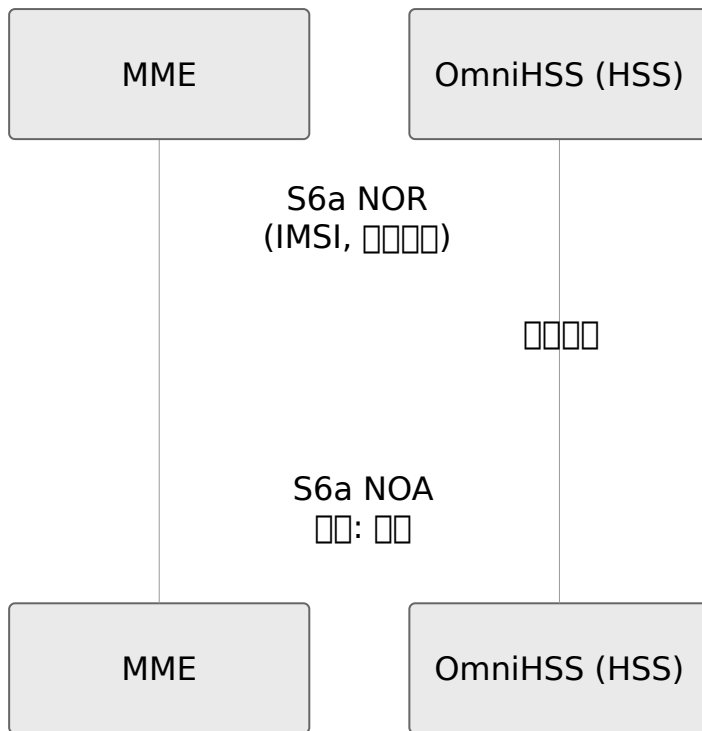
UE ID (PUR/PUA)

MME ID HSS ID



Database (NOR/NOA)

MME ID HSS ID

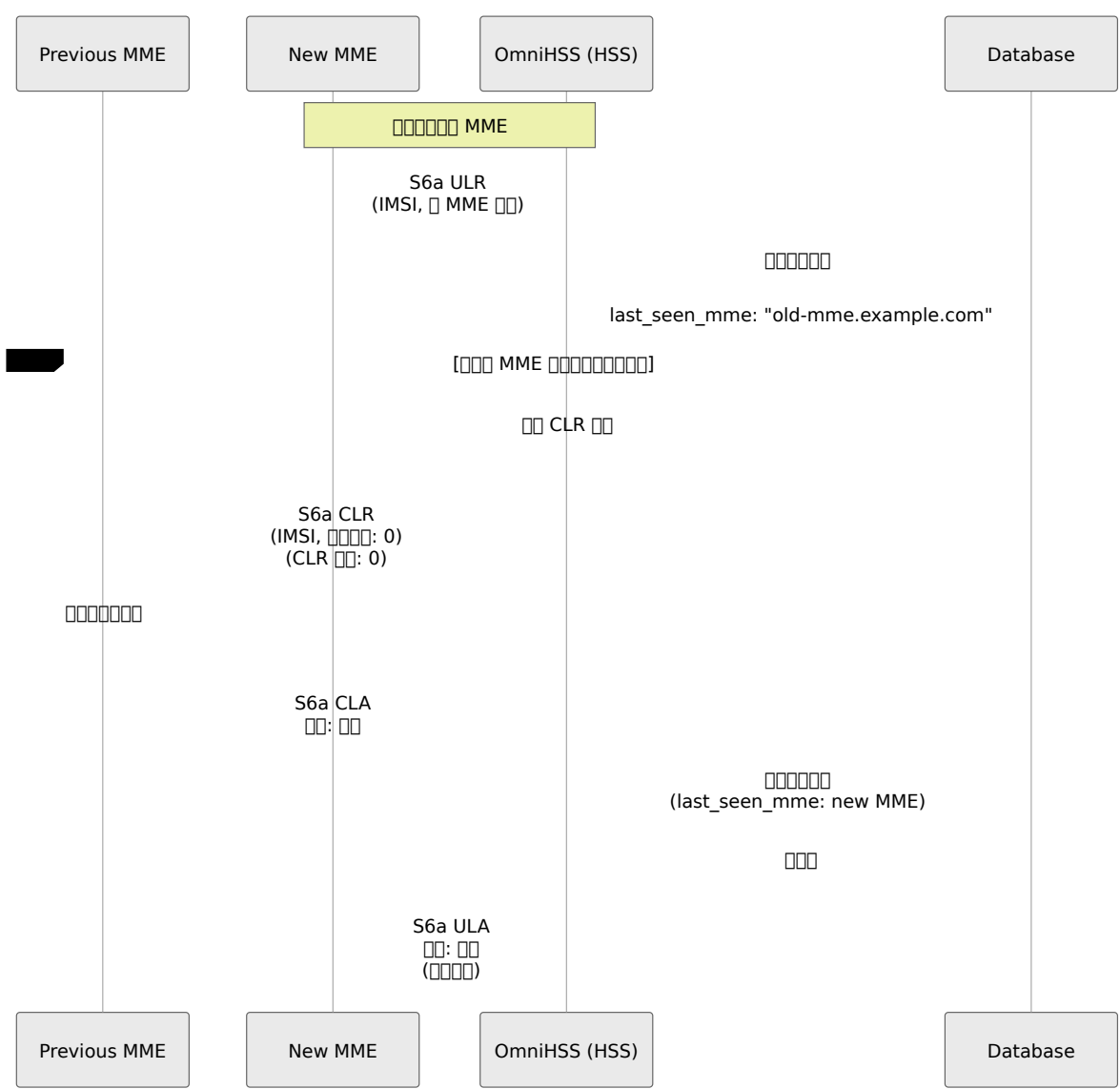


□□□□□□ (CLR/CLA)

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

□□ CLR (MME □□)

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

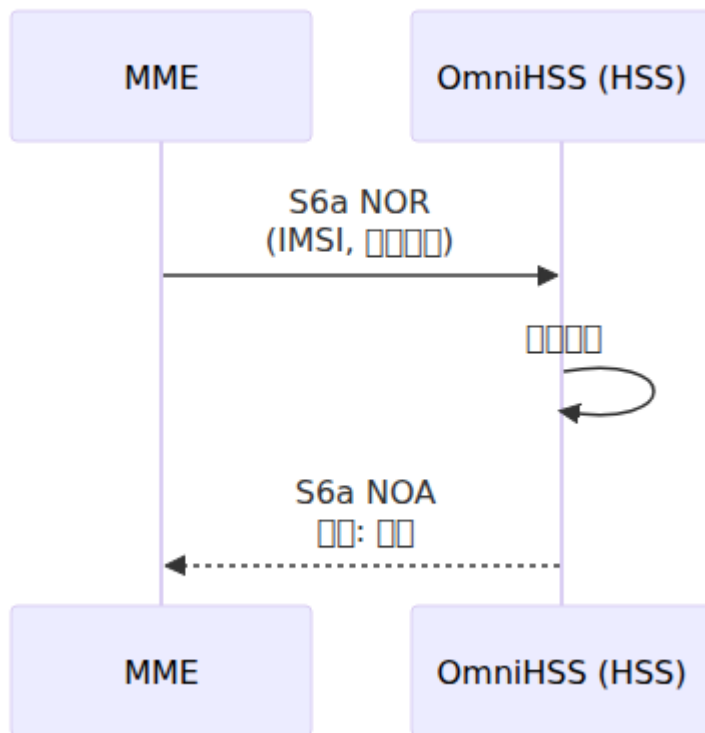


AVP (CLR):

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

CLR (API)

API CLR



AVP (CLR):

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

OmniHSS 3GPP TS 29.272

項目	値	項目	項目
MME 番号	0	MM MME 番号	MM MME 番号 ULR
SGSN 番号	1	SGSN 番号	3G/2G 番号
番号	2	番号	API 番号
番号 IWF	3	番号	番号
番号	4	番号	番号

CLR 番号

CLR 番号 AVP 番号番号番号番号番号

項目	値	項目
S6a/S6d 番号	0	1 = 番号 S6a 番号
番号	1	1 = UE 番号

番号 CLR 番号:

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

IMSI 番号

OmniHSS 番号番号 (IMSI) 番号 MME 番号番号 MSISDN番号番号 IMSI 番号 CLR 番号

番号 1: 番号 MSISDN番号 IMSI

□□ A:

- IMSI: 999000123456789
- MSISDNs: ["+1234567890", "+9876543210"]
- last_seen_mme: "mme01.operator.com"

□□□□□□□□ MME □□

- □□□□ **CLR** □ "mme01.operator.com"□IMSI □ 999000123456789
- □□ MSISDN □□□□□□□□□□□□ SIM□
- □□□ AVP □□ IMSI□□□□ MSISDNs

□□ 2: □□□□□□ **IMSI**□□□□ **MSISDN**

OmniHSS □□ □□ **MSISDN** □□□□□ MSISDN □□□□□□□□□□□□□□□□/□□□□□

□□ A:

- IMSI: 9990001111111111
- MSISDN: "+1234567890"
- last_seen_mme: "mme01.operator.com"

□□ B□□□□□:

- IMSI: 9990002222222222
- MSISDN: "+1234567890" # □□ MSISDN□□□ SIM/IMSI
- last_seen_mme: "mme02.operator.com"

□□□ B □□□□

- □□□ **CLR**□□□ IMSI = □□□□□
- □□ A □□□ mme01 □□
- □□ B □ mme02 □□
- □□□□□□□□□□□□□□□□□□□□

□□ 3: □□□ **CLR** □□□ **MSISDN** □□

□□□

- □□□□ **CLR** □□□□ last_seen_mme
- □□ **MSISDN** □□ IMSI □□□□□□□□□□

- IMSI 000 MME 00000

000000

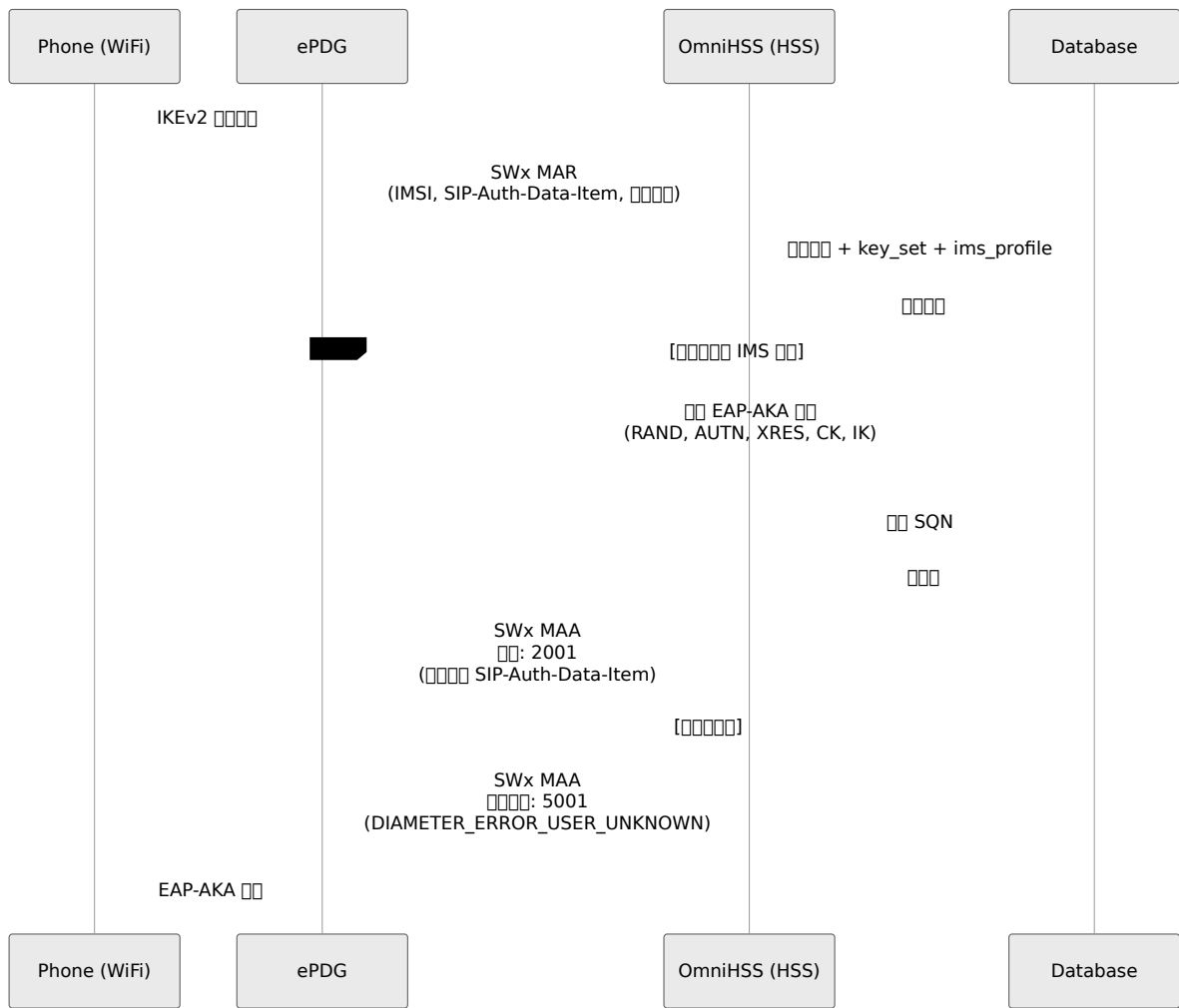
1. **IMSI** 000: CLR 00000 0 **IMSI**00000 MSISDN0 subscriber_state 0000 (IMSI)
00 last_seen_mme0
2. 0000: 000000000000 MME 00000 CLR 0000000000000000
3. 00000000 **MME** 0000 **CLR:** 00 last_seen_mme 0 nil000000000000 ULR 00000
CLR0
4. 000000: 00 CLR00 ULR 000000000 Subscription-Data AVP00000 MME 00000
000
5. 00: CLR 0000000000000000 MME 0 ULA 000000000000 MME 0 CLA0
6. **CLA** 00: OmniHSS 00 CLA 0000000000000000 398 00 :discard000000000000
0000 HSS 000

SWx 00 (0 3GPP 00 / WiFi 00)

SWx 00000 ePDG00000000000000 HSS0000 WiFi 000000000000

00000000 (MAR/MAA)

ePDG 00 EAP-AKA 00000000 WiFi 00000



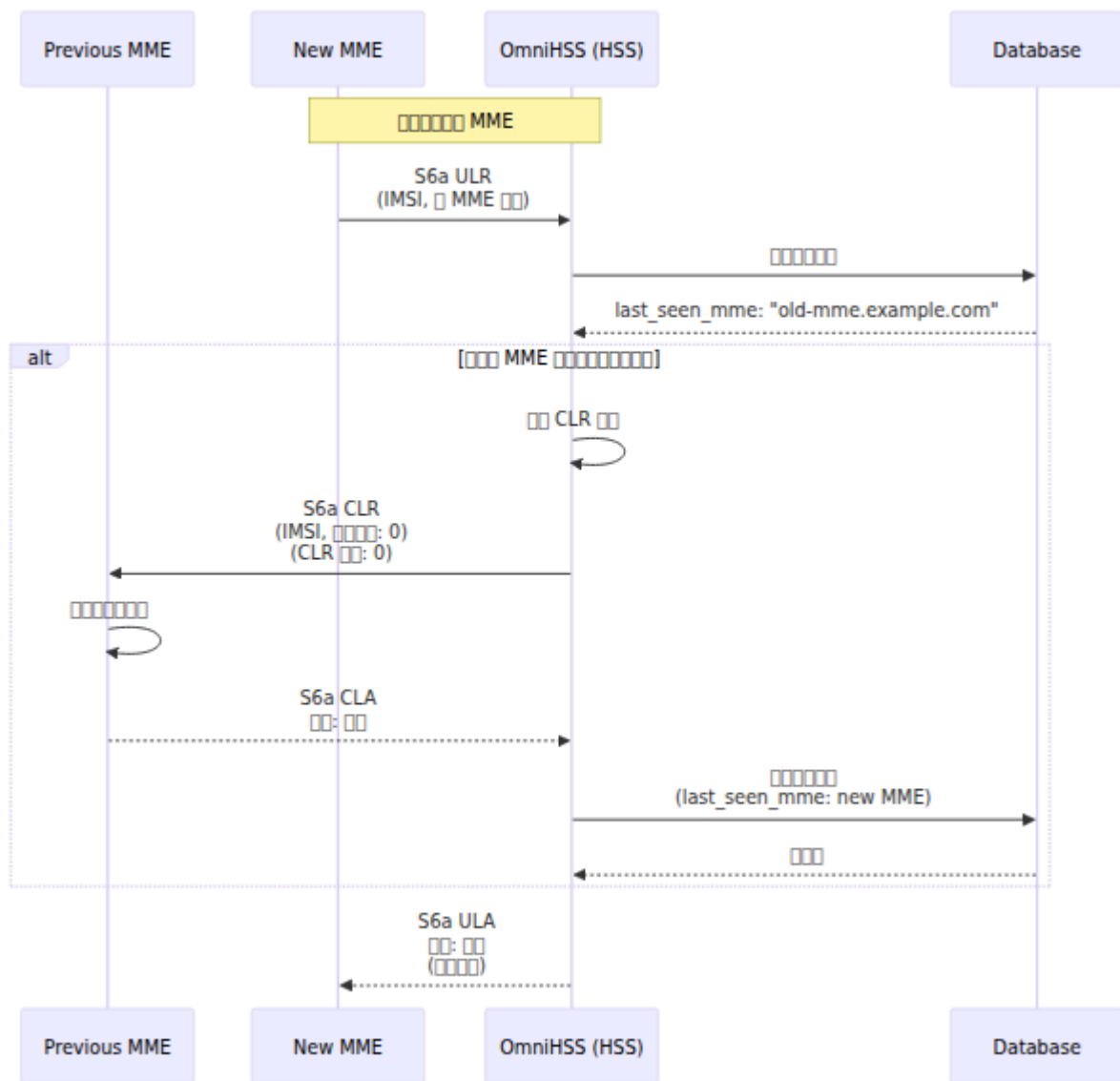
AVP:

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

SQN : UE SQN ePDG SIP-Authorization AVP OmniHSS AUTS 3GPP TS 33.102 IND SQN

(SAR/SAA)

ePDG WiFi



[MME]:

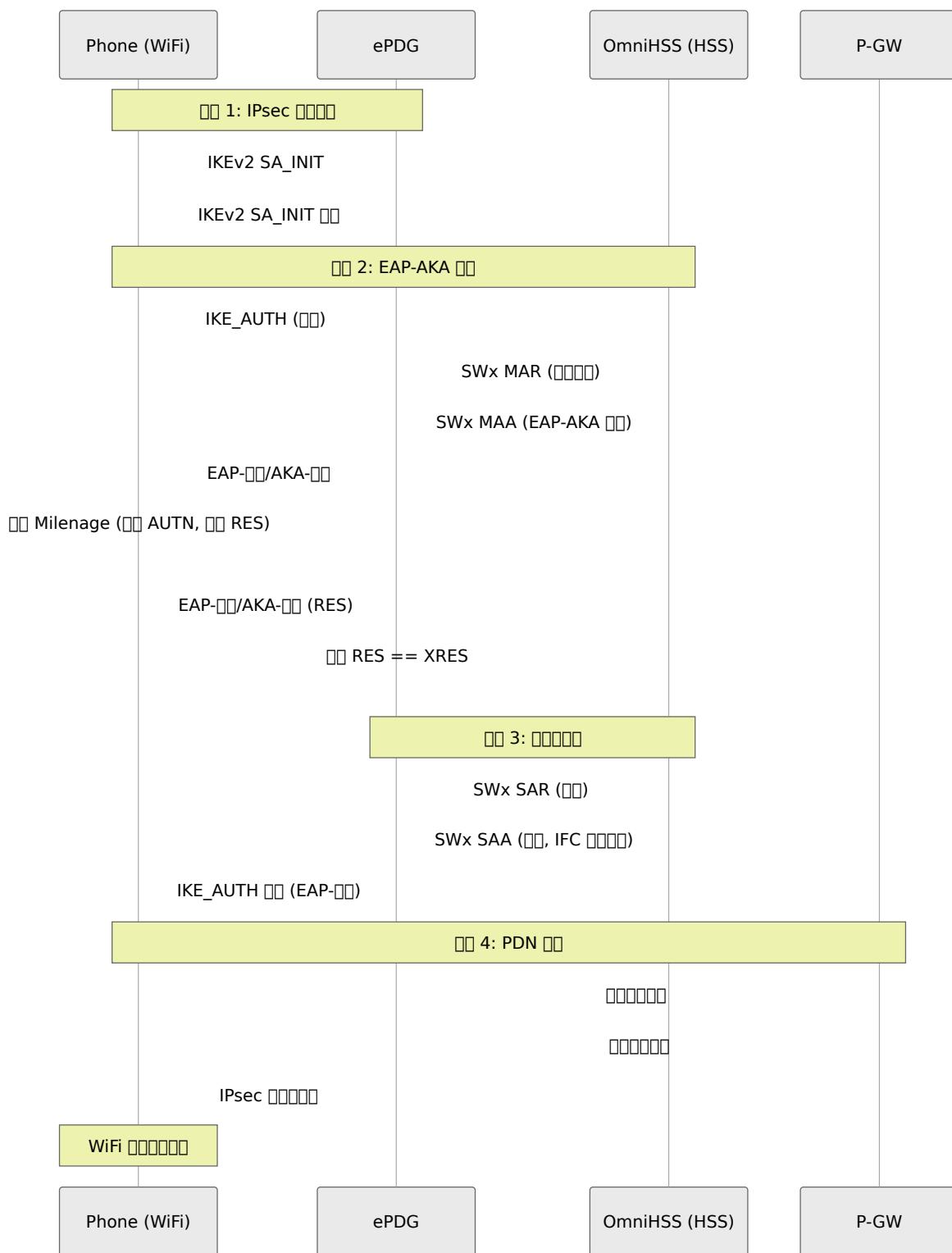
코드	번호	설명
NO_ASSIGNMENT	0	할당되지 않음
REGISTRATION	1	WiFi 등록
RE_REGISTRATION	2	재등록
UNREGISTERED_USER	3	등록되지 않은 사용자
TIMEOUT_DEREGISTRATION	4	등록 시간 초과
USER_DEREGISTRATION	5	사용자 등록 해제
AUTHENTICATION_FAILURE	7	인증 실패
ADMINISTRATIVE_DEREGISTRATION	8	관리자 등록 해제

AVP:

- 코드: IMSI, 전화번호, ePDG 주소, APN
- 코드: IFC XML, 3GPP 네트워크

WiFi

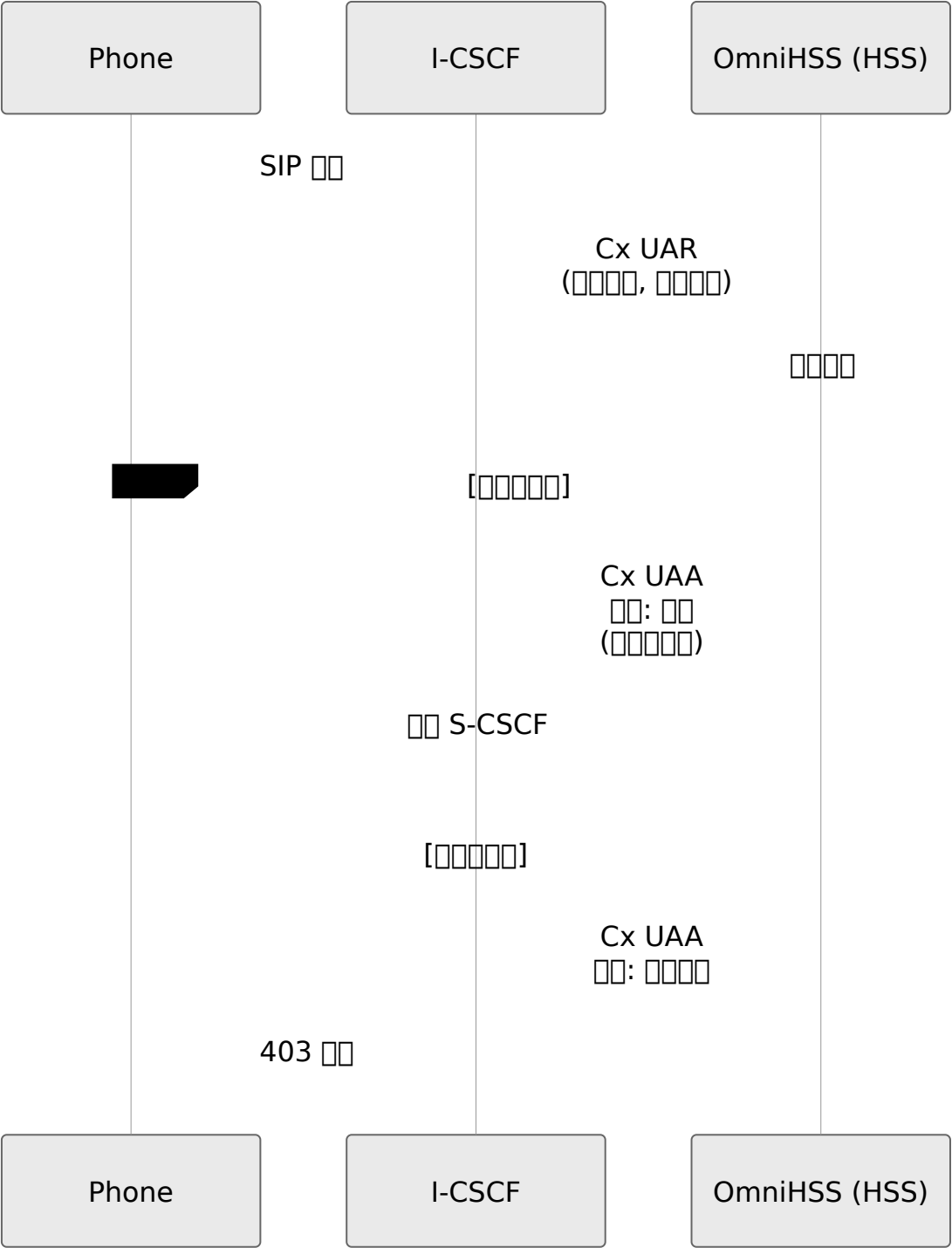
WiFi



Cx 消息 (IMS)

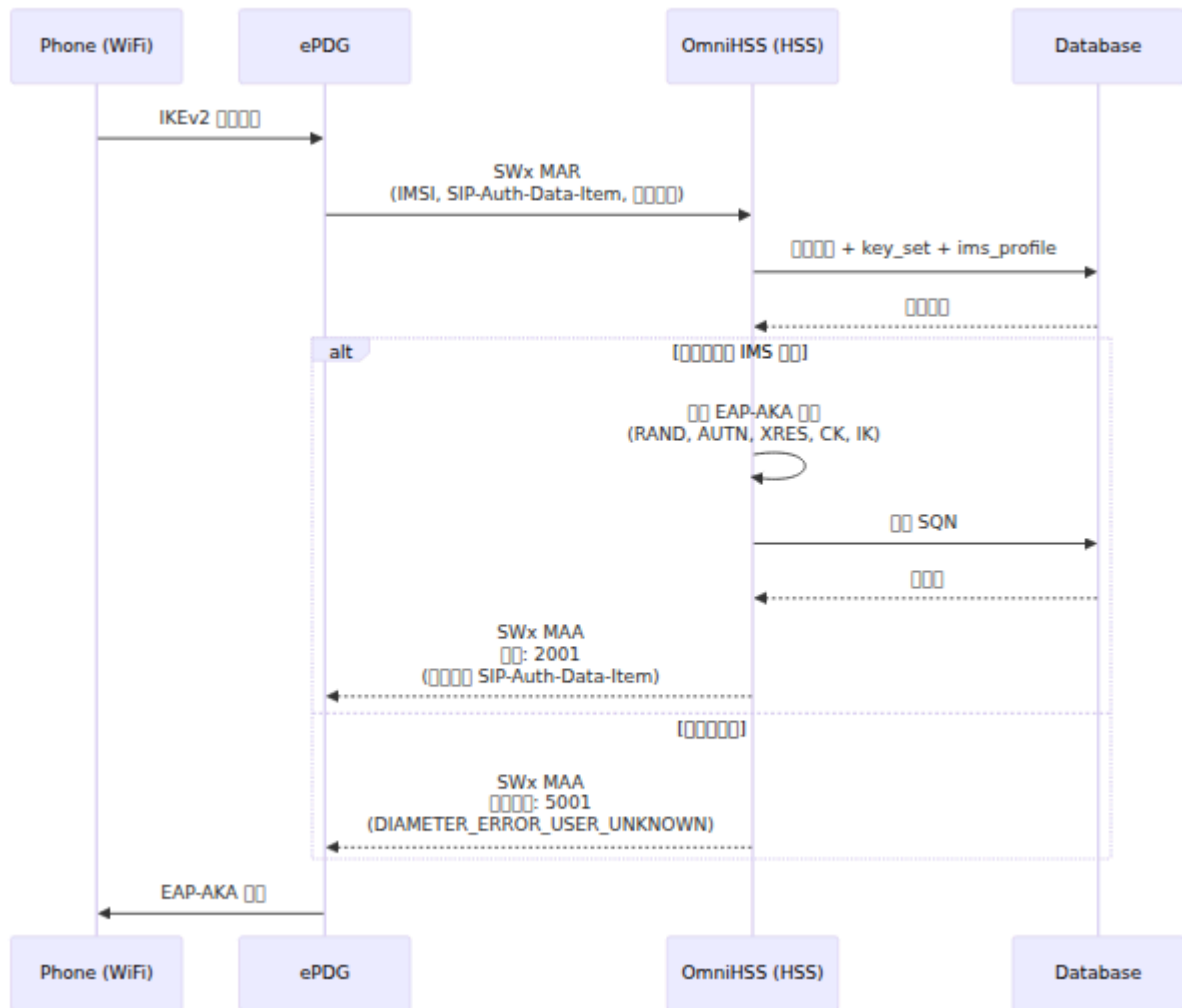
注册消息 (UAR/UA)

I-CSCF 注册消息



Authentication (SAR/SAA)

S-CSCF IMS/Non-IMS Authentication

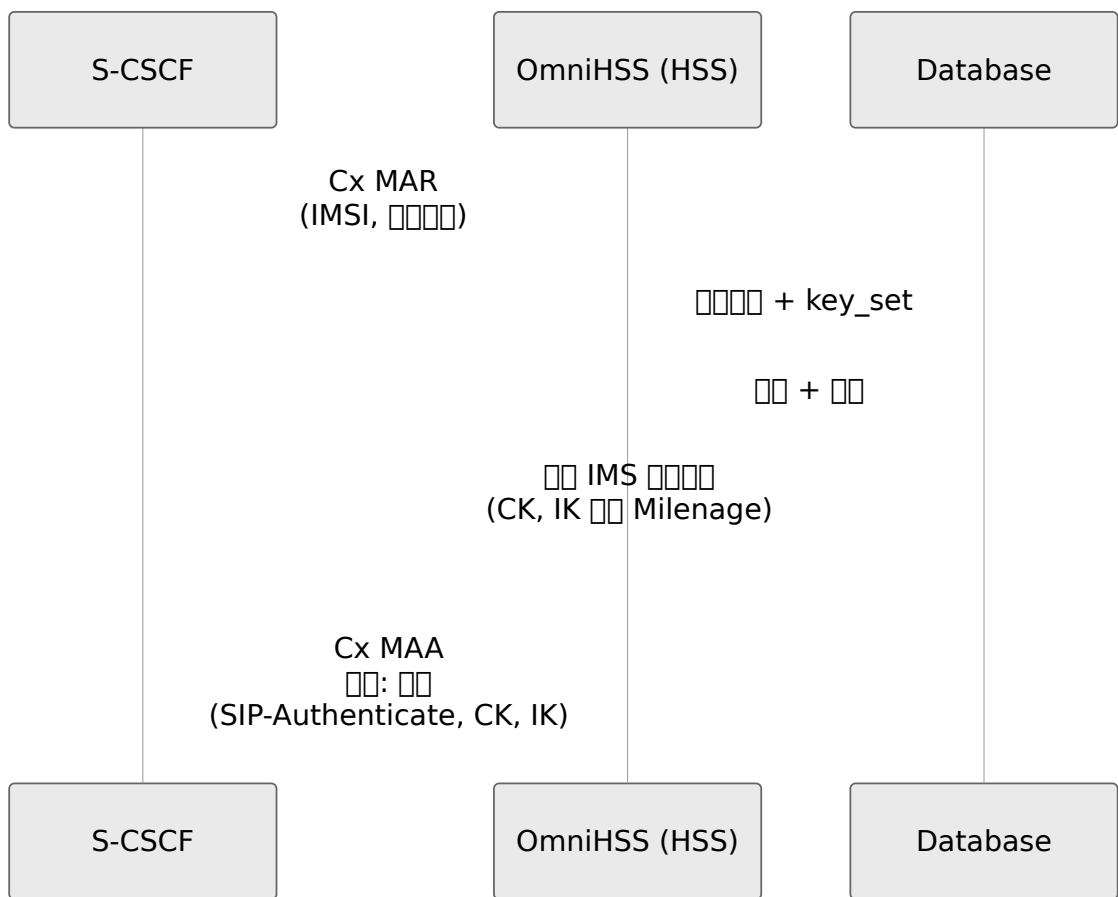


IFC :

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

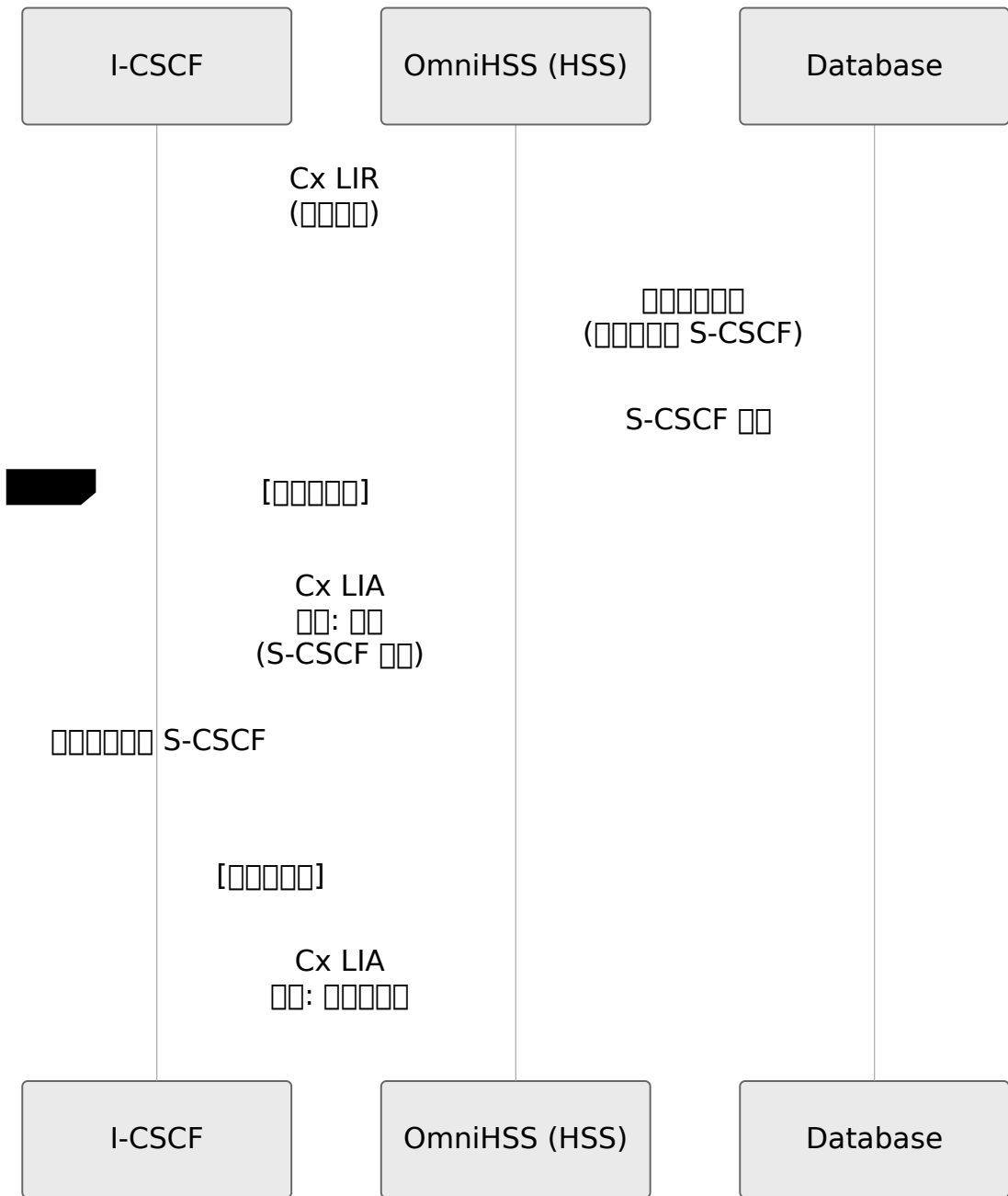
Authentication (MAR/MAA)

S-CSCF IMS Authentication



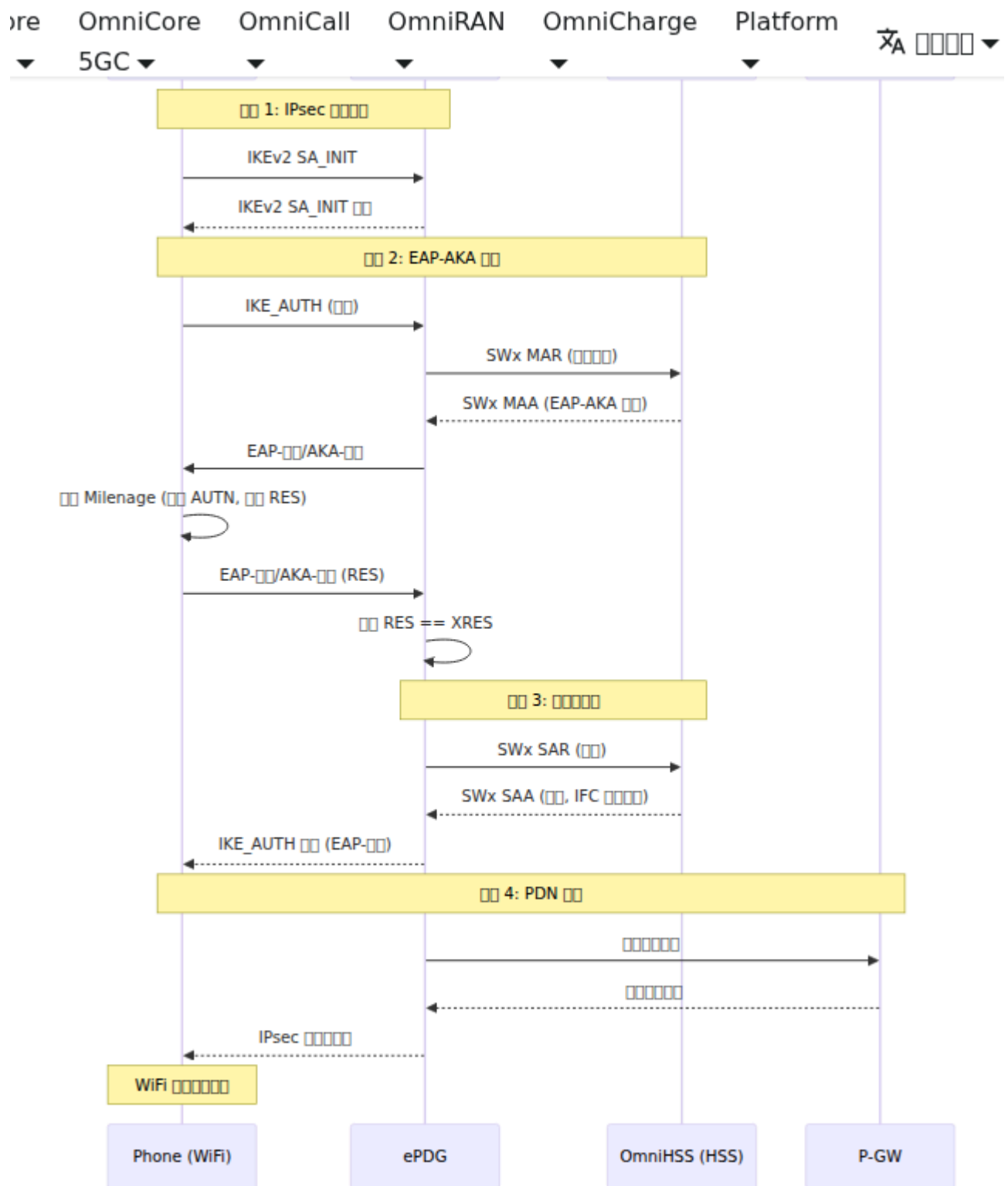
000000 (LIR/LIA)

I-CSCF 0000 S-CSCF 00000000



Sh (IMS)

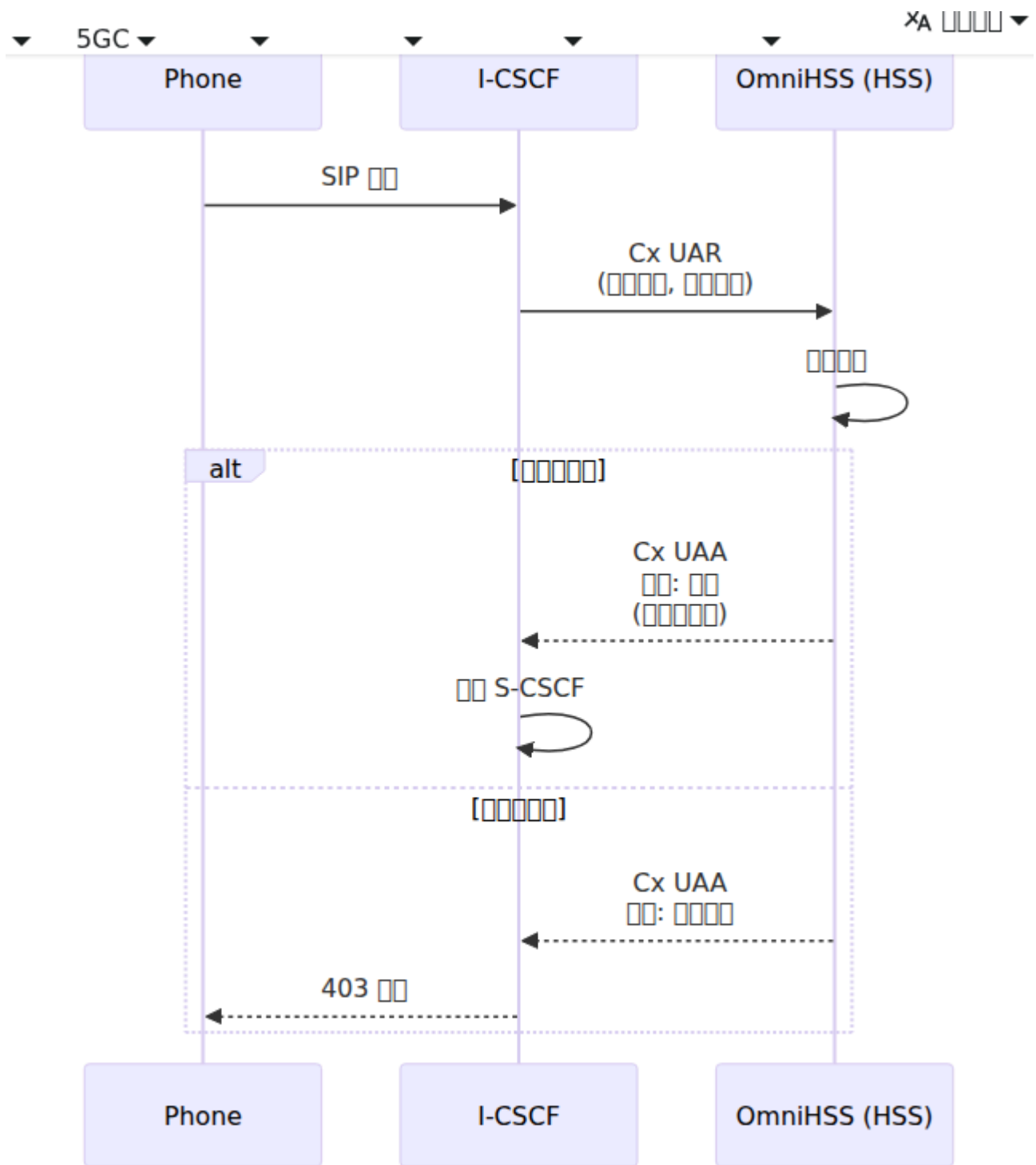
(UDR/UDA)



Authentication (PUR/PUA)

Authentication and Authorization

□ □ □ □ □ □ □ □ □ □ □ □ □ □

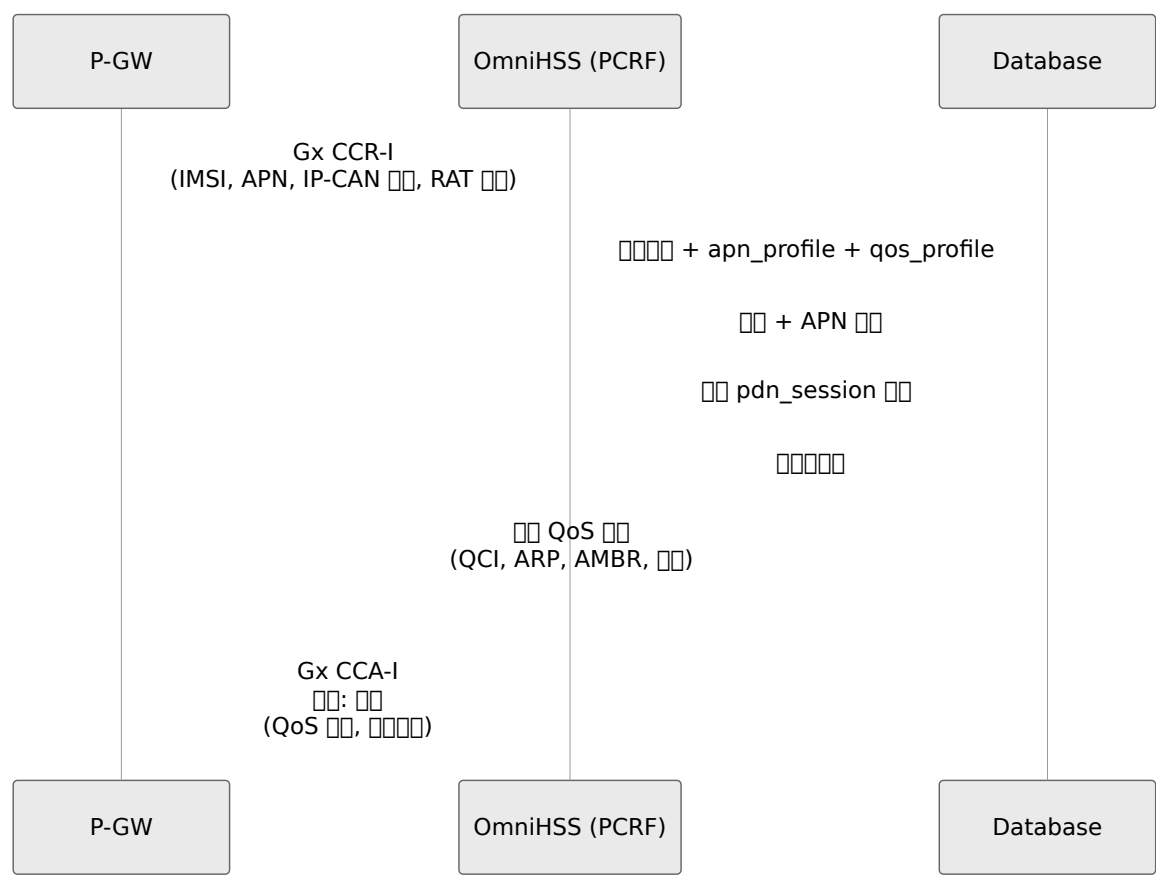


Gx □□ (□□□□)

PCRF QoS

网络策略 - 下行 (CCR-I/CCA-I)

P-GW 下行 PDN 网络策略

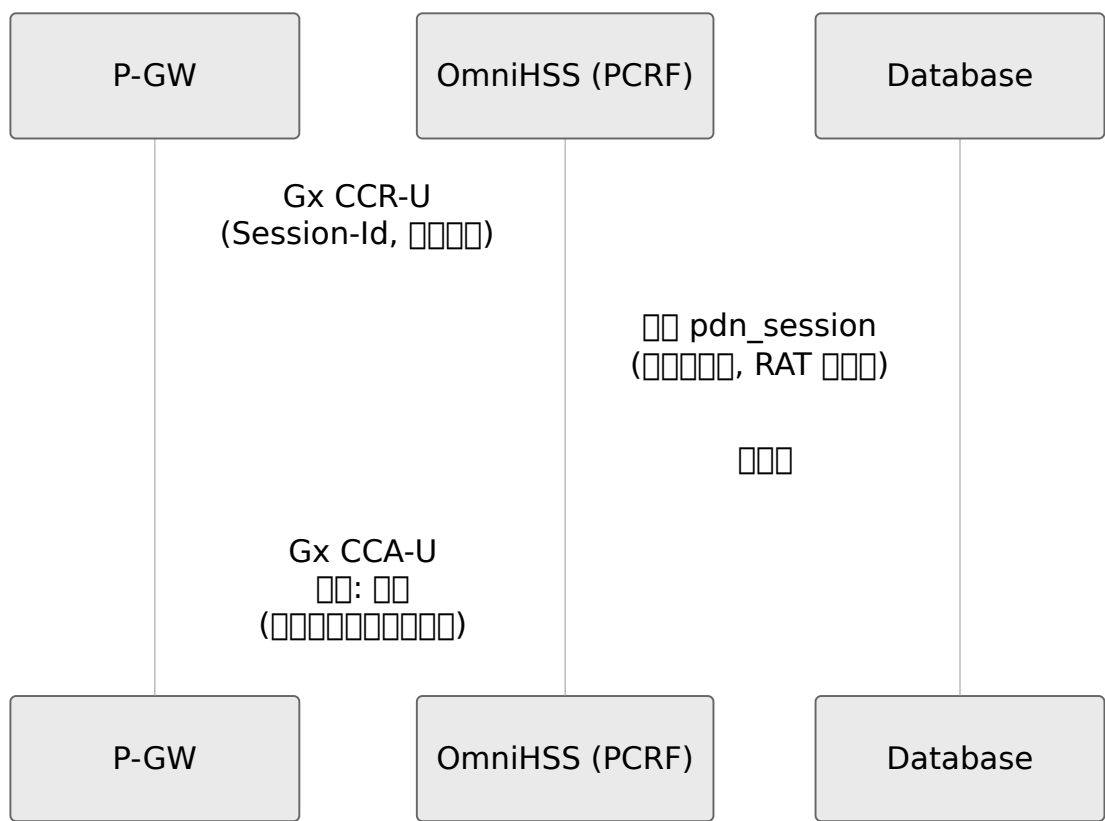


AVP:

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

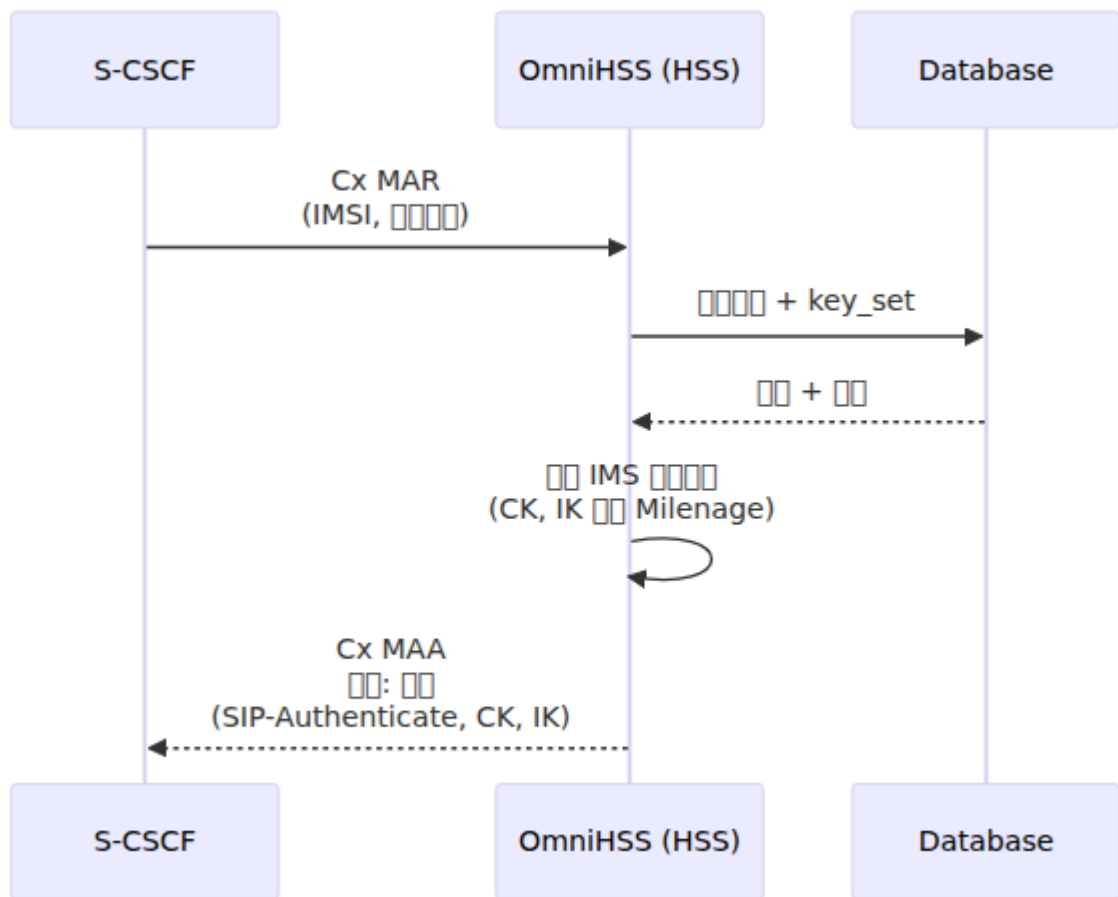
网络策略 - 上行 (CCR-U/CCA-U)

P-GW 上行



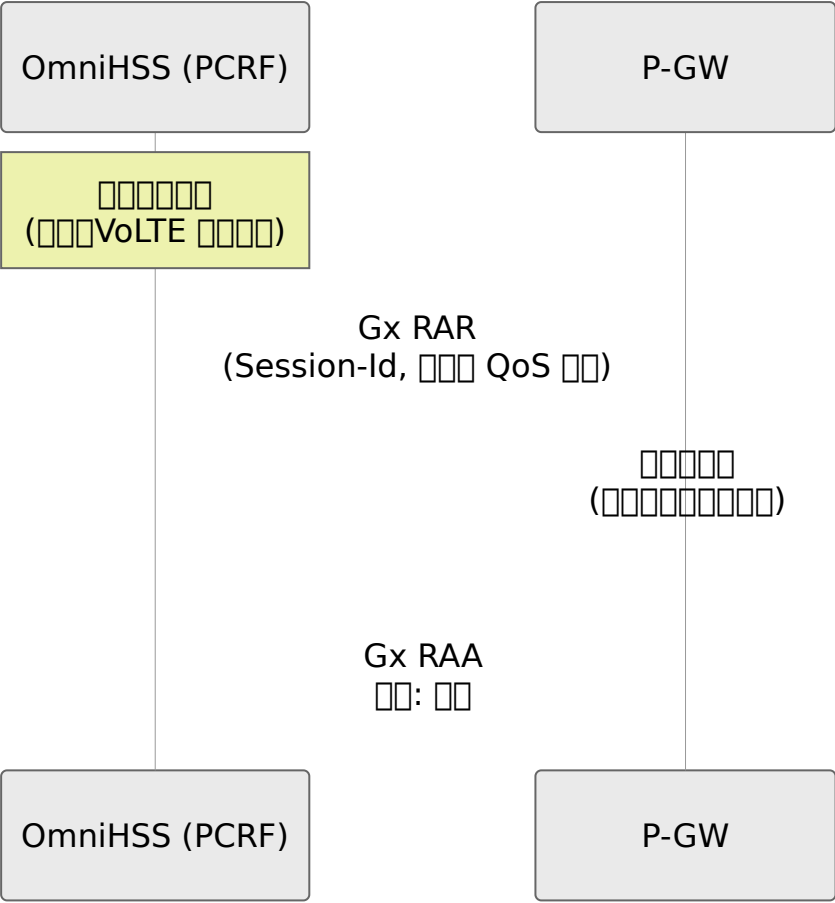
[] - [] (**CCR-T/CCA-T**)

P-GW [] PDN []



(RAR/RAA)

OmniHSS (PCRF) P-GW



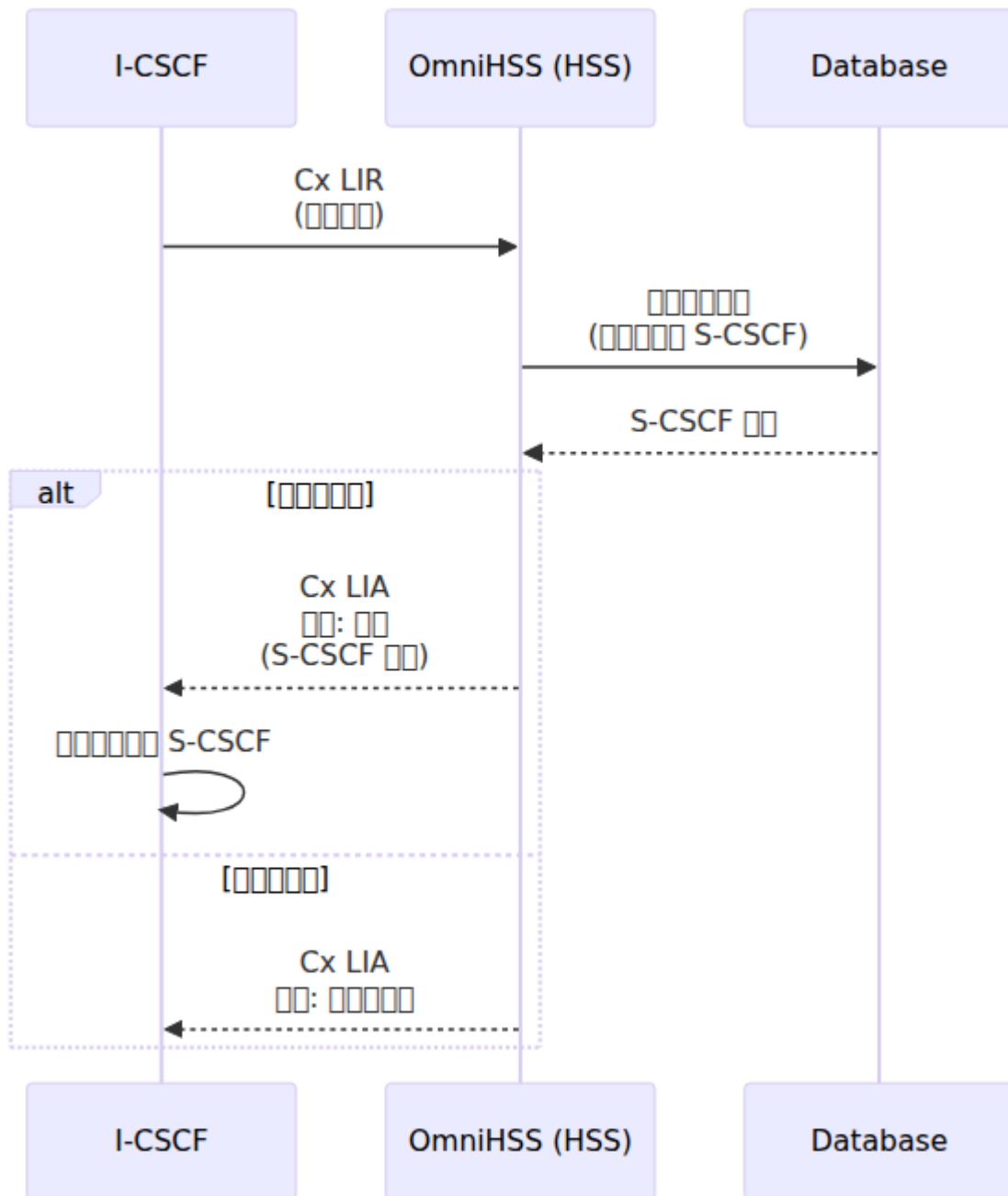
Rx (IMS)

OmniHSS Rx PCRF IMS

PCRF VoLTE

AA (AAR/AAA)

P-CSCF IMS

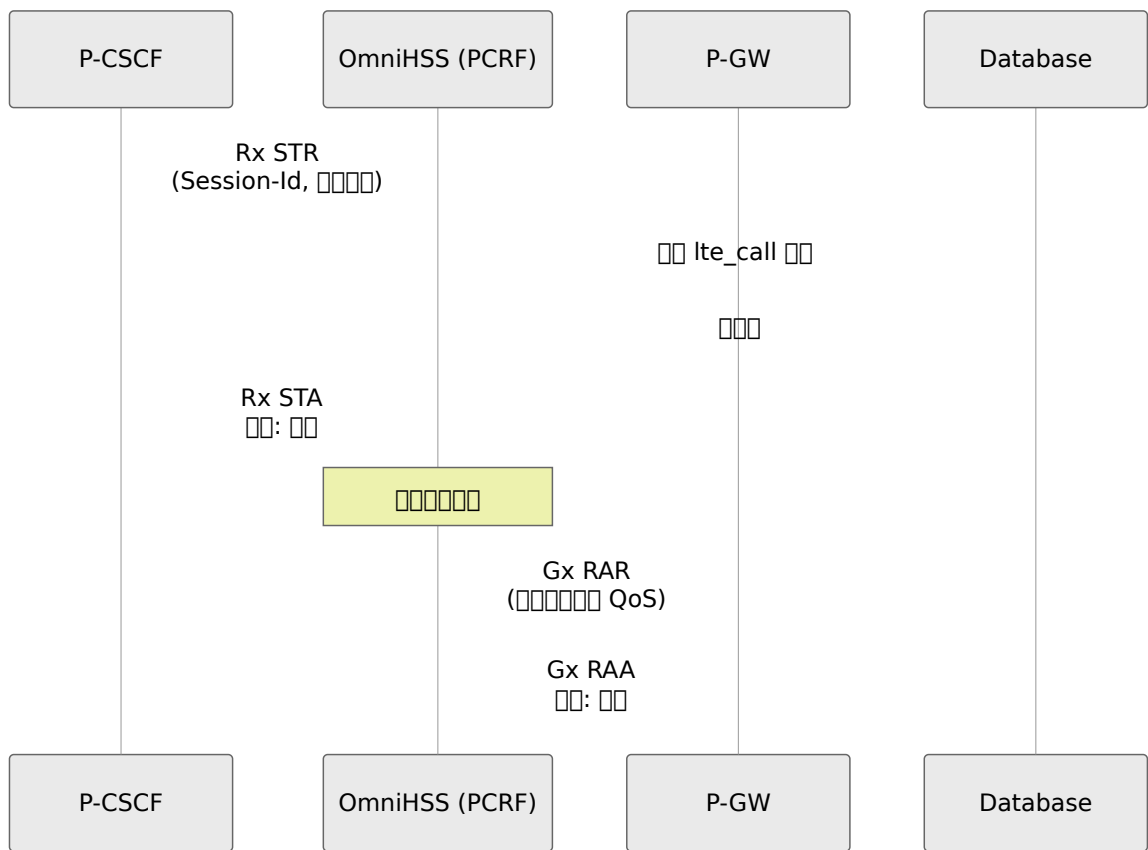


□□□□:

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

□□□□□□ (STR/STA)

P-CSCF □□ IMS □□□□□



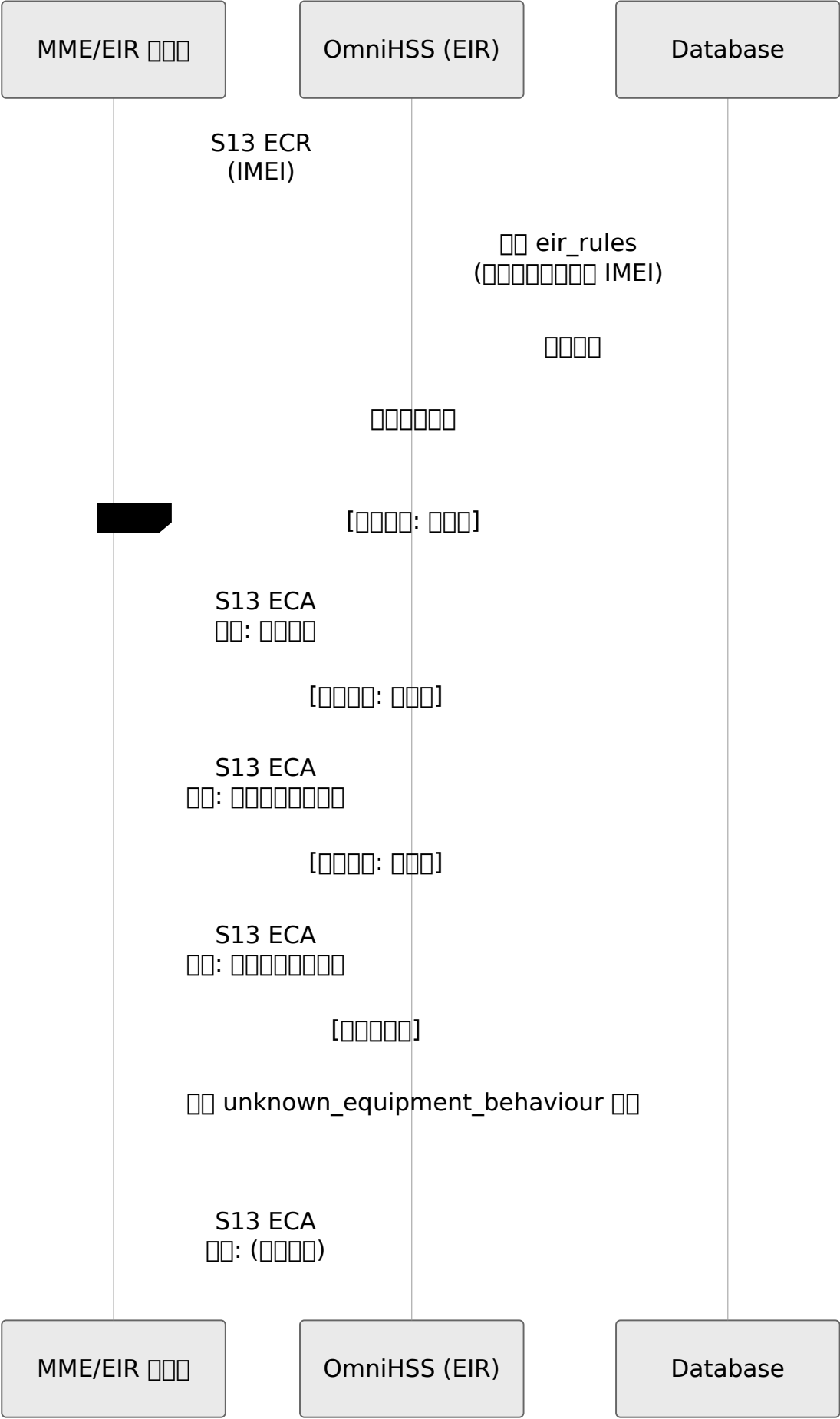
S13 (EIR)

OmniHSS S13 EIR

EIR IMEI

ME (ECR/ECA)

EIR MME

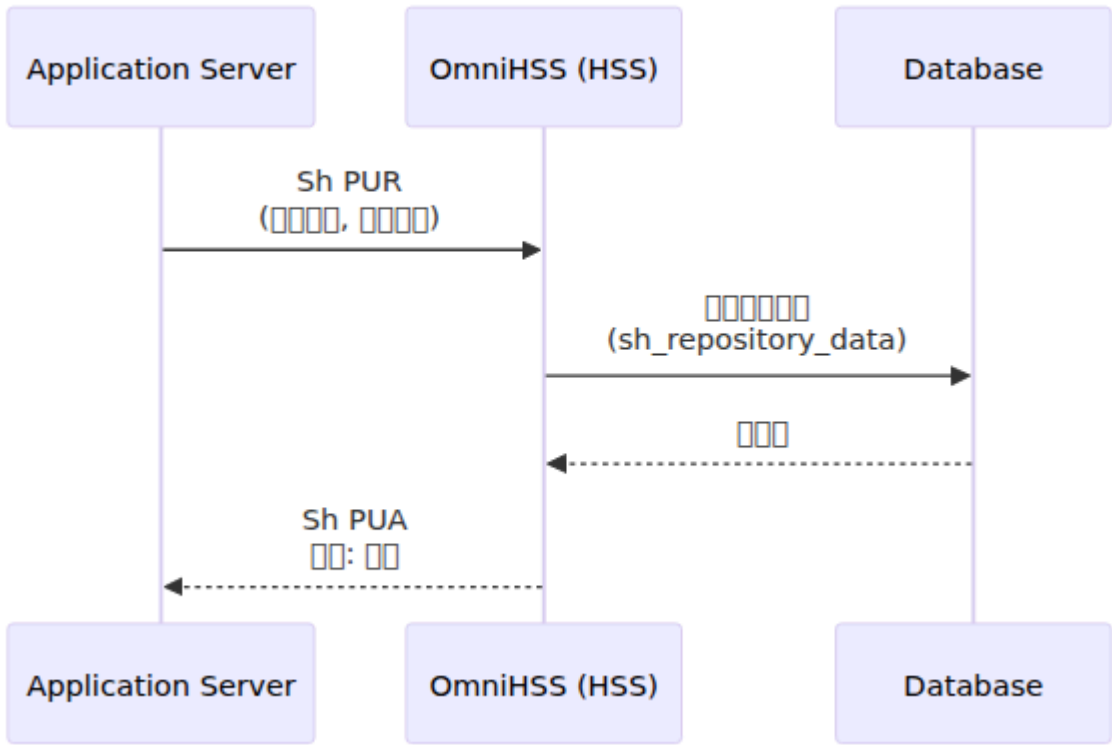


000000:

- 0000 (0) - 0000000000
- 00000000 (1) - 000000
- 00000000 (2) - 00000000

00000000: VoLTE 00

000 VoLTE 00000000000000

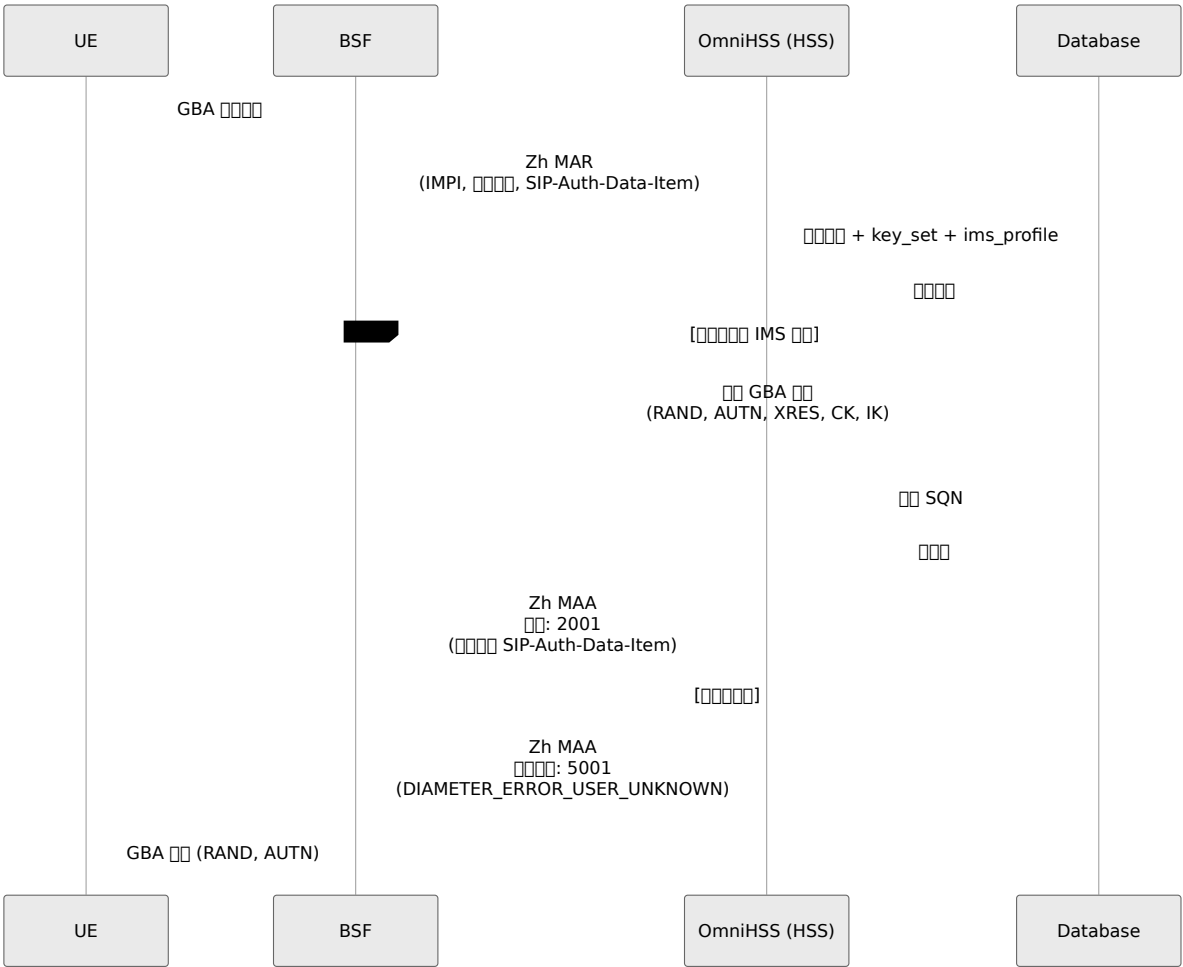


Zh 00 (00000000)

Zh 000000000000 (BSF) 000 HSS0000000000 (GBA) 000000000000

00000000 (MAR/MAA)

BSF 00 GBA 000000000000



AVP:

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

GBA parameters:

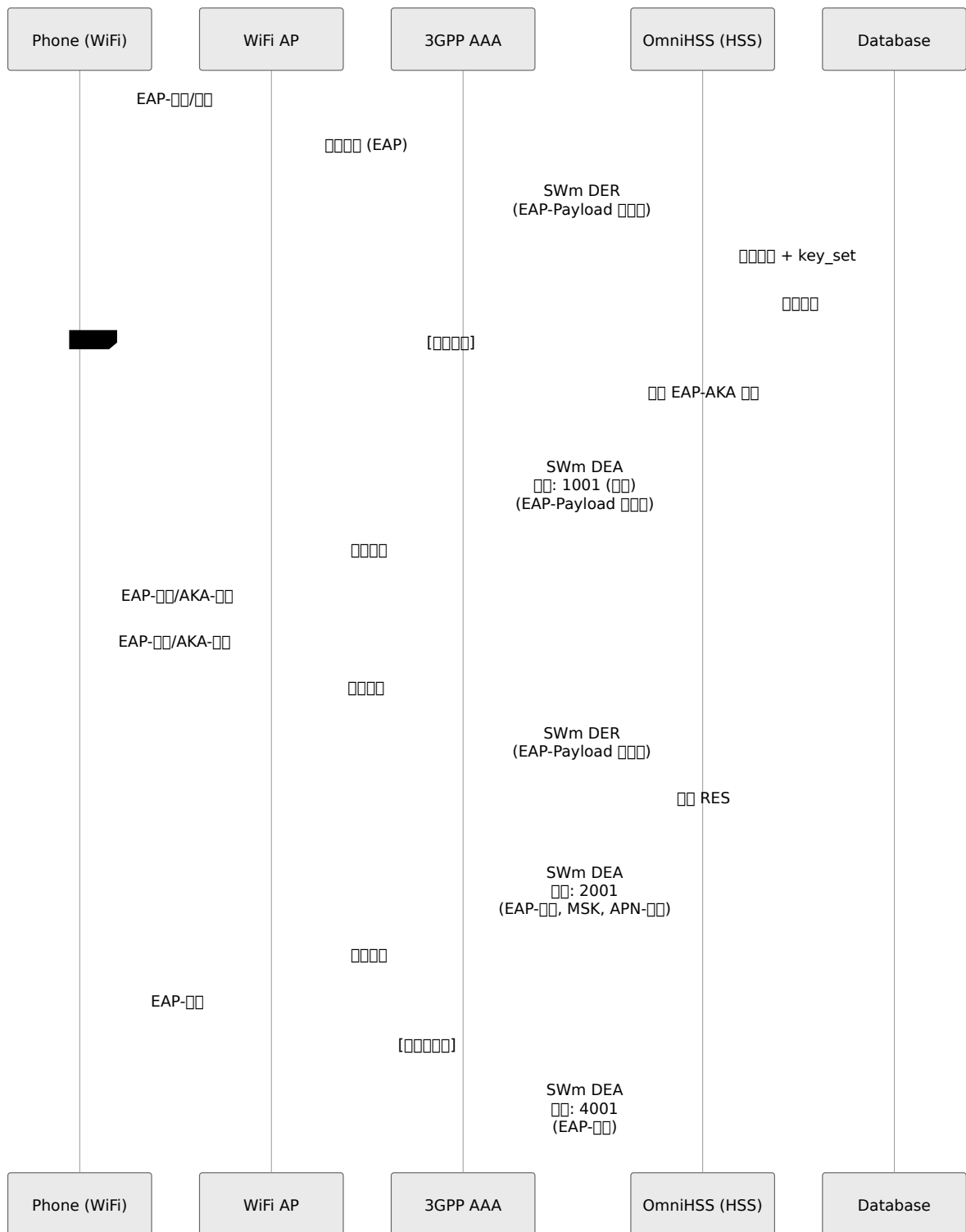
Parameter	Value
MBMS	IMSI, IMSI, IMSI
HTTPS	IMSI, IMSI, IMSI
NAF	IMSI, IMSI, IMSI
Presence	IMSI, IMSI, IMSI

SWm (3GPP)

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

Diameter-EAP- (DER/DEA)

EAP WiFi

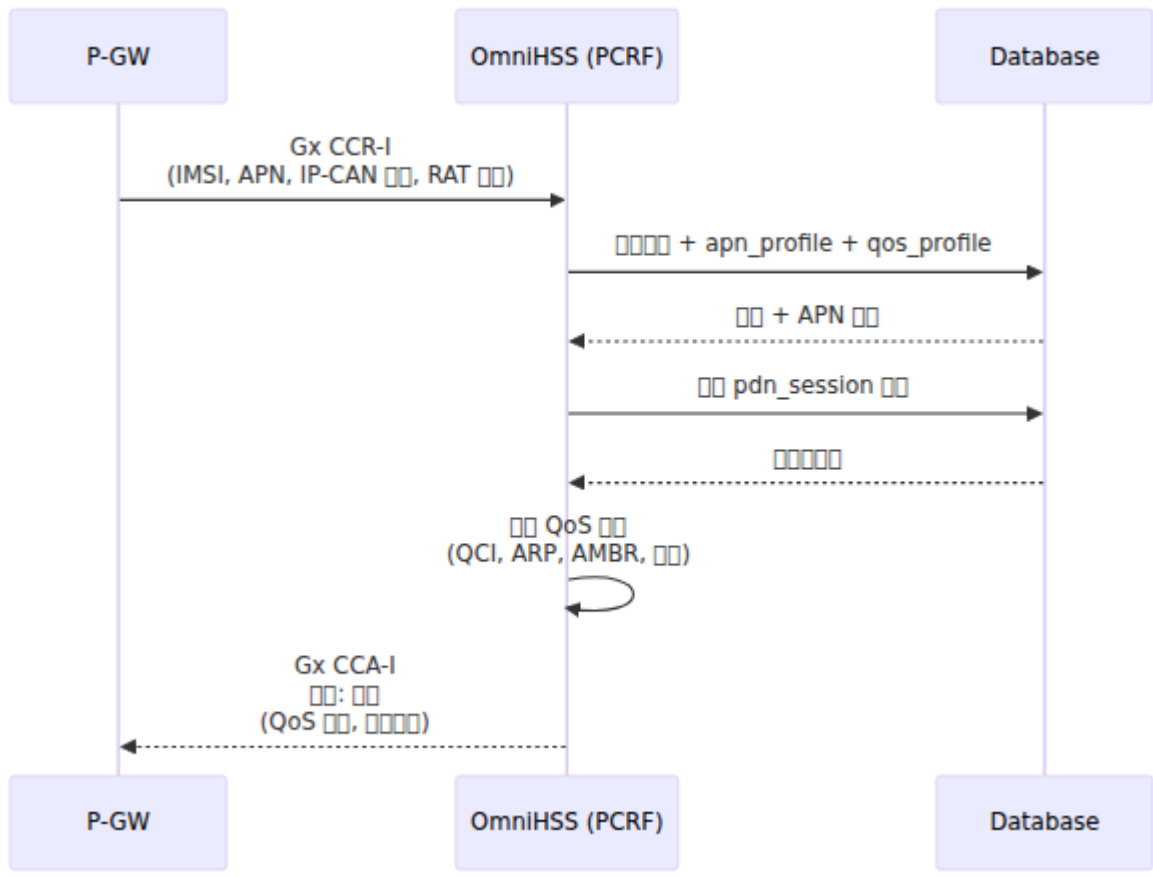


AVP:

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

AA- (AAR/AAA)

EAP

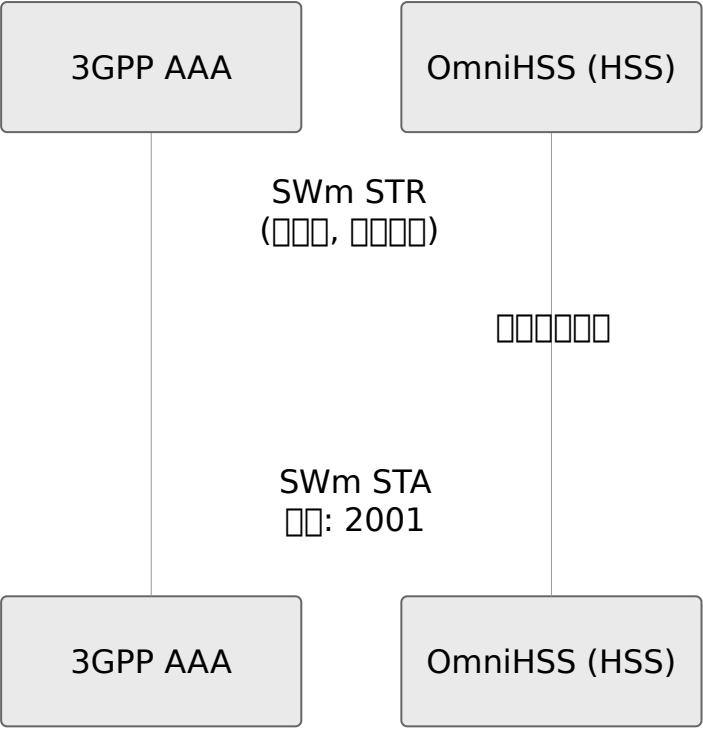


AVP:

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

(STR/STA)

WiFi



STR:

STR	STR	STR
DIAMETER_LOGOUT	1	STR
DIAMETER_SERVICE_NOT_PROVIDED	2	STR
DIAMETER_BAD_ANSWER	3	STR
DIAMETER_ADMINISTRATIVE	4	STR
DIAMETER_LINK_BROKEN	5	STR
DIAMETER_AUTH_EXPIRED	6	STR
DIAMETER_USER_MOVED	7	STR
DIAMETER_SESSION_TIMEOUT	8	STR

Network Interface

Network Interface (S6a AIR)

Parameters:

- Subscriber Identity (Ki, OPC, AMF)
- SQN (Sequence Number)
- Subscription Data

Network Interface (S6a ULR)

Parameters:

- EPC (Evolved Packet Core) APN
- Subscription Data
- MME (Mobility Management Entity) Address

IMS Network Interface (Cx SAR)

Parameters:

- IMS (IP Multimedia Subsystem) Address
- IFC (Inter-Functionality Code) XML
- S-CSCF (Serving Call Session Control Function) Address
- MSISDN (Mobile Station International Subscriber Number)

PDN Network Interface (Gx CCR-I)

Parameters:

- APN (Access Point Name) EPC (Evolved Packet Core) APN Address
 - APN QoS (Quality of Service) Parameters
 - PDN (PDN Address) Address
-

← □□□□□

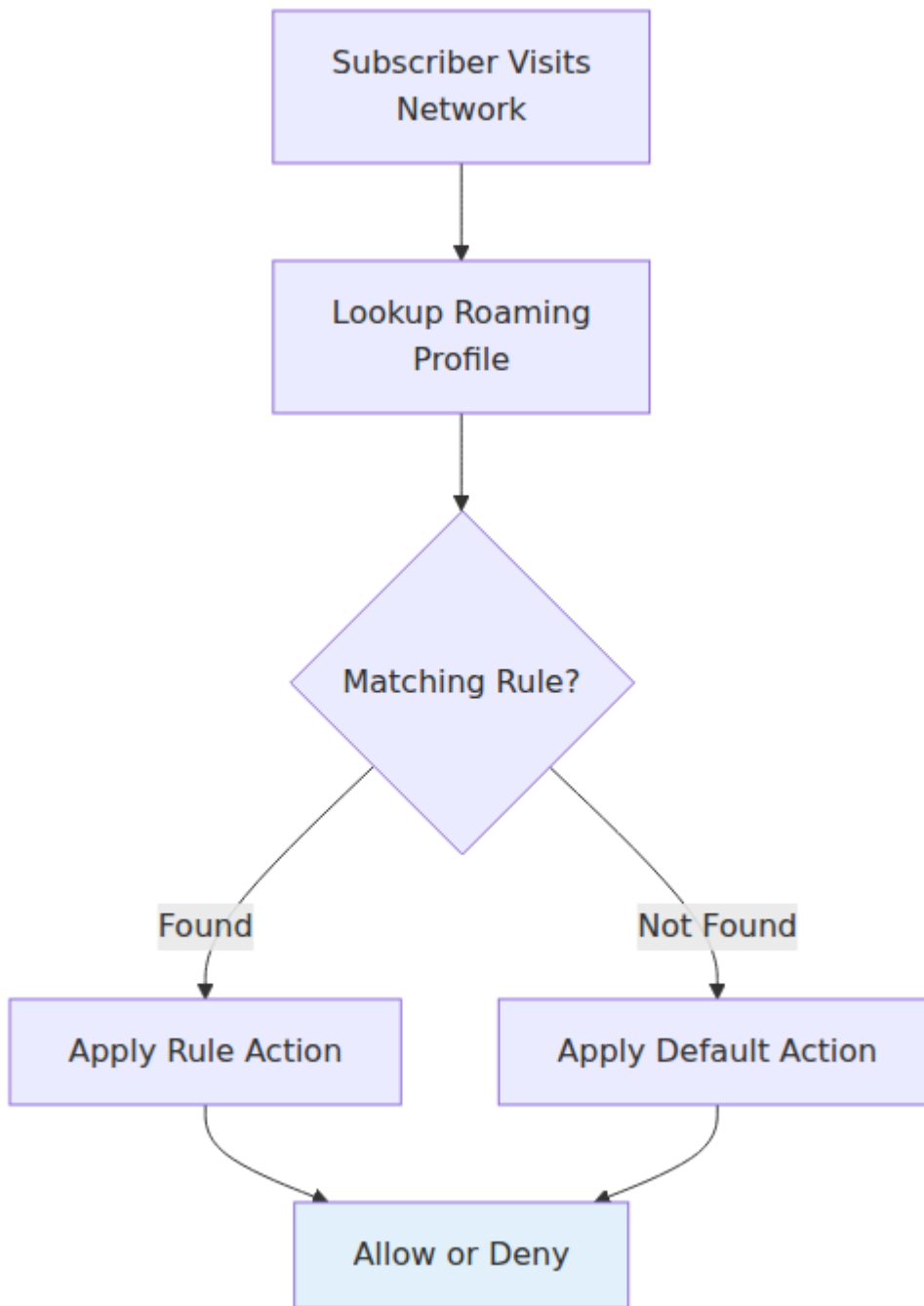
OmniHSS

←

11

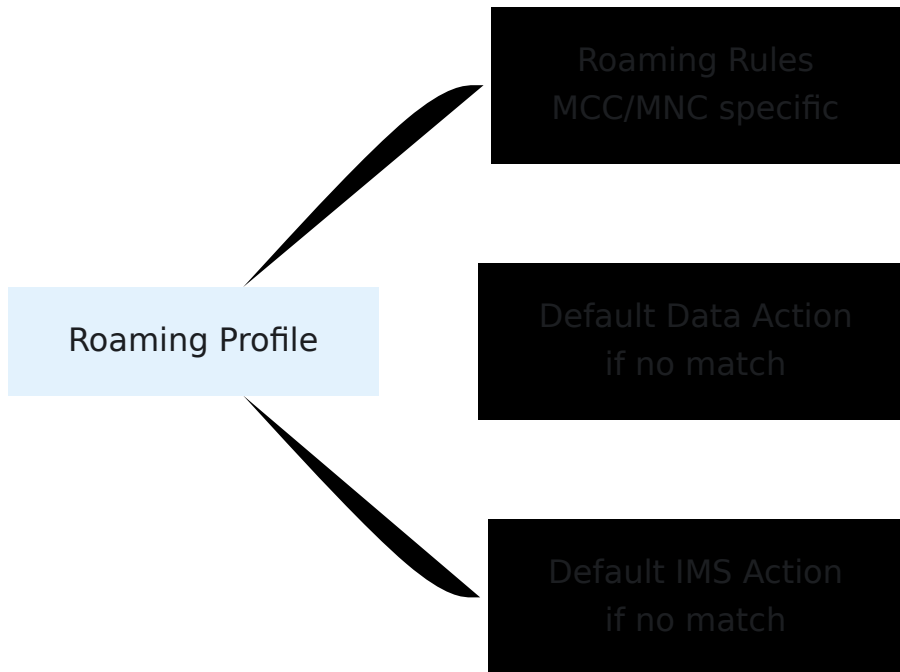
OmniHSS IMS

□□□□□□



Roaming Profile

Roaming Profile



Roaming Profile

Roaming Profile is a configuration for MCC/MNC specific

Roaming Profile

- `name` - Roaming Profile Name
- `mcc` - MCC (3 digits)
- `mnc` - MNC (2-3 digits)
- `data_action` - "allow" or "deny"
- `ims_action` - "allow" or "deny"

Roaming Profile

Roaming Profile is a configuration for MCC/MNC specific

Roaming Profile

- `name` - 任意
 - `data_action_if_no_rules_match` - "allow" 或 "deny"
 - `ims_action_if_no_rules_match` - "allow" 或 "deny"
-

任意

任意

```
# 任意
```

```
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": []
  }
}'
```

任意

```
# 任意
```

```
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": []
  }
}'
```

API Examples

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"  
  }  
}'
```

API Calls

Create Rule

```
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')
```

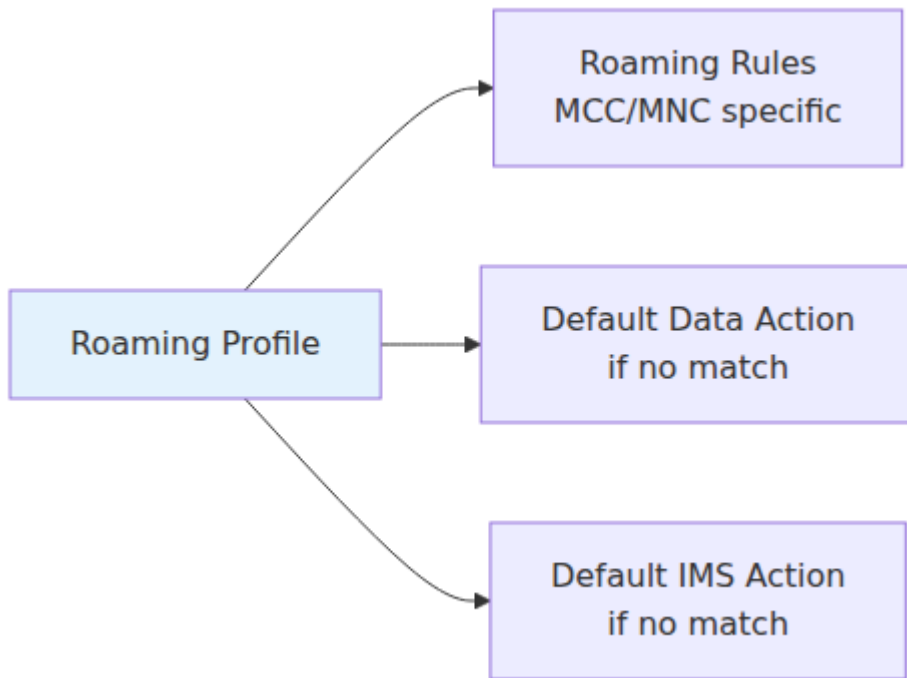
Create Profile

```
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]
  }
}"
```

Test

1. Test

Test the API calls

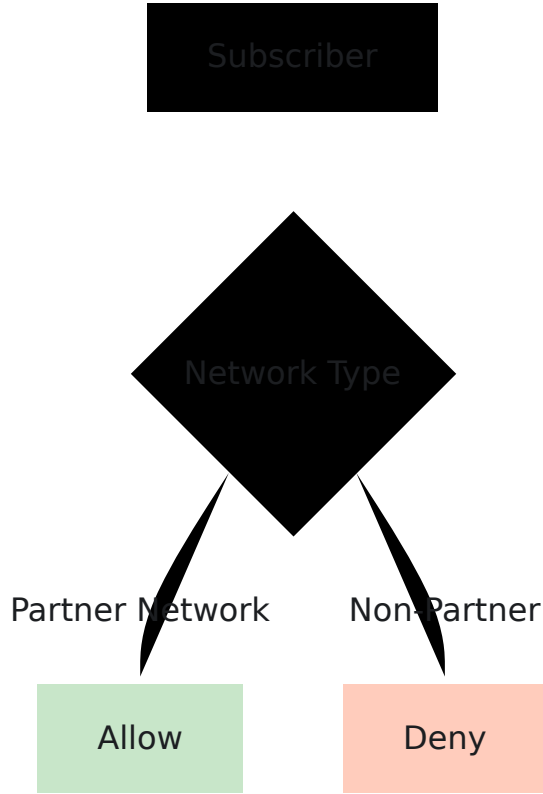


□□□

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

□□ **2**□□□□□□□□□

□□□□□□□□□□□□□□□□

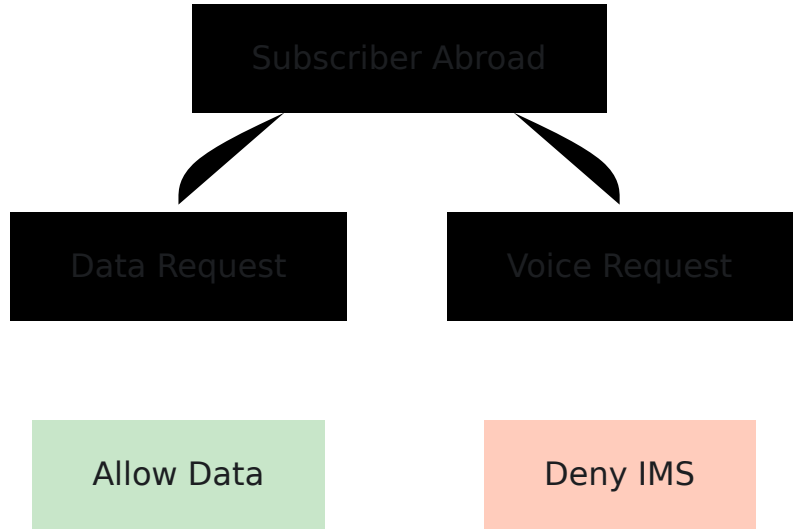


□□□

- □□□□□□□
- □□□□□□□□□□□□□□□□ MCC/MNC□

□□ **3**□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ WiFi □□□□□□□



□□□

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

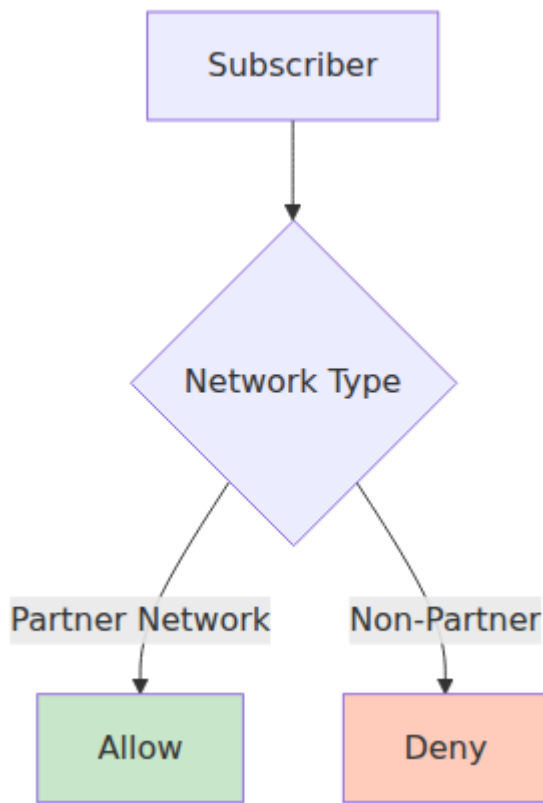
4

MME/OmniHSS

MCC/MNC

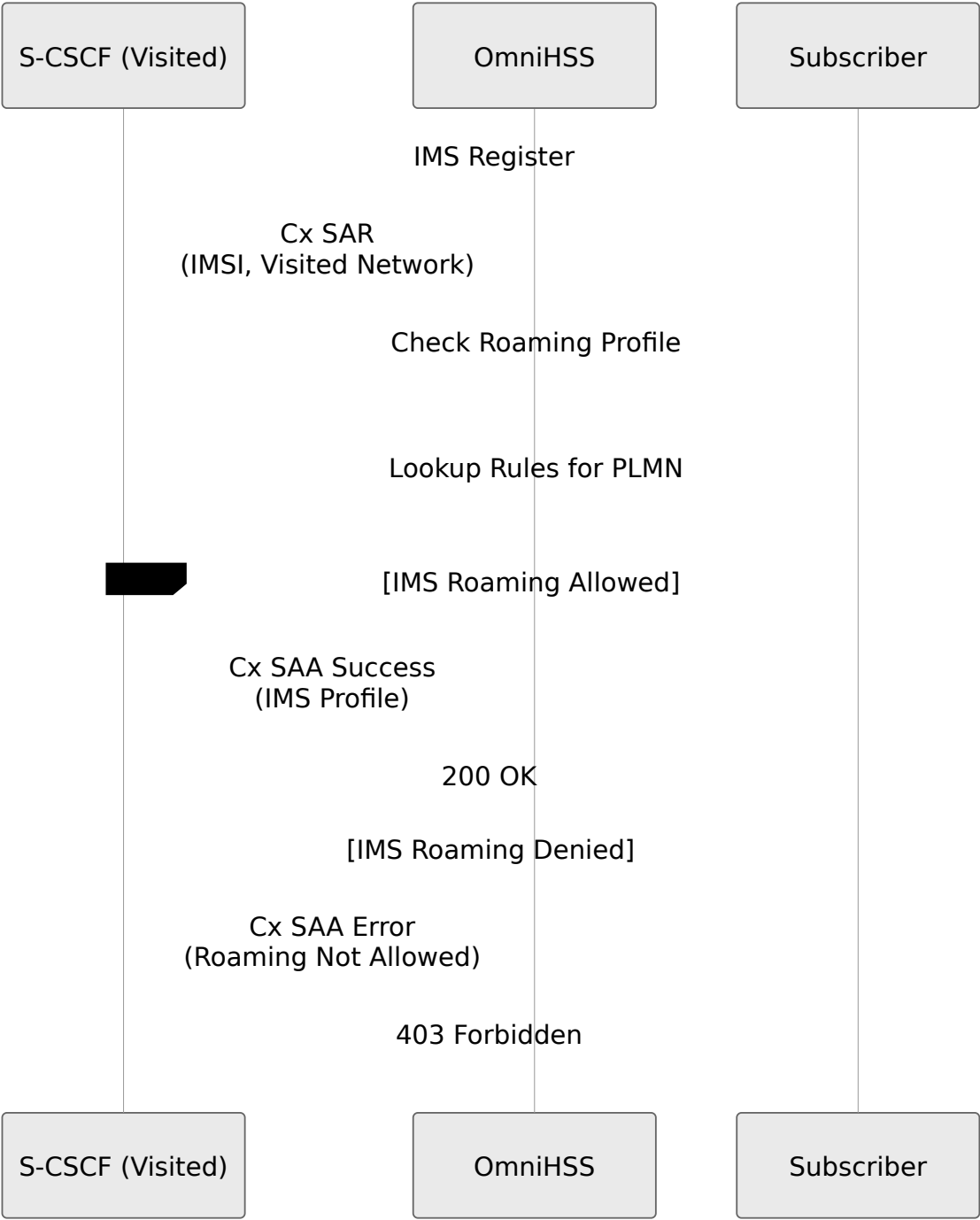
MCC

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



Cx IMS

IMS



□□□□□□

□□□□□□□□□□□□

□□□□□□□□

- □□□□□□□□□□□□□□□□

- 國際漫游功能

國際漫游功能

- 國際漫游功能 MCC/MNC 設定
- 國際漫游功能
- 國際漫游功能 data_action

國際漫游功能 IMS

國際漫游功能 IMS

- 國際漫游功能
- data_action ims_action
- 國際漫游功能 IMS

國際漫游功能

國際漫游功能

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

國際漫游功能

國際漫游功能

1. 國際漫游功能 - 國際漫游功能
2. 國際漫游功能 - 國際漫游功能
3. 國際漫游功能 - 國際漫游功能
4. 國際漫游功能 - 國際漫游功能

□□□□

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

□□□□

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

OmniHSS

□□□□□□

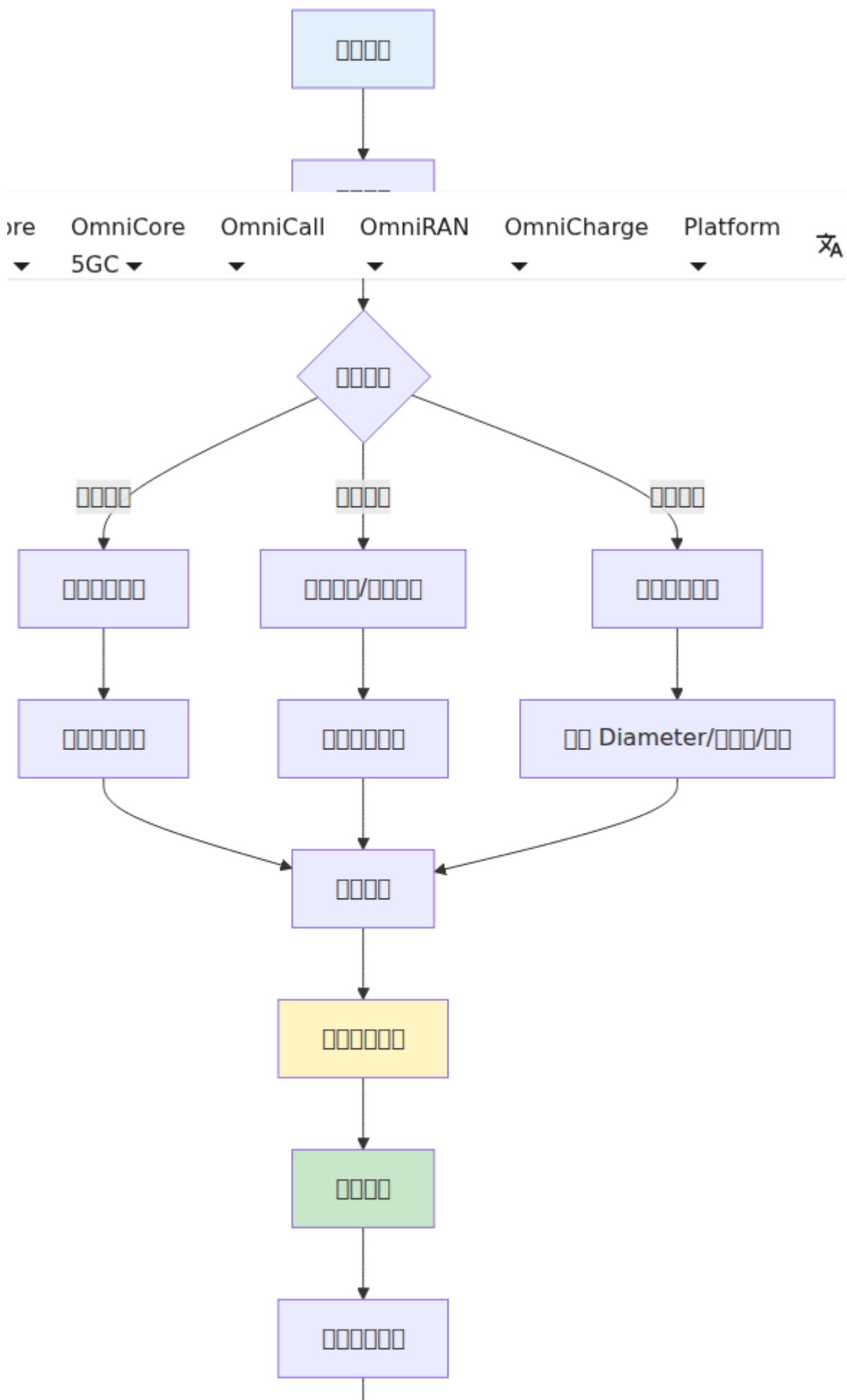
← □□□□□

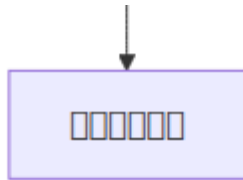
□□

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

□□□□□□

□□□□□□□





□□□□

□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□

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

2. □□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

3. □□□□

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

4. □□□□

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

API

API

- API
- “API”
- API

API

API 1

API

- API
- API

API

1. API key_set_id

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

2. API

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

3. Ki OPC SIM

API

- API
- SIM

API 2 SQN

API

- 查詢所有用戶
- 查詢“SQN 號碼”
- 查詢用戶

查詢用戶

1. 查詢所有用戶的 SQN 值
2. 查詢用戶的 SQN 號碼
3. 查詢用戶的 SQN 值

查詢用戶

- 查詢 AUTS 的 SQN 號碼
- 查詢用戶的 SQN 值 0 號碼

查詢用戶 SQN 號碼

查詢 3 號碼

查詢

- 查詢用戶
- 查詢用戶

查詢用戶

1. 查詢用戶

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

2. 查詢 `enabled` 用戶 `true`

查詢用戶

- 查詢用戶


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

4 EPC

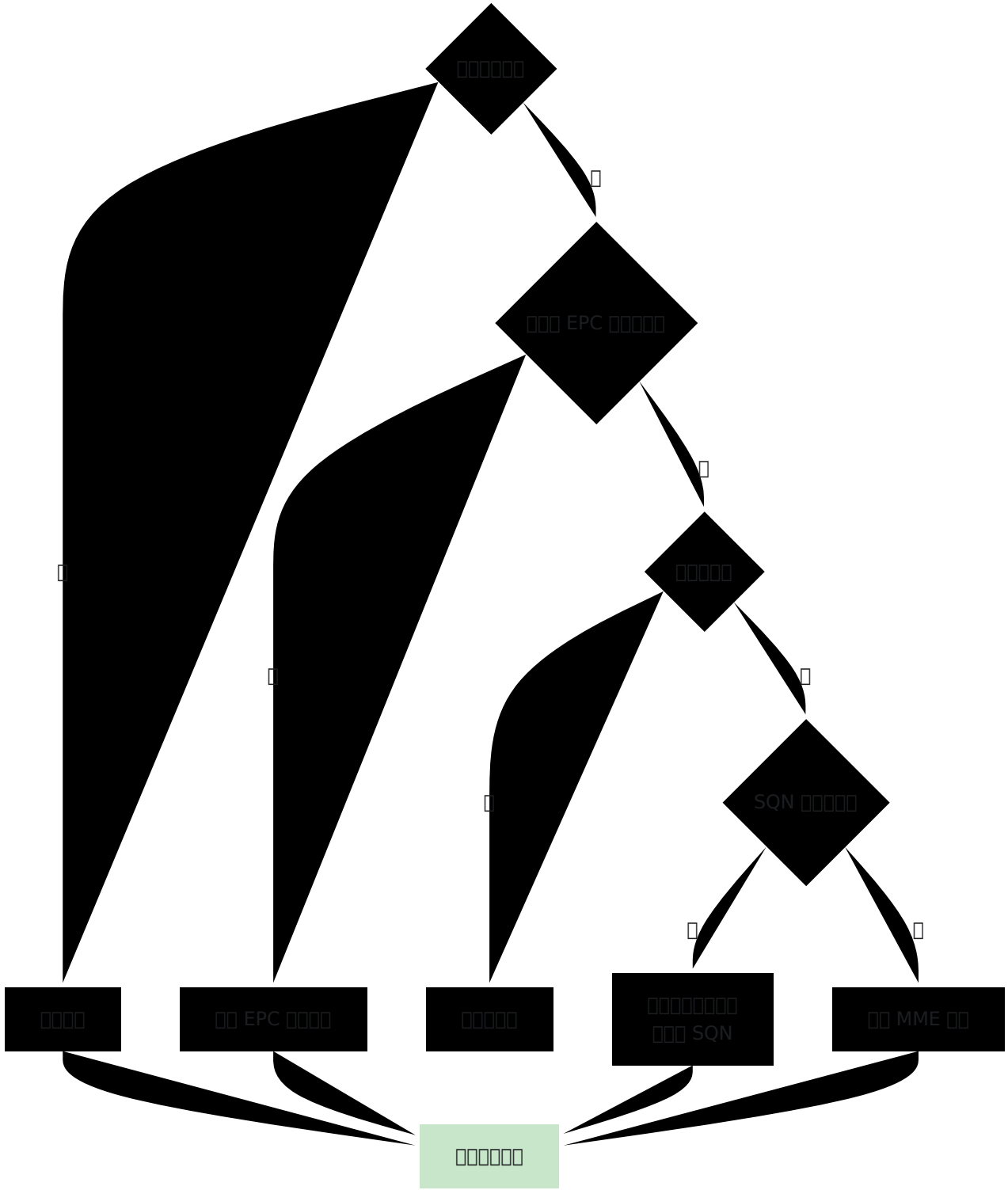
-
- “ EPC ”

1.  `epc_profile_id`
2. EPC

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

- EPC

□□□□



Diameter

11

- 4 0000 0000 Diameter 00000000
- “00000000”00
- 0000000000

00 **1**00000000

000

- 
- 
- Ping 

□□□□□

- ## 1. OmniHSS 概要

```
ping [PEER_IP]
```

- ## 2. Diameter

```
telnet [PEER_IP] 3868
```

3. Diameter 3868

□□□□□

- 
- 
- 

第2章 Diameter 配置

目标

- 配置
- CER/CEA 配置
- 配置

配置

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

配置

- 配置 Diameter 配置 runtime.exs
- 配置 OmniHSS
- 配置

第3章 TLS Diameter

目标

- TLS 配置
- 配置
- “配置”“配置”配置

配置

1. 配置 priv/cert/ 配置
2. 配置

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

3. 配置设备

4. 配置设备支持 TLS

配置

- 配置设备
- 配置设备
- 配置设备 OmniHSS

4. 配置设备

配置

- 配置设备
- 配置设备
- “配置设备”

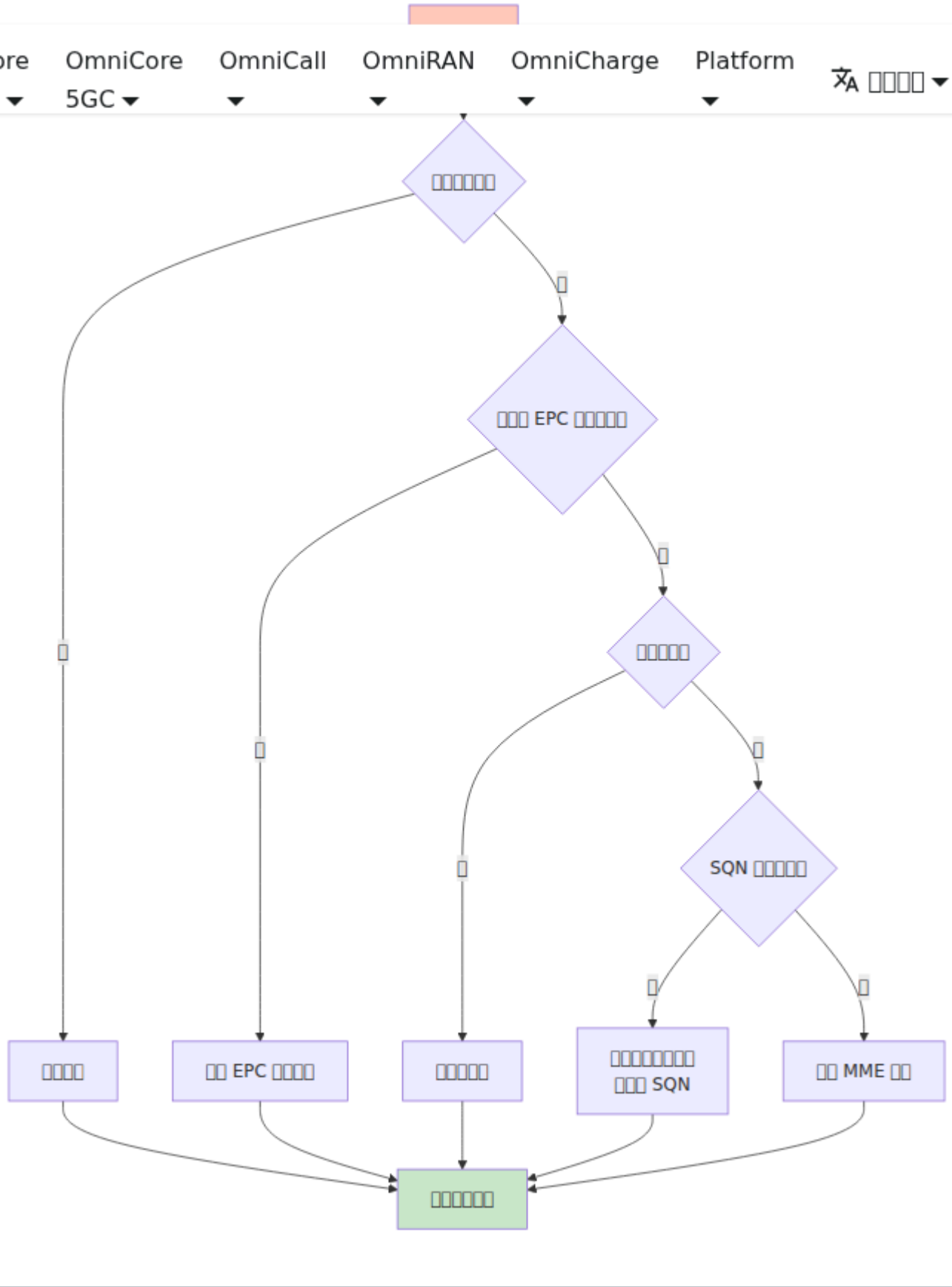
配置

1. 配置 Diameter 设备
2. 配置设备 S6a/Cx/Sh
3. 配置 CER/CEA

配置

- 配置设备 Diameter
- 配置设备
 - MME 设备 S6a 16777251
 - S-CSCF 设备 Cx 16777216
 - P-GW 设备 Gx 16777238

Diameter 网络架构



環境構築

前提

- API 500 行
- テーブル構造
- “テーブル”名
- テーブル

環境構築

1. テーブル構造

環境

- API 500 行
- テーブル構造
- “テーブル”名

環境

1. テーブル構造

```
# テーブル PostgreSQL
psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME]

# テーブル MySQL
mysql -h [DB_HOST] -u [DB_USER] -p [DB_NAME]
```

2. テーブル構造

3. テーブル構造

環境

- テーブル構造
- テーブル構造

- 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. 00000000000000
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

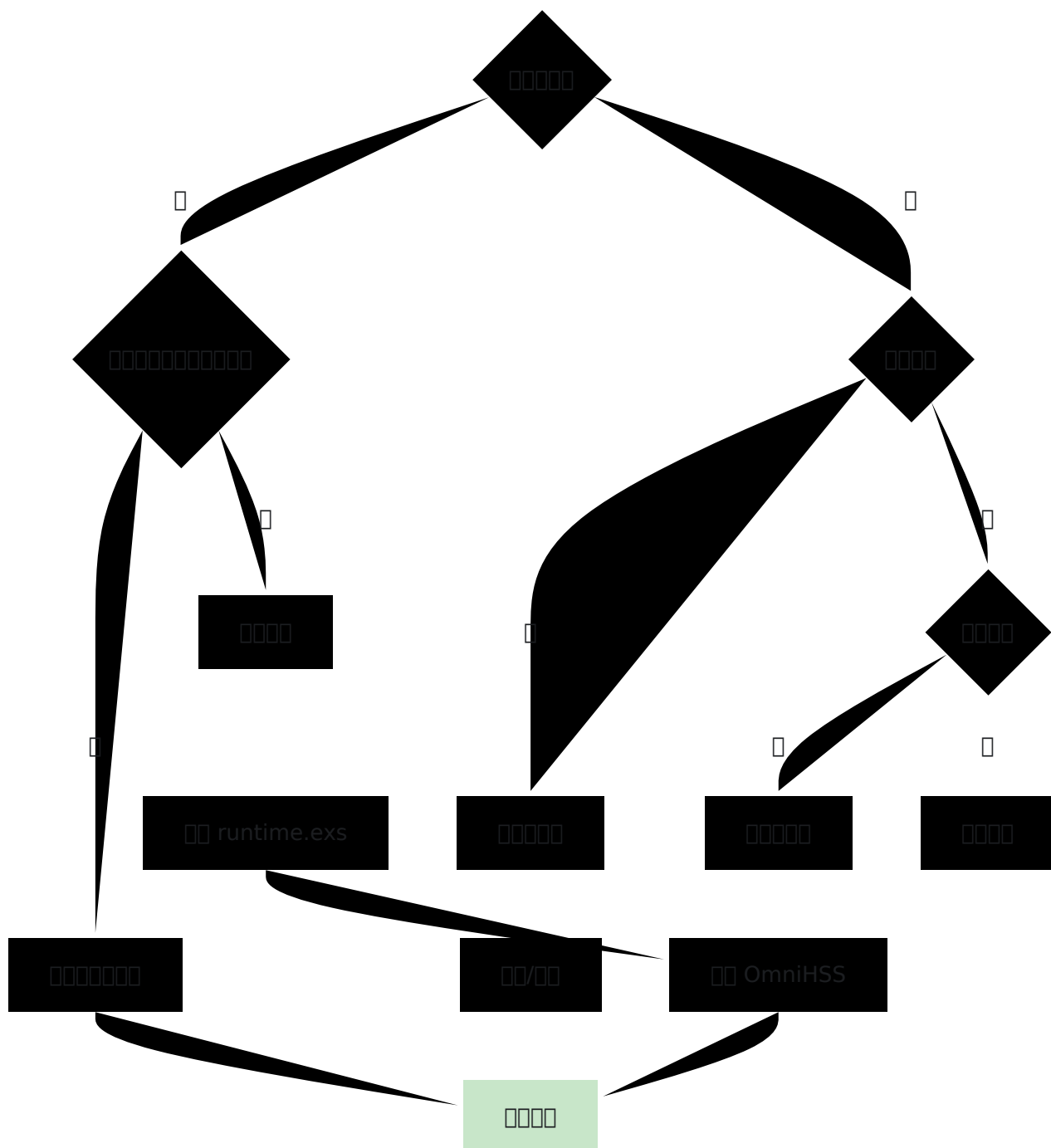
1. 000000000000
2. 00000000
3. 00000000
4. 000000000000

00000

- 00000
- 00000000
- 0000000000
- 000000000000

□ □ □ □ □ □ ◆ ◆ ◆ □ □

□□□□□



EPC 网络

网络

- 网络核心 LTE 网络
- MME 网络
- 网络 PDN 网络

网络核心

网络 1 网络

网络

- 网络核心网络
- “网络”网络
- 网络核心网络

网络

1. 网络 roaming_profile_id
2. 网络
3. 网络 MCC/MNC
4. 网络 网络

网络

- 网络 MCC/MNC 网络
- 网络
- 网络 网络

网络 2 网络 APN 网络

网络

- 网络 PDN 网络
- MME 网络“网络 APN”网络

- 認證與授權

認證

1. 向 EPC 註冊並獲取 APN 資訊
2. 向 APN 請求網路資源
3. 向 APN 請求認證

認證

- 向 APN 請求 EPC 認證
- 向 APN 請求授權
- 向 APN 請求 QoS 資訊

第 3 步 MME 認證

認證

- 認證與授權
- MME 認證
- Diameter 認證

認證

1. 向 Diameter 請求
2. MME 請求“認證”
3. MME 請求 S6a 認證

認證

- Diameter 認證
- MME 認證
- MME 認證

第 4 步認證

認證

- 認證與授權

- 00000000
- 00000000

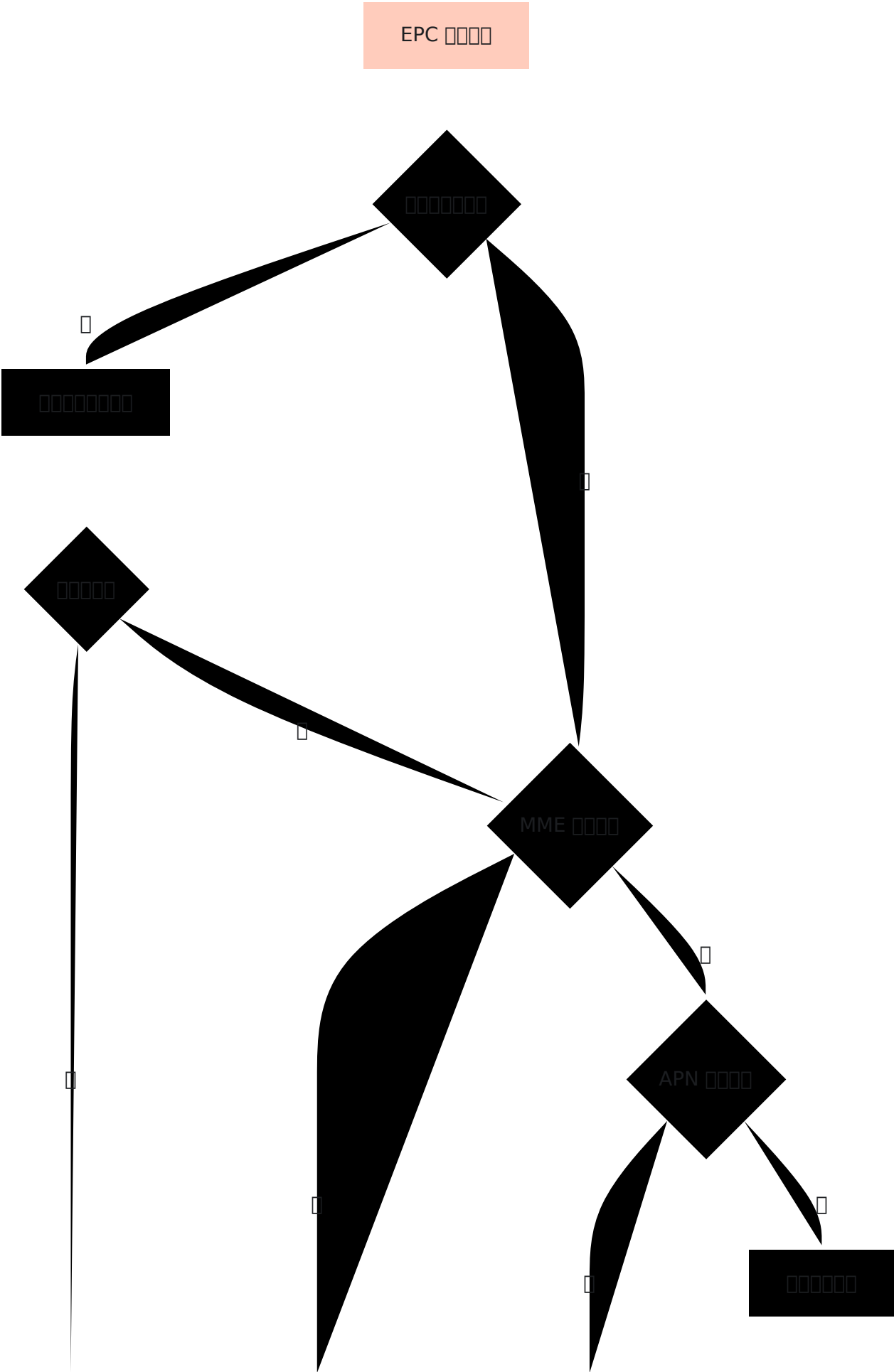
000000

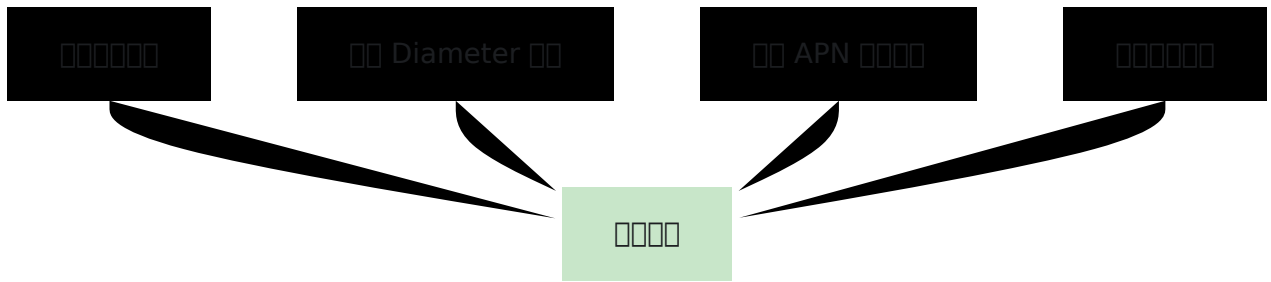
1. 0000000000
2. 0000000000 MME 00
3. 0000000000

000000

- 000000000000
- 000000000000 MME
- 000000000000

EPC □□□□□□□□





IMS

IMS

- IMS VoLTE
- IMS “IMS”
- IMS

IMS

IMS 1 IMS

IMS

- IMS IMS
- IMS

IMS

- IMS `ims_enabled`
- IMS `ims_profile_id`

IMS

- 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 3GPP 網路 IFC 網路

網路

- IMS 網路 3GPP 網路
- IMS 網路 IFC 網路

網路

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

網路

- IMS 網路 IFC 網路 IMS 網路
- IMS 網路 3GPP 網路 IFC 網路

IMS 4G 網路 IMS 網路

網路

- IMS 網路 3GPP 網路
- IMS 網路
- IMS 網路 IMS 網路

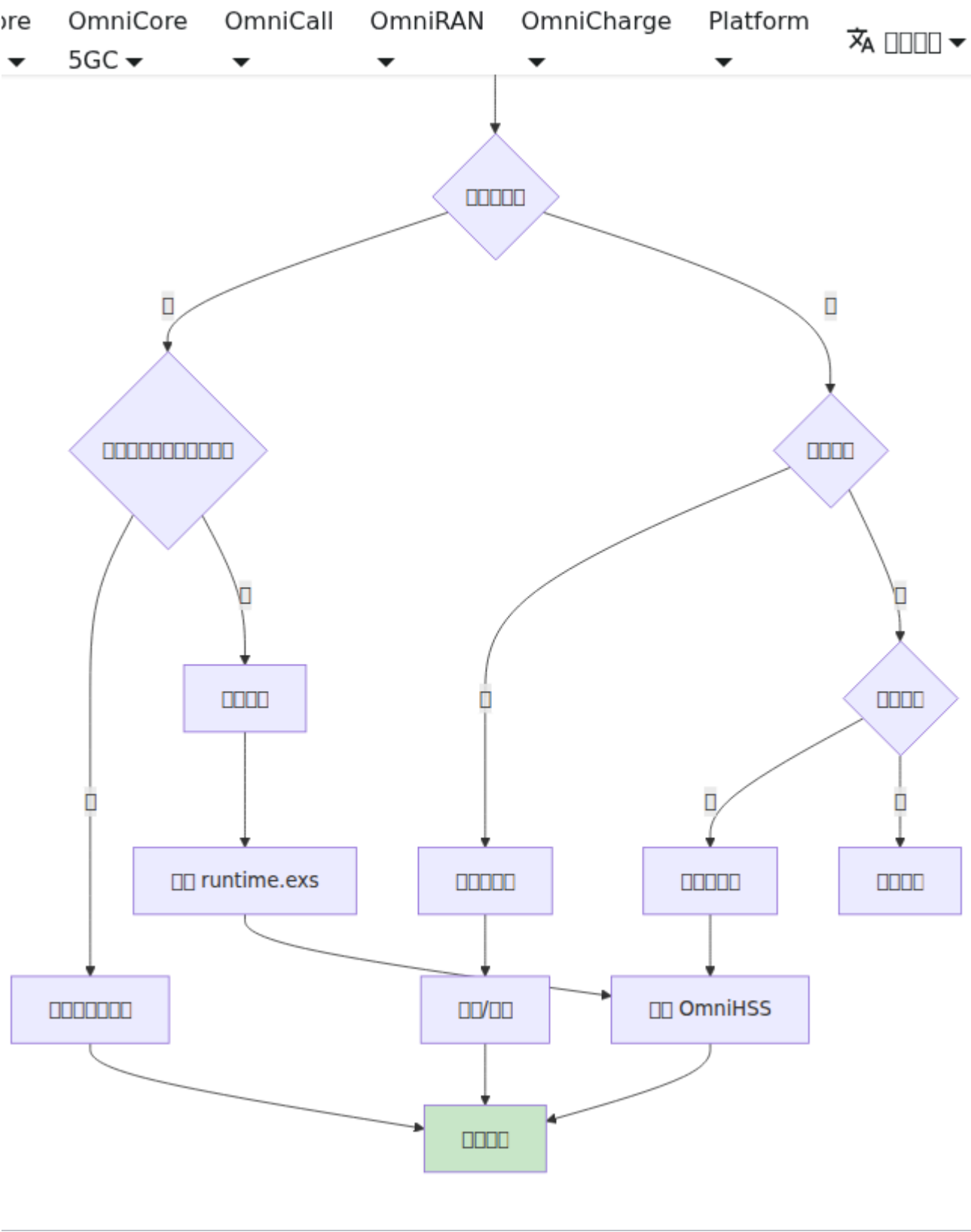
□□□□

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

□□□□

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

IMS 网络架构图



VoLTE 网络

网络

- IMS 网络
- 网络
- 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

网络 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
- OmniHSS Rx 設定
- IMS 設定

3 QoS/設定

設定

- 設定
- 設定
- 設定

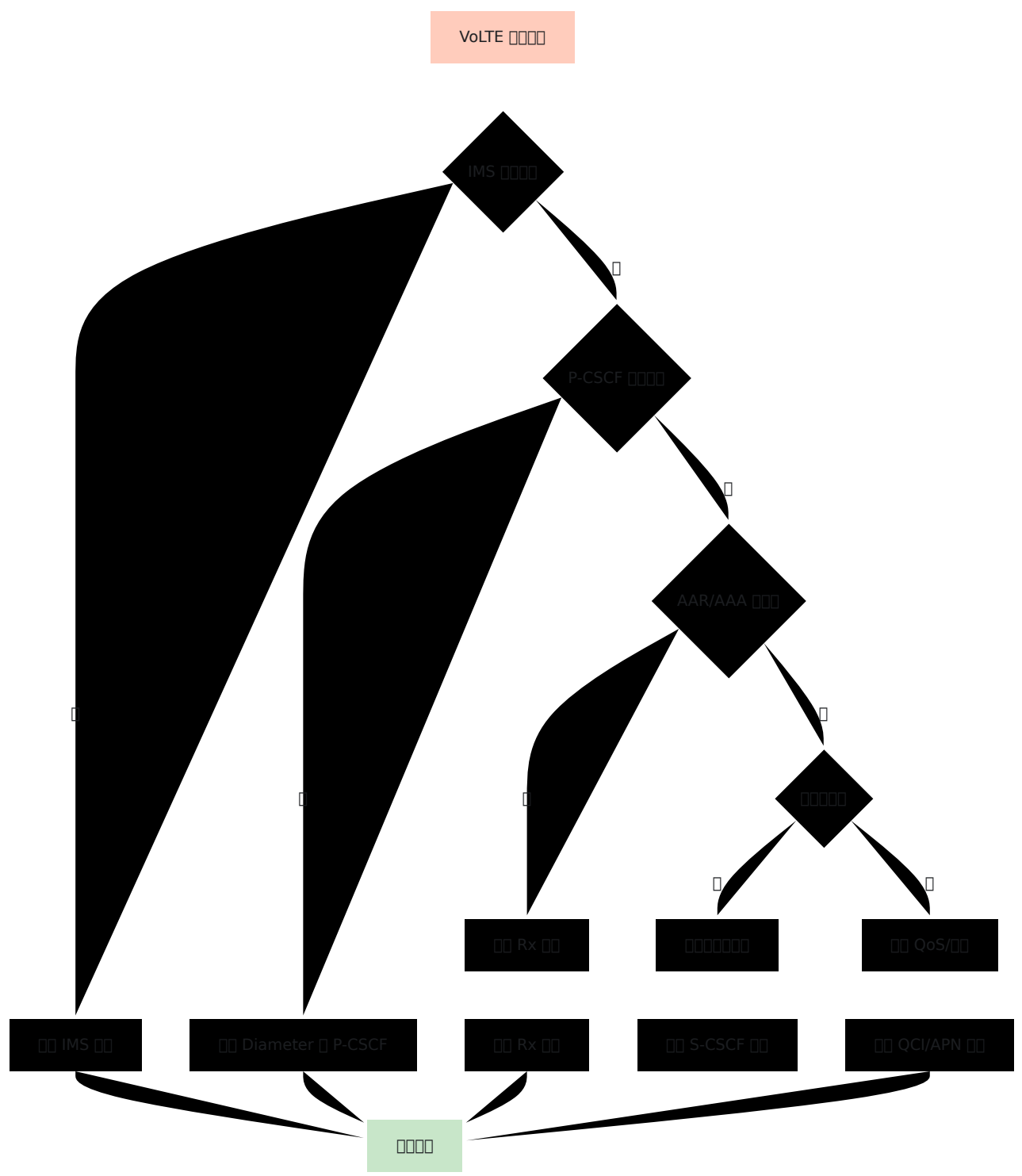
設定

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 网络架构图



認證

認證

- 認證過程
- 認證結果
- 認證失敗處理

認證過程

認證 1 認證過程

認證

- 認證過程
- 認證結果

認證過程

1. 認證過程 `roaming_profile_id`
2. 認證結果

認證過程

- 認證過程 認證結果

認證 2 認證過程

認證

- 認證過程
- 認證結果

認證過程

1. 認證過程 MME 認證過程 MCC/MNC
2. 認證結果
3. 認證 MCC/MNC 認證過程

4. 設定 roaming_rule

設定例

- curl コマンドで設定

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

3. IMS 設定

設定例

- 設定例
- IMS 設定
- 設定例

設定例

1. 設定例
2. data_action ims_action
3. 設定例

設定例

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

設定例

EIR 查詢

查詢

- 查詢號碼
- 查詢號碼
- EIR 查詢

查詢號碼

查詢 1IMEI 查詢號碼

查詢

- 查詢號碼/查詢
- 查詢號碼

查詢

- 查詢 EIR 查詢
- 查詢號碼
- 查詢號碼 IMEI
- 查詢號碼/查詢

查詢

- 查詢號碼 EIR 查詢
- 查詢號碼
- 查詢號碼

查詢 2MME 查詢 S13 查詢

查詢

- EIR 查詢
- 查詢號碼

查詢

1. 00 MME 00000000 S13 00
2. 00 MME Diameter 00000000
3. 00 S13 00000000
4. 00 MME 00

000000

- 00 MME 00 S13 00 EIR 00
- 00 Diameter 000000 S13 000000167772520
- 00000000 MME 0000

00 **3**00000000

0000

- 00000000000000000000

000000

1. 00000 EIR 00
2. 0000000000000000
3. 00 000000000000

000000

- 0000000000000000 0.* 000000 IMEI
- 00000000000000000000
- 00000000000000000000

000000

00

- API 00000
- Diameter 00000
- CPU 00000000

- 00000000

00000000

00 **1**0000000

000

- 000000
- 000 CPU 0
- 0000

00000

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

00000

- 00000
- 0000000
- 0000000
- 0000000
- 000 00000

00 **2**000000

000

- 0000000
- 000000000
- 0000000

00000

1. 0000000

2. 鋼管
3. 鋼管管口
4. 鋼管管口

鋼管

- 鋼管
- 鋼管管口
- 鋼管管口
- 鋼管管口

3 Diameter 鋼管

鋼管

- Diameter 鋼管
- 鋼管管口
- 鋼管管口

鋼管

1. 鋼管 Diameter 鋼管
2. 鋼管
3. 鋼管管口
4. 鋼管管口

鋼管

- 鋼管管口
- 鋼管管口
- 鋼管管口
- 鋼管管口 Diameter 鋼管

4 鋼管

鋼管

- OmniHSS 鋼管

- 安裝環境
- 安裝步驟

安裝環境

1. 安裝 OmniHSS 安裝環境
2. 安裝環境
3. 安裝環境
4. 安裝 Erlang VM 安裝

安裝步驟

- 安裝 OmniHSS 安裝環境
 - 安裝環境
 - 安裝 runtime.exs 安裝 Erlang VM 安裝
 - 安裝環境
-

安裝環境

安裝

- 安裝環境
- 安裝環境
- 安裝環境
- 安裝環境

安裝環境

安裝 1MME 安裝/安裝

安裝

- 安裝 MME 安裝環境
- 安裝 MME 安裝環境
- 安裝

□□□□□

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

□□□□□

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

□□ **2**□□□□□□□□

□□□

- □□□□□□□□□□□□□□
- PDN □□□□□□□□□□
- □□□□□□

□□□□□

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

□□□□□

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

□□ **3**□□□□□□□

□□□

- □□□□□□□□□□
- □□□□□
- □□□□□□

API 仕様

1. 概要
2. 接続方法
3. エラーコード

API 仕様

- 接続方法
- エラーコード
- 注意事項

API 仕様

概要

- API 仕様
- API 仕様
- 接続方法/エラーコード
- 500 以内

接続方法

API 1.0 仕様

概要

- 400 ~ 422 以内
- 接続方法
- エラーコード

API 仕様

1. 概要
2. API 仕様

3. 認証失敗時

4. 認証成功

認証失敗

- 認証失敗時 API 呼び出し
- 認証失敗原因
- 認証失敗時の ID 取得

第 2 回

認証

- 認証失敗
- 認証“key_set_id 取得”
- 認証成功

認証成功

1. 認証失敗
2. 認証成功時の処理
 - key_set_id → 取得
 - epc_profile_id → EPC 取得
 - ims_profile_id → IMS 取得

認証失敗

- 認証失敗原因
- 認証失敗時の ID
- 認証失敗時の API 呼び出し

第 3 回

認証

- 500 以内
- API 呼び出し
- 認証失敗

□□□□

- □□ □□□□

□□□□□□□□

□□□□□□□□

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
```

データベース

```
# 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]
```

検索

検索条件 **IMSI**

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

検索条件 **authentication.*fail**

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

検索条件 **Diameter**

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

検索条件 **database.*error**

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

□□□□

□□□□

□□□□□□□□□□/□□□□□□

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

□□□□□□

□□□□□□□□

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

□□□□□□

□□□□□□□□□□

- □□□□□□
- □□□□□□□□
- □□□□□□
- □□□□

□□□□□□□□□□□□□□□□

- 000000000000
- 0000000000
- 0000
- 0000

0000000000

0000

0000	00	0000
"0000000000"	0000000000	00 00000
"SQN 0000"	SQN000	000000
"000000"	000 IMSI	00 IMSI000000
"000000"	enabled=false	0000

Diameter 00

0000	00	0000
"Diameter 00000000"	0000	000000
"CER/CEA 0000"	000000	00 Diameter 00
"0000000000"	00000000000000	0000000000
"TLS 0000"	0000	0000

API 参数

参数名	类型	说明
"key_set_id"	字符串	唯一标识符
"key_set_id"	字符串	唯一标识符
"key_set_id"	字符串	唯一标识符
"key_set_id"	字符串	唯一标识符
"key_set_id"	字符串	唯一标识符

API 接口

参数名	类型	说明
"key_set_id 参数"	字符串	唯一标识符
"IMSI 参数"	字符串 IMSI	唯一标识符 IMSI 唯一标识符
"key_set_id"	字符串	唯一标识符

OmniHSS Webhook 簡介

← 目錄

簡介

- 簡介
 - Webhook 簡介
 - Webhook 類型
 - Webhook 格式
 - 配置
 - 驗證
 - 錯誤處理
 - 測試
-

簡介

OmniHSS 提供 **webhooks** 功能，用於接收來自 HSS 的即時通知。IMS 系統會透過 OmniHSS 的 webhook 端點，以 HTTP POST 方式發送通知。

Webhooks

Webhooks 是 HTTP 協議的一部分，OmniHSS 提供一個端點，用於接收來自 HSS API 的通知。

概要

- 概要 - Webhooks の概要
- 概要 - Webhook の 5 種類
- 概要 - 各 Webhook の概要
- 概要 - 各 Webhook の URL

Webhook の概要

Webhook

OmniHSS の Webhooks

EPC/LTE

Webhook	API	概要
update_location_request	S6a ULR	位置更新要求
authentication_information_request	S6a AIR	認証情報要求
purge_request	S6a PUR	MME のデータ削除要求
cancel_location_answer	S6a CLA	MME の位置更新応答

IMS

API	Interface	Function
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)

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

IMSI

API	Interface	Function
imsi_switch	ULR for different IMSI on same SIM	IMSI SIM switch

Webhook

Webhook

OmniHSS webhook URL 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"
  }
}
```

Schema

Field	Type	Description
event	string	Event name, e.g. update_location_request
event_id	string	Unique identifier for the event, e.g. webhook UUID
timestamp	string	Timestamp in ISO 8601 format
subscriber	object	Subscriber information, e.g. GET /api/subscriber/:id
event_context	object	Contextual information for the event

Example

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 消息

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

Webhooks

Webhooks

Webhooks 是 OmniHSS API 的一部分

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. □□□□□□□□

□□□□□□□□□□□□□□□□□□□□


```
// 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
    });
  }

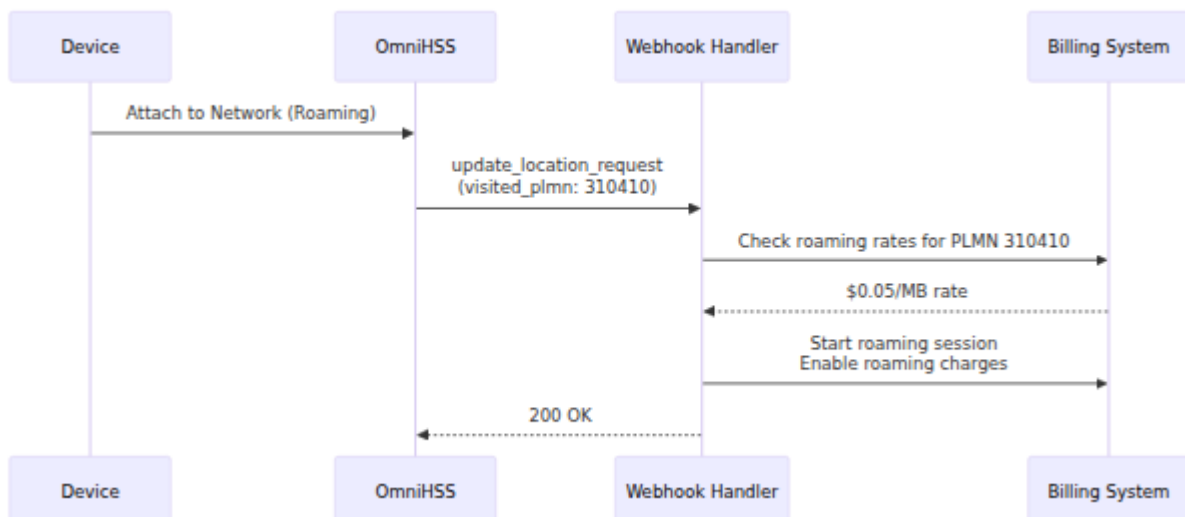
  res.status(200).send();
});
```

Sequence Diagram

- MME
- HSS
- IMS

3. Sequence Diagram

Sequence Diagram



□□□□□□

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 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 是 3GPP 定义的 VoLTE 和 IMS

```

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

  if (event === 'ims_registration' && !subscriber.ims_enabled) {
    // IMS ON - Enable IMS
    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']
    });
  }

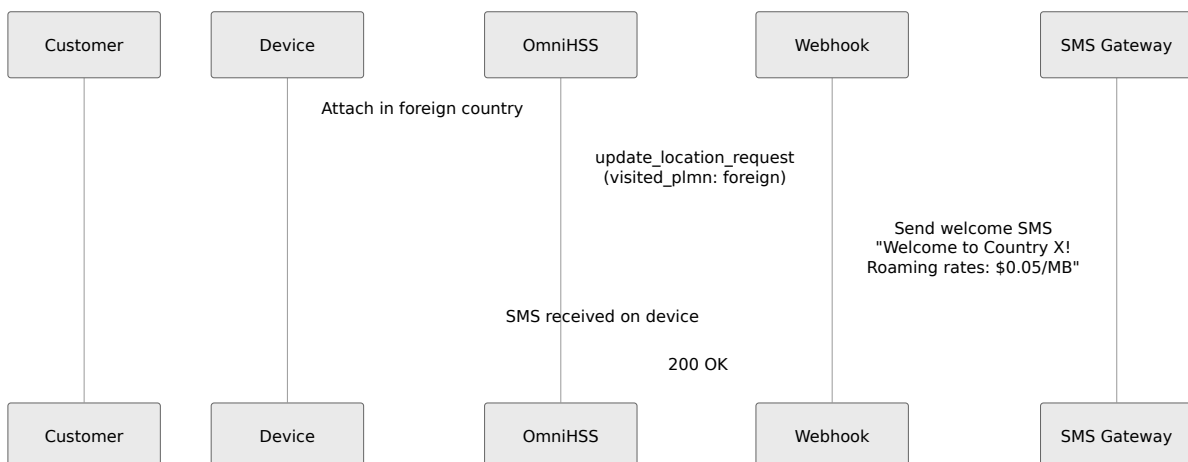
  res.status(200).send();
});

```

5. 测试

测试环境搭建

测试数据准备



测试用例

- "IMEI [IMEI] is not valid"
- "IMEI 80% is not valid"
- "IMEI is not valid VoLTE IMEI"
- "IMEI is not valid"

6. IMSI SIM

IMSI SIM is a type of SIM card that can be used to switch between different IMSI numbers.

```
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;

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

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

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

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

7. 設定項目

OmniHSS 設定項目

設定項目

- CRM 項目 - 設定項目
- 項目 - 設定項目
- 項目 - 設定項目
- 項目 - 設定項目
- 項目 - 設定項目

設定項目

Webhook 項目/項目

項目 webhooks 項目 OmniHSS項目項目

```
# 項目項目 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 項目項目 X-OmniHSS-Signature 項目

X-OmniHSS-Signature:
sha256=5d7a8f9b2c1e3a4d6f7e8b9c0a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a

項目項目


```
const crypto = require('crypto');

function verifyWebhook(req) {
  const signature = req.headers['x-omnihss-signature'];
  const secret = process.env.WEBHOOK_SECRET;
  const payload = JSON.stringify(req.body);

  const expectedSignature = 'sha256=' +
    crypto.createHmac('sha256', secret)
      .update(payload)
      .digest('hex');

  return crypto.timingSafeEqual(
    Buffer.from(signature),
    Buffer.from(expectedSignature)
  );
}

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

  // OK webhook...
  res.status(200).send();
});
```

□ □ □ □

1. **HTTPS** - webhook TLS
2. - webhook
3. - webhook
4. **IP** - webhook OmniHSS IP
5. - webhook
6. - webhook
7. - webhook

Webhook

Webhook 支持以下事件

- IMSI 注册/注销
- MSISDN 注册/注销
- 网络注册/注销 PLMN/MME
- 网络切换

Webhook 支持以下配置

- **GDPR** - 通过 webhook 接收 GDPR 数据
- 网络 - 接收网络注册/注销事件
- 网络 - 通过 webhook 接收网络切换事件
- 网络 - 通过 TLS 接收 webhook 数据
- 网络 - 接收 webhook 配置更新事件

配置

Webhook 配置

配置

- 通过 webhook 接收数据
- Webhook 配置更新事件

配置示例

1. 通过 webhook 接收数据

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

2. 通过 webhook 接收数据

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

3. HSS を

- webhook を
-
- DNS を

4. 設定

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

Webhook

- HSS の webhook を
- Webhook を HSS を

1. 設定

- 5 の HTTP 200
-

2. 設定

```
// 10 - 10초 지연
app.post('/webhook', (req, res) => {
  processData(req.body); // 10초 지연
  res.status(200).send();
});

// 2 - 2초 지연
app.post('/webhook', (req, res) => {
  res.status(200).send(); // 2초 지연
  processData(req.body); // 2초 지연
});
```

Webhooks

1. 개요

- 이벤트 발생 시 호출되는 API
- `event_id` 이벤트 식별자

2. 특징

- OmniHSS 시스템에서 발생하는 이벤트
- 이벤트 발생 시 webhook 호출

3. 사용법

1. `event_id` 이벤트 식별자

```

const processedEvents = new Set();

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

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

  processedEvents.add(eventId);

  // 処理 webhook...
  processWebhook(req.body);

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

```

Webhook 問題

問題

- 発生 HTTP 4xx 或 5xx
- HSS 側 webhook 失敗

原因

1. 401 Unauthorized - 認証失敗

- 側 webhook 署名失敗
- 署名失敗

2. 400 Bad Request - 不正なリクエスト

- 側 webhook 署名失敗
- 署名失敗 Content-Type 不正

3. 500 Internal Server Error - サーバエラー

- 署名失敗

- 認證與驗證

認證

驗證

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

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

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

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

  } catch (error) {
    console.error('Webhook processing error:', error);
    // 200 狀態碼與訊息
    res.status(200).json({ status: 'error', message: error.message });
  }
});
```

認證與驗證

認證

- 呼叫 webhook 認證與驗證
- 認證 null 值

驗證

1. 認證與驗證 - 認證與驗證IMS認證

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 统计
rate(omnihss_webhook_success_total[5m]) /
rate(omnihss_webhook_attempts_total[5m])

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

Webhook 配置

配置 webhook 接收来自设备的数据

配置示例

```
{
  "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
}
```