

API 入門

← API 入門

概要

- API 概要
- API 仕様

API 仕様

400 エラー

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

原因

- JSON 形式エラー
- パラメータエラー
- 認証エラー

404 エラー

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

原因

- 0000/00000/000000
- URL 00 ID 000

422 00000000

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

000

- 0000
- 000000000
- 00000000

500 00000000

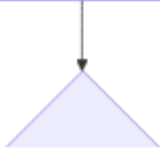
```
{
  "error": "Internal server error"
}
```

000

- 00000000
 - 0000000000
-



API Request



OmniCharge

OmniRAN

Downloads

🔍

[Omnitouch Website](#)

Invalid JSON

400 Bad Request

Valid

Authorized?

No

401 Unauthorized

Yes

Resource Exists?

No

404 Not Found

Yes

Data Valid?

No

422 Validation Error

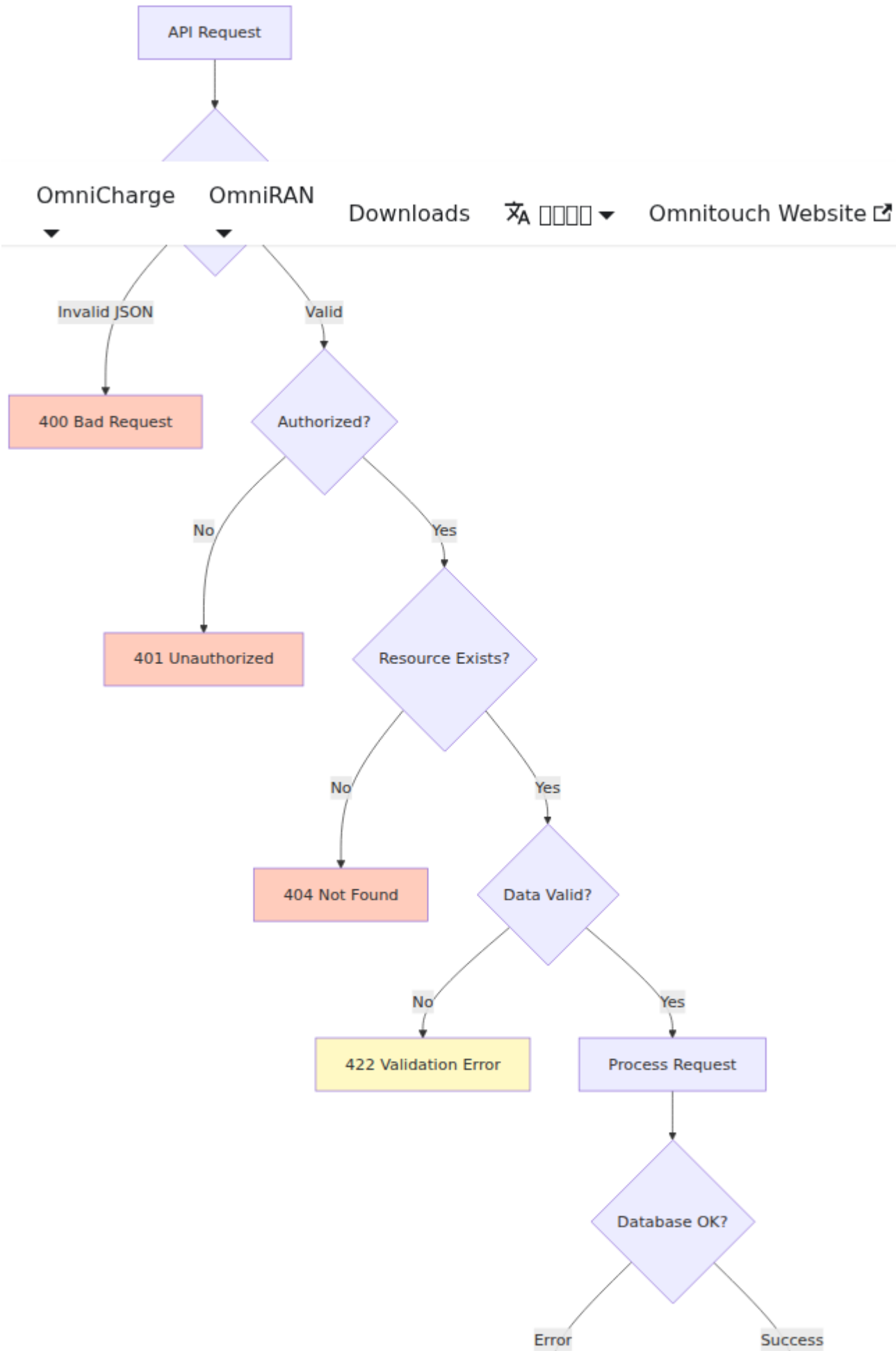
Yes

Process Request

Database OK?

Error

Success





← API

API 開箱

← API 開箱

簡介

- 開箱 API
- 開箱 IP 地址

API 介紹

API 是應用程式介面，用於在不同系統之間進行資料交換。

API 的安裝方法如下：
Linux: `apt-get install jq` 或 `brew install jq`
Mac: `brew install jq`

API 的用途如下：

- 獲取資料
- API 地址
- EPC 地址
- 查詢

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": "인터넷 QoS",
  "allocation_retention_priority": 8,
  "apn_ambr_dl_kbps": 50000,
  "apn_ambr_ul_kbps": 25000,
  "pre_emption_capability": true,
  "pre_emption_vulnerability": true,
  "qci": 9
}' | jq -r '.response.id')
```

3. APN 식별자 생성

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

4. APN 프로파일 생성

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

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

5. 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\": \"\",
  \"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.

```
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 "ID: $SUBSCRIBER_ID"
```

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

- MSISDN
- IMS
-

- 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

専用 IP

専用 IP あり

専用 IP あり 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 (IP 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 (IP) - 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

- **IP IP** 物联网 - 物联网 IP 物联网
- 物联网物联网 - 物联网 IP 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: HTTPS
- ポート: 8443

API: API 形式はJSON形式

リクエスト:

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

レスポンス:

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

コマンド:

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

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

リクエスト

レスポンス JSON形式

リクエスト:

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

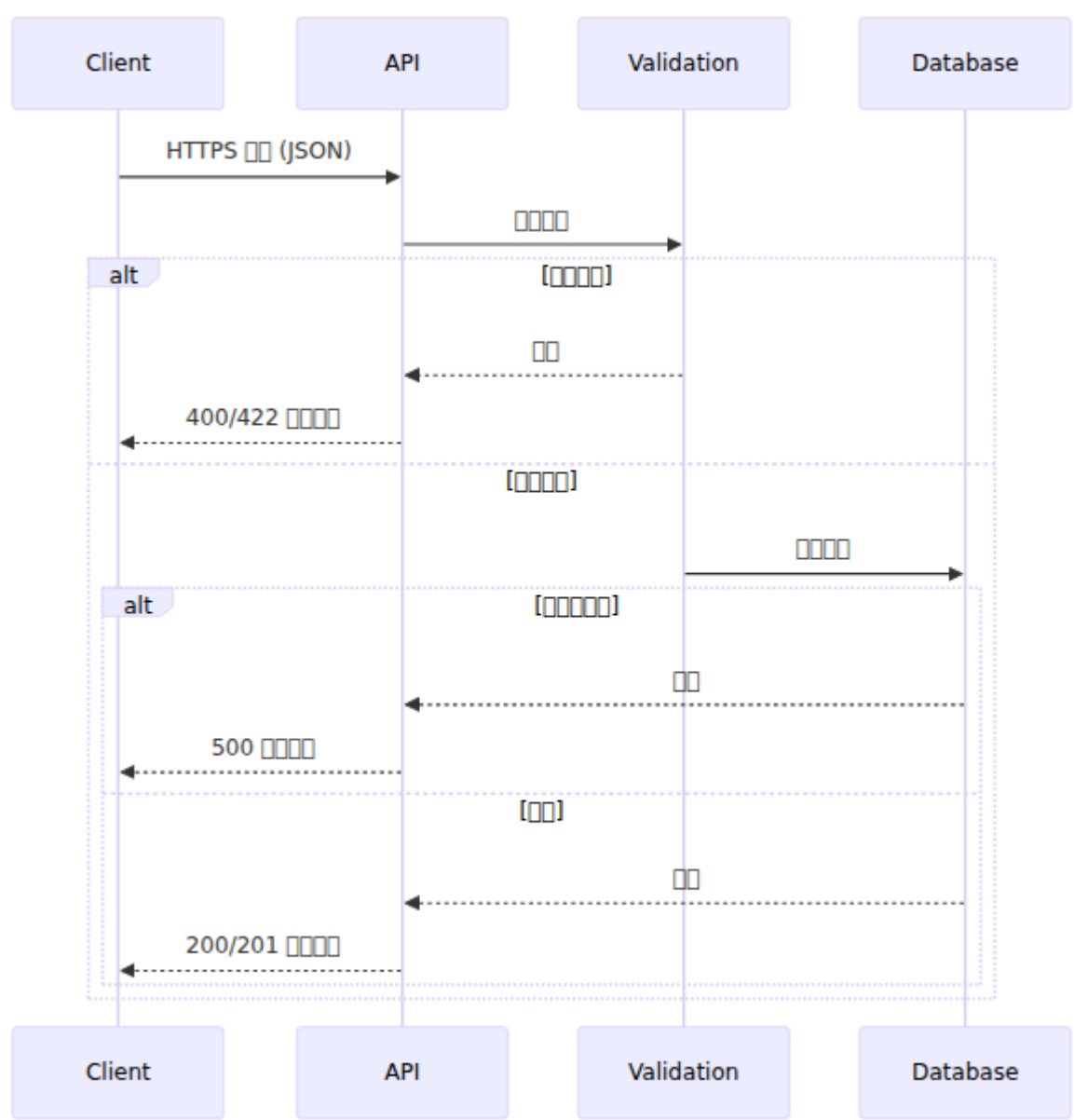
예외:

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

HTTP 코드

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

API 流程图



请求

响应

数据库交互

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:

項目	型	説明
id	integer	サブスクリプション ID

リクエスト:

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

IMSI 取得

IMSI 取得

リクエスト: GET /api/subscriber/imsi/:imsi

レスポンス:

項目	型	説明	長さ
imsi	string	IMSI	14-15 桁

リクエスト:

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

レスポンス: IMSI 001001123456789

MSISDN 取得

MSISDN 取得

リクエスト: GET /api/subscriber/msisdn/:msisdn

レスポンス:

項目	型別	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: 成功した場合はJSON形式で返ります

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 APN 000 00 IP ID 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:



□□ : □□□□□□



00: 000 EPC 0000

[illegible]

□□: IMSI □□□

□□□□

□□□□□□□

201 〇〇〇

□□□□

□□□□□□□

□□: PUT /api/subscriber/:id

□□□□:

□□	□□	□□
id	integer	□□□□□ ID

□□□:

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

□□□□□:

- enabled - □□/□□□□□□
- ims_enabled - □□/□□ IMS □□
- sim_id - □□ SIM □ □□
- key_set_id - □□ □□□□□□□□
- epc_profile_id - □◆◆◆ □□□□□□□□
- ims_profile_id - □□ □□□□□□□□
- roaming_profile_id - □□ □□□□
- msisdns - □□□□□□□□ □□□□
- static_ips - □□□□□ APN □ □□ IP

- `custom_attributes` - 自定义属性

响应:

- `imsi` - IMSI 国际移动用户识别码

错误:

- `400` - 请求错误

请求:

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

响应:

- 响应体: `{"enabled": false}`
- 响应头: `{"ims_enabled": false}`
- 响应体: `{"epc_profile_id": 2}` (EPC 配置)
- 响应体: `{"roaming_profile_id": 3}` (漫游配置)

删除

删除接口

请求: `DELETE /api/subscriber/:id`

响应:

参数	类型	说明
<code>id</code>	integer	用户 ID

요청:

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

요청: PDN 연결을 삭제하고 IMSI를 반환합니다.

요청: 요청 본문:

- **IMSI** - IMSI 번호
- **SIM** - SIM 카드 번호
- **MSISDN** - MSISDN 번호
- **MSISDNs** - MSISDN 번호 목록

응답:

응답: CLR 응답 본문은 MME에 반환됩니다.

요청: POST /api/subscriber/cancel_location

요청:

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

요청:

필드	타입	필수	설명
imsi	string	예	IMSI 번호 14-15 자리

응답:

```
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"
}
```

注意:

- 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 MSISDN へ通知

失敗:

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

- 00: 000000000000

0 IMSI 00:

00 CLR 000 MSISDN 000:

1. 00 MSISDN000 IMSI:

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

// 00: 0000 CLR000 MSISDN 0000000000
```

2. 00 IMSIs00000000:

```
// 0000000000 MSISDN 0000 IMSIs0000000000
// 00 A: IMSI 0010011111111111, MSISDN "+1234567890"
// 00 B: IMSI 0010012222222222, MSISDN "+1234567890"

POST /api/subscriber/cancel_location
{"imsi": "0010011111111111"}

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

0000:

- 00 **IMSI**: CLR 000 IMSI 00000000 MSISDN
- 00: CLR 0000000000000000 CLR 00000000 MME 0000
- 000 **MME** 00: 00 MME 0000000000 CLR000 HSS 000
- 00: 000 IMSI 0000000000

0000:

- 000000000000
- 0 IMSI 00
- S6a 0000

MSISDN

MSISDN [MSISDN](#)

MSISDN

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

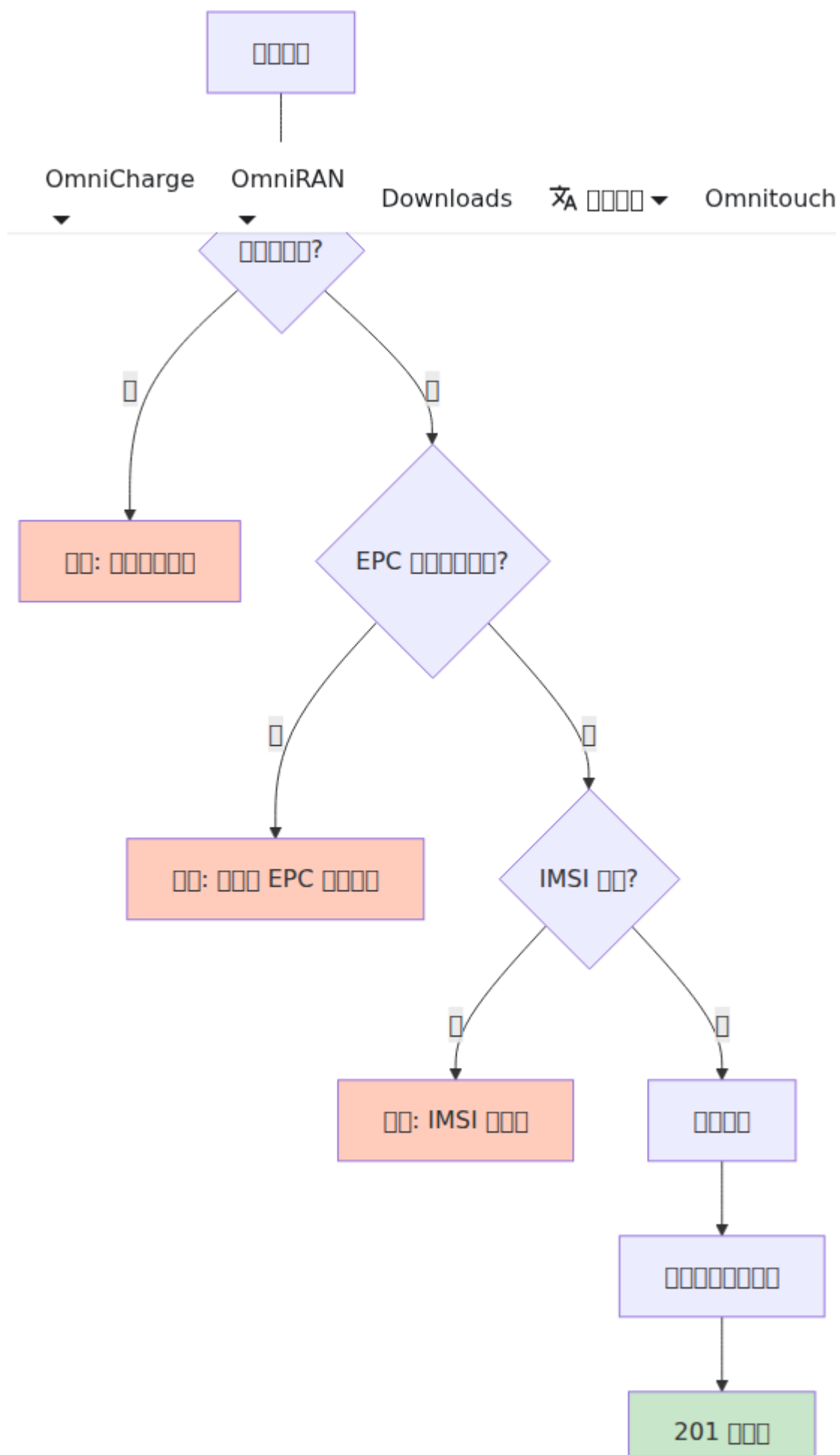
示例:

```
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:



刪除 **MSISDN** 及 **IMSI** 記錄

刪除 MSISDN

刪除記錄

請求: `DELETE /api/msisdn/:id`

響應:

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

SIM 記錄

SIM 記錄包含 SIM 記錄的 ICCID 及 PIN/PUK 值。OTA 更新 SIM 記錄。

請求:

- IMSI** - 刪除 SIM 記錄

獲取 SIM

獲取 SIM 記錄

請求: `GET /api/sim`

響應:

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

SIM

獲取 SIM 記錄

Request: GET /api/sim/:id

Response:

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

API SIM

API endpoint SIM

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 string

Response:

- sim_vendor - string
- batch_name - string
- is_esim - eSIM flag

- `pin1`, `pin2` - PIN 1000
- `puk1`, `puk2` - PIN 1000
- `adm1-adm10` - 1000
- `kic`, `kid` - OTA 1000000000000000

Request:

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

POST SIM

POST SIM 1000

Request: `PUT /api/sim/:id`

Request:

```
curl -k -X PUT https://hss.example.com:8443/api/sim/1 \
-H "Content-Type: application/json" \
-d '{
  "sim": {
    "batch_name": "10000000"
  }
}'
```

POST SIM

POST SIM 1000

Request: `DELETE /api/sim/:id`

00: 0000000000000000 SIM

00000

00000000 Milenage 0000000000000000KiOPC/OPAMFSQLN0000 00 00000000
00

00:

- 0000 - 000000000000

00000

0000000000

00: GET /api/key_set

0000:

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

00000

00000000

00: GET /api/key_set/:id

0000:

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

0000:

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

□□□□□

□□□□□□□□□□□□

□□: POST /api/key_set

□□□:

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

□□□□:

- `ki` - 128 □□□□32 □□□□□□□□
- `opc` □ `op`□□OPC □□□ OP □□□
- `authentication_algorithm` - □□□□□ "milenage"

□□□□:

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

요청:

- 요청 바이트 크기
- KiOPCOP: 32 바이트 크기 128 비트
- AMF: 4 바이트 크기 16 비트

요청:

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

요청: 요청 바이트 크기 요청 바이트 크기 API 요청

요청

요청 바이트 크기

요청: PUT /api/key_set/:id

요청: 요청 바이트 크기 요청 바이트 크기 요청 바이트 크기

요청: 요청 바이트 크기 요청 바이트 크기 요청 바이트 크기

요청

요청 바이트 크기

요청: DELETE /api/key_set/:id

例: 以下は、EPC のデフォルト値を示す。

例: 以下は、EPC のデフォルト値を示す。

EPC のデフォルト値

EPC のデフォルト値は、以下に示すように設定されている。

例: EPC のデフォルト値

例: GET /api/epc/profile

例: EPC のデフォルト値

例: GET /api/epc/profile/:id

例: EPC のデフォルト値

例: POST /api/epc/profile

例:

```
{
  "apn_profiles": [],
  "name": "デフォルト",
  "network_access_mode": "デフォルト",
  "tracking_area_update_interval_seconds": 600,
  "ue_ambr_dl_kbps": 100000,
  "ue_ambr_ul_kbps": 50000
}
```

例:

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 프로파일 ID 문자	문자	[] 또는 [1, 2, 3]

예시:

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
-H "Content-Type: application/json" \
-d '{
  "apn_profiles": [],
  "name": "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

PUT EPC profile

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

Response: EPC profile is updated successfully

DELETE EPC profile

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

Response: EPC profile is deleted successfully

IMS profile

IMS IP address is used to identify the IMS network. IFC is used to identify the IMS network. IMS profile is used to identify the IMS network.

GET IMS profile

Request: GET /api/ims/profile

POST IMS profile

Request: POST /api/ims/profile

Response:

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

Response:

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

IFC profile:

IFC profile is used to identify the IMS network.

変数	変数	値
{{ imsi }}	IMSI	001001123456789
{{ msisdns }}	MSISDN 番号	["14155551234", "14155555678"]
{{ mcc }}	国番号	001
{{ mnc }}	ネットワーク番号	001

例:

IFC として **Liquid** を使用して Jinja2 を使って IMS への XML を生成

1. 変数: IMS 番号と MSISDN 番号を指定する `{{ imsi }}` と `{% for msisdns in msisdns %}`
2. 変数: API 番号と XML 番号
3. 変数: IMS 番号 MAA/SAA/HSS
 - IMS 番号
 - MSISDN 番号
 - `{{ imsi }}` → IMSI
 - `{{ msisdns }}` → MSISDN
 - `{{ mcc }}` → 国番号
 - `{{ mnc }}` → ネットワーク番号
 - Cx/Diameter 番号 XML 番号 S-CSCF

例:

```
<!-- 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
- name は一意である

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 **PCRF** API **QoS** 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 📡📡

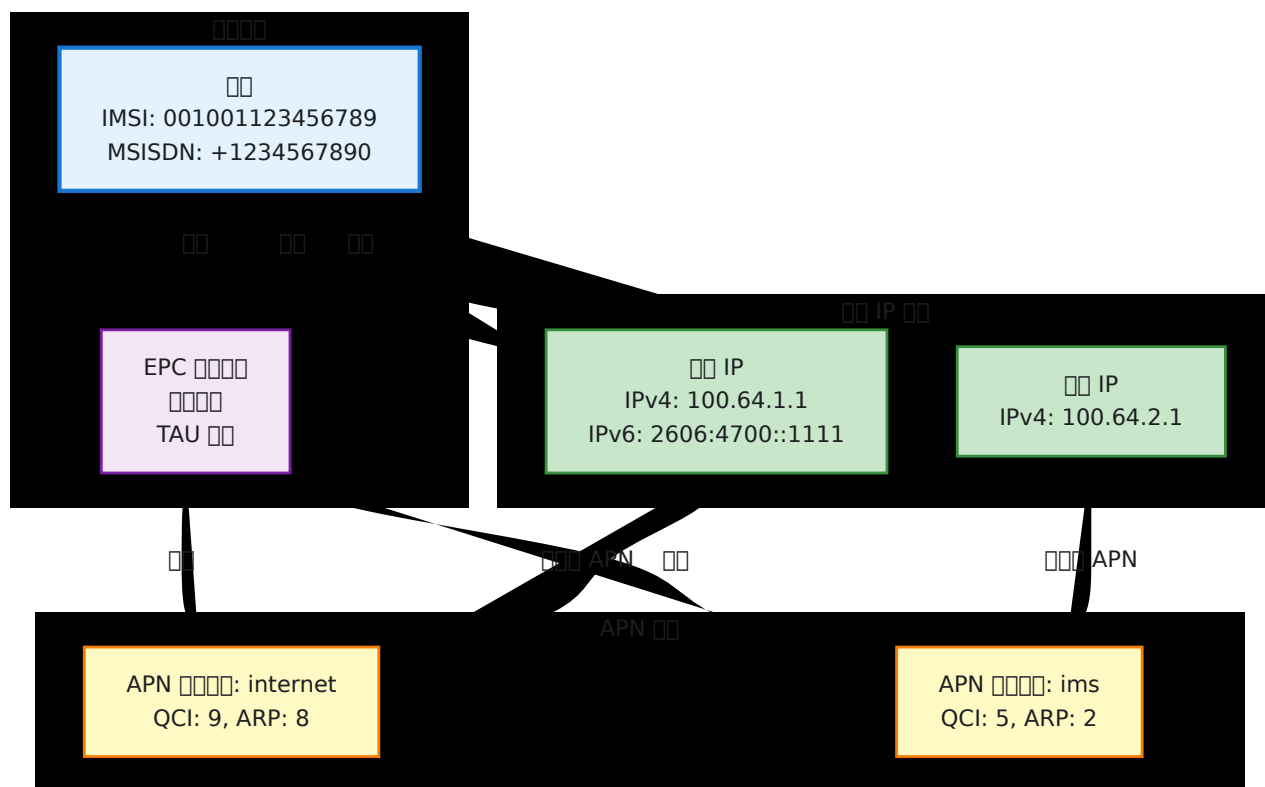
11

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

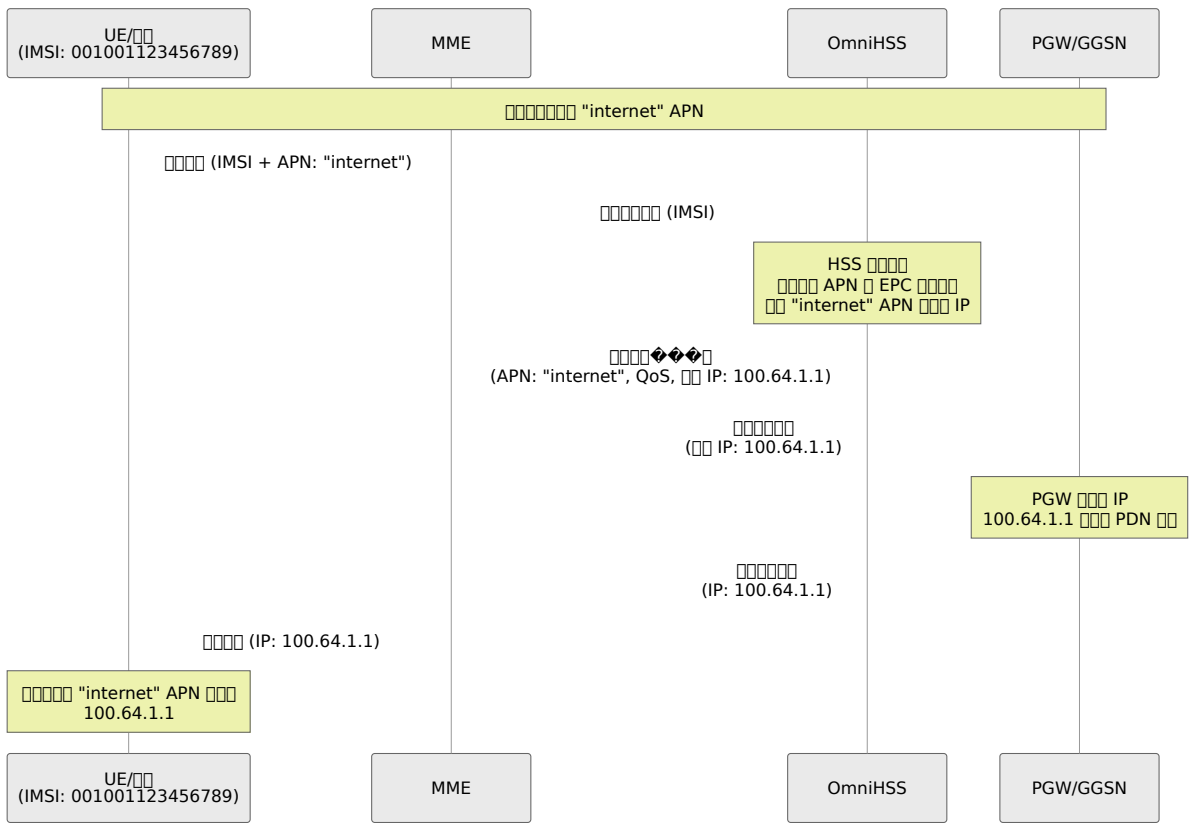
IP

IP アドレス APN APN IP4 / IPv6 DHCP
設定

11

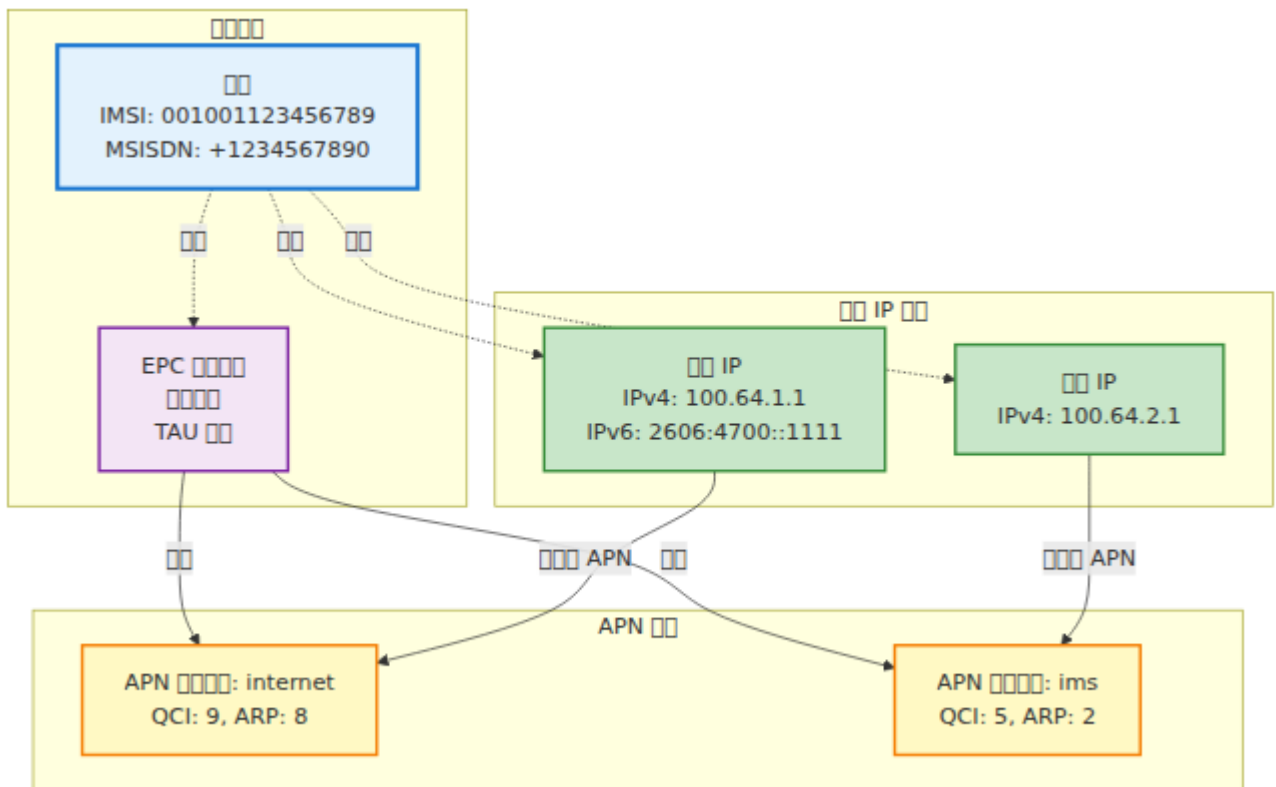


□ □ □ □ □ □ □ □ :



网络 - APN 网络:

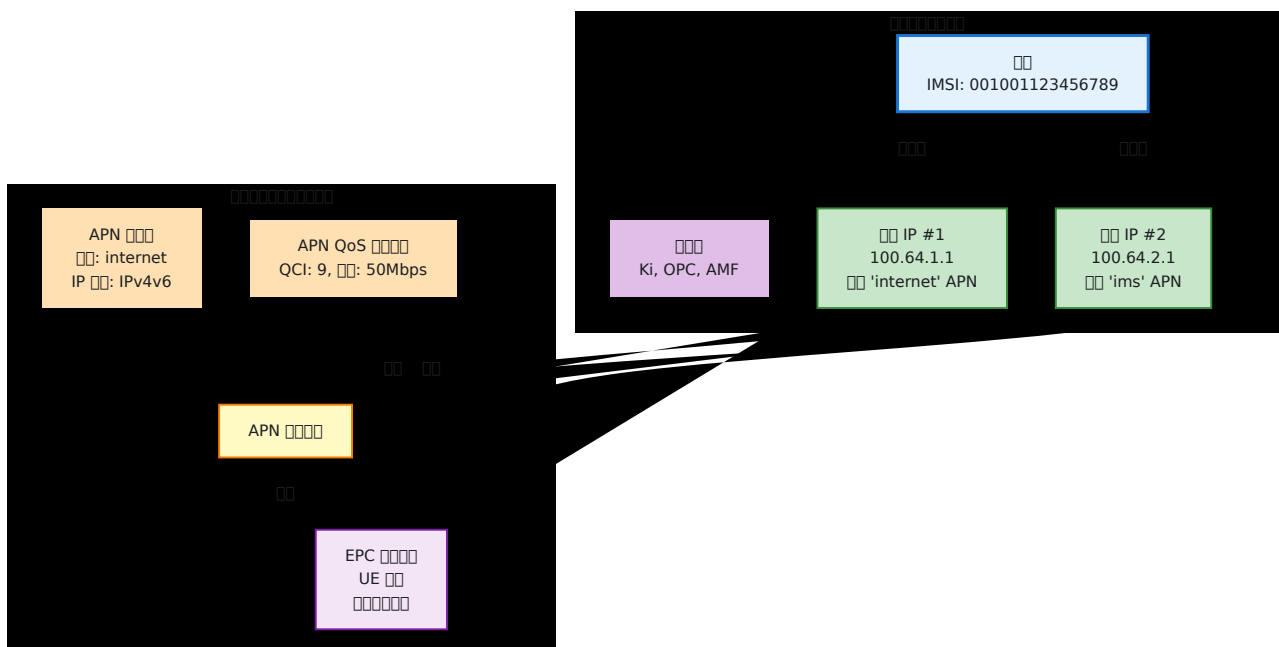
网络 S6a 网络 APN-网络 AVP 网络



网络:

1. **APN identifier**: APN identifier 0, 1, 2...
2. **APN type**: APN type `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
7. **VPLMN**: 0 - IP

Example:



Example:

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

□□□□:

- IoT □□□□ IP □□
- □□□□□□□□□□□□□□ IP □□□□□□□□
- □□□□ IP □□□□□□□□
- □□□ IP □□□□□□□□
- □□ IP □□□□□□□□

□□□□ IP

□□□□□□ IP □□□

□□: GET /api/epc/static_ip

□□□□:

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

□□□□:

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

🌐 IP

🌐 IP 📄

🔍: GET /api/epc/static_ip/:id

📄:

🔍	📄	📄
id	integer	🌐 IP 📄 ID

📄:

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

靜態 IP

將 APN 與靜態 IP 綁定

請求: `POST /api/epc/static_ip`

請求:

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

響應:

- `apn_profile_id` - 綁定靜態 IP 的 APN 配置 ID
- 綁定靜態 IP 的 `ipv4_static_ip` 與 `ipv6_static_ip`

響應:

- `ipv4_static_ip` - IPv4 靜態 IP 地址
- `ipv6_static_ip` - IPv6 靜態 IP 地址

IP 地址:

- IPv4: 綁定靜態 IP 地址 `100.64.1.1`
- IPv6: 綁定靜態 IP 地址 `2606:4700:4700::1111`
- IPv4 與 IPv6 靜態 IP 地址綁定 IP 地址池
 - 綁定靜態 IP 地址池
 - 綁定 IP 地址池與靜態 IP 地址池 APN
 - 綁定靜態 IP 地址池與靜態 IP 地址池

響應:

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>
両方	✓	✓	両方指定可能

例:

IPv4 のみ 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 のみ 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"
  }
}'
```

両方 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:

項目	型別	説明
id	integer	静的 IP 設定 ID

例:

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

フィールド:

- ipv4_static_ip - IPv4 静的 IP
- ipv6_static_ip - IPv6 静的 IP
- apn_profile_id - APN プロファイル ID

レスポンス:

- id - 静的 IP 設定 ID

注: 静的 IP 設定は、PDN 設定と IP アドレス設定を必要とします。

例:

```
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"
  }
}'
```

静的 IP

静的 IP 設定

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

Response:

Field	Type	Description
id	integer	Static IP ID

Example:

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

Notes:

- Static IP ID
- APN (Access Point Name) is required
- Static IP address must be unique
- Static IP must be assigned to a PDN

Response: Static IP ID, PDN, and Static IP address are returned.

Static IP Management

Static IP address must be unique and assigned to a PDN.

Example:

- Static IP address is unique
- Static IP address is assigned to a PDN

Static IP address:

```
# 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 APN:

IP IP APN

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

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

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

[]:

- ✓ []: [] APN [] IP
- ✗ []: [] APN [] IP

[] - [] **APN:**

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

# 响应:
{
  "errors": {
    "static_ips": [
      "静态 IP APN 中 IP 地址池 IP 100.64.1.1 已
internet 100.64.1.2 已 internet"
    ]
  }
}
```

错误:

- 静态 IP - 地址池
- 静态 IP - 地址池
- 静态 IP 地址池 - 地址池

漫游

漫游功能允许设备在 IMS 网络中漫游 MCC/MNC 网络

漫游配置

GET /api/roaming/profile

漫游配置

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 网络 漫游规则
- ims_action_if_no_rules_match - IMS 漫游规则

请求

请求: GET /api/roaming/rule

请求

请求: POST /api/roaming/rule

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

필드:

- `imei` - 15자리 숫자
- `imei` - Diameter 프로토콜

EIR

OmniHSS는 S13 Diameter 프로토콜을 사용하여 EIR에서 IMEI 정보를 조회합니다.

이 API는 EIR에서 S13 프로토콜을 사용하여 IMEI 정보를 조회합니다.

EIR

GET /api/eir/rule

EIR

POST /api/eir/rule

配置:

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

参数:

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

动作:

- 0 - 忽略
- 1 - 警告
- 2 - 阻止

备注:

- IMEI 由 15 位数字组成
- 前 3 位为 TAC (Type Allocation Code)
- 后 12 位为 SNR (Serial Number)

来源:

- 来源 1 - S13 设备 EIR 数据
- 来源 2 - OmniHSS EIR 数据

附录

设备识别与追踪

- **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

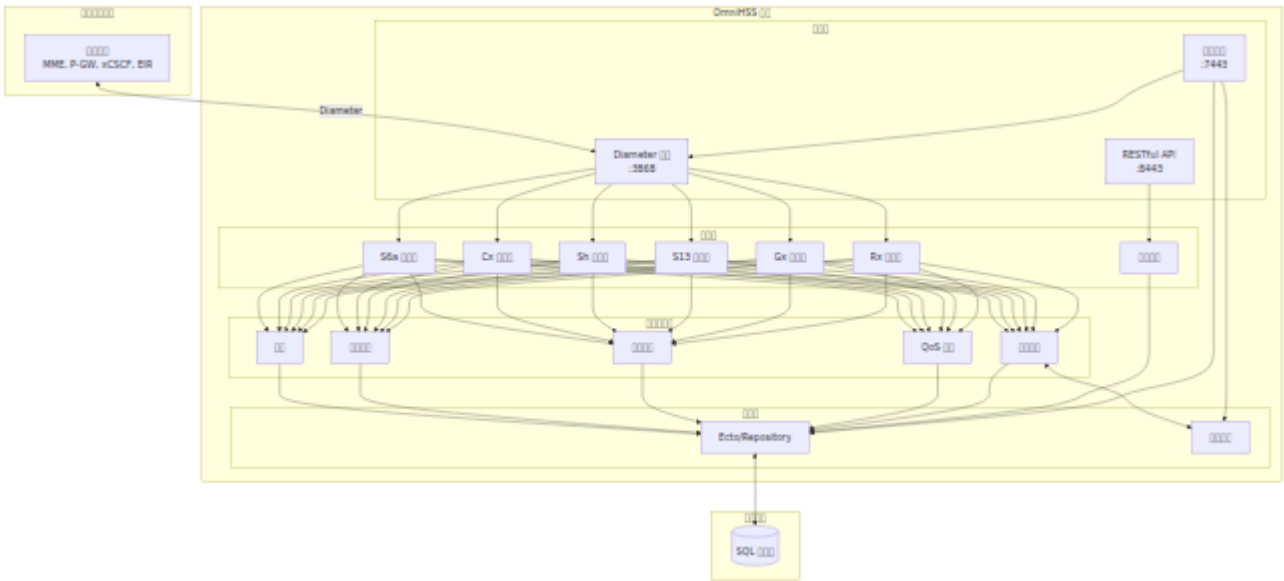
← API 入門

OmniHSS

←

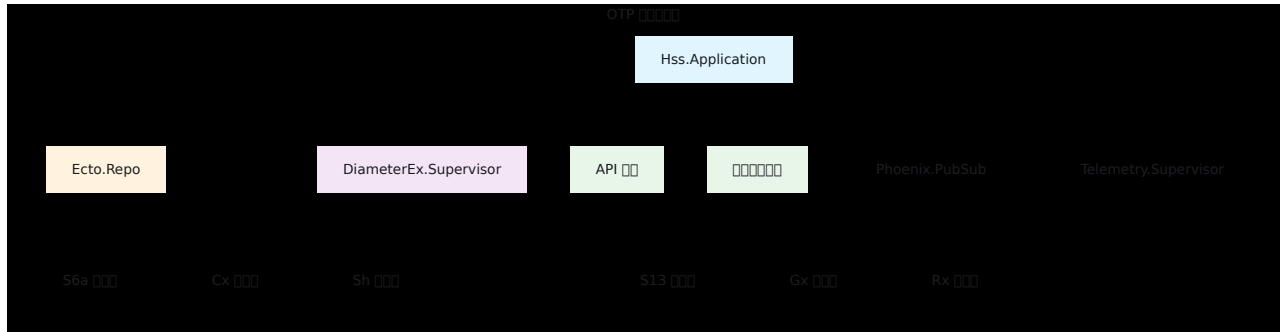
-
-
- Diameter
-
-
-
-

OmniHSS is an Elixir / Erlang/OTP based implementation of the HSS (Home Subscriber System) interface.



□□□□

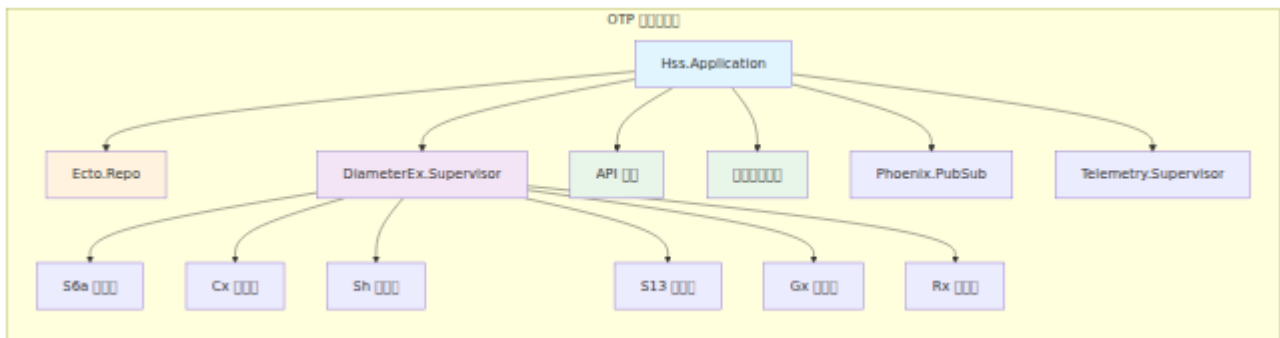
□□□□



Diameter □□□□

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

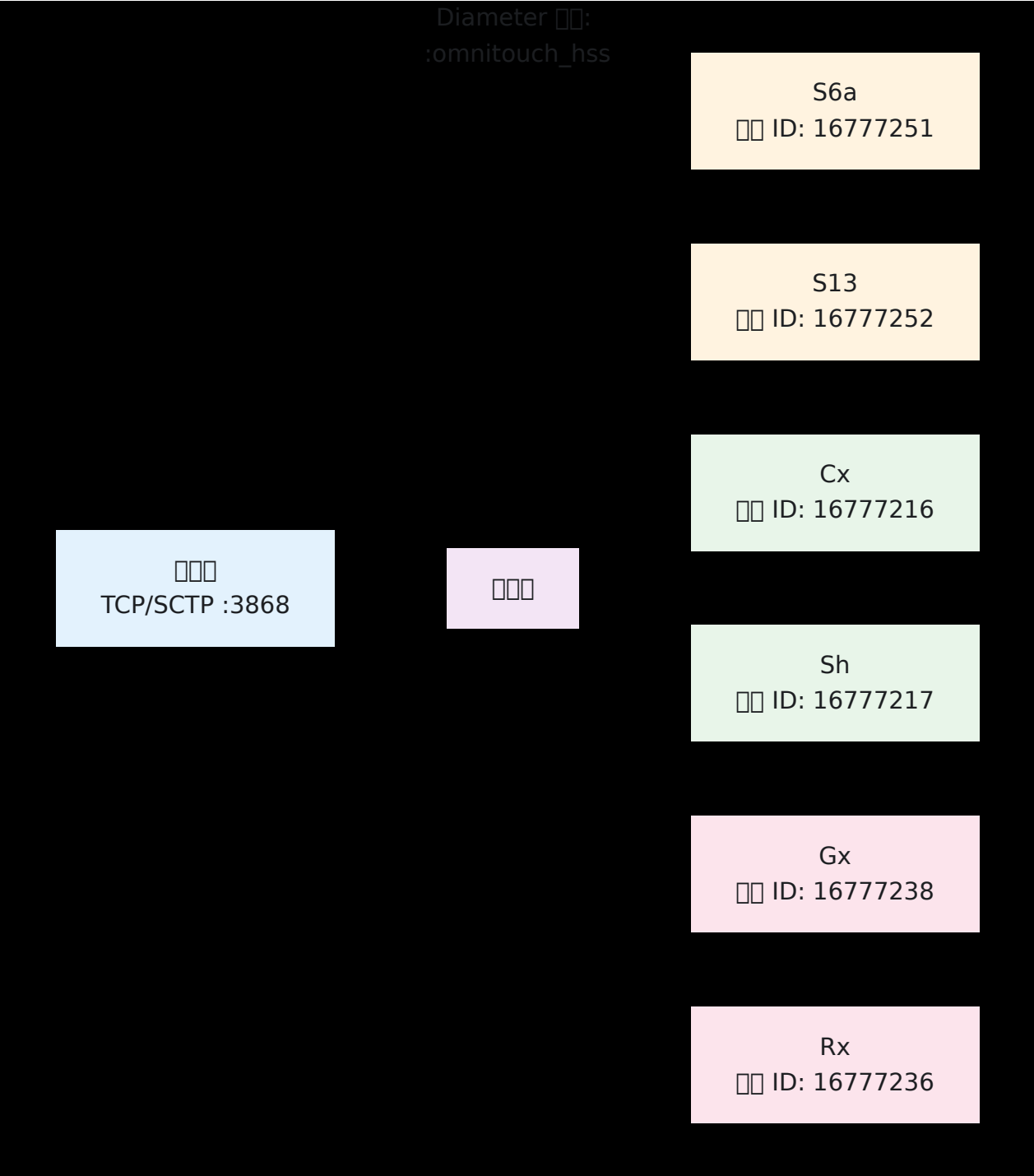
1. □ **DiameterEx** □□ - □□□□□ Diameter □□ ID
2. □□□□ - □□ AVP□□□□□□□□□□
3. □□□□□□ - □□□□□□□□□□□□
4. □□□□ - □□ AVP □□ Diameter □□□□
5. □□□□ - □□□□□□ Diameter □□□□



Diameter 网络

Diameter 网络

OmniHSS 网络部署 Diameter 网络



□□□□□□



□□□□□□

Configured

□□□□

Connecting

□□□□

Connected

□□□□

□□□□□□

□□□□□□

□□□□

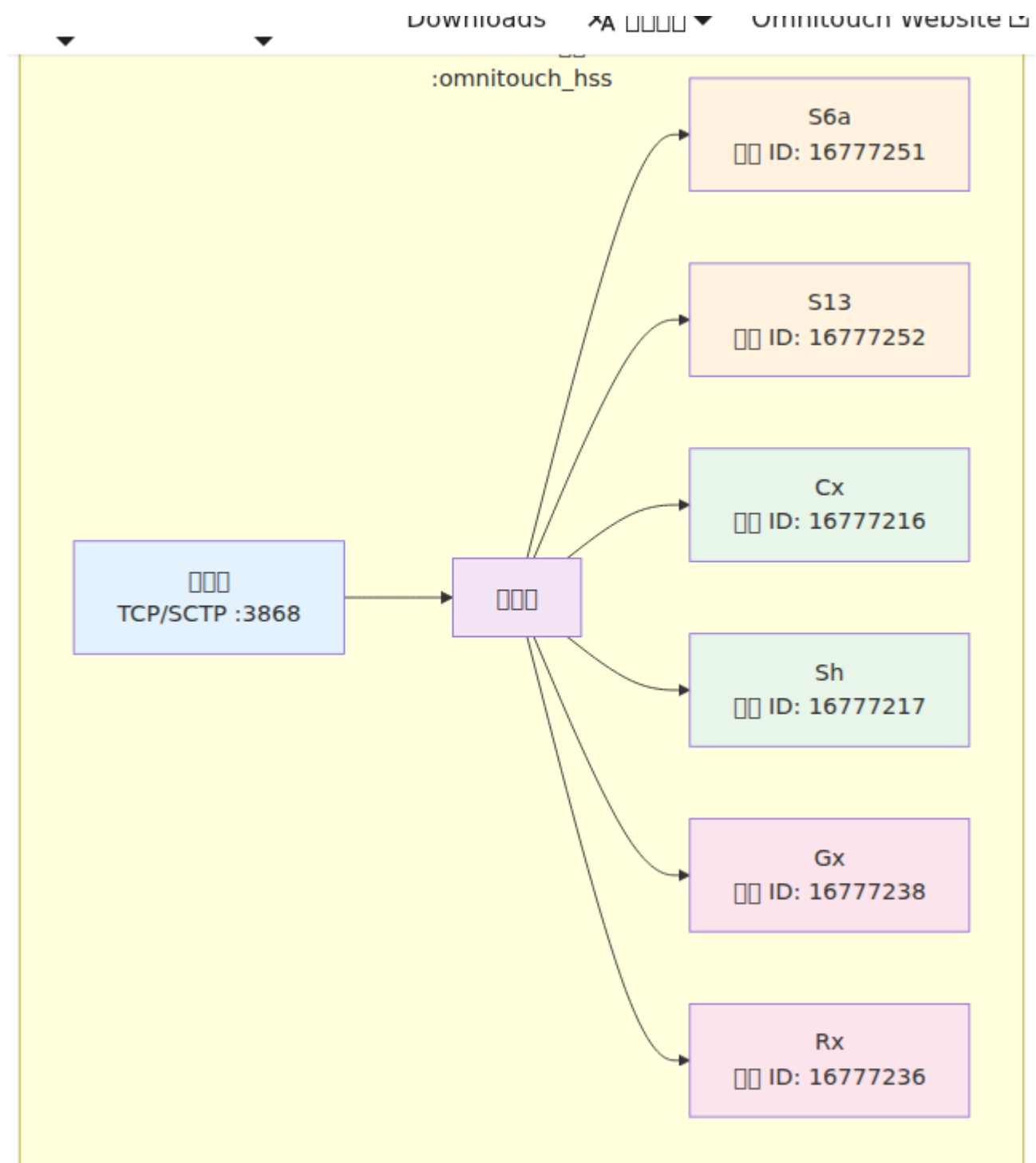
□□□□ Diameter □□



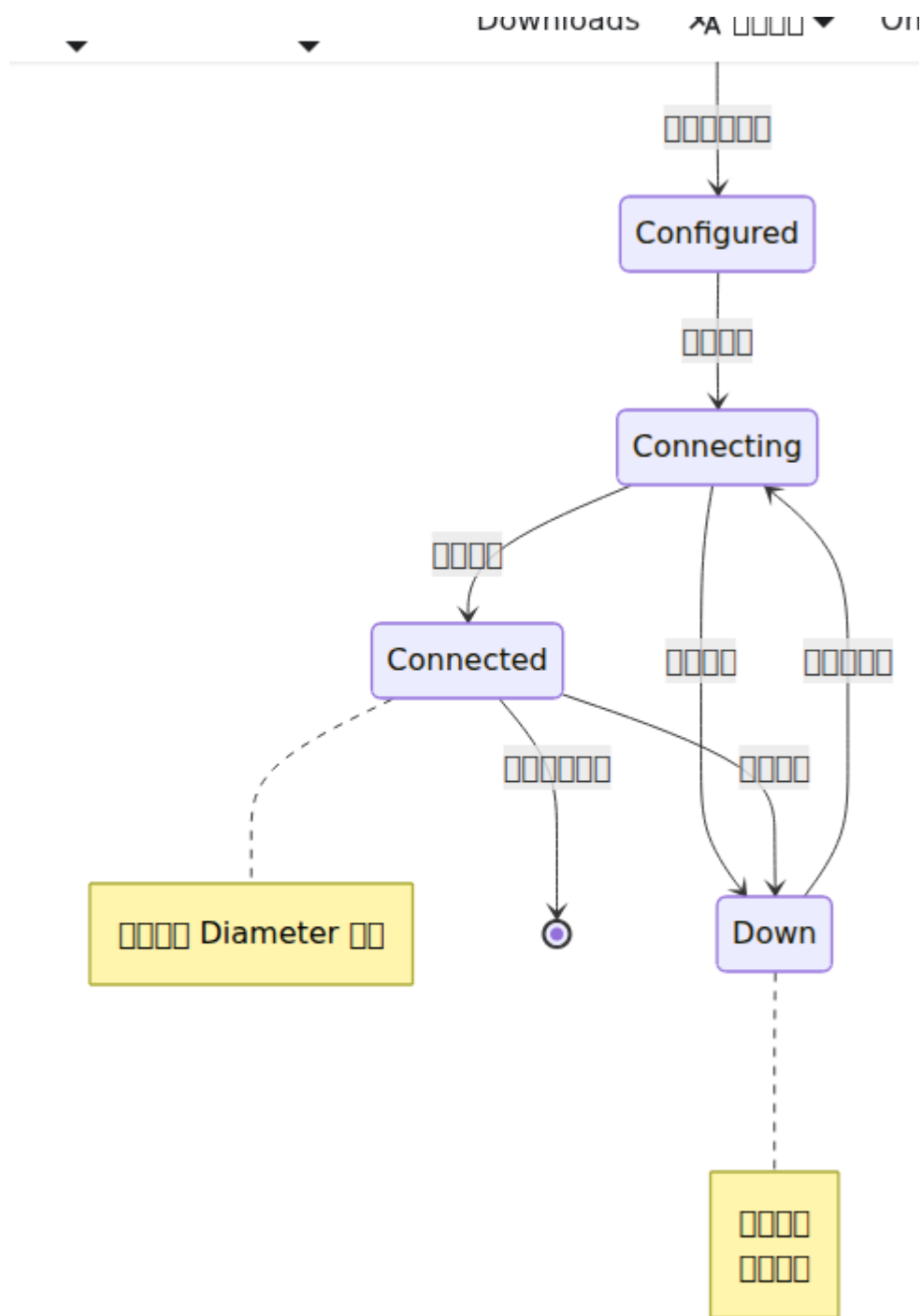
Down

□□□□
□□□□

Diameter 网络

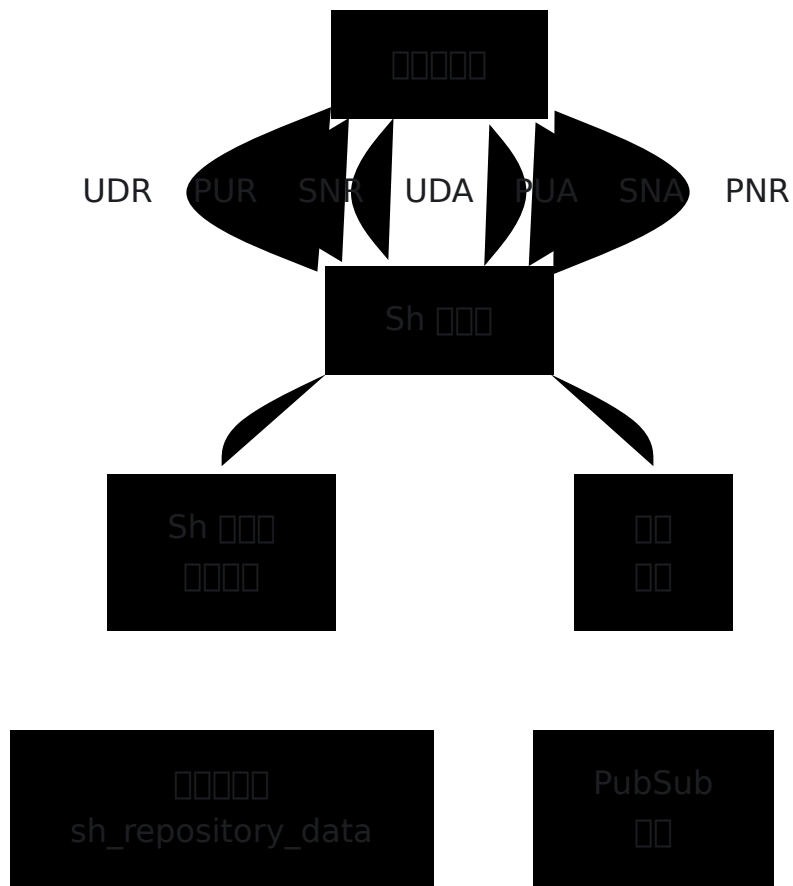


IMS 000000



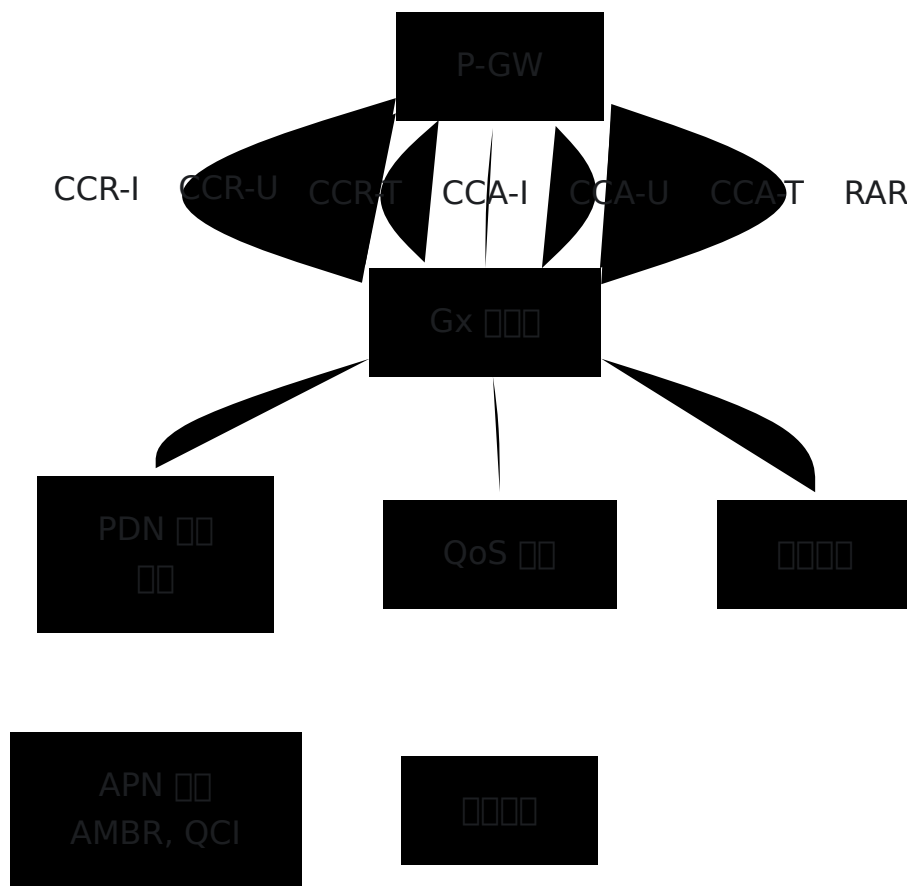
Sh (IMS)

IMS



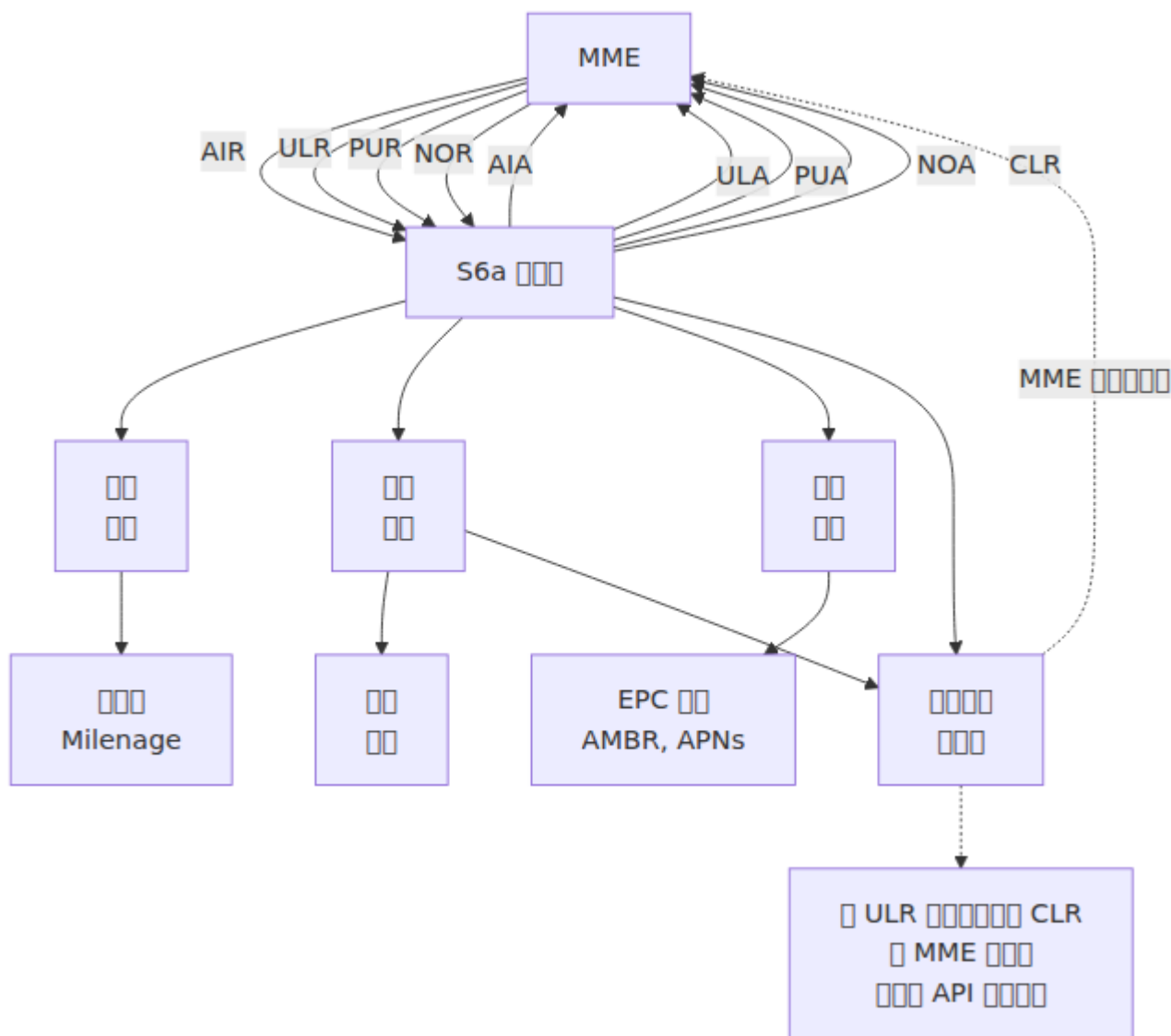
Gx ()

PCRF



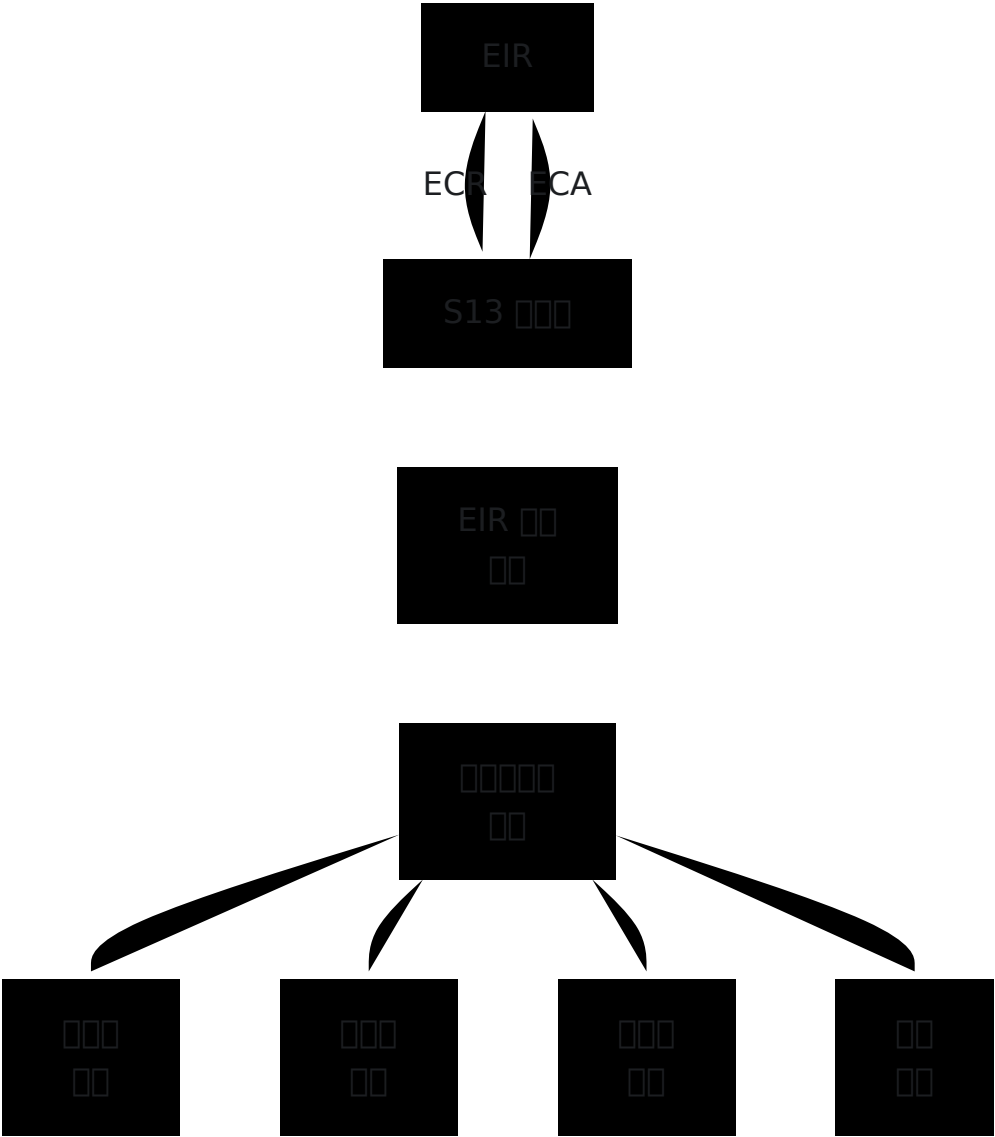
Rx (IMS)

IMS VoLTE PCRF



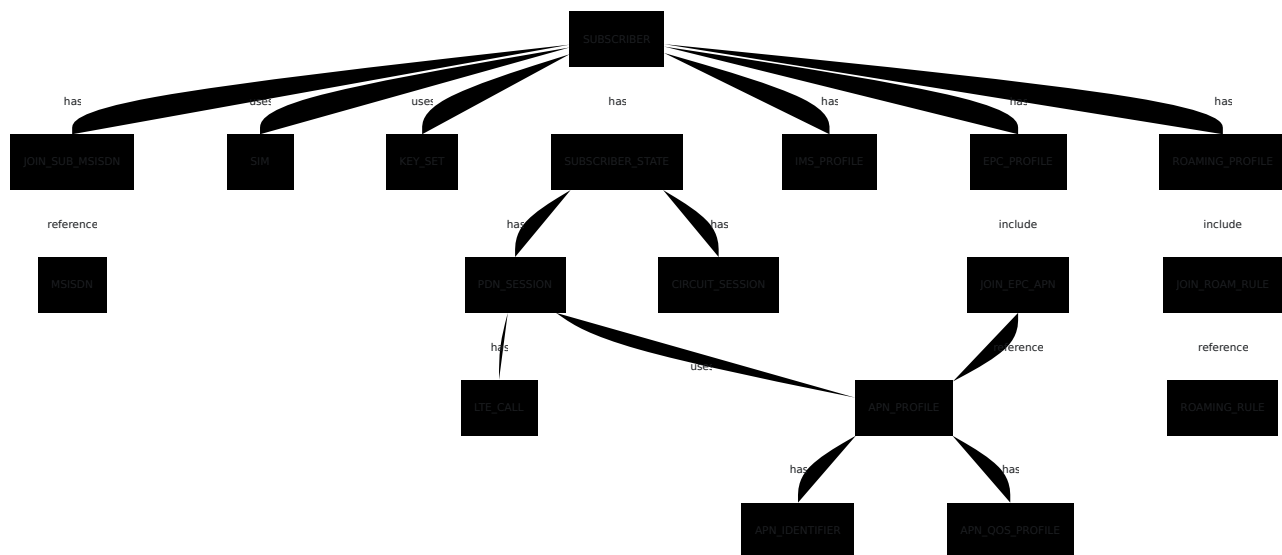
S13 (EIR)

IMEI (EIR) (EIR) (EIR)

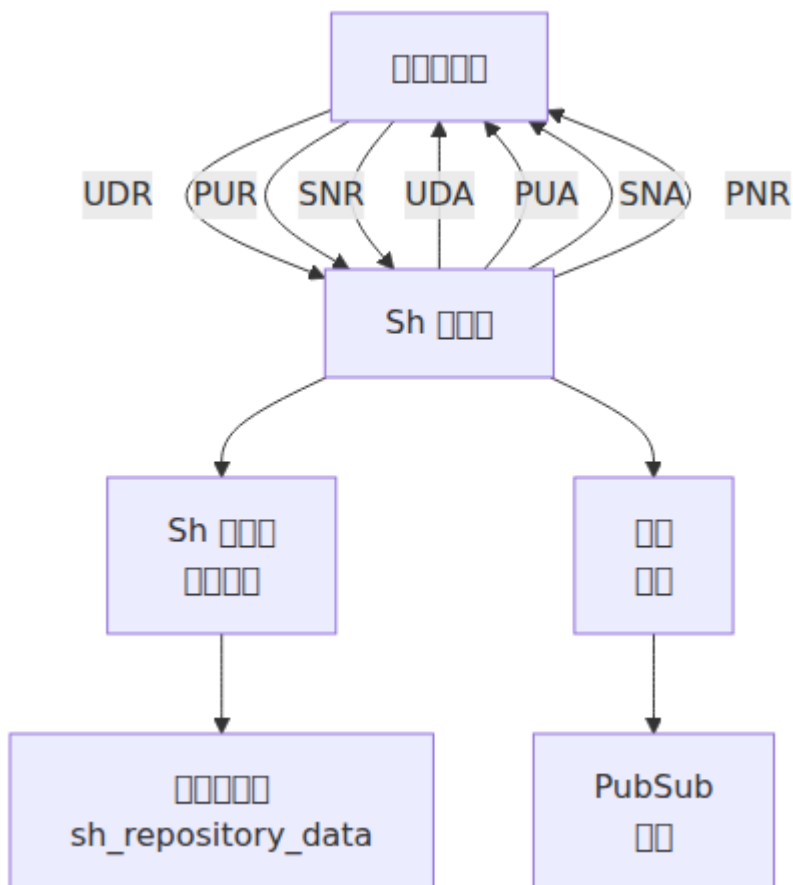


□□□

□□□□□□□

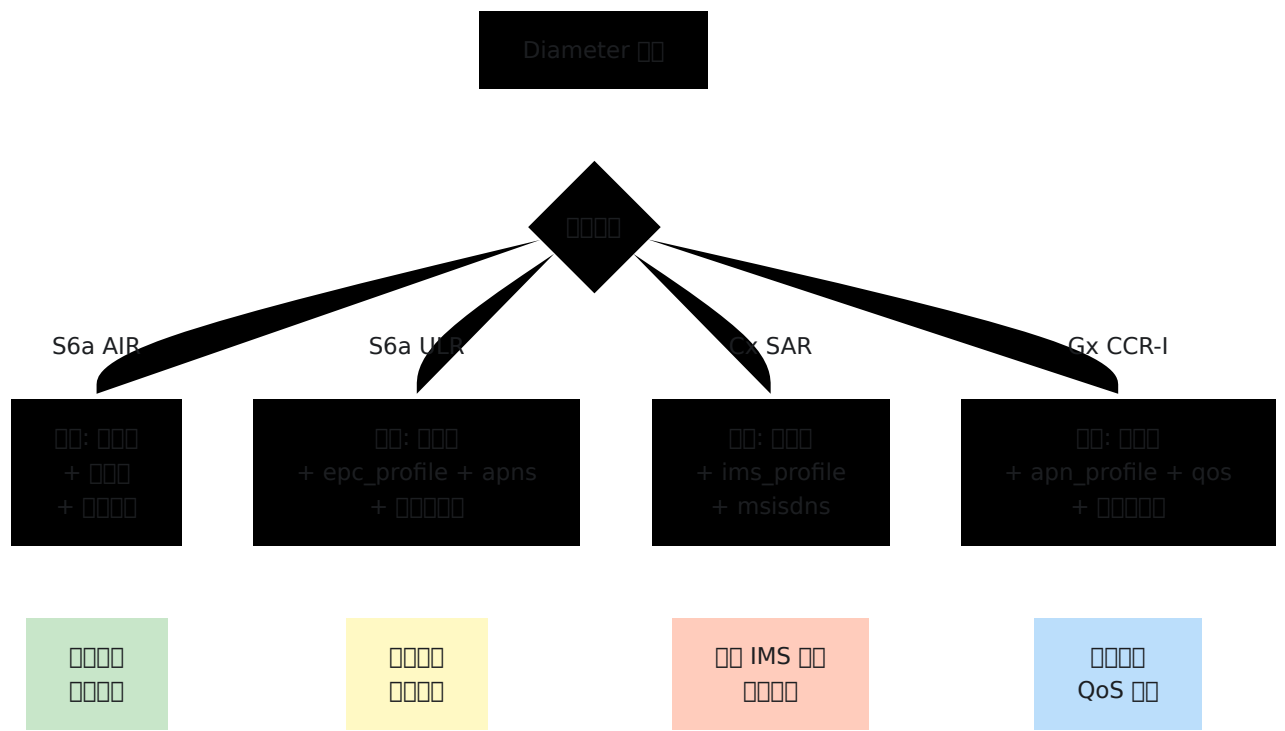


Ecto □□□□□



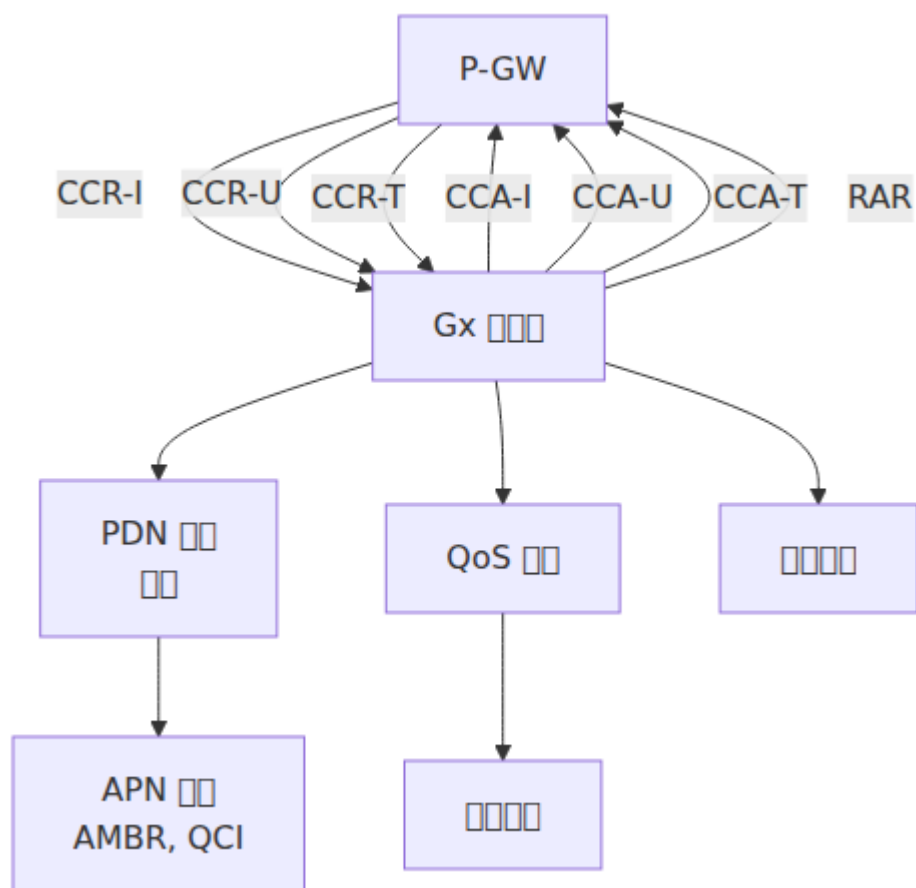
网络架构

网络 Diameter 网络架构

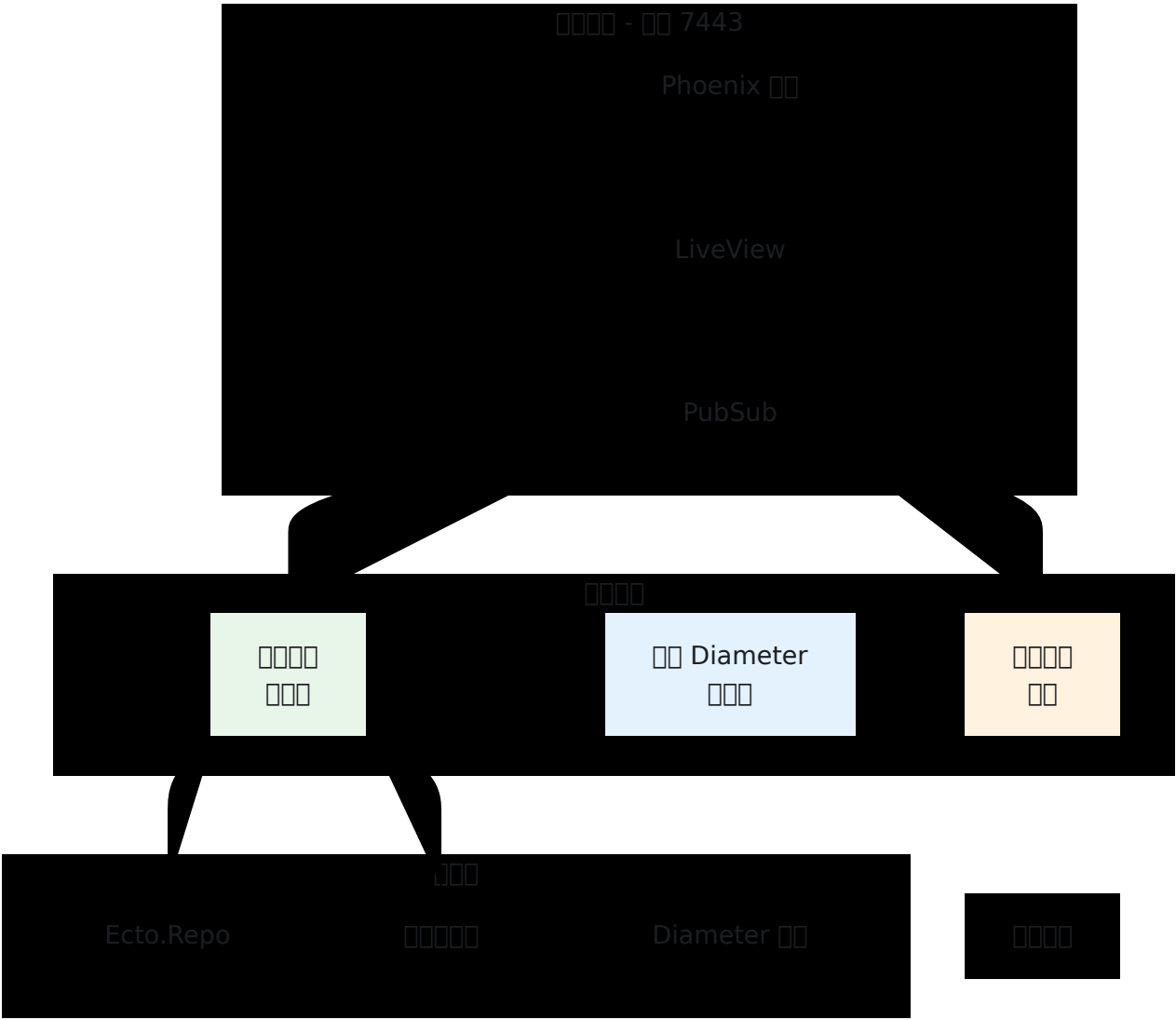


□□□□

API □□

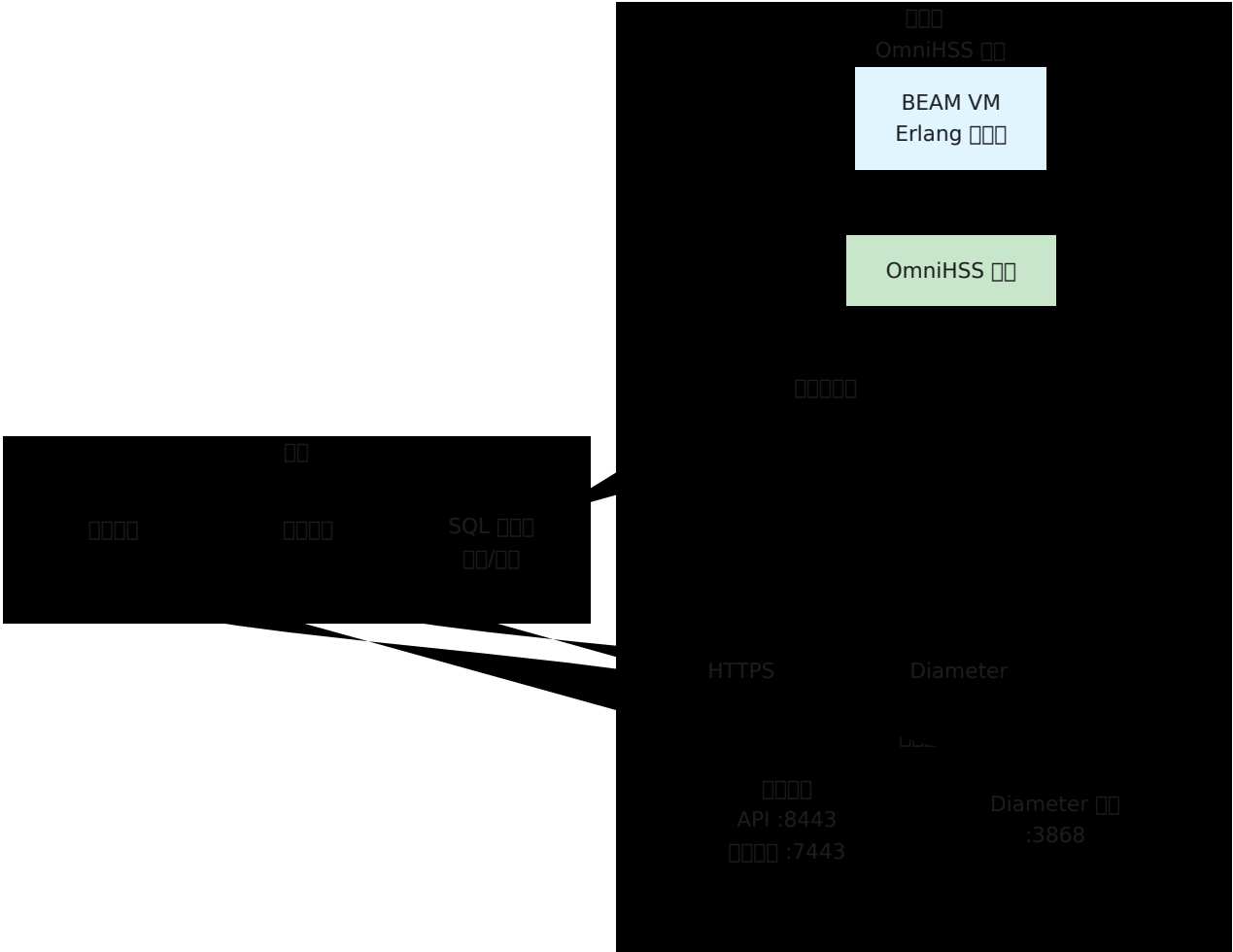


☐☐☐☐☐☐



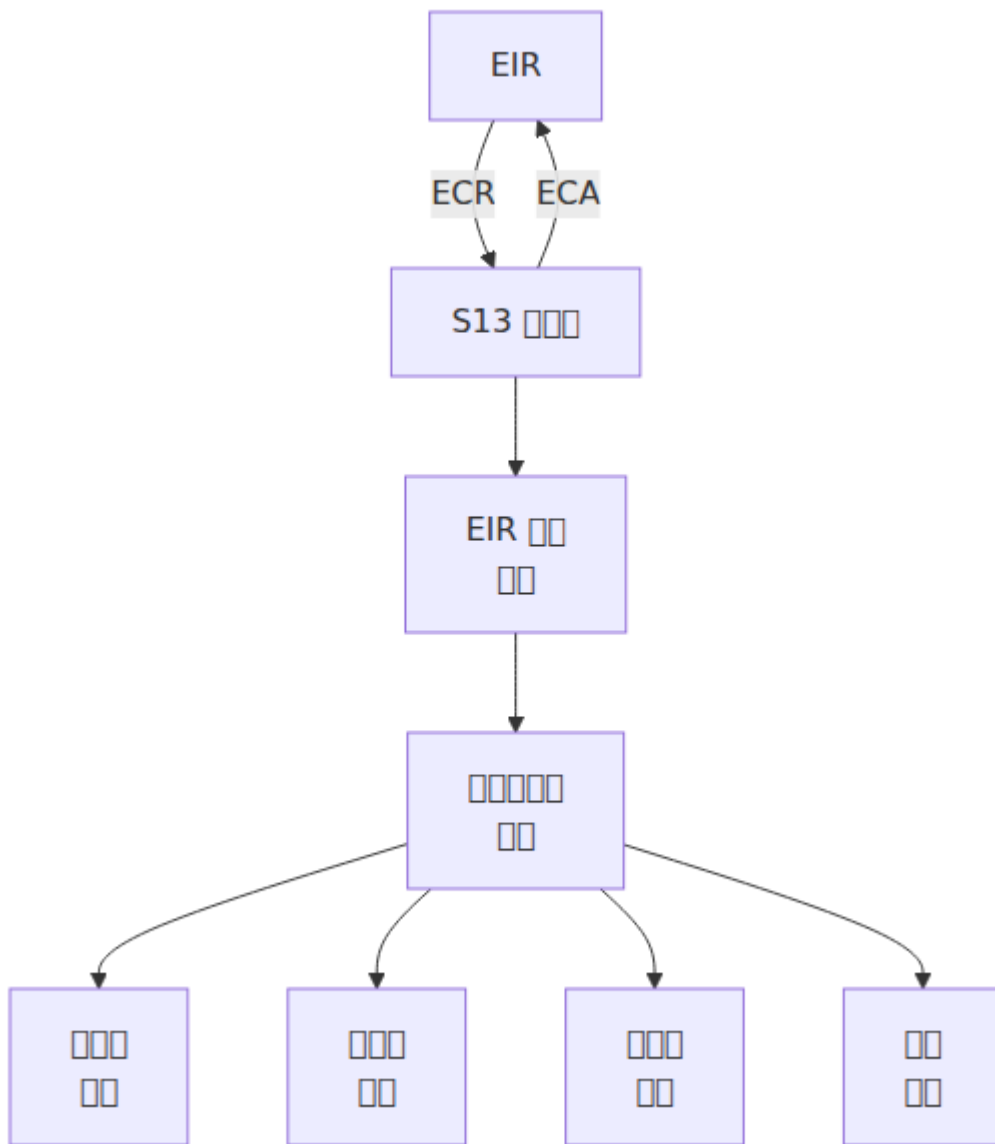
□□□□

□□□□□



□□□□□□□□

□□□□□□□□□□□□



□□□□□□

1. □□□

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

2. □□□

- □□ Diameter □□□□□□□□□□□□

- 鋼管の強度と剛性
- 鋼管の加工と施工

3. 鋼管

- 鋼管 Diameter の範囲と単位
- 鋼管の強度と剛性の計算
- 鋼管の加工と施工

4. 鋼管

- 鋼管の強度と剛性の計算
- 鋼管の加工と施工
- 鋼管の強度と剛性の計算

5. 鋼管

- 鋼管の強度と剛性の計算
- 鋼管の加工と施工
- Diameter の範囲と単位
- 鋼管の強度と剛性の計算

OmniHSS 架构图

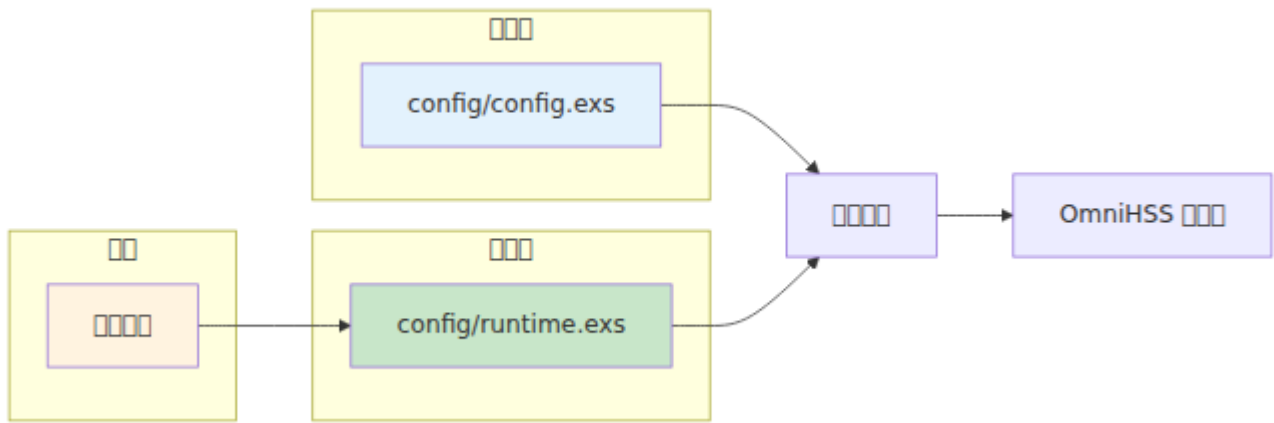
← 架构图

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 (config)

config/config.exs is a file that contains the following information:

- config/runtime.exs
- API config
- config

config/runtime.exs (runtime)

config/runtime.exs is a file that contains the following information:

- config/runtime.exs
- Diameter config
- PLMN config
- IMS S-CSCF config
- config

config/runtime.exs

config/runtime.exs is a file that contains the following information: HSS config

```
# 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",

  # 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 URL 列表	是	["https://10.0.0.1:8443/api"]
licensee	许可证 持有者 名称	是	"ACME Telecom"
product_name	许可证 产品名称	是	"omnihss"

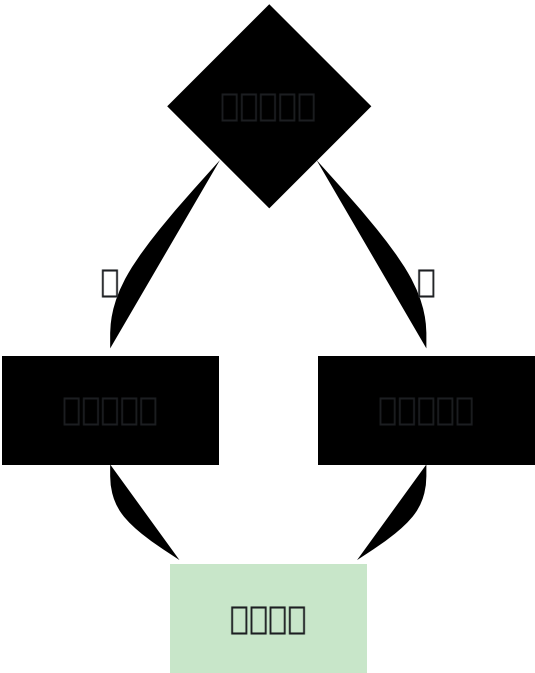
许可证

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

□□□□□

□□□□□

□□□□



□□□□□□

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")),

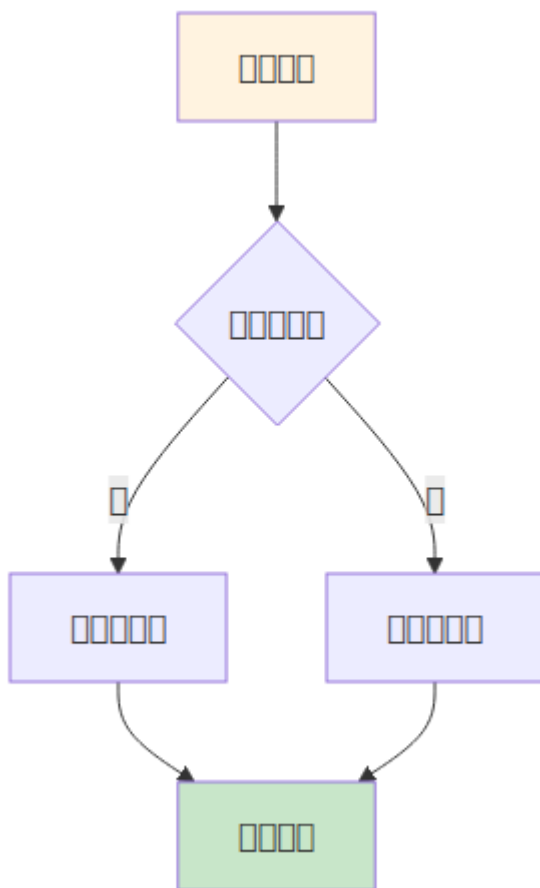
  # タイムアウト
  timeout: 15_000,
  connect_timeout: 15_000,

  # 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名	"root"	データベースユーザ名
password	SQL データベースパスワード	"password"	データベースパスワード
hostname	SQL データベースホスト名	"localhost"	データベース FQDN または IP
database	データベース名	"omnihss"	データベース名
pool_size	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 □□□□□

[illegible]

```
# 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 ☐☐

Diameter ☐☐☐☐

```
# config/runtime.exs

diameter_config = %{
  service_name: :omnitouch_hss,

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

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

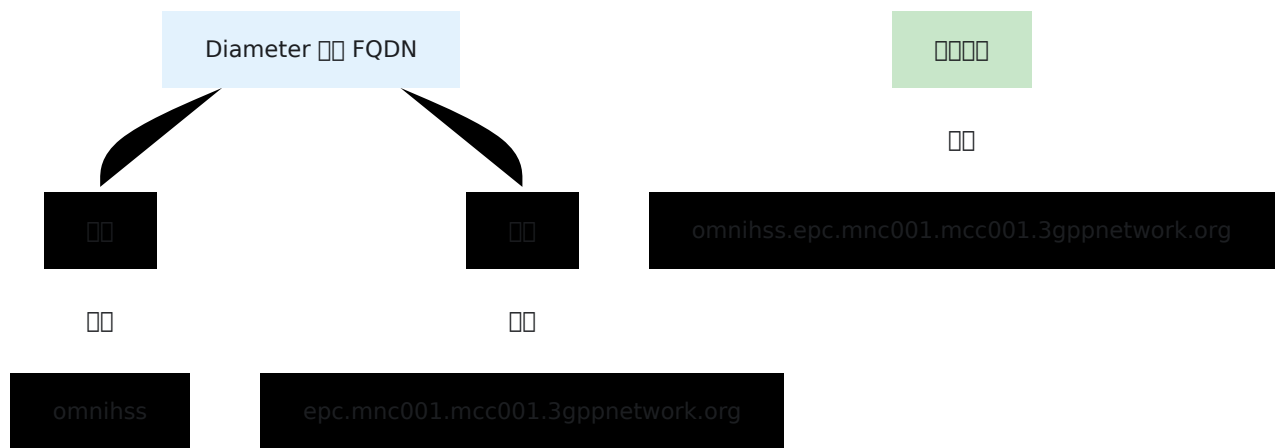
  # ☐☐☐☐
  product_name: "OmniHSS",
  vendor_id: 10415, # 3GPP
  supported_vendor_ids: [5535, 10415],

  # ☐☐☐☐
  request_timeout: 5000,

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

config :hss, :diameter, diameter_config
```

Diameter



□□□□□

- HSS 参数 "omnihss" "hss01"
- 网络 PLMN 标识 Diameter 网络
"epc.mnc001.mcc001.3gppnetwork.org"
- 主机名 {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]
```

```
  }
```

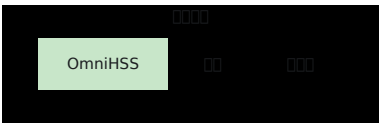
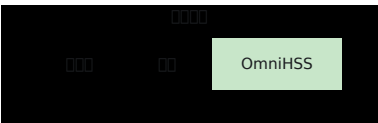
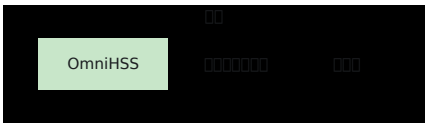
```
]
```

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
```

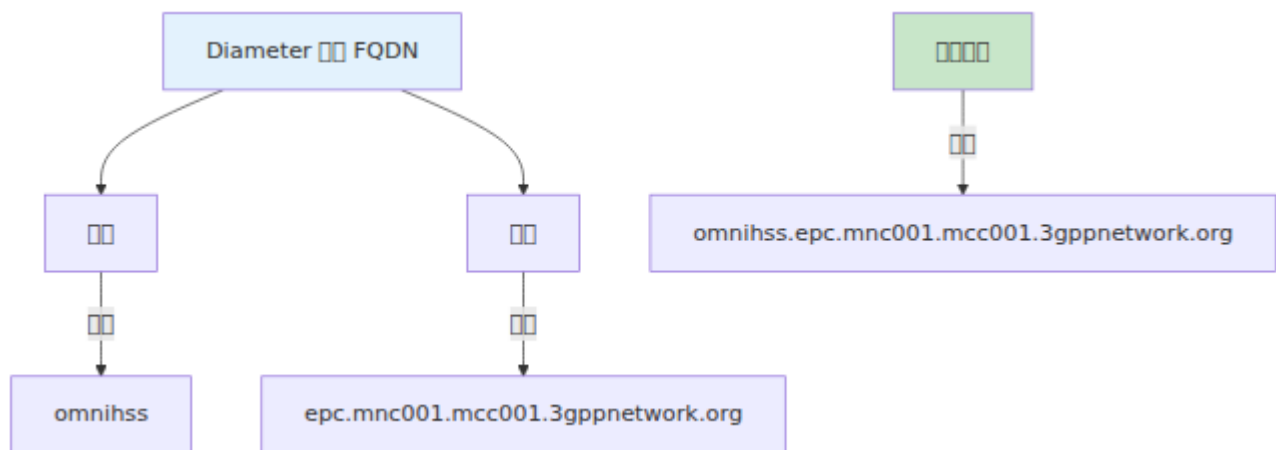
```
  license_enforced: true,
```

```
  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



PLMN

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

□□□□□□

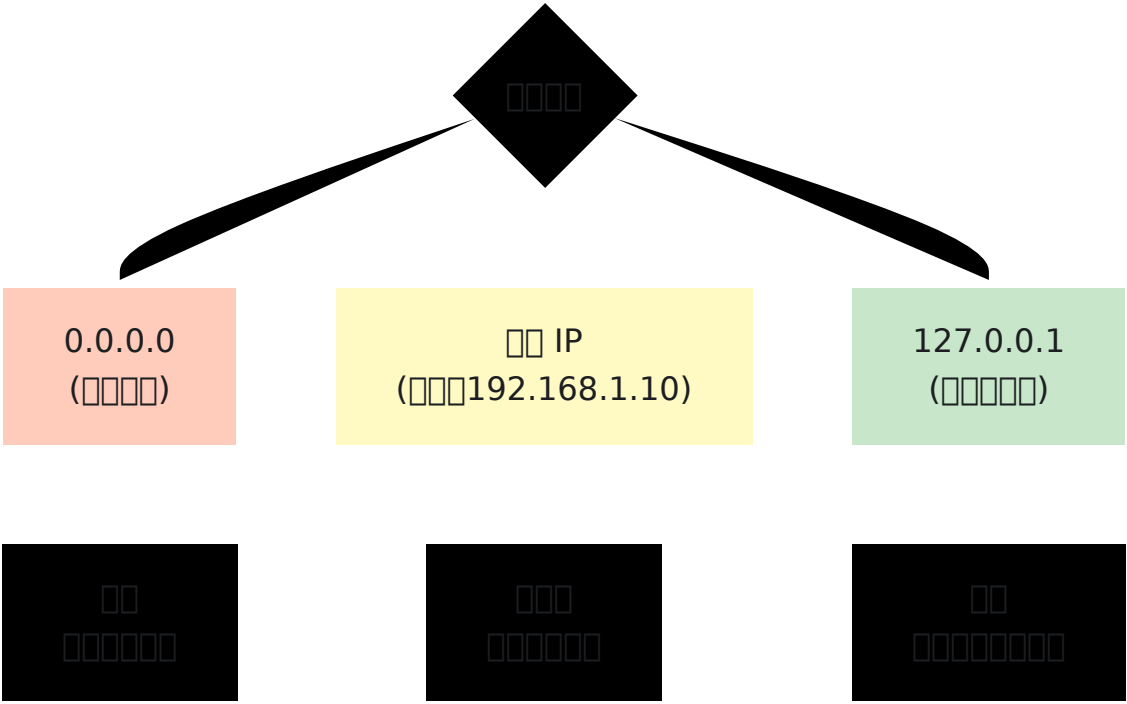
```
# config/runtime.exs

# 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

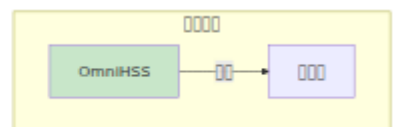
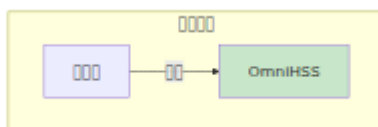
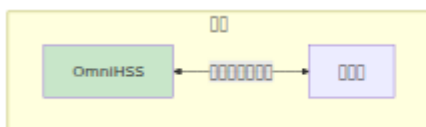
S-CSCF

```
# 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"
```

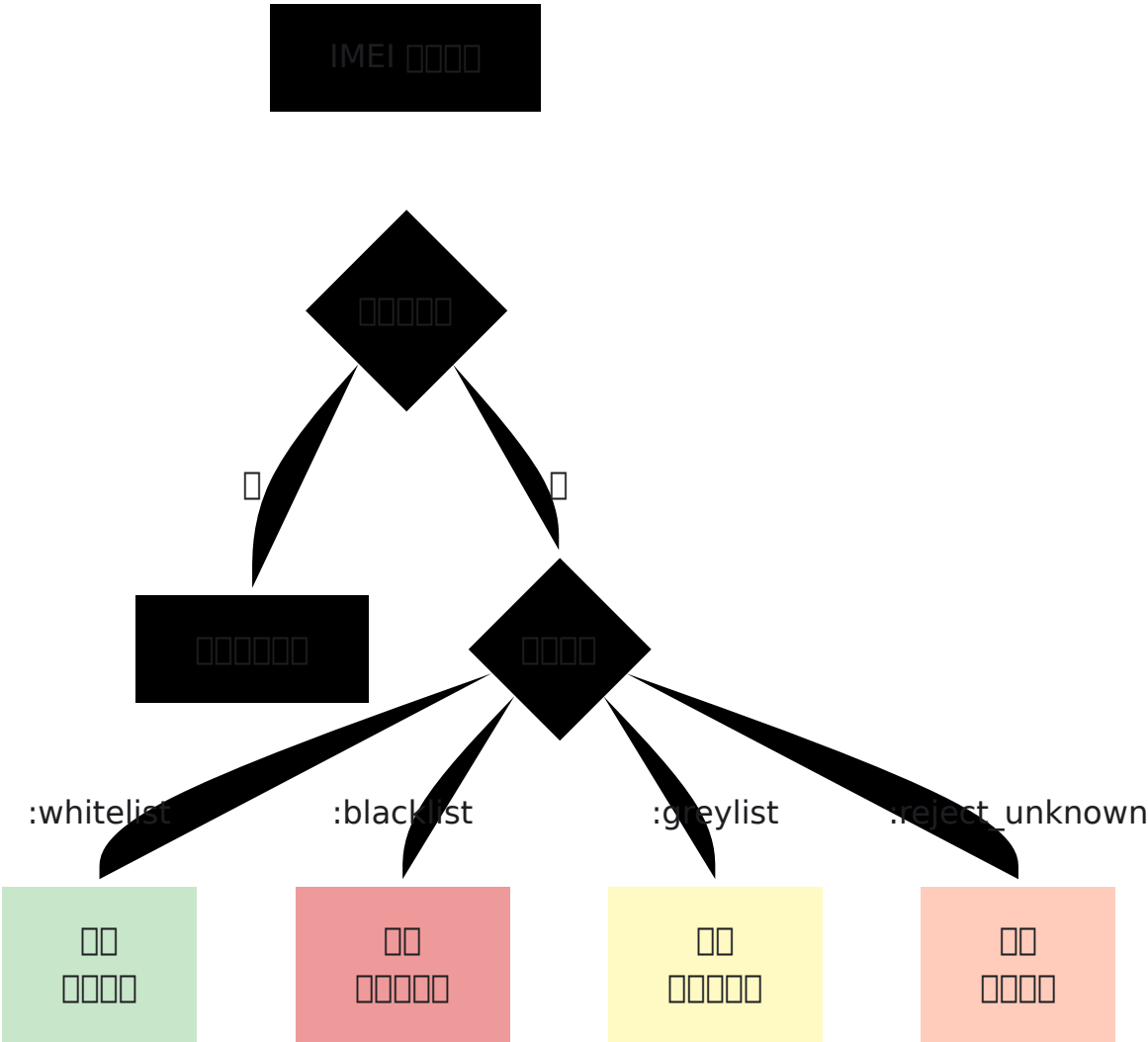
EIR

EIR

```
# config/runtime.exs

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

IMEI 黑名单



黑名单

黑名单	黑名单	黑名单
:whitelist	黑名单 IMEI	黑名单
:blacklist	黑名单 IMEI	黑名单
:greylist	黑名单 IMEI	黑名单
:reject_unknown_equipment	黑名单	黑名单

黑名单 :whitelist 黑名单 :greylist 黑名单 :blacklist 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"
]
```

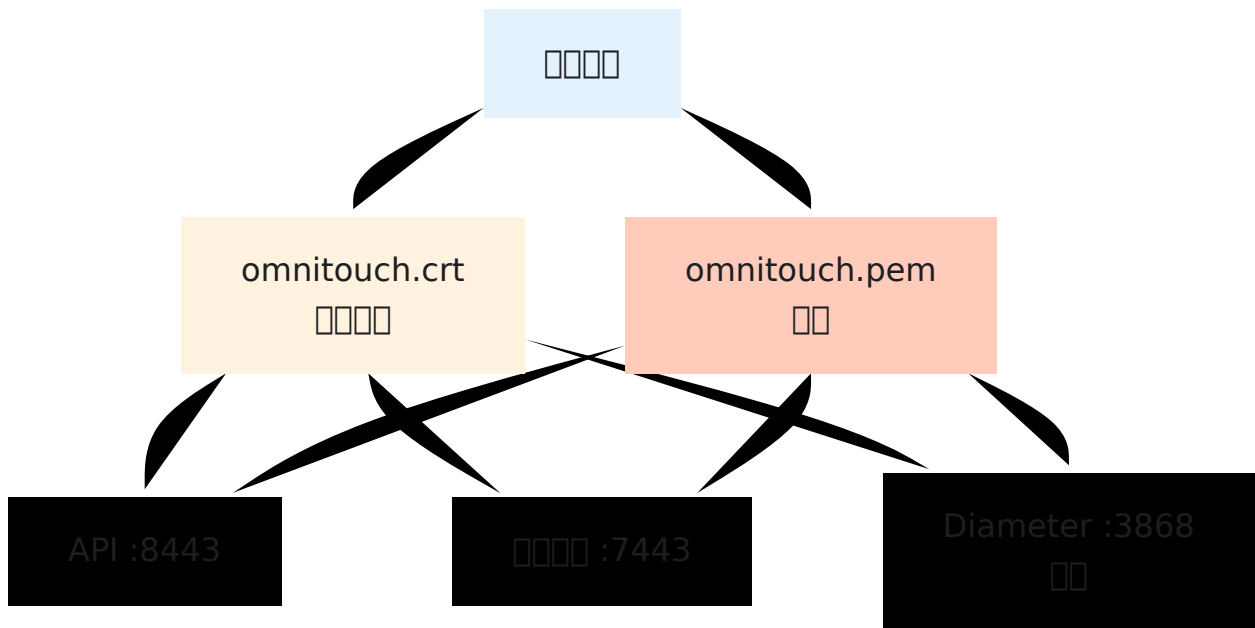
環境構築

```
# 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

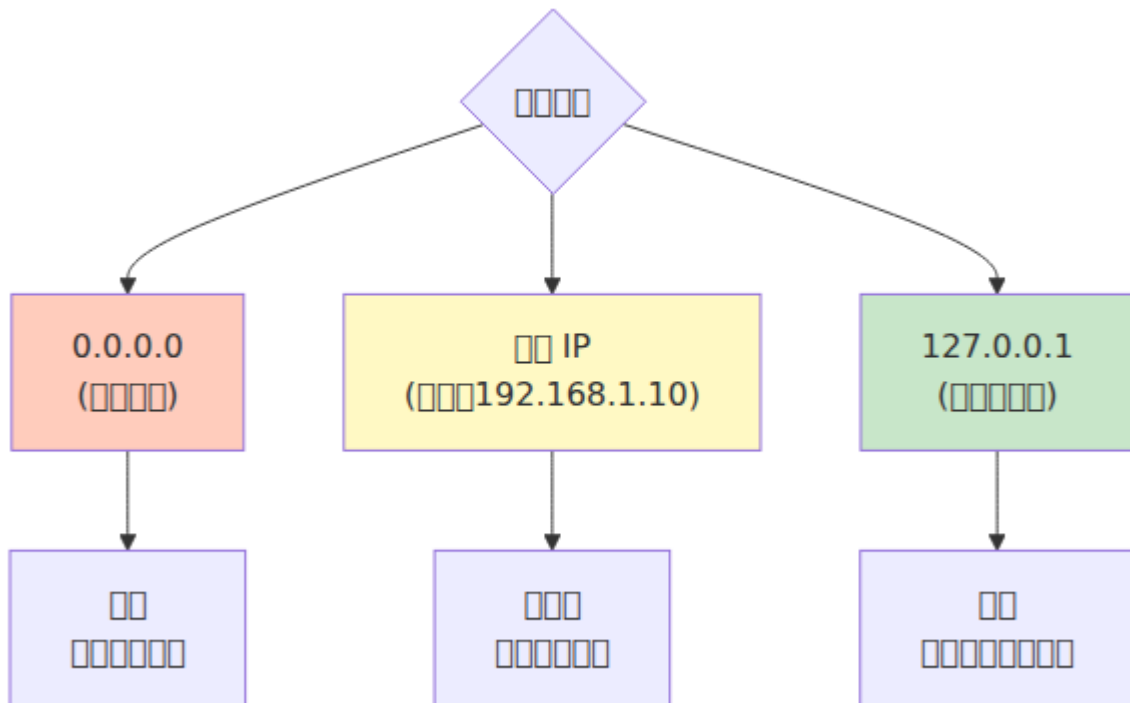
- 証明書 X.509 形式
- 証明書
- 証明書
- CN と SAN 証明書

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 数据库 DNS 配置

IMS 数据库配置 IMS 数据库

☐ 配置 S-CSCF 数据库

☐ 配置 S-CSCF 数据库

☐ 配置 IMS 数据库

数据库

☐ 配置 EIR 数据库

☐ 配置数据库

☐ 配置数据库

数据库

数据库

1. 数据库

```
数据库数据库数据库
```

2. 数据库

```
配置 https://[hostname]:7443  
数据库数据库
```

3. API 配置

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

4. Diameter 配置

```
数据库数据库 Diameter 配置  
数据库数据库
```

5. 資料庫

資料庫系統
資料庫 SQL 查詢

omnihss

```
# config/runtime.exs - 配置
```

```
import Config
```

```
#
```

```
=====
```

```
# 数据库配置
```

```
#
```

```
=====
```

```
config :hss, Hss.Repo,
```

```
  username: System.fetch_env!("DATABASE_USERNAME"),
```

```
  password: System.fetch_env!("DATABASE_PASSWORD"),
```

```
  hostname: System.get_env("DATABASE_HOSTNAME", "db.omnihss.internal"),
```

```
  database: System.get_env("DATABASE_NAME", "omnihss"),
```

```
  port: String.to_integer(System.get_env("DATABASE_PORT", "3306")),
```

```
  pool_size: String.to_integer(System.get_env("DATABASE_POOL_SIZE", "10")),
```

```
  timeout: 15_000,
```

```
  connect_timeout: 15_000,
```

```
  ssl: true,
```

```
  ssl_opts: [
```

```
    cacertfile: "/etc/ssl/certs/mysql-ca.pem",
```

```
    verify: :verify_peer
```

```
  ]
```

```
#
```

```
=====
```

```
# 许可证配置
```

```
#
```

```
=====
```

```
config :license_client,
```

```
  license_server_api_urls: [System.get_env("LICENSE_SERVER_URL",
```

```
"https://license.example.com:8443/api")],
```

```
  licensee: System.get_env("LICENSE_ORGANIZATION", "omnihss"),
```

```
  product_name: "omnihss"
```

```
#
```

```
=====
```

```
# 配置 PLMN 和 HSS 地址
```

```
#
```

```
=====
```

```

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

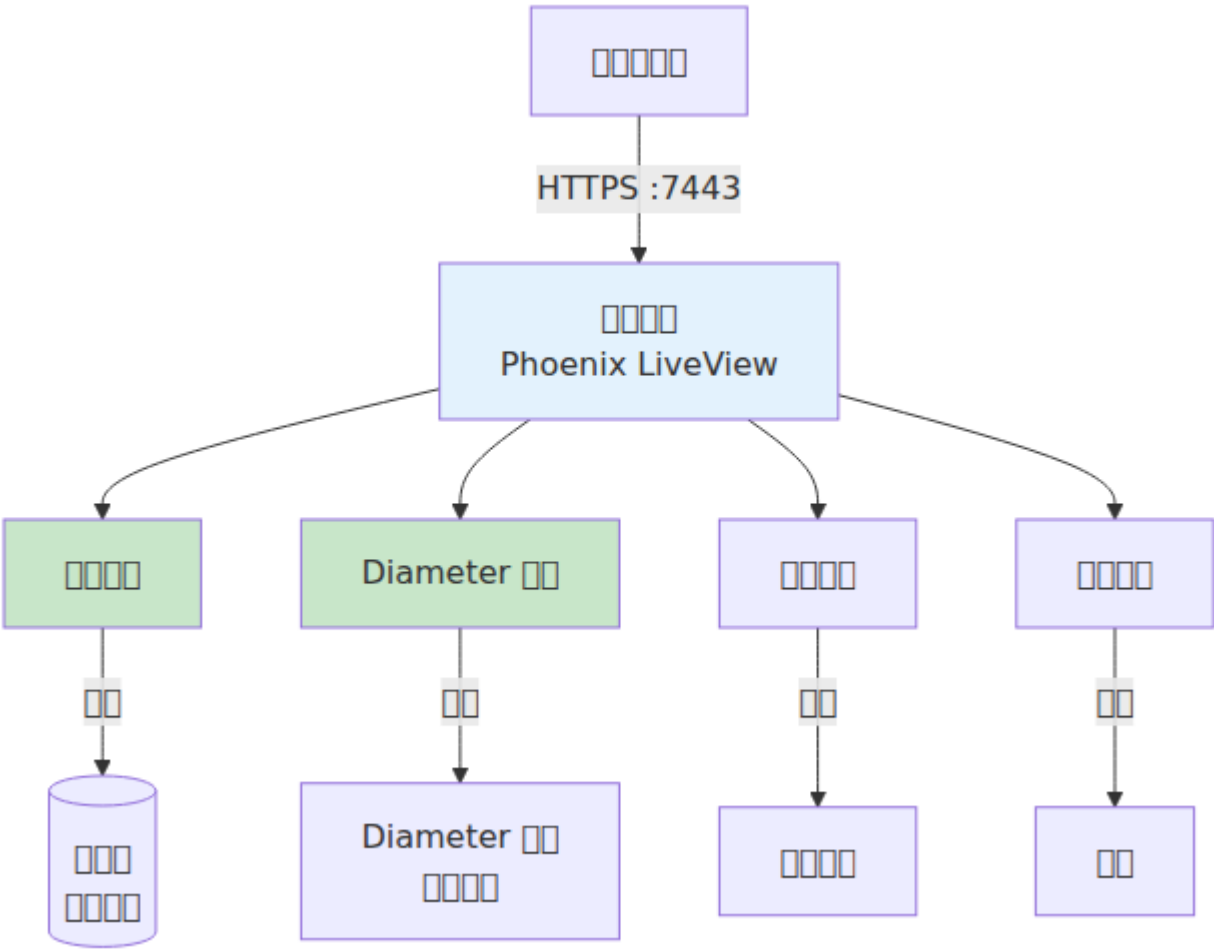
開箱

- 開箱 - 開箱開箱
- 開箱 - 開箱開箱開箱開箱
- **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

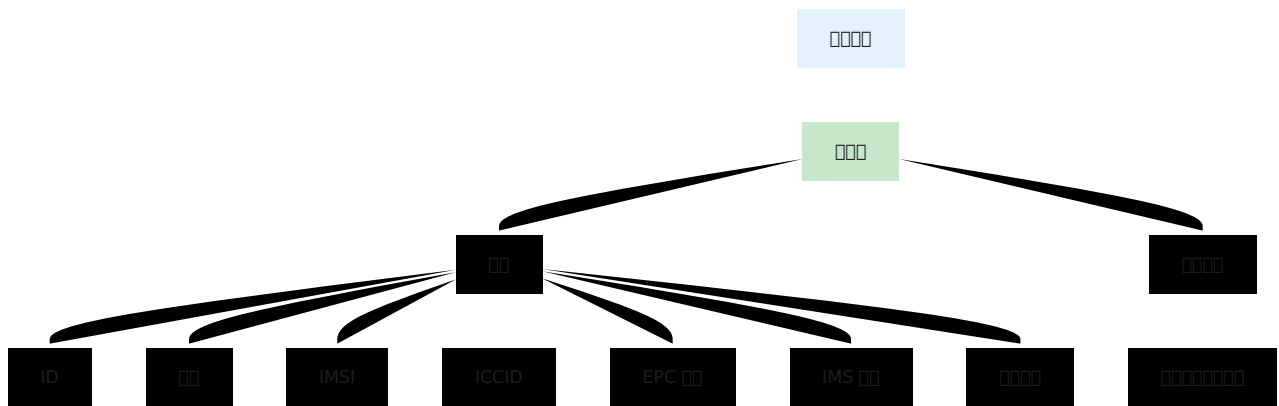
証明書を確認 Internet Explorer

証明書

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

証明書を確認

--	--	--	--



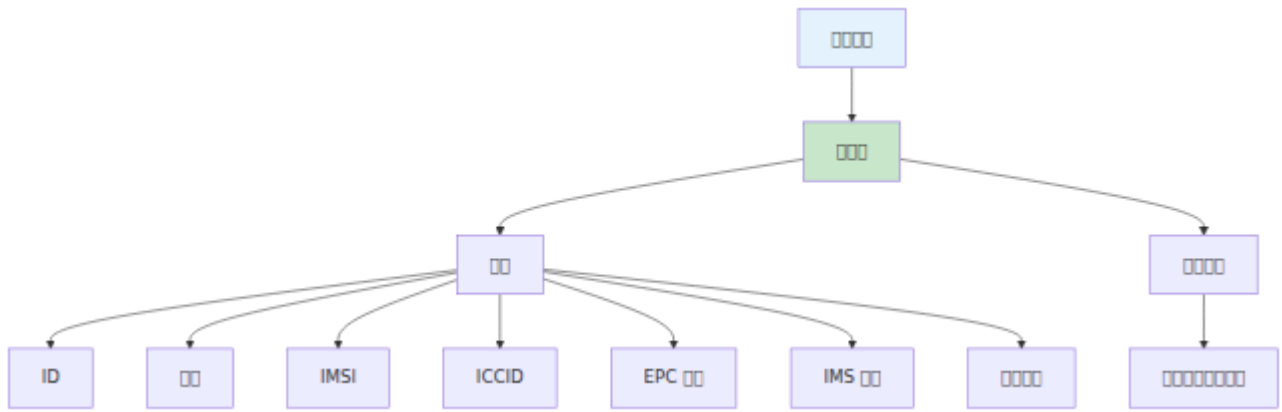
11

欄	項目	説明
ID	電話番号 ID	番号
番号	番号	✓ 番号 / ✕ 番号
IMSI	国際移動 subscriber ID	14-15 番号
ICCID	SIM 番号 ID	19-20 番号 "N/A"
EPC 番号	電子郵便番号	電子郵便 ID
IMS 番号	インターネット移動番号	インターネット ID 番号 "N/A"
番号	番号	インターネット ID 番号 "N/A"

--	--	--	--	--	--	--	--

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□□□□



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

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

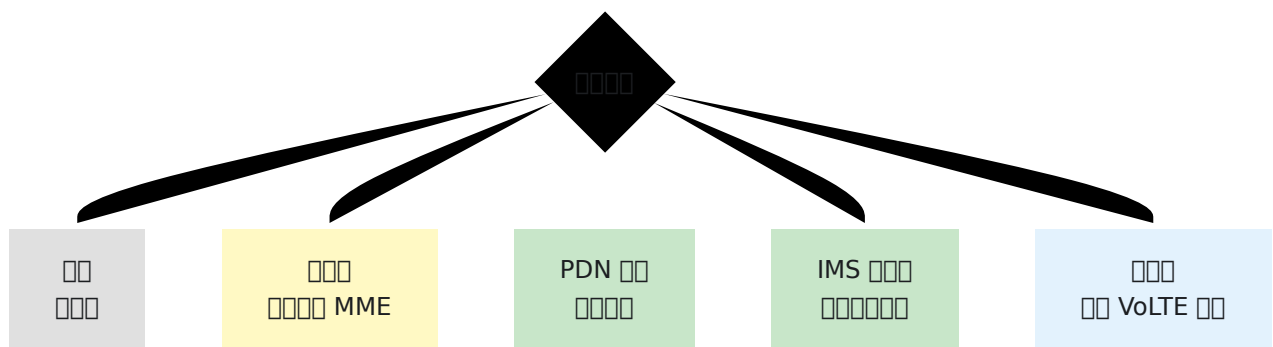
IMS

- **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

□□□

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

□□□□□□



□□□□□□□□

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

□□□□□□

□□□□□□ **1** □ □□□□□□□□□□□□□□□□□□□

□□□□□□

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

□□□□□□

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

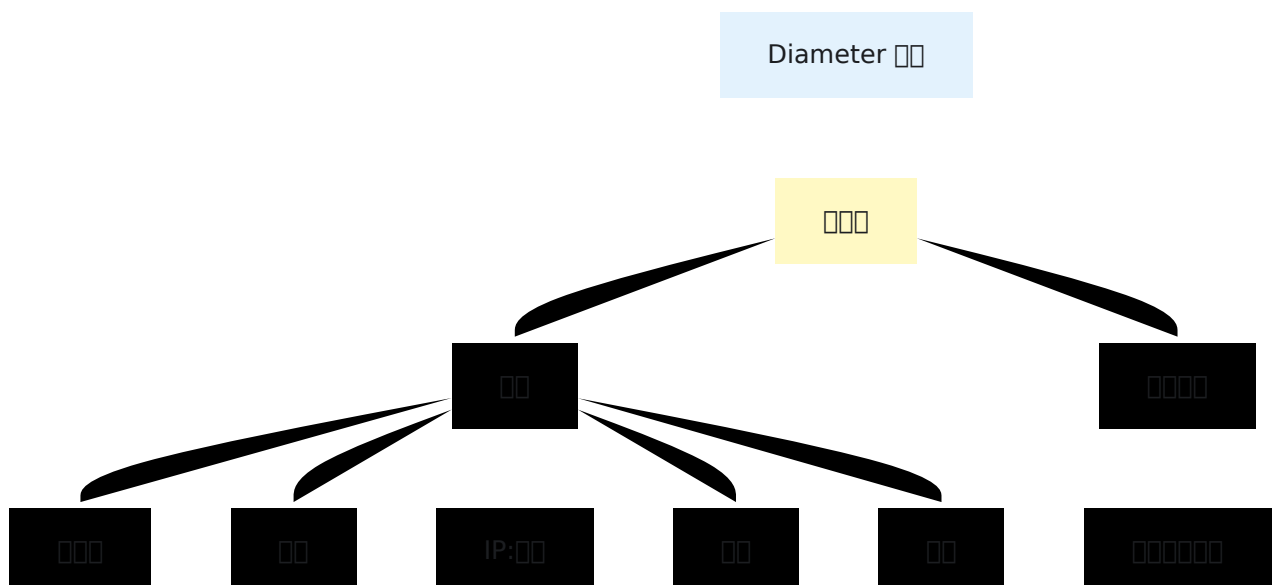
- 网络功能虚拟化
- 网络 PDN 网络
- 网络 VoLTE 网络

Diameter 网络

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

Diameter 网络 Diameter 网络

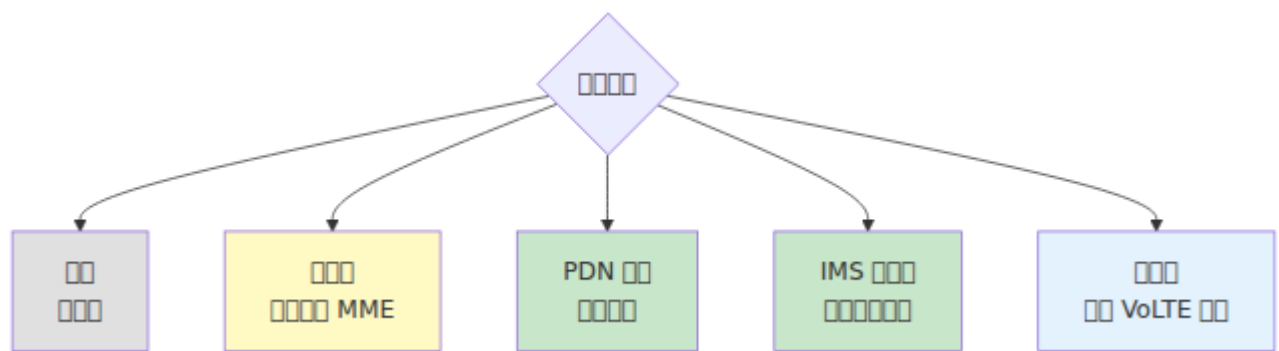
网络



□□

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

□□□□



□□□□□□□□

□□□□□□□□□□□□□□

□□□□□

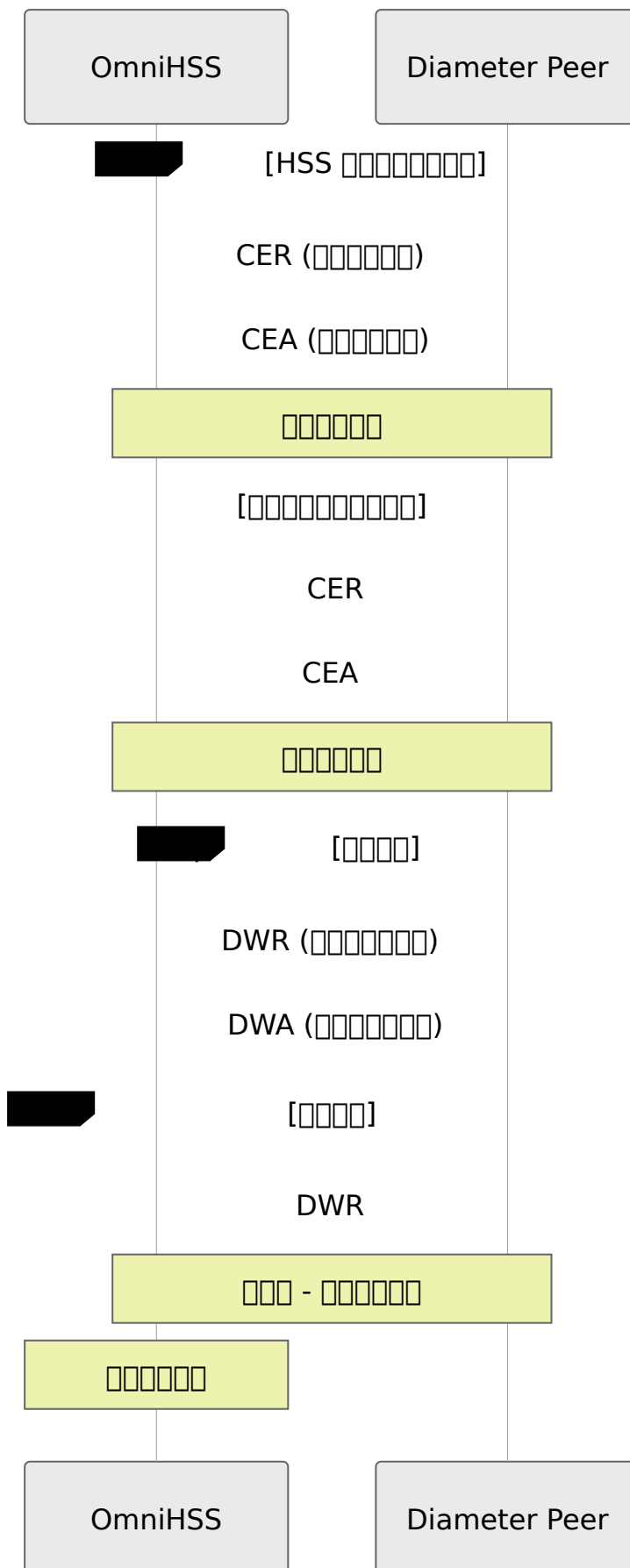
- □□□□ - □ HSS □□□□□
- □□□□ - □□□□□□□□
- □□ **ID** - □□□ Diameter □□

□□ **ID** □□□

- 16777251 - S6a (MME)

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

--	--	--	--	--	--



□□□□

Diameter □□□ **1** □ □□□□□

□□□□

1. □□□□□

- □□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□

2. □□□□

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

3. □□□□

- □□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□

□□□□

□□□□□□□□

□□□□□□

1. □□□□□□

2. □□□□□□□□□□□□

3. □□□□□□□□

4. Diameter □□□□□□

5. □□□□□□□□□□ TLS□

□□□□□□□□

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

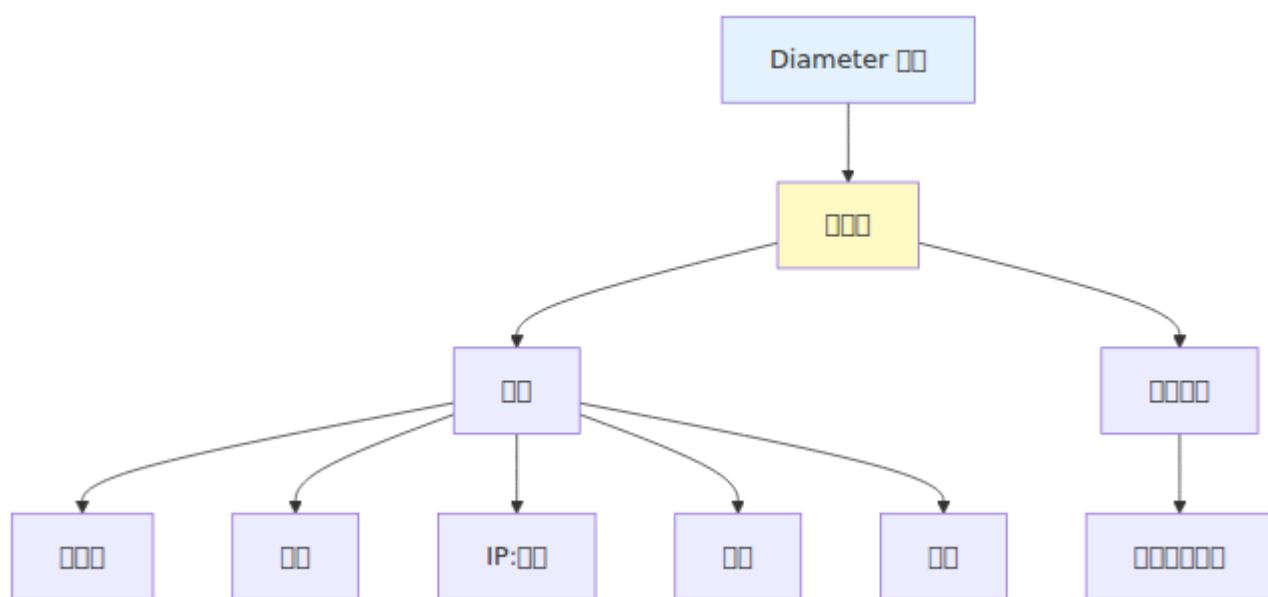
- 1.
- 2.
- 3.
4. Diameter

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

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

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

□□□□



□□□□

1. □□□□

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

2. □□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

3. □□□□

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

部署

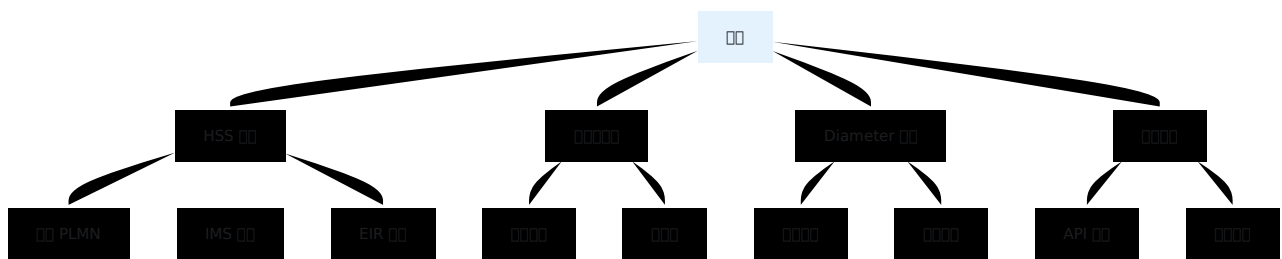
URL: `https://[hostname]:7443/configuration`

部署 OmniHSS 部署图

部署

- 部署 - 部署图
- 部署 - 部署图
- 部署 - 部署图

部署



部署

1. 部署

- 部署 runtime.exs 部署图
- 部署图
- 部署 Diameter 部署图

2. 部署

- 部署图
- 部署图
- 部署图

3. 部署

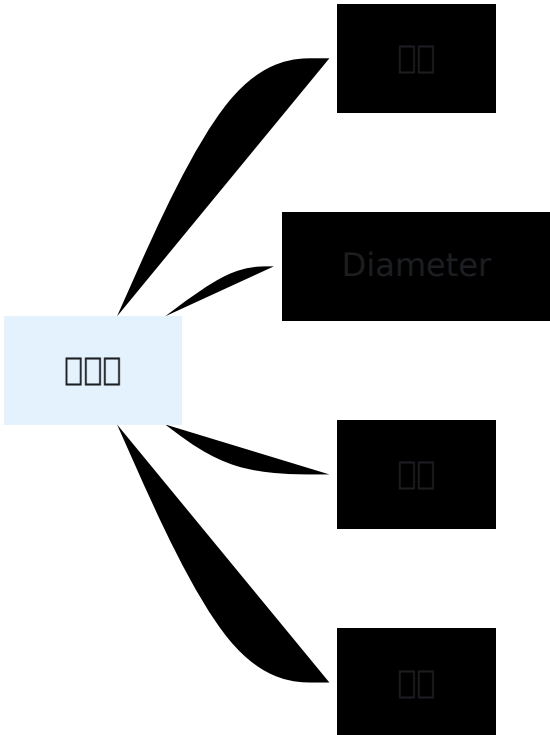
- 部署图

- □□□□□□□□

[illegible]

--	--	--	--	--

--	--	--	--	--



□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □

□ □

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

□□□□□□□□

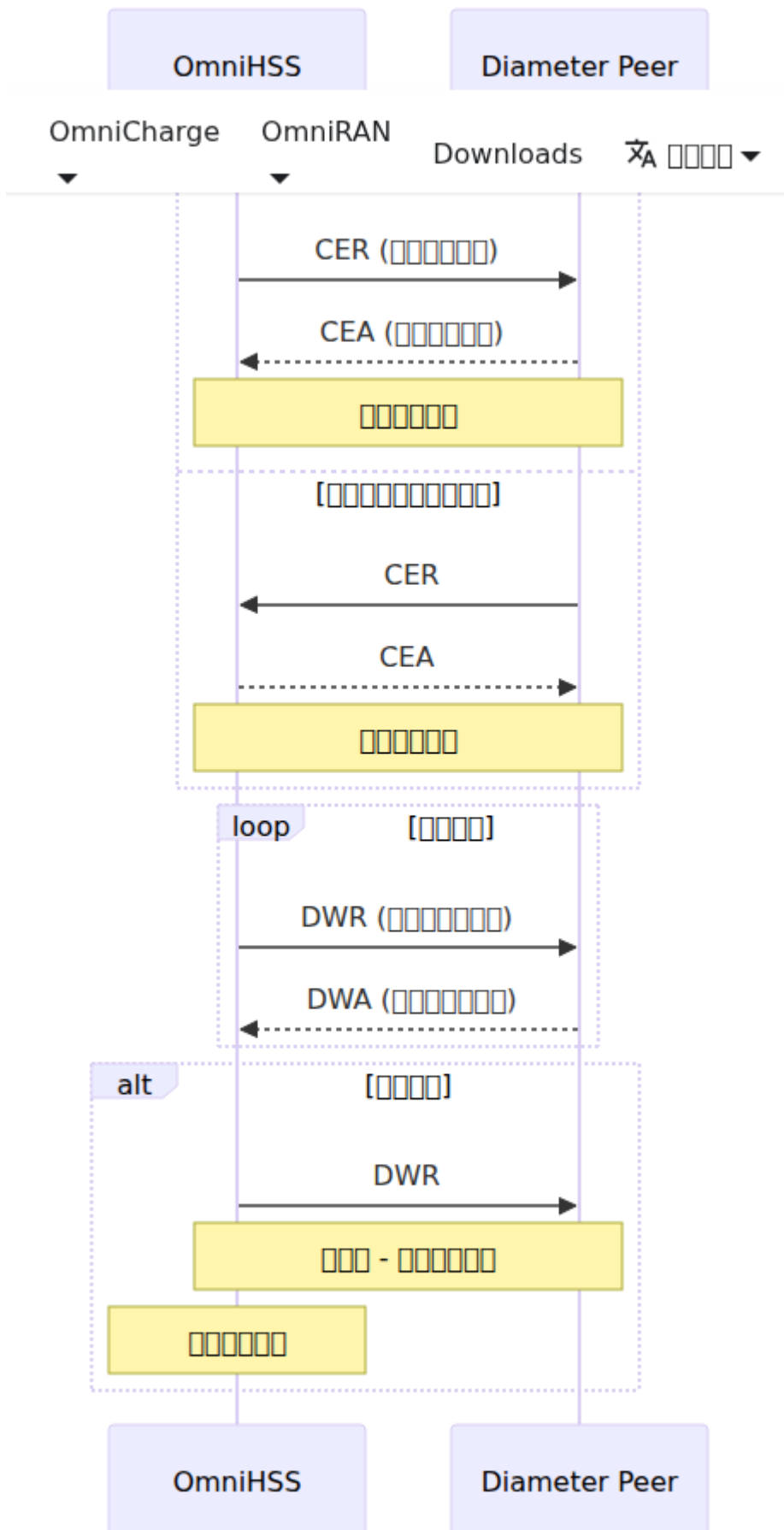


Table 1

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% achieved

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	

動作

- whitelist - 許可
- blacklist - 拒否
- greylist - 監視

IMEI/IMEISV と IMSI

動作

動作	動作	動作	動作
許可	0	許可	許可
許可	1	許可/拒否	許可
許可	2	許可	許可

S13

動作

ECR/ECM

MME/SGSN → HSS

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 字节
- Vendor-Specific-Application-Id

EIR

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

AVPs

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

消息

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

Android

Android

EIR

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

Android

1. IMEISV

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

2. IMEI

- IMEI → "35979139461611"
- EIR
-

3. IMSI

- IMSI → "999999876543210"
- EIR
-
-

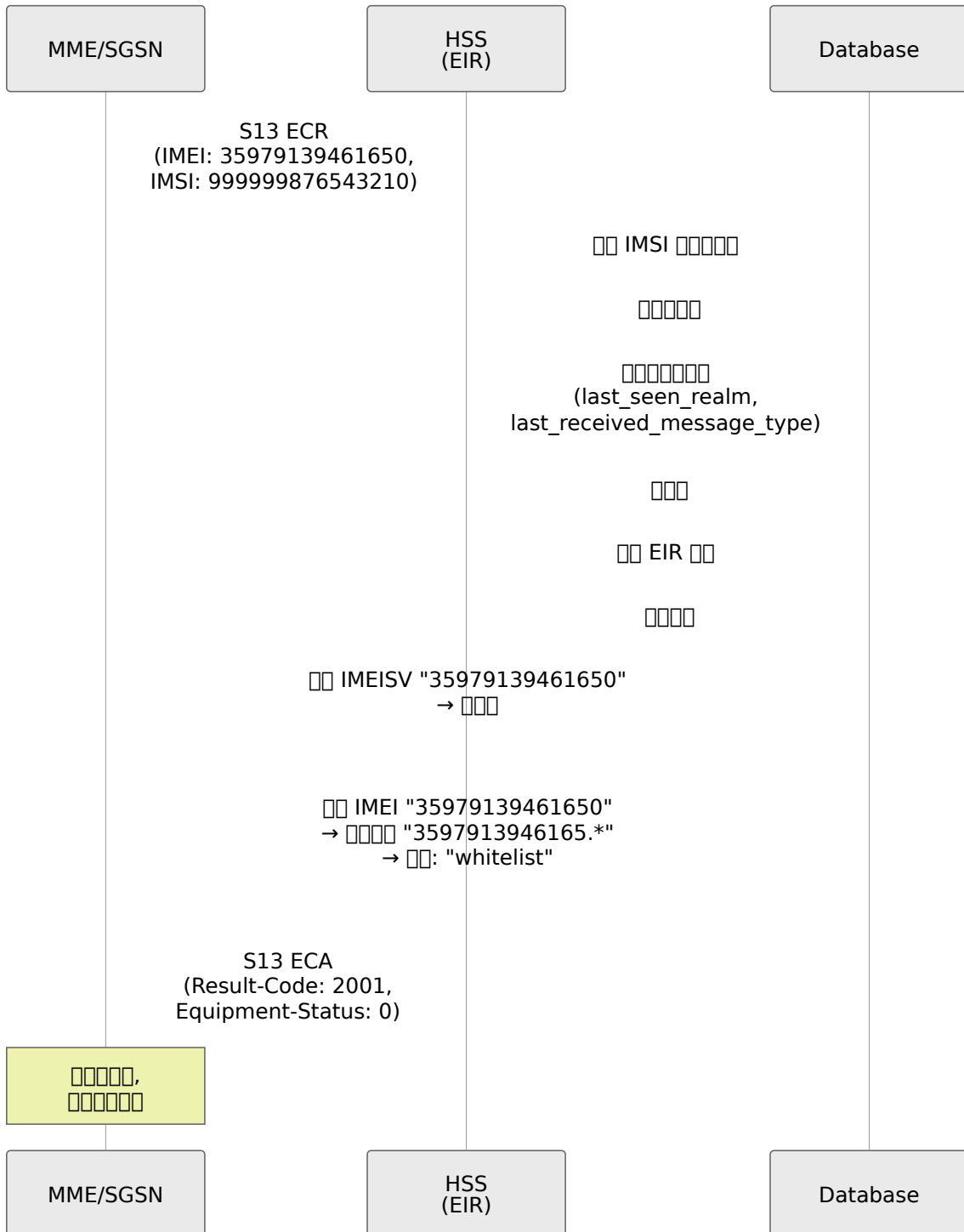
4.

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

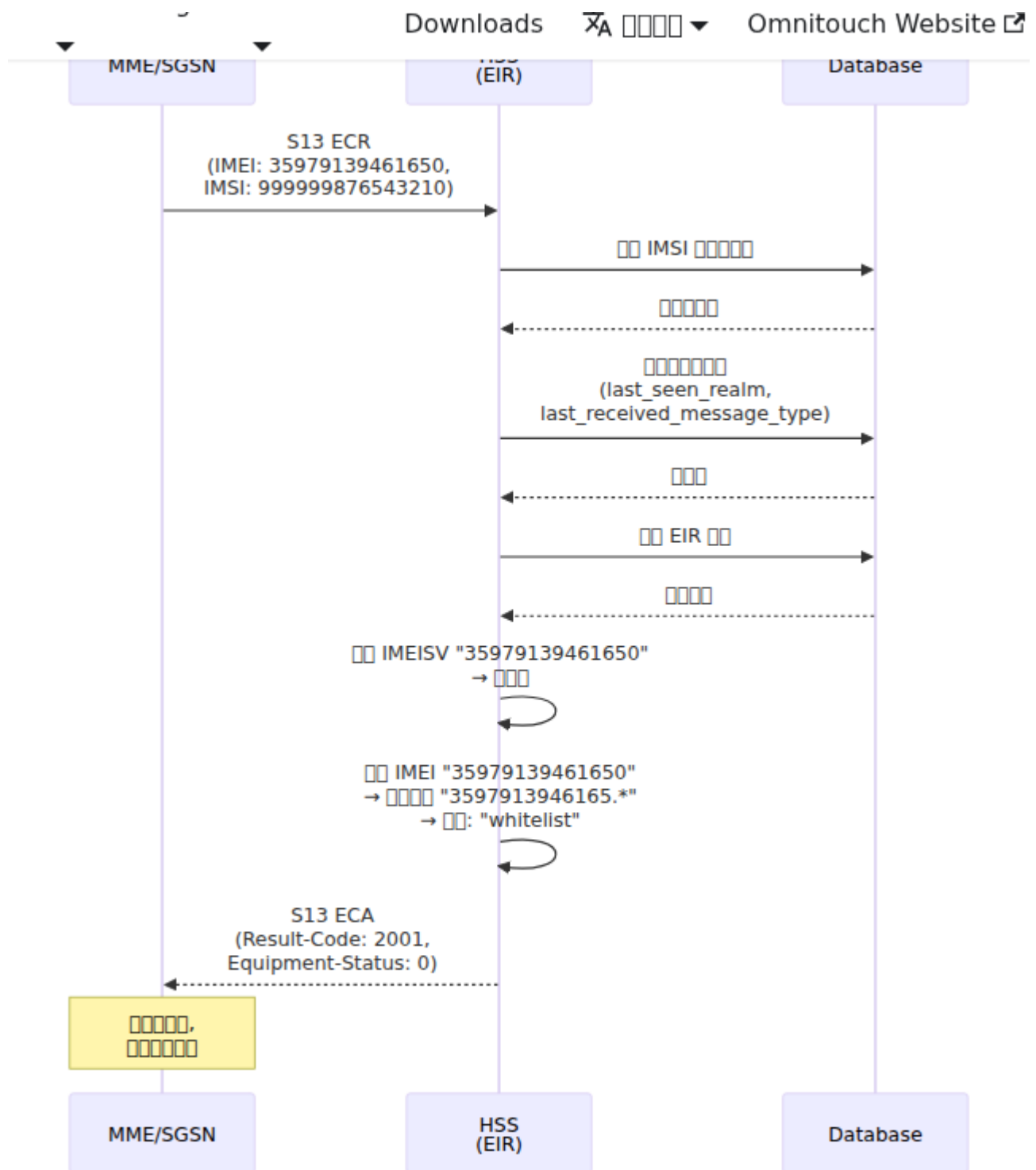
<code>"35979139461650"</code>	IMEI	/
<code>"3597913946165.*"</code>	IMEI	/
<code>"3597913946161108"</code>	IMEISV	
<code>"999999876543210"</code>	IMSI	
<code>"359791.*"</code>	TAC	

□□□□□□

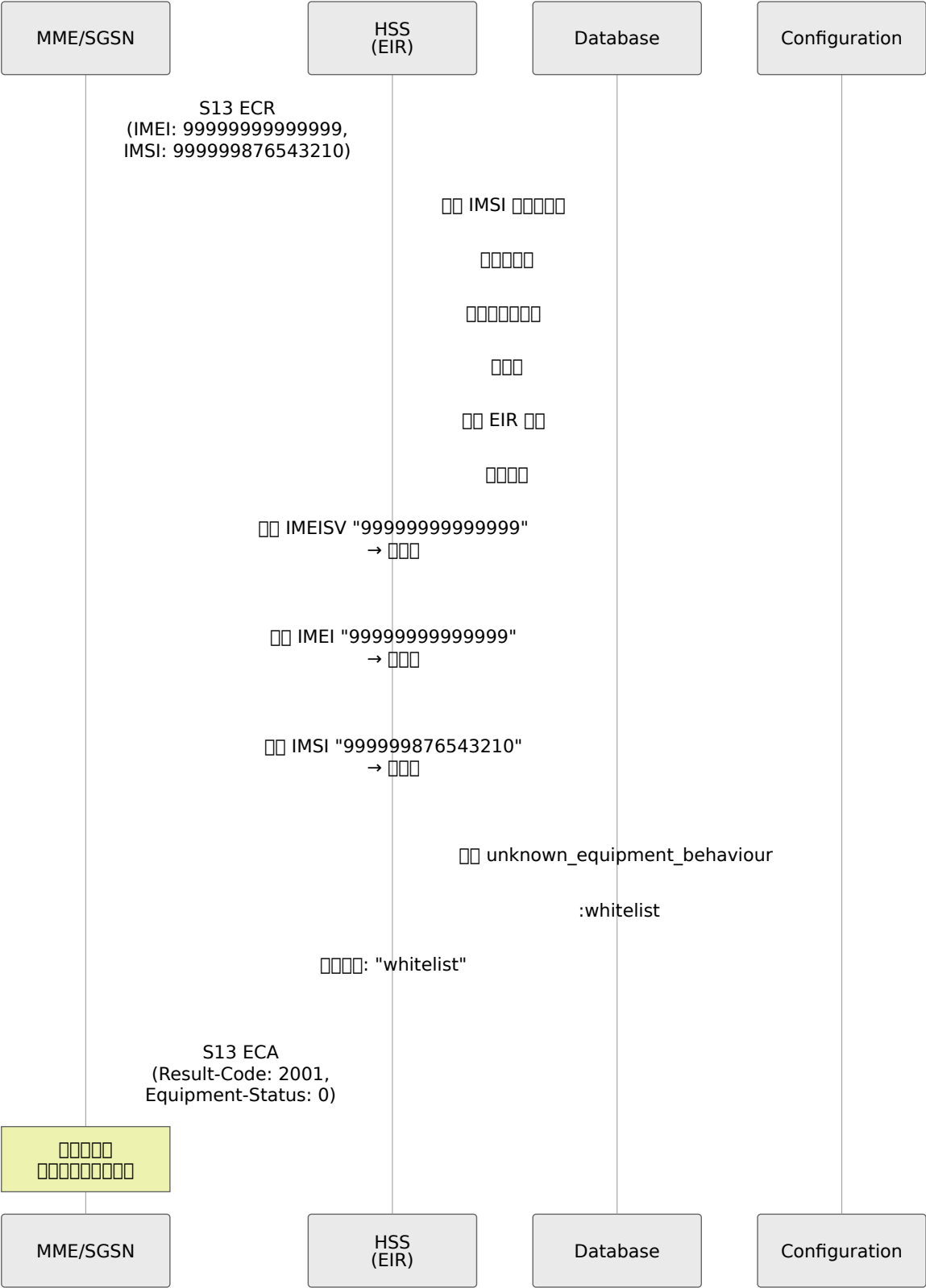
□□ 1□□□□□ - □□□□□ IMEI



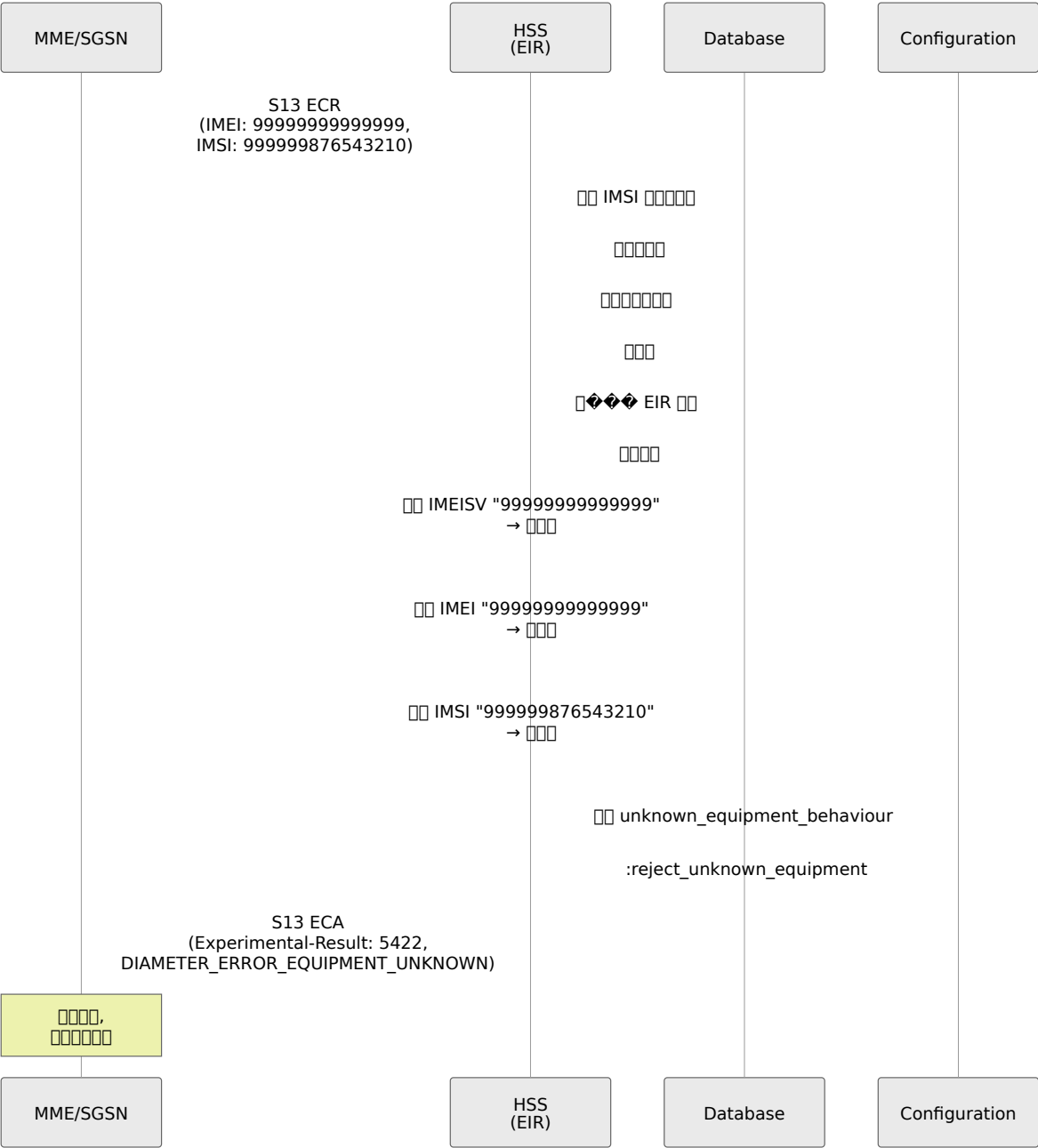
2. IMEI Whitelisting



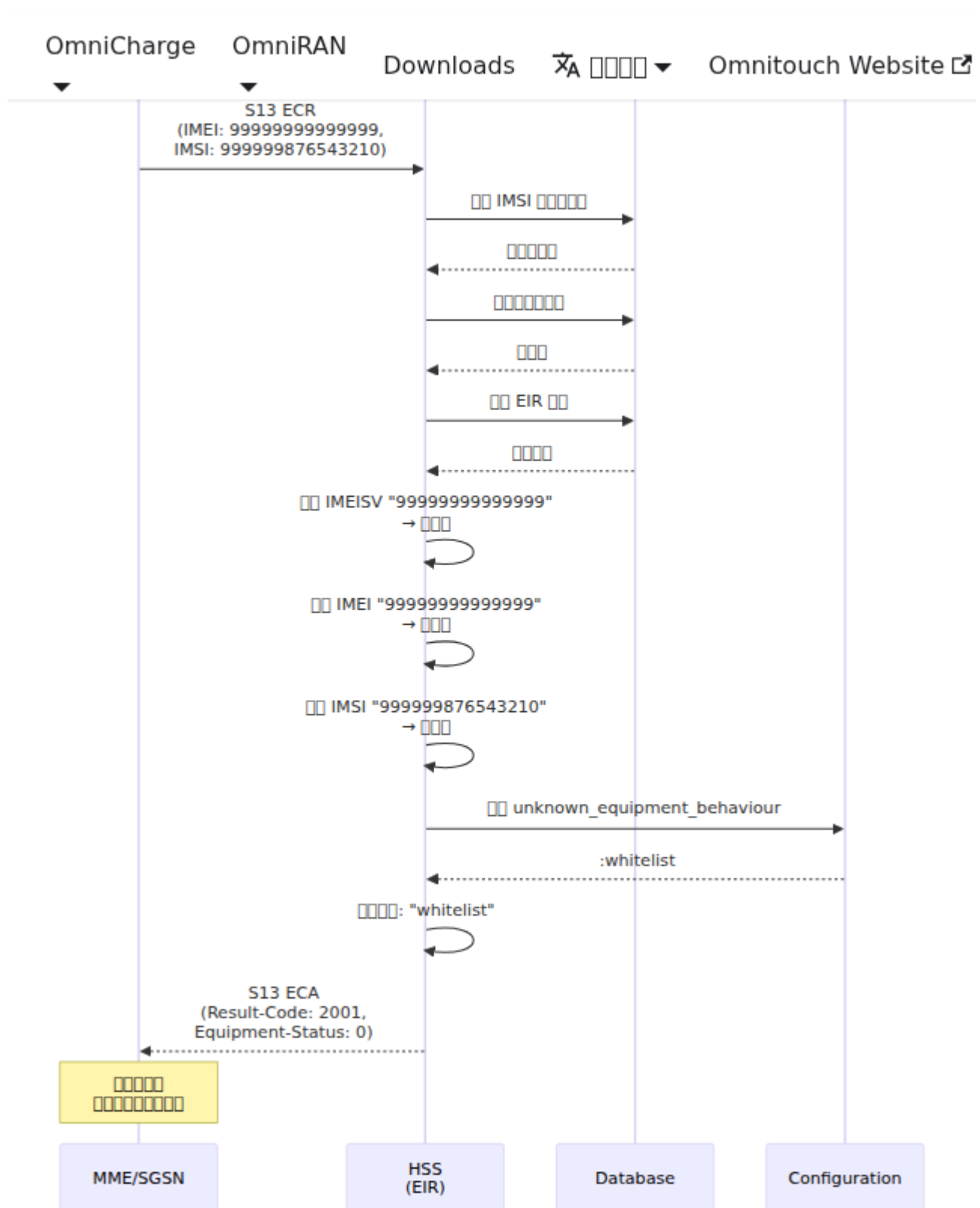
3 -



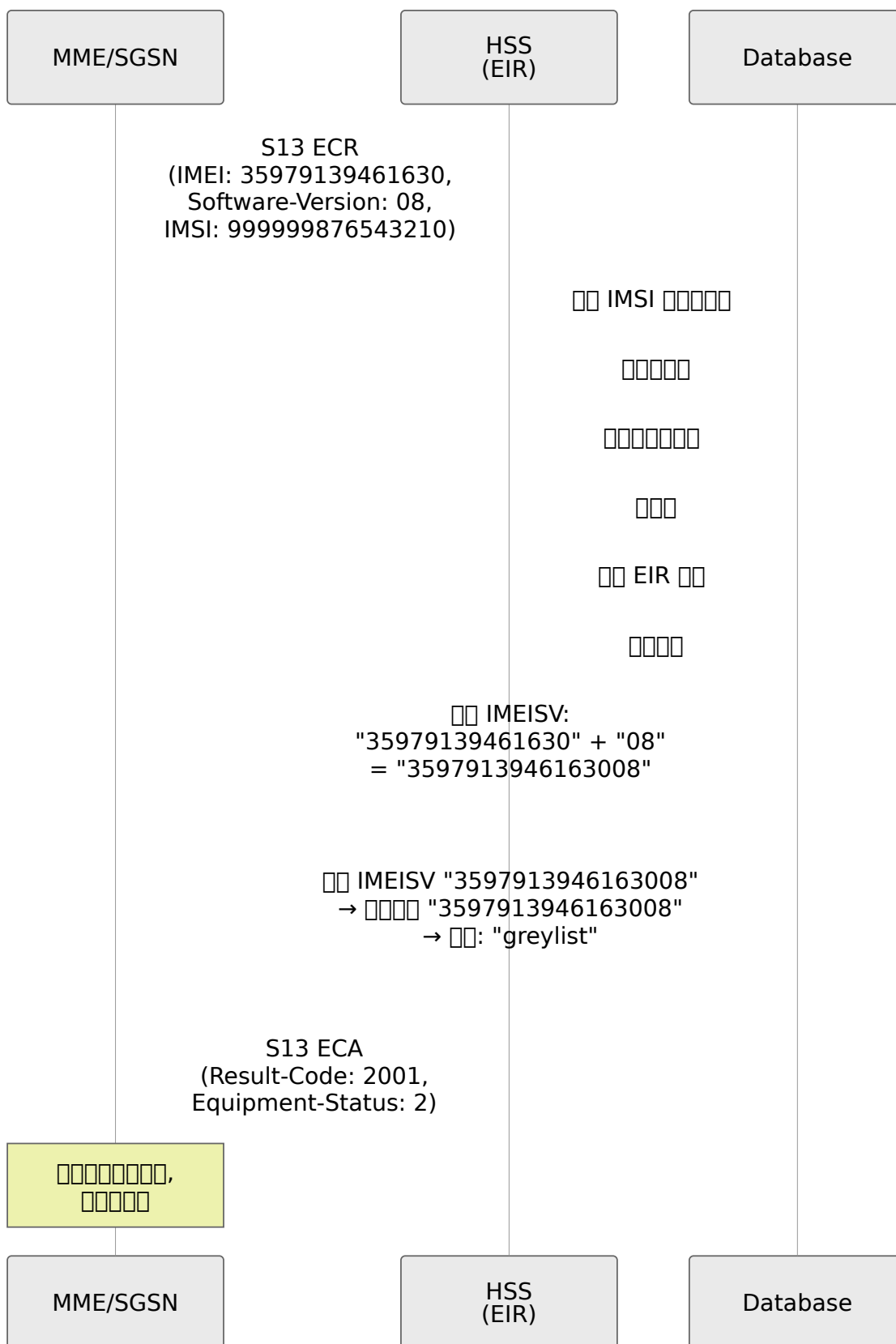
4 -



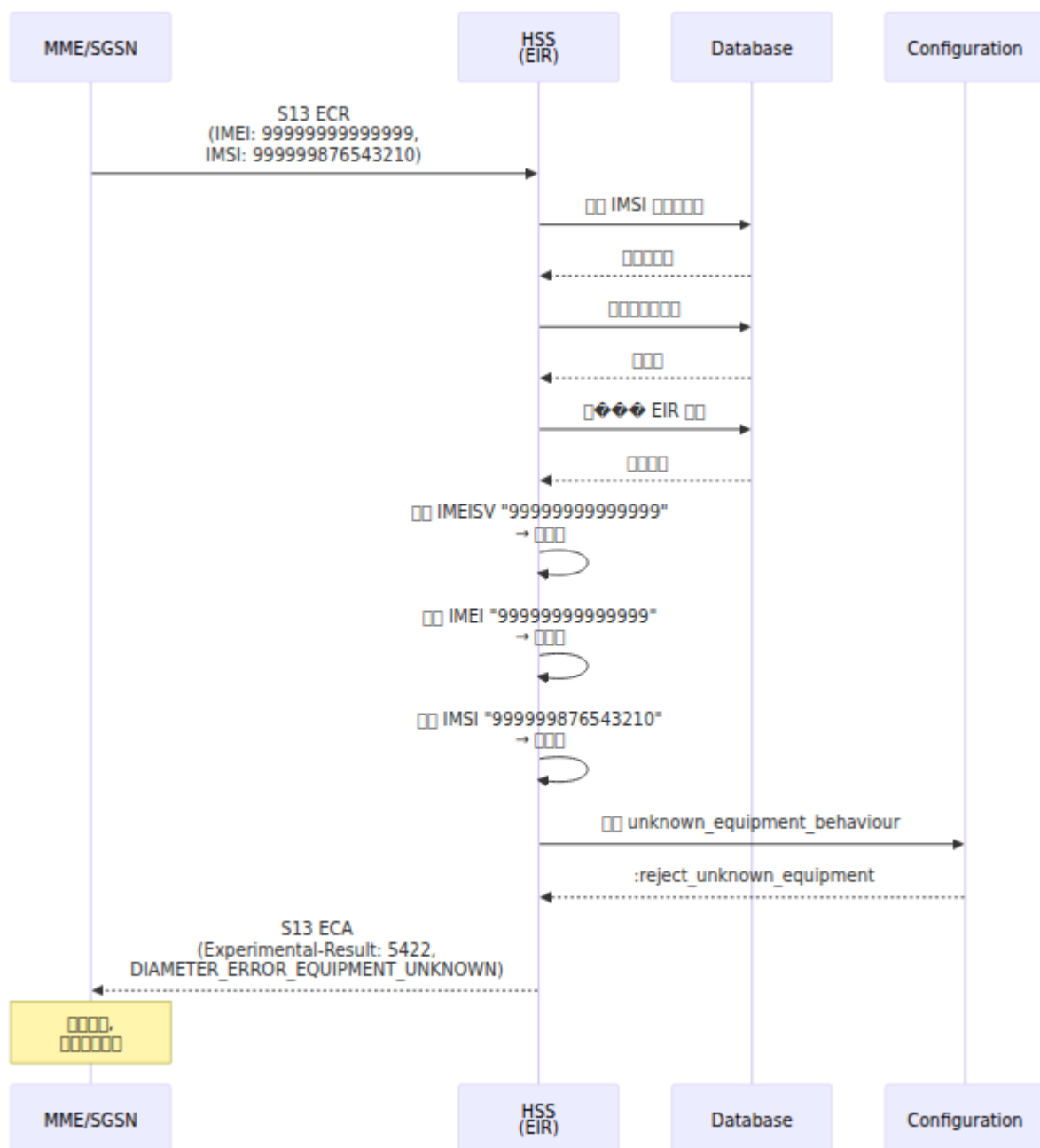
5 -



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	0000000000000000
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
}
```

黑名單 IMEI 後面加上 "05" 即可

測試

測試

EIR 測試工具

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

- **EIR** `imei_blacklist` - `imei_whitelist`
- **REST API** `imei_blacklist` - `imei_whitelist`

`imei_blacklist`

`imei_blacklist`

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

`imei_blacklist`

- `whitelist` - `imei_blacklist`
- `blacklist` - `imei_blacklist`
- `greylist` - `imei_blacklist`
- `reject_unknown_equipment` - `imei_blacklist`

`imei_blacklist`

IMEI `imei_blacklist`

IMEI `imei_blacklist`EIR

- `imei_blacklist` IMEI `imei_blacklist`
- `imei_blacklist`
- `API` `imei_blacklist`

`imei_blacklist`

EIR `imei_blacklist` ID`imei_blacklist`

- `1`: `"359791.*"` `"whitelist"` (`imei_blacklist`)
- `2`: `"35979139461640"` `"blacklist"` (`imei_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 RBI00000000002 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

OmniHSS

←

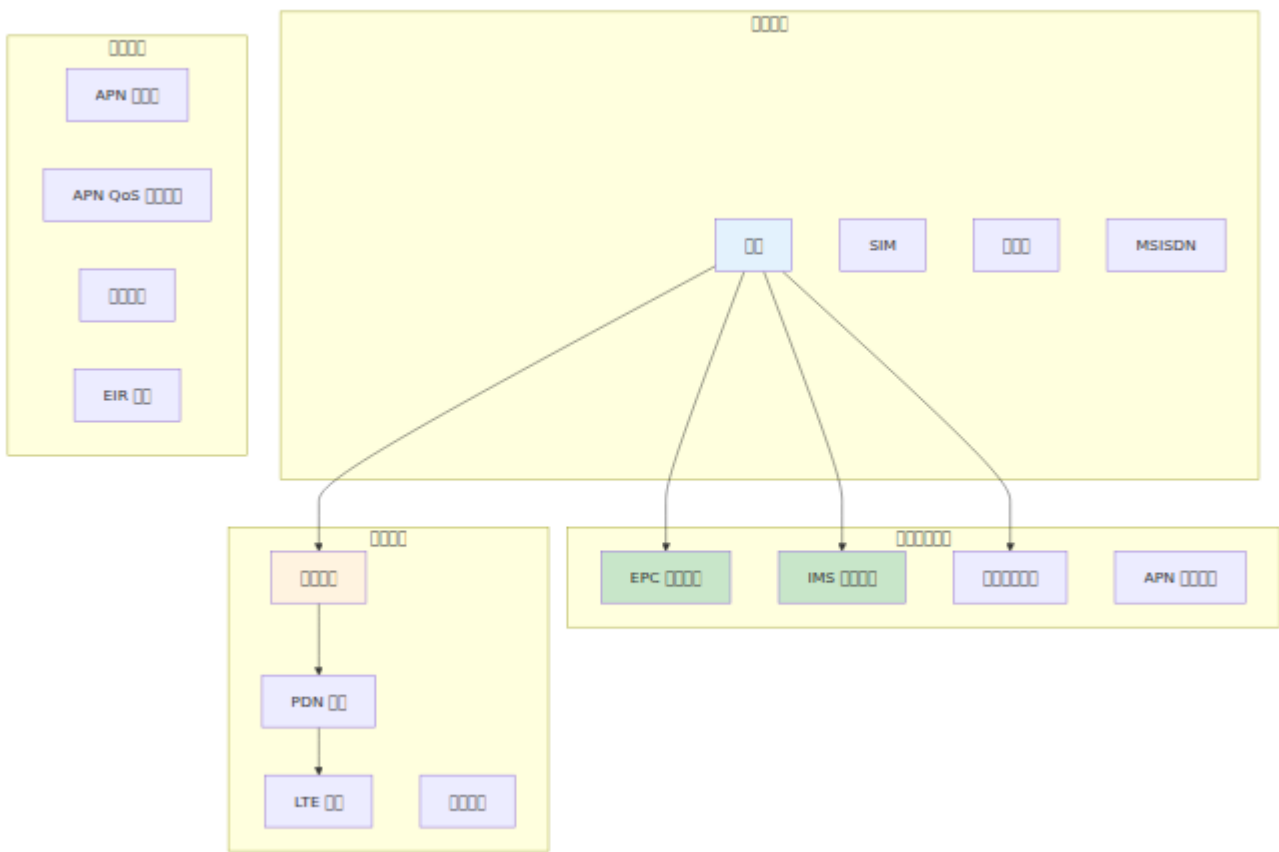
11

- 
- 
- 
- 
- 
- 
- 

444

[illegible]

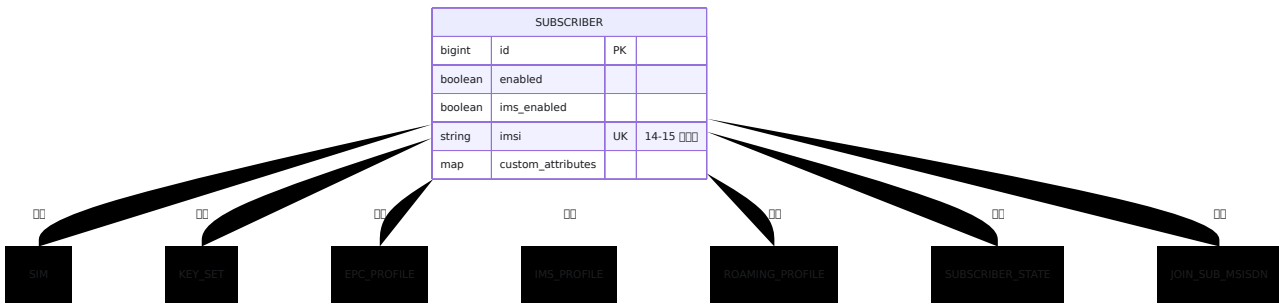
□□□□



□□□□

□□

□□□□□□□□□□□□□□



□□:

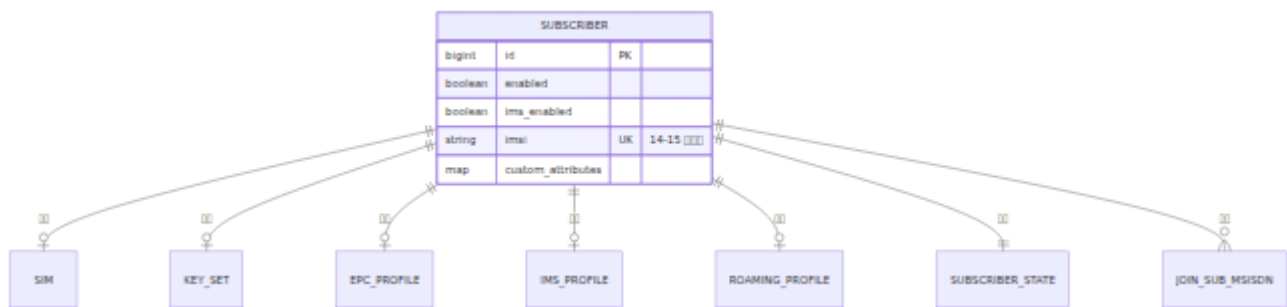
欄名	型	説明	例
id	bigint	ID	00000
enabled	boolean	IMS 有効フラグ	例: true
ims_enabled	boolean	IMS 有効フラグ	例: true
imsi	string	IMSI	14-15 桁の数字
custom_attributes	map	カスタム属性	例
sim_id	bigint	SIM ID	例
key_set_id	bigint	キーセット ID	例
epc_profile_id	bigint	EPC プロファイル ID	例
ims_profile_id	bigint	IMS プロファイル ID	例
roaming_profile_id	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	是否必填
iccid	string	ICCID ID	是
sim_vendor	string	SIM 厂商	是
batch_name	string	批次名	是
is_esim	boolean	是否是 eSIM	是
pin1, pin2	string	PIN 码	是
puk1, puk2	string	PUK 码	是
adm1 - adm10	string	ADM 码	是
kic, kid	binary	OTA 密钥	是

说明:

- ICCID 是 SIM 卡
- SIM 卡唯一标识符
- PIN/PUK 是 SIM 卡
- ADM 是 SIM 卡
- KIC/KID 是 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

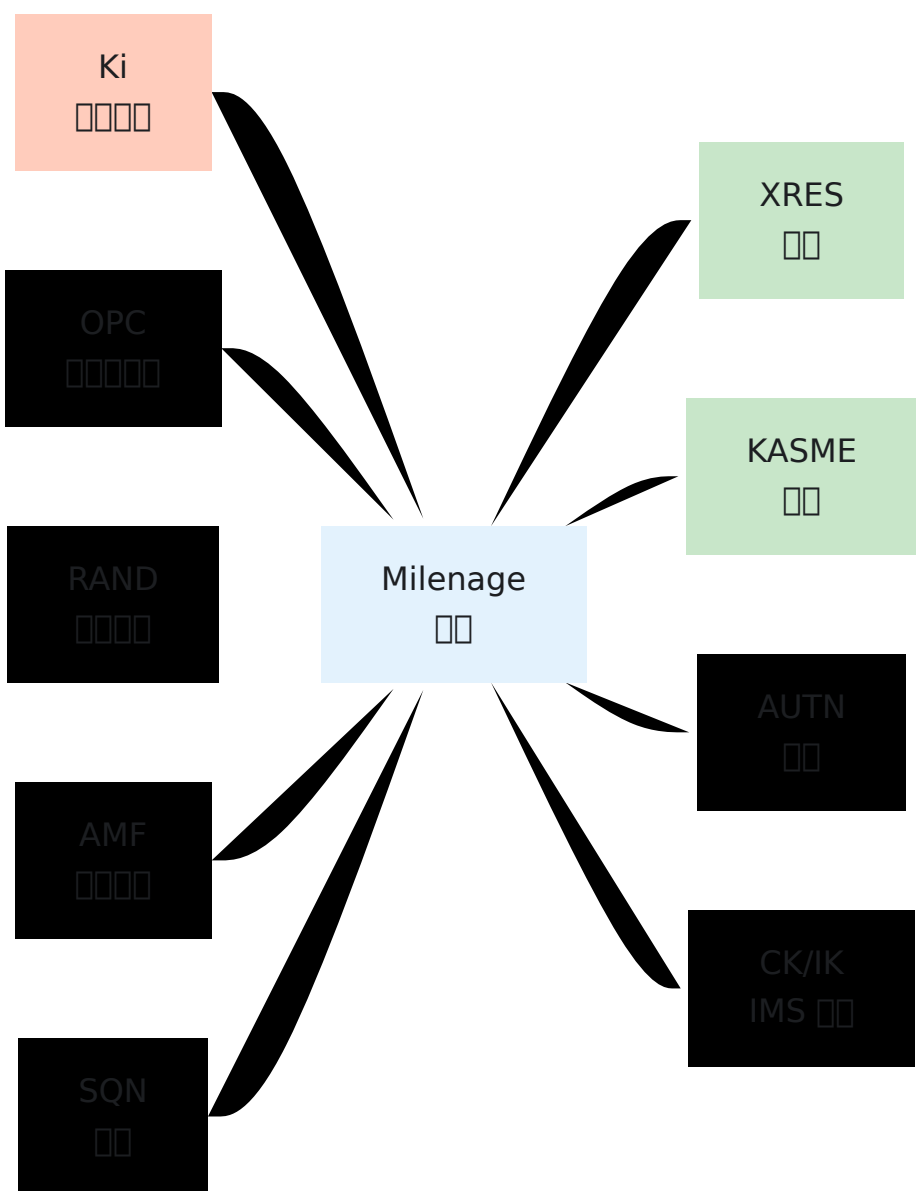
00:

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	認証モード	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:

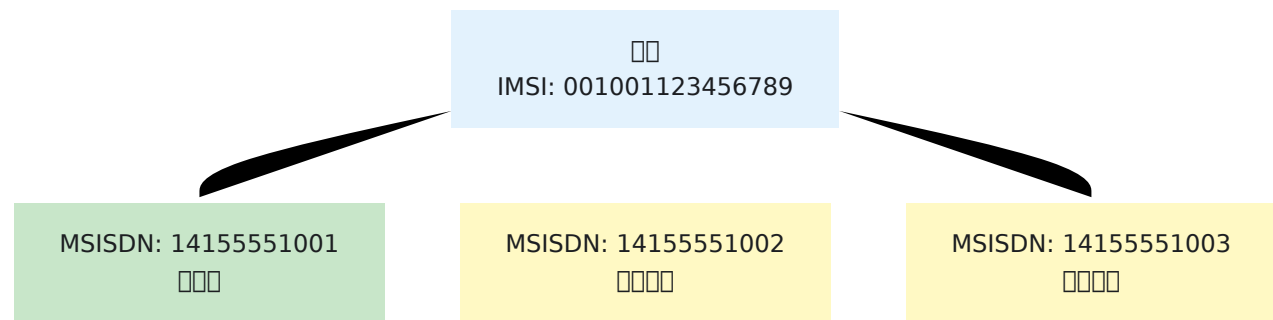
- 秘密鍵
- Ki SIM 秘密鍵
- オペレーションコード OP オペレーションコード OP
- SQN シーケンス番号
- Milenage 認証アルゴリズム

項目:



MSISDN

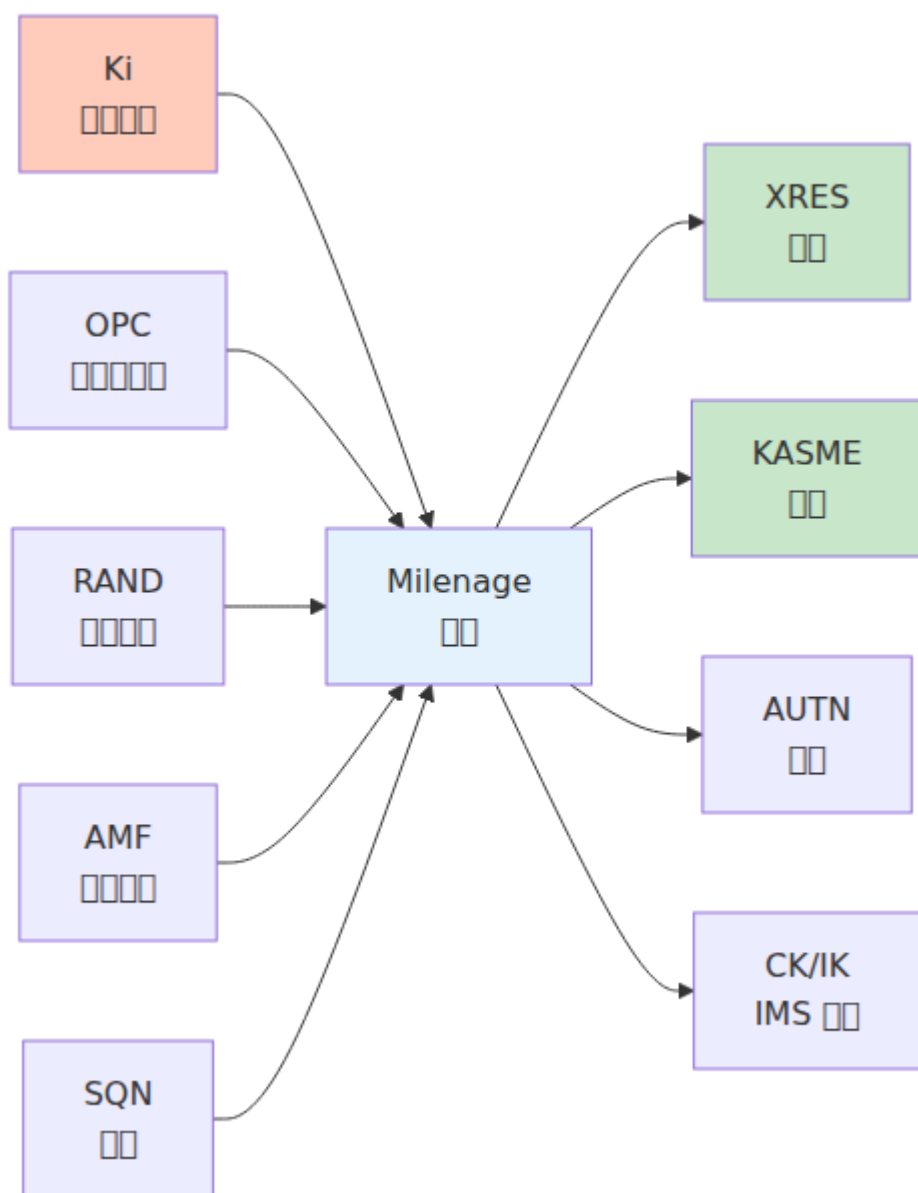
MSISDN 0000000000



□□□□□□

EPC □□□□

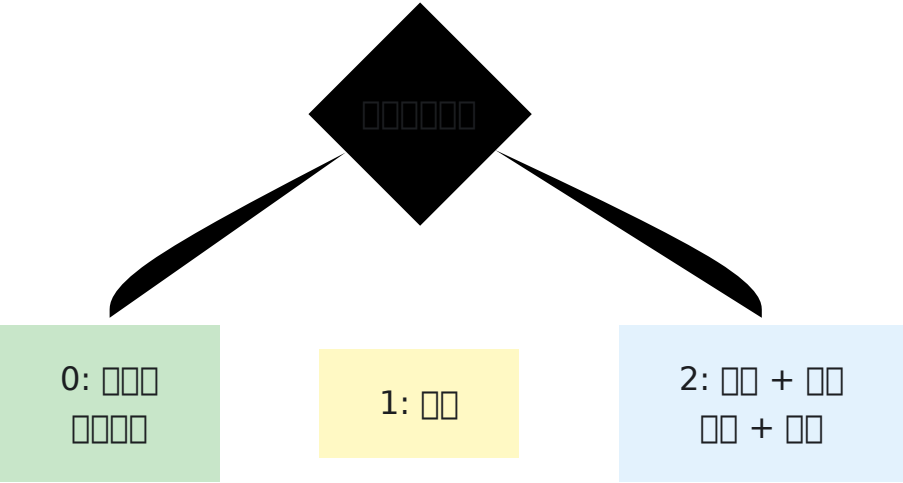
EPC □□□□□□ LTE □□□□□□□□



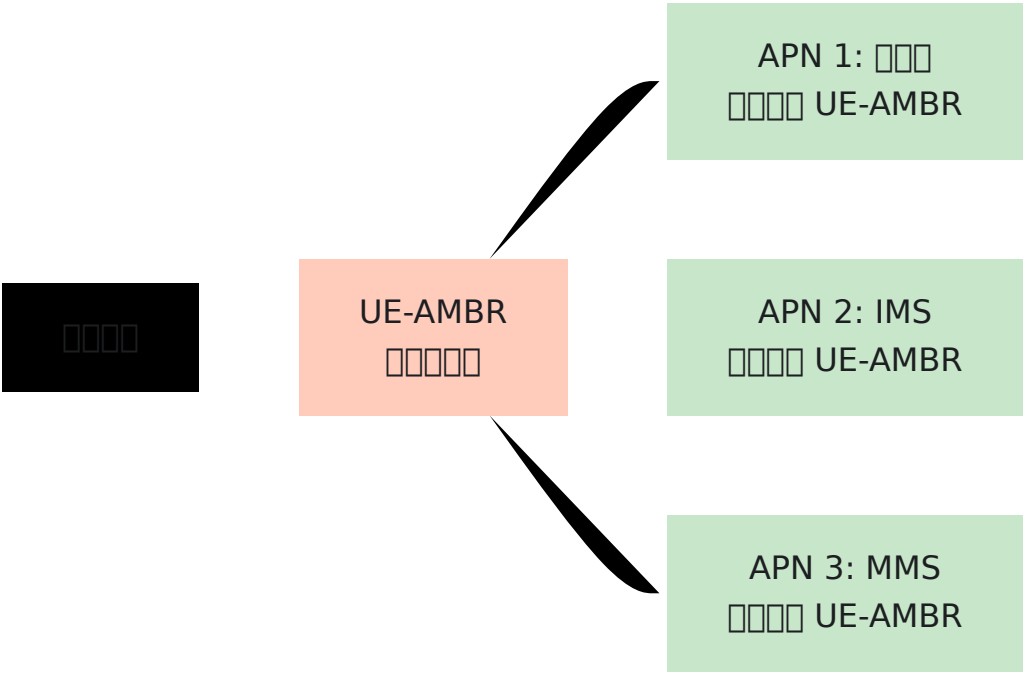
□□:

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

説明:

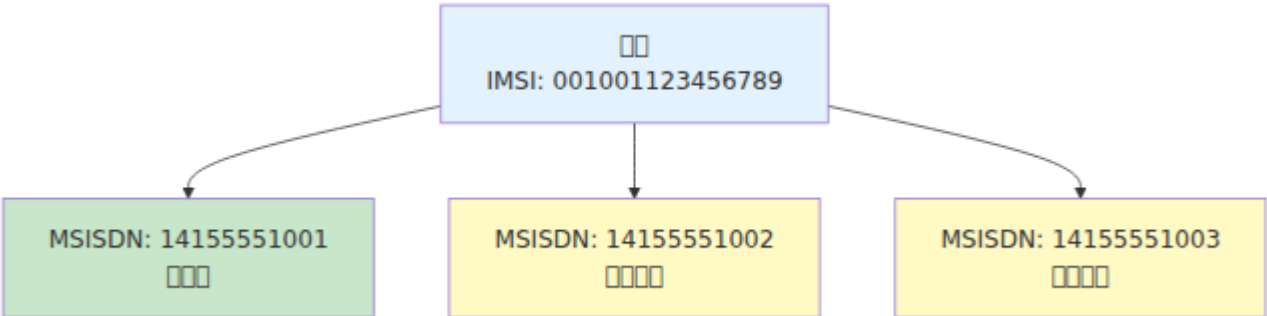


AMBR説明:



IMS

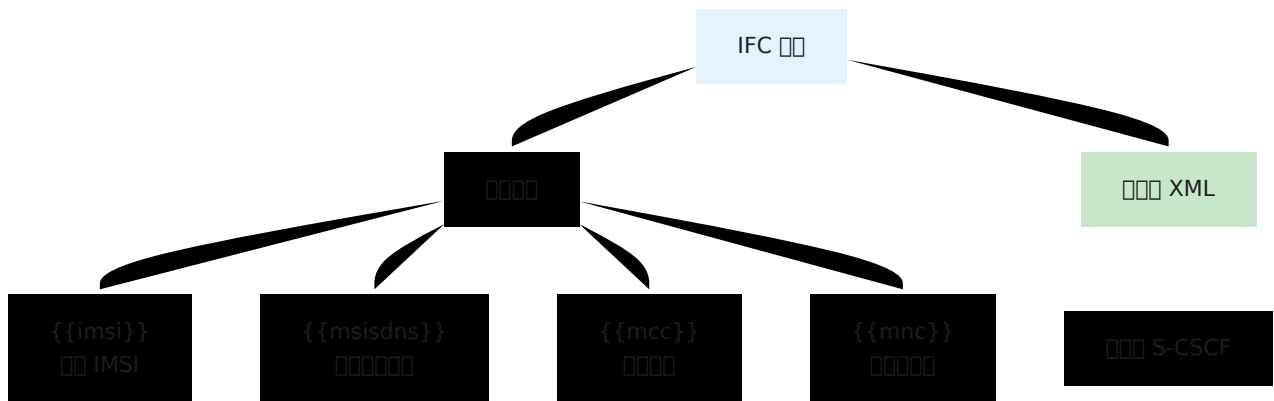
IMS



IMS:

name	string		
ifc_template	text	XML	XML

IFC

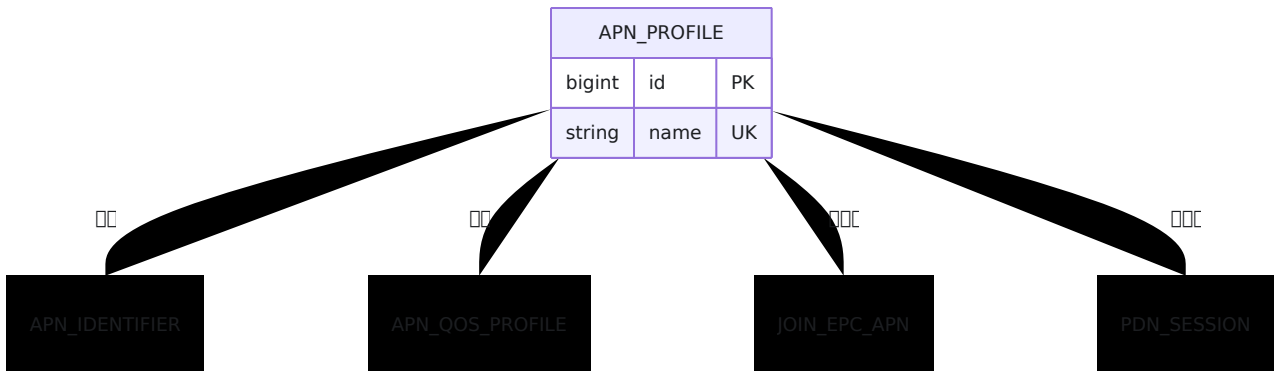


IFC:

- 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

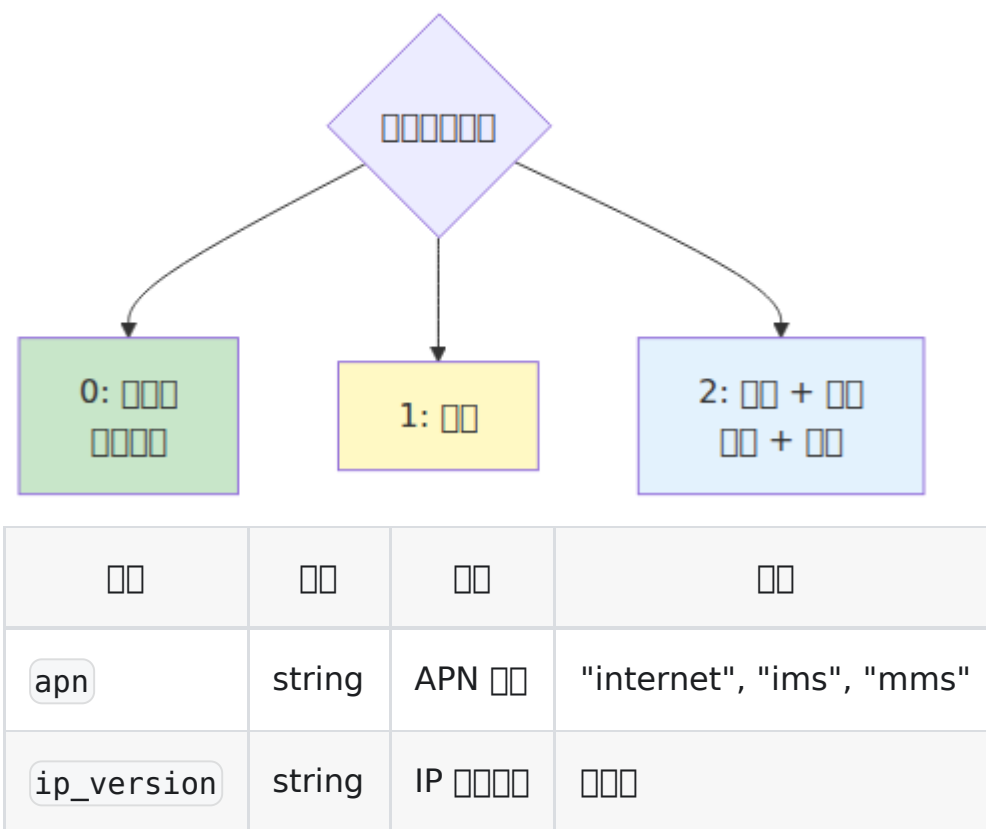
APN

APN is the Access Point Name

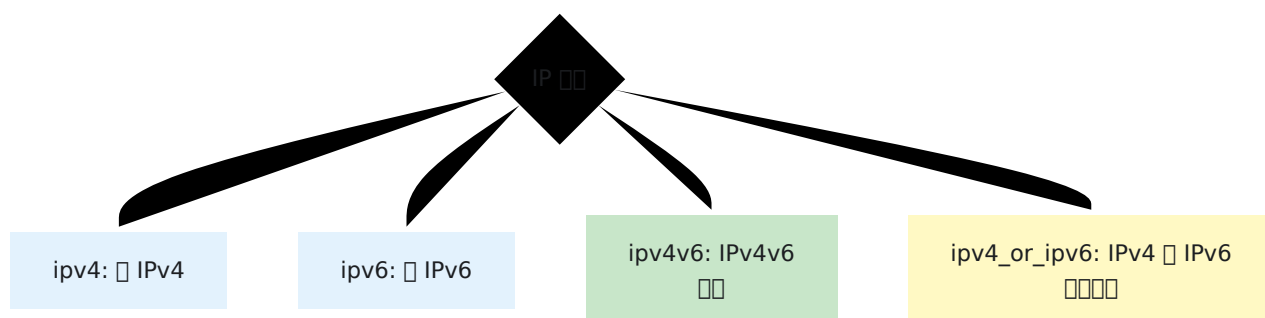


APN:

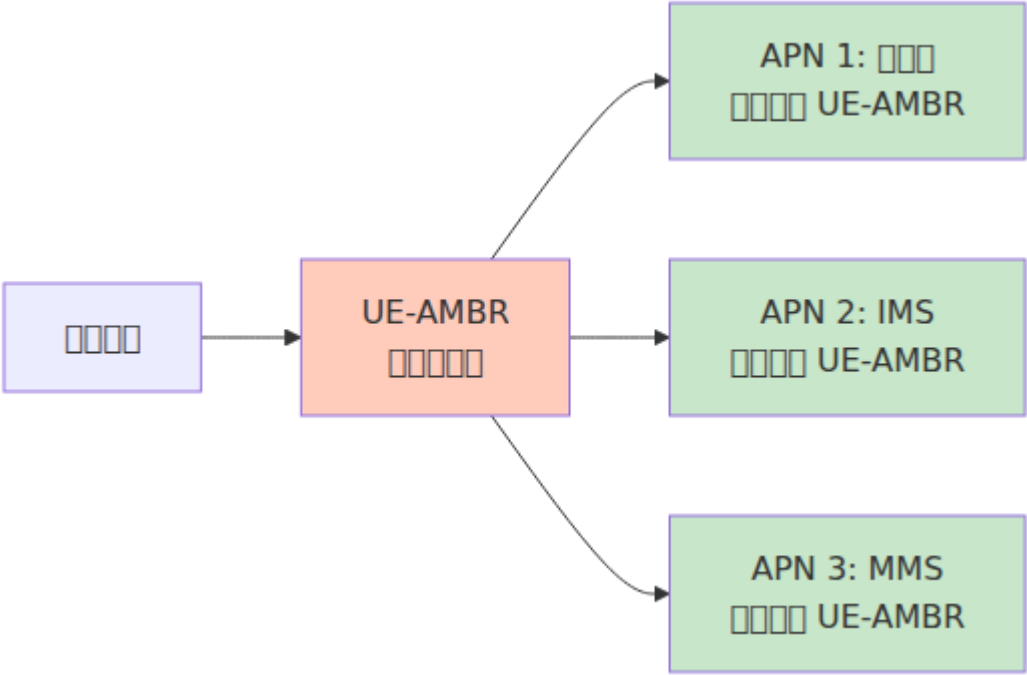
APN is the Access Point Name



IP 版本:



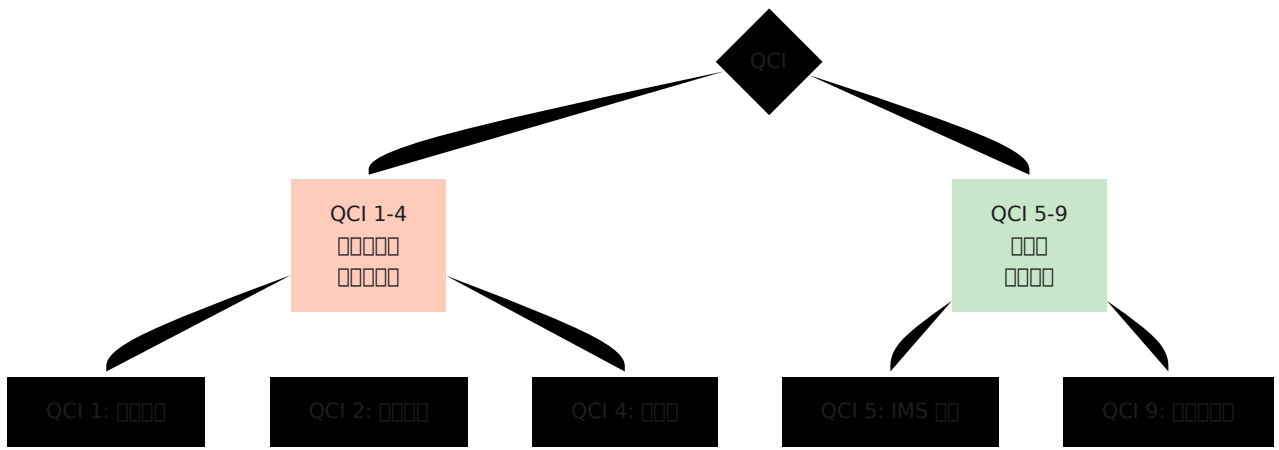
APN QoS 名称



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 :



0000000

000000000000000000000000

ROAMING_PROFILE			
bigint	id	PK	
string	name	UK	
string	data_action_if_no_rules_match		00000
string	ims_action_if_no_rules_match		00000

00

000

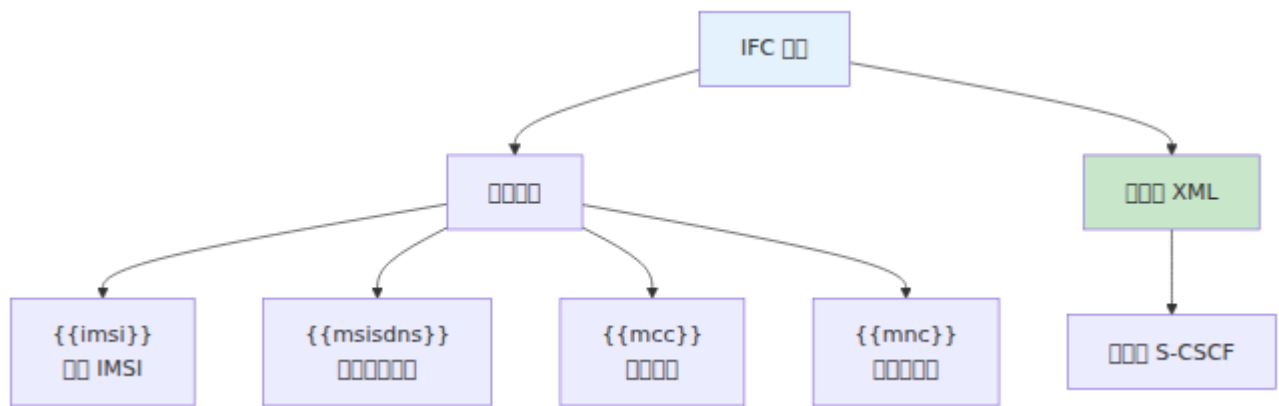
JOIN_ROAM_RULE

SUBSCRIBER

00

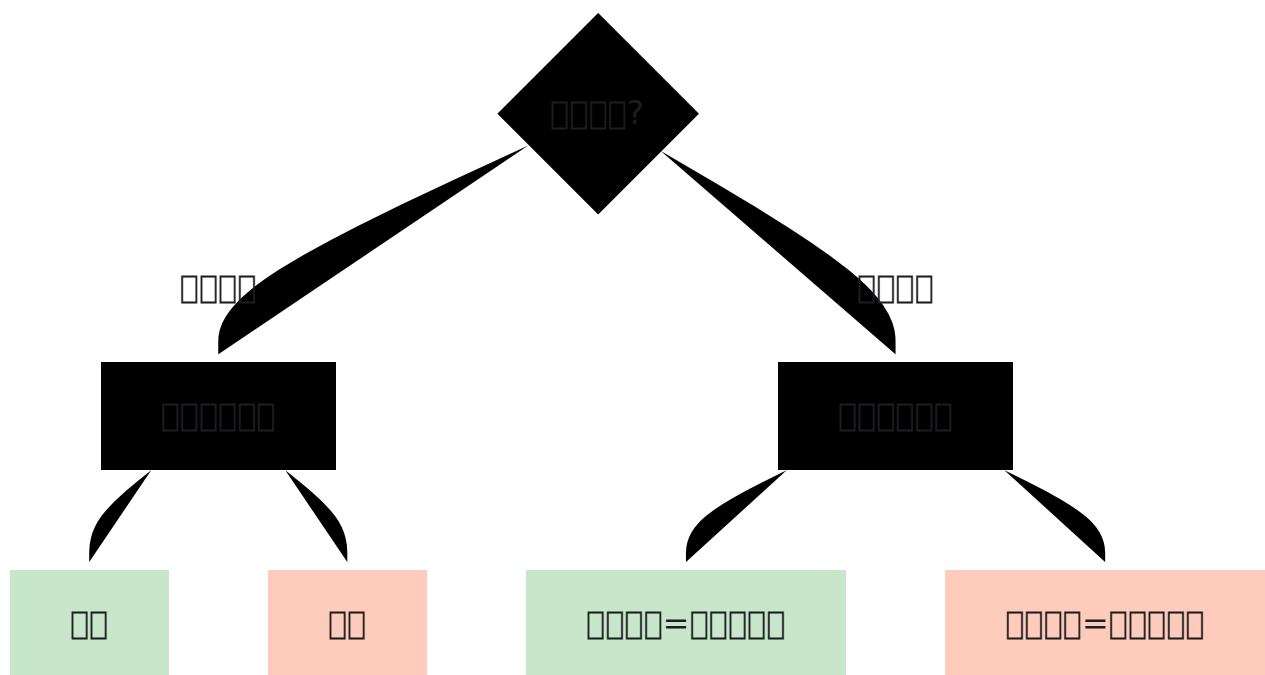
ROAMING_RULE

网络:



网络:

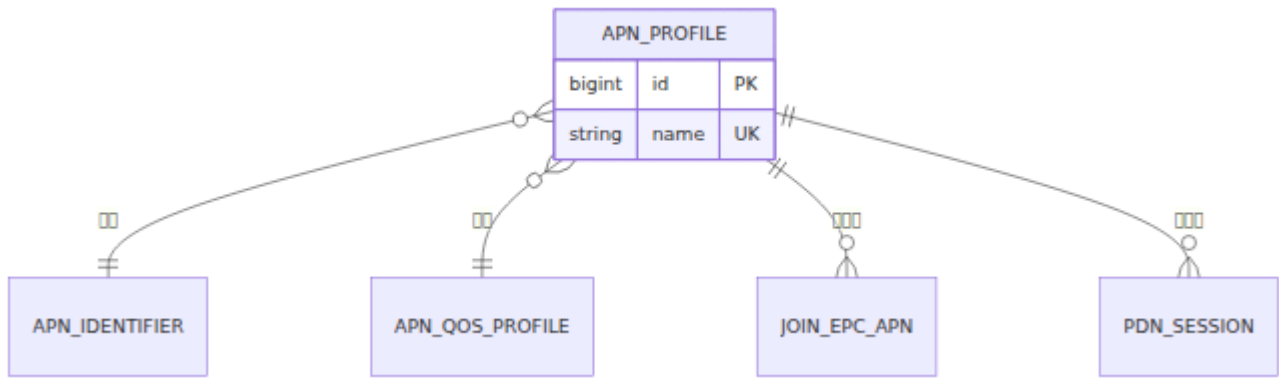
网络
MCC: 310, MNC: 410



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
- last_seen_realm - MME
- last_seen_rat_type - LTE5G

IMS:

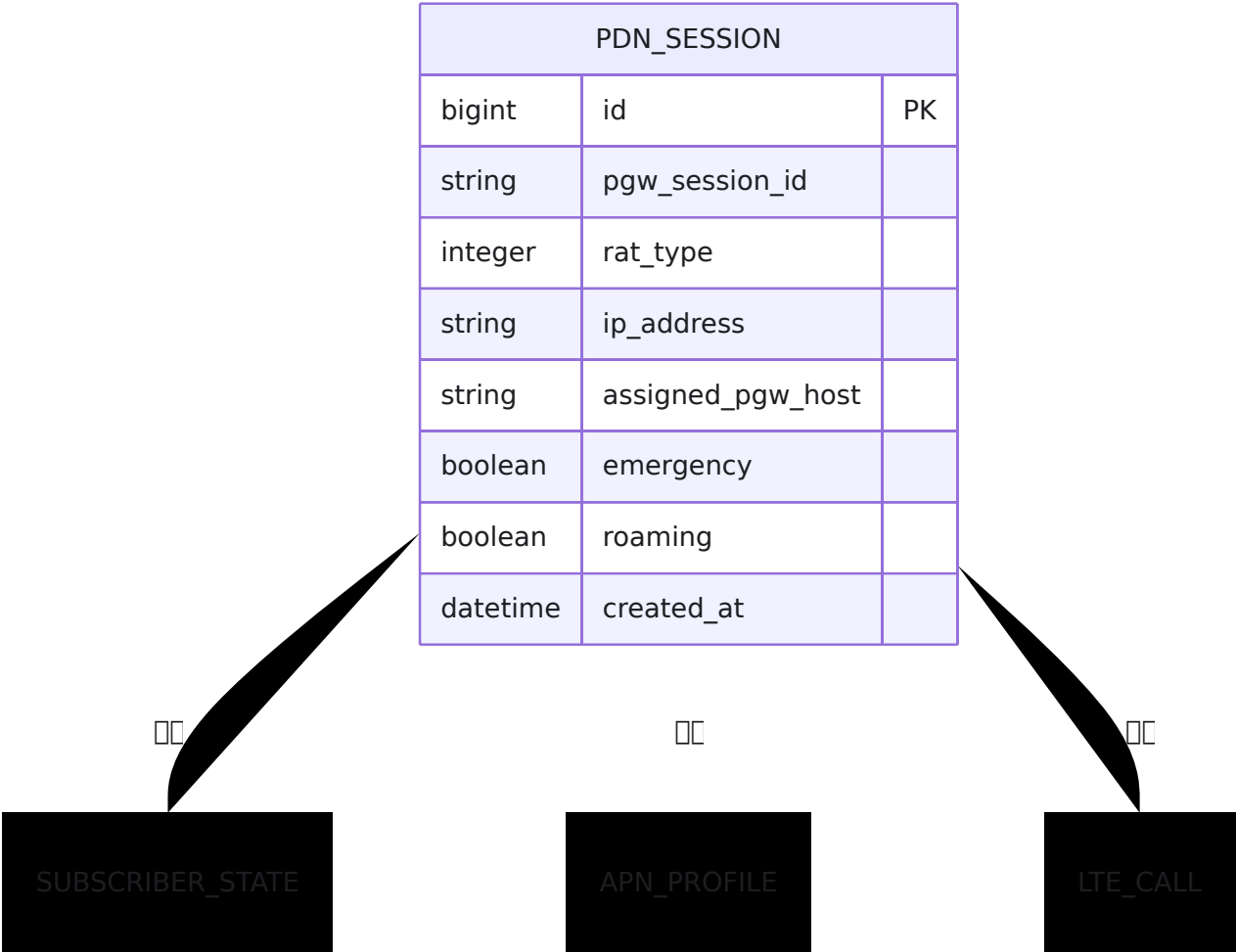
- assigned_scscf - S-CSCF
- ims_public_identity - SIP URI
sip:+14155551234@ims.example.com
- sh_repository_data - IMS

数据库:

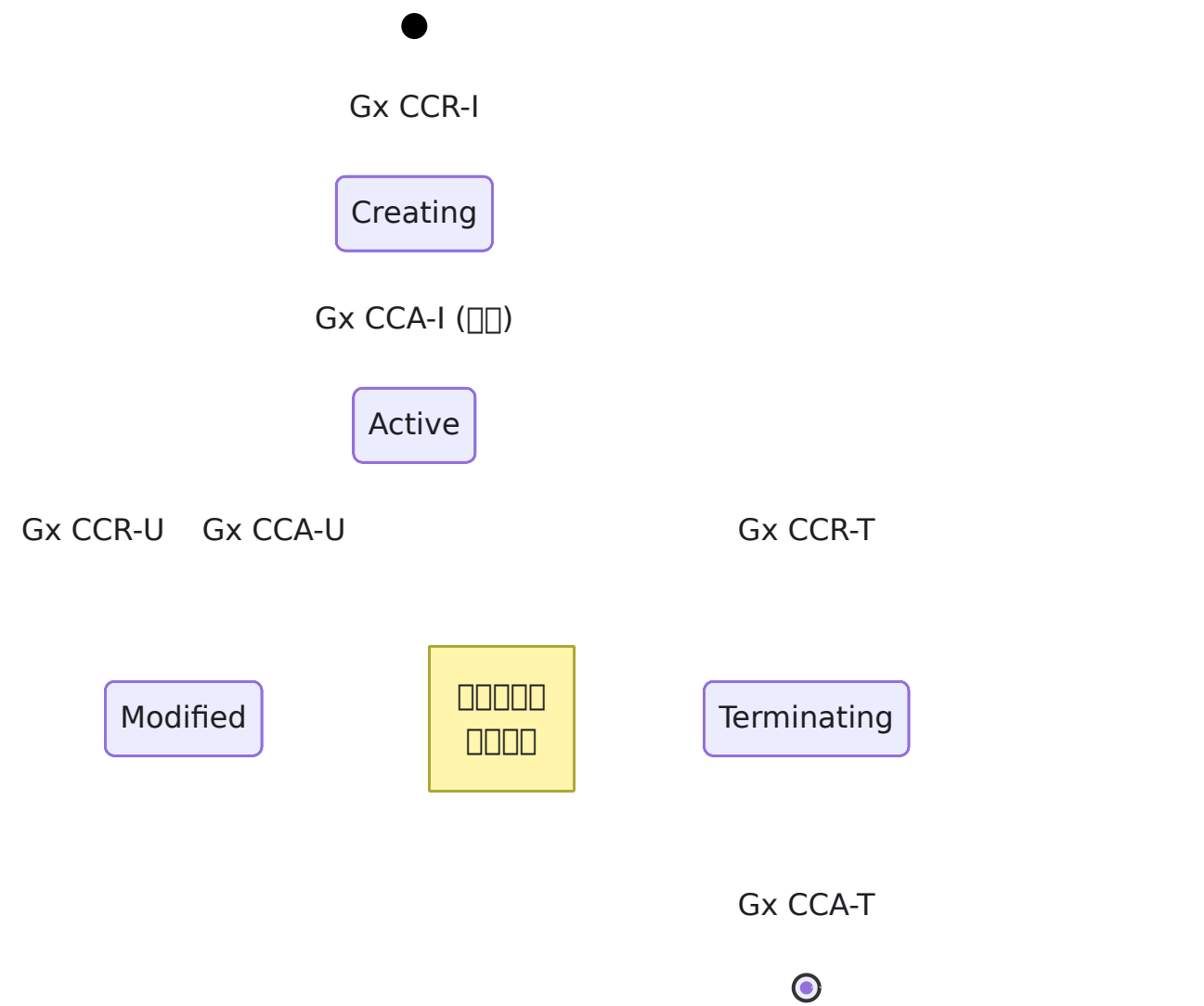
- last_seen_at - 000000000000
- last*_at 000000000000

PDN

PDN 0000000000000000



PDN 000000:



LTE

LTE VoLTE

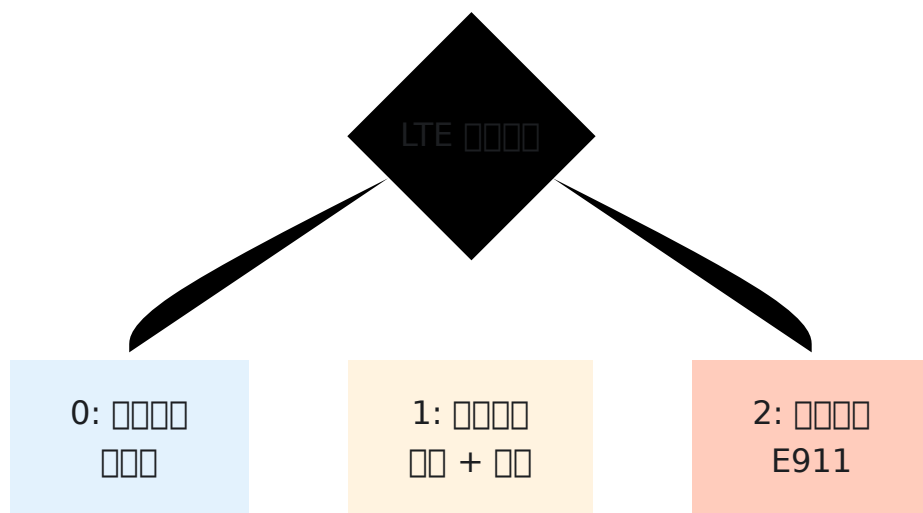
instant pricing estimates, or contact us for a tailored solution to your needs.

Dimensioning Tool

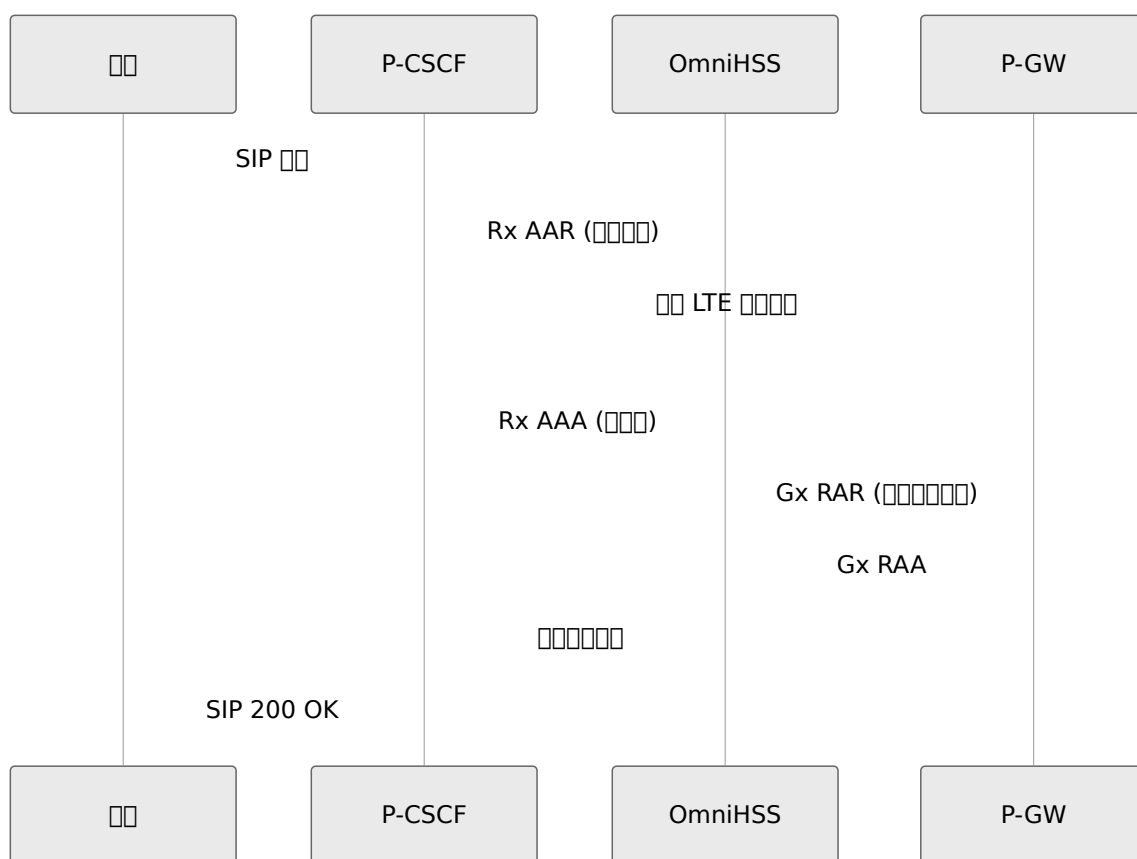
Pricing Calculator

v6: IPv4 IPv6

ip:



VoLTE Network:



□□□□□

□□□□□

Q

Use our free tools to dimension your network and get instant pricing estimates, or contact us for a tailored solution to your needs.

Dimensioning Tool

Pricing Calculator

QCI 9: □□□□□

□□□□

□□□□□□□□□□□□□□□□





Plan Your Mobile Network

Use our free tools to dimension your network and get instant pricing estimates, or contact us for a tailored solution to your needs.

Dimensioning Tool

Pricing Calculator

Get in Touch

ROAMING_RULE

□□□□□□

□□□□□□□□



□□□□□□

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 □□
□□□□



□□□□□

□□□□□

□□□□□

×

Plan Your Mobile Network

Use our free tools to dimension your network and get instant pricing estimates, or contact us for a tailored solution to your needs.

Dimensioning Tool

Pricing Calculator

Get in Touch

□□

□□

□□□□=□□□□□

□□□□=□□□□□

□□□□□□

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

□□□□□□

PDN_SESSION

Plan Your Mobile Network

Use our free tools to dimension your network and get instant pricing estimates, or contact us for a tailored solution to your needs.

Dimensioning Tool

Pricing Calculator

Get in Touch

SUBSCRIBER_STATE

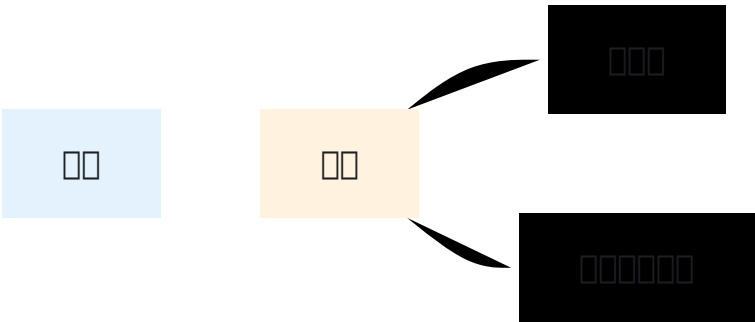
APN_PROFILE

LTE_CALL

□□□□□□

OmniHSS □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□



□□: S6a AIR - □□□□□□□□□□□□

[illegible]

Plan Your Mobile Network

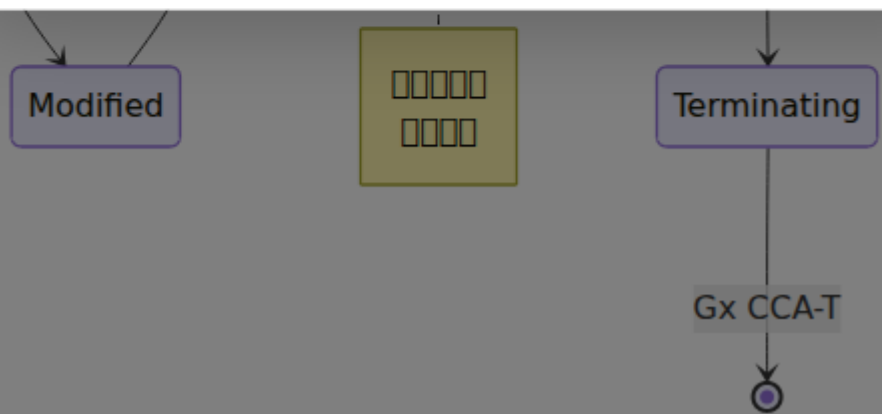
Use our free tools to dimension your network and get instant pricing estimates, or contact us for a tailored solution to your needs.

Dimensioning Tool

Pricing Calculator

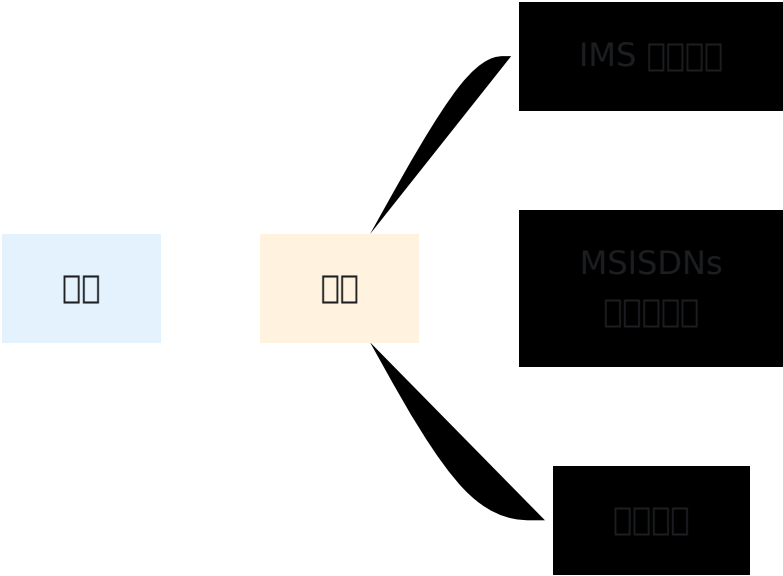
Get in Touch

Gx



□□: S6a ULR - □□□□□ EPC □□□□□□

IMS 数据



数据: Cx SAR - 数据 IMS 数据

← 数据 | 数据: API 数据 →

Diameter 参考リンク

← 参考リンク

参考リンク mermaid 参考リンク Diameter 参考リンク OmniHSS 参考リンク

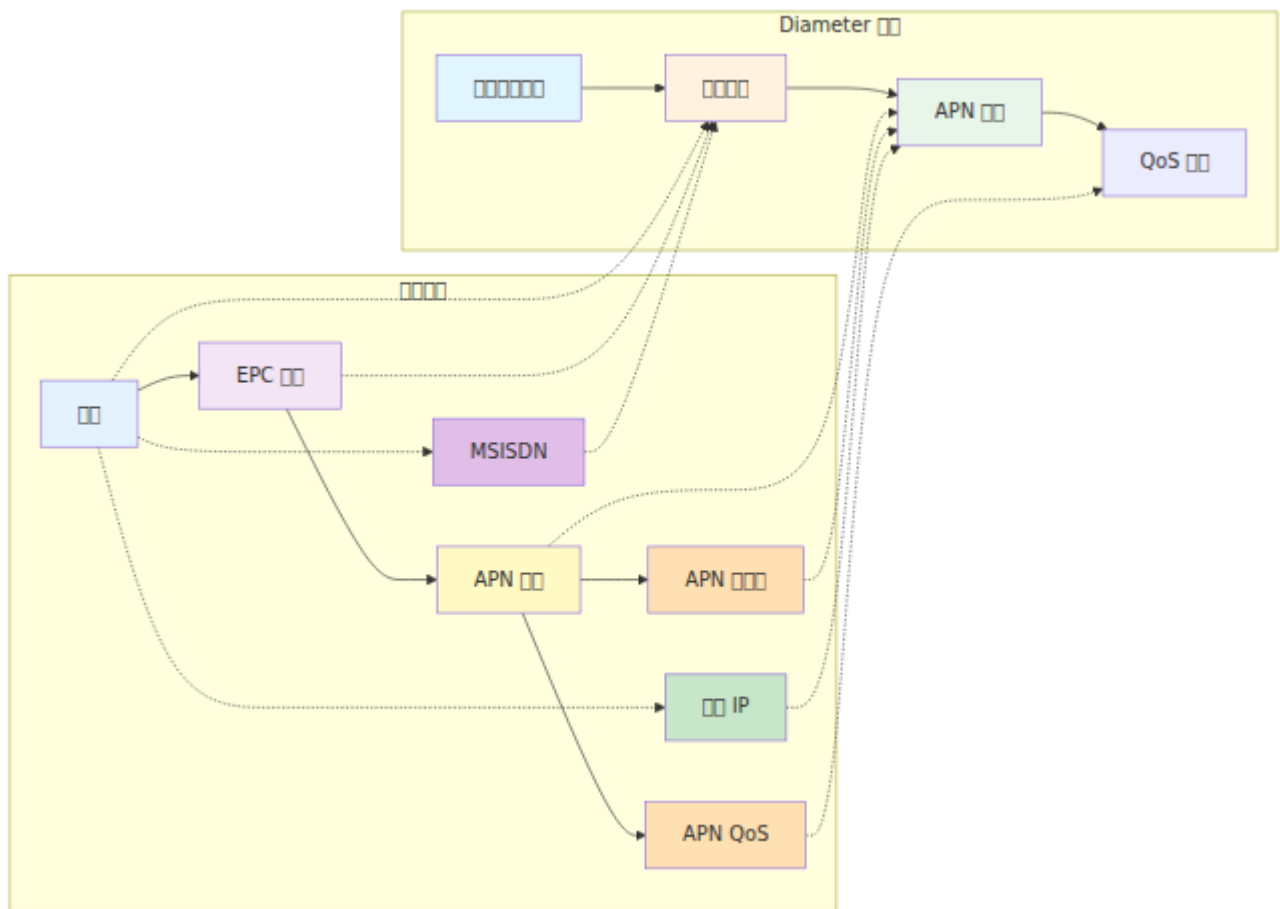
参考リンク

- 参考リンク (S6a ULA)
 - 参考リンク (S6a AIA)
 - 参考リンク (Cx SAA)
 - 参考リンク (Gx CCA)
 - 参考リンク (Sh UDA)
 - ME 参考リンク (S13 ECA)
-

参考リンク (S6a ULA)

参考リンク HSS 参考リンク LTE 参考リンク MME参考リンク Diameter AVP 参考リンク

□□□□□



□□□□□□

□□□□□	□□	□
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

- AMBR** 値は 10000 kbps から 100000 kbps の範囲で指定してください。
- IP** 値は 0=IPv4, 1=IPv6, 2=IPv4v6, 3=IPv4_or_IPv6 のいずれかを指定してください。
- enabled** 値は true → 0 (SERVICE_GRANTED) / false → 1 (OPERATOR_DETERMINED_BARRING) のいずれかを指定してください。
- 静的 IP アドレスは APN ごとに指定する必要があります。
- 静的 IP アドレスは static_ips 項目で指定する必要があります。

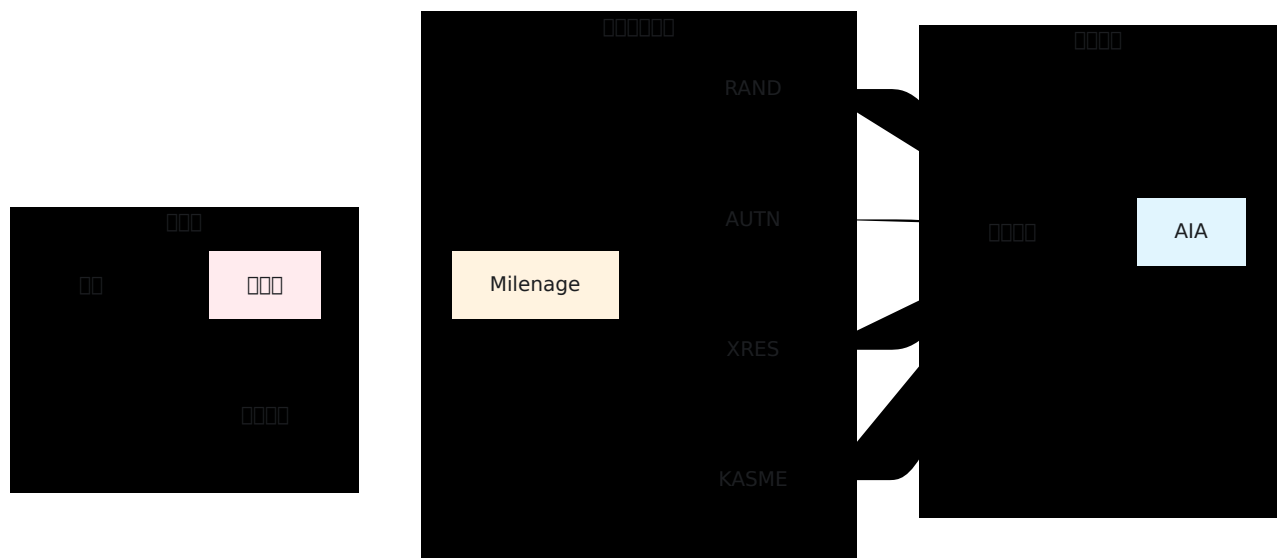
関連項目

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

网络选择策略 (S6a AIA)

网络选择策略 LTE/EPC 网络选择策略

网络选择策略



网络选择策略

1. 网络选择策略网络选择策略网络选择策略 `key_set` 网络选择策略
2. **SQL** 网络选择策略网络选择策略网络选择策略
3. 网络选择策略3GPP TS 35.206 - 网络选择策略
4. **KASME** 网络选择策略 TS 33.401 网络选择策略 KDF 网络选择策略 CK||IK 网络选择策略

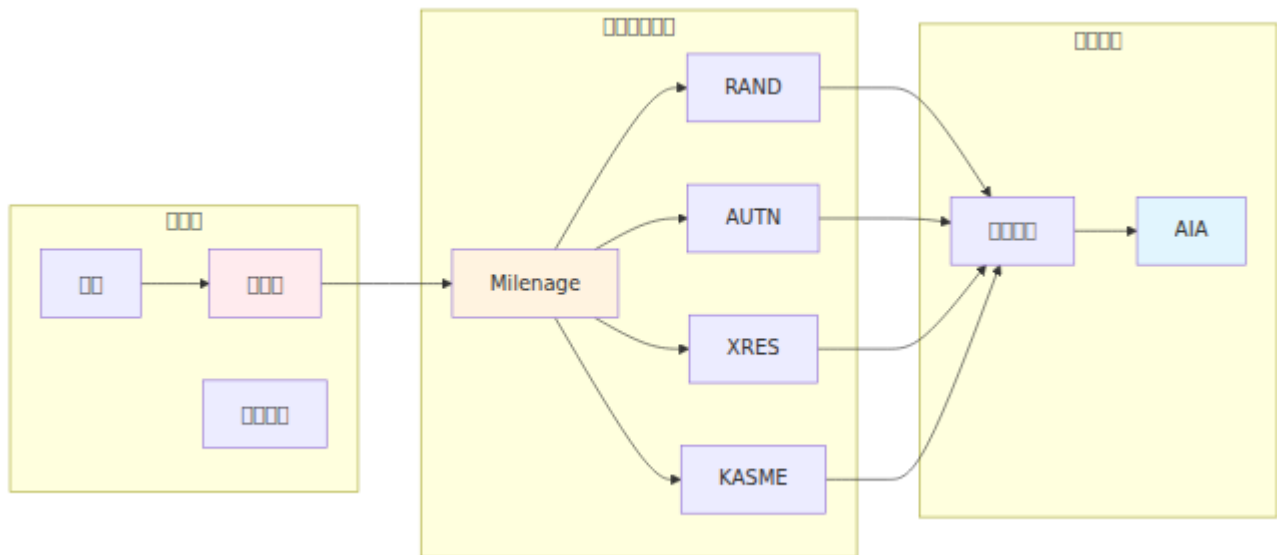
网络选择策略

- 网络选择策略 SQL网络选择策略
- Ki/OPc 网络选择策略 HSS网络选择策略
- AUTN 网络选择策略SQL网络选择策略 AMF 网络选择策略
- 网络选择策略 UE 网络选择策略

AAA (Cx SAA)

AAA HSS IMS S-CSCF

AAA



AAA

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

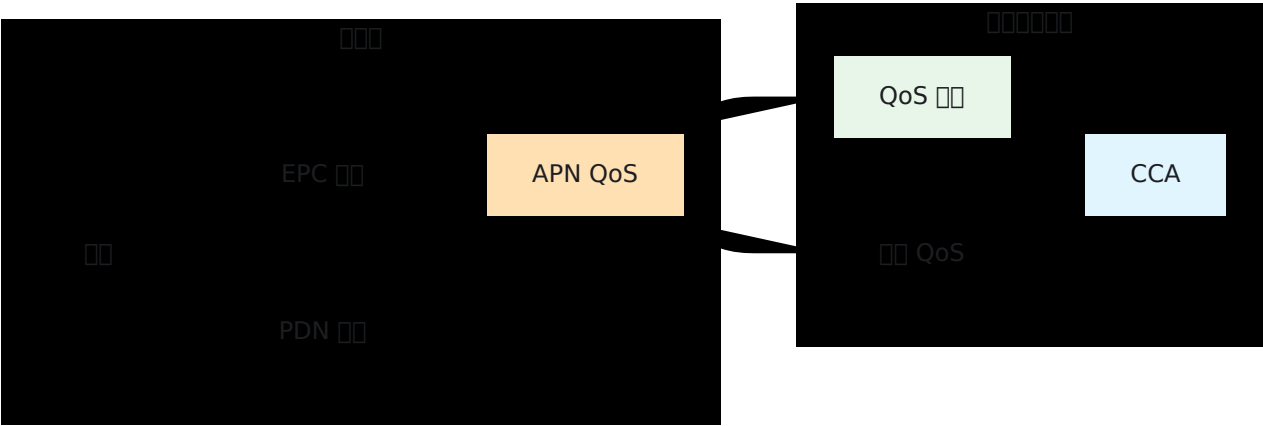
IFC AAA

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

AAA (Gx CCA)

AAA PCRF PGW

PDN



QoS

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

QoS

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

Sh UDA

HSS Sh AS

Sh



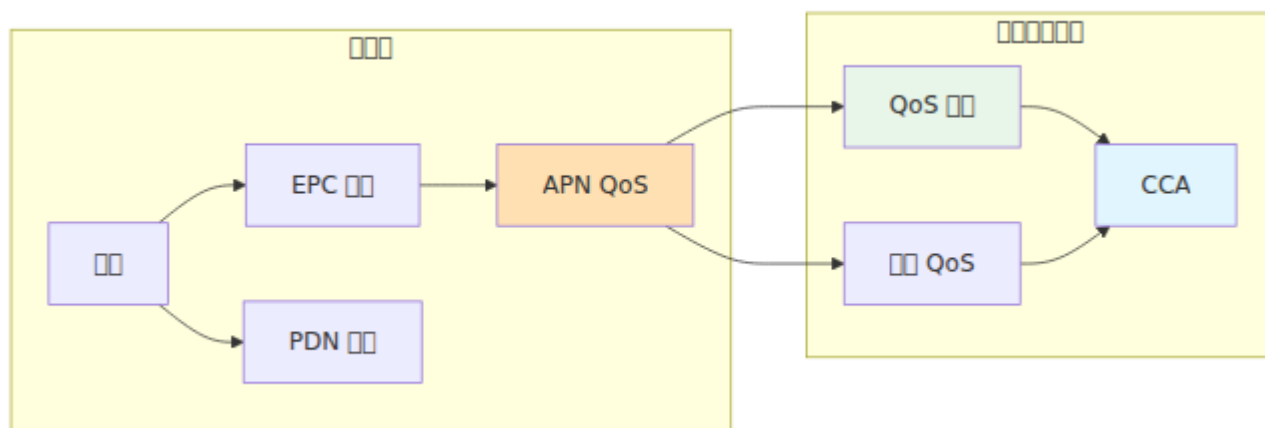
□ □ □ □ □

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

ME (S13 ECA)

ME 0000000 EIR 00000 MME 00 IMEI 000

□ □ □ □ □



□□□□□

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

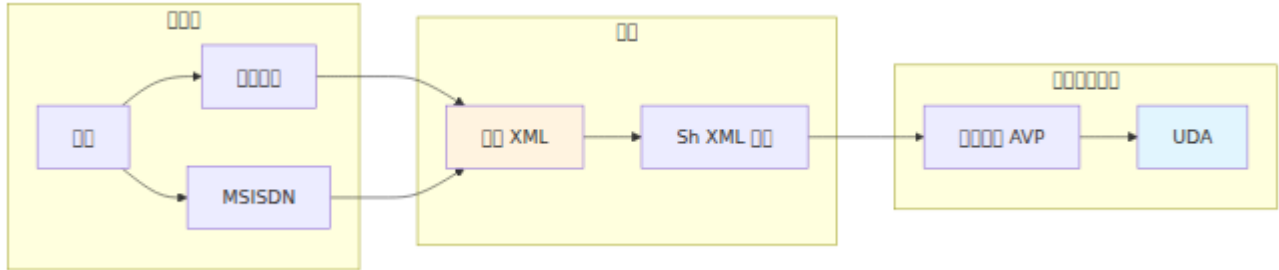
111

- IMEI
 - TAC
 -
 -
-

Diameter AVP

□□□□□

□□□□□□



□□□□

□□□□□□

□□□□□□ 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

← 返回主頁 | **API 說明** → | 關於我們 →

OmniHSS 部署架构图

← 部署图

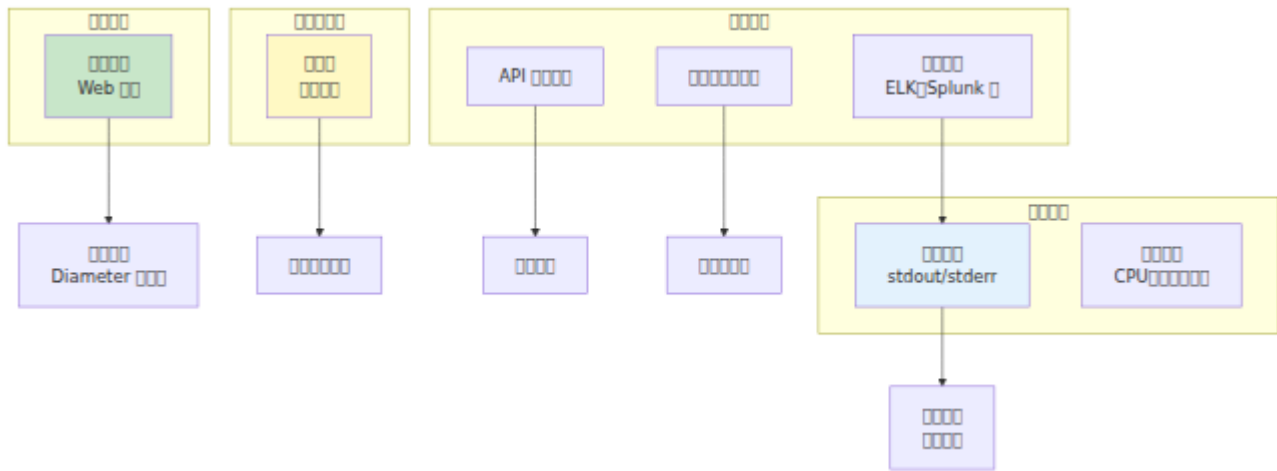
部署

- 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



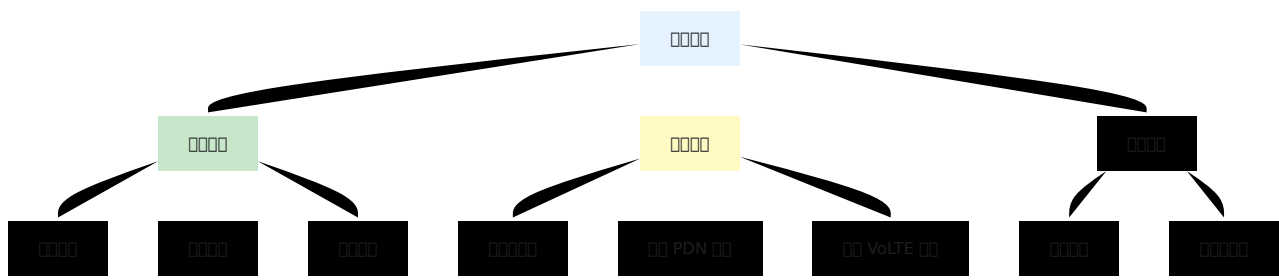
IMS Overview

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

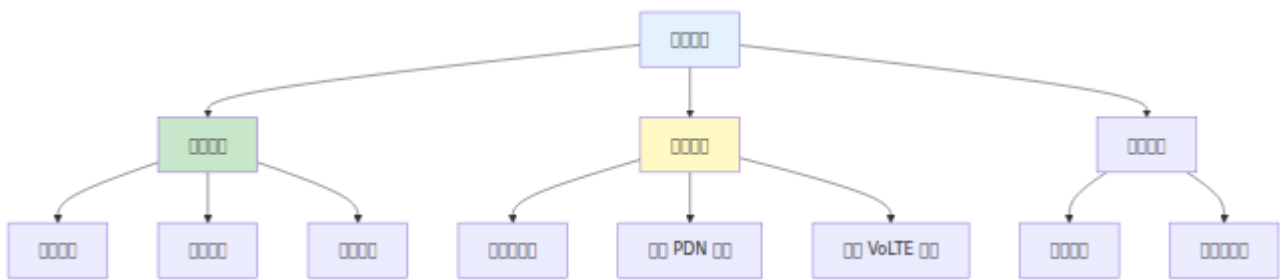
1. Item
2. Item
3. Item

4. 4G/LTE 3GPP 网络架构

Diameter 网络

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

网络



网络

网络

网络	网络	网络
MME	网络	网络 LTE 网络
P-GW	网络	网络
S-CSCF	网络	网络 IMS 网络
P-CSCF	网络	网络 VoLTE 网络
I-CSCF	网络	IMS 网络
AS	网络-网络	网络

网络

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

网络

項目	単位	測定項目	測定結果
通話品質	単位 Erlang	通話品質	> 90% 通話
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

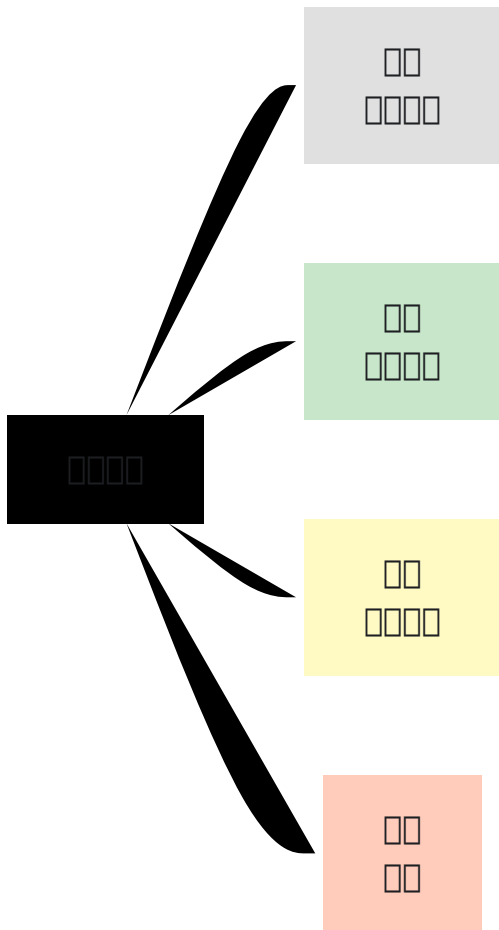
- 0000000000000000
- 000 00000000
- 0000000000
- 0000000000000000

000000

000000

OmniHSS 000000 **stdout/stderr**0000000000000000

0000



Network Configuration

Diameter Network:

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

Database:

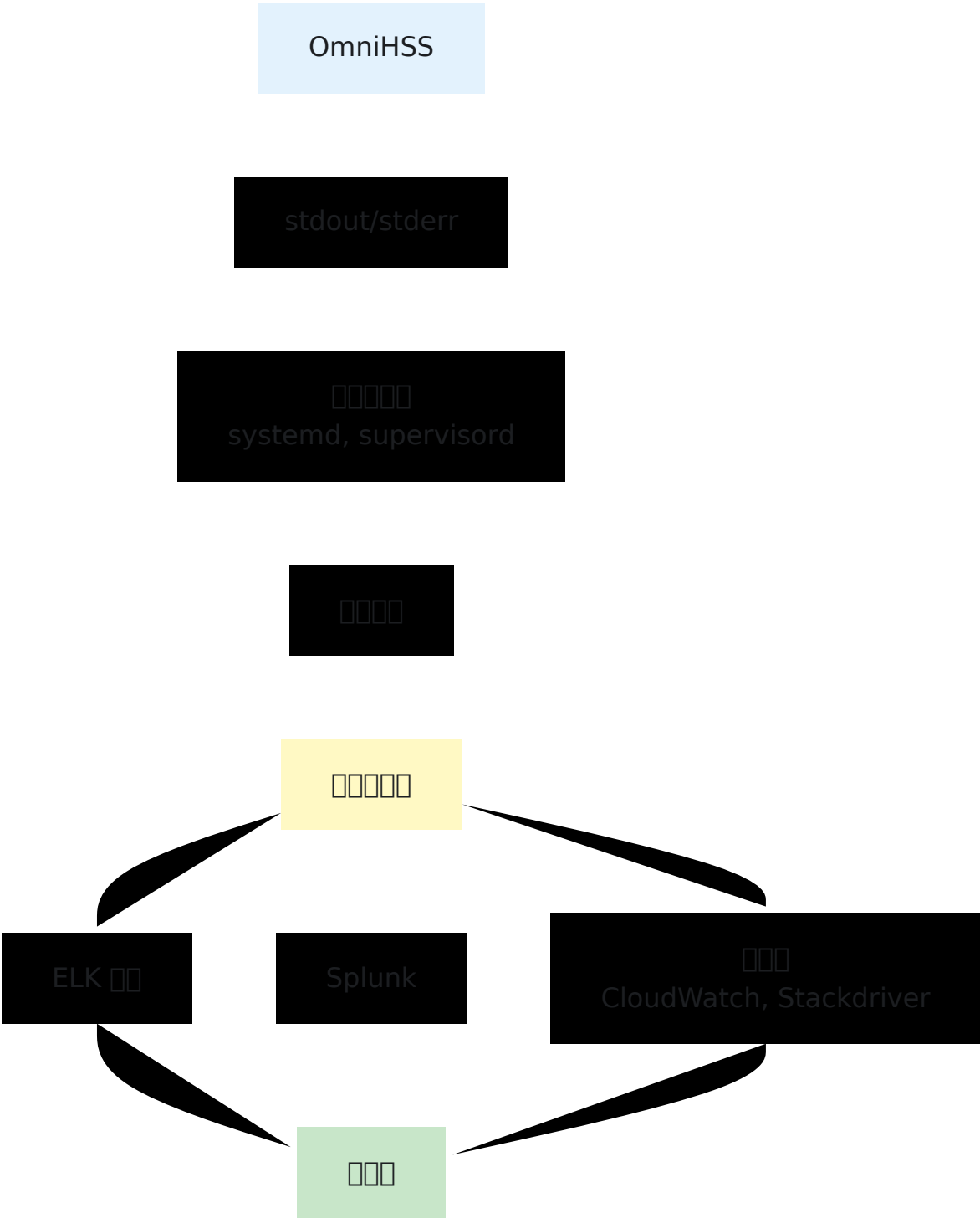
```
[info] Database connection established
[error] Database connection lost: timeout
[error] Database query failed: deadlock detected
```

Logs:

[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

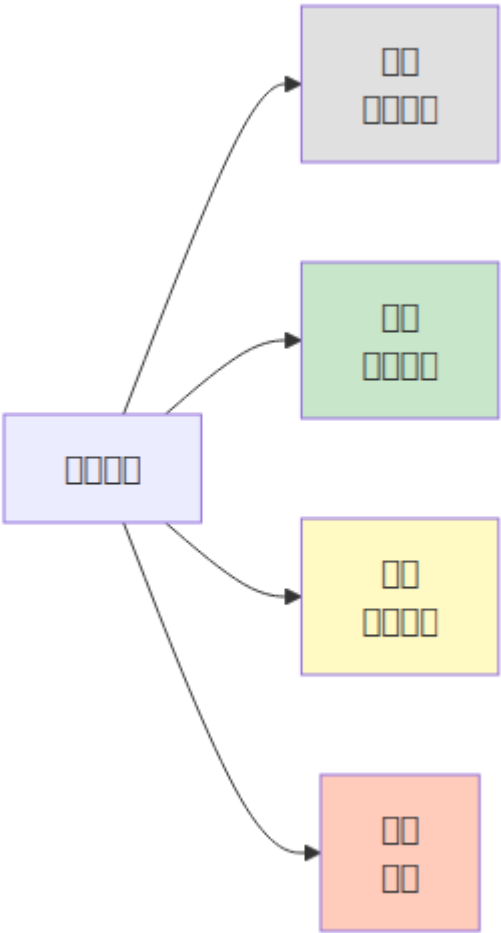


Table 1: KPI

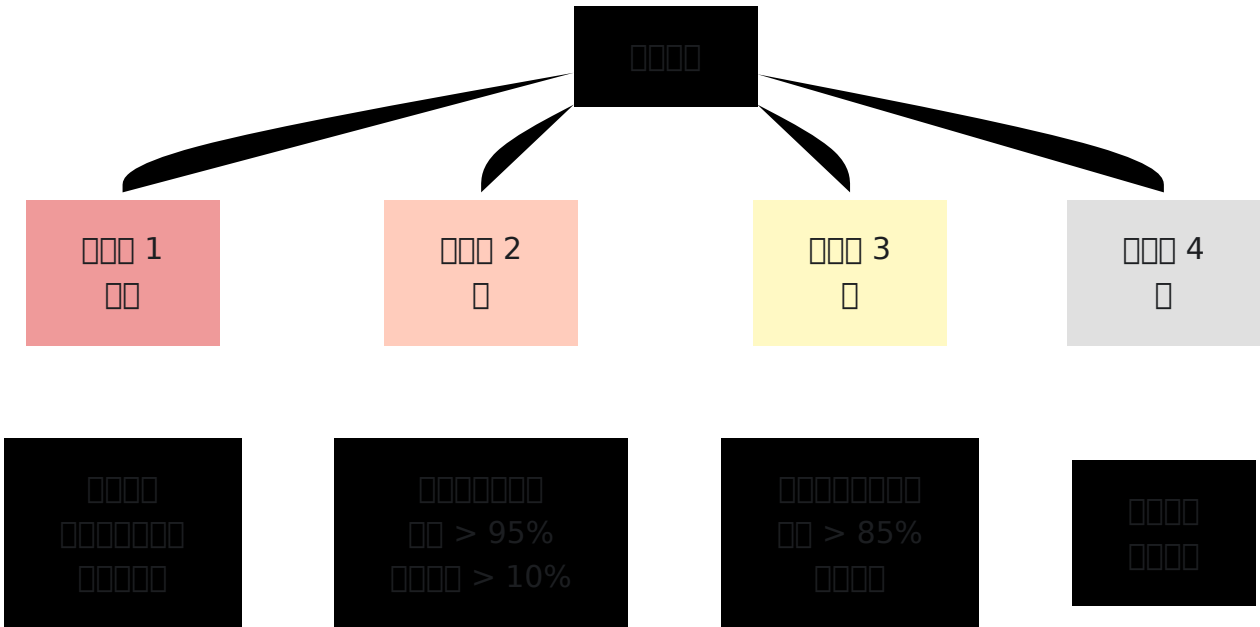
KPI	Target	Current	Goal
Availability	99.99%	< 99.95%	< 99.9%
Diameter Success Rate	99.9%	< 99.5%	< 99%
Latency	> 99%	< 99%	< 95%
Diameter Latency	< 100ms	> 200ms	> 500ms
Throughput	< 50ms	> 100ms	> 500ms
Drop Rate	< 0.1%	> 0.5%	> 1%

Table 2: KPI

Target	Current	Goal
Availability	80%	80% Availability
PDN Latency	70%	70% Latency
Throughput	80% MB	80% Throughput
Drop Rate	80%	80% Drop Rate

□□□□

□□□□□



□□□□

□□□□ (P1)

□□□□□:

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

□□ **Diameter** □□□□□:

- □□□□□□
- □□: □□□□□□□□□□

□□□□□:

- □□□□□ SQL □□□

- 時間: 10分以内

問題 (P2)

問題 Diameter 問題:

- MME 問題
- P-GW 問題
- S-CSCF 問題
- 時間: 15 分以内

問題:

- 時間 > 95%
- 時間: 10分以内

問題:

- 10% 以内
- 時間: 10分以内

問題 (P3)

問題:

- 問題
- 問題
- 時間: 1 分以内

問題:

- 時間 > 85%
- 時間: 10分以内

問題:

- 時間 > 1%
- 時間: 10分以内

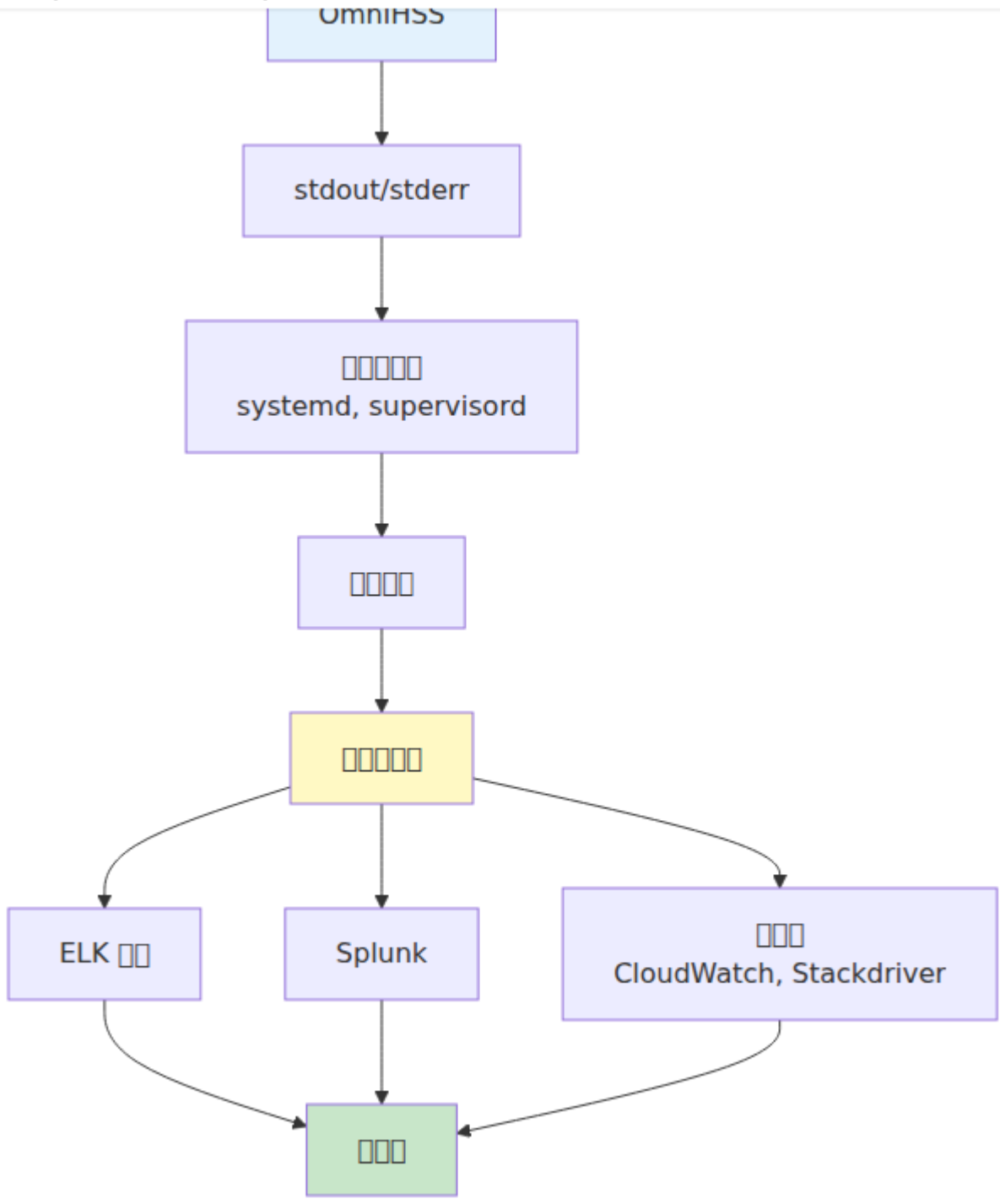
□□□□□ (P4)

□□□□:

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

□□□□:

- □□□□□□□□□□
- □□: □□□□□□□



□□□□□□

□□□□

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

□□□□

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

□□□□

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

OmniHSS ◻MSISDN◻◻ IMSI◻◻

←

11

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

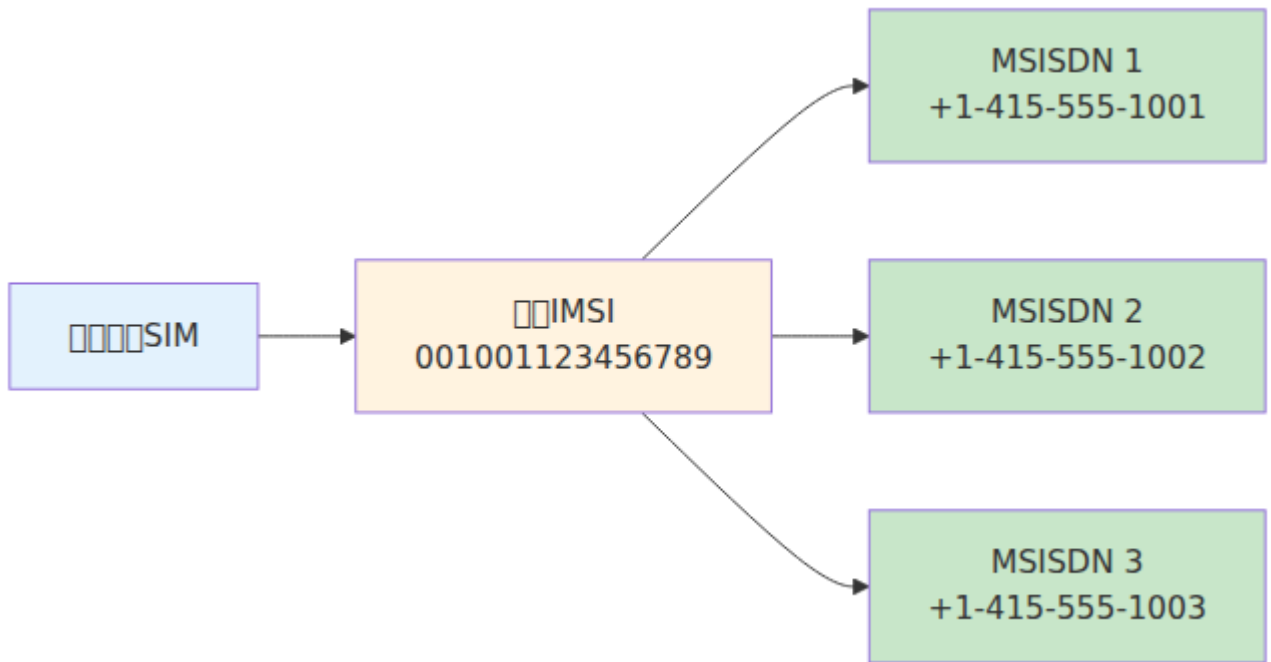
11

OmniHSS

MSISDN

IMSI →

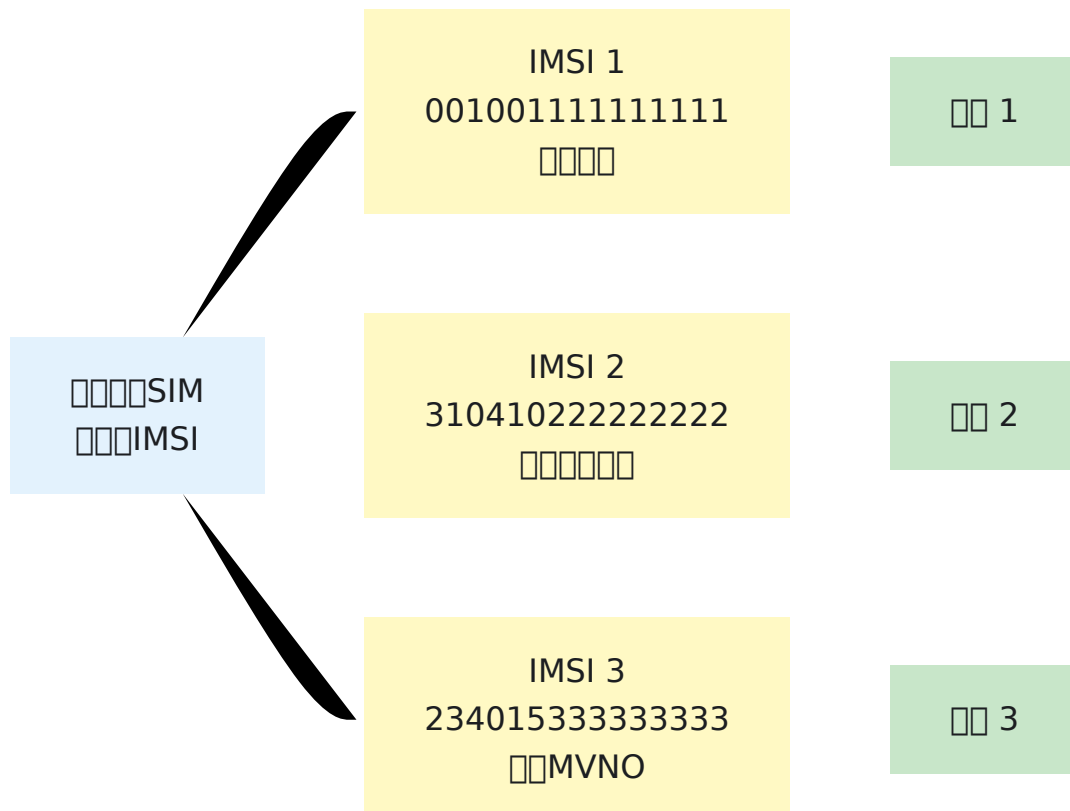
IMSI	MSISDN
------	--------



IMSI SIM

SIM → IMSI

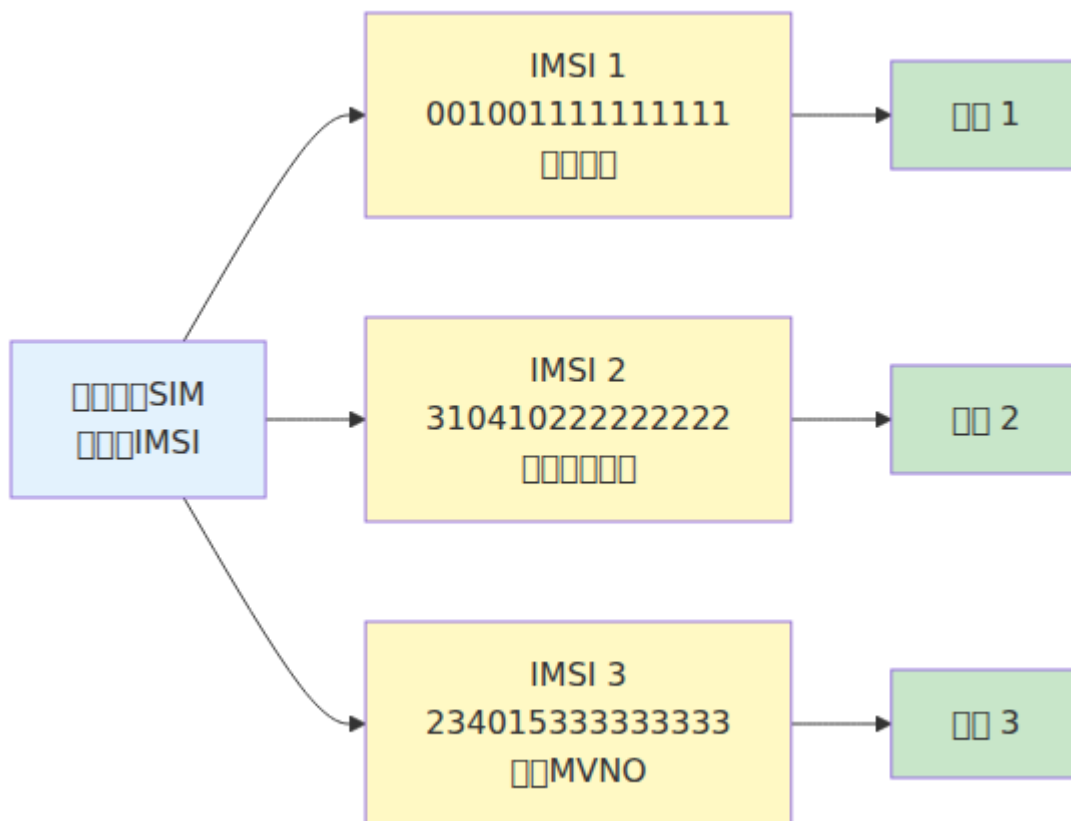
SIM IMSI MVNO



IMSISDN

HSSMSISDNIMSMSISDNIMS

- IMSI - SIMIMSI
- MSISDN -
- IMS - MSISDNIMS
- - EPCIMS

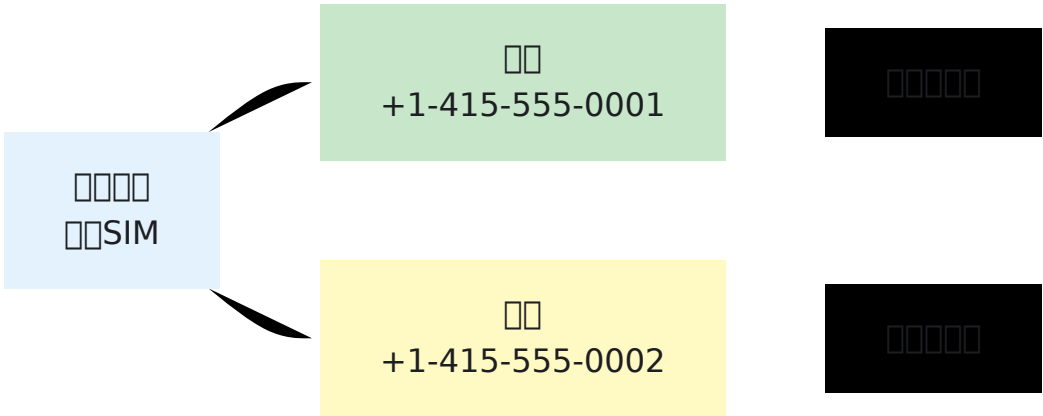


MSISDNMSISDN

□□□□

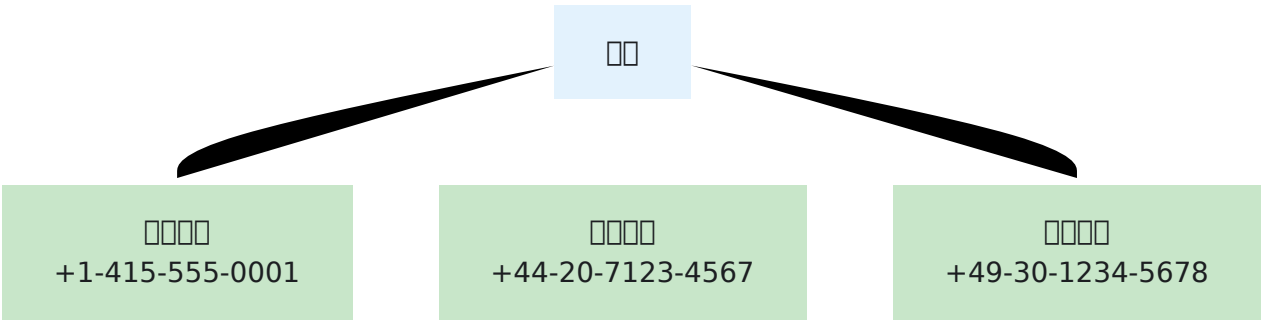
1. □□□□□□

□□□□□□□□□□□□□□□□□□



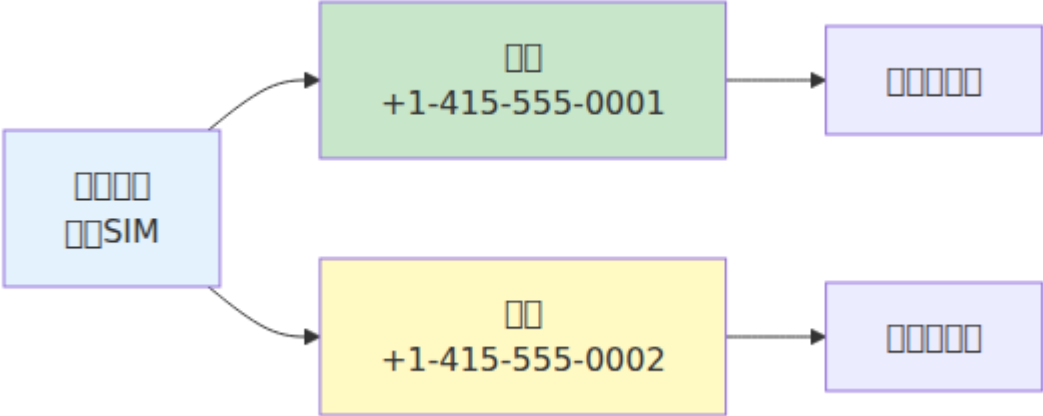
2. □□□□

□□□□□□□□□□□□□□□□



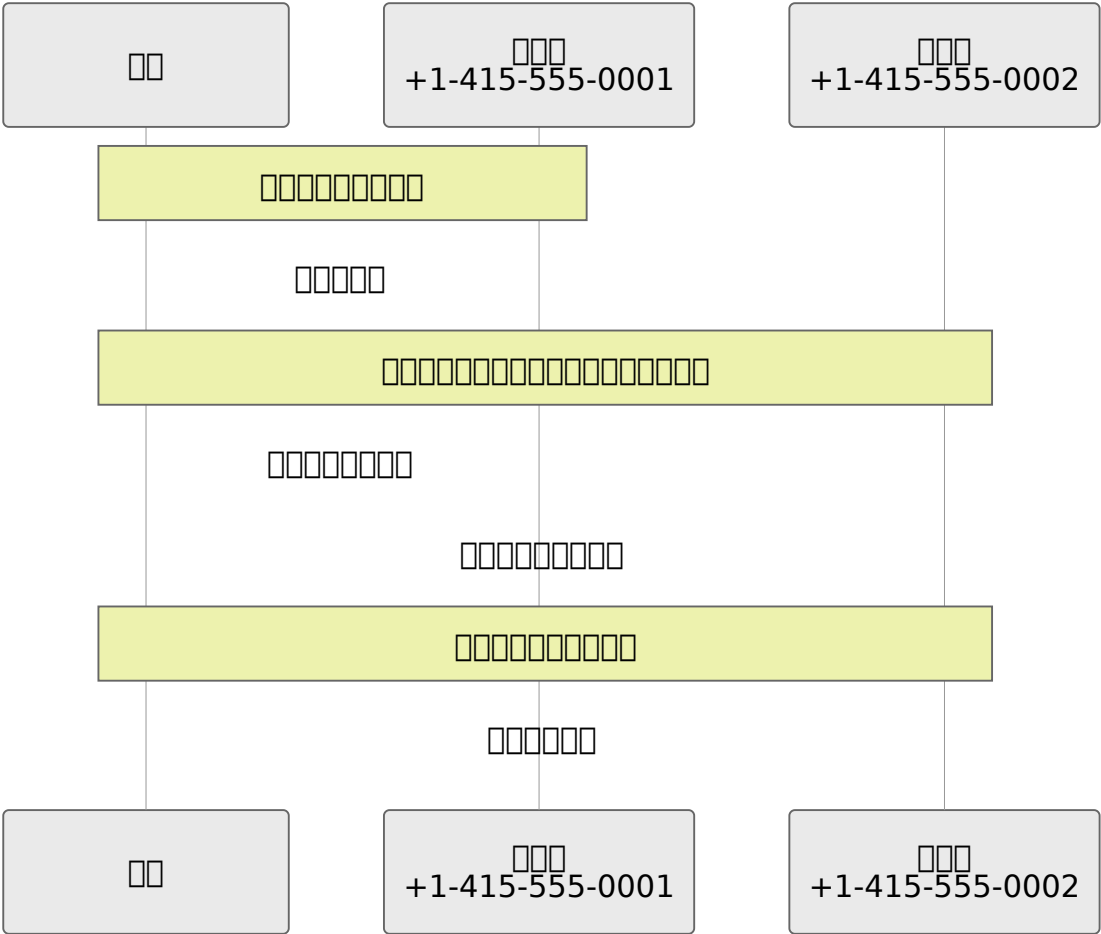
3. □□□□

□□□□□□□□□□□□□□□□



OmniHSS SIM/IMSI MSISDN

4.



MSISDN

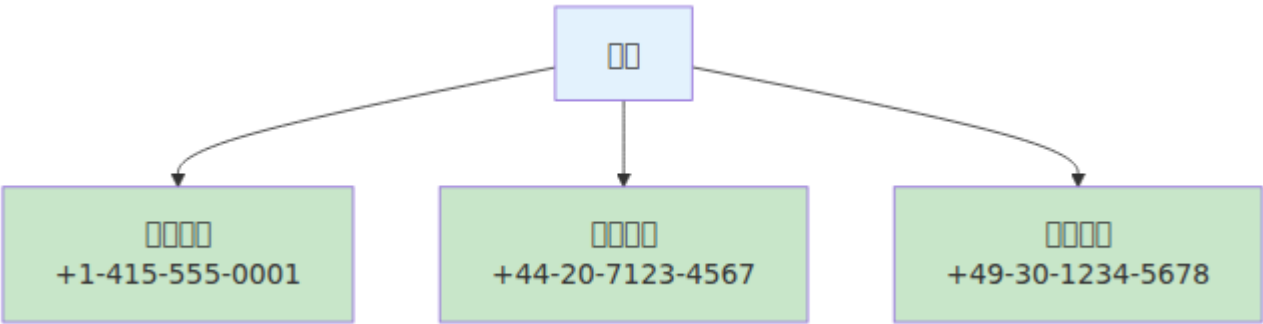
MSISDN

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

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

MSISDN

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



MSISDN

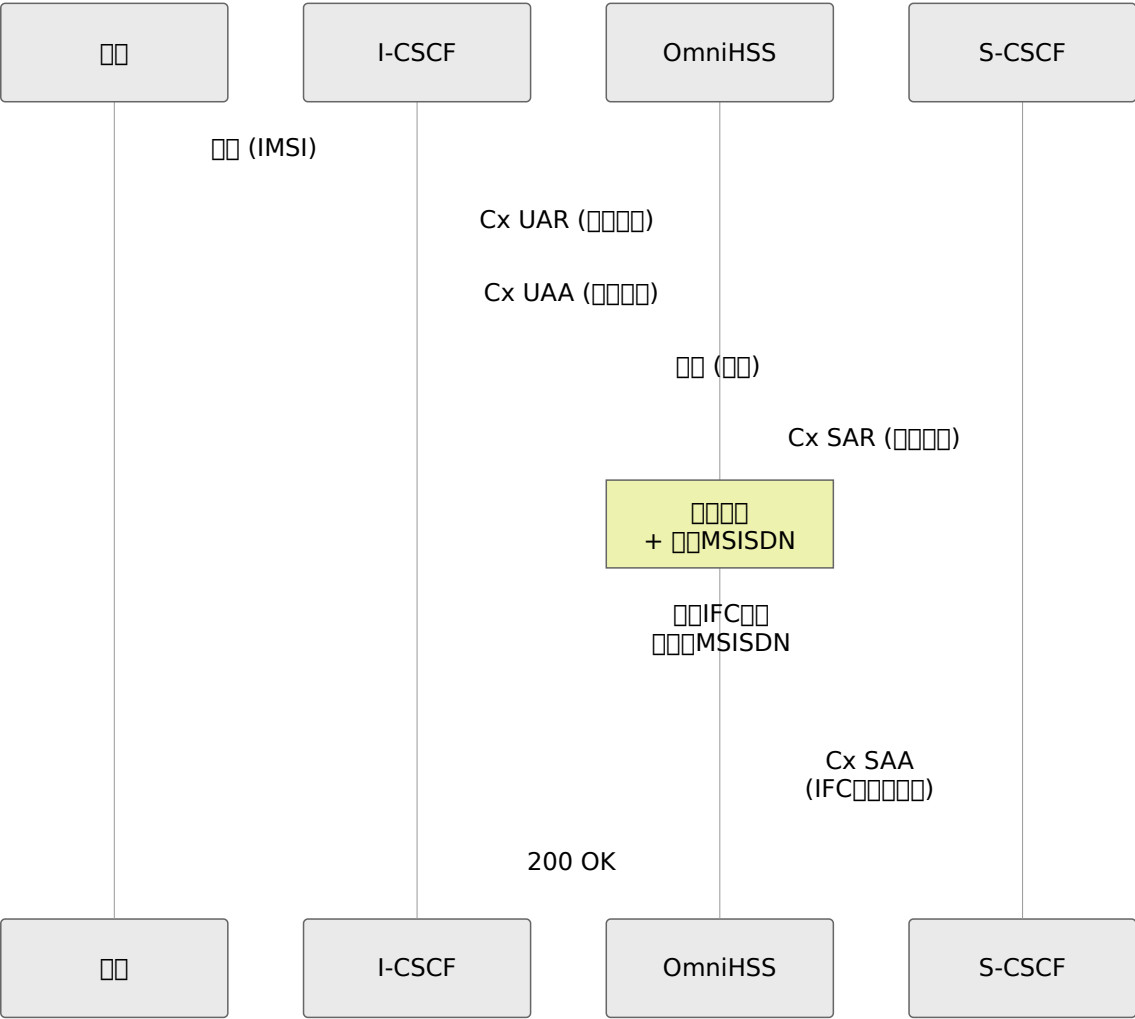
-
- msisdn
-

□□□□□ID□IMS□□□□□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>

```

MSISDN

- `{{msisdns}}` - MSISDN

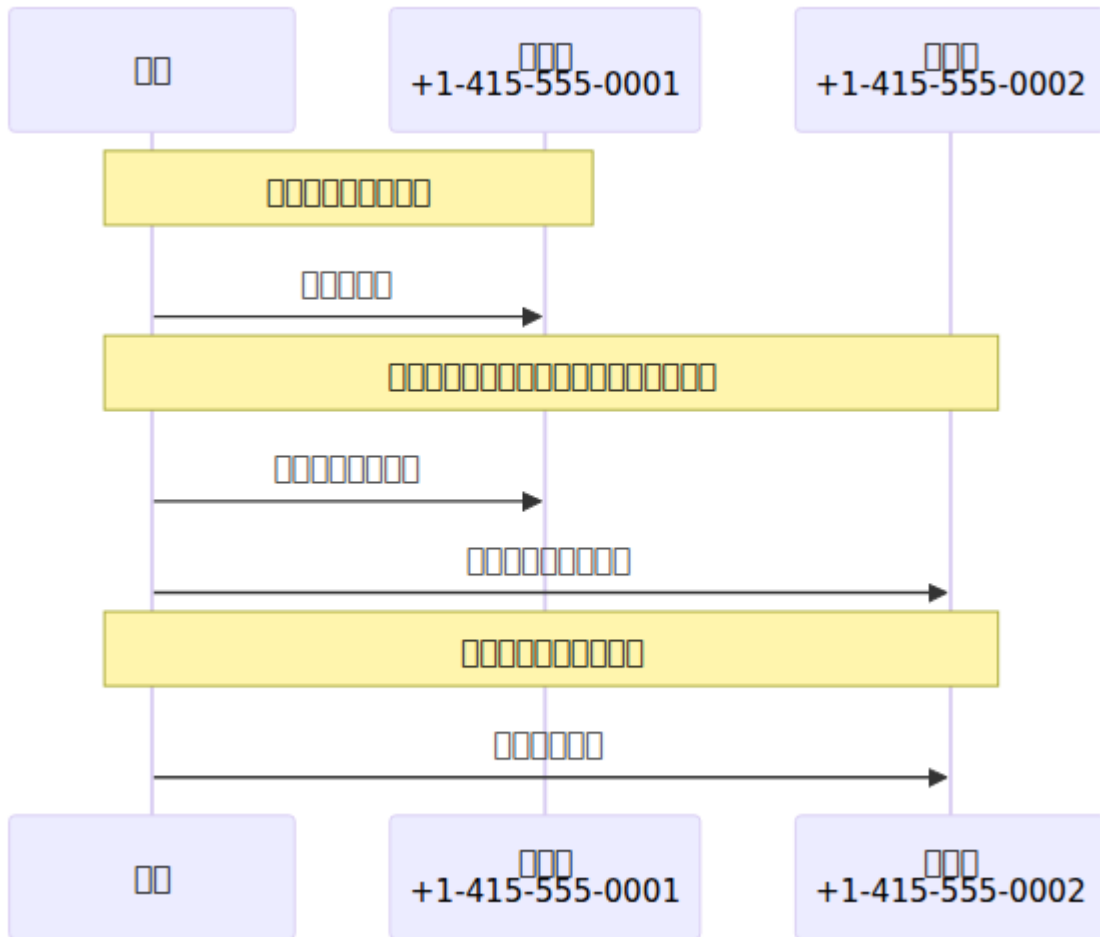
URI

MSISDN IMS



MSISDN

IMS SIP URI



INVITE

200 OK

SIP INVITE

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

From P-Asserted-Identity

MSISDN

MSISDN **IMS**

- S-CSCF
-

1. MSISDN

- IMSI
- MSISDN

2. IMS

- `{{msisdns}}`
- XML

3. HSS

- IMS Cx SAR
- MSISDN

4. IMS

-
- S-CSCF

MSISDN

-
- “” “”

1. MSISDN

- MSISDN
-

2. MSISDN

- 獲取MSISDN
- 通過API獲取MSISDN

獲取MSISDN

獲取

- 獲取MSISDN
- 獲取MSISDN

獲取MSISDN

1. 獲取IMS

- S-CSCF
- SIP URI

2. 獲取IMS

- IFC
- 獲取IMS

3. 獲取IMS

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

APIMSISDN

獲取

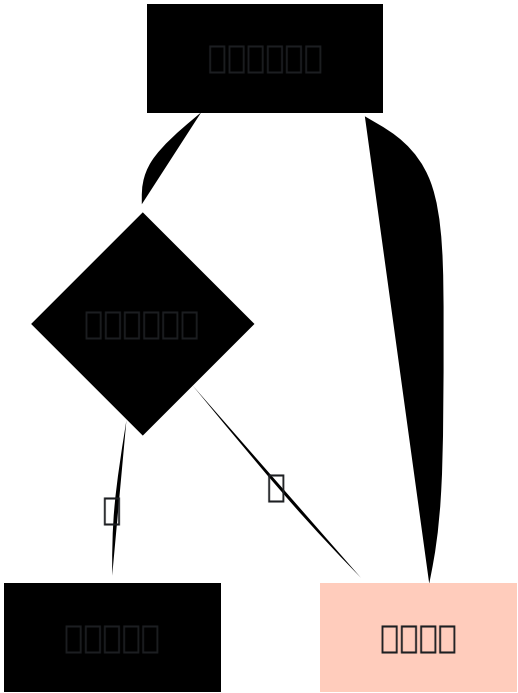
- API/api/subscriber/msisdn/:msisdn

獲取

獲取MSISDN

1MSISDN

MSISDN



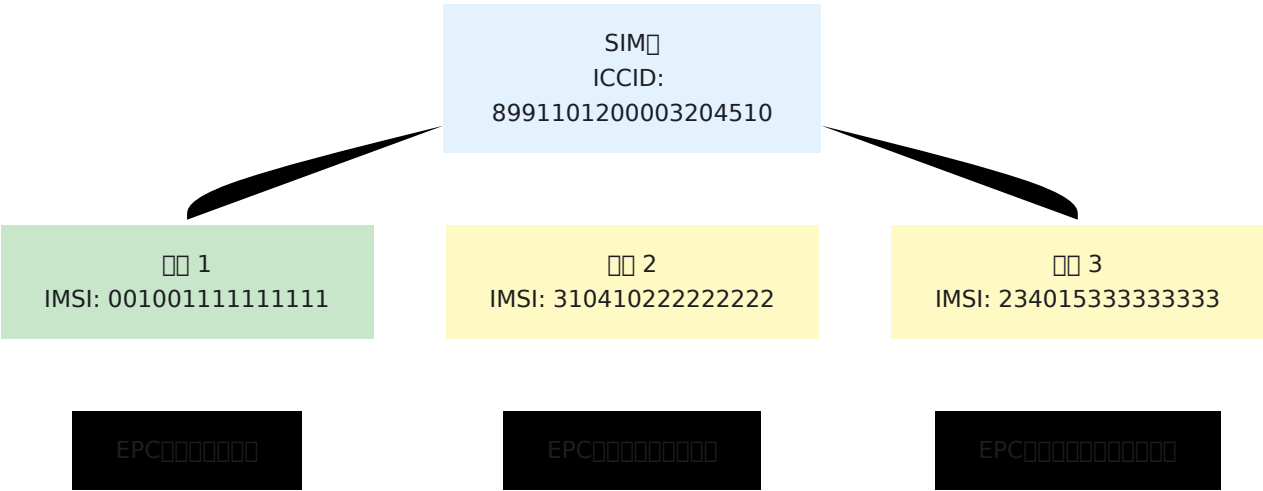
IMSI SIM

IMSI SIM IMSI IMSI

IMSI SIM IMSI HSS IMSI

OmniHSS

OmniHSS IMSI SIM IMSI SIM



1. 1. 1. 1.

1. 1. 1. 1.

- IMSI 001-001
- IMSI 310-410
- IMSI 234-015
- IMSI

2. MVNO

- IMSI MVNO
- IMSI
- MVNO

3. IoT/M2M

- IMSI 1
- IMSI 2
- IMSI 3
-

4. 4. 4. 4.

- IMSI
-
-

IMSI

- IMSI Ki OPC
- IMSI
-

- EPC APN
- IMS ? ?
- IMS

- IMSI SIM sim_id
- ICCID
- SIM

- SIM IMSI
-
- HSS IMSI



```
# 1. SIM卡注册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卡注册密钥
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. 注册 1卡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
  }}'

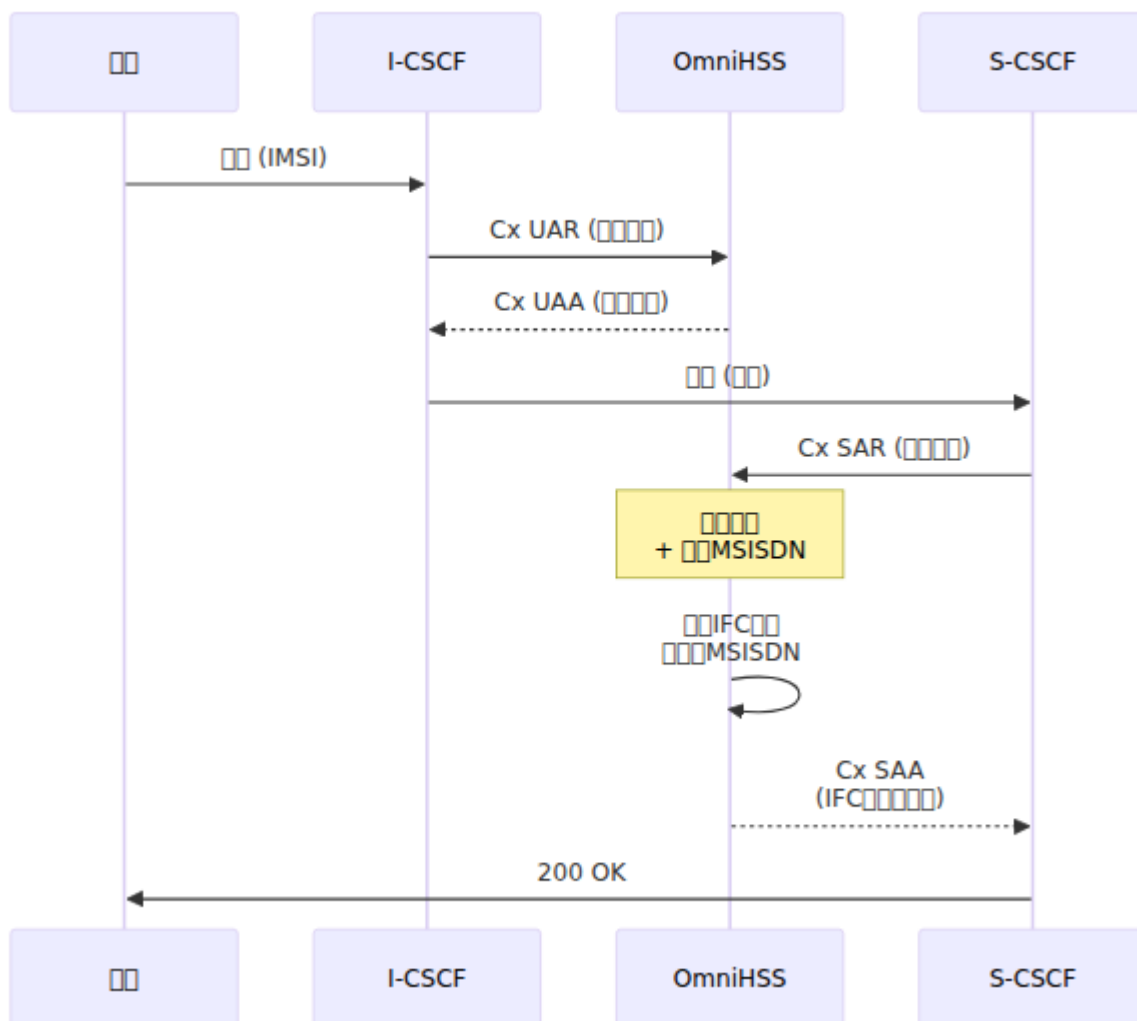
# 4. IMSI 2卡注册密钥
KEYSET2=$(curl -k -X POST https://hss.example.com:8443/api/key_set \
  -d '{"key_set": {"ki": "111111111111111...", "opc": "2222222222..."}}' \
  | jq -r '.data.id')

# 5. 注册 2卡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卡注册IMSI卡注册...
```

□□□□□□

□□IMSI□□□□□□



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

IMSI□□□□□□

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

□□□□IMSI□□

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

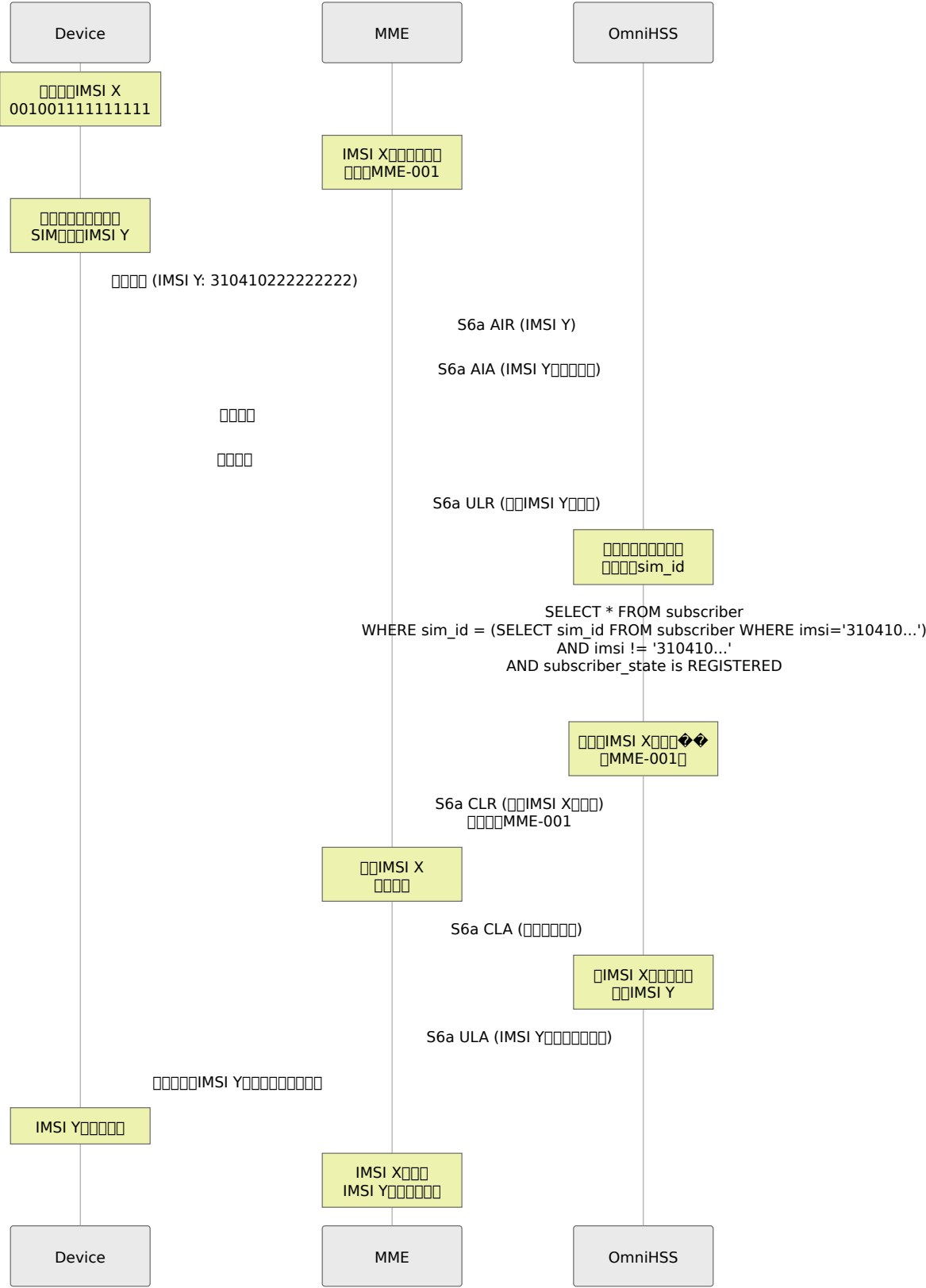
- □□□□□MME□□□□IMSI X
- □□HSS□□□□□IMSI Y□□IMSI X□□□□SIM□□□□□□□□□□□□□□□□

- HSS

IMSI X

IMSI

IMSI



□□□□□□

□□□□□□

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

□□□□□□

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

□□□□□□

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

IMSI□□□□□□

□□/SIM□□□□□□□IMSI□□□□□

1. □□□□□□

- □□IMSI□□□□□□
- □□□□□□□□□IMSI

2. □□□□□

- □□□□□□□□□
- SIM□□□□□□□IMSI

3. □□□□□

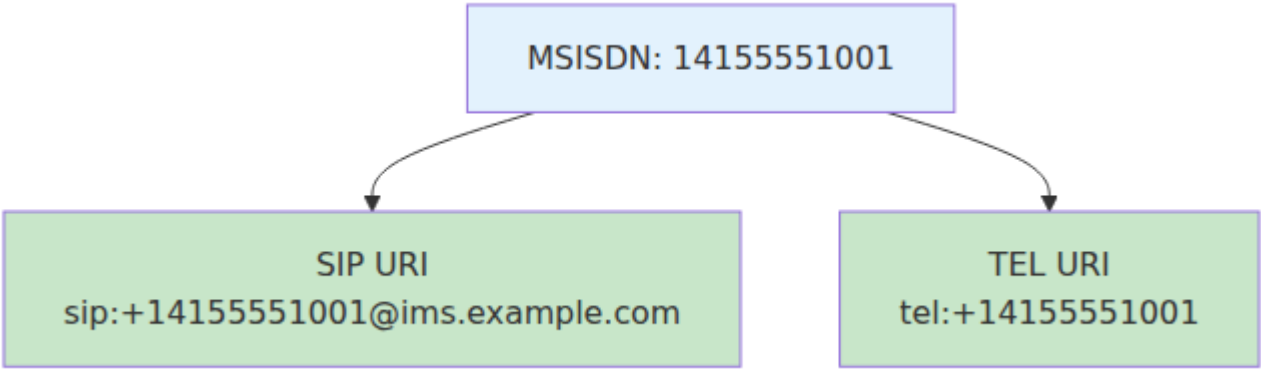
- SIM□□□□□□□□□□□□□□□IMSI□
- □□MCC/MNC□□□□□□

4. □□□□□

- 國際移動用戶識別碼IMSI
- 國際IMSI號碼

IMS

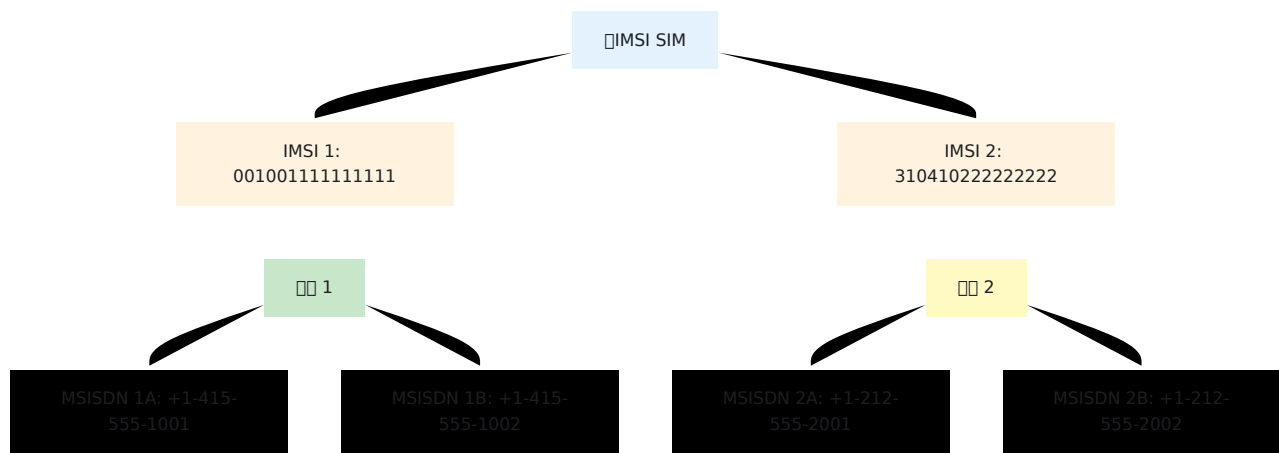
國際移動用戶識別碼IMSI



1. 國際移動用戶識別碼IMSI國際移動用戶識別碼HSS“”
2. SIM國際移動用戶識別碼IMSI SIM國際移動用戶識別碼sim_id“”
3. subscriber_state國際移動用戶識別碼MME/SGSN國際移動用戶識別碼
4.
 - 國際移動用戶識別碼IMSI
 - 國際移動用戶識別碼IMSI
 - SIM國際移動用戶識別碼IMSI

IMSI + MSISDN

SIM國際移動用戶識別碼IMSI國際移動用戶識別碼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□

如何识别IMSI

如何识别IMSI

1. 如何IMSI识别
2. 如何IMSI识别key_set
3. 如何识别EPC
4. 如何识别

如何识别IMSI

- 如何识别/SIM识别HSS
- HSS识别IMSI
- 如何识别IMSI

如何识别MSISDN

如何识别

1. 如何MSISDN识别
2. 如何IMS识别{{msisdns}}
3. 如何IMS识别
4. 如何S-CSCF识别

如何识别

- 如何识别
- 如何识别HSS
- HSS识别

如何识别

如何识别MSISDN

✓ 如何SIM识别 ✓ 如何识别 ✓ 如何识别 ✓ 如何识别 ✓ 如何识别 ✓ 如何识别

IMSI SIM

✓ 12345678 ✓ 12345678 ✓ 12345678 ✓ 12345678 ✓ 12345678 ✓ 12345678

1234

✓ 12345678 ✓ 12345678 12345678 ✓ 12345678 ✓ 12345678 ✓ 12345678

← 12345678

PCRF

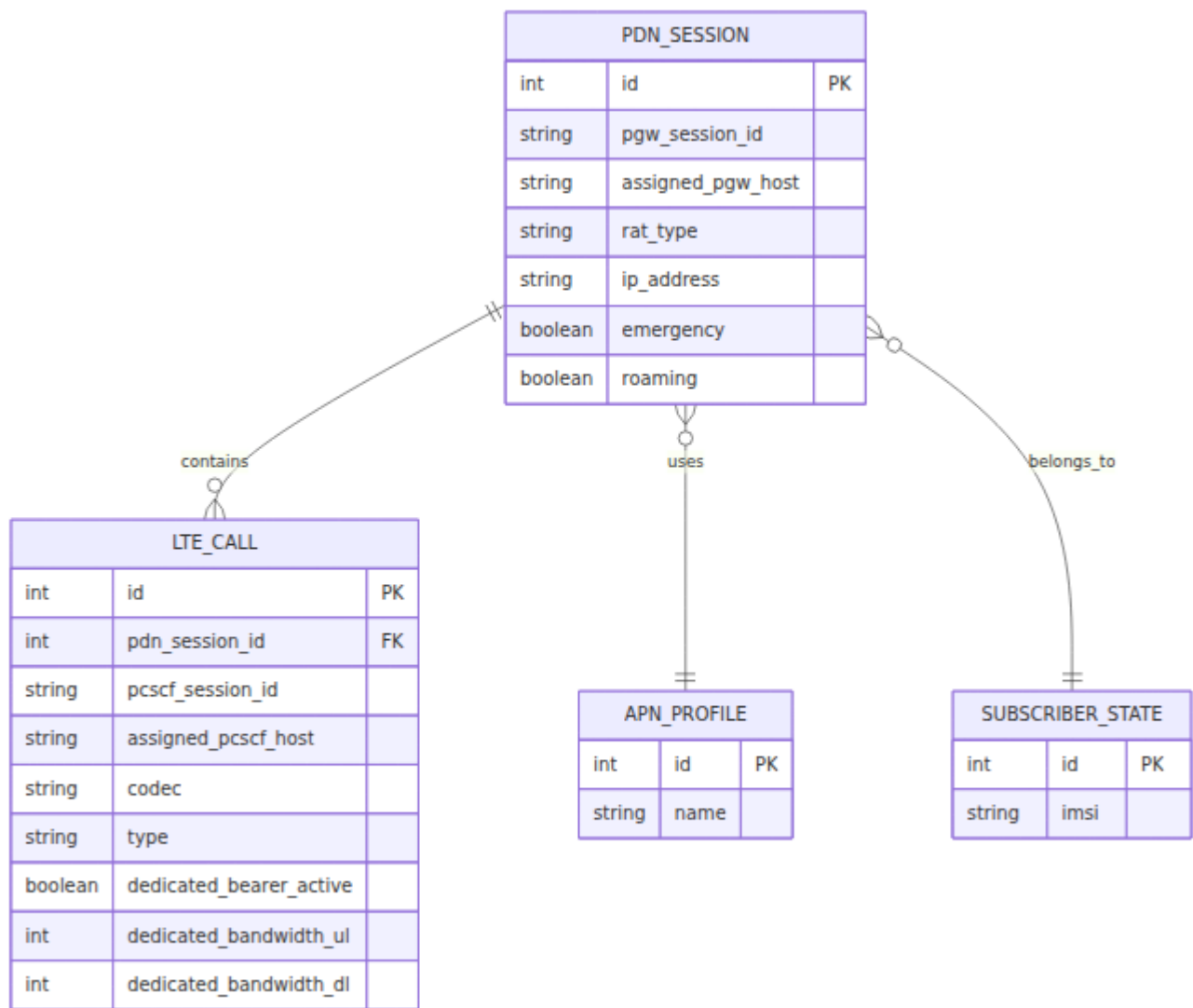
HSS PCRF PCRF LTE QoS

- **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. CCR-I

PGW PDN

AVP

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

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

PCRF ــــــــ

1. ــــــــ IMSI ــــــــــــــــ
2. ــــــــ APN ــــــــ QoS ــــــــ
3. ــــــــــــــــــــــــ
4. ــــــــ APN ــــــــ QoS ــــــــ

ــــــــ AVP ــــــــ

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

2. ــــــــــــــــ - ــــــــ CCR-U ــــــــ

ــــــــ PGW ــــــــــــــــــــــــــــــــRAT ــــــــــــــــ

PCRF ــــــــ

1. ــــــــ ID ــــــــــــــــ
2. ــــــــــــــــRAT ــــــــــــــــ
3. ــــــــــــــــــــــــ

ــــــــ Result-Code 2001 ــــــــــــــــــــــــ

3. ــــــــــــــــ - ــــــــ CCR-T ــــــــ

ــــــــ PGW ــــــــ PDN ــــــــ

PCRF ــــــــ

1. ــــــــ ID ــــــــ
2. ــــــــــــــــــــــــ
3. ــــــــ

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** - 端对端加密和认证
- **SSL** - 端到端加密和认证
- **SSH** - 端到端加密和认证
- **VPN** - 端到端加密和认证
- **QoS** - 端到端服务质量

QoS ☐☐☐☐

QoS APN

```
{
  "QoS-Class-Identifier": 9,           // QCI9 = 0000
  "APN-Aggregate-Max-Bitrate-UL": 50000, // kbps
  "APN-Aggregate-Max-Bitrate-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 ☐ ☐

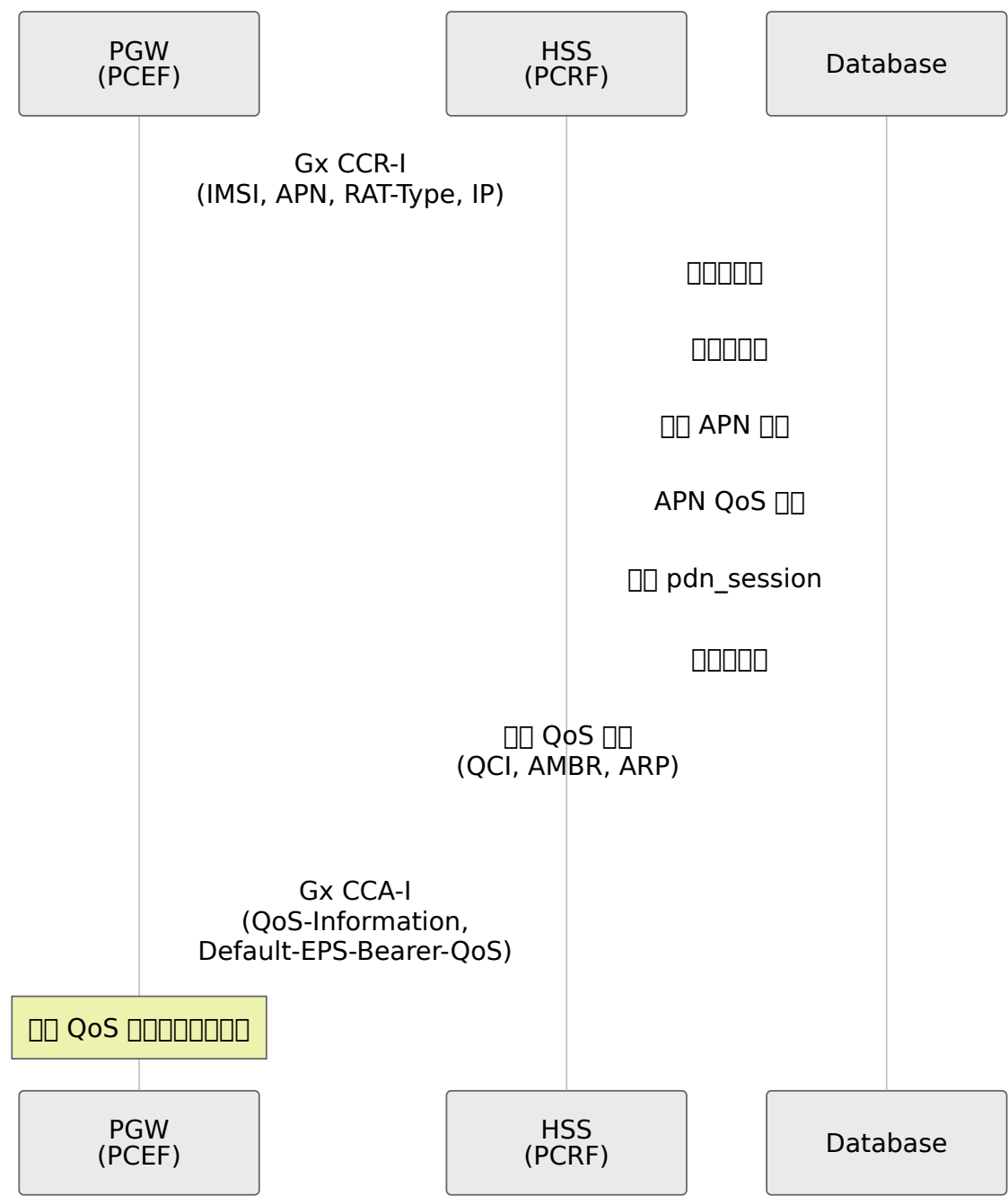
□ □ □ □ □

1. AA □□□AAR□/ AA □□□AAA□

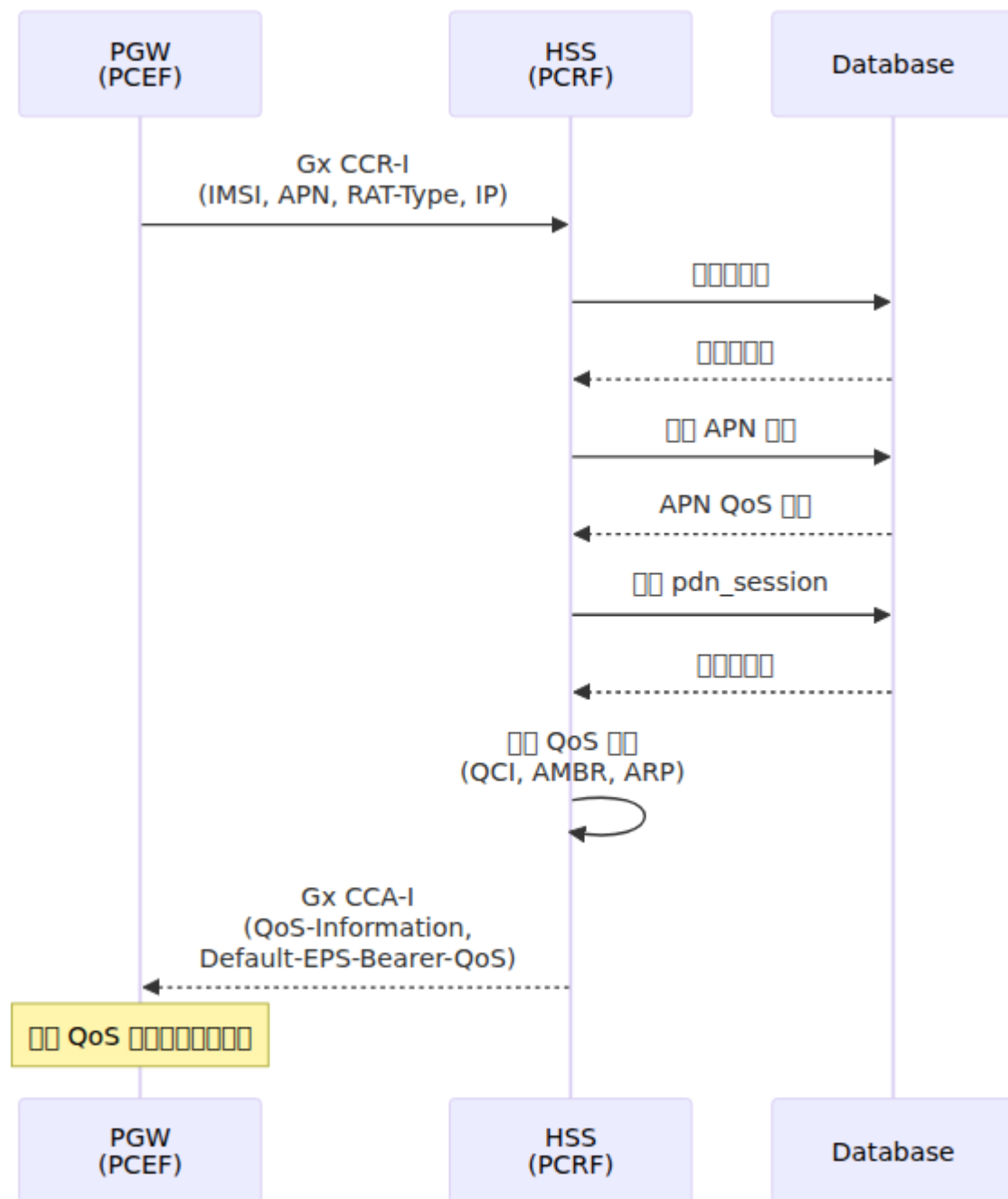
1. 00 P-CSCF 00 ID 000000
2. 00 **Gx RAR** 0 **PGW** 000000
3. 00000000
4. 00 STA 00

Result-Code 2001

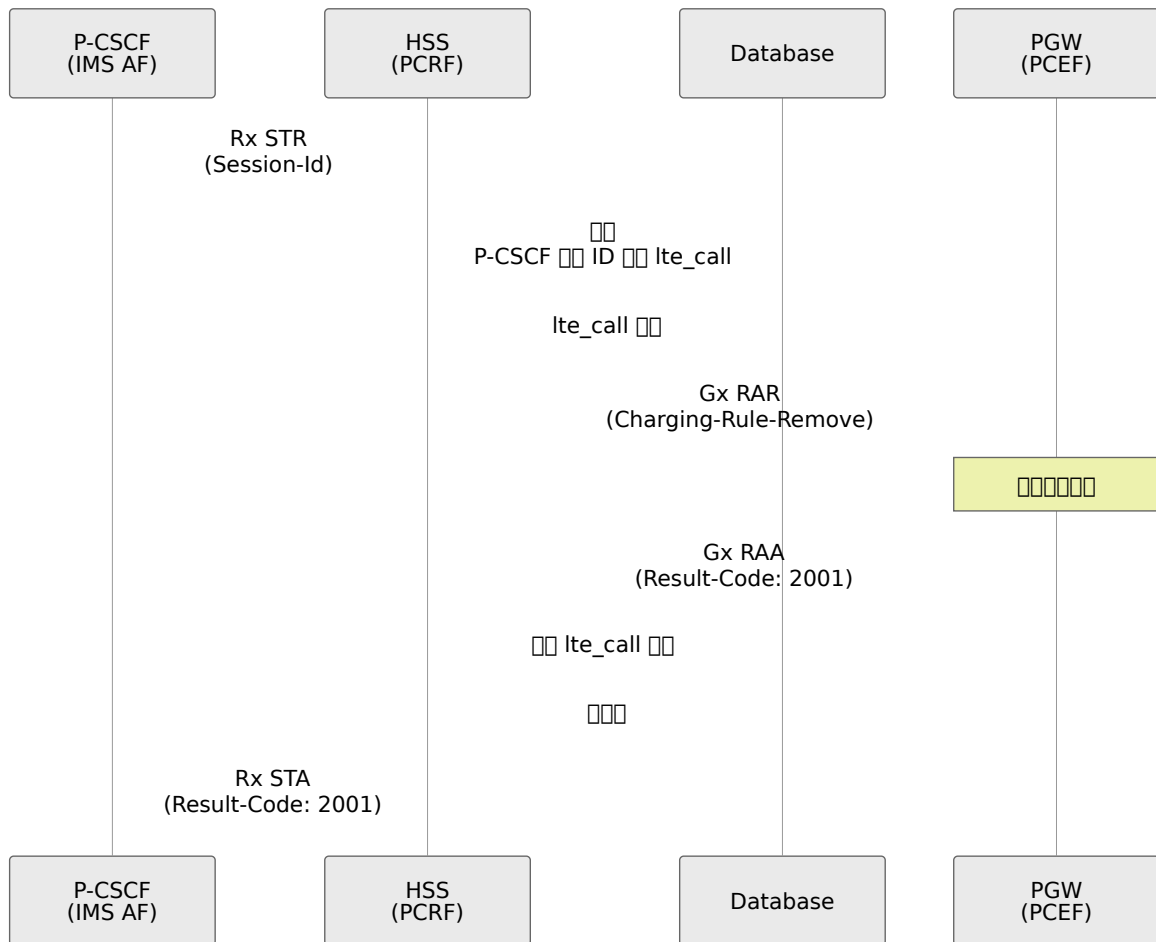
1 PDN



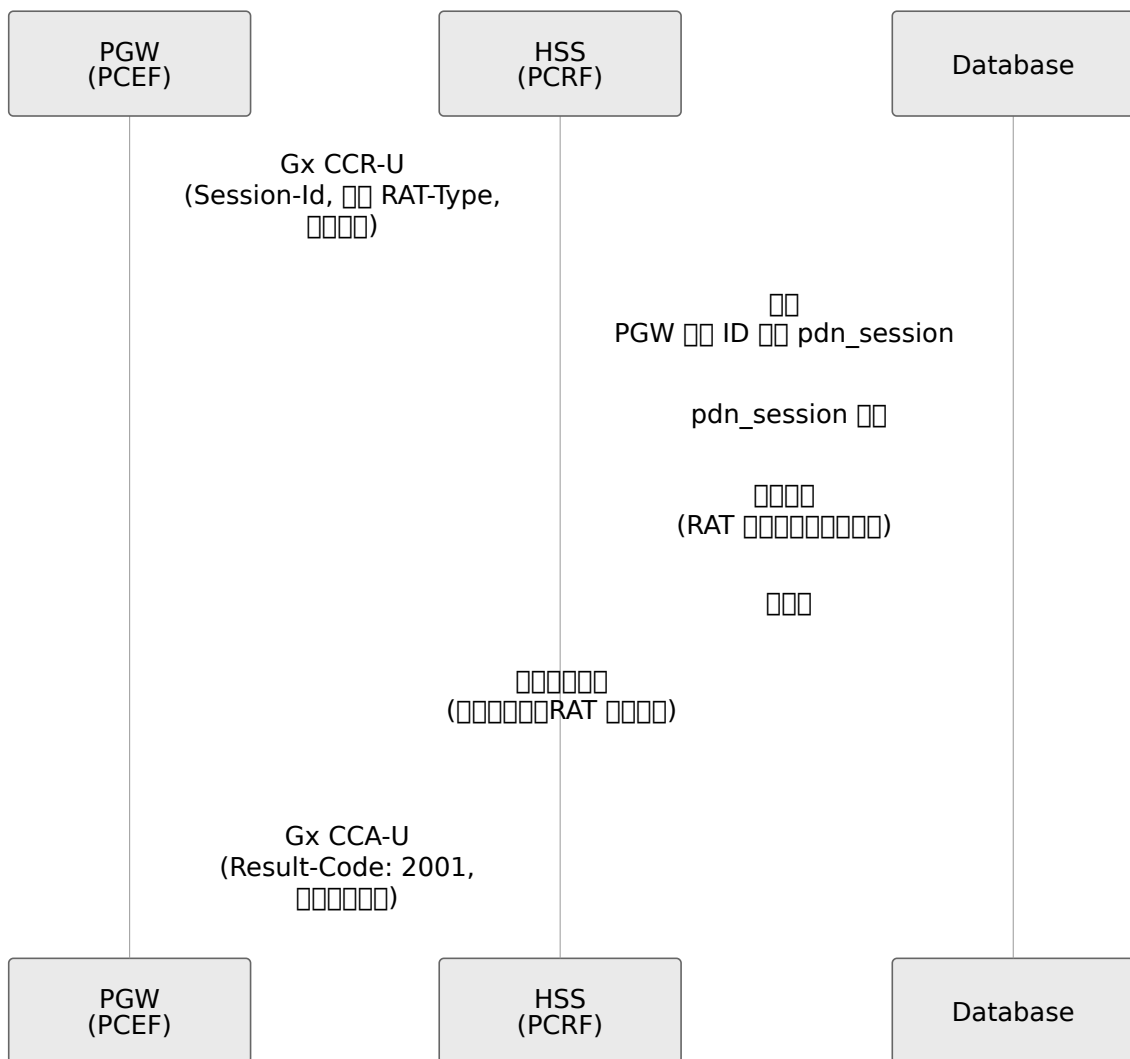
2 VoLTE Rx AAR → Gx RAR



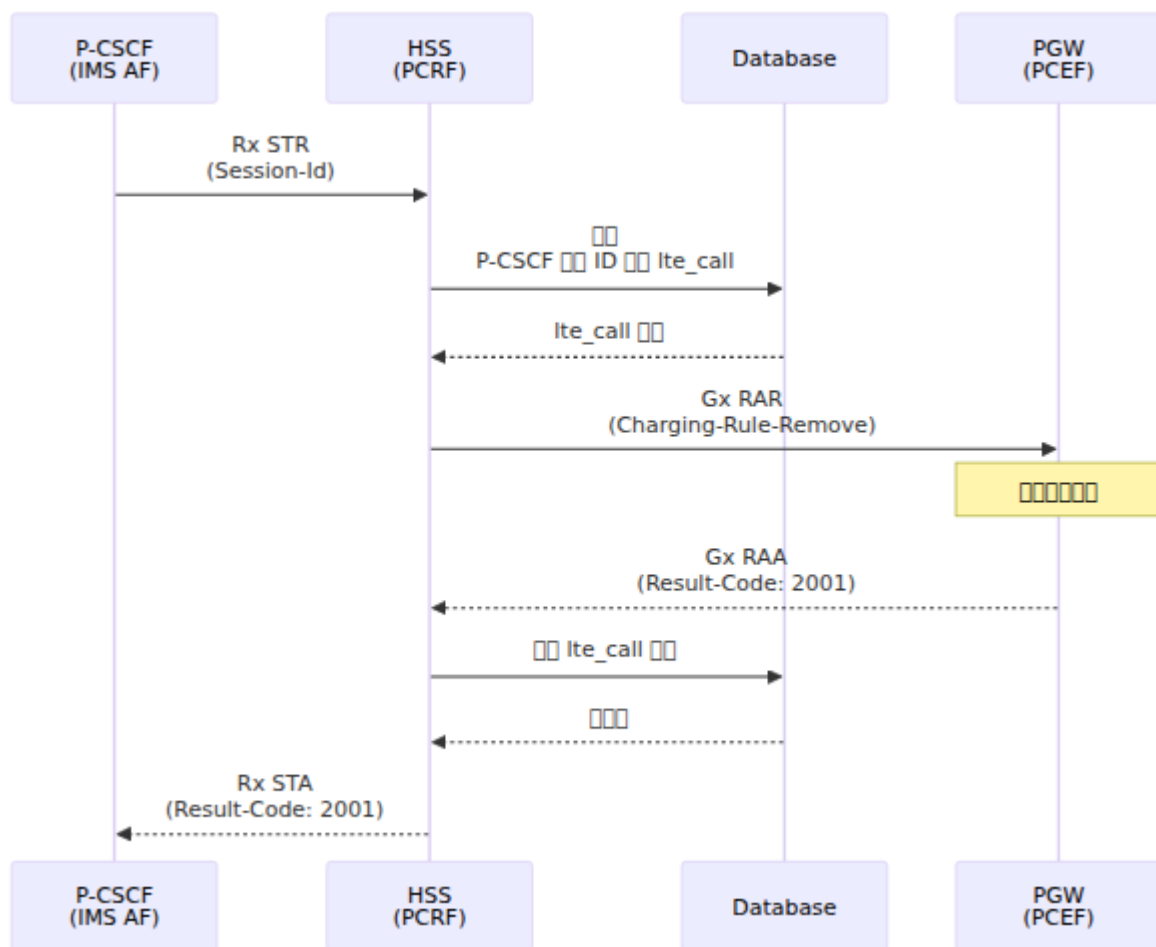
3 VoLTE Rx STR → Gx RAR



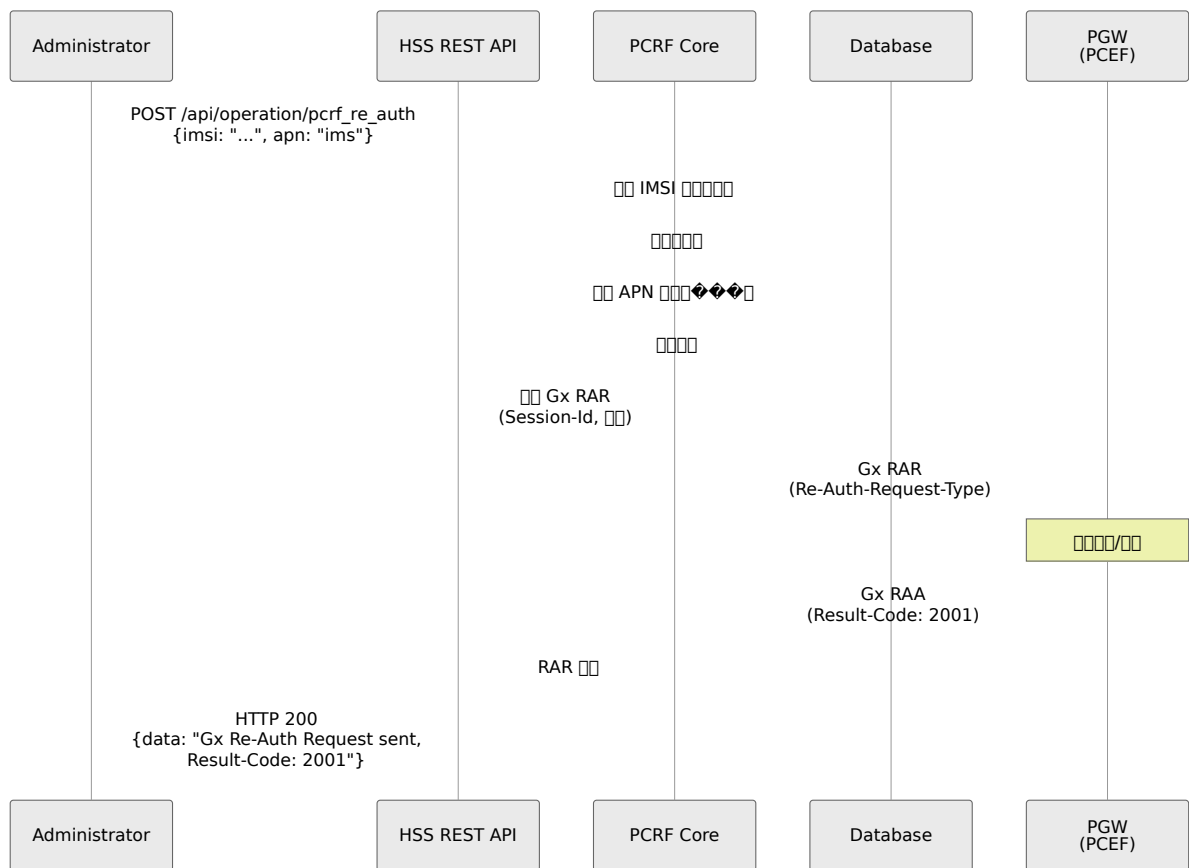
4 PDN



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"
}
```

失敗 HTTP 400

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

API

PCRF は APN と QoS を REST API で取得

PCRF は APN と QoS を取得し、QCI、AMBR、ARP を APN と PDN
PGW に Gx として RAR を送信する

図 1

図 1-1

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

1. APN 取得

APN から IP 取得

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

API 명세

QoS API

```
PATCH /api/apn/qos_profile/{id}
PUT /api/apn/qos_profile/{id}
```

예시 - QoS 프로파일 생성

```
# QoS ID 1 생성
curl -X PATCH https://hss.example.com:8443/api/apn/qos_profile/1 \
-H "Content-Type: application/json" \
-d '{
  "apn_qos_profile": {
    "apn_ambr_dl_kbps": 150000,
    "apn_ambr_ul_kbps": 75000
  }
}'
```

시퀀스

1. QoS 프로파일 생성
2. QoS 프로파일 APN PDN
3. PGW Gx RAR
4. PGW QoS
5. API

100 QoS ID 1 "internet" APN 100 API 150 Mbps / 75 Mbps

APN QoS

API

```
DELETE /api/apn/identifier/{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
 - QCI
 - Max-Requested-Bandwidth AVP
 - Flow-Description AVP

4.1.1

消息	消息	消息	消息
2001	消息	DIAMETER_SUCCESS	消息
5001	消息	消息	IMSI 消息
5002	消息	消息	PDN 消息/消息
5063	消息	消息	IMS 消息

4.1.2

4.1.2.1

PCRF 消息

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

4.1.2.2

PCRF 消息

- 消息 Gx 消息/消息
- 消息 **TFT**消息
- 消息
- 消息

消息 TFT 消息

□□□□

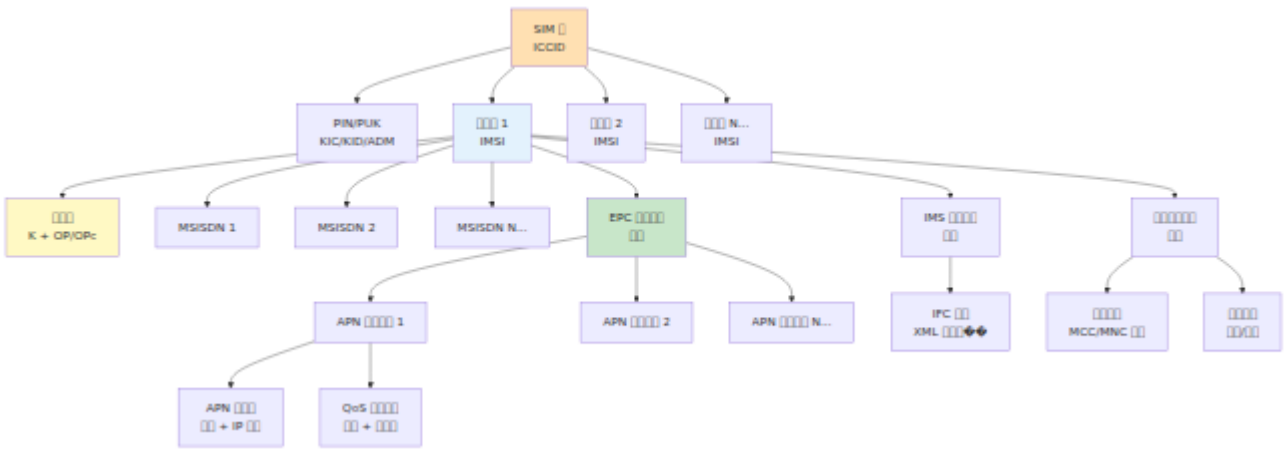
- Diameter □□ - □□□□□□
- API □□ - □□□ API □□
- □□ - □□ HSS □□
- □□□□ - □□□□ Diameter AVP □□□

OmniHSS 数据库结构

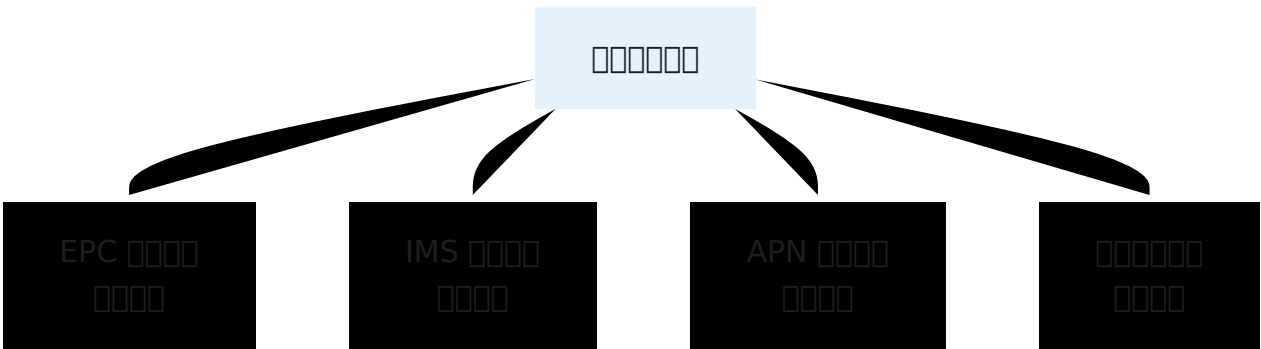
← 数据库

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 EPC 数据库的层级关系和主要数据表。

パラメータ

パラメータ名	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" または "packet_and_circuit"
<code>tracking_area_update_interval_seconds</code>	TAU seconds	54 秒

API EPC プロファイル

```
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 プロファイル

パラメータ:

- `ue_ambr_dl_kbps`: 10 Mbps (10,000 Kbps)
- `ue_ambr_ul_kbps`: 5 Mbps (5,000 Kbps)

単位:

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

50:

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

100:

- 1 Gbps (1,000,000 Kbps)
- 500 Mbps (500,000 Kbps)

IMS

IMS IFC

IFC

IFC S-CSCF XML

50:

- `{{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>"
  }
}'
```

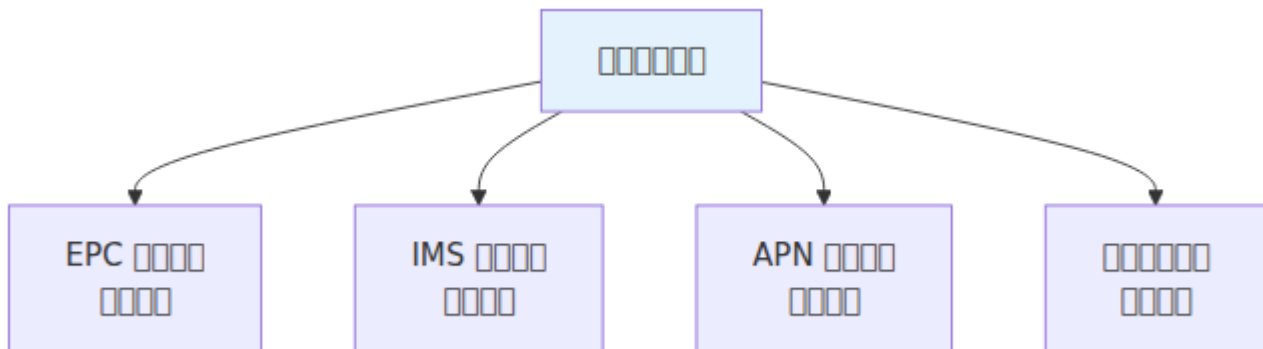
IFC 概要

```
<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 何者

APNは、ネットワークを識別するための文字列です。

APN 種類



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 用
- `mms` - MMS 用
- `vzwadmin` - VZW 用

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、ネットワークの性能を最適化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 📡

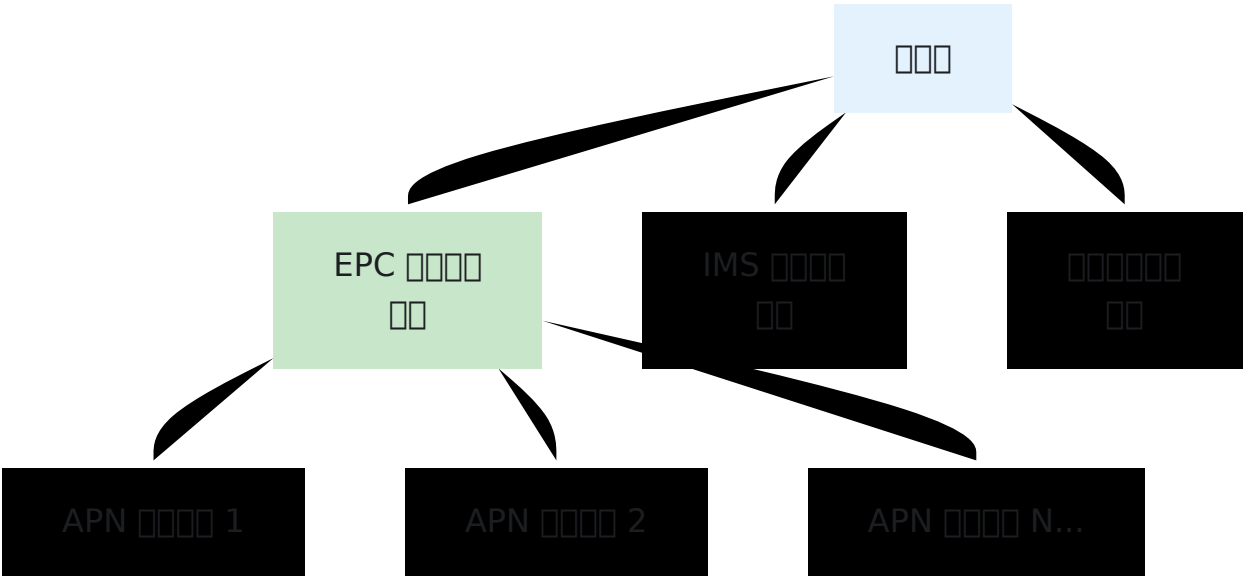
📡 APN 📡 ID 📡 EPC 📡 ID 📡 APN 📡 EPC 📡

□□□□□□

□□□ □□□□□ □□□□□□

□□□□□□

□□□□□□□□



API 操作

```
# 创建 EPC 和 IMS 配置
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
    }
  }'
```

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

配置项

配置项

1. 配置项 - 配置项名称和值
2. 配置项 - 配置项名称和值
3. 配置项 - 配置项名称
4. 配置项 - 配置项名称和值

Network Architecture

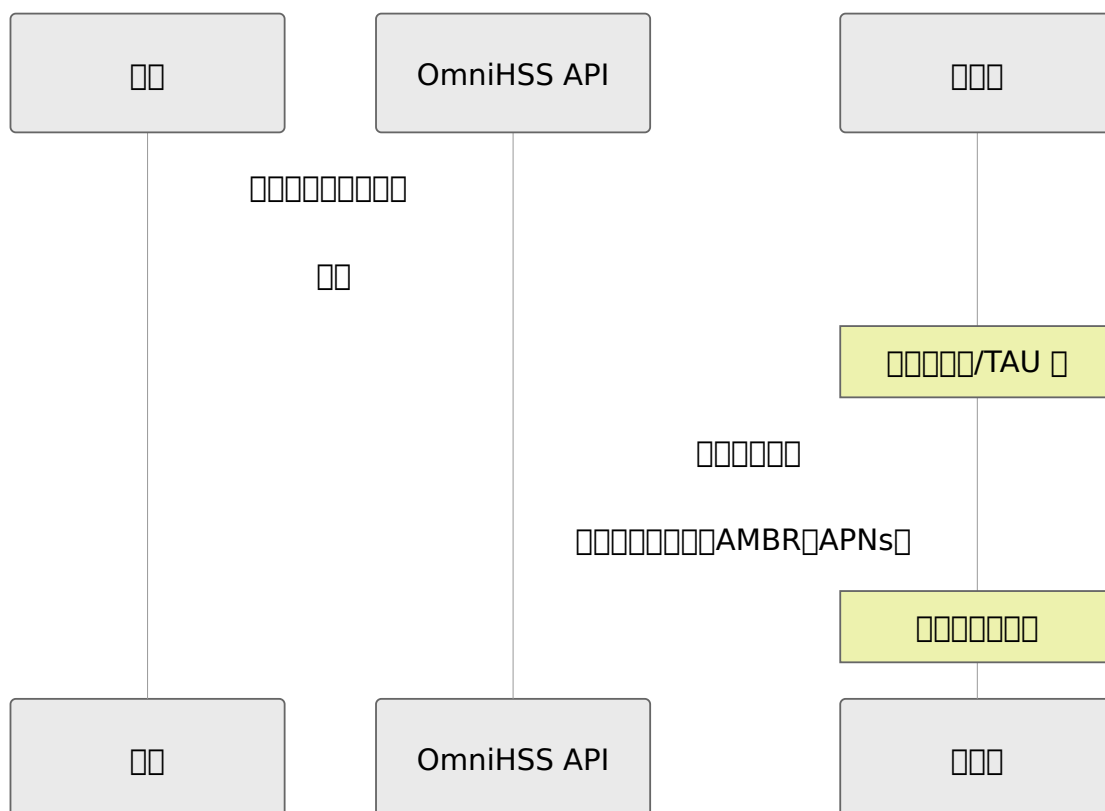
[Network] - [Core] - [Access]

Network:

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

Core Network

Core Network Components



Core: Core Network Components

- Core Network Components
- Core Network Components
- IMS Core Network Components

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

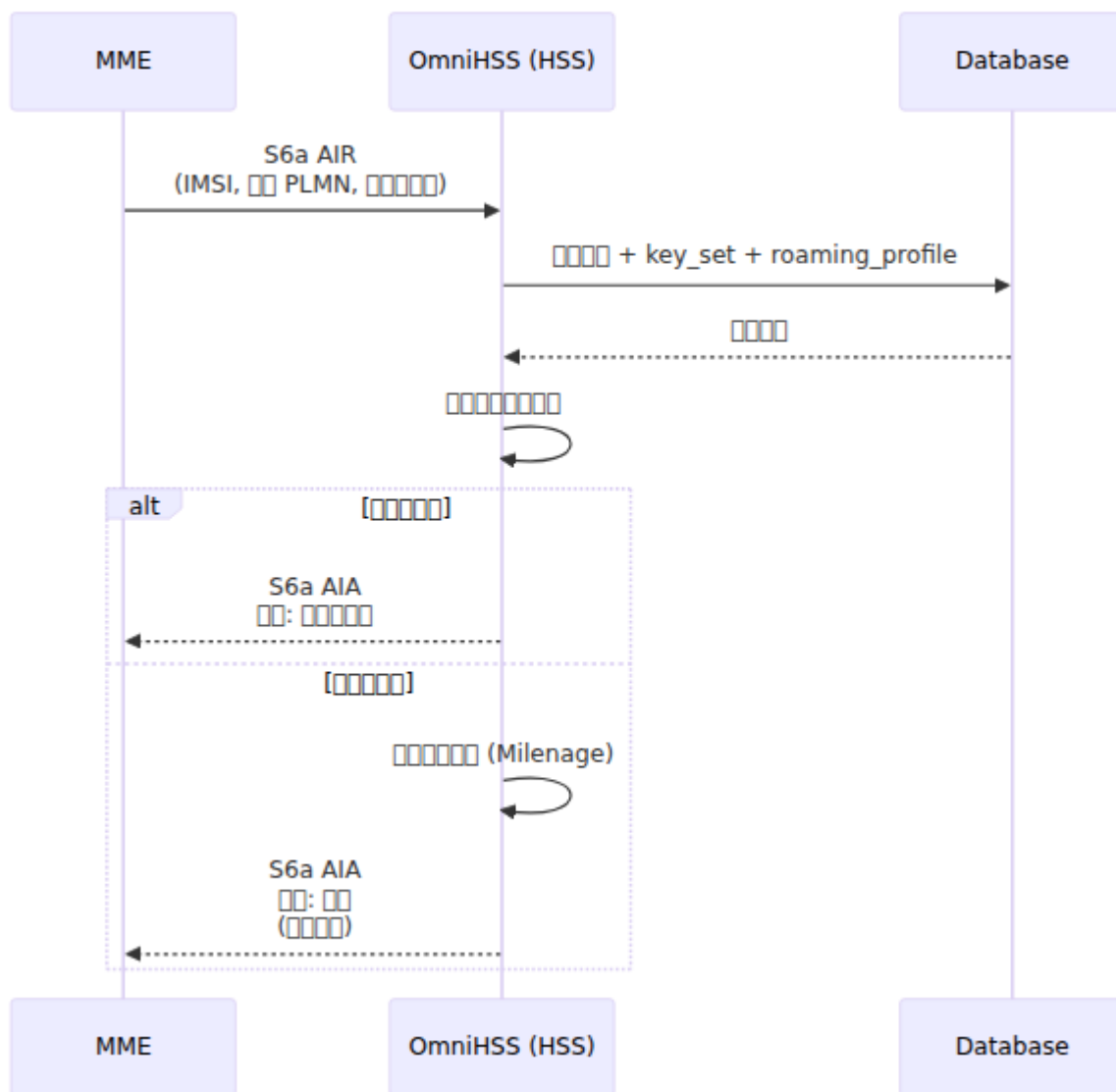
設定

OmniHSS 設定 Diameter 設定

S6a 設定 (LTE/EPC)

設定 (AIR/AIA)

MME 設定

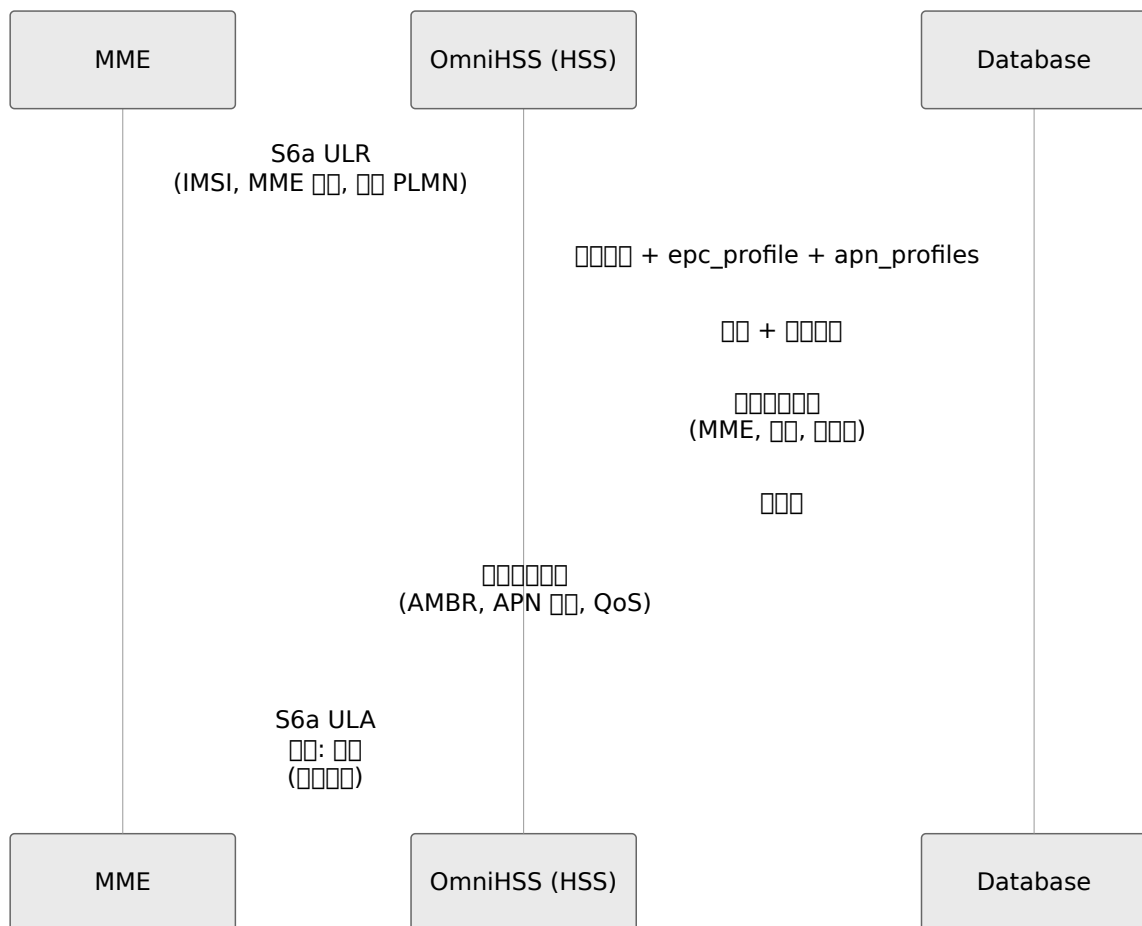


AVP:

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

ULR/ULA

MME HSS

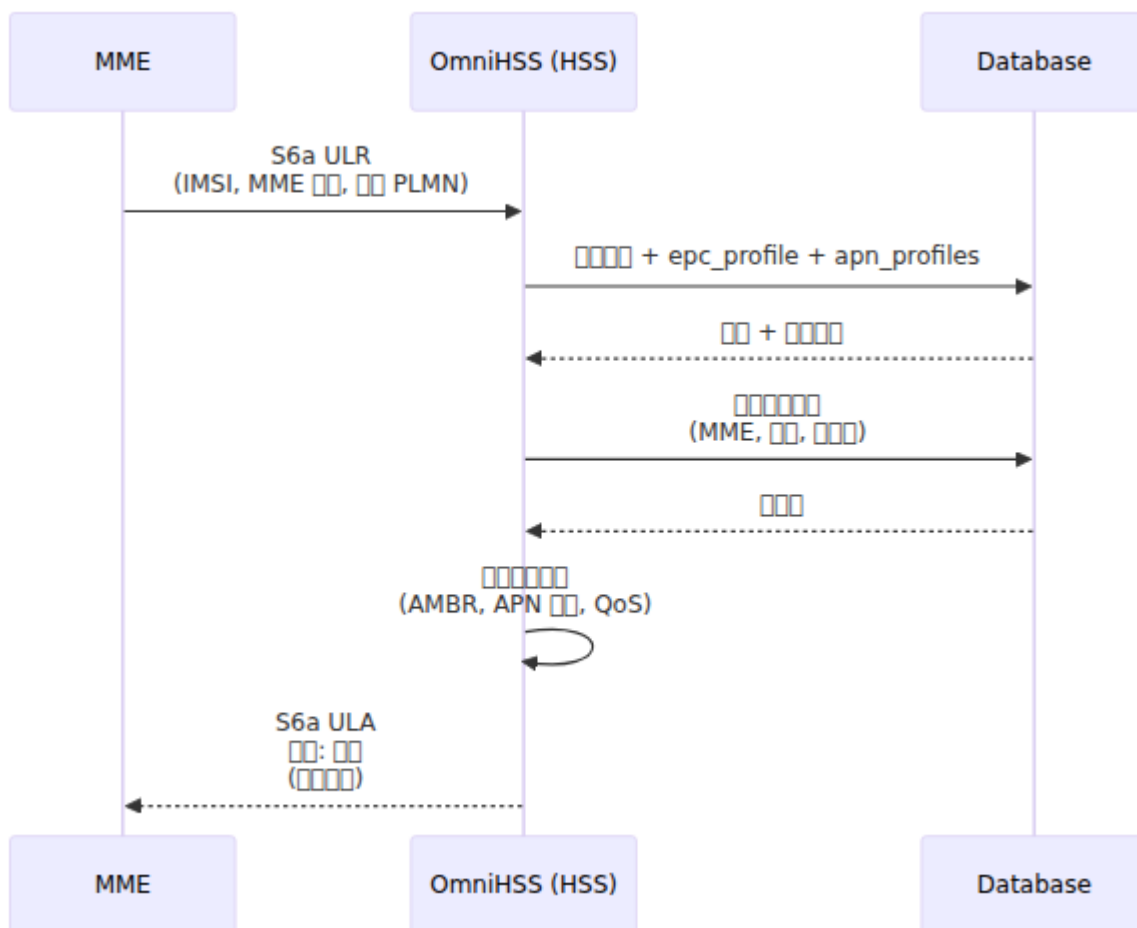


AVP:

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

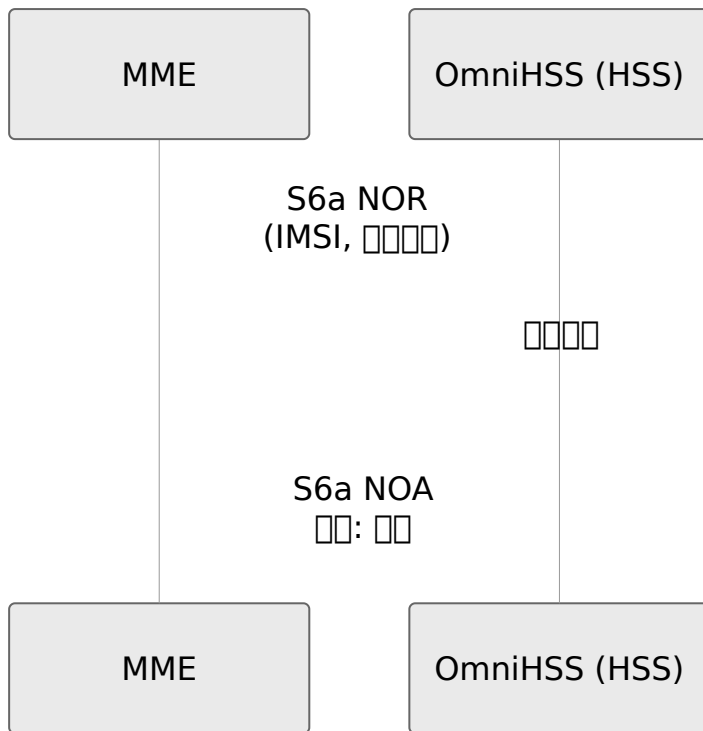
UE ID (PUR/PUA)

MME ID HSS ID



Database (NOR/NOA)

MME ID HSS ID

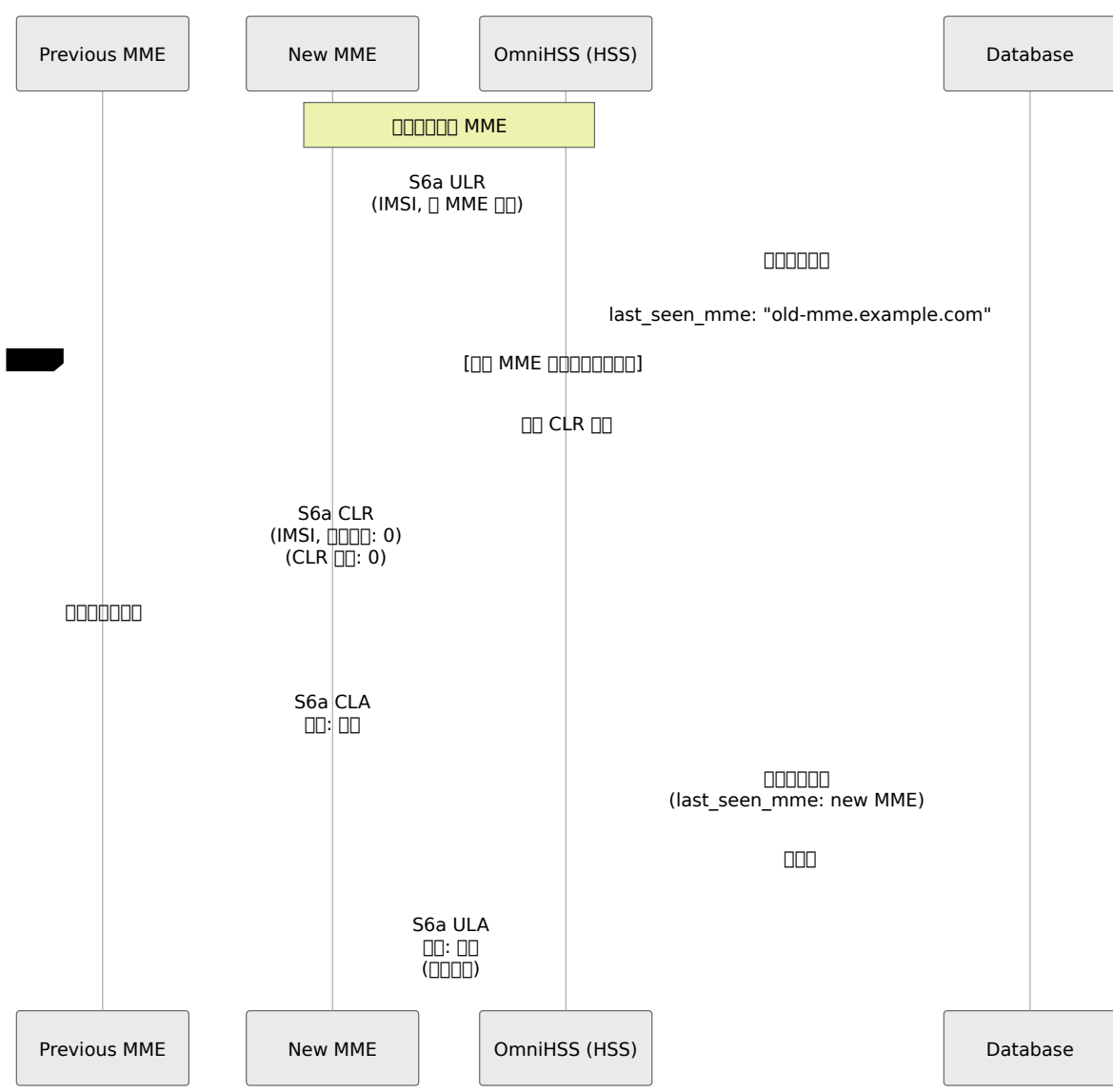


□□□□□□ (CLR/CLA)

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

□□ **CLR (MME □□)**

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

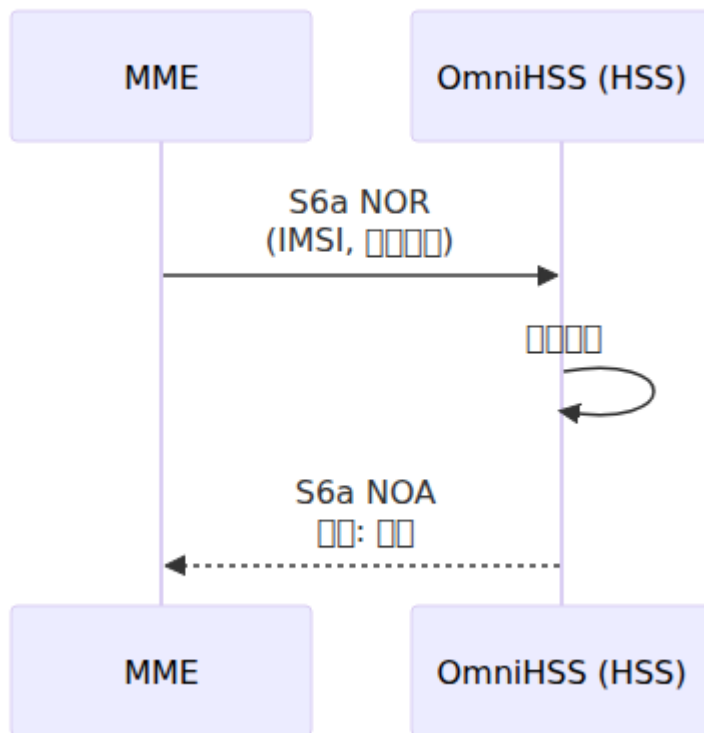


AVP (CLR):

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

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-Flags AVP 番号

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

CLR-Flags 番号:

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

IMSI 番号

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

1: 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**s□□□□ **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
- □□ **MSISDNs** □□ IMSI □□□□□□□□□□

- IMSI 000 MME 00000

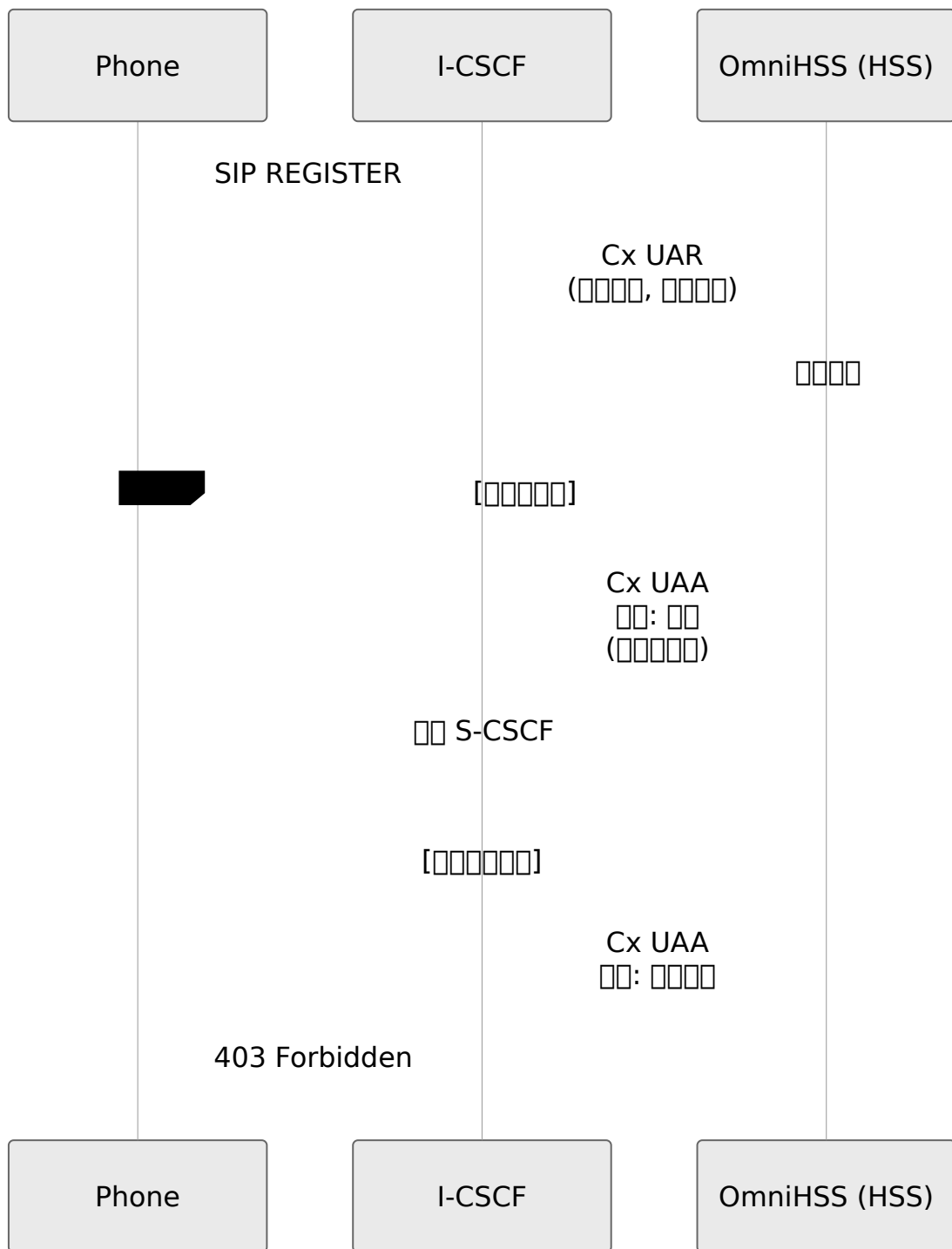
1111

1. **IMSI** 消息: CLR 消息包含 **IMSI** 消息的 MSISDN 和 **subscriber_state** 消息 (IMSI) 消息 **last_seen_mme**
2. 消息: 消息包含 MME 消息的 CLR 消息的 **last_seen_mme** 消息
3. 消息包含 **MME** 消息的 **CLR**: 消息 **last_seen_mme** 消息 **nil** 消息包含 ULR 消息的 CLR 消息
4. 消息: 消息 CLR 消息 ULR 消息包含 **Subscription-Data** AVP 消息 MME 消息 消息
5. 消息: CLR 消息包含 MME 消息 ULA 消息 MME 消息 CLA 消息
6. **CLA** 消息: OmniHSS 消息 CLA 消息包含 398 消息 **:discard** 消息 HSS 消息

Cx 11 (IMS)

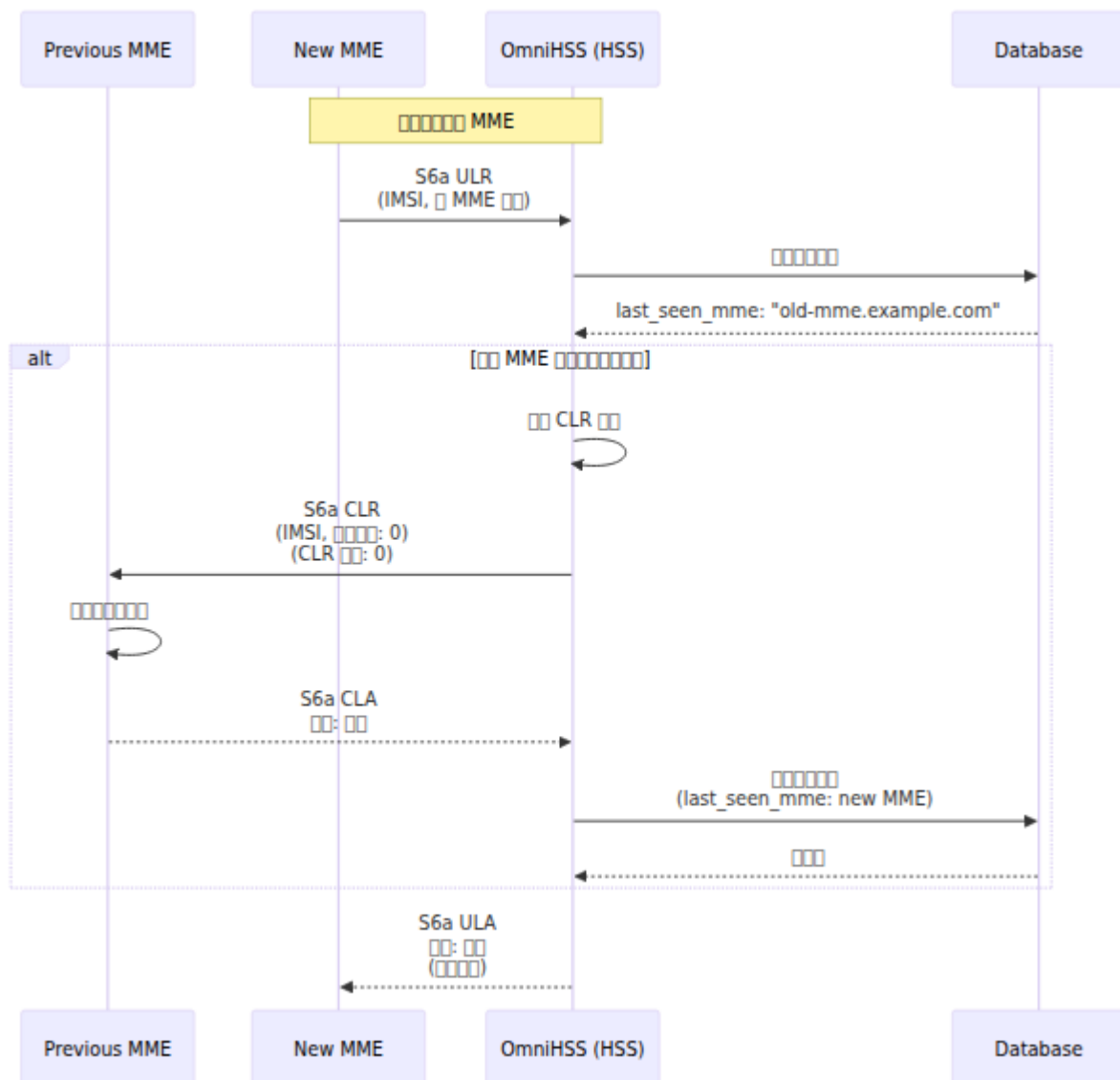
□□□□□□ (UAR/UAA)

I-CSCF



Sequence Diagram (SAR/SAA)

S-CSCF SAA/IMS SAA

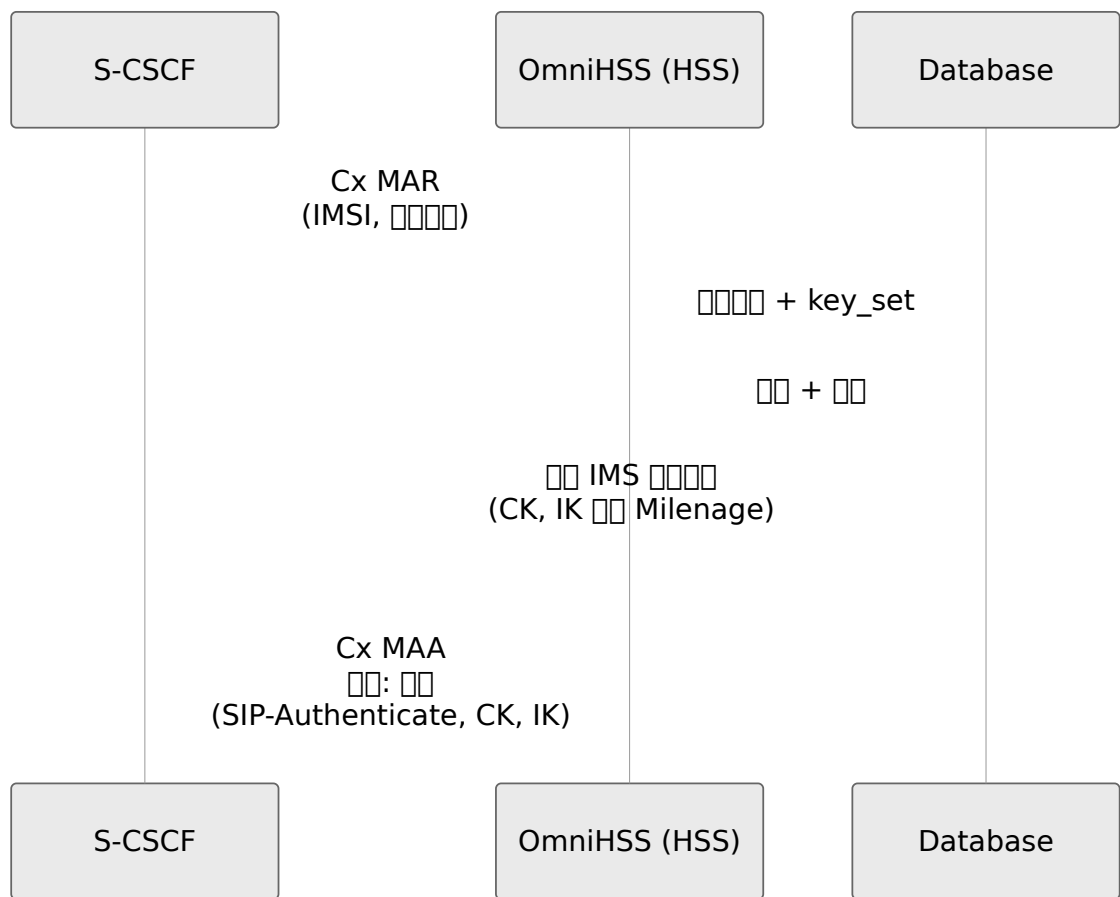


IFC :

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

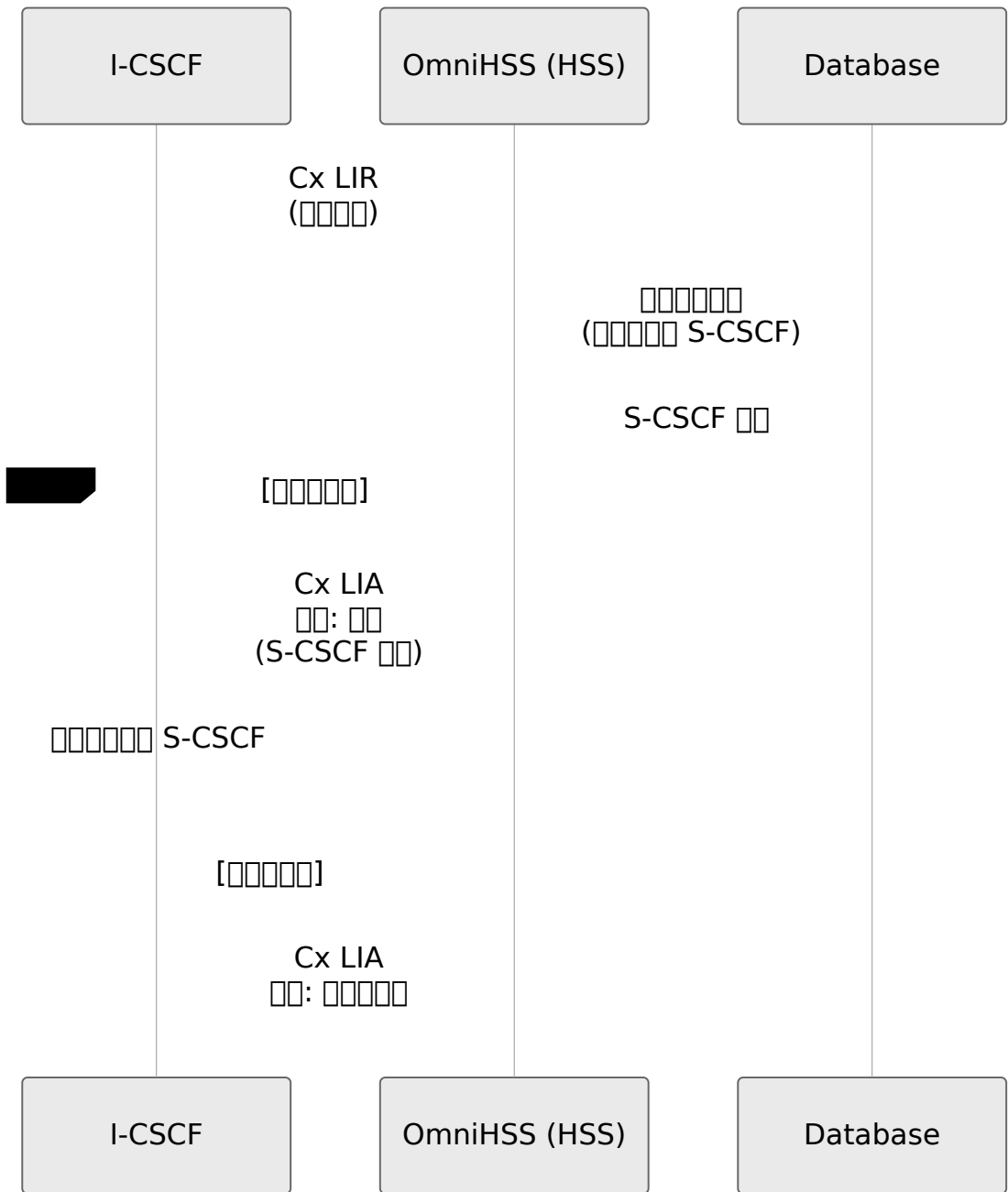
(MAR/MAA)

S-CSCF IMS



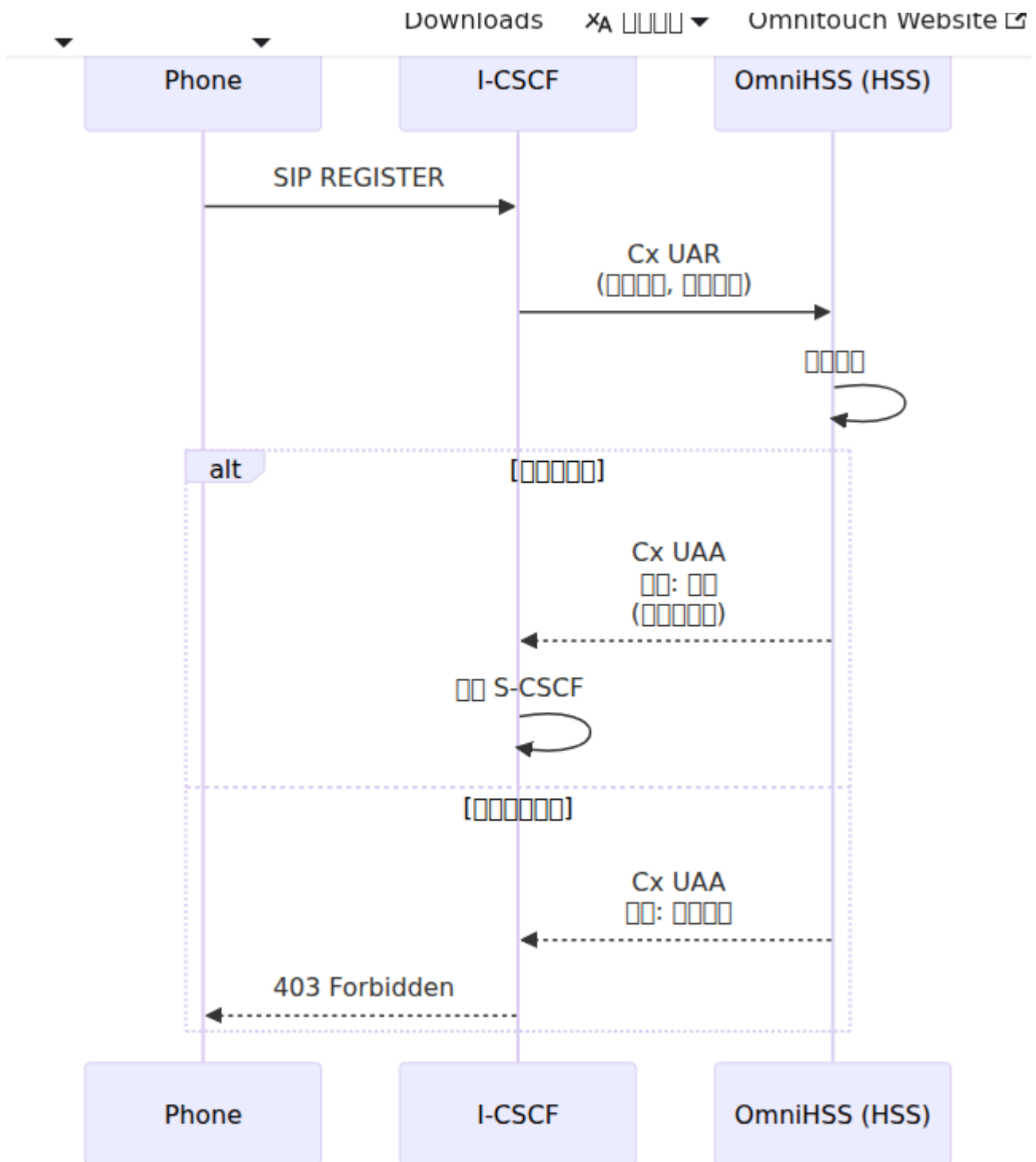
000000 (LIR/LIA)

I-CSCF 0000 S-CSCF 00000000

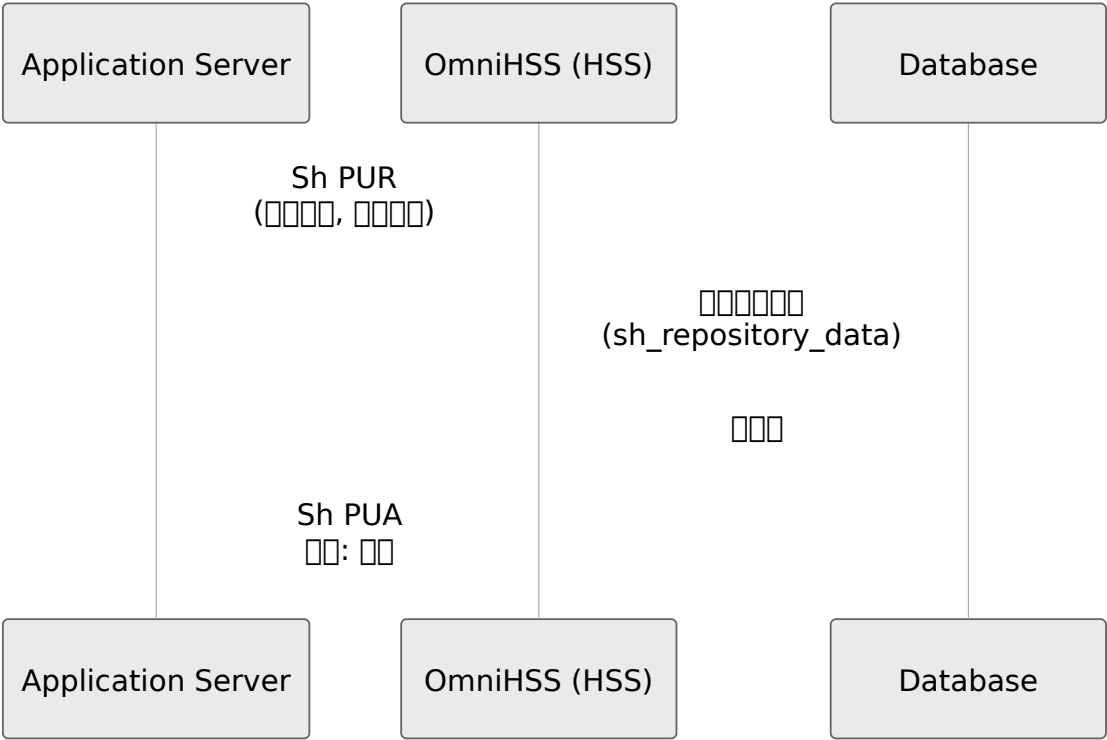


Sh (IMS)

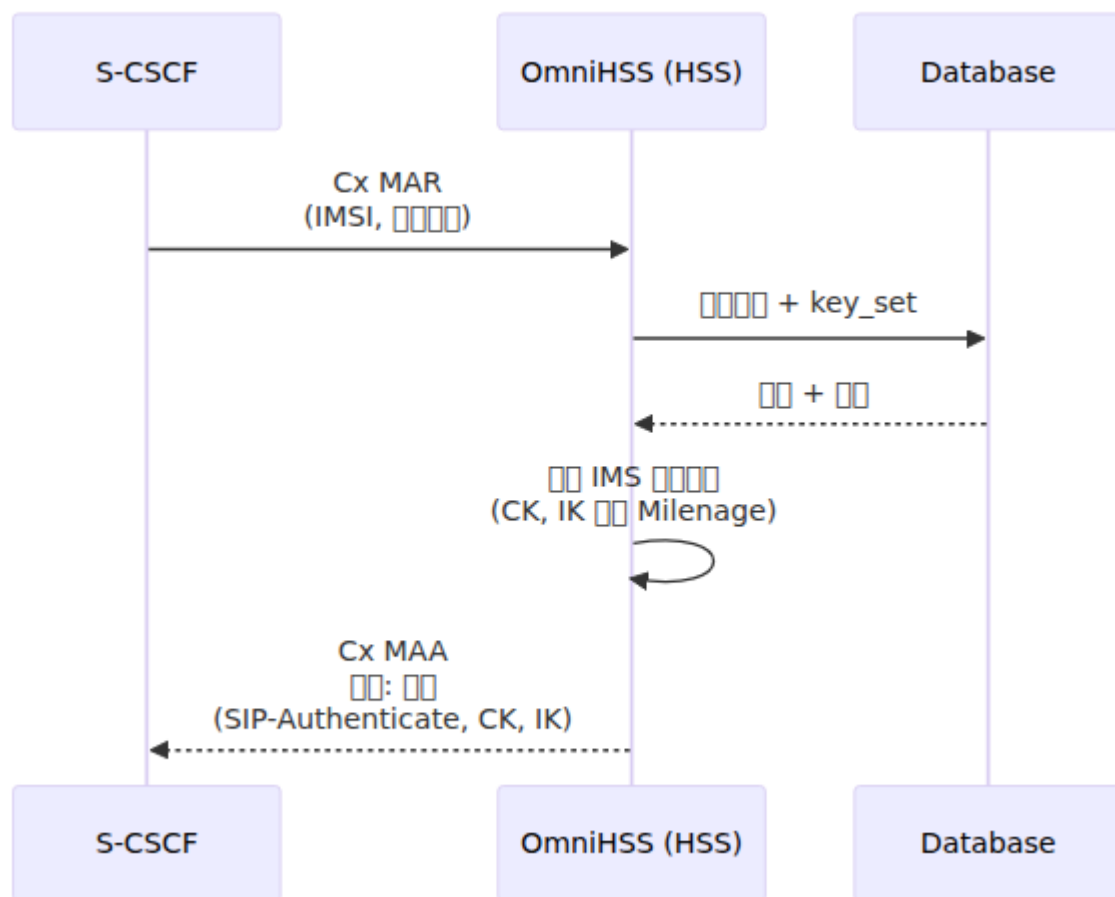
(UDR/UDA)



(PUR/PUA)



(SNR/SNA)

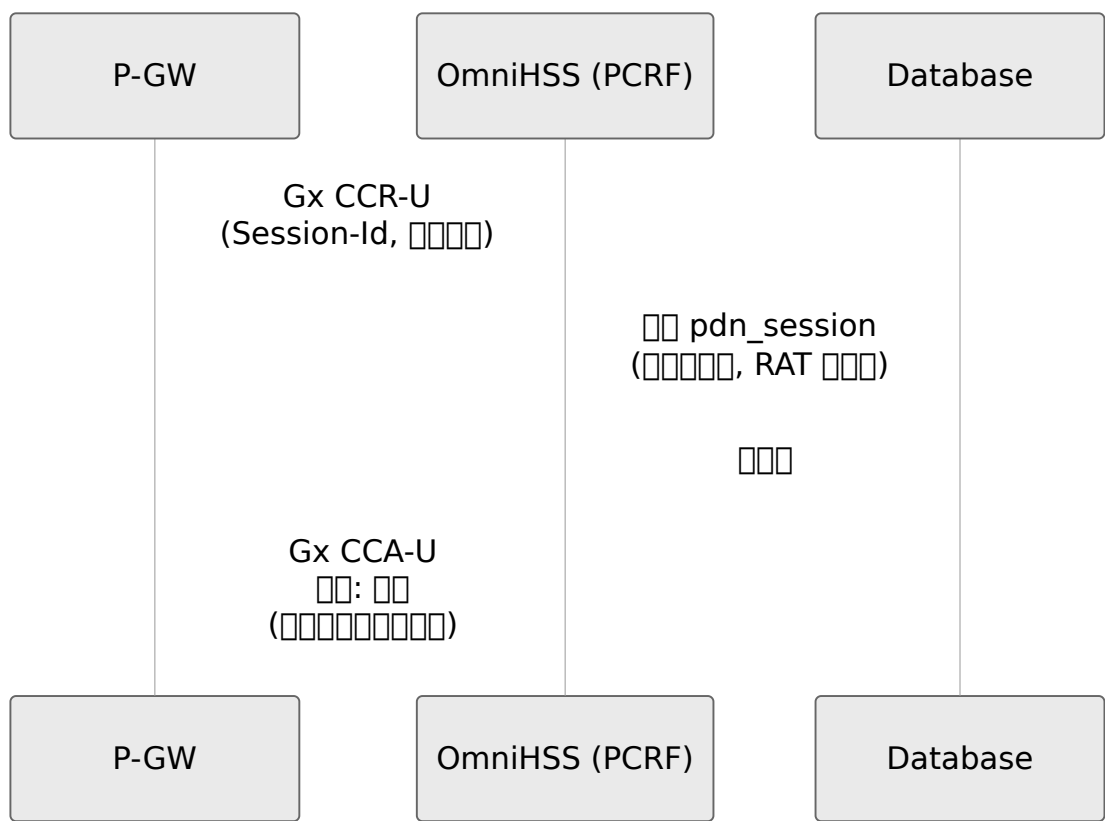


AVP:

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

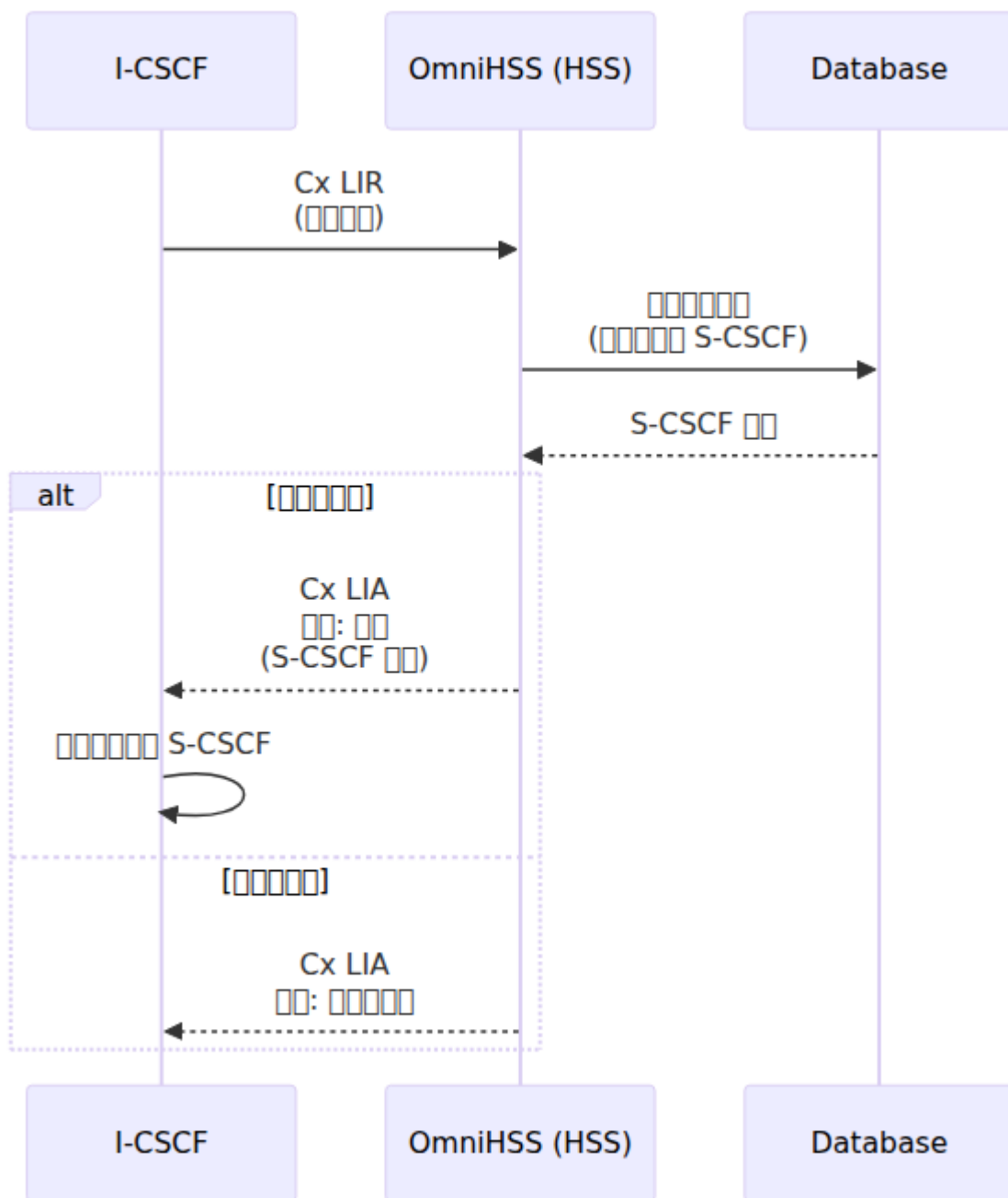
Authentication - (CCR-U/CCA-U)

P-GW authentication



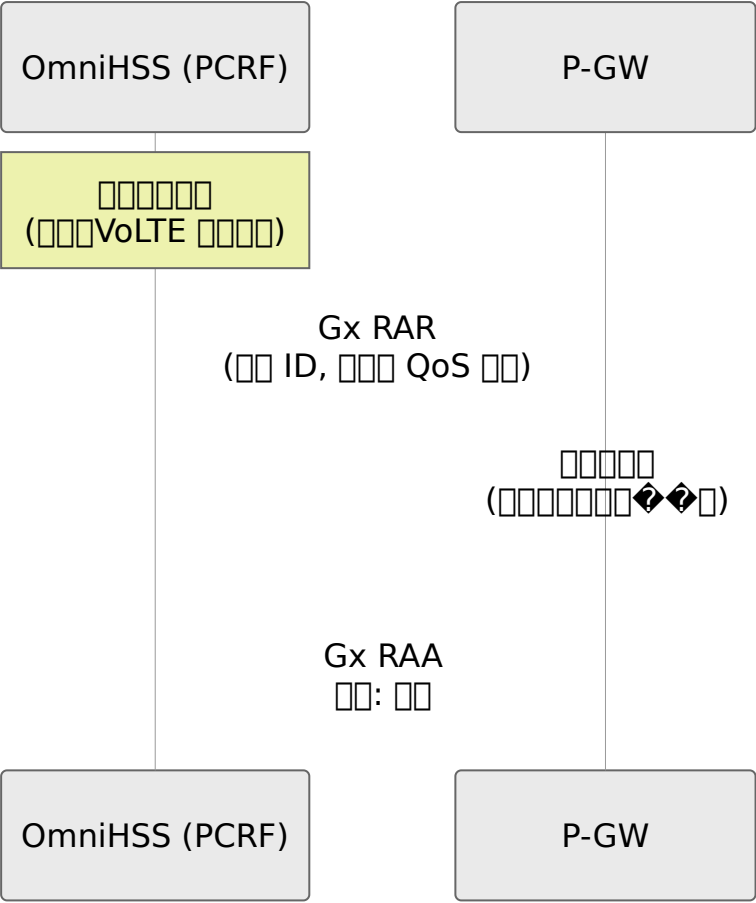
0000000 - 00 (CCR-T/CCA-T)

P-GW 00 PDN 000000



(RAR/RAA)

OmniHSS (PCRF) P-GW



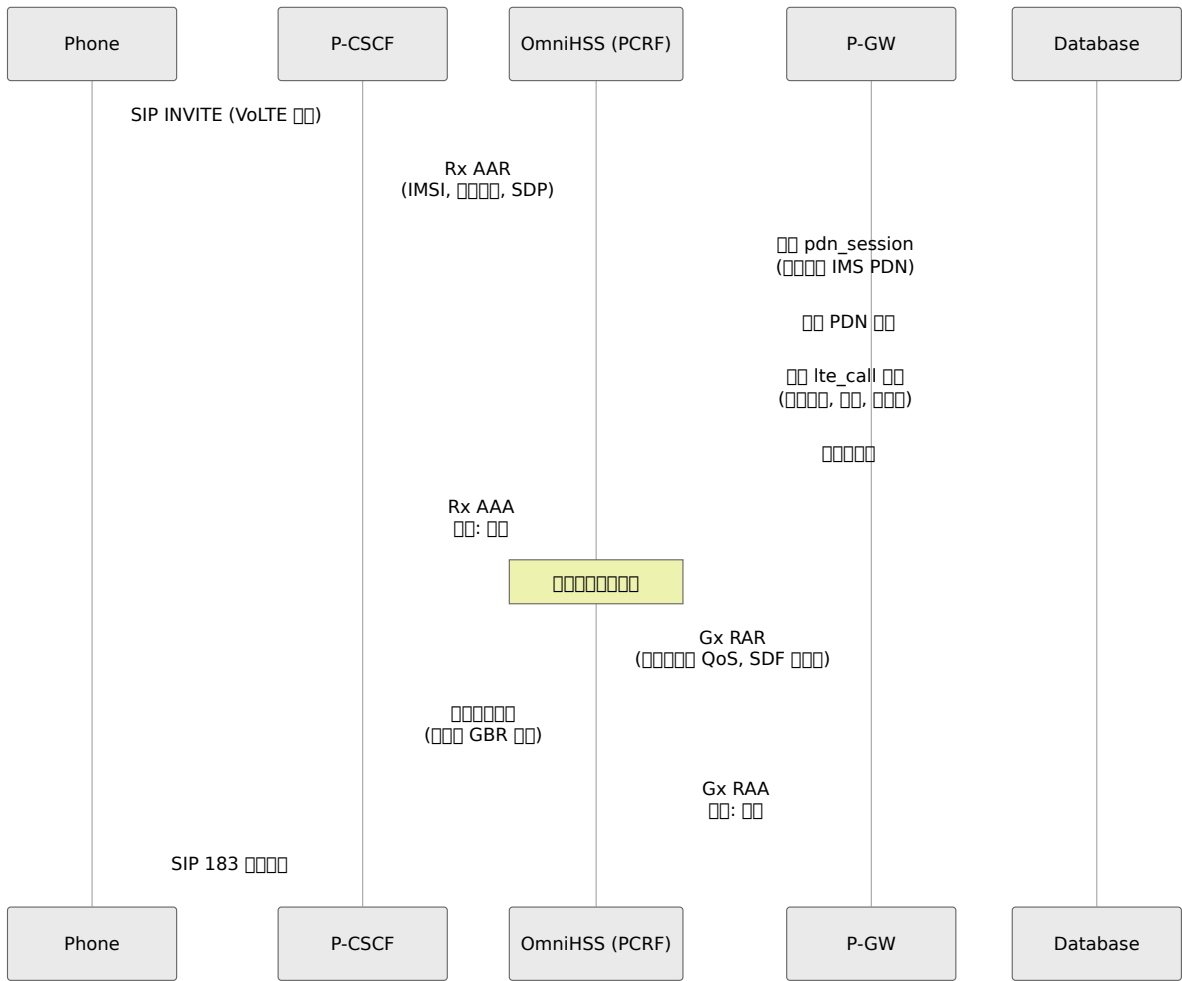
Rx (IMS)

OmniHSS Rx PCRF IMS

PCRF VoLTE

AA (AAR/AAA)

P-CSCF IMS



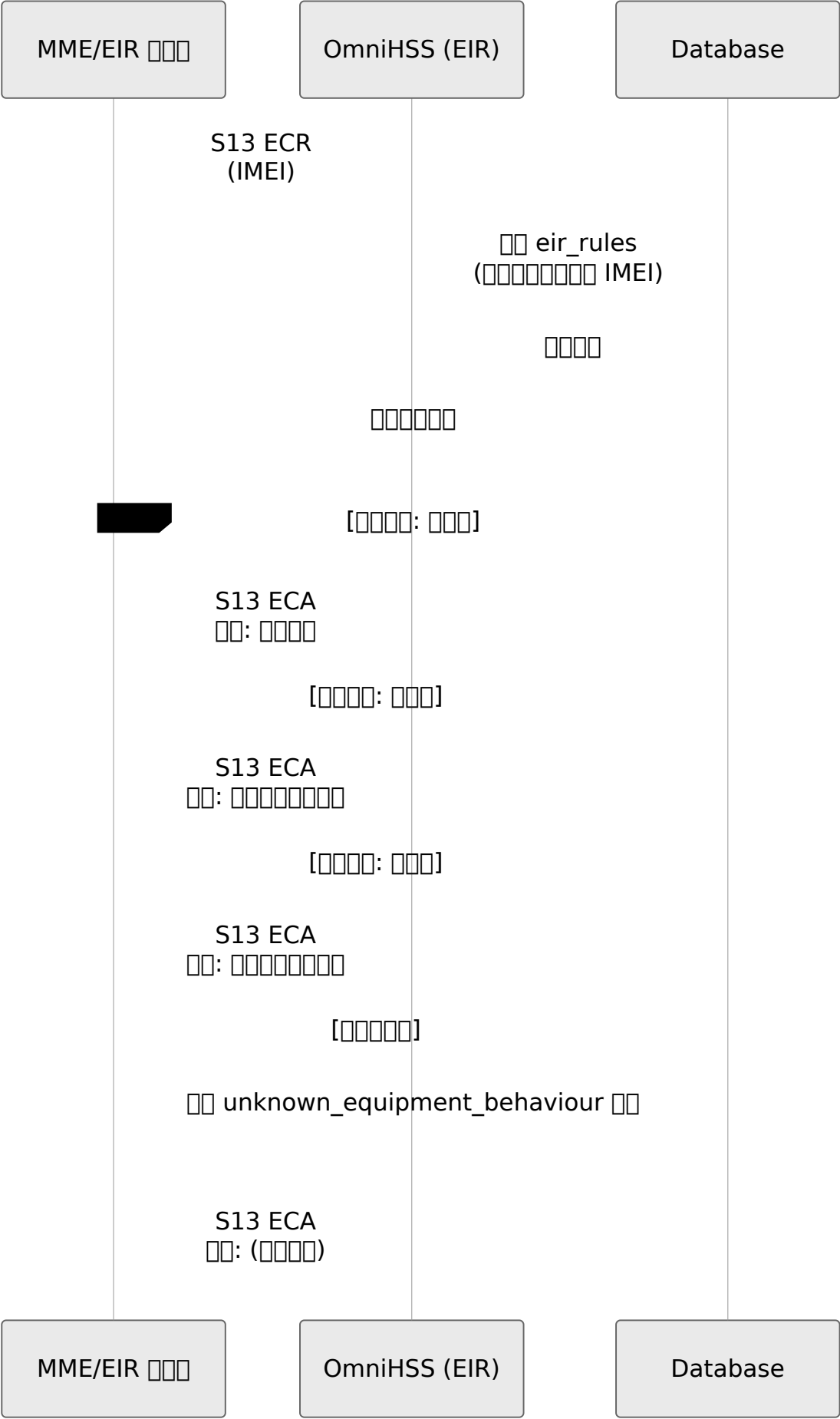
IMS:

- SDP (IP/Port)
- IP/Port (IP/Port)
- SDF (IP/Port)
- Gx RAR (IP/Port)

IMS (STR/STA)

P-CSCF IMS (STR/STA)

00 EIR 00000 MME000000000

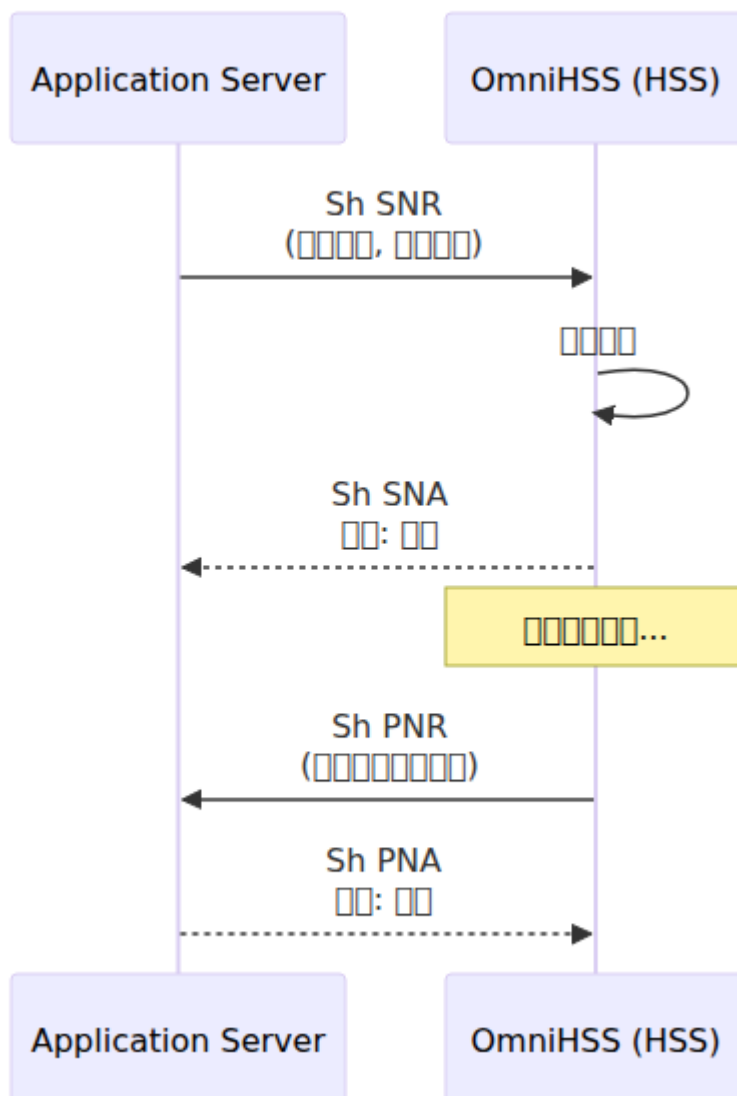


□□□□□:

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

□□□□□□□: VoLTE □□

□□□ VoLTE □□□□□□□□□□□□



Network Elements

Network Element (S6a AIR)

Parameters:

- Authentication (Ki, OPC, AMF)
- SQN (Sequence Number)
- Subscription Information

Network Element (S6a ULR)

Parameters:

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

IMS Network Element (Cx SAR)

Parameters:

- IMS (IP Multimedia Subsystem) Information
- IFC (Initial Filter Criteria) XML
- S-CSCF (Serving Call Session Control Function) Information
- MSISDN (Mobile Station International Subscriber Number)

PDN Network Element (Gx CCR-I)

Parameters:

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

← □□□□□

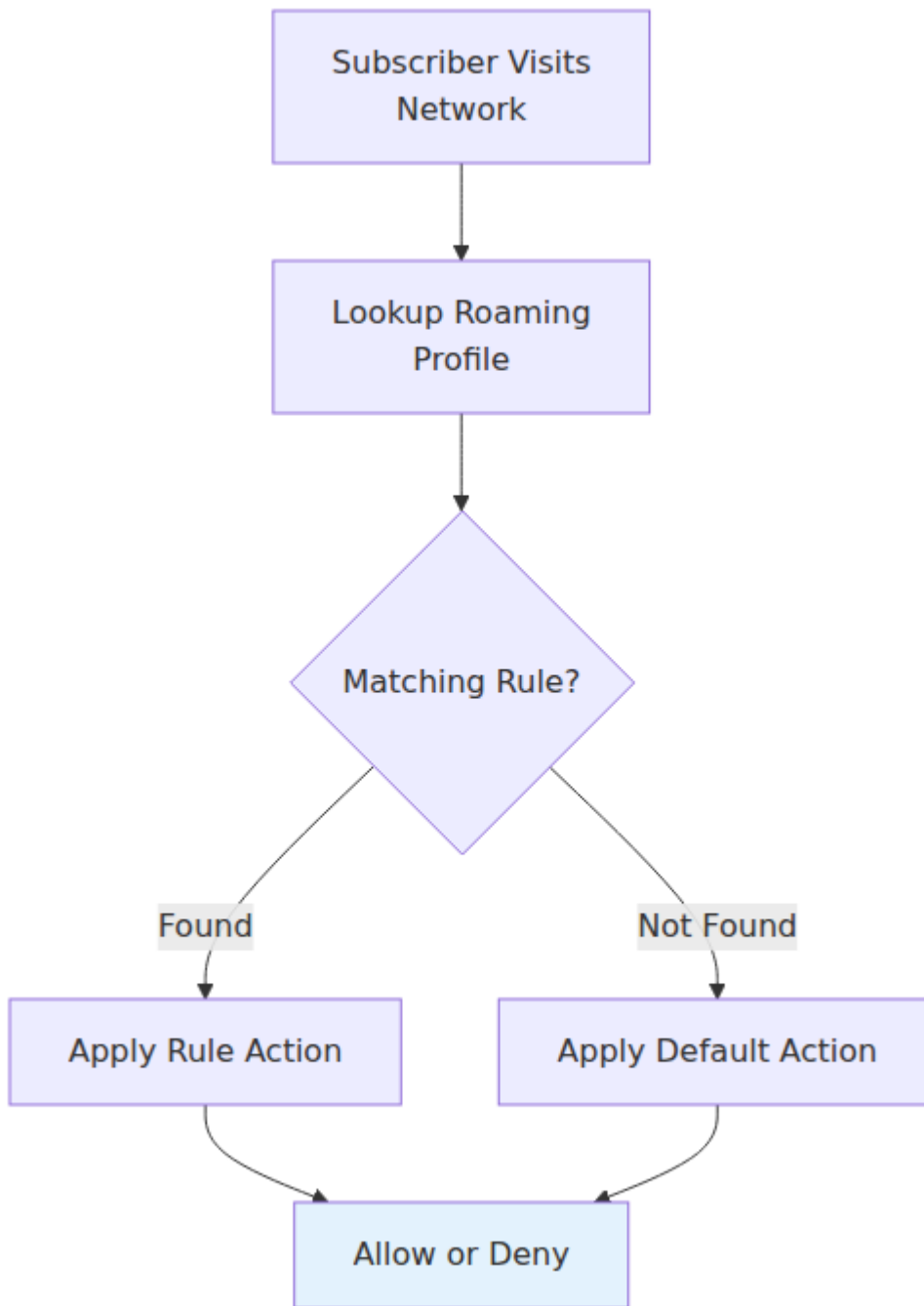
OmniHSS

←

11

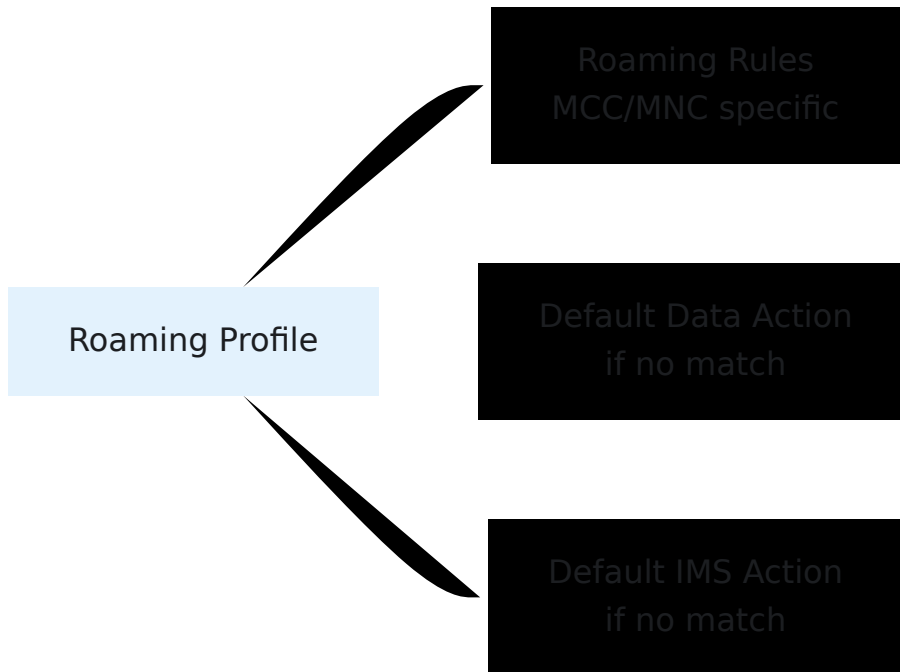
OmniHSS IMS

□□□□□□



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" 〇 "deny"
 - `ims_action_if_no_rules_match` - "allow" 〇 "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')
```

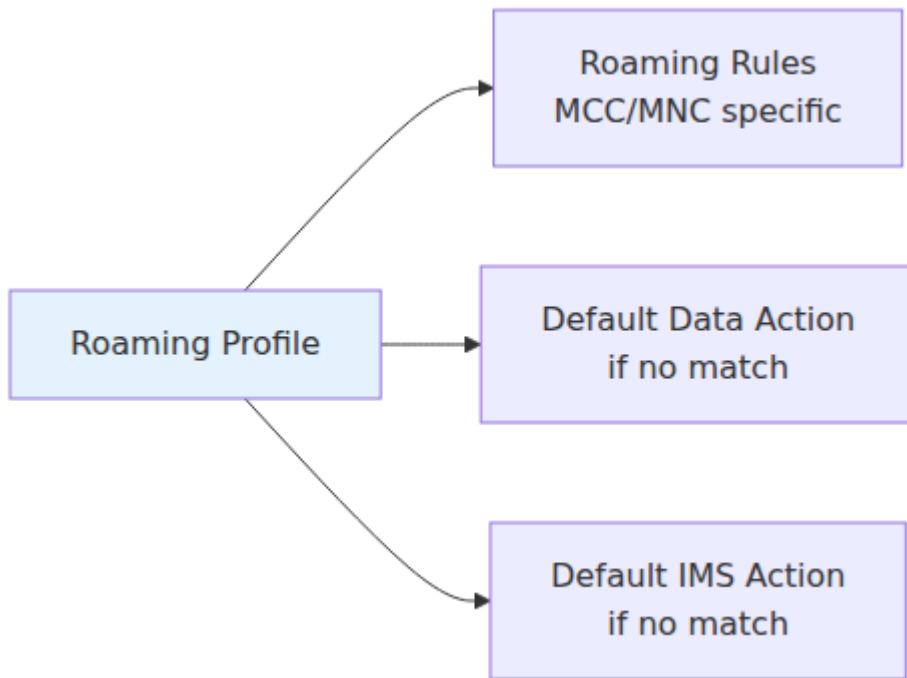
Create Profile

```
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

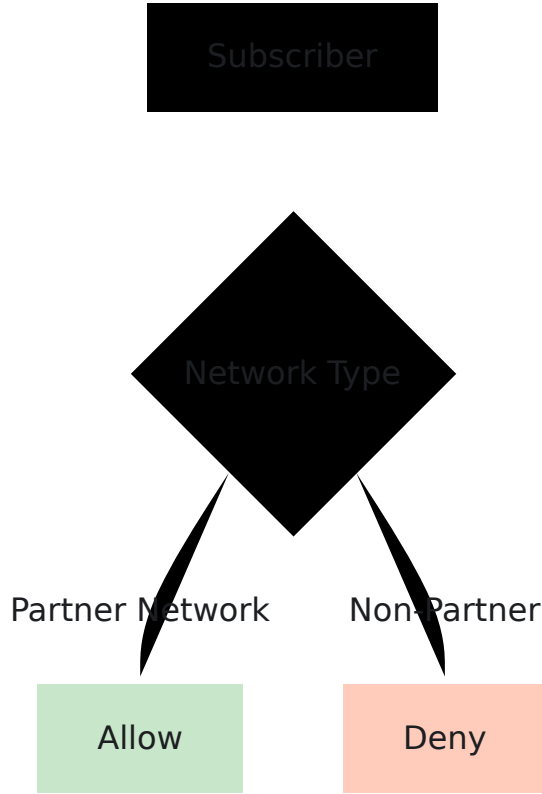


□□□

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

□□ **2**□□□□□□□□□

□□□□□□□□□□□□□□□□

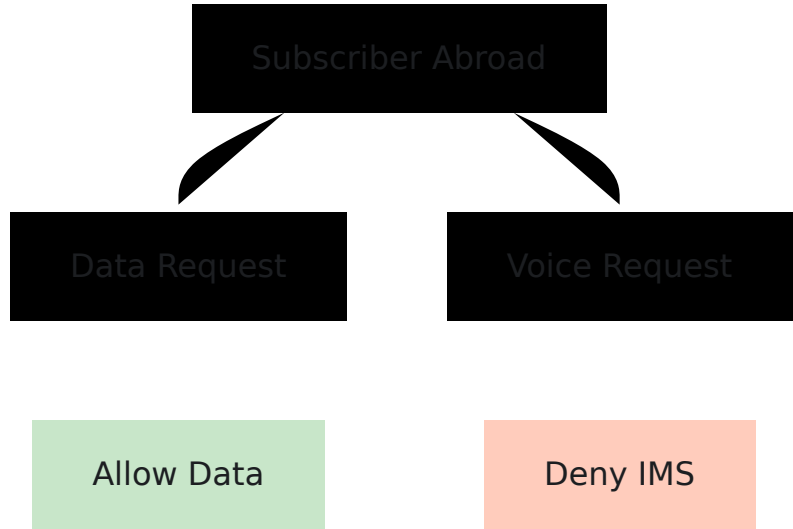


□□□

- □□□□□□□
- □□□□□□□□□□□□□□□□ MCC/MNC□

□□ **3**□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ WiFi □□□□□□□



□□□

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

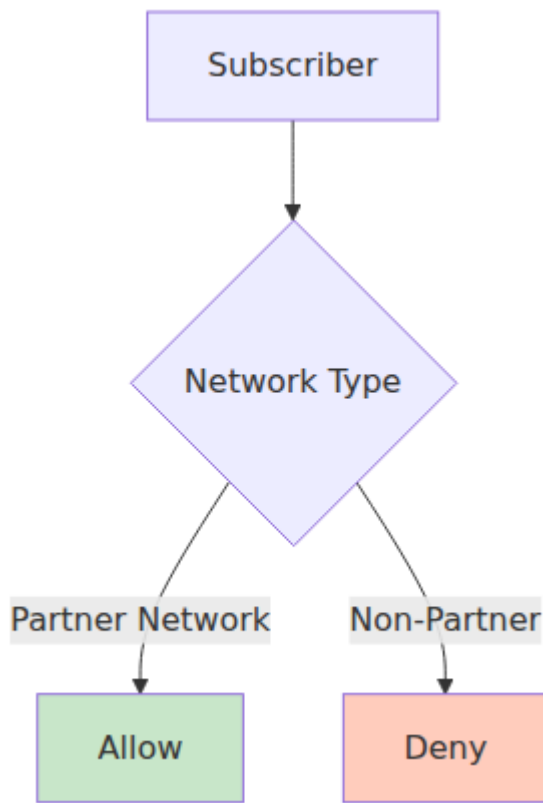
4

MME/OmniHSS

MCC/MNC

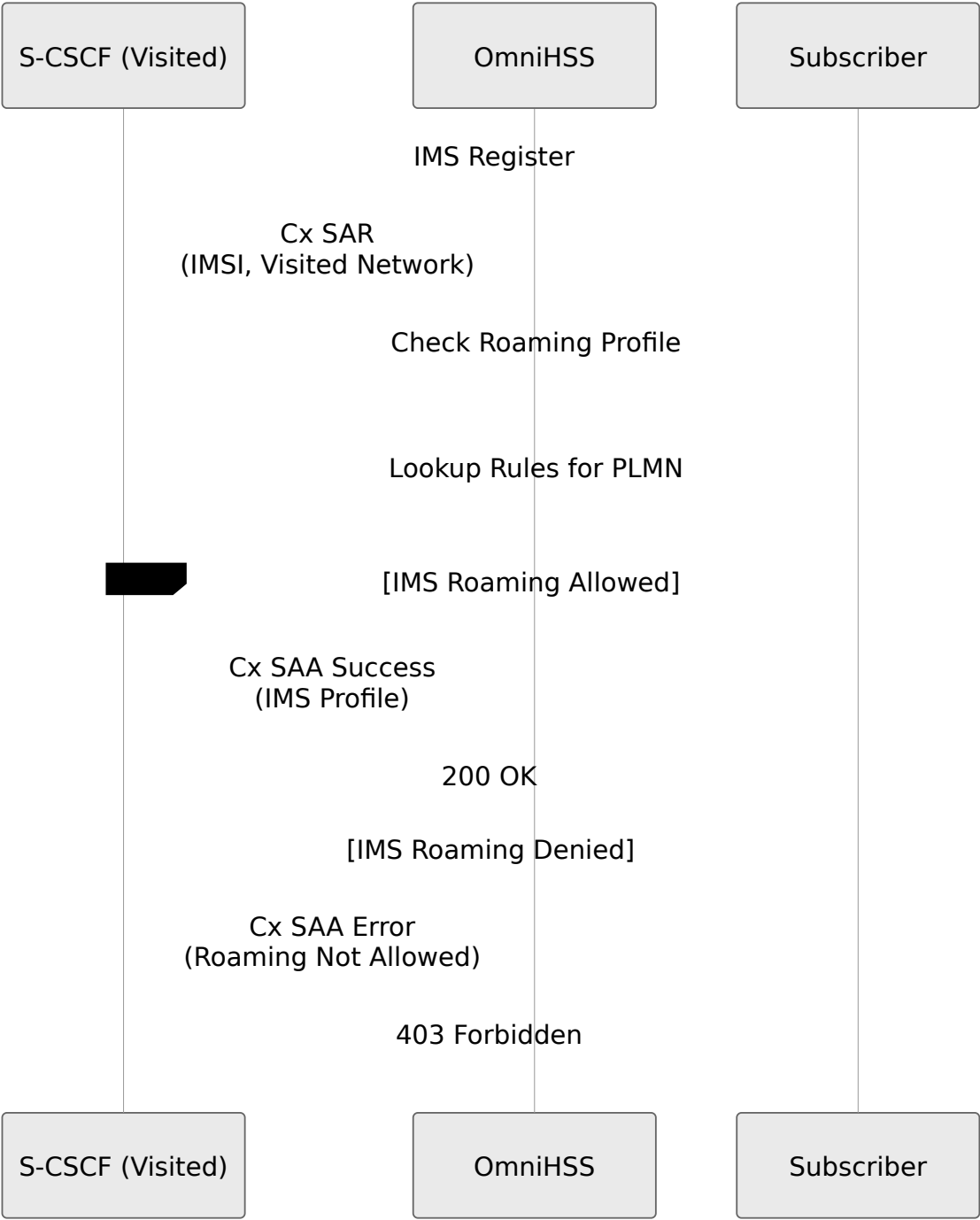
MCC

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



□□□□□□

□□□□□□□□□□□□

□□□□□□□□

- □□□□□□□□□□□□□□□□

- 检查是否漫游

检查是否漫游

- 检查是否漫游 MCC/MNC 是否匹配
- 检查是否漫游 是否匹配
- 检查是否漫游 data_action 是否

检查是否漫游 IMS

检查是否 IMS 是否

- 检查是否漫游
- 检查 data_action 是否 ims_action 是否
- 检查是否漫游 IMS 是否

检查是否漫游

检查是否漫游

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

检查是否漫游

检查是否漫游

1. 检查是否漫游 - 检查是否漫游
2. 检查是否漫游 - 检查是否漫游
3. 检查是否漫游 - 检查是否漫游
4. 检查是否漫游 - 检查是否漫游

□□□□

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

□□□□

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

OmniHSS

□□□□□□

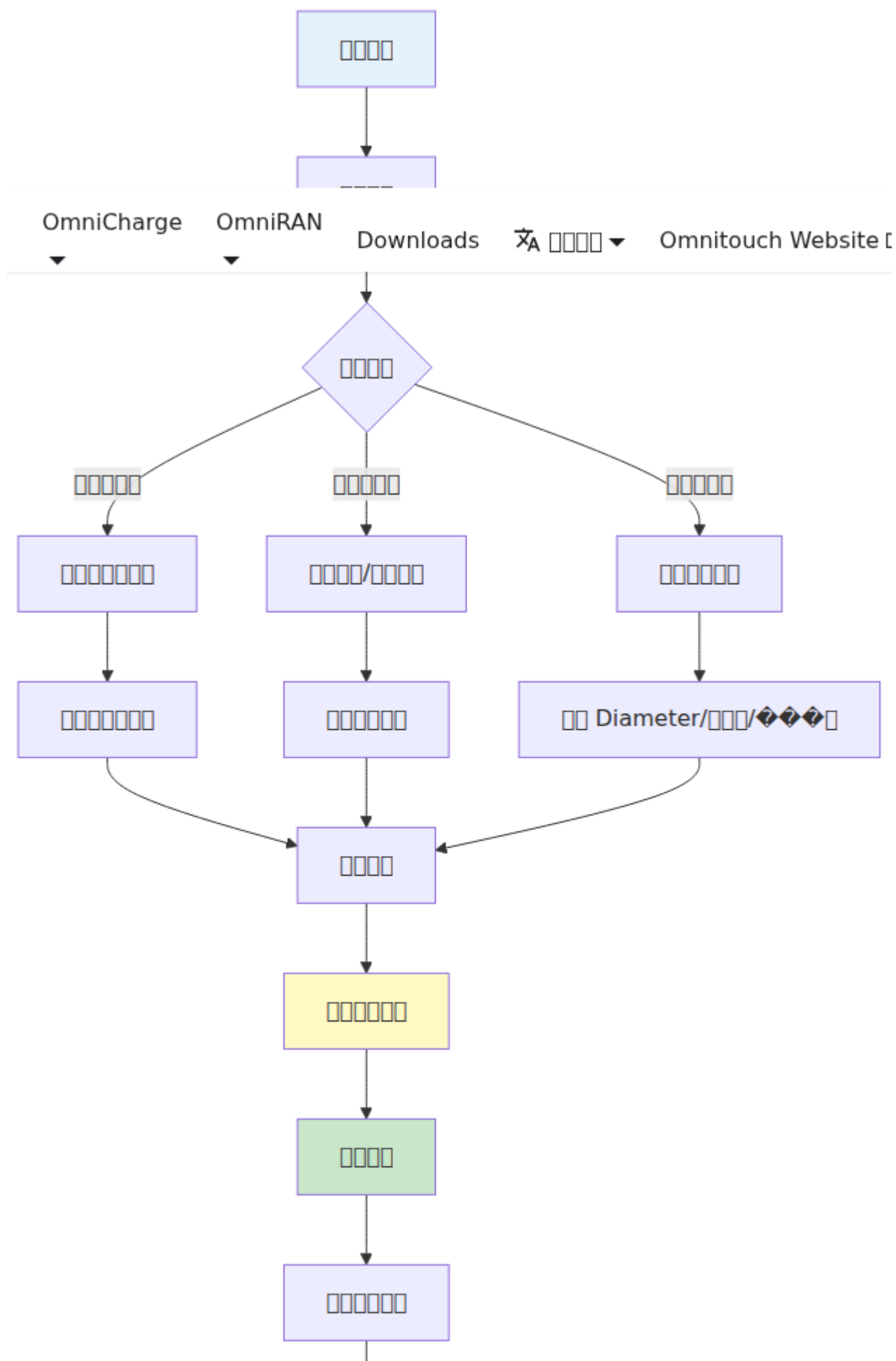
← □□□□□

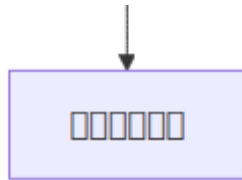
□□

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

□□□□□□

□□□□□□□





□□□□□

□□□□□□□□□□□□□□

1. □□□□□□□□□□□□□□

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

2. □□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

3. □□□□

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

4. □□□□

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

API 說明

簡介

- 系統架構圖
- “API 端點”表
- 認證與權限

系統架構圖

第 1 部分：系統架構

系統架構

- 系統架構圖
- 系統架構圖

API 端點

1. 獲取 key_set_id

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

2. 獲取 key_set_id

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

3. Ki 與 OPC 與 SIM 卡相關

API 端點

- 系統架構圖
- 系統架構圖SIM 卡相關

第 2 部分：SQL 查詢

SQL 查詢

- 0000000000000000
- 000“SQN 0000”
- 000000

000000

1. 000000000000 SQN 0
2. 00000000 SQN 000000
3. 0000000000 SQN 0

000000

- 00000000 AUTS 00SQN 00000000
- 0000000000000000 SQN 0 00000000000000

000 00 SQN 000000000000000000000000

00 **3**00000000

000

- 0000000000
- 000000000000

000000

1. 000000000000

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

2. 00 **enabled** 000 **true**

00❓❓❓00

- 00000000

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

4 EPC

-
- “ EPC ”

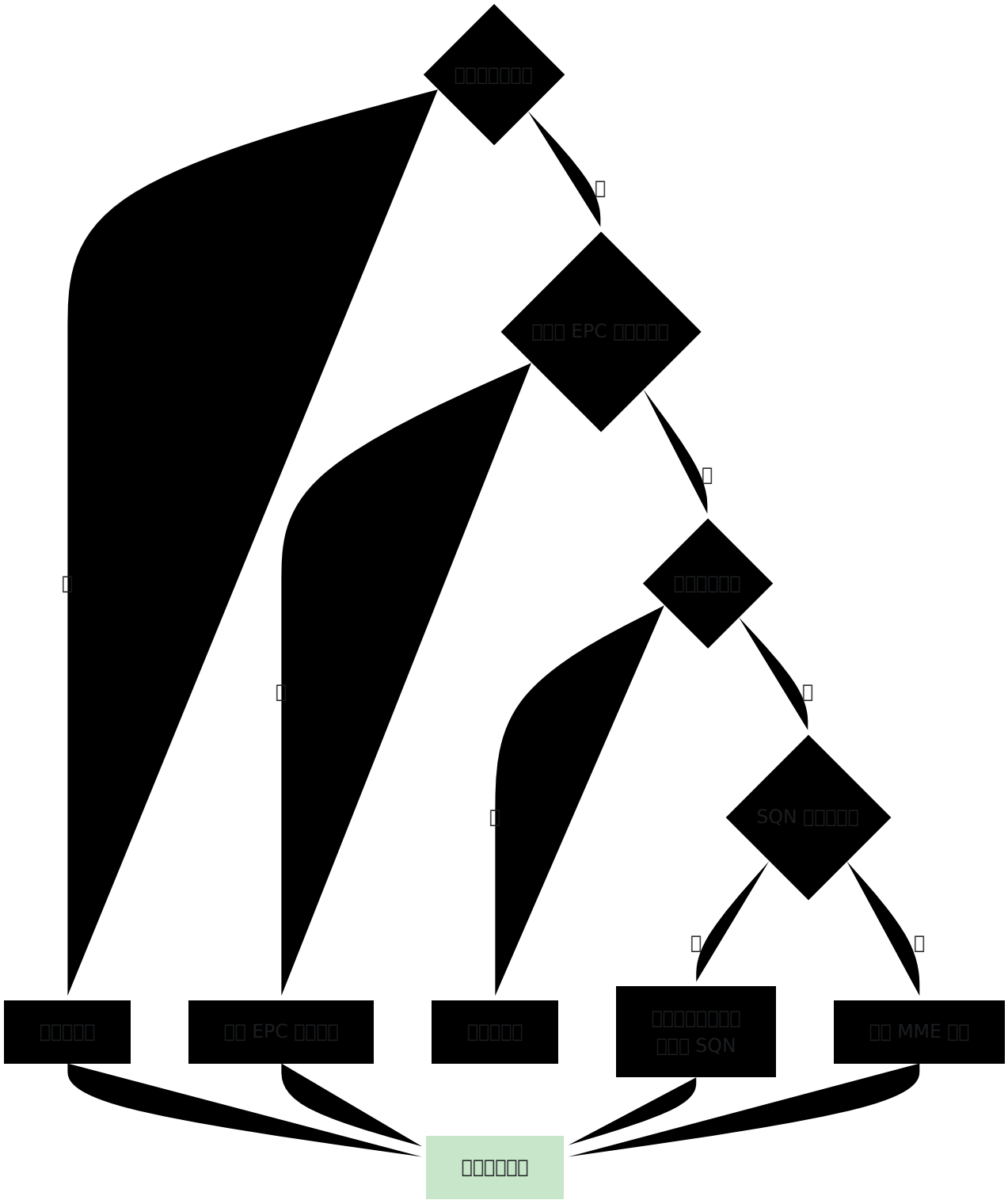
1. `epc_profile_id`
2. EPC

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


- EPC

□□□□□□□□□□□□

□□□□□□



Diameter 检查

检查

- 检查 [PEER_IP] 的 Diameter 服务是否正常
- “配置是否正确”检查
- 检查日志输出

检查步骤

检查 1: 检查 IP

检查

- 检查 IP 地址
- 检查端口
- Ping 检查

检查

1. 检查 OmniHSS 配置是否正确

```
ping [PEER_IP]
```

2. 检查 Diameter 服务是否正常

```
telnet [PEER_IP] 3868
```

3. 检查日志输出 Diameter 服务 3868

检查

- 检查 IP 地址
- 检查端口
- 检查日志输出

第2章 Diameter 配置

简介

- 配置目标
- CER/CEA 配置
- 配置流程

配置步骤

1. 配置 runtime.exs 中的 Diameter 配置
 - 配置 origin_host 为服务器地址
 - 配置 origin_realm 为 realm
 - 配置 IP 地址
2. 配置 CER/CEA 配置
3. 配置 OmniHSS 的 origin_host

配置完成

- 配置 Diameter 配置 runtime.exs
- 配置 OmniHSS
- 配置流程

第3章 TLS Diameter 配置

简介

- TLS 配置
- 配置目标
- “配置”“配置”

配置步骤

1. 配置 priv/cert/ 配置
2. 配置

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

3. 設定確認

4. 設定確認 TLS

確認

- 設定確認
- 設定確認
- 設定確認 OmniHSS

4. 設定確認

確認

- 設定確認
- 設定確認
- “設定確認”

確認

1. 設定確認 Diameter 設定
2. 設定確認 S6a/Cx/Sh 設定
3. 設定確認 CER/CEA 設定

確認

- 設定確認 Diameter 設定
- 設定確認
 - MME 設定 S6a/16777251
 - S-CSCF 設定 Cx/16777216
 - P-GW 設定 Gx/16777238

Diameter □□□□□□□



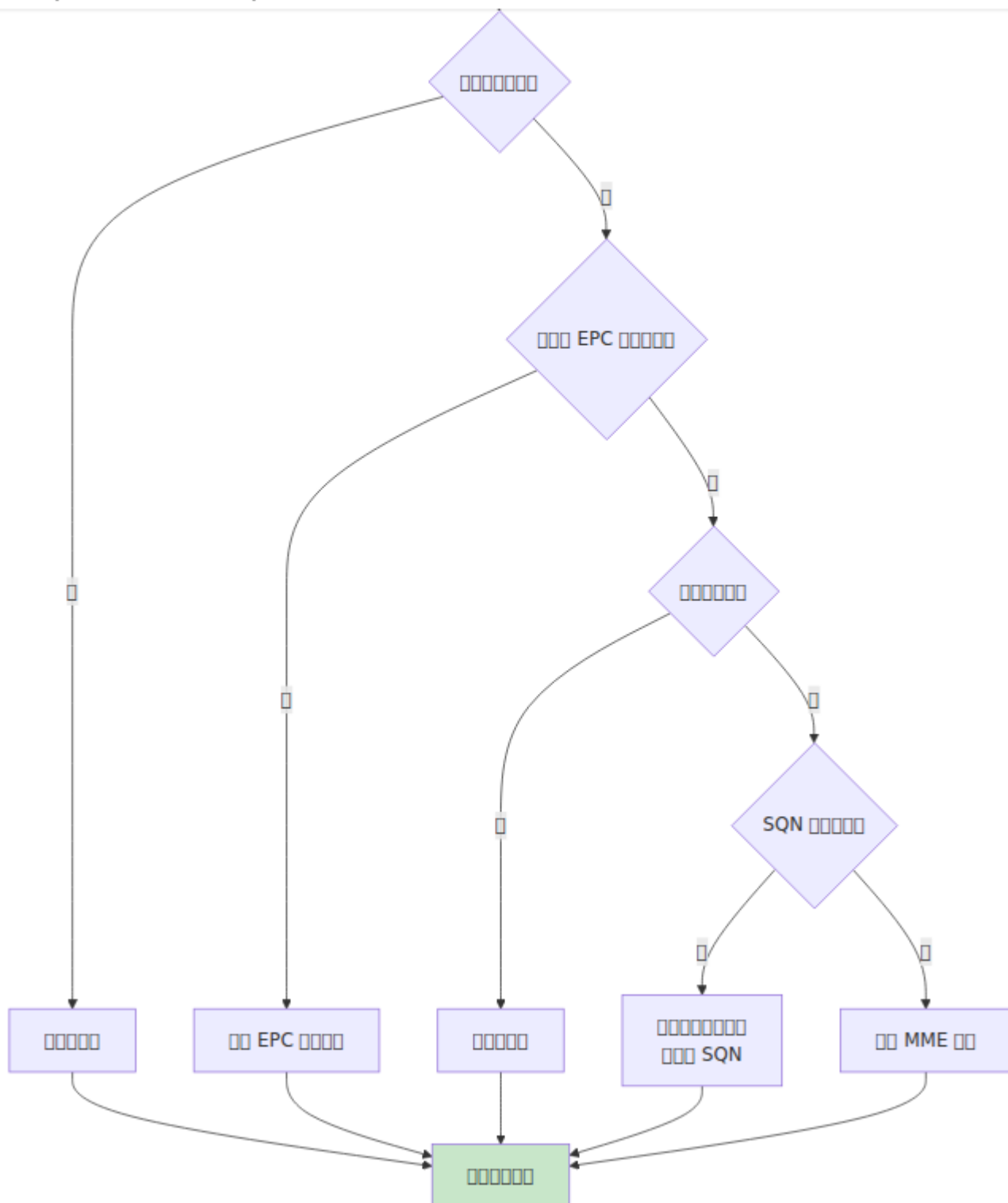
OmniCharge

OmniRAN

Downloads

🔍 □□□□ ▼

[Omnitouch Website](#) ↗



環境構築

前提

- API 500 以上
- 開発環境
- “開発環境”
- 開発環境

開発環境構築

1. 開発環境構築

開発環境

- API 500 以上
- 開発環境
- “開発環境”

開発環境

1. 開発環境構築

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

2. 開発環境構築

3. 開発環境構築

開発環境

- 開発環境
- 開発環境

- 0000000000000000

00 2000000000

000

- “000000”00
- OmniHSS 00000000

00000

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

00000

- 0 runtime.exs 000 00000
- 00000000000000
- 000000000 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

- 0000000000000000

00 **4**0000

000

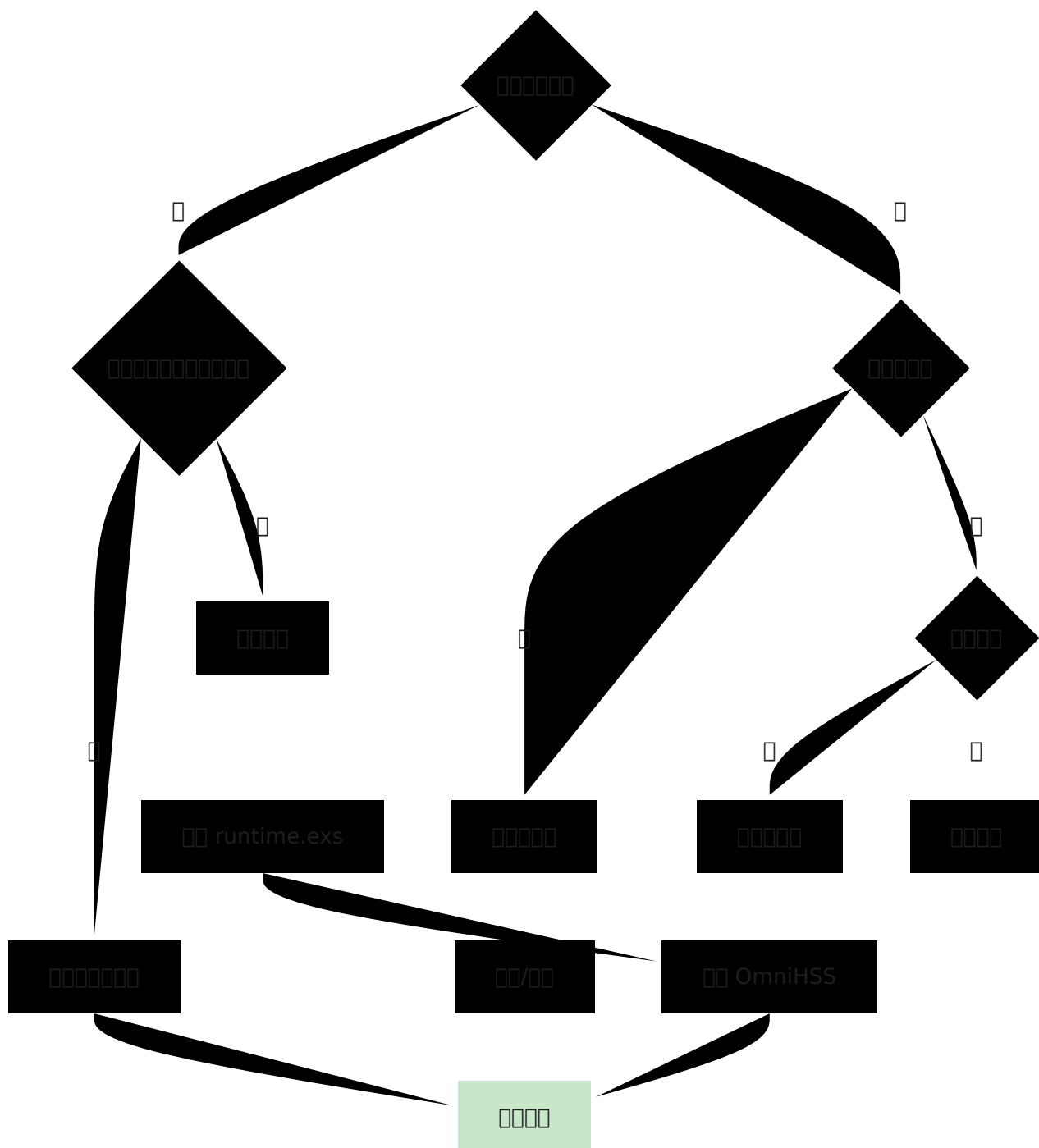
- API 00000
- 00000000
- 000 CPU 0

00000

1. 0000000000000000
2. 000000000
3. 00000000
4. 00000000000000

00000

- 00000
- 00000000
- 0000000000
- 00000000000000



EPC 问问??问

问问

- 问问问问问 LTE 问问
- MME 问问问
- 问问 PDN 问问

问问问问问问问

问问 **1**问问问问

问问

- 问问问问问问问问问问问问问
- “问问问问”问问
- 问问问问问问问问问问问

问问问问

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

问问问问

- 问问问 MCC/MNC 问问 问问问
- 问问问问问问问问问问问
- 问问问问问 问问问

问问 **2**问问 **APN** 问问

问问

- 问问问 PDN 问问问
- MME 问问“问问 APN”问问

- 網路服務品質管理

網路服務

1. 網路 EPC 網路服務品質管理 APN 網路服務
2. 網路 APN 網路服務品質管理
3. 網路 APN 網路服務品質管理

網路服務

- 網路 APN 網路服務品質管理 EPC 網路服務
- 網路 APN 網路服務品質管理
- 網路 APN QoS 網路服務品質管理

網路 3GPP MME 網路服務

網路服務

- 網路服務品質管理
- 網路 MME 網路服務
- Diameter 網路服務

網路服務

1. 網路 網路 Diameter 網路服務
2. 網路 MME 網路服務“網路服務”
3. 網路 MME 網路服務 S6a 網路服務

網路服務

- 網路 Diameter 網路服務
- 網路 MME 網路服務
- 網路 MME 網路服務

網路 4GPP 網路服務

網路服務

- 網路服務品質管理

- 00000000
- 0000000000

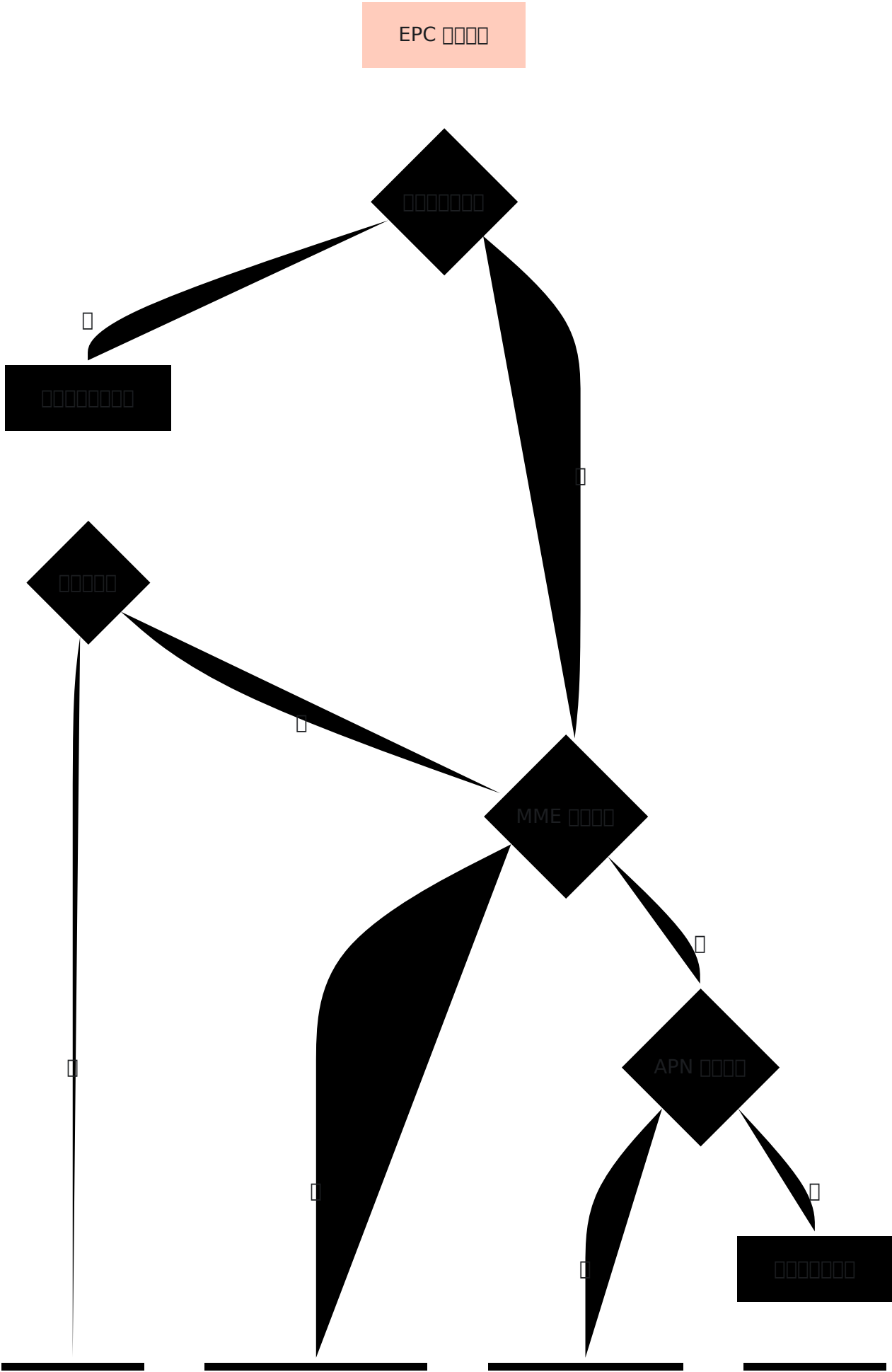
00000

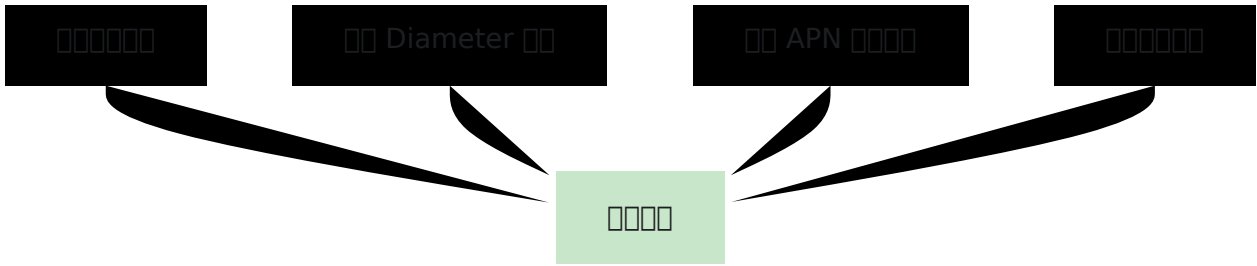
1. 000000000000
2. 00000000 MME 00
3. 0000000000

00000

- 00000000000000
- 000000000000 MME
- 000000000000

EPC □□□□□□□□





IMS

IMS

- IMS VoLTE
- IMS“IMS”
- IMS

IMS

1 IMS

IMS

- IMS
- IMS

IMS

- IMS `ims_enabled`
- IMS `ims_profile_id`

IMS

- IMS
- IMS

2 S-CSCF

IMS

- 簡 IMS 簡簡
- 簡簡 IMS 簡簡 Diameter 簡

簡簡簡

1. 簡 簡簡 Diameter 簡
2. 簡 S-CSCF 簡簡簡簡
3. 簡 S-CSCF 簡簡 Cx 簡簡

簡簡簡

- 簡 Diameter 簡簡 簡 S-CSCF
- 簡 S-CSCF 簡

簡 **3**簡簡簡簡 IFC 簡

簡簡

- 簡簡簡~~???~~簡簡簡簡簡
- 簡簡簡 IFC 簡簡簡

簡簡簡

1. 簡簡簡簡 IMS 簡簡
2. 簡 IFC 簡簡簡簡
3. 簡 IFC XML 簡

簡簡簡

- 簡簡簡 IFC 簡簡 IMS 簡簡
- 簡 IFC 簡簡簡簡 簡簡簡簡

簡 **4**簡IMS 簡簡簡

簡簡

- IMS 簡簡簡簡簡簡
- 簡簡簡
- 簡簡簡簡 IMS 簡

□□□□

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

□□□□

- □□ □□□□ □□ IMS
- □□□□□□□□□□□ IMS □□

IMS □□□□□□□□

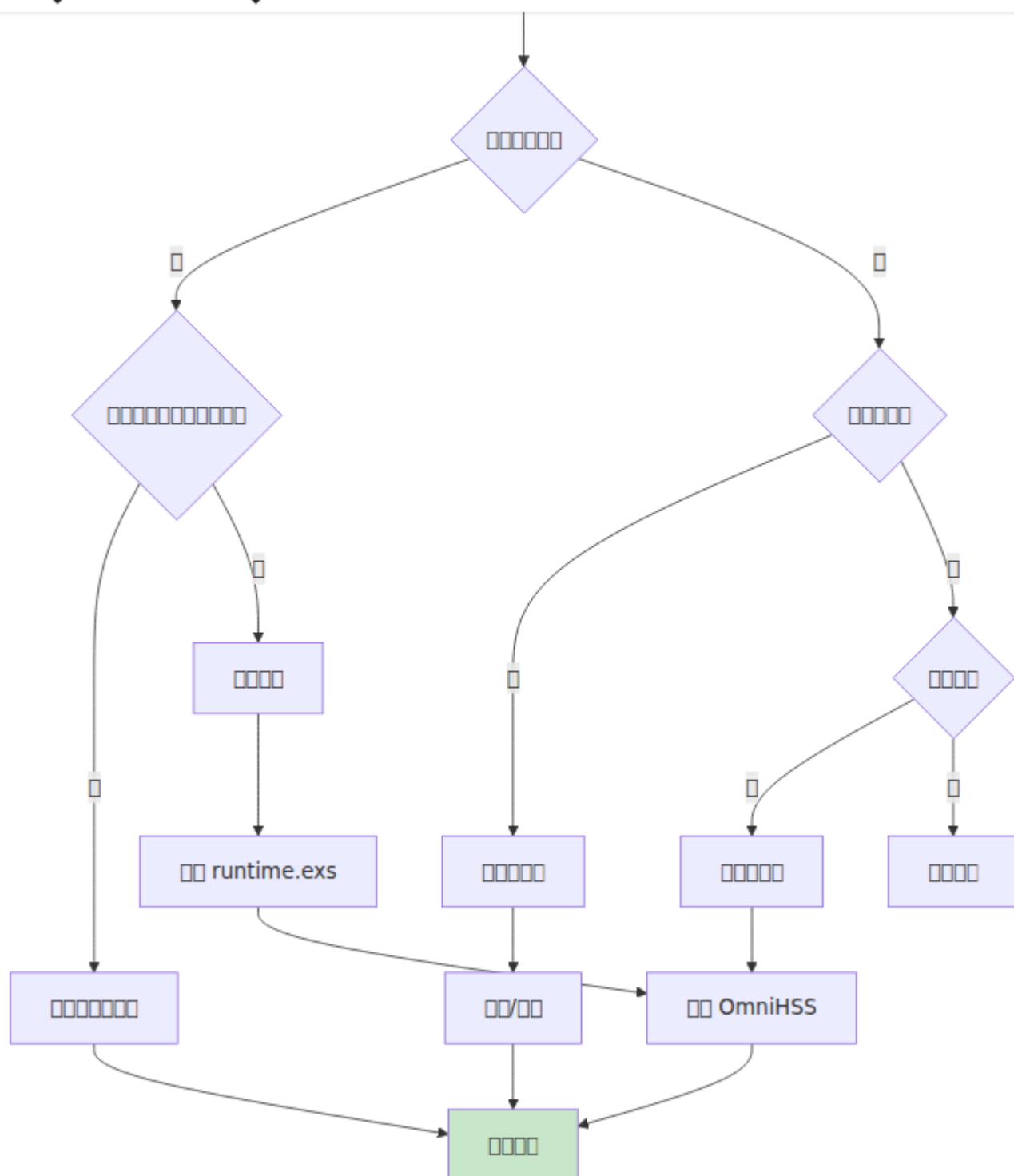
OmniCharge

OmniRAN

Downloads

文A □□□□ ▼

[Omnitouch Website](#) 



VoLTE 网络

网络

- IMS 网络
- 网络
- 网络
- 网络“网络”

网络网络

网络 1P-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. 設定する Rx Diameter
2. 設定する AAR/AA-IP
3. IP AAA/AA-IP

設定

- IP P-CSCF 設定 AAR 設定
- IP OmniHSS Rx 設定
- 設定する IMS

3QoS/設定

設定

- 設定
- 設定
- 設定

設定

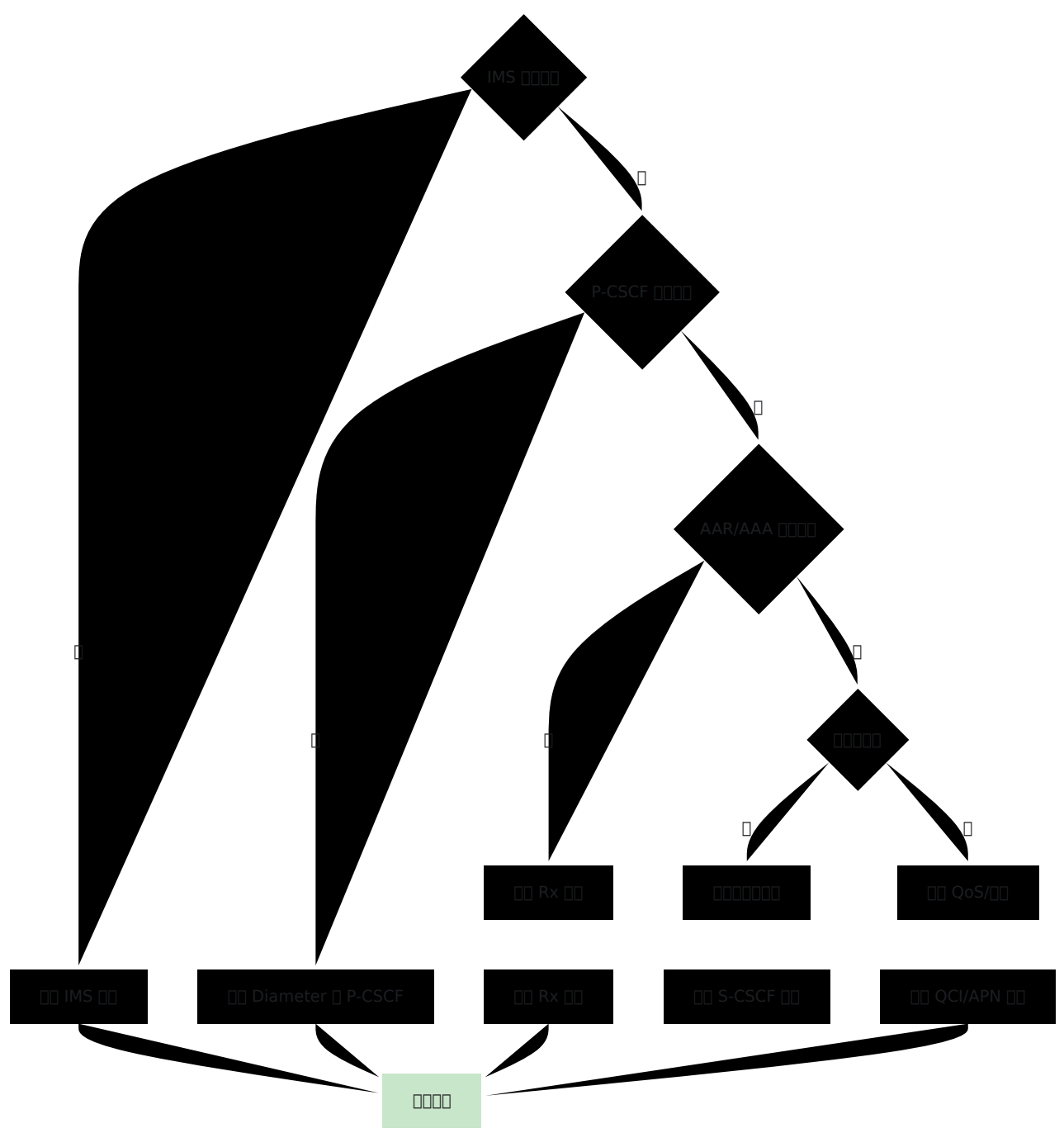
1. 設定 APN APN QoS 設定
2. QCI 設定 QCI 1
3. P-GW 設定 Gx/PCRF 設定

設定

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

VoLTE 网络架构

VoLTE 网络架构



AAA

AA

- AAA
- AAA
- AAA

AAAAA

AA **1**AAAA

AAA

- AAA
- AAA

AAAA

1. AAAA `roaming_profile_id`
2. AAAA null

AAAA

- A AAA AAA

AA **2**AAAA

AAA

- AAA
- AAA

AAAA

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

4. 設定 roaming_rule

設定方法

- curl を利用して設定

```
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

- IMS 設定
- IMS/IMS 設定
- IMS 設定

設定

1. IMS 設定
2. data_action 設定 ims_action 設定
3. IMS 設定

設定

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

設定

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

1. 查詢 EIR 查詢
2. 查詢號碼
3. 查詢號碼 IMEI
4. 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 查詢 查詢 查詢

查詢

1. 查詢 MME 資料庫 S13 查詢
2. 查詢 MME Diameter 資料庫
3. 查詢 S13 資料庫
4. 查詢 MME 查詢

查詢

- 查詢 MME 查詢 S13 查詢 EIR 查詢
- 查詢 Diameter 資料庫 S13 查詢 16777252
- 查詢 MME 查詢

查詢 3 查詢

查詢

- 查詢

查詢

1. 查詢 EIR 查詢
2. 查詢
3. 查詢

查詢

- 查詢 . * 查詢 IMEI
- 查詢
- 查詢

查詢

查詢

- API 查詢
- Diameter 查詢
- CPU 查詢

- 環境問題

環境問題

第1章 環境問題

環境

- 環境問題
- 環境 CPU
- 環境

環境

1. 環境問題
2. 環境
3. 環境
4. 環境

環境

- 環境
- 環境
- 環境
- 環境
- 環境

第2章 環境問題

環境

- 環境
- 環境
- 環境

環境

1. 環境

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 MME 安裝
- 安裝 MME 安裝
- 安裝

□□□□

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

□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□
- MME □□□□□□ Cancel-Location

□□ **2**□□□□□□□□

□□□

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

□□□□□

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

□□□□□

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

□□ **3**□□□□□□□

□□□

- □□□□□□□□□
- □□□□□
- □□□□□□

□ □ □ □ ◆ ◆

1.
2.
3.

--	--	--	--	--

- 
- 
- 

API

11

- API 10000
- API 10000
- 10000/10000
- 500 00

--	--	--	--	--	--	--	--	--

1

000

- 400 □ 422 □□
- □□□□□
- □□□□□

□ □ □ □ □

1.
2. API

3. 登録済みのキー

4. キーセット

手順

- 登録済みのキー API 呼び出し
- キーセットの取得
- 登録済みのキーセット ID の取得

ステップ 2: キーセット

手順

- キーセットの取得
- キーセット“key_set_id”の取得
- キーセットの取得

手順

1. キーセットの取得
2. キーセットの取得
 - key_set_id → キーセット
 - epc_profile_id → EPC キーセット
 - ims_profile_id → IMS キーセット

手順

- キーセットの取得
- キーセット ID の取得
- キーセットの取得

ステップ 3: キーセット

手順

- 500 キー
- キー API 呼び出し
- キーセットの取得

□□□□

- □□ □□□□

□□□□□□□□

□□□□□□□□

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

IMSI 検索

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

認証失敗

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

Diameter 接続エラー

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

データベースエラー

```
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
- 00000000
- 0000
- 0000

0000000000

00000000

0000	00	0000
"000000000000"	0000000000	00 00000
"SQN 0000"	SQN 000	000000
"0000000"	00 IMSI	00 IMSI000000
"0000000"	enabled=false	00000

Diameter 00

0000	00	0000
"Diameter 00000000"	0000	000000
"CER/CEA 0000"	00000	00 Diameter 00
"0000000000"	0000000000000000	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
 - 安全
-

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，使用 HTTP POST 方法發送通知。

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 - Webhook の設定方法
- 概要 - Webhook の利用方法

Webhook の概要と利用方法について詳しく説明します。

Webhook 概要

OmniHSS の Webhooks の概要

EPC/LTE 概要

項目	項目	項目
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 の位置情報削除要求
cancel_location_answer	S6a CLA	MME の位置情報削除応答

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: update_location_request
event_id	string	Unique webhook event UUID
timestamp	string	Event timestamp in ISO 8601 format
subscriber	object	Subscriber information. GET /api/subscriber/:id
event_context	object	Event context information

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 表

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

表

表 Webhooks

Webhooks 表 OmniHSS API 表

表 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
    });
  }

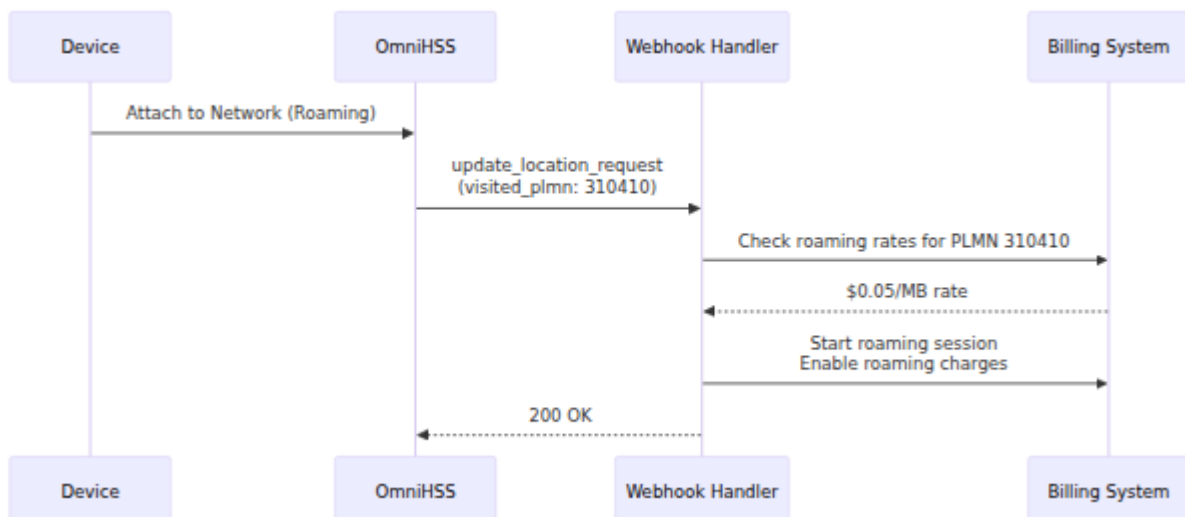
  res.status(200).send();
});
```

Sequence Diagram

- MME
-
-
- 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 注册 - 启用 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']
    });
  }

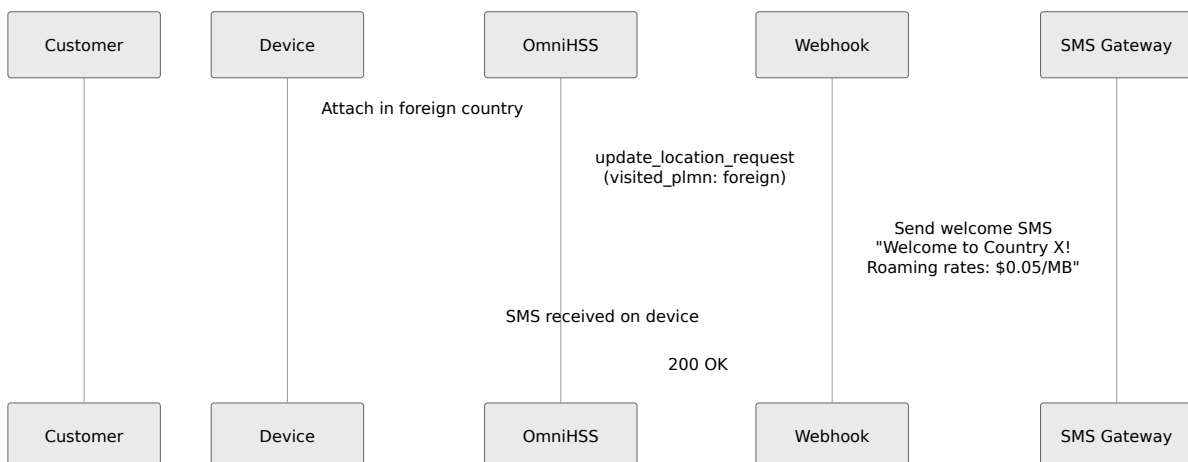
  res.status(200).send();
});

```

5. 流程图

流程图描述如下：

涉及的角色和系统：



流程图结束

- "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 事件 通知

- IMSI 通知
- MSISDN 通知
- 通知 PLMN MME
- 通知

通知

- **GDPR** - 通知 webhook 通知 GDPR
- 通知 - 通知
- 通知 - 通知 webhook 通知
- 通知 - 通知 TLS 通知 webhook 通知
- 通知 - 通知 webhook 通知

Webhook

Webhook 通知

通知

- 通知 webhook 通知
- Webhook 通知

通知

1. 通知 **webhook** 通知

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

2. 通知 **webhook** 通知

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

3. HSS

- webhook
-
- DNS

4.

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

Webhook

- HSS webhook
- Webhook HSS

1.

- 5 HTTP 200
-

2.

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

// 20 - 20초 지연
app.post('/webhook', (req, res) => {
  res.status(200).send(); // 20초 지연
  processData(req.body); // 20초 지연
});
```

Webhooks

1. 개요

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

2. 특징

- 이벤트 발생 시 OmniHSS 시스템에서 호출되는 API
- 이벤트 발생 시 webhook

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 ~ 5xx
- HSS から webhook 失敗

原因

1. 401 Unauthorized - 認証失敗

- 正しい webhook URL ではない
- 署名が正しい

2. 400 Bad Request - 不正なリクエスト

- 正しい webhook URL ではない
- 正しい Content-Type ではない

3. 500 Internal Server Error - サーバエラー

- サーバエラー

- 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

配置项

```
{
  "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
}
```

OmniHSS

概要

OmniHSS は 4G LTE (EPC) と IMS (IP 多メディアサブシステム) の統合認証・課金・ポリシー決定 (HSS) を提供します。

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

機能

HSS は LTE と IMS の両方に対応しています。

- 認証 - ユーザーの認証
- 課金 - ユーザーの課金
- ポリシー決定 - ユーザーのポリシー決定
- 認証 - ユーザーの認証
- 課金 - ユーザーの課金
- 認証 - ユーザーの認証
- 課金 - ユーザーの課金 (EIR) の管理

インターフェース

外部インターフェース

- S6a** - LTE/EPC の HSS と MME のインターフェース
- Cx** - IMS の HSS と CSCF のインターフェース
- Sh** - IMS の HSS と AS のインターフェース
- S13** - HSS と EIR (OmniHSS の EIR) のインターフェース
- Gx** - HSS と PCRF (OmniHSS の PCRF) のインターフェース
- Rx** - IMS の HSS と PCRF (OmniHSS の PCRF) のインターフェース
- PLMN - PLMN の IMS の HSS のインターフェース

- **MSISDN** - 電話番号
- **RESTful API** - 様々な API から OmniHLR へ
- **Web** 経由 - 様々なサービス

機能

OmniHSS 機能

- **MME** (LTE) - LTE ネットワーク
- **P-GW** (PDN GW) - 様々な OmniHSS へ (PCRF 経由)
- **P-CSCF** (IMS) - IMS ネットワーク
- **I-CSCF** (IMS) - IMS ネットワーク
- **S-CSCF** (IMS) - IMS ネットワーク
- **AS** (IMS) - IMS ネットワーク
- **OmniHLR** - 様々な API から OmniHSS へ HLR

接続

接続方法

接続

- **API** - 様々な API から Diameter へ
- **API** - 様々な API から Diameter へ
- **API** - 様々な API から Diameter へ

接続

- **API** - 様々な Web サービス
- **API** - 様々な Web サービス
- **API** - 様々な Web サービス
- **API** - 様々な API サービス
- **Webhooks** - 様々な Web サービス

네트워크

- **APN** - EPC의 IMS의 APN 정보
- **APN** - 네트워크
- **APN** - Diameter 프로토콜
- **PCRF** - 프로토콜 (Gx/Rx 프로토콜 QoS VoLTE)
- **EIR** - 프로토콜 (S13 프로토콜 IMEI 프로토콜)
- **MSISDN** 프로토콜 **IMSI** 프로토콜 - 프로토콜 프로토콜 IMSI

네트워크 프로토콜

네트워크

네트워크 (Web 프로토콜)

URL: https://[hostname]:7443

네트워크 프로토콜 Diameter 프로토콜

API 프로토콜

URL: https://[hostname]:8443

RESTful API 프로토콜

네트워크 프로토콜

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

네트워크

1. 프로토콜 - 프로토콜
2. **???** 프로토콜 **Diameter** 프로토콜 - 프로토콜 Diameter 프로토콜

3. 部署 - 部署 API 端点 `/api/subscriber/imsi/:imsi`
4. 数据库 - 数据库连接池 SQL 连接

部署部署部署

部署部署

部署部署 stdout/stderr 部署部署部署部署 (systemd, supervisord 等) 部署

部署部署

- **Diameter** 部署 - 部署 Diameter 部署部署部署部署
- 部署部署 - 部署 runtime.exs 部署部署部署
- 部署部署 - 部署部署部署部署部署部署

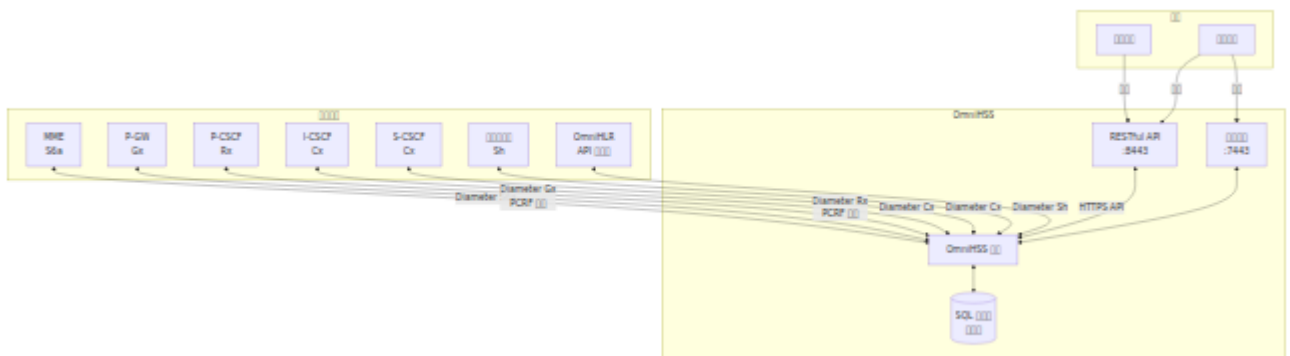
部署部署

- **API** 部署 - `GET /api/status`
- 部署部署 - 部署部署部署部署部署
- 部署 - 部署 SQL 部署部署部署部署

部署部署部署

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

□□□□



□□□

□□□□□□□□□□□□□□□□□□□□

- □ □□□□ □□□□□□□□□□
- □□ □□□□ □□□□□□□□□□
- □□ □□□□ □□□□□□□□
- □□ **API** □□ □□□□□□□□

□□□□□ 1.0

□□□□ Omnitouch □□□□