

REST API

←

OmniSS7 REST API Swagger UI

- 1.
- 2. HTTP
- 3. Swagger UI
- 4. API
- 5.
- 6.
- 7.
- 8. Prometheus
- 9.

OmniSS7 REST API MAP API

- MAP SRI SRI-for-SM UpdateLocation
- MAP
- Prometheus

API



HTTP

	HTTP	
IP	0.0.0.0	
	8080	
	Plug.Cowboy	

URL: http://[server-ip]:8080

/ HTTP

HTTP

```
config :omniss7,  
  start_http_server: true # false
```

: true

: HTTP REST API/Swagger UI

Swagger UI

API Swagger UI API

Swagger UI

URL: http://[server-ip]:8080/swagger

□□:

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

Swagger JSON

OpenAPI□□□□□□□□□□

URL: `http://[server-ip]:8080/swagger.json`

□□:

- □□□Postman□□□API□□□
- □□□□□□
- API□□□□□□

API□□

□□MAP□□□□□□□□□□ `POST /api/{operation}`

API

API	Method	Description	Timeout
/api/sri	POST	Send Routing Info	10s
/api/sri-for-sm	POST	Send Routing Info for SM	10s
/api/send-auth-info	POST	Send Auth Info	10s
/api/MT-forwardSM	POST	MT Forward SM	10s
/api/forwardSM	POST	Forward SM	10s
/api/updateLocation	POST	Update Location	10s
/api/prn	POST	Print	10s
/metrics	GET	Prometheus	N/A
/swagger	GET	Swagger UI	N/A
/swagger.json	GET	OpenAPI	N/A

API: MAP API 10

SendRoutingInfo (SRI)

API: POST /api/sri

API:

```
{
  "msisdn": "1234567890",
  "gmsc": "5551234567"
}
```

表:

項目	項目	項目	項目
msisdn	項目	項目	項目MSISDN
gmsc	項目	項目	項目MSC項目

表 (200 OK):

```
{
  "result": {
    "imsi": "001001234567890",
    "msrn": "5551234999",
    "vlr_number": "5551234800",
    ...
  }
}
```

表 (504 Gateway Timeout):

```
{
  "error": "timeout"
}
```

cURL:

```
curl -X POST http://localhost:8080/api/sri \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "gmsc": "5551234567"
}'
```

SendRoutingInfoForSM (SRI-for-SM)

이 API는 SMS 전송을 위한 정보를 보냅니다.

요청: `POST /api/sri-for-sm`

응답:

```
{
  "msisdn": "1234567890",
  "service_center": "5551234567"
}
```

응답:

필드	타입	길이	설명
msisdn	문자열	15	발신자 MSISDN
service_center	문자열	15	서비스 센터 번호

상태 (200 OK):

```
{
  "result": {
    "imsi": "001001234567890",
    "msc_number": "5551234800",
    "location_info": {...},
    ...
  }
}
```

cURL:

```
curl -X POST http://localhost:8080/api/sri-for-sm \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "service_center": "5551234567"
}'
```

SendAuthenticationInfo

요청 파라미터

메소드: `POST /api/send-auth-info`

요청 바디:

```
{
  "imsi": "001001234567890",
  "vectors": 3
}
```

응답:

項目	項目	項目	項目
imsi	IMSI	IMSI	IMSI
vectors	IMSI	IMSI	IMSI

IMSI (200 OK):

```
{
  "result": {
    "authentication_sets": [
      {
        "rand": "0123456789ABCDEF...",
        "xres": "...",
        "ck": "...",
        "ik": "...",
        "autn": "..."
      }
    ],
    ...
  }
}
```

cURL:

```
curl -X POST http://localhost:8080/api/send-auth-info \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "vectors": 3
}'
```

MT-ForwardSM

MT-ForwardSM

POST /api/MT-forwardSM

Request:

```
{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}
```

Response:

Field	Type	Length	Description
imsi	String	15	International Mobile Subscriber Identity (IMSI)
destination_service_centre	String	10	Destination Service Centre (GT)
originating_service_center	String	10	Originating Service Centre (GT)
smsPDU	String	255	Short Message Service (SMS) TPDU

Response: smsPDU (Hexadecimal string)

Response (200 OK):

```
{
  "result": {
    "delivery_status": "success",
    ...
  }
}
```

cURL:

```
curl -X POST http://localhost:8080/api/MT-forwardSM \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}'
```

ForwardSM

短信MO-SMS

请求: POST /api/forwardSM

响应: MT-ForwardSM

cURL:

```
curl -X POST http://localhost:8080/api/forwardSM \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}'
```

UpdateLocation

HLR/VLR

请求: POST /api/updateLocation

响应:

ProvideRoamingNumber (PRN)

MSRN

POST /api/prn

:

```
{
  "msisdn": "1234567890",
  "gmsc": "5551234567",
  "msc_number": "5551234800",
  "imsi": "001001234567890"
}
```

:

msisdn			MSISDN
gmsc			MSC GT
msc_number			MSC
imsi			IMSI

(200 OK):

```
{
  "result": {
    "msrn": "5551234999",
    ...
  }
}
```

cURL:

```
curl -X POST http://localhost:8080/api/prn \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "gmsc": "5551234567",
  "msc_number": "5551234800",
  "imsi": "001001234567890"
}'
```

테스트

테스트: API 호출 성공

테스트:

- API 호출 성공/실패
- 데이터 검증
- 에러 처리

결과

응답 데이터: **JSON** 형식

응답:

HTTP 상태: 200 OK

응답:

```
{
  "result": {
    // 데이터
  }
}
```

错误码

HTTP码:

- 400 Bad Request - 无效的JSON
- 504 Gateway Timeout - MAP响应超时10秒
- 404 Not Found - 找不到

响应:

```
{
  "error": "timeout"
}
```

请求:

```
{
  "error": "invalid request"
}
```

错误码

错误码

码	HTTP码	码	描述
无效JSON	400	无效的JSON	JSON无效
找不到	404	找不到	找不到
超时	504	MAP响应超时10秒	MM3UA响应HLR/VLR超时
错误	404	错误	错误URL

□□□□

□□MAP□□□□□□□□□□**10**□□□□

1. □□□□□MapClient GenServer
2. □□□□□□10□
3. □□□□□□ → □□504 Gateway Timeout
4. □□□□□□ → □□200 OK□□□

□□□□□□:

- □□M3UA□□□□□Web UI → M3UA□□□
- □□□□□□□HLR/VLR/MSC□□□□□
- □□□□□□
- □□SS7□□□□□□□□□□

□□□Prometheus□

□API□□Prometheus□□□□□□□

□□□□

URL: `http://[server-ip]:8080/metrics`

□□: Prometheus□□□□

□□□□:

```
# HELP map_requests_total Total MAP requests
# TYPE map_requests_total counter
map_requests_total{operation="sri"} 42
map_requests_total{operation="sri_for_sm"} 158
map_requests_total{operation="updateLocation"} 23

# HELP cap_requests_total Total CAP requests
# TYPE cap_requests_total counter
cap_requests_total{operation="initialDP"} 87
cap_requests_total{operation="requestReportBCSMEEvent"} 91

# HELP map_request_duration_milliseconds Duration of MAP
request/responses in ms
# TYPE map_request_duration_milliseconds histogram
map_request_duration_milliseconds_bucket{operation="sri",le="10"}
5
map_request_duration_milliseconds_bucket{operation="sri",le="50"}
12
map_request_duration_milliseconds_bucket{operation="sri",le="100"}
35
...

# HELP map_pending_requests Number of pending MAP TID waiters
# TYPE map_pending_requests gauge
map_pending_requests 3
```

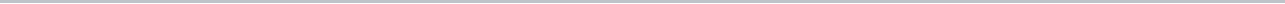
メトリクス

メトリクス	単位	操作	説明
<code>map_requests_total</code>	回 分	operation	MAP操作の総回数を示すメトリクス
<code>cap_requests_total</code>	回 分	operation	CAP操作の総回数を示すメトリクス
<code>map_request_duration_milliseconds</code>	ミリ秒 分	operation	MAP操作の平均実行時間
<code>map_pending_requests</code>	回	-	現在待機中のMAP操作の回数

Prometheus

設定ファイル `prometheus.yml`

```
scrape_configs:  
  - job_name: 'omniss7'  
    static_configs:  
      - targets: ['server-ip:8080']  
    metrics_path: '/metrics'  
    scrape_interval: 15s
```



□□□□

Python□□

```
import requests
import json

# SRI-for-SM□□
url = "http://localhost:8080/api/sri-for-sm"
payload = {
    "msisdn": "1234567890",
    "service_center": "5551234567"
}

response = requests.post(url, json=payload, timeout=15)

if response.status_code == 200:
    result = response.json()
    print(f"□□: {result}")
elif response.status_code == 504:
    print("□□ - □□□□□□")
else:
    print(f"□□: {response.status_code} - {response.text}")
```

JavaScript

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

async function sendSRI() {
  try {
    const response = await
    axios.post('http://localhost:8080/api/sri', {
      msisdn: '1234567890',
      gmsc: '5551234567'
    }, {
      timeout: 15000
    });

    console.log('SRI:', response.data);
  } catch (error) {
    if (error.code === 'ECONNABORTED') {
      console.error('SRI - Connection Aborted');
    } else {
      console.error('SRI:', error.response?.data || error.message);
    }
  }
}

sendSRI();
```

Bash/cURL

```
#!/bin/bash

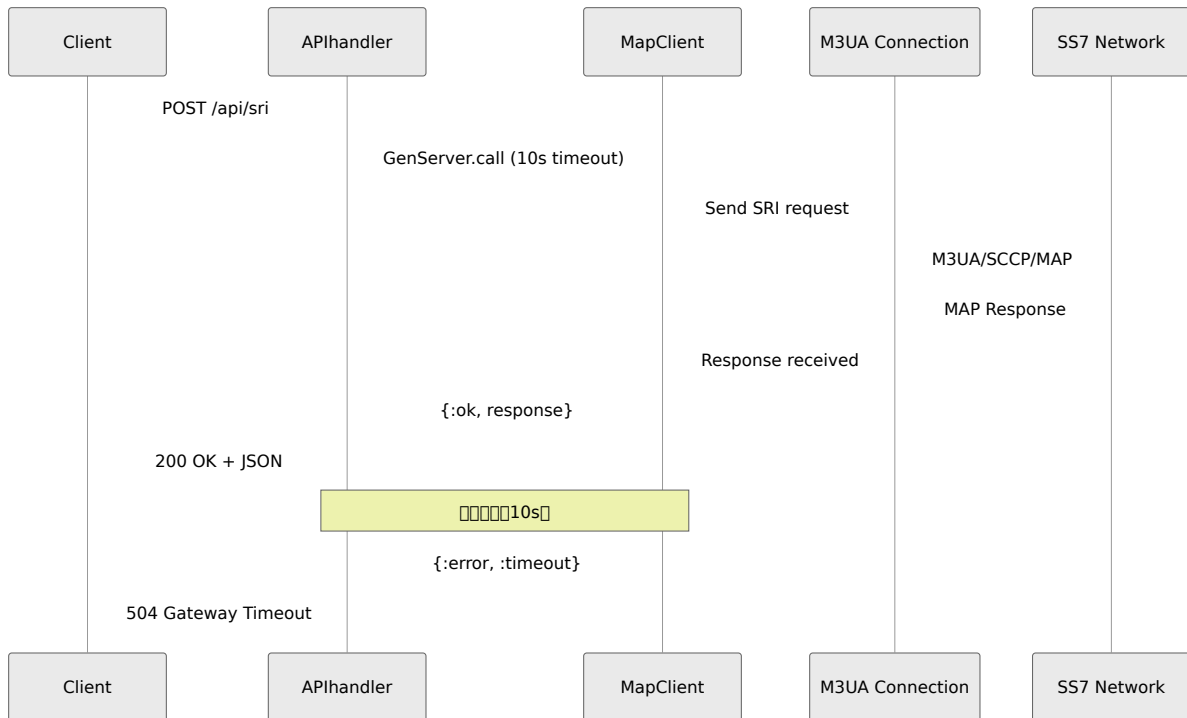
# UpdateLocation
response=$(curl -s -w "\n%{http_code}" -X POST
http://localhost:8080/api/updateLocation \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "vlr": "5551234800"
}')

http_code=$(echo "$response" | tail -n 1)
body=$(echo "$response" | sed '$d')

if [ "$http_code" -eq 200 ]; then
  echo "OK: $body"
elif [ "$http_code" -eq 504 ]; then
  echo "Error - timeout"
else
  echo "Error $http_code: $body"
fi
```



API



OmniSS7 REST API

MAP - SRI SRI-for-SM UpdateLocation SMS

Swagger UI - API

Prometheus -

MAP 10

HTTP - 8080 start_http_server

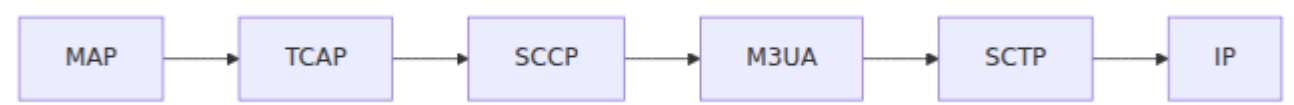
Web UI Web UI

SS7

← SS7

SS7OmniSS7

SS7



MAP

MAP	SS7	SS7
updateLocation	2	updateLocation
cancelLocation	3	cancelLocation
provideRoamingNumber	4	provideRoamingNumber
sendRoutingInfo	22	sendRoutingInfo
mt-forwardSM	44	mt-forwardSM
sendRoutingInfoForSM	45	sendRoutingInfoForSM
mo-forwardSM	46	mo-forwardSM
sendAuthenticationInfo	56	sendAuthenticationInfo

TCAP

- **BEGIN** -
 - **CONTINUE** -
 - **END** -
 - **ABORT** -
-

SCCP

- **E.164** - 447712345678
- **E.212** - IMSI 234509876543210
- **E.214** -

SSN

- **SSN 6**: HLR
 - **SSN 7**: VLR
 - **SSN 8**: MSC/SMSC
 - **SSN 9**: GMLC
 - **SSN 10**: SGSN
-

SMS TPDU

- **SMS-DELIVER** (MT) -
- **SMS-SUBMIT** (MO) -
- **SMS-STATUS-REPORT** -
- **SMS-COMMAND** -

短信

- **GSM7** - 7位GSM编码短信 160字节
 - **UCS2** - 16位Unicode编码短信 70字节
 - **8-bit** - 8位编码短信 140字节
-

M3UA

- **DOWN** - SCTP故障
 - **CONNECTING** - SCTP连接
 - **ASPUP_SENT** - 发送ASPUP
 - **INACTIVE** - ASP空闲
 - **ASPAC_SENT** - 发送ASPAC
 - **ACTIVE** - 活动
-

SS7

14位ITU 24位ANSI

ITU

- 3
 - 8
 - 3
-

SCCP

- **0** - 网络
- **1** - 网络
- **2** - 网络
- **3** - 网络

- **4** - 00000
- **5** - MTP00
- **6** - 0000
- **7** - 000
- **8** - 0000000

MAP00000

00	00	00
1	unknownSubscriber	0000HLR0
27	absentSubscriber	000000
34	systemFailure	0000
35	dataMissing	00000000
36	unexpectedDataValue	000000

00000

- ← 00000
- STP00
- MAP000000
- 000000
- HLR00
- 0000

CAMEL 開発環境 - 構築

目次

このドキュメントは LiveView を利用して CAMEL/CAP の開発環境を構築するための UI を提供します。
InitialDP は CAMEL の初期設定

目次

1. CAMEL 開発環境 LiveView

目次

- 開発環境 UI を提供するための CAMEL 環境
- 開発環境の構成
 - **InitialDP** - 初期設定
 - **Connect** - 接続

- **ReleaseCall** - 释放/释放
- **RequestReportBCSMEvent** - 请求报告
- **Continue** - 继续
- **ApplyCharging** - 应用计费/应用计费

接口

- 接口
- 接口
- 接口 SCCP/M3UA 接口
 - 接口
 - SSN接口
 - OPC/DPC接口
- 接口 20 接口
- 接口 OTID 接口
- 接口/接口
- 接口

接口 /camel_request

2. 接口 EventLog 接口 CAMEL

接口

- paklog_camel/2 - 接口 CAMEL/CAP 接口
- lookup_cap_opcode_name/1 - CAP 接口
- find_cap_opcode/1 - 接口 JSON 接口 CAP 接口
- extract_cap_tids/1 - 接口 CAP 接口 OTID/DTID
- format_cap_to_json/1 - 接口 CAP PDU 接口 JSON 接口

接口 CAP 接口

```

0 => "initialDP"
5 => "connect"
6 => "releaseCall"
7 => "requestReportBCSMEEvent"
8 => "eventReportBCSM"
10 => "continue"
13 => "furnishChargingInformation"
35 => "applyCharging"
... ( 47  )

```

- CAMEL / JSON
- TCAP / /
- SCCP
-
-
- "CAP:"

3. CapClient

- paklog_camel/2
- MAP (paklog) CAP (paklog_camel)
- sccp_m3ua_maker/2
- handle_payload/1

LiveView

```
# config/runtime.exs

config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "SS7 页面"},
    {SS7.Web.TestClientLive, "/client", "SS7 测试"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {SS7.Web.HlrLinksLive, "/hlr_links", "HLR 页面"},
    {SS7.Web.CAMELSessionsLive, "/camel_sessions", "CAMEL 页面"},
    {SS7.Web.CAMELRequestLive, "/camel_request", "CAMEL 请求页面"}
  ],
  page_order: ["/events", "/client", "/m3ua", "/hlr_links",
    "/camel_sessions", "/camel_request",
    "/application", "/configuration"]
```

##

请求流程

1. 访问 `https://your-server:8087/camel_request`
2. 返回响应
3. 返回响应
4. 返回响应“SCCP/M3UA 页面”
5. 返回响应 [RequestType] 页面

请求

InitialDP

1. 返回响应100
2. 返回响应A-Party
3. 返回响应B-Party
4. 返回 → OTID
5. OTID 返回响应

返回 **Connect, ReleaseCall**

1. InitialDP OTID
2. OTID
3. OTID

InitialDP

-
- ISDN
- ISDN

Connect

-

ReleaseCall

- 16 = 17 = 31 =

RequestReportBCSMEvent

- BCSM Answer, oDisconnect

Continue

- OTID

ApplyCharging

- 1-864000 -
- -

SCCP

- GT
- GT

- SSN 146 = gsmSSF
- SSN 146

M3UA

- OPC 5013
- DPC 5011

JSON

CAMEL JSON

- /
- **TCAP** /
- **CAP** "CAP:initialDP" "CAP:connect"
- **SCCP** /
- **TIDs** OTID/DTID
- JSON CAP PDU

```
{
  "map_event": "CAP:initialDP",
  "direction": "outgoing",
  "tcap_action": "Begin",
  "otid": "A1B2C3D4",
  "sccp_called": {
    "SSN": 146,
    "GlobalTitle": {
      "Digits": "55512341234",
      "NumberingPlan": "isdn_tele",
      "NatureOfAddress_Indicator": "international"
    }
  },
  "event_message": "{ ... full CAP PDU ... }"
}
```

□□□□

UI □□□□ 20 □□□□□□□□

- □□□
- □□□□□□□□□□□□□□□□
- OTID□□ 8 □□□□□□□□
- □□□□□□/□□□
- □□□□□□□□□□□□

□□□□

□□□□□□□□□□

- □□□□ OTID
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□□

1. □□□□□□

- □□ InitialDP → □□ OTID
- □□□□□□

2. □□□□□□

- □□ RequestReportBCSMEEvent → □□□□
- □□ ApplyCharging → □□□□□□□□□□□□□□□□290 □□
- □□ Connect → □□□□□□□□
- □□□□ ReleaseCall → □□

3. □□□□□□

- □□□□□□□□
- □□ CAMEL □□□□□□

- 消息 "CAP:" 消息

ApplyCharging - 消息

消息

ApplyCharging 消息是网络侧向终端侧发送的消息，用于通知终端侧开始计费。消息中包含计费相关信息，如计费类型、计费周期等。

消息

- 消息格式
- 消息内容
- 消息长度
- **OCS** 消息格式

消息

消息 **maxCallPeriodDuration**

- 消息格式 1-86400 秒
- 消息内容
- 消息长度
 - 60 = 1 秒
 - 290 = 4 秒 50 毫秒
 - 3600 = 1 分钟
 - 86400 = 24 小时

消息 **releaseIfDurationExceeded**

- 消息格式 true/false
- 消息 true
- 消息内容
 - true 释放/不释放
 - false 释放/不释放 gsmSCF 消息

□□□□

ApplyCharging □□□□ TCAP Continue□□□□

- **TCAP**□□□□□□□□□□
- □□□□35□applyCharging□
- □□□ApplyChargingArg □□□
 - aChBillingChargingCharacteristics□□□□□□□□□□
 - timeDurationCharging□□□□□□□□□□
 - partyToCharge□□□□□□□□□□sendingSideID□

□□□□

□□□□□□□□□□ 5 □□

1. □□ **InitialDP** □□□□□□

```
□□□□□100
□□□447700900123
□□□447700900456
→ OTID□A1B2C3D4
```

2. □□ **ApplyCharging** □□ 5 □□□□

```
□□□□□300□□□□
□□□□□true
→ □□ OTID□A1B2C3D4
```

3. □□ **Connect** □□□□

```
□□□□447700900456
→ □□ OTID□A1B2C3D4
```

4. 5 □□□□300 □□□

- □□□□□□□□□□

- gsmSCF 呼叫转移

呼叫转移

1. 呼叫 **Connect** 呼叫 **ApplyCharging**

- 呼叫转移呼叫转移
- 呼叫转移呼叫转移

2. 呼叫 **RequestReportBCSMEEvent** 呼叫

- 呼叫 **oAnswer** 呼叫 **oDisconnect** 呼叫
- 呼叫转移呼叫转移
- 呼叫转移呼叫转移

3. 呼叫转移

- 呼叫转移呼叫转移
- 呼叫转移呼叫转移
- 呼叫转移 60-300 呼叫转移

4. 呼叫转移

- 呼叫 **release=false** 呼叫转移
- 呼叫转移呼叫转移

呼叫转移

呼叫转移

- 呼叫 **OTID** 呼叫 InitialDP
- 呼叫转移呼叫转移 1-864000 呼叫
- 呼叫转移 SSF 呼叫 ApplyCharging
- 呼叫转移呼叫转移 1 呼叫转移

呼叫

呼叫转移 ApplyCharging 呼叫

- 000000000000 ApplyCharging 00
- 00000000 "CAP:applyCharging"
- **CAMEL** 0000000000000000
- **TCAP** 00000000/0000

0000

0000

- LiveView 00000000
- OTID 00000000
- 00000000 20 000
- 0000000000000000

0000

- 000000 CapRequestGenerator 00
- 000000 TCAP/CAP 00
- 00 :TCAPMessages 00000000
- 00 CapClient.sccp_m3ua_maker/2 0000 SCCP 0

0000

- 00 M3UA 0000 :camelgw_client_asp
- 00000000 1
- 00 SCCP/M3UA 00

0000

- 0000000000
- 00000000 OTID
- UI 00000000
- 00000000

□□□□

□□□□□

1. □□□□/□□
2. □□□□□□□
3. □□□□□□□
4. □□□□□□□□
5. □□□□□□
6. □□□□□□□□□□
7. □□□□□ PCAP □□
8. CAP □□□□

□□□□

- □□□□ MAP □□□□ `paklog` □□□
- □ MAP □□□□□□□□□□□
- □□□□□ SCCP/M3UA □□□□
- □ CAMELSessionsLive □□□□□□□□□
- □□□□ M3UA □□□□

□□□□□

- `config/runtime.exs` - □□□

□□□

- □□□ CapRequestGenerator
- □□ M3UA □□□ CapClient
- □□□□□□□□ M3UA.Server
- □□□□□□□□□ EventLog
- Phoenix LiveView □□

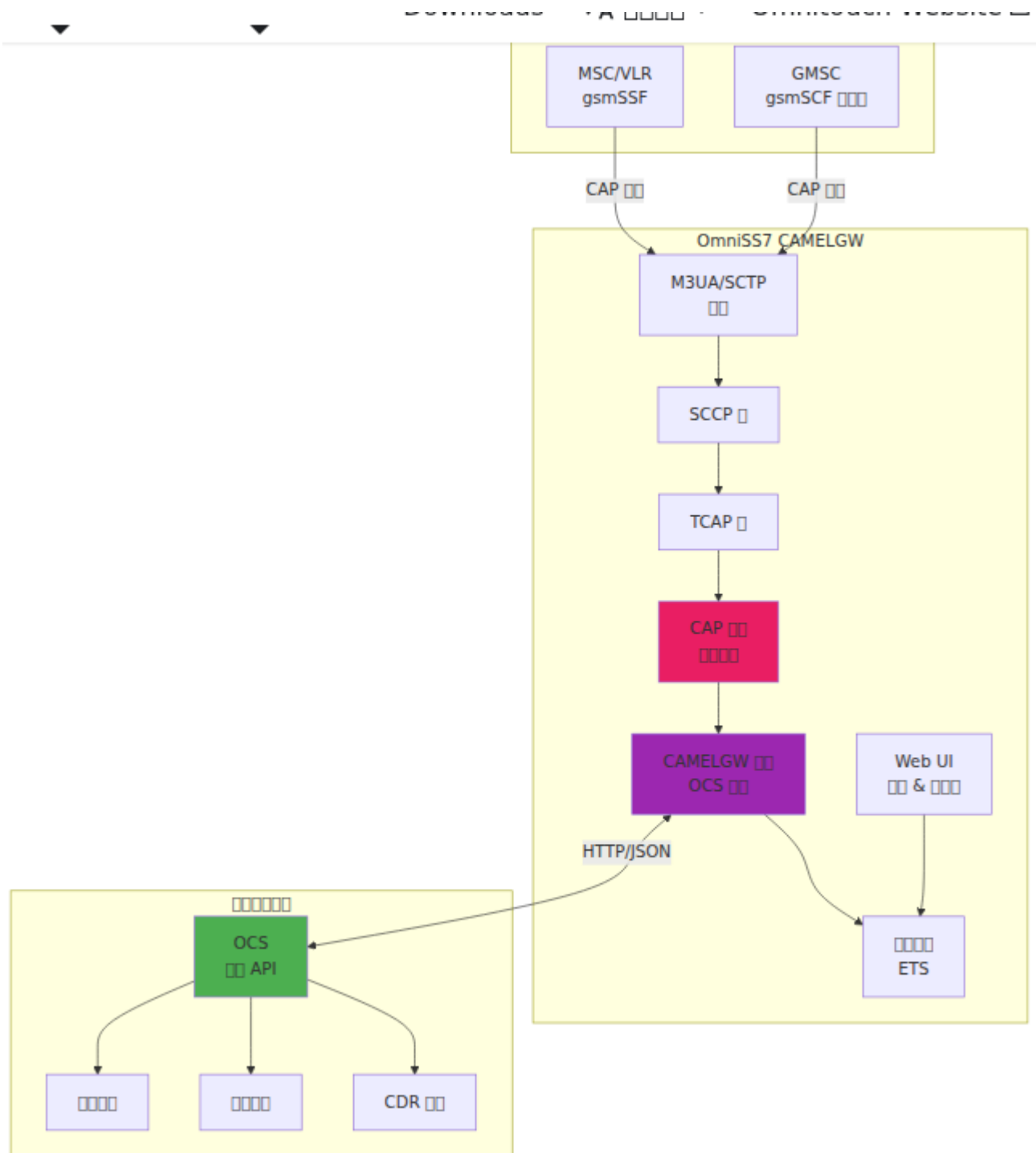
- 00 UI 0000000000

CAP 简介

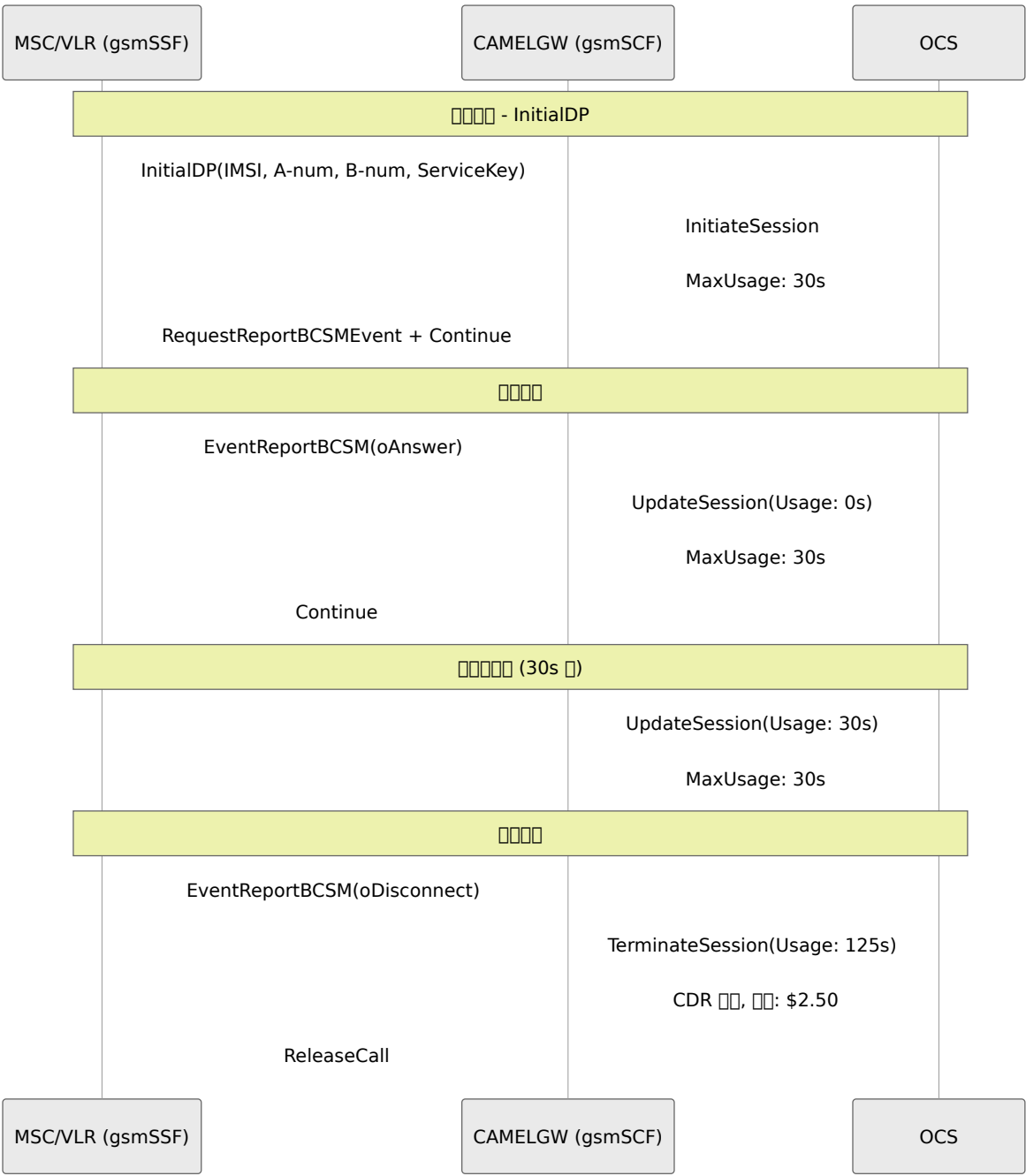
OmniSS7 CAMELGW 简介 CAP 简介

版本	名称	描述
CAP v1	CAMEL 1	语音业务
CAP v2	CAMEL 2	短信业务 SMS
CAP v3	CAMEL 3	支持 GPRS 业务
CAP v4	CAMEL 4	支持 GPRS 业务

简介 CAP v2 业务



□□□□□□



□□

□□□□

- □□□□□□ OmniSS7
- □ MSC/GMSC (gsmSSF) □ M3UA □□

- `ocscs` (OCS) `API` `ocscs`

`CAMEL`

`config/runtime.exs` `CAMEL`

```

config :omniss7,
  # 配置 - 配置 CAP/CAMEL 配置
  cap_client_enabled: true,
  camelgw_mode_enabled: true,

  # 配置
  map_client_enabled: false,
  hlr_mode_enabled: false,
  smsc_mode_enabled: false,

  # CAP/CAMEL 配置
  # 配置 CAP 版本
  # 版本: :v1, :v2, :v3, :v4
  cap_version: :v2,

  # OCS 配置
  ocs_enabled: true,
  ocs_url: "http://your-ocs-server/api/charging",
  ocs_timeout: 5000, # 超时
  ocs_auth_token: "your-api-token" # 配置 OCS 认证

  # CAMEL 配置 M3UA 配置
  # 配置 ASP 配置 CAP 配置
  cap_client_m3ua: %{
    mode: "ASP",
    callback: {CapClient, :handle_payload, []},
    process_name: :camelgw_client_asp,

    # 配置 CAMEL GW 配置
    local_ip: {10, 179, 4, 13},
    local_port: 2905,

    # 配置 MSC/GMSC - gsmSSF 配置
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,

    # M3UA 配置
    routing_context: 1,
    network_appearance: 0,
    asp_identifier: 13
  }

```

Web UI

Web UI CAMEL

```
config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "SS7"},
    {SS7.Web.TestClientLive, "/client", "SS7"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {SS7.Web.CAMELSessionsLive, "/camel_sessions", "CAP"},
    {SS7.Web.CAMELRequestLive, "/camel_request", "CAP"}
  ],
  page_order: ["/events", "/client", "/m3ua", "/camel_sessions",
    "/camel_request", "/application", "/configuration"]
```

3GPP CAP

3GPP 23.068 gsmSSF → gsmSCF

消息	消息 ID	消息名称	消息内容
InitialDP	0	InitialDP - InitialDP	handle_initial_dp/1
EventReportBCSM	6	EventReportBCSM EventReportBCSM EventReportBCSM	handle_event_report_bcsn/1
ApplyChargingReport	71	ApplyChargingReport gsmSSF ApplyChargingReport	handle_apply_charging_report
AssistRequestInstructions	16	AssistRequestInstructions gsmSRF AssistRequestInstructions	handle_assist_request_instructions

gsmSCF → gsmSSF

구분	구분 번호	구분 번호	구분
Connect	20	구분 구분 구분 구분 구분 구분 구분 구분 구분	CapRequestGenerator.connect_rec
Continue	31	구분 구분 구분 구분 구분 구분 구분 구분 구분	CapRequestGenerator.continue_re
ReleaseCall	22	구분 구분/ 구분 구분 구분 구분	CapRequestGenerator.release_cal
RequestReportBCSMEvent	23	구분 구분 구분	CapRequestGenerator.request_rep

項目	項目 項目	項目 項目	項目
		項目 項目 項目 項目 項目	
ApplyCharging	35	項目 項目 項目 項目 項目 項目 項目	CapRequestGenerator.apply_charg

Web UI 項目

CAMEL 項目

URL: `http://localhost/camel_sessions`

項目項目 CAMEL 項目

項目

- 項目項目 - 項目 2 項目
- 項目 - OTID項目 ID項目項目
- CAP** 項目 - 項目 InitialDP 項目項目 (CAP v1/v2/v3/v4)
- 項目 - IMSI項目A 項目B 項目項目
- 項目 - 項目項目項目
- 項目項目 - 項目項目項目

項目

- ID - IMSI - OTID

- InitialDP
-
- CDR

CAP InitialDP CAP MSC CAP

CAMEL

URL: http://localhost/camel_request

CAP

- InitialDPConnectReleaseCall
-
- **SCCP/M3UA** -
- 20
- OTID
- /

1. **InitialDP** -

-
- A
- B

2. **Connect** -

-

3. **ReleaseCall** -

- 16=17=31=

4. **RequestReportBCSMEEvent** -

- oAnsweroDisconnecttAnswertDisconnect

5. **Continue** -

-

6. **ApplyCharging** -

- 1-864000
-
- CAMEL

SCCP

-
-
- SSN146 = gsmSSF
- SSN146

M3UA

- OPC5013
- DPC5011

OCS

1. (InitialDP)

MSC InitialDP CAMELGW

1. CAP - CAP v1/v2/v3/v4
2. CAP - IMSI/

3. **MSC** - `InitiateSession` API
4. **SessionStore** - MaxUsage `30` **ms**
5. **SessionStore** - `SessionStore` (ETS `1`) **ms** CAP **ms**
6. **MSC** - `RequestReportBCSMEvent` + `Continue` **ms** CAP **ms**

ms

```
# ms InitialDP ms
%{
  imsi: "310150123456789",
  calling_party_number: "14155551234",
  called_party_number: "14155556789",
  service_key: 1,
  msc_address: "19216800123",
  cap_version: :v2 # ms
}

# ms ms
{:ok, %{max_usage: 30}} # ms 30 ms

# SessionStore ms
%{
  call_id: "CAMEL-4B000173",
  initial_dp_data: %{...},
  cap_version: :v2, # ms
  start_time: 1730246400,
  state: :initiated
}
```

2. **ms** (EventReportBCSM - oAnswer)

ms

1. **ms** **oAnswer** **ms** - **ms** MSC
2. **ms** **OCS** - `UpdateSession` `ms=0`
3. **ms** **ms** - OCS **ms**
4. **ms** **ms** - `SessionStore` `ms` `:answered`
5. **ms** **ms** - `MSC` `ms` `Continue`

3. **ms**

0000000000000000

```
# 30
OCS.Client.update_session(call_id, %{}, current_usage)
```

MaxUsage 0 → ReleaseCall

4. EventReportBCSM - oDisconnect

000000

- oDisconnect - MSC
- 000000 - 000000
- OCS - TerminateSession API
- CDR - OCS 000000
- 0000 - SessionStore 000
- ReleaseCall - 0000 MSC

CDR

CDR OCS 00000000

CAMEL CDR

- Account - IMSI 00000
 - Destination - 00000
 - OriginID - 0000000 (CAMEL-OTID)
 - Usage - 0000000000
 - Cost - 0000
 - IMSI - 000 IMSI
 - CallingPartyNumber - A
 - CalledPartyNumber - B
 - MSCAddress - MSC 000
 - ServiceKey - CAMEL 0000
-

00

0000000000000000

1. 0000000000

`http://localhost/camel_request`

2. 00 InitialDP

- 00000000 "InitialDP"
- 000000 100
- 000000 14155551234
- 000000 14155556789
- 00 "00 InitialDP 00"
- 000000 OTID

3. 000000

- 00000000 `http://localhost/camel_sessions`
- 000000 "0000" 000000

4. 00000000

- 00000000
- 00 "EventReportBCSM"
- 000000 oAnswer
- 00 "00 EventReportBCSM 00"
- 00000000 "0000"

5. 000000

- 00 "ReleaseCall"
- 000000 16 000000
- 00 "00 ReleaseCall 00"
- 00000000 "0000"

□□□□ MSC □□□□

MSC CAMEL

MSC/VLR CAMEL

```
# 0000 MSC 00
ADD CAMELSERVICE:
    SERVICEID=1,
    SERVICEKEY=100,
    GSMSCFADDR="55512341234", # CAMELGW 0000
    DEFAULTCALLHANDLING=CONTINUE;

ADD CAMELSUBSCRIBER:
    IMSI="310150123456789",
    SERVICEID=1,
    TRIGGERTYPE=TERMCALL;
```

□□□□

00 CAMELGW 00000000 CAP 000

```
# tail -f /var/log/omniss7/omniss7.log

# grep "CAP:" /var/log/omniss7/omniss7.log

# curl http://localhost/api/events | jq '.[] | select(.map_event | startswith("CAP:"))'
```

□ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
# 100 InitialDP
for i in {1..100}; do
    curl -X POST http://localhost/api/camel/initial_dp \
        -H "Content-Type: application/json" \
        -d '{
            "service_key": 100,
            "calling_number": "1415555'$i'",
            "called_number": "14155556789"
        }'
    sleep 0.1
done
```

□□□□□

Prometheus □□

CAMELGW □ <http://localhost:8080/metrics> □□□□□

□□ **CAP** □□□□

- `cap_requests_total{operation}` - □□□□□□□□□□initialDP□
requestReportBCSMEEvent□□□□□ CAP □□

□□ **MAP/API** □□□

- `map_requests_total{operation}` - □□□□□□□□□ MAP □□
- `map_request_duration_milliseconds{operation}` - □□□□□□□□□
- `map_pending_requests` - □□□ MAP □□□□□

M3UA STP □□□□□□□ STP □□□□

- `m3ua_stp_messages_received_total{peer_name,point_code}` - □□□□□□□□□□
- `m3ua_stp_messages_sent_total{peer_name,point_code}` - □□□□□□□□□
- `m3ua_stp_routing_failures_total{reason}` - □□□□□□□□□□

□□□□□

```
# CAP 测试
curl http://localhost:8080/metrics | grep cap_requests_total

# 测试 InitialDP
curl http://localhost:8080/metrics | grep
'cap_requests_total{operation="initialDP"}'

# MAP 测试
curl http://localhost:8080/metrics | grep map_pending_requests
```

测试

```
# 测试 M3UA 测试
curl http://localhost/api/m3ua-status

# 测试 OCS 测试
curl http://localhost/api/ocs-status

# 测试 测试
curl http://localhost/api/camel/sessions/count
```

测试

在 `config/runtime.exs` 中添加配置

```
config :logger,
  level: :info # 测试:debug, :info, :warning, :error

# 测试 CAP 测试
config :logger, :console,
  metadata: [:cap_operation, :otid, :call_id]
```

環境構築

環境構築 **CAP** 実行

環境構築コマンドを実行 MSC 実行 InitialDP

実行

1. M3UA 実行 `curl http://localhost/api/m3ua-status`
2. MSC CAMEL 実行 `gsmSCF` 実行
3. SCCP 実行 `CAMELGW`
4. 実行 `SCTP` 実行 2905

実行

```
# M3UA 実行
tcpdump -i eth0 sctp

# MSC 実行 CAMELGW
ss -tuln | grep 2905
```

環境構築 **OCS** 実行

環境構築 `INSUFFICIENT_CREDIT` 実行

実行

1. OCS 実行 `curl http://your-ocs-server/api/health`
2. 実行 OCS 実行
3. OCS 実行
4. OCS 実行
5. 実行

実行

- 実行 `runtime.exe` 実行 OCS URL 実行
- 実行 OCS 実行

- curl 调用 OCS API
- 数据解析

数据解析

EventReportBCSM 消息 "消息"

OTID 消息

消息

1. 消息 OTID
2. 消息
3. DTID 消息 Continue/End 消息 OTID 消息

```
# 消息
iex> CAMELGW.SessionStore.list_sessions()
```

数据解析

消息 Failed to decode InitialDP

CAP 消息

消息

1. CAP 消息 MSC 消息
2. ASN.1 消息
3. PCAP 消息 Wireshark 消息

```
# CAP 消息
tcpdump -i eth0 -w cap_trace.pcap sctp port 2905
```

```
# Wireshark 消息 m3ua
wireshark cap_trace.pcap
```

OmniSS7

OmniSS7 CAP 配置

OmniSS7 配置 CAP 参数

```
config :omniss7,  
  cap_version_map: %{\n    100 => :v2, # 100 CAP v2\n    200 => :v3, # 200 CAP v3\n    300 => :v4  # 300 CAP v4\n  },\n  cap_version: :v2 #
```

OmniSS7

CAMEL 配置 OmniSS7 参数

- 配置 CAP 参数 (v1/v2/v3/v4)
- 配置 OCS 参数
- 配置 (Connect, Release, Continue)
- 配置 ETS 参数
- 配置 Web UI 参数
- 配置
- 配置 CDR 参数
- 配置

配置

- CAMEL 配置
- 配置 - CAP 参数

OmniSS7 CAMEL 配置

版本 1.0

日期 2025-10-26

□□□□□□

← □□□□

□□□□□□ OmniSS7 □□□□□□□□□□

□□

1. Web UI □□
2. API □□
3. □□□□□
4. □□□□
5. □□□□□ SCTP □□□

Web UI □□

Web UI □□□□□□□ Web □□□□□□□□

API

- API - API SS7 API
- API - API
- API - API
- **M3UA** API - M3UA API STP API
- API - API SMSG API

API Web UI

1. API Web UI
2. API `http://localhost`
3. API

Swagger API

API API

http://your-server/swagger

Web UI

config/runtime.exs

```
config :control_panel,
  #
  page_order: ["/events", "/application", "/configuration"],

  # Web
  web: %{
    listen_ip: "0.0.0.0",      # IP 0.0.0.0
    port: 80,                 # HTTP 443
    hostname: "localhost",    # URL
    enable_tls: false,        # true HTTPS
    tls_cert: "cert.pem",     # TLS
    tls_key: "key.pem"        # TLS
  }
```

項目	型別	デフォルト値	説明
page_order	配列	<code>["/events", "/application", "/configuration"]</code>	ページ順序
listen_ip	文字列	<code>"0.0.0.0"</code>	Web サービス IP アドレス
port	整数	<code>80</code>	HTTP または HTTPS ポート番号 (443)
hostname	文字列	<code>"localhost"</code>	URL ホスト名
enable_tls	ブール値	<code>false</code>	TLS を有効にするかどうか
tls_cert	文字列	<code>"cert.pem"</code>	TLS 証明書ファイル名
tls_key	文字列	<code>"key.pem"</code>	TLS 私鍵ファイル名

ログ

ファイル `config/runtime.exs` を参照してください。

```
config :logger,
  level: :debug # 値: :debug, :info, :warning, :error
```

ログレベル

- `:debug` - デバッグ
- `:info` - インフォ
- `:warning` - ウォーニング
- `:error` - エラー

API 接口

API 接口 URL

`http://your-server/api`

接口说明

- **200** - 成功
- **400** - 参数错误
- **504** - 网关超时

OpenAPI 接口

`http://your-server/swagger.json`

接口文档

Prometheus 接口

`http://your-server/metrics`

接口说明

M3UA/SCTP 接口

- SCTP 连接管理
- M3UA ASP 消息
- 路由/负载均衡

M2PA

- DOWN → ALIGNMENT → PROVING → READY
- /
- SCTP

STP

- /
-
-

MAP

- MAP
-
-

CAP

- CAP
- CAMEL

SMSc

-
-
-

Grafana

OmniSS7 Prometheus Grafana

□□□□

□□□□

1. □□□□

- □□□□ VLAN □
- □□□□□□□□
- □□□□□□□□ SCTP

2. **Web UI** □□

- □□□□□□□□ TLS
- □□□□□□□□□□
- □□□□ IP

3. **API** □□

- □□□□□□
- □□ API □□□ OAuth
- □□□□□□□□

□□□□

1. **TPS** □□

- □□□□ TPS
- □□□□□□
- □□ SCTP □□□

2. □□□□□

- □□□□
- □□□□□
- □□□□□

3. **M3UA** □□

- SCTP
-
-

SCTP

SCTP

SCTP M3UA IP

- SCTP
- - SCTP
-
- SCTP

M3UA IP SCTP

1. IP
2. SCTP
3. SCTP
- 4.
5. SCTP

SCTP IP IP

IP

```
# 00 IP - 0000
local_ip: {10, 179, 4, 10}
```

00 **IP**00000000

```
# 00 IP - 00000
# 000 IP 00000 IP 00000
local_ip: [{10, 179, 4, 10}, {10, 179, 4, 11}]
```

0000

00 **1**00000000 **STP** 000

```
# STP 00000000
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "Partner_STP_Redundant",
      role: :client,
      # 000000000000 IP 00000
      local_ip: [{213, 57, 23, 200}, {213, 57, 23, 201}],
      local_port: 0,
      # 000000000000
      remote_ip: [{213, 57, 23, 100}, {213, 57, 23, 101}],
      remote_port: 2905,
      routing_context: 1,
      point_code: 100,
      network_indicator: :international
    }
  ]
```

00 **2**00000000 **MAP** 000

```
# MAP 配置
config :omniss7,
  map_client_enabled: true,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    # 本地 IP 地址
    local_ip: [{10, 0, 0, 100}, {10, 0, 0, 101}],
    local_port: 2905,
    # STP 配置
    remote_ip: [{10, 0, 0, 1}, {10, 0, 0, 2}],
    remote_port: 2905,
    routing_context: 1
  }
}
```

3. STP 配置

```
# STP 配置
config :omniss7,
  m3ua_stp: %{
    enabled: true,
    # 本地 IP 地址
    local_ip: [{172, 16, 0, 10}, {172, 16, 0, 11}],
    local_port: 2905,
    point_code: 100
  }
}
```

4. 配置示例

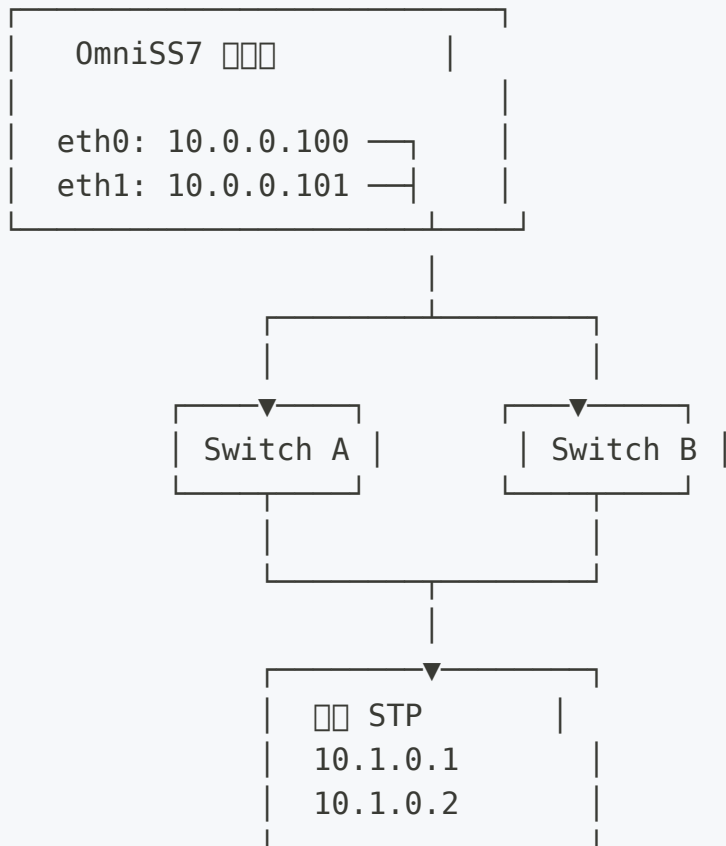
```

# 配置m3ua对等体
config :omniss7,
  m3ua_peers: [
    # 配置对等体1 - 本地IP
    %{
      peer_id: 1,
      name: "Legacy_STP",
      role: :client,
      local_ip: {10, 0, 0, 1},      # 本地IP地址
      local_port: 0,
      remote_ip: {10, 0, 0, 10},
      remote_port: 2905,
      routing_context: 1,
      point_code: 100
    },
    # 配置对等体2 - 冗余IP
    %{
      peer_id: 2,
      name: "Redundant_STP",
      role: :client,
      local_ip: [{10, 0, 0, 2}, {10, 0, 0, 3}], # IP地址列表
      local_port: 0,
      remote_ip: [{10, 0, 0, 20}, {10, 0, 0, 21}],
      remote_port: 2905,
      routing_context: 2,
      point_code: 200
    }
  ]
]

```

配置m3ua对等体

配置1个NIC对等体



□□□

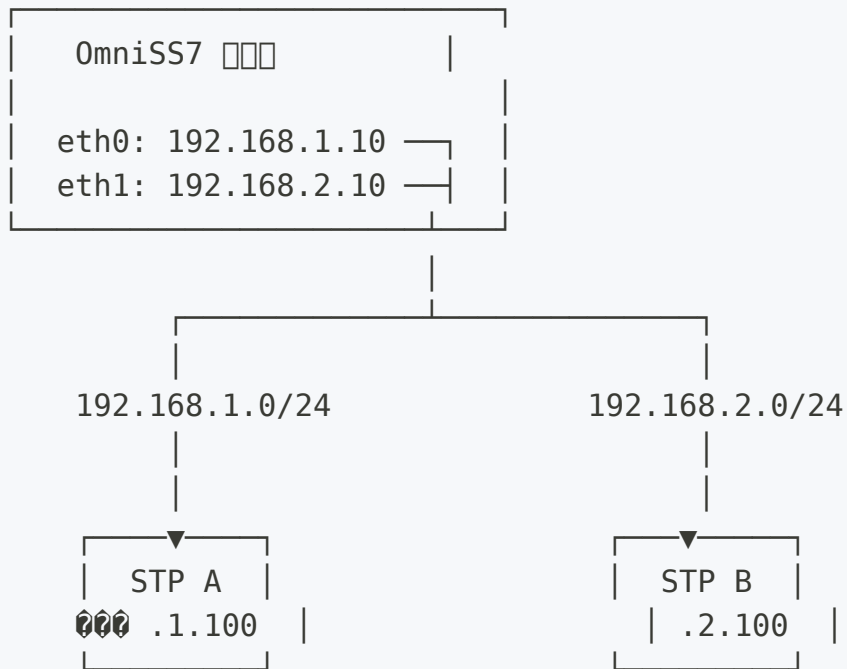
```

local_ip: [{10, 0, 0, 100}, {10, 0, 0, 101}] # □□ NIC
remote_ip: [{10, 1, 0, 1}, {10, 1, 0, 2}] # □□□□
  
```

□□□

- □□□ NIC □□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□ 1 □

□□ **2**□□□□□□



```

local_ip: [{192, 168, 1, 10}, {192, 168, 2, 10}]
remote_ip: [{192, 168, 1, 100}, {192, 168, 2, 100}]

```

-
-
-

```

[info] SCTP  2  IP
[info] STP  2  IP

```

```
[warning] [MULTIHOMING] 10.0.0.100 Partner_STP
assoc_id=1
[info] [MULTIHOMING] 10.0.0.101 Partner_STP assoc_id=1
[info] [MULTIHOMING] 10.0.0.100 Partner_STP assoc_id=1
```

Web UI

Web UI

M3UA

- IP 10.0.0.100
- IP 10.0.0.100 (+1) 10.0.0.100 (+2)
- IP /

1.

- NIC
-
-
- -

2. IP

- IP -
- - IP
- -

3.

```
# 断网
sudo ip link set eth0 down

# 断网日志
tail -f /var/log/omniss7.log | grep MULTIHOMING

# 恢复网络
sudo ip link set eth0 up
```

4. 断网测试

- 断网前 IP
- 断网后 IP
- 断网后网络状态

5. 断网恢复

```
# 断网前 IP 断网 Sctp
iptables -A INPUT -p sctp --dport 2905 -s 10.0.0.0/24 -j ACCEPT
iptables -A INPUT -p sctp --dport 2905 -s 10.1.0.0/24 -j ACCEPT
```

断网

断网前 IP

断网后 IP

断网

1. 断网 Erlang Sctp `erl -eval 'gen_sctp:open(9999, [binary, {ip, {127,0,0,1}}]).'`
2. 断网 Sctp `lsmod | grep sctp`
3. 断网 Sctp `sudo modprobe sctp`
4. 断网 IP `ip addr show`

断网前 IP

断网后 IP

目標

1. SCTP の概要
2. 標準化の経緯
3. 標準化された IP の SCTP
4. SCTP の特徴

標準化の経緯

標準化された UP と DOWN の経緯

目標

1. 標準化 - 標準化
2. SCTP の標準化 - 標準化
3. 標準化 SCTP の
4. 標準化 MTU の

標準化

- 標準化SCTP の標準化
- 標準化標準化標準化
- 標準化標準化 1 標準化
- 標準化標準化

標準化

- 標準化 IP の標準化
- 標準化標準化 IP の IP の
- 標準化標準化 STP、HLR、SMSc の MAP の標準化
- **Erlang** の標準化 SCTP の Erlang

標準化

標準化

- M3UA の標準化

- MAP 〇〇〇〇〇
- API 〇〇〇〇
- 〇〇〇〇〇〇

--	--	--	--	--

- M3UA 1
- MAP 10%
- 1000
- API 5%

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

□ □ □ □ □ ◆ ◆

[illegible]

□□□□ (:logger)

```
config :logger,
  level: :debug # :debug | :info | :warning | :error
```

Web UI (control_panel)

```
config :control_panel,
  page_order: ["/events", "/application", "/configuration"],
  web: %{
    listen_ip: "0.0.0.0",
    port: 80,
    hostname: "localhost",
    enable_tls: false,
    tls_cert: "cert.pem",
    tls_key: "key.pem"
  }
end
```

項目	型	単位	初期値	説明
page_order	整数		["/events", "/application", "/configuration"]	ページ番号
web.listen_ip	文字列		"0.0.0.0"	Web サービス IP アドレス
web.port	整数		80	HTTP/HTTPS ポート
web.hostname	文字列		"localhost"	ホスト名
web.enable_tls	ブール値		false	HTTPS の有効化
web.tls_cert	文字列	TLS 証明書	"cert.pem"	TLS 証明書
web.tls_key	文字列	TLS 秘密鍵	"key.pem"	TLS 秘密鍵

M3UA STP 設定 (:omniss7)

```
config :omniss7,
  m3ua_stp: %{
    enabled: false,
    local_ip: {127, 0, 0, 1},
    local_port: 2905
  },
  enable_gt_routing: true,
  m3ua_peers: [...],
  m3ua_routes: [...],
  m3ua_gt_routes: [...]
```

項目	型別	デフォルト	説明
m3ua_stp.enabled	boolean	false	M3UA STP を有効にするかどうか
m3ua_stp.local_ip	list	{127, 0, 0, 1}	M3UA の IP アドレス
m3ua_stp.local_port	integer	2905	M3UA の SCTP ポート番号
enable_gt_routing	boolean	false	グローバルルーティングを有効にするかどうか

M3UA 設定

項目	型	制約	説明
peer_id	整数	必須	ピアID
name	文字列 100文字以内	必須	名前
role	整数	必須	:client または :server
local_ip	整数 4文字列 100文字以内	文字列 :client	ローカルIPアドレス {10, 0, 0, 1} の場合 [{10, 0, 0, 1}, {10, 0, 0, 2}]
local_port	整数	文字列 :client	ローカルポート番号
remote_ip	整数 4文字列 100文字以内	必須	リモートIPアドレス {10, 0, 0, 10} の場合 [{10, 0, 0, 10}, {10, 0, 0, 11}]
remote_port	整数	文字列 :client	リモートポート番号
routing_context	整数	必須	M3UA コンテキスト
point_code	整数	必須	SS7 ポイントコード
network_indicator	整数	必須	:international または :national

M3UA 定義

項目	型	必須	説明
dest_pc	整数	○	宛先ポート
peer_id	整数	??	ピア識別子
priority	整数	○	優先度
network_indicator	整数	○	:international ○ :national

M3UA GT 項目

項目	型	必須	説明
gt_prefix	整数	○	GTプレフィックス
peer_id	整数	○	ピア識別子
priority	整数	○	優先度
description	整数	○	説明
source_ssn	整数	○	送信元 SSN
dest_ssn	整数	○	宛先 SSN

MAP 項目 (:omniss7)

```
config :omniss7,  
  map_client_enabled: false,  
  map_client_m3ua: %{  
    mode: "ASP",  
    callback: {MapClient, :handle_payload, []},  
    process_name: :map_client_asp,  
    local_ip: {10, 0, 0, 100},  
    local_port: 2905,  
    remote_ip: {10, 0, 0, 1},  
    remote_port: 2905,  
    routing_context: 1  
  }  
}
```

項目	型	デフォルト	説明	注
<code>map_client_enabled</code>	boolean	false		MAP クライアント
<code>map_client_m3ua.mode</code>	enum	"ASP"		M3UA モード "ASP" " "SGP"
<code>map_client_m3ua.callback</code>	function	{MapClient, :handle_payload, []}		
<code>map_client_m3ua.process_name</code>	string	:map_client_asp		
<code>map_client_m3ua.local_ip</code>	string	-		IP
<code>map_client_m3ua.local_port</code>	integer	2905		SCTP ポート
<code>map_client_m3ua.remote_ip</code>	string	-		STP/SGP IP
<code>map_client_m3ua.remote_port</code>	integer	2905		SCTP ポート
<code>map_client_m3ua.routing_context</code>	string	-		M3UA コンテキスト

配置参数 (:omniss7)

```
config :omniss7,
  auto_flush_enabled: false,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: nil,
  auto_flush_tps: 10
```

参数	类型	默认值	说明
auto_flush_enabled	布尔	false	是否开启 SMS 自动刷新
auto_flush_interval	整数	10000	自动刷新的间隔
auto_flush_dest_smsc	字符串/nil	nil	指定 SMSC 名称 nil = 默认
auto_flush_tps	整数	10	每秒刷新次数

HTTP API 配置 (:omniss7)

配置 HTTP API 参数

```
config :omniss7,
  smsc_api_base_url: "https://10.5.198.200:8443",
  frontend_name: "omni-smsc01" # 指定 hostname_SMSc
```

API 接口

項目	項目 項目	項目	項目
smssc_api_base_url	項目 項目 項目	項目	項目項目 API 項目 URL
frontend_name	項目 項目 項目	項目	項目項目項目 項目

項目 **API** 項目

- POST /api/frontends - 項目項目項目項目
- POST /api/messages_raw - 項目 SMS 項目
- GET /api/messages - 項目項目項目 smsc 項目
- PATCH /api/messages/{id} - 項目項目項目項目
- PUT /api/messages/{id} - 項目項目項目
- POST /api/events - 項目項目項目
- GET /api/status - 項目項目項目

項目項目

項目項目項目項目 API 項目項目 5 項目項目項目項目項目項目

- 項目項目項目項目SMSc項目
- 項目
- 項目項目項目
- 項目項目JSON 項目

項目項目

- 項目項目項目項目項目項目 SSL 項目
- HTTP 項目 5 項目項目
- 項目項目項目 ISO 8601 項目
- API 項目 JSON 項目項目/項目項目

□□□□

- ← □□□□
- STP □□
- MAP □□□□
- □□□□□
- HLR □□

OmniSS7 □ Omnitouch □□□□□

OmniSS7

← 概要

OmniSS7 のインストールと起動

目次

1. 概要
 2. インストール
 3. HLR の設定
 4. SMSc の設定
 5. STP の設定
 6. NAT の設定
 7. M3UA の設定
 8. HTTP の設定
 9. その他
 10. 起動
-

インストール

OmniSS7 のインストールは `config/runtime.exs` を編集して行う。

- **STP** - 設定
- **HLR** - 設定
- **SMSc** - 設定

インストールコマンド: `config/runtime.exs`

配置

配置项

配置项	默认值	是否必填	配置说明	配置位置
map_client_enabled	false	否	MAP 客户端 M3UA 是否启用	否
hlr_mode_enabled	false	否	HLR 模式是否启用	HLR
smsc_mode_enabled	false	否	SMSc 模式是否启用	SMSc

配置:

```
config :omniss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: true,  
  smsc_mode_enabled: false
```

HLR 配置

HLR 配置项

HLR API

項目	項目名	項目値	項目単位	項目説明
HLR API URL	hlr_api_base_url	-		HLR API URL (SSL 対応)
HLR Service Center GT Address	hlr_service_center_gt_address	-		UpdateLocation 用 HLR GT Address
SMSC Service Center GT Address	smsc_service_center_gt_address	-		SRI-for-SM 用 SMSC GT Address

例:

```
config :omniss7,  
  hlr_api_base_url: "https://10.180.2.140:8443",  
  hlr_service_center_gt_address: "55512341111",  
  smsc_service_center_gt_address: "55512341112"
```

MSISDN ↔ IMSI

MSISDN から IMSI を取得するための HLR への MSISDN ↔ IMSI

項目	単位	値	注	説明
hlr_imsi_plmn_prefix	1桁 1桁	"50557"	1桁	IMSI の MCC + MNC の値
hlr_msisdn_country_code	1桁 1桁	"61"	1桁	IMSI → MSISDN の変換に使用する国コード
hlr_msisdn_nsn_offset	1桁 1桁	0	1桁	MSISDN の NSN のオフセット
hlr_msisdn_nsn_length	1桁 1桁	9	1桁	MSISDN の NSN の長さ

例2 の設定:

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "99",      # 2桁 国コード
  hlr_msisdn_nsn_offset: 2,           # 2桁 NSN オフセット
  hlr_msisdn_nsn_length: 9           # 9桁 NSN
```

例3 の設定:

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "999",     # 3桁 国コード
  hlr_msisdn_nsn_offset: 3,           # 3桁 NSN オフセット
  hlr_msisdn_nsn_length: 8           # 8桁 NSN
```

注: ns_n_offset の値は NSN の長さに依存する

- 国コード "9" の場合 → ns_n_offset: 1
- 国コード "99" の場合 → ns_n_offset: 2

- 0000 "999"03 00→ nsn_offset: 3

InsertSubscriberData (ISD) 00

0 UpdateLocation 00000 VLR 0000000000000000 ISD 0000000000000000 HLR 0000
InsertSubscriberData 000

00	0 0	00	0 0	0
isd_network_access_mode	0 0	:packetAndCircuit	0	00000000 :packetAndCircuit 0 :circuitOnly
isd_send_ss_data	0 0	true	0	000000000000 ISD
isd_send_call_barring	0 0	true	0	000000000000 ISD

00:

```
config :omniss7,  
  isd_network_access_mode: :packetAndCircuit,  
  isd_send_ss_data: true,  
  isd_send_call_barring: true
```

CAMEL 00

00 CAMEL 0000000000000000 CAMEL 0000000000000000 HLR 0000 CAMEL 000

項目	単位	項目	単位	
camel_service_key	11_110			11 S
camel_trigger_detection_point	:termAttemptAuthorized			CAM :te
camel_gsmcf_gt_address	(GT)			CAM

例:

```
config :omniss7,  
  camel_service_key: 11_110,  
  camel_trigger_detection_point: :termAttemptAuthorized
```

例 VLR

例 PRN HLR

項目	単位	項目	単位	項目
home_vlr_prefixes	["5551231"]			VLR GT

例:

```
config :omniss7,  
  home_vlr_prefixes: ["5551231", "5551234"]
```

SMSc 配置

SMS 配置

SMSc API 配置

配置项	数据类型	默认值	是否必填	说明
<code>smsc_api_base_url</code>	字符串	-	是	SMSc API 的 URL，支持 SSL 加密连接
<code>smsc_name</code>	字符串	"{hostname}_SMSc"	是	SMSc 的名称
<code>smsc_service_center_gt_address</code>	字符串	-	是	SMSc 的服务中心地址

示例：

```
config :omniss7,
  smsc_api_base_url: "https://10.179.3.219:8443",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112"
```

注意：配置项 **5** 必须在 `SMS.FrontendRegistry` 中配置

配置参数

参数	类型	默认值	必填	说明
auto_flush_enabled	布尔	true	否	是否启用 SMS 自动刷新
auto_flush_interval	整数	10_000	否	自动刷新的间隔时间
auto_flush_dest_smsc	字符串	-	否	自动刷新的 SMSC 地址
auto_flush_tps	整数	10	否	自动刷新的 TPS 限制

示例:

```
config :omniss7,  
  auto_flush_enabled: true,  
  auto_flush_interval: 10_000,  
  auto_flush_dest_smsc: "ipsmgw",  
  auto_flush_tps: 10
```

STP 配置

M3UA 配置参数 STP 配置

配置 STP 参数

参数	数据类型	默认值	是否必填	说明
m3ua_stp.enabled	布尔	false	否	是否启用 M3UA STP 功能
m3ua_stp.local_ip	IP 地址	{127, 0, 0, 1}	否	本地 IP 地址。SCTP 地址列表: [{10, 0, 0, 1}, {10, 0, 0, 2}]
m3ua_stp.local_port	端口	2905	否	本地端口
m3ua_stp.point_code	点码	-	否	STP 点码 SS7 点码

配置 IP:

```
config :omniss7,
  m3ua_stp: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 2905,
    point_code: 100
  }
```

配置 SCTP 参数:

```
config :omniss7,
  m3ua_stp: %{
    enabled: true,
    # 两个 IP 地址
    local_ip: [{10, 179, 4, 10}, {10, 179, 4, 11}],
    local_port: 2905,
    point_code: 100
  }
```

注意: 两个 SCTP 地址都配置为 **双向 SCTP 连接**

配置项

名称	类型	默认值	必填	说明
enable_gt_routing	布尔	false	否	是否 PC 支持 GT 路由

配置:

```
config :omniss7,
  enable_gt_routing: true
```

配置 NAT

配置 GT 地址时，需要配置 **双向 NAT**

名称	类型	默认值	必填	说明
gt_nat_enabled	布尔	false	否	是否/是否 GT NAT
gt_nat_rules	列表	[]	是	GT 规则

配置: gt_nat_rules 配置 GT 规则

- `calling_prefix`: 國際區號 GT 碼
- `response_gt`: 國際區號

例:

```
config :omniss7,
  gt_nat_enabled: true,
  gt_nat_rules: [
    # 8772 國際 GT 碼 "55512341112"
    %{calling_prefix: "8772", response_gt: "55512341112"},
    # 8773 國際 GT 碼 "55512341111"
    %{calling_prefix: "8773", response_gt: "55512341111"},
    # 國際區號
    %{calling_prefix: "", response_gt: "55512311555"}
  ]
```

例: 國際區號 GT NAT 碼

M3UA 協議

MAP 協議 M3UA 協議 MAP 協議

項目	型	値	注	説明
<code>map_client_m3ua.mode</code>	enum	-	必須	モード名 "ASP" または "SGP"
<code>map_client_m3ua.callback</code>	void (*)	-	必須	コールバック関数 {MapClient, :handle_payload, []}
<code>map_client_m3ua.process_name</code>	string	-	必須	プロセス名
<code>map_client_m3ua.local_ip</code>	ipaddr_t	-	必須	ローカル IP アドレス {10, 0, 0, 1} または SCTP アドレス [{10, 0, 0, 1}, {10, 0, 0, 2}]
<code>map_client_m3ua.local_port</code>	uint16_t	2905	必須	SCTP ポート
<code>map_client_m3ua.remote_ip</code>	ipaddr_t	-	必須	リモート STP/SGW IP アドレス {10, 0, 0, 10} または SCTP アドレス [{10, 0, 0, 10}, {10, 0, 0, 11}]
<code>map_client_m3ua.remote_port</code>	uint16_t	2905	必須	SCTP ポート
<code>map_client_m3ua.routing_context</code>	uint32_t	-	必須	M3UA ルーティング ID

ローカル IP:

```

config :omniss7,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }

```

####SCTP ####:

```

config :omniss7,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    # #### IP ####
    local_ip: [{10, 179, 4, 11}, {10, 179, 4, 12}],
    local_port: 2905,
    # #### IP #### STP ####
    remote_ip: [{10, 179, 4, 10}, {10, 179, 4, 20}],
    remote_port: 2905,
    routing_context: 1
  }

```

| : SCTP ##### Sctp #####

HTTP

REST API HTTP #####

項目	項目	項目	項目	項目
start_http_server	項目	true	項目	項目/項目 HTTP 項目項目 8080

項目項目項目項目:

- **IP:** 0.0.0.0項目項目
- 項目: 8080
- 項目: Plug.Cowboy

項目:

```
config :omniss7,  
  start_http_server: true # 項目 false 項目
```

API 項目:

- REST API: http://[server-ip]:8080/api/*
- Swagger UI: http://[server-ip]:8080/swagger
- Prometheus 項目: http://[server-ip]:8080/metrics

項目項目項目項目 **API** 項目

項目項目項目

Mnesia 項目項目項目項目

項目	項目	項目	項目	項目
mnesia_storage_type	項目	:disc_copies	項目	Mnesia 項目項目 :disc_copies 項目 :ram_copies

例:

```
config :omniss7,  
  mnesia_storage_type: :disc_copies # 例  
  # mnesia_storage_type: :ram_copies # 例
```

例:

- `:disc_copies` - 例 - 例
- `:ram_copies` - 例 - 例

Mnesia 例:

- `m3ua_peer` - M3UA 例
- `m3ua_route` - 例
- `m3ua_gt_route` - 例

例: `Mnesia.{node_name}/` 例

例

例

例

例	例	例
MAP 例: 10 例	例 MAP 例 10 例	例
ISD 例: 10 例	例 ISD 例 10 例	例

HTTP

HTTP IP: 0.0.0.0		
HTTP : 8080	REST API 8080	

SSL

HLR API SSL:	SSL	
SMSc API SSL:	SSL	

5	SMSc 5	

Web UI

5	
2	

□□□□

□□ **HLR** □□

```
config :omniss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: true,  
  smsc_mode_enabled: false,  
  
  hlr_api_base_url: "https://10.180.2.140:8443",  
  hlr_service_center_gt_address: "55512341111",  
  smsc_service_center_gt_address: "55512341112",  
  
  map_client_m3ua: %{  
    mode: "ASP",  
    callback: {MapClient, :handle_payload, []},  
    process_name: :hlr_client_asp,  
    local_ip: {10, 179, 4, 11},  
    local_port: 2905,  
    remote_ip: {10, 179, 4, 10},  
    remote_port: 2905,  
    routing_context: 1  
  }
```

☐☐ **SMSc** ☐☐

```
config :omniss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: false,  
  smsc_mode_enabled: true,  
  
  smsc_api_base_url: "https://10.179.3.219:8443",  
  smsc_name: "ipsmgw",  
  smsc_service_center_gt_address: "55512341112",  
  
  auto_flush_enabled: true,  
  auto_flush_interval: 10_000,  
  auto_flush_dest_smsc: "ipsmgw",  
  auto_flush_tps: 10,  
  
  map_client_m3ua: %{  
    mode: "ASP",  
    callback: {MapClient, :handle_payload, []},  
    process_name: :stp_client_asp,  
    local_ip: {10, 179, 4, 12},  
    local_port: 2905,  
    remote_ip: {10, 179, 4, 10},  
    remote_port: 2905,  
    routing_context: 1  
  }
```

配置 STP

```
config :omniss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: false,  
  smsc_mode_enabled: false,  
  
  enable_gt_routing: true,  
  mnesia_storage_type: :disc_copies,  
  
  m3ua_stp: %{  
    enabled: true,  
    local_ip: {10, 179, 4, 10},  
    local_port: 2905,  
    point_code: 100  
  },  
  
  map_client_m3ua: %{  
    mode: "ASP",  
    callback: {MapClient, :handle_payload, []},  
    process_name: :stp_client_asp,  
    local_ip: {10, 179, 4, 10},  
    local_port: 2906,  
    remote_ip: {10, 179, 4, 11},  
    remote_port: 2905,  
    routing_context: 1  
  }
```

配置

配置: 32

配置:

- 配置: 3 配置
- HLR 配置: 13 配置
- SMSc 配置: 7 配置

- STP 数量: 5 个
- M3UA 数量: 8 个
- HTTP 数量: 1 个
- 接口: 1 个

配置项名称:

- `hlr_api_base_url` HLR 接口
 - `hlr_service_center_gt_address` HLR 接口
 - `smsc_api_base_url` SMSc 接口
 - `smsc_service_center_gt_address` SMSc/HLR 接口
 - 配置 `map_client_m3ua.*` 接口
 - `m3ua_stp.point_code` 指定 STP
-

接口

- **HLR 接口** - HLR 接口
- **SMSc 接口** - SMSc 接口
- **STP 接口** - STP 接口
- **API 接口** - REST API 接口
- **Web UI 接口** - Web 接口

□□□□ NAT □□

11

GT NAT OmniSS7 GT GT
GT GT GT /
GT GT

GT NAT

[illegible]

111

1. `called_prefix` `calling_prefix`
- 2.
3. `weight` `=`
4.
 -
 -
 -
 -

--	--	--	--	--

□ □ □ □ □

- `weight` = 权重
- `response_gt` GT

- calling_prefix 000000 GT 00
- called_prefix 000000 GT 00

000

```
gt_nat_rules: [  
  # 0000000000000000 - 000000  
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,  
  response_gt: "111111"},  
  
  # 0000 - 000000  
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"},  
  %{called_prefix: "555", weight: 10, response_gt: "333333"},  
  
  # 00000 - 000000  
  %{weight: 100, response_gt: "999999"}  
]
```

00

000000

000000000000000000000000 GT 000000

- 00 **A** 0000 GT 111111 0000 111111 0000
- 00 **B** 0000 GT 222222 0000 222222 0000

00 GT NAT000000000000000000000000 GT NAT000 OmniSS7 0000000000000000

0000

000 HLR 0 SSMSc 00000000000000

- 0000 0000 GT 555000
- 0000000 **1** 00 GT 555001
- 0000000 **2** 00 GT 555002

GT NAT 000000000000000000000000 GT 000000

□□□□□

□□□□□□□□□□

- □□□□□□ GT □□□□ GT
- □□□□□□□□ GT
- □□□□□□□□ GT □□□□

□□□□

□□□□□□

1. □□□□□OmniSS7 □□□□□□□□□□ SCCP □□□
 - □□□ GT□55512341112□□□□ GT□
 - □□□ GT□877234567□□□□ GT□
2. **GT NAT** □□□□□□□□ GT 877234567 □□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□□8772 □□ 877234567□
4. □□ **GT** □□□□□□□□□□□ response_gt□□□□□55512341112□
5. □□□□□□SCCP □□□□□
 - □□□ GT□877234567□□□□ - □□□□ GT□
 - □□□ GT□55512341112□NAT'd GT□

□□□□□□□□

GT NAT □□□ SS7 □□□□□□□□

SCCP □□□□□□□□

- □□□□□□□□ SCCP □□/□□ GT □□
- ISD□InsertSubscriberData□□□□
- UpdateLocation □□

- 配置

MAP 配置

- **SRI-for-SM** 配置 `networkNode-Number` 和 `SMS-Sc GT`
- **UpdateLocation** 配置 `hlr-Number`
- **InsertSubscriberData** 配置 `ISD` 和 `HLR GT`

配置

配置

配置 `config/runtime.exs`

```
config :omniss7,
  # 配置 GT NAT
  gt_nat_enabled: true,

  # 配置 GT NAT 规则
  gt_nat_rules: [
    # 配置 1 规则 "8772" 和 "55512341112"
    %{calling_prefix: "8772", response_gt: "55512341112"},

    # 配置 2 规则 "8773" 和 "55512341111"
    %{calling_prefix: "8773", response_gt: "55512341111"},

    # 配置默认规则
    %{calling_prefix: "", response_gt: "55512311555"}
  ]
```

配置

配置 `config/runtime.exs` 中的 `NAT` 配置

項目	型	範囲	説明
<code>gt_nat_enabled</code>	bool	bool	GT NAT 有効/無効
<code>gt_nat_rules</code>	list	list	GT NAT ルール

定義

GT NAT の定義

```
%{
  calling_prefix: "8772",      # 発信元 GT 番号
  called_prefix: "555",       # 着信元 GT 番号
  weight: 10,                 # 優先度 = 10
  response_gt: "55512341112" # 応答 GT
}
```

実装

- `calling_prefix` 発信元 GT 番号
 - `String.starts_with?/2` で確認
 - 空文字列 `""` は `nil` と見做す
 - GT 番号
- `called_prefix` 着信元 GT 番号
 - `String.starts_with?/2` で確認
 - 空文字列 `""` は `nil` と見做す
 - GT 番号
- `weight` 優先度
 - 0 から 100 の範囲
 - 0 は最低
 - 100 は最高

- **response_gt** 00000000000000000000
 - 00000000 E.164 000000
 - 00000000 GT 0000

000000 **calling_prefix** 0 **called_prefix** 00000000000000000000000000000000/000000

00000000

000000 00000000 0000000000 0000000000

000000

1. 000000000000000000
 - 000000 **calling_prefix**0000000000 GT
 - 000000 **called_prefix**0000000000 GT
 - 00000000000000000000
 - 00000000000000000000
2. 000000000000000000
 - 0000000000 - 0000000000
 - 0000000000000000 - 00 = 000000
3. 0000000000000000

0000

```

# []
gt_nat_rules: [
  # 1 - 
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,
  response_gt: "111111"},

  # 10 - 
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, #
  %{called_prefix: "555", weight: 10, response_gt: "333333"}, #
  # 100 - 
  %{weight: 100, response_gt: "444444"} # 
]

# 
# "877234567" "555123" -> "111111" 1 
# "877234567" "999999" -> "222222" 10 
# "999999999" "555123" -> "333333" 10 
# "999999999" "888888" -> "444444" 100

```

1

SMS GT

```

config :omniss7,
  gt_nat_enabled: true,

  # SMS Sc GT NAT
  smsc_service_center_gt_address: "5551000",

  # GT NAT
  gt_nat_rules: [
    # A 4412 GT 5551001
    %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},

    # B 4413 GT 5551002
    %{calling_prefix: "4413", weight: 10, response_gt: "5551002"},

    # SMS Sc GT
    %{weight: 100, response_gt: "5551000"}
  ]

```

GT NAT

```

44121234567 SRI-for-SM
  GT5551001 A
  GT44121234567 A GT

GT NAT
  "44121234567" "4412"
  response_gt "5551001"

SRI-for-SM 44121234567
  GT44121234567
  GT5551001NAT'd
  networkNode-Number5551001 MAP

```

2 GT HLR

HLR GT

```

config :omniss7,
  gt_nat_enabled: true,
  hlr_service_center_gt_address: "555000", # [] HLR GT

  gt_nat_rules: [
    # [] VLR[] 5551
    %{calling_prefix: "5551", weight: 10, response_gt: "555100"},

    # [] VLR[] 5552
    %{calling_prefix: "5552", weight: 10, response_gt: "555200"},

    # [] VLR[] 5553
    %{calling_prefix: "5553", weight: 10, response_gt: "555300"},

    # []
    %{weight: 100, response_gt: "555000"}
  ]

```

3

GT GT

```

config :omniss7,
  gt_nat_enabled: true,
  hlr_service_center_gt_address: "123456789", # [] GT

  gt_nat_rules: [
    # []
    %{calling_prefix: "555", weight: 10, response_gt:
"987654321"}, # [] GT
    %{calling_prefix: "666", weight: 10, response_gt:
"987654321"}, # [] GT

    # [] GT
    %{weight: 100, response_gt: "123456789"} # [] GT
  ]

```

例 4 電話番号別対応

電話番号別 GT 対応を定義する GT 設定 GT

```
config :omniss7,  
  gt_nat_enabled: true,  
  
  gt_nat_rules: [  
    # 電話番号 SMS GT 5551xxx 対応 GT  
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},  
  
    # 電話番号 GT 5552xxx 対応 GT  
    %{called_prefix: "5552", weight: 10, response_gt: "555200"},  
  
    # 電話番号 GT 5553xxx 対応 GT  
    %{called_prefix: "5553", weight: 10, response_gt: "555300"},  
  
    # 共通  
    %{weight: 100, response_gt: "555000"}  
  ]
```

例

```
電話番号 GT 555100 SMS GT  
GT 441234567
```

```
GT NAT  
GT "555100" 対応 "5551"  
response_gt "555100"
```

```
電話番号 GT 555100
```

例 5 番号 + 対応

電話番号別 GT 対応

```

config :omniss7,
  gt_nat_enabled: true,

  gt_nat_rules: [
    # 0000 A 0000 SMS GT - 00000000 10
    %{calling_prefix: "4412", called_prefix: "5551", weight: 1,
    response_gt: "555101"},

    # 0000 B 0000 SMS GT - 00000000 10
    %{calling_prefix: "4413", called_prefix: "5551", weight: 1,
    response_gt: "555102"},

    # 00000000 SMS GT - 0000000000 100
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},

    # 0000 A 0000 GT - 00000000 100
    %{calling_prefix: "4412", weight: 10, response_gt: "555200"},

    # 0000 - 00000000 1000
    %{weight: 100, response_gt: "555000"}
  ]

```

00000

```

# 0000 A 00 SMS GT
000"441234567"0000"555100"
→ 0000 1 00000000 → "555101"

# 0000 A 0000 GT
000"441234567"0000"555200"
→ 0000 10 00000000 → "555200"

# 00000000 SMS GT
000"999999999"0000"555100"
→ 0000 10 00000000 → "555100"

# 0000000000 GT
000"999999999"0000"555200"
→ 0000 100 000 → "555000"

```

配置

GT NAT 配置 OmniSS7 配置

HLR 配置

GT NAT 配置

- UpdateLocation 配置 HLR GT
- InsertSubscriberData 配置 HLR GT
- SendAuthenticationInfo 配置
- 配置

配置 HLR 配置 HLR 配置

配置

```
config :omniss7,  
  hlr_mode_enabled: true,  
  hlr_service_center_gt_address: "5551234567", # 配置 HLR GT  
  
  gt_nat_enabled: true,  
  gt_nat_rules: [  
    %{calling_prefix: "331", weight: 10, response_gt:  
      "5551234568"}, # 配置  
    %{calling_prefix: "44", weight: 10, response_gt:  
      "5551234569"}, # 配置  
    %{weight: 100, response_gt: "5551234567"} # 配置  
  ]
```

SMSc 配置

GT NAT 配置

- SRI-for-SM 配置 networkNode-Number 配置 - 配置 SRI-for-SM 配置
- MT-ForwardSM 配置

配置 SMSc 配置 SMSc 配置

□□□

```
config :omniss7,
  smsc_mode_enabled: true,
  smsc_service_center_gt_address: "5559999", # □□ SSMSc GT

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "1", weight: 10, response_gt: "5559991"},
# □□
    %{calling_prefix: "44", weight: 10, response_gt: "5559992"},
# □□
    %{calling_prefix: "86", weight: 10, response_gt: "5559993"},
# □□
    %{weight: 100, response_gt: "5559999"} # □□□□□
  ]
```

CAMEL □□□□

GT NAT □□□

- □□ SCCP □□□□gsmSCF GT □□□□□□
- CAMEL/CAP □□□□□InitialDP□EventReportBCSM □□
- RequestReportBCSMEvent □□
- ApplyCharging ◆◆◆◆□
- Continue □□

□□□

```

config :omniss7,
  camelgw_mode_enabled: true,
  camel_gsmSCF_gt_address: "55512341112", # [] gsmSCF GT

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "555", weight: 10, response_gt:
"55512341111"}, # [] A
    %{calling_prefix: "666", weight: 10, response_gt:
"55512311555"}, # [] B
    %{weight: 100, response_gt: "55512341112"} # []
  ]

```

[] [] gsmSCF [] gsmSSF [] gsmSCF GT []
 []GT NAT [] gsmSSF [] GT

[] [] [] [] []

[] **GT NAT** []

GT NAT []

```

# []
[info] GT NAT [SRI-for-SM []]: [] GT 877234567 -> [] GT
55512341112
[info] GT NAT [UpdateLocation ISD]: [] GT 331234567 -> [] GT
55512341111
[info] GT NAT [MAP BEGIN []]: [] GT 441234567 -> [] GT 55512311555

```

[] NAT []

- "SRI-for-SM []" - [] SRI-for-SM []
- "UpdateLocation ISD" - [] InsertSubscriberData []
- "UpdateLocation END" - [] UpdateLocation END []
- "MAP BEGIN []" - ❖❖ MAP BEGIN []
- "ISD ACK" - ISD []

- "HLR 0000" - HLR 00000
- "CAMEL 00" - CAMEL/CAP 00000gsmSCF0

00

00000000 GT NAT 000

```
# 00 GT NAT 00
iex> GtNat.validate_config()
{:ok, [
  %{calling_prefix: "8772", weight: 10, response_gt:
"55512341112"},
  %{calling_prefix: "8773", weight: 10, response_gt:
"55512341111"}
]}

# 00000000
iex> GtNat.enabled?()
true

# 00000000
iex> GtNat.get_rules()
[
  %{calling_prefix: "8772", weight: 10, response_gt:
"55512341112"},
  %{calling_prefix: "8773", weight: 10, response_gt:
"55512341111"}
]
```

00 GT NAT

00000000 GT NAT 000

```
# 000000 GT 000000called_gt 0 nil
iex> GtNat.translate_response_gt("877234567", nil, "default_gt")
"55512341112"
```

```
# 00000000 GT 00000
iex> GtNat.translate_response_gt("877234567", "555123",
"default_gt")
"55512341112"
```

```
# 00000000nil 00 GT
iex> GtNat.translate_response_gt_with_logging("877234567", nil,
"default_gt", "test")
# 000GT NAT [test]: 00 GT 877234567 -> 00 GT 55512341112
"55512341112"
```

```
# 0000000000 GT
iex> GtNat.translate_response_gt_with_logging("877234567",
"555123", "default_gt", "test")
# 000GT NAT [test]: 00 GT 877234567, 00 GT 555123 -> 00 GT
55512341112
"55512341112"
```

```
# 00000000000000
iex> GtNat.translate_response_gt("9999999999", "888888",
"default_gt")
"default_gt"
```

0000

000**GT NAT** 0000

00 **1**0000000

```
iex> Application.get_env(:omniss7, :gt_nat_enabled)
true # 00 true
```

00 **2**0000000000

```
iex> Application.get_env(:omniss7, :gt_nat_rules)
[%{calling_prefix: "8772", response_gt: "55512341112"}, ...] # []
[]
```

3 "GT NAT"

GT

GT

```
# 
gt_nat_rules: [
  %{weight: 1, response_gt: "111111"},          # 
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"} # 
]

# 
gt_nat_rules: [
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, # 
  %{weight: 100, response_gt: "111111"} # 
]
```

GT NAT

NAT'd GT

- SCCP GT
- SRI-for-SM networkNode-Number
- UpdateLocation ISD HLR GT
- UpdateLocation END

- ISD
- MAP BEGIN

GT NAT

GT NAT $O(n)$ n

- 100
-
-

- $10 < 1\mu s$
- $50 < 5\mu s$
- $100 < 10\mu s$

100

- $10 \approx 1 \text{ KB}$
- $100 \approx 10 \text{ KB}$

□□□□

1. □□□□□□□□□□

```
gt_nat_rules: [  
  {%calling_prefix: "8772", weight: 10, response_gt: "111111"},  
  {%calling_prefix: "8773", weight: 10, response_gt: "222222"},  
  {%weight: 100, response_gt: "default_gt"} # □□□□□□□□□□  
]
```

2. □□□□□□□□□□

```
# □□□□□□□□□□□□□□  
{%calling_prefix: "331", weight: 10, response_gt: "..."} # □□  
{%calling_prefix: "44", weight: 10, response_gt: "..."} # □□  
  
# □□□□□□□□□□□□□□  
{%calling_prefix: "3", weight: 5, response_gt: "..."} # □□□□  
{%calling_prefix: "331", weight: 100, response_gt: "..."} # □□□□□  
□□□□□□□□□□
```

3. □□□□□□

```
gt_nat_rules: [  
  # □□□□ XYZ - □□□□□GT □□□4412xxxxxxx□  
  # □□ 10□□□□□□□□□□  
  {%calling_prefix: "4412", weight: 10, response_gt: "5551001"},  
  
  # □□□□ ABC - □□□□□GT □□□33123xxxxxxx□  
  # □□ 10□□□□□□□□□□  
  {%calling_prefix: "33123", weight: 10, response_gt: "5551002"}  
]
```

4. 設定

```
# 設定 iex 環境
iex> GtNat.translate_response_gt("44121234567", nil, "default")
"5551001" # 設定

# 設定 GT 環境
iex> GtNat.translate_response_gt("44121234567", "555123",
"default")
"5551001" # 設定
```

5. 設定

INFO 設定 GT NAT 設定

設定

STP 設定

GT NAT 設定 STP 設定 GT 設定 GT NAT 設定

STP 設定 STP 設定

CAMEL 設定

GT NAT 設定 CAMEL/CAP 設定

SCCP 設定

- CAMEL 設定 GT
- gsmSSF GT 設定

設定

- `camel_gsmscf_gt_address` - gsmSCF GT 設定
- 設定 GT
- GT NAT 設定

###

```
#  gsmSSF 555123456  gsmSCF
#  =55512341112=555123456
# GT NAT  "555" -> response_gt="55512341111"
#  =555123456=55512341111
```

####

GT NAT M3UA

####

GT NAT

1.

```
#  runtime.exs
config :omniss7,
  gt_nat_enabled: false, # 
  gt_nat_rules: [
    # 
    %{calling_prefix: "877", weight: 10, response_gt:
"111111"},
    %{weight: 100, response_gt: "999999"} # 
  ]
```

2.

```
# 
mix compile

#  iex 
iex -S mix
iex> GtNat.validate_config()
```

3. 確認

```
gt_nat_enabled: true # 確認 true
```

4. 確認

```
tail -f log/omniss7.log | grep "GT NAT"
```

5. 確認

- 確認
- 確認 24 時間
- 確認 `gt_nat_enabled: false`

確認

確認

- 確認 "GT NAT" 確認
- 確認 `GtNat.validate_config()` 確認
- 確認
- 確認 OmniSS7 確認

確認

- HLR 確認 - HLR 確認
- SMSC 確認 - SMSc 確認
- STP 確認 - STP 確認
- 確認 - 確認

HLR 簡介

← 目錄

本節介紹 OmniSS7 與 網路層 (HLR/HSS) 的 **OmniHSS** 網路層架構

OmniHSS 簡介

OmniSS7 HLR 透過 **SS7** 網路與 **OmniHSS** 網路層 HSS 網路層

- **OmniSS7 (HLR 端)** 透過 SS7/MAP 網路層 SCCP 網路層
- **OmniHSS (HSS 端)** 透過網路層 HSS 網路層

OmniHSS 簡介

OmniHSS 網路層 HSS 網路層

- **IMSI** 網路層 MSISDN 網路層 IMSI 網路層 eSIM 網路層
- 網路層 Milenage 3G/4G/5G 網路層 COMP128 2G 網路層
- 網路層 CS 網路層 PS 網路層
- 網路層 CAMEL 網路層
- **API** 網路層 RESTful HTTP API 網路層 CRM 網路層
- 網路層

OmniHSS 網路層 OmniSS7 網路層 HTTPS API 網路層 OmniHSS 網路層 MAP 網路層 UpdateLocation 網路層 SendAuthenticationInfo 網路層 SendRoutingInfo 網路層

OmniSS7 HLR 網路層 網路層 OmniHSS 網路層 OmniSS7 網路層 SS7/MAP 網路層 **OmniHSS** 網路層

IMSI 簡介

OmniHSS 網路層 **IMSI** 網路層 MSISDN 網路層 IMSI 網路層

- 電話番号から自動的に IMSI を取得する
- **eSIM** 対応デバイス eSIM をインストールする
- 電話番号から MSISDN を取得する
- **SIM** 対応デバイス SIM を挿入する
- 電話番号から IMSI を取得する

手順

- 1. IMSI を取得する (Ki/OPc/OPd)
- 2. IMSI を取得する (Ki/OPc/OPd)
- 3. 電話番号から IMSI を取得する
- 4. OmniSS7 から IMSI を取得する (OmniHSS/OmniHSS から取得する)
- 5. 電話番号から IMSI を取得する (IMSI から取得する)

1. IMSI を取得

```

1. MSISDN: +1-555-123-4567
   ├── IMSI 1: 310260123456789 (Milenage から取得)
   ├── IMSI 2: 208011234567890 (Milenage から取得)
   └── IMSI 3: 440201234567891 (COMP128 から取得)

```

2. IMSI を取得する (OmniHSS から IMSI を取得する)

□□

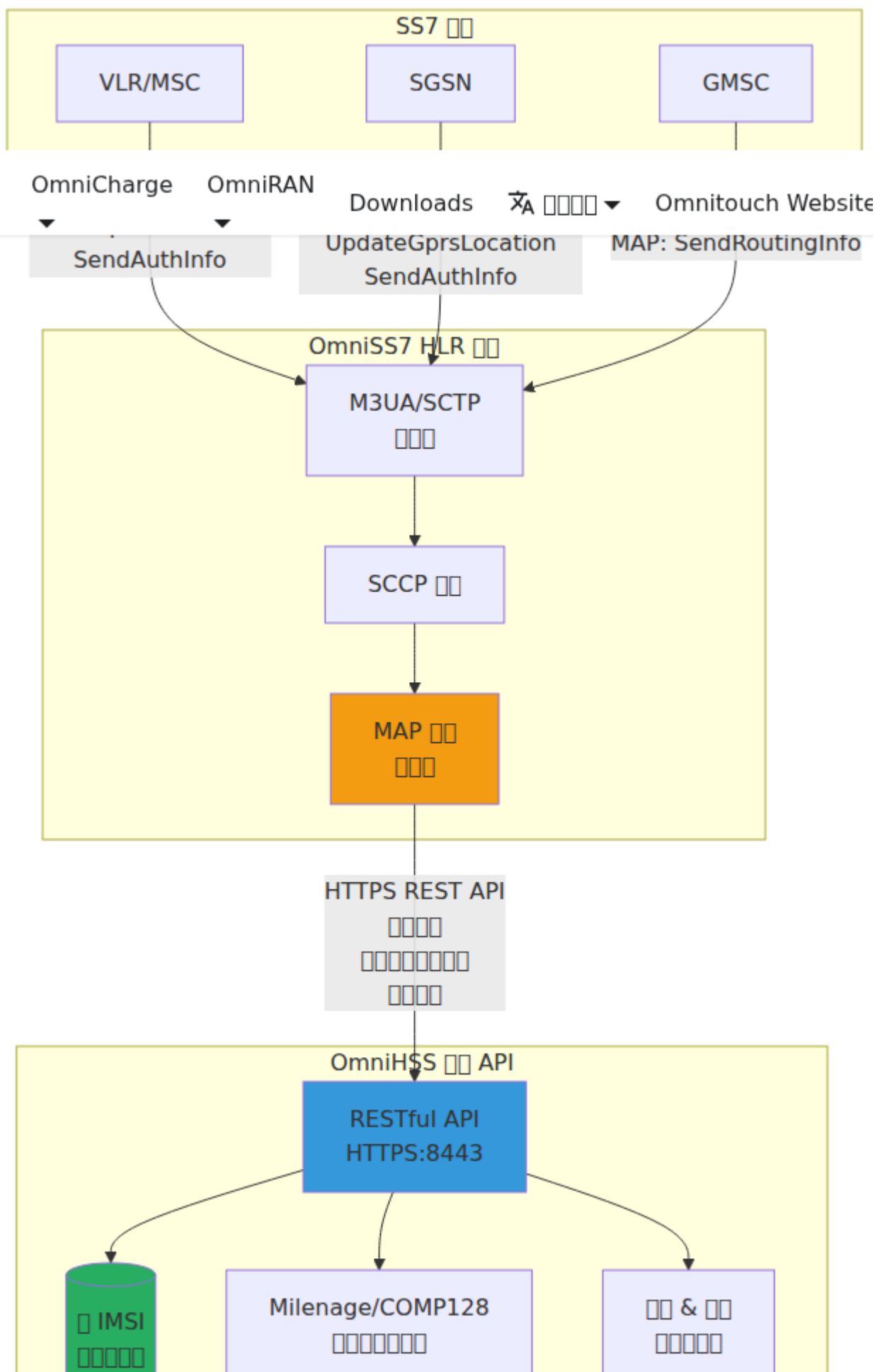
1. OmniHSS □□
2. □ IMSI □□
3. □□□ HLR □□□
4. □□ HLR □□
5. □□□□□
6. □□□□□□
7. □□□□
8. CAMEL □□
9. □□□□□□
10. HLR □□
 - □□□□□□
 - SendRoutingInfo (SRI)
 - UpdateLocation / ISD
 - SendRoutingInfoForSM
 - □□□□□□

HLR

HLR OmniSS7

-
-
- VLR
- SMS

HLR ☐ ☐



HLR

OmniSS7 HLR HLR

HLR

OmniSS7 `config/runtime.exs` HLR

1. `config/runtime.exs`
2. 53-174
 - 1 STP 53-85
 - 2 HLR 87-123
 - 3 SMS Sc 125-174
3. #
4. HLR 87-123 #
- 5.
6. `iex -S mix`

HLR

HLR

```

config :omniss7,
  # 配置 - 配置 HLR 配置
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,

  # OmniHSS 配置 API 配置
  hlr_api_base_url: "https://10.180.2.140:8443",

  # HLR 配置 GT 配置 SMS 配置
  hlr_service_center_gt_address: "1234567890",

  # MSISDN ↔ IMSI 配置
  # 配置MSISDN ↔ IMSI 配置
  hlr_imsi_plmn_prefix: "50557",
  hlr_msisdn_country_code: "61",
  hlr_msisdn_nsn_offset: 0,
  hlr_msisdn_nsn_length: 9,

  # InsertSubscriberData 配置
  # 配置:packetAndCircuit:packetOnly 配置 :circuitOnly
  isd_network_access_mode: :packetAndCircuit,

  # 配置 ISD #2配置
  isd_send_ss_data: true,

  # 配置 ISD #3配置
  isd_send_call_barring: true,

  # CAMEL 配置 SendRoutingInfo 配置
  # CAMEL 配置
  camel_service_key: 11_110,

  # CAMEL 配置
  # 配置:termAttemptAuthorized:tBusy:tNoAnswer:tAnswer
  camel_trigger_detection_point: :termAttemptAuthorized,

  # 配置 VLR 配置
  # 配置“配置”配置 VLR 配置
  # 配置 VLR 配置 SRI 配置
  # 配置 PRN 配置 MSRN
  home_vlr_prefixes: ["123456"],

```

```
# M3UA
# ASP MAP UpdateLocation SendAuthInfo
map_client_m3ua: %{
  mode: "ASP",
  callback: {MapClient, :handle_payload, []},
  process_name: :hlr_client_asp,
  # HLR
  local_ip: {10, 179, 4, 11},
  local_port: 2905,
  # STP
  remote_ip: {10, 179, 4, 10},
  remote_port: 2905,
  routing_context: 1
}
```

項目	型	デフォルト値	注
hlr_api_base_url	String	""	Optional
hlr_service_center_gt_address	String	""	HLR URA HLR Home
smc_service_center_gt_address	String	""	SP SMS
hlr_smc_alert_gts	String	[]	Up ale SM
hlr_alert_location_expiry_seconds	Integer	172800	SM aler seconds
hlr_imsi_plmn_prefix	String	"50557"	MSIS PLM ↔ IM
hlr_msisdn_country_code	String	"61"	MSIS country IMSI
hlr_msisdn_nsn_offset	Integer	0	MSIS offset IMSI
hlr_msisdn_nsn_length	Integer	9	MSIS

属性	数据类型	默认值	说明
isd_network_access_mode	枚举	:packetAndCircuit	接入网络模式
isd_send_ss_data	布尔	true	是否发送SS数据
isd_send_call_barring	布尔	true	是否发送呼叫限制
camel_service_key	枚举	11_110	发送CAMEL服务密钥
camel_trigger_detection_point	枚举	:termAttemptAuthorized	CAMEL触发检测点
home_vlr_prefixes	列表	["5551231"]	归属VLR前缀列表
local_ip	字符串		本地IP地址
local_port	整数	2905	本地端口
remote_ip	字符串		远端IP地址
remote_port	整数	2905	远端端口
routing_context	枚举	1	M3U路由上下文

□□ **HLR** □□□□□□□□

```
h1r_mode_enabled: true Web UI
```

- **SS7** - 3GPP
- **SS7** - MAP
- **M3UA** - 3GPP
- **HLR** - HLR API + *HLR*
- - 3GPP
- - 3GPP

□□□□□□□□ **SMSc** □□ □□□□□□□□

□ □ □ □

- hlr_service_center_gt_address
- OmniHSS** OmniHSS API
- API** OmniHSS API 5
- MAP** MAP SRI UpdateLocation SendAuthInfo 10
- ISD** UpdateLocation InsertSubscriberData (ISD) 10
- STP M3UA MAP
-
- Web UI** Web Web UI
- API** REST API Swagger UI API

--	--	--	--	--

OmniHSS 与 OmniSS7 的 RESTful API 接口

OmniHSS

OmniHSS

- IMSI
- KiOPc Milenage COMP128
- QoS
- VLR/MSC SGSN/GGSN
- CAMEL gsmSCF
- CLIP/CLIR
- /

OmniHSS Milenage COMP128

OmniSS7 sendAuthenticationInfo MAP

1. OmniSS7 MAP IMSI
2. OmniSS7 OmniHSS API
3. OmniHSS Ki OPc
4. OmniHSS RANDXRESCKIKAUTN
5. OmniSS7 MAP VLR/SGSN

OmniHSS API

OmniSS7 HTTPS REST API OmniHSS

```
config :omniss7,  
  hlr_api_base_url: "https://omnihss-server:8443"
```

OmniSS7 SS7 MAP OmniHSS

- IMSI MSISDN
- Ki/OPc
- UpdateLocation
-

updateLocation MAP OmniSS7 OmniHSS VLR

1. UpdateLocation IMSI VLR GT MSC GT
2. OmniHSS
3. OmniHSS VLR/MSC
4. InsertSubscriberData (ISD) VLR
5. UpdateLocation VLR hlr_service_center_gt_address HLR GT
6. alertServiceCenter SSMSc GT hlr_ssmc_alert_gts

hlr_service_center_gt_address UpdateLocation HLR VLR/MSC HLR

UpdateLocation HLR alertServiceCenterMAP 64 SSMSc SSMSc SSMSc

SSMSc

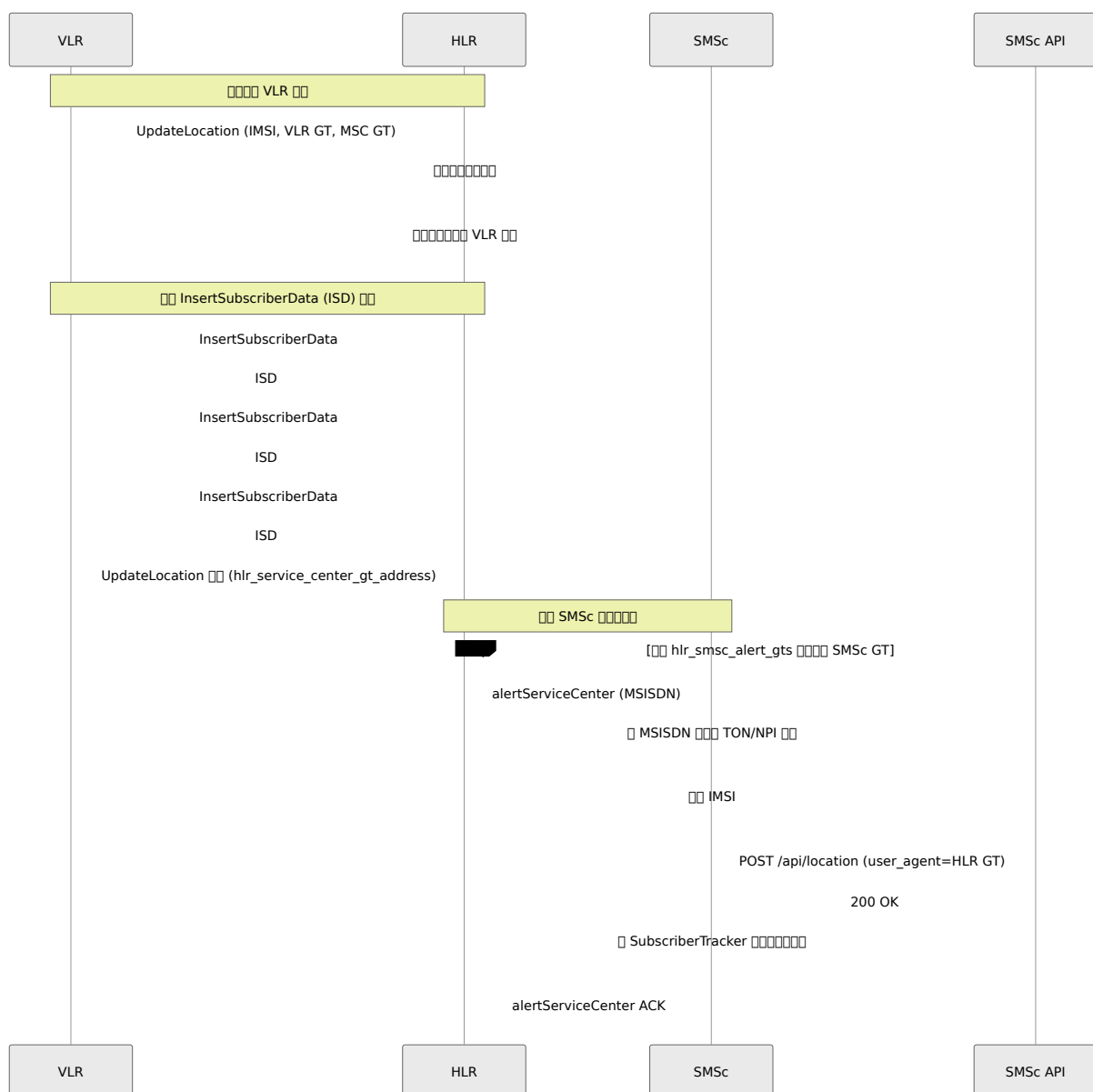
```

config :omniss7,
  # [] UpdateLocation [][] alertServiceCenter [] SMSc GT []
  hlr_smsc_alert_gts: [
    "15559876543",
    "15559876544"
  ],

  # SMSc [][] alertServiceCenter []48 []
  hlr_alert_location_expiry_seconds: 172800

```

[][]



[][]

如何 UpdateLocation

1. HLR 的 `hlr_smsc_alert_gts` 如何 SMSc GT 的 `alertServiceCenter`
2. 如何 MSISDN
3. HLR 的 `hlr_service_center_gt_address` 如何 GT
4. SCCP 如何 SSN=6 的 HLR 如何 SSN=8 的 SMSc

SMSc 如何

- 如何 **MSISDN** 的 **TON/NPI** 如何 “19123123213” → “123123213”
- 如何 POST 的 `/api/location`
- 如何 **API** 的 `user_agent` 如何 **HLR GT** 如何 HLR 如何
- 如何 `hlr_alert_location_expiry_seconds` 如何
- 如何 SMSc 如何

如何

如何 Web UI 如何 如何 `alertServiceCenter` 如何

1. 如何 “如何” 如何
2. 如何 “如何” 如何
3. 如何 MSISDN 的 SMSc GT 的 HLR GT 如何
 - SMSc GT 如何 `hlr_smsc_alert_gts` 如何
 - HLR GT 如何 `hlr_service_center_gt_address`
4. 如何 “如何 `alertServiceCenter`”

如何 SMSc 如何 UpdateLocation 如何 `phx-blur` 如何

InsertSubscriberData (ISD) 如何

如何 UpdateLocation 如何 HLR 如何 **InsertSubscriberData (ISD)** 如何
VLR 的 ISD 如何

如何 如何 如何 如何 ISD 如何

ISD 如何

HLR 000000 3 0000 ISD 0000

1. **ISD #1**000000 - 00000000

- IMSI
- MSISDN
- 0000
- 000000serviceGranted
- 000000
- 000000
- 000000

2. **ISD #2**0000 - 000000SS0000

- 000000000000000000000000
- 0000
- 0000
- 0000
- 0000000000

3. **ISD #3**0000 - 00000000

- 00000000BAOC
- 0000000000BOIC
- 000000

0000

```
# InsertSubscriberData 00
# 00000000:packetAndCircuit0:packetOnly 0 :circuitOnly
isd_network_access_mode: :packetAndCircuit,

# 00 ISD #2000000000
isd_send_ss_data: true,

# 00 ISD #3000000000
isd_send_call_barring: true,
```

ISD

isd_network_access_mode ISD

ISD	ISD	ISD
:packetAndCircuit	ISD GPRS/LTE	ISD - ISD
:packetOnly	ISD/LTE	ISD
:circuitOnly	ISD/SMS	ISD

ISD

ISD

ISD - ISD

```
isd_send_ss_data: true,  
isd_send_call_barring: true,
```

ISD

```
isd_send_ss_data: false,  
isd_send_call_barring: false,
```

ISD + ISD

```
isd_send_ss_data: true,  
isd_send_call_barring: false,
```

ISD

ISD UpdateLocation

```

VLR → HLR: UpdateLocation ( )
HLR → VLR: InsertSubscriberData #1 ( ) - 
VLR → HLR: ISD #1 ACK ( )
HLR → VLR: InsertSubscriberData #2 ( ) - SS [ ]
VLR → HLR: ISD #2 ACK ( )
HLR → VLR: InsertSubscriberData #3 ( ) - [ ]
VLR → HLR: ISD #3 ACK ( )
HLR → VLR: UpdateLocation ( )

```

`isd_send_ss_data` `isd_send_call_barring` `false` ISD `UpdateLocation`

- `:packetAndCircuit` ISD
- `/M2M` `:packetOnly` SS /
- VLR - `isd_send_ss_data`
- ISD

CAMEL

CAMEL `SendRoutingInfo`

GMSC MSC `SendRoutingInfo` SRI HLR GMSC CAMEL

CAMEL

CAMEL

CAMEL GSM/UMTS

-
-
- VPN

- 0000
- 000000000000
- 0000000

0000

```
# CAMEL 0000 SendRoutingInfo 000
# CAMEL 00000000
camel_service_key: 11_110,

# CAMEL 0000
# 000:termAttemptAuthorized:tBusy:tNoAnswer:tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,
```

0000

camel_service_key 000 gsmSCF000000000000 CAMEL 0000000000000000
0000

0000	0000
11_110	000000000000
100	0000000
200	??0000000000
300	0000000VPN
000	0000000

00000

```
# 000000000000
camel_service_key: 11_110,

# 00 VPN 00
camel_service_key: 300,
```

00000

camel_trigger_detection_point 000000000000 CAMEL 000

000	00	0000
:termAttemptAuthorized	000000000000	000000000000
:tBusy	0000	000000
:tNoAnswer	00000	00000000
:tAnswer	0000	000000000

00000

00000000000000000000

```
camel_trigger_detection_point: :termAttemptAuthorized,
```

00000000000000000000

```
camel_trigger_detection_point: :tBusy,
```

00000000000000000000

```
camel_trigger_detection_point: :tAnswer,
```

0 CAMEL 0 SRI 00

SendRoutingInfo CAMEL

```
GMSC → HLR: SendRoutingInfo ( )
HLR → GMSC: SRI ( )
- IMSI
- VLR
-
- CAMEL
  * 11_110
  * gsmSCF < >
  * termAttemptAuthorized
  * continueCall
```

GMSC gsmSCF CAMEL

- gsmSCF
- :termAttemptAuthorized
- 11_110
- defaultCallHandling: :continueCall gsmSCF

VLR VLR

HLR **SendRoutingInfo** SRI " " VLR VLR

VLR

- VLR CAMEL SRI
- VLR PRN MSRN SRI

□□

```
# □□ VLR □□
# □□□"□□"□□□ VLR □□□□□□
# □□□□□ VLR □□□□□□□□□□□□□□□□□ SRI □□
# □□□□□□□□□□□□□□□ PRN □□□ MSRN
home_vlr_prefixes: ["555123"],
```

□□□□□

```
# □□□□□□□
home_vlr_prefixes: ["555123"],

# □□□□□□□□□□□□□□□□□□□□□□
home_vlr_prefixes: ["555123", "555124", "555125"],
```

□□□□□

1. □□□□□□□□□□

□□□□ VLR □□□□□□□□□□□□□□□□

```
GMSC → HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR □□□□□ API □□□□□□□□
HLR □□ VLR □□□"5551234567"
HLR □□□VLR □ "555123" □□ → □□□□
HLR → GMSC: SRI □□□□ CAMEL □□□□□□
- IMSI
- VLR □□□"5551234567"
- gsmSCF □□□MSC□□"5551234501"
- CAMEL □□□□□11_110
- □□□□□□termAttemptAuthorized
```

2. □□□□□□□□ **PRN**□

□□□□ VLR □□□□□□□□□□□□□□□□

GMSC → HLR: SendRoutingInfo (MSISDN: "1234567890")

HLR 0000 API 000000

HLR 00 VLR 000 "49170123456"

HLR 000VLR 00 "555123" 00 → 00

HLR → MSC: ProvideRoamingNumber (PRN):

- MSISDN: "1234567890"
- IMSI: "999999876543210"
- MSC 00: "49170123456"
- GMSC 00: "5551234501"

MSC → HLR: PRN 0000 MSRN: "49170999888777"

HLR → GMSC: SRI 0000000000

- IMSI
- VLR 00: "49170123456"
- 0000 (MSRN): "49170999888777"

00000000

0000 **SRI** 00

```
%{
  imsi: "999999876543210",
  extendedRoutingInfo: {
    :camelRoutingInfo, %{
      gmscCamelSubscriptionInfo: %{
        "t-CSI": %{
          serviceKey: 11_110,
          "gsmSCF-Address": "5551234501",
          defaultCallHandling: :continueCall,
          "t-BcsmTriggerDetectionPoint": :termAttemptAuthorized
        }
      }
    }
  },
  subscriberInfo: %{
    locationInformation: %{"vlr-number": "5551234567"},
    subscriberState: {:notProvidedFromVLR, :NULL}
  }
}
```

0000 **SRI** 00

```
%{
  imsi: "999999876543210",
  extendedRoutingInfo: {
    :routingInfo, %{
      roamingNumber: "49170999888777" # PRN MSRN
    }
  },
  subscriberInfo: %{
    locationInformation: %{"vlr-number": "49170123456"},
    subscriberState: {:notProvidedFromVLR, :NULL}
  }
}
```

PRN

PRN

MSC/VLR PRN

MSISDN	SRI	
IMSI	HLR API	IMSI
MSC	HLR API	MSC serving_msc
GMSC	SRI	SRI GMSC
CAMEL		GMSC CAMEL

PRN

HLR PRN

- MSRN

□□□□□

- □□ PRN □□ → □ SRI □□□□□□ 27□□□□□□
- □□ PRN □□ → □ SRI □□□□□□ 27□□□□□□
- □□□□□□ MSRN → □ SRI □□□□□□ 27□□□□□□

□□□□□

□□□□□□□□□□

```
# □□ VLR □□□ "555123" □□□□□□□□  
home_vlr_prefixes: ["555123"],
```

- VLR 5551234567 → □□□CAMEL □□□
- VLR 5551235001 → □□□CAMEL □□□
- VLR 49170123456 → □□□PRN + MSRN □□□

□□□□□□□

```
# □□□□□□□□□□□□□□  
home_vlr_prefixes: ["555123", "555124", "555125"],
```

- VLR 5551234567 → □□□□□ 1□
- VLR 5552341234 → □□□□□ 2□
- VLR 5553411111 → □□□□□ 3□
- VLR 44201234567 → □□□□□□

□□□□□

□□□□ PRN □□□□□□□□□□□□□□ VLR □□□□□□

```
# □□ VLR □□□□□□□□□□ PRN □□□□  
home_vlr_prefixes: [],
```

PRN

- PRN VLR PRN + PRN
- PRN VLR PRN
- PRN PRN
- PRN
- PRN PRN PRN

PRN

PRN

- PRN `home_vlr_prefixes` PRN VLR PRN
- PRN VLR PRN

PRN PRN

- PRN MSC/VLR PRN
- PRN MSC PRN M3UA/SCCP PRN

PRN SRI PRN MSRN PRN

- PRN PRN PRN
- PRN PRN PRN `extract_msrn_from_prn/1`

HLR PRN

PRN MAP PRN

- `updateLocation` PRN 2 - PRN VLR PRN
- `sendAuthenticationInfo` PRN 56 - PRN
- `sendRoutingInfo` PRN 22 - PRN CAMEL PRN MSRN
- `sendRoutingInfoForSM` PRN 45 - PRN SMS PRN MSC GT
- `cancelLocation` PRN 3 - PRN VLR PRN

- insertSubscriberData 0000 7 - 000000

000000

000000 HLR 000000000000

SendRoutingInfo (SRI) 00

0000000000000000

HLR 00000000 VLR 0000000000000000

0000000**CAMEL** 000

00000 VLR 000000 home_vlr_prefixes 00000

00000

項目	API	説明
IMSI	OmniHSS API	OmniHSS から取得した IMSI
VLR 番号	OmniHSS API	現在接続している VLR [circuit_session.assigned_vlr]
提供状況	あり	なし notProvidedFromVLR
extendedRoutingInfo	-	[camelRoutingInfo]
gsmSCF 番号	OmniHSS API	現在接続している MSC [circuit_session.assigned_msc]
提供状況	runtime.exs	CAMEL サービスキー [camel_service_key]
検出ポイント	runtime.exs	CAMEL 検出ポイント [camel_trigger_detection_point]
CAMEL サービス	あり	CAMEL サービス
接続先	あり	gsmSCF 番号

MSRN 番号

VLR 番号 home_vlr_prefixes

番号

項目	API	説明	注
IMSI	OmniHSS API	OmniHSS から取得した IMSI	"
VLR 名	OmniHSS API	VLR 名 [circuit_session.assigned_vlr]	"
住所	住所	住所 [notProvidedFromVLR]	:
extendedRoutingInfo	-	[routingInfo]	-
住所MSRN	PRN 住所	住所MSRN	"

住所

```
1. OmniSS7 住所 SendRoutingInfo 住所
2. OmniSS7 住所 OmniHSS API 住所
3. OmniSS7 住所 VLR 住所 home_vlr_prefixes

住所 VLR 住所
→ 住所 CAMEL 住所

住所 VLR 住所
→ 住所 MSC 住所PRN
→ 住所 PRN 住所 MSRN
→ 住所 MSRN 住所
```

住所

- OmniSS7 住所 OmniHSS 住所
- OmniHSS 住所 IMSI住所 VLR/MSC 住所
- OmniSS7 住所 MAP 住所

住所

```
# runtime.exs
home_vlr_prefixes: ["555123"], # VLR
```

- serving_vlr serving_msc null 27
- 1
- PRN 27
- PRN 27

UpdateLocation InsertSubscriberData

VLR

UpdateLocation

HLR	runtime.exs	HLR hlr_service_center_gt_address	"5551234568"
TCAP		ISD	END

InsertSubscriberData #1

項目	項目	項目	項目
IMSI	項目	項目 UpdateLocation 項目	"999999876543210"
MSISDN	OmniHSS API	項目 OmniHSS 項目項目項目	"555123456"
項目	項目	項目項目	"\n" (0x0A)
項目項目	項目	項目項目	:serviceGranted
項目項目項目	項目	項目項目項目	[<<31>>]
項目項目項目	項目	項目項目項目	[<<17>>, "!", "\"]
項目項目項目	runtime.exs	項目/項目項目 項目 isd_network_access_mode 項目	:packetAndCircuit

InsertSubscriberData #2項目項目項目 - 項目

項目	項目	項目	項目項目
項目 SS	項目	項目項目項目	isd_send_ss_data: true
項目項目	項目	項目項目項目項目項目項目項目項目項目	項目項目
項目項目	項目	項目項目項目項目	項目項目
項目項目	項目	項目項目項目	項目項目

InsertSubscriberData #3項目項目項目 - 項目

項目	項目	項目	項目
項目	項目	項目	isd_send_call_barring: true
BAOC	項目	項目項目SS 項目 146	項目
BOIC	項目	項目項目項目SS 項目 147	項目
項目	項目	項目	項目

ISD 項目

- ISD #1項目 - 項目項目
- ISD #2項目 isd_send_ss_data: true 項目
- ISD #3項目 isd_send_call_barring: true 項目

SendRoutingInfoForSM (SRI-for-SM) 項目

項目 SMS 項目 MSC/SMSC 項目項目 SMSc 項目 SMS 項目項目項目 HLR 項目 SRI-for-SM 項目項目項目項目

項目項目

項目	項目	項目	項目	
IMSI	項目	項目 MSISDN 項目項目 IMSI	PLMN_PREFIX + zero_padded_MSISDN	"00100
項目項目 項目	runtime.exs	項目 SMS 項目 SMSC GT 項目	smsc_service_center_gt_address	"55512

項目項目 runtime.exs 項目

```
# 0000 GT 0000 SRI-for-SM 000000
# 000000 SMS Sc 000 MT-ForwardSM 00000000
smsc_service_center_gt_address: "5551234567", # 00

# MSISDN ↔ IMSI 0000
# PLMN 000MCC001 = 000000+ MNC001 = 000000
hlr_imsi_plmn_prefix: "001001", # 000000000000
```

MSISDN ↔ IMSI 00

000000

0000000 OmniSS7 000 SRI-for-SM 0000000 IMSI

- **hlr_imsi_plmn_prefix** 00000000 IMSI 0 MCC+MNC 000000“50557”00
MCC=5050MNC=570
- **hlr_msisdn_country_code** 000000 IMSI→MSISDN 0000000000000000“61”000
0000“1”0000/0000
- **hlr_msisdn_nsn_offset** 0MSISDN 00000000NSN00000000000000 0000
MSISDN 0000000000000000000000
- **hlr_msisdn_nsn_length** 0000 MSISDN 0000 NSN 000

0000000000000000 0000000 MSISDN ↔ IMSI 000

000000 **MSISDN** 0 **IMSI** 0000

SendRoutingInfoForSM 0SRI-for-SM000000 HLR 000000 **IMSI**0000000000000000
00 SMS Sc 0000000 **MSISDN**00000000

00000000

- SMS Sc 0000000 MSISDN 0 SRI-for-SM0000“5551234567”0
- HLR 0000000000000000000000 IMSI
- HLR 0 SRI-for-SM 000000 IMSI
- SMS Sc 0000 MT-ForwardSM 000000 MSC/VLR 0000 IMSI

OmniSS7 0000 - 00 **IMSI**0

OmniSS7 如何透過 SS7 訊號取得 MSISDN 與 IMSI 號碼？如何透過 IMSI 號碼取得 MSISDN 號碼？

1. 如何透過 HLR 查詢到 IMSI 號碼？SS7 訊號 SRI-for-SM 如何？
2. 如何透過 SRI-for-SM 查詢到 HLR 號碼？IMSI 號碼 - IMSI 號碼 MSISDN 如何？

如何？

MSISDN 如何？IMSI 如何？MCC+MNC 如何？

```
IMSI = PLMN_PREFIX + zero_padded_MSISDN
```

如何？

- **PLMN_PREFIX** = MCC + MNC 如何？"001001" 如何？
- **MSISDN** 如何？
- 如何？IMSI 號碼 15 位

如何？

```
# 如何？
plmn_prefix = "001001" # MCC 001 + MNC 01

# 如何？SRI-for-SM 如何？MSISDN TBCD 如何？
msisdn = "555123456" # 9 位

# 如何？1 如何？
subscriber_digits = 15 - String.length("001001") # = 9 位

# 如何？2 如何？MSISDN 如何？
padded_msisdn = String.pad_left("555123456", 9, "0") # =
"555123456" 如何？

# 如何？3 如何？PLMN 如何？+ 如何？MSISDN
imsi = "001001" <> "555123456" # = "001001555123456" 如何？15 位
```

如何？

MSISDN	PLMN	MSISDN	IMSI	
"555123456"	"001001" (6)	9	"555123456"	"001001555123456"
"99"	"001001" (6)	9	"000000099"	"001001000000099"
"999999999"	"001001" (6)	9	"999999999"	"001001999999999"
"91123456789"	"001001" (6)	9	"555123456"	"001001555123456"

- MSISDN"99" → "000000099"
- MSISDN"91123456789" → "555123456"
- IMSI 15

IMSI → MSISDN

SMSc IMSI MSISDN

```
# 提取 SRI-for-SM 中的 IMSI
imsi = "001001555123456"

# 提取 15 位 PLMN 前缀
plmn_prefix = "001001"
subscriber_portion = String.slice(imsi, 6, 9) # = "555123456"

# 提取 20 位 MSISDN 号码
msisdn = String.replace_leading(subscriber_portion, "0", "") # =
"555123456"
```

数据表 1

完整 IMSI	PLMN 前缀	中间部分	后三位	完整 MSISDN
"001001555123456"	"001001"	"555123456"	"555123456"	"555123456"
"0010010000000099"	"001001"	"0000000099"	"99"	"99"
"0010019999999999"	"001001"	"9999999999"	"9999999999"	"9999999999"

数据表 2

- 提取 MSISDN 中的 IMSI
- 提取 IMSI 中的 MSISDN
- 提取 `hlr_imsi_plmn_prefix`
- 提取 IMSI 前缀
- 提取 HLR 前缀
- 每 15 个 IMSI 提取 15 个

MSISDN 提取

HLR 中的 SRI-for-SM 中的 MSISDN 中的 TBCD 提取

1. **TBCD** 提取 TBCD 中的 TON/NPI 中的“91”
2. 提取 TBCD 中的 TON/NPI
3. 提取 TBCD 中的 TON/NPI

4. 手机号码 PLMN 码 + 号码MSISDN

流程图

1. SRI-for-SM 消息携带 IMSI 号码，HLR 返回 IMSI 号码，
2. UpdateLocation 消息携带 IMSI 号码，SendAuthenticationInfo 消息
携带 IMSI 号码

消息

1. SMSc → HLR: SRI-for-SM 消息
- MSISDN 号码 "91123456789" 号码 TON/NPI
2. HLR 返回
- TBCD 号码 "91123456789"
- 号码 "91123456789" 11 号码
- 号码 9 号码 "555123456" 号码 9 号码
- 号码 PLMN "001001" + "555123456" = "001001555123456"
- 号码 SMSC GT "5551234567"
3. HLR → SMSc: SRI-for-SM 消息
- IMSI "001001555123456" 号码 15 号码
- 号码 "5551234567" 号码 MT-ForwardSM 号码
4. SMSc 消息 MT-ForwardSM 号码 "5551234567" IMSIN 号码 "001001555123456"

消息

消息 runtime.exs 消息

```
# PLMN 号码 MCC 001 = 号码 + MNC 01 = 号码  
hlr_imsi_plmn_prefix: "001001",  
  
# NSN 号码 MSISDN 号码  
hlr_msisdn_country_code: "1",      # 号码 IMSI→MSISDN  
hlr_msisdn_nsn_offset: 1,          # 号码 1 号码  
hlr_msisdn_nsn_length: 10          # 号码 10 号码 NSN
```

NSN 号码

MSISDN "68988000088" NSN

- hlr_msisdn_nsn_offset NSN
- hlr_msisdn_nsn_length MSISDN NSN

		MSISDN	nsn_offset	nsn_length	NSN
1 CC	"9"	"95551234567"	1	10	"5551234567"
2 CC	"99"	"99412345678"	2	9	"412345678"
3 CC	"999"	"99988000088"	3	8	"88000088"

1. MSISDN → IMSI MSISDN NSN PLMN

```
MSISDN: "99988000088"  
NSN: String.slice("99988000088", 3, 8) = "88000088"  
NSN: "088000088"9  
IMSI: "547050" + "088000088" = "547050088000088"
```

2. IMSI → MSISDN PLMN

```
IMSI: "547050088000088"  
PLMN: "088000088"  
NSN: "88000088"  
MSISDN: "+999" + "88000088" = "+99988000088"
```

API 목록 - SRI-for-SM 목록

API 목록

목록

이름	종류	설명
OmniHSS API	OmniHSS API 목록	IMSI, MSISDN, circuit_session, VLR/MSD
runtime.exs	OmniSS7 API	smc_service_center_gt_address, camel_service_key, isd_network_access_mode
API	목록	SS 목록
API	MAP API	UpdateLocation, IMSI, SRI, MSISDN
API	목록	SRI-for-SM, IMSI, hlr_imsi_prefix + NSN

목록

runtime.exs 목록

- hlr_service_center_gt_address - UpdateLocation
- smc_service_center_gt_address - SRI-for-SM, MT-ForwardSM

runtime.exs 목록

- camel_service_key - 11_110
- camel_trigger_detection_point - :termAttemptAuthorized

- `isd_network_access_mode` - `type: packetAndCircuit`
- `isd_send_ss_data` - `type: true`
- `isd_send_call_barring` - `type: true`
- `hlr_imsi_plmn_prefix` - `type: "001001"` `type: MSISDN↔IMSI` `type: PLMN` `type:`

OmniHSS

OmniHSS REST API

- IMSI MSISDN
- VLR/MSC
-
-

OmniSS7

- ←
-
- MAP
-
-

OmniHSS OmniHSS OmniHSS IMSI

OmniSS7 Omnitouch

MAP

←

OmniSS7

MAP

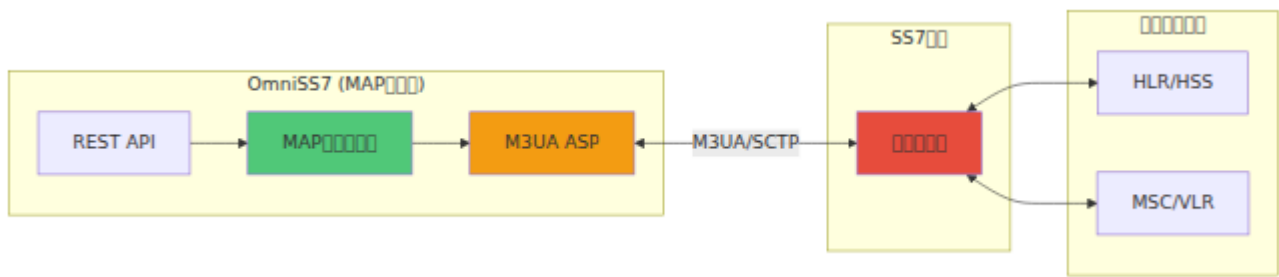
MAP

1. MAP
2. MAP
3. MAP
4. API
- 5.
- 6.

MAP

MAP OmniSS7** ASP M3UA STP SGP / MAP
**

- HLR SRI SRI-for-SM
-
- PRN



MAP

config/runtime.exs MAP M3UA

配置

```
config :omniss7,  
  # 配置MAP客户端  
  map_client_enabled: true,  
  
  # MAP客户端M3UA客户端ASP客户端STP/SGP  
  map_client_m3ua: %  
    mode: "ASP", # M3UA客户端"ASP"客户端"SGP"客户端  
  }  
  callback: {MapClient, :handle_payload, []}, # 回调函数  
  process_name: :map_client_asp, # 进程名称  
  local_ip: {10, 0, 0, 100}, # 本地IP  
  local_port: 2905, # 本地SCTP端口  
  remote_ip: {10, 0, 0, 1}, # 远程STP/SGP IP  
  remote_port: 2905, # 远程STP/SGP端口  
  routing_context: 1 # M3UA客户端  
}
```

配置

```
config :omniss7,  
  # 配置MAP  
  map_client_enabled: true,  
  
  # M3UA  
  map_client_m3ua: %{  
    mode: "ASP",  
    callback: {MapClient, :handle_payload, []},  
    process_name: :map_client_asp,  
    local_ip: {10, 0, 0, 100},  
    local_port: 2905,  
    remote_ip: {10, 0, 0, 1},      # STP IP  
    remote_port: 2905,  
    routing_context: 1  
  }  
  
config :control_panel,  
  web: %{  
    listen_ip: "0.0.0.0",  
    port: 443,  
    hostname: "ss7-gateway.example.com",  
    enable_tls: true,  
    tls_cert: "/etc/ssl/certs/gateway.crt",  
    tls_key: "/etc/ssl/private/gateway.key"  
  }
```

MAP

1. SM SRI-for-SM

HLR SMS MSC HLR SRI-for-SM HLR SRI-for-SM

API `POST /api/sri-for-sm`

Request

```
{
  "msisdn": "447712345678",
  "serviceCenter": "447999123456"
}
```

Response

```
{
  "result": {
    "imsi": "234509876543210",
    "locationInfoWithLMSI": {
      "networkNode-Number": "447999555111"
    }
  }
}
```

cURL

```
curl -X POST http://localhost/api/sri-for-sm \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "447712345678",
  "serviceCenter": "447999123456"
}'
```

2. SRI

HLR

API `POST /api/sri`

Request

```
{
  "msisdn": "447712345678",
  "gmsc": "447999123456"
}
```

□□□

```
{
  "result": {
    "imsi": "234509876543210",
    "extendedRoutingInfo": {
      "routingInfo": {
        "roamingNumber": "447999555222"
      }
    }
  }
}
```

3. □□□□□□PRN□

□□□□MSC□□□□□□□□MSRN□□

API□□□ POST /api/prn

□□□

```
{
  "msisdn": "447712345678",
  "gmsc": "447999123456",
  "msc_number": "447999555111",
  "imsi": "234509876543210"
}
```

4. 认证信息

HLR 认证信息

API POST /api/send-auth-info

请求

```
{
  "imsi": "234509876543210",
  "vectors": 5
}
```

响应

```
{
  "result": {
    "authenticationSetList": [
      {
        "rand": "0123456789ABCDEF0123456789ABCDEF",
        "xres": "ABCDEF0123456789",
        "ck": "0123456789ABCDEF0123456789ABCDEF",
        "ik": "FEDCBA9876543210FEDCBA9876543210",
        "autn": "0123456789ABCDEF0123456789ABCDEF"
      }
    ]
  }
}
```

5. 位置更新

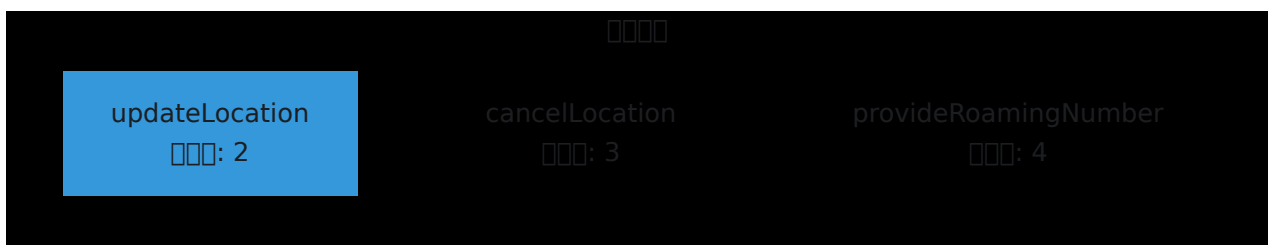
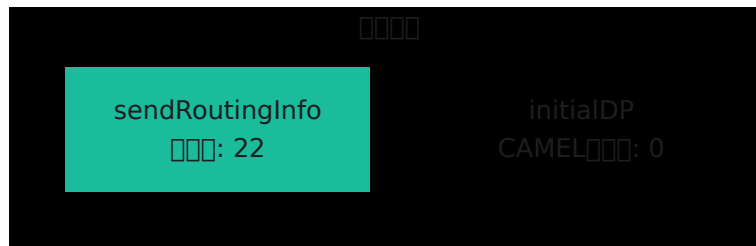
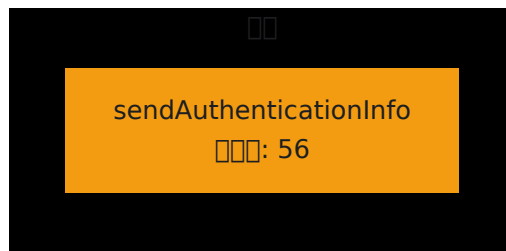
HLR 位置更新信息 HLR 位置更新信息

API POST /api/updateLocation

请求

```
{
  "imsi": "234509876543210",
  "vlr": "447999555111"
}
```

MAP



API

Swagger UI

Swagger UI는 Swagger 2.0 사양을 사용하여 API를 설명하는 OpenAPI 사양을 기반으로 생성된 웹 인터페이스입니다.

Swagger UI

1. `http://your-server/swagger`
2. Swagger API
3. Swagger UI

Swagger UI

1. Swagger UI `/api/sri-for-sm`
2. Swagger UI
3. Swagger UI
4. Swagger UI
5. Swagger UI

API

- **200** - 성공적으로 반환
- **400** - 잘못된 요청
- **504** - Swagger UI 10초 이상 응답 없음

MAP

MAP

MAP

- `map_requests_total` - MAP 요청 수
 - `operation` `sri` `sri_for_sm` `prn` `authentication_info`

- `map_request_errors_total` - MAP
 - `operation`
- `map_request_duration_milliseconds` - MAP
 - `operation`
- `map_pending_requests` - MAP

Prometheus

```
#             SRI-for-SM  
increase(map_requests_total{operation="sri_for_sm"}[1h])

# SRI                  
rate(map_request_duration_milliseconds_sum{operation="sri"}[5m]) /
rate(map_request_duration_milliseconds_count{operation="sri"}[5m])

#   MAP          
sum(rate(map_request_errors_total[5m])) by (operation)

#             
map_pending_requests
```

MAP

- API 504
- HLR/MS

1. M3UA

```
# IEx
:sys.get_state(:map_client_asp)
```

2. STP

3. SCCP

4. SCCP

SCCP

- API SCCP
- “SCCP unitdata service”

SCCP

- STP
- HLR SSN 6
-

- STP
-
-

- ←
- - Web UI API
- STP -
- -

- 0000 - 0000

OmniSS7 Omnitouch 000000

OmniSS7 (SMSc) OmniMessage

← Overview

OmniSS7 SMSc SS7 OmniMessage SMS

OmniMessage

OmniSS7 SMSc SS7 OmniMessage SMS

- OmniSS7 (SMSc SS7) SS7/MAP SCCP
- OmniMessage (SMS)

OmniMessage

OmniMessage SMS

-
- (DLR)
- SMSc OmniMessage
-
- TPS
- API RESTful HTTP API
-

OmniMessage OmniSS7 HTTPS API

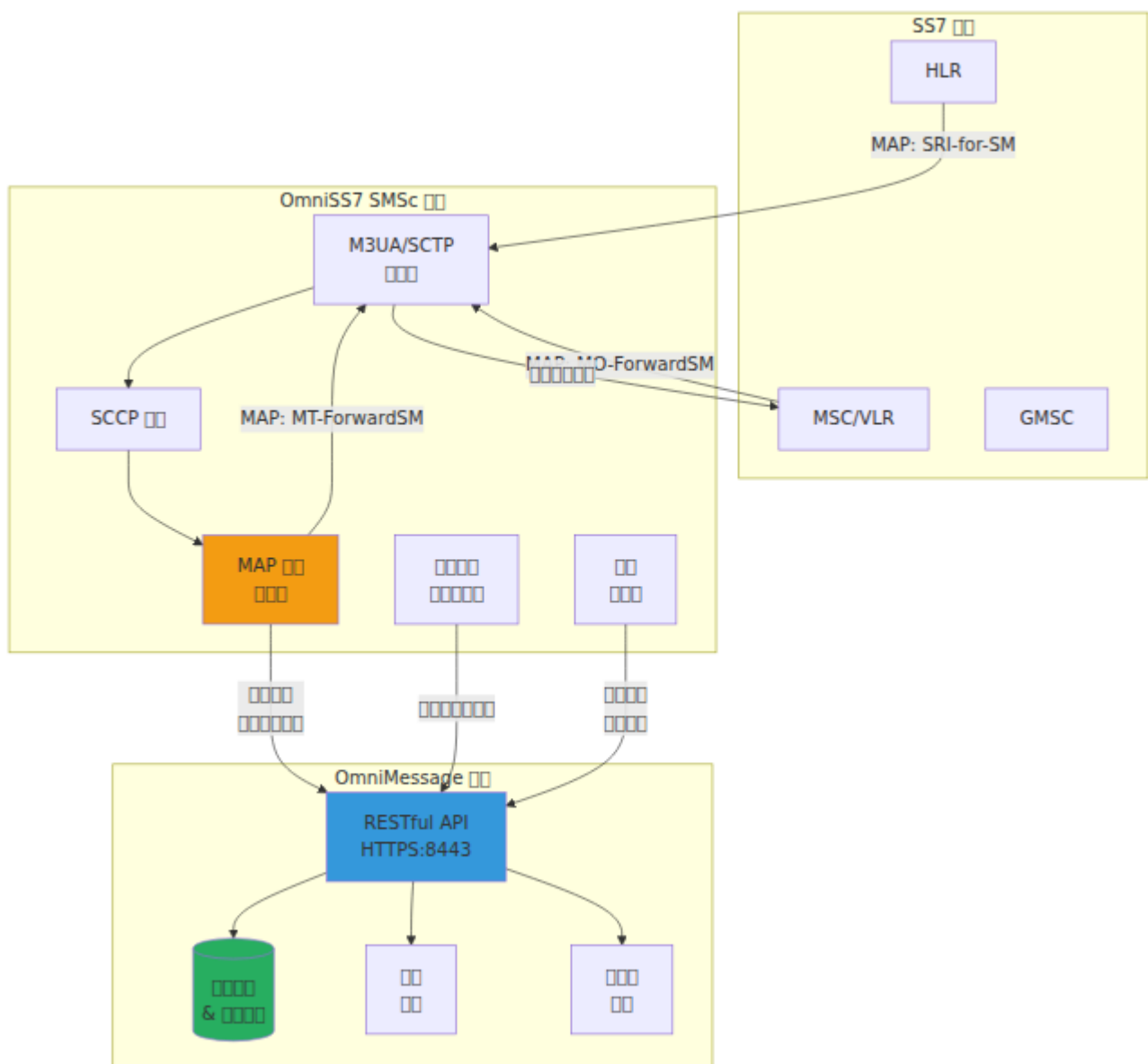
OmniSS7 SMSc OmniMessage OmniSS7 SS7/MAP OmniMessage

1. OmniMessage 消息
2. 消息的接收者
3. 消息的发送者
4. HTTP API 消息
5. 消息的接收者
6. 消息的发送者
7. SMSG 消息
8. 消息的接收者
9. 消息的发送者
10. 消息的接收者

OmniSS7 SS7
OmniMessage

- **MT-SMS** □□□□□□□□□□□□□□□□
- **MO-SMS** □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□□



□□ SMSG □□

OmniSS7 □□□□□□□□□□□□□□ SMSG□□□□□□□□□□ SMSG □□□

□□□ SMSG □□

OmniSS7 □ `config/runtime.exs` □□□□□□□□□□□□□□ SMSG □□□

1. □□ `config/runtime.exs`

2. 00 00000000 53-204 000
 - 00 10STP 0000 53-95 00
 - 00 20HLR 0000 97-142 00
 - 00 30SMSc 0000 144-204 00
3. 000 0000000000000000 #0
4. 0000 SMSc 00000 144-204 000 #0
5. 00000000 0000
6. 00 00000 iex -S mix

SMSc 0000

000 SMSc 00000000

```

config :omniss7,
  # 配置 - 配置 STP + SMS 配置
  # 配置 map_client_enabled 为 true 配置 SMS 配置
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,

  # OmniMessage 配置 API 配置
  smsc_api_base_url: "https://10.179.3.219:8443",
  # SMS 配置
  smsc_name: "ipsmgw",
  # 配置 GT 配置
  smsc_service_center_gt_address: "5551234567",

  # 配置 SMS 配置
  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: "ipsmgw",
  auto_flush_tps: 10,

  # M3UA 配置
  # 配置 ASP 配置/配置 MAP SMS 配置
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :stp_client_asp,
    # 配置 SMS 配置
    local_ip: {10, 179, 4, 12},
    local_port: 2905,
    # 配置 STP 配置
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }

config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "SS7 配置"},
    {SS7.Web.TestClientLive, "/client", "SS7 配置"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {SS7.Web.RoutingLive, "/routing", "配置"},
    {SS7.Web.RoutingTestLive, "/routing_test", "配置"},
    {SS7.Web.SmscLinksLive, "/smsc_links", "SMS 配置"}
  ]

```

```
],
  page_order: ["/events", "/client", "/m3ua", "/routing",
"/routing_test", "/smsc_links", "/application", "/configuration"]
```

□□□□□□□□

□□□□□□□□□□□□□□□□□□ □□□□□

参数	数据类型	默认值	说明
smsc_api_base_url	字符串	空	OmniMessage 的 API 地址
smsc_name	字符串	"{hostname}_SMSc"	SMSc 名称
smsc_service_center_gt_address	字符串	空	SMSc 的 GT 地址
auto_flush_enabled	布尔	true	是否启用自动刷新
auto_flush_interval	整数	10_000	自动刷新的间隔
auto_flush_dest_smsc	字符串	空	SMSc 的目的地
auto_flush_tps	整数	10	自动刷新的 TPS
local_ip	字符串	空	SMSc 的本地 IP
local_port	整数	2905	SCTP 端口
remote_ip	字符串	空	SS7 的 STP IP

項目	項目名	値	説明
remote_port	リモートポート	2905	SCTPポート
routing_context	ルーティングコンテキスト	1	M3UA ルーティング ID

SMS Sc 機能

smc_mode_enabled: true map_client_enabled: true Web UI 設定

- SS7 - 接続
- SS7 - MAP 接続
- M3UA - 接続
- STP - 接続
- STP - 接続
- SMSc API + SMS 接続 ← SMSc 接続
- 接続
- 接続

HLR 機能

設定

- SMSc 設定 map_client_enabled: true
- OmniMessage OmniMessage API 設定 smc_api_base_url
- 5 SMS.FrontendRegistry OmniMessage
- API OmniMessage API 設定 5
- MAP MAP SRI-for-SM MT-ForwardSM 設定 10
- SMS 設定
- STP M3UA MAP SMS 設定
- 接続

- **Web UI** 웹 UI Web UI
- **API** REST API Swagger UI API

HTTP API ☐☐

OmniMessage

OmniSS7 HTTPS REST API OmniMessage

```
config :omniss7,
  # OmniMessage API URL
  smsc_api_base_url: "https://10.5.198.200:8443",
  # SMSC hostname_SMSC
  smsc_name: "omni-smsc01",
  # GT
  smsc_service_center_gt_address: "5551234567"
```

□□□□□

項目	項目名	項目値	項目説明
項目名	smsc_api_base_url	"https://localhost:8443"	C A
項目名	smsc_name	"{hostname}_SMS" {hostname}_SMS	[S
項目名	smsc_service_center_gt_address	"5551234567"	[[C [F [[S

項目名

項目名 OmniMessage 5 SMS.FrontendRegistry 項目名
OmniMessage 項目名

- 項目名
- 項目名
- 項目名
- 項目名 SMS 項目名

項目名

- 項目名5 項目名
- 項目名 smsc_mode_enabled: true 項目名

項目名

```
{
  "frontend_name": "omni-smsc01",
  "configuration": "{...}",
  "frontend_type": "SS7",
  "hostname": "sm-sc-server01",
  "uptime_seconds": 12345
}
```

スクリプトは `sm-sc_name` 変数にスクリプトのホスト名を `"{hostname}_SMSc"` と

OmniMessage API

OmniSS7 は SS7 ネットワークを MAP 形式に変換する OmniMessage

- 送信元アドレス
- 送信元アドレス (MO) 番号
- 送信元アドレス (MT) 番号
- 送信元アドレス / 送信元
- 送信元アドレス

URI	Method	Request	Response
/api/frontends	POST	JSON Body	{ "frontend_name": "...", "frontend_type": "SMS", "hostname": "...", "uptime_seconds": ... }
/api/messages_raw	POST	JSON SMS Body	{ "source_msisdn": "...", "source_smsc": "...", "message_body": "..." }
/api/messages	GET	JSON Body	Body: smsc: <smsc_name>
/api/messages/{id}	PATCH	JSON JSON	{ "deliver_time": "...", "dest_smsc": "..." }
/api/messages/{id}	PUT	JSON Body	{ "dest_smsc": null }
/api/locations	POST	JSON/JSON JSON Body	{ "msisdn": "...", "imsi": "...", "location": "...", "ims_capable": true, "csfb": false, "expires": "...", "user_agent": "...", "ran_location": "...", "imei": "...", "registered": "..." }
/api/events	POST	JSON Body	{ "message_id": ..., "name": "...", "description": "..." }
/api/status	GET	JSON	-

API 詳細

API 詳細は JSON 形式で提供されます。

- 返回HTTP 200-201JSON 数据
- 返回HTTP 4xx/5xx错误码
- 时间ISO 8601 格式 "2025-10-21T12:34:56Z"
- 唯一 ID

API 接口

SMS 接口

1. SMS.APIClient

API 接口 OmniMessage HTTP API

- `frontend_register/4` - OmniMessage
- `insert_message/3` - SMS Python 3
- `insert_location/9` - /
- `get_message_queue/2` -
- `mark_dest_smsc/3` -
- `add_event/3` -
- `flush_queue/2` - SRI-for-SM + MT-forwardSM
- `auto_flush/2` -

2. SMS.FrontendRegistry

-
- 5
- `smc_name`
-

3. SMS.Utils

SMS

- `generate_tp_scts/0` - TPDU SMS

□□□□□

□□ **SMS** □□□□□□□

M3UA 与 SCTP 接口

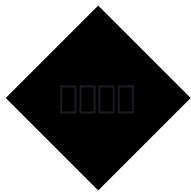
M3UA 消息格式

与 SCCP 接口

与 SCCP 接口

与 TCAP/MAP 接口

与 MAP 接口



Forward-SM

与 SMS TPDU

与 接口

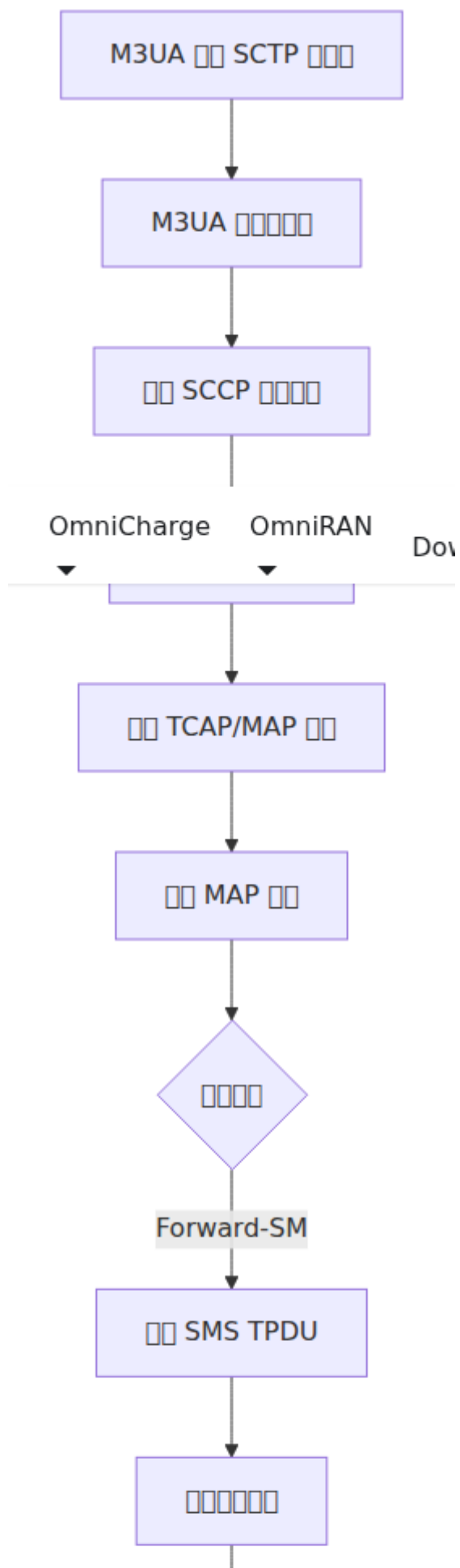
□□□□□□

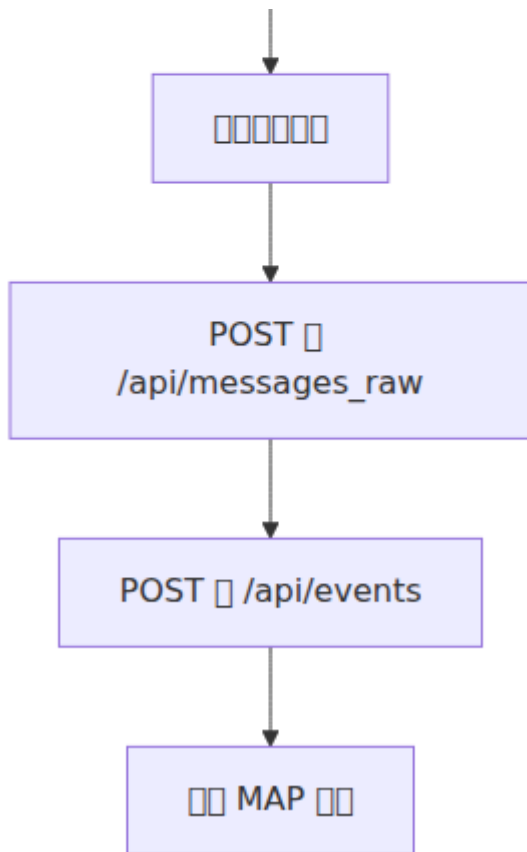
POST □
/api/messages_raw

POST □ /api/events

□□ MAP □□

□□ **SMS** □□□□□□□

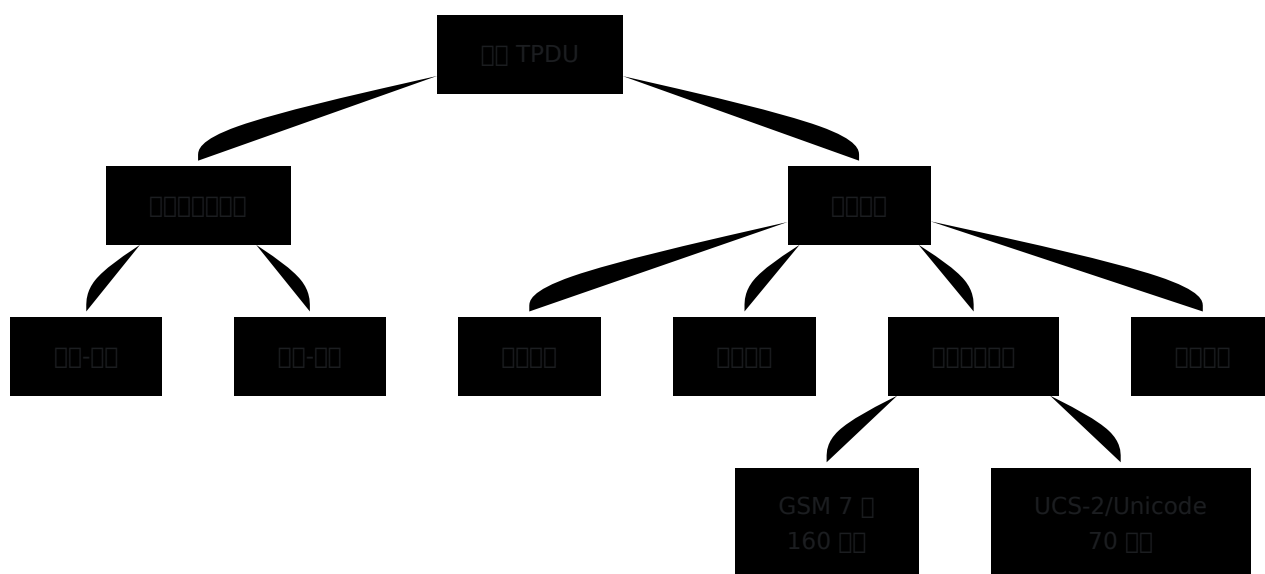




MAP

- **SRI-for-SM** SMS Sc HLR MSISDN SMS HLR
 - IMSI MSISDN - MSISDN ↔ IMSI
 - SMSC GT MT-ForwardSM
 - HLR SRI-for-SM
- **MT-forwardSM** SMS Sc SMS MSC/VLR

SMS TPDU 结构



配置 alertServiceCenter

SMS Sc 配置 HLR 的 **alertServiceCenter** 配置项

HLR 配置 alertServiceCenter 配置项 HLR 配置项

配置 alertServiceCenter

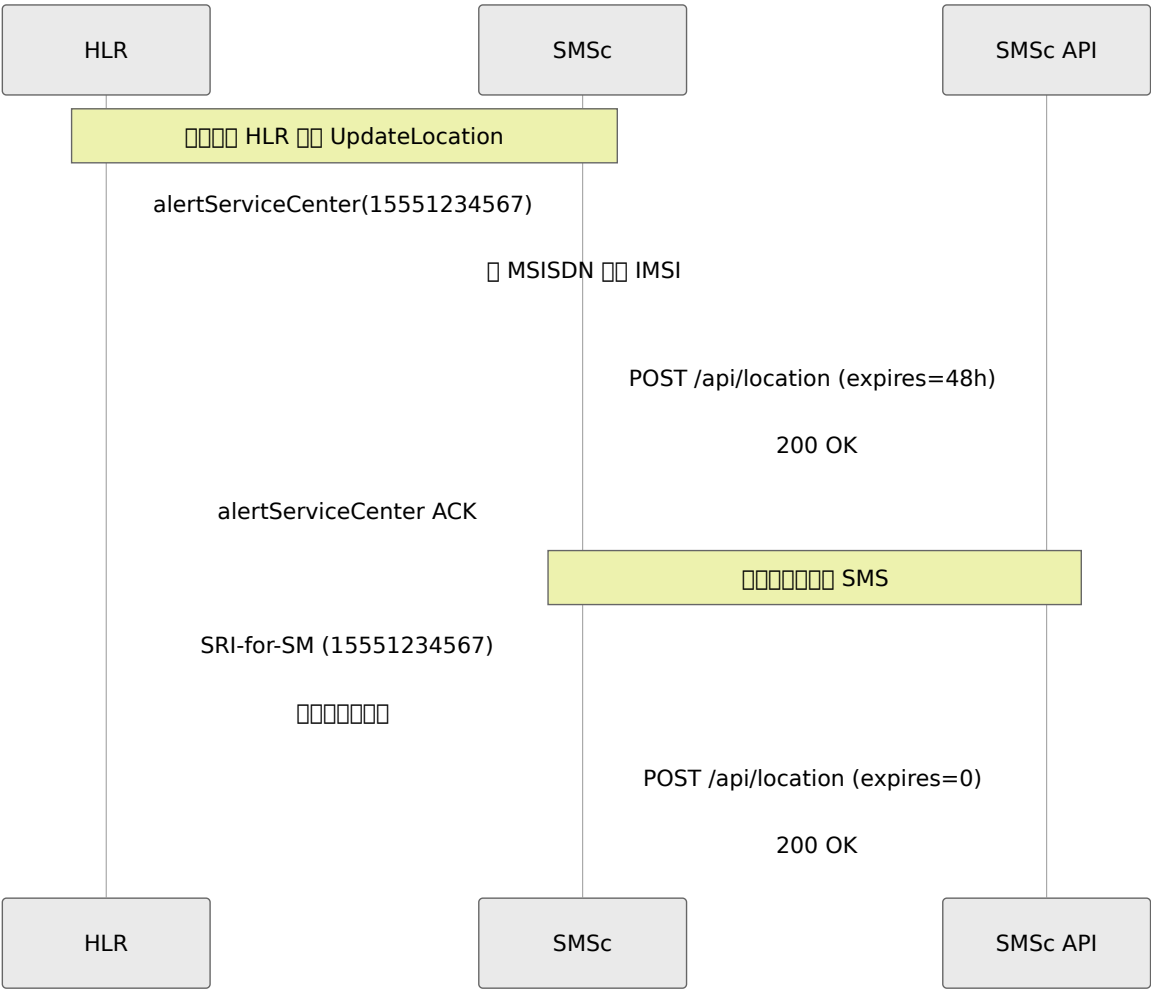
HLR 配置 UpdateLocation 配置项 VLR/MS Sc HLR 配置 **alertServiceCenter** MAP 配置 64 配置 SMS Sc 配置

配置

配置 HLR 配置

```
config :omniss7,  
  # SMS Sc alertServiceCenter 配置48 配置  
  hlr_alert_location_expiry_seconds: 172800
```


□□□



API □□

POST /api/location

```
{
  "msisdn": "15551234567",
  "imsi": "001010123456789",
  "location": "ipsmgw",
  "ims_capable": false,
  "csfb": true,
  "expires": "2025-11-01T12:00:00Z",
  "user_agent": "15551111111",
  "ran_location": "SS7",
  "imei": "",
  "registered": "2025-10-30T12:00:00Z"
}
```

user_agent alertServiceCenter HLR GT SMS Sc HLR

expires

SMS Sc SS7

SMS Sc SS7 (MO) source_smsc "SS7_GT_15551234567"

1. SS7 → source_smsc "SS7"
2. →
3. SS7 →

SMS Sc

項目	値	説明
SS7 番号 (MO-FSM)	"SS7_GT_15551234567"	発信元 - 発信番号
API/SMPP	"ipsmgw" と "api_gateway"	接続先
SMSc	"smsc-node-01"	送信先

JSON

メッセージイベントの構造

```
{
  "message_id": 12345,
  "name": "SMS",
  "description": "SMS - source_smsc 'SS7_GT_15551234567' から 'SS7' 経由で送信"
```

API エンドポイント

- Web UIでSS7 イベント (/events)
- REST API の events エンドポイント
- イベントのフィルタリング

例

送信元が SS7 を経由して SMS を送信した場合

JSON

送信元が SS7 を経由して SMS を送信した場合

1. 00000 → MSC/VLR → SMSc0000 M0-ForwardSM
2. SMSc 0 GT 15551234567 00 M0-FSM
3. SMSc 00000source_smsc = "SS7_GT_15551234567"
4. 0000000000000000
5. SMSc 0000 source_smsc 00 "SS7" → 00
6. 000000"00000"
7. 00000000
8. 0000 SRI-for-SM 0 MT-ForwardSM00000000

000000000000 8 0000000000 SS7 0000000000000000

SMSc 000000

SMSc 0000 0000000 GenServer00000000 alertServiceCenter 000000000000000000

00

00000000

- 000000000000000000
- **HLR** 000000 alertServiceCenter 0 HLR
- 0000000000000000/00000000
- 00000000000000000000000000
- **Web UI** 0000000000000000000000

00000000

000000000000000000

項目	項目	項目
msisdn	電話番号	"15551234567"
imsi	IMSI	"001010123456789"
hlr_gt	alertServiceCenter HLR GT	"15551111111"
messages_sent	MT-FSM	5
messages_received	MO-FSM	2
status	:active :failed	:active
updated_at	Unix	1730246400



alertServiceCenter

Active

SRI-for-SM | alertServiceCenter

Failed

/



alertServiceCenter

- 000000000000
- 00 status = :active
- 00 HLR GT
- 000000000000

SRI-for-SM

- `messages_sent`
- `updated_at`

SRI-for-SM

- `status = :failed`
- `xxxxxxxxxxxx`

□□□□□□□□

- ETS 規格
- Web UI

Web UI - SMSc ☐☐☐☐☐

□□□□□□ 2 □

00000000 SMC 00000000 config/runtime.exe 0000 SMC 000000000000
 0000

[illegible]

11

1. □□□□

- MSISDN IMSI HLR GT
- /□□□□□□□□
- /□□□□□□□□
-

- 0000000000

2. 0000

- 000000
- 0000000
- 0000000
- 00 HLR 000

3. 00

- 0000000000000000
- 0000000000

0000

SMSc 000000		00: 3			
MSISDN	IMSI	HLR GT	Msgs S/R	00	
15551234567	001010123456789	15551111111	5/2	● 00	
15559876543	001010987654321	15551111111	0/0	● 00	
15551112222	001010111222233	15552222222	3/1	○ 00	

000

000 00: 3 | 00: 2 | 00: 1 | 00 HLR: 2

API 00

0000000000000000

```
# 注册 alertServiceCenter 接口
SMSc.SubscriberTracker.alert_received(msisdn, imsi, hlr_gt)

# 注册消息接口
SMSc.SubscriberTracker.message_sent(msisdn)
SMSc.SubscriberTracker.message_received(msisdn)

# 注册 SRI-for-SM 接口
SMSc.SubscriberTracker.mark_failed(msisdn)

# 注册删除接口
SMSc.SubscriberTracker.remove_subscriber(msisdn)

# 注册查询接口
SMSc.SubscriberTracker.get_active_subscribers()
SMSc.SubscriberTracker.get_subscriber(msisdn)
SMSc.SubscriberTracker.count_subscribers()
SMSc.SubscriberTracker.clear_all()
```

□□

□□□□□□□□□□□□

- **alertServiceCenter** □□□□□□□□□□□□□□□□ `alert_received/3`
- **SRI-for-SM** □□□□□□□□□□□□□□□□ `messages_sent`
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□ `mark_failed/1`
- □□□□□□□□□□ SRI-for-SM □□□□□□□□□□□□□□□□ `mark_failed/1`

□□□□ SMS □□

□□□ □□□□□□□□□□ SMS □□□

□□□□□□□□□□□□ □□□□□□□□□□□□□□□□

配置

```
config :omniss7,  
  auto_flush_enabled: true,           # 是否/是否启用  
  auto_flush_interval: 10_000,        # 刷新间隔  
  auto_flush_dest_smsc: nil,          # 刷新目标nil = 空  
  auto_flush_tps: 10                  # 刷新速率
```

配置项

- 配置 `auto_flush_interval` 刷新 API 刷新间隔
- 配置 `auto_flush_dest_smsc` 刷新目标
- 配置 `auto_flush_tps` 刷新速率
- 配置 `auto_flush_enabled`
 - 配置 **SRI-for-SM** 刷新间隔 HLR 刷新间隔
 - HLR 刷新间隔 IMSI 刷新 MSISDN 刷新
 - HLR 刷新 SMSC GT 刷新 MT-ForwardSM 刷新
 - 刷新间隔 HLR 刷新 SRI-for-SM 刷新
 - 配置 **MT-forwardSM** 刷新 MSC/VLR
 - 配置 API 刷新间隔/刷新
 - 配置 API 刷新速率

配置 `auto_flush_dest_smsc` SRI-for-SM 刷新间隔 MSISDN 刷新 IMSI 刷新 SMSC GT 刷新
刷新 IMSI 刷新 HLR 刷新 SRI-for-SM 刷新

SMSc 配置

配置项

SMS 配置

- `smc_queue_depth` - 队列深度
- `smc_messages_delivered_total` - 消息交付总数
- `smc_messages_failed_total` - 消息失败总数

- `smsc_delivery_duration_milliseconds` - 消息传递时间

配置

```
# 队列深度
smsc_queue_depth

# 每5分钟计算一次
rate(smsc_messages_delivered_total[5m]) /
(rate(smsc_messages_delivered_total[5m]) +
rate(smsc_messages_failed_total[5m]))

# 消息传递时间
rate(smsc_delivery_duration_milliseconds_sum[5m]) /
rate(smsc_delivery_duration_milliseconds_count[5m])
```

消息传递 SMSG

消息传递流程

步骤

1. 消息传递请求
2. 消息传递接收
3. 消息传递处理
4. 通过 M3UA 消息传递接收
5. 通过 TPS 消息

消息传递配置

配置

- TPS 配置
- HLR 配置
- 消息传递

- 00000000

00000

- 00 auto_flush_tps
- 0000 HLR 000
- 0000000000

MT-forwardSM API

00 API 00 SMS

API 000 POST /api/MT-forwardSM

000

```
{
  "imsi": "234509876543210",
  "destination_serviceCentre": "447999555111",
  "originating_serviceCenter": "447999123456",
  "smsPDU":
  "040B917477218345F600001570301857140C0BD4F29C0E9281C4E1F11A"
}
```

000

```
{
  "result": "success",
  "message_id": "12345"
}
```



OmniSS7 詳細

- ← 詳細
- HLR 詳細 - HLR 詳細
 - **SRI-for-SM** 詳細 - MSISDN と IMSI の関係
- 詳細 - Web UI/API 詳細
- MAP 詳細 - MAP 詳細
- 詳細 - 詳細

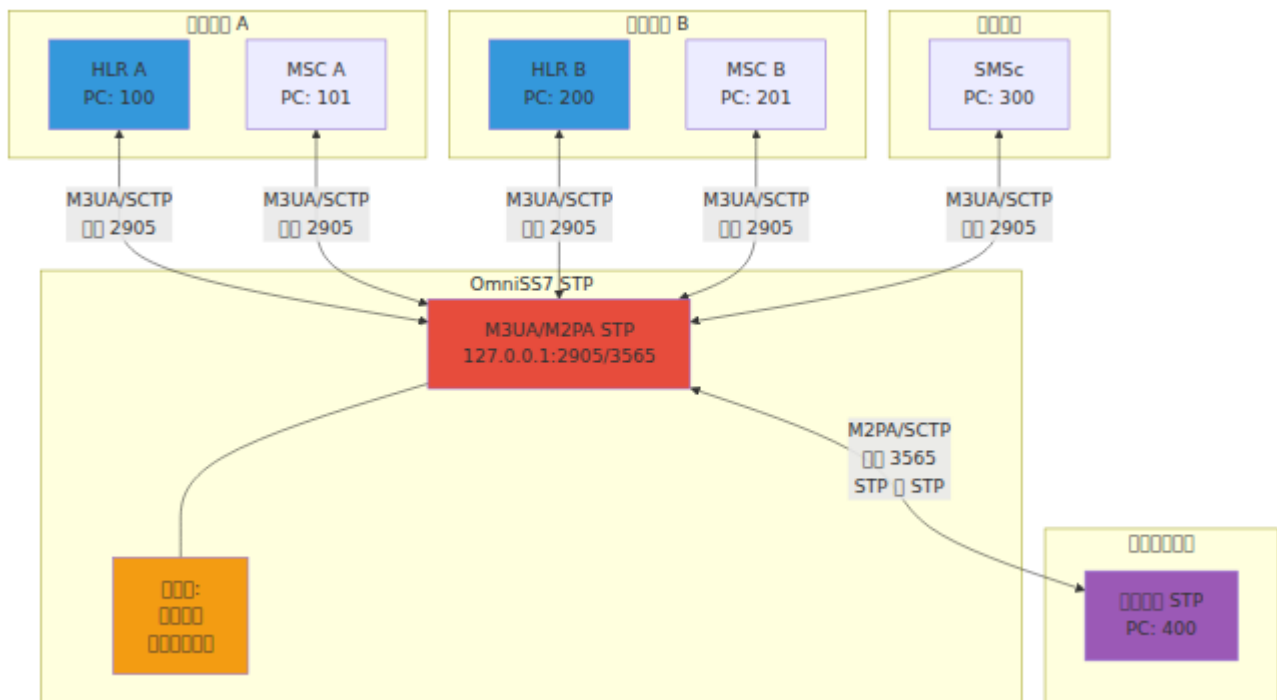
OmniMessage 詳細 詳細詳細詳細詳細詳細詳細詳細詳細詳細 **OmniMessage** 詳細
詳細OmniMessage 詳細詳細詳細詳細詳細詳細詳細詳細詳細詳細

OmniSS7 と Omnitouch 詳細

STP 関

- 端末 (PC) 端末 (GT) 間 SS7 関
- 端末 SS7 端末 IP 間 M3UA/SCTP 関
- 端末端末端末端末端末端末端末
- 端末端末端末端末端末端末
- 端末端末端末端末端末端末

STP 概要



STP 機能概要

ASP (アプリケーションサーバ)

- 各アプリケーション SGP/STP 接続
- 各アプリケーション
- 各アプリケーション STP 接続

SGP (サービスゲートウェイ)

- 各アプリケーション ASP 接続
- 各アプリケーション
- 各アプリケーション STP

AS (アプリケーションサーバ)

- 各アプリケーション ASP 接続

- 0000000000000000
 - 000000 ASP 00000000
-

00 M3UA STP 00

OmniSS7 0000000000000000 STP0000000000 STP 000

0000 STP 00

OmniSS7 0 `config/runtime.exs` 0000000000000000 STP 000

1. 00 `config/runtime.exs`
2. 00 00000000 53-174 000
 - 00 10STP 0000 53-85 00
 - 00 20HLR 0000 87-123 00
 - 00 30SMSc 0000 125-174 00
3. 000 00000000000000 #0
4. 0000 STP 00000 53-85 000 #0
5. 00000000 0000
6. 00 00000 `iex -S mix`

STP □□□□

□□□ STP □□□□□□□□

```

config :omniss7,
  # STP - STP
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: false,

  # M3UA
  # ASP STP/SGW
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :stp_client_asp,
    #
    local_ip: {10, 179, 4, 10},
    local_port: 2905,
    # STP/SGW
    remote_ip: {10, 179, 4, 11},
    remote_port: 2905,
    routing_context: 1
  }

```

STP

STP STP

項目	型	初期値	説明	デフォルト
map_client_enabled	bool	true	MAP クライアントの有効化	true
local_ip	IP アドレス		ローカル IP アドレス {10, 0, 0, 1} 範囲 [{10, 0, 0, 1}, {10, 0, 0, 2}]	{10, 179, 4, 10}
local_port	ポート	2905	ローカル SCTP ポート	2905
remote_ip	IP アドレス		リモート STP/SGW IP アドレス	{10, 179, 4, 11}
remote_port	ポート	2905	リモート SCTP ポート	2905
routing_context	整数	1	M3UA ルーティング ID	1
enable_gt_routing	bool	false	グローバルルーティング PC 有効化	true

ローカル IP アドレス local_ip とリモート IP アドレス remote_ip は SCTP 接続に使用されます。
SCTP 接続

STP 接続

map_client_enabled: true Web UI 設定

- SS7 接続 - 設定
- SS7 接続 - MAP 接続
- M3UA - 設定
- 接続 - 接続 ← STP 接続
- 接続 - 接続 ← STP 接続
- 接続 - 接続
- 接続 - 接続

HLR 和 SMSc 的部署

部署

- SCTP 端口 IP 为 132
- M3UA 端口 2905
- 部署
- 部署 Web UI 和 API 部署 **Mnesia** 部署 部署
- 部署 `runtime.exs` 部署 Mnesia 部署
- 部署
- **Web UI** 部署 Web 部署 **Web UI**
- **API** 部署 REST API 部署 Swagger UI 部署 **API**

部署 STP

部署 `map_client_enabled: true` 部署 STP 部署 **M3UA STP** 部署

部署 STP

部署 `config/runtime.exs`

```
config :omniss7,  
  m3ua_stp: %{  
    enabled: true,  
    local_ip: {127, 0, 0, 1},    # IP  
    local_port: 2905,           #  
    point_code: 100             # STP  
  }
```

STP 配置

属性	类型	默认值	描述	范围
enabled	布尔	false	是否启用 STP 服务	true
local_ip	IP	{127, 0, 0, 1}	本地 IP 地址	{0, 0, 0, 0}
local_port	端口	2905	本地端口	2905
point_code	点码	00	STP 点码 SS7 点码	100

STP 配置

- 配置 M3UA 和 MAP 服务
- 配置 STP 服务
- 配置 STP 服务 HLR、MSC 和 SMSC

配置 STP 服务需要配置 `map_client_m3ua` 和 `m3ua_stp`

Mnesia (Mnesia)

Mnesia 是 Erlang 的分布式数据库

配置

- Runtime.exs** 配置 `config/runtime.exs` 配置 `m3ua_peers`、`m3ua_routes` 和 `m3ua_gt_routes`
- Web UI** 配置 `Web UI` 配置 Mnesia
- 配置 `runtime.exs` 配置 Mnesia
- 配置 Web UI 配置

Mnesia 配置

配置 Mnesia 数据库时，需要指定存储类型和副本数。

```
config :omniss7,  
  mnesia_storage_type: :disc_copies # 指定 :ram_copies 副本数
```

配置项	类型	默认值	说明
<code>:disc_copies</code>	列表	<code>[{disc_copies, 1}]</code>	指定使用磁盘存储的副本数
<code>:ram_copies</code>	列表	<code>[{ram_copies, 1}]</code>	指定使用内存存储的副本数

配置 `:disc_copies`

Mnesia 部署

Mnesia 部署需要配置 Mnesia 节点。

- 配置 `Mnesia.{node_name}/Mnesia.nonnode@nohost/`
- 配置 `m3ua_peer` `m3ua_route` `m3ua_gt_route`

部署

部署步骤

- Runtime.exs** - 配置 Mnesia 节点
- Web UI** - 部署 Web UI 界面
- REST API** - 部署 REST API 接口

部署 `runtime.exs` 部署 Web UI 部署 REST API

部署 M3UA

部署 M3UA 需要配置 STP、HLR、MSC、SMSC 等组件。配置 `config/runtime.exs`

配置 STP 和 MSC

```
config :omniss7,
  m3ua_peers: [
    # 配置 STP 客户端
    %{
      peer_id: 1,
      name: "Partner_STP_West",
      role: :client,
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 10},
      remote_port: 2905,
      routing_context: 1,
      point_code: 100,
      network_indicator: :international
    },
    # 配置 HLR 客户端
    %{
      peer_id: 2,
      name: "Local_HLR",
      role: :client,
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 20},
      remote_port: 2905,
      routing_context: 2,
      point_code: 200,
      network_indicator: :international
    },
    # 配置 MSC 服务器
    # 配置 :server STP 服务器
    %{
      peer_id: 3,
      name: "Remote_MSC",
      role: :server,
      remote_ip: {10, 0, 0, 30},
      remote_port: 2905,
      routing_context: 3,
```

```

    point_code: 300,
    network_indicator: :international
  },

  # 中国
  %{
    peer_id: 4,
    name: "Dynamic_Client",
    role: :server,
    remote_ip: {10, 0, 0, 40},      # 公网 IP
    remote_port: 0,                 # 0 = 任意端口
    routing_context: 4,
    point_code: 400,
    network_indicator: :international
  }
]

```

フィールド定義

フィールド名	型	制約	説明
peer_id	整数	必須	ピアの一意の識別子
name	文字列 最大 255 文字	必須	ピアの名前
role	文字列	必須	:client または :server
local_ip	IP アドレス または SCTP アドレス	必須	ローカル IP アドレス {10, 0, 0, 1} または SCTP アドレス [{10, 0, 0, 1}, {10, 0, 0, 2}]
local_port	整数	必須	ローカルポート番号
remote_ip	IP アドレス または SCTP アドレス	必須	リモート IP アドレス {10, 0, 0, 10} または [{10, 0, 0, 10}, {10, 0, 0, 11}]
remote_port	整数	必須	リモートポート番号
routing_context	文字列	任意	M3UA ルーティングコンテキスト
point_code	整数	任意	SS7 ポイントコード
network_indicator	文字列	任意	:international または :national

SCTP アドレスは `local_ip` と `remote_ip` の IP アドレスまたは SCTP アドレスで指定されます。

フィールドの値

例: `:server` の `remote_port` は 12345

- `remote_port: 2905` `peer_id`
 - `peer_id`
 - `peer_id`
- `remote_port: 0` `peer_id`
 - `peer_id`/`peer_id`
 - `IP` `IP`
 - `peer_id`

peer_id

```
# 10.5.198.200:2905 peer_id
%{
  peer_id: 1,
  name: "Strict_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 2905,
  # ... peer_id
}

# 10.5.198.200 peer_id
%{
  peer_id: 2,
  name: "Flexible_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 0, # peer_id
  # ... peer_id
}
```

M2PA peer_id

OmniSS7 peer **M3UA** peer **M2PA** peer SS7 peer

M2PA MTP2 IETF RFC 4165 IP SCTP SS7 MTP3

M3UA M2PA

	M3UA	M2PA
	ASP/SGW	
	SS7 IP	
	ASPUP/ASPAC	MTP2
		24 BSN/FSN
	SS7 IP STP	
RFC	RFC 4666	RFC 4165

M3UAM2PA

M3UA

M3UA

- STP
- SS7 IP
- HLRMSCSMSC STP
- (SGW)
-
- RFC 4666

□□ M3UA □□□□□ HLR□ MSC□ SMSC□ VLR □□□□□ STP□

□□□□ **M2PA**□□□□□□□□

M2PA

- [illegible]

□□□□□□ M2PA □□□□□□HLR□MSC□SMSC□□□□□□ STP - □□□ M3UA□M2PA □□□□ STP
□ STP □□□□□□□□□□□□□□□□□□

□□ **M2PA** □□□

M2PA □□□□□□ M3UA □□□□□□□□□□ protocol □□

M2PA

M2PA config/runtime.exs m3ua_peers

M2PA ☐☐☐☐☐☐

項目	値	説明
protocol	:m2pa	プロトコル M2PA のみサポートされています :m3uaはサポートされていません
role	:client または :server	クライアントまたはサーバー
local_port	ポート番号	ローカル Sctp ポート番号 M2PA ポート 3565
remote_port	ポート番号	リモート Sctp ポート番号 M2PA ポート 3565
point_code	ポイントコード	ポイントコード
adjacent_point_code	ポイントコード	隣接ポイントコード M2PA ポート

プロトコル M2PA のみ ポート **3565** のみサポートされています M3UA ポート 2905 はサポートされていません

M2PA のインストール

M2PA のインストール手順

1. 実行 - インストール
2. 実行 - インストール 1
3. 実行 - インストール 2
4. 実行 - インストール

インストール完了後

Web UI から M2PA をインストール

Web UI から M2PA をインストール

1. Web UI
2. “インストール” ボタン

3. 如何“配置设备”
4. 如何“M2PA (RFC 4165)” 配置设备
5. 如何 配置设备
 - 配置设备名称和地址
 - 配置M2PA
 - 配置设备名称和地址
 - 配置设备 PC
 - 配置/设备 IP 地址
 - 配置/设备名称和地址 3565 和 M2PA
 - 配置设备名称和地址
6. 如何“配置设备”

配置设备名称和地址

- 如何 - M3UA 配置
- 如何 - M2PA 配置

M2PA 配置

M2PA 配置 OmniSS7 配置设备名称和地址

- 配置设备名称 M2PA 和 M3UA 配置设备
- 配置设备名称 M2PA 配置设备
- 配置设备名称 M2PA 和 M3UA 配置设备名称和地址
- 配置设备名称 M2PA 配置设备名称 M3UA配置设备

M2PA 配置

M2PA 配置 Prometheus 配置设备名称和地址

配置设备

- `m2pa_messages_sent_total` - 配置设备名称 MTP3 配置
- `m2pa_messages_received_total` - 配置设备名称 MTP3 配置
- `m2pa_bytes_sent_total` - 配置 M2PA 配置设备
- `m2pa_bytes_received_total` - 配置 M2PA 配置设备

link_name point_code adjacent_pc

- m2pa_link_state_changes_total - DOWN → ALIGNMENT → PROVING → READY
 - link_name from_state to_state

- m2pa_errors_total -
 - decode_error - M2PA
 - encode_error - M2PA
 - sctp_send_error - SCTP
 - link_name error_type

- Prometheus <http://your-server:8080/metrics>
-

M2PA

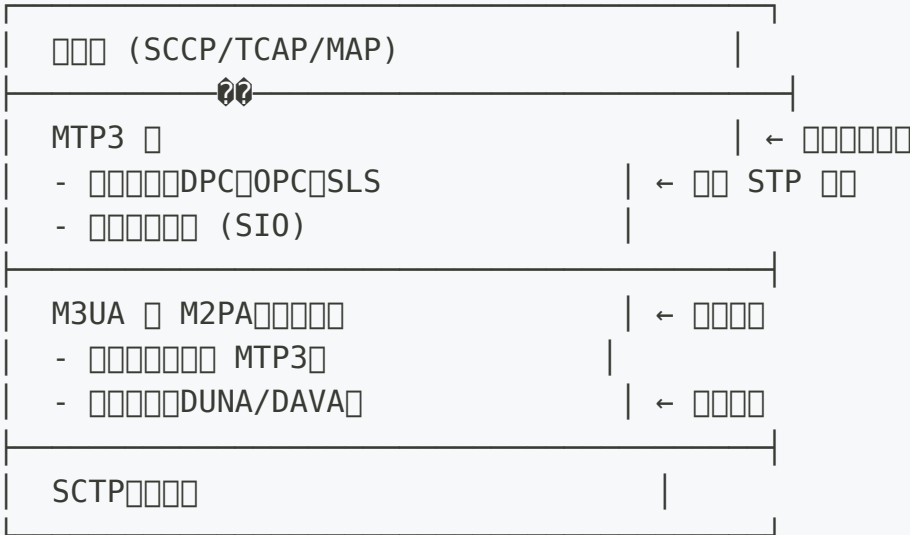
1. 3565 M2PA
- 2.
3. SCTP/IP 132
- 4.
- 5.
- 6.

MTP3 (DPC)

SS7 프로토콜

SS7 프로토콜은 전화망에서 사용되는 프로토콜이다.

SS7 프로토콜은



SS7 프로토콜은

1. MTP3 프로토콜

- MTP3 프로토콜은 DPC/OPC를 사용한다.
- M3UA 프로토콜은 528을 사용한다.
- M2PA 프로토콜은 사용된다.
- STP 프로토콜은 DPC 프로토콜을 사용한다.
- 신호 전송 프로세서 (STP)는 사용된다.

2. M3UA 프로토콜

- M3UA 프로토콜은 DUNA/DAVA/SCON/DUPU를 사용한다.
- 신호 전송 프로세서 (STP)는 사용된다.
- 신호 전송 프로세서 (STP)는 사용된다.
- 신호 전송 프로세서 (STP)는 사용된다.

STP 프로토콜

- **M3UA** **STP** **528** **MTP3** **MTP3** **DPC** **OPC** **SLS** **MTP3** **DPC**
- **M2PA** **STP** **M2PA** **MTP3** **MTP3** **DPC**
- **M3UA** **DUNA** **DAVA** **SCON** **M3UA**

`config/runtime.exs`

```

config :omniss7,
  m3ua_routes: [
    # 100000000 PC 100 10000 STP
    %{
      dest_pc: 100,          # 10000
      peer_id: 1,            # 1000000
      priority: 1,           # 1000000000000000
      network_indicator: :international
      # mask: 14              # 1000000 140000000
    },

    # 100000000 PC 200 10000 2000 HLR
    %{
      dest_pc: 200,
      peer_id: 2,
      priority: 1,
      network_indicator: :international
    },

    # 100000000 PC 300 100000000
    %{
      dest_pc: 300,
      peer_id: 3,            # 1000
      priority: 1,
      network_indicator: :international
    },
    %{
      dest_pc: 300,
      peer_id: 4,            # 1000000000000000
      priority: 2,
      network_indicator: :international
    }
  ]
]

```

mask 14 mask

10000

1. STP M3UA M2PA
2. STP M3UA M2PA **MTP3**
3. STP MTP3 **(DPC)**

4. ☐ DPC
5. ☐
6. ☐ MTP3 ☐ M3UA ☐ M2PA
7. ☐
8. ☐

--

□□□ 14 □□□□□ 0-16383□□□□□□□□□□□□□□□□□□□□ /14□□□□□□□□□□ □□□□ □□□□

□□□□ □□□□

□□□□

□□□□□□□□ PC □□□□□ DPC □□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

階	階数	階
/14	1 PCs	階
/13	2 PCs	階
/12	4 PCs	階
/11	8 PCs	階
/10	16 PCs	階
/9	32 PCs	階
/8	64 PCs	階
/7	128 PCs	階
/6	256 PCs	階
/5	512 PCs	階
/4	1,024 PCs	階
/3	2,048 PCs	階
/2	4,096 PCs	階
/1	8,192 PCs	階 PCs
/0	16,384 PCs	階 PCs/階

階

階 mask 階 階 14 階

階 1 階

```
# 1000000000000000 PC
%{
  dest_pc: 1000,
  peer_id: 1,
  priority: 1,
  network_indicator: :international
}
# 000000 14 - 0000 PC 1000

# 1000000000000000
%{
  dest_pc: 1000,
  peer_id: 1,
  priority: 1,
  mask: 14, # 000000
  network_indicator: :international
}
# 0000 PC 1000
```

2

```
%{
  dest_pc: 1000,
  peer_id: 2,
  priority: 1,
  mask: 12, # 00 4 PCs
  network_indicator: :international
}
# 000PC 100001001002003
```

3

```
%{
  dest_pc: 1000,
  peer_id: 3,
  priority: 1,
  mask: 8,
  network_indicator: :international
}
# 64 PCs
# PC 1000-1063 64
```

4/

```
%{
  dest_pc: 0,
  peer_id: 4,
  priority: 10,
  mask: 0,
  network_indicator: :international
}
# 0-16383
# PCs
```

```

config :omniss7,
  m3ua_routes: [
    # [] PC 1000 []
    %{
      dest_pc: 1000,
      peer_id: 1,
      priority: 1,
      network_indicator: :international
      # [] 14[]
    },

    # [] PCs 1000-1063 []
    %{
      dest_pc: 1000,
      peer_id: 2,
      priority: 1,
      mask: 8, # [] 64 PCs
      network_indicator: :international
    },

    # [] PCs []/[]
    %{
      dest_pc: 0,
      peer_id: 3,
      priority: 10, # []
      mask: 0, # [] PCs
      network_indicator: :international
    }
  ]

```

DPC 1000 []

1. [] /14 [] PC 1000 [] - []
2. [] /8 [] PC 1000-1063 [] - []
3. [] /0 [] PCs - []

DPC 1015 []

1. [] /14 [] PC 1000
2. [] /8 [] PC 1000-1063 [] - []
3. [] /0 [] PCs - []

DPC 5000

1. /14
2. /8
3. /0 PCs -

1. mask /14
2. /14 mask: 14
3. /0 /13
4. /0
- 5.
- 6.

(GT)

IMSI / NAT

□□□□

- □□ GT □□□□ `config/runtime.exs` □□□ `enable_gt_routing: true`

GT 配置

```
config :omniss7,
  # 是否 GT 配置
  enable_gt_routing: true,

  m3ua_gt_routes: [
    # 配置 44 的 GT 路由 1
    %{
      gt_prefix: "44",          # 配置 GT 前缀
      peer_id: 1,               # 配置 GT 对等体 ID
      priority: 1,              # 配置 GT 路由优先级
      description: "配置 44"    # 配置 GT 路由描述
    },

    # 配置 1 的 GT 路由 2
    %{
      gt_prefix: "1",
      peer_id: 2,
      priority: 1,
      description: "配置 1"
    },

    # 配置 447 的 GT 路由 3
    %{
      gt_prefix: "447",         # 配置 GT 前缀
      peer_id: 3,               # 配置 GT 对等体 ID
      priority: 1,
      description: "配置 447"
    },

    # SSN 配置 4
    %{
      gt_prefix: "555",         # 配置 GT 前缀
      source_ssn: 8,            # 配置 SSN = 8 的 SMSC 对等体 ID
      peer_id: 4,
      dest_ssn: 6,              # 配置 SSN 6 的 HLR 对等体 ID
      priority: 1,
      description: "配置 61 的 SMS 对等体"
    }
  ]
]
```

GT □□□□

GT □□□□□□□□□□□□□□

SCCP

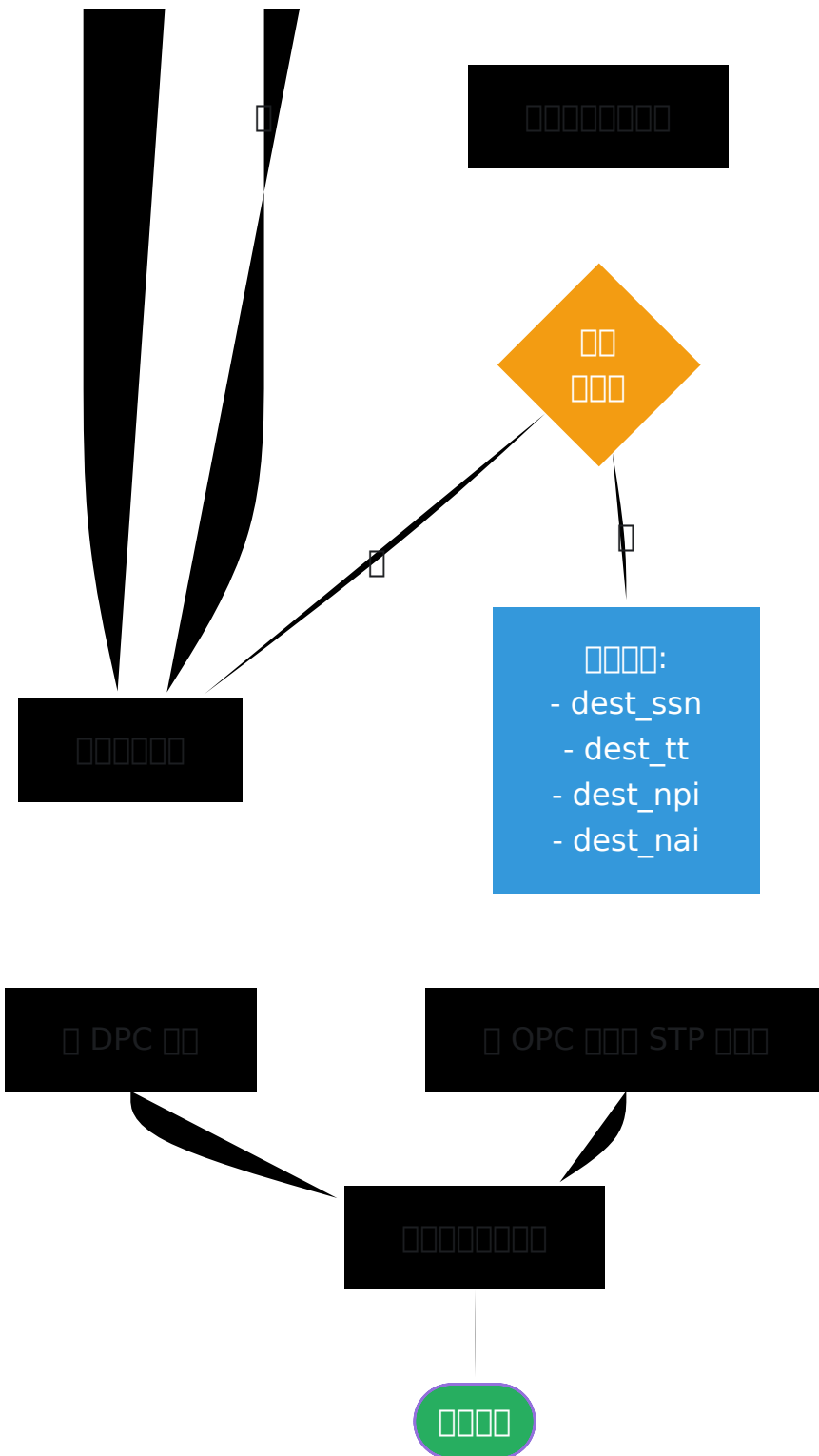
GTSSNTTNPI
NAI

GT
SSN

GTSSNTTNPI
+ NAI

SSN
TT

- SSN:
1. GT
 2. SSN >
 3. TT >
 4. NPI >
 5. NAI >
 - 6.



000000

1. 000000STP 0000 GT 00000000000000000000

◦ 0000GT "447712345678" 00 "44" 0 "447"00 "447" 0000000000

2. **SSN** 00000000

- 00000 source_ssn 0000 SCCP 000 SSN 0000000000
- 00 source_ssn 0 nil 0000000000 SSN 000000

3. TT/NPI/NAI 00000000

- 00000 source_tt 0 source_npi 0 source_nai 0000000000000000
- nil 0000000000000000

4. 0000000000

- 00000000000000000000
- 00000000GT 0000 → SSN → TT → NPI → NAI → 000000

5. 0000000000

- 00000 dest_ssn 0 dest_tt 0 dest_npi 0 dest_nai 0STP 0000000000

- 國際電報號碼

6. 呼叫號碼

- 國際 GT 號碼 STP 號碼 DPC 號碼

國際 GT 號碼 NPI 和 NAI

國際 GT 號碼 SSN 號碼 STP 號碼 SCCP 號碼

- 國際 (TT) 號碼
- 國際 (NPI) 號碼 ISDN 號碼
- 國際 (NAI) 號碼

國際號碼

國際號碼

- `source_tt` 國際號碼
- `source_npi` 國際號碼
- `source_nai` 國際號碼
- `nil` = 國際號碼

國際號碼

國際號碼

- `dest_tt` 000000000000
- `dest_npi` 0000000000000000
- `dest_nai` 0000000000000000
- `nil` 0 = 000000000000

00000000

0000000000000000000000000000

1. 00 GT 0000
2. 000 SSN 00000 SSN
3. 000 TT 00000 TT
4. 000 NPI 00000 NPI
5. 000 NAI 00000 NAI
6. 00000000

0000

```

config :omniss7,
  enable_gt_routing: true,

m3ua_gt_routes: [
  # 1
  %{
    gt_prefix: "44",
    peer_id: 1,
    source_tt: 0,      # TT=0
    dest_tt: 3,        # TT=3
    priority: 1,
    description: "TT 0→3"
  },

  # 2 NPI NAI
  %{
    gt_prefix: "1",
    peer_id: 2,
    source_npi: 1,     # NPI=1ISDN/
    source_nai: 4,     # NAI=4
    dest_nai: 3,       # NAI=3
    priority: 1,
    description: "→ NAI"
  },

  # 3 SSN
  %{
    gt_prefix: "33",
    source_ssn: 8,     # SMSC
    source_tt: 0,      # TT=0
    dest_ssn: 6,       # SSN HLR
    dest_tt: 2,        # TT=2
    dest_npi: 1,       # NPI=1ISDN
    dest_nai: 4,       # NAI=4
    peer_id: 3,
    priority: 1,
    description: "SMS"
  },

  # 4 TT NPI
  %{
    gt_prefix: "49",
    source_tt: nil,    # TT
  }
]

```

```

    source_npi: 6,      # 00 NPI=60000
    dest_npi: 1,        # 000 NPI=10ISDN
    peer_id: 4,
    priority: 1,
    description: "0000000000"
  }
]

```

00 TT/NPI/NAI 0

0000 (TT)0

- 0 = 00
- 1 = 00
- 2 = 00
- 3 = 0000

0000000 (NPI)0

- 0 = 00
- 1 = ISDN/000E.1640
- 3 = 000X.1210
- 4 = 000F.690
- 6 = 00000E.2120

0000000 (NAI)0

- 0 = 00
- 1 = 0000
- 2 = 0000000000
- 3 = 0000000
- 4 = 0000

0000000

00000000

- GT: "447712345678"

- SSN: 8
- TT: 0
- NPI: 1
- NAI: 4

□□□□□□□□

```
# □□ A□□□□ TT
%{gt_prefix: "447", peer_id: 1, priority: 1}

# □□ B□□□ TT
%{gt_prefix: "447", source_tt: 0, peer_id: 2, priority: 1}

# □□ C□□□ TT + NPI
%{gt_prefix: "447", source_tt: 0, source_npi: 1, peer_id: 3,
priority: 1}
```

□□□□□ C □□□□□□□□□□ GT + TT + NPI□

□□□□□□ C □ dest_tt dest_npi dest_nai □□□□□□

GT 規則

GT	SSN	TT	NPI	NAI	説明	例
447712345678	6	-	-	-	"447" → GT 3	447712345678
441234567890	6	-	-	-	"44" → GT 1	441234567890
12125551234	6	-	-	-	"1" → GT 2	12125551234
555881234567	8	-	-	-	"555"SSN 8→ GT 4 SSN 6	555881234567
555881234567	6	-	-	-	"555"SSN → GT X	555881234567
441234567890	6	0	1	4	"44"TT=0 → GT 1 TT 3	441234567890
12125551234	8	0	1	4	"1"TT=0 NPI=1 NAI=4	12125551234

TT/NPI/NAI 規則

1. GT
 - SSN 8桁
 - SSN 6桁
 - SSN 6桁 TT=0 NPI=1 NAI=4
2. GT
 - SSN 8桁

- 國際電報號碼 NPI=6 國際 PSTN NPI=1

3. 國際電報號碼

- 國際電報號碼 NAI
- 國際電報號碼 NAI=4 國際電報號碼 NAI=3 國際電報號碼

4. 國際電報號碼

- 國際電報號碼
- 國際 TT=0 國際 A TT=2 國際 B

5. 國際電報號碼

- 國際電報號碼
- STP 國際電報號碼

國際電報號碼

國際電報號碼

國際電報號碼 國際電報號碼 國際電報號碼

國際電報號碼

國際電報號碼 enabled 國際電報號碼

3. `if ($?) { "DROP route matched for DPC 999" } else { "DROP route matched for GT 999" }`

4. `[:error, :dropped]`

[illegible]

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

DROP -

111

1. 0000000000 DROP 000000 99
2. 000000000000000000 000001
3. 00000000000000000000000000000000 DROP 000000
4. 0000000000000000 DROP 0000

55

1234 10,000 1234000000 - 1234999999 3
 1234567890 1234555000 1234111222

```

config :omniss7,
  m3ua_gt_routes: [
    # 1234567890 DROP 1234567890
    %{
      gt_prefix: "1234",
      peer_id: 0,          # DROP
      priority: 99,        # 1000 = 10000 = 10000
      description: "1234* 1234567890"
    },

    # 123456789012345678901234567890
    %{
      gt_prefix: "1234567890",
      peer_id: 1,          # 1234567890 1
      priority: 1,         # 1000 = 10000 = 10000
      description: "1234567890 1"
    },

    %{
      gt_prefix: "1234555000",
      peer_id: 1,
      priority: 1,
      description: "1234555000 2"
    },

    %{
      gt_prefix: "1234111222",
      peer_id: 1,
      priority: 1,
      description: "1234111222 3"
    }
  ]
]

```

1234567890

GT	Configuration	Configuration	Result
1234567890	1. "1234567890" 2. "1234" DROP 3. 99	"1234567890"   "1234" DROP	1 1
1234555000	1. "1234555000" 2. "1234" DROP 3. 99	"1234555000" "1234" DROP	1 1
1234111222	1. "1234111222" 2. "1234" DROP 3. 99	"1234111222" "1234" DROP	1 1
1234999999	"1234" DROP 99	"1234" DROP	+ "
1234000000	"1234" DROP 99	"1234" DROP	+ "

- 3 1
- 1234*
-

```
[INFO] DROP route matched for GT 1234999999
[INFO] DROP route matched for GT 1234000000
```

DROP

```

config :omniss7,
  m3ua_routes: [
    # DROP route /8 1000-1063
    %{
      dest_pc: 1000,
      peer_id: 0,
      priority: 99,
      mask: 8,
      network_indicator: :international
    },

    # 1000 PCs
    %{dest_pc: 1010, peer_id: 1, priority: 1, network_indicator: :international},
    %{dest_pc: 1020, peer_id: 1, priority: 1, network_indicator: :international},
    %{dest_pc: 1030, peer_id: 1, priority: 1, network_indicator: :international}
  ]

```

1000 PCs 1010-1020 & 1030 route 1000-1063 1000 PCs

Drop

```

# Test
tail -f logs/app.log | grep "DROP route matched"

# Output
[INFO] DROP route matched for GT 1234999999
[INFO] DROP route matched for DPC 1050

```

Web UI

- 1000 1000 1000
- **INFO** 1000
- "DROP route matched"

1. ⚠️ 0000000000 DROP 0000000000
 2. 0 00000 description 0000000000
 3. 0 000000000090-9900 DROP 00000000000000
 4. 0 00000000000000 DROP 0000
 5. 0 000000000000000000
-

00000000 SSN 00000000

000000 (SSN)

000000000000

- **SSN 6** 0HLR0000000000
- **SSN 7** 0VLR0000000000
- **SSN 8** 0MSC00000000/ SMSC000000
- **SSN 9** 0GMLC0000000000

00 **SSN**    0000

00000000 SMS 00000000 HLR

```

m3ua_gt_routes: [
  # 8 SMSC 6 HLR
  %{
    gt_prefix: "44",
    source_ssn: 8, # 8 SMSC
    peer_id: 1,
    dest_ssn: 6, # 6 HLR
    priority: 1,
    description: "SMS to HLR"
  },

  # 6 HLR
  %{
    gt_prefix: "44",
    source_ssn: 6, # 6 HLR
    peer_id: 1,
    dest_ssn: nil, # SSN
    priority: 1,
    description: "SSN"
  }
]

```

STP

1.

Web UI

- <http://localhost>
- M3UA
-

IEx

```
# 確認ステータス
M3UA.STP.get_peers_status()

# 確認
# [
#   %{peer_id: 1, name: "Partner_STP_West", status: :active,
# point_code: 100, ...},
#   %{peer_id: 2, name: "Local_HLR", status: :active, point_code:
# 200, ...}
# ]
```

2. 送信

```
# 確認 M3UA の DPC 100
test_payload = <<1, 2, 3, 4>> # 送信
M3UA.STP.route_by_pc(100, test_payload, 0)

# 確認ステータス
# 確認 "Routing message: OPC=... -> DPC=100 via peer 1"
```

3. 消息路由

```
# 通过 GT 查询
M3UARouting.lookup_peer_by_gt("447712345678")

# 返回结果
# {:ok, {:m3ua_peer, 3, "UK_Mobile_Peer", ...}, nil}

# 通过 SSN 和 GT 查询
M3UARouting.lookup_peer_by_gt("555881234567", 8)

# 返回结果 SSN 查询
# {:ok, {:m3ua_peer, 4, "SMS_HLR_Peer", ...}, 6}
```

4. 指标

通过 Prometheus 查询 `/metrics`

查询结果

```
# 接收消息总数
m3ua_stp_messages_received_total{peer_name="Partner_STP_West",point_code="1523"}

# 发送消息总数
m3ua_stp_messages_sent_total{peer_name="Local_HLR",point_code="200"}

# 路由失败次数
m3ua_stp_routing_failures_total{reason="no_route"} 5
m3ua_stp_routing_failures_total{reason="no_gt_route"} 2
```

STP 消息

消息

消息接收

- `m3ua_stp_messages_received_total` - 000000000000
 - 000 `peer_name` 0 `point_code`
- `m3ua_stp_messages_sent_total` - 000000000000
 - 000 `peer_name` 0 `point_code`

□ □ □ □ □ □ □

- `m3ua_stp_routing_failures_total` - 000000000000
 - 000 reason 000 no_route 0 no_gt_route 0

1111

- 0000000000000000
- 0000000000000000
 - no_route 000000000000
 - no_gt_route 000000000000 PC 000000

□ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □

1.

```
m3ua_stp_messages_received_total{peer_name="Source_Peer"} > 0
```

2. □□□□□□□□

```
m3ua_stp messages_sent_total{peer_name="Dest_Peer"} > 0
```

3. □□□□□□

```
m3ua stp routing failures total{reason="no route"} > 0
```

□ □

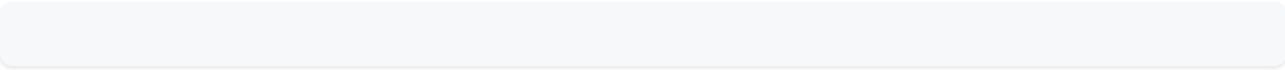
M3UA □□□□□□□□

□□ M3UA

M3UA□MTP3 □□□□□□□□□□□□ SS7 □□□□ IP □□□□ SCTP □□□□□

M3UA □□□□

M3UA □□□□□□□□



Web UI 簡介

← 目錄

OmniSS7 Web UIPhoenix LiveView 簡介

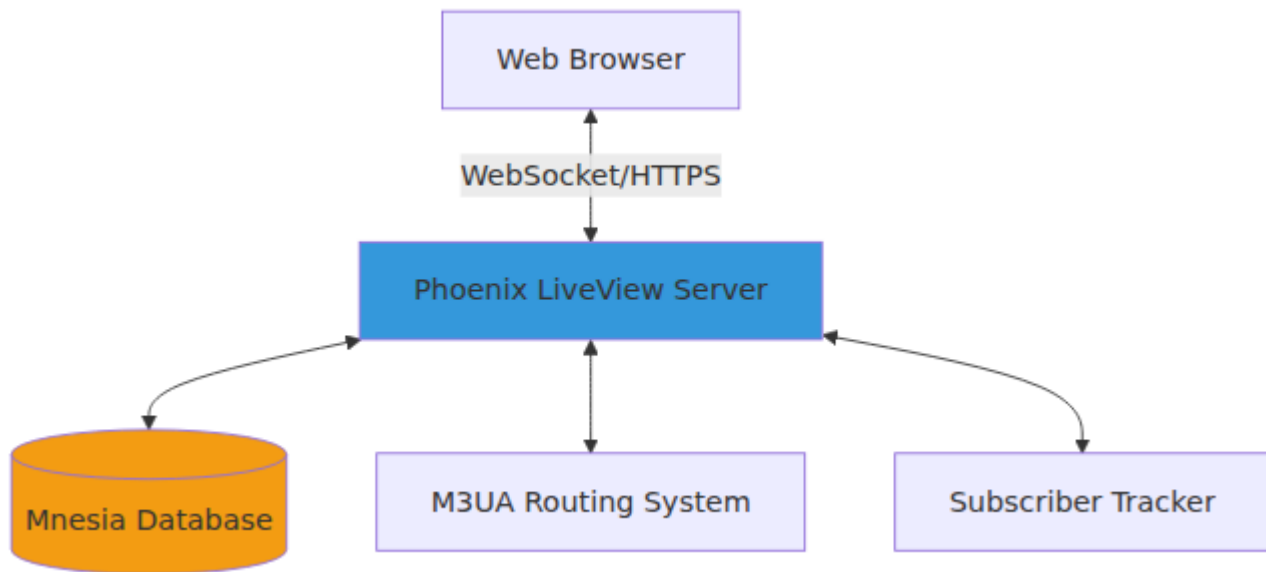
目錄

1. 簡介
2. Web UI
3. 安裝
4. 配置
5. 使用
6. 故障排除

簡介

OmniSS7 Web UI 是 **Phoenix LiveView** 的 Web 界面，用於管理 STP、HLR 和 SMS 服務。

Web UI



Configuration

- Port: HTTPS
- Port: 443 `config/runtime.exs` port
- IP: 0.0.0.0
- Path: `priv/cert/`

URL: `https://[server-ip]:443`

Web UI

Configuration

1. **SSL** Path: `priv/cert/` SSL files
 - `omnitouch.crt` - Certificate
 - `omnitouch.pem` - Private key
2. Command: `iex -S mix`
3. Port: 443 HTTPS

□□□□□□□

□□	STP □□	HLR □□	SMSc □□	□□
SS7 □□	□	□	□	□□□□ SCCP □□□□
SS7 □□□	□	□	□	□□ MAP □□□□
M3UA	□	□	□	M3UA □□□□
□□	□	□	□	M3UA □□□□□
□□□□	□	□	□	□□□□□□□
HLR □□	□	□	□	HLR API □□□□□□□
□□□□	□	□	□	□□□□□□□□□HLR□
SMSc □□	□	□	□	SMSc API □□□□□□□
SMSc □□	□	□	□	□□□□□□□SMSc□
□□□□	□	□	□	□□□□□□□
□□	□	□	□	□□□□□

□□□□

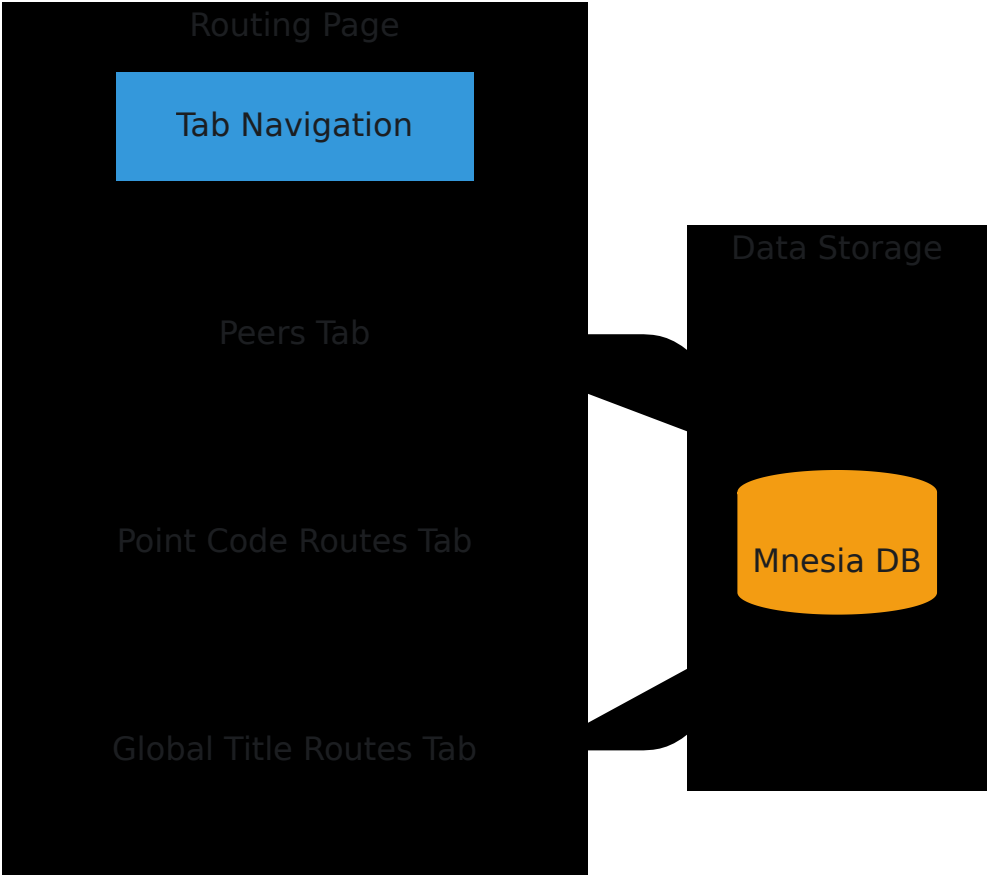
□□: /routing

□□: STP, SMSc

□□□□: □ 5 □

□□□□□□□□□□□□□□□□□□ M3UA □□□□

□□□□



Peers □□□

□□ M3UA □□□□□□ STP□HLR□MSC□SMSC□□

□□□□

項目	説明	値
ID	識別子	1
名前	名前	"STP_West"
タイプ	タイプ	client, server, stp
ポート	ポート SS7 番号	100
IP	IP アドレス	10.0.0.10:2905
ステータス	ステータス	active, aspup, down
コメント	コメント	-

手順

1. Peers 設定
2. Peer 設定
 - ID: 識別子
 - 名前: 名前
 - タイプ: client, server, stp
 - ポート: SS7 番号
 - IP: IP アドレス
 - ポート: 2905
 - ステータス: M3UA ID
 - コメント: international, national
3. "Add Peer"

注意: Mnesia 設定

手順

1. 00 000000 "Edit" 00
2. 000000 0000
3. 00 "Update Peer"

00: 0000000 ID00000000000000000000000000000000

□□□□

1. 删除字符串 "Delete" 的字符
2. 删除字符串 "Delete" 的字符

□□□□□□□

項目	単位	数値
active	個	10000000000
aspup	個	ASP 1000000000
down	個	10000000

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

□□□□□□□□□□□□

□□□□

項目	説明	値
PC	zone.area.id	1.2.3 (100)
	PC の	/14 (), /8 ()
ID		1
		"STP_West"
	1 =	1
		international
	/	-

- "Point Code Routes"
- zone.area.id 1.2.3 0-16383
 - : /14
 - ID:
 - : 1 =
 - : international national
- "Add Route"

- 3-8-3 : zone.area.id 1.2.3
- : 0-16383 1100

14 0-16383

IP	Number of PCs	Subnet
/14	1 (PCs)	
/13	2 PCs	
/8	64 PCs	
/0	16,384 PCs	

Example:

- PC 1000 /14 → PC 1000
- PC 1000 /8 → PC 1000-1063 (64 PCs)
- PC 0 /0 → All PCs

Example:

Example:

Example:

Example: SCCP

Example:

```
config :omniss7,  
  enable_gt_routing: true
```

Example:

SSN	名前
6	HLR (ホームランダム)
7	VLR (バリエーション)
8	MSC (移動交換機)
9	EIR (エラーイン)
10	AUC (アクセス)
142	RANAP
145	gsmSCF (グローバル)
146	SGSN

SSN 名前

- **SSN**: ホームランダム SSN
- **SSN**: バリエーション SSN
 - = ホーム SSN
 - = バリエーション SSN

例: SSN=6HLR バリエーション SSN=7VLR

インストール

インストール Mnesia データベース

インストール

1. **Web UI** 例: config/runtime.exs Mnesia
2. データベース: Mnesia データベース
3. **Runtime.exs** 例: config/runtime.exs Mnesia データベース

🔗🔗

📄📄📄📄📄📄

- 1. 📄📄📄: 📄📄📄📄📄📄📄
- 2. 📄📄📄: 📄📄📄📄📄📄📄1 = 📄📄📄
- 3. 📄📄: 📄📄 **active** 📄📄📄

📄📄📄

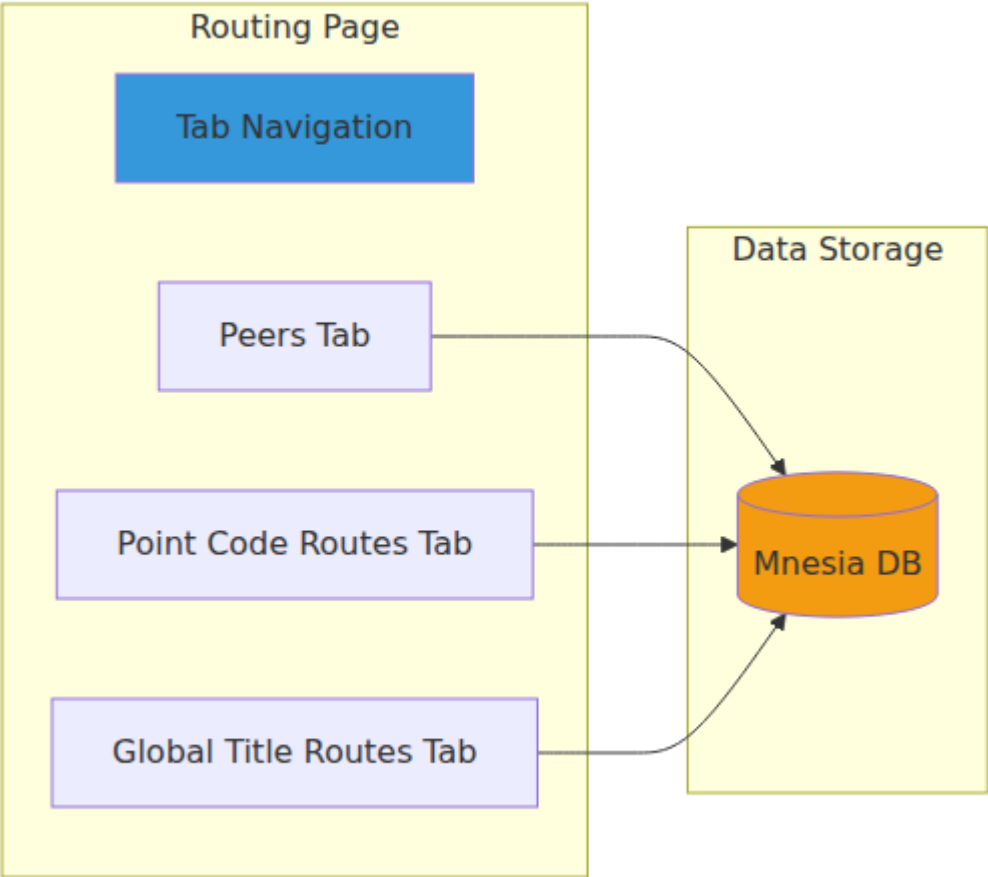
📄: /subscribers

📄: 📄 HLR

📄📄: 📄 2 📄

📄📄📄📄📄 UpdateLocation 📄📄📄📄

📄📄📄



数据表

字段	名称	值
IMSI	IMSI	"50557123456789"
VLR 标识	VLR GT 标识	"555123155"
MSC 标识	MSC GT 标识	"555123155"
时间戳	UpdateLocation 时间戳	"2025-10-25 14:23:45 UTC"
持续时间	持续时间	"2h 15m 34s"

数据表

数据表头信息

- ❓❓❓: 数据表头
- 标识 **VLRs**: VLR 标识
- 标识 **MSCs**: MSC 标识

数据表

数据表头: 数据表头

标识: 数据表头

标识: 数据表头

数据表

标识 2 数据表头

SMSc 简介

URL: /smsc_subscribers

部署: 1 SMSc

副本数: 2 个

与 HLR 交互 alertServiceCenter 实现短信服务

功能

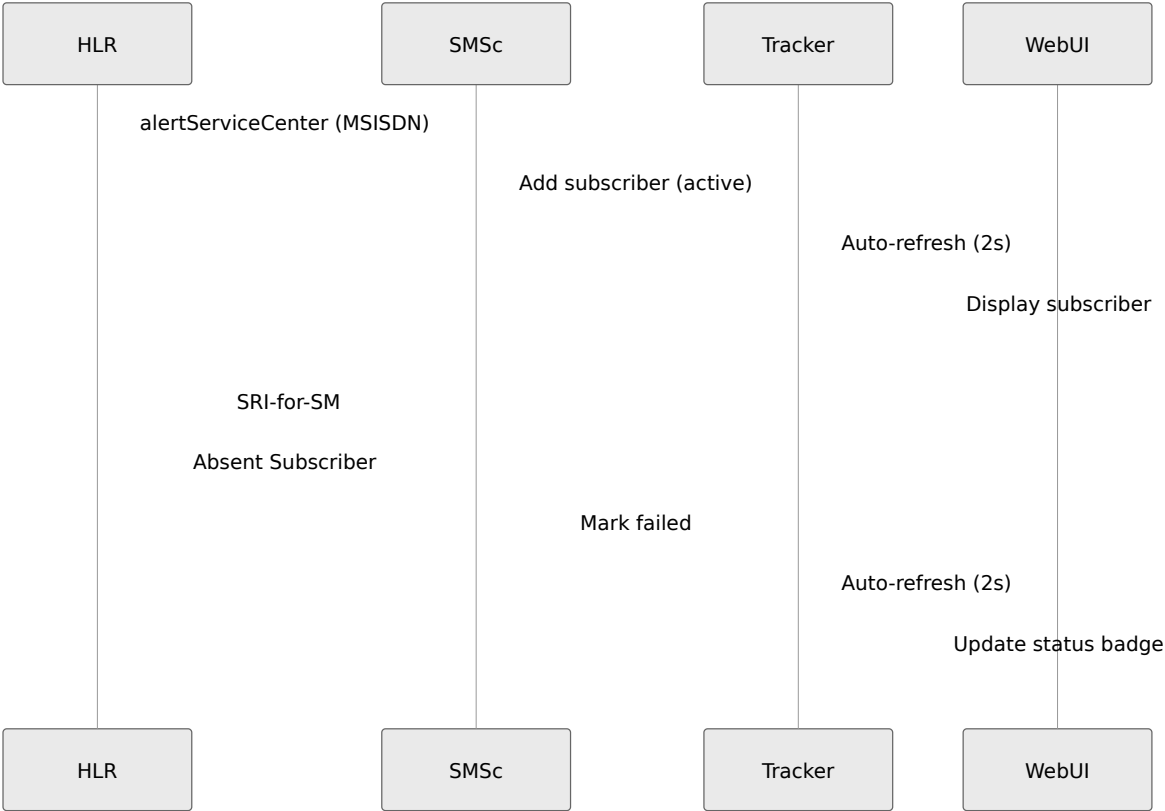


Table 1

Field	Description	Value
MSISDN	MSISDN	"15551234567"
IMSI	IMSI	"001010123456789"
HLR GT	HLR GT	"15551111111"
MT-FSM	MT-FSM	5
MO-FSM	MO-FSM	2
Status	Active or Failed	● Active
Timestamp	Timestamp	"2025-10-30 14:23:45 UTC"
Duration	Duration	"15m 34s"

Legend

- **Active** (●): alertServiceCenter
- **Failed** (○): SRI-for-SM

Table 2

Table 3

- Field: MSISDN
- Field: IMSI
- Field: HLR GT
- Field: HLRs: HLR

□□□□

□□□□: □□□□□□□□□□

□□□□□□: □□□□□□□□□□

□□: □□□□□□□□□□□□□□

□□:

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□ HLR □□□□□□

□□□□□□

□□□□□□□□□□□□

- □□□□□□: □ SRI-for-SM □□□□□ MT-FSM □□□
- □□□□□□: □□□□□□ MO-FSM □□□

□□□□

□□□ 2 □ □□□□□□□□□□□□□□□□□

□□□□

□□□□□□

□□□Web UI □□□□□□□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□ (Ctrl+F / Cmd+F)
2. □□□□□□□□□□□□ GT □□

部署

部署环境

1. 节点 1: 部署 REST API 服务
2. 节点 2: 部署 config/runtime.exs 服务
3. 节点 3: 部署 Web UI 服务

部署

部署 Web UI 服务

- 部署 Mnesia 数据库
- 部署 config/runtime.exs 服务

部署

1. **Mnesia**: 部署 Mnesia.{node_name}/ 服务
2. 部署: 部署 config/runtime.exs 服务

部署

部署环境

节点	部署	部署
节点 1	5 个	部署服务
节点 2	2 个	部署服务
M3UA 节点	部署	部署

WebSocket 部署: 部署 Phoenix LiveView WebSocket 服务

部署: 部署 WebSocket 服务

SSL

SSL 設定

1. **HTTPS** 設定: `priv/cert/omnitouch.crt` と `.pem` を指定
2. ポート **443**: HTTPS を有効にする
3. 証明書: `iex -S mix` を実行して生成
4. 証明書: SSL 証明書をインストール

データベース

1. **Mnesia** 設定: `mnesia_storage_type: :disc_copies` を指定
2. **Mnesia** 設定: Mnesia を有効にする
3. 証明書: Mnesia をインストール

WebSocket

1. **WebSocket** 設定: WebSocket を有効にする
2. 設定: WebSocket を有効にする
3. 証明書: (F5) を有効にする

API

- **STP** API - 証明書
- **HLR** API - 証明書
- **API** API - REST API
- **API** API - 証明書

UI

OmniSS7 Web UI をインストール

□ □□□□ - □□□□□□□□□□

□ □□□□ - Mnesia □□□□□□□□□□□□

□ □□□□□ **UI** - □□□□□□□□□□STP/HLR/SMSc□□□□□

□ □□□□□ - □□□□□□□□□□□□□□

□ □□□□ - □□□□□□□□□□

□□□□□□□□□□□□□□ **API** □□□

