

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 配置

配置信息

项目	配置	说明
协议	HTTP	默认
IP 地址	0.0.0.0	监听所有接口
端口	8080	默认端口
插件	Plug.Cowboy	默认

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

配置 HTTP 服务

配置 HTTP 服务

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

配置: `true`

配置: HTTP 服务 REST API/Swagger UI

Swagger UI

API 文档 **Swagger UI** API 文档

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 SM Routing Info	10s
/api/send-auth-info	POST	Send Auth Info	10s
/api/MT-forwardSM	POST	Forward SM	10s
/api/forwardSM	POST	Forward SM	10s
/api/updateLocation	POST	Update Location	10s
/api/prn	POST	Print	10s
/api/usss/send	POST	Send USSD	10s
/metrics	GET	Prometheus	N/A
/swagger	GET	Swagger UI	N/A
/swagger.json	GET	OpenAPI	N/A

API: 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 項目):

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

cURL 表:

```
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	필수	설명
msisdn	문자열	예	발신 MSISDN
service_center	문자열	예	서비스 센터 번호

상태: (200 OK):

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

cURL 실행:

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

SendAuthenticationInfo

요청 데이터

메소드: `POST /api/send-auth-info`

응답:

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

메소드:

項目	項目	項目	項目
imsi	IMSI	1	IMSI
vectors	3	1	3

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

SMS

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 is a hexadecimal string representing the SMS TPDU.

Response (200 OK):

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

cURL Request:

```
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 SMS 消息 MO-SMS

请求: POST /api/forwardSM

响应: MT-ForwardSM

cURL 请求:

```
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 位置更新 VLR

请求: POST /api/updateLocation

响应:

```
{
  "imsi": "001001234567890",
  "vlr": "5551234800"
}
```

□□:

□□	□□	□□	□□
imsi	□□□	□	□□ IMSI
vlr	□□□	□	VLR □□  □□□

□□ (200 OK):

```
{
  "result": {
    "hlr_number": "5551234567",
    "subscriber_data": {...},
    ...
  }
}
```

□□: □ HLR □□□□□□□□□□ 10 □□□□ InsertSubscriberData (ISD) □□□

cURL □□:

```
curl -X POST http://localhost:8080/api/updateLocation \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "vlr": "5551234800"
}'
```

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",
  "gsmc": "5551234567",
  "msc_number": "5551234800",
  "imsi": "001001234567890"
}'
```

USSD

USSD `ussd_gateway_enabled: true` USSD `USSD`

POST `/api/ussd/send`

:

```
{
  "msisdn": "+254712345678",
  "text": "1 ",
  "callback_url": "http://billing-app:9000/ussd"
}
```

:

<code>msisdn</code>			MSISDN
<code>text</code>			USSD GSM 7
<code>callback_url</code>			URL HTTP POST

(200 OK):

```
{
  "session_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "status": "sent"
}
```

503 (503 status):

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

cURL command:

```
curl -X POST http://localhost:8080/api/usss/send \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "+254712345678",
  "text": "USSD service 1 minute",
  "callback_url": "http://billing-app:9000/usss"
}'
```

Notes

503: API Unavailable

Notes:

- API Unavailable/503
 - USSD service timeout
 - USSD service unavailable
-

□□□□

□□□□□ **JSON** □□□

□□□□

HTTP □□: 200 OK

□□:

```
{
  "result": {
    // □□□□□□□□
  }
}
```

□□□□

HTTP □□:

- 400 □□□□ - □□□□□ JSON
- 504 □□□□ - MAP □□□□□10 □□
- 404 □□□ - □□□□

□□:

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

□

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

API

API

API	HTTP	API	API
JSON	400	JSON	JSON
	400		
	504	MAP 10	M3UA HLR/VLR
	404		URL

API

MAP 10

- MapClient GenServer
- 10
- 504
- 200 OK

:

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

(Prometheus)

API Prometheus

□□□□

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

□□: Prometheus □□□□

□□□□:

```
# HELP map_requests_total □ MAP □□□
# 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 □ CAP □□□
# TYPE cap_requests_total counter
cap_requests_total{operation="initialDP"} 87
cap_requests_total{operation="requestReportBCSMEEvent"} 91

# HELP map_request_duration_milliseconds MAP □□/□□□□□□□□□□
# 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 □□□□ MAP TID □□□□□
# TYPE map_pending_requests gauge
map_pending_requests 3

# HELP omniss7_license_status □□□□□□□□1 = □□□0 = □□□
# TYPE omniss7_license_status gauge
omniss7_license_status 1
```

項目

項目	単位	種別	説明
map_requests_total	回 分	operation	MAP 操作の総回数を示す
cap_requests_total	回 分	operation	CAP 操作の総回数を示す
map_request_duration_milliseconds	ミリ秒 分	operation	MAP 操作の平均実行時間
map_pending_requests	回	-	現在 MAP 操作中の回数
ussd_requests_total	回 分	direction	USSD 操作の総回数を示す
ussd_active_sessions	回	-	現在 USSD 操作中の回数
omniss7_license_status	回	-	ライセンスステータス 1 = 有効 0 = 無効

Prometheus 設定

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

sendSRI();
```

Bash/cURL ☐☐

```
#!/bin/bash

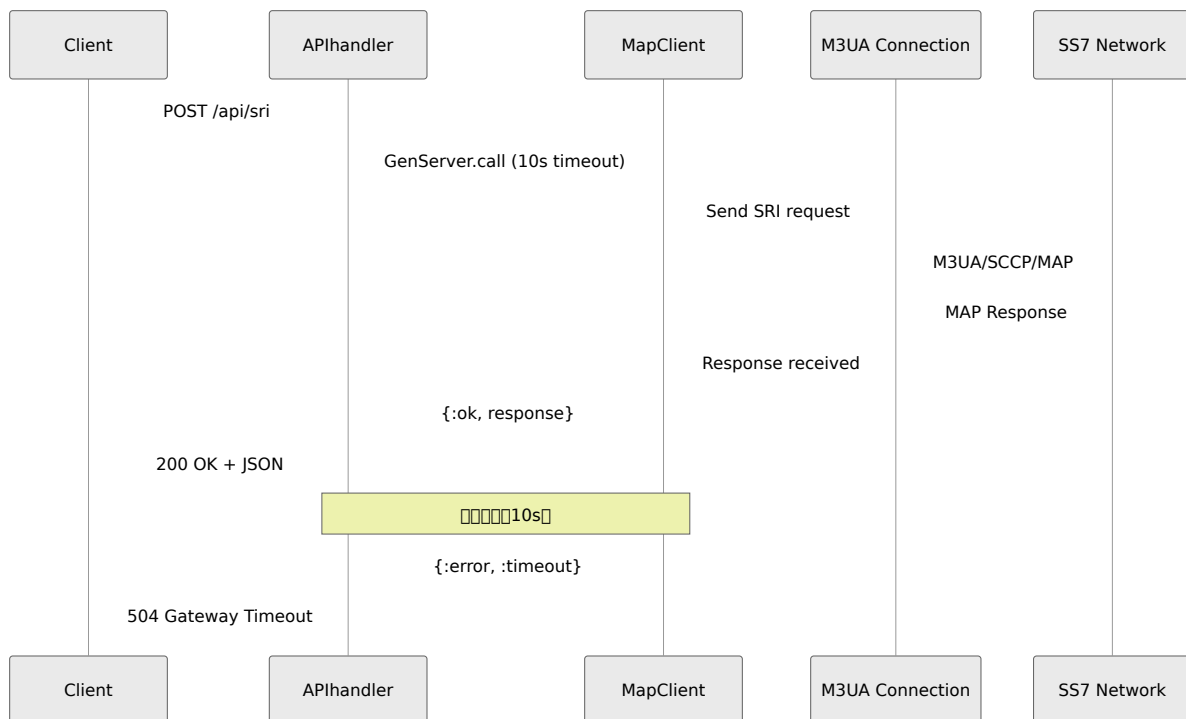
# UpdateLocation ☐☐
response=$(curl -s -w "\n%{http_code}" -X POST
http://localhost:8080/api/updateLocation \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "vlr": "5551234800"
}')

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

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



API 测试



OmniSS7 REST API 测试

MAP 测试 - 测试 SRI/SRI-for-SM/UpdateLocation/SMS 测试

Swagger UI - 测试 API 测试

Prometheus 测试 - 测试测试

测试测试 - 测试 MAP 测试 10 测试

HTTP 测试 - 测试 8080 测试测试 `start_http_server` 测试

Web UI 测试测试 Web UI 测试

测试测试测试测试 测试测试

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 GSM 短信 SMS 160 字节
 - **UCS2** - 16 Unicode 短信 SMS 70 字节
 - **8-bit** - 8 位短信 SMS 140 字节
-

M3UA

- **DOWN** - SCTP 故障
 - **CONNECTING** - SCTP 连接
 - **ASPUP_SENT** - ASP 启动
 - **INACTIVE** - ASP 空闲
 - **ASPAC_SENT** - ASP 激活
 - **ACTIVE** - ASP 活动
-

SS7

SS7 14 ITU 24 ANSI

SS7 ITU

- 3 字节
 - 8 字节
 - 3 字节
-

SCCP

- **0** - 默认
- **1** - 默认
- **2** - 默认
- **3** - 默认

- **4** - 00000
- **5** - MTP00
- **6** - 0000
- **7** - 000
- **8** - 000000

MAP0000

00	00	00
1	unknownSubscriber	0000HLR0
27	absentSubscriber	00000
34	systemFailure	0000
35	dataMissing	0000000
36	unexpectedDataValue	00000

0000

- ← 00000
- STP00
- MAP00000
- 000000
- HLR00
- 0000

CAMEL ☐☐☐☐☐☐

CAMEL 00 (CAMEL00) 0000 OmniSS7 0000000000 (IN) 00000000 CAMEL 0000
(CAP) 0000000000000000

□□□ **CAMEL**□

CAMEL (GSM) GSM UMTS

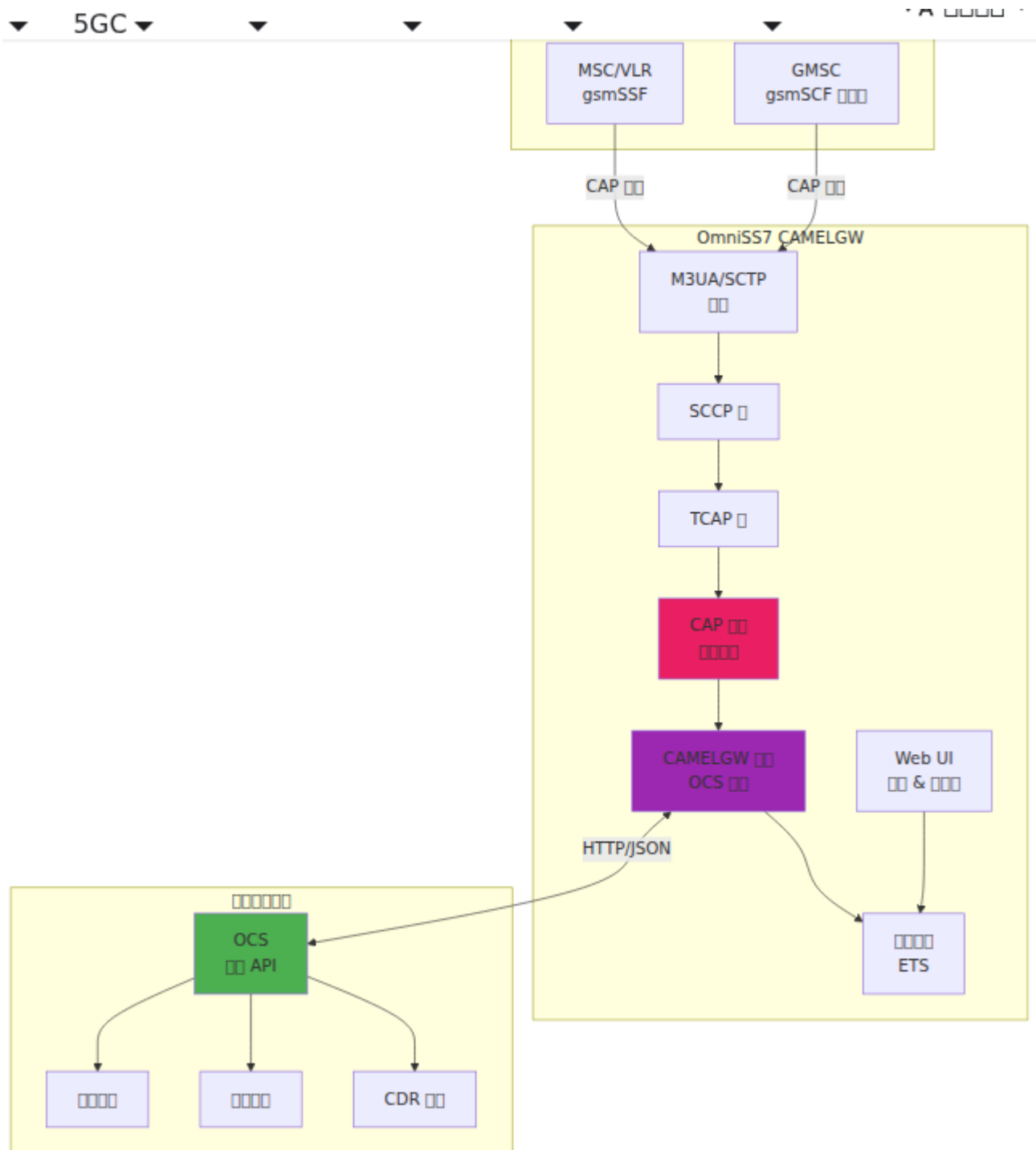
- 000000 - 0000000000
- 0000 - 0000000000
- 000000 - 0000/0000000000
- 000000 - 000000
- 0000 - 000000/0000

CAP 简介

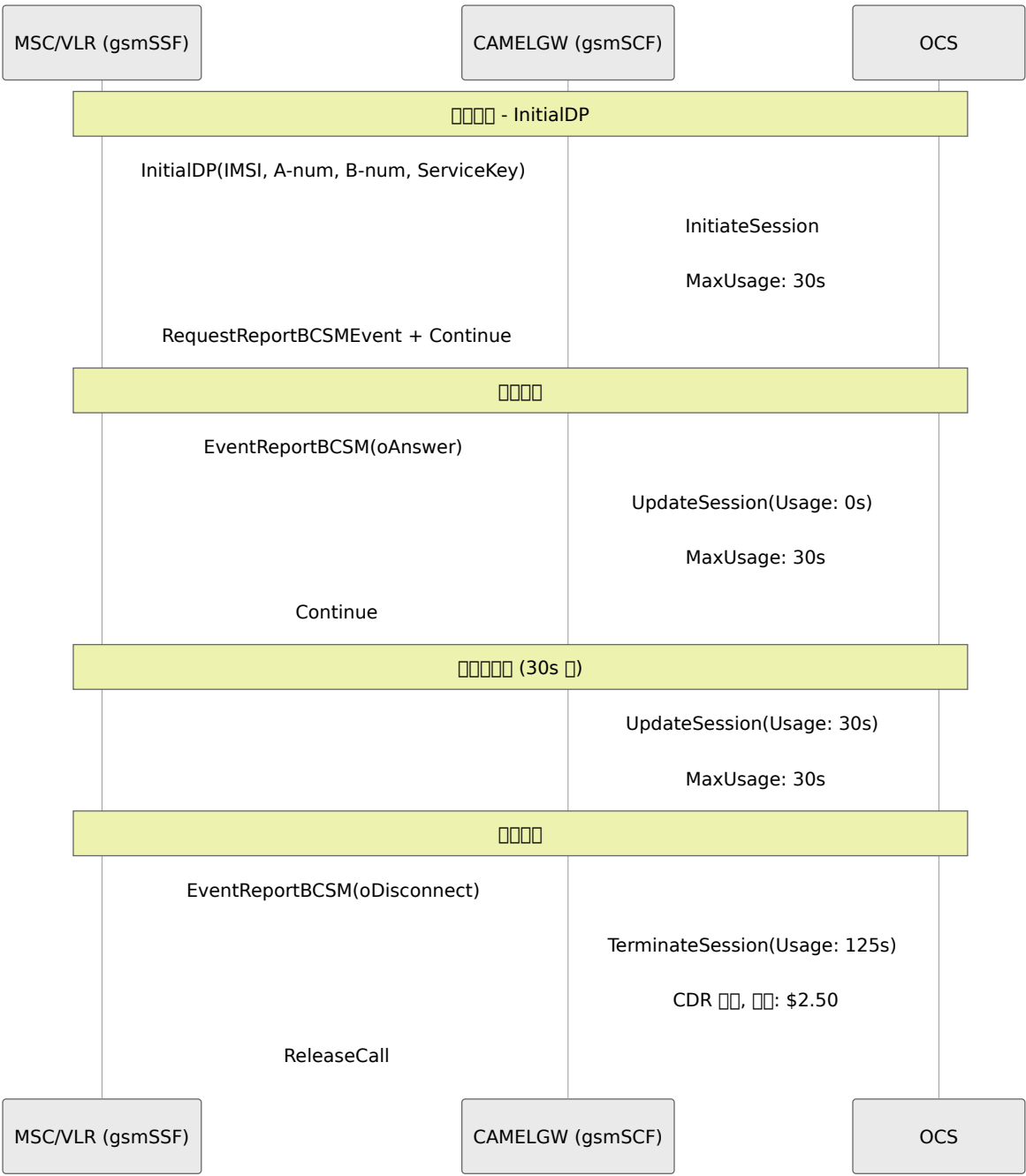
OmniSS7 CAMELGW 简介 CAP 简介

名称	简介	功能
CAP v1	CAMEL 1	呼叫转移
CAP v2	CAMEL 2	呼叫转移 SMS
CAP v3	CAMEL 3	呼叫 GPRS
CAP v4	CAMEL 4	呼叫转移

简介 CAP v2



□□□□□□



□□

□□□□

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

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

`CAMEL`

`config/runtime.exe` `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, # 配置
  ocs_auth_token: "your-api-token" # 配置 OCS 配置

  # 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"]
```

CAP

gsmSSF → gsmSCF

消息	消息ID	消息名称	处理函数
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/1
AssistRequestInstructions	16	AssistRequestInstructions gsmSRF AssistRequestInstructions	handle_assist_request_instructions/1

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 項目項目

URL: `http://localhost/camel_sessions`

項目項目項目 CAMEL 項目項目

項目

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

項目項目

- ID - IMSI - OTID

InitialDP

- InitialDP
- -
- CDR

CAP InitialDP CAP MSC CAP

CAMEL

URL: http://localhost/camel_request

CAP

InitialDP

- InitialDP
- -
- SCCP/M3UA
- 20
- OTID
- -

InitialDP

1. InitialDP -

-
- A
- B

2. Connect -

-

3. ReleaseCall -

- 16=17=31=

4. **RequestReportBCSMEvent** -

- 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) `ms` `CAP` `ms`
6. **MSC** - `RequestReportBCSMEvent` + `Continue` `ms` `CAP` `ms`

msc

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

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

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

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

msc

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

3. `msc`

0000000000000000

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

MaxUsage 0 → ReleaseCall

4. EventReportBCSM - oDisconnect

000000

1. oDisconnect - MSC
2. -
3. OCS - TerminateSession API
4. CDR - OCS
5. - SessionStore
6. ReleaseCall - MSC

CDR

CDR OCS

CAMEL CDR

- Account - IMSI
 - Destination -
 - OriginID - (CAMEL-OTID)
 - Usage -
 - Cost -
 - IMSI - IMSI
 - CallingPartyNumber - A
 - CalledPartyNumber - B
 - MSCAddress - MSC
 - ServiceKey - CAMEL
-

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"

0000 MSC 0000

00 MSC CAMEL 00

000 MSC/VLR 0000 CAMEL 000

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

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

0000

00 CAMELGW 00000000 CAP 000

```
# 000000
tail -f /var/log/omniss7/omniss7.log

# 00 CAP 00
grep "CAP:" /var/log/omniss7/omniss7.log

# 00000000JSON 000
curl http://localhost/api/events | jq '.[ ] | select(.map_event | startswith("CAP:"))'
```

0000

00000000000000000000

```
# 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]
```

準備

準備環境 **CAP** 環境

環境 準備環境環境環境 MSC 環境 InitialDP

環境

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

環境環境

```
# 環境 M3UA 環境
tcpdump -i eth0 sctp

# 環境 MSC 環境環境 CAMELGW
ss -tuln | grep 2905
```

環境**OCS** 環境

環境 `INSUFFICIENT_CREDIT` 環境環境

環境

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

環境環境

- 環境 `runtime.exe` 環境 OCS URL 環境
- 環境 OCS 環境環境環境

- 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 简介

OmniSS7 CAP 简介

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

OmniSS7

CAMEL 简介 OmniSS7 简介

- OmniSS7 CAP 简介 (v1/v2/v3/v4)
- OmniSS7 OCS 简介
- OmniSS7 (Connect, Release, Continue)
- OmniSS7 ETS 简介
- OmniSS7 Web UI 简介
- OmniSS7 简介
- OmniSS7 CDR 简介
- OmniSS7 简介

OmniSS7

- CAMEL 简介
- OmniSS7 - CAP 简介

OmniSS7 CAMEL 简介

OmniSS7 1.0

OmniSS7 2025-10-26

CAMEL 開発者 - 開発者

開発者

開発者 LiveView 開発者 CAMEL/CAP 開発者 UI開発者
InitialDP 開発者 CAMEL 開発者

開発者

1. CAMEL 開発者 LiveView

開発者

- 開発者 UI開発者 CAMEL 開発者
- 開発者
 - InitialDP** - 開発者
 - Connect** - 開発者

- **ReleaseCall** - 00/0000
- **RequestReportBCSMEEvent** - 000000
- **Continue** - 000000
- **ApplyCharging** - 00000000/000000

000000

- 0000000000
- 0000000000000000
- 00 SCCP/M3UA 0000000000
 - 00/00000000
 - SSN0000000000
 - OPC/DPC000000
- 0000000000 20 0000
- 00 OTID 000000
- 00/00000000
- 000000

0000 /camel_request

2. 0000 EventLog 0000 CAMEL

0000

- paklog_camel/2 - 0000 CAMEL/CAP 000000
- lookup_cap_opcode_name/1 - CAP 000000
- find_cap_opcode/1 - 0 JSON 0000 CAP 0000
- extract_cap_tids/1 - 0 CAP 000000 OTID/DTID
- format_cap_to_json/1 - 0 CAP PDU 0000 JSON 00

0000 CAP 0000

```

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 []"},
    {SS7.Web.TestClientLive, "/client", "SS7 [][]"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {SS7.Web.HlrLinksLive, "/hlr_links", "HLR []"},
    {SS7.Web.CAMELSessionsLive, "/camel_sessions", "CAMEL []"},
    {SS7.Web.CAMELRequestLive, "/camel_request", "CAMEL [][][]"}
  ],
  page_order: ["/events", "/client", "/m3ua", "/hlr_links",
    "/camel_sessions", "/camel_request",
    "/application", "/configuration"]
```

[]

[][][][][]

1. [][] https://your-server:8087/camel_request
2. [][][][][][]
3. [][][]
4. [][][]“[] SCCP/M3UA []”[][][]
5. []“[] [RequestType] []”

[][][]

InitialDP[][][]

1. [][][][][][]100
2. [][][][]A-Party
3. [][][][]B-Party
4. [][] → [][] OTID
5. OTID [][][][][][]

[][][] **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 □□□□□□

- 前4文字 "CAP:" 後8文字

ApplyCharging -

11

ApplyCharging

11

- [illegible]

11

maxCallPeriodDuration

- 1-864000 00
- 000000000000000000000000
- 00
 - 60 = 1 00
 - 290 = 4 00 50 00000000
 - 3600 = 1 00
 - 86400 = 24 00

releaselfDurationExceeded

- `Boolean` `true/false`
- `Boolean` `true`
- `Boolean` `Boolean`
 - `true` `Boolean`/`Boolean`
 - `false` `Boolean` `gsmSCF` `Boolean`

□□□□

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 **RequestReportBCSMEvent** 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 □□

- 用户 UI 交互流程图

目次

← 概要

OmniSS7 概要

目次

1. Web UI
2. API
3. 設定
4. 操作
5. 拡張機能 SFTP

Web UI

Web UI 概要 Web 操作

API

- API - API SS7 API
- API - API
- API - API
- **M3UA** API - M3UA API STP API
- **SMS** API - API SMS API SMSc API

API Web UI

1. API Web API
2. API API `http://localhost`
3. API 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	デフォルト値	説明
page_order	配列	<code>["/events", "/application", "/configuration"]</code>	ページの順序
listen_ip	文字列	<code>"0.0.0.0"</code>	Web サービスの IP アドレス
port	整数	<code>80</code>	HTTP サービスのポート番号 (443 は HTTPS のポート番号)
hostname	文字列	<code>"localhost"</code>	URL のホスト名
enable_tls	ブール値	<code>false</code>	TLS を有効にするかどうか (HTTPS には必要)
tls_cert	文字列	<code>"cert.pem"</code>	TLS 証明書ファイルのパス
tls_key	文字列	<code>"key.pem"</code>	TLS 私鍵ファイルのパス

ログ設定

config/runtime.exs を編集する

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

ログレベル

- `:debug` - デバッグログ
 - `:info` - インフォログ
 - `:warning` - ワARNINGログ
 - `:error` - エラーログ
-

API 設定

API 設定 URL

```
http://your-server/api
```

ステータスコード

- **200** - OK
- **400** - 不正なリクエスト
- **504** - タイムアウト

OpenAPI 設定

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

監視設定

Prometheus 設定

```
http://your-server/metrics
```

□□□□□□

M3UA/SCTP □□□

- SCTP □□□□□□
- M3UA ASP □□□□
- □□/□□□□□□□□□□

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.

- 設定
- 設定
- 設定

3. M3UA

- SCTP
-
-

SCTP

SCTP

SCTP は SCTP 対応の M3UA 対応の IP 対応の

- SCTP
- - SCTP
- NIC
- SCTP

M3UA IP SCTP

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

##

SCTP **IP** **IP** **IP**

IP

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

IP

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

###

role: :server **remote_ip**
- SCTP SCTP INIT IP

STP SCTP INIT IP SCTP IP
remote_ip **role: :client**

```
# ✓ [] - [] remote_ip
%#123;
    role: :server,
    remote_ip: [#123;10, 0, 2, 100#125;, # []
    # ...
#125;

# x [] - []
%#123;
    role: :server,
    remote_ip: [#123;10, 0, 2, 100#125;, [#123;10, 0, 2,
101#125;], # []
    # ...
#125;
```

[] 1 [] STP [] - []

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

[] 2 [] MAP []

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

配置 3 个 STP 节点

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

配置 4 个节点 - 配置网络接口

```

# HLR
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "HLR",
      role: :server, # HLR
      # IP
      local_ip: [{10, 0, 1, 10}, {10, 0, 1, 11}],
      local_port: 2905,
      # HLR SCTP INIT IP
      # SCTP HLR IP
      remote_ip: {10, 0, 2, 100}, # IP
      remote_port: 0, #
      routing_context: 1,
      point_code: 100,
      network_indicator: :international
    }
  ]

```

5

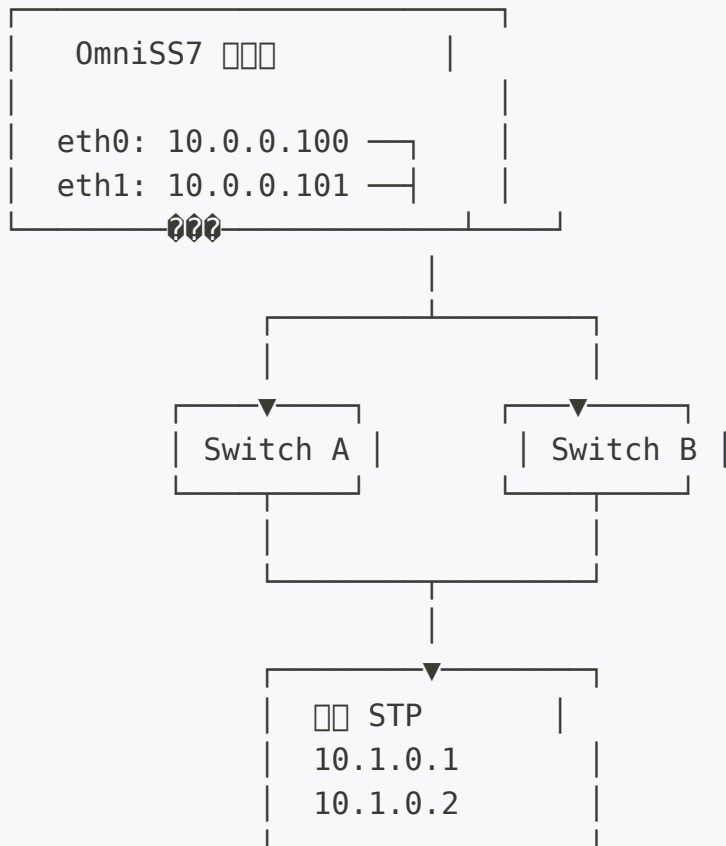
```

# 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
    %{
      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



Host

```

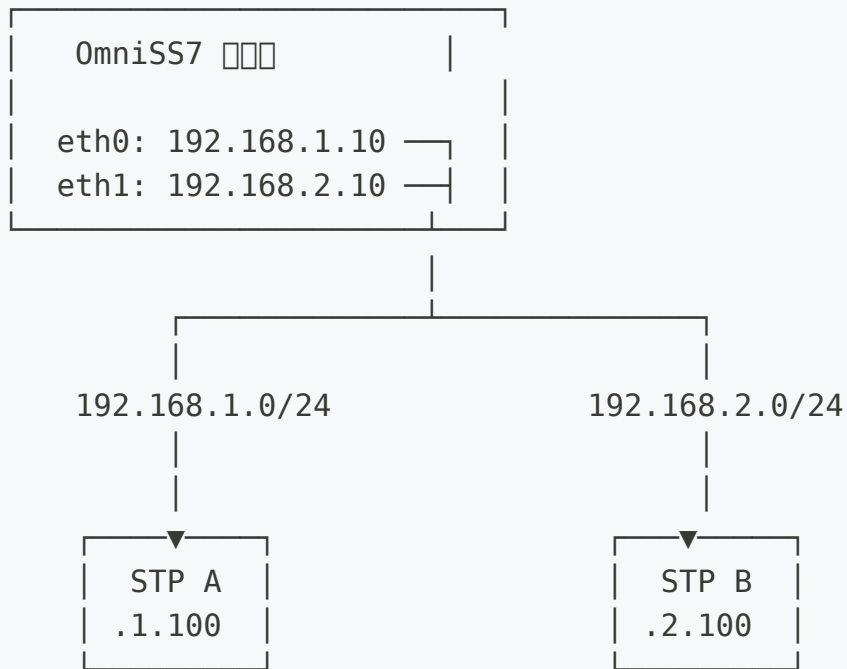
local_ip: [{10, 0, 0, 100}, {10, 0, 0, 101}] # Host NIC
remote_ip: [{10, 1, 0, 1}, {10, 1, 0, 2}] # Switches

```

Switches

- Host NICs
- Switches
- Switch 1

2 Switches



□□□

```

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 -
- IP
-

3.

```
# 断网
sudo ip link set eth0 down

# 查看断网原因
tail -f /var/log/omniss7.log | grep MULTIHOMING

# 恢复网络
sudo ip link set eth0 up
```

4. 配置网络

- 配置网络接口 IP
- 配置网络接口子网掩码
- 配置网络接口网关

5. 配置防火墙

```
# 配置防火墙规则 IP 地址 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
```

测试

配置网络接口

配置网络接口 IP 地址

测试

1. 测试 Erlang SCTP 配置 `erl -eval 'gen_sctp:open(9999, [binary, {ip, {127,0,0,1}}]).'`
2. 查看 SCTP 模块 `lsmod | grep sctp`
3. 加载 SCTP 模块 `sudo modprobe sctp`
4. 查看网络接口 IP 地址 `ip addr show`

配置网络接口

配置网络接口子网掩码

問題

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 IP の UP と DOWN の経緯

問題

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 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%

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

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

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

□□□□ (: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		<code>["/events", "/application", "/configuration"]</code>	ページ順序
web.listen_ip	文字列		<code>"0.0.0.0"</code>	Web サービス IP アドレス
web.port	整数		<code>80</code>	HTTP/HTTPS ポート
web.hostname	文字列		<code>"localhost"</code>	ホスト名
web.enable_tls	ブール値		<code>false</code>	HTTPS を有効にする
web.tls_cert	文字列	有効な TLS	<code>"cert.pem"</code>	TLS 証明書
web.tls_key	文字列	有効な TLS	<code>"key.pem"</code>	TLS キー

SCTP SocketHandler (:omniss7)

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

項目	項目	値	説明
sctp_handler.enabled	項目	false	STP 有効
sctp_handler.local_ip	項目	{127, 0, 0, 1}	M3UA IP
sctp_handler.local_port	項目	2905	M3UA Sctp ポート
enable_gt_routing	項目	false	

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 10.0.0.0	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 10.0.0.0	文字列	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	○	宛先ポート番号
peer_id	整数	○	対話相手の識別子
priority	整数	○	優先度 (0: 低, 1: 高)
network_indicator	整数	○	:international 国際ネットワーク, :national 国内ネットワーク

M3UA GT 項目

項目	型	必須	説明
gt_prefix	文字列	○	GTプレフィックス
peer_id	整数	○	対話相手の識別子
priority	整数	○	優先度 (0: 低, 1: 高)
description	文字列	○	GTの説明
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	デフォルト値	説明
<code>map_client_enabled</code>	boolean	<code>false</code>	MAP クライアントの有効化フラグ
<code>map_client_m3ua.mode</code>	enum	<code>"ASP"</code>	M3UA モード。ASP (Application Specific Part) または SGP (Signaling Gateway Part) を指定します。
<code>map_client_m3ua.callback</code>	function	<code>{MapClient, :handle_payload, []}</code>	M3UA パayload のハンドリングに使用するコールバック関数
<code>map_client_m3ua.process_name</code>	string	<code>:map_client_asp</code>	プロセス名
<code>map_client_m3ua.local_ip</code>	string	<code>-</code>	ローカル IP アドレス
<code>map_client_m3ua.local_port</code>	integer	<code>2905</code>	ローカル SCTP ポート番号
<code>map_client_m3ua.remote_ip</code>	string	<code>-</code>	リモート STP/SGP の IP アドレス
<code>map_client_m3ua.remote_port</code>	integer	<code>2905</code>	リモート SCTP ポート番号
<code>map_client_m3ua.routing_context</code>	string	<code>-</code>	M3UA ルーティングコンテキスト

SMS 配置 (:omniss7)

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

配置项	类型	默认值	说明
auto_flush_enabled	布尔	false	是否开启 SMS 自动推送
auto_flush_interval	整数	10000	自动推送间隔 (毫秒)
auto_flush_dest_smsc	字符串/nil	nil	推送 SMS 的 SMSC 地址 nil = 默认
auto_flush_tps	整数	10	每秒推送条数

HTTP API 配置 (:omniss7)

SMS 推送使用 HTTP API 推送

```
config :omniss7,
  smsc_api_base_url: "https://10.5.198.200:8443",
  frontend_name: "omni-smsc01" # 推送 SMS 的 hostname_SMS
```

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	SMS 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
- PUT /api/messages/{id} - 項目項目
- POST /api/events - 項目項目
- GET /api/status - 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SMSc
- 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 SSL 項目
- HTTP 項目 5 項目
- 項目項目 ISO 8601 項目
- API 項目 JSON 項目項目/項目項目

□□□□

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

OmniSS7 □ Omnitouch □□□□□

OmniSS7

← 00000

OmniSS7 0000000000

00

1. 00
2. 000000
3. HLR 0000
4. SMSc 0000
5. STP 0000
6. CAMEL 000000
7. 0000 NAT 00
8. M3UA 0000
9. 000000
10. 00000
11. 0000

00

OmniSS7 0000 config/runtime.exs 0000000000000000

- **STP** 00 - 0000000000
- **HLR** 00 - 0000000000000000
- **SMSc** 00 - 0000000000
- **CAMEL GW** 00 - CAMEL 0000000000

0000: config/runtime.exs

Configuration

Configuration

Item	Item	Value	Item	Item
map_client_enabled	map_client_enabled	false	map MAP protocol M3UA	map
hlr_mode_enabled	hlr_mode_enabled	false	hlr HLR	HLR
smsc_mode_enabled	smsc_mode_enabled	false	smsc SMSc	SMSc
cap_client_enabled	cap_client_enabled	false	cap CAP CAMEL	CAMEL GW
camelgw_mode_enabled	camelgw_mode_enabled	false	camelgw CAMEL	CAMEL GW
ussd_gateway_enabled	ussd_gateway_enabled	false	ussd_gateway USSD (HTTP/JSON)	USSD GW

Example:

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

HLR Configuration

HLR Configuration

HLR API

Property	Type	Default	Required	Description
hlr_api_base_url	String	-	Yes	HLR API URL
hlr_api_verify_ssl	Boolean	false	Yes	Whether to verify SSL certificate for HLR API
hlr_service_center_gt_address	String	-	Yes	UpdateLocation HLR GT address
smsc_service_center_gt_address	String	-	Yes	SRI-for-SM HLR GT address

Example:

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

AlertServiceCenter

UpdateLocation HLR GT address alertServiceCenter GT address

項目	単位	初期値	単位	説明
hlr_smsc_alert_gts	分	[]	分	alertServiceCenter SMS Sc からの送信
hlr_alert_location_expiry_seconds	秒	172800	秒	48 時間 0 分 0 秒

例:

```
config :omniss7,  
  hlr_smsc_alert_gts: [  
    "15559876543",  
    "15559876544"  
  ],  
  hlr_alert_location_expiry_seconds: 172800 # 48 時間
```

MSISDN ↔ IMSI 変換

MSISDN から IMSI を取得する HLR 変換 MSISDN ↔ IMSI

項目	単位	値	注	説明
hlr_imsi_plmn_prefix	桁 桁 桁	"50557"	桁	桁 IMSI 桁桁 PLMN 桁桁 (MCC+MNC)
hlr_msisdn_country_code	桁 桁 桁	"61"	桁	IMSI→MSISDN 桁桁桁桁桁桁 桁桁
hlr_msisdn_nsn_offset	桁 桁	0	桁	MSISDN 桁 NSN 桁桁桁桁桁桁 桁桁桁桁桁桁桁桁
hlr_msisdn_nsn_length	桁 桁	9	桁	桁 MSISDN 桁桁桁桁桁桁桁桁桁桁 桁

桁桁2 桁桁桁桁桁:

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "99",      # 桁桁 2 桁桁桁桁
  hlr_msisdn_nsn_offset: 2,           # 桁桁 2 桁桁桁桁
  hlr_msisdn_nsn_length: 9            # 桁桁 9 桁 NSN
```

桁桁3 桁桁桁桁桁:

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "999",     # 桁桁 3 桁桁桁桁
  hlr_msisdn_nsn_offset: 3,           # 桁桁 3 桁桁桁桁
  hlr_msisdn_nsn_length: 8            # 桁桁 8 桁 NSN
```

桁桁: 桁 nsn_offset 桁桁桁桁桁桁桁桁桁桁桁桁桁桁 NSN桁桁桁桁

- 桁桁桁桁 "9"桁1 桁桁→ nsn_offset: 1
- 桁桁桁桁 "99"桁2 桁桁→ nsn_offset: 2

- 号码 "999" 3 位 → `nsn_offset: 3`

InsertSubscriberData (ISD) 配置

通过 UpdateLocation 接口向 VLR 下发 ISD 配置信息。HLR 下发 InsertSubscriberData 配置。

配置项	数据类型	配置值	生效范围	说明
<code>isd_network_access_mode</code>	枚举	<code>:packetAndCircuit</code>	全网	网络接入模式 :packetAndCircuit: 包电路 :packetOnly: 包 :circuitOnly: 电路
<code>isd_send_ss_data</code>	布尔	<code>true</code>	全网	是否下发 ISD #2
<code>isd_send_call_barring</code>	布尔	<code>true</code>	全网	是否下发 ISD #3

配置:

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

CAMEL 配置

通过 CAMEL 接口下发 CAMEL 配置信息。HLR 下发 CAMEL 配置。

項目	単位	初期値	単位	説明
camel_service_key	整数	11_110	整数	サービスキー
camel_trigger_detection_point	文字列	:termAttemptAuthorized	文字列	CAM :termAttemptAuthorized :termAttemptAuthorized :termAttemptAuthorized
camel_gsmcf_gt_address	文字列	(GSMCF GT)	文字列	GSMCF GT CAM GT

例:

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

例 VLR 例

例 VLR 例 PRN 例 HLR 例

項目	単位	初期値	単位	説明
home_vlr_prefixes	文字列	["5551231"]	文字列	例 VLR GT 例

例:

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

SMSc 配置

配置项说明

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 接口 URL
<code>smsc_api_verify_ssl</code>	布尔值	false	否	是否验证 SMSc API 的 SSL 证书
<code>smsc_name</code>	字符串	"{hostname}_SMSc"	否	SMSc 名称
<code>smsc_service_center_gt_address</code>	字符串	-	否	短信服务中心 GT 地址

示例:

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

注意: 配置项 5 在 SMS.FrontendRegistry 中配置

配置项

配置项	类型	默认值	必填	说明
auto_flush_enabled	布尔	true	否	是否开启 SMS 自动刷新
auto_flush_interval	整数	10_000	否	自动刷新的间隔时间(毫秒)
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 配置项在 STP 配置中配置

STP

sctp_handler.enabled		false		SCTP SocketHandler
sctp_handler.local_ip		{127, 0, 0, 1}		IP IP: {10, 0, 0, 1} IP SCTP [{10, 0, 0, 1}, {10, 0, 0, 2}]
sctp_handler.local_port		2905		
sctp_handler.point_code		-		STP SS7

IP:

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

SCTP

```
config :omniss7,
  sctp_handler: %{
    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 地址都必须在配置文件中指定 **SCTP** 地址

配置项

名称	类型	默认值	是否必填	说明
<code>enable_gt_routing</code>	布尔	<code>false</code>	否	是否 PC 支持 GT 路由

注意:

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

M3UA/M2PA 配置

配置项 `m3ua_peers` 用于配置 M3UA 和 M2PA 地址簿 **STP** 地址簿 **M2PA** 地址簿

配置项

項目	型	説明	注	備考
peer_id	整数	-	○	ピアID
name	文字列	-	○	名前
protocol	文字列	:m3ua	○	プロトコル :m3ua または :m2pa
role	文字列	:client	○	役割 :client, :server, :asp, :sgp
local_ip	文字列	-	○	ローカル IP アドレス
local_port	整数	-	○	ローカル Sctp ポート M3UA: 2905, M2PA: 3565
remote_ip	文字列	-	○	リモート IP アドレス
remote_port	整数	-	○	リモート Sctp ポート
routing_context	文字列	-	○	M3UA ルーティングコンテキスト M3UA

項目	型	初期値	必須	説明
point_code	整数	-	必須	ポイントコード
network_indicator	文字列	:international	必須	ネットワークインジケータ :international 国際 :national 国内
initiate_connection	ブール値	true	必須	SCTP を開始するかどうか

M2PA 設定

項目	型	初期値	必須	説明
adjacent_point_code	整数	-	必須	隣接する M2PA のポイントコード

注意

M2PA を開始する SctpSocketHandler は M2PA を開始する SctpSocketHandler の隣接する SctpSocketHandler を M2PA 設定で指定する

隣接する M3UA 設定:

```

config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "HLR_East",
      protocol: :m3ua,
      role: :sgp,
      local_ip: {10, 179, 4, 10},
      local_port: 2905,
      remote_ip: {10, 179, 4, 20},
      remote_port: 2905,
      point_code: 100,
      network_indicator: :international
    }
  ]

```

配置M2PA 配置:

```

config :omniss7,
  sctp_handler: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 3565,
    point_code: 100
  },
  m3ua_peers: [
    %{
      peer_id: 2,
      name: "M2PA_Link_STP_West",
      protocol: :m2pa,
      role: :client,
      local_ip: {10, 179, 4, 10},
      local_port: 3565,
      remote_ip: {10, 179, 4, 30},
      remote_port: 3565,
      point_code: 100,
      adjacent_point_code: 200
    }
  ]

```

M3UA 配置

配置 M3UA 路由表

名称	类型	值	单位	描述
m3ua_routes	列表	[]	无	配置 M3UA 路由表

配置 M3UA 路由表 m3ua_routes

- dest_pc: 目的 PC 地址
- peer_id: 对端 ID
- priority: 优先级 - 值越大 = 优先级越高
- network_indicator: 网络指示符 :international 或 :national

配置:

```
config :omniss7,  
  m3ua_routes: [  
    # 配置 1 个 PC 100  
    %{dest_pc: 100, peer_id: 1, priority: 1, network_indicator:  
:international},  
    # 配置 2 个 PC 200  
    %{dest_pc: 200, peer_id: 2, priority: 1, network_indicator:  
:international},  
    # 配置 3 个 PC 300  
    %{dest_pc: 300, peer_id: 3, priority: 1, network_indicator:  
:international},  
    %{dest_pc: 300, peer_id: 4, priority: 2, network_indicator:  
:international}  
  ]
```

M3UA 配置

配置 M3UA 路由表 SCCP 网络指示符

項目	型	初期値	単位	説明
m3ua_gt_routes	配列	[]	個	GTルート情報

初期値は m3ua_gt_routes 配列が空配列

項目

- **gt_prefix**: GTプレフィックス - 10桁
- **peer_id**: ピアID - 0 から 9999
- **priority**: 優先度 - 0 から 255
- **description**: 説明

項目

- **source_ssn**: 発信元SSN
- **source_tt**: 発信元TT
- **source_npi**: 発信元NPI
- **source_nai**: 発信元NAI

項目

- **dest_ssn**: 宛先SSN
- **dest_tt**: 宛先TT
- **dest_npi**: 宛先NPI
- **dest_nai**: 宛先NAI

項目

- **TT (TT)**: 0=国, 1=地域, 2=都市, 3=その他
- **NPI (NPI)**: 0=国, 1=ISDN(E.164), 6=その他(E.212)
- **NAI (NAI)**: 0=国, 1=地域, 3=都市, 4=その他
- **SSN (SSN)**: 6=HLR, 7=VLR, 8=MSC, 9=EIR, その他

項目:

```

config :omniss7,
  m3ua_gt_routes: [
    # 1234
    %{gt_prefix: "1234", peer_id: 1, priority: 1, description: "1234"},
    %{gt_prefix: "44", peer_id: 2, priority: 1, description: "44"},

    # 61
    %{
      gt_prefix: "61",
      peer_id: 3,
      priority: 1,
      description: "61 TT 0→1 61",
      source_tt: 0, # 61 TT=0
      dest_tt: 1 # 61 TT=1
    },

    # NPI 49
    %{
      gt_prefix: "49",
      peer_id: 1,
      priority: 1,
      description: "49→ISDN NPI 49",
      source_npi: 6, # 49 NPI=6/E.212
      dest_npi: 1 # 49 NPI=1/ISDN/E.164
    },

    # SSN 86
    %{
      gt_prefix: "86",
      source_ssn: 8, # 86 SSN=8/MS
      peer_id: 3,
      dest_ssn: 6, # 86 SSN=6/HLR
      priority: 1,
      description: "86",
      source_tt: 0,
      dest_tt: 2,
      source_npi: 6,
      dest_npi: 1,
      source_nai: 4,
      dest_nai: 3
    },
  ],

```

```
# CAMEL/CAP
%{
  gt_prefix: "",
  peer_id: 1,
  priority: 99,
  description: "CAMEL"
}
]
```

CAMEL

CAMEL CAP

CAMEL

CAMEL/CAP    `cap_client_enabled: true` `camelgw_mode_enabled: true`

CAP

Field	Type	Value	Unit	Description
<code>cap_version</code>	String	<code>:v2</code>	String	CAP version <code>:v1</code> , <code>:v2</code> , <code>:v3</code> , <code>:v4</code>
<code>camel_gsmcf_gt_address</code>	String	(GSMCF GT)	String	GSMCF GT address CAMEL

CAP

- `:v1` → GSMCF GT address CAMEL
- `:v2` → GSMCF GT address CAMEL
- `:v3` → GSMCF GT address CAMEL

- : v4

[illegible]

11

```
config :omniss7,
  cap_client_enabled: true,
  camelgw_mode_enabled: true,
  cap_version: :v2,
  camel_gsmscf_gt_address: "68988411553"
```

CGrateS

CGrateS [○○○○○○○○○○○○○○○○/○○○○○○](#)

項目	型	デフォルト値	必須	説明
cgrates_enabled	bool	false	否	CGRateS を有効にするかどうか
cgrates_url	string	-	否 CGRateS の URL	CGRateS JSON-RPC の URL
cgrates_tenant	string	"cgrates.org"	否	CGRateS テナント名
cgrates_request_type	string	"*prepaid"	否	CGRateS リクエストタイプ "*prepaid", "*postpaid", "*pseudoprepaid"
cgrates_timeout	int	5000	否	CGRateS リクエストタイムアウト (ms)

配置:

```
config :omniss7,  
  cgrates_enabled: true,  
  cgrates_url: "http://localhost:2080/jsonrpc",  
  cgrates_tenant: "cgrates.org",  
  cgrates_request_type: "*prepaid",  
  cgrates_timeout: 5000
```

CAP M3UA 配置

CAMEL 配置 M3UA 配置 CAP 配置

配置项	配置值	配置说明	配置位置
cap_client_m3ua	配置值	-	配置位置

配置:

```
config :omniss7,  
  cap_client_m3ua: %{\br/>    mode: "ASP",  
    callback: {CapClient, :handle_payload, []},  
    process_name: :camelgw_client_asp,  
    local_ip: {10, 5, 198, 200},  
    local_port: 2905,  
    remote_ip: {10, 179, 4, 10},  
    remote_port: 2905,  
    routing_context: 4,  
    opc: 5013,      # 配置值  
    dpc: 5011      # 配置值  
  }
```

配置 NAT

配置GT = 配置 NAT

配置项	类型	默认值	说明	配置位置
gt_nat_enabled	布尔	false	是否启用 GT NAT	配置 GT NAT
gt_nat_rules	列表	[]	GT NAT 规则列表	配置 GT NAT 规则

配置 GT NAT 规则

- calling_prefix: 呼叫 GT 前缀
- called_prefix: 被叫 GT 前缀
- weight: 权重 = 100 - 1000
- response_gt: 响应 GT

配置

- 配置 weight = 100
- 配置
- calling_prefix 和 called_prefix 配置

：

```

config :omniss7,
  gt_nat_enabled: true,
  gt_nat_rules: [
    # 0000000000000000 "8772" 0000 "555"
    %{calling_prefix: "8772", called_prefix: "555", weight: 1,
response_gt: "111111"},

    # 0000000000000000 "8772"
    %{calling_prefix: "8772", weight: 10, response_gt:
"68988411553"},

    # 0000000000000000 "555"
    %{called_prefix: "555", weight: 10, response_gt:
"68988411554"},

    # 00000000 "8773"
    %{calling_prefix: "8773", weight: 10, response_gt:
"68988411554"},

    # 000000000000000000000000
    %{weight: 100, response_gt: "68988411555"}
  ]

```

00: GT NAT 00 00000000000000

M3UA 0000

M3UA 000000 MAP 00000000000000000000 MAP 000000

項目	型	デフォルト値	必須	説明
map_client_m3ua.mode	enum	-	必須	動作モード。"ASP" は、ASP として動作する。
map_client_m3ua.callback	void (*)	-	必須	コールバック関数。{MapClientM3ua} :handle_payload
map_client_m3ua.process_name	string	-	必須	プロセス名。
map_client_m3ua.local_ip	ip_address	-	必須	ローカル IP アドレス。{10, 0, 0, 1} は、10.0.0.1 を表す。[{10, 0, 0, 1}, {10, 0, 0, 2}] は、10.0.0.1 と 10.0.0.2 を表す。
map_client_m3ua.local_port	port	2905	必須	ローカル SCTP ポート。
map_client_m3ua.remote_ip	ip_address	-	必須	リモート STP/SGW IP アドレス。[{10, 0, 0, 10}, [{10, 0, 0, 10}, {10, 0, 0, 11}]] は、10.0.0.10 と 10.0.0.11 を表す。
map_client_m3ua.remote_port	port	2905	必須	リモート SCTP ポート。
map_client_m3ua.routing_context	string	-	必須	M3UA ルーティングコンテキスト ID。
map_client_m3ua.opc	port	5013	必須	オペレーティングポート。MAP プロトコルは、5013 で動作する。

項目	項目名	項目値	項目単位	項目説明
	map_client_m3ua.dpc	5011		MAP 処理の DPC 番号 (2-114-3) 5011
	map_client_m3ua.receive_watchdog	true		ASP 処理の watchdog 機能の有効化 (true) receive_watchdog 処理の Sctp 処理の有効化 (false) 有効化
	map_client_m3ua.receive_watchdog_idle	15		ASP 処理の watchdog 機能の有効化 (true) receive_watchdog 処理の Sctp 処理の有効化 (false) 有効化

項目名 IP:

```
config :omniss7,  
  map_client_m3ua: %{\br/>    mode: "ASP",  
    callback: {MapClient, :handle_payload, []},  
    process_name: :hmr_client_esp,  
    local_ip: {10, 179, 4, 11},  
    local_port: 2905,  
    remote_ip: {10, 179, 4, 10},  
    remote_port: 2905,  
    routing_context: 1,  
    opc: 5013,      # 項目名 (2-114-5)  
    dpc: 5011      # 項目名 (2-114-3)  
  }  
}
```

項目名: X-Y-Z 項目名 (X * 2048) + (Y * 8) + Z 項目名 2-114-5 =
(2 * 2048) + (114 * 8) + 5 = 5013

❏❏❏SCTP ❏❏❏:

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

00: 00 SCTP 0000000000000000 000000 SCTP 0000

5555

000000 SCTP 000???

0000000000 ESTABLISHED 0000000000000000

000000000000000000000000000000000000

§3.8 M3UA BEAT 0000 — SCTP 0000000000000000 SG 0000000000000000 M3UA BEAT0000000000000000000000000000

項目	型別	初期値	説明
receive_watchdog	bool 型	true	パケット受信監視機能を有効にするかどうか。false の場合は、SCTP パケットの受信が停止しても、プロセスは正常に動作し続けます。
receive_watchdog_idle	int 型	15	パケット受信監視機能が検出された場合、プロセスを再起動するまでのアイドル時間（秒）。 receive_watchdog: false の場合は無効です。

00 – 00 ASP 00000000:

```

config :omniss7,
  map_client_m3ua: %{
    mode: "ASP",
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1,
    receive_watchdog: false      # 00 - 00 SG 00000000
  }

```

00 - 00000000:

```

config :omniss7,
  map_client_m3ua: %{
    mode: "ASP",
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1,
    receive_watchdog: true,
    receive_watchdog_idle: 30    # 0 30 00000000000015 00
  }

```

00000000

0000000000000000Web 000API 0000000000

配置

配置	配置	配置	配置	配置
license_client.license_server_api_urls	配置	-	配置	配置 API URL
license_client.licensee	配置	-	配置	配置

配置:

```
config :license_client,  
  license_server_api_urls: ["https://localhost:10443/api"],  
  licensee: "OmniTouch Network Services Pty. Ltd."
```

Web

Property	Type	Default Value
control_panel.parent_application_readable_name	String	"OmniSS7 Stack"
control_panel.use_additional_pages	Boolean	False
control_panel.page_order	List<String>	["ControlPanelWeb.Endpoint", "ControlPanelWeb.Endpoint.https"]
ControlPanelWeb.Endpoint.url.host	String	"0.0.0.0"
ControlPanelWeb.Endpoint.https.port	Integer	8087
ControlPanelWeb.Endpoint.https.keyfile	String	"priv/cert/omnitest.key"
ControlPanelWeb.Endpoint.https.certfile	String	"priv/cert/omnitest.crt"

□□:

```
config :control_panel, ControlPanelWeb.Endpoint,  
  url: [host: "0.0.0.0", path: "/"],  
  https: [  
    port: 8087,  
    keyfile: "priv/cert/omnitouch.pem",  
    certfile: "priv/cert/omnitouch.crt"  
  ],  
  parent_application_readable_name: "OmniSS7 Stack STP"  
  
config :control_panel,  
  use_additional_pages: [  
    {SS7.Web.EventsLive, "/events", "SS7 □□"},  
    {SS7.Web.M3UAStatusLive, "/m3ua", "□□□□"}  
  ],  
  page_order: ["/events", "/m3ua", "/application",  
"/configuration"]
```

REST API 設定

項目	型別	デフォルト値	必須	説明
start_http_server	boolean	true	○	HTTP サーバを起動するかどうか 8080ポートで起動
api_ex.api.port	integer	8445	○	API サーバのポート番号
api_ex.api.listen_ip	string	"0.0.0.0"	○	API サーバのリスニング IP
api_ex.api.product_name	string	"OmniSS7"	○	API サーバの製品名
api_ex.api.title	string	"API - OmniSS7"	○	Swagger UI のタイトル
api_ex.api.hostname	string	"localhost"	○	API サーバのホスト名
api_ex.api.enable_tls	boolean	true	○	API サーバに TLS を有効にするかどうか
api_ex.api.tls_cert_path	string	"priv/cert/omnitouch.crt"	○	API サーバの TLS 証明書ファイルのパス

項目	単位	設定値	単位	説明
api_ex.api.tls_key_path	文字列	"priv/cert/omnitouch.pem"	文字列	API の TLS 鍵のパス

例:

```
config :omniss7,
  start_http_server: true

config :api_ex,
  api: %{
    port: 8445,
    listen_ip: "0.0.0.0",
    product_name: "OmniSS7",
    title: "API - OmniSS7",
    hostname: "localhost",
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem"
  }
```

API 例:

- REST API: `https://[server-ip]:8445/api/*`
- Swagger UI: `http://[server-ip]:8080/swagger`
- Prometheus 用: `http://[server-ip]:8080/metrics`

配置

配置项	是否必需	默认值	数据类型	说明
logger.level	否	:debug	字符串	日志级别 (:warni
logger.backends	否	[:console]	列表	日志后端 (SS7.We Web
logger.default_formatter.format	否	-	字符串	日志格式
logger.default_formatter.metadata	否	[]	列表	日志元数据
logger.default_formatter.truncate	否	:infinity	字符串	日志截断

配置:

```
config :logger,  
  level: :debug,  
  backends: [:console, SS7.Web.LoggerBackend]  
  
config :logger, :default_formatter,  
  format: "[$date] [$time] [$level] $message\n",  
  metadata: [:error_code, :file],  
  truncate: :infinity
```

配置

Mnesia 配置

配置项	类型	默认值	说明
<code>mnesia_storage_type</code>	字符串	<code>:disc_copies</code>	Mnesia 存储类型 <code>:disc_copies</code> 磁盘 <code>:ram_copies</code> 内存

示例:

```
config :omniss7,  
  mnesia_storage_type: :disc_copies # 磁盘  
  # mnesia_storage_type: :ram_copies # 内存
```

注意:

- `:disc_copies` - 磁盘存储 - 持久化
- `:ram_copies` - 内存存储 - 非持久化

Mnesia 配置:

- `m3ua_peer` - M3UA 对端
- `m3ua_route` - M3UA 路由
- `m3ua_gt_route` - M3UA 网关路由

配置: `Mnesia.{node_name}/` 配置

部署

部署 部署到生产环境

項目

項目	設定	説明
MAP 項目数 10 件	項目 MAP 項目 10 項目	項目数
ISD 項目 10 件	項目 ISD 項目 10 項目	項目数

HTTP 項目

項目	設定	説明
HTTP IP: 0.0.0.0	項目/Swagger 項目項目項目項目	項目数
HTTP 項目: 8080	項目/Swagger 項目項目 8080 項目	項目数

項目項目

項目	設定	説明
項目項目 5 項目	SMSc 項目 5 項目項目項目	項目数

Web UI 項目項目

項目	項目
項目項目	5 項目
項目項目	2 項目

□□□□

□□ **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  
  }
```

☐☐ SMSG ☐☐

```
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,  
  
  sctp_handler: %{  
    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 75+

配置:

- 配置 5 个
- HLR 配置 17 个
- SMSc 配置 8 个

- STP 节点 5+ 节点名称 m3ua_peers|m3ua_routes|m3ua_gt_routes 节点
- CAMEL GW 节点 14 节点
- 节点 NAT 2 节点
- M3UA 节点 8 节点
- 节点名称 Web API 节点 23 节点
- 节点 1 节点

节点名称:

HLR 节点:

- hlr_api_base_url
- hlr_service_center_gt_address
- smsc_service_center_gt_address
- 节点 map_client_m3ua.* 节点 8 节点

SMSc 节点:

- smsc_api_base_url
- smsc_service_center_gt_address
- auto_flush_dest_smsc 节点名称
- 节点 map_client_m3ua.* 节点 8 节点

STP 节点:

- sctp_handler.point_code 节点名称 SCTP 节点名称
- sctp_handler.local_ip
- sctp_handler.local_port

CAMEL GW 节点:

- cgrates_url 节点名称 CGrates
- 节点 cap_client_m3ua.* 节点 8 节点

节点名称:

- license_client.license_server_api_urls
- license_client.licensee

□□□□

- **HLR** □□ - HLR □□□□
- **SMSc** □□ - SMSc □□□□
- **STP** □□ - STP □□□□
- **API** □□ - REST API □□
- **USSD** □□□□ - USSD □□□□ HTTP □□□□
- **Web UI** □□ - Web □□□□

□□□□ NAT □□

11

GT NAT OmniSS7 GT GT
GT /
GT GT

GT NAT

GT NAT

1111

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

□ □ □ □ □

□ □ □ □ □

- `weight` 权重 = 重要性
- `response_gt` 响应 GT

[illegible]

- 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 SMS0 0000000000000000

- 0000 0000 GT 555000
- 000000 **1** 00 GT 555001
- 000000 **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	項目	項目	項目
gt_nat_enabled	項目	項目	項目/項目 GT NAT 項目
gt_nat_rules	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

```
%{
  calling_prefix: "8772",      # 項目項目項目 GT 項目項目
  called_prefix: "555",       # 項目項目項目 GT 項目項目
  weight: 10,                  # 項目項目項目項目 = 項目項目
  response_gt: "55512341112"  # 項目項目項目項目 GT
}
```

項目項目

- **calling_prefix**項目項目項目項目 GT 項目項目項目
 - 項目項目 String.starts_with?/2 項目
 - 項目項目 "" 項目 nil 項目項目項目項目項目 GT
 - 項目項目項目項目 GT
- **called_prefix**項目項目項目項目 GT 項目項目項目
 - 項目項目 String.starts_with?/2 項目
 - 項目項目 "" 項目 nil 項目項目項目項目項目 GT
 - 項目項目項目項目 GT
- **weight**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 0
 - 項目項目項目項目項目

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

00000 **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 Sc 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"}
  ]

```

Example

```

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 設定の 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 SMS GT 5552xxx 対応 GT  
    %{called_prefix: "5552", weight: 10, response_gt: "555200"},  
  
    # 電話番号 SMS GT 5553xxx 対応 GT  
    %{called_prefix: "5553", weight: 10, response_gt: "555300"},  
  
    # 番号  
    %{weight: 100, response_gt: "555000"}  
  ]
```

例

```
電話番号 GT 555100 SMS GT  
GT 441234567 対応  
  
GT NAT 対応  
GT "555100" 対応 "5551"  
response_gt "555100"  
  
電話番号 GT 555100 対応
```

例 5 番号 + 電話番号別対応

電話番号別対応 GT 設定

```

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"

# 000000000nil 00 GT
iex> GtNat.translate_response_gt_with_logging("877234567", nil,
"default_gt", "test")
# 000GT NAT [test]: 00 GT 877234567 -> 00 GT 55512341112
"55512341112"

# 00000000000 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**000000000

```
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

OmniSS7 與 HLR/HSS 的 OmniHSS 簡介

OmniHSS 簡介

OmniSS7 HLR 透過 SS7 與 OmniHSS 連接，HSS 提供以下功能：

- **OmniSS7 (HLR 端)** 透過 SS7/MAP 與 SCCP 連接
- **OmniHSS (HSS 端)** 提供以下功能：

OmniHSS 功能

OmniHSS 提供以下功能：

- **IMSI** 與 MSISDN 的 IMSI 與 eSIM 管理
- Milenage 3G/4G/5G COMP128 2G 加密
- CS 與 PS 服務管理
- CAMEL 服務
- **API** 提供 RESTful HTTP API 與 CRM 集成
- 其他功能

OmniHSS 透過 OmniSS7 的 HTTPS API 與 OmniHSS 的 MAP 協議，提供 UpdateLocation、SendAuthenticationInfo 與 SendRoutingInfo 等功能。

OmniSS7 HLR 透過 SS7/MAP 與 OmniHSS 連接，OmniHSS 提供以下功能：

IMSI 簡介

OmniHSS 提供 **IMSI** 與 MSISDN 的 IMSI 管理

- 國際行動電話用戶身份識別碼 IMSI 號碼
- **eSIM** 電子SIM卡 eSIM 號碼
- 行動電話號碼 MSISDN 號碼
- **SIM** 卡 SIM 卡號碼
- 國際行動電話用戶身份識別碼 IMSI 號碼

□□□□□

- IMSI 000000000000000KiOPc0000
- IMSI 0000000000000000
- 0000000000 IMSI 000000
- OmniSS7 IMSI 00 OmniHSSOmniHSS 0000000000
- 0000000 IMSI 000000000000000 IMSI 0000000

IMSI

☐☐ MSISDN: +1-555-123-4567

└ IMSI 1: 310260123456789 (□□□□□□ - Milenage □□□□)

└ IMSI 2: 208011234567890 (□□□□□□ - Milenage □□□□)

└ IMSI 3: 440201234567891 (□□□□□□ - COMP128 □□□□)

0000 IMSI 00000000000000000000000000000000OmniHSS 00 IMSI 0000000000000000 IMSI 000
 00000000

□□

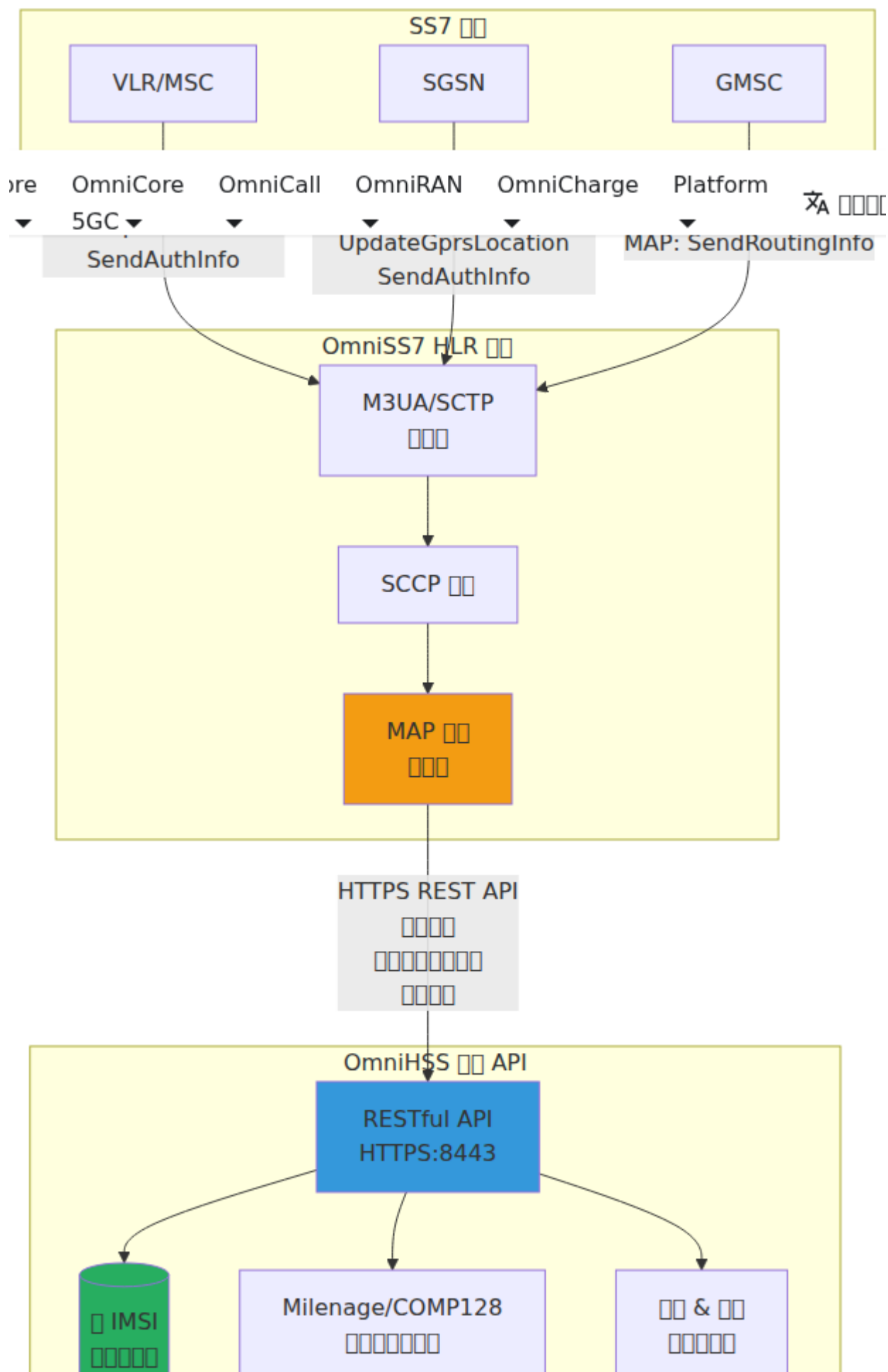
1. OmniHSS □□
2. □ IMSI □□
3. □□□ HLR □□□
4. □□ HLR □□
5. □□□□□
6. □□□□□□
7. □□□□
8. CAMEL □□
9. □□□□□□
10. HLR □□
 - □□□□□□
 - SendRoutingInfo (SRI)
 - UpdateLocation / ISD
 - SendRoutingInfoForSM
 - □□□□□□

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 - HLR 配置
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,

  # OmniHSS 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 :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
}

```

参数	数据类型	默认值	说明
hlr_api_base_url	字符串	空字符串	OmniCore HLR API 基础 URL
hlr_service_center_gt_address	字符串	空字符串	HLR 服务中心 GT 地址 OmniCore HLR 服务地址
smc_service_center_gt_address	字符串	空字符串	短信服务中心 GT 地址 短信服务中心地址
hlr_smc_alert_gts	字符串	[]	HLR 短信服务中心告警 GT 地址 短信服务中心告警地址 短信服务中心地址
hlr_alert_location_expiry_seconds	整数	172800	HLR 短信服务中心告警过期时间 短信服务中心告警过期时间 短信服务中心地址
hlr_imsi_plmn_prefix	字符串	"50557"	HLR IMSI PLMN 前缀 IMSI PLMN 前缀 IMSI PLMN 前缀
hlr_msisdn_country_code	字符串	"61"	HLR MSISDN 国家代码 IMSI PLMN 前缀 IMSI PLMN 前缀
hlr_msisdn_nsn_offset	整数	0	HLR MSISDN NSN 偏移量 IMSI PLMN 前缀 IMSI PLMN 前缀
hlr_msisdn_nsn_length	整数	9	HLR MSISDN NSN 长度 IMSI PLMN 前缀 IMSI PLMN 前缀

属性名	属性種別	属性値	属性説明
isd_network_access_mode	enum	:packetAndCircuit	Interdiction mode
isd_send_ss_data	boolean	true	Send SS data
isd_send_call_barring	boolean	true	Send call barring
camel_service_key	enum	11_110	Service key
camel_trigger_detection_point	enum	:termAttemptAuthorized	Trigger detection point
home_vlr_prefixes	list	["5551231"]	Home VLR prefixes
local_ip	ip		Local IP
local_port	port	2905	Local port
remote_ip	ip		Remote IP
remote_port	port	2905	Remote port
routing_context	enum	1	Routing context

HLR 設定

`hlr_mode_enabled: true` Web UI 設定

- **SS7** 設定 - 設定
- **SS7** 設定 - MAP 設定
- **M3UA** - 設定
- **HLR** 設定 - HLR API 設定 + 設定 ← *HLR*
- 設定 - 設定
- 設定 - 設定

設定 **SMSc** 設定

設定

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

設定

OmniHSS 設定 OmniSS7 設定 RESTful API 設定

OmniHSS

OmniHSS

- IMSI
- Ki/OPc 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 及 MSISDN 的查詢
- 查詢 Ki/OPc 的查詢
- 查詢 UpdateLocation 的查詢
- 查詢的查詢

查詢

查詢

查詢 **updateLocation** MAP 的 OmniSS7 及 OmniHSS 的 VLR 查詢

1. **UpdateLocation** 的 IMSI 的 VLR GT 的 MSC GT
2. **OmniHSS** 的查詢
3. **OmniHSS** 的 VLR/MSC 的查詢
4. **InsertSubscriberData (ISD)** 的 VLR 的查詢
5. **UpdateLocation** 的 VLR 的 `hlr_service_center_gt_address` 的 HLR GT
6. **alertServiceCenter** 的 SMSc GT 的 `hlr_smsc_alert_gts` 的查詢

查詢 `hlr_service_center_gt_address` 的 UpdateLocation 的 HLR 的查詢
查詢 VLR/MSC 的 HLR

查詢

查詢 UpdateLocation 的 HLR 的 **alertServiceCenter** 的 MAP 的 64 的 SMSc 的 SMSc 的 SMSc 的 SMSc

查詢

查詢 SMSc 的查詢

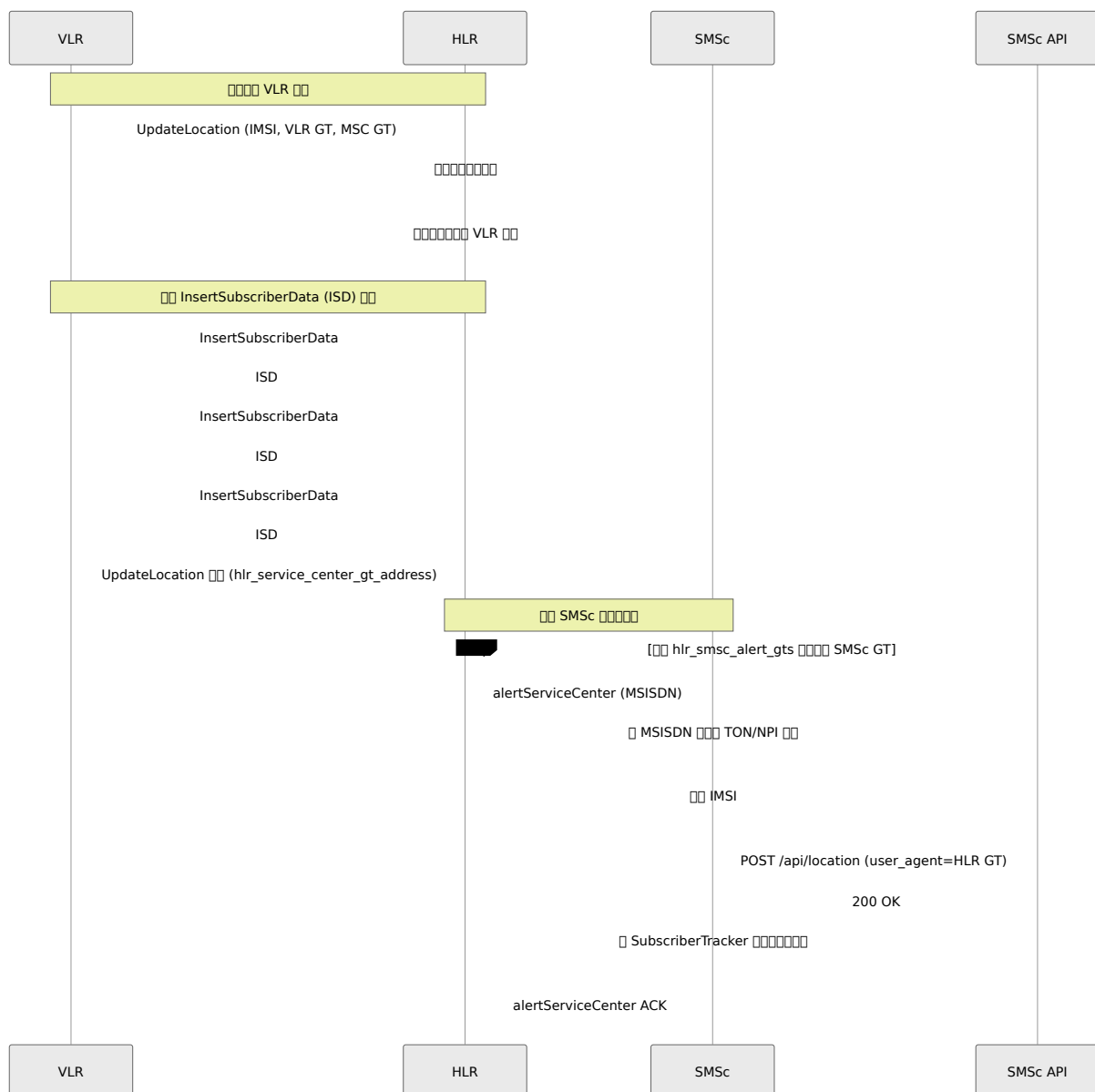
```

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,
```

이동통신

isd_network_access_mode 이동통신망 액세스 모드

값	설명	비고
:packetAndCircuit	이동통신망 액세스 모드 GPRS/LTE	데이터 - 음성
:packetOnly	이동통신망 액세스 모드/LTE	데이터 전용
:circuitOnly	이동통신망 액세스 모드/SMS	음성 전용

이동통신 ISD

이동통신망 액세스 모드 ISD

이동통신 ISD - 데이터 전용

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

이동통신망 액세스 모드

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

이동통신 + 데이터 전용

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

ISD

이동통신 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	???000000000
300	0000000VPN
000	0000000

00000

```
# 0000000000000000
camel_service_key: 11_110,

# 00 VPN 00
camel_service_key: 300,
```

000000

camel_trigger_detection_point 0000000000000000 CAMEL 000

000	00	0000
:termAttemptAuthorized	0000000000000000	0000000000000000
:tBusy	0000	000000
:tNoAnswer	000000	0000000000
:tAnswer	0000	0000000000

000000

000000000000000000000000

```
camel_trigger_detection_point: :termAttemptAuthorized,
```

000000000000000000000000

```
camel_trigger_detection_point: :tBusy,
```

000000000000000000000000

```
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

00

```
# 00 VLR 00
# 000"00"000 VLR 000000
# 00000 VLR 00000000000000000000 SRI 00
# 0000000000000000 PRN 000 MSRN
home_vlr_prefixes: ["555123"],
```

00000

```
# 000000
home_vlr_prefixes: ["555123"],

# 00000000000000000000
home_vlr_prefixes: ["555123", "555124", "555125"],
```

00000

1. 0000000000

0000 VLR 0000000000000000

```
GMSC → HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR 0000 API 000000
HLR 00 VLR 000"5551234567"
HLR 000VLR 0 "555123" 00 → 0000
HLR → GMSC: SRI 0000 CAMEL 00000
- IMSI
- VLR 000"5551234567"
- gsmSCF 000MSC00"5551234501"
- CAMEL 0000011_110
- 000000termAttemptAuthorized
```

2. 000000000 PRN0

0000 VLR 0000000000000000

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-COI": %{
          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 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 PRN MSRN PRN

- 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 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	VLR 番号 [circuit_session.assigned_vlr]
フラグ	なし	フラグ notProvidedFromVLR
extendedRoutingInfo	-	[camelRoutingInfo]
gsmSCF 番号	OmniHSS API	MSC 番号 [circuit_session.assigned_msc]
サービスキー	runtime.exs	CAMEL サービスキー [camel_service_key]
トリガー検出ポイント	runtime.exs	CAMEL トリガー検出ポイント [camel_trigger_detection_point]
CAMEL サービス	なし	CAMEL サービス
gsmSCF サービス	なし	gsmSCF サービス

MSRN 番号

VLR 番号 home_vlr_prefixes

フラグ

項目	API	説明	注
IMSI	OmniHSS API	OmniHSS から取得した IMSI	"
VLR 名	OmniHSS API	現在接続している VLR [circuit_session.assigned_vlr]	"
住所	なし	なし [notProvidedFromVLR]	:
extendedRoutingInfo	-	[routingInfo]	-
電話番号MSRN	PRN 名	電話番号から取得した MSRN	"

処理フロー

```
1. OmniSS7 から SendRoutingInfo 受信
2. OmniSS7 から OmniHSS API 経由で住所取得
3. OmniSS7 から VLR 経由で home_vlr_prefixes 取得

VLR 経由で住所取得
→ CAMEL 経由で住所取得

VLR 経由で住所取得
→ MSC から PRN 取得
→ PRN から MSRN 取得
→ MSRN から住所取得
```

結論

- OmniSS7 から OmniHSS 経由で住所取得
- OmniHSS から IMSI 経由で VLR/MSC 経由で住所取得
- OmniSS7 から住所取得 MAP 経由

補足

```
# 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	: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 HLR 項目 SRI-for-SM 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	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 SMSc 000 MT-ForwardSM 00000000
smsc_service_center_gt_address: "5551234567", # 00

# MSISDN ↔ IMSI 0000
# PLMN 000MCC001 = 00000+ MNC001 = 000000
hlr_imsi_plmn_prefix: "001001", # 000000000000
```

MSISDN ↔ IMSI 00

000000

0000000 OmniSS7 000 SRI-for-SM 0000000 IMSI0

- **hlr_imsi_plmn_prefix** 00000000 IMSI 0 MCC+MNC 000000“50557”00
MCC=5050MNC=570
- **hlr_msisdn_country_code** 0000000 IMSI→MSISDN 0000000000000000“61”000
0000“1”0000/0000
- **hlr_msisdn_nsn_offset** 0MSISDN 00000000NSN0000000000000 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 SMSc 0000000 **MSISDN**00000000

00000000

- SMSc 0000000 MSISDN 0 SRI-for-SM0000“5551234567”0
- HLR 000000000000000000000000 IMSI
- HLR 0 SRI-for-SM 000000 IMSI
- SMSc 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 碼 - IMSI 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 碼 "001001" 如何？
- **MSISDN** 如何？
- 如何透過 IMSI 碼 15 碼

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 碼

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

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

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

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
msisdn = String.replace_leading(subscriber_portion, "0", "") # =
"555123456"
```

数据表

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

数据表

- 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
- 提取 15 位 IMSI 前缀 15 位

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 前缀 + 手机号码 MSISDN

流程图

1. SRI-for-SM 消息中包含 IMSI 号码，HLR 返回 IMSI 号码，
2. HLR 返回 IMSI 号码，UpdateLocation 消息中包含 SendAuthenticationInfo 消息，
3. HLR 返回 IMSI 号码，UpdateLocation 消息中包含 SendAuthenticationInfo 消息，

流程图

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"

流程图

流程图 runtime.exs 流程图

```
# 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 詳細 - SRI-for-SM 詳細

API 詳細

概要

API	対象	説明
OmniHSS API	OmniHSS への接続 詳細	IMSI/MSISDN/ circuit_session 詳細 VLR/MSC
runtime.exs	OmniSS7 詳細 詳細	smc_service_center_gt_address camel_service_key isd_network_access_mode
API	接続 詳細	接続SS 詳細
API	MAP 詳細 詳細	UpdateLocation 詳細 IMSI/ SRI / MSISDN
API	接続	SRI-for-SM 詳細 IMSI/hlr_imsi_prefix + NSN

詳細

runtime.exs 詳細

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

runtime.exs 詳細

- camel_service_key - 11_110
- camel_trigger_detection_point - :termAttemptAuthorized

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

OmniHSS 類型

OmniHSS 類型 REST API 類型

- IMSI 類型 MSISDN 類型
- 類型類型VLR/MSC 類型
- 類型類型
- 類型類型類型

類型

OmniSS7 類型

- ← 類型
- 類型
- MAP 類型
- 類型
- 類型

OmniHSS 類型 類型類型類型類型類型類型類型類型類型 **OmniHSS** 類型OmniHSS 類型 類型類型類型類型類型類型類型類型 IMSI 類型

OmniSS7 類型 Omnitouch 類型

MAP

←

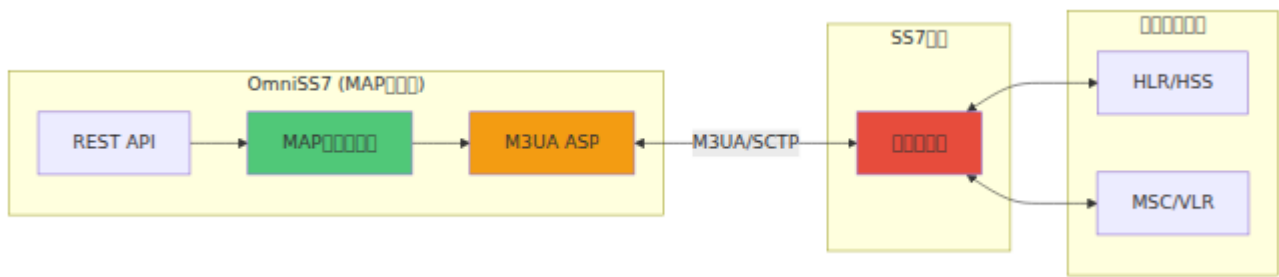
OmniSS7MAPMAP

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

□□□□

```
config :omniss7,  
  # □□MAP□□□□□  
  map_client_enabled: true,  
  
  # MAP□□□□M3UA□□□□□ASP□□□□□STP/SGP□  
  map_client_m3ua: %{  
    mode: "ASP", # M3UA□□□"ASP"□□□□□□"SGP"□□□  
  }  
  □□  
    callback: {MapClient, :handle_payload, []}, # □□□□□□□□□  
    process_name: :map_client_asp, # □□□□□□□  
    local_ip: {10, 0, 0, 100}, # □□IP□□  
    local_port: 2905, # □□SCTP□□  
    remote_ip: {10, 0, 0, 1}, # □□STP/SGP IP  
    remote_port: 2905, # □□STP/SGP□□  
    routing_context: 1 # M3UA□□□□□  
  }
```

配置文件

```
config :omniss7,  
  # 启用MAP客户端  
  map_client_enabled: true,  
  
  # M3UA配置  
  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 接口 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

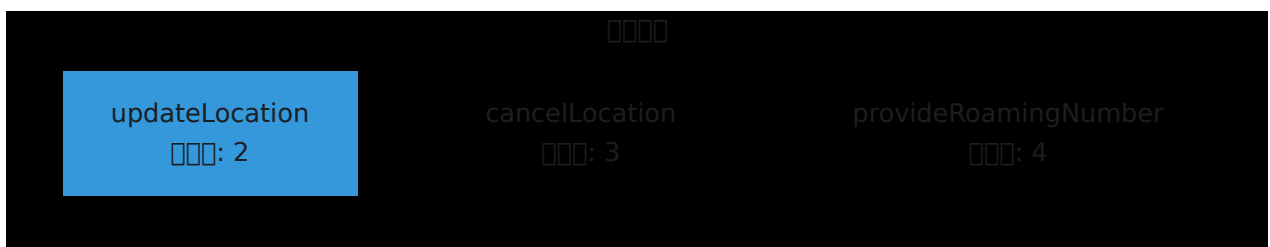
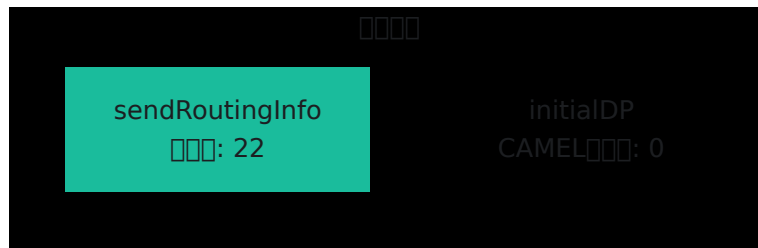
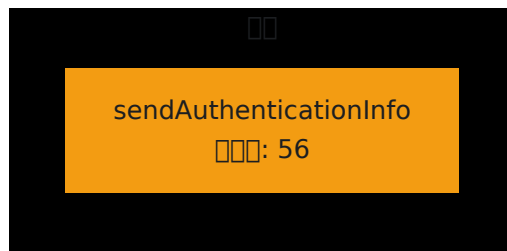
HLR 位置更新信息 HLR 位置更新信息

API 接口 POST /api/updateLocation

请求

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

MAP



API

Swagger UI

Swagger UI

Swagger UI

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

1. `/api/sri-for-sm`
2. “”
- 3.
4. “”
- 5.

API

- **200** -
- **400** -
- **504** - SS7 10

MAP

- `map_requests_total` - MAP
 - `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) Overview

← Introduction

OmniSS7 is a platform that provides a unified interface for sending and receiving SMS messages. It supports various protocols, including SS7, and integrates with external systems like OmniMessage.

OmniMessage Integration

OmniSS7 SMSc supports SS7 and integrates with OmniMessage for sending and receiving SMS messages.

- **OmniSS7 (SMSc)** supports SS7/MAP and SCCP protocols.
- **OmniMessage (SMS)** supports sending and receiving SMS messages.

OmniMessage Features

OmniMessage provides the following features for SMS:

- Supports various SMS protocols and formats.
- Supports Delivery Receipts (DLR) and status reports.
- **SMSc** supports sending and receiving SMS messages.
- Supports various SMS services and applications.
- Supports TPS (Text-to-Push) and other advanced features.
- **API** provides a RESTful HTTP API for integration.
- Supports various SMS gateways and providers.

OmniMessage integrates with OmniSS7 via HTTPS API for sending and receiving SMS messages.

OmniSS7 SMSc supports various protocols and integrates with OmniMessage for sending and receiving SMS messages. OmniMessage supports SS7/MAP and integrates with OmniMessage for sending and receiving SMS messages.



1. OmniMessage
2. SMS
3. SMSc
4. HTTP API
- 5.
- 6.
- 7.
8. SMSc
9. VLR
- 10.
- 11.
- 12.

SMS

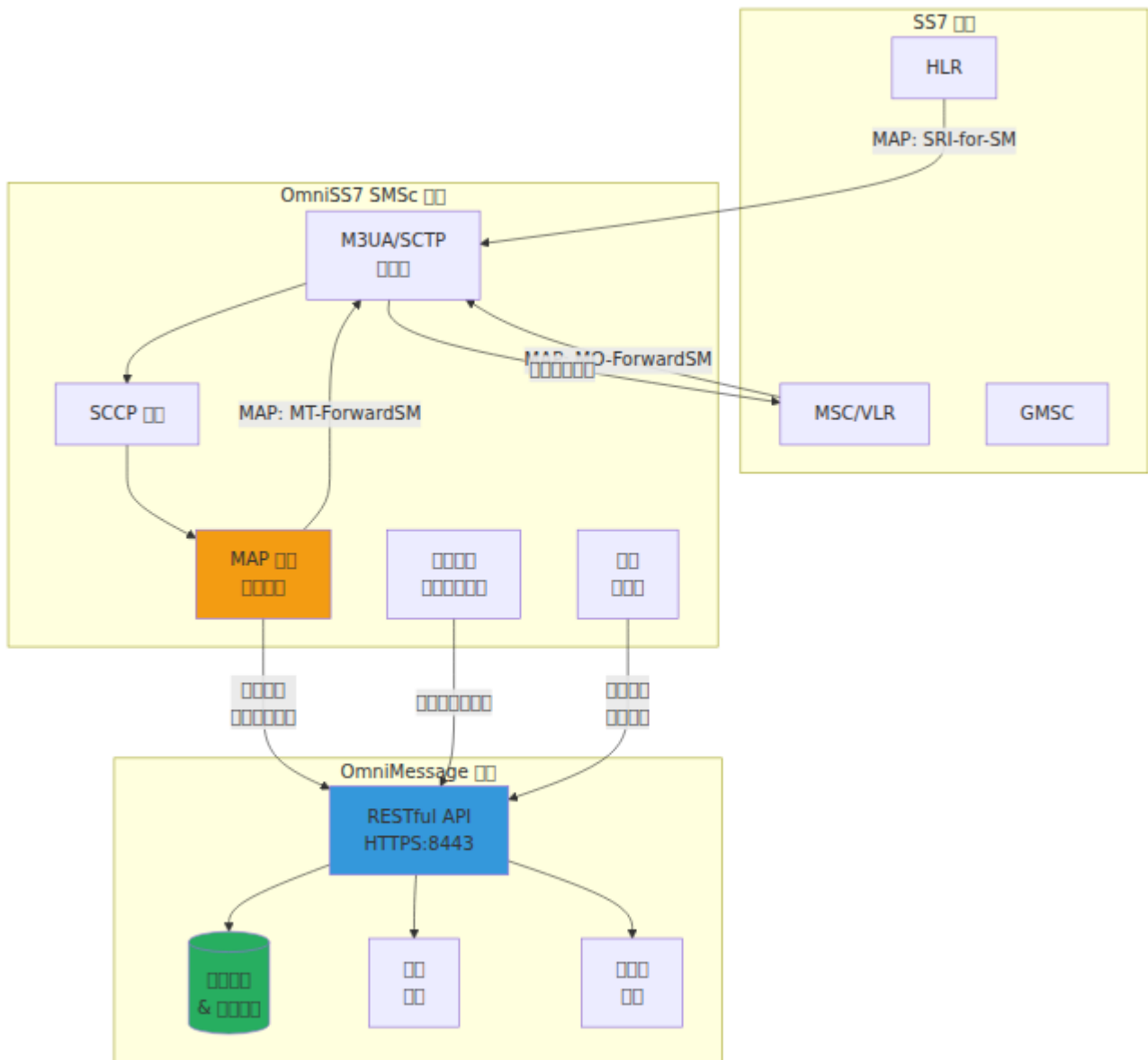
OmniSS7 SS7

OmniMessage

SMS OmniSS7 SMSc

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

SMS □□□□



□□ SMSc □□

OmniSS7 SSMSc SSMSc

□□□ SMSc □□

OmniSS7 □ config/runtime.exs □□□□□□□□□□□□□□□□ SMC □□

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 0000 #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 消息 API 地址
smsc_name	字符串	"{hostname}_SMSc"	消息 SMSc 名称
smsc_service_center_gt_address	字符串	空	消息服务地址
auto_flush_enabled	布尔	true	是否自动刷新
auto_flush_interval	整数	10_000	自动刷新间隔
auto_flush_dest_smsc	字符串	空	消息 SMSc 地址
auto_flush_tps	整数	10	自动刷新 TPS
local_ip	字符串	空	消息 SMSc 本地 IP
local_port	整数	2905	消息 SCTP 端口
remote_ip	字符串	空	SS7 消息 STP IP

項目	項目名	項目値	項目説明
remote_port	項目名	2905	項目 Sctp 項目
routing_context	項目名	1	M3UA 項目項目 IC

項目 SMSc 項目項目項目項目

項目 smsc_mode_enabled: true 項目 map_client_enabled: true 項目Web UI 項目項目

- 項目 **SS7** 項目 - 項目項目
- 項目 **SS7** 項目項目 - MAP 項目項目
- 項目 **M3UA** - 項目項目
- 項目 項目 - 項目項目項目 STP
- 項目 項目項目 - 項目項目項目 STP
- 項目 **SMSc** 項目 - SMSc API 項目 + SMS 項目項目 ← 項目 SMSc
- 項目 項目 - 項目項目
- 項目 項目 - 項目項目

HLR 項目 項目項目項目項目

項目項目

- SMSc 項目項目 map_client_enabled: true 項目項目項目項目
- OmniMessage** 項目項目OmniMessage API 項目項目項目項目 smsc_api_base_url 項目項目
- 項目項目項目項目 5 項目 項目項目 SMS.FrontendRegistry 項目項目 OmniMessage 項目項目
- API** 項目項目項目項目 OmniMessage API 項目項目項目項目 5 項目項目
- MAP** 項目項目項目SRI-for-SM 項目項目 10 項目項目項目MT-ForwardSM 項目項目 30 項目項目項目項目項目 2G/CS 項目項目項目 VLR 項目項目
- 項目項目項目項目項目項目 SMS 項目項目
- 項目項目 STP 項目項目 M3UA 項目項目項目項目/項目項目 MAP SMS 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

- **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	smc_api_base_url	"https://localhost:8443"	C A
項目名	smc_name	"{hostname}_SMS" {hostname}_SMS	[S
項目名	smc_service_center_gt_address	"5551234567"	[[C [F [[S

項目名

項目名 OmniMessage 項目名 5 項目名 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 smc_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` 変数にホスト名を代入し、`"{hostname}_SMSc"` のように

OmniMessage API

OmniSS7 は SS7 ネットワークを介して MAP メッセージを送受信する OmniMessage

- 送信メッセージ
- 送信メッセージ (MO) の受信
- 送信メッセージ (MT) の受信
- 送信メッセージの送信/受信
- 送信メッセージの送信

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

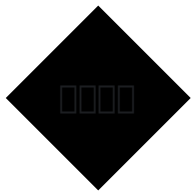
M3UA 接口

与 SCCP 接口

与 SCCP 接口

与 TCAP/MAP 接口

与 MAP 接口



Forward-SM

与 SMS TPDU

接口

POST

POST `/api/messages_raw`

POST `/api/events`

MAP

SMS

HSS/HLR API (`hlr_api_base_url`) IP-SM-GW MT-SMS SRI-for-SM HSS API IMSI VLR **VLR** HSS

M3UA 消息 SCTP 消息

M3UA 消息

消息 SCCP 消息

Core 5GC OmniCore OmniCall Omni

消息 TCAP/MAP 消息

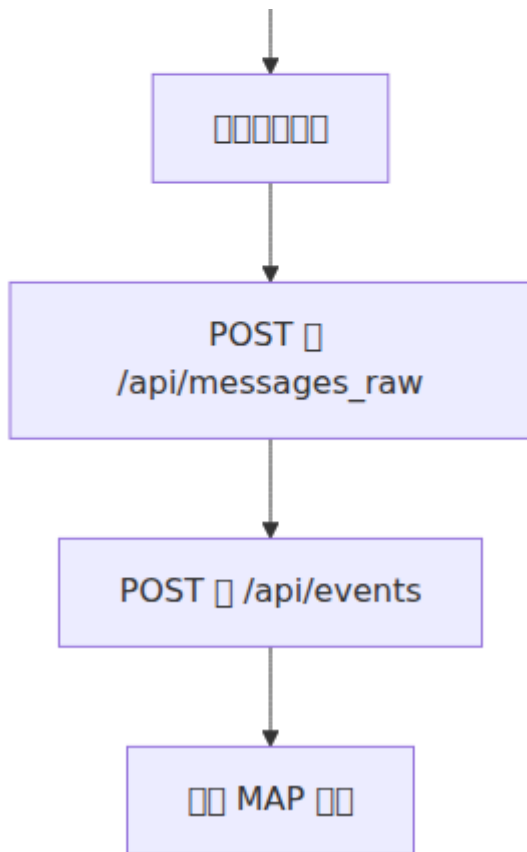
消息 MAP 消息

消息

Forward-SM

消息 SMS TPDU

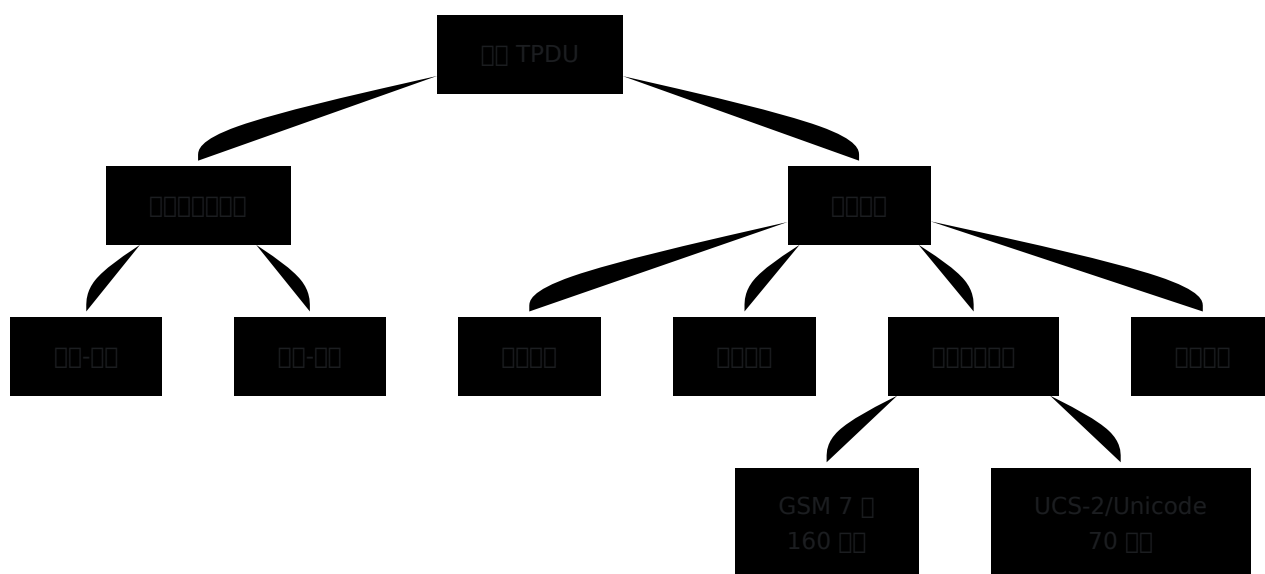
消息



Sequence

- **VLR** sends HSS API request to VLR with IMSI + VLR ID and TTL. HSS API returns **VLR ID**.
- **HSS API** returns `hlr_api_base_url` to IP-SM-GW. MSISDN is sent to HSS API. **IMSI** is sent to **VLR**. MT-ForwardSM is sent to VLR — `SRI-for-SM`.
- HSS API returns VLR ID to HLR. `alertServiceCenter` is returned.
- **SRI-for-SM** returns HSS API request. HSS returns `SRI-for-SM` to HLR. IMSI is returned. **HLR ID** **SRI-for-SM**.
- VLR sends MT-ForwardSM to SCCP. VLR returns HSS API request to VLR.
- **MT-ForwardSM** returns MT-ForwardSM to **30**. 2G/CS VLR.

SMS TPDU 结构



配置项

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

HLR 配置项 alertServiceCenter 配置项 HLR 配置项

配置 alertServiceCenter

HLR 配置 UpdateLocation 配置 VLR/MSC 配置 HLR 配置 **alertServiceCenter** MAP 配置 64 配置 SMS Sc 配置

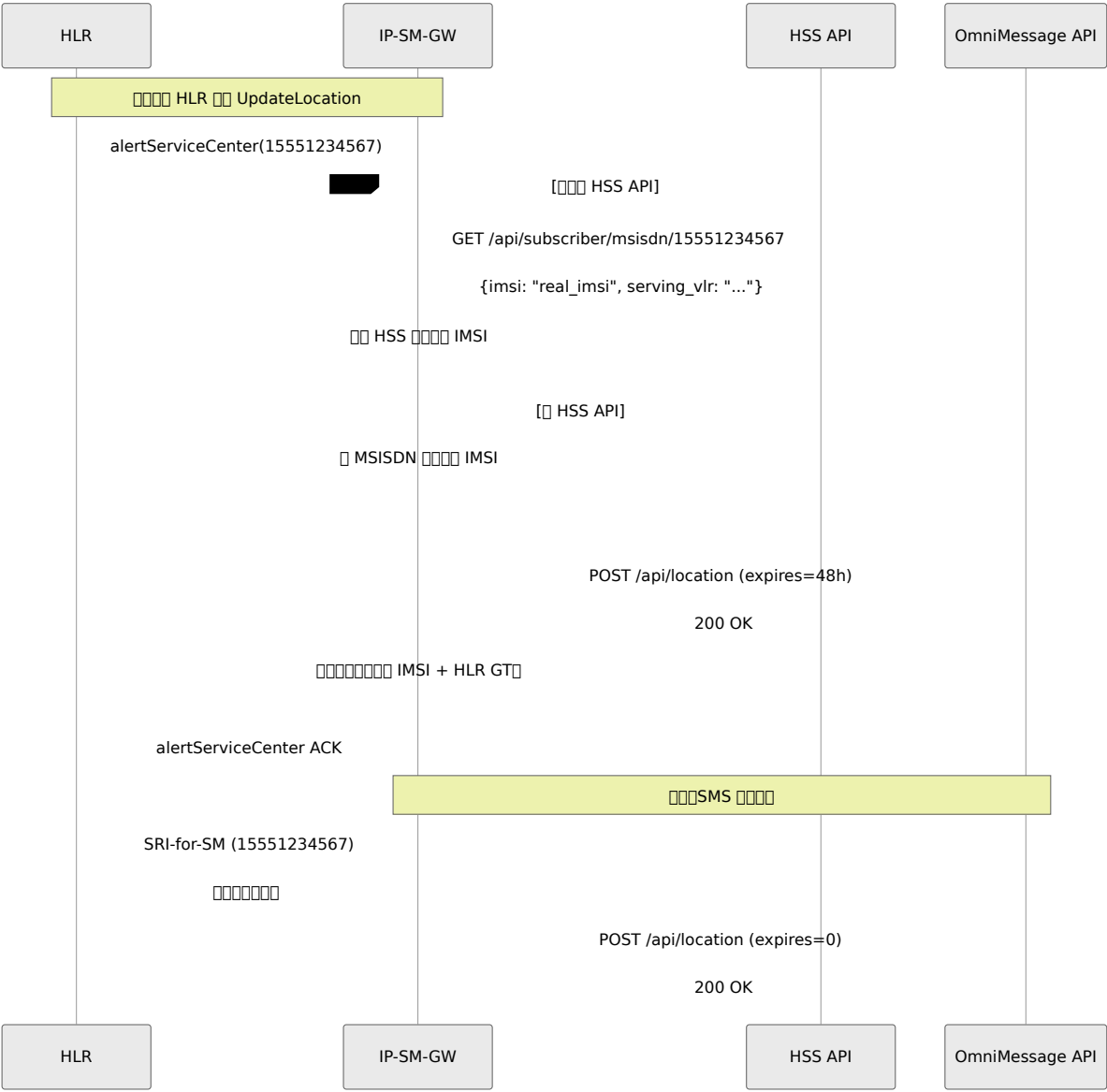
配置

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 SCS HLR
expires

SS7

SCS SS7 MO→MT

SS7

SCS SS7 (MO) SMS source_smsc "SS7_MO_15551234567"

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

SS7

SS7 IP-SM-GW MSC MSC MO→MT

- `source_smsc` `"SS7"`
 - `source_smsc` Web API/SMPP `source_smsc` SS7 `source_smsc`

SMSC

`source_smsc`

Source	Destination	Event
SS7 MO-FSM	"SS7_M0_15551234567"	MO - MO
SS7 MO-FSM	"SS7_M0_15551234567"	MO - MT MO→MT
API/SMPP	"ipsmgw" "api_gateway"	API
SMSc	"smsc-node-01"	SMSc

Example

Example

```
{
  "message_id": 12345,
  "name": "SMS",
  "description": "SMS - source_smsc 'SS7_M0_15551234567' to 'SS7'
  destination_smsc"
}
```

Example

- **Web UI** SS7 `/events`
- `events` API `events`
- `events`

11

```
smc_mode_enabled:
true  VLR  hlr_api_base_url
```

1 SS7 MO-SMS

1. 网络 → MSC/VLR → IP-SM-GW → MO-ForwardSM
2. IP-SM-GW 向 MO-FSM 发送 source_smsc = "SS7_M0_2471900"
3. 网络接收
4. 网络 VLR 向 HSS 发送 → 网络
5. IP-SM-GW 向 "SS7" 发送 source_smsc 网络 + 网络 → 网络
6. 网络 "网络"
7. 网络 SRI-for-SM 向 MT-ForwardSM 发送

2 SS7 MO-SMS

1. 网络 → MSC/VLR → IP-SM-GW → MO-ForwardSM
2. IP-SM-GW 向 MO-FSM 发送 source_smsc = "SS7_MO_2471900"
3. 网络接收并处理
4. 网络 HSS API 接口 → 网络
5. IP-SM-GW 向 SS7 网络
6. HSS 向 IMSI + VLR → MT-ForwardSM 向 VLR

SMSc ☐ ☐ ☐ ☐ ☐

SMSc GenServer alertServiceCenter

11

□□□□□□

- □□□□□□□□□□

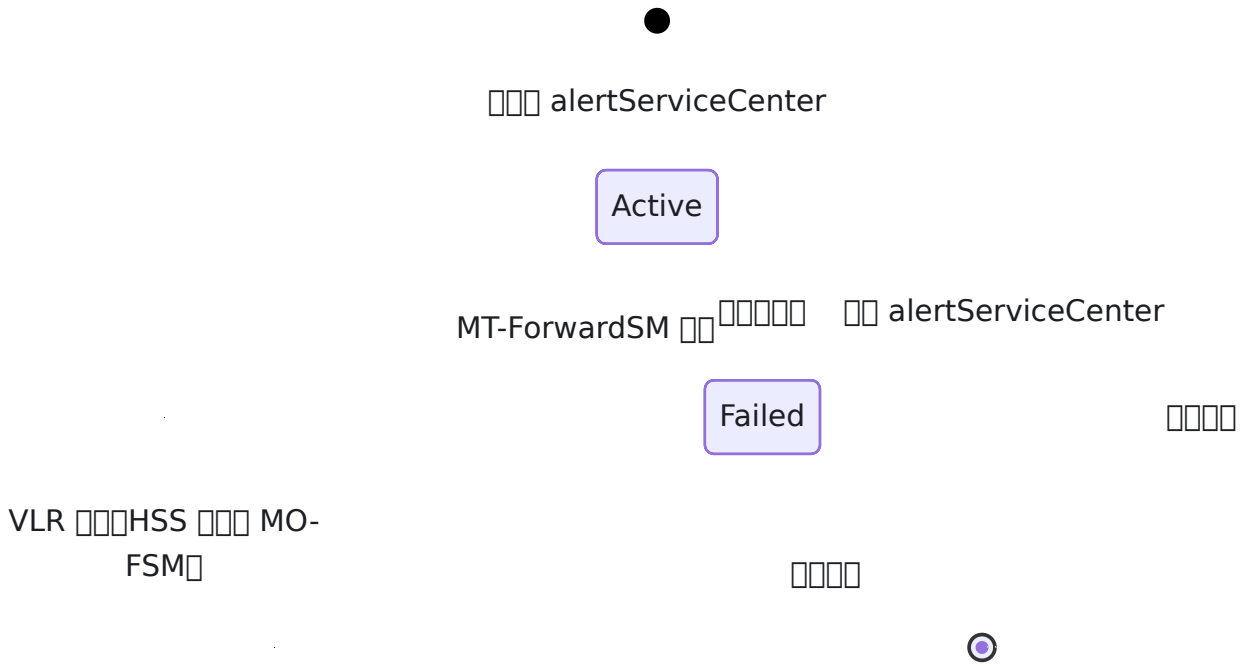
- **HLR** 儲存 alertServiceCenter 的 HLR
- 儲存 IMSI/HSS 號碼
- 儲存 alertServiceCenter 的 GT
- **Web UI** 儲存 VLR GT 和 VLR 號碼

資料庫

儲存 alertServiceCenter 的 HLR

欄位	說明	值
msisdn	號碼	"15551234567"
imsi	號碼 IMSI 號碼 HSS 號碼	"001010123456789"
hlr_gt	號碼 alertServiceCenter 的 GT	"15551111111"
messages_sent	發送 MT-FSM 次數	5
messages_received	接收 MO-FSM 次數	2
status	:active 或 :failed	:active
updated_at	最後更新 Unix 時間	1730246400
vlr_address	號碼 VLR GT 號碼	"14155550100"
vlr_cached_at	VLR 最後更新 Unix 時間	1730246400

--	--	--	--



11

alertServiceCenter

- 電話番号を保持
- 番号 `status = :active`
- 番号 HLR GT
- 番号 HSS API 番号 IMSI 番号 IMSI
- 番号 VLR 番号

□ **HSS API** □□□ **VLR** □□□□□□

- 移動局 VLR からの IMSI
- 移動局 VLR からの HSS

MO-ForwardSM

- SCCP → GT → VLR → MO-ForwardSM → GT → VLR
-

MT-ForwardSM

- 00000 VLR 0000000000000000 HSS API

0 SRI-for-SM 0000

- 00 messages_sent 000
- 00 updated_at 000

0 SRI-for-SM 0000

- 00 status = :failed
- 0000000000000000

0000000000

- 0 ETS 0000
- 00000 Web UI 0

Web UI - SMS Sc 00000

000 /smc_subscribers 000000 2 0

00000000 SMS Sc 0000000000 config/runtime.exs 00000 SMS Sc 0000000000000000
000000

SMS Sc 000 000000000000000000

00

1. 0000

- MSISDN 0 IMSI 0 HLR GT 0 VLR GT 000000 VLR 000
- 00/000000
- 00000000/0000000000
- 0000000000000000
- 00000000000000

2. 0000

- 00000000

- 数据库连接
- 数据库操作
- 向 HLR 写入

3. 数据

- 数据库连接
- 数据库操作

数据

SMS 数据库				表名
MSISDN	IMSI	HLR GT	VLR GT	Msg Svc
15551234567	001010123456789	15551111111	14155550100	5/2
15559876543	001010987654321	15551111111	-	0/6
15551112222	001010111222233	15552222222	14155550200	3/1

数据表 | 表名 | 表名 | 表名 | 表名

API 接口

数据库连接

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

#
SMSc.SubscriberTracker.message_sent(msisdn)
SMSc.SubscriberTracker.message_received(msisdn)

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

#
SMSc.SubscriberTracker.remove_subscriber(msisdn)

#
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`

VLR □□□□

□□□ IP-SM-GW □□□□□□□□ HSS API □□□□□□□□□□□□ VLR □□□□□□□□□□ MT □□□□□□□□□□ HSS□

□□□□

VLR □□□□□□□□□□□□

1. **HSS API** 配置 HSS API 配置 VLR 配置 IMSI 配置 VLR 配置
2. **MO-ForwardSM** 配置 MO-ForwardSM 配置 SCCP 配置 GT 配置 VLR/MSC 配置 VLR 配置

配置 MT 配置 IMSI 配置 TTL 配置 HSS API

配置

配置 VLR 配置

- **MT-ForwardSM** 配置 SCCP 配置 VLR 配置 VLR 配置
- **TTL** 配置 `vlr_cache_ttl_seconds` 配置 TTL 配置

配置 alertServiceCenter 配置 VLR 配置 VLR 配置

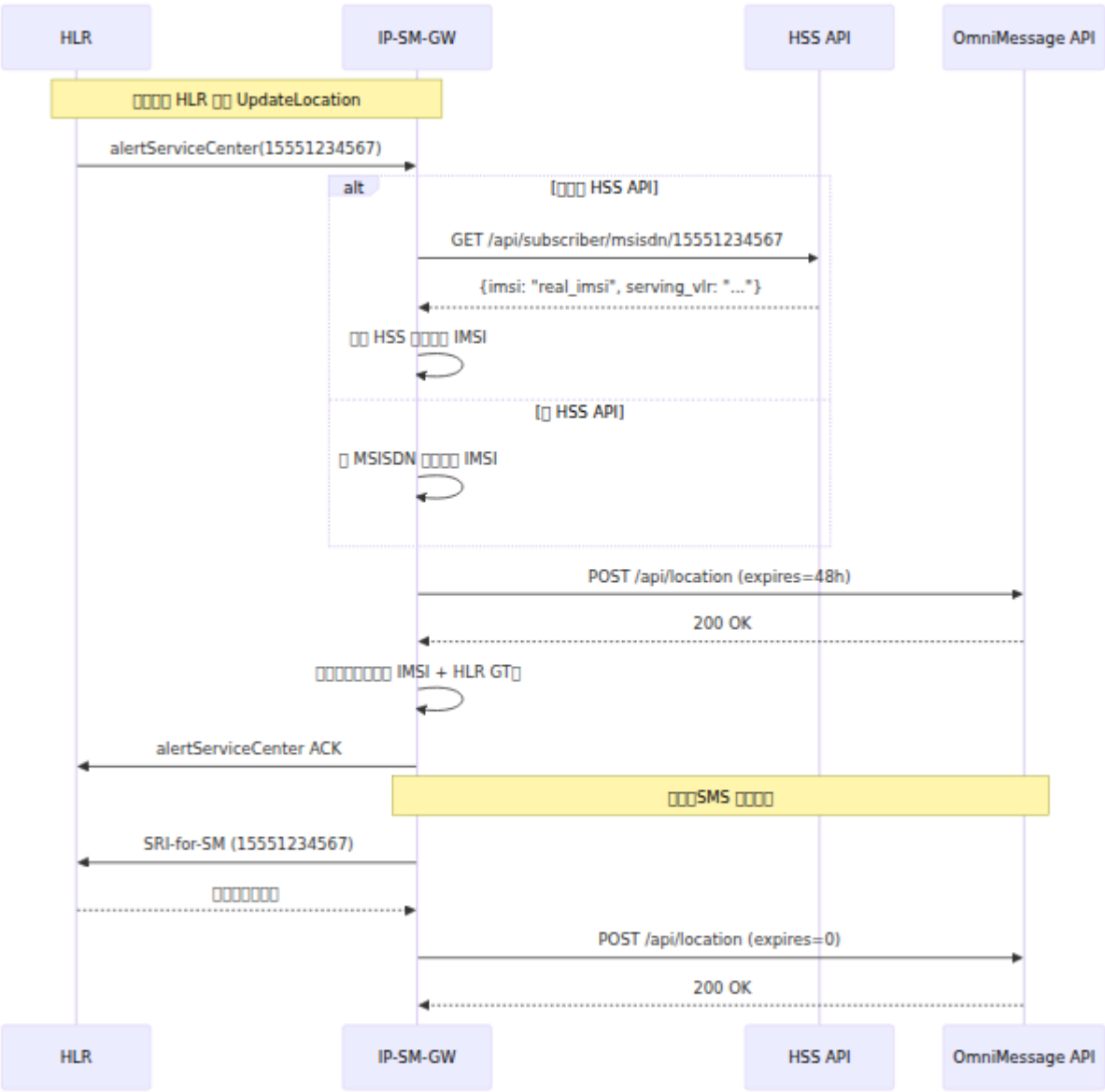
配置

```
config :omniss7,
  # HSS/HLR API 配置 VLR 配置
  hlr_api_base_url: "https://10.179.2.140:8443",

  # VLR 配置 TTL 配置 3600 = 1 配置
  vlr_cache_ttl_seconds: 3600
```

配置	配置	配置	配置
<code>hlr_api_base_url</code>	配置	配置	<code>nil</code>
	配置	配置	<code>nil</code>
<code>vlr_cache_ttl_seconds</code>	配置	配置	<code>3600</code>

□□□



□□□□ 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` 设置 API 刷新间隔
- 配置 `auto_flush_dest_smsc` 设置
- 配置 `auto_flush_tps` 设置
- 配置
 - 配置 **SRI-for-SM** 设置 HLR 刷新
 - HLR 设置 MSISDN 设置 IMSI
 - HLR 设置 MT-ForwardSM 设置 SMSC GT 设置
 - 配置 HLR 设置 **SRI-for-SM** 设置
 - 配置 MSC/VLR 设置 **MT-forwardSM**
 - 配置 API 设置刷新/设置
 - 配置 API 设置

配置 **SRI-for-SM** 设置 MSISDN 设置 IMSI 设置 GT 设置
配置 IMSI 设置 **HLR** 设置 **SRI-for-SM** 设置

SMSc 配置

配置

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秒计算一次
rate(smsc_messages_delivered_total[5m]) /
(rate(smsc_messages_delivered_total[5m]) +
rate(smsc_messages_failed_total[5m]))

# 平均时间
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. 消息传递初始化
2. 消息传递
3. 消息传递完成
4. 通过 M3UA 消息传递
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

0000

- 00 auto_flush_tps
- 00 HLR 000
- 00000000

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

M3UA & M2PA STP 目錄

← 目錄

目錄 OmniSS7 的 目錄 (STP) 目錄

目錄

1. 目錄 STP?
 2. STP 目錄
 3. 目錄 TDM 目錄
 4. 目錄 STP 目錄
 5. 目錄
 6. M2PA 目錄
 - M3UA 與 M2PA 目錄
 - 目錄 M2PA 目錄
 - SLTM 目錄
 - 目錄 Web UI 目錄 M2PA
 - M2PA 目錄
 7. 目錄
 8. 目錄
 9. 目錄
 - 目錄
 - DROP 目錄 - 目錄
 10. 目錄
 11. 目錄
 12. 目錄
 13. M3UA 目錄
 - M3UA 目錄
 - 目錄 IP 目錄
-

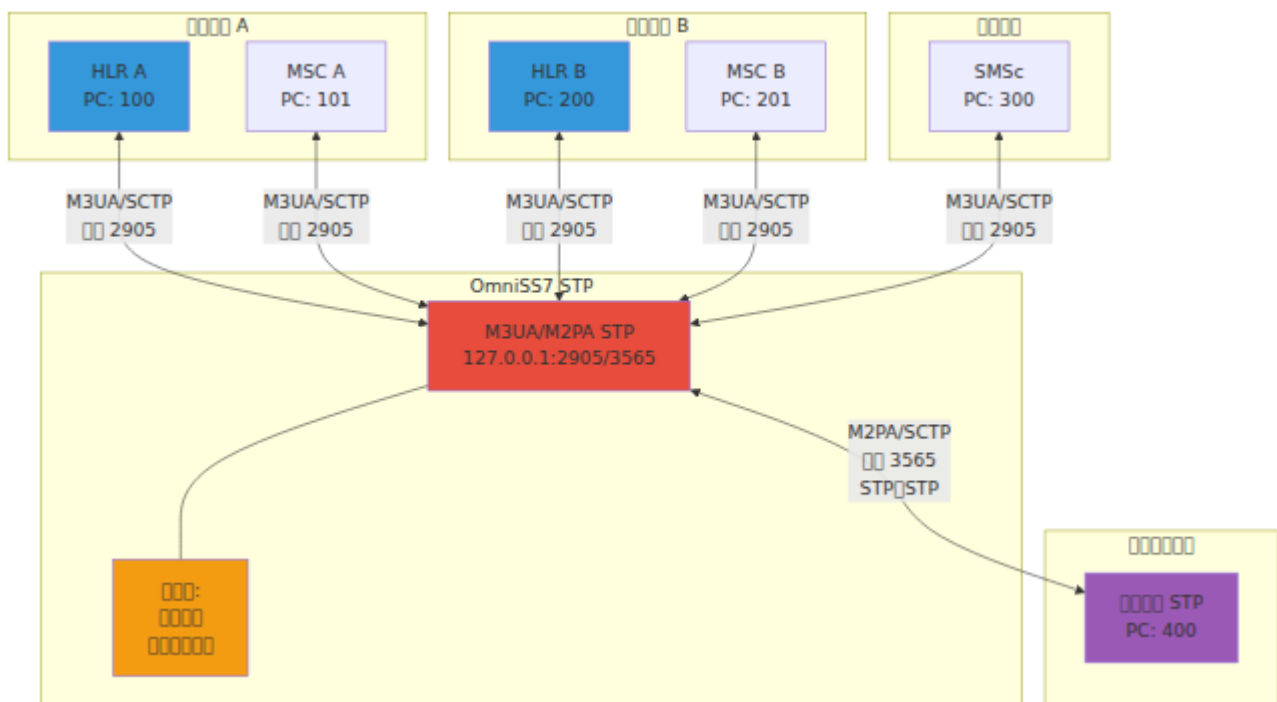
STP (STP)?

STP (STP) SS7 IP

STP

- PC GT SS7
- SS7 IP M3UA/SCTP
-
-
-

STP 概要



STP 構成要素

ASP (アプリケーションサーバ)

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

SGP (サービスゲートウェイ)

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

AS (アプリケーションサーバ)

- アプリケーション ASP 接続

- 网络部署方案
- 网络 ASP 部署方案

网络部署方案 TDM 网络

OmniSS7 网络 IP 网络 SIGTRAN 网络 (M3UA/M2PA 网络 SCTP)网络 TDM 网络 SS7 网络 网络 (SGW) 网络

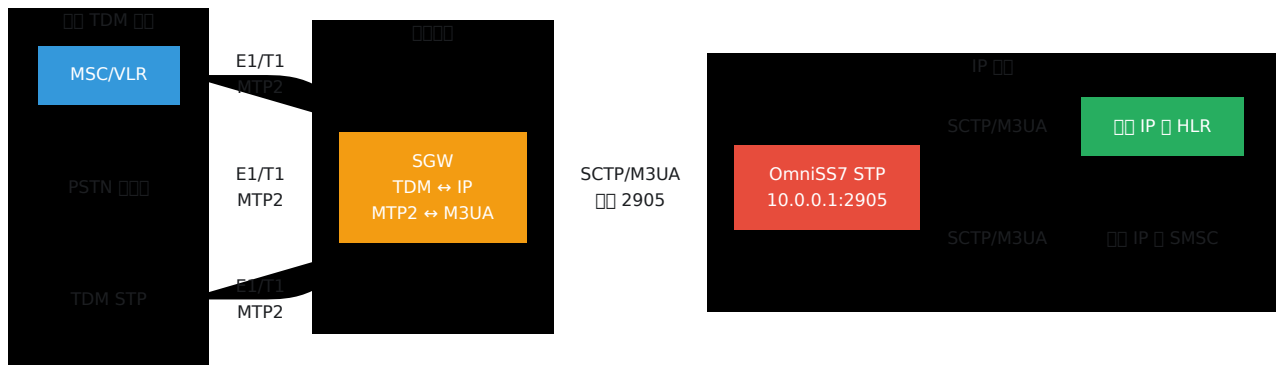
网络

网络 (SGW) 网络 SS7 网络

- TDM 网络 MTP1/MTP2/MTP3 网络 E1/T1 网络 SS7
- IP 网络 M3UA 网络 M2PA 网络 SCTP/IP 网络 SIGTRAN

SGW 网络 OmniSS7 网络 IP 网络 IP 网络 TDM 网络 STP 网络 HLR 网络

TDM 网络 IP 网络



□□□□□

□□	TDM SS7	□□□□	IP (SIGTRAN)
□□□□	SCCP/TCAP/MAP	← □□ →	SCCP/TCAP/MAP
□□	MTP3	□□	M3UA/M2PA
□□	MTP2	□□	SCTP
□□	MTP1 (E1/T1)	□□	IP/Ethernet

SGW □ TDM □□□ MTP1/MTP2□□□ IP □□□ M3UA □ M2PA □□ MTP3 □□□□□□□□□□SCCP□
TCAP□MAP□CAP□□□□□□□

□□ **OmniSS7** □□□□□

OmniSS7 □□ **ASP** (□□□□□□□□) □□□□□□□□□□ SGW □□ **SGP** (□□□□□□□)□

□□□□

```

config :omniss7,
  # [] ASP []
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :sgw_connection,
    # [] (OmniSS7)
    local_ip: {10, 0, 0, 1},
    local_port: 0, # []
    # []
    remote_ip: {10, 0, 0, 100}, # SGW IP []
    remote_port: 2905, # SGW M3UA []
    routing_context: 1 # SGW []
  },

  # [] STP []
  peers: [
    %{
      peer_id: 1,
      name: "TDM_Gateway",
      role: :client, # OmniSS7 []
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 100}, # SGW IP []
      remote_port: 2905,
      routing_context: 1,
      point_code: 100, # SGW []
      network_indicator: :international
    }
  ],

  # [] TDM []
  m3ua_routes: [
    %{
      dest_pc: 100, # TDM MSC []
      peer_id: 1, # [] SGW []
      priority: 1,
      network_indicator: :international
    },
    %{
      dest_pc: 200, # TDM HLR []
      peer_id: 1,
      priority: 1,

```

```

    network_indicator: :international
  }
]

```

SGW 設定項目

設定項目 OmniSS7 設定項目

項目	項目	項目
IP	OmniSS7 の IP	外部 IP
ポート	OmniSS7 の SCTP ポート	2905
ネットワーク	SGW の AS ネットワーク	SGW ネットワーク
タイプ	SGW ネットワーク TDM ネットワーク	ネットワーク
接続	ASP 接続	接続

設定項目

1 IP 設定 TDM 設定

OmniSS7 の SGW ネットワーク TDM の HLR 設定 (SRI-SM, PRN)

```

OmniSS7 (ASP) → SGW (SGP) → TDM ネットワーク → HLR
M3UA                MTP2                MTP3

```

2 OmniSS7 の IP 設定 TDM 設定 STP

OmniSS7 の IP ネットワーク TDM ネットワーク

```

IP SMSC → OmniSS7 STP → SGW → TDM HLR
      ↓
      IP ネットワーク HLR

```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□□ M3UA STP □□

OmniSS7 STP STP

配置 STP 配置

OmniSS7 在 `config/runtime.exs` 中配置 STP 配置

1. 在 `config/runtime.exs`
2. 配置 STP 配置 (在 53-174 行)
 - 配置 1 个 STP 配置 (在 53-85 行)
 - 配置 2 个 HLR 配置 (在 87-123 行)
 - 配置 3 个 SMSc 配置 (在 125-174 行)
3. 配置 STP 配置 (在 53-85 行)
4. 配置 STP 配置 (在 53-85 行)
5. 配置 STP 配置
6. 配置 STP 配置 `iex -S mix`

STP 配置

配置 STP 配置

```

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

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

```

STP/SGW

STP/SGW STP/SGW

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	list of str		ローカル IP アドレス {10, 0, 0, 1} のリスト [10, 0, 0, 1], {10, 0, 0, 2}]	{10, 179, 4, 10}
local_port	int	2905	ローカル SCTP ポート	2905
remote_ip	list of str		リモート STP/SGW IP アドレス	{10, 179, 4, 11}
remote_port	int	2905	リモート SCTP ポート	2905
routing_context	int	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 の有効化	true

ローカル IP アドレス local_ip とリモート IP アドレス remote_ip の SCTP ポート番号は、SCTP ポート番号

STP の設定

map_client_enabled: true Web UI の設定

- SS7 の設定
- SS7 の MAP の設定
- M3UA の設定
- ローカル IP アドレス ← STP の設定
- リモート IP アドレス ← STP の設定
- ローカル ポート
- リモート ポート

HLR 和 SMSc 的部署

部署

- 部署 SCTP 服务 (IP 为 132) 部署
- 部署 M3UA 服务 2905部署
- 部署数据库
- 部署 Web UI 和 API 部署 **Mnesia** 部署 部署 部署
- 部署 `runtime.exs` 部署 Mnesia 部署
- 部署
- **Web UI**部署 **Web UI** 部署 Web 部署
- **API** 部署 **API** 部署 REST API 部署 Swagger UI 部署

部署 STP

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

部署 STP

部署 `config/runtime.exs`

```
config :omniss7,  
  sctp_handler: %{  
    enabled: true,  
    local_ip: {127, 0, 0, 1},    # 本地 IP  
    local_port: 2905,           # 本地端口  
    point_code: 100             # STP 点代码  
  }
```

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 和 MSC 服务

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

Mnesia (Mnesia)

Mnesia 是 Erlang 的分布式数据库，用于存储 Mnesia 的数据。

配置

1. **Runtime.exs** 配置 `config/runtime.exs` 中的 `peers` 配置
配置 `m3ua_peers` 和 `m3ua_routes` 以及 `m3ua_gt_routes`
2. **Web UI** 配置 `Web UI` 中的 `Mnesia` 配置
3. 配置 `runtime.exs` 中的 `Mnesia` 配置
4. 配置 `Web UI` 中的 `Mnesia` 配置

Mnesia 配置

配置 Mnesia 数据库

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

配置	配置	配置	配置
<code>:disc_copies</code>	配置	配置	配置
<code>:ram_copies</code>	配置	配置	配置

配置 `:disc_copies`

Mnesia 配置

Mnesia 配置 Mnesia 配置

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

配置

配置

1. **Runtime.exs** - 配置
2. **Web UI** - 配置 **Web UI** 配置
3. **REST API** - 配置 **API** 配置

配置 `runtime.exs` 配置 `Web UI` 配置

配置 M3UA 配置

配置 M3UA 配置 STP HLR MSC SMSC 配置 `config/runtime.exs`

□□□□□□□

| □□□□□□ `peers` □□□□□□□□□ `m3ua_peers` □□□□□□□□□□□□□□□□

```

config :omniss7,
  peers: [
    # STP :client
    %{
      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
    },

    # HLR :client
    %{
      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
    },

    # MSC :server
    # :server STP
    %{
      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
    }
  ]

```

```

},

# 配置动态客户端
%{
    peer_id: 4,
    name: "Dynamic_Client",
    role: :server,
    remote_ip: {10, 0, 0, 40},      # 设置 IP
    remote_port: 0,                 # 0 = 任意端口
    routing_context: 4,
    point_code: 400,
    network_indicator: :international
}
]

```

□ □ □ □ □ □ □

項目	型	範囲	説明
peer_id	型	型	ピアの識別子
name	型 型	型	名前 (最大 255 文字)
role	型	型	:client または :server
local_ip	型 型 型	型 型 型	ローカル IP アドレス (例: {10, 0, 0, 1}) または SCTP アドレス (例: [{10, 0, 0, 1}, {10, 0, 0, 2}])
local_port	型	型 型	ローカルポート番号 (0 から 65535)
remote_ip	型 型 型	型	リモート IP アドレス (例: {10, 0, 0, 1}) または SCTP アドレス (例: [{10, 0, 0, 1}, {10, 0, 0, 2}])
remote_port	型	型	リモートポート番号 (0 から 65535)
routing_context	型	型	M3UA ルーティングコンテキスト
point_code	型	型	ポイントコード (SS7)
network_indicator	型	型	:international または :national

SCTP 프로토콜은 TCP와 UDP 프로토콜과 유사하지만, SCTP 프로토콜은 IP 프로토콜의 SctpInit() 함수를 사용하여 SCTP 프로토콜을 IP 프로토콜의 remote_ip 주소로 연결합니다.

SCTP `local_ip` `remote_ip` `IP` **SCTP**

配置节点

每个节点都需要配置 `server` 和 `remote_port` 属性

- 配置 `remote_port: 2905` 的节点
 - 作为严格对等节点
 - 作为严格对等节点
- 配置 `remote_port: 0` 的节点
 - 作为严格对等节点/对等节点
 - 指定 IP 地址
 - 指定端口

示例

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

# 配置 10.5.198.200 节点
%{
  peer_id: 2,
  name: "Flexible_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 0, # 指定端口
  # ... 其他配置
}
```

M2PA

OmniSS7 M3UA M2PA SS7

M2PA

M2PAMTP2 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** (Service Gateway) 提供 IP 地址并管理 IP 地址分配
- 提供 IP 地址分配/管理
- 提供 IP 地址分配 (RFC 4666)

M3UA 提供 HLR、MSC、SMSC、VLR 与 STP

M2PA 提供

M2PA 提供

- **STP** 与 **STP** 提供 STP 提供
- **TDM** 提供 M2PA 提供 SS7 TDM
- **MTP2** 提供 MTP2 提供
- 提供 M2PA 提供

M2PA 提供 HLR、MSC、SMSC 与 STP - M3UA、M2PA 提供 STP 提供

M2PA 提供

M2PA 提供 M3UA 提供 protocol 提供

M2PA 提供

M2PA 提供 config/runtime.exs 提供 peers 提供 M3UA 与 M2PA 提供 protocol 提供

M2PA 提供

項目	値	説明
protocol	:m2pa	この M2PA 接続のタイプは :m3ua
role	:client :server	この接続の役割
local_port		この SCTP 接続の M2PA ポート番号は 3565
remote_port		この SCTP 接続の M2PA ポート番号は 3565
point_code		この接続の点番号
adjacent_point_code		この接続の隣接点番号 M2PA 接続
send_slrm		この SLTM 接続の SLTM 接続
network_indicator		:international :national -

この M2PA 接続のポート番号は 3565 であり、M3UA 接続のポート番号は 2905

SLTM 接続

SLTM 接続は SLTA 接続と MTP3 接続の M2PA 接続の READY 状態の接続であり、ITU-T Q.707 接続の SLTM 接続の SLTA 接続

SLTM 接続

send_slrm 接続の SLTM 接続

<code>send_slm</code> []	[]
<code>true</code>	[] [] [] [] [] READY [] [] SLTM [] [] SLTA
<code>false</code>	[] [] [] [] [] [] SLTM [] [] SLTA
[] [] [] [] []	[] [] Q.707 [] SCTP [] [] [] SLTM [] SCTP [] [] [] []

[] [] [] [] **Q.707** []

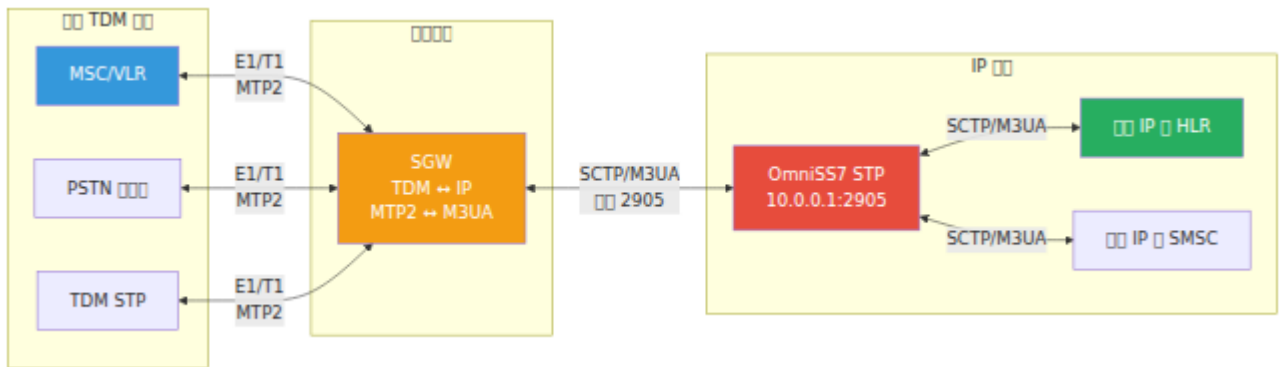
- [] `initiate_connection: true` [] SCTP [] [] / [] [] → [] [] [] [] [] SLTM
- [] `initiate_connection: false` [] SCTP [] [] / [] [] → [] [] SLTM

[] [] **SLTM** [] []

[] [] [] [] Q.707 [] [] [] [] [] [] [] [] [] []

```
# [] [] [] [] [] SLTM [] [] SCTP [] []
%{
  peer_id: 100,
  name: "Partner_STP",
  protocol: :m2pa,
  role: :client,
  local_ip: {10, 0, 0, 1},
  local_port: 3565,
  remote_ip: {10, 0, 0, 2},
  remote_port: 3565,
  point_code: 7415,
  adjacent_point_code: 15528,
  network_indicator: :international,
  send_slm: true # [] [] [] [] SLTM [] [] [] SCTP [] []
}
```

SLTM [] []



SLTM

READY

SLTM

"waiting for peer to send SLTM"

`send_slm: true`

SLTM SLTA

1. `adjacent_point_code` SLTM DPC
2. `:international` `:national`
- 3.

`adjacent_point_code` STP `sctp_handler.point_code`

M2PA

M2PA

1. **Down** -
2. **Alignment** - 1
3. **Proving** - 2
4. **Ready** -

○○○○○○○○○○○○○○○○○○○○○○○○○○○○

○○ Web UI ○○ M2PA ○○○

Web UI ○ ○○ ○○○○○ M2PA ○○○○○○○○○○○

1. ○○ ○○○○○
2. ○○ "○○○" ○○○
3. ○○ "○○○○○○"
4. ○○ "M2PA (RFC 4165)" ○○○○○○○
5. ○○ ○○○○○
 - ○○○○○○○○○○○○○○○
 - ○○○M2PA
 - ○○○○○○○○○○○
 - ○○○○○ PC
 - ○○/○○ IP ○○
 - ○○/○○○○○○○○ M2PA ○ 3565
 - ○○○○○○○○○○○○○
6. ○○ "○○○○○"

○○○○○○○○○○○○○○○○○○○○○○○○○○○○

- ○○ - M3UA ○○○
- ○○ - M2PA ○○○

M2PA ○○○○

M2PA ○○○○ OmniSS7 ○○○○○○○○○○○

- ○○○○○○ M2PA ○ M3UA ○○
- ○○○○○○○○ M2PA ○○○○○○○
- ○○○○○○M2PA ○ M3UA ○○○○○○○○○○○○○○○
- ○○○○○○○○○○ M2PA ○○○○○○ M3UA○○○○○

M2PA 简介

M2PA 通过 Prometheus 暴露监控指标

指标

- `m2pa_messages_sent_total` - 发送 MTP3 消息
- `m2pa_messages_received_total` - 接收 MTP3 消息
- `m2pa_bytes_sent_total` - 发送 M2PA 字节
- `m2pa_bytes_received_total` - 接收 M2PA 字节

链路状态指标 `link_name` `point_code` `adjacent_pc`

指标

- `m2pa_link_state_changes_total` - 链路状态变化
DOWN → ALIGNMENT → PROVING → READY
 - `link_name` `from_state` `to_state`

指标

- `m2pa_errors_total` - 错误总数
 - `decode_error` - M2PA 解码错误
 - `encode_error` - M2PA 编码错误
 - `sctp_send_error` - SCTP 发送错误
 - `link_name` `error_type`

指标

- Prometheus 地址 `http://your-server:8080/metrics`
- 配置示例

M2PA 部署

1. 部署 M2PA 3565 端口
2. 配置 Prometheus 监控
3. 配置 SCTP/IP 地址 132.0.0.0
4. 配置防火墙规则

5. `SocketHandler` 클래스를 구현합니다.
6. `SocketHandler` 클래스를 `M2PA` 클래스에 등록합니다.

M2PA 클래스

M2PA 클래스는 **SCTP.SocketHandler** 인터페이스를 구현하며 M2PA 클래스는 SCTP.SocketHandler 인터페이스를 구현합니다.

예제

- SCTP.SocketHandler 인터페이스
- `M2PA` 클래스를 구현합니다.

예제

```
# Sctp.SocketHandler
sctp_handler: %{
  enabled: true,
  local_ip: {10, 179, 4, 10},
  local_port: 3565,
  point_code: 100
}

# M2PA
peers: [
  %{
    peer_id: 1,
    name: "M2PA_Link_1",
    protocol: :m2pa,
    role: :client,
    local_ip: {10, 179, 4, 10},
    local_port: 3565,
    remote_ip: {10, 179, 4, 20},
    remote_port: 3565,
    point_code: 100,
    adjacent_point_code: 200
  },
  %{
    peer_id: 2,
    name: "M2PA_Link_2",
    protocol: :m2pa,
    role: :client,
    local_ip: {10, 179, 4, 10},
    local_port: 3565, #
    remote_ip: {10, 179, 4, 30},
    remote_port: 3565,
    point_code: 100,
    adjacent_point_code: 300
  }
]
```

SCTP

- unordered
- PPID 5 M2PA RFC 4165

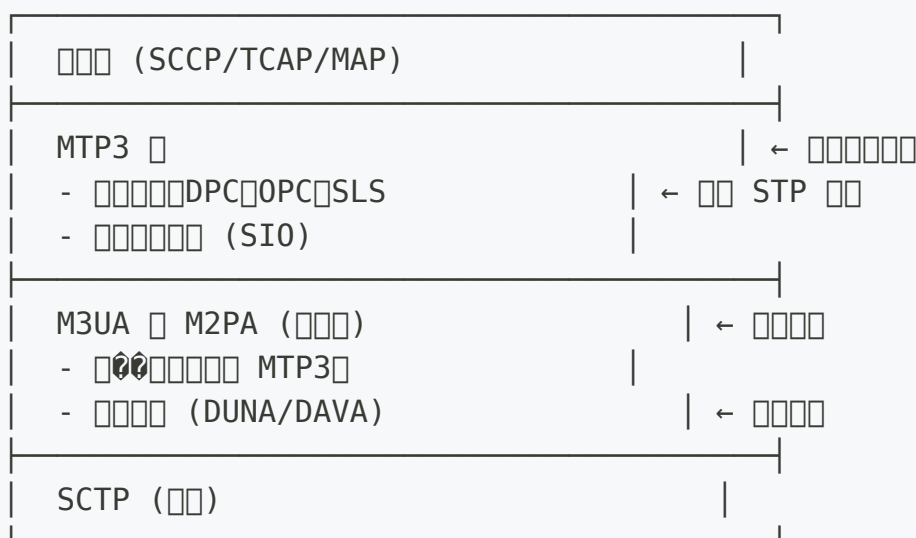
SS7

SS7 MTP3 DPC

SS7

SS7

SS7



SS7

1. MTP3

- MTP3 DPC OPC
- M3UA 528
- M2PA
- STP DPC
-

2. M3UA

- M3UA DUNA DAVA SCON DUPU
-

- 000000000000/000
- 0000000000

STP 00000000

- 00 **M3UA DATA** 000STP 0000000000 5280000 MTP3 0000000 MTP3 0000
0DPC0OPC0SLS00MTP3 00 DPC 00000000
- 00 **M2PA** 0000000STP 0 M2PA 000000000 MTP3 0000000 MTP3 0000000
DPC0
- **M3UA** 000000000000DUNA0DAVA0SCON00 M3UA 000000000000000000
0000000000

00000000

00000000 config/runtime.exs0

```

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,            # 10000
      priority: 1,
      network_indicator: :international
    },
    %{
      dest_pc: 300,
      peer_id: 4,            # 1000000000000000
      priority: 2,
      network_indicator: :international
    }
  ]
]

```

mask 14 mask

10000

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

4. 00000000 DPC000000
5. 000000000000 000000000000000000 00000000
6. 0 MTP3 00000 M3UA DATA 0 M2PA 0000000000000000
7. 000000000000
8. 00000000000000000000000000000000

0000

000 14 00000 0-163830000000000000000000 /140000000000 0000 0000 000
0 0000

0000

0000000000 PC 0000000 DPC 00000000 00000000000000000000000000000000

0000000

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

階数

階名 mask 階数 階名

階名 1階

```
# 000000000000 PC
%{
    dest_pc: 1000,
    peer_id: 1,
    priority: 1,
    network_indicator: :international
}
# 0000 14 - 0000 PC 1000

# 000000000000
%{
    dest_pc: 1000,
    peer_id: 1,
    priority: 1,
    mask: 14, # 000000
    network_indicator: :international
}
# 0000 PC 1000
```

2000

```
%{
    dest_pc: 1000,
    peer_id: 2,
    priority: 1,
    mask: 12,
    network_indicator: :international
}
# 4 PCs
# PC 1000 1001 1002 1003
```

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 peer_id
      priority: 1,             # 配置 priority
      description: "44 GT"     # 配置 description
    },

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

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

    # SSN 配置 4
    %{
      gt_prefix: "555",
      source_ssn: 8,          # 配置 source_ssn = 8 为 SMSC
      peer_id: 4,
      dest_ssn: 6,            # 配置 dest_ssn 为 6 为 HLR
      priority: 1,
      description: "61 配置 SMS 配置"
    }
  ]
]
```

GT □□□□

GT □□□□□□□□□□□□□□

SCCP

GTSSNTTNPINAI

GT
?

GT + SSN + TT + NPI
+ NAI

?

：

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

- 消息源 source_ssn 为 SCCP 消息 SSN 标识
- 消息 source_ssn 为 nil 标识 SSN 标识

3. TT/NPI/NAI 标识

- 消息源 source_tt 为 source_npi 为 source_nai 标识
- nil 标识

4. 消息源

- 消息源
- 消息源 GT 标识 → SSN → TT → NPI → NAI → 消息源

5. 消息源

- 消息源 dest_ssn 为 dest_tt 为 dest_npi 为 dest_nai 为 STP 标识

- 00000000000000000000

6. 00000000

- 00000000 GT 0000STP 00000000 DPC 000000

00 GT 0000000000NPI 0 NAI

00 GT 0000 SSN 0000STP 000000 SCCP 0000000000000000

- 0000 (**TT**)0000000000000000
- 00000000 (**NPI**)0000000000000000ISDN00000000
- 00000000 (**NAI**)000000000000000000000000

0000000000

000000000000000000000000

- source_tt000000000000000000
- source_npi00000000000000000000
- source_nai00000000000000000000
- nil 0 = 000000000000

0000000000

0000000000000000

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

00000000

00000000000000000000000000000000

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	GT + SSN SSN 6
555881234567	6	-	-	-	"555" → SSN → GT X	GT SSN
441234567890	6	0	-	-	"44" → TT=0 → GT 1	GT + TT TT 3
12125551234	8	0	1	4	"1" → TT=0 NPI=1 NAI=4	GT+TT+NPI+NAI

TT/NPI/NAI 規則

1. GT

- SSN
- SSN
- SSN TT=0 3桁 → SSN TT=1

2. GT

- SSN

- 國際號碼 NPI=6 PSTN NPI=1

3. 國際號碼

- 國際號碼 NAI
- 國際號碼 NAI=4 國際號碼 NAI=3

4. 國際號碼

- 國際號碼
- 國際號碼 TT=0 國際號碼 A TT=2 國際號碼 B

5. 國際號碼

- 國際號碼
- STP 國際號碼

國際號碼

國際號碼

國際號碼

國際號碼

國際號碼 ☐ enabled 國際號碼

```
config :omniss7,
  m3ua_routes: [
    # 國際號碼
    %{
      dest_pc: 100,
      peer_id: 1,
      priority: 1,
      network_indicator: :international,
      enabled: true # 國際號碼
```

USSD 開發

← 目錄

OmniSS7 **USSD** 透過 SS7/MAP USSD 透過 HTTP/JSON 透過 HTTP 透過 USSD 透過

目錄

1. 目錄
2. 目錄
3. 目錄 USSD 目錄
4. 目錄
5. HTTP 目錄
6. 目錄 USSD (API)
7. 目錄
8. 目錄
9. 目錄
10. 目錄
11. 目錄

目錄

USSD 目錄 USSD 目錄

- 目錄 (目錄) — 目錄 *100# 目錄 MAP processUnstructuredSS-Request 目錄 59 目錄 HTTP 目錄 SS7 目錄
- 目錄 (目錄) — 目錄 REST API 目錄 USSD 目錄 SS7 目錄 MAP unstructuredSS-Request 目錄 60 目錄

目錄 目錄 — 目錄

□□□□

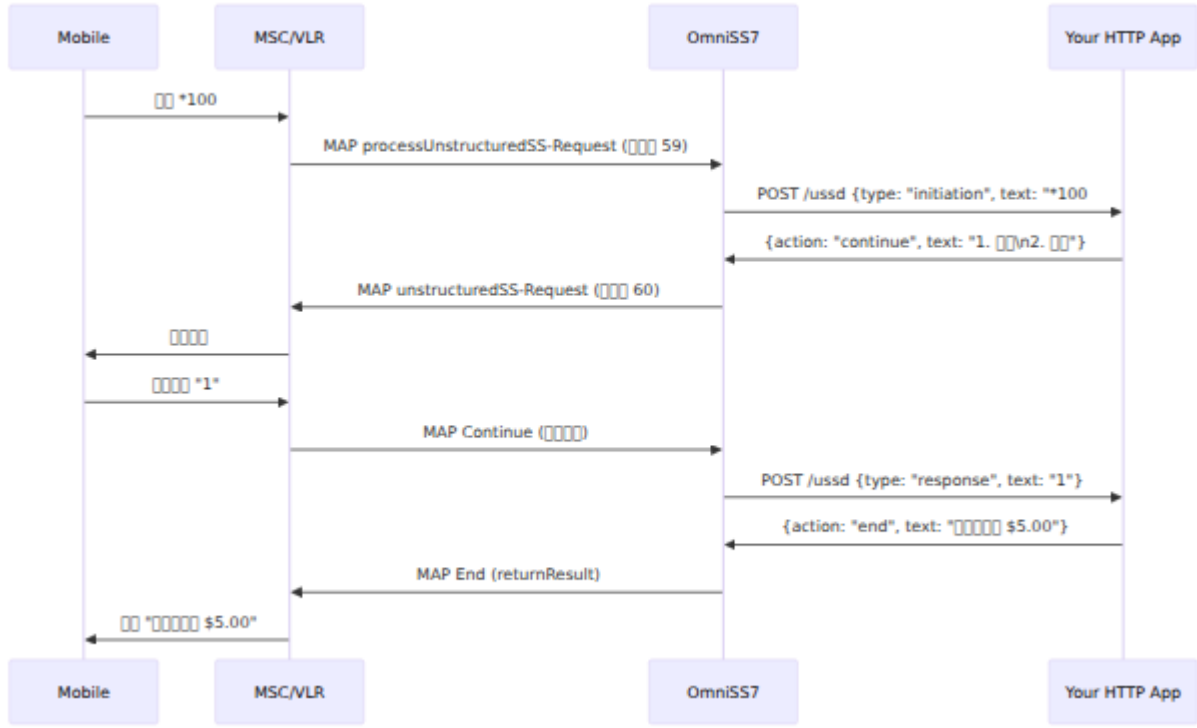
□□	□
□□	□□□□ HTTP POST
□□	GSM 7 □□□□□□ (DCS 0x0F) □□ 3GPP TS 23.038
□□□□□□	182 □□□□□□□□
□□□□	□□□□□□□□□□ UUID
□□	□□□□ SS7 □□□
□□	□□□□□□□□□□ URL

3GPP □□

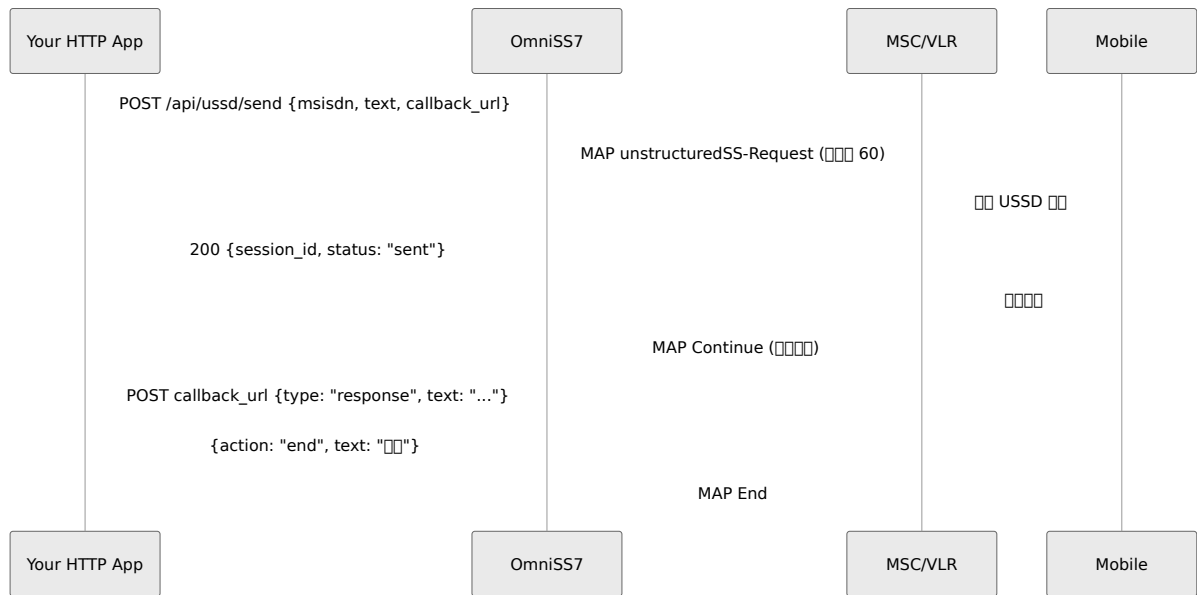
□□	□□□
3GPP TS 23.090	USSD □□□□ — □□□□□
3GPP TS 24.090	USSD □□□□ — □□□□□
3GPP TS 29.002	MAP □□ — USSD-Arg, USSD-Res, □□□ 59/60/61
3GPP TS 23.038	GSM 7 □□□□□□□□□□□□□□

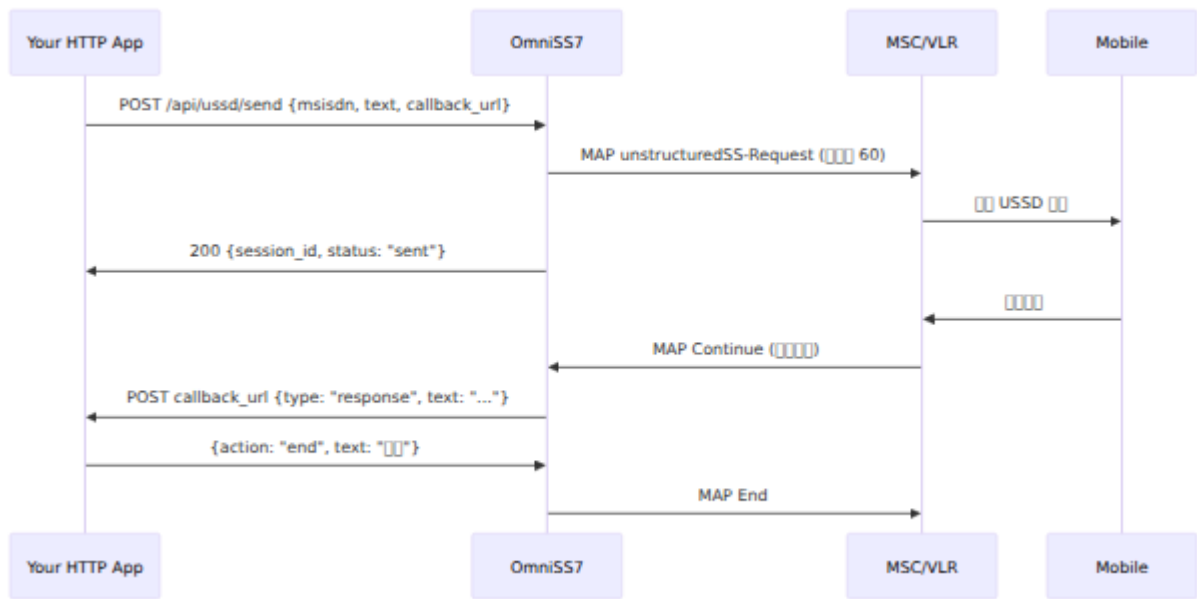
□□

□□□□□□ (□□)



□□□□□□ (□□□□□)





[[USSD]]

USSD [[[[[[**MAP**]]]]]]]]]]]]]]]]]]]

```
config :omniss7,
  map_client_enabled: true,
  ussd_gateway_enabled: true
```

]]]]]]]]]]]] M3UA]]]]]]] **MAP**]]]]]]]]]] M3UA]]]]]



USSD 配置

```
config :omniss7,  
  ussd_gateway_enabled: true,  
  ussd_gateway: %{  
    # 配置 - 配置  
    routes: [  
      %{pattern: "*100", url: "http://balance-app:9000/ussd"},  
      %{pattern: "*200", url: "http://topup-app:9000/ussd"},  
      %{pattern: "*", url: "http://default-app:9000/ussd"}  
    ],  
  
    # 配置  
    session_timeout_ms: 180_000, # 配置3000  
    turn_timeout_ms: 30_000, # 配置3000  
  
    # HTTP 配置  
    http_timeout_ms: 5_000, # 配置 HTTP POST 500  
  
    # 配置  
    max_text_length: 182 # GSM 7 配置
```

配置

配置	数据类型	默认值	说明
<code>ussd_gateway_enabled</code>	布尔	<code>false</code>	是否启用 USSD 网关
<code>ussd_gateway.routes</code>	列表	<code>[]</code>	路由配置，每个元素是一个字典，包含 <code>pattern</code> 和 <code>url</code> 两个键，分别表示路由模式和 URL
<code>ussd_gateway.session_timeout_ms</code>	整数	<code>180_000</code>	会话超时时间，单位为毫秒
<code>ussd_gateway.turn_timeout_ms</code>	整数	<code>30_000</code>	转接超时时间，单位为毫秒
<code>ussd_gateway.http_timeout_ms</code>	整数	<code>5_000</code>	HTTP 请求超时时间，单位为毫秒
<code>ussd_gateway.max_text_length</code>	整数	<code>182</code>	USSD 消息最大文本长度

路由

`routes` 路由配置


```
{
  "session_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "msisdn": "+254712345678",
  "type": "response",
  "text": "1",
  "turn": 2
}
```

欄名

欄名	型	説明
session_id	文字列	セッションの UUID。USSD のセッション固有の識別子
msisdn	文字列	送信元 MSISDN。MAP のセッション固有の識別子
type	文字列	メッセージの種類。"initiation" または "response"
text	文字列	メッセージの本文。*100# のメッセージの場合は 1 桁
turn	整数	1 回目のメッセージ以降のメッセージの回数

応答メッセージ

応答メッセージは JSON で送信され、action と text が含まれる。

応答メッセージの例

```
{
  "action": "continue",
  "text": "1. 123\n2. 456\n3. 789"
}
```

応答メッセージの例

```
{
  "action": "end",
  "text": "□□□□ $5.00"
}
```

--	--	--	--

項目	形式	値	説明
action	文字列	空文字列	"continue" を返すと、再度質問を生成する。 "end" を返すと、生成を終了する。
text	文字列	空文字列	生成されたテキスト。 max_text_length (182) を超えないようにする。 \n を含む。

□□□□□ **USSD (□□ API)**

USSD

11

POST /api/usss/send

11

```
{
  "msisdn": "+254712345678",
  "text": "XXXXXXXXXXXX 1 XXXXX",
  "callback_url": "http://billing-app:9000/usssd"
}
```

Parameters

Field	Type	Required	Description
msisdn	String	Yes	MSISDN of the recipient
text	String	Yes	USSD message in GSM 7-bit encoding
callback_url	String	No	URL to call back after the message is sent

Response

200 OK:

```
{
  "session_id": "xyz-789-abc-123",
  "status": "sent"
}
```

Errors

HTTP Status	Error Message	Details
400	<code>{"error": "invalid request", "required": ["msisdn", "text", "callback_url"]}</code>	Missing required fields
400	<code>{"error": "gsm7_encode_failed", ...}</code>	USSD message not in GSM 7-bit encoding
500	<code>{"error": "send_failed", ...}</code>	M3UA configuration error
503	<code>{"error": "USSD gateway not enabled"}</code>	ussd_gateway_enabled is false

cURL

```
curl -X POST http://localhost:8080/api/uszd/send \  
-H "Content-Type: application/json" \  
-d '{  
  "msisdn": "+254712345678",  
  "text": "XXXXXXXXXXXXXXXX 1 XXXXX",  
  "callback_url": "http://billing-app:9000/uszd"  
}'
```

XXXXXXXXXX

USSD XXXXXXXXXXXX session_id XXXX

XXXX



MAP XXX 59

Begin

XXXX "continue"

WaitingForReply

XXXX "end" XXXX/XX

XXXX "continue"
XXXX (MAP Continue)

XXXX XXXX XXXXX

Processing

XXXX "end"
XXXX/XX



MAP End

MAP End 消息的 "end" 属性

MAP Continue

MAP Continue 消息的 "continue" 属性

- 1. 在 `UssdGateway.Registry` 中注册 **GenServer**
- 2. 在 60 秒内发送 `unstructuredSS-Request` 消息 MAP Continue
- 3. 设置 `turn_timeout_ms` 属性
- 4. 设置 `turn_timeout_ms` 属性
- 5. 在消息的 "end" 属性中设置

MAP 消息

消息	属性	说明
MAP End	30 秒	在 30 秒内发送 MAP End 消息
MAP Continue	3 秒	在 3 秒内发送 MAP Continue 消息
HTTP 消息	5 秒	在 5 秒内发送 HTTP 消息 MAP 消息

MAP 消息

MAP 消息的 "MAP" 属性

項目	エラーメッセージ	MAP コード
HTTP ステータス 5xx	サーバーエラー MAP End	34 (systemFailure)
不正な JSON	JSON 解析エラー MAP End	34 (systemFailure)
USSD メッセージ max_text_length	メッセージ長超過エラー	N/A (メッセージ長超過)
データベース接続エラー	データベース接続エラー MAP End	34 (systemFailure)
メッセージ長超過	メッセージ長超過 MAP End	34 (systemFailure)
GenServer 再起動エラー	GenServer 再起動エラー	N/A (再起動)
USSD サービス未サポート	サービス未サポート	21 (facilityNotSupported)

監視

USSD サービスは `/metrics` エンドポイントで 8080 ポートで Prometheus にメトリクスを提供します。

USSD メトリクス

メトリクス: `ussd_requests_total`
単位: 回
ラベル: `direction` USSD サービス
方向:

- `direction` — `"inbound"` 受信メッセージ `"outbound"` 送信メッセージ

👉: `ussd_active_sessions`

👉: 📊

👉: 📊📊 USSD 📊📊

👉: `map_request_duration_milliseconds`

👉: 📊

👉: USSD 📊📊📊📊📊📊📊

👉:

- `operation` — `"ussd_send"` 📊📊📊📊📊

👉 Prometheus 📊

```
# 📊📊 USSD 📊📊  
rate(ussd_requests_total[5m])  
  
# 📊📊  
ussd_active_sessions  
  
# 📊 USSD 📊 (p95)  
histogram_quantile(0.95,  
rate(map_request_duration_milliseconds_bucket{operation="ussd_send"}  
[5m]))
```



Python (Flask)

```
from flask import Flask, request, jsonify

app = Flask(__name__)
sessions = {}

@app.route('/ussd', methods=['POST'])
def ussd():
    data = request.json
    session_id = data['session_id']
    text = data['text']
    turn = data['turn']

    if data['type'] == 'initiation':
        sessions[session_id] = {'state': 'main_menu'}
        return jsonify({
            'action': 'continue',
            'text': '1. 2. 3. '
        })

    state = sessions.get(session_id, {}).get('state')

    if state == 'main_menu':
        if text == '1':
            del sessions[session_id]
            return jsonify({
                'action': 'end',
                'text': '$5.00'
            })
        elif text == '2':
            sessions[session_id]['state'] = 'buy_airtime'
            return jsonify({
                'action': 'continue',
                'text': ''
            })
        else:
            del sessions[session_id]
            return jsonify({
                'action': 'end',
```

```

        'text': '👉👉👉👉👉👉'
    })

elif state == 'buy_airtime':
    del sessions[session_id]
    return jsonify({
        'action': 'end',
        'text': f'👉👉👉👉👉👉'
    })

return jsonify({'action': 'end', 'text': '👉👉👉👉👉👉'})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=9000)

```

Node.js (Express)

```
const express = require('express');
const app = express();
app.use(express.json());

const sessions = new Map();

app.post('/ussd', (req, res) => {
  const { session_id, text, type } = req.body;

  if (type === 'initiation') {
    sessions.set(session_id, { state: 'main_menu' });
    return res.json({
      action: 'continue',
      text: '0000\n1. 0000\n2. 0000'
    });
  }

  const session = sessions.get(session_id);
  if (!session) {
    return res.json({ action: 'end', text: '000000' });
  }

  if (session.state === 'main_menu' && text === '1') {
    sessions.delete(session_id);
    return res.json({ action: 'end', text: '000000 $5.00' });
  }

  sessions.delete(session_id);
  return res.json({ action: 'end', text: '0000' });
});

app.listen(9000, () => console.log('USSD 000000 9000 0000'));
```

問題

USSD 機能の検証

問題: USSD 機能が正常に動作しない

原因:

- `ussd_gateway_enabled` が `false`
- 設定が正しくありません
- M3UA 設定が正しくありません

解決:

1. 設定ファイルに `ussd_gateway_enabled: true`
2. 設定ファイルの権限を `* 777` に変更
3. Web UI から M3UA 設定を確認

問題

問題: USSD 機能が正常に動作しない HTTP POST

原因:

- URL が OmniSS7 ではありません
- 設定が OmniSS7 ではなく HTTP
- 設定が正しくありません

解決:

1. 設定ファイルに `curl -v http://your-app:9000/ussd` が OmniSS7
2. 設定ファイルの権限を `* 777` に変更
3. 設定ファイルの権限を `* 777` に変更

問題

問題: "systemFailure" エラーが発生

原因:

- `turn_timeout_ms` 参数设置不正确
- `http_timeout_ms` 参数设置不正确
- OmniSS7 参数设置不正确

解决:

1. 将 `turn_timeout_ms` 参数 30 秒设置正确
2. 将 `http_timeout_ms` 参数设置正确
3. 将 OmniSS7 参数设置正确

GSM 7 编码

错误: 收到 `gsm7_encode_failed` 错误 /api/usd/send 返回 400 错误

原因:

- 使用 GSM 7 编码的字符集包含 CJK 字符

解决:

- 将 USSD 消息中的 GSM 编码转换为 ASCII 编码/转码
- 参考 [3GPP TS 23.038 6.2.1](#) 章节

部署

- **API** 包 — 提供 REST API 接口 `/api/usd/send`
- **MAP** 包 — USSD 消息 M3UA 接口
- **配置** — 配置文件
- **监控** — Web UI 和 Prometheus 集成

Web UI 簡介

← 目錄

OmniSS7 Web UIPhoenix LiveView 簡介

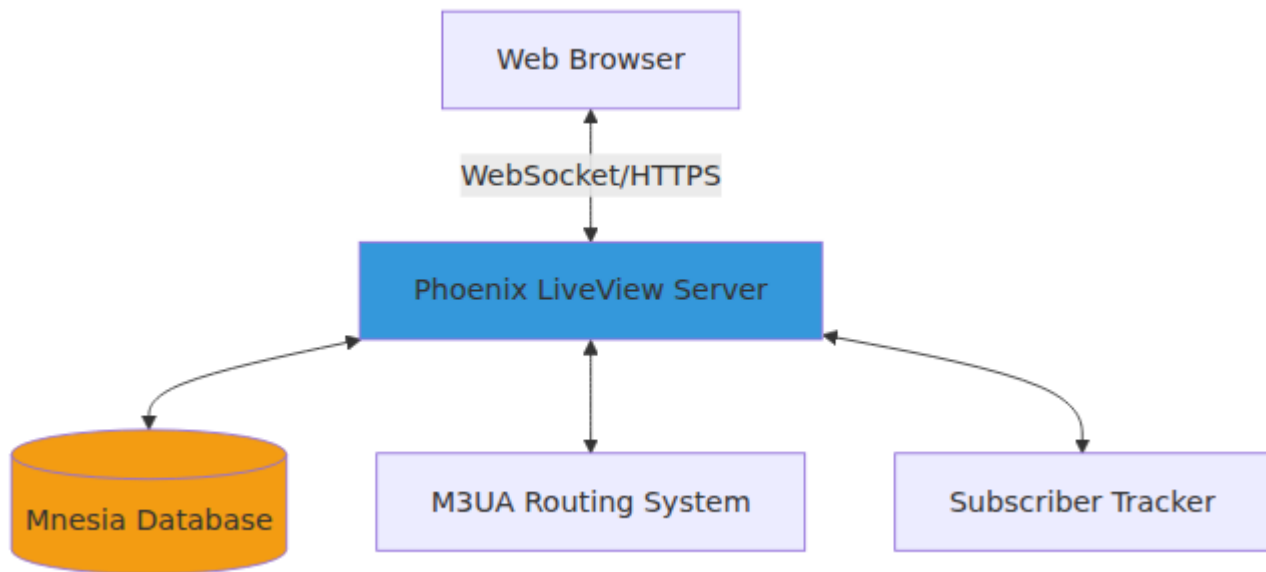
目錄

1. 簡介
2. Web UI
3. 安裝
4. 配置
5. 使用
6. 其他

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
- Port: `priv/cert/`

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

Web UI

Configuration

1. **SSL** Port: `priv/cert/` SSL Port
 - `omnitouch.crt` - Certificate
 - `omnitouch.pem` - Private Key
2. Port: `iex -S mix` port
3. Port: Port 443 HTTPS port

□□□□□□□

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

□□□□

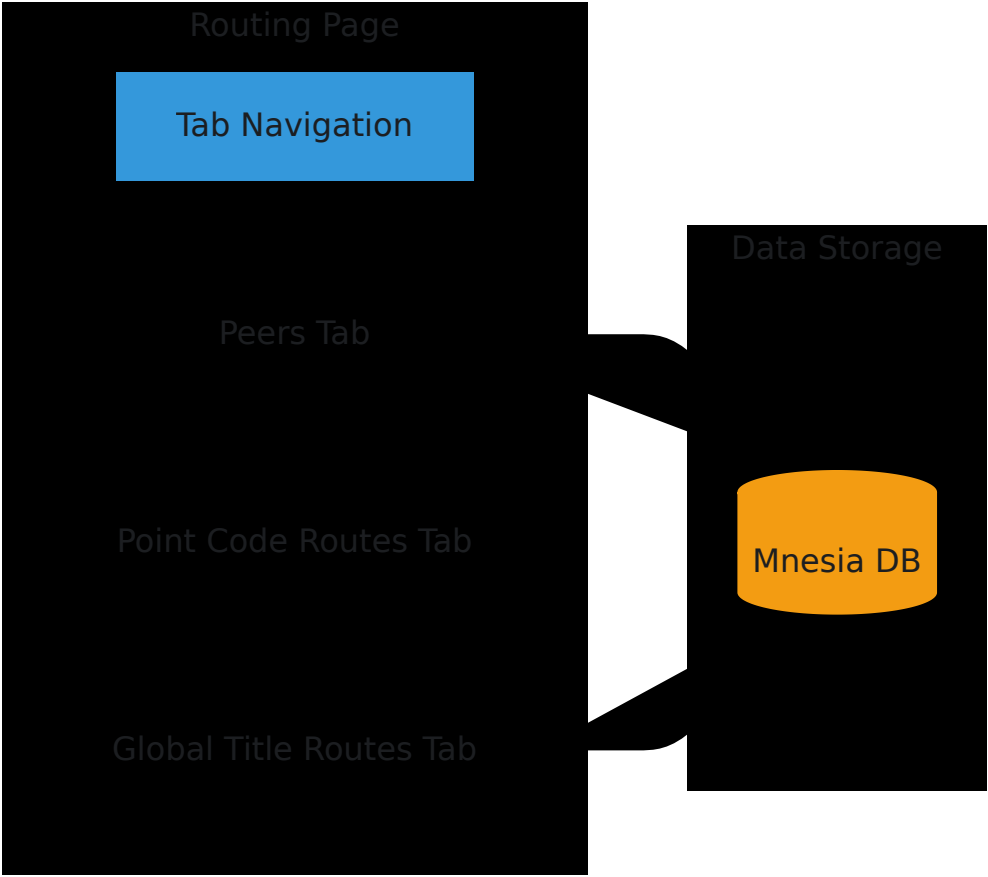
□□: /routing

□□: STP, SMSc

□□□□: □ 5 □

□□□□□□□□□□□□□□□□ M3UA □□□□

□□□□



Peers □□□

□□ M3UA □□□□□□ STP□HLR□MSC□SMSC□□

□□□□

項目	項目名	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 識別子
 - コメント: international, national
3. "Add Peer"

注: Mnesia 設定

手順

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

00: 000000 ID000000000000000000

□□□□

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

□□□□□□□

項目	項目	項目
active	項目	項目項目項目項目項目項目項目項目
aspup	項目	ASP 項目項目項目項目項目項目項目項目
down	項目	項目項目項目項目項目項目項目項目

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

□□□□□□□□□□□□

□□□□

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

1. "Point Code Routes"
2.
 - zone.area.id 1.2.3 0-16383
 - : /14
 - ID:
 - : 1 =
 - : international national
3. "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	

IP:

- PC 1000 /14 → PC 1000
- PC 1000 /8 → PC 1000-1063 64 PCs
- PC 0 /0 →

IP SCCP

IP:

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


SSN	名前
6	HLR (ホームランダム)
7	VLR (バリエーションランダム)
8	MSC (移動交換機)
9	EIR (エラーインジケータランダム)
10	AUC (アクセスユーザランダム)
142	RANAP
145	gsmSCF (グローバルシステムランダム)
146	SGSN

SSN 名前

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

例: **SSN**=6ホームHLRランダム SSN=7バリエーションVLR

インストール

インストール **Mnesia** ランタイムランタイムランタイム

インストール

1. **Web UI** 例: `config/runtime.exs` Mnesia
2. ランタイム: `runtime.exs` Mnesia ランタイム
3. **Runtime.exs** 例: `config/runtime.exs` ランタイム Mnesia ランタイム

🔗🔗

📄📄📄📄📄📄

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

📄📄📄

📄: /subscribers

📄: 📄 HLR

📄📄: 📄 2 📄

📄📄📄📄📄 UpdateLocation 📄📄📄📄

📄📄📄

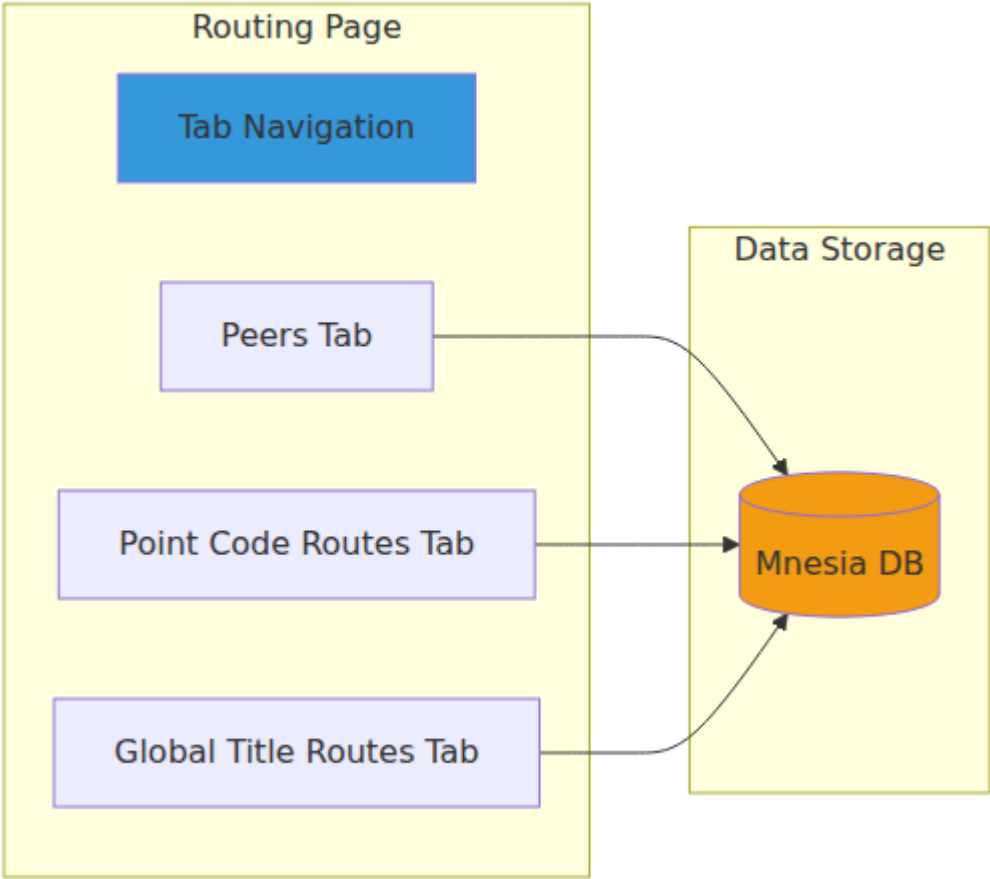


Table 1

Field	Label	Value
IMSI	IMSI	"50557123456789"
VLR ID	VLR GT ID	"555123155"
MSC ID	MSC GT ID	"555123155"
Timestamp	UpdateLocation Time	"2025-10-25 14:23:45 UTC"
Duration	Call Duration	"2h 15m 34s"

Table 2

Table 2: Call Details

- **Call ID:** 123456789
- **VLRs:** VLR 1, VLR 2
- **MSCs:** MSC 1, MSC 2

Table 3

Table 3: Location Data

Table 3: Location Data

Table 3: Location Data

Table 4

Table 4: Call Summary

SMSc 接口

接口: /smc_subscribers

接口: SMSc

接口: 2 个

接口 HLR 接口 alertServiceCenter 接口

接口

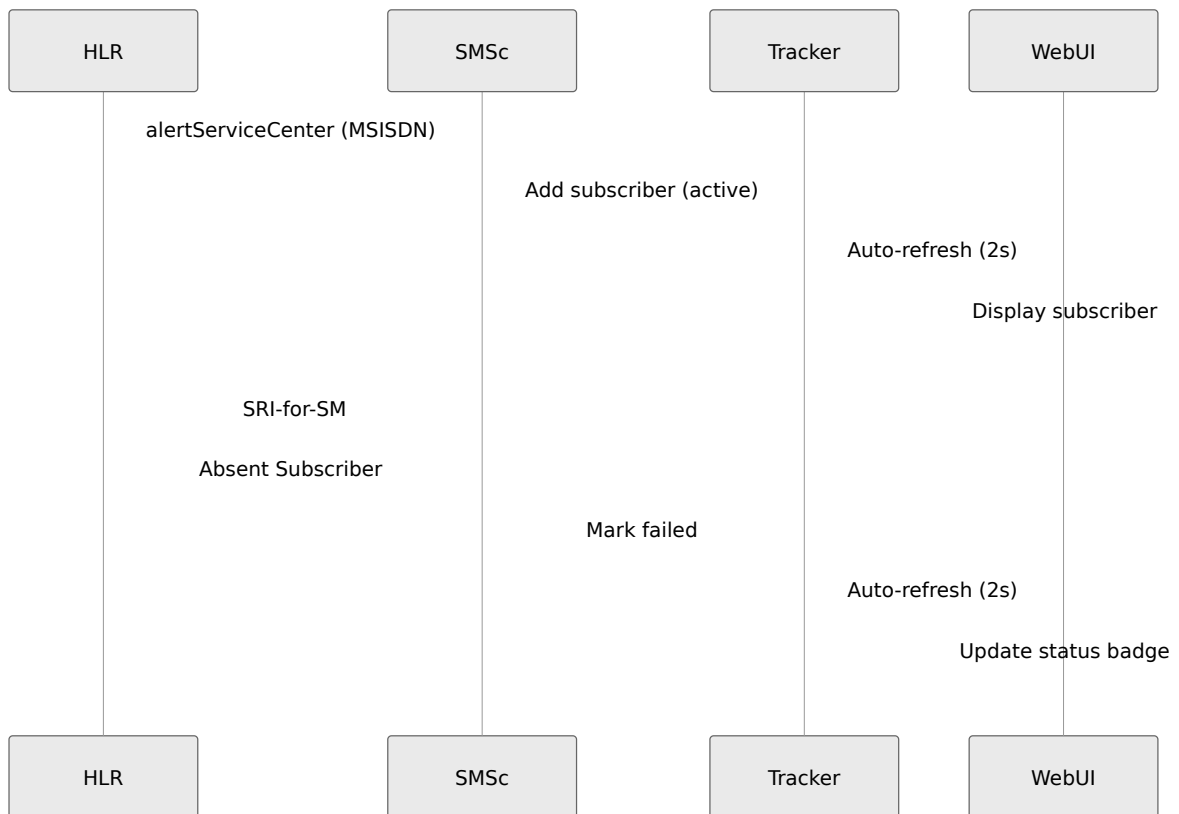


Table 1

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"

Table 2

- **Active** (Status): The status of the alertServiceCenter is Active.
- **Failed** (Status): The status of the SRI-for-SM is Failed.

Table 3

Table 4

- Field: MSISDN
- Field: IMSI
- Field: HLR GT
- Field: HLRs: The HLRs are listed below.

□□□□

□□□□: □□□□□□□□□□

□□□□□□: □□□□□□□□□□

□□: □□□□□□□□□□□□□□

□□:

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□ 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 証明書を読み込む

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 を有効にする

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 を有効にする

□ □□□□ - □□□□□□□□□□

□ □□□□ - Mnesia □□□□□□□□□□□□

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

□ □□□□□ - □□□□□□□□□□□□□□

□ □□□□ - □□□□□□□□□□

□□□□□□□□□□□□□□ **API** □□□

