

REST API 簡介

← 目錄

OmniSS7 REST API 與 Swagger UI 簡介

目錄

1. 簡介
2. HTTP 協議
3. Swagger UI
4. API 簡介
5. 安裝
6. 配置
7. 測試
8. 部署
9. 監控 (Prometheus)
10. 總結

簡介

OmniSS7 是一個 **9090** 端口的 REST API，支持 SS7 協議，MAP 協議，CAP 協議，CAMEL 協議，SCCP/M3UA 協議。API 地址 **47** 端口 **POST /api/*** 接口，支持 Swagger 接口。

- 支持 MAP 協議，支持 GPRS 協議
- 支持 CAP/CAMEL 協議
- 支持 SCCP/M3UA 協議
- 支持 Prometheus 監控

OmniSS7/SS7 MAP 協議，MAP 協議，MAP 協議，MAP 協議

API 概要

OmniSS7 は **HTTP** を通じて **9090** 番ポートで API を提供します。

ポート	プロトコル	エンドポイント	機能
9090	HTTP (Plug.Cowboy)	APIhandler	MAP/CAP REST API /api/* /metrics /
8087	HTTPS (Bandit, Phoenix)	ControlPanelWeb.Endpoint	Web UI
8445	HTTPS / TLS	:api_ex REST API	REST API

API 概要



HTTP API

API 概要

/api/* /metrics /swagger Plug.Cowboy HTTP API

項目	値	説明
プロトコル	HTTP	API 提供プロトコル
IP アドレス	0.0.0.0	すべての IP アドレスにバインド
ポート	9090	API config :omniss7, api_port: <port> 9090
プラグイン	Plug.Cowboy	API 提供プラグイン

API URL `http://[server-ip]:9090`

API 提供ポート: 8087 Bandit API :api_ex REST API 提供ポート: 8445

API 提供 HTTP サーバ

API :9090 HTTP サーバ

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

API true

API :9090 HTTP REST API / Swagger UI / API 提供ポート: 8087
:api_ex :8445 start_web_interfaces

Swagger UI

API 提供 Swagger UI API 提供

Swagger UI

URL `http://[server-ip]:9090/swagger`

000

- API
-
- /
-

Swagger JSON

OpenAPI

URL `http://[server-ip]:9090/swagger.json`

111

- Postman API
-
- API

API ☐ ☐

POST /api/{operation} JSON SS7
10 MAP

MAP

000000000000 200 = 000 400 = 00/00000000000000000000 504 = 000000000000
 0 SS7 0000 500 = 000000000000 503 = 00000000USSD00 404 = 00000000 000000

接口

接口	接口	接口
接口	POST /api/updateLocation	接口 VLR 接口 HLR接口 HLR 接口 接口 InsertSubscriberData (ISD) 接口
接口	POST /api/cancelLocation	接口 VLR 接口
接口 接口	POST /api/prn	接口 MSC 接口 MSRN
接口 MS	POST /api/purgeMS	接口 HLR 接口
接口	POST /api/sendIdentification	接口 VLR 接口 IMSI/接口TMSI 接口
接口	POST /api/reset	HLR 接口 VLR接口 { "status": "sent" } 接口
接口	POST /api/restoreData	VLR 接口 HLR 接口

API

API	API	API
POST /api/sri-for-sm	POST /api/sri-for-sm	HLR SMS
MT-SM	POST /api/MT-forwardSM	SM MSC/SGSN smsPDU
MO-SM	POST /api/forwardSM	SM smsPDU
POST /api/sendSM	POST /api/sendSM	GSM-7 SMS-DELIVER
POST /api/deliverPDU	POST /api/deliverPDU	SMS-DELIVER PDU SS7
POST /api/reportSM-DeliveryStatus	POST /api/reportSM-DeliveryStatus	HLR SM
POST /api/readyForSM	POST /api/readyForSM	HLR SMS
POST /api/alertServiceCentre	POST /api/alertServiceCentre	SMSC

API

API	API	API
POST /api/send-auth-info	POST /api/send-auth-info	HLR
POST /api/sendIMSI	POST /api/sendIMSI	MSISDN HLR IMSI

□□□□

□□	□□	□□
□□□□□ □	POST /api/sri	□□ HLR □□□□□□□□□□ □□□
□□□□□ □	POST /api/provideSubscriberInfo	□ VLR □□□□□□□/□□□
□□□□□ □	POST /api/deleteSubscriberData	□ VLR □□□□□□□□□□□□ withdraw □□□□
□□□□□ □	POST /api/anyTimeInterrogation	gsmSCF □□ HLR □□ □□□□□/□□□
□□□□□ □□□	POST /api/anyTimeSubscriptionInterrogation	□□ HLR □□□□□□□□
□□□□□ □	POST /api/anyTimeModification	□ HLR □□□□□□□□□
□□□□□ □□□□	POST /api/noteSubscriberDataModified	HLR □□ gsmSCF □□ □□□□
□□ MM □ □	POST /api/noteMM-Event	□ gsmSCF □□□□□□□□ □□

接口

接口	接口	接口
SS 接口	POST /api/interrogateSS	SS 接口
SS 接口	POST /api/registerSS	SS 接口 / CFU 接口
SS 接口	POST /api/eraseSS	SS 接口
SS 接口	POST /api/activateSS	SS 接口
SS 接口	POST /api/deactivateSS	SS 接口
SS 接口	POST /api/registerPassword	SS 接口 ss_code 接口
SS 接口	POST /api/getPassword	SS 接口 SS 接口
SS 接口	POST /api/forwardCheckSS-Indication	VLR 接口 {"status": "sent"} 接口

SS 接口 interrogateSS 接口 registerSS 接口 eraseSS 接口 activateSS 接口 deactivateSS 接口 ss_code 接口 hlr_gt 接口 forwarded_to 接口 no_reply_time 接口 basic_service 接口

位置 (LCS)

名称	接口	说明
向 LCS 提供位置信息	POST /api/sendRoutingInfoForLCS	GMLC 向 HLR 提供 LCS 信息
提供位置信息	POST /api/provideSubscriberLocation	提供位置信息
报告位置信息	POST /api/subscriberLocationReport	向 GMLC 报告位置信息

GPRS

名称	接口	说明
向 GPRS 提供位置信息	POST /api/updateGprsLocation	向 SGSN 提供 HLR 位置信息
向 GPRS 提供位置信息	POST /api/sendRoutingInfoForGprs	向 HLR 提供 GPRS 位置信息

其他

名称	接口	说明
检查 IMEI	POST /api/checkIMEI	检查 EIR 是否黑名单

USSD

API	Method	Response
USSD Send	POST /api/ussd/send	200 OK ussd_gateway_enabled: true 503 Service Unavailable

CAP / CAMEL

API	Method	Response
CAP InitialDP	POST /api/cap/initialDP	200 OK otid
CAP Connect	POST /api/cap/connect	200 OK destination_number + otid
CAP Continue	POST /api/cap/continue	200 OK otid
CAP ReleaseCall	POST /api/cap/releaseCall	200 OK cause + otid
CAP ApplyCharging	POST /api/cap/applyCharging	200 OK duration + otid

CAP InitialDP otid initialDP {"status":"sent","otid":"<hex>"} CAMEL

API Endpoints

API	Method	Description
SCCP	POST <code>/api/raw/sccp</code>	Retrieve SCCP configuration parameters for SSNs and OPC/DPC
M3UA	POST <code>/api/raw/m3ua</code>	Retrieve M3UA configuration parameters for OPC/DPC

API Endpoints

API	Method	Description
<code>/metrics</code>	GET	Prometheus metrics endpoint
<code>/swagger</code>	GET	Swagger UI
<code>/swagger.json</code>	GET	OpenAPI specification

API

API endpoint for sending routing information (SRI) to the MAP server

SendRoutingInfo (SRI)

API endpoint for sending routing information (SRI)

Method: POST `/api/sri`

API

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

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

項目200 OK項目

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

cURL 項目

```
curl -X POST http://localhost:9090/api/sri \
-H "Content-Type: application/json" \
-d '{"msisdn": "1234567890", "gmsc": "5551234567"}'
```

SendRoutingInfoForSM (SRI-for-SM)

項目 SMS 項目項目項目項目項目項目

項目 POST /api/sri-for-sm

項目項目


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

項目	項目	項目	項目
msisdn	項目	項目	項目 MSISDN
service_center	項目	項目	項目項目項目

cURL 項目

```
curl -X POST http://localhost:9090/api/sri-for-sm \
-H "Content-Type: application/json" \
-d '{"msisdn": "1234567890", "service_center": "5551234567"}'
```

UpdateLocation

項目 HLR 項目項目項目VLR 項目項目 HLR 項目項目項目 InsertSubscriberData (ISD) 項目

項目 POST /api/updateLocation

項目項目

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

Field	Type	Length	Description
imsi	String	15	IMSI
vlr	String	15	VLR Number

USSD

Enable USSD gateway: `ussd_gateway_enabled: true`

USSD

POST /api/ussd/send

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

Field	Type	Length	Description
msisdn	String	15	MSISDN
text	String	15	USSD GSM 7
callback_url	String	15	URL HTTP POST

200 OK

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

503

```
{
  "error": "USSD gateway not enabled"
}
```

API

- API
-
-

JSON

HTTP 200 OK

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

□□□□

HTTP □□□

- 400 Bad Request - □□□□□□□□
- 500 Internal Server Error - □□□□□SS7 □□□□□□□□□□
- 503 Service Unavailable - □□□□□□USSD □□□
- 504 Gateway Timeout - □□□□□□□ 10 □□
- 404 Not Found - □□□□

□□□

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

□

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

0000

0000

00	HTTP 00	00	0000
00 JSON	400	00000000 JSON	00 JSON 00
00/00 00	400	000000000000000000000000 "required" 000	00000000000000000000
0000	500	SS7 0000000000	00 "error" 00000 SS7 0000
0000	503	USSD 00000	00 ussd_gateway_enabled: true
00	504	0000000000 10 00	00 M3UA 000HLR/VLR 000
000	404	0000	0000 URL

0000

0000000000 **10** 0000

1. 00000 MapClient GenServer
2. 000000 10 0
3. 000000 → 00 504 Gateway Timeout
4. 000000 → 00 200 OK 000

00000000

- 00 M3UA 00000Web UI → M3UA 000
- 00000000HLR/VLR/MSC00000
- 0000000

- SS7

(Prometheus)

API Prometheus

URL `http://[server-ip]:9090/metrics`

Prometheus

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

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

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

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

# HELP omniss7_license_status Current license status (1 = valid, 0
= invalid)
# TYPE omniss7_license_status gauge
omniss7_license_status 1
```

□ □ □ □

[illegible]

MAP / CAP / USSD /

項目	単位	種別	説明
map_requests_total	回 分	operation	MAP 要求の総数
map_request_errors_total	回 分	operation	MAP 要求のエラーの総数
cap_requests_total	回 分	operation	CAP 要求の総数
cap_request_errors_total	回 分	operation	CAP 要求のエラーの総数
map_request_duration_milliseconds	ミリ秒 分	operation	MAP 要求の平均実行時間
map_pending_requests	回 分	-	現在 MAP 待ちの要求数
ussd_requests_total	回 分	direction	USSD 要求の総数/方向
ussd_active_sessions	回 分	-	現在 USSD 中のセッション数
omniss7_license_status	回 分	-	ライセンスステータス 1 = 有効 0 = 無効

SCCP

項目	単位	説明	備考
sccp_messages_total	メッセージ数 [?] [?] [?] [?]	message_type, direction	SCCP [?] udt/xudt/ [?] sent/r
sccp_decode_errors_total	エラー数	-	SCCP
sccp_segments_received_total	セグメント数	-	XUDT/LU
sccp_reassembly_completed_total	セグメント数	-	SCCP
sccp_reassembly_timeouts_total	エラー数	-	

TCAP

項目	単位	説明	備考
tcap_dialogues_total	対話数 [?] [?] [?]	type	begin/continue/end/abc
tcap_aborts_total	エラー数 [?] [?] [?]	reason	TCAP P-Abort → unrecognized_message_type/un [?] p_abort_other [?] U-Abort → u
tcap_decode_errors_total	エラー数 [?] [?] [?]	-	TCAP
tcap_active_dialogues	対話数 [?] [?] [?]	-	MAP

SCTP

項目	単位	項目	説明
sctp_association_state	個	remote_ip, remote_port	remote_ip1 = up[
sctp_association_state_changes_total	個	event	イベント comm_up/co
sctp_send_errors_total	個	reason	SCTP 送信エラー
sctp_heartbeat_sent_total	個	type	M3UA 送信
sctp_heartbeat_received_total	個	type	M3UA 受信

項目 / SMSG

項目	単位	項目	説明
sms_messages_total	個	direction, operation	送信方向 (mo/mt) と 送信操作 (submit/deliver) の SMS の総数
sms_delivery_failures_total	個	reason	SMS 送信失敗の理由
sms_tracked_subscribers	個	-	SubscriberTracker を 利用している端末の 数

M2PA/MTP2 機能

項目	単位	項目	説明
m2pa_messages_sent_total	個 個 個	link_name, point_code, adjacent_pc	M2PA 送信メッセージの総数 MTP3 送信メッセージの総数
m2pa_messages_received_total	個 個 個	link_name, point_code, adjacent_pc	M2PA 受信メッセージの総数 MTP3 受信メッセージの総数
m2pa_bytes_sent_total	個 個 個	link_name, point_code, adjacent_pc	M2PA 送信バイトの総数
m2pa_bytes_received_total	個 個 個	link_name, point_code, adjacent_pc	M2PA 受信バイトの総数
m2pa_link_state_changes_total	個 個 個	link_name, from_state, to_state	M2PA リンク状態変更の総数
m2pa_errors_total	個 個 個	link_name, error_type	M2PA エラーの総数

M3UA STP

指标	维度	属性	说明
m3ua_stp_messages_received_total	peer_name, point_code		M3UA 接收消息总数
m3ua_stp_messages_sent_total	peer_name, point_code		M3UA 发送消息总数
m3ua_stp_routing_failures_total	reason		STP 路由失败总数 no_route/no_gt_

Prometheus 配置

配置文件 prometheus.yml

```
scrape_configs:  
  - job_name: 'omniss7'  
    static_configs:  
      - targets: ['server-ip:9090']  
    metrics_path: '/metrics'  
    scrape_interval: 15s
```

□□□□

Python □□

```
import requests
import json

# SRI-for-SM □□
url = "http://localhost:9090/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:9090/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

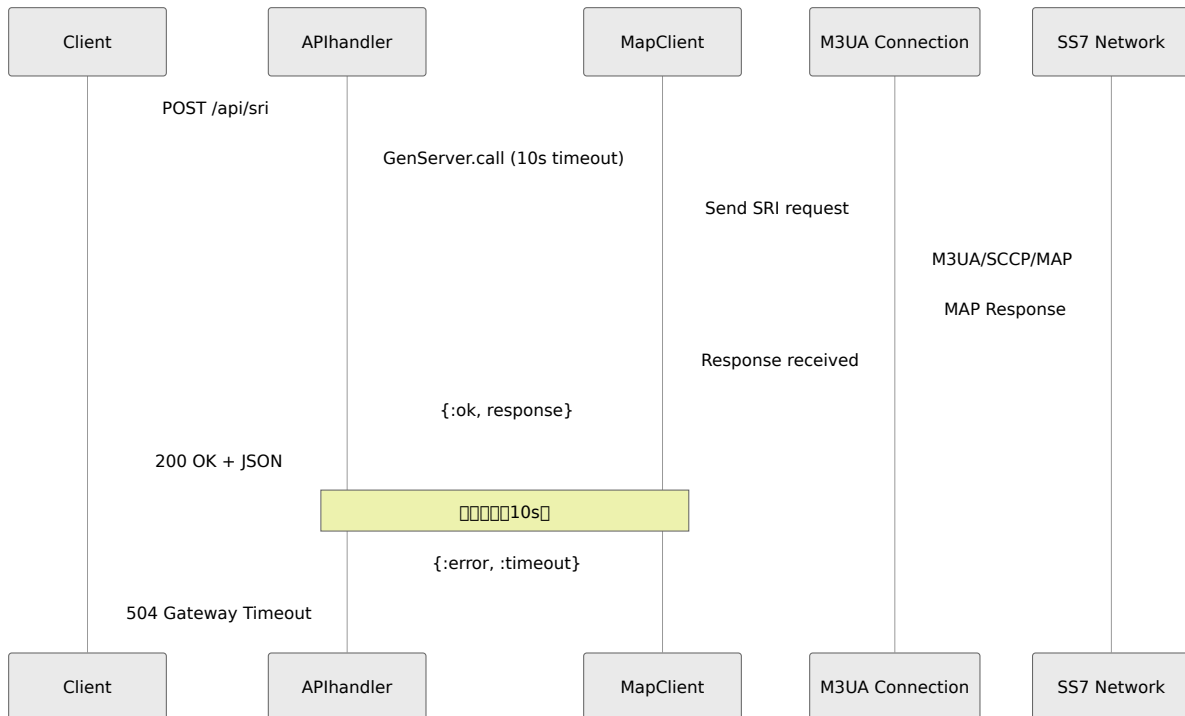
# UpdateLocation ☐☐
response=$(curl -s -w "\n%{http_code}" -X POST
http://localhost:9090/api/updateLocation \
-H "Content-Type: application/json" \
-d '{
    "imsi": "001001234567890",
    "vlr": "5551234800"
}')

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

- **47** 测试用例覆盖所有功能，包括GPRS、USSD、CAP/CAMEL 测试
- **Swagger UI** `/swagger` 测试接口文档
- **Prometheus** `/metrics` 测试接口文档
- **10** 测试用例覆盖所有功能，包括 504 测试
- 使用 `Plug.Cowboy` **HTTP** 测试接口 **9090** `start_http_server` 测试接口文档
地址: 8087 `:api_ex` REST 接口: 8445 测试接口

测试用例覆盖所有功能，包括 MAP 测试接口 **MAP** 测试接口

测试 Web UI 测试接口 **Web UI** 测试接口

□□□□□□□□□□ □□□□

□□□□□□□□

← □□□□

SS7□□□OmniSS7□□□□□□□□

SS7□□□



MAP□□□

□□□□□□□□3GPP TS 29.002□MAP local□□□□□□□□□□OmniSS7□□□/□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□**MAP□□□□□□**□

項目	番号	説明
updateLocation	2	位置更新CS
cancelLocation	3	VLR/SGSNからの位置更新取消
provideRoamingNumber	4	MSRN提供
noteSubscriberDataModified	5	gsmSCFからのサブスクリバデータ変更通知
insertSubscriberData	7	VLR/SGSNへのサブスクリバデータ挿入
deleteSubscriberData	8	VLR/SGSNからのサブスクリバデータ削除
registerSS	10	CFUへの登録
eraseSS	11	SSデータの削除
activateSS	12	SSのアクティベーション
deactivateSS	13	SSのデアクティベーション
interrogateSS	14	SSの照会
registerPassword	17	SSへのパスワード登録
getPassword	18	SSからのパスワード取得
sendRoutingInfo	22	SRIの送信
updateGprsLocation	23	SGSNからのGPRS位置更新
sendRoutingInfoForGprs	24	GGSNへのGPRS SRI送信
reset	37	HLRからのVLR/SGSNリセット
forwardCheckSS-Indication	38	SSチェック指示の転送

名前	番号	説明
checkIMEI	43	EIRデータベース参照
mt-forwardSM	44	SMSメッセージのMT転送
sendRoutingInfoForSM	45	SMSメッセージのSRI-for-SM送信
mo-forwardSM	46	SMSメッセージのMO転送
reportSM-DeliveryStatus	47	HLRからのSMS配信状況報告
sendIdentification	55	VLRからのIMSI/auth送信
sendAuthenticationInfo	56	認証情報の送信
restoreData	57	VLRからのデータ復元
sendIMSI	58	HLRからのMSISDNからのIMSI送信
processUnstructuredSS-Request	59	USSDメッセージの処理
unstructuredSS-Request	60	USSDメッセージの送信
unstructuredSS-Notify	61	USSDメッセージの通知
anyTimeSubscriptionInterrogation	62	gsmSCFからの任意時刻のサブスクリプション照会
alertServiceCentre	64	SMSセンターからのアラート
anyTimeModification	65	gsmSCFからの任意時刻の修正
readyForSM	66	HLRからのSMS準備完了通知
purgeMS	67	VLR/SGSNからのMSの削除
provideSubscriberInfo	70	サブスクリプション情報の提供

消息	消息	消息
anyTimeInterrogation	71	gsmSCF向GMLC/LCS
provideSubscriberLocation	83	GMLC向LCS
sendRoutingInfoForLCS	85	GMLC向LCS
subscriberLocationReport	86	GMLC向LCS
noteMM-Event	89	gsmSCF向GMLC

TCAP消息

- **BEGIN** - 开始
- **CONTINUE** - 继续
- **END** - 结束
- **ABORT** - 中止

SCCP消息

消息

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

消息SSN

- **SSN 6**: HLR
- **SSN 7**: VLR

- **SSN 8:** MSC/SMSC
 - **SSN 9:** GMLC
 - **SSN 10:** SGSN
-

SMS TPDU

TPDU

- **SMS-DELIVER** (MT) - 000000
- **SMS-SUBMIT** (MO) - 000000
- **SMS-STATUS-REPORT** - 0000
- **SMS-COMMAND** - 0000

TPDU

- **GSM7** - 7-bit GSM 000000 SMS 1600000
 - **UCS2** - 16-bit Unicode 0000 SMS 700000
 - **8-bit** - 8-bit 000000 SMS 140000
-

M3UA

- **DOWN** - SCTP 00
 - **CONNECTING** - SCTP 000
 - **ASPUP_SENT** - 00ASPUP 00
 - **INACTIVE** - ASP 00000000
 - **ASPAC_SENT** - 00ASPAC 00
 - **ACTIVE** - 000000
-

SS7

14 ITU 24 ANSI

ITU

- 3
- 8
- 3

SCCP

- 0 -
- 1 -
- 2 -
- 3 -
- 4 -
- 5 - MTP
- 6 -
- 7 -
- 8 -

MAP

項目	項目	項目
1	unknownSubscriber	HLR
27	absentSubscriber	
34	systemFailure	
35	dataMissing	
36	unexpectedDataValue	

- ←
- STP
- MAP
-
- HLR
-

OmniSS7Omnitouch

CAMEL 系統概要

概要

CAMEL 1 (CAMEL1) は OmniSS7 経由で IN (IN) から CAMEL 1 (CAP) 経由で OmniSS7 経由で **gsmSCF** (gsmSCF) から MSC/VLR/GMSC へ **gsmSSF** 経由で CGrateS 経由で

主要な CAMEL 1

CAMEL 1 (CAMEL 1) は GSM 経由で UMTS 経由で

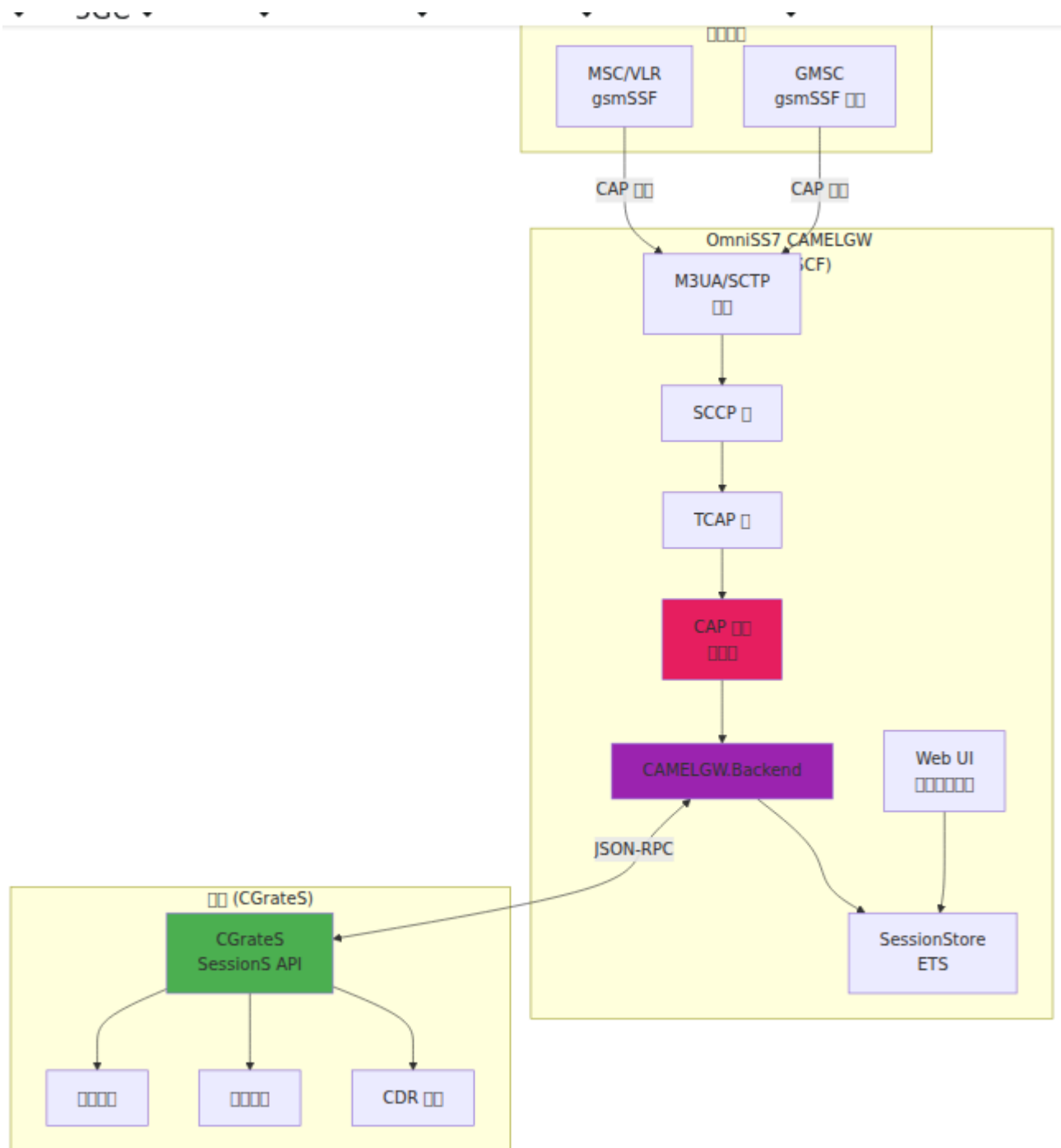
- 1 (1) - 1 (1)
- 1 (1) - 1 (1)
- 1 (1) - 1 (1)/1 (1)
- 1 (1) - 1 (1)
- 1 (1) - 1 (1)/1 (1)

CAP 表

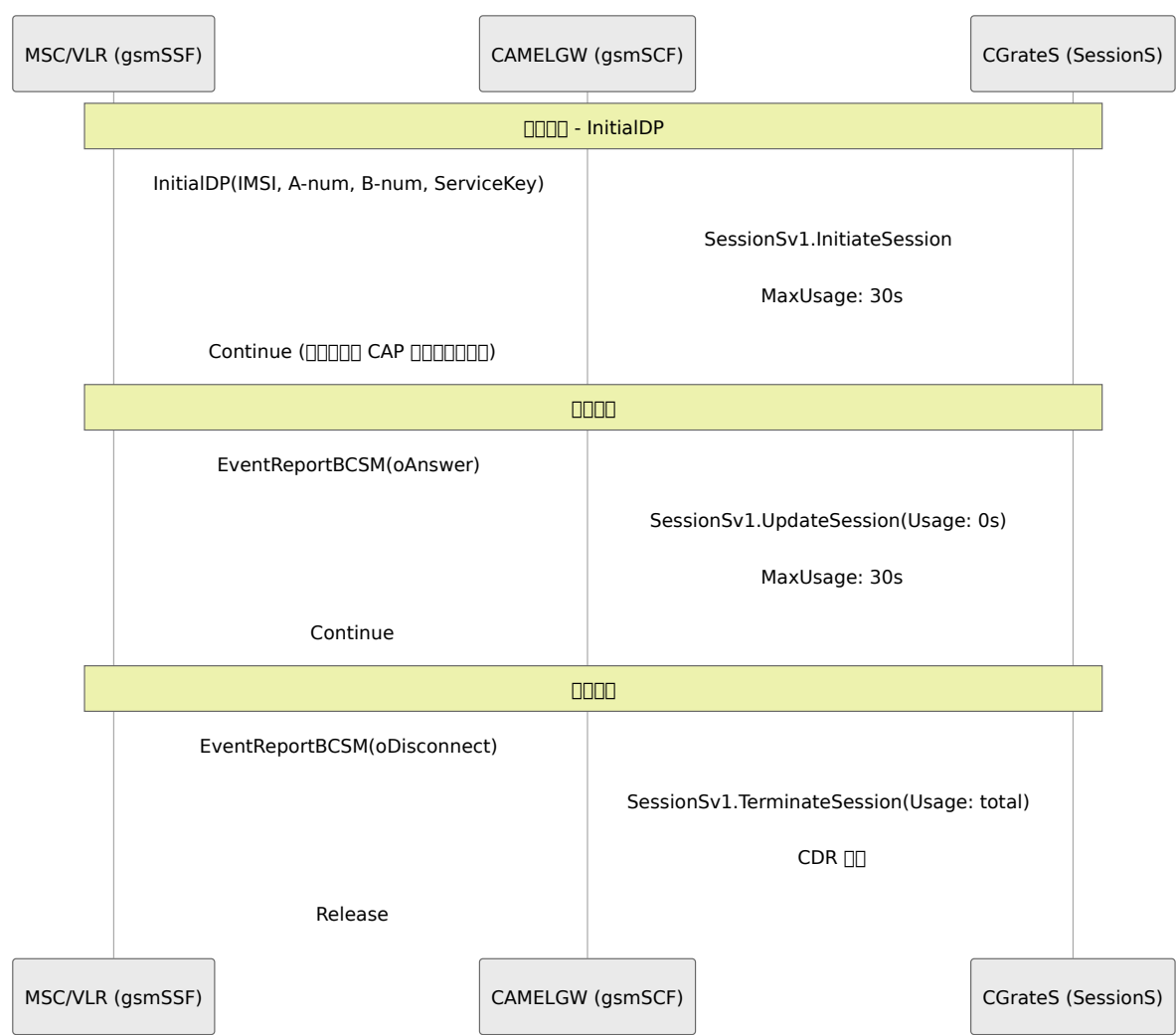
OmniSS7 CAMEL GW 表 CAP 表 TCAP 表 OID

表名	表名	OID
CAP v1	CAMEL 表 1	0.4.0.0.1.0.50.0
CAP v2	CAMEL 表 2	0.4.0.0.1.0.50.1
CAP v3	CAMEL 表 3	0.4.0.0.1.21.3.4
CAP v4	CAMEL 表 4	0.4.0.0.1.23.3.4

表名: CAP v2 (表名) InitialDP 表名 MSC 表 CAP 表



□□□□□□



□□

□□□□

- OmniSS7
- MSC/GMSC (gsmSSF) M3UA
- JSON-RPC CGrateS (CAP)

配置 CAMEL 参数

CAMEL 参数配置在 `cap_client_enabled` 中 CAP/M3UA 参数 CAMEL 参数 `cgrates_enabled` 配置 CGrateS 参数 `config/runtime.exs` 配置

```
config :omniss7,
  # 配置 CAP/CAMEL 参数 (M3UA ASP, SessionStore, 参数)
  cap_client_enabled: true,

  # 配置参数 (参数)
  map_client_enabled: false,
  smsc_mode_enabled: false,

  # 配置版本/参数 CAP 参数 (:v1 | :v2 | :v3 | :v4)
  cap_version: :v2,

  # CGrateS 参数
  cgrates_enabled: true,

  # CAP 与 M3UA 参数 (参数 ASP 参数 MSC/GMSC gsmSSF)
  cap_client_m3ua: %{
    routing_context: 1,
    opc: 5013,
    dpc: 5011
  }
```

配置 `cap_client_m3ua` 参数: CAMEL 参数 `:routing_context` `:opc` `:dpc` 参数 CAP 参数 M3UA ASP 参数 `:camelgw_client_asp`

配置 `camelgw_mode_enabled` `hlr_mode_enabled` `smsc_mode_enabled` 配置 CAMEL 参数 CAMEL 参数 `cap_client_enabled` `cgrates_enabled`

配置

項目	値	説明
<code>cap_client_enabled</code>	<code>false</code>	CAP/CAMEL クライアント (M3UA ASP, SessionStore, CAP 機能)
<code>cgrates_enabled</code>	<code>false</code>	CAMELGW.Backend の CGRateS 機能
<code>cap_version</code>	<code>:v2</code>	使用する CAP のバージョン/OID
<code>map_client_enabled</code>	<code>false</code>	MAP クライアント
<code>smsc_mode_enabled</code>	<code>false</code>	SMS の SCS (MAP 機能)

CGRateS 設定

項目	値	説明
<code>cgrates_enabled</code>	<code>false</code>	CGRateS サービスの有効化
<code>cgrates_url</code>	<code>http://localhost:2080/jsonrpc</code>	CGRateS JSON-RPC エンドポイント
<code>cgrates_tenant</code>	<code>cgrates.org</code>	使用する SessionS のテナント
<code>cgrates_request_type</code>	<code>*prepaid</code>	CGRateS のリクエストタイプ (プリペイド/ポストペイド)
<code>cgrates_timeout</code>	<code>5000</code>	HTTP リクエストのタイムアウト

Web UI

Web UI ☐ CAMEL ☐

```
config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "SS7 ☐☐"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "☐☐☐"},
    {SS7.Web.CAMELSessionsLive, "/camel_sessions", "CAMEL ☐☐"},
    {SS7.Web.CAMELRequestLive, "/camel_request", "CAMEL ☐☐☐☐"}
  ],
  page_order: [
    "/events", "/m3ua", "/camel_sessions", "/camel_request",
    "/application", "/configuration"
  ]
end
```

項目名	パス	クラス名
セッション	/m3ua	SS7.Web.M3UAStatusLive
CAMEL セッション	/camel_sessions	SS7.Web.CAMELSessionsLive
CAMEL リクエスト	/camel_request	SS7.Web.CAMELRequestLive

□□□ **CAP** □□

CAP 00000000000000000000 CAP 00000000 (00000000000000000000)000
0000000 00000 000000000000000000000000

□□□□ (gsmSSF → gsmSCF)

TCAP 00000000

項目	説明
InitialDP	CGateS <code>InitiateSession</code> メッセージを 0 から 60 (最大 InitialDP/InitialDPSMS) まで送信
AssistRequestInstructions	CGateS TCAP メッセージを 1 回送信
EventReportBCSM	BCSM イベント (oAnswer, oDisconnect, ...) を CGateS に送信
Continue	続行
ConnectToResource	接続
RequestReportBCSMEvent	BCSM イベントを要求
ApplyCharging	CGateS BER を適用
Connect	CGateS BER を適用
ReleaseCall	呼び出しを解放

CGateS TCAP メッセージ (送信 / 受信)

接続 (gsmSCF → gsmSSF)

gsmSSF からの OmniSS7 接続

名前	値
InitialDP	0
Connect	20
ReleaseCall	22
RequestReportBCSMEvent	23
Continue	31
ApplyCharging	35
InitialDPSMS	60
NoteMM-Event	89 (MAP)

InitialDPSMS (値 60) は MO-SMS CAMEL として smsSSF の SMS-CSI / mo-sms-CSI として gsmSCF として SMSC として CAP ASN.1 として id-ac-cap3-sms-AC として

NoteMM-Event は MAP として (値 89 として CAP) として VLR の gsmSCF の CAMEL M-CSI として MAP ASN.1 として (:TCAPMessages.encode(:MapSpecificPDUs, ...)) として mm-EventReporting-v3 として

は CAP TCAP として :GenericTCAP.encode(:GenericPDUs, ...) として MAP の NoteMM-Event として :TCAPMessages

Web UI

CAMEL

URL: CAMEL . URL: /camel_sessions (HTTPS, 8087)

□□□□□□ CAMEL □□□□□

- 000000000000
- 0000 - OTID000 ID00000000
- **CAP** 00 0 InitialDP 00000000
- 0000 - IMSI0A 000B 00000000
- 0000 - initiated0 answered0 terminated0

IMEI: ID IMSI OTID

REST API `/camel-sessions` ()

CAMEL ☐ ☐ ☐ ☐ ☐

□□: CAMEL □□□□□ · □□: /camel_request (HTTPS, □□ 8087)

□□□□□□ CAP □□□□□□□□□□□□□□□□

```

00000000: InitialDPConnectReleaseCallRequestReportBCSMEEvent
ApplyChargingContinue

```

CAP : CAP (v1/v2/v3/v4 OID)
 CAP Ver

SCCP/M3UA

項目	値	説明
GT	99988415011	GT番号
GT	()	GT番号
SSN	146	gsmSSF
SSN	146	gsmSSF
OPC	5013	OPC番号
DPC	5011	DPC番号

CGrateS

OmniSS7 CGrateS **SessionS** JSON-RPC API
CGrateS.SessionsClient cgrates_url CAP SessionS

CAP	SessionS
InitialDP	SessionSv1.InitiateSession
EventReportBCSM (oAnswer)	SessionSv1.UpdateSession
EventReportBCSM (oDisconnect)	SessionSv1.TerminateSession

1. InitialDP

MSC InitialDP CAMELGW

- 1. CAP ()

2. **InitialDP** (IMSI/)
3. **CGrateS** SessionSv1.InitiateSession
4. **MaxUsage** ()
5. **CAMELGW.SessionStore** (ETS) OTID CAP CAMEL -
<hex OTID> ID
6. **CAP** **MSC**

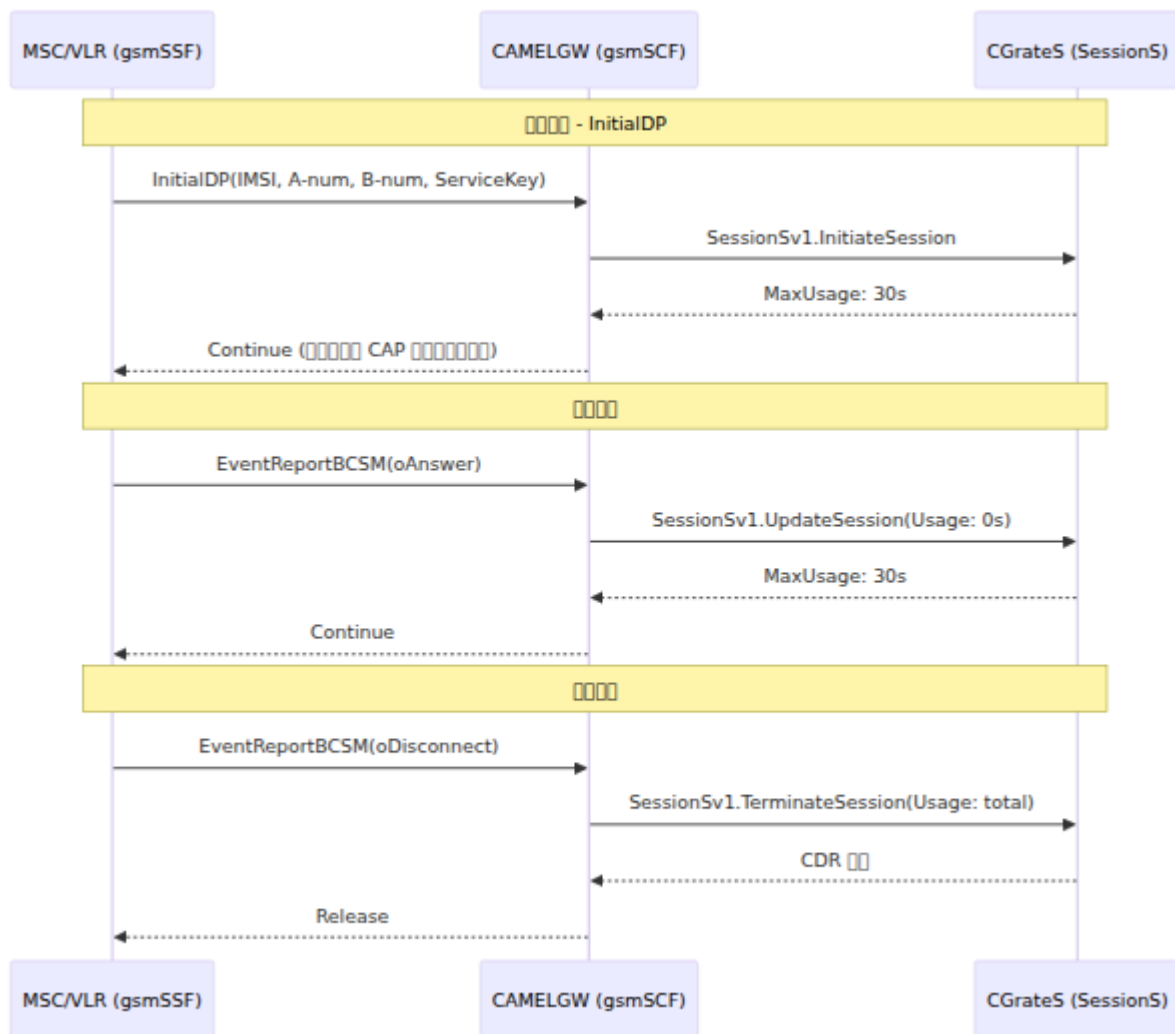
SessionS **IMSI** **Destination** **ServiceKey** **MSCAddress** **CallingPartyNumber** **CalledPartyNumber** **OriginID** ID

2. (EventReportBCSM - oAnswer)

oAnswer answered 0
SessionSv1.UpdateSession **CGrateS** **???**

3. (EventReportBCSM - oDisconnect)

oDisconnect terminated
SessionSv1.TerminateSession **CDR** **CGrateS**



CAMEL CDR

CDR CGrateS

- Account - IMSI
- Destination -
- OriginID - (CAMEL-<hex OTID>)
- Usage -
- IMSI CallingPartyNumber CalledPartyNumber
- MSCAddress - MSC
- ServiceKey - CAMEL

API

API 端點與參數

1. GET `https://your-server:8087/camel_request`
2. POST **InitialDP** 呼叫 CAP 事件 OTID
3. GET `https://your-server:8087/camel_sessions` 查詢 `initiated` 狀態
4. POST **EventReportBCSM** 呼叫 `oAnswer` 事件 `answered`
5. POST **ReleaseCall** 呼叫 `16` (事件) 結束

呼叫 MSC 事件

呼叫 MSC/VLR 呼叫 CAMEL 事件 (透過 `gsmSCF` 事件) 透過 OmniSS7 呼叫 InitialDP 事件
呼叫 MSC 事件

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

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

查詢

OmniSS7 查詢 JSON 事件 `OmniLogger.JsonFormatter` 查詢事件

```
# 查詢 CAP 事件 map_event 事件  
tail -f /var/log/omniss7/omniss7.log | jq 'select(.map_event |  
  toString | startswith("CAP:"))'
```

🔍 🔍 🔍 🔍

Prometheus 🔍

CAMELGW 🔍 `http://localhost:9090/metrics` 🔍 🔍 🔍

CAP 🔍:

- `cap_requests_total{operation}` - 🔍 🔍 CAP 🔍 🔍 🔍 🔍 snake_case 🔍 🔍
`initial_dp` 🔍 `connect` 🔍 `release_call` 🔍 `continue` 🔍 `apply_charging` 🔍
- `cap_request_errors_total{operation}` - 🔍 🔍 CAP 🔍 🔍 / 🔍 🔍 🔍

MAP / API 🔍:

- `map_requests_total{operation}` - 🔍 🔍 🔍 MAP 🔍 🔍
- `map_pending_requests` - 🔍 🔍 MAP TID 🔍 🔍 🔍

M3UA STP 🔍 (🔍 🔍 STP 🔍):

- `m3ua_stp_routing_failures_total{reason}` - 🔍 🔍 🔍 🔍 🔍

```
# 🔍 CAP 🔍 🔍 🔍 🔍  
curl http://localhost:9090/metrics | grep cap_requests_total  
  
# 🔍 InitialDP (🔍 snake_case 🔍)  
curl http://localhost:9090/metrics | grep  
'cap_requests_total{operation="initial_dp"}'
```

🔍 🔍 🔍

```
# M3UA 🔍 🔍 🔍 (TLS API 🔍 🔍 8445)  
curl -k https://localhost:8445/m3ua-status  
  
# 🔍 CAMEL 🔍 🔍 (🔍 session_count + 🔍 🔍 🔍 🔍 🔍)  
curl -k https://localhost:8445/camel-sessions
```


□□ /api/ocs-status □ /api/events □ /api/camel/sessions/count □□□□□□□□
/camel-sessions □

□□□□

□ config/runtime.exs □□□□□□□□

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

□□□□□□ config/config.exs □□□□□□□□ JSON□

```
config :logger, :default_formatter,  
  format: {OmniLogger.JsonFormatter, :format},  
  metadata: :all,  
  truncate: :infinity
```

□□□□

□□: □□□□ **CAP** □□

□□: □□□□□□□□ MSC □□□□ InitialDP□

□□:

1. M3UA □□□□: curl -k https://localhost:8445/m3ua-status
2. MSC CAMEL □□□□ (□□□□□gsmSCF □□)
3. SCCP □□ (gsmSCF □□□□□□□□ CAMELGW)
4. □□□□□ (□□ SCTP □□ 2905)

□□: **CGrateS** □□

□□: □□□□ not_configured □ http_error □ cgrates_error □

##:

1. `cgrates_enabled: true` 且 `cgrates_url` 配置正确 JSON-RPC 接口
2. 网络 (IMSI 号码) 正确 CGrateS 网络
3. `cgrates_tenant` 且 `cgrates_request_type` 正确 CGrateS 网络
4. `cgrates_timeout` 正确 CGrateS 网络

配置正确 CAMEL GW 网络配置正确

##: 配置

##: EventReportBCSM 网络 "配置"

##: OTID 配置

##: 配置 OTID 网络 DTID/OTID 网络

##: 配置

##: 配置 "InitialDP 网络"

##: CAP 配置

##: 配置 CAP 网络 MSC 网络 PCAP 网络

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

##

CAMEL 配置 OmniSS7 网络 gsmSCF 网络

- **CAP** 网络 (v1/v2/v3/v4) 配置
- 网络 **CGrateS SessionS JSON-RPC API** 配置
- 网络 (ConnectReleaseContinueApplyCharging)
- **ETS** 网络 **OTID** 网络

- **Web UI** **API**
- **API** (Web UI **/camel-sessions** API)
- **CGrateS** **CDR** **API**

API

- **API** - **CAP** **API**

API: OmniSS7 CAMEL **API**

API: 1.1

OmniSS7

← 快速入门

OmniSS7 快速入门

目录

1. Web UI
2. API
3. 部署
4. 配置
5. SCTP

Web UI

Web UI 默认使用 HTTPS 端口 8087，默认配置 `force_ssl` 为 `true`，使用 TLS 证书。
访问地址：`https://localhost:8087/`

API

API config/config.exs API page_order API

- API (/application) - API
- API (/configuration) - API
- **SS7** API (/events) - SS7 API
- API (/logs) - API
- **SS7** API (/client) - MAP API
- API (/m3ua) - M3UA/M2PA API
- API (/routing) - API
- API (/routing_test) - API
- **SMSc** API (/smc_links) - SMSc (SMSc)
- **SMSc** API (/smc_subscribers) - SMSc (SMSc)
- **HLR** API (/hmr_links) - HLR API (HLR)
- API (/subscribers) - UpdateLocation API
- **CAMEL** API (/camel_sessions) - CAMEL API (CAMEL GW)
- **CAMEL** API (/camel_request) - CAP API (CAMEL GW)

- **MSISDN/IMSI** `/msisdn_imsi_test` -

Web UI

- 1.
2. 8087 `https://localhost:8087/`
- 3.

Swagger API

API MAP API HTTP 9090

`http://your-server:9090/swagger`

Web UI

Web UI `config :control_panel, ControlPanelWeb.Endpoint` `config :control_panel`

```
config :control_panel, ControlPanelWeb.Endpoint,  
  force_ssl: [rewrite_on: [:x_forwarded_proto]],  
  server: true,  
  url: [host: "0.0.0.0", path: "/"],  
  https: [  
    port: 8087,  
    keyfile: "priv/cert/omnitouch.pem",  
    certfile: "priv/cert/omnitouch.crt"  
  ]  
  
config :control_panel,  
  page_order: [  
    "/application", "/configuration", "/events", "/logs",  
    "/client",  
    "/m3ua", "/routing", "/routing_test", "/smsc_links",  
    "/smsc_subscribers",  
    "/hlr_links", "/subscribers", "/camel_sessions",  
    "/camel_request",  
    "/msisdn_imsi_test"  
  ]
```

□□□□□

項目	型別	デフォルト値
ControlPanelWeb.Endpoint.url.host	String	"0.0.0.0"
ControlPanelWeb.Endpoint.https.port	Integer	8087
ControlPanelWeb.Endpoint.https.keyfile	String	"priv/cert/omnitouch.pem"
ControlPanelWeb.Endpoint.https.certfile	String	"priv/cert/omnitouch.crt"
ControlPanelWeb.Endpoint.force_ssl	Boolean	[rewrite_on: [:x_forwarded_proto]]
page_order	Array	[]

--	--	--	--

```

0000 OmniLogger.JsonFormatter 0000 JSON 00000000000000000000
Logger.metadata 000000000000STP 00 OPC/DPC/GTs00000000/0 MSISDNs0000 otid/0
000000000000 config/config.exs 0000

```

```
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all,
  truncate: :infinity
```

□ □ □ □ □ □ □ □ □

```
config :logger,
  level: :debug # []: :debug, :info, :warning, :error
```

□□□□□

項目	型	初期値
<code>:logger.level</code>	enum	<code>:debug</code>
<code>:logger.default_formatter.format</code>	string	<code>{OmniLogger.JsonFormatter, :format}</code>
<code>:logger.default_formatter.metadata</code>	list of string	<code>:all</code>
<code>:logger.default_formatter.truncate</code>	boolean	<code>:infinity</code>

API 一覧

OmniSS7 提供の HTTP サービス

サービス名	ポート	プロトコル	API 一覧
MAP 提供 API	9090	HTTP	MAP <code>/api/*</code> 提供 <code>/swagger</code> <code>/swagger.json</code> <code>/met</code>
JSON 提供 API (api_ex)	8445	HTTPS	提供 (<code>/api/status</code> <code>/api/m3ua-status</code> <code>/api/routing</code> <code>/api/subscribers</code> <code>/api/sessions</code> ...)

MAP API 接口 URL

`http://your-server:9090/api`

接口说明

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

OpenAPI 接口

`http://your-server:9090/swagger.json`

接口地址

Prometheus 接口

`http://your-server:9090/metrics`

接口地址

M3UA/SCTP 接口

- SCTP 接口地址
- M3UA ASP 接口地址
- 接口/接口地址

M2PA 接口

- 接口地址DOWN → ALIGNMENT → PROVING → READY
- 接口地址/接口地址

- 網際網路網路網路網路SCTP

STP 網路

- 網路網路/網路網路
- 網路網路網路
- 網路網路網路網路

MAP 網路網路

- 網路網路 MAP 網路
- 網路網路網路網路
- 網路網路網路

CAP 網路

- 網路網路 CAP 網路
- CAMEL 網路網路

SMSc 網路

- 網路網路
- 網路網路
- 網路網路網路

Grafana 網路

OmniSS7 網路 Prometheus 網路 Grafana 網路

網路網路

網路網路

1. 網路網路

- 網路網路 VLAN 網路

- 認證與授權
- 傳輸層安全 SCTP

2. Web UI 簡介

- 網路安全 TLS
- 網路安全認證與授權
- 網路 IP

3. API 簡介

- 網路安全
- 網路 API 認證 OAuth
- 網路安全認證與授權

網路安全

1. TPS 簡介

- 網路安全 TPS
- 網路安全
- 網路 SCTP 認證

2. 網路安全

- 網路 SMS 網路 API 認證與授權 5 認證與授權
- 網路 `auto_flush_tps` 網路 `auto_flush_interval` 認證與授權
- 網路安全認證與授權

3. M3UA 簡介

- 網路 SCTP 認證
 - 網路安全
 - 網路安全認證與授權
-

📖📖📖📖📖 SCTP 📖📖

📖📖 SCTP 📖📖📖

SCTP 📖📖📖 SCTP 📖📖📖📖📖📖📖 M3UA 📖📖📖📖📖📖📖📖📖📖📖📖 IP 📖📖📖📖📖📖📖📖📖

📖📖📖📖

- 📖📖📖📖📖📖📖📖📖📖SCTP 📖📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖📖📖📖 - SCTP 📖📖📖📖📖📖📖
- 📖📖📖📖📖📖📖📖📖📖📖📖 NIC 📖📖
- 📖📖📖📖SCTP 📖📖📖📖📖📖📖📖📖📖📖

📖📖📖📖

📖📖 M3UA 📖📖📖📖 IP 📖📖📖SCTP📖

1. 📖📖📖📖 **IP**📖📖📖📖📖📖📖📖 IP 📖📖
2. 📖📖📖📖SCTP 📖📖📖📖📖📖📖📖📖📖📖📖
3. 📖📖📖📖📖📖📖📖📖SCTP 📖📖📖📖📖
4. 📖📖📖📖📖📖📖📖📖📖📖📖📖
5. 📖📖📖📖📖📖📖📖SCTP 📖📖📖📖📖📖📖📖

📖📖

SCTP 📖📖📖📖 **IP** 📖📖📖 📖📖📖 IP 📖📖📖📖📖

📖📖 **IP**📖📖📖

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

📖📖 **IP**📖📖📖📖📖

```
# 00 IP - 000000
# 000 IP 00 IP000 IP 000000
local_ip: [{10, 179, 4, 10}, {10, 179, 4, 11}]
```

□□□□

□□□□□□□□□□□□□□□□

`role: :server` remote_ip
- SCTP INIT IP

`STP SctpInit` IP address SCTP parameters
IP address 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;
```

00 1000000 STP 000000000 - 000

```
# STP 配置
config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "Partner_STP_Redundant",
      role: :client, # 主 - 从
      # 本地 IP 地址
      local_ip: [{213, 57, 23, 200}, {213, 57, 23, 201}],
      local_port: 0,
      # 远端 IP 地址 - 远端 IP 地址
      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 配置
config :omniss7,
  map_client_enabled: true,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hmr_client_asp,
    # 本地 IP 地址
    local_ip: [{10, 0, 0, 100}, {10, 0, 0, 101}],
    local_port: 2905,
    # STP 配置
    remote_ip: [{10, 0, 0, 1}, {10, 0, 0, 2}],
    remote_port: 2905,
    routing_context: 1
  }
}
```

3. STP 配置


```
# 配置 STP 配置
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 配置 - 配置配置配置配置配置

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

配置 5 配置配置配置配置配置

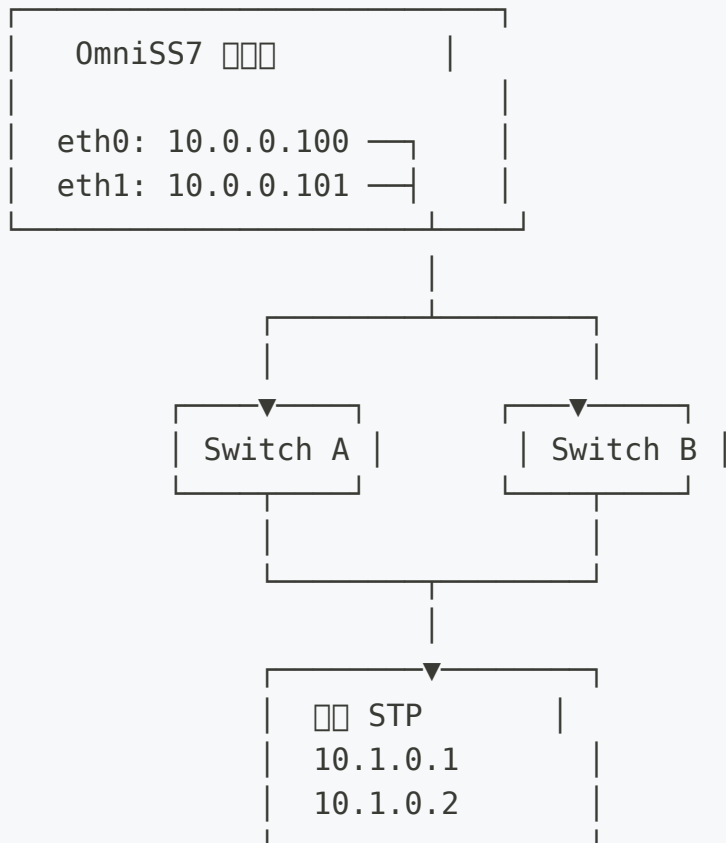
```

# 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



□□□

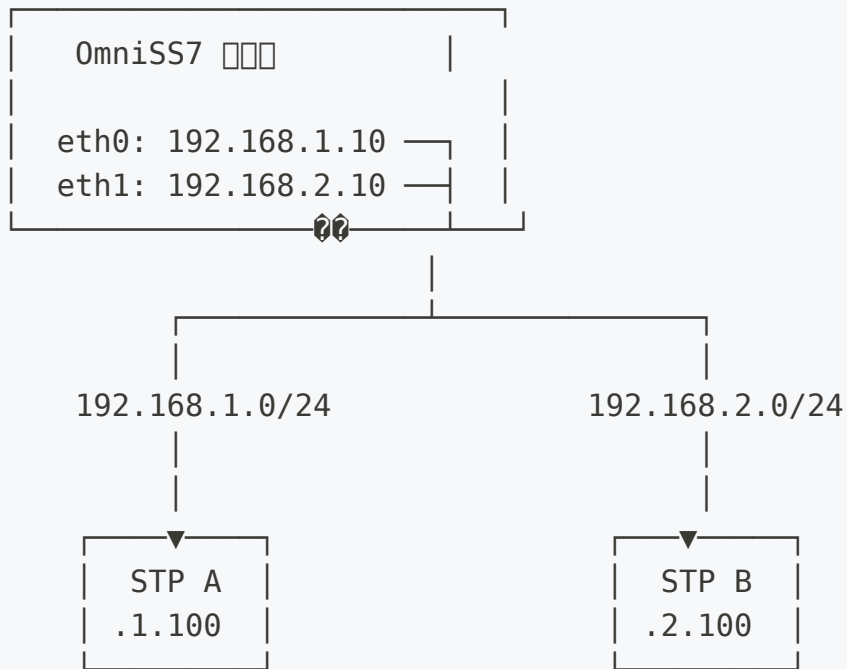
```

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

□□□

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

□□ **2**□□□□□□



□□□

```

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

□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

```

[info] SCTP □□□□□□□□□□ 2 □□□ IP
[info] STP □□□□□□□□□□□□ 2 □□□ IP
  
```

□□□□□□□□□□

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

Web UI

Web UI

M3UA

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

1.

- NIC
-
-
- -

2. IP

- IP IP -
- - IP
- -

3.

```
# 断网
sudo ip link set eth0 down

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

# 恢复网络
sudo ip link set eth0 up
```

4. 断网测试

- 断网前 IP
- 断网后 IP
- 断网后网络状态

5. 断网恢复

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

断网

断网前 IP

断网后 IP

断网

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

断网前 IP

断网后 IP

目標

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

標準化の経緯

標準化された UP と DOWN の経緯

目標

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

標準化

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

標準化

- 標準化 IP 標準化
- 標準化 IP 標準 IP 標準
- 標準化 STP、HLR、SMSc と MAP 標準化
- **Erlang** 標準化 SCTP と Erlang

標準化

標準化

- M3UA 標準化

- MAP □□□□□
- API □□□□
- □□□□□□

□□□□□

- M3UA ≥ 1 μm
- MAP $\geq 10\%$
- $\text{PDI} \geq 1000$
- API $\geq 5\%$

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

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

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

□□□□ (:logger)

```
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all,
  truncate: :infinity

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


項目	型	デフォルト	説明
<code>:logger.level</code>	enum	<code>:debug</code>	
<code>:logger.default_formatter.format</code>	string	<code>{OmniLogger.JsonFormatter::format}</code>	
<code>:logger.default_formatter.metadata</code>	array of string	<code>:all</code>	
<code>:logger.default_formatter.truncate</code>	boolean	<code>:infinity</code>	

Web UI 項目 (`:control_panel`)

```
config :control_panel, ControlPanelWeb.Endpoint,  
  force_ssl: [rewrite_on: [:x_forwarded_proto]],  
  server: true,  
  url: [host: "0.0.0.0", path: "/"],  
  https: [  
    port: 8087,  
    keyfile: "priv/cert/omnitouch.pem",  
    certfile: "priv/cert/omnitouch.crt"  
  ]  
  
config :control_panel,  
  page_order: [  
    "/application", "/configuration", "/events", "/logs",  
    "/client",  
    "/m3ua", "/routing", "/routing_test", "/smc_links",  
    "/smc_subscribers",  
    "/hlr_links", "/subscribers", "/camel_sessions",  
    "/camel_request",  
    "/msisd_n_imsi_test"  
  ]
```

項目	型	デフォルト値	説明
ControlPanelWeb.Endpoint.url.host	String	"0.0.0.0"	
ControlPanelWeb.Endpoint.https.port	Integer	8087	
ControlPanelWeb.Endpoint.https.keyfile	String	"priv/cert/omnitouc"	
ControlPanelWeb.Endpoint.https.certfile	String	"priv/cert/omnitouc"	
ControlPanelWeb.Endpoint.force_ssl	Boolean	[rewrite_on: :x_forwarded_proto]	
page_order	Integer	1	

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	項目	項目	項目
sctp_handler.enabled	項目	項目	false	項目項目 STP 項目
sctp_handler.local_ip	項目	項目	{127, 0, 0, 1}	項目項目 M3UA 項目 IP
sctp_handler.local_port	項目	項目	2905	M3UA 項目 SCTP 項目
enable_gt_routing	項目	項目	false	項目項目項目項目

M3UA 項目項目項目

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

M3UA 定義

項目	型	必須	説明
dest_pc	整数	○	宛先ポート
peer_id	整数	○	ピアID
priority	整数	○	優先度
network_indicator	整数	○	:international 国際 :national 国内

M3UA GT 項目

項目	型	必須	説明
gt_prefix	整数	○	GTプレフィックス
peer_id	整数	○	ピアID
priority	整数	○	優先度
description	文字列	○	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
map_client_enabled	bool	false	MAP クライアントの有効化
map_client_m3ua.mode	enum	"ASP"	M3UA モード "ASP" (Asynchronous Signaling Point) "SGP" (Signaling Gateway Point)
map_client_m3ua.callback	func	{MapClient, :handle_payload, []}	MAP クライアントのコールバック関数
map_client_m3ua.process_name	string	:map_client_asp	MAP クライアントのプロセス名
map_client_m3ua.local_ip	string	-	ローカル IP アドレス
map_client_m3ua.local_port	integer	2905	ローカル SCTP ポート番号
map_client_m3ua.remote_ip	string	-	リモート STP/SGP の IP アドレス
map_client_m3ua.remote_port	integer	2905	リモート SCTP ポート番号
map_client_m3ua.routing_context	string	-	M3UA ルーティングコンテキスト

オプション (:omniss7)


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

項目	型	必須	デフォルト	説明
auto_flush_enabled	Boolean	否	false	自動フラッシュの有効/無効
auto_flush_interval	Integer	否	10000	自動フラッシュの間隔(ミリ秒) ???
auto_flush_dest_smsc	String/nil	否	nil	自動フラッシュの宛先 SMSC 名 nil = 不明
auto_flush_tps	Integer	否	10	自動フラッシュの TPS

HTTP API 項目 (:omniss7)

HTTP API 項目

```
config :omniss7,  
  smsc_api_base_url: "https://omnimessage.example.com:8443",  
  smsc_name: "omni-smsc01" # 宛先 SMSC 名
```

API 項目

項目	項目	項目	項目
smc_api_base_url	項目	項目	"https://omnimessage.example.com:8443"
smc_name	項目	項目	項目

API 項目

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

項目

項目 API 項目 85 項目

- 項目 SS7
- 項目
- 項目
- 項目JSON 項目

項目

- 項目 SSL
- HTTP 項目 5 項目

- 符合 ISO 8601 格式
 - API 返回 JSON 格式/内容
-

功能

- 返回
 - STP 设置
 - MAP 设置
 - 设置
 - HLR 设置
-

OmniSS7 与 Omnitouch 兼容

□□□□

← □□□□

□□□□□□□□ OmniSS7 □□□□□□□□□□

□□

1. □□
2. □□□□
3. □□□□□□
4. HLR □□□□
5. SMSc □□□□
6. STP □□□□
7. CAMEL □□□□□□
8. □□□□ NAT □□
9. M3UA □□□□
10. SCCP □□□□□□
11. □□□□
12. USSD □□□□
13. □□□ / □□□□
14. □□□□□□
15. □□□□□□
16. □□□□

□□

OmniSS7 □□□□□□□□□□□□□□□□

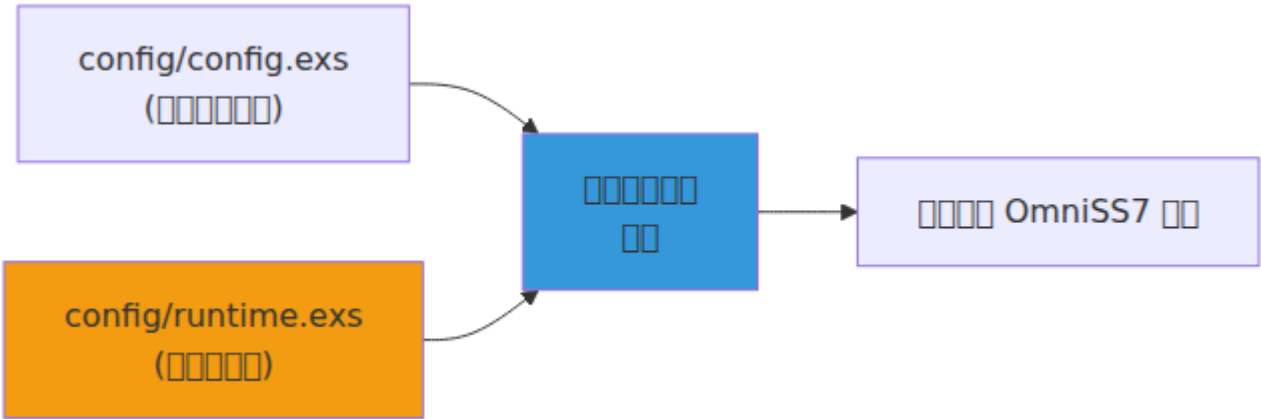
- **STP** □□ - □□□□□□□□□□
- **HLR** □□ - □□□□□□□□□□□□

- **SMSc** -
- **CAMEL GW** - CAMEL

??? HLR + SMSc

OmniSS7

config/config.exs	/ :api_ex REST	
config/runtime.exs ()	4	
config/runtime.exs ()	 Ansible/Jinja GTM3UA API	



```

config :omniss7, ...
config/runtime.exs config/runtime.exs

```

Application.get_env(:omniss7, ...)

map_client_enabled		false	MAP M3UA	
hlr_mode_enabled		false	HLR	HLR
smsc_mode_enabled		false	SMSc	SMSc
cap_client_enabled		false	CAMEL CAP	CAMEL GW
camelgw_mode_enabled		false	CAMEL	CAMEL GW
ussd_gateway_enabled		false	USSD HTTP/JSON	USSD GW

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

HLR 配置

HLR 配置参数

HLR API 配置

名称	数据类型	默认值	必填	说明
hlr_api_base_url	字符串	-	是	HLR API 基地址
hlr_api_status_path	字符串	"/api/status"	是	HLR API 状态检查路径 HLR API 基地址 + 该路径 = HLR API 状态检查地址 例如：hlr_api_base_url = "http://192.168.1.1:8080"，hlr_api_status_path = "/api/status"，则 HLR API 状态检查地址为 "http://192.168.1.1:8080/api/status"
hlr_service_center_gt_address	字符串	-	是	HLR 服务中心 GT 地址
smsc_service_center_gt_address	字符串	-	是	SMSC 服务中心 GT 地址

SSL 配置参数 hlr_api_verify_ssl 用于配置 HLR API 是否使用 TLS 加密通信。
:api 配置参数 verify_ssl 用于配置 API 是否使用 SSL 加密通信。

配置

```
config :omniss7,
  hlr_api_base_url: "https://omnihss.example.com:8443",
  hlr_api_status_path: "/api/status",
  hlr_service_center_gt_address: "55512341111",
  smsc_service_center_gt_address: "55512341112"
```

AlertServiceCenter

UpdateLocation HLR SMC GT alertServiceCenter

hlr_smc_alert_gts	 	[]		alertServiceCer SMC
hlr_alert_location_expiry_seconds		172800		4 0

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

MSISDN ↔ IMSI

MSISDN IMSI HLR MSISDN ↔ IMSI

項目	単位	値	注	説明
hlr_imsi_plmn_prefix	桁 桁 桁	"50557"	桁	IMSI の MCC + MNC 部分
hlr_msisdn_country_code	桁 桁 桁	"61"	桁	IMSI → MSISDN 変換用の国番号
hlr_msisdn_nsn_offset	桁 桁	0	桁	MSISDN の NSN 部分のオフセット
hlr_msisdn_nsn_length	桁 桁	9	桁	MSISDN の NSN 部分の長さ

例2: 国番号 61

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "99",      # 桁 2 の国番号
  hlr_msisdn_nsn_offset: 2,           # 桁 2 のオフセット
  hlr_msisdn_nsn_length: 9            # 桁 9 の NSN
```

例3: 国番号 99

```
config :omniss7,
  hlr_imsi_plmn_prefix: "50557",      # MCC 505 + MNC 57
  hlr_msisdn_country_code: "999",     # 桁 3 の国番号
  hlr_msisdn_nsn_offset: 3,           # 桁 3 のオフセット
  hlr_msisdn_nsn_length: 8            # 桁 8 の NSN
```

国番号 nsn_offset の関係

- 国番号 "9" の 1 桁 → nsn_offset: 1
- 国番号 "99" の 2 桁 → nsn_offset: 2

- 0000 "999"03 00 → nsn_offset: 3

InsertSubscriberData (ISD) 00

0 UpdateLocation 00000 VLR 0000000000000000 ISD 0000000000000000 HLR 0000
InsertSubscriberData 000

00	0 0	00	0 0	
isd_network_access_mode	0 0	:packetAndCircuit	0	00000 00 :packetAndCi 0 :circuitOnly
isd_send_ss_data	0 0	true	0	000000000000 ISD
isd_send_call_barring	0 0	true	0	000000000000 ISD

000

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

CAMEL 00

0000 CAMEL 0000000000000000 CAMEL 0000000000000000 HLR 0000 CAMEL 000

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

例

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

例 **VLR**

例 PRN HLR

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

例

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

SMSc 配置

配置说明

SMSc API 配置

名称	数据类型	默认值
smsc_api_base_url	字符串	"https://omnimessage.example.com"
smsc_api_status_path	字符串	"/api/status"
smsc_name	字符串	"{hostname}_SMSc"
smsc_service_center_gt_address	字符串	-

SSL 验证配置 `smsc_api_verify_ssl` 用于验证 API 是否使用 TLS 加密通信。
:api 配置 `verify_ssl` 用于验证 API 是否使用 SSL 加密通信。

配置

```
config :omniss7,
  smsc_api_base_url: "https://omnimessage.example.com:8443",
  smsc_api_status_path: "/api/status",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112"
```

配置文件中 5 个配置项 SMS.FrontendRegistry 配置

配置项

配置项	类型	默认值	是否必填	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
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

配置

```
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 STP 配置

STP

sctp_handler.enabled		false		SCTP SocketHandler
sctp_handler.local_ip		{127, 0, 0, 1}		IP {10, 0, 0, 1} SCTP IP [{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,
    # IP
    local_ip: [{10, 179, 4, 10}, {10, 179, 4, 11}],
    local_port: 2905,
    point_code: 100
  }
```

SCTP SCTP

enable_gt_routing		false		PC GT

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

M3UA/M2PA

m3ua_peers M3UA M2PA STP M2PA

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


```

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

```

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	格式	范围	描述
m3ua_routes	列表	[]	1	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 - 1 = 最高
- 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	値	注	備考
m3ua_gt_routes	配列	[]	???	配列要素は以下を参照

配列要素は m3ua_gt_routes の要素を参照

要素は

- **gt_prefix** 宛先プレフィックス - 必須
- **peer_id** ピア ID - 0 以上
- **priority** 優先度 - 0 以上
- **description** 説明

要素は

- **source_ssn** 送信元 SSN
- **source_tt** 送信元 TT
- **source_npi** 送信元 NPI
- **source_nai** 送信元 NAI

要素は

- **dest_ssn** 宛先 SSN
- **dest_tt** 宛先 TT
- **dest_npi** 宛先 NPI
- **dest_nai** 宛先 NAI
- **route_on_ssn** SSN をルートに使用するかどうか **false** GT を使用する **true**
 GTT ITU-T Q.714 §2.4 STP 宛先 SSN を使用する **true**
 STP 宛 → GTT 宛 SSN

注

- **(TT)** 0=1=2=3=4
- **(NPI)** 0=1=ISDN(E.164)6=E.212
- **(NAI)** 0=1=3=4=5
- **(SSN)** 6=HLR7=VLR8=MSC9=EIR


```

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
    },

    # 86 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
    },
  ],

```

```
# 00/0000
%{
    gt_prefix: "",
    peer_id: 1,
    priority: 99,
    description: "000000"
}
```

CAMEL

CAMEL CAP

CAMEL ☐☐☐☐

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

CAP ☐ ☐ ☐ ☐

項目	形式	単位	長さ	説明
cap_version	2バイト	バージョン	2	CAP のバージョン番号。v1, v2, v3, v4 のいずれか。
camel_gsmSCF_gt_address	2バイト	(GSM SCF GT アドレス)	2	CAMEL の GSM SCF GT アドレス。

CAP ☐ ☐ ☐ ☐ ☐

- **v1** → **OID: 0.4.0.0.1.0.50.0**
- **v2** → **OID: 0.4.0.0.1.0.50.1** -
- **v3** → **OID: 0.4.0.0.1.21.3.4**

- :v4 → 0.4.0.0.1.23.3.4

OID 0.4.0.0.1.23.3.4

```
config :omniss7,
  cap_client_enabled: true,
  camelgw_mode_enabled: true,
  cap_version: :v2,
  camel_gsmscf_gt_address: "99988411553"
```

CGrateS

CGrateS /

cgrates_enabled		false		CGrateS
cgrates_url		-		CGrateS JSON-RPC U
cgrates_tenant		"cgrates.org"		CGrateS
cgrates_request_type		"*prepaid"		*prepaid *postpaid *pseudoprepaid
cgrates_timeout		5000		CGrateS

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	配置	配置	配置
cap_client_m3ua	配置	-	配置

配置

```
config :omniss7,
  cap_client_m3ua: %{
    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
    dpc: 5011      # 配置
  }
```

配置 NAT 规则

配置规则时，配置规则名称 GT 规则名称 = 规则名称
规则名称 = 规则名称 NAT 规则

名称	类型	默认值	说明	备注
gt_nat_enabled	布尔	false	是否启用	是否/是否 GT NAT 规则
gt_nat_rules	列表	[]	规则列表	规则名称 GT NAT 规则

规则名称 gt_nat_rules 规则名称

- calling_prefix 规则名称 GT 规则名称
- called_prefix 规则名称 GT 规则名称
- weight 规则名称 = 规则名称 - 规则100
- response_gt 规则名称

规则名称

- 规则 weight 规则名称 = 规则名称
- 规则名称
- calling_prefix 规则 called_prefix 规则名称

规则

```
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:  
"99988411553"},  
  
    # 0000000000000000 "555"  
    %{called_prefix: "555", weight: 10, response_gt:  
"99988411554"},  
  
    # 00000000 "8773"  
    %{calling_prefix: "8773", weight: 10, response_gt:  
"99988411554"},  
  
    # 000000000000000000000000  
    %{weight: 100, response_gt: "99988411555"}  
  ]
```

000000000000000000 GT NAT 000

M3UA 0000

MAP 000000 M3UA 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}のhandle_payに指定する。
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。IPアドレスは、{0, 0, 1}の形式で指定する。例: {10, 0, 0, 1}、{10, 0, 0, 1}、{10, 0, 0, 1}。
map_client_m3ua.local_port	uint16_t	2905	必須	ローカルSCTPポート。
map_client_m3ua.remote_ip	ip_address	-	必須	リモートSCTP/SGW IPアドレス。例: {10, 0, 0, 1}、[{10, 0, 0, 1}, {10, 0, 0, 1}]。
map_client_m3ua.remote_port	uint16_t	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。
map_client_m3ua.opc	uint16_t	5013	必須	オペレーティングポイントコード (OPC)。MAP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
	map_client_m3ua.receive_watchdog	true		ASP からの SCTP 接続監視の有効化 receive_watchdog_enable 設定値 false 有効 true 無効
	map_client_m3ua.receive_watchdog_idle	15		SCTP 接続監視のアイドルタイムアウト receive_watchdog_idle 設定値 true 有効 false 無効

IP 番号

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

X-Y-Z

(X * 2048) + (Y * 8) + Z

2-114-5 =

(2 * 2048) + (114 * 8) + 5 = 5013

配置SCTP 配置

```
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,
    # 远端 IP 地址 STP 地址
    remote_ip: [{10, 179, 4, 10}, {10, 179, 4, 20}],
    remote_port: 2905,
    routing_context: 1
  }
```

配置SCTP 配置

配置

配置SCTP 配置 ESTABLISHED 配置

配置SCTP 配置 M3UA BEAT 配置NOTIFY 配置 RFC 4666 §3.8 M3UA BEAT 配置SCTP 配置 SG 配置 M3UA BEAT 配置

配置	配置	配置	配置
receive_watchdog	配置	true	配置 false 配置 SCTP 配置
receive_watchdog_idle	配置	15	配置 receive_watchdog: false 配置

配置 ASP 配置

```

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
  }

```

000000000000

```

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
  }

```

SCCP 000000

OmniSS7 0000000000000000 SCCP 0000000000SST00ITU-T Q.714 §5.3.400

local_subsystems 0000000000000000SSN000000 SSN 0 SST 0000 Subsystem-Allowed (SSA)000000 SSN 0000 Subsystem-Prohibited (SSP)0


```
config :omniss7,  
  #   
  screening_default_action: :deny
```

ss7_screen_rule Mnesia

USSD

USSD HTTP/JSON `ussd_gateway_enabled:`
`true` `ussd_gateway` USSD

項目	単位	初期値	範囲	説明
ussd_gateway.routes	配列	[]	文字列、整数、浮点数、真偽値、配列	URL へリダイレクトする URL のパターンを指定します。% {pattern: "<prefix>", url: "<callback URL>"} の形式で指定します。 パターンは "*" を含むことができます。
ussd_gateway.session_timeout_ms	ミリ秒	180_000	0 ~ 300,000	セッションのタイムアウト時間（ミリ秒）を指定します。 3 秒未満の値は有効ではありません。
ussd_gateway.turn_timeout_ms	ミリ秒	30_000	0 ~ 300,000	ターンタイムアウト時間（ミリ秒）を指定します。 3 秒未満の値は有効ではありません。
ussd_gateway.http_timeout_ms	ミリ秒	5_000	0 ~ 300,000	HTTP リクエストのタイムアウト時間（ミリ秒）を指定します。 3 秒未満の値は有効ではありません。
ussd_gateway.max_text_length	文字数	182	0 ~ 255	USSD メッセージの最大文字数を指定します。 GSM 7 ビットエンコーディングを使用します。 0 を指定すると、デフォルト値（182）が適用されます。

USSD メッセージのパターンは pattern という名前で指定します。pattern は次の形式で指定します。

Parse error on line 3: ...} B -->|\"*100\"| C[\"POST U -----
^ Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '-)', 'STADIUMEND',
'SUBROUTINEEND', 'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND',
'TRAPEND', 'INVTRAPEND', 'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'STR'



□ □ □ / □ □ □ □

OmniSS7 Web

項目	項目	項目	項目
start_web_interfaces	項目	true	項目 項目項目 UI :api_ex REST 項目 SS7 項目項目 false 項目項目 項目 SS7.Events 項目項目 ENABLE_WEB_INTERFACES 項目 項目 true/1 項目 false/0 項目項目
start_http_server	項目	true	項目 項目 Plug.Cowboy HTTP 項目項目 api_port 項目 REST /api/* 項目 /metrics 項目 /swagger 項目 項目 REST API 項目項目
api_port	項目	9090	項目 項目 Plug.Cowboy HTTP 項目 TCP 項目 項目 REST /api/* 項目 /metrics 項目 /swagger 項目 項目項目 Prometheus 項目項目
headless	項目	false	項目 項目項目 true 項目 OmniSS7 項目 項目/項目項目 MAP 項目 CAP 項目 USSD 項目 項目項目項目 Web UI 項目項目 項目 SCTP 項目項目 M3UA 項目 項目 OmniSS7 項目項目項目
vlr_cache_ttl_seconds	項目	3600	項目 項目 SSMc 項目項目 VLR 項目項目 項目

項目項目項目

```
config :omniss7,  
  headless: true
```

項目 Web 項目 REST API 項目

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

Configuration

Configure the Web API interface

Configuration

Field	Type	Default	Required	Description
license_client.license_server_api_urls	String	-	Yes	API URL
license_client.licensee	String	-	Yes	Licensee

Example

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

Web

Property	Access	Value
control_panel.parent_application_readable_name	Read	"OmniSS7 Stack"
control_panel.use_additional_pages	Read	[]
control_panel.page_order	Read	[]
ControlPanelWeb.Endpoint.url.host	Read	"0.0.0.0"
ControlPanelWeb.Endpoint.https.port	Read	8087
ControlPanelWeb.Endpoint.https.keyfile	Read	"priv/cert/omnit"
ControlPanelWeb.Endpoint.https.certfile	Read	"priv/cert/omnit"

□□□

```
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	単位	設定値	単位	説明
start_http_server	bool	true	bool	HTTP サーバを起動するかどうか 9090ポートで起動
api_ex.api.port	int	8445	int	API サーバのポート番号
api_ex.api.listen_ip	str	"0.0.0.0"	str	API サーバのリスニング IP
api_ex.api.product_name	str	"OmniSS7"	str	API サーバの製品名
api_ex.api.title	str	"API - OmniSS7"	str	Swagger UI のタイトル
api_ex.api.hostname	str	"localhost"	str	API サーバのホスト名
api_ex.api.enable_tls	bool	true	bool	API サーバに TLS を有効にするかどうか
api_ex.api.tls_cert_path	str	"priv/cert/omnitouch.crt"	str	API サーバの TLS 証明書ファイルのパス

項目	値	項目	値	説明
api_ex.api.tls_key_path		"priv/cert/omnitouch.pem"		API の TLS 証明書

設定

```
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"  
  }
```

ポート番号

ポート	プロトコル	サービス	説明
9090	HTTP (Plug.Cowboy)	APIhandler	MAP/CAP REST /api/* /metrics start_http_server
8087	HTTPS (Bandit/Phoenix)	ControlPanelWeb.Endpoint	Web アプリ UI Web start_web_interface
8445	HTTPS / TLS	:api_ex REST API	API の CRUD 操作 Web アプリ

API 設定

- MAP/CAP REST API + Swagger: `http://[server-ip]:9090/...` API 設定
- `:api_ex` REST 設定: `https://[server-ip]:8445/api/*`
- Web 設定: `https://[server-ip]:8087/`
- Prometheus 設定: `http://[server-ip]:9090/metrics`

API 設定 SSL 設定

設定 HLR/SMSc API 設定*_verify_ssl 設定:api 設定

項目	単位	値	説明
<code>api.base_url</code>	文字列	<code>smsc_api_base_url</code>	API 接続する URL
<code>api.timeout</code>	秒	<code>5000</code>	接続タイムアウト
<code>api.verify_ssl</code>	ブール値	<code>false</code>	TLS 接続時に SSL 証明書を確認するかどうか 設定項目 <code>hlr_api_verify_ssl</code> <code>smsc_api_verify_ssl</code>

設定

```
config :omniss7,  
  api: %{\br/>    base_url: "https://backend.example.com:8443",  
    timeout: 5000,  
    verify_ssl: false  
  }
```

配置

配置	类型	默认值	是否必填	说明
<code>logger.level</code>	字符串	<code>:debug</code>	否	日志级别，支持 <code>:debug</code> 、 <code>:info</code> 、 <code>:warn</code> 、 <code>:error</code> 。
<code>logger.backends</code>	数组	<code>[:console]</code>	否	日志后端，支持 <code>:console</code> 、 <code>SS7.Web.LoggerBackend</code> 、 <code>Web UI</code> 。
<code>logger.default_formatter.format</code>	字符串	-	否	默认日志格式。
<code>logger.default_formatter.metadata</code>	数组	<code>[]</code>	否	默认日志元数据。
<code>logger.default_formatter.truncate</code>	字符串/数字	<code>:infinity</code>	否	默认日志截断长度。

示例

```
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
```


Mnesia

mnesia_storage_type		:disc_copies		Mnesia :disc_copies :ram_copies

```

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

```

- :disc_copies -
- :ram_copies -

Mnesia

- m3ua_peer - M3UA
- m3ua_route -
- m3ua_gt_route -

Mnesia.{node_name}/

#

□□

□	□□	□□□□
MAP □□□□ 10 □	□□ MAP □□□ 10 □□□□	□□□□□
ISD □□□ 10 □	□□ ISD □□□ 10 □□□□	□□□□□

HTTP □□□

□	□□	□□□□
HTTP IP□ 0.0.0.0	□□/Swagger □□□□□□□ □□□□	□□□□□
HTTP □□□ 9090	□□/Swagger □□□□□ 9090 □□□	□□ <code>config :omniss7, api_port:</code> <code><port></code> □□□ <code>9090</code> □

□□□□

□	□□	□□□□
□□□□□ 5 □□	SMSc □ 5 □□□□□□□	□□□□□

Web UI □□□□

□□	□□
□□□□	5 □
□□□□	2 □

□□□□

□□ **HLR** □□

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

☐☐ **SMSc** ☐☐

```
config :omniss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: false,  
  smsc_mode_enabled: true,  
  
  smsc_api_base_url: "https://omnimessage.example.com: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  
  }
```

STP 配置

```
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 85+

配置

- 配置 6 项
- HLR 配置 17 项
- SMSc 配置 8 项

- STP 0005+ 000000 m3ua_peers m3ua_routes m3ua_gt_routes 000
- CAMEL GW 00014 000
- 0000 NAT 02 000
- M3UA 0008 000
- SCCP 0000001 000
- USSD 0005 000
- 000 / 0004 000
- 000000000Web API 0000023 000
- 00001 000

00000000

HLR 000

- hlr_api_base_url
- hlr_service_center_gt_address
- smsc_service_center_gt_address
- 00 map_client_m3ua.* 00080

SMSc 000

- smsc_api_base_url
- smsc_service_center_gt_address
- auto_flush_dest_smsc 0000000000
- 00 map_client_m3ua.* 00080

STP 000

- sctp_handler.point_code 00000 SCTP 00000
- sctp_handler.local_ip
- sctp_handler.local_port

CAMEL GW 000

- cgrates_url 00000 CGrateS
- 00 cap_client_m3ua.* 00080

□□□□

- `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

111

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

□ □ □ ◆ ◆ □

□ □ □ □ □

- weight [0.0, 0.0, 0.0, 0.0, 0.0]
- response_gt [0] 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

00000000000000000000 GT 000000

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

0000 GT NAT00000000000000000000 GT NAT000 OmniSS7 00000000000000

0000

000 HLR 0 SSMSc 000000000000

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

GT NAT 000000000000000000 GT 000000

□□□□□

□□□□□□□□□□

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

□□□□

□□□□□□

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

□□□□□□□□

GT NAT □□□ SS7 □□□□□□

SCCP □□□□□□□

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

- 配置

MAP 配置

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

配置

配置

配置 `config/runtime.exs`

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

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

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

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

配置

配置 `config/runtime.exs` 中的 `NAT` 配置

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

設定

設定ファイル

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

パラメータ

- calling_prefix** 発信元 GT 番号
 - 必須
 - 空文字列 "" は nil として扱われる
 - GT 番号
- called_prefix** 着信元 GT 番号
 - 必須
 - 空文字列 "" は nil として扱われる
 - GT 番号
- weight** 優先度
 - 整数 = 優先度
 - 0 >=
 - 必須

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

00000 **calling_prefix** 0 **called_prefix** 000000000000000000000000/000000

00000000

00000000 000000000000000000000000

000000

1. 0000000000000000
 - 000000 **calling_prefix**00000000 GT 00
 - 000000 **called_prefix**00000000 GT 00
 - 00000000000000000000
 - 00000000000000000000
2. 00000000000000000000
 - 000000000 - 00000000
 - 00000000000000 - 00 = 00000
3. 000000000000

0000


```

# []
gt_nat_rules: [
  # 1[] - []
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,
  response_gt: "11111"},

  # 10[] - []
  %{calling_prefix: "8772", weight: 10, response_gt: "22222"}, #
  []
  %{called_prefix: "555", weight: 10, response_gt: "33333"}, #
  []

  # 100[] - []
  %{weight: 100, response_gt: "44444"} # []
]

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

```

[]

[] **1**[][][][][]

🔗🔗[][][] SMS[] GT []

```

config :omniss7,
  gt_nat_enabled: true,

  # SMSc 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"},

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

```

Example

```

44121234567 SRI-for-SM
  GT5551001 GT A
  GT44121234567 A GT

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

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

```

2 GT HLR

HLR GT

```

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

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

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

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

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

```

3

GT GT

```

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

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

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

```

例 4 電話番号別対応

電話番号別 GT 対応を定義する GT 設定 GT

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

例

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

例 5 電話番号 + 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 - 00000000 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"

```

GT NAT

GT NAT 支持 OmniSS7 和 SS7

HLR 支持

GT NAT 支持

- UpdateLocation 支持 HLR GT
- InsertSubscriberData 支持 HLR GT
- SendAuthenticationInfo 支持
- 支持

HLR 支持 HLR 支持

支持

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

SMSc 支持

GT NAT 支持

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

SMSc 支持 SMSc 支持

□□□

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

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

CAMEL □□□□

GT NAT □□□

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

□□□

```

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

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

```

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

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

[] **GT NAT** [] [] []

GT NAT [] [] [] [] [] [] [] [] [] []

```

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

```

[] [] [] [] [] [] NAT [] [] [] [] []

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

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

00

00000000 GT NAT 000

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

# 00000000
iex> GtNat.enabled?()
true

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

00 GT NAT

00000000 GT NAT 000

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

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

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

# 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"

# 000000000000000
iex> GtNat.translate_response_gt("999999999", "888888",
"default_gt")
"default_gt"
```

0000

000**GT NAT** 000

00 **1**0000000

```
iex> Application.get_env(:omniss7, :gt_nat_enabled)
true # 000 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` 確認 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 的關係 OmniHSS

OmniHSS 簡介

OmniSS7 HLR 透過 SS7 與 OmniHSS 連接 HSS 資料庫

- **OmniSS7 (HLR 端)** 透過 SS7/MAP 與 SCCP 連接
- **OmniHSS (HSS 端)** 透過 HSS 資料庫連接

OmniHSS 功能

OmniHSS 提供以下功能：

- **IMSI** 資料庫 IMSI 與 MSISDN 對應 eSIM 卡
- Milenage 3G/4G/5G COMP128 2G 加密
- CS 與 PS 網路選擇
- CAMEL 呼叫
- **API** 提供 RESTful HTTP API 與 CRM 系統
- 資料庫查詢與更新

OmniHSS 透過 OmniSS7 的 HTTPS API 與 OmniHSS 透過 MAP 的 UpdateLocation 與 SendAuthenticationInfo 與 SendRoutingInfo

OmniSS7 HLR 透過 SS7/MAP 與 OmniHSS 連接 HSS 資料庫
OmniSS7 透過 SS7/MAP 與 OmniHSS 連接 HSS 資料庫
OmniHSS 簡介

IMSI 簡介

OmniHSS 提供 IMSI 與 MSISDN 對應 IMSI 卡

- 電話番号から IMSI を取得する
- **eSIM** 対応 eSIM 対応端末
- 電話番号から MSISDN を取得する
- **SIM** 対応 SIM 対応
- 電話番号から IMSI を取得する

番号

- 番号 IMSI から Ki/OPc
- 番号 IMSI から
- 番号 IMSI から
- OmniSS7 番号 IMSI から OmniHSS/OmniHSS
- 番号 IMSI から IMSI

番号 **IMSI** 番号

番号 MSISDN: +1-555-123-4567
└ IMSI 1: 310260123456789 (番号 - Milenage 番号)
└ IMSI 2: 208011234567890 (番号 - Milenage 番号)
└ IMSI 3: 440201234567891 (番号 - COMP128 番号)

番号 IMSI 番号 OmniHSS 番号 IMSI 番号 IMSI 番号

□□

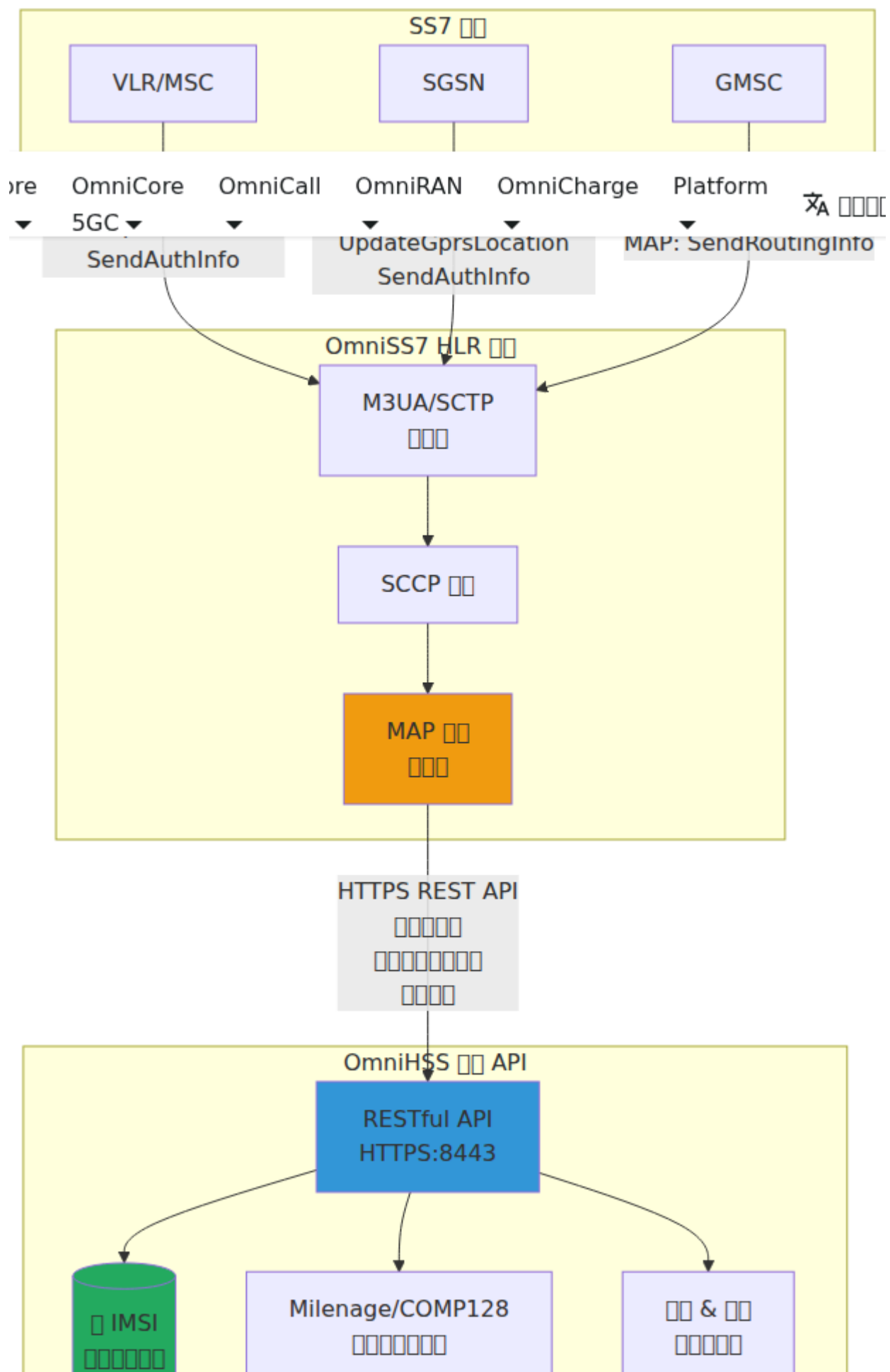
1. OmniHSS □□
2. □ IMSI □□
3. □□□ HLR □□□
4. □□ HLR □□
5. □□□□□□
6. □□□□□□
7. □□□□
8. CAMEL □□
9. □□□□□□□□
10. HLR □□
 - □□□□□□
 - SendRoutingInfo (SRI)
 - UpdateLocation / ISD
 - SendRoutingInfoForSM
 - □□□□□□

HLR

HLR OmniSS7

-
-
- VLR
- SMS

HLR ☐ ☐



HLR

OmniSS7 STP HLR SMSc `config/config.exs`

HLR

OmniSS7 HLR SMSc HLR

- `config/config.exs`
- `config :omniss7`
 - `map_client_enabled: true` MAP
 - `hlr_mode_enabled: true` HLR
 - `smsc_mode_enabled: false` SMSc
- HLR API GT ISD/CAMEL M3UA
-
-

`config/runtime.exs` HLR
`config/config.exs` `config/test.exs`
`config/config.exs`

HLR

HLR `config/config.exs`

```

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

  # OmniHSS 网络 API 配置
  hlr_api_base_url: "https://omnihss.example.com:8443",

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

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

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

  # 网络 ISD #2网络网络网络
  isd_send_ss_data: true,

  # 网络 ISD #3网络网络网络
  isd_send_call_barring: true,

  # CAMEL 网络 SendRoutingInfo 网络
  # CAMEL 网络网络网络
  camel_service_key: 11_110,

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

  # 网络 VLR 配置
  # 网络“网络”网络 VLR 网络网络
  # 网络网络 VLR 网络网络网络网络网络 SRI 网络
  # 网络网络网络网络网络 PRN 网络 MSRN
  # 网络网络网络 []网络 = 网络 VLR 网络网络
  home_vlr_prefixes: ["555123"],

```



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

項目名	データ型	初期値	説明
hlr_api_base_url	文字列	空	OmniCore API のベース URL
hlr_service_center_gt_address	文字列	空	HLR サービスセンターの GT アドレス
smsc_service_center_gt_address	文字列	空	SMS サービスセンターの GT アドレス
hlr_smsc_alert_gts	文字列	[]	HLR SMS サービスセンターの alert GT 一覧
hlr_alert_location_expiry_seconds	整数	172800	SMS alert の有効期限 (秒)
hlr_imsi_plmn_prefix	文字列	"50557"	MSISDN の PLMN 前缀 (IMSI 50557)
hlr_msisdn_country_code	文字列	"61"	MSISDN の国コード (IMSI 61)
hlr_msisdn_nsn_offset	整数	0	MSISDN の NSN オフセット (IMSI 0)
hlr_msisdn_nsn_length	整数	9	MSISDN の NSN 長さ (IMSI 9)

参数	数据类型	默认值	说明
isd_network_access_mode	枚举	:packetAndCircuit	接入模式
isd_send_ss_data	布尔	true	是否发送SS数据
isd_send_call_barring	布尔	true	是否发送呼叫限制
camel_service_key	字符串	11_110	发送的Service Key
camel_trigger_detection_point	枚举	:termAttemptAuthorized	CAMEL触发检测点
home_vlr_prefixes	字符串	[]	归属VLR前缀
local_ip	字符串		本地IP地址
local_port	字符串	2905	本地端口
remote_ip	字符串		远端IP地址
remote_port	字符串	2905	远端端口
routing_context	字符串	1	M3UA路由上下文

HLR

`hlr_mode_enabled: true` Web UI

- **SS7** -
- **SS7** - MAP
- - M3UA/SCTP
- **HLR** - HLR API + *HLR*
- -
- -

SMSc

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

OmniHSS OmniSS7 RESTful API

OmniHSS

OmniHSS

- IMSI
- Ki/OPc Milenage COMP128
- QoS
- VLR/MSC SGSN/GGSN
- CAMEL
- CLIP/CLIR
- /

OmniHSS Milenage COMP128
OmniSS7 **sendAuthenticationInfo** MAP

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

OmniHSS API

OmniSS7 HTTPS REST API OmniHSS

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

OmniSS7 SS7 MAP OmniHSS

- IMSI MSISDN
- Ki/OPc
- UpdateLocation
-

updateLocation MAP OmniSS7 OmniHSS VLR

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

hlr_service_center_gt_address UpdateLocation HLR VLR/MSC HLR

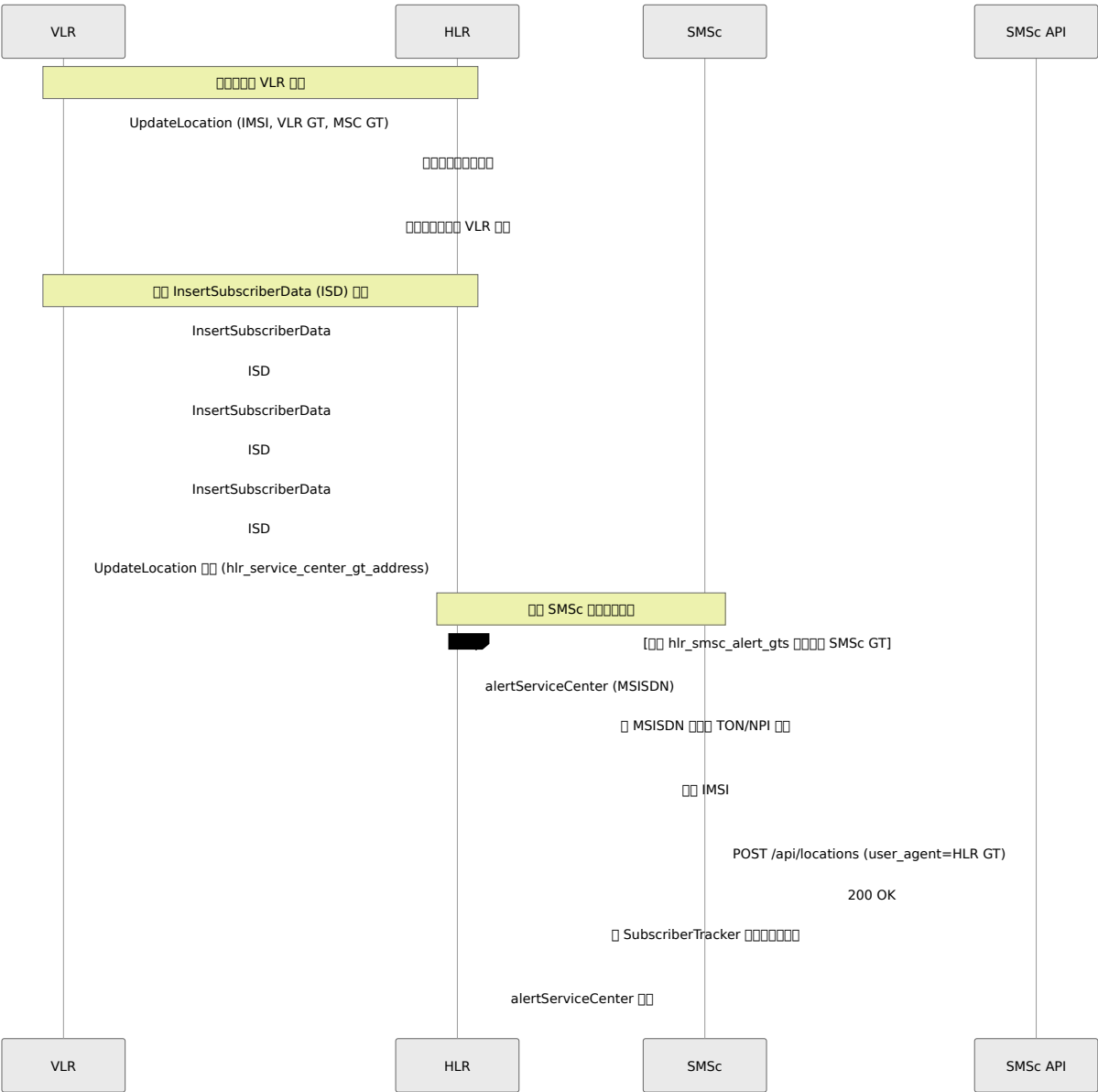
UpdateLocation HLR alertServiceCenterMAP 64 SSMSc SSMSc SMSc

SSMSc

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

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

[] []



[] []

如何调用 UpdateLocation 接口

1. HLR 中 `hlr_smsc_alert_gts` 字段的 SMSc GT 为 `alertServiceCenter`
2. 如何设置 MSISDN
3. HLR 中 `hlr_service_center_gt_address` 字段的 GT
4. SCCP 中 SSN=6 的 HLR 中 SSN=8 的 SMSc

SMSc 如何设置

- 将 **MSISDN** 中的 **TON/NPI** 从 “19123123213” → “123123213”
- 通过 `POST /api/locations`
- 设置 **API** 中的 `user_agent` 为 **HLR GT** 中的 HLR 地址
- 设置 `hlr_alert_location_expiry_seconds` 为 0
- 将 SMSc 设置为 0

如何

Web UI 中如何设置 `alertServiceCenter` 字段

1. 设置 “MSISDN” 字段
2. 设置 “SMSc GT” 字段
3. 设置 MSISDN 和 SMSc GT 中的 HLR GT 字段
 - SMSc GT 中 `hlr_smsc_alert_gts` 字段
 - HLR GT 中 `hlr_service_center_gt_address` 字段
4. 设置 “`alertServiceCenter`” 字段

如何设置 SMSc 字段中的 `UpdateLocation` 字段中的 `phx-blur` 字段

InsertSubscriberData (ISD) 接口

如何调用 `UpdateLocation` 接口中的 HLR 中的 **InsertSubscriberData** 接口中的 ISD 字段

如何设置 ISD 字段

ISD 接口

HLR 000000 3 0000 ISD 0000

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

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

2. **ISD #2**0000 - 00000SS0000

- 000000000000000000000000
- 0000
- 0000
- 0000
- 0000000000

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

- 00000000BAOC
- 0000000000BOIC
- 000000

0000

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

# 00 ISD #2000000000
isd_send_ss_data: true,

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

ISD 設定

isd_network_access_mode 設定

設定	説明	値
:packetAndCircuit	GPRS/LTE 両方	パケット - 電路
:packetOnly	GPRS/LTE	パケット
:circuitOnly	SMS	電路

ISD 設定

ISD 設定

ISD 設定 - 設定

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

ISD 設定

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

ISD 設定 + 設定

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

ISD 設定

ISD 設定 UpdateLocation

```

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

- 
- 
- 

□□□□

```
# CAMEL 000000 SendRoutingInfo 0000
# CAMEL 0000000000
camel_service_key: 11_110,

# CAMEL 000000
# 000:termAttemptAuthorized:tBusy:tNoAnswer:tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,
```

□□□□

camel_service_key gsmSCF CAMEL

項目	内容
11_110	11月11日
100	100円
200	200円
300	300円
400	400円
500	500円
600	600円
700	700円
800	800円
900	900円
1000	1000円

□□□□□

```
# 000000000000
camel_service_key: 11_110,

# 00 VPN 00
camel_service_key: 300,
```

00000

camel_trigger_detection_point 000000000000 CAMEL 000

000	00	0000
:termAttemptAuthorized	000000000000	000000000000
:tBusy	0000	000000
:tNoAnswer	00000	00000000
:tAnswer	0000	0000000000

00000

00000000000000000000

```
camel_trigger_detection_point: :termAttemptAuthorized,
```

00000000000000000000

```
camel_trigger_detection_point: :tBusy,
```

00000000000000000000

```
camel_trigger_detection_point: :tAnswer,
```

0 CAMEL 0 SRI 00

SendRoutingInfo CAMEL

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

GMSC gsmSCF CAMEL

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

VLR VLR

HLR **SendRoutingInfo** SRI " " VLR
VLR

VLR

- VLR CAMEL SRI
- VLR PRN MSRN SRI

□□

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

□□□□□

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

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

□□□□□

1. □□□□□□□□□□

□□□□□ VLR □□□□□□□□□□□□□□□□

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

2. □□□□□□□□□□ **PRN**□

□□□□□ VLR □□□□□□□□□□□□□□□□

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

HLR 0000 API 00000000

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"
- 00000MSRN0: "49170999888777"

00000000

00000 **SRI** 00

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

00000 **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 VLR PRN
- PRN PRN
- PRN
- PRN PRN ??? PRN

PRN

PRN

- PRN home_vlr_prefixes PRN VLR PRN
- PRN VLR PRN

PRN PRN

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

PRN SRI PRN MSRN PRN

- PRN PRN PRN
- PRN PRN PRN extract_msrn_from_prn/1

HLR PRN

PRN MAP PRN

OmniSS7 PRN TCAP PRN MAP PRN

번호	기능명	기능 설명
2	updateLocation	VLR에서 ISD 번호 + alertServiceCenter
23	updateGprsLocation	SGSN에 위치 정보 업데이트
3	cancelLocation	VLR에서 위치 정보 삭제
7	insertSubscriberData	Subscriber Data를 VLR에 저장
22	sendRoutingInfo (SRI)	MSRN/CAMEL 번호를 전송
45	sendRoutingInfoForSM (SRI-for-SM)	IMSI + SMSC GT를 전송하여 SMS 전송
44	MT-Forward-SM	발신 SMS 전송
46	MO-Forward-SM	수신 SMS 전송
47	reportSM-DeliveryStatus	SMS-GMSC에 전송 상태 보고
64	alertServiceCenter	발신 SMS에 서비스 센터 정보
66	readyForSM	MSC/SGSN가 SMS 전송을 준비
56	sendAuthenticationInfo	인증 정보 전송
57	restoreData	VLR에서 데이터 복원
67	purgeMS	HLR에서 데이터 삭제
59	processUnstructuredSS-Request	USSD 요청 처리
71	anyTimeInterrogation (ATI)	위치 조회/확인

項目 ID	項目 名	項目 説明
10	registerSS	登録処理

登録処理 facilityNotSupported 21 登録処理

登録処理

登録処理 HLR 登録処理

SendRoutingInfo (SRI) 登録

登録処理

HLR 登録処理 VLR 登録処理

登録処理 **CAMEL** 登録

登録処理 VLR 登録処理 home_vlr_prefixes 登録

登録

項目	項目	項目
IMSI	OmniHSS API	OmniHSS から取得した IMSI
VLR 番号	OmniHSS API	現在接続している VLR circuit_session.assigned_vlr
提供状況	有	notProvidedFromVLR
extendedRoutingInfo	-	camelRoutingInfo
gsmSCF 番号	OmniHSS API	現在接続している MSC circuit_session.assigned_msc
サービスキー	config.exs	CAMEL サービスキー camel_service_key
トリガー検出ポイント	config.exs	CAMEL camel_trigger_detection_point
CAMEL サービス	有	CAMEL サービス
接続先	有	gsmSCF 接続先

MSRN 番号

VLR 番号 home_vlr_prefixes

番号

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

住所MSRN

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

住所 VLR 住所
  → 住所 CAMEL 住所

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

住所

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

住所

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

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

UpdateLocation InsertSubscriberData

VLR

UpdateLocation

HLR	config.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	config.exs	項目/項目項目 項目 isd_network_access_mode 項目	:packetAndCircuit

InsertSubscriberData #2項目項目項目 - 項目

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

ISD #2 項目項目

- 項目項目項目項目SS 項目 21
- 項目項目項目項目SS 項目 41
- 項目項目項目項目SS 項目 42

- 國際號碼SS 00 620
- 國際號碼SS 00 430
- 國際號碼SS 00 510
- CLIP/CLIR 00

InsertSubscriberData #3國際號碼 - 00

00	00	00	國際號碼
國際號碼	00	國際號碼	isd_send_call_barring: true
BAOC	00	國際號碼SS 00 1460	國際號碼
BOIC	00	國際號碼SS 00 1470	國際號碼
國際號碼	00	國際號碼	國際號碼

ISD 國際號碼

- ISD #1國際號碼 - 國際號碼
- ISD #2國際號碼 isd_send_ss_data: true 國際號碼
- ISD #3國際號碼 isd_send_call_barring: true 國際號碼

SendRoutingInfoForSM (SRI-for-SM) 國際號碼

國際號碼 SMS 國際號碼 MSC/SMSC 國際號碼 SMSc 國際號碼 SMS 國際號碼 HLR 國際號碼 SRI-for-SM 國際號碼

HLR 國際號碼 **SCCP** 國際號碼 **GT** 國際號碼

1. 國際號碼 IMSI 國際號碼 IP-SM-GW 國際號碼 GT 國際號碼 HLR 國際號碼 IMSI 國際號碼 MT-ForwardSM 國際號碼 IP-SM-GW

項目	項目	項目	説明	値
IMSI	IMSI	MSISDN MSISDN IMSI	PLMN_PREFIX + zero_padded_MSISDN	"001001"
サービス センター	config.exs	SMSC GT アドレス SMS サービス	smsc_service_center_gt_address	"555123"

2. サービスセンター / IP-SM-GW サービスセンター GT アドレス
home_smc_gts サービスセンター HLR サービスセンター IMSI サービスセンター MSC サービスセンター
SMSC サービス MT-ForwardSM サービス MSCサービス MSCサービス

項目	項目	項目	項目
IMSI	HLR サービス	サービス IMSI	"505570123456789"
サービス センター	HLR サービス	サービス UpdateLocation サービス serving_msc	"5559876543"

サービス config/config.exs サービス

```
# 0000 GT 0000000000 SRI-for-SM 000000
# 000000 SMSc 0 MT-ForwardSM 00000000
smsc_service_center_gt_address: "5551234567", # 00

# 000000000000 SMSc / IP-SM-GW 000 SCCP 00 GT0
# 00000 SRI-for-SM 0000000 MSC0000000000
# 000000000000000000000000000000000000
home_smsc_gts: ["5551234567"], # 00000 000

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

AAA HLR AAAA serving_msc AAAA MSC AA HLR AAA
MAP updateLocation AAA MSC AAAA AAAA AAAA AAAA HLR AA GSUPAAAAA
AAAAA

MSISDN ↔ IMSI

□□□□□

OmniSS7 SRI-for-SM MSISDN IMSI

- **hlr_imsi_plmn_prefix** IMSI MCC+MNC “50557”
MCC=505 MNC=57
- **hlr_msisdn_country_code** IMSI→MSISDN “61”
“1”
- **hlr_msisdn_nsn_offset** MSISDN NSN 0
MSISDN
- **hlr_msisdn_nsn_length** MSISDN NSN

MSISDN ↔ IMSI

□□□□ **MSISDN** □ **IMSI** □□□□

MAP 消息 **SendRoutingInfoForSM** SRI-for-SM HLR 消息 **IMSI**
 消息 **SMS** Sc 消息 **MSISDN**

□□□□□□

- SSMc □□ SRI-for-SM□□□ MSISDN□□□□“5551234567”□
- HLR □□□□□□□□□□□□□□□□ IMSI
- HLR □ SRI-for-SM □□□□□ IMSI
- SSMc □□□□ MT-ForwardSM □□□ MSC/VLR □□□□ IMSI

OmniSS7 □□□ - □□ IMSI□

OmniSS7 □□□□□□□□□□□ MSISDN □□ □□ IMSI□□□□□□□□□□□□□□□□ MSISDN □
IMSI □□□□□□□□□□□□□□□□

1. □□□□□□ HLR □□□□□□□□□□ IMSI □□□□□ SS7 □□□□□ SRI-for-SM □□□□□
2. □□□□□ SRI-for-SM □□□□□□□□□□ HLR □□□□□ IMSI □□ - IMSI □□□□ MSISDN □□□□□

□□□□□

MSISDN □□□□□ IMSI □□□□□□□□MCC+MNC □□□□□□□

IMSI = PLMN_PREFIX + zero_padded_MSISDN

□□□

- **PLMN_PREFIX**□MCC + MNC□□□□□“001001”□□□□□□□□
- **MSISDN**□□□□□□□□□□□□□□□□
- □□□□□□□□□□□ IMSI □□□□ 15 □□□

□□□□□

```

# []
plmn_prefix = "001001" # MCC 001 + MNC 01

# [] SRI-for-SM [] MSISDN TBCD []
msisdn = "555123456" # 9 []

# [] 1 []
subscriber_digits = 15 - String.length("001001") # = 9 []

# [] 2 [] MSISDN []
padded_msisdn = String.pad_leading("555123456", 9, "0") # =
"555123456" []

# [] 3 [] PLMN [] + [] MSISDN
imsi = "001001" <> "555123456" # = "001001555123456" [] 15 []

```

[][][]

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 如何从 IMSI 获取 MSISDN

```
# 如何从 SRI-for-SM 获取 IMSI
imsi = "001001555123456"

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

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

数据表

15 位 IMSI	PLMN 前缀	15 位 PLMN 前缀	15 位 PLMN 前缀	20 位 MSISDN
"001001555123456"	"001001"	"555123456"	"555123456"	"555123456"
"0010010000000099"	"001001"	"0000000099"	"99"	"99"
"0010019999999999"	"001001"	"9999999999"	"9999999999"	"9999999999"

数据表

- 如何从 MSISDN 获取 IMSI
- 如何从 IMSI 获取 MSISDN
- 如何从 hlr_imsi_plmn_prefix
- 如何从 IMSI 获取
- 如何从 API 获取
- 如何从 15 位 IMSI 获取 15 位

MSISDN 数据表

HLR 如何 SRI-for-SM 获取 MSISDN 如何 TBCD 如何

1. **TBCD** 如何 TBCD 如何 TON/NPI 如何“91”
2. 如何

3. 00000000000000000000000000000000
4. 00000 PLMN 00 + 0000 MSISDN

00000

0 SRI-for-SM 00000000 IMSI 0000000000000000 HLR 0000000000 IMSI0000000000
0000000000000000 IMSI 00000000000000000000000000000000 UpdateLocation 0
SendAuthenticationInfo 00000

0000

1. SMSc → HLR: SRI-for-SM 00
 - MSISDN0TBCD00"91123456789"000 TON/NPI0
2. HLR 000
 - TBCD 000"91123456789"
 - 000000"91123456789"011 0000
 - 00 9 0000"555123456"00000 9 00
 - 00 PLMN0"001001" + "555123456" = "001001555123456"
 - 000000 SMSC GT0"5551234567"
3. HLR → SMSc: SRI-for-SM 00
 - IMSI0"001001555123456"000000 15 00
 - 00000000"5551234567"000 MT-ForwardSM 0000
4. SMSc 0 MT-ForwardSM 000 "5551234567"0IMSIN 0 "001001555123456"

000

00000 config/config.exs 0000

```
# PLMN 000MCC0001 = 000000+ MNC001 = 000000
hlr_imsi_plmn_prefix: "001001",

# NSN 00000 MSISDN 00000000
hlr_msisdn_country_code: "1",      # 00000000IMSI→MSISDN0
hlr_msisdn_nsn_offset: 1,          # 00 1 00000
hlr_msisdn_nsn_length: 10          # 00 10 0 NSN
```

NSN 00000

MSISDN "99988000088" NSN "88000088"

- hlr_msisdn_nsn_offset NSN
- hlr_msisdn_nsn_length MSISDN

		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"  
NSN: "088000088"  
NSN: "88000088"  
MSISDN: "+999" + "88000088" = "+99988000088"
```

API 接口 - SRI-for-SM 接口

API 接口

接口

接口	接口	接口
OmniHSS API	OmniHSS 接口	IMSI/MSISDN 接口 circuit_session 接口 VLR/MSC
config.exs	OmniSS7 接口	smc_service_center_gt_address camel_service_key isd_network_access_mode
接口	接口	接口SS 接口
接口	接口 MAP 接口	接口 UpdateLocation 接口 IMSI/MSISDN
接口	接口	SRI-for-SM 接口 IMSI/hlr_imsi_prefix + NSN

接口

config/config.exs 接口

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

config/config.exs 接口

- camel_service_key - 接口 11_110
- camel_trigger_detection_point - 接口 :termAttemptAuthorized

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

OmniHSS

OmniHSS REST API

- IMSI MSISDN
- VLR/MSC
-
-

OmniSS7

- ←
-
- MAP
-
-

OmniHSS OmniHSS OmniHSS IMSI

OmniSS7 Omnitouch

MAP

←

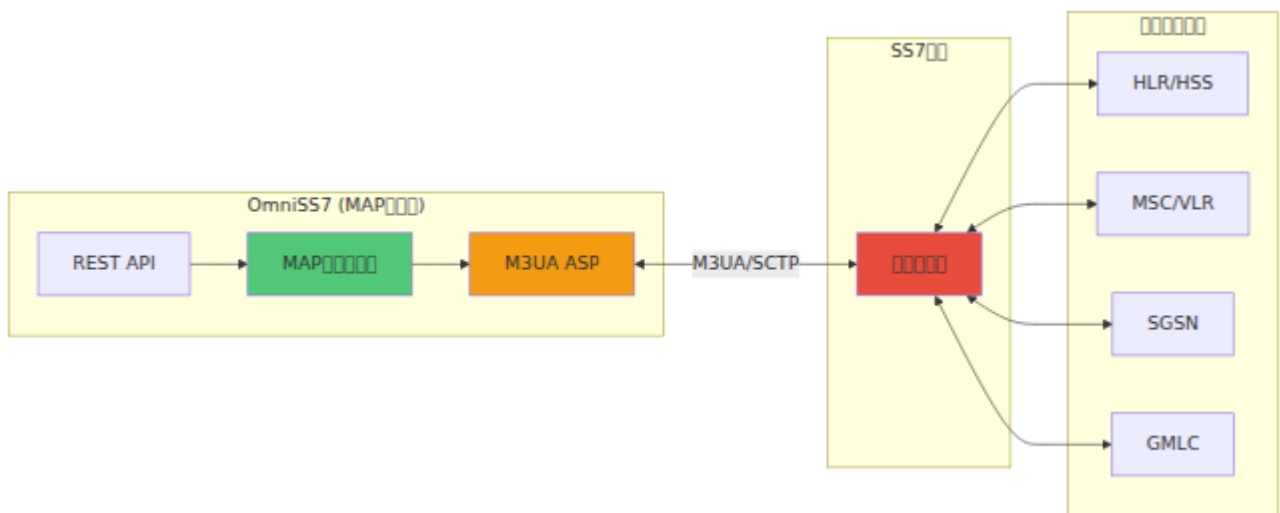
OmniSS7 **MAP** OmniSS7 ASP SS7
HLR MSC/VLR SGSN GMLC MAP 3GPP TS
29.002

1. MAP
2. MAP
3. MAP
- 4.
5. API
- 6.
- 7.

MAP

MAP是OmniSS7的ASP，SCTP M3UA STP SGP
MAP通过HTTP API与TCAP/MAP SCCP M3UA SS7
OmniSS7

- MS
-
- SRI-for-SM MO/MT-Forward SM
-
- /
- / / SS
- LCS GPRS



MAP

MAP config/runtime.exs M3UA M3UA

```
config :omniss7,  
  # MAP  
  map_client_enabled: true,  
  
  # MAPM3UAASPSTP/SGP  
  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  
  }
```

map_client_m3ua

項目	型	デフォルト	説明
mode	enum	-	M3UAをASPとして動作させるか、STP/SGPとして動作させるか
callback	void (*)	-	{Module, Function, Args}で指定されたM3UAの処理関数を呼び出す。{MapClient, :handle_payload, []}
process_name	string	-	M3UA ASPの名称
local_ip	string	-	SCTPのローカルIPアドレス {10, 0, 0, 100}
local_port	uint16_t	2905	SCTPのローカルポート番号IANAでM3UAに割り当てられている2905
remote_ip	string	-	STP/SGPのIPアドレス
remote_port	uint16_t	2905	STP/SGPのSCTPポート番号
routing_context	string	-	M3UAのルーティングコンテキスト

map_client_enabled: false MAPのクライアント map_client_m3ua を有効にする

SCCPのポート番号/SSNは10000から10009まで指定可能

MAP

POST /api/<endpoint> 3GPP TS 29.002 MAP API SS7

	/api/updateLocation	2	VLRHLR
	/api/cancelLocation	3	VLR
	/api/prn	4	MSCMSRN
MS	/api/purgeMS	67	HLR
	/api/sendIdentification	55	VLRIMSI/
	/api/reset	37	HLRVLR
	/api/restoreData	57	VLRHLR

0000000000

00	00	00 0	00
00000 0	/api/sri	22	00HLR00000000 00000
00000 0	/api/provideSubscriberInfo	70	0VLR000000/0 00
00000 0	/api/deleteSubscriberData	8	00VLR00000000
0000	/api/anyTimeInterrogation	71	gsmSCF0HLR0 000000/000
00000 0	/api/anyTimeSubscriptionInterrogation	62	00HLR00000000
0000	/api/anyTimeModification	65	00HLR00000000
00000 0000	/api/noteSubscriberDataModified	5	HLR00gsmSCF 000000
00MM 00	/api/noteMM-Event	89	0gsmSCF0000 00000

□□

□□	□□	□□ □	□□
□□□□□□□□	/api/sri-for-sm	45	□□HLR□□□□□□□□□□□□ □□
MT-□□□□	/api/MT-forwardSM	44	□□□□□□□□□□□□ MSC/SGSN□
MO-□□□□	/api/forwardSM	46	□SMSC□□□□□□□□□□
□□□□ □PDU□	/api/sendSM, /api/deliverPDU	44	□□□□□SMS-DELIVER PDU□
□□□□□□□□	/api/reportSM- DeliveryStatus	47	□HLR□□□□□□□□□□
□□□□□□	/api/readyForSM	66	□□HLR□□MS□□□□□□□□
□□□□□□	/api/alertServiceCentre	64	□□SMSC□□□□□□□□□□

□□□□□□□□

□□	□□	□□□	□□
□□□□□□□□	/api/send-auth-info	56	□HLR□□□□□□□□□□
□□IMSI	/api/sendIMSI	58	□MSISDN□□□HLR□□IMSI□
□□IMEI	/api/checkIMEI	43	□□□□□□□□

□□□□

□□	□□	□□ □	□□
□□SS	/api/registerSS	10	□□□□□□□□□□ □/CFU□□
□□SS	/api/eraseSS	11	□□□□□□□
□□SS	/api/activateSS	12	□□□□□□□
□□SS	/api/deactivateSS	13	□□□□□□□
□□SS	/api/interrogateSS	14	□□□□□□□□□□
□□□□	/api/registerPassword	17	□□SS□□□
□□□□	/api/getPassword	18	□□/□□SS□□  
□□□□SS□ □	/api/forwardCheckSS- Indication	38	□VLR□□□□□□□□

3GPP TS 29.274 LCS and GPRS

Interface	Service	Service ID	Notes
Uu	/api/sendRoutingInfoForLCS	85	GMLC to HLR for LCS
Uu	/api/provideSubscriberLocation	83	HLR to GMLC
Uu	/api/subscriberLocationReport	86	GMLC to HLR
Uu	/api/sendRoutingInfoForGprs	24	HLR to GPRS
Uu	/api/updateGprsLocation	23	HLR to SGSN

LCS is a 3GPP standard for location services. It defines the MAP interface between HLR and GMLC for LCS. The GMLC provides the LCS service to the user. The HLR provides the location information to the GMLC. The GPRS service is also defined for location services. The GPRS service is used for location services. The GPRS service is used for location services. The GPRS service is used for location services.

CAP / CAMEL

API	API	API
CAP InitialDP	/api/cap/initialDP	gsmSCF CAMEL
CAP Connect / Continue / ReleaseCall / ApplyCharging	/api/cap/connect, /api/cap/continue, /api/cap/releaseCall, /api/cap/applyCharging	CAMEL CAMEL
SCCP	/api/raw/sccp	SCCP
M3UA	/api/raw/m3ua	M3UA

SRI-for-SM

HLRHLRSRI-for-SM

SCCPSMScSRI-for-SMGTMISDNSSN
6/HLRSTPGTTSMScHLR GTSR
HLRGTTHLRMSC

名称:omniss7	类型	值
hlr_gt_address	SRIRGT	"61455501111"
home_msisdn_prefixes	MSISDNHRLR GT	["6777"]

home_msisdn_prefixesSRIRGTSSN 6
MSISDNGTSRIRGTfor-SMcccp_opts:called_party,
:called_ssnSCCP

POST /api/sri-for-sm

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

```
{
  "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"}'
```

POST /api/sri

POST /api/sri

Request

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

POST /api/prn

POST /api/prn

Request

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

POST /api/send-auth-info

POST /api/send-auth-info

Request

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

Response


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

□□□□

□□□□□VLR□□□HLR□HLR□□□□□□□□□□□□□□□□□□□□HLR□□□□□□□□□□

□□□ POST /api/updateLocation

□□□

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

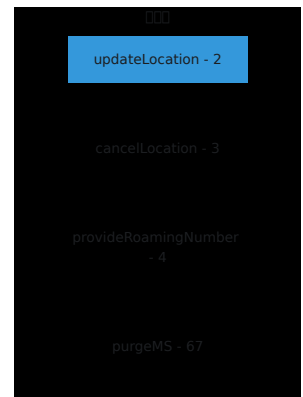
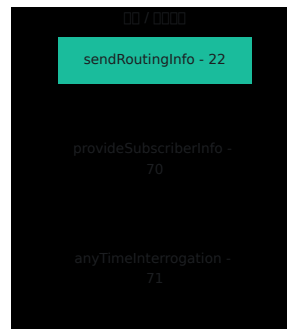
□□□□□**ATI**□

□□□ POST /api/anyTimeInterrogation

□□□

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

□□□□



API

Swagger UI

Swagger UI

1. `http://your-server/swagger`
2. `/api/sri-for-sm`
- 3.
4. `SS7`

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

API

□□	□□
200	□□□□□□□□□□
400	□□□□□□□□□□
504	□□□□□□□□□□SS7□□□□□10□□□

[illegible]

MAP

Prometheus /metrics

map_requests_total operation sri
sri_for_sm prn authentication_info update_location

map_request_errors_total operation

map_request_duration_milliseconds operation

map_pending_requests

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])

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

# 
map_pending_requests
```

MAP

504

API 504 HLR/MS

- M3UA STP/SGP
- STP
- SCCP / GT SSN

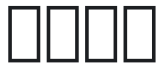
1. M3UA STP
2. STP SCTP
3. HLR SSN 6
4. SCCP

SCCP /

SCCP UDT

- STP
- HLR SSN 6
- /

- GT STP /
-
-



- ← 送信機
- 送信機が受信機にメッセージを送信
- 送信機がWeb UI/APIを通じて送信
- STPが送信機にメッセージを送信
- HLRが送信機にHLRメッセージを送信
- 送信機が受信機にメッセージを送信
- CAMELが送信機にCAP/CAMELメッセージを送信

OmniSS7 Omnitouch

OmniSS7 (SMSc) Overview

← Introduction

OmniSS7 provides a **SMSc (SMSC)** interface for **OmniMessage** to send and receive SMS messages.

OmniMessage

OmniSS7 SMSc interface **SS7** interface **OmniMessage** interface SMS interface

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

OmniMessage

OmniMessage interface SMS interface

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

OmniMessage interface OmniSS7 interface HTTPS API interface
OmniMessage interface interface interface interface

OmniSS7 SMSc interface interface interface interface interface interface interface
OmniMessage interface interface interface interface SS7/MAP interface interface interface interface interface interface interface
interface interface **OmniMessage** interface



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



OmniSS7 SS7

OmniMessage

□□□□□ □ OmniSS7 □□□□ SSMSc □□□□□□

- **MT-SMS** 發送 SMS 訊息
- **MO-SMS** 接收 SMS 訊息
- 發送 SMS 訊息
- 接收 SMS 訊息
- 發送 SMS 訊息

The diagram illustrates the SS7 to IP migration architecture, showing the flow of signaling and data between various components:

- SS7 to IP Migration:**
 - SS7 []** (top right) connects to **HLR []**.
 - HLR []** sends **MAP: SRI-for-SM** to **M3UA/SCTP []** in the **OmniSS7 SMSc []** block.
 - OmniSS7 SMSc []** contains **M3UA/SCTP []**, **SCCP []**, and **MAP []**.
 - M3UA/SCTP []** sends **MAP: MT-ForwardSM** to **MAP []**.
 - MAP []** sends **MAP: D-ForwardSM** to **MSC/VLR []** and **GMSC []**.
- IP to SS7 Migration:**
 - RESTful API HTTPS:8443 []** (bottom center) receives input from **OmniMessage []**.
 - OmniMessage []** receives input from **MAP []** and **SCCP []**.
 - RESTful API HTTPS:8443 []** sends output to **OmniSS7 SMSc []** and **OmniMessage []**.

OmniSS7 数据库连接池STP、HLR、SMSc数据库连接池 **config/config.exs** 数据库连接池

OmniSS7 SSMSc SSMSc SSMSc

1. 打开 `config/config.exs`
2. 打开 `config :omniss7` 配置项
 - `map_client_enabled: true` 打开/MAP 功能
 - `smsc_mode_enabled: true` 打开 SMSG 功能
 - `hlr_mode_enabled: false` 关闭 HLR 功能
3. 打开 SMSG 功能 API 接口 SMSG 接口 GT 功能 M3UA 功能
4. 打开 配置项
5. 打开 配置项

打开 `config/runtime.exs` 配置项 `smc_mode_enabled` 打开 SMSG 功能
 打开 `config/config.exs` 打开 `config/test.exs` 配置项
 打开 `config/config.exs`

SMSG 功能

打开 SMSG 功能 `config/config.exs` 配置项

```

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://omnimessage.example.com:8443",
  # 配置 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_smsgw: "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
  }

```

Web UI 配置 `config :control_panel` 配置 `config/config.exs` 配置

`use_additional_pages` 配置 OmniSS7 配置 16 配置

配置 peer-status 配置 **"Peers"**

```

config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "SS7 []"},
    {SS7.Web.TestClientLive, "/client", "SS7 [][]"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "Peers"},
    {SS7.Web.SCTPConnectionsLive, "/sctp", "SCTP []"},
    {SS7.Web.RoutingLive, "/routing", "[]"},
    {SS7.Web.RoutingTestLive, "/routing_test", "[][][]"},
    {SS7.Web.SmscLinksLive, "/smc_links", "SMSc []"},
    {SS7.Web.SmscSubscribersLive, "/smc_subscribers", "SMSc []
    []"},
    {SS7.Web.HlrLinksLive, "/hmr_links", "HLR []"},
    # ... [][]CAMEL[]MSISDN/IMSI [][]
  ],
  page_order: [
    "/application", "/configuration", "/events", "/logs",
    "/client", "/m3ua",
    "/routing", "/routing_test", "/smc_links",
    "/smc_subscribers", "/hmr_links",
    # ... [][]
  ]

```

[][][]

[][][] ? ? ? [] []

項目	データ型	デフォルト値	説明
<code>smsc_api_base_url</code>	文字列	<code>""</code>	OmniMessage 通信 API の URL
<code>smsc_name</code>	文字列	<code>"{hostname}_SMS"</code>	SMS 通信 SMS 通信の識別子
<code>smsc_service_center_gt_address</code>	文字列	<code>""</code>	サービスセンターの GT アドレス
<code>auto_flush_enabled</code>	ブール値	<code>true</code>	自動フラッシュの有効化
<code>auto_flush_interval</code>	整数	<code>10_000</code>	自動フラッシュのインターバル (ミリ秒)
<code>auto_flush_dest_smsc</code>	文字列	<code>nil</code>	自動フラッシュの宛先 SMS 通信の識別子。 <code>nil</code> = 送信元 SMS 通信の識別子
<code>auto_flush_tps</code>	整数	<code>10</code>	自動フラッシュの TPS (メッセージ/秒)
<code>local_ip</code>	文字列	<code>""</code>	SMS 通信の IP アドレス
<code>local_port</code>	整数	<code>2905</code>	SCTP ポート
<code>remote_ip</code>	文字列	<code>""</code>	SS7 通信 STP IP アドレス

項目	単位	値	説明
remote_port	ポート	2905	SCTP ポート
routing_context	コンテキスト	1	M3UA ルーティング IC

SMS Sc 設定

smc_mode_enabled: true  map_client_enabled: true Web UI 設定

- SS7 - 設定
- SS7 - MAP 設定
- Peers - M3UA/SCTP 設定
- STP - 設定
- STP - 設定
- SMSc 設定 - SMSc API + SMS 設定 ← SMSc 設定
- 設定
- 設定

HLR 設定

設定

- SMSc 設定 map_client_enabled: true 設定
- OmniMessage 設定 OmniMessage API 設定 smc_api_base_url 設定
- 設定 ~85 設定 SMS.FrontendRegistry 設定 OmniMessage 設定
- API 設定 OmniMessage API 設定 5 設定
- MAP 設定 SRI-for-SM 10 設定 MT-ForwardSM 30 設定 2G/CS 設定 VLR 設定
- 設定 SMS 設定
- STP 設定 M3UA 設定/設定 MAP SMS 設定
- 設定

- **Web UI**  Web  **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://omnimessage.example.com:8443",
  # hostname_SMSC hostname_SMSC
  smsc_name: "omni-smsc01",
  # GT
  smsc_service_center_gt_address: "5551234567"
```

□□□□□

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

項目名

項目名 OmniMessage 項目名 SMS.FrontendRegistry 項目名 ~85 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~85 項目名@registration_interval 85_000 項目名
- 項目名 smsc_mode_enabled: true 項目名

項目名

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

スクリプトは `sm-sc_name` 変数にホスト名を代入して `"{hostname}_SMSc"` とします。

OmniMessage API

OmniSS7 は SS7 ネットワークを MAP 形式に変換する OmniMessage API を提供します。

- 送信元アドレスを指定
- 送信元アドレス (MO) を指定
- 送信元アドレス (MT) を指定
- 送信元アドレスを指定/受信
- 送信元アドレスを指定

URI	Method	Request	Response
/api/frontends	POST	JSON	<pre>{ "frontend_name": "...", "configuration": "...", "frontend_type": "...", "hostname": "...", "uptime_seconds": ... }</pre>
/api/messages_raw	POST	JSON SMS TPDU	<pre>{ "rp_originator": "...", "rp_destination": "...", "direction": "...", "message_body": "...", "send_time": "...", "smc_node_name": "...", "source_smc": "...", "source_type": "..." }</pre>
/api/messages	GET	JSON	<pre>{ "smc": <smc_r> }</pre>
/api/messages/{id}	PATCH	JSON	<pre>{ "deliver_time": "...", "dest_smc": "..." }</pre>
/api/messages/{id}	PUT	JSON	<pre>{ "dest_smc": null }</pre>
/api/locations	POST	JSON	<pre>{ "msisd": "...", "...", "location": "...", "ims_capable": true, "csfb": false, "...", "user_age": "...", "ran_location": "...", "...", "imei": "...", "registered": "..." }</pre>
/api/events	POST	JSON	<pre>{ "message_id": "...", "name": "...", "description": "..." }</pre>

URI	Method	Request	Response
/api/status	GET	Request Header sm-sc-api-status-path Request Body /api/status SMSC API Client	-

API Overview

The API returns JSON responses.

- Returns HTTP 200-201 JSON responses.
- Returns HTTP 4xx/5xx responses for errors.
- Returns ISO 8601 timestamps "2025-10-21T12:34:56Z".
- Returns ID.

API Endpoints

SMS API Endpoints

1. SMSC.APIClient

The API returns OmniMessage via HTTP API.

- frontend_register/4 - Returns OmniMessage.
- insert_message/1 - Returns SMS via /api/messages_raw.
- insert_location/9 - Returns location.
- get_message_queue/2 - Returns message queue.
- mark_dest_sm-sc/3 - Returns destination.
- add_event/3 - Returns event.
- flush_queue/2 - Returns SRI-for-SM + MT-forwardSM.
- auto_flush/2 - Returns auto flush.

2. SMS.FrontendRegistry

□□□□□□□□□□□□

- □□□□□□□
- □ ~85 □□□□□
- □□□□□□ `smsc_name` □□□□□□□□
- □□□□□□□□□□□□□□

3. SMS.Utils

SMS □□□□□□□□

- `generate_tp_scts/0` - □□ TPDU □□□ SMS □□□
-

□□□□□

□□ **SMS** □□□□□□□

M3UA 与 SCTP 接口

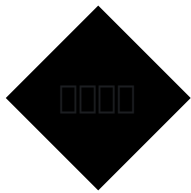
M3UA 接口

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 HSS API
IMSI VLR MT-SMS SRI-for-SM **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

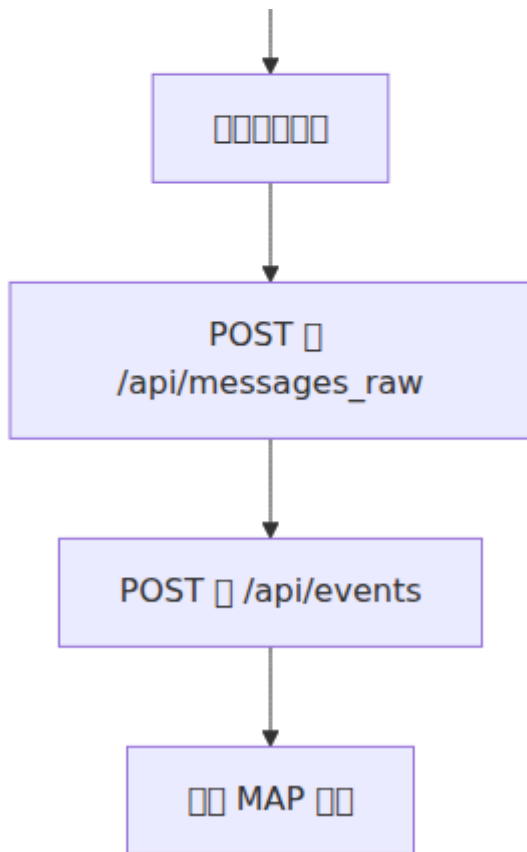
消息 MAP 消息

消息

Forward-SM

消息 SMS TPDU

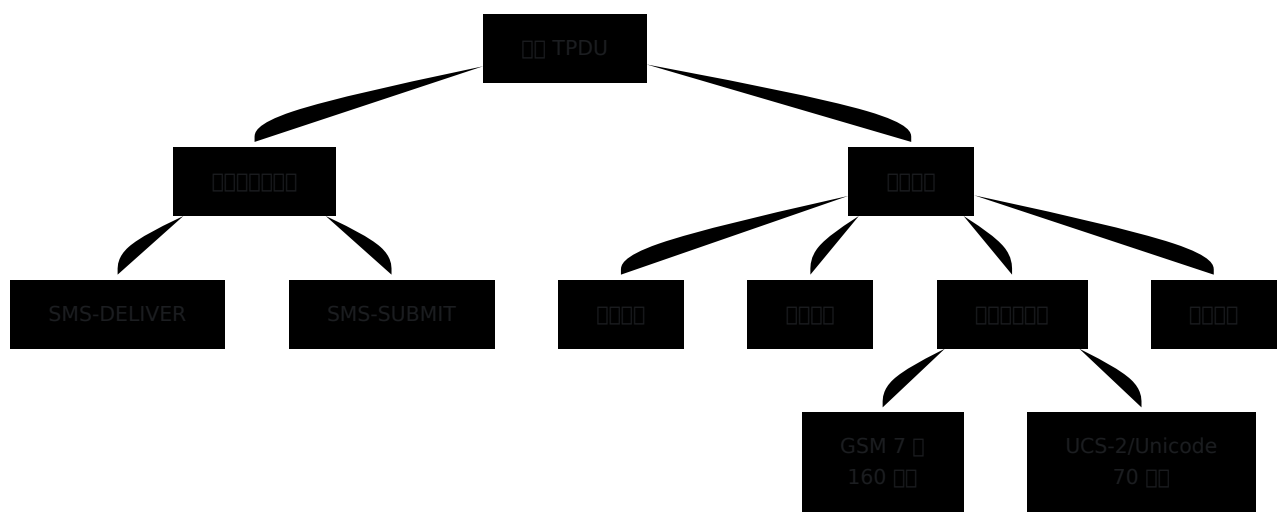
消息



Sequence

- **VLR** sends HSS API request to VLR containing 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** via IP-SM-GW. MT-ForwardSM is sent to VLR via SRI-for-SM.
- HSS API returns VLR ID to HLR. `alertServiceCenter` is returned.
- **SRI-for-SM** returns HSS API ID. HSS returns SRI-for-SM ID. HLR returns IMSI. HLR returns **SRI-for-SM**.
- VLR sends MT-ForwardSM to SCCP. VLR returns HSS API ID to VLR.
- **MT-ForwardSM** returns MT-ForwardSM ID. **30** is returned. 2G/CS VLR ID.

SMS TPDU 结构



短信服务

SMS Sc 服务 HLR 中 **alertServiceCenter** 参数用于指示短信服务的地址。

HLR 中 **alertServiceCenter** 参数指示 HLR 中短信服务的地址。

HLR 中 **alertServiceCenter**

HLR 中 **UpdateLocation** 消息中 VLR/MS Sc 向 HLR 发送 **alertServiceCenter** MAP 消息 64 字节 SMS Sc 地址。

HLR

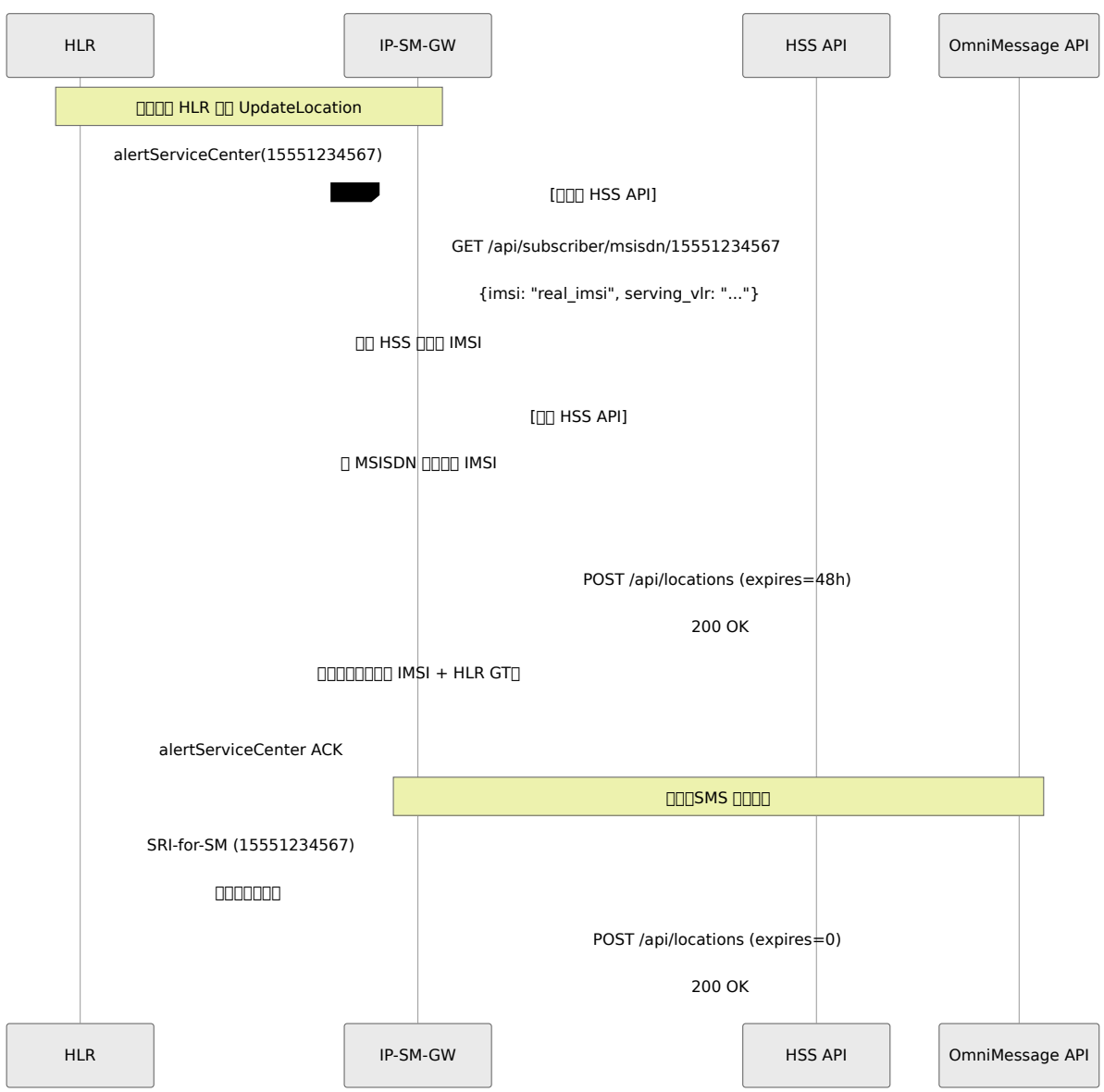
HLR 中 **HLR**

```
config :omniss7,
  # SMS Sc 中 alertServiceCenter 参数 48 字节
  hlr_alert_location_expiry_seconds: 172800
```

HLR

HLR 中 **alertServiceCenter** 参数

□□□



API □□

POST /api/locations

```
{
  "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 SSMSc HLR

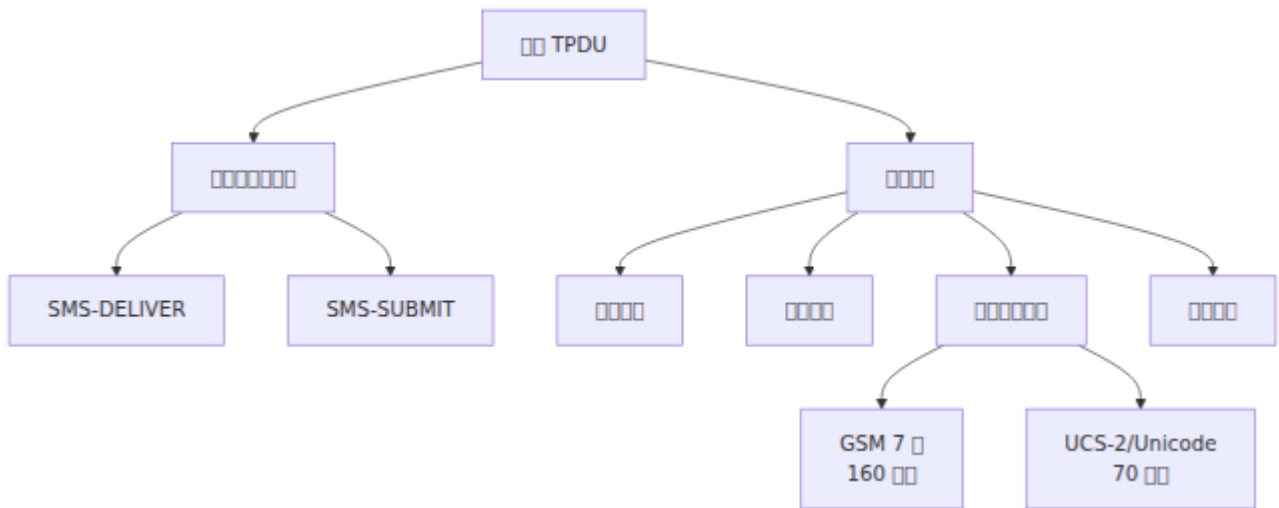
expires

SSMSc SS7 MO→MT

SSMSc SS7 MO SMS source_smsc "SS7_M0_15551234567"

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

SSMSc SS7 MO→MT MSC MSC IP-SM-GW



SS7

SS7 is a signaling protocol used in mobile networks.

1. **VLR** is a database that stores information about the current location of a mobile phone. It is used to route calls and messages to the phone.
2. **HSS API** is a protocol used to communicate between the HSS and the VLR. It is used to retrieve the IMSI of a mobile phone and to update the VLR with the current location of the phone.

The HSS API is used to retrieve the IMSI of a mobile phone and to update the VLR with the current location of the phone.

SS7

SS7 is a signaling protocol used in mobile networks.

- **source_smsc** is a parameter used to identify the source of the SMS. It is used to route the SMS to the correct destination.
- **source_smsc** is a parameter used to identify the source of the SMS. It is used to route the SMS to the correct destination.
- **source_smsc** is a parameter used to identify the source of the SMS. It is used to route the SMS to the correct destination.

- 送信元SS7情報 Web API/SMPP 送信 SS7 送信元

送信元 SMSC

送信元情報 source_smsc 送信元

送信元	送信元情報	送信元
SS7 送信元MO-FSM送信元	"SS7_M0_15551234567"	送信元 - 送信元
SS7 送信元MO-FSM送信元	"SS7_M0_15551234567"	送信元 - 送信元 MO→MT 送信元
送信元 API/SMPP	"ipsmgw" 送信元 "api_gateway"	送信元
送信元 SMSc	"smsc-node-01"	送信元

送信元

送信元情報送信元送信元送信元

```
{
  "message_id": 12345,
  "name": "送信元",
  "description": "送信元 - source_smsc 'SS7_M0_15551234567' 送信元 'SS7'
  送信元送信元送信元送信元"
}
```

送信元送信元送信元

- Web UI/SS7 送信元 (/events)
- 送信元 API 送信元 events 送信元
- 送信元送信元送信元

11

```

smc_mode_enabled:
true  VLR  hlr_api_base_url

```

--

1 SS7 MO-SMS

1. 0000 → MSC/VLR → IP-SM-GW0000 MO-ForwardSM0
2. IP-SM-GW 00 MO-FSM000000source_smsc = "SS7_MO_9991900"
3. 00000000000000
4. 0000 VLR 000 HSS 000 → 000
5. IP-SM-GW 000 "SS7" 0 source_smsc 0 + 000000 → 00
6. 00000"0000"
7. 000 SRI-for-SM 0 MT-ForwardSM00000000

2 SS7 MO-SMS

1. 终端 → MSC/VLR → IP-SM-GW → MO-ForwardSM
2. IP-SM-GW 向 MO-FSM 发送 source_smsc = "SS7_MO_9991900"
3. 终端收到消息
4. 终端 HSS API 接口 → 终端
5. IP-SM-GW 向 SS7
6. HSS 向 IMSI + VLR → MT-ForwardSM 向 VLR

SMSc ☐ ☐ ☐ ☐ ☐

SMSc GenServer alertServiceCenter

11

□□□□□□

- □□□□□□□□□□□□

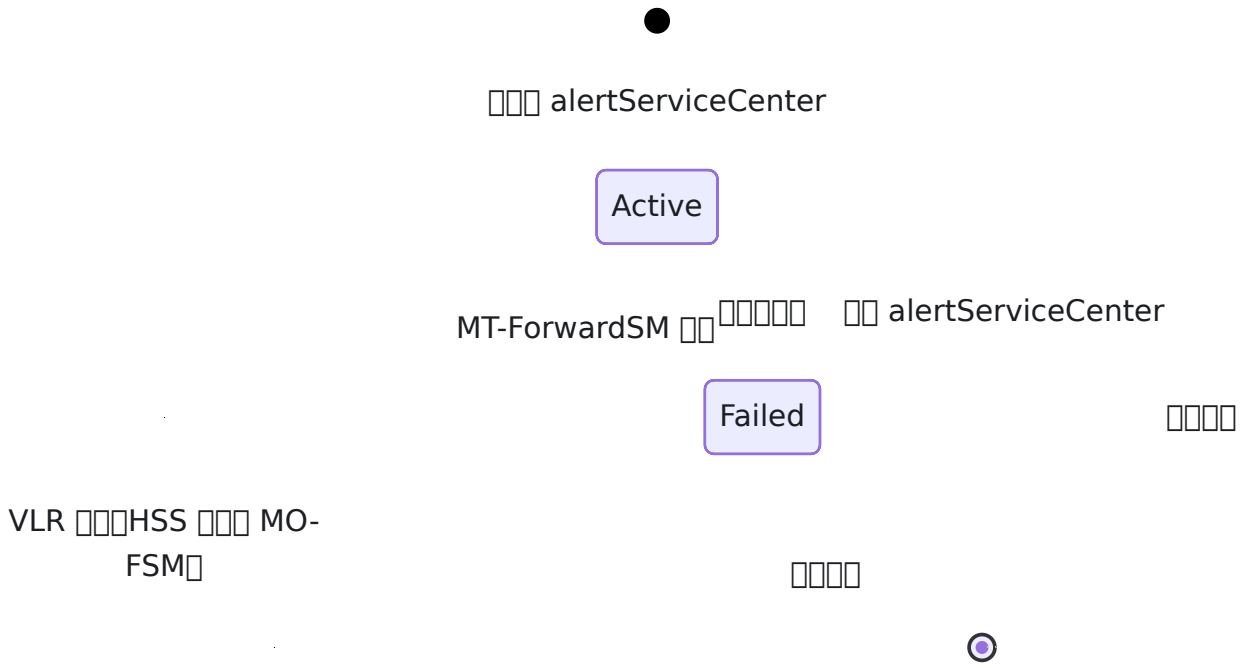
- **HLR** alertServiceCenter → HLR
- alertServiceCenter/AlertServiceCenter
- alertServiceCenter/AlertServiceCenter
- **Web UI** alertServiceCenter

□ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □

項目	説明	値
msisdn	電話番号	"15551234567"
imsi	IMSI HSS IMSI	"001010123456789"
hlr_gt	alertServiceCenter HLR GT	"15551111111"
messages_sent	MT-FSM	5
messages_received	MO-FSM	2
status	:active :failed	:active
updated_at	Unix	1730246400
vlr_address	VLR GT	"14155550100"
vlr_cached_at	VLR Unix	1730246400

--	--	--	--



11

alertServiceCenter

- 000000000000
- 00 status = :active
- 00 HLR GT
- 0 HSS API 000 IMSI0000000000000000 IMSI
- 0000 VLR 00000000

□ **HSS API** □□□ **VLR** □□□□□□

- 電話番号 VLR 番号 IMSI
- 電話番号 VLR 番号 HSS

MO-ForwardSM

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

□ **MT-ForwardSM** □□□□□□

- VLR 向 HSS API

□ SRI-for-SM □□□□

- `messages_sent`
- `updated_at`

SRI-for-SM

- `status = :failed`
- `XXXXXXXXXX`

□ □ □ □ □ □ □ □ □

- ETS 1000
- 1000 Web UI

Web UI - SMSc ☐☐☐☐☐

```
000/smsc_subscribers
```

□□□□□□ 2 □

SMSc 配置文件 `config/config.exs` 中 SMSc 配置项如下：

SMSc

11

1. 問題
 - MSISDN、IMSI、HLR GT、VLR GT、VLR 番号
 - 番号/番号
 - 番号/番号
 - 番号番号番号
 - 番号番号番号
2. 問題

- 查詢狀態
- 查詢記錄
- 查詢詳情
- 新增 HLR 記錄

3. 查詢

- 查詢所有記錄
- 查詢指定記錄

查詢

SMSc 查詢結果				頁數: 3
MSISDN	IMSI	HLR GT	VLR GT	Msg S/
15551234567	001010123456789	15551111111	14155550100	5/2
15559876543	001010987654321	15551111111	-	0/0
15551112222	001010111222233	15552222222	14155550200	3/1

查詢 頁數: 3 | 頁數: 2 | 頁數: 1 | 新增 HLR: 2

API 查詢

查詢所有記錄

```
# 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VLR 配置TTL 配置HSS API

配置VLR 配置

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

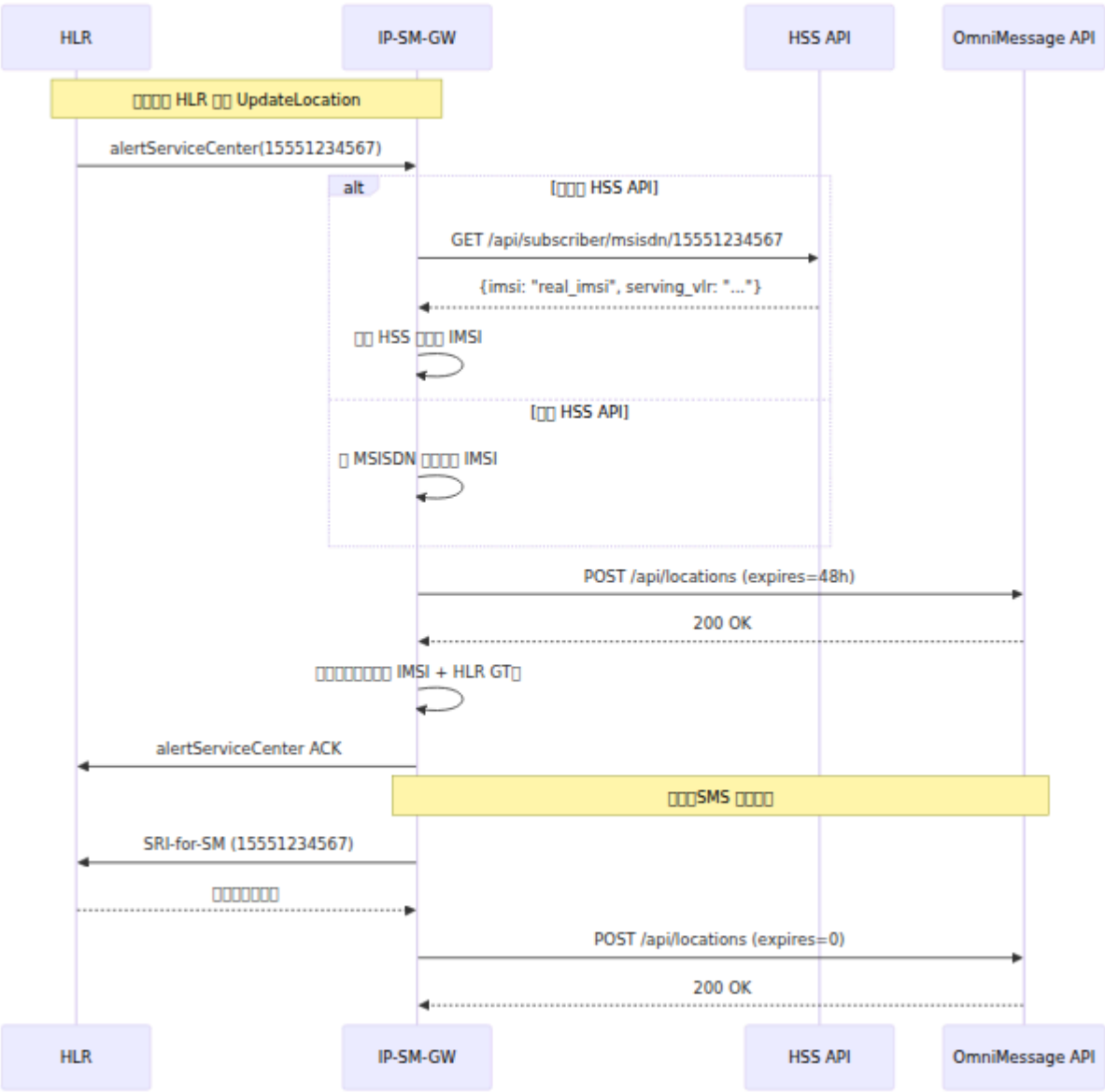
配置 alertServiceCenter 配置VLR 配置

配置

```
config :omniss7,  
  # HSS/HLR API 配置VLR 配置  
  hlr_api_base_url: "https://omnihss.example.com: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>
<code>vlr_cache_ttl_seconds</code>	配置	配置	<code>3600</code>

□□□



□□□□ SMS □□

□□□□ □□□□□□□□□□ SMS □□□

□□□□□□□□□□□□ □□□□□□□□□□□□

項目	単位	項目	SMS 項目
map_requests_total	個	operation	MAP 項目 SMS 項目 sri_for_sm forward_sm MTforward_sm prn
map_request_duration_milliseconds	ms	operation	MAP 項目/項目 項目 SRI-for-SM 項目 MT-ForwardSM

項目項目項目項目 SMS 項目項目 map_requests_total 項目 cap_requests_total 項目
ussd_requests_total 項目 ussd_active_sessions 項目 map_pending_requests 項目
omniss7_license_status 項目 map_request_duration_milliseconds 項目

項目項目

```
# 項目 5 項目 MT-ForwardSM 項目項目  
rate(map_requests_total{operation="MTforward_sm"}[5m])  
  
# 項目 5 項目 SRI-for-SM 項目項目  
rate(map_requests_total{operation="sri_for_sm"}[5m])  
  
# MT-ForwardSM 項目項目項目項目 5 項目  
rate(map_request_duration_milliseconds_sum{operation="MTforward_sm"}  
[5m]) /  
rate(map_request_duration_milliseconds_count{operation="MTforward_sm"  
[5m])
```


SMSc

SMSc

SMSc

1. SMSc
2. SMSc
3. SMSc
4. SM M3UA SMSc
5. SM TPS SM

SMSc

SMSc

- TPS SMSc
- HLR SMSc
- SMSc
- SMSc

SMSc

- SM `auto_flush_tps`
- SM HLR SMSc
- SMSc

MT-forwardSM API

API SMS

API `POST /api/MT-forwardSM`

snake_case destination_service_centre
smsPDU SMS TPDU

```
{
  "imsi": "234509876543210",
  "destination_service_centre": "447999555111",
  "originating_service_center": "447999123456",
  "smsPDU":
    "040B917477218345F600001570301857140C0BD4F29C0E9281C4E1F11A"
}
```

HTTP 200 result MT-ForwardSM TCAP
message_id "success"

```
{
  "result": "<formatted TCAP result>"
}
```

MAP HTTP 504 {"error": "timeout"}/ HTTP
400 {"error": "invalid request"}

OmniSS7

- ←
- HLR - HLR
 - SRI-for-SM - MSISDN IMSI
- Web UI/API
- MAP - MAP
-

```

OmniMessage 000 000000000000000000000000000000000000 OmniMessage 00
000OmniMessage 000000000000000000000000000000000000

```

OmniSS7 □ Omnitouch Network Services □□

M3UA & M2PA STP 目錄

← 目錄

本書介紹了 OmniSS7 的 STP 架構

目錄

1. STP
2. STP 架構
3. TDM 架構
4. STP 層
5. 協議
6. M2PA 協議
 - M3UA 與 M2PA 協議
 - M2PA 協議
 - SLTM 協議
 - Web UI 與 M2PA
 - M2PA 協議
7. 協議
8. 協議
9. SCCP 協議
 - GTT 與 SSN 協議
 - 協議
 - 協議
10. 協議 (協議)
11. 協議 (SCCP 協議)
12. 協議
 - 協議
 - DROP 協議 - 協議
13. 協議
14. 協議

15.

16. M3UA 消息

- M3UA □□□□□□
- □ IP □□□□□□

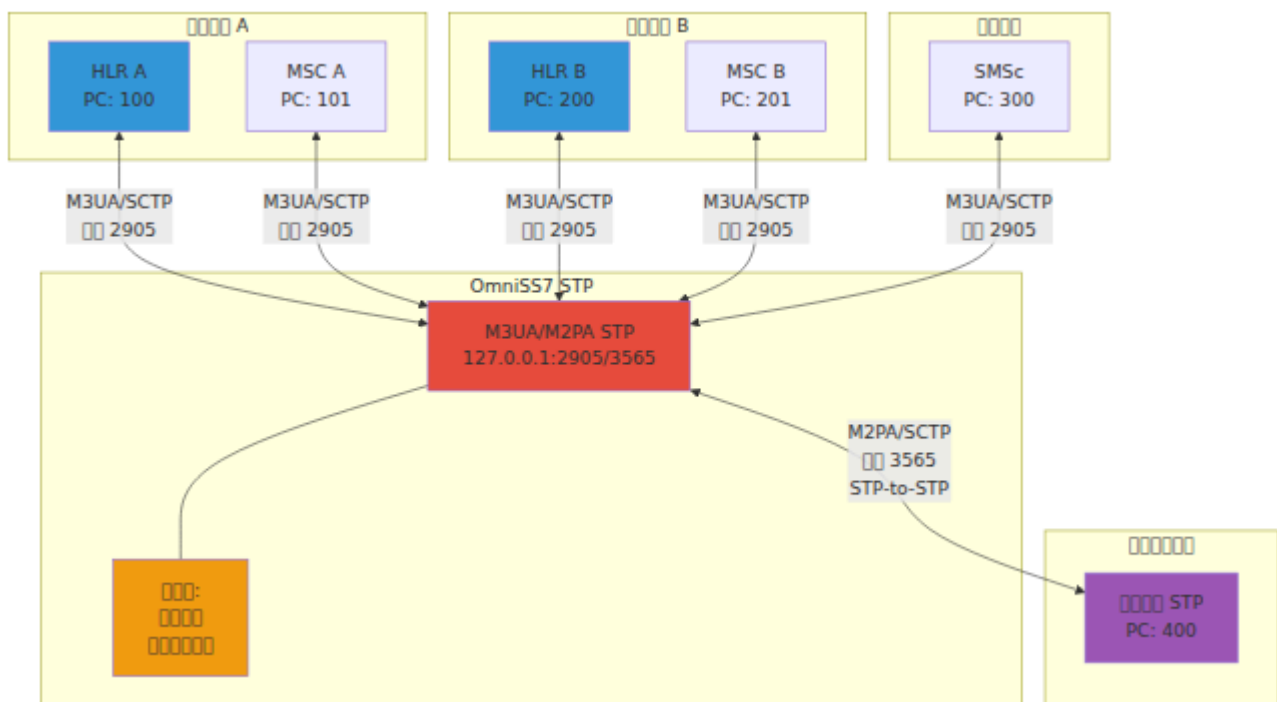
□ □ □ □ □ □ □ **(STP)**

STP SS7 IP

STP ☐ ☐

- 物理网络 (PC) 物理 (GT) 的 SS7 网络
- 物理网络 SS7 网络 IP 的 M3UA/SCTP 网络
- 物理网络网络网络网络网络网络网络
- 物理网络网络网络网络网络网络
- 物理网络网络网络网络网络网络

STP 概要



STP 機能概要

ASP (アプリケーションサーバ)

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

SGP (サービスゲートウェイ)

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

AS (アプリケーションサーバ)

- 各アプリケーション ASP 接続

- 0000000000000000
- 000000 ASP 0000000000

0000000000 TDM 000000

OmniSS7 000000 IP 0000000000 SIGTRAN 00 (M3UA/M2PA 00 SCTP)00000000 00 TDM
0 SS7 00 00000000000000 0000 (SGW) 000000000000

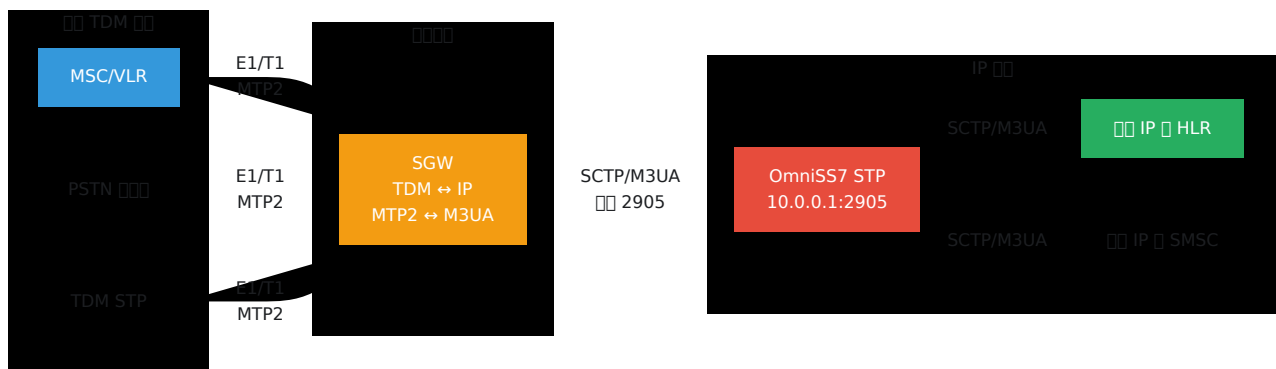
000000000000

0000 (SGW) 000000000000000000000000 SS7 0000

- TDM 0000 E1/T1 000000 SS7000 MTP1/MTP2/MTP3
- IP 0000 SCTP/IP 0 SIGTRAN0000 M3UA 0 M2PA

SGW 0000000000000000 IP 0000000000 OmniSS7000000000 IP 000000 TDM 0000000000STP0
HLR00000000

TDM 0 IP 00



□□□□□

□□	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

1. IP ネットワーク TDM

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

```

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

```

2. OmniSS7 の IP ネットワーク TDM ネットワーク 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

```

3 SGW

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

```
peers: [
    %{
        peer_id: 1,
        name: "SGW_Primary",
        role: :client,
        remote_ip: {10, 0, 0, 100},
        remote_port: 2905,
        point_code: 100,
        # ... 
    },
    %{
        peer_id: 2,
        name: "SGW_Backup",
        role: :client,
        remote_ip: {10, 0, 0, 101},
        remote_port: 2905,
        point_code: 100,
        # ... 
    }
],

m3ua_routes: [
    # SGW_Primary 
    %{dest_pc: 100, peer_id: 1, priority: 1, network_indicator:
:international},
    # SGW_Backup 
    %{dest_pc: 100, peer_id: 2, priority: 2, network_indicator:
:international}
]
```

□□ M3UA STP □□

OmniSS7 STP :omniss7 STP

環境構築

ファイル	説明
config/config.exe	Web UIの8445番ポートのAPIを管理する
config/stp_example_runtime.exe	STPのsctp_handlerを実行するPC側のGT
config/runtime.exe	STPのsctp_handlerを実行するサーバー側のGT

環境構築は config/runtime.exe を実行してサーバー側のSTPを起動し、 config/stp_example_runtime.exe を実行してPC側のGTを実行する。

STP の実行

STPを実行するには以下の手順を実行する。

- config/stp_example_runtime.exe を実行してSTPのSCTPを実行するPC側のGTを実行する。
- config :omniss7, ... を実行して config/runtime.exe を実行する。
- IP アドレスを指定して実行する。
- 実行コマンドは iex -S mix

STP 실행

`config/stp_example_runtime.exe` 실행 STP 실행을 위한 STP 실행 `sctp_handler` 실행
실행 GT 실행을 위한 PC 및 GT 실행

```

config :omniss7,
  # SCTP 配置 STP 配置 - 配置 SCTP 配置
  sctp_handler: %{
    enabled: true,
    local_ip: {127, 0, 0, 1},
    local_port: 2905,
    point_code: 101          # STP 配置
  },

  # 配置 PC 配置
  enable_gt_routing: true,

  # M3UA 配置 PC 配置 GT 配置
  m3ua_peers: [ %{peer_id: 1, name: "STP_West", role: :client,
...} ],
  m3ua_routes: [ %{dest_pc: 100, peer_id: 1, priority: 1,
network_indicator: :international} ],
  m3ua_gt_routes: [ %{gt_prefix: "44", peer_id: 2, priority: 1,
description: "配置"} ]

```

配置 M3UA 配置 配置 配置

配置

配置

配置 config :omniss7 配置 STP 配置

項目	型	初期値	説明	デフォルト値
sctp_handler	Map		STP の ハンドラ。 STP	<code>{enabled: true, local_port: 2905, point_code: 101}</code>
enable_gt_routing	Boolean	false	GT ルーティング を PC に 有効にする	true
m3ua_peers	List	[]	M3UA / M2PA の ピア。 M3UA	
m3ua_routes	List	[]	MTA の ルート。 M3UA	
m3ua_gt_routes	List	[]	GT の ルート。 M3UA	

local_ip / remote_ip は IP アドレス。 Sctp は Sctp のポート番号。

STP の設定

Web UI の

- SS7 -
- M3UA -
- ← STP
- ← STP
-
-

📋 功能特性

- 支持 SCTP 协议 (IP 地址 132) 的通信
- 支持 M3UA 协议 2905 的通信
- 支持配置和部署
- 支持通过 Web UI 和 API 进行配置和管理 **Mnesia** 数据库
- 支持 `runtime.exs` 配置文件管理 Mnesia 数据库
- 支持配置和部署
- **Web UI** 支持 Web UI 界面 支持 Web 管理
- **API** 支持 API 接口 REST API 支持 Swagger UI 文档

📄 STP 配置

支持配置 M3UA STP 配置 `config/stp_example_runtime.exs` 文件

📄 STP 配置

支持配置 `config :omniss7` 配置

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


STP 配置

属性	类型	默认值	说明	配置值
<code>enabled</code>	Boolean	<code>false</code>	是否启用 STP	<code>true</code>
<code>local_ip</code>	Tuple	<code>{127, 0, 0, 1}</code>	本地 IP 地址	<code>{0, 0, 0, 0}</code>
<code>local_port</code>	Integer	<code>2905</code>	本地端口	<code>2905</code>
<code>point_code</code>	Integer	<code>0</code>	STP 点代码 (SS7)	<code>100</code>

配置 STP

- 配置 M3UA 和 MAP 接口
- 配置 **STP** 接口
- 配置 STP 接口 HLR、MSC 和 SMSC

配置 STP 接口 `map_client_m3ua` 和 `sctp_handler`

配置 (Mnesia)

配置 Mnesia 数据库

配置

- Runtime.exs** 配置 `config/runtime.exs` 中的 `peers` 配置
配置 `m3ua_peers`、`m3ua_routes` 和 `m3ua_gt_routes`
- Web UI** 配置 `Web UI` 中的 Mnesia 配置
- 配置 `runtime.exs` 中的 Mnesia 配置
- 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 `M3UARouting.init_database` 配置

配置	配置
<code>m3ua_peer</code>	M3UA / M2PA 配置
<code>m3ua_route</code>	配置 (DPC → 配置)
<code>m3ua_gt_route</code>	配置 (配置 → 配置 SSN/TT/NPI/NAI 配置)
<code>m3ua_dest_state</code>	配置 MTP3 配置 (TFP/TFA/TFR, Q.704 §13)
<code>ss7_screen_rule</code>	配置 (SCCP 配置) 配置/配置

□□□□

□□□□□□□□□□

- 1. **Runtime.exe** - □□□□□□□□
- 2. **Web UI** - □□□□□□□□ **Web UI** □□
- 3. **REST API** - □□□□□□□□ **API** □□

□□□□□□ `runtime.exe` □□□□□□□□ Web UI □□□□□□□□□□□□

□□ **M3UA** □□□

□□□□ M3UA □□□□□□ STP□HLR□MSC□SMSC□□□□□□□□ `config/runtime.exe`□

□□□□□□□

| □□□□□□ `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
}
]

```

参数列表

参数	数据类型	是否必填	说明
peer_id	字符串	否	对端标识符
name	字符串	否	对端名称
role	枚举	否	:client 客户端 :server 服务器
local_ip	IP 地址	否	本地 IP 地址 {10, 0, 0, 1} 表示 Sctp 地址 [{10, 0, 0, 1}, {10, 0, 0, 2}]
local_port	端口	否	本地端口 0 表示任意
remote_ip	IP 地址	否	对端 IP 地址 - 对端 IP 地址
remote_port	端口	否	对端端口 0 表示任意 = 对端端口
routing_context	字符串	否	M3UA 路由上下文
point_code	字符串	否	对端 SS7 点码
network_indicator	枚举	否	:international 国际 :national 国内

对端 Sctp 地址由对端 IP 地址、对端端口、对端点码、对端网络标识符组成。Sctp 地址由本地 IP 地址、本地端口、本地点码、本地网络标识符组成。Sctp 地址由对端 IP 地址、对端端口、对端点码、对端网络标识符组成。

对端 Sctp 地址由对端 IP 地址、对端端口、对端点码、对端网络标识符组成。local_ip 本地 IP 地址、remote_ip 对端 IP 地址、Sctp 地址由对端 IP 地址、对端端口、对端点码、对端网络标识符组成。

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
 - 作为服务器
 - 监听 2905 端口
- 配置 `remote_port: 0` 的节点
 - 作为客户端/对等节点
 - 使用 IP 地址
 - 连接到服务器

示例

```
# 配置 192.0.2.10:2905 的节点
%{
  peer_id: 1,
  name: "Strict_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 2905,
  # ... 其他配置
}

# 配置 192.0.2.10 的节点
%{
  peer_id: 2,
  name: "Flexible_Peer",
  role: :server,
  remote_ip: {10, 5, 198, 200},
  remote_port: 0, # 连接到服务器
  # ... 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

- 需要SGW
- 需要/
- RFC 4666

M3UA HLR MSC SMSC VLR STP

M2PA

M2PA

- STP STP
- TDM M2PA SS7 TDM
- MTP2 MTP2
- M2PA

M2PA HLR MSC SMSC STP - M3UA M2PA STP

M2PA

M2PA M3UA protocol

M2PA

M2PA config/runtime.exs peers M3UA M2PA protocol

M2PA

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 3565
remote_port		この SCTP 接続の M2PA ポート番号 3565
point_code		この接続の点番号
adjacent_point_code		この接続の M2PA 点番号
send_slrm		この SLTM 接続の SLTM 番号
network_indicator		:international 国際ネットワーク :national 国内ネットワーク

この M2PA 接続 3565 は M3UA 2905 接続と接続

SLTM 接続

SLTM 接続は SLTA 接続と MTP3 接続の M2PA 接続 READY 状態
ITU-T Q.707 接続 SLTM 接続 SLTA 接続と接続

SLTM 接続

send_slrm 接続の SLTM 番号

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

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

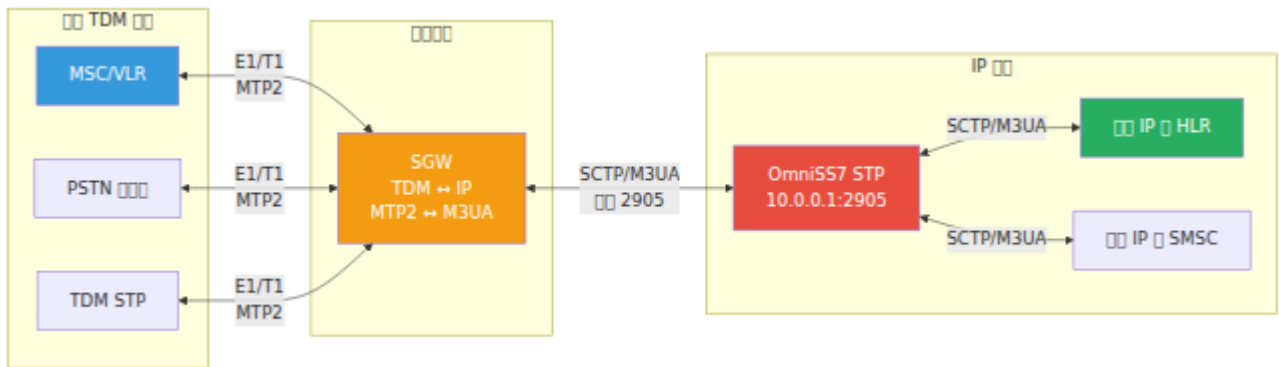
- [] `initiate_connection: true` [] SCTP [] [] [] → [] [] [] [] [] SLTM
- [] `initiate_connection: false` [] SCTP [] [] [] → [] [] SLTM

[] [] **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_sltn: true # [] [] [] [] SCTP [] [] [] [] SLTM
}
```

SLTM [] []



SLTM

READY

SLTM

"waiting for peer to send SLTM"

`send_slrm: 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:9090/metrics`
- 配置

M2PA 部署

1. 安装 3565 版本的 M2PA
2. 配置
3. 配置 SCTP IP 为 132
4. 配置

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

M2PA 클래스

M2PA 클래스는 **SCTP.SocketHandler** 인터페이스를 구현합니다. `M2PA` 클래스는 `SCTP.SocketHandler` 인터페이스를 구현하는 `SCTP` 클래스를 상속받습니다.

예제

- `SCTP.SocketHandler` 인터페이스를 구현합니다.
- `M2PA` 클래스를 `SCTP` 클래스에 등록합니다.

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 RFC 4165 M2PA

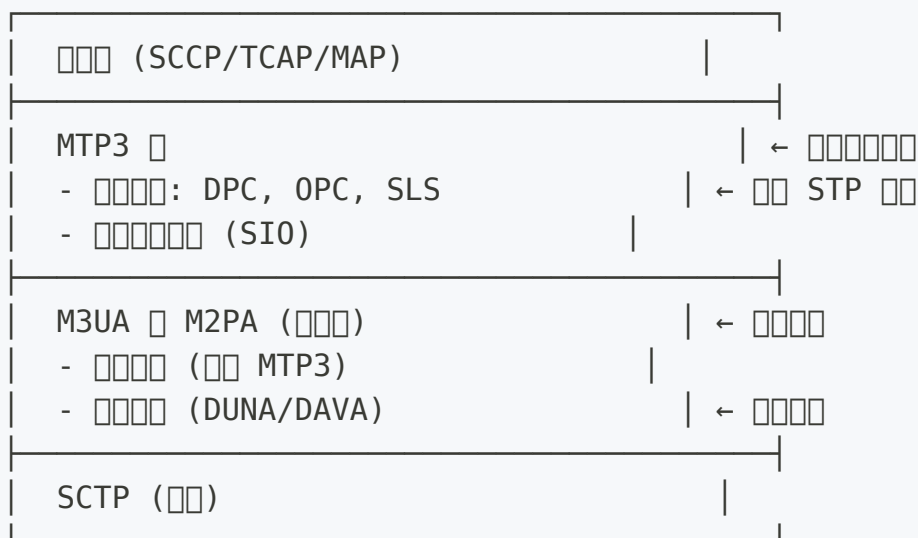
SS7

SS7 MTP3 DPC

SS7

SS7

SS7



SS7

1. MTP3

- SS7 MTP3 DPC OPC
- SS7 M3UA 528
- SS7 M2PA
- SS7 DPC
- SS7

2. M3UA

- SS7 M3UA DUNA DAVA SCON DUPU
- SS7

- 000000000000/000000000000
- 0000000000

STP 00000000

- 00 **M3UA DATA** 000STP 0000000000 5280000 MTP3 00000000 MTP3 0000
0DPC0OPC0SLS0000 MTP3 00 DPC 000000
- 00 **M2PA** 00000000STP 0 M2PA 0000000000 MTP3 00000000 MTP3 00000000
DPC0
- **M3UA** 000000000000DUNA0DAVA0SCON00 M3UA 000000000000000000000000
0000

00000000

00000000 config/runtime.exs0

```
config :omniss7,
  m3ua_routes: [
    # 100000000 PC 100 1000 1000 STP
    %{
      dest_pc: 100, # 1000
      peer_id: 1, # 10000
      priority: 1, # 1000000000000000
      network_indicator: :international
      # mask: 14 # 1000000 140000000
    },

    # 100000000 PC 200 1000 2000 HLR
    %{
      dest_pc: 200,
      peer_id: 2,
      priority: 1,
      network_indicator: :international
    },

    # 100000000 PC 300 1000000000
    %{
      dest_pc: 300,
      peer_id: 3, # 1000
      priority: 1,
      network_indicator: :international
    },
    %{
      dest_pc: 300,
      peer_id: 4, # 1000000000000000
      priority: 2,
      network_indicator: :international
    }
  ]
]
```

mask 14 mask

□ □ □ □

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

- 4. 00000000 DPC000000
- 5. 00000000000000 0000000000000000 00000000 000
- 6. 0 MTP3 00000 M3UA DATA 0 M2PA 00000000000000
- 7. 000000000000
- 8. 00000000000000000000000000000000

0000

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

0000

0000000000 PC 000000 DPC 00000000 00000000000000000000000000000000

000000

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

階

階 mask 階 階 14 階

階 1 階

```
# 0000000000000000 PC
%{
    dest_pc: 1000,
    peer_id: 1,
    priority: 1,
    network_indicator: :international
}
# 00000 14 - 0000 PC 1000

# 0000000000000000
%{
    dest_pc: 1000,
    peer_id: 1,
    priority: 1,
    mask: 14, # 00000000
    network_indicator: :international
}
# 00000 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: [
    # 1000番 PC 1000番
    %{
      dest_pc: 1000,
      peer_id: 1,
      priority: 1,
      network_indicator: :international
      # mask 14番
    },

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

    # 0番/1000番 PC
    %{
      dest_pc: 0,
      peer_id: 3,
      priority: 10, # 10番
      mask: 0, # 0番 PC
      network_indicator: :international
    }
  ]

```

DPC 1000

- mask /14 1000番 PC 1000番 - 1000番
- mask /8 1000番 PC 1000-1063番 - 1000番
- mask /0 0番 PC - 1000番

DPC 1015

- mask /14 1000番 PC 1000番
- mask /8 1000番 PC 1000-1063番 - 1000番
- mask /0 0番 PC - 1000番

DPC 5000

- 1. mask /14
- 2. mask /8
- 3. mask /0 PC -

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

GT

IMSI / NAT

□□□□

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

GT 配置

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

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

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

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

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

GT □□□□

GT □□□□□□□□□□□□□□

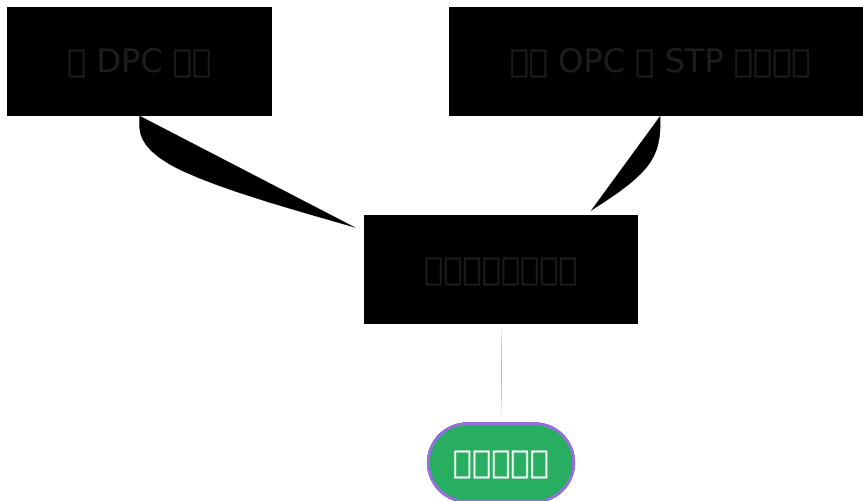
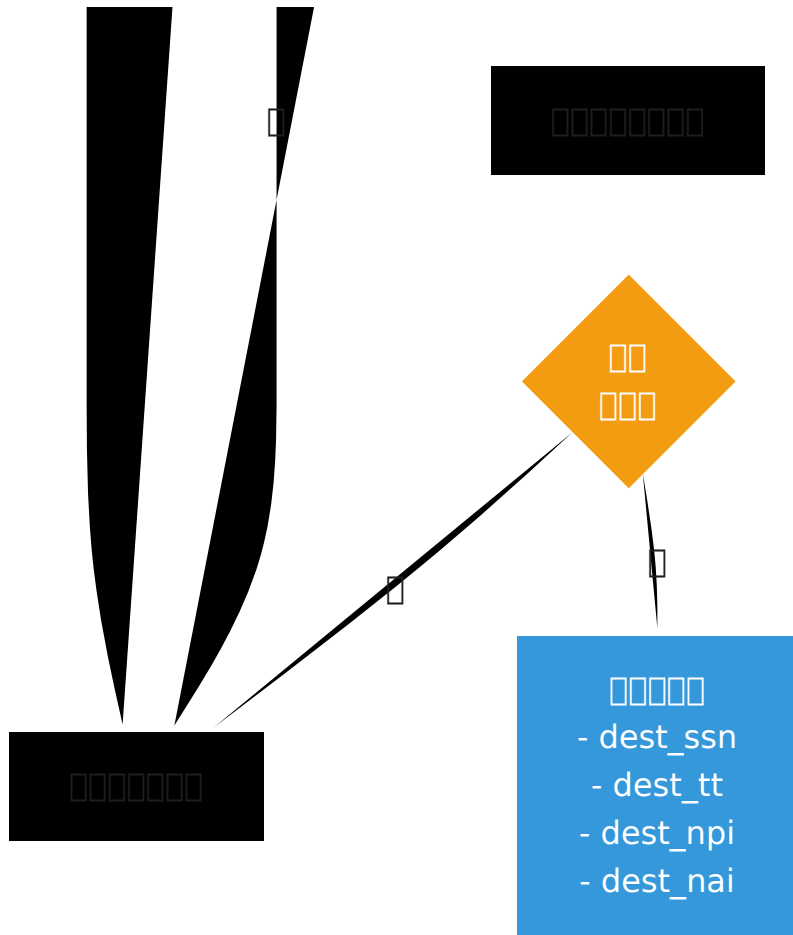
SCCP

GT, SSN, TT, NPI,
NAI

GT

GT + SSN + TT + NPI
+ NAI

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



GT

1. GT STP GT GT

◦ GT "447712345678" "44" "447" "447" GT

2. SSN GT

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

3. TT/NPI/NAI 00000000

- 00000 source_tt 0 source_npi 0 source_nai 0000000000000000
- nil 0000000000000000

4. 0000000000

- 00000000000000000000
- 0000000GT 0000 → SSN → TT → NPI → NAI → 000000

5. 0000000000

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

- 0000000000000000

6. 00000000

- 0000 GT 00000STP 000000 DPC 000000

00 GT 0000000000NPI 0 NAI

00 GT 000 SSN 00000STP 00000 SCCP 0000000000000000

- 00000**TT**0000000000000000
- 00000000**NPI**0000000000000000ISDN00000000
- 00000000**NAI**0000000000000000000000000000

0000000000

000000000000000000000000

- `source_tt` 0000000000000000
- `source_npi` 0000000000000000000000
- `source_nai` 0000000000000000000000
- `nil` 0 = 000000000000

0000000000

0000000000000000

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

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=1 ISDN/
    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=1 ISDN
    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

000000TT00

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

00000000NPI00

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

00000000NAI00

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

00000000

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	規則	結果
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	1	4	"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

- 規則
 - 規則
 - 規則
 - 規則 TT=0 規則 TT=1
- 規則
 - 規則

- 國際電訊號碼 NPI=6 國際 PSTN NPI=1

3. 呼叫建立

- 國際電訊號碼 NAI
- 國際電訊號碼 NAI=4 國際電訊號碼 NAI=3 國際電訊號碼

4. 呼叫建立

- 國際電訊號碼
- 國際 TT=0 國際 A TT=2 國際 B

5. 呼叫建立

- 國際電訊號碼
- STP 國際電訊號碼

SCCP 呼叫

國際電訊號碼 STP 國際 SCCP 國際 國際電訊號碼 國際電訊號碼 國際 STP 國際電訊號碼
 國際 UDT 0x09 國際 XUDT 0x11 國際 LUDT 0x13 國際

國際電訊號碼 SSN 國際

國際電訊號碼 STP 國際電訊號碼 國際 GT 國際 GT 國際電訊號碼 國際 GT 國際電訊號碼
 GT 國際電訊號碼 GT 國際 ITU-T Q.714 §2.4 國際電訊號碼 國際 GT 國際電訊號碼
 國際電訊號碼 國際 SSN 國際電訊號碼 國際 GT 國際 SSN 國際電訊號碼
 國際 國際電訊號碼 GTT 國際 MSS 國際 `providerAbort(abnormalDialogue)`

國際 / STP 國際 國際電訊號碼 國際 GTT 國際 SSN 國際電訊號碼 國際 SSN 國際電訊號碼
 GT 國際 `route_on_ssn: true` 國際電訊號碼

```

m3ua_gt_routes: [
  # 44 - 44 GT to 44 GTT
  %{gt_prefix: "44", peer_id: 2, priority: 1, description: "44
  "},

  # 1234 SSN 1234 - 1234 SSN
  %{
    gt_prefix: "1234",
    peer_id: 1,
    priority: 1,
    dest_ssn: 8,
    route_on_ssn: true,
    description: "1234 SSN 1 MSC"
  }
]

```

The API endpoint `route_on_ssn` is `POST /api/routing/gt-routes` with `false`
 indicating that the GT is not a SSN (ITU-T Q.714 §2.4).

XUDT / LUDT

XUDT and LUDT are **SCCP** protocols used for STP.

1. Q.714 §2.4
2. Q.714 §2.8.6
3.


```

m3ua_stp_routing_failures_total{reason="hcounter_violation"}
      
```
4. XUDTS/LUDTS

UDT

UDTs / XUDTS / LUDTS

UDTs are used for STP and SCCP.
 ITU-T Q.714 §4.2

UDT	UDTS
UDT (0x09)	UDTS (0x0A)
XUDT (0x11)	XUDTS (0x12)
LUDT (0x13)	LUDTS (0x14)

UDT, XUDT, LUDT, UDTS, XUDTS, LUDTS MTP 0x09, 0x0A, 0x11, 0x12, 0x13, 0x14

SCCP
UDT/XUDT/LUDT

XUDT/LUDT?
UDT

UDT

GT DPC

UDT

UDT

UDT

UDT

UDT
UDT

UDTS/XUDTS/LUDTS
UDTS

+

UDTS/XUDTS/LUDTS

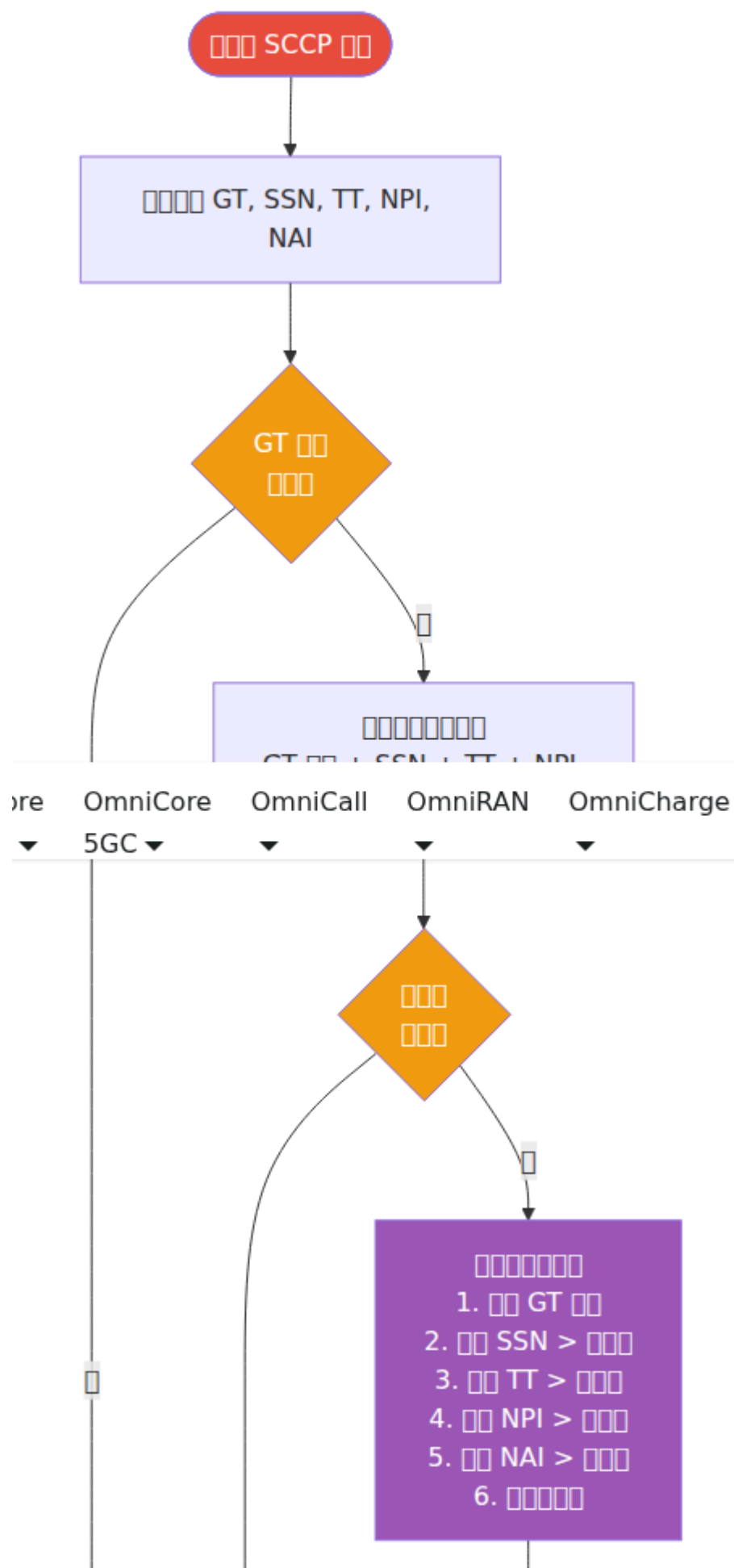
STP MTP3 ITU-T Q.704 §13.2.3
m3ua_dest_state Mnesia

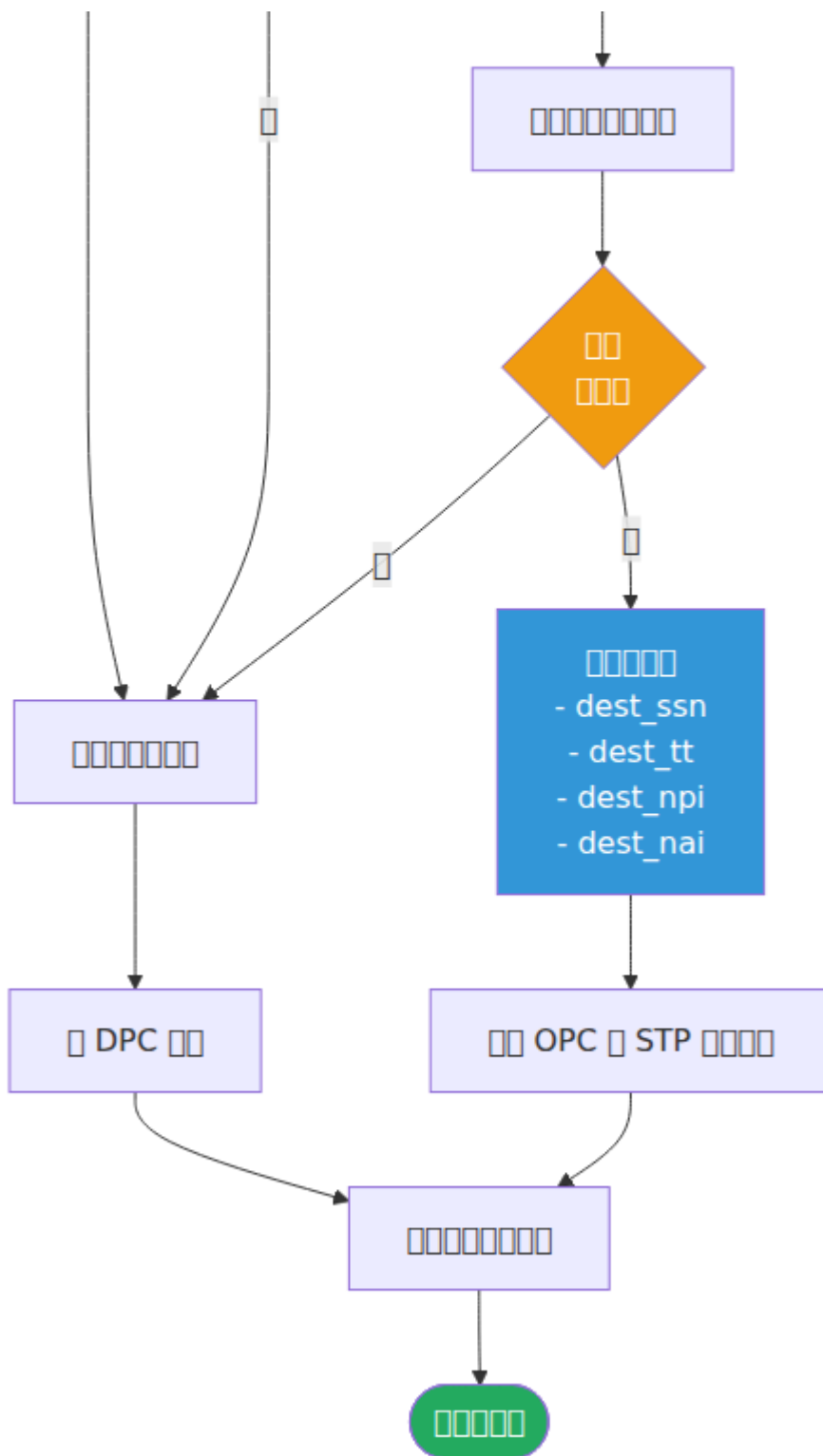
TFP / TFA / TFR

		STP
TFP		:prohibited
TFA		
TFR		:restricted

UDTS/XUDTS/LUDTS

- STP {error, :prohibited}Q.704 §13.2.3
m3ua_stp_routing_failures_total{reason="prohibited"}
- DAUD“”blanketDAVA
DUNA RFC 4666 §3.4.3





SCCP

STP STP SS7

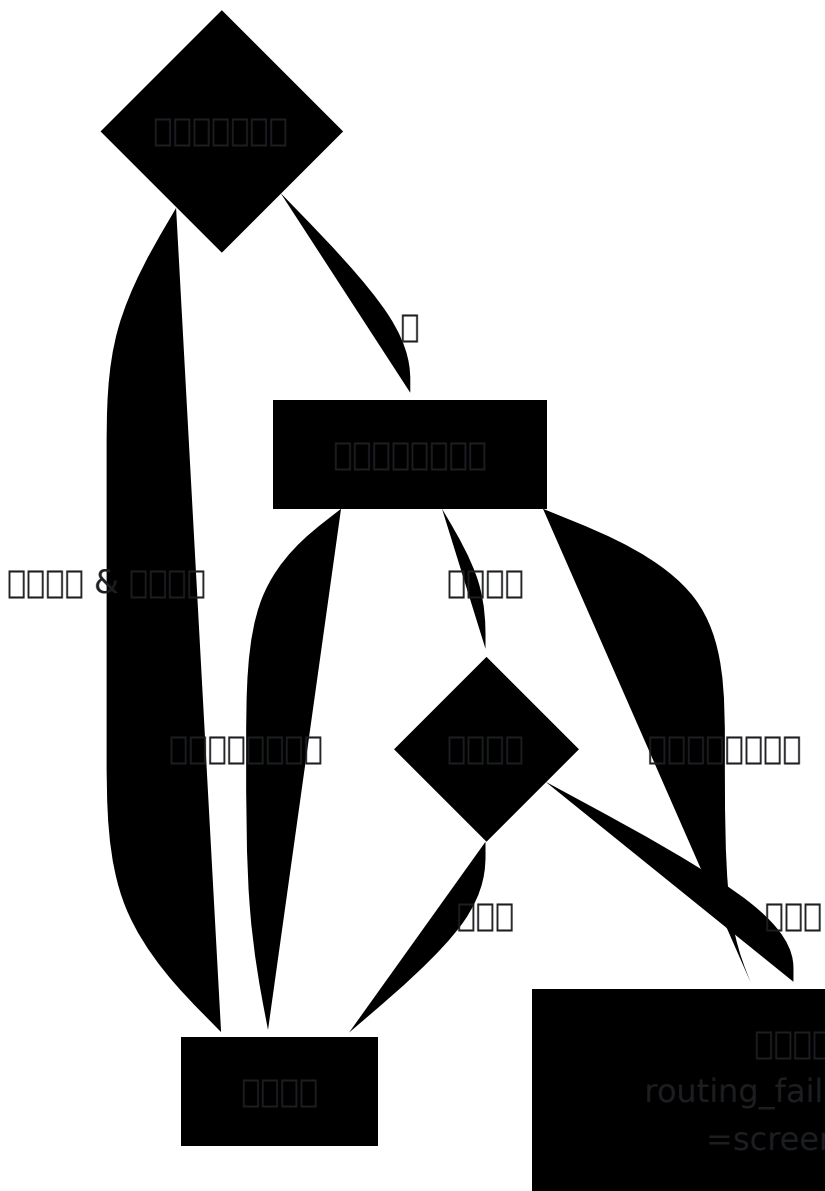
M3UA

項目	項目
項目	項目
OPC	項目
DPC	項目
項目	MTP3 SI
項目 GTA 項目	SCCP 項目
項目 GTA 項目	SCCP 項目
SSN	SCCP 項目

項目 allow 項目 deny 項目 config
:omniss7, :screening_default_action項目 :allow項目

項目 m3ua_stp_routing_failures_total{reason="screened"}項目
項目 UDTS/項目

11



☐ :allow ☐ :deny ☐

設定

項目	値	説明	備考
screening_default_action	:allow :deny	:allow	デフォルトで拒否する場合は :deny を指定する デフォルト

```
config :omniss7,  
  # 拒否する  
  screening_default_action: :deny
```

ss7_screen_rule Mnesia データベースに保存される
allow/deny フラグは enabled フラグが有効な場合にのみ適用される

インストール

前提条件

インストールには以下の環境が必要

OS

インストールには enabled フラグが必要


```
config :omniss7,  
  m3ua_routes: [  
    # []  
    %{  
      dest_pc: 100,  
      peer_id: 1,  
      priority: 1,  
      network_indicator: :international,  
      enabled: true # []  
    },  
  
    # []  
    %{  
      dest_pc: 200,  
      peer_id: 2,  
      priority: 1,  
      network_indicator: :international,  
      enabled: false # []  
    }  
  ],  
  
  m3ua_gt_routes: [  
    # [] GT []  
    %{  
      gt_prefix: "44",  
      peer_id: 1,  
      priority: 1,  
      description: "[] - []",  
      enabled: false  
    }  
  ]
```

□ □ □ □ □

- `enabled` `enabled: true`
-
- Web UI `/`

111

- □□□□□□□

- □□□□□□
- A/B □□□□□□□□
- □□□□□□□□

DROP

DROP [peer_id: 0]

DROP

```
config :omniss7,
  m3ua_routes: [
    # 00000000 DROP 00
    %{
      dest_pc: 999,
      peer_id: 0,          # peer_id=0 000 DROP
      priority: 1,
      network_indicator: :international
    }
  ],

  m3ua_gt_routes: [
    # 00 GT 000 DROP 00
    %{
      gt_prefix: "999",
      peer_id: 0,          # peer_id=0 000 DROP
      priority: 99,
      description: "0000000"
    }
  ]
]
```

DROP □□□□□□

□□□□ DROP □□□□

1. `peer_id: 0`
2. `peer_id: 0`

3. 如果 **INFO** 消息 "DROP route matched for DPC 999" 与 "DROP route matched for GT 999"
4. 消息格式 { :error, :dropped }

消息格式 消息格式 INFO 消息格式

消息格式

DROP 消息格式 消息格式 - 消息格式

消息

1. 消息格式 消息 **DROP** 消息格式 99
2. 消息格式 消息格式 消息格式 1
3. 消息格式 消息格式 消息格式 DROP 消息格式
4. 消息格式 消息格式 DROP 消息格式

消息格式

消息格式 GT 消息格式 1234 消息格式 10,000 消息格式 1234000000 - 1234999999 消息格式 3 消息格式 1234567890 消息格式 1234555000 消息格式 1234111222

```

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

    # 1234567890 1
    %{
      gt_prefix: "1234567890",
      peer_id: 1,          # 1234567890 1
      priority: 1,         # 1234 = 1234567890 = 1234567890
      description: "1234567890 1"
    },

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

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

```

1234567890

GT	Route	Drop Route	Count
1234567890	Route "1234567890" 1 Route "1234" DROP 99	"1234567890" Route	Count 1
1234555000	Route "1234555000" 1 Route "1234" DROP 99	"1234555000" Route	Count 1
1234111222	Route "1234111222" 1 Route "1234" DROP 99	"1234111222" Route	Count 1
1234999999	Route "1234" DROP 99	"1234" DROP	Count + Count
1234000000	Route "1234" DROP 99	"1234" DROP	Count + Count

Count

- Count 3 Count 1
- Count 1234* Count
- Count

Count

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

Count **DROP** Count

Count

```

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

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

```

PC 1010 1020 1030 1000-1063 PC

DROP

```

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

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

```

Web UI

- INFO
- "DROP route matched"

1. ⚠️ 0000000000 DROP 0000000000
 2. 0 00000 description 000000000000
 3. 0 000000000090-9900 DROP 0000000000000000
 4. 0 0000000000 DROP 0000
 5. 0 000000000000000000
-

00000000 SSN 00000000

00000000SSN0

000000000000

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

00 SSN 000000

00000000 SMS 000000000000 HLR0

```

m3ua_gt_routes: [
  # 000000 SMS 000000 HLR00 SSN 80SMSC0000 60HLR0
  %{
    gt_prefix: "44",
    source_ssn: 8,                # 0000 SSN 80SMSC0
    peer_id: 1,
    dest_ssn: 6,                 # 000 SSN 60HLR0
    priority: 1,
    description: "00 SMS 0 HLR"
  },

  # 0000000000000000SSN 6000000000
  %{
    gt_prefix: "44",
    source_ssn: 6,                # 0000 SSN 60HLR0
    peer_id: 1,
    dest_ssn: nil,               # 0 SSN 00
    priority: 1,
    description: "00000000"
  }
]

```

00 STP 0000

000000000000000000000000

1. 00000000

00 Web UI

- 000 <http://localhost>
- 00 M3UA 0000
- 00000000 000**ACTIVE**

00 IEx 0000


```
# 取得ピアのステータス
SCTP.SocketHandler.get_peers_status()

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

2. 送信テスト

```
# DPC 100 のピア M3UA 送信
test_payload = <<1, 2, 3, 4>> # テストペイロード
SCTP.SocketHandler.route_by_pc(100, test_payload, 0)

# 送信結果
# "Routing message: OPC=... -> DPC=100 via peer 1"
```

3. 配置路由

```
# 配置 GT 路由
M3UARouting.lookup_peer_by_gt("447712345678")

# 配置 {ok, peer, rewrites} 列表 peer 列表 13 个 m3ua_peer 列表
# `rewrites` 列表 MAP nil = 列表
# {ok, {m3ua_peer, 3, "UK_Mobile_Peer", ...},
#      %{dest_ssn: nil, dest_tt: nil, dest_npi: nil, dest_nai:
#      nil}}
```

```
# 配置 SSN 与 GT 列表 SSN 8 → 6 列表
M3UARouting.lookup_peer_by_gt("555881234567", 8)

# SSN 列表
# {ok, {m3ua_peer, 4, "SMS_HLR_Peer", ...},
#      %{dest_ssn: 6, dest_tt: nil, dest_npi: nil, dest_nai:
#      nil}}
```

4. 配置监控

在 `/metrics` 中 Prometheus 监控

配置

```
# 配置 m3ua_stp_messages_received_total
m3ua_stp_messages_received_total{peer_name="Partner_STP_West",point_code=1523}

# 配置 m3ua_stp_messages_sent_total
m3ua_stp_messages_sent_total{peer_name="Local_HLR",point_code="200"}

# 配置 m3ua_stp_routing_failures_total
m3ua_stp_routing_failures_total{reason="no_route"} 5
m3ua_stp_routing_failures_total{reason="no_gt_route"} 2
m3ua_stp_routing_failures_total{reason="no_active_peer"} 1
m3ua_stp_routing_failures_total{reason="prohibited"} 3
m3ua_stp_routing_failures_total{reason="hcounter_violation"} 4
m3ua_stp_routing_failures_total{reason="screened"} 7
```

STP 配置

配置

配置

- `m3ua_stp_messages_received_total` - 接收到的消息数量
 - `peer_name` `point_code`
- `m3ua_stp_messages_sent_total` - 发送的消息数量
 - `peer_name` `point_code`

配置

- `m3ua_stp_routing_failures_total` - 路由失败的数量
 - `reason`

reason	说明
no_route	没有路由
no_gt_route	没有大于路由 PC 的路由
no_active_peer	没有活跃的邻居/对等体
prohibited	被禁止 TFP Q.704 §13
hcounter_violation	SCCP 计数器违规 XUDT/LUDT Q.714 §2.8.6

配置

- 配置
- 配置
 - `no_route` 没有路由
 - `no_gt_route` 没有大于路由 PC 的路由
 - `no_active_peer` 没有活跃的邻居/对等体
 - `prohibited` 被禁止

- `hcounter_violation` 00000000000000000000000000000000

000000000000

000000000000

1. 000000000000

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

2. 000000000000

```
m3ua_stp_messages_sent_total{peer_name="Dest_Peer"} > 0
```

3. 00000000

```
m3ua_stp_routing_failures_total{reason="no_route"} > 0
```

00000000000000000000000000000000

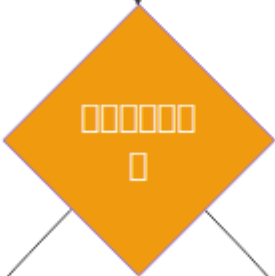
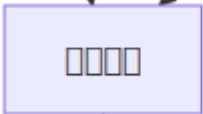
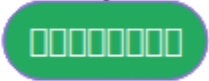
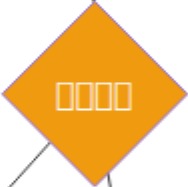
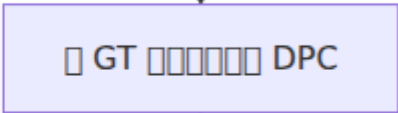
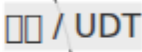
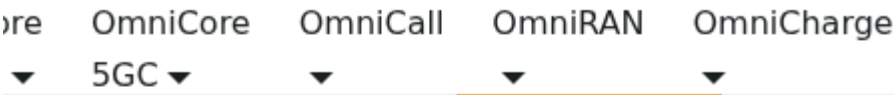
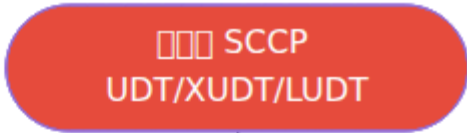
M3UA 00000000

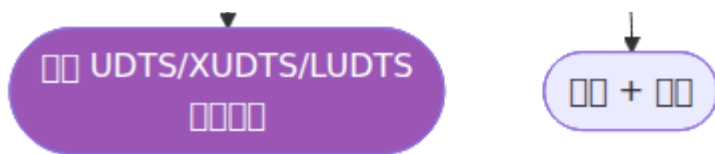
00 M3UA

M3UA 0MTP3 0000000000000000 SCTP 0 IP 00000 SS7 000

M3UA 0000

M3UA 0000000000





UDTS

- **DOWN** - SCTP UDTS
- **CONNECTING** - SCTP UDTS
- **ASPUP_SENT** - ASPUP UDTS
- **INACTIVE** - ASP UDTS ASPAC UDTS ACTIVE
- **ASPAC_SENT** - ASPAC UDTS
- **ACTIVE** - UDTS
- **ASPDOWN_SENT** - UDTS

Web UI M3UA

Web UI M3UA

M3UA

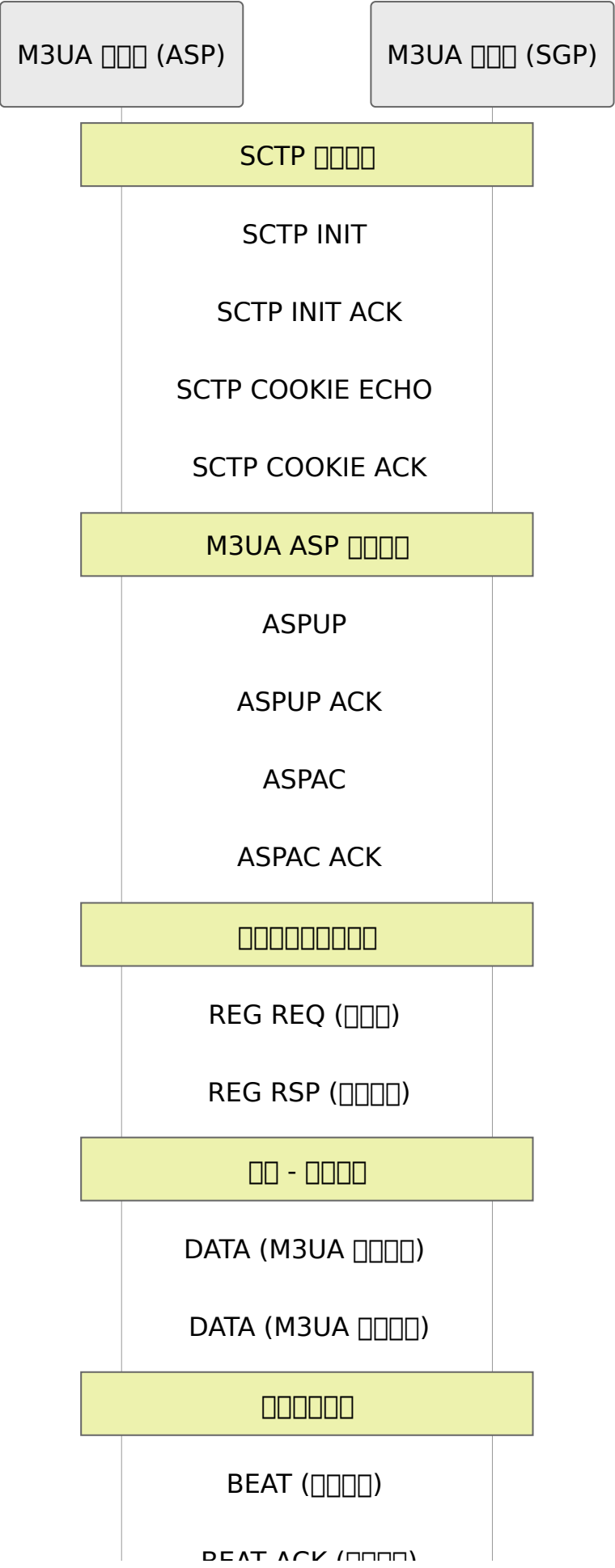
1. Web UI
2. "M3UA"
- 3.

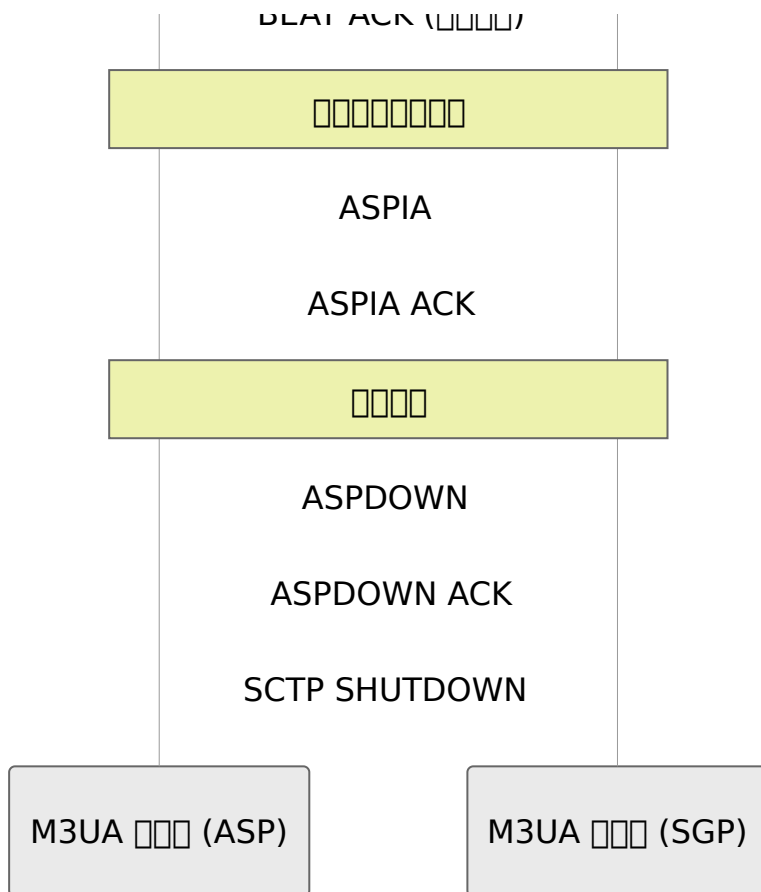
M3UA

項目	説明
名前	testASP
PID	
状態	UP DOWN
ASP 状態	M3UA 状態 ACTIVE INACTIVE
Assoc/SCTP	SCTP 状態
IP	IP:
IP	IP:
RC	ID

- **UP** -
- **DOWN** -
- **ASP 状態** - M3UA 状態
- **Assoc/SCTP** - SCTP 状態

M3UA □□□





RFC 4666 M3UA 0000

OmniSS7 000 RFC 4666 000000 M3UA 000000

번호	구분	코드	상태	설명
0 (MGMT)	0	ERR	실패	오류 발생
0 (MGMT)	1	NTFY	실패	통지
1 (XFER)	1	DATA	실패	데이터 전송 PC 및 GT 실패
2 (SSNM)	1	DUNA	실패	데이터 전송 실패
2 (SSNM)	2	DAVA	실패	데이터 전송 실패
2 (SSNM)	3	DAUD	실패	데이터 전송 실패 PC 및 DAVA 전송/수신 PC 및 DUNA
2 (SSNM)	4	SCON	실패	데이터 전송 실패
2 (SSNM)	5	DUPU	실패	데이터 전송 실패
2 (SSNM)	6	DRST	실패	데이터 전송 실패
3 (ASPSM)	1	ASPUP	실패	ASPUP/ACK ASPUP ACK
3 (ASPSM)	2	ASPDN	실패	ASPDN ASPDN ACK 전송 실패
3 (ASPSM)	3	BEAT	실패	BEAT BEAT ACK 전송 실패
3 (ASPSM)	4	ASPUP ACK	실패	ASPUP ACK UP
3 (ASPSM)	5	ASPDN ACK	실패	ASPDN ACK DOWN

번호	순 번	메시지	종류	비고
3 (ASPSM)	6	BEAT ACK	응답	응답
4 (ASPTM)	1	ASPAC	요청	응답/응답 ASPAC ACK
4 (ASPTM)	2	ASPIA	요청	응답 ASPIA ACK
4 (ASPTM)	3	ASPAC ACK	응답	응답 ACTIVE
4 (ASPTM)	4	ASPIA ACK	응답	응답 INACTIVE
9 (RKM)	1	REG REQ	SGP 요청	응답 REG RSP
9 (RKM)	2	REG RSP	ASP 응답	응답
9 (RKM)	3	DEREG REQ	SGP 요청	응답 DEREG RSP
9 (RKM)	4	DEREG RSP	ASP 응답	응답

BEAT/BEAT ACK은 STP에서 BEAT을 전송하면 BEAT ACK을 전송하는 프로토콜이다.
SCTP 프로토콜이다.

RKM은 ASP에서 REG REQ를 SGP에서 RK를 전송하면 DPC에서
REG RSP를 전송하는 프로토콜이다.

ASP에서 **ASPDN**은 SGP에서 ASPDN을 전송하면 ASPDN ACK을 전송하는
프로토콜이다.

ASP ☐ **ASPIA** ☐ SGP ☐ ASPIA ☐ ASPIA ACK ☐ INACTIVE ☐

M3UA □□□□□□

000000**RC** 0 M3UA 0000 32 00000000000000000000AS00000000000000000000
 00000000000000000000

□□□□□□□□

RFC 4666 3.3.1 子网掩码

- 0000000000000000AS
- M3UA DATA 0000000000 6
- 0000000000 AS 000000
- 0000000000000000

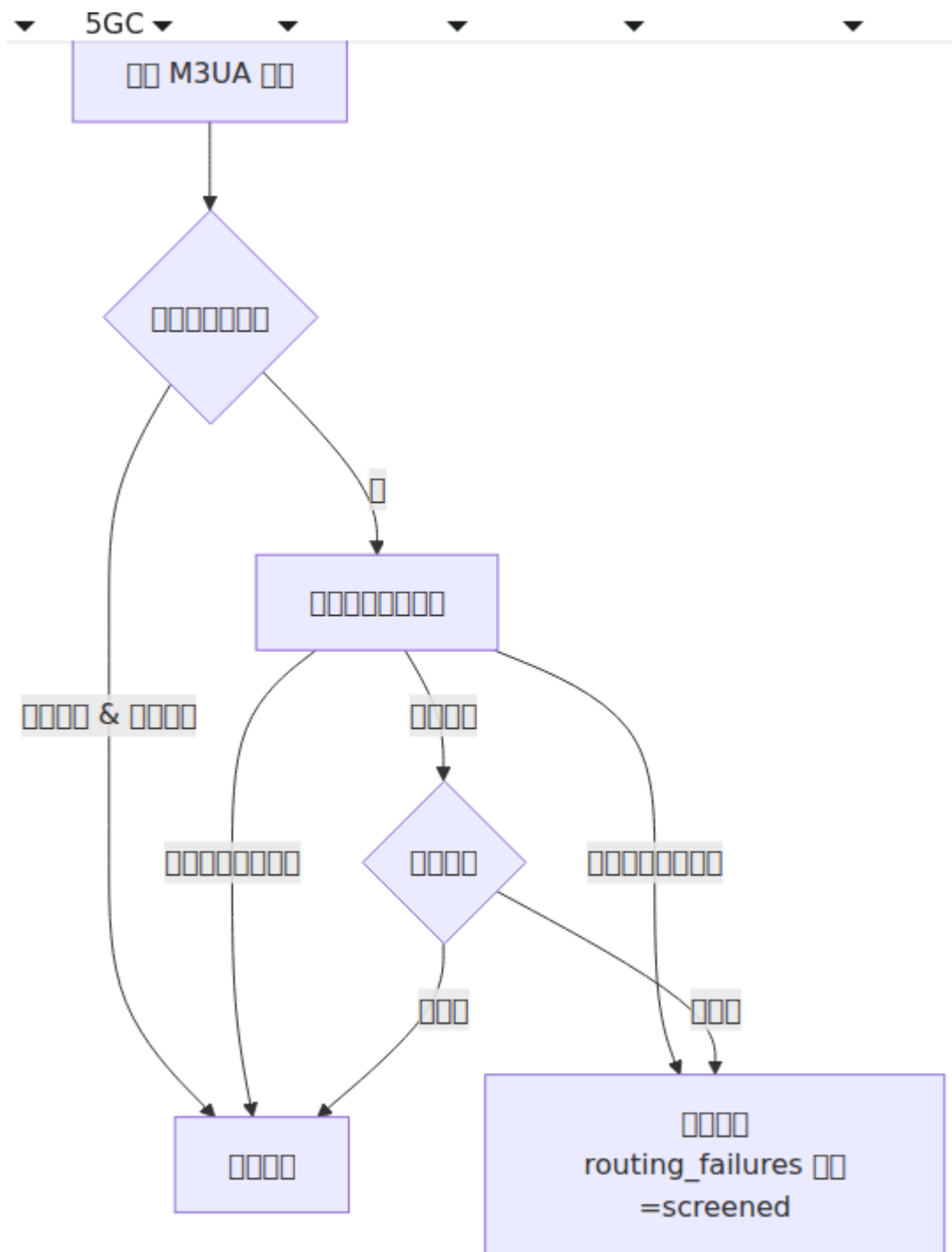
M3UA □□□□□□□□



0000000000 60000 M3UA DATA 0000000000000000 52800000000000 MTP3/SCCP 0000

OmniSS7 □□□□□□□□

OmniSS7



MAP □□□□ **CAP** □□□

MAP CAP

項目	RC 項目	説明
初期設定	初期設定 RC	初期 - 初期
初期設定 (SRI, PRN, IDP, ...)	初期設定 RC	map_client_m3ua cap_client_m3ua routing_context
初期設定 RC	初期設定 RC	初期設定
初期設定 RC	初期設定 RC	初期設定

初期設定

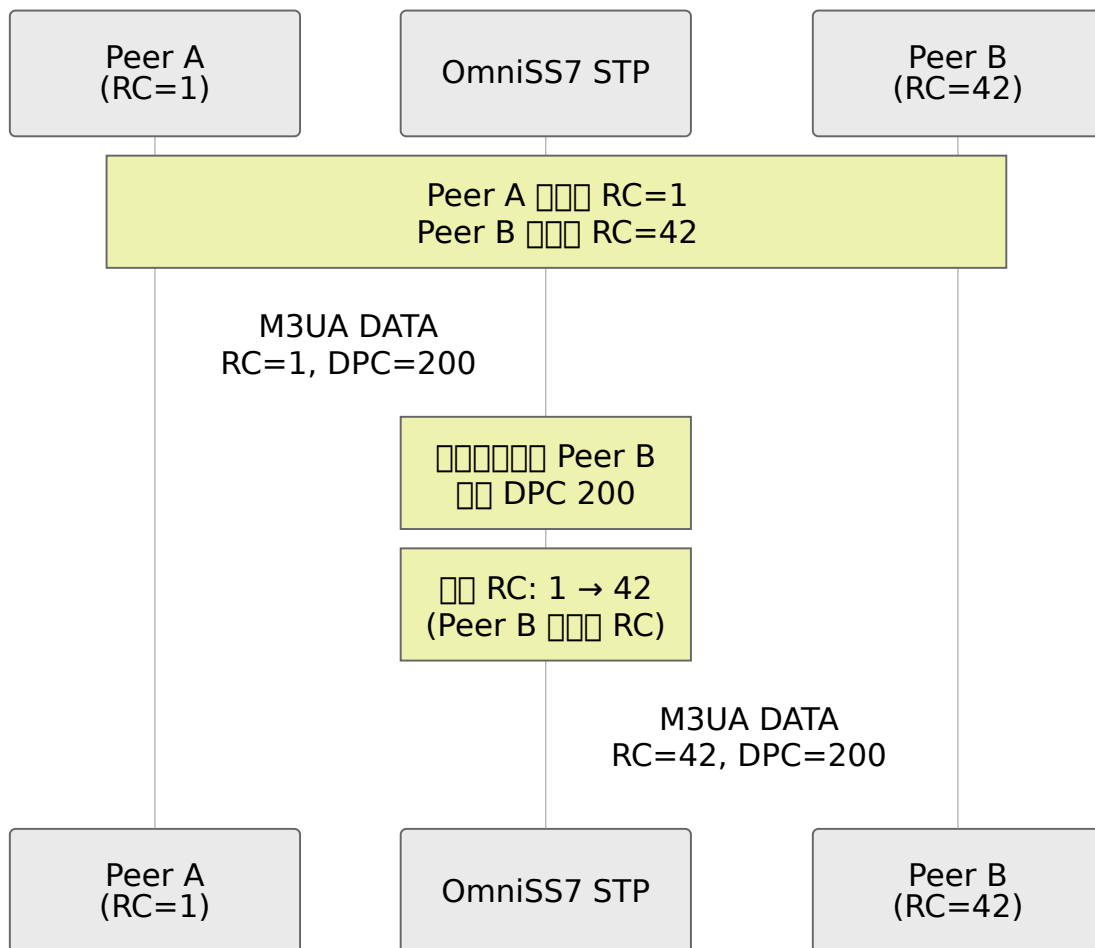
```
config :omniss7,  
  # MAP 初期 RC 初期  
  map_client_m3ua: %{\br/>    mode: "ASP",  
    local_ip: {10, 0, 0, 1},  
    local_port: 2905,  
    remote_ip: {10, 0, 0, 10},  
    remote_port: 2905,  
    routing_context: 1    # 初期設定 MAP 初期 (SRI, PRN, ...)  
  },  
  
  # CAP 初期 RC 初期  
  cap_client_m3ua: %{\br/>    mode: "ASP",  
    local_ip: {10, 0, 0, 1},  
    local_port: 2906,  
    remote_ip: {10, 0, 0, 20},  
    remote_port: 2906,  
    routing_context: 2    # 初期設定 CAP 初期 (InitialDP, ...)  
  }
```

STP 初期設定

STP RC (Routing Control) は RFC 4666 に基づいて定義されています。

1. RC (Routing Control) AS (Autonomous System) として定義
2. 各 AS は RC (Routing Control) を持つ
3. STP (Shortest Path Tree) AS (Autonomous System) RC (Routing Control) として定義

STP RC (Routing Control) の構成



STP

Routing Control (RC) の構成


```

config :omniss7,
  m3ua_peers: [
    %{
      peer_id: 1,
      name: "Network_A_STP",
      role: :server,
      remote_ip: {10, 0, 0, 100},
      remote_port: 2905,
      routing_context: 1,          # 10000000 RC
      point_code: 100,
      network_indicator: :international
    },
    %{
      peer_id: 2,
      name: "Network_B_HLR",
      role: :client,
      local_ip: {10, 0, 0, 1},
      local_port: 0,
      remote_ip: {10, 0, 0, 200},
      remote_port: 2905,
      routing_context: 42,        # 10000000 RC
      point_code: 200,
      network_indicator: :international
    }
  ]

```

10000000 1 RC=1000000000000000 2 100 STP

1. 100 DPC 1000000200 → 1000 20
2. 1 RC 1 1000 4200000 2 1000 RC
3. 10000000000 2

RC 10000

RC	MAP	CAP	STP
MAP RC	MAP RC	MAP RC	MAP RC
CAP RC	CAP RC	CAP RC	CAP RC
STP	STP RC	N/A	STP RC

RC

routing_context nil

- MAP/CAP RC
- MAP/CAP RC
- STP RC RC

IP

IP MSC Sctp remote_port: 0 IP

OmniSS7 M3UA ASPAC RC routing_context RC

- 检查 SCTP 配置

检查

1. 检查配置 `ping remote_ip`
2. 检查配置 SCTP 端口 132
3. 检查 STP/SGP 配置
4. 检查配置 `remote_ip` 和 `remote_port`
5. 检查配置 SCTP 配置

检查配置 **ASP** 配置

检查

- SCTP 配置
- ASP 配置 INACTIVE 和 ASPUP_SENT

检查

1. 检查配置
2. 检查 STP 配置
3. 检查配置 ASPUP/ASPAC 配置
4. 检查配置/配置

检查   检查

检查

- ASP 配置 ACTIVE
- 检查配置

检查

1. 检查配置
2. 检查 SCCP 配置 GT 配置 SSN 配置
3. 检查配置
4. 检查 `/events` 配置 SCCP 配置
5. 检查 STP 配置

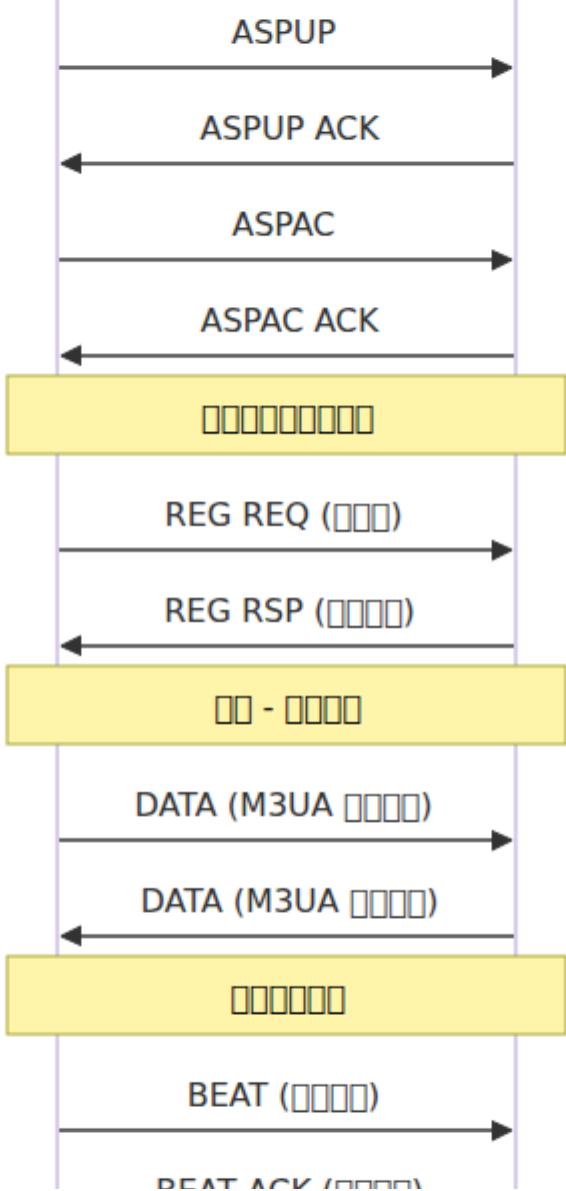
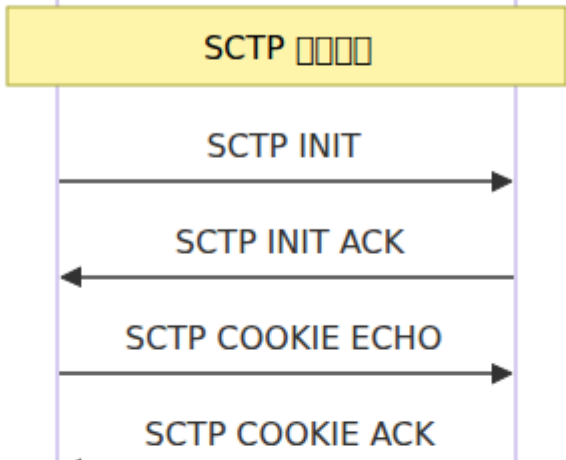
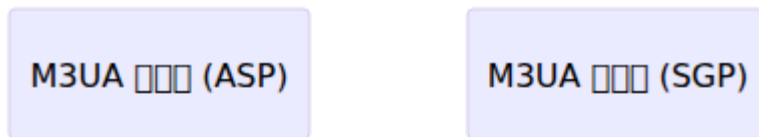
M2PA 概要

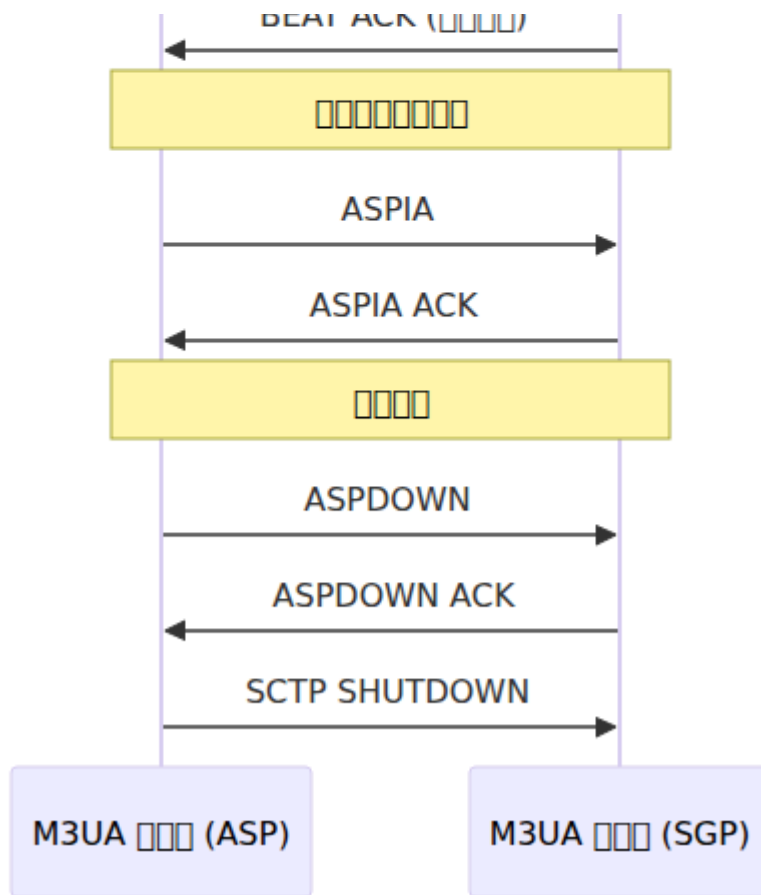
概要 M2PA

M2PAはMTP2 仕様を拡張した RFC 4165 仕様を SCTP 上で MTP3 として
ASP/SGP 間で M3UA として M2PA として TDM SS7 接続する

M2PA 機能

M2PA 機能





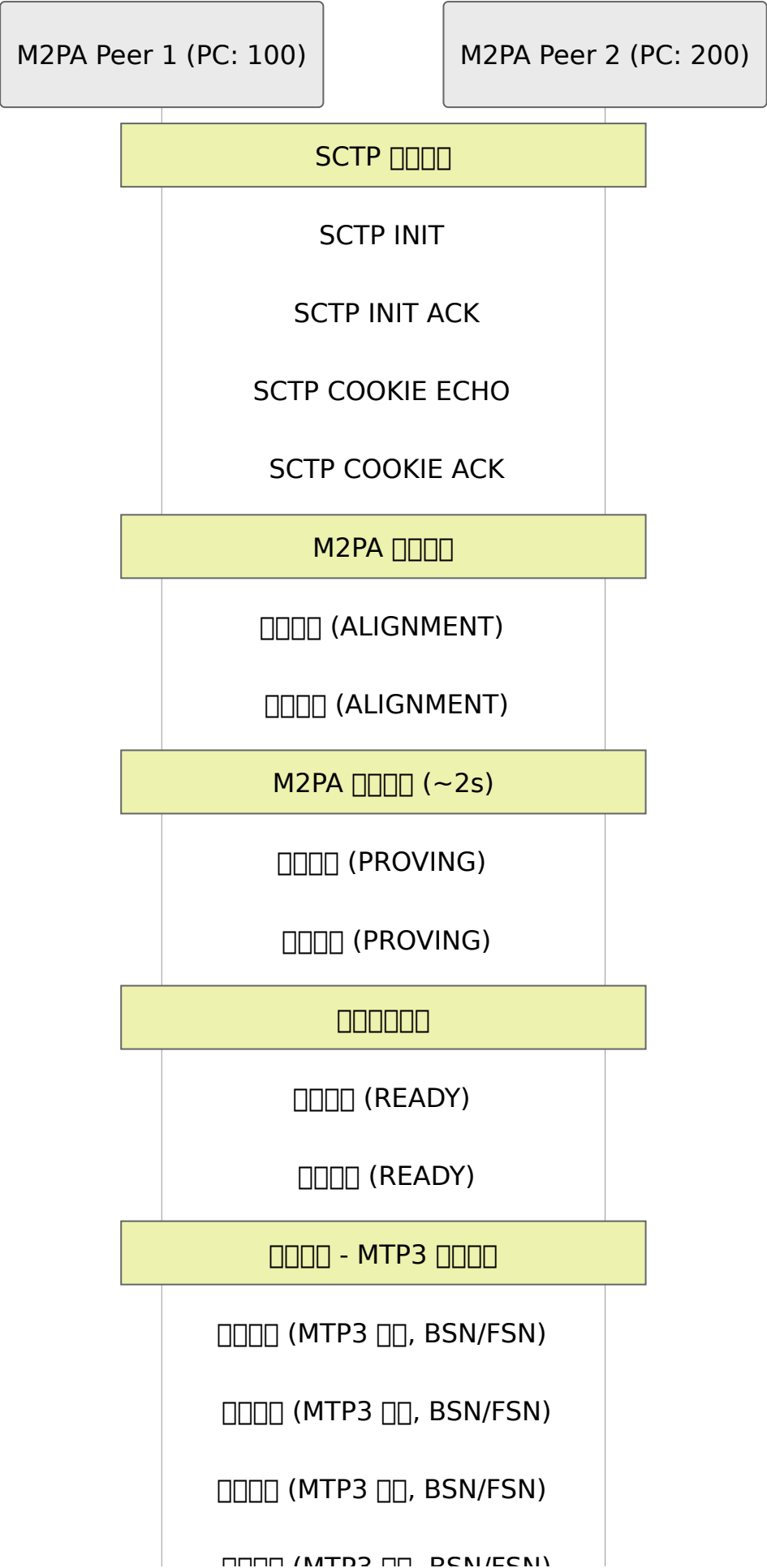
Sequence diagram

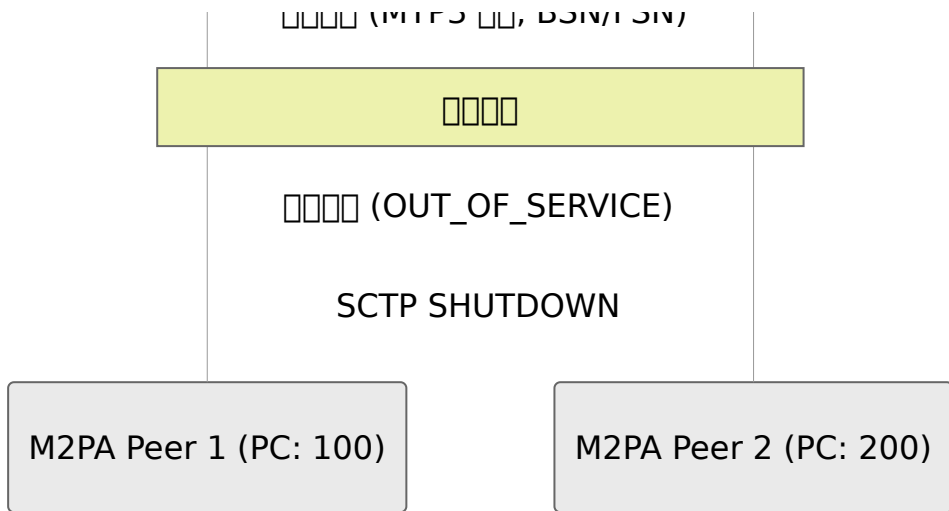
- **DOWN** - Sctp SHUTDOWN
- **CONNECTING** - Sctp SHUTDOWN
- **ALIGNMENT** - ASPIA (~1)
- **PROVING** - ASPIA ACK (~2)
- **READY** - ASPIA MTP3 SHUTDOWN
- **ALIGNMENT** - ASPDOWN - ASPDOWN ACK

Sequence diagram

1. **Sctp** SHUTDOWN Sctp SHUTDOWN → CONNECTING
2. ASPIA ASPIA → ALIGNMENT
3. ASPIA ASPIA ACK → PROVING
4. ASPIA ASPIA MTP3 SHUTDOWN → READY

M2PA □□□





Web UI M2PA

Web UI M2PA

1. Web UI
2. "Routing Management"
3. "M3UA/M2PA Peers"

M2PA

項目	説明
デバイス ID	一意の識別子
リンク	接続先 (例: M2PA_Link_STP_A)
リンク名	リンク名 "M2PA" を含む
ステータス	接続状態
デバイス PC	接続先 PC
ポート	ポート IP: 192.168.1.3565
ポート	ポート IP: 192.168.1.1
モード	接続モード: READY, ALIGNMENT, DOWN

接続モード

- **READY**モード - 接続準備完了
- **ALIGNMENT**モード - 接続調整中
- **PROVING**モード - 接続確認中
- **DOWN**モード - 接続切断

接続モード M2PA について

接続モード ALIGNMENT

接続

- 接続モード ALIGNMENT 中に接続エラーが発生した場合
- 接続モード PROVING から READY へ移行

接続

1. 接続モード ALIGNMENT 中に接続エラーが発生した場合

2. SCTP 132
3. point_code adjacent_point_code
- 4.
5. ALIGNMENT

PROVING

- PROVING READY
- 2-3

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

DOWN ↔ READY

- READY DOWN
-

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

- 00000000 READY
- 00 MTP3 000000

000

1. 00000000000000000000
2. 00 MTP3 0000000000
3. 000000 DPC 0000000000
4. 00 /events 0000000000
5. 000000BSN/FSN000000

00000

- ← 000000
- 000000 - Web UI, API, 00
- MAP 000000 - 00 MAP 00
- 000000 - 0000
- 0000 - 0000

OmniSS7 0 Omnitouch 000000

←

11

1. 二维码
2. 二维码
3. 二维码 USSD 二维码
4. 二维码
5. HTTP 二维码
6. 二维码 USSD (二维码 API)
7. 二维码
8. 二维码
9. 二维码
10. 二维码
11. 二维码

11

USSD USSD USSD

- 100# MAP processUnstructuredSS-Request
59 HTTP SS7
- REST API USSD SS7 MAP
unstructuredSS-Request 60

□□□□

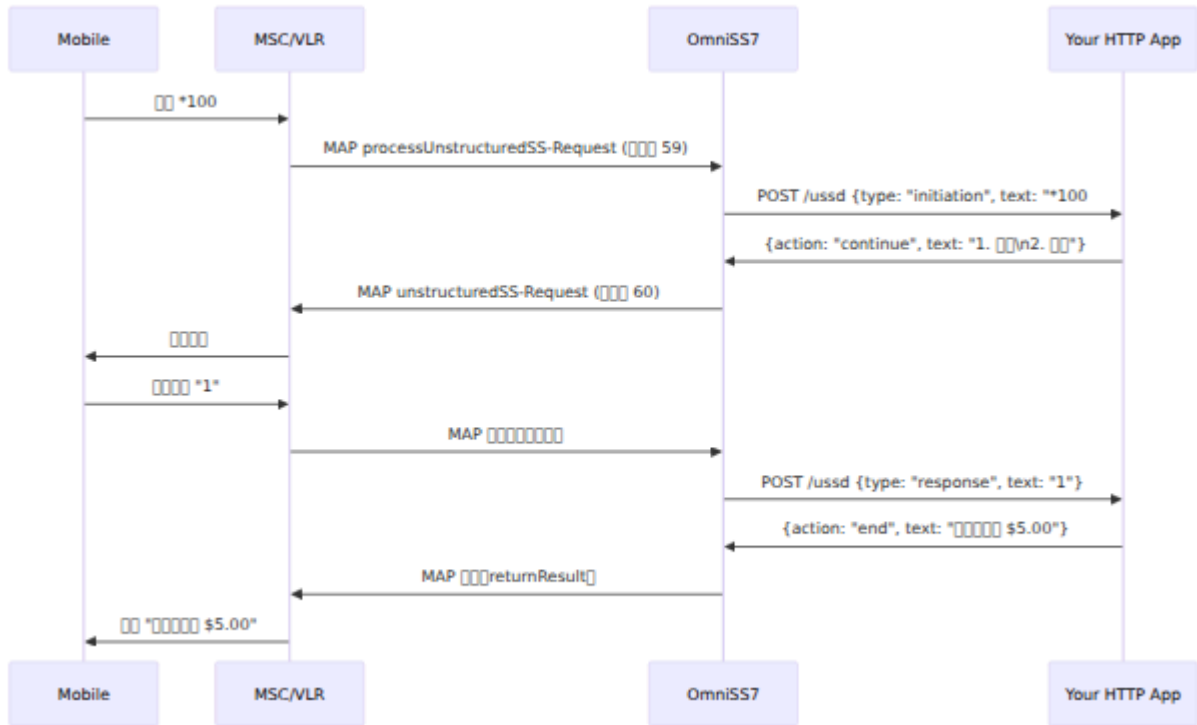
□□	□
□□	□□□□ HTTP POST
□□	GSM 7 □□□□□□□□DCS 0x0F□□□□ 3GPP TS 23.038
□□□□□□	182 □□□□□□□□
□□□□	□□□□□□□□□□ UUID
□□□□	□□□□ SS7 □□□
□□	□□□□□□□□ URL

3GPP □□

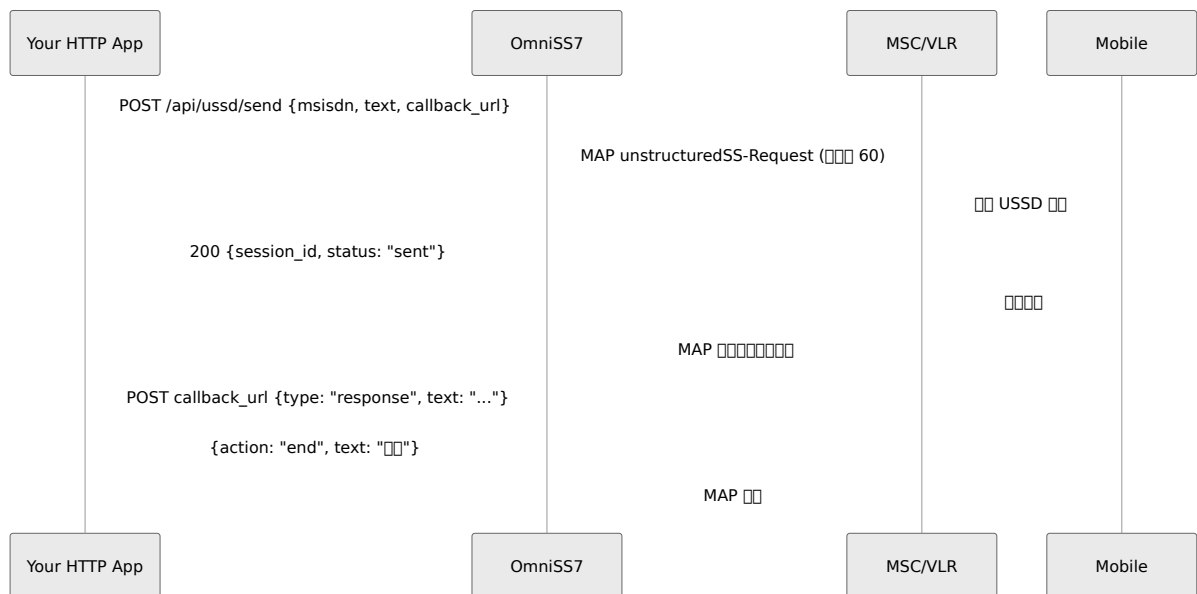
□□	□□□
3GPP TS 23.090	USSD □□□□□□□□□□
3GPP TS 24.090	USSD □□□□□□□□□□
3GPP TS 29.002	MAP □□□USSD-Arg□USSD-Res□□□□ 59/60/61
3GPP TS 23.038	GSM 7 □□□□□□□□□□□□□□

□□

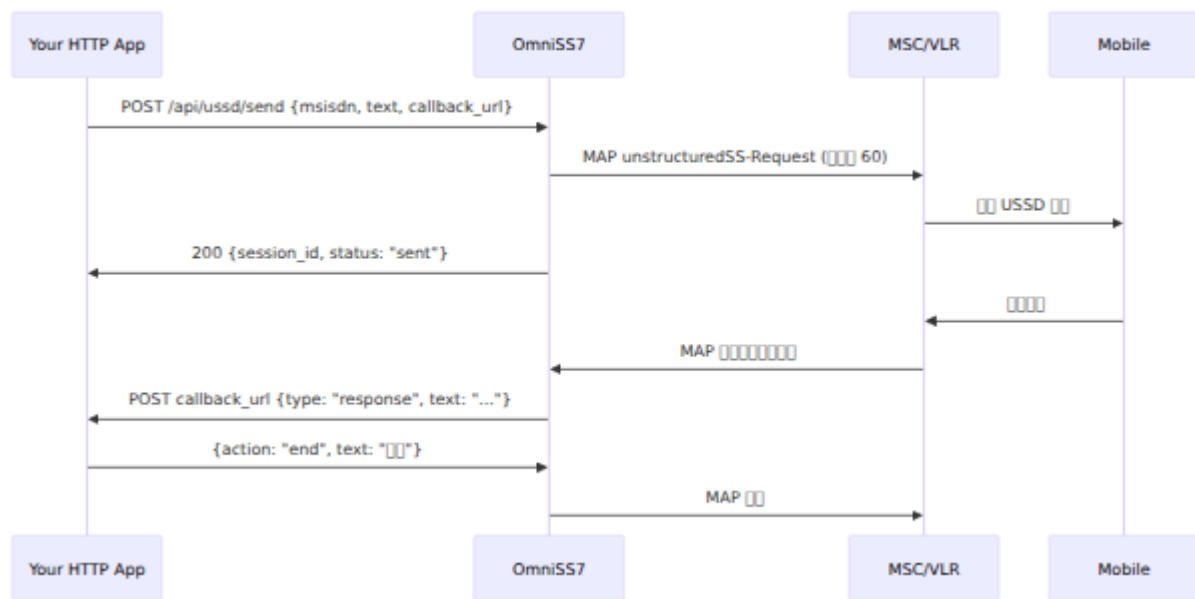
□□□□□□□□



□□□□□□□□□□



□□□□



□□ 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 - 路由  
    routes: [  
      %{pattern: "*100", url: "http://balance-app:9000/ussd"},  
      %{pattern: "*200", url: "http://topup-app:9000/ussd"},  
      %{pattern: "*", url: "http://default-app:9000/ussd"}  
    ],  
  
    # 会话  
    session_timeout_ms: 180_000,      # 会话超时3 分钟  
    turn_timeout_ms: 30_000,          # 会话超时30 秒  
  
    # HTTP 配置  
    http_timeout_ms: 5_000,           # 超时 HTTP POST 5 秒  
  
    # 文本  
    max_text_length: 182              # GSM 7 位编码  
  }
```

配置

配置	数据类型	默认值	说明
ussd_gateway_enabled	boolean	false	USSD 是否启用
ussd_gateway.routes	array	[]	USSD 路由配置，每个路由包含 pattern 和 url，URL 可以是完整的 URL 或相对 URL
ussd_gateway.session_timeout_ms	integer	180_000	USSD 会话超时时间，单位为毫秒
ussd_gateway.turn_timeout_ms	integer	30_000	USSD 转接超时时间，单位为毫秒
ussd_gateway.http_timeout_ms	integer	5_000	USSD HTTP 请求超时时间，单位为毫秒
ussd_gateway.max_text_length	integer	182	USSD 文本最大长度

配置

routes 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、action と text

の 2 つのフィールドを持つ

```
{
  "action": "continue",
  "text": "1. 1\n2. 2\n3. 3"
}
```

応答メッセージの例

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

□□□□

項目	形式	説明
action	文字列	"continue" (継続) または "end" (終了) を指定する
text	文字列	max_text_length (最大文字数) は 182 以下で、\n は改行を表す

□□□□□ USSD (□□ API)

USSD

□□

POST /api/ussd/send

□□

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

Parameters

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

Response

200 OK

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

Example

HTTP 코드	응답	설명
400	<pre>{"error": "invalid request", "required": ["msisdn", "text", "callback_url"]}</pre>	JSON 형식 오류
500	<pre>{"error": " :gsm7_encode_failed, ...}"}</pre>	GSM 7 인코딩 오류 500 400
500	<pre>{"error": "..."} error</pre>	M3UA 오류 error
503	<pre>{"error": "USSD gateway not enabled"} ussd_gateway_enabled false</pre>	ussd_gateway_enabled false

JSON 형식 오류 400 invalid request
GSM 7 인코딩 오류 500 error

cURL

```
curl -X POST http://localhost:9090/api/ussd/send \  
-H "Content-Type: application/json" \  
-d '{  
  "msisdn": "+254712345678",  
  "text": "1",  
  "callback_url": "http://billing-app:9000/ussd"  
}'
```

응답

USSD 응답 session_id

□□□□



□□ MAP □□□ 59

Begin

□□□□ "continue"

WaitingForReply

□□□□ "end" □□□□/□□

□□□□□□MAP □□□ □□□□ "continue"

□□□□ □□□□ □□□□□

Processing

□□□□ "end"
□□□□/□□



□□□□

□□□□□□□□□□□□ "end" □□□□□□□□□□□□□□□□□□□□ MAP □□□□□□□□

□□□□

□□□□□□□□□□ "continue" □□□□

1. □□□□□ □□□□□ `UssdGateway.Registry` □□□□ □□ **GenServer**
2. □□□□□□□□ 60□`unstructuredSS-Request`□□ MAP □□□□□□□□
3. □□□□□□□□□□□□ `turn_timeout_ms`□
4. □□□□□□□□□□□□
5. □□□□□□□□□□□□ "end" □□□□□□

項目	説明	MAP 状態
HTTP ステータス 5xx	システムエラー MAP 状態	34 (systemFailure)
JSON 解析エラー	システムエラー MAP 状態	34 (systemFailure)
USSD メッセージ max_text_length	システムエラー	N/A (サポート外)
システムエラー	システムエラー MAP 状態	34 (systemFailure)
システムエラー	システムエラー MAP 状態	34 (systemFailure)
GenServer 状態	システムエラー	N/A (サポート外)
USSD メッセージ	システムエラー	21 (facilityNotSupported)

監視

USSD メトリクス `/metrics` 9090ポート Prometheus 監視

USSD 監視

メトリクス: `ussd_requests_total` 方向: 受信 方向: 送信 USSD 状態:

- `direction: "inbound"` 送信 `"outbound"` 受信

メトリクス: `ussd_active_sessions` 方向: 受信 方向: 送信 USSD 状態:

```
00: 0000000000 0000000000000000 0 00000000000000000000
ussd_requests_total 00000000
```

```
map_request_duration_milliseconds: 000 00: USSD 0000000000000000
00 00:
```

- operation: "ussd_send"

❏ Prometheus ❏

```
# 统计所有 USSD 请求
rate(ussd_requests_total[5m])

# 统计 USSD 请求 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'████ ${text} ████████'
    })

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 を有効にする

USSD を有効にするには、以下の手順に従ってください。

1. 設定を確認する

- `ussd_gateway_enabled` が `false`
- 設定が正しく適用されている
- M3UA が正常に動作している

2. 設定を変更する

1. `ussd_gateway_enabled: true`
2. 設定が正しく適用されている * 確認する
3. Web UI から M3UA を確認する

トラブルシューティング

USSD が正常に動作しない場合は、以下のエラーメッセージを確認してください。

エラーメッセージ

- URL が OmniSS7 に設定されている
- OmniSS7 が HTTP を使用している
- 設定が正しく適用されている

3. 設定を確認する

1. OmniSS7 の URL を `curl -v http://your-app:9000/ussd`
2. HTTP の設定を確認する
3. 設定が正しく適用されている

結論

USSD が正常に動作しない場合は、"systemFailure" エラーメッセージを確認してください。

配置项

- `turn_timeout_ms` 超时时间
- `http_timeout_ms` HTTP 超时时间
- OmniSS7 配置项

配置项

1. 设置 `turn_timeout_ms` 为 30 秒
2. 设置 `http_timeout_ms` 为 30 秒
3. 设置 OmniSS7 配置项

GSM 7 配置

当 `gsm7_encode_failed` 为 `1` 时，`/api/ussd/send` 返回 `500` 状态码，`error` 字段为 `{:gsm7_encode_failed, ...}`

配置项

- 配置 GSM 7 编码表，支持 CJK 语言

配置项

- 配置 USSD 消息，GSM 7 编码，ASCII 编码，UTF-8 编码
- 配置 3GPP TS 23.038 6.2.1 章节

接口

- **API** 接口，REST API，`/api/ussd/send`
- **MAP** 接口，USSD 消息，M3UA 消息
- 接口，消息
- 接口，Web UI，Prometheus 接口

Web UI 簡介

← 目錄

OmniSS7 Web UIPhoenix LiveView 簡介

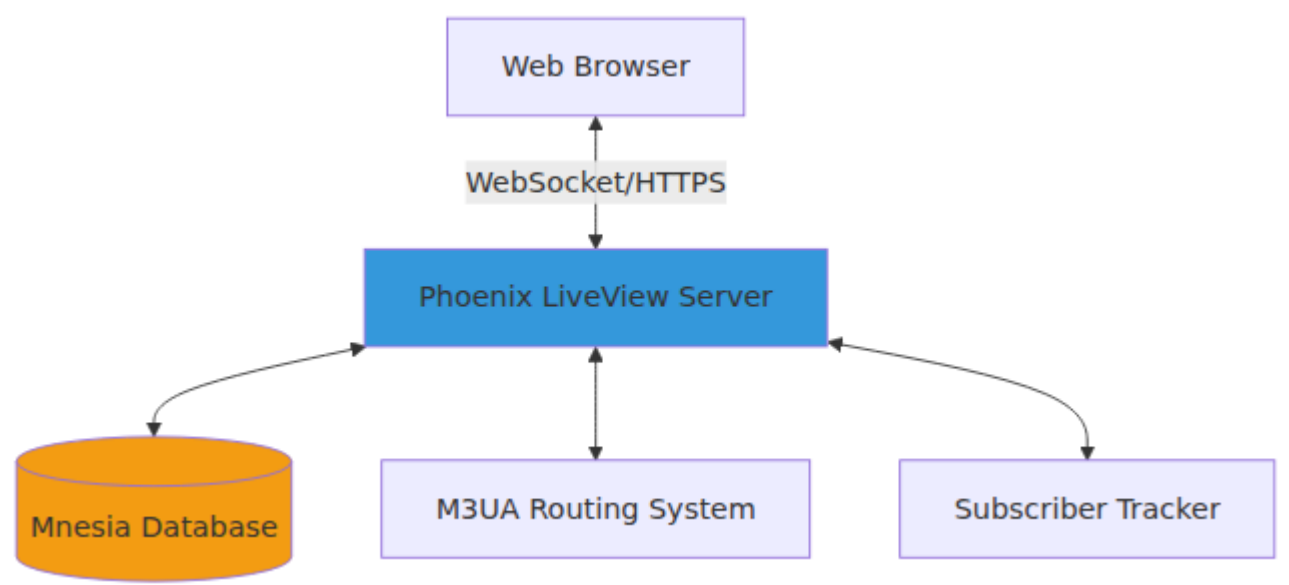
目錄

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

簡介

OmniSS7 Web UI 是 **Phoenix LiveView** 的 Web 界面，用於配置和監控 STP、HLR 和 SMS 服務。

Web UI



Configuration

Item	Value
Protocol	HTTPS (<code>force_ssl</code>)
Port	<code>8087</code> <code>config :control_panel, ControlPanelWeb.Endpoint</code> → <code>https: [port: 8087]</code>
IP	<code>0.0.0.0</code>
Cert File	<code>priv/cert/omnitouch.crt</code>
Key File	<code>priv/cert/omnitouch.pem</code>
App	Bandit (Phoenix)

URL: `https://[server-ip]:8087`

Web UI

準備

1. **SSL** 鍵: 鍵 `priv/cert/` 証明書 SSL 鍵

- `omnitech.crt` - 証明書
- `omnitech.pem` - 証明書

2. 証明書: 鍵 `iex -S mix` 証明書

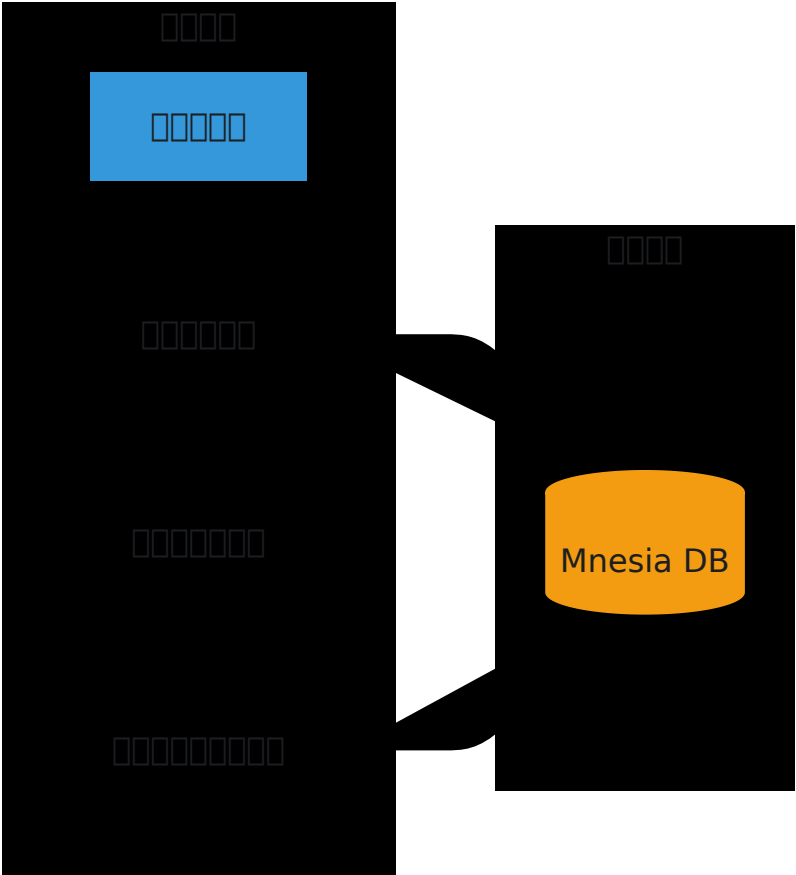
3. 鍵: 鍵 `8087` 鍵 HTTPS 鍵

インストール

インストール `config/config.exs` 証明書 `use_builtin_pages /`
`use_additional_pages` 証明書 `STP/HLR/SMSc` 証明書
鍵

API (URL)	API 名称	STP API	HLR API	SMSc API	API 描述
/application	应用	是	是	是	应用管理 API
/configuration	配置	是	是	是	配置管理 API
/license	许可证	是	是	是	许可证管理 API
/events	SS7 事件	是	是	是	SS7 事件 API SCCP 消息
/logs	日志	是	是	是	日志管理 API
/client	SS7 客户端	是	是	是	SS7 客户端 API MAP 消息
/m3ua	M3UA	是	是	是	M3UA 消息 / ASP 消息
/sctp	SCTP 连接	是	是	是	SCTP 连接 API
/routing	路由	是	是	是	M3UA 路由 API
/routing_test	路由测试	是	是	是	路由测试 API
/hlr_links	HLR 链接	是	是	是	HLR API 链接 管理 API
/subscribers	订阅者	是	是	是	订阅者管理 API (HLR)
/smc_links	SMSc 链接	是	是	是	SMSc API 链接 管理 API

□□□□



□□□□□□

□□ M3UA □□□□□□□□ STP□HLR□MSC□SMSC□□

□□□□□

項目	説明	値
ID	識別子	1
名前	名前	"STP_West"
タイプ	タイプ	client, server, stp
ポート	ポート SS7	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	-

設定

1. 名前
2. 名前
 - ID: 識別子
 - 名前: 名前
 - タイプ: client, server, stp
 - ポート: SS7
 - IP: IP アドレス
 - ポート: 0
 - IP: IP アドレス
 - ポート: 2905
 - ID: M3UA ID
 - ID: international, national
3. 名前 "名前"

名前: Mnesia

名前

- 1. 在 数据库 中 "id" 列
- 2. 数据库 名称
- 3. 在 "数据库"

在: 数据库 ID 数据库名称

数据库

- 1. 在 数据库 中 "id" 列
- 2. 在 数据库 名称

数据库

id	id	id
active	id	数据库名称
aspup	id	ASP 数据库名称
down	id	数据库名称

数据库

数据库名称

数据库

項目	説明	値
PC 名	zone.area.id 4桁	1.2.3 (100)
PC	PC 名	/14 (100) /8 (100)
ID	ID	1
名前	名前	"STP_West"
名前	名前1 = 名前	1
名前	名前	international
名前	名前/名前	-

名前

1. 名前 "名前" 名前
2. 名前
 - 名前: 名前 zone.area.id 4桁 1.2.3 0-16383
 - 名前: 名前 /14 名前
 - 名前 ID: 名前
 - 名前: 名前1 = 名前
 - 名前: 名前 international 名前 national
3. 名前 "名前"

名前: 名前

- 3-8-3 名前: zone.area.id 4桁 1.2.3
- 名前: 0-16383 1100

名前

名前

名前 14 名前 0-16383 名前

PC	PCs	PCs
/14	1 (PCs)	PCs
/13	2 PCs	PCs
/8	64 PCs	PCs
/0	16,384 PCs	PCs/PCs

PC:

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

PCs

PCs

PCs

PC SCCP PCs

PC: PCs

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

PCs

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)
10	AUC (アクセスユーニタ)
142	RANAP
145	gsmSCF (グローバルサービスコントロール)
146	SGSN

SSN について

- **SSN**: サービスネットワーク
- **SSN**: サービスネットワーク
 - = サービスネットワーク
 - = サービスネットワーク

例: SSN=6 (HLR) サービスネットワーク SSN=7 (VLR)

インストール

インストール Mnesia サービスネットワーク

インストール

1. **Web UI** 例: サービスネットワーク Mnesia
2. サービスネットワーク: サービスネットワーク Mnesia
3. **Runtime.exs** 例: `config/runtime.exs` サービスネットワーク Mnesia

□□□□

□□□□□□□□□□□□

1. □□□□□□: □□□□□□□□□□□□
 2. □□□□□□: □□□□□□□□□□□□1 = □□□□□□
 3. □□□□□□: □□□□□□□□ active □□□□□□□□
-

□□□□□□

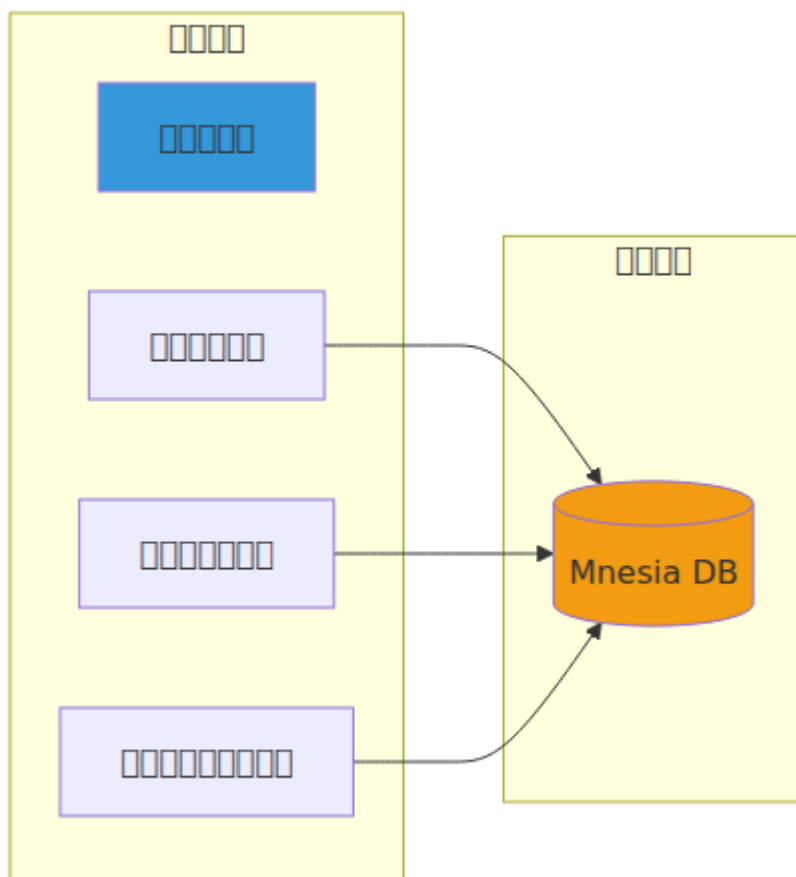
□□: /subscribers

□□: □ HLR

□□□□: □ 2 □

□□□□ UpdateLocation □□□□□□□□□□□□

□□□□□



呼叫数据

参数	描述	值
IMSI	国际移动用户识别码	"50557123456789"
VLR 号码	拜访位置寄存器 GT 号码	"555123155"
MSC 号码	移动交换中心 GT 号码	"555123155"
时间戳	最后 UpdateLocation 时间戳	"2025-10-25 14:23:45 UTC"
持续时间	呼叫持续时间	"2h 15m 34s"

呼叫原因

呼叫原因: 移动发起

- 原因: 移动发起
- 原因 **VLRs**: 拜访 VLR 号码
- 原因 **MSCs**: 移动交换中心 号码

呼叫类型

呼叫类型: 语音呼叫

呼叫: 语音呼叫

呼叫: 语音呼叫

呼叫结果

呼叫 2 次 失败原因: 无响应

SMSc 测试

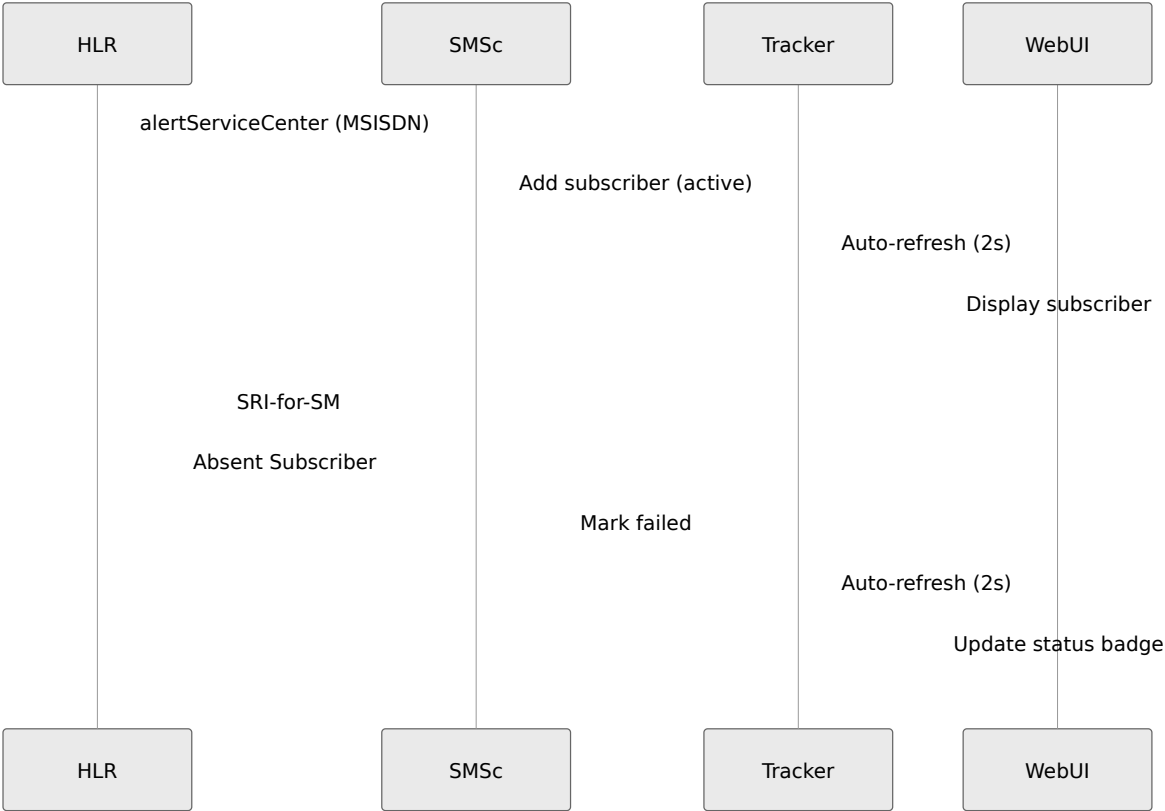
URL: /smc_subscribers

URL: SMC

URL: 2

HLR 测试 alertServiceCenter 测试

测试



Parameters

Field	Description	Value
MSISDN	Subscriber's phone number	"15551234567"
IMSI	Subscriber's IMSI	"001010123456789"
HLR GT	HLR's alertServiceCenter or HLR GT	"15551111111"
MT-FSM	MT-FSM type	5
MO-FSM	MO-FSM type	2
Mode	Mode (MT or MO)	● MT
Time	Time	"2025-10-30 14:23:45 UTC"
Duration	Duration	"15m 34s"

Parameters

- MT (MT): alertServiceCenter
- MO (MO): SRI-for-SM

Parameters

Parameters

- MT: MT
- MO: MO
- HLRs: HLRs

□□□□

□□□□: □□□□□□□□□□

□□□□□□: □□□□□□□□□□

□□: □□□□□□□□□□□□□□□□

□□:

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□ 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 服务
- 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 `priv/cert/omnitouch.pem` を指定
2. ポート **8087**: SSL 設定を有効にし、`8087` で HTTPS を開始
3. 証明書: 証明書と秘密鍵を指定 `iex -S mix`
4. 証明書: SSL 証明書と秘密鍵を指定

SSL 証明書

1. **Mnesia** 設定: `mnesia_storage_type: :disc_copies`
2. **Mnesia** 設定: `Mnesia` を開始
3. 証明書: 証明書と秘密鍵を指定 `Mnesia` を開始

WebSocket

1. **WebSocket** 設定: `WebSocket` を開始
2. 設定: 設定を指定
3. 証明書: 証明書 (F5)

API

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



OmniSS7 Web UI

- □□□□ - □□□□□□□□□□
- □□□□ - Mnesia □□□□□□□□□□
- □□□□ **UI** - □□□□□□□□ (STP/HLR/SMSc) □□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□

□□□□□□□□□□□□□□ **API** □□□

國際電碼

← 國際電碼

國際電碼由國際電碼聯盟（ITU）制定，用於國際電話號碼的編碼。國際電碼由國際電碼聯盟（ITU）制定，用於國際電話號碼的編碼。

國際電碼

- 國際電碼
- 國際電碼由國際電碼聯盟（ITU）制定
- 國際電碼由國際電碼聯盟（ITU）制定
- 國際電碼由國際電碼聯盟（ITU）制定
- 國際電碼
- 國際電碼

國際電碼

國際電碼

國際電碼由國際電碼聯盟（ITU）制定

- 0 = 國際
- 1 = 國際
- 2 = 國際
- 3 = 國際

國際電碼由國際電碼聯盟（ITU）制定

- 0 = 國際
- 1 = ISDN/國際E.164
- 3 = 國際X.121
- 4 = 國際F.69

- 6 = 00000E.212

00000000**NAI**00

- 0 = 00
- 1 = 0000
- 2 = 00000000
- 3 = 000000
- 4 = 0000

000000

- 00000source_tt, source_npi, source_nai0000000000
- 000000dest_tt, dest_npi, dest_nai0000000000
- 00000nil0000000000000000

000000**TT**0000

00 **1**0000000**TT**00

000 0000000000GT0TT=0000000000HLR00TT=100000

```
%{
  gt_prefix: "44",
  peer_id: 1,
  priority: 1,
  description: "000000TT000000",
  source_tt: 0,      # 0000TT=0
  dest_tt: 1        # 000TT=1
}
```

000000

- 000GT=4412345678900TT=0
- 00000gt_prefix="44"0source_tt=0

- GT=441234567890TT=1

2

TT

```
# TT=0A
%{
  gt_prefix: "33",
  peer_id: 1,
  priority: 1,
  description: "TT=0A",
  source_tt: 0
},

# TT=2B
%{
  gt_prefix: "33",
  peer_id: 2,
  priority: 1,
  description: "TT=2B",
  source_tt: 2
}
```

- GT="33123456789"TT=0 → 1
- GT="33123456789"TT=2 → 2

NPI

1 ISDN NPI

NPI=6E.212 PSTN NPI=1E.164

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

[illegible]

2

[illegible]

```
# 00000000NPI=3
%{
  gt_prefix: "1",
  peer_id: 1,
  priority: 1,
  description: "000000",
  source_npi: 3      # 00NPI=3000/X.121
},

# 00000000NPI=1
%{
  gt_prefix: "1",
  peer_id: 2,
  priority: 1,
  description: "000000",
  source_npi: 1      # 00NPI=10ISDN
}
```

00000**NAI**000

00 **1**000000**NAI**00

000 0000000NAI=4000000000NAI=3000000000

```
%{
  gt_prefix: "1",
  peer_id: 1,
  priority: 1,
  description: "00000→0000",
  source_nai: 4,      # 00NAI=4000000
  dest_nai: 3        # 000NAI=3000000
}
```

00000

- 000GT=12125551234NAI=4000000
- 000GT=12125551234NAI=3000000

00 **2**000000**NAI**00

000 00000000000000000000

```
# 00000000HLR
%{
  gt_prefix: "44",
  peer_id: 1,
  priority: 1,
  description: "00000000HLR",
  source_nai: 1      # 00NAI=10000000
},

# 00000000MSC
%{
  gt_prefix: "44",
  peer_id: 2,
  priority: 1,
  description: "00000000MSC",
  source_nai: 3      # 00NAI=3000000000
}
```

0000

00 **1SMS**00000

000 0000HLR0000000000000000SMS000

```
%{
  gt_prefix: "86",
  source_ssn: 8,          # SSN=8 SMSC
  peer_id: 1,
  dest_ssn: 6,           # SSN=6 HLR
  priority: 1,
  description: "SMS",
  source_tt: 0,          # TT=0
  dest_tt: 2,            # TT=2
  source_npi: 6,         # NPI=6
  dest_npi: 1,           # NPI=1 ISDN
  source_nai: 4,         # NAI=4
  dest_nai: 3            # NAI=3
}
```

說明

- SMS SSN=8 GT="86"
-
- SSN SMSC 8 HLR 6
-

2

TT

```
# AAAA[TT]ISDN NPI
%{
  gt_prefix: "44",
  peer_id: 1,
  priority: 1,
  description: "AAAAA/ISDN",
  source_tt: 0,          # [TT]=0
  dest_tt: 2,           # [TT]=2[ ]
  source_npi: 6,        # [NPI]
  dest_npi: 1           # [ISDN NPI]
},

# BB[TT][NPI]
%{
  gt_prefix: "44",
  peer_id: 2,
  priority: 2,          # [ ]
  description: "BB[TT]",
  source_tt: 2,         # [TT]
  dest_tt: 1            # [TT]
  # NPI[ ] = [ ]
}
```

AAAAA

[] **1**AAAAA

[] AAAAAAAAAAAAAAAAAAAAAA

AAAAA AAAAAAA???

```
%{
  gt_prefix: "555",
  peer_id: 1,
  priority: 1,
  description: "HRLR",
  # 
  source_tt: 1,          # 
  dest_tt: 0,            # 
  source_npi: 1,         # ISDN
  dest_npi: 0,           # 
  source_nai: 4,         # 
  dest_nai: 0            # 
}
```

2

```
# 0000A00TT=00000000TT=1
%{
  gt_prefix: "1",
  peer_id: 1,
  priority: 1,
  description: "000000A000",
  source_tt: 0,
  dest_tt: 1,
  dest_npi: 1,          # 00ISDN NPI
  dest_nai: 4           # 0000NAI
},

# 0000B00TT=300000000000TT=1
%{
  gt_prefix: "44",
  peer_id: 2,
  priority: 1,
  description: "000000B000",
  source_tt: 3,
  dest_tt: 1,
  dest_npi: 1,
  dest_nai: 4
}
```

00 **3**000000000

000 000000000000000000000000

000000 0000000000000000


```
# 00000000000000000000
%{
  gt_prefix: "12025",      # 00000000
  peer_id: 1,
  priority: 1,
  description: "00000000",
  source_nai: 4,           # 0000000000000000
  dest_nai: 3,             # 000000
  source_npi: 6,           # 000000
  dest_npi: 1              # 0HLR0ISDN
}
```

00 40000000

0000 00000000000000000000

000000 00000000000000000000

```
# 0000000000000000
%{
  gt_prefix: "999",
  peer_id: 1,
  priority: 1,
  description: "00000000",
  source_tt: 0,           # 0000
  dest_tt: 2,            # 0000
  source_npi: 0,         # 0000
  dest_npi: 1,           # 00ISDN
  source_nai: 0,         # 0000
  dest_nai: 3            # 0000
},

# 0000000000000000
%{
  gt_prefix: "999",
  peer_id: 1,
  priority: 2,
  description: "000000",
  source_tt: 2,          # 000000
  # dest_*[] = 000000
}
```

00000000

00Web UI

1. 000 00 → 0000000
2. 00 “00000000TT/NPI/NAI”
3. 00000000000000
4. 00 0000
5. 00 0000 0000000

API

GT API 在 `/routing/gt-routes` 端口 **8445** 通过 TLS 的 JSON 格式
`prefix` `gt_prefix` `peer_id` `TT/NPI/NAI`

```
curl -k -X POST https://localhost:8445/routing/gt-routes \
-H "Content-Type: application/json" \
-d '{
  "prefix": "44",
  "peer_id": 1,
  "priority": 1,
  "description": "TT",
  "source_tt": 0,
  "dest_tt": 1
}'
```

参数	类型	必填	说明
<code>prefix</code>	字符串	是	GT 前缀
<code>peer_id</code>	整数	是	对等体 ID = 0
<code>priority</code>	整数	是	优先级 1
<code>description</code>	字符串	否	描述
<code>source_ssn</code> / <code>dest_ssn</code>	整数	否	源/目标 SSN
<code>source_tt</code> / <code>dest_tt</code>	整数	否	源/目标 TT
<code>source_npi</code> / <code>dest_npi</code>	整数	否	源/目标 NPI
<code>source_nai</code> / <code>dest_nai</code>	整数	否	源/目标 NAI

DELETE `/routing/gt-routes/{prefix}:{peer_id}`
`/routing/gt-routes/44:1`

請求パラメータ

請求パラメータは以下の通りです。

- GT番号“441234567890”
- TTNPINAI
- “”
- 請求元番号

レスポンス

ヘッダ

ヘッダは以下の通りです。

- GT番号
- source_ssn番号
- source_tt番号
- source_npi番号
- source_nai番号
- 請求元番号

ボディ

ボディはdest_で始まります。

- dest_tt/dest_npi/dest_nai“”
- dest_*

フッター

:bag

- GT番号 : bag

- `{gt_prefix, source_ssn}`
 - `TT/NPI/NAI` + `SSN`
 -
-

- 1.
 - 2.
 3. `INFO`
 - 4.
 5. `source_*`
 - 6.
 - 7.
-

- `←`
 - `STP` -
 - `API` -
 - `API` - REST API
-

OmniSS7 `Omnitouch`

M3UA STP 配置指南

← 配置指南

简介

M3UA STP 配置 (PC) 需要配置 MTP3 (GT) 和 SCCP 配置 SS7 网络。PC 配置

配置

配置

- GT 配置 SCCP 配置 GT → GT 配置
- PC 配置 MTP3 配置 DPC → PC 配置

GT 配置

- 配置 MTP3 配置
- 配置 GT "447123456789"
 - 配置 "447" 配置 3 配置 "44" 配置 2 配置
 - 配置 "447" 配置

配置

配置 GT 配置

配置 config/runtime.exs 配置

```
config :omniss7,  
  enable_gt_routing: true, # 配置 false 配置 PC 配置
```

GT

```
config :omniss7,
  m3ua_gt_routes: [
    # 1234 1
    %{gt_prefix: "1234", peer_id: 1, priority: 1, description: "1"},
    # 44 2
    %{gt_prefix: "44", peer_id: 2, priority: 1, description: "2"},
    # 447xxx
    %{gt_prefix: "447", peer_id: 2, priority: 1, description: "2"},
    # 61 3
    %{gt_prefix: "61", peer_id: 3, priority: 1, description: "3"}
  ]
```

STP `config/stp_example_runtime.exs`

1 GT

SCCP UDT Called GT = "447700123456"

- GT "447700123456"
- "447" → 2
- 2

例 2 例題 GT 例題 PC

例題 SCCP UDT 例 Called GT = "331234567890" 例 DPC = 100

例題

- 例題 GT 例 "331234567890"
- 例題 "33" 例 GT 例
- 例題 PC 例
- DPC 100 → 例題 1 例題 PC 例題
- 例題例題例題 1

例 3 例題 GT

例題例題 SCCP 例 M3UA DATA 例 MTP3 例 DPC = 200

例題 例題

- 例題 SCCP 例題 GT
- 例題 PC 例
- DPC 200 → 例題 2
- 例題例題例題 2

例題例題

例題例題 M3UARouting 例題例題 iex -S mix 例題例題

📄 GT 📄

```
# 📄 GT 📄
{:ok, routes} = M3UARouting.get_all_gt_routes()

# 📄 14 📄
# {:m3ua_gt_route, gt_prefix, source_ssn, peer_id, dest_ssn,
# priority,
# description, enabled, source_tt, dest_tt, source_npi, dest_npi,
# source_nai, dest_nai}
#
# 📄source_ssn 📄/📄📄📄📄📄 nil📄
[
  {:m3ua_gt_route, "1234", nil, 1, nil, 1, "📄📄", true, nil, nil,
  nil, nil, nil, nil},
  {:m3ua_gt_route, "44", nil, 2, nil, 1, "📄📄", true, nil, nil,
  nil, nil, nil, nil},
  {:m3ua_gt_route, "447", nil, 2, nil, 1, "📄📄", true, nil, nil,
  nil, nil, nil, nil},
  {:m3ua_gt_route, "61", nil, 3, nil, 1, "📄📄📄", true, nil, nil,
  nil, nil, nil, nil}
]
```

📄 GT 📄

add_gt_route/3 📄📄📄📄📄📄📄📄 ID📄📄📄📄📄 priority📄 description📄
source_ssn📄 dest_ssn📄 TT/NPI/NAI 📄📄📄📄📄📄 enabled📄📄📄📄📄

```
# 添加gt_route(gt_prefix, peer_id, opts \ [])

# 添加路由
M3UARouting.add_gt_route("49", 4, priority: 1, description: "HLR")

# 添加 SMS 路由 SSN 8 到 SMSC SSN 6 到 HLR
M3UARouting.add_gt_route("55", 4,
  source_ssn: 8,
  dest_ssn: 6,
  priority: 1,
  description: "HLR 到 SMS 路由"
)
```

删除 GT 路由

```
# 删除gt_prefix, peer_id, 删除 source_ssn
M3UARouting.remove_gt_route("49", 4)
```

查找 GT 路由

```
# 查找路由 GT
# 查找lookup_peer_by_gt(global_title, source_ssn \ nil, opts \ [])
{:ok, peer, rewrites} =
M3UARouting.lookup_peer_by_gt("447700123456")

# `peer` 是 13 字节的 m3ua_peer 结构
# {:m3ua_peer, peer_id, peer_name, protocol, role, local_ip,
  local_port,
# remote_ip, remote_port, routing_context, point_code,
  network_indicator, status}
#
# `rewrites` 是路由重写规则 MAP
# nil 表示“无重写”
#   %{dest_ssn: nil, dest_tt: nil, dest_npi: nil, dest_nai: nil}
```

路由重写规则 "55" 到 dest_ssn: 6

```
{:ok, peer, rewrites} =  
M3UARouting.lookup_peer_by_gt("5512345678", 8)  
# rewrites => %{dest_ssn: 6, dest_tt: nil, dest_npi: nil,  
dest_nai: nil}
```

```
peer_id: 0 DROP {:error, :dropped} GT {:error,  
:no_gt_route}
```

STP

GT

```
[info] M3UA DATA: OPC=1000, DPC=200, Called GT="447700123456"  
1  
[info] GT 447700123456  
[info] 1 2
```

GT

```
[info] M3UA DATA: OPC=1000, DPC=200, Called GT="331234567"  
1  
[info] GT 331234567  
[info] 1 2 PC
```

SCCP 表

表 SCCP 表

表	表	表
UDT (表)	0x09	表GT 表
UDTS (表)	0x0A	表
XUDT (表)	0x11	表GT 表
XUDTS (表)	0x12	表
LUDT (表)	0x13	表GT 表
LUDTS (表)	0x14	表

XUDT 表 LUDT 表 SCCP 表 表 STP 表 表
表 STP 表

表 GT 表

- **GTI 0-4**表 GT 表
- **BCD** 表
- 表

表

- **GT** 表 $O(n)$ 表 $n = \text{GT 表}$
- 表
- 表 GT 表 < 1000 表

--	--	--	--

GT

Called GT=nil

□□□□□

1. SCCP 与 UDT/XUDT/LUDT
2. 与 UDT/XUDT/LUDT
3. SCCP 与 UDT/XUDT/LUDT

□ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □

□□□□□

1. 取得 GT 的公鑰
2. 取得 `peer_id` 的公鑰
3. 公鑰 = 公鑰

□□□ **PC** □□□

□□□□□ GT □□□□□

□□□□□

1. `enable_gt_routing` `true`
2. PC `enable_dpc`
3. PC `enable_dpc`

□□□□

1.

- 在 "447" 和 "448" 和 "449" 之间添加路由
- 在 "44" 之前添加路由

2. 配置路由

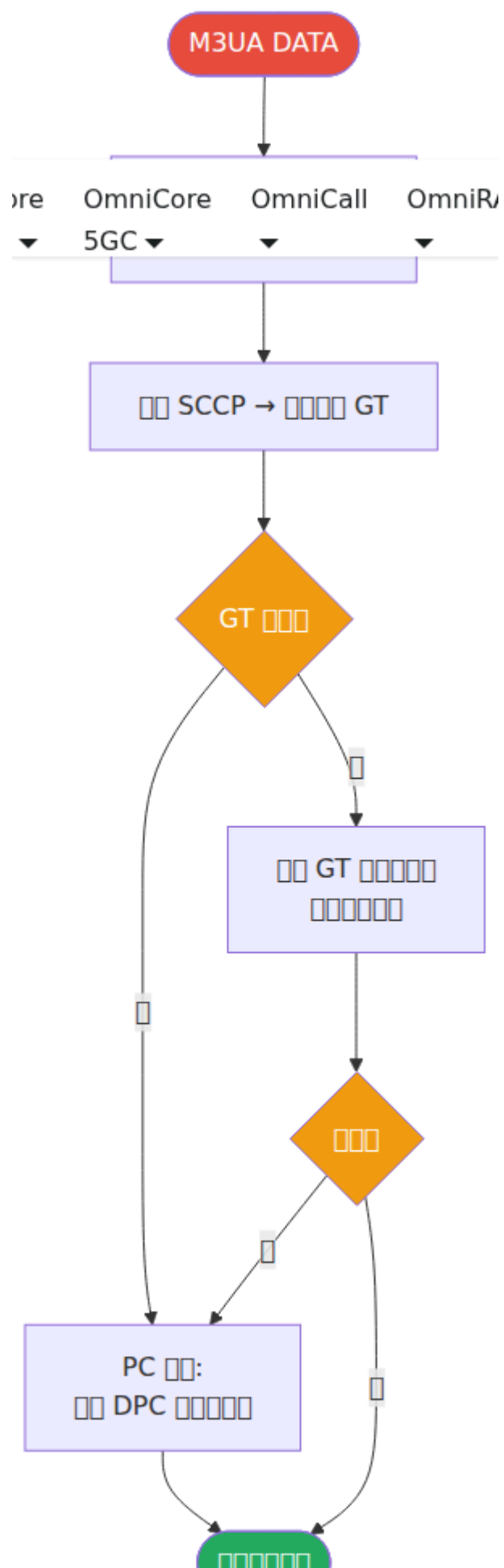
```
M3UARouting.add_gt_route("1234", 1, priority: 1,  
  description: "ACME Corp - 内部网络")
```

3. 配置路由 `lookup_peer_by_gt/1` 函数

4. 配置路由

5. 配置路由 PC 网络







OmniSS7 Omnitouch

