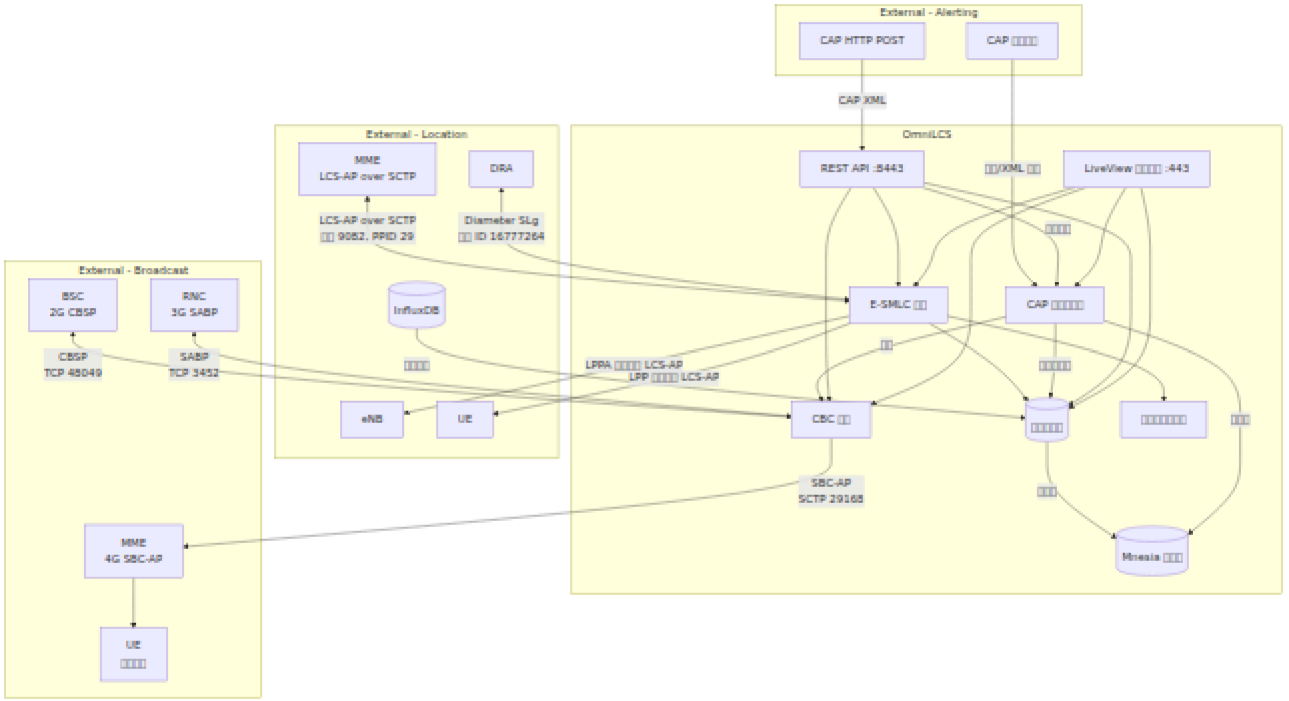
OmniLCS □作指南

OmniLCS □一个集成的 LTE/GSM 位置和小区广播平台。它结合了 **E-SMLC**（演进服务移动位置中心）用于用户设备定位，以及 **CBC**（小区广播中心）用于公共警报和商业广播服务，所有这些都在一个单一的 Elixir/OTP 应用程序中。

架构概述



功能概述

E-SMLC -- 位置服务

- 定位方法：小区 ID、增强小区 ID（E-CID）、GNSS/A-GPS、OTDOA
- **LCS-AP over SCTP**（**SLs** 接口）：根据 3GPP TS 29.171 使用原生 LCS-AP 与 MME 通信，端□ 9082, PPID 29
- **LPPA/LPP** 中继：通过 LCS-AP 连接导向信息将定位协议消息发送到 eNB 和 UE
- 小区数据库：存储小区位置以用于小区 ID 和 OTDOA 定位。支持从华为 U2020 XLSX（GSM/UMTS/LTE/NR）、特定厂商的 CSV、JSON 和 InfluxDB 同步导入。持久化到 Mnesia，以便在重启后生存

- **OTDOA** □□□□□□□□□□□□□□□□ RSTD □□□□ UE □□
- □□□□□□□□□□□□□□□□□□ CSV □□□□□□□□□□□□□□□ InfluxDB □□□□
- □□□□□□□ IMSI □□□□□□□□□□□□□□□□□□□□□□□□□□□ Mnesia □□□□□ KML/CSV □□
- □□□□□□□□□□ IMSI □□□□ GNSS □□□ E-CID □□□□□RSRP/RSRQ□□□□□□□□□□□□□□□ InfluxDB □□□□□□□□□□□□□□□□□ KML/CSV □□

GMLC -- □□□□□□□□

- **Le** □□□□□ 3GPP TS 29.172 □□□ LCS □□□□PSAP□□□□□□□□□□□□□□ Diameter □□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ InfluxDB □□□□□ Diameter LRR □□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□/□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□ Diameter □□□□□□□□LRR□□□

CBC -- □□□□

- **2G CBSP**□□□□ 48049 □□ TCP □□□□□□□□ BSC □□□
- **3G SABP**□□□□ 3452 □□ TCP□Iu-BC □□□3GPP TS 25.419□□□□□ TS 25.414 §7.1.3.3□□CBC □□□□□□ RNC □□□□□□□□□□ RNC □□□□□□□□/□□□□□
- **4G SBC-AP**□□□□ MME □□□□ SCTP □□□□□□ 29168
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ CBS □□□□□□□□□□□□□□□□□□ 3GPP TS 23.038
- □□□
- □□□□□□□GSM 7 □□ UCS-2 □□□CBS □□□□□□□□□□□
- □□□□□□□□ ETWS□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- **Mnesia** □□□□□□□□□□□ CAP □□□□□□□□□ Mnesia disc_copies □□□□□□□□□□□□□□□□□□□□ □□
- **PWS** □□□□□□□□ MMEs □ PWS-Restart-Indication □ PWS-Failure-Indication
- **CAP** □□□□□□□□ HTTP POST □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□CAP□v1.2 XML□□□□□□□□ □□□□□ TAC/LAC□□□□□□□□□□□□□□□□□□□□□□

□□□□

- **REST API**□□□□ 8443 □□ HTTPS□□□□□□□□□□□□□□□□□□□□□□□□

- **LiveView** 控制面板：在端口 443 上的 HTTPS，具有实时仪表板、位置测试、小区数据库管理、Diameter 对等体监控和 CBC 广播组成

文档结构

文档	描述
配置参考	所有配置参数及其表格和示例
小区广播操作	CBSP（2G）、SABP（3G）和 SBC-AP（4G）操作，消息格式化，故障排除
CAP 警报操作	CAP v1.2 摄取，多边形解析，审批工作流，馈送轮询
位置服务	E-SMLC 定位方法，LCS-AP 接口，小区数据库，OTDOA
用户跟踪与驾驶测试	定期跟踪，虚拟驾驶测试活动，RSRP/RSRQ 测量，KML/CSV 导出
GMLC 与 Le 接口	外部 LCS 客户端访问，定期和地理围栏位置，InfluxDB 日志记录
REST API 参考	所有 API 端点及其请求/响应示例
Web 接口指南	LiveView 控制面板页面和功能

接口摘要

接口	协议	传输	端口	方向	**3GPP 参考**
SLs (E-SMLC ↔ MME)	LCS-AP	SCTP	9082	OmniLCS 连接到 MMEs	TS 29.171
SLg/Le (GMLC ↔ LCS 客户端/DRA)	Diameter	SCTP	3868	从 LCS 客户端入站，向 DRA 出站	TS 29.172
CBSP	CBSP	TCP	48049	BSC 连接到 OmniLCS	TS 48.049
SABP (Iu-BC)	SABP	TCP	3452	向 RNC 出站 / 从 RNC 入站	TS 25.414 §7.1.3.3
SBC-AP	SBC-AP	SCTP	29168	OmniLCS 连接到 MMEs	TS 29.168
REST API	HTTPS	TCP	8443	客户端连接到 OmniLCS	--
控制面板	HTTPS	TCP	443	浏览器连接到 OmniLCS	--
小区同步	HTTP	TCP	8086	OmniLCS 查询 InfluxDB	--

监督树

OmniLCS 在一对一监督下启动以下进程:

1. **OmniLcs.Persistence** -- Mnesia 初始化和小区数据库及 CAP 警报的磁盘持久性

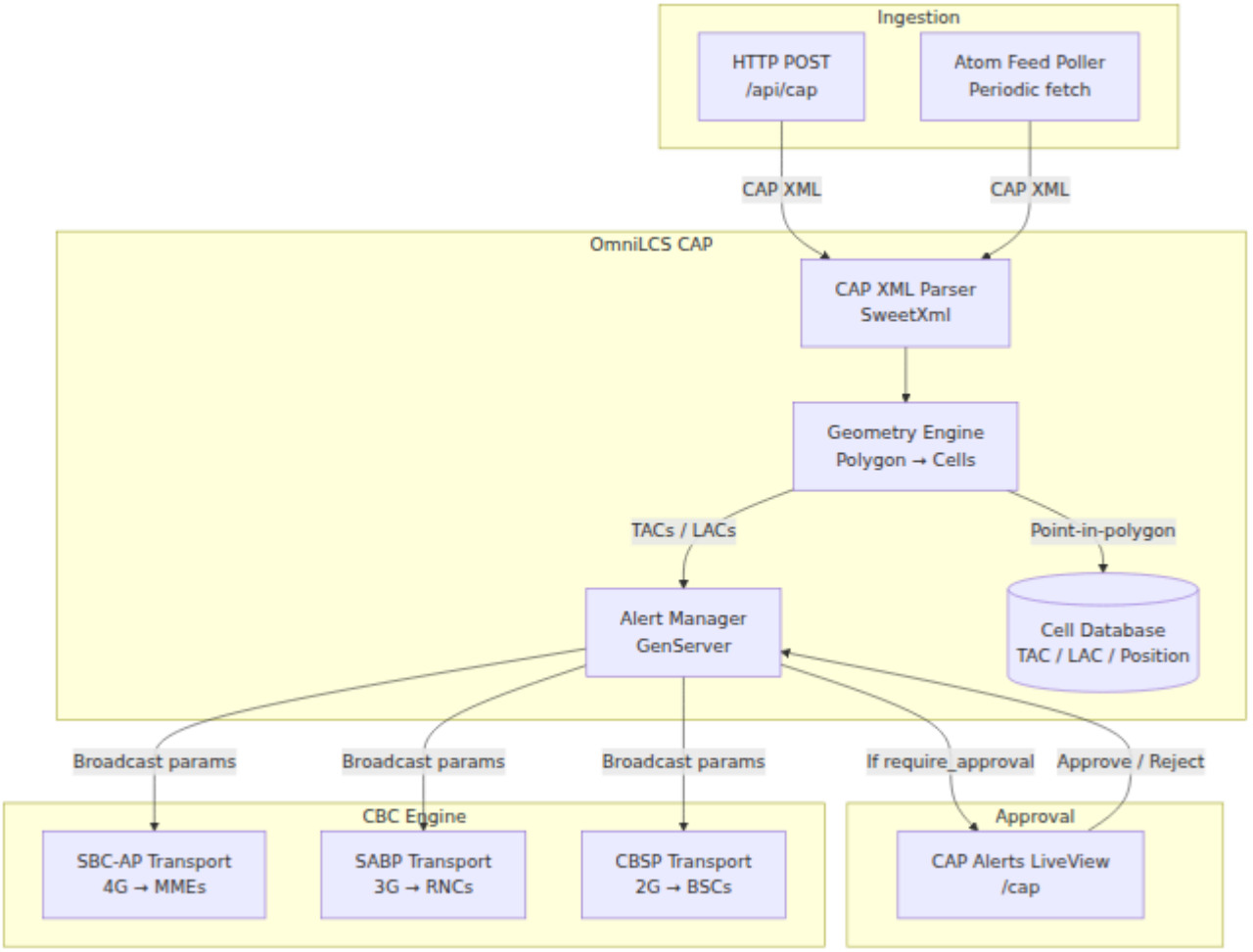
2. **OmniLcs.Context** -- ETS 表初始化（cell_database、location_session、pending_transactions）；在启动时从 Mnesia 加载持久化小区
3. **OmniLcs.InfluxDb** -- InfluxDB 连接池
4. **OmniLcs.Esmlc.LocationLogger** -- 位置修正日志记录到 CSV 和 ETS
5. **Task.Supervisor** -- 异步任务执行
6. **OmniLcs.Sls.SctpTransport** -- SLs LCS-AP SCTP 客户端连接到 MMEs
7. **DiameterEx.Supervisor** -- Diameter 服务和对等体管理（SLg 和其他 Diameter 接口）
8. **OmniLcs.Esmlc.CellSync** -- 定期 InfluxDB 小区同步
9. **OmniLcs.Cbc.CbspConnectionSupervisor** -- 用于 2G CBSP TCP 连接的 DynamicSupervisor
10. **OmniLcs.Cbc.CbspTransport** -- CBSP TCP 监听器（端口 48049）
11. **OmniLcs.Cbc.SabpConnectionSupervisor** -- 用于 3G SABP TCP 连接的 DynamicSupervisor
12. **OmniLcs.Cbc.SabpTransport** -- SABP TCP 监听器（端口 3452）
13. **OmniLcs.Cbc.Engine** -- CBC 广播状态管理（2G、3G 和 4G）
14. **OmniLcs.Cbc.SctpTransport** -- SBC-AP SCTP 客户端连接到 MMEs
15. **OmniLcs.Cap.AlertManager** -- CAP 警报生命周期管理（解析、解决、批准、广播）；持久化到 Mnesia
16. **OmniLcs.Cap.FeedPoller** -- 定期 CAP Atom 喂送轮询
17. **OmniLcs.Gmlc.ClientRegistry** -- 授权的外部 LCS 客户端管理
18. **OmniLcs.Gmlc.SessionSupervisor** -- 用于定期和触发位置会话的 DynamicSupervisor
19. **OmniLcs.Tracking.SessionSupervisor** -- 用于用户跟踪监视会话的 DynamicSupervisor
20. **OmniLcs.DriveTest.CampaignSupervisor** -- 用于虚拟驾驶测试活动的 DynamicSupervisor

CAP 警报□作指南

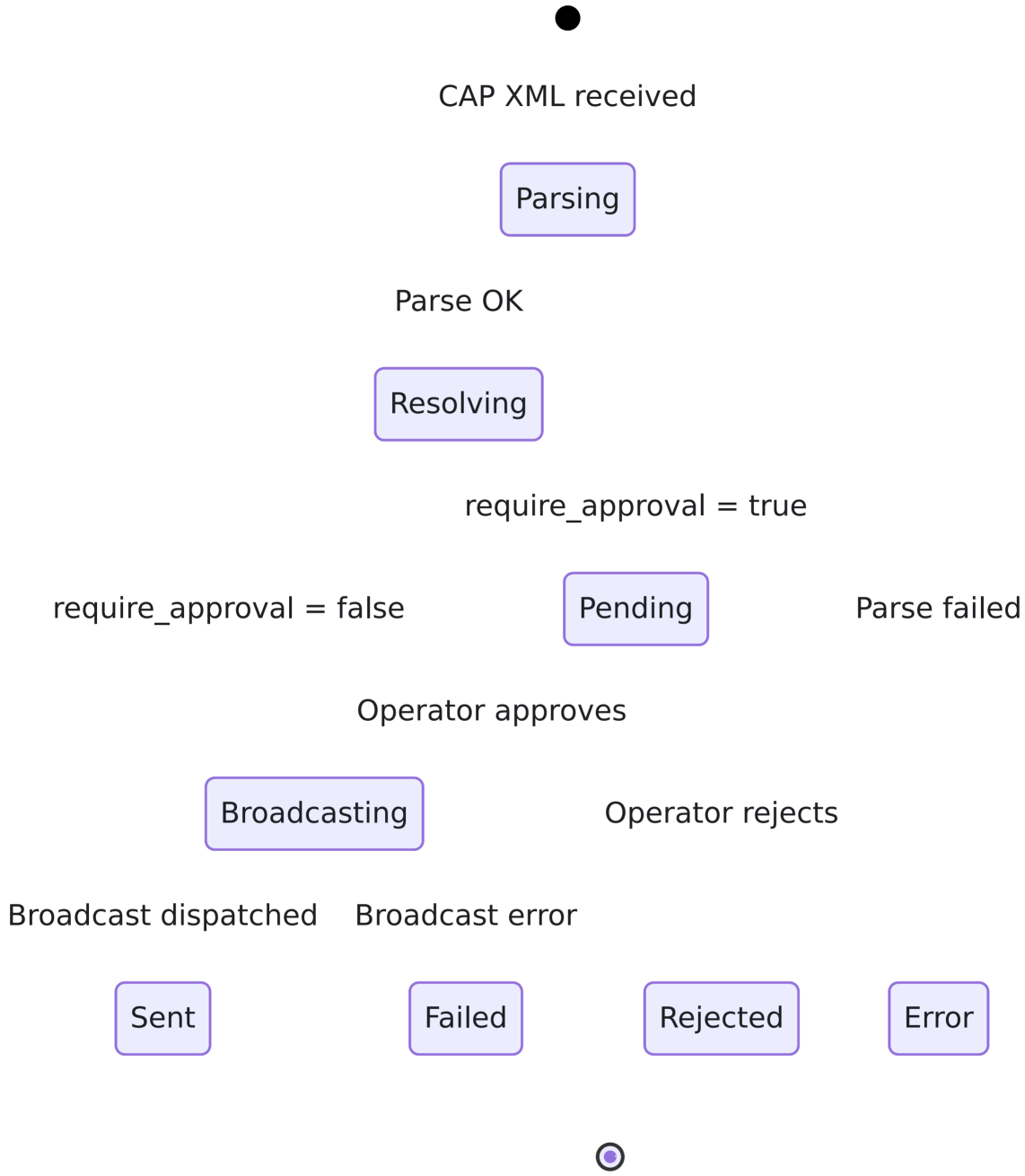
OmniLCS 接受来自外部警报机构（天气服务、应急管理机构等）的 通用警报协议 **(CAP) v1.2** XML 消息，解析地理警告区域到网络的蜂窝基础设施，并通过现有的 CBC 引擎在 4G (SBC-AP)、3G (SABP) 和 2G (CBSP) 接口上触发蜂窝广播。

支持两种摄取方法：**HTTP POST**（推送）用于直接集成和 **Atom** 订阅轮询（拉取）用于监控标准 CAP 订阅源。可配置的 手动审批步骤 允许操作员在广播前预览警报。

□□


Ingestion
HTTP POST
/api/cap
Atom Feed Poller
Periodic fetch
CAP XML
CAP XML
OmniLCS CAP
CAP XML Parser
SweetXml
Geometry Engine
Polygon → Cells
TACs / LACs
Point-in-polygon
Alert Manager
GenServer
Cell Database
TAC / LAC / Position
Broadcast params
Broadcast params
Broadcast params
If require_approval
Approve / Reject
CBC Engine
SBC-AP Transport
4G → MMEs
SABP Transport
3G → RNCs
CBSP Transport
2G → BSCs
Approval
CAP Alerts LiveView
/cap

警报生命周期



当接收到 CAP 警报（通过 HTTP POST 或订阅轮询）时：

1. **CAP XML** 解析器 提取警报信封、信息块、警告区域（多边形）和 CB 参数
2. 几何引擎 使用射线投射（点在多边形内）将每个多边形与蜂窝数据库匹配，将警告区域解析为 TAC（4G）、SAI（3G）和 LAC（2G）的列表
3. 如果 `require_approval` 为 `true`，警报将排队为 待处理 并出现在操作员 LiveView UI 中以供预览和审批
4. 如果 `require_approval` 为 `false`，警报将自动批准并立即发送到 CBC 引擎
5. 在广播时，CBC 引擎向所有连接的 MMEs、RNCs 和 BSCs 发送写替换警告请求（4G SBC-AP）、写替换（3G SABP）和/或 写替换（2G CBSP）

□□

CAP □□

```
config :omnilcs, :cap,
  # Require operator approval before broadcasting
  require_approval: true,

  # PLMN identity for broadcast messages
  plmn: %{mcc: "001", mnc: "01"},

  # Use cell coverage radius for polygon matching (vs center-point
only)
  coverage_aware: false,

  # Atom feed sources to poll (empty = no polling)
  feeds: []
```

CAP 参数

参数	类型	必需	默认	描述
`require_approval`	boolean	否	`true`	当为 `true` 时，警报在广播前排队等待操作员批准。当为 `false` 时，警报将自动批准并立即广播。
`plmn`	map	否	`%{mcc: "001", mnc: "01"}`	广播消息中使用的 PLMN 身份（MCC/MNC）。必须与服务网络匹配。
`plmn.mcc`	string	□	`"001"`	移动国家代码（3 位数字）。
`plmn.mnc`	string	□	`"01"`	移动网络代码（2-3 位数字）。
`coverage_aware`	boolean	否	`false`	当为 `true` 时，多边形匹配使用每个蜂窝的覆盖半径来确定交集，而不仅仅□中心点。对于多边形边缘的大蜂窝非常有用。
`feeds`	list	否	`[]`	要轮询的 CAP Atom 订阅配置列表。请参见 订阅配置。

订阅配置

`feeds` 列表中的每个条目定义了一个要监控的 CAP Atom 订阅源。

```
config :omnilcs, :cap,
  feeds: [
    %{url: "https://alerts.weather.gov/cap/us.php?x=1",
poll_interval_seconds: 60},
    %{url: "https://feeds.meteoalarm.org/api/v1/warnings/atom",
poll_interval_seconds: 120}
  ]
```

参数	类型	必需	默认	描述
`url`	string	□	--	CAP Atom 订阅的 URL。必须返回嵌入 CAP 警报条目的 Atom XML（RFC 4287）。
`poll_interval_seconds`	integer	否	`60`	订阅轮询之间的秒数。较低的值提供更快的警报摄取，但增加网络流量。

订阅轮询器跟踪已见警报标识符以避免重新处理。每个新的 `<entry>` 在订阅中提取其嵌入的 `<alert>` XML 并提交给警报管理器。

配置示例：完整 CAP 设置

```
config :omnilcs, :cap,
  require_approval: true,
  plmn: %{mcc: "001", mnc: "01"},
  coverage_aware: true,
  feeds: [
    %{url: "https://alerts.weather.gov/cap/us.php?x=1",
poll_interval_seconds: 60}
  ]
```

工作原理：来自 NWS 天气订阅的 CAP 警报每 60 秒检查一次。新警报被解析，其多边形警告区域通过覆盖感知交集与蜂窝数据库匹配。匹配的警报出现在 CAP 警报 LiveView 页面供操作员审核。操作员可以预览警报详情（消息文本、严重性、匹配的蜂窝、TACs/LACs），然后批准广播或拒绝丢弃。

蜂窝数据库：TAC、LAC 和 RAT 字段

CAP 多边形解析依赖于数据库中的蜂窝具有填充的 **TAC**（4G）、**LAC**（2G 和 3G）、**SAC**（3G）和 **RAT** 字段。这些字段决定了每个蜂窝使用的广播接口。

CAP □□□□□

□□	□□	□□
`tac`	integer	□□□□□□□□□ 4G SBC-AP □□□□□ TAC □□□□□□□□□ MMEs □ TAI □□□□
`lac`	integer	□□□□□□□□□ 2G CBSP □ 3G SABP □□□□□ LAC □□□□□□□□□ BSCs □□□□□□□□□ RNCs □ SAI □□□□
`rat`	string	□□□□□□□ `"4g"` □ `"3g"` □ `"2g"` □□□□□ `tac` □ `lac` □□□□□□□□□□

□□□□□□□□□□□□□□□□□

- **REST API** □□□□□□□□□□□ `POST /api/cells` □ `PUT /api/cells/:id` □
- **JSON** □□□ `priv/cells.json` □□□□□□□□ `"tac"` □ `"lac"` □ `"rat"` □□
- □□□□ □□□□□□ LiveView □□

□ InfluxDB □□□□□□□□□ TAC/LAC/RAT □□□□□□InfluxDB □□□□□□□□□□□□□ `nil` □□□□

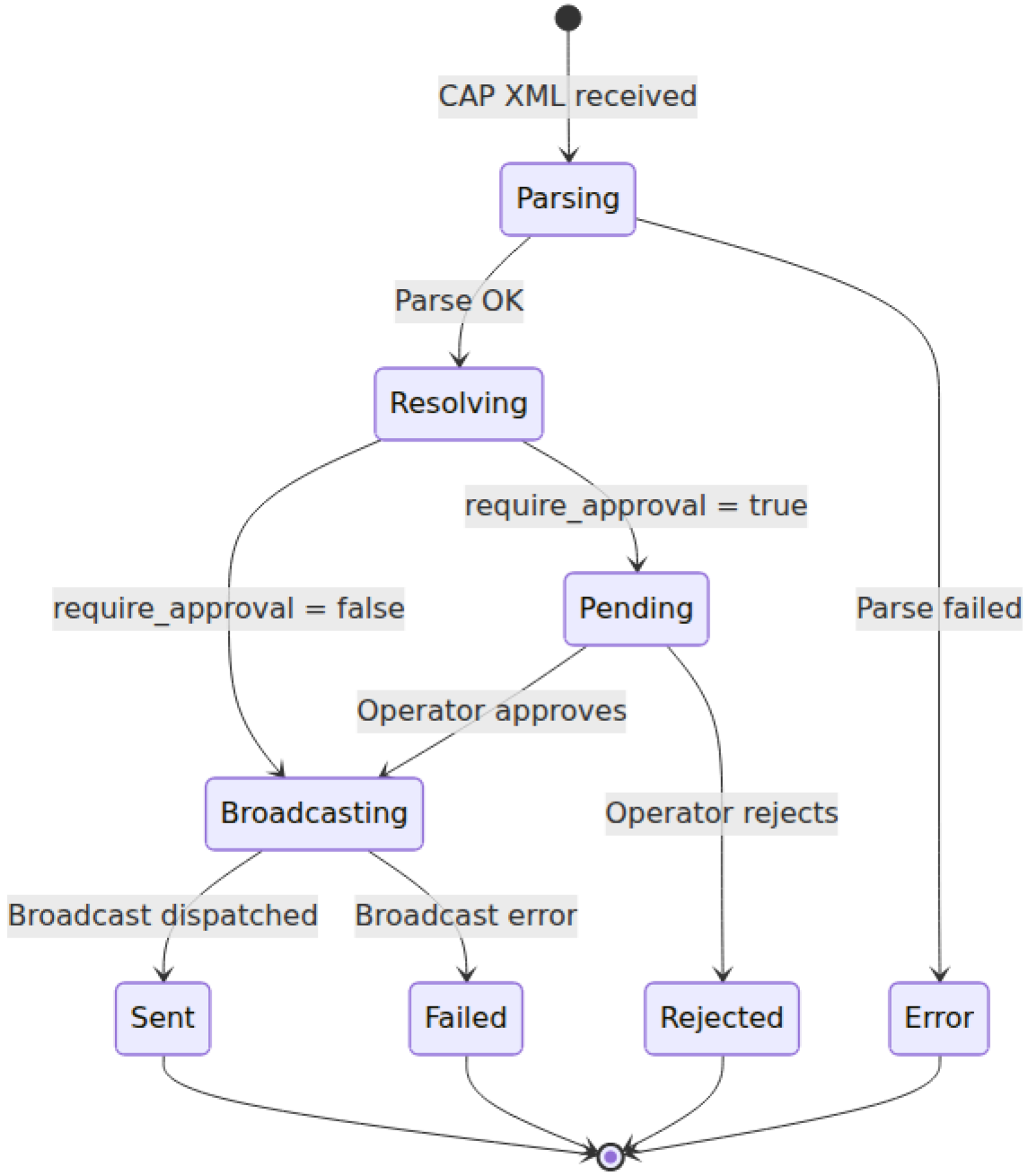
蜂窝 JSON 示例

```json
[
  {
    "cell_id": "eNB-001-cell-01",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "pci": 100,
    "earfcn": 1300,
    "radius": 500,
    "tac": 100,
    "lac": null,
    "rat": "4g"
  },
  {
    "cell_id": "BTS-001-cell-01",
    "latitude": 40.7130,
    "longitude": -74.0065,
    "pci": null,
    "earfcn": null,
    "radius": 2000,
    "tac": null,
    "lac": 5001,
    "rat": "2g"
  }
]
```

多边形解析

当 CAP 警报包含多边形警告区域时，几何引擎确定哪些蜂窝位于每个多边形内。

多边形匹配的工作原理



标准模式（`coverage_aware: false`）：如果蜂窝的中心点（纬度/经度）位于多边形内，则该蜂窝匹配，使用射线投射算法。

覆盖感知模式（`coverage_aware: true`）：如果蜂窝的覆盖区域（建模为具有蜂窝 `radius` 字段的圆）与多边形相交，则该蜂窝匹配。这可以捕捉到中心刚好在多边形外但覆盖范围延伸到多边形内的蜂窝。

CAP 多边形格式

CAP v1.2 将多边形定义为以空格分隔的 `lat,lon` 对。第一个和最后一个点应相同（闭合多边形）：

```
38.47,-120.14 38.34,-119.95 38.52,-119.74 38.62,-119.89
38.47,-120.14
```

CAP XML 格式

解析器根据 OASIS 标准处理 CAP v1.2。提取的关键元素：

警报信封

元素	描述
`<identifier>`	唯一的警报标识符
`<sender>`	警报发起者
`<sent>`	警报发布的时间戳
`<status>`	`Actual`、`Exercise`、`System`、`Test`、`Draft`
`<msgType>`	`Alert`、`Update`、`Cancel`、`Ack`、`Error`
`<scope>`	`Public`、`Restricted`、`Private`

信息块

每个 `<info>` 块包含特定语言/受众的警报内容：

□□	��□
`<category>`	□□□□□□□□□□□□□□□□□□□□□□
`<event>`	□□□□□□□□□□□“□□□□□”□
`<urgency>`	□□□□□□□□□□□□□□□
`<severity>`	□□□□□□□□□□□□□□□
`<certainty>`	□□□□□□□□□□□□□□□□□□
`<headline>`	□□□□□
`<description>`	□□□□□□□□□□□□□□□□□□□
`<instruction>`	□□□□□

CB □□

□□□□□□□□□ `<info>` □□□ `<parameter>` □□□□□

□□□□	□□
`CBMessageIdentifier`	□□□ 16 □ CB □□ ID
`CBRepetitionInterval`	□□□□□□□□□
`CBNumberOfBroadcasts`	□□□□□□

□□□□□□□ CAP XML □□□□□□□□□□□□□□□ ID `0x1112`□□□□□ 30 □□10 □□□□□

□□□

□□ `<info>` □□ `<area>` □□□□□

元素	描述
`<areaDesc>`	可读的区域描述
`<polygon>`	定义警告区域的以空格分隔的 `lat,lon` 坐标对
`<circle>`	中心点和半径（供将来使用）

REST API

POST /api/cap

提交 CAP XML 警报以进行处理。

请求体

```
{
  "xml": "<alert xmlns=\"urn:oasis:names:tc:emergency:cap:1.2\">...</alert>"
}
```

参数	类型	必需	描述
`xml`	string	□	完整的 CAP v1.2 XML 警报文档

响应 **(201)** -- 警报成功提交

```
{
  "status": "ok",
  "data": {
    "id": "a1b2c3d4-e5f6-...",
    "status": "pending",
    "source": "http_post",
    "received_at": "2025-01-15T10:30:00Z",
    "matched_cells": 42,
    "tacs": [100, 101, 102],
    "lacs": [5001, 5002],
    "mcc": "001",
    "mnc": "01",
    "broadcast_params": {
      "message_id": 4370,
      "repetition_period": 30,
      "num_broadcasts": 10,
      "message_text": "Tornado Warning for Springfield County...",
      "event": "Tornado Warning",
      "severity": "Extreme",
      "urgency": "Immediate"
    }
  }
}
```

如果 `require_approval` 为 `true`，则 `status` 字段将为 `"pending"`，如果自动批准，则为 `"sent"`。

错误响应

状态	原因
400	`"xml field is required"`
422	解析错误详情

GET /api/cap

列出所有州的所有警报。

响应 **(200)**

```
{
  "status": "ok",
  "data": {
    "pending": [...],
    "active": [...],
    "history": [...]
  }
}
```

每个数组包含与 POST 响应相同格式的警报对象。

GET /api/cap/:id

通过 ID 获取单个警报。

路径参数

参数	类型	描述
`id`	string	警报 UUID

响应 **(200)**

返回警报对象。

错误响应

状态	原因
404	`"Alert not found: <id>"`

PUT /api/cap/:id

批准或拒绝待处理的警报。

请求体

```
{
  "action": "approve",
  "operator": "operator1"
}
```

参数	类型	必需	描述
`action`	string	□	`"approve"` 或 `"reject"`
`operator`	string	否	审计轨迹的操作员名称。默认为 `"unknown"`。

响应 **(200)**

返回带有新状态（`"sent"`、`"broadcasting"` 或 `"rejected"`）的更新警报对象。

错误响应

状态	原因
400	`"action must be 'approve' or 'reject'"`
404	`"Alert not found: <id>"`

Web 界面：CAP 警报页面

路径：`/cap`
刷新：每 3 秒 + 实时 PubSub 更新

CAP 警报页面提供操作员审批工作流和广播监控界面。

统计卡片

顶部有四个摘要卡片：

卡片	描述
待审批	等待操作员操作的警报数量（当 > 0 时为琥珀色）
活动广播	当前正在广播的警报数量
总处理	历史中的总警报（已发送 $+$ 已拒绝 $+$ 失败）
审批模式	当前模式：“手动”或“自动”

待处理警报面板

仅在 `require_approval` 为 `true` 时可见。显示等待操作员批准的警报。

列	描述
时间	警报接收时间（HH:MM:SS）
事件	警报事件类型（例如，“龙卷风警告”）
严重性	警报严重性级别
蜂窝	通过多边形解析匹配的蜂窝数量
状态	待处理徽章（琥珀色）
操作	预览、批准和拒绝按钮

预览 展开行以显示：

详情	描述
描述	完整的警报描述文本（广播消息内容）
来源	警报接收方式（`http_post` 或 `feed_poll`）
TACs	匹配的用于 4G 广播的跟踪区域代码
LACs	匹配的用于 2G 和 3G 广播的位置区域代码
消息 ID	CB 消息标识符
PLMN	用于广播的 MCC/MNC

批准 通过 CBC 引擎触发立即广播到所有连接的 MMEs（4G）、RNCs（3G）和 BSCs（2G）。

拒绝 将警报移动到历史记录，状态为 `:rejected`。

活动广播面板

显示当前正在广播的警报：

列	描述
事件	警报事件类型
消息 ID	CB 消息标识符
TACs	目标跟踪区域代码
开始时间	广播开始时间
状态	正在广播（蓝色）或已发送（绿色）徽章

警报历史面板

显示已发送、已拒绝和失败的警报（最后 200 条）：

列	描述
时间	发送或接收时间戳
事件	警报事件类型
严重性	警报严重性级别
蜂窝	匹配的蜂窝数量
TACs/LACs	目标 TACs 和 LACs
状态	已发送（绿色）、已拒绝（红色）或失败（红色）徽章

实时更新

该页面订阅 `cap:alerts` PubSub 主题。当新警报到达或警报状态更改时，页面会自动刷新。当收到新的待处理警报时，会出现吐司通知（“收到新的 CAP 警报 - 等待批准”）。

警报状态持久化

警报状态持久化到 `priv/cap_alerts.json` 中，格式为 JSON，遵循与 CBC 引擎的 `priv/active_broadcasts.json` 相同的模式。在启动时，警报管理器重新加载此文件以恢复待处理和活动警报。

历史记录限制为 **200** 条 以防止无限增长。

Atom 订阅轮询

订阅轮询器支持国家天气服务和警报机构使用的标准 CAP Atom 订阅。订阅格式遵循 RFC 4287（Atom 订阅格式），并包含 CAP 警报条目。

预期的订阅格式

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <entry>
    <id>urn:oid:2.49.0.1.840.0.abc123</id>
    <title>Tornado Warning</title>
    <content type="text/xml">
      <alert xmlns="urn:oasis:names:tc:emergency:cap:1.2">
        <!-- □整的 CAP 警报 XML -->
      </alert>
    </content>
  </entry>
</feed>
```

去重

轮询器跟踪它已见过的每个条目的 `<id>`。具有先前见过的 ID 的条目将被跳过。去重集合保存在内存中，并在应用程序重启时重置。

错误处理

订阅获取失败（网络错误、HTTP 错误、XML 解析错误）被记录为警告，并不会影响后续的轮询周期。轮询器将继续按照其配置的间隔进行，无论单个获取结果如何。

监督

CAP 子系统向 OmniLCS 监督器添加两个进程：

进程	描述
`OmniLcs.Cap.AlertManager`	警报生命周期 GenServer。管理待处理/活动/历史状态、多边形解析和广播调度。
`OmniLcs.Cap.FeedPoller`	Atom 订阅轮询 GenServer。为每个配置的订阅 URL 调度定期获取。

两者自动启动，并与现有的 CBC 和 E-SMLC 进程一起使用 `:one_for_one` 策略进行监���。

故障排除

没有蜂窝与多边形匹配

症状：警报被接受，但 `matched_cells` 为 0，TAC/LAC 列表为空。

可能的原因：

- 蜂窝数据库中没有地理位置在警告多边形内的蜂窝
- 蜂窝存在于多边形区域内，但缺少 `tac` 和 `lac` 字段
- CAP XML 中的多边形坐标顺序或格式错误

解决方案：

1. 通过 `GET /api/cells` 验证警告区域中的蜂窝□否设置了 `tac` 和/或 `lac` 值
2. 检查蜂窝位置（纬度/经度）□否正确
3. 如果蜂窝靠近多边形边缘，请尝试启用 `coverage_aware: true`
4. 验证 CAP 多边形坐标与地图的匹配

警报卡在待处理状态

症状：警报显示为待处理，但操作员无法批准。

可能的原因：

- `require_approval` 为 `true`，且没有操作员批准警报
- LiveView 页面未连接（检查浏览器 WebSocket 连接）

解决方案：

1. 导航到控制面板中的 `/cap` 并单击批准
2. 或者，使用 REST API：`PUT /api/cap/<id>`，内容为 `{"action": "approve", "operator": "operator1"}`

订阅轮询器未获取警报

症状：配置的订阅 URL 有新警报，但它们未出现在 OmniLCS 中。

可能的原因：

- 订阅 URL 不正确或无法访问
- 订阅格式不□嵌入 CAP 的标准 Atom
- TLS 证书问题
- 警报条目不包含嵌入的 `<content>` 和 `<alert>` XML

解决方案：

1. 检查日志中□否有 `CAP FeedPoller: Failed to fetch` 消息
2. 验证订阅 URL □否返回有效的 Atom XML
3. 确认订阅包含 `<entry>` 元素，并且 `<content>` 中包含 `<alert>` XML
4. 对于 TLS 问题，检查订阅 URL □否可以从 OmniLCS 服务器访问

广播已发送但 **UE** 未接收

症状：警报状态显示为“已发送”，但没有 UE 接收到广播消息。

可能的原因：

- TAC/LAC 值与实际网络配置不匹配
- MME 或 BSC 拒绝了广播（检查 CBC 4G / CBC 页面以获取响应详情）
- 消息编码问题

解决方案：

1. 验证 TAC 值与 MME 的跟踪区域配置□否匹配
2. 检查 CBC 4G 页面以获取写替换警告响应状态
3. 检查 CBC 页面以获取写替换完成/失败消息
4. 在警报历史中查看广播结果（展开警报行）

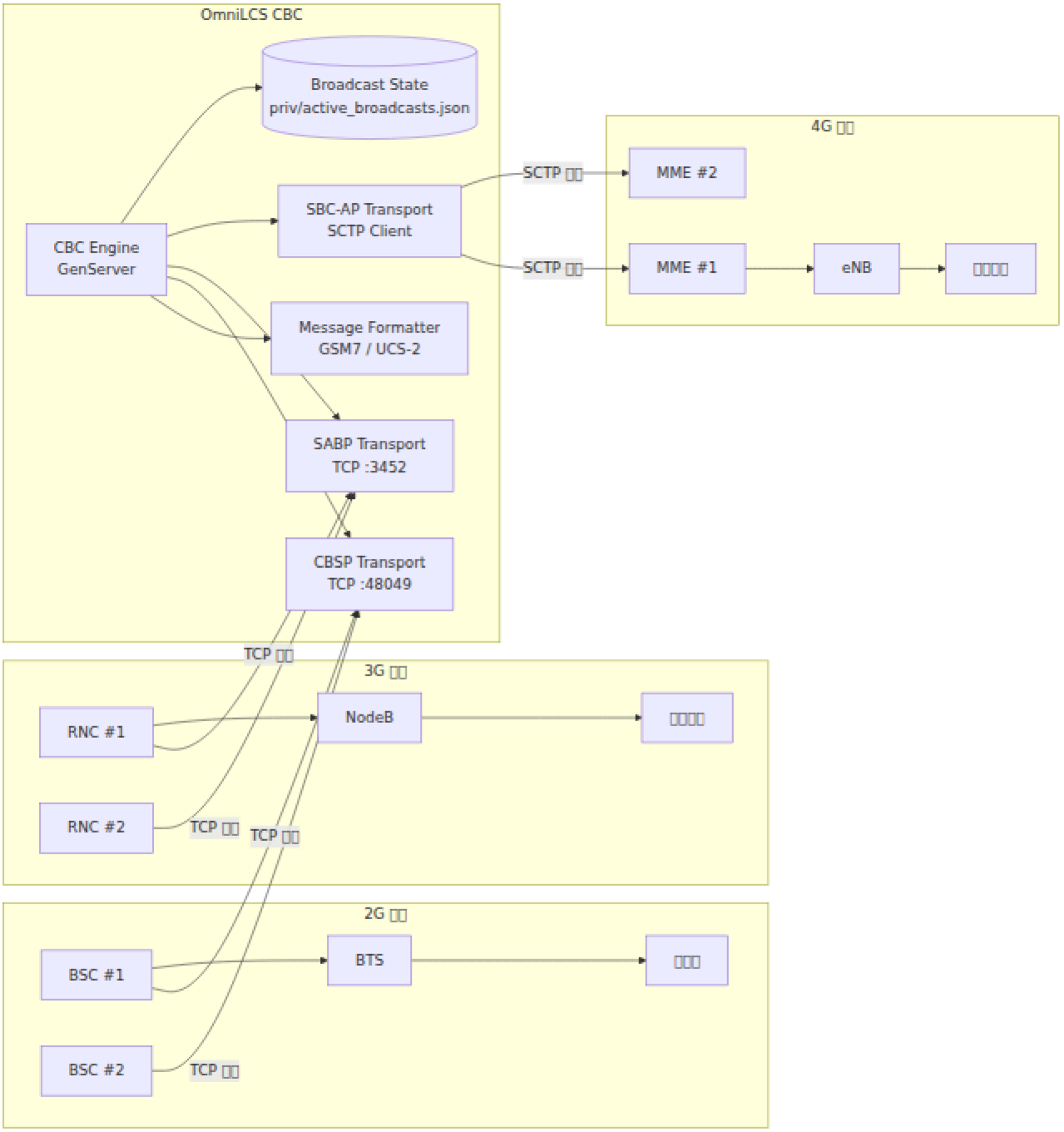
3GPP 和标准参考

规范	标题
OASIS CAP v1.2	通用警报协议版本 1.2
RFC 4287	Atom 订阅格式
TS 29.168	蜂窝广播中心与演进分组核心（SBC-AP）接口
TS 48.049	蜂窝广播中心协议（CBSP）
TS 23.041	蜂窝广播服务（CBS）的技术实现
TS 23.038	字母表和特定语言的信息

短信广播□作指南

OmniLCS 实现了一个短信广播中心 (CBC)，支持通过 CBSP 的 2G 网络、通过 SABP 的 3G UTRAN 网络和通过 SBC-AP 的 4G LTE 网络。CBC 可以同时在所有连接的无线接入网络元素中发送、更新和停止广播消息。

架构



2G CBSP □作

连接模型

CBSP 根据 3GPP TS 48.049 使用 TCP 传输。CBC 监听 TCP 端口 **48049**（IANA 注册），BSC 建立入站连接。

- 每个接受的连接在 DynamicSupervisor 下生成一个连接处理程序
- 连接在 `:cbsp_connections` ETS 表中进行跟踪
- 连接状态通过 PubSub 广播到 LiveView UI

保持连接

CBC 实现了主动保持连接:

参数	默认值	描述
保持连接间隔	30 秒	CBC 向每个对等体发送 KEEP-ALIVE 的频率
保持连接超时	10 秒	在标记为不健康之前等待 KEEP-ALIVE COMPLETE 的时间

CBC 同时发送主动的 KEEP-ALIVE 消息，并对 BSC 发起的 KEEP-ALIVE 消息以 KEEP-ALIVE COMPLETE 作出响应。

CBSP 消息类型

消息	方向	代码	描述
WRITE-REPLACE	CBC -> BSC	0x01	发送或更新广播消息
WRITE-REPLACE COMPLETE	BSC -> CBC	0x02	BSC 接受广播
WRITE-REPLACE FAILURE	BSC -> CBC	0x03	BSC 拒绝广播
KILL	CBC -> BSC	0x04	停止广播
KILL COMPLETE	BSC -> CBC	0x05	广播已停止
KILL FAILURE	BSC -> CBC	0x06	停止广播失败
LOAD-QUERY	CBC -> BSC	0x07	查询无线资源加载
LOAD-QUERY COMPLETE	BSC -> CBC	0x08	返回加载信息
LOAD-QUERY FAILURE	BSC -> CBC	0x09	加载查询失败
STATUS-QUERY	CBC -> BSC	0x0A	查询广播交付状态
STATUS-QUERY COMPLETE	BSC -> CBC	0x0B	返回状态信息
STATUS-QUERY FAILURE	BSC -> CBC	0x0C	状态查询失败
RESET	CBC -> BSC	0x10	在 BSC 上重置广播状态
RESET COMPLETE	BSC -> CBC	0x11	重置已确认
RESET FAILURE	BSC -> CBC	0x12	重置失败
RESTART	BSC -> CBC	0x13	BSC 重启（信息性）
FAILURE	BSC -> CBC	0x14	BSC 故障指示

消息	方向	代码	描述
ERROR INDICATION	任意	0x15	协议错误
KEEP-ALIVE	任意	0x16	连接监督
KEEP-ALIVE COMPLETE	任意	0x17	保持连接响应

CBSP 消息封装

每个 CBSP 消息在传输中为：

```
+----------+----------+------+
| □度      | 类型     | IEs  |
| (3 字节) | (1 字节) | ...  |
+----------+----------+------+
```

3 字节的长度字段包括类型字节和所有 IEs（���包括 3 个长度���节本身）。

小区列表格式

小区列表 IE 确定广播目标的小区。支持的格式：

区分符	值	格式	描述
CGI	0x00	MCC+MNC+LAC+CI	完整小区全球标识
LAC+CI	0x01	PLMN+LAC+CI	位置区域代码 + 小区标识
CI	0x02	仅 CI	仅小区标识
LAI	0x04	MCC+MNC+LAC	位置区域标识
LAC	0x05	仅 LAC	位置区域代码
所有在 BSC	0x06	(无小区)	BSC 管理的所有小区

通道指示符

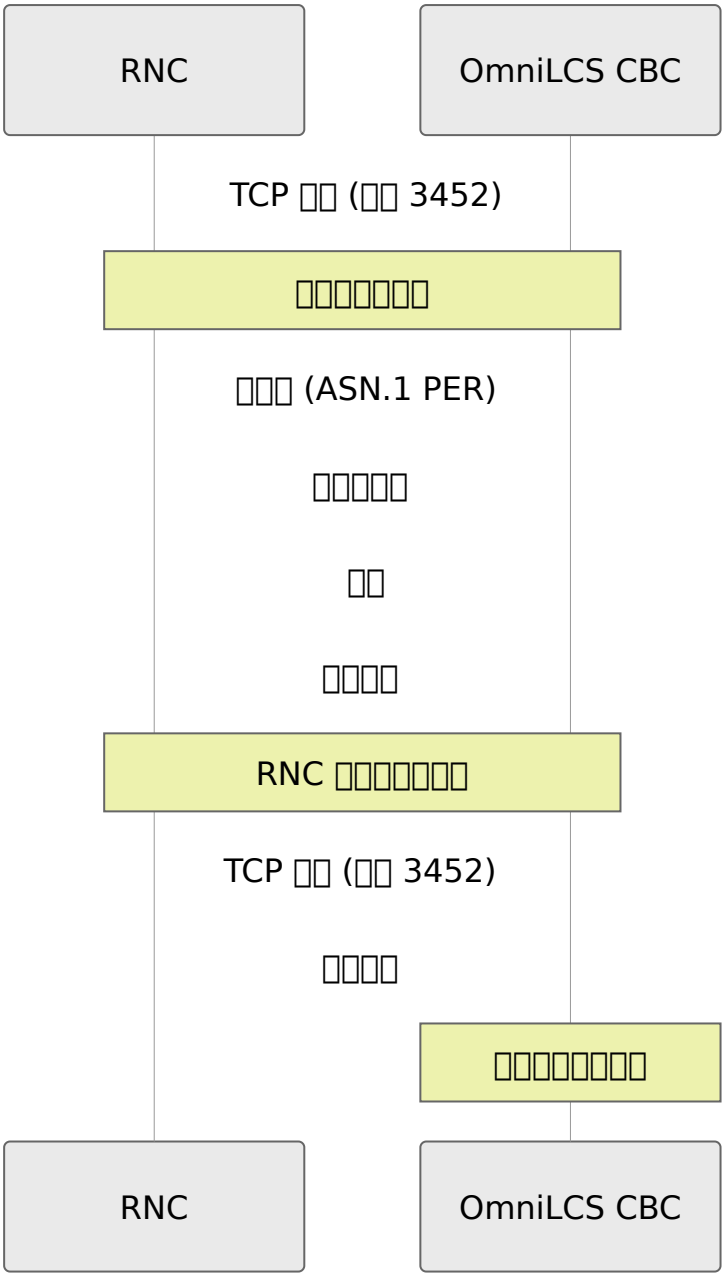
值	名称	描述
0	基本 CBCH	主小区广播信道
1	扩展 CBCH	扩展小区广播信道（更高容量）

3G SABP 作

服务区域广播协议 (SABP) 提供 CBC 与 3G UTRAN 网络中的 RNC 之间的 Iu-BC 接口，符合 3GPP TS 25.419。SABP 使用 ASN.1 PER 对 TCP 进行对齐编码。

连接模型

SABP 根据 TS 25.414 §7.1.3.3 使用 TCP 传输。SABP 的 IANA 注册端口为 **3452**。根据 TS 25.419 §5，CBC 发起正常操作的连接（写替换、杀死等），而 RNC 仅在故障/重启指示时发起连接。双方在建立新连接时使用目标端口 3452。



- 每个接受的 TCP 连接在 SABP 连接 DynamicSupervisor 下生成一个连接处理程序
- 连接在 `:sabp_connections` ETS 表中进行跟踪
- 连接状态变化通过 PubSub 广播到 LiveView UI
- 由于 SABP 没有协议级的保持连接机制，因此启用了 TCP 保持连接套接字选项

SABP 消息封装

SABP 消息通过 TCP 封装，带有 4 字节的大端长度前缀：

```
+--------------------+----------------------------+
| □度 (4 字节)       | ASN.1 PER 编码的 PDU       |
| 大端 uint32        | ...                        |
+--------------------+----------------------------+
```

长度字段指定后续 ASN.1 PER 编码有效负载的字节数。多个消息可以在单个 TCP 段中到达；连接处理程序会根据需要缓冲和重新组装。

SABP 程序

程序	代码	类别	方向	描述
写替换	0	类别 1	CBC -> RNC	发送或更新广播消息
杀死	1	类别 1	CBC -> RNC	停止活动广播
加载状态查询	2	类别 1	CBC -> RNC	查询无线资源加载
消息状态查询	3	类别 1	CBC -> RNC	查询广播交付状态
重置	4	类别 1	CBC -> RNC	在 RNC 上重置广播状态
重启指示	5	类别 2	RNC -> CBC	RNC 重启，重新发送活动广播
故障指示	6	类别 2	RNC -> CBC	RNC 故障指示
错误指示	7	类别 2	任意	报告协议错误

SABP 信息元素

SABP 协议使用 ASN.1 定义的信息元素 (IEs)。在写替换中使用的关键 IEs:

IE	ID	关键性	类型	描述
广播消息内容	0	拒绝	二进制	编码的 CBS 消息内容
类别	1	忽略	枚举	消息优先级（高、正常、背景、默认）
原因	2	忽略	整数	故障/错误指示的原因值
数据编码方案	4	忽略	8 位	消息内容的编码方式（例如，0x0F 表示 GSM 7 位）
消息标识符	6	拒绝	16 位	CB 消息标识符
新序列号	7	拒绝	16 位	新/更新广播的序列号
请求的广播数量	9	拒绝	整数	总广播数量 (0..65535)
旧序列号	10	忽略	16 位	被替换消息的序列号（可选）
重复周期	13	拒绝	整数	重复之间的秒数 (1..4096)
服务区域列表	15	拒绝	列表	目标服务区域（SAI 列表）

服务区域标识符 (SAI)

SABP 将广播定向到特定的服务区域，服务区域通过 SAI 识别。每个 SAI 由以下组成：

字段	大小	描述
PLMN 标识	3 字节	根据 3GPP TS 24.008 编码的 BCD 编码 MCC+MNC
LAC	2 字节	位置区域代码
SAC	2 字节	服务区域代码

广播类别

可选的类别 IE 设置广播消息的优先级：

类别	描述
高优先级	最高优先级，抢占其他广播
正常优先级	标准优先级
背景优先级	低优先级，不抢占
默认优先级	未指定类别时的默认值

重启指示和故障指示处理

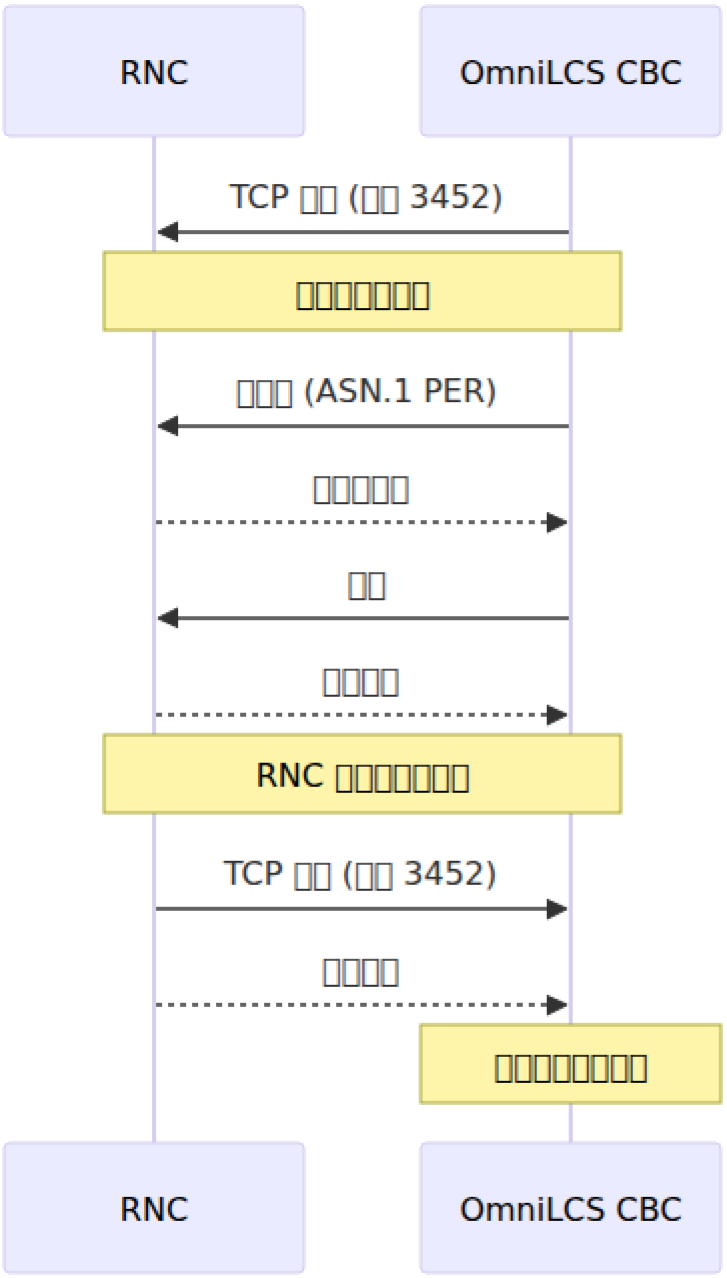
当 CBC 接收到来自 RNC 的 重启指示 时，RNC 已重启并丢失其广播状态。CBC 应重新发送所有活动广播到该 RNC。

当 CBC 接收到 故障指示 时，RNC 报告某些服务区域失去了广播能力。这将被记录并作为遥测事件发出。

4G SBC-AP 作

连接模型

根据 3GPP TS 29.168 第 4.3 节，**CBC** 向每个 **MME** 发起 **SCTP** 关联。MME 监听端口 **29168**（IANA 注册）。SBC-AP 的 SCTP 负载协议标识符（PPID）为 **24**。



该传���实现了自动重连，采用指数退避策略：

□□	□	□□
□□□□□□	5 □	□□□□□□□□□□□□□□□
□□□□□□	60 □	□□□□□□
SCTP □□□□	10 □	□□□□□□
SCTP □□□□□□□□	7	□□□□□□□□□
SCTP RTO □□	30 □	□□□□□□
SCTP RTO □□	1 □	□□□□□□

SBC-AP □□

□□	□□	□□	□□	□□
□□□□□	0	□□ 1	CBC -> MME	□□□□□□□□□
□□□	1	□□ 1	CBC -> MME	□□□□□
□□□	2	□□ 2	□□	□□□□□
PWS □□□	3	□□ 2	MME -> CBC	MME □□□□□□□□□□□
PWS □□□	4	□□ 2	MME -> CBC	eNB ��□□□□□□□

□□□□□□□ IEs

IE	ID	□□□	□□	□□
□□□□□	5	□□	16 □	CB □□□□□□□□□□0x1112 □□ CMAS□
□□□	11	□□	16 □	□□□□□□□□□□□□□□□□□□□□□□□□
TAIs □□	14	□□	□□	□□□□□□□□□□
□□□□□□□	15	□□	□□	□□□□□TAI □□□□□□□□□□□□□□
□□□□	10	□□	□□	□□□□□□□ (0..4096)
□□□□□□□	7	□□	□□	□□□□□ (0..65535)
□□□□	18	□□	2 □□	ETWS □□□□ + □□□□
□□□□□□	3	□□	8 □	□□□□□□□□
□□□□□□	16	□□	□□□	□□□ CBS □□
OMC ID	19	□□	□□□	□□□□□□□
□□□□□□□□□	20	□□	□□□	□□□□□

停止警告请求 IEs

IE	ID	关键性	描述
消息标识符	5	拒绝	要停止的 CB 消息标识符
序列号	11	拒绝	要停止的广播的序列号
TAIs 列表	14	拒绝	（可选）限制停止到特定 TAIs
警告区域列表	15	忽略	（可选）限制停止到特定区域

PWS 重启指示处理

当 CBC 接收到来自 MME 的 PWS 重启指示时，它会重新发送所有活动广播（状态为 `:sent` 或 `:acknowledged`）到该 MME。这确保了 MME 重启后的广播连续性。

PWS 故障指示处理

当 CBC 接收到 PWS 故障指示时，它会将受影响的广播标记为 `:pws_failure` 响应，以便报告 MME 关联。这将被记录为警告。

消息格式

GSM 7 位编码

CBS 消息的默认编码。每个 CBS 页面最多可容纳 **93** 个七位字节，打包为 **82** 字节。

GSM 7 位默认字母表映射标准拉丁字符以及希腊字母和特殊符号。基本字母表中没有的字符使用扩展表（转义序列 0x1B + 扩展代码），消耗两个七位字节。

警告消息内容的二进制结构：

```
+----------+-----------------------------------------------+
| 页数     | 页 1 (82 字节打包 + 1 字节□度)                  |
| (1 字节) | 页 2 ...                                       |
+----------+-----------------------------------------------+
```

UCS-2 编码

用于非拉丁文字（CJK、阿拉伯语、泰语等）。每个 CBS 页面最多可容纳 **40** 个 **UCS-2** 字符（80 字节的 UTF-16BE 数据），填充到 82 字节。

数据编码方案 (DCS)

编码	**DCS** 值	描述
GSM 7 位	0x0F	GSM 7 位默认字母表，语言未指定
UCS-2	0x48	UCS-2 (UTF-16BE)，语言未指定

DCS 可以在每个广播中显式设置，或者根据编码参数自动推导。

序列号结构

根据 3GPP TS 23.041 第 9.4.1.2.1 节，16 位序列号具有内部结构：

```
+----+------------+--------+
| GS | 消息代码   | 更新   |
| 2b | 10 位      | 4 位   |
+----+------------+--------+
```

字段	位数	描述
地理范围（GS）	15-14	0 = 小区范围立即，1 = PLMN 范围，2 = LA/TA 范围，3 = 小区范围
消息代码	13-4	在其范围内识别广播（0..1023）
更新编号	3-0	消息更新时递增（0..15）

`MessageFormatter.build_serial_number/3` 辅助函数根据这些组件构建序列号。

警告类型（ETWS）

警告类型 IE 每个 3GPP TS 23.041 第 9.3.24 节为 2 字节:

警告类型	八位组 **1** 值	描述
地震	0x00	地震警告
海啸	0x01	海啸警告
地震 + 海啸	0x02	组合地震和海啸
测试	0x03	测试警告
其他	0x04	其他紧急情况

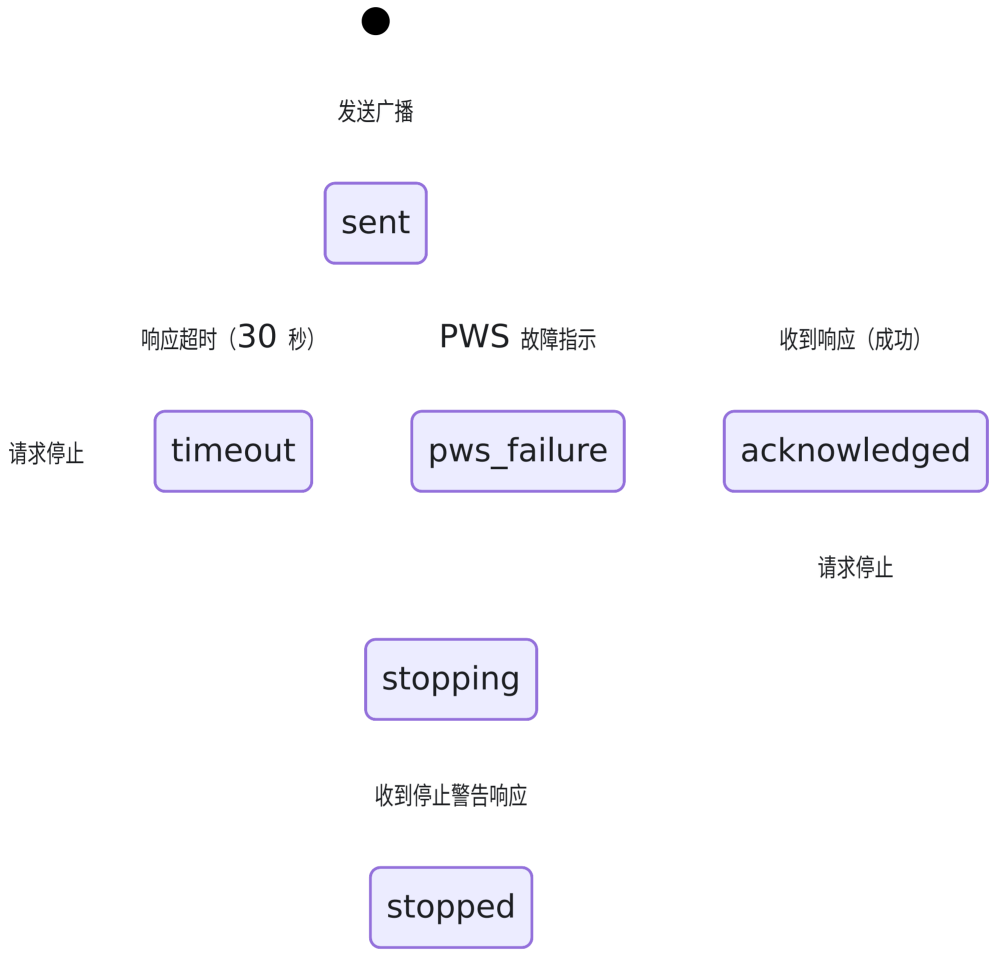
八位组 2 激活标志（默认 0xC0）：

位	值	描述
第 8 位	0x80	紧急用户警报（激活警报音/振动）
第 7 位	0x40	弹出（自动在屏幕上显示消息）

默认情况下，两个位都被设置（0xC0），以确保最大可见性。

广播状态管理

状态生命周期



持久性

2G、3G 和 4G 的活动广播以 JSON 格式持久化到 `priv/active_broadcasts.json`。在启动时，Engine 重新加载此文件以恢复广播状态。这允许 CBC 在应用程序重启后重新发送活动警告。持久性文件存储三个独立的广播映射：`active_broadcasts_4g`、`active_broadcasts_3g` 和 `active_broadcasts_2g3g`。

响应跟踪

对于 4G 广播，每个 `send_broadcast_4g` 启动一个 30 秒的响应超时计时器。来自 MMEs 的响应通过消息标识符和序列号进行关联。当所有预期的 MME 对等体响应时，计时器被取消。超时的对等体被记录为 `:timeout` 原因。

活动广播视图

活动广播页面提供了 2G、3G 和 4G 网络中所有正在进行的广播的统一视图。

广播历史

Engine 维护最近 100 个 4G 广播的滚动历史，以供 LiveView UI 和 API 查询。

故障排除

没有连接的 MMEs (SBC-AP)

1. 验证 `mme_peers` 配置包含正确的 IP 地址和端口
2. 检查 `local_ip` 是否可以从 MME 网络访问
3. 在日志中查找 SCTP 连接错误：`SBC-AP: 无法连接到 MME`
4. 验证 SCTP 是否未被防火墙阻塞（协议 132）
5. 确认 MME 正在监听端口 29168

没有连接的 BSCs (CBSP)

1. 验证 `listen_ip` 和 `listen_port` 配置在 `:cbsp` 下

2. 检查 TCP 端口 48049 □否未被防火墙阻塞
3. 查找接受错误：`CBSP 接受失败`
4. 确认 BSC 已配置 CBC 的 IP 地址和端口

没有连接的 RNCs (SABP)

1. 验证 `listen_ip` 和 `listen_port` 配置在 `:sabp` 下
2. 检查 TCP 端口 3452 □否未被防火墙阻塞
3. 查找接受错误：`SABP 接受失败`
4. 确认 RNC 已配置 CBC 的 IP 地址和端口
5. 检查日志中□否有 `SABP 连接被对等体关闭` 或 `SABP TCP 错误` 消息

广播未交付

1. 检查活动广播表以获取响应状态
2. 在 CBSP/SABP 消息日志中查找 WRITE-REPLACE FAILURE 或 KILL FAILURE 消息
3. 验证小区列表格式□否与 BSC/RNC/MME 预期的匹配
4. 对于 4G，检查 TAC 值□否对目标 MME 有效
5. 对于 3G，检查服务区域标识符 (SAIs) □否对目标 RNC 有效
6. 检查广播历史以获取超时条目

编码问题

1. 对于非拉丁文本，确保选择了 `:ucs2` 编码
2. 验证 DCS □否与编码匹配（0x0F 表示 GSM 7 位，0x48 表示 UCS-2）
3. GSM 7 位只能表示 GSM 默认字母表；不支持的字符将被替换为 `?`

3GPP 参考

规范	标题
TS 25.419	UTRAN Iu-BC 接口：服务区域广播协议 (SABP)
TS 29.168	短信广播中心与演进分组核心的接口 (SBC-AP)
TS 48.049	短信广播中心协议 (CBSP)
TS 23.041	短信广播服务 (CBS) 的技术实现
TS 23.038	字母表和特定语言信息 (DCS, GSM 7 位)
TS 24.008	移动无线接口层 3 规范 (PLMN 标识编码)

OmniLCS 配置参考

所有配置都在 `config/config.exs` 中定义。本文档涵盖每个配置部分和参数。

控制面板（LiveView UI）

控制面板提供基于网页的管理界面，通过 HTTPS 在 443 端口提供服务。

```
config :control_panel,
  parent_application: :omnilcs,
  parent_application_version: "1.0.0",
  parent_application_readable_name: "OmniLCS",
  home_page: ControlPanelWeb.ApplicationLive,
  use_builtin_pages: [...],
  use_additional_pages: [...],
  page_order: ["/dashboard", "/location", "/cells", "/diameter",
               "/cbc", "/cbc4g", "/application", "/configuration",
"/log"],
  licensee_name: "Omnitouch"
```

参数	类型	描述
`parent_application`	atom	OTP 应用程序名称
`parent_application_version`	string	在 UI 中显示的版本
`parent_application_readable_name`	string	人类可读的产品名称
`home_page`	module	默认登录页面模块
`use_builtin_pages`	list	内置页面（应用程序、配置、日志）
`use_additional_pages`	list	自定义 OmniLCS 页面（仪表板、位置、单元、直径、CBC 2G、CBC 3G、CBC 4G、CAP 警报）
`page_order`	list	导航栏中的选项卡顺序
`licensee_name`	string	页脚中显示的被许可方名称

控制面板端点

```
config :control_panel, ControlPanelWeb.Endpoint,
  server: true,
  https: [
    port: 443,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ],
  secret_key_base: "...",
  check_origin: false,
  pubsub_server: ControlPanel.PubSub,
  live_view: [signing_salt: "LcsLvSlt"]
```

参数	类型	默认	
`server`	boolean	`true`	启用 HT
`https.port`	integer	`443`	HTTPS
`https.keyfile`	string	`"priv/cert/omnitouch.pem"`	TLS 私
`https.certfile`	string	`"priv/cert/omnitouch.crt"`	TLS 证
`secret_key_base`	string	--	Phoen (使用 `mi` `phx.ge` 生成)
`check_origin`	boolean	`false`	否验证 WebSo
`pubsub_server`	atom	`ControlPanel.PubSub`	用于 Live PubSu
`live_view.signing_salt`	string	`"LcsLvSlt"`	LiveVi

REST API

REST API 通过 api_ex 框架在 8443 端口上通过 HTTPS 提供服务。

```
config :api_ex,
  api: %{
    port: 8443,
    listen_ip: "0.0.0.0",
    product_name: "OmniLCS",
    title: "API - OmniLCS",
    hostname: "localhost",
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem",
    routes: [...]
  }
```

参数	类型	默认	描述
`port`	integer	`8443`	API 监听端口
`listen_ip`	string	`"0.0.0.0"`	API 绑定地址
`product_name`	string	`"OmniLCS"`	OpenAPI 规范中的产品名称
`title`	string	`"API - OmniLCS"`	Swagger UI 中的 API 标题
`hostname`	string	`"localhost"`	OpenAPI 规范 URL 的主机名
`enable_tls`	boolean	`true`	启用 HTTPS
`tls_cert_path`	string	`"priv/cert/omnitouch.crt"`	TLS 证书路径
`tls_key_path`	string	`"priv/cert/omnitouch.pem"`	TLS 私钥路径
`routes`	list	--	路由定义列表（路径、模块、操作）

注册的 **API** 路由

路径	控制器	□作
`/status`	`OmniLcs.Api.StatusController`	`index`
`/location`	`OmniLcs.Api.LocationController`	`index`, `create`, `show`
`/cells`	`OmniLcs.Api.CellController`	`index`, `create`, `show`, `update`, `delete`
`/cap`	`OmniLcs.Api.CapController`	`index`, `create`, `show`, `update`

E-SMLC 设置

一般的 E-SMLC 配置。

```
config :omnilcs,
  esmlc_name: "OmniLCS",
  cell_database_path: "priv/cells.json"
```

参数	类型	默认	描述
`esmlc_name`	string	`"OmniLCS"`	E-SMLC 实例名称（在状态 API 中使用）
`cell_database_path`	string	`"priv/cells.json"`	用于导入的单元数据库 JSON 文件路径

CBSP (2G 单元广播)

CBSP TCP 监听器的配置。BSC 通过此端口连接到 CBC。

```
config :omnilcs, :cbsp,
  listen_ip: "0.0.0.0",
  listen_port: 48049
```

参数	类型	默认	描述
`listen_ip`	string	`"0.0.0.0"`	绑定 CBSP TCP 监听器的 IP 地址
`listen_port`	integer	`48049`	CBSP 连接的 TCP 端口 (IANA 注册的 CBSP 端口)

SABP (3G 单元广播)

SABP TCP 监听器的配置。RNC 通过 Iu-BC 接口连接到此端口的 CBC，符合 3GPP TS 25.419。

```
config :omnilcs, :sabp,
  listen_ip: "0.0.0.0",
  listen_port: 3452
```

参数	类型	默认	描述
`listen_ip`	string	`"0.0.0.0"`	绑定 SABP TCP 监听器的 IP 地址
`listen_port`	integer	`3452`	来自 RNC 的 SABP 连接的 TCP 端口 (IANA 注册的，符合 TS 25.414 §7.1.3.3)

SBC-AP (4G 单元广播)

SBC-AP SCTP 连接的配置。根据 3GPP TS 29.168，CBC 向每个 MME 对等体发起 SCTP 关联。

```
config :omnilcs, :sbcap,
  local_ip: "10.5.198.200",
  mme_peers: [
    %{host: "mme01", ip: "10.179.2.100", port: 29168},
    %{host: "mme02", ip: "10.179.2.101", port: 29168}
  ]
```

参数	类型	默认	描述
`local_ip`	string	`"0.0.0.0"`	绑定 SCTP 套接字的本地 IP 地址
`mme_peers`	list	`[]`	MME 对等体配置列表

MME 对等体配置

`mme_peers` 中的每个条目都□一个包含以下字段的映射：

字段	类型	必需	默认	描述
`host`	string	否	`ip` 的值	人类可读的 MME 主机名（用于日志和 UI）
`ip`	string	□	--	MME IP 地址
`port`	integer	否	`29168`	MME SBC-AP SCTP 端口（IANA 注册）

SBC-AP 传输使用 SCTP 负载协议标识符（PPID）24。在连接失败时，它会以指数退避的方式重试，从 5 秒开始，最长可达 60 秒。

InfluxDB

单元位置数据定期从 InfluxDB 实例同步。

```
config :omnilcs, OmniLcs.InfluxDb,
  database: "nokia-monitor",
  host: "172.19.3.68",
  port: 8086,
  auth: [method: :basic, username: "monitor", password: "..."],
  http_opts: [recv_timeout: 30_000],
  pool: [max_overflow: 10, size: 5]
```

参数	类型	默认	描述
`database`	string	--	InfluxDB 数据库名称
`host`	string	--	InfluxDB 服务器主机名或 IP
`port`	integer	`8086`	InfluxDB HTTP API 端口
`auth.method`	atom	`:basic`	认证方法
`auth.username`	string	--	InfluxDB 用户名
`auth.password`	string	--	InfluxDB 密码
`http_opts.recv_timeout`	integer	`30000`	HTTP 接收超时（毫秒）
`pool.size`	integer	`5`	连接池大小
`pool.max_overflow`	integer	`10`	最大池溢出连接

单元同步每 5 分钟自动运行，启动后初始延迟为 10 秒。也可以通过 REST API 或控制面板手动触发。

SLs 接口 (LCS-AP 通过 SCTP)

SLs 接口通过 SCTP 将 E-SMLC 连接到 MME，符合 3GPP TS 29.171。OmniLCS 在 9082 端口上向每个配置的 MME 对等体发起 SCTP 关联，PPID 为 29。

```
config :omnilcs, :sls,
  local_ip: "10.5.198.200",
  mme_peers: [
    %{host: "mme01", ip: "10.179.1.15", port: 9082}
  ]
```

参数	类型	默认	描述
`local_ip`	string	`"0.0.0.0"`	绑定 SCTP 套接字的本地 IP 地址
`mme_peers`	list	`[]`	MME 对等体配置列表

MME 对等体配置 (SLs)

`mme_peers` 中的每个条目都□一个包含以下字段的映射：

字段	类型	必需	默认	描述
`host`	string	否	`ip` 的值	人类可读的 MME 主机名（用于日志和 UI）
`ip`	string	□	--	MME IP 地址
`port`	integer	否	`9082`	MME LCS-AP SCTP 端口（IANA 注册的 LCS-AP）

SLs 传输使用 SCTP 负载协议标识符 (PPID) 29。在连接失败时，它会以指数退避的方式重试，从 5 秒开始，最长可达 60 秒。

SCTP 调优参数：

参数	值	描述
心跳间隔	10 秒	对等地址心跳
路径最大重传次数	7	在声明路径失败之前
RTO 最大值	30 秒	最大重传超时
RTO 最小值	1 秒	最小重传超时
SACK 延迟	200 毫秒	选择性确认延迟

Diameter (SLg 和其他接口)

Diameter 配置控制 OmniLCS（作为 GMLC）与 MME 之间的 SLg 接口，通过 DRA 路由。这与使用原生 LCS-AP 通过 SCTP 的 SLs 接口分开的。

```
config :diameter_ex,
  diameter: %{
    service_name: :omnitouch_esmlc,
    listen_ip: "10.5.198.200",
    listen_port: 3868,
    host: "amanaki",
    realm: "epc.mnc380.mcc313.3gppnetwork.org",
    product_name: "OmniLCS",
    request_timeout: 5000,
    peer_selection_algorithm: :random,
    allow_undefined_peers_to_connect: true,
    log_unauthorized_peer_connection_attempts: true,
    control_module: OmniLcs.Control.Diameter,
    vendor_id: 10415,
    supported_vendor_ids: [5535, 10415],
    applications: [...],
    peers: [...]
  }
```

服务参数

参数	类型	默认
`service_name`	atom	`:omnitouch_esml`
`listen_ip`	string	--
`listen_port`	integer	`3868`
`host`	string	--
`realm`	string	--
`product_name`	string	`"OmniLCS"`
`request_timeout`	integer	`5000`
`peer_selection_algorithm`	atom	`:random`

□□	□□	□□
`allow_undefined_peers_to_connect`	boolean	`true`
`log_unauthorized_peer_connection_attempts`	boolean	`true`
`control_module`	module	`OmniLcs.Control`
`processor_module`	module	`DiameterEx.Proc`
`vendor_id`	integer	`10415`
`supported_vendor_ids`	list	`[5535, 10415]`

Diameter □□□□

```
applications: [
  %{
    application_name: :slg,
    application_dictionary: :diameter_gen_3gpp_slg,
    vendor_specific_application_ids: [
      %{vendor_id: 10415, auth_application_id: 16_777_264,
acct_application_id: nil}
    ]
  }
]
```

字段	类型	描述
`application_name`	atom	应用程序标识符 (`:slg` 用于 GMLC 到 MME 接口)
`application_dictionary`	atom	Erlang Diameter 字典模块
`vendor_specific_application_ids`	list	Vendor-Specific-Application-Id AVP 值

SLg 应用程序使用应用程序 ID **16777264** 和 3GPP Vendor-Id **10415**。请注意，E-SMLC 到 MME 接口 (SLs) 使用原生 LCS-AP 通过 SCTP，而不是 Diameter。

Diameter 对等体

```
peers: [
  %{
    host: "omni-nick2-dra01.epc.mnc380.mcc313.3gppnetwork.org",
    realm: "epc.mnc380.mcc313.3gppnetwork.org",
    ip: "10.179.2.233",
    port: 3868,
    tls: false,
    transport: :diameter_sctp,
    initiate_connection: true
  }
]
```

字段	类型	默认	描述
`host`	string	--	对等 Diameter 主机身份 (FQDN)
`realm`	string	--	对等 Diameter 领域
`ip`	string	--	对等 IP 地址
`port`	integer	`3868`	对等 Diameter 端口
`tls`	boolean	`false`	启用对等连接的 TLS
`transport`	atom	`:diameter_sctp`	传输协议 (`:diameter_sctp` 或 `:diameter_tcp`)
`initiate_connection`	boolean	`true`	OmniLCS □否发起与此对等体的连接

GMLC / Le 接口

网关移动位置中心和与外部 LCS 客户端的 Le 接口的配置。有关请求流、延迟会话和 InfluxDB 日志的完整详细信息，请参见 GMLC & Le 接口操作指南。

```
config :omnilcs, :gmlc,
  enabled: true,
  allow_unknown_clients: false,
  authorized_clients: [
    %{
      name: "psap-01",
      type: :emergency_services,
      allowed_methods: [:cell, :ecid, :gnss, :otdoa],
      rate_limit: 100,
      description: "□□ PSAP"
    }
  ],
  allow_deferred: true,
  max_periodic_sessions: 100,
  max_triggered_sessions: 50,
  default_periodic_poll_interval_ms: 60_000,
  default_triggered_poll_interval_ms: 30_000,
  influx_logging: true
```

参数	类型	默认	描述
`enabled`	boolean	`false`	启用 GMLC Le 接口处理程序
`allow_unknown_clients`	boolean	`false`	接受来自未列出客户端的请求
`authorized_clients`	list	`[]`	授权的 LCS 客户端定义（请参见 GMLC 指南）
`allow_deferred`	boolean	`true`	接受周期性和触发的位置请求
`max_periodic_sessions`	integer	`100`	最大并发周期性会话
`max_triggered_sessions`	integer	`50`	最大并发触发会话
`default_periodic_poll_interval_ms`	integer	`60000`	默认周期性修正间隔（毫秒）
`default_triggered_poll_interval_ms`	integer	`30000`	默认地理围栏轮询间隔（毫秒）
`influx_logging`	boolean	`true`	将所有 GMLC 位置修正写入 InfluxDB

CAP 警报

CAP（通用警报协议）警报摄取和广播的配置。有关警报生命周期、多边形解析和操作员批准工作流的完整详细信息，请参见 CAP 警报操作指南。

```
config :omnilcs, :cap,
  require_approval: true,
  plmn: %{mcc: "001", mnc: "01"},
  coverage_aware: false,
  feeds: []
```

参数	类型	默认	描述
`require_approval`	boolean	`true`	当为 `true` 时，警报排队等待操作员批准。当为 `false` 时，警报自动广播。
`plmn`	map	`%{mcc: "001", mnc: "01"}`	广播消息的 PLMN 身份 (MCC/MNC)
`coverage_aware`	boolean	`false`	使用单元覆盖半径进行多边形交集（与仅中心点相比）
`feeds`	list	`[]`	CAP Atom 订阅 URL

订阅配置

```
feeds: [
  %{url: "https://alerts.weather.gov/cap/us.php?x=1",
poll_interval_seconds: 60}
]
```

字段	类型	必需	默认	描述
`url`	string	□	--	CAP Atom 订阅 URL
`poll_interval_seconds`	integer	否	`60`	轮询之间的秒数

日志记录

```
config :logger,
  backends: [:console, ControlPanel.Logger]
```

参数	类型	描述
`backends`	list	日志记录后端。`:console` 写入 stdout；`ControlPanel.Logger` 将日志馈送到网页 UI 中的日志页面。

□□□□□□

```
import Config

config :control_panel,
  parent_application: :omnilcs,
  parent_application_version: "1.0.0",
  parent_application_readable_name: "OmniLCS",
  home_page: ControlPanelWeb.ApplicationLive,
  use_builtin_pages: [
    {ControlPanelWeb.ApplicationLive, "/application", "□□"},
    {ControlPanelWeb.ConfigurationLive, "/configuration", "□□"},
    {ControlPanelWeb.LogLive, "/log", "□□"}
  ],
  use_additional_pages: [
    {OmniLcs.Web.DashboardLive, "/dashboard", "□□□"},
    {OmniLcs.Web.LocationLive, "/location", "□□"},
    {OmniLcs.Web.CellDatabaseLive, "/cells", "□□"},
    {OmniLcs.Web.DiameterLive, "/diameter", "□□"},
    {OmniLcs.Web.CbcLive, "/cbc", "CBC 2G"},
    {OmniLcs.Web.Cbc3gLive, "/cbc3g", "CBC 3G"},
    {OmniLcs.Web.Cbc4gLive, "/cbc4g", "CBC 4G"},
    {OmniLcs.Web.CapAlertsLive, "/cap", "CAP □□"}
  ],
  page_order: [
    "/dashboard", "/location", "/cells", "/diameter",
    "/cbc", "/cbc3g", "/cbc4g", "/cap", "/application",
"/configuration", "/log"
  ],
  licensee_name: "Omnitouch"

# REST API
config :api_ex,
  api: %{
    port: 8443,
    listen_ip: "0.0.0.0",
    product_name: "OmniLCS",
    title: "API - OmniLCS",
    hostname: "localhost",
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem",
```

```
    routes: [
      %{path: "/status", module: OmniLcs.Api.StatusController,
actions: [:index]},
      %{path: "/location", module: OmniLcs.Api.LocationController,
actions: [:index, :create, :show]},
      %{path: "/cells", module: OmniLcs.Api.CellController,
actions: [:index, :create, :show, :update, :delete]},
      %{path: "/cap", module: OmniLcs.Api.CapController, actions:
[:index, :create, :show, :update]}
    ]
  }

# 控制面板 HTTPS 端点
config :control_panel, ControlPanelWeb.Endpoint,
  server: true,
  url: [host: "0.0.0.0", path: "/"],
  https: [port: 443, keyfile: "priv/cert/omnitouch.pem", certfile:
"priv/cert/omnitouch.crt"],
  adapter: Bandit.PhoenixAdapter,
  secret_key_base: "REPLACE_WITH_64_BYTE_RANDOM_SECRET",
  check_origin: false,
  pubsub_server: ControlPanel.PubSub,
  live_view: [signing_salt: "LcsLvSlt"]

# 日志记录
config :logger,
  backends: [:console, ControlPanel.Logger]

# E-SMLC 一般设置
config :omnilcs,
  esmlc_name: "OmniLCS",
  cell_database_path: "priv/cells.json"

# CBSP (2G 单元广播) - BSC 通过此端口连接
config :omnilcs, :cbsp,
  listen_ip: "0.0.0.0",
  listen_port: 48049

# SABP (3G 单元广播) - RNC 通过此端口连接
config :omnilcs, :sabp,
  listen_ip: "0.0.0.0",
  listen_port: 3452

# SLs (LCS-AP 通过 SCTP) - E-SMLC 连接到这些 MMEs 进行定位
```

```
config :omnilcs, :sls,
  local_ip: "10.5.198.200",
  mme_peers: [
    %{host: "mme01", ip: "10.179.1.15", port: 9082}
  ]

# SBC-AP (4G 单元广播) - OmniLCS 连接到这些 MMEs
config :omnilcs, :sbcap,
  local_ip: "10.5.198.200",
  mme_peers: [
    %{host: "mme01", ip: "10.179.2.100", port: 29168},
    %{host: "mme02", ip: "10.179.2.101", port: 29168}
  ]

# CAP 警报摄取和广播
config :omnilcs, :cap,
  require_approval: true,
  plmn: %{mcc: "001", mnc: "01"},
  coverage_aware: false,
  feeds: []

# InfluxDB 用于单元位置同步
config :omnilcs, OmniLcs.InfluxDb,
  database: "nokia-monitor",
  host: "172.19.3.68",
  port: 8086,
  auth: [method: :basic, username: "monitor", password:
"REPLACE_WITH_PASSWORD"],
  http_opts: [recv_timeout: 30_000],
  pool: [max_overflow: 10, size: 5]

# Diameter (SLg 接口到 DRA/MME - GMLC 角色，而不是 E-SMLC SLs)
config :diameter_ex,
  diameter: %{
    service_name: :omnitouch_esmlc,
    listen_ip: "10.5.198.200",
    listen_port: 3868,
    host: "amanaki",
    realm: "epc.mnc380.mcc313.3gppnetwork.org",
    product_name: "OmniLCS",
    request_timeout: 5000,
    peer_selection_algorithm: :random,
    allow_undefined_peers_to_connect: true,
    log_unauthorized_peer_connection_attempts: true,
```

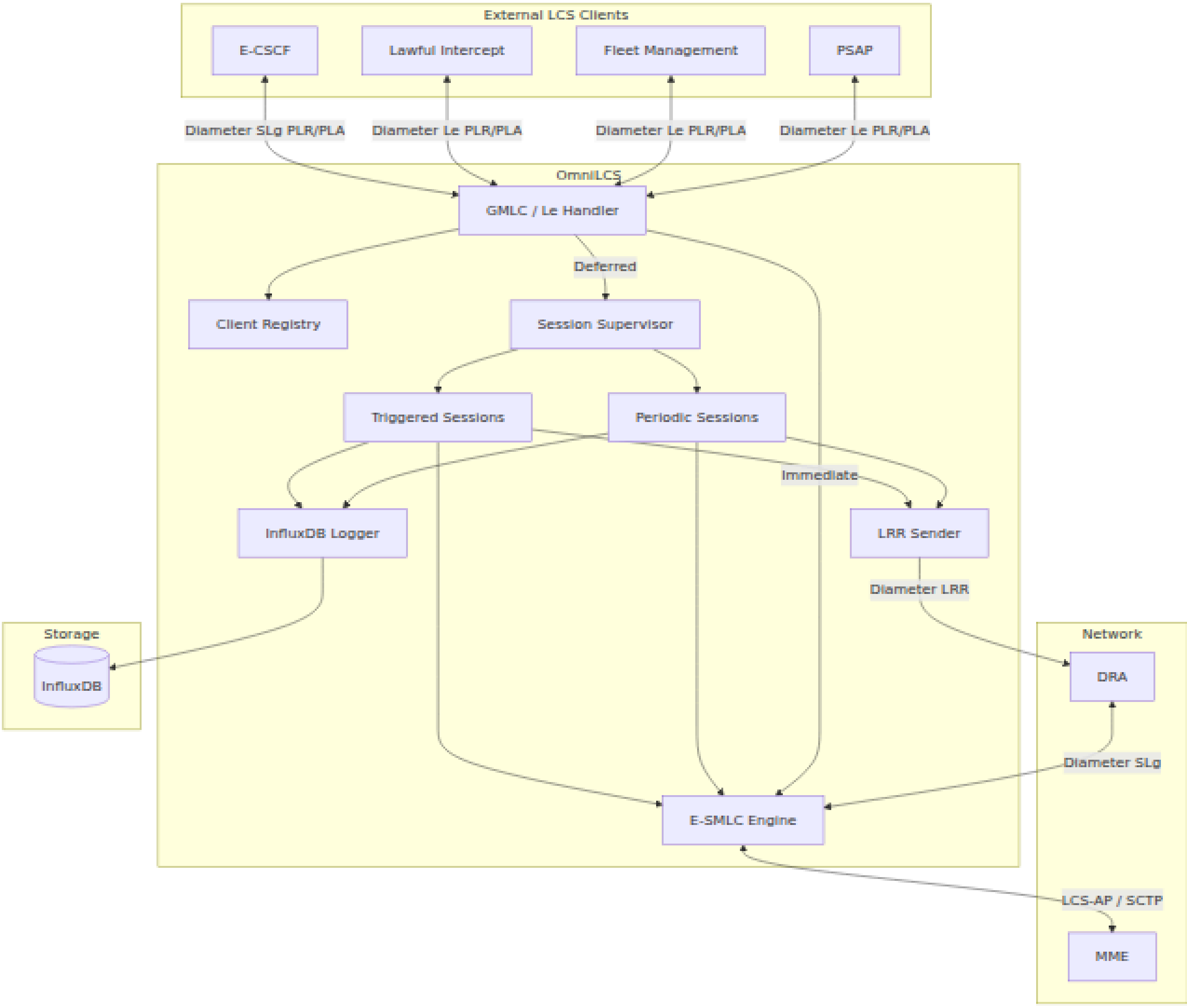
```
    control_module: OmniLcs.Control.Diameter,
    processor_module: DiameterEx.Processor,
    auth_application_ids: [],
    acct_application_ids: [],
    vendor_id: 10415,
    supported_vendor_ids: [5535, 10415],
    applications: [
      %{
        application_name: :slg,
        application_dictionary: :diameter_gen_3gpp_slg,
        vendor_specific_application_ids: [
          %{vendor_id: 10415, auth_application_id: 16_777_264,
acct_application_id: nil}
        ]
      }
    ],
    peers: [
      %{
        host: "dra01.epc.mnc380.mcc313.3gppnetwork.org",
        realm: "epc.mnc380.mcc313.3gppnetwork.org",
        ip: "10.179.2.233",
        port: 3868,
        tls: false,
        transport: :diameter_sctp,
        initiate_connection: true
      }
    ]
  }
```

GMLC & Le 接口□作指南

OmniLCS 包含一个网关移动位置中心（GMLC）功能，向外部 LCS 客户端暴露 **Le** 接口。这允许公共安全应答点（PSAP）、车队管理系统、合法拦截平台和其他外部方通过 3GPP TS 29.172 中定义的基于 Diameter 的 Le 参考点请求订阅者位置。

GMLC 支持 即时 和 延迟 位置请求。延迟请求包括 周期性位置（定期报告位置）和 触发位置（当订阅者进入、离开或停留在某个地理区域时报告位置）。

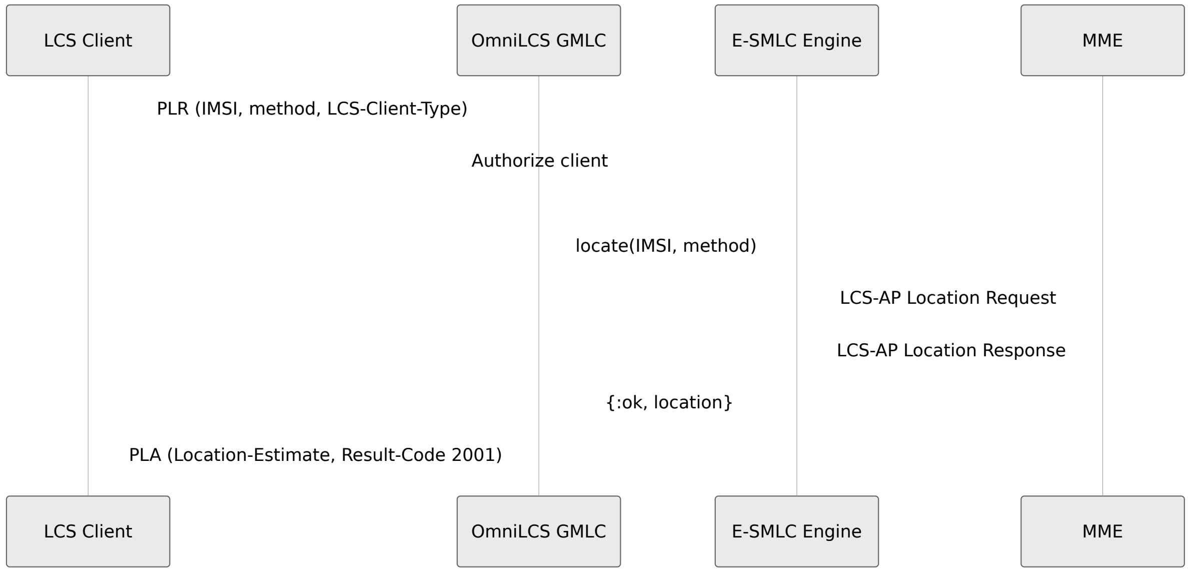
架构



请求流程

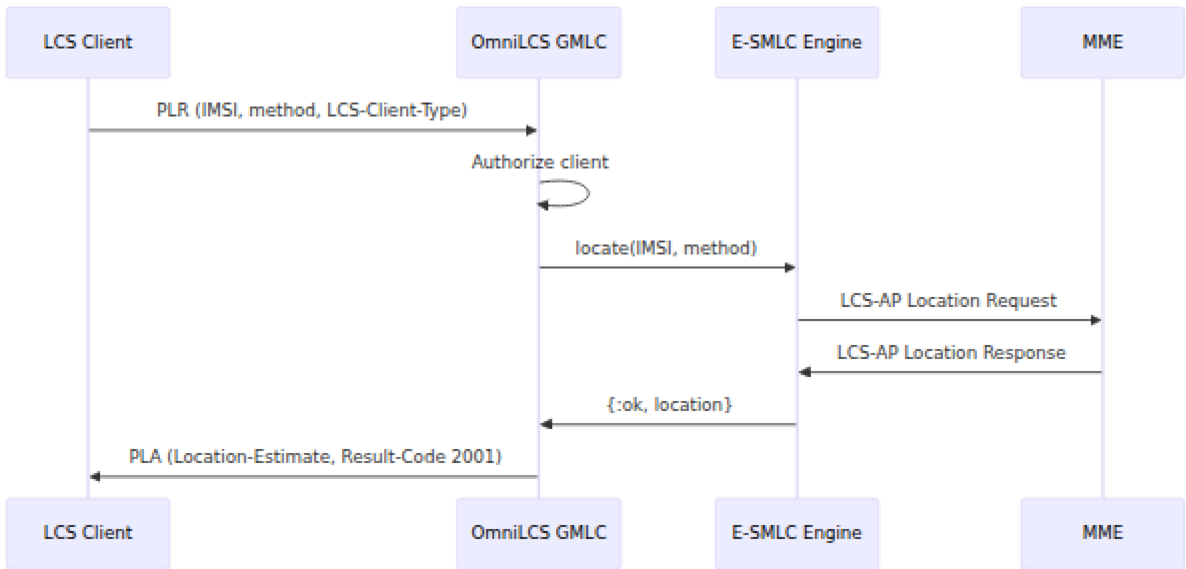
即时位置请求

外部 LCS 客户端通过 Diameter 发送提供位置请求（PLR）。OmniLCS 授权客户端，运行定位引擎，并在提供位置应答（PLA）中返回结果。



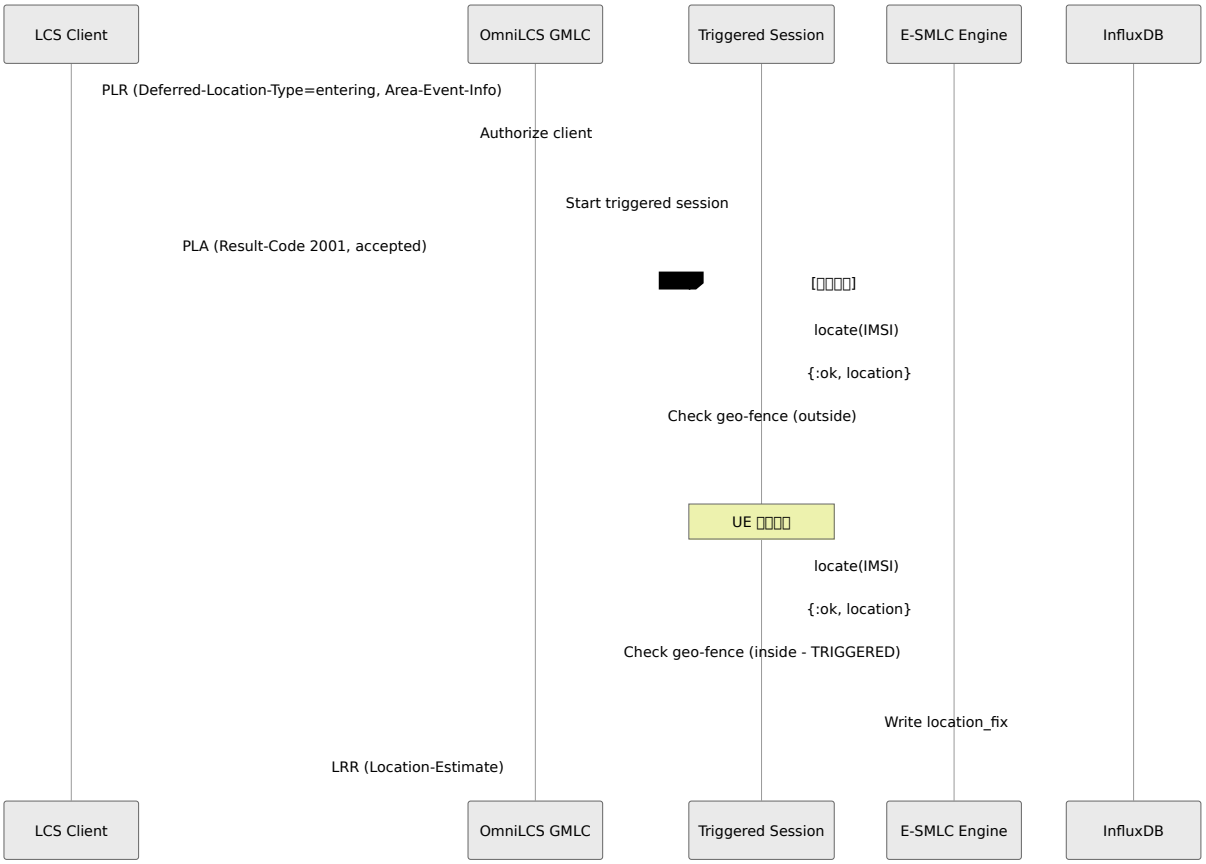
周期性位置请求

当PLR包含设置了周期位的`Deferred-Location-Type`和分组AVP的`Periodic-LDR-Information`时，OmniLCS创建一个周期会话，在指定的时间间隔内运行位置修正，并通过位置报告请求（LRR）交付每个结果。



触发（地理围栏）位置请求

当PLR包含区域事件信息时，OmniLCS创建一个触发会话，轮询UE位置，并在满足触发条件时触发。


LCS Client
OmniLCS GMLC
Triggered Session
E-SMLC Engine
InfluxDB
PLR (Deferred-Location-Type=entering, Area-Event-Info)
Authorize client
Start triggered session
PLA (Result-Code 2001, accepted)
[□□□□]
locate(IMSI)
{:ok, location}
Check geo-fence (outside)
UE □□□□
locate(IMSI)
{:ok, location}
Check geo-fence (inside - TRIGGERED)
Write location_fix
LRR (Location-Estimate)
LCS Client
OmniLCS GMLC
Triggered Session
E-SMLC Engine
InfluxDB

配置

GMLC 配置

```
config :omnilcs, :gmlc,
  # 启用/禁用 GMLC Le 接口
  enabled: true,

  # 允许来自未授权列表中的客户端的请求
  allow_unknown_clients: false,

  # 授权的外部 LCS 客户端
  authorized_clients: [
    %{
      name: "psap-01",
      type: :emergency_services,
      allowed_methods: [:cell, :ecid, :gnss, :otdoa],
      rate_limit: 100,
      description: "主要 PSAP"
    },
    %{
      name: "fleet-mgmt",
      type: :value_added_services,
      allowed_methods: [:cell, :ecid],
      rate_limit: 50,
      description: "车队管理系统"
    }
  ],

  # 允许延迟（周期性/触发）位置请求
  allow_deferred: true,

  # 最大并发周期性会话
  max_periodic_sessions: 100,

  # 最大并发触发（地理围栏）会话
  max_triggered_sessions: 50,

  # 默认周期性会话的间隔（毫秒）
  default_periodic_poll_interval_ms: 60_000,

  # 默认触发会话的轮询间隔（毫秒）
```

```
default_triggered_poll_interval_ms: 30_000,

# 将所有位置修正记录到 InfluxDB
influx_logging: true
```

GMLC 参数

参数	类型	必需	默认	描述
`enabled`	布尔值	否	`false`	启用 GMLC Le 接口处理[当禁用时，所有 PLR 被视为标 SLg 请求。
`allow_unknown_clients`	布尔值	否	`false`	接受未列在 `authorized_clier` 中的客户端的请求。在集成测试期 置为 `true` 以允许模式。
`authorized_clients`	列表	否	`[]`	授权的外部 LCS 客户端配置[请参见下面的 客户端参数。
`allow_deferred`	布尔值	否	`true`	接受延迟（周期性和触发）位置请 当 `false` 时，仅处理即时[
`max_periodic_sessions`	整数	否	`100`	最大并发周期性位置会话数。当超 返回 Diameter 3004 （TOO_BUSY）。
`max_triggered_sessions`	整数	否	`50`	最大并发触发位置会话数。
`default_periodic_poll_interval_ms`	整数	否	`60000`	当 PLR 中未指定时，周期性位 正之间的默认间隔。
`default_triggered_poll_interval_ms`	整数	否	`30000`	当在 Area-Event-Info AVP 中未指定时，触发会话的[轮询间隔。
`influx_logging`	布尔	否	`true`	将 GMLC 会话中的所有位置 写入 InfluxDB。

□□	□□	□□	□□	□□
	□			

□□□□□

□□ `authorized_clients` □□□□□

□□	□□	□□	□□	□□
`name`	□□□	□	--	□□□□□□□ PLR □□ LCS-EPS-Client-Name □ LCS-Requestor-ID-String AVP □□□
`type`	□□	□	`:any`	□□□ LCS-Client-Type: `:emergency_services`, `:value_added_services`, `:plmn_operator_services`, `:lawful_intercept_services`, □ `:any`□
`allowed_methods`	□□	□	`[:cell, :ecid, :gnss, :otdoa]`	□□□□□□□□□□□□□□□□
`rate_limit`	□□	□	`0`	□□□□□□□□□□□ `0` □□□□□□□
`description`	□□□	□	`""`	□□□□□□□□□□□

LCS 客户端类型

根据 3GPP TS 29.172 第 7.4.4 节:

值	类型	描述
0	紧急服务	PSAP, E-CSCF 紧急呼叫路由
1	增值服务	车队管理、资产跟踪、基于位置的服务
2	PLMN 运营商服务	运营商内部服务（O&M，网络优化）
3	合法拦截服务	执法位置请求

延迟位置类型

周期性位置

周期性位置会话在固定的时间间隔内执行可配置数量的位置修正。每个修正：

1. 由 E-SMLC 引擎使用请求的位置方法执行
2. 作为带有会话元数据的 `location_fix` 测量写入 InfluxDB
3. 通过标准位置记录器（CSV + ETS）记录
4. 通过 Diameter LRR 交付给发起的 LCS 客户端

当报告计数耗尽时，会话终止。

使用的 **Diameter AVP**:

AVP	□□	□□
`Deferred-Location-Type`	1480	□□□ 4 □□□□□□ LDR
`Periodic-LDR-Information`	2025	□□□□□□□□□ AVP
`Reporting-Amount`	2026	□□□□□□□□□□
`Reporting-Interval`	2027	□□□□□□□□□□

□□□□□□□□□

□□□□□□□□ UE □□□□□□□□□□

□□	□□
`imsi`	□□□ IMSI
`method`	□□□□□□□□□cell, ecid, gnss, otdoa□
`source`	□□□□□□□□□□□
`session_type`	`periodic`, `triggered`, □ `immediate`
`client_name`	□□□ LCS □□□□□□

□□□

□□	□□	□□
`latitude`	□□□	WGS84 □□□□□
`longitude`	□□□	WGS84 □□□□□
`altitude`	□□□	□□□□□□□□□
`uncertainty`	□□□	□□□□□□□□□□
`confidence`	□□	□□□□□0-100□
`duration_ms`	□□	□□□□□□□□□

□□ InfluxQL □□□

```
-- 订阅者的最新修正
SELECT * FROM location_fix WHERE imsi = '001010000000001' ORDER BY
time DESC LIMIT 10

-- 最近一小时的周期性会话跟踪
SELECT latitude, longitude FROM location_fix
WHERE session_type = 'periodic' AND imsi = '001010000000001' AND
time > now() - 1h

-- 按方法的修正成功率
SELECT COUNT(*) FROM location_fix WHERE time > now() - 24h GROUP
BY method

-- 地理围栏触发事件
SELECT * FROM location_fix WHERE session_type = 'triggered' AND
time > now() - 24h
```

REST API

延迟位置 API 可在 `https://<host>:8443/api/deferred_location` 访问。

列出活动会话

```
GET /api/deferred_location
```

响应：

```
{
  "status": "ok",
  "count": 2,
  "data": [
    {
      "session_id": "a1b2c3d4-...",
      "type": "periodic",
      "imsi": "001010000000001",
      "method": "cell",
      "client_name": "rest-api",
      "status": "active",
      "interval_ms": 60000,
      "remaining_reports": 7,
      "total_reports": 10,
      "started_at": "2026-04-09T10:00:00Z",
      "last_fix_at": "2026-04-09T10:03:00Z"
    }
  ]
}
```


□□□□□□□

```
POST /api/deferred_location
Content-Type: application/json

{
  "type": "periodic",
  "imsi": "001010000000001",
  "method": "cell",
  "interval_seconds": 60,
  "count": 10
}
```

创建触发会话

```
POST /api/deferred_location
Content-Type: application/json

{
  "type": "triggered",
  "imsi": "001010000000001",
  "method": "cell",
  "event_type": "entering",
  "poll_interval_seconds": 30,
  "max_reports": 0,
  "areas": [
    {
      "type": "circle",
      "center": {"lat": -33.8688, "lon": 151.2093},
      "radius_meters": 500
    }
  ]
}
```

取消会话

```
DELETE /api/deferred_location/:session_id
```

指标

Le 接口请求

指标: `omnilcs_gmlc_le_request_total` 类型: 计数器 描述: 来自外部 LCS 客户端的总 Le 接口请求 标签:

- `client_type` -- LCS 客户端类型（emergency_services, value_added_services 等）
- `result` -- 请求结果: `received`, `success`, `error`, `unauthorized`

指标: `omnilcs_gmlc_le_request_duration` 类型: 直方图 描述: Le 请求处理时间（毫秒） 标签:

- `client_type` -- LCS 客户端类型

位置报告请求

指标：`omnilcs_gmlc_lrr_total` 类型：计数器 描述：发送给外部客户端的总位置报告请求 标签：

- `session_type` -- `periodic` 或 `triggered`
- `result` -- `sent` 或 `error`

会话跟踪

指标：`omnilcs_gmlc_session_periodic_active` 类型：测量 描述：活动周期性位置会话的数量

指标：`omnilcs_gmlc_session_triggered_active` 类型：测量 描述：活动触发/地理围栏位置会话的数量

指标：`omnilcs_gmlc_session_total` 类型：计数器 描述：创建的总延迟会话 标签：

- `type` -- `periodic` 或 `triggered`

地理围栏事件

指标：`omnilcs_gmlc_geofence_trigger_total` 类型：计数器 描述：触发的总地理围栏事件 标签：

- `event_type` -- `entering`, `leaving` 或 `being_inside`

InfluxDB 写入

指标：`omnilcs_gmlc_influx_write_total` 类型：计数器 描述：来自 GMLC 会话的总 InfluxDB 位置修正写入 标签：

- `result` -- `success` 或 `error`

示例 **Prometheus** 查询：

```
# Le 请求速率
rate(omnilcs_gmlc_le_request_total[5m])

# 按类型的延迟会话创建速率
rate(omnilcs_gmlc_session_total[5m])

# 地理围栏触发速率
rate(omnilcs_gmlc_geofence_trigger_total[5m])

# LRR 交付错误比例
sum(rate(omnilcs_gmlc_lrr_total{result="error"}[5m]))
  / sum(rate(omnilcs_gmlc_lrr_total[5m]))
```

故障排除

Le 请求返回 5012 (UNABLE_TO_COMPLY)

症状：外部 LCS 客户端对所有 PLR 接收到结果代码 5012。

可能原因：

- 配置中未启用 GMLC（`enabled: false`）
- 客户端不在 `authorized_clients` 中且 `allow_unknown_clients` 为 `false`
- E-SMLC 引擎无法连接到任何 MME（检查 SL 连接状态）

解决方案：

1. 验证 `config :omnilcs, :gmlc, enabled: true` 已设置
2. 检查客户端名称与 PLR 中的 `LCS-EPS-Client-Name` AVP 匹配
3. 验证 SL 连接在仪表板或 SL 页面上建立

周期会话未启动

症状：带有周期性延迟类型的 PLR 返回 3004 (TOO_BUSY)。

可能原因：

- 达到 `max_periodic_sessions` 限制
- 会话监督员未运行

解决方案：

1. 检查 GMLC 控制面板页面上的活动会话计数
2. 如有需要，增加 `max_periodic_sessions`
3. 验证 `OmniLcs.Gmlc.SessionSupervisor` 是否在应用页面中运行

位置修正未出现在 InfluxDB 中

症状：周期/触发会话正在运行，但 InfluxDB 中没有数据。

可能原因：

- `influx_logging` 设置为 `false`
- InfluxDB 连接中断
- 数据库不存在

解决方案：

1. 验证 GMLC 配置中的 `influx_logging: true`
2. 检查 InfluxDB 连接（元同步状态一个良好的代理）
3. 直接查询 InfluxDB: `SHOW MEASUREMENTS` 应包括 `location_fix`

地理围栏未触发

症状：触发会话处于活动状态但从未触发。

可能原因：

- 区域定义对于定位方法的精度过小
- 错误的事件类型（例如，当订阅者已经在外部时使用 `:leaving`）
- 定位方法返回 `nil` 坐标

解决方案：

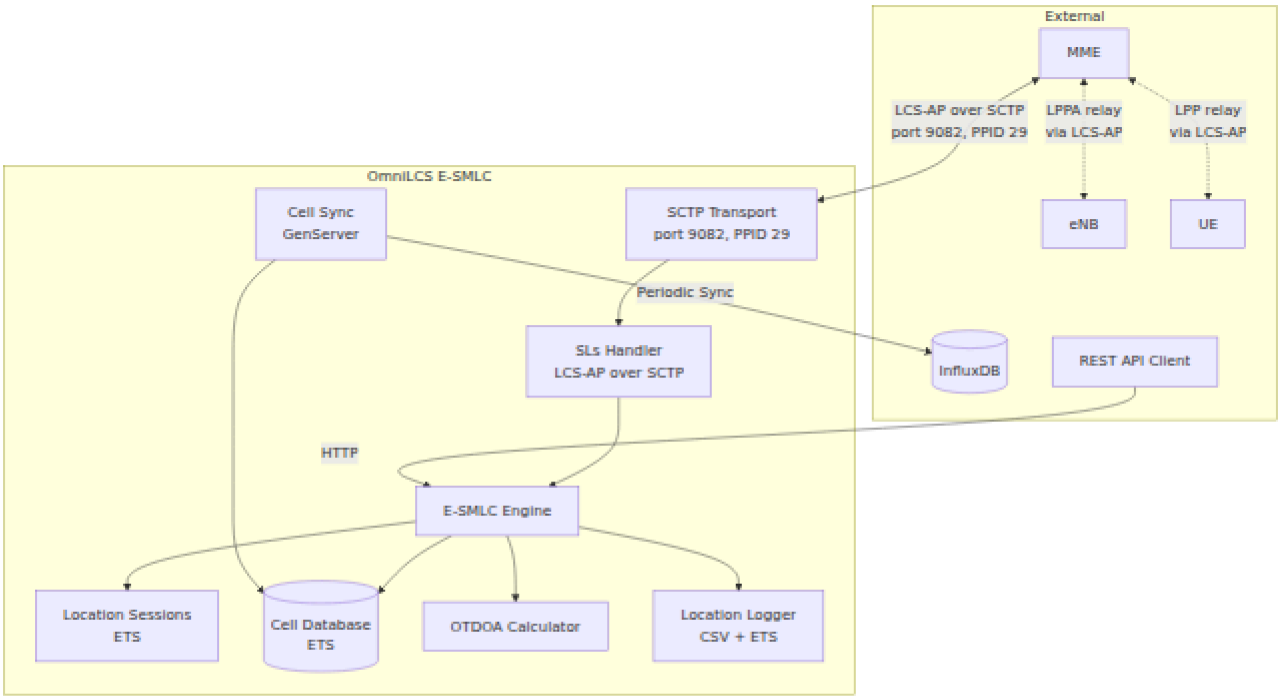
1. 检查定位方法的精度 -- 小区 ID 具有公里级的不确定性，可能对于小地理围栏过于粗糙

2. 对于较小区域，使用更精确的方法（`:ecid` 或 `:gnss`）
3. 验证区域坐标□否正确（纬度/经度顺序）

E-SMLC 位置服务指南

OmniLCS 实现了一个演进服务移动位置中心 (E-SMLC)，为 LTE 网络提供用户设备 (UE) 定位。E-SMLC 使用多种定位方法确定 UE 位置，并通过 SLs 接口使用原生 LCS-AP 通过 SCTP 与 MME 通信，符合 3GPP TS 29.171。

架构



定位方法

小区 **ID**

最简单的定位方法。返回服务小区在小区数据库中的地理位置。

- 准确性：小区半径（通常为 100 米 - 5 公里，具体取决于小区类型）
- 延迟：毫秒（本地数据库查找）
- **UE** 支持要求：无
- **eNB** 支持要求：无

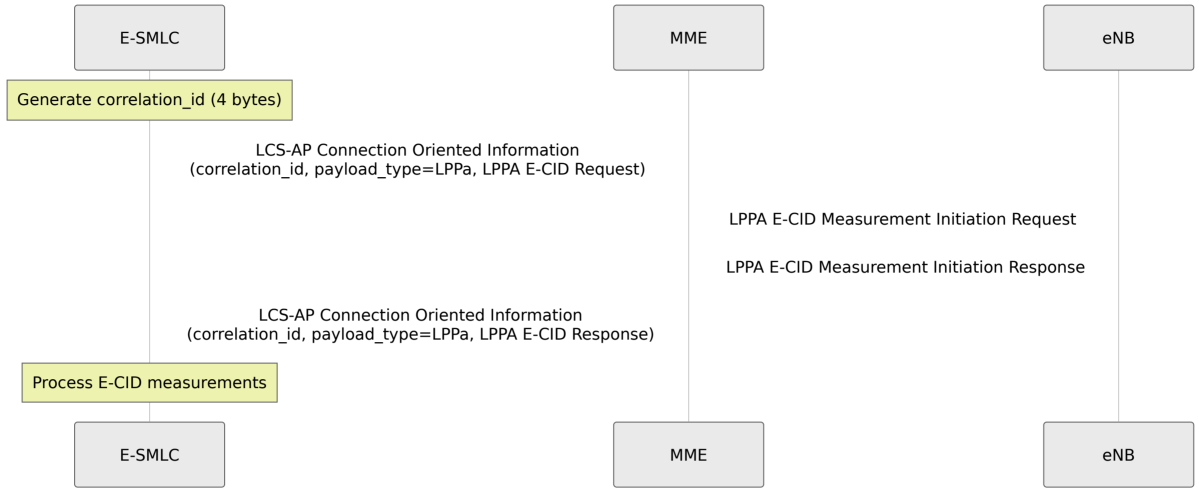
引擎首先尝试在本地小区数据库中查找服务小区。如果未找到，它将通过 MME 使用 LCS-AP 面向连接的信息向 eNB 发送 LPPA E-CID 测量启动请求（仅包含 cell_id 数量）以获取小区标识。

增强小区 **ID (E-CID)**

使用 LPPA 从 eNB 获取无线测量，包括时间提前和信号强度。

- 准确性：50 米 - 500 米

- 延迟: 1-5 秒
- **UE** 支持要求: 无
- **eNB** 支持要求: □ (LPPA)



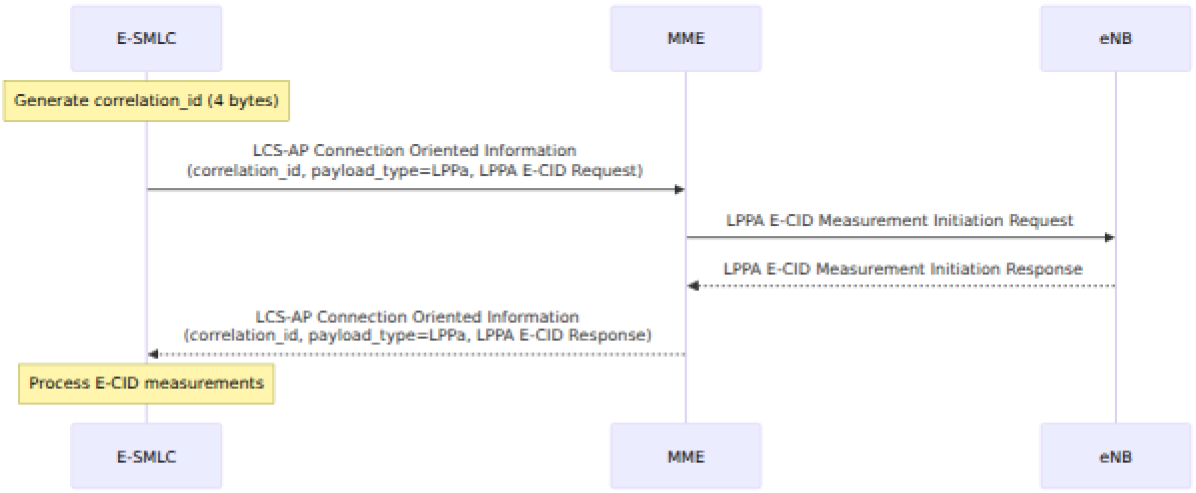
请求的测量数量:

数量	描述
小区 ID	服务小区标识
时间提前类型 2	UE 到 eNB 的传播延迟（距离估计）
RSRP	参考信号接收功率
RSRQ	参考信号接收质量

GNSS / A-GPS

请求 UE 通过 LPP 协议报告其 GPS 坐标。

- 准确性: 5 米 - 50 米
- 延迟: 5-30 秒
- **UE** 支持要求: □ (GNSS 能力)
- **eNB** 支持要求: 无

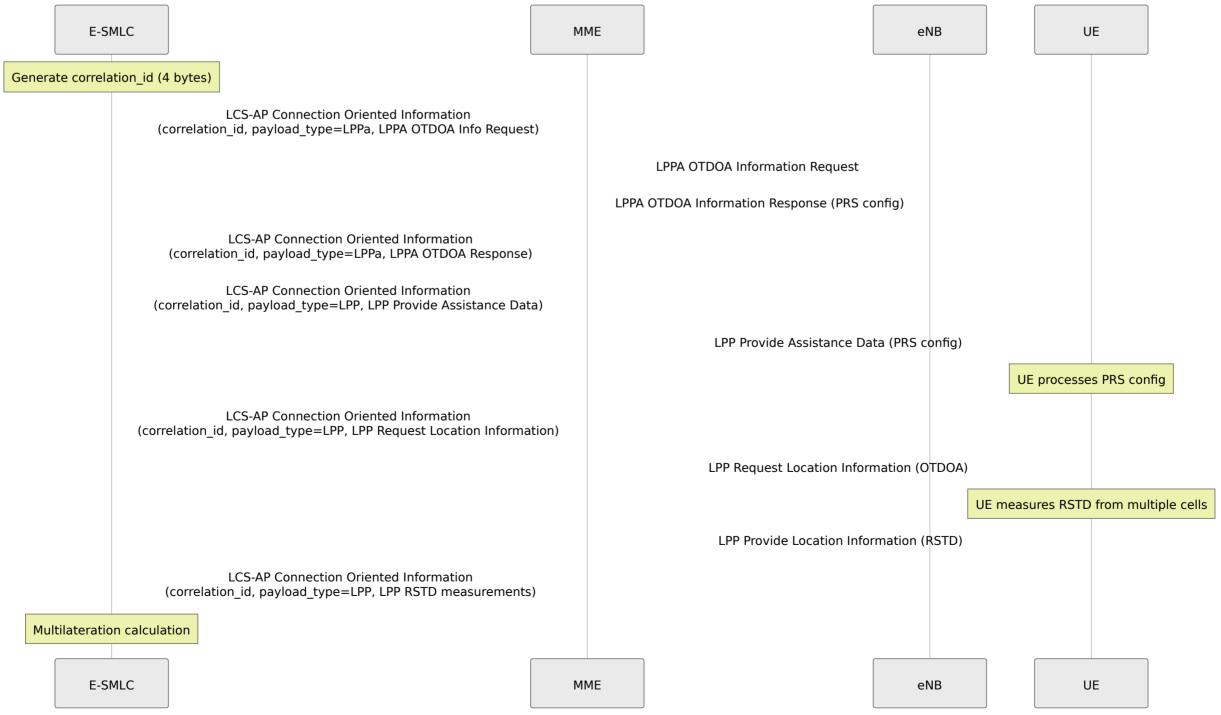


E-SMLC 通过 MME 通过 LCS-AP 向 UE 发送 LPP 请求位置信息消息以获取 GNSS。UE 执行 GNSS 定位并通过 LPP 报告其坐标，经过 MME 转发至连接导向信息消息。

OTDOA

观察到的到达时间差。使用来自多个小区的定位参考信号 (PRS) 测量进行多边定位。

- 准确性: 10 米 - 100 米
- 延迟: 5-15 秒
- **UE** 支持要求: □ (OTDOA 能力)
- **eNB** 支持要求: □ (已配置 PRS, LPPA)



OTDOA 步骤:

1. 通过 LPPA 从 eNB 请求 OTDOA 小区信息（通过 LCS-AP 连接导向信息转发）
2. 接收 PRS 配置、小区 ID、参考小区和邻近小区的 EARFCN
3. 通过 LPP 向 UE 发送 PRS 辅助数据（通过 LCS-AP 连接导向信息转发）
4. 通过 LPP 请求 UE 的 RSTD 测量
5. 接收 RSTD 测量
6. 运行多边定位以计算位置

SLs 接口 -- LCS-AP over SCTP

E-SMLC 使用 LCS 应用协议 (LCS-AP) 通过 SCTP 与 MME 通信，符合 3GPP TS 29.171。OmniLCS 向每个配置的 MME 对等体发起 SCTP 关联。

传输详细信息

参数	值
协议	LCS-AP over SCTP
SCTP PPID	29
默认端口	9082 (IANA 注册)
连接方向	E-SMLC 发起到 MME
编码	ASN.1 PER 对齐

LCS-AP 程序

程序	代码	描述
位置请求 / 位置响应	0	E-SMLC 请求 MME 定位 UE
面向连接的信息	1	转发 LPPA/LPP PDU（与 UE 关联，使用 correlation_id）
无连接的信息	2	转发非 UE 关联的 LPPA PDU
位置中止	3	中止正在进行的定位程序
重置	4	重置所有定位程序

关联 ID

每个与 UE 关联的 LCS-AP 事务使用关联 **ID** -- 一个 4 字节的二进制值，将位置请求、所有连接导向信息交换和位置响应绑定在一起，形成一个单一的定位会话。E-SMLC 在每个会话开始时使用 `:crypto.strong_rand_bytes(4)` 生成一个随机的关联 ID。

关联 ID 作为 IE (ID 2) 包含在:

- 位置请求 (E-SMLC -> MME)
- 位置响应 (MME -> E-SMLC)
- 连接导向信息（双向）
- 位置中止 (E-SMLC -> MME)

APDU 隧道

LPP 和 LPPA PDU 被包含在 LCS-AP 连接导向信息 消息中。有效载荷类型 IE (ID 15) 标识内容：

有效载荷类型	值	描述
`:lPP`	0	LPP PDU (UE 定位协议, TS 36.355)
`:lPPa`	1	LPPa PDU (eNB 定位协议, TS 36.455)

APDU IE (ID 1) 携带二进制 LPP 或 LPPa PDU。MME 在 E-SMLC 和目标 eNB 或 UE 之间透明地转发这些。

入站消息 (MME -> E-SMLC)

位置响应 (程序代码 **0**, 成功结果 / 失败结果)

MME 发送位置响应作为位置请求的结果。

IE	ID	重要性	描述
关联 ID	2	拒绝	来自原始位置请求���关联 ID
位置估计	12	拒绝	GAD 编码位置（成功时）
定位数据	16	拒绝	定位方法信息（成功时）
速度估计	21	拒绝	UE 的速度（可选，成功时）
准确性满足指示	0	拒绝	□否满足请求的准确性（成功时）
LCS 原因	11	忽略	失败原因（失败时）

连接导向信息（程序代码 **1**，发起消息）

MME 将来自 eNB 或 UE 的 LPPA/LPP PDU 转发回 E-SMLC。

IE	ID	重要性	描述
关联 ID	2	拒绝	会话关联
有效载荷类型	15	拒绝	`:lPP` 或 `:lPPa`
APDU	1	拒绝	二进制 LPP 或 LPPa PDU

重置请求（程序代码 **4**，发起消息）

MME 请求 E-SMLC 重置所有活动程序。E-SMLC 以重置确认响应。

IE	ID	重要性	描述
LCS 原因	11	忽略	重置原因

出站消息 (E-SMLC -> MME)

位置请求（程序代码 **0**，发起消息）

E-SMLC 请求 MME 定位 UE。

IE	ID	重要性	描述
关联 ID	2	拒绝	4 字节会话关联 ID
位置类型	13	拒绝	`geographic-Information`、`assistance-Information` 或 `last-known-location`
E-UTRAN 小区 ID	4	忽略	UE 服务小区的 E-CGI
LCS 客户端类型	8	拒绝	LCS 客户端类型（可选）
LCS 优先级	9	拒绝	位置请求优先级（可选）
LCS QoS	10	拒绝	服务质量要求（可选）
IMSI	7	忽略	UE IMSI（可选）
IMEI	6	忽略	UE IMEI（可选）
包含速度	5	拒绝	请求速度估计（可选）

连接导向信息（程序代码 **1**，发起消息）

E-SMLC 通过 MME 向 eNB 或 UE 发送 LPPA/LPP PDU。

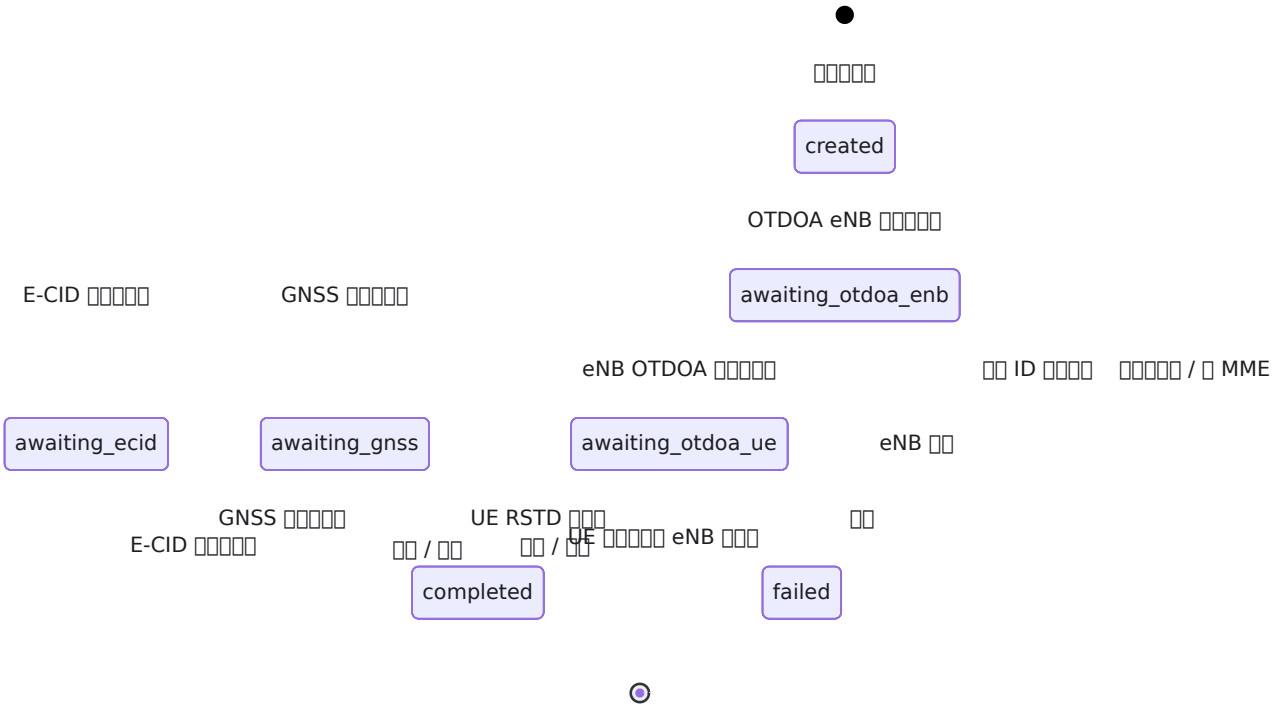
IE	ID	重要性	描述
关联 ID	2	拒绝	会话关联
有效载荷类型	15	拒绝	`:lPP` 或 `:lPPa`
APDU	1	拒绝	二进制 LPP 或 LPPa PDU

无连接信息（程序代码 **2**，发起消息）

E-SMLC □□ MME □□□ UE □□□ LPPA PDU□□□□eNB □□□□□□

IE	ID	□□□	□□
□□□	19	□□	E-SMLC □□□□□□
□□ ID	3	□□	□□ eNB □□□□□□
APDU	1	□□	□□□ LPPa PDU

□□□□□□□□



□□□□□□□

□□	□□
`session_id`	□□□□□□□□□□ `esmlc-1234567890-1` □
`imsi`	UE □□□
`mme_host`	□□□□□ MME
`method`	□□□□□□□
`state`	□□□□□□
`created_at`	□□□□□□□
`updated_at`	□□□□□□□□□
`completed_at`	□□□□□
`lppa_transactions`	□□□ LPPA PDU □□
`lpp_transactions`	□□□ LPP PDU □□
`measurements`	□□□□□□
`result`	□□□□□□

□□ 1 □□□□□□□□□ `LocationSession.cleanup_old_sessions/1` □□□

□□□□□□□

E-SMLC □□ ETS □ (`:pending_transactions`) ���□□ LCS-AP □□□□□□□□□□

4. 在 `receive` 块中等待响应
5. 当 SLs 处理程序接收到匹配的响应时，它通过关联 ID 查找待处理事务，并将结果发送给等待的进程

小区数据库管理

小区数据库存储每个小区站点的地理位置和无线参数。它用于小区 ID 定位和 OTDOA 计算。

小区记录字段

字段	类型	必需	描述
`cell_id`	any	□	唯一小区标识符
`latitude`	float	□	小区纬度（十进制度）
`longitude`	float	□	小区经度（十进制度）
`pci`	integer	否	物理小区标识 (0-503)
`earfcn`	integer	否	E-UTRA 绝对无线频率信道号
`radius`	integer	否	小区覆盖半径（米）（默认: 1000）
`azimuth`	float	否	天线方位角（度）
`height`	float	否	天线高度（米）
`prs_config`	map	否	OTDOA 的 PRS 配置

PRS 配置字段

字段	类型	描述
`bandwidth`	integer	PRS 带宽（资源块数，6、15、25、50、75、100）
`config_index`	integer	PRS 配置索引（0-4095）
`num_dl_frames`	integer	连续 DL 子帧的数量
`cp_length`	atom	循环前缀长度（`:normal` 或 `:extended`）
`num_antenna_ports`	integer	天线端口数量（1、2 或 4）

InfluxDB 小区同步

小区位置定期从 InfluxDB 同步：

参数	值	描述
同步间隔	5 分钟	自动同步周期
初始延迟	10 秒	启动后第一次同步的延迟
同步超时	60 秒	同步操作的最大时间

同步也可以通过以下方式手动触发：

- REST API: `POST /api/cells/sync`
- LiveView UI: 在小区页面上点击“从 InfluxDB 同步”按钮

JSON 导入

可以从 JSON 文件导入小区：

```
[
  {
    "cell_id": "001-01-0001-01",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "pci": 100,
    "earfcn": 1300,
    "radius": 500,
    "prs_config": {
      "bandwidth": 50,
      "config_index": 0,
      "num_dl_frames": 1
    }
  }
]
```

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ REST API □□□`GET /api/cells/nearby?lat=X&lon=Y&radius=R`□□□□□□□□□□□□□

OTDOA □□

OTDOA □□□□□□□□□□□□ RSTD□□□□□□□□□□□□□□□□ UE □□□

□□

1. **RSTD** □□□□□`dd = RSTD * Ts * c`□□□ Ts = 1/(15000 * 2048) □□c = □□
2. □□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□
3. □□
4. □□□□□□□□□□□□□□□ 1 □□□□□ 50 □□□
5. □□□□□□□□□□□□□□□□□□□□□□□□

□□

- □□□□ 2 □□□□□□□□□□□□□□□ = □□ 3 □□□□□□□□□□□

- 建议 3 个或更多邻近小区以获得明确的 2D 定位
- 所有小区必须在小区数据库中具有已知位置
- 小区位置通过 PCI、ECGI 或 cell_id 解析

位置的 REST API 参考

请参阅 REST API 参考 以获取完整的端点文档。关键位置端点：

端点	方法	描述
`/api/location`	POST	请求新的 UE 位置
`/api/location`	GET	列出最近的位置修正
`/api/location/:imsi`	GET	某个 IMSI 的最后已知位置
`/api/location/:imsi/history`	GET	某个 IMSI 的位置历史
`/api/location/:imsi/history/csv`	GET	位置历史的 CSV 导出

故障排除

没有 SLs 对等体连接

1. 验证 `:sls` 配置：`local_ip` 必须可以从 MME 网络访问
2. 检查每个 `mme_peers` 条目□否具有正确的 IP 地址和端口（默认 9082）
3. 在日志中查找 SCTP 连接错误：`SLs: Failed to connect to MME`
4. 确认 SCTP 没有被防火墙阻止（IP 协议 132）
5. 确认 MME 正在监听 LCS-AP 的 9082 端口

位置请求返回 "no_mme_host"

E-SMLC 无法确定将 LPPA/LPP 消息发送到哪个 MME。确保：

1. 至少建立了一个 SLs SCTP 关联
2. 对于 REST API 请求，提供 `mme_host` 参数
3. 检查仪表板页面上的 SLs 连接状态

MME 没有 LCS-AP 响应

1. 验证 SCTP 关联□否处于 `:established` 状态，通过 `SctpTransport.get_connections/0`
2. 检查日志中的 SCTP 心跳失败
3. 确认 MME 支持 LCS-AP（TS 29.171）
4. 检查请求和响应之间的关联 ID □否正确匹配

小区 ID 定位返回无坐标

小区数据库中没有匹配的小区。操作：

1. 触发 InfluxDB 同步：`POST /api/cells/sync`
2. 通过 REST API 或 LiveView UI 手动添加小区
3. 从 JSON 文件导入小区

OTDOA 超时

eNB 在超时期间未响应 OTDOA 信息。可能原因：

1. eNB 不支持 LPPA OTDOA 信息程序
2. eNB 上未配置 PRS
3. MME 和 eNB 之间的网络路径问题

GNSS 超时

UE 在超时内未报告 GNSS 坐标。可能原因：

1. UE 不支持 GNSS 定位
2. UE 在室内（无卫星可见）
3. 未提供 GNSS 辅助数据（冷启动需要更长时间）

3GPP 参考

规范	标题
TS 29.171	MME 与 E-SMLC 之间的 LCS 应用协议 (LCS-AP) (SLs 接口)
TS 29.172	GMLC 与 MME 之间的 EPC LCS 协议 (SLg Diameter 接口)
TS 36.455	eNB 与 E-SMLC 之间的 LTE 定位协议 A (LPPa)
TS 36.355	UE 与 E-SMLC 之间的 LTE 定位协议 (LPP)
TS 23.032	通用地理区域描述 (GAD 编码)

OmniLCS REST API 参考

OmniLCS REST API 通过 HTTPS 在端口 **8443** 提供服务。基本 URL 为 `https://<host>:8443/api`。

API 文档（Swagger UI）可在 `https://<host>:8443/api/docs` 获取。OpenAPI 架构位于 `https://<host>:8443/api/schema`。

认证

API 当前不强制执行认证。访问应在网络层面进行限制。

响应格式

所有响应使用 JSON。成功响应包括 `"status": "ok"` 字段。错误响应包括 `"status": "error"` 和 `"reason"` 字段。

系统状态

GET /api/status

返回系统健康和操作状态。

响应 **(200)**

```
{
  "status": "ok",
  "version": "1.0.0",
  "name": "OmniLCS",
  "diameter_peers": [
    {
      "host": "dra01.epc.mnc380.mcc313.3gppnetwork.org",
      "realm": "epc.mnc380.mcc313.3gppnetwork.org",
      "state": "Connected",
      "transport": "sctp"
    }
  ],
  "active_sessions": 2,
  "completed_sessions": 47,
  "cells_loaded": 128,
  "cell_sync": {
    "last_sync": "2025-01-15T10:30:00Z",
    "last_result": "ok (128 cells)",
    "sync_count": 42
  },
  "uptime_seconds": 86400
}
```

字段	类型	描述
`version`	string	应用版本
`name`	string	实例名称
`diameter_peers`	array	连接的 Diameter 对等体及其状态
`active_sessions`	integer	进行中的位置会话数量
`completed_sessions`	integer	已完成的位置会话数量
`cells_loaded`	integer	数据库中的小区数量
`cell_sync`	object	InfluxDB 同步状态
`uptime_seconds`	integer	进程运行时间（秒）

位置服务

POST /api/location

请求 UE 的新位置。

请求体

```json
{
  "imsi": "001010000000001",
  "method": "gnss",
  "timeout": 30000,
  "mme_host": "mme01.epc.mnc380.mcc313.3gppnetwork.org",
  "accuracy": 50
}
```

参数	类型	必需	默认	描述
`imsi`	string	□	--	要定位的 UE 的 IMSI
`method`	string	否	`"cell"`	定位方法：`"cell"`、`"ecid"`、`"gnss"`、`"otdoa"`、`"h`
`timeout`	integer	否	`30000`	超时时间（毫秒）
`mme_host`	string	否	--	目标 MME Diameter 主机。如果省略，则使用连接的对等体。
`accuracy`	integer	否	--	期望的精度（米）。覆盖方法选择。

当提供 `accuracy` 时，方法会自动选择：

精度（米）	选择的方法
<= 50	GNSS
<= 200	OTDOA
<= 500	E-CID
> 500	小区 ID

响应 **(200)**

```
{
  "status": "ok",
  "imsi": "001010000000001",
  "method": "gnss",
  "latitude": 40.7128,
  "longitude": -74.0060,
  "altitude": null,
  "uncertainty": 10.5,
  "confidence": null,
  "source": "gnss",
  "duration_ms": 5230,
  "timestamp": "2025-01-15T10:30:00Z"
}
```


字段	类型	描述
`imsi`	string	UE 标识符
`method`	string	请求的方法
`latitude`	float/null	十进制度的纬度
`longitude`	float/null	十进制度的经度
`altitude`	float/null	高度（米，若可用）
`uncertainty`	float/null	位置不确定性（米）
`confidence`	float/null	置信度百分比
`source`	string	实际使用的定位源
`duration_ms`	integer	耗时（毫秒）
`timestamp`	string	ISO 8601 时间戳

错误响应

状态	原因	描述
400	`"imsi is required"`	缺少 IMSI 参数
404	`"User not found"`	IMSI 不明
404	`"User not connected"`	UE 未附加到网络
422	`"No MME host available for this subscriber"`	没有 MME 路由请求
504	`"Positioning timed out"`	定位方法未能及时完成
500	(变化)	内部错误

GET /api/location

列出最近完成的位置修正。

查询参数

参数	类型	默认	描述
`limit`	integer	`50`	最大结果数量

响应 **(200)**

```
{
  "status": "ok",
  "data": [
    {
      "imsi": "001010000000001",
      "method": "gnss",
      "state": "completed",
      "latitude": 40.7128,
      "longitude": -74.0060,
      "uncertainty": 10.5,
      "source": "gnss",
      "created_at": "2025-01-15T10:29:55Z",
      "completed_at": "2025-01-15T10:30:00Z"
    }
  ],
  "count": 1
}
```

GET /api/location/:imsi

通过 IMSI 获取 UE 的最后已知位置。

路径参数

参数	类型	描述
`imsi`	string	UE 的 IMSI

响应 **(200)**

```
{
  "status": "ok",
  "imsi": "001010000000001",
  "latitude": 40.7128,
  "longitude": -74.0060,
  "altitude": null,
  "uncertainty": 10.5,
  "confidence": null,
  "source": "gnss",
  "timestamp": "2025-01-15T10:30:00Z"
}
```

错误响应

状态	原因
404	`"No location found for IMSI"`
404	`"No completed location for IMSI"`

GET /api/location/:imsi/history

获取 IMSI 的位置修正历史。

路径参数

参数	类型	描述
`imsi`	string	UE 的 IMSI

查询参数

参数	类型	默认	描述
`from`	string	--	时间范围开始（ISO 8601 或日期）
`to`	string	--	时间范围结束（ISO 8601 或日期）
`limit`	integer	`100`	最大结果数量

响应 **(200)**

```
{
  "status": "ok",
  "data": [
    {
      "timestamp": "2025-01-15T10:30:00Z",
      "imsi": "001010000000001",
      "method": "gnss",
      "latitude": 40.7128,
      "longitude": -74.0060,
      "altitude": null,
      "uncertainty": 10.5,
      "confidence": null,
      "source": "gnss",
      "duration_ms": 5230
    }
  ],
  "count": 1
}
```

GET /api/location/:imsi/history/csv

将 IMSI 的位置历史导出为 CSV 文件。

路径参数

参数	类型	描述
`imsi`	string	UE 的 IMSI

查询参数

与 `/api/location/:imsi/history` 相同。

响应 **(200)**

返回带有标题的 CSV 文件下载:

```
timestamp,imsi,method,latitude,longitude,altitude,uncertainty,confide
```

Content-Type: `text/csv` Content-Disposition: `attachment; filename="location_history_<imsi>_<date>.csv"`

小区数据库

GET /api/cells

列出数据库中的所有小区。

响应 **(200)**

```
{
  "status": "ok",
  "data": [
    {
      "cell_id": "001-01-0001-01",
      "latitude": 40.7128,
      "longitude": -74.0060,
      "pci": 100,
      "earfcn": 1300,
      "radius": 500,
      "azimuth": null,
      "height": null,
      "prs_config": null,
      "updated_at": "2025-01-15T10:00:00Z"
    }
  ],
  "count": 1
}
```

GET /api/cells/:id

通过 cell_id 获取单个小区。

路径参数

参数	类型	描述
id	string	小区标识符

响应 **(200)**

```
{
  "status": "ok",
  "data": {
    "cell_id": "001-01-0001-01",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "pci": 100,
    "earfcn": 1300,
    "radius": 500,
    "azimuth": null,
    "height": null,
    "prs_config": {
      "bandwidth": 50,
      "config_index": 0,
      "num_dl_frames": 1,
      "cp_length": null,
      "num_antenna_ports": null
    },
    "updated_at": "2025-01-15T10:00:00Z"
  }
}
```


错误响应

状态	原因
404	`"Cell not found: <id>"`

POST /api/cells

创建一个新的小区。

请求体

```
{
  "cell_id": "001-01-0001-01",
  "latitude": 40.7128,
  "longitude": -74.0060,
  "pci": 100,
  "earfcn": 1300,
  "radius": 500,
  "azimuth": 120.0,
  "height": 30.0,
  "prs_config": {
    "bandwidth": 50,
    "config_index": 0,
    "num_dl_frames": 1,
    "cp_length": "normal",
    "num_antenna_ports": 2
  }
}
```

参数	类型	必需	默认	描述
`cell_id`	string	□	--	唯一的小区标识符
`latitude`	float	□	--	小区纬度（-90 到 90）
`longitude`	float	□	--	小区经度（-180 到 180）
`pci`	integer	否	--	物理小区标识（0-503）
`earfcn`	integer	否	--	E-UTRA 绝对无线电频率信道号
`radius`	integer	否	`1000`	覆盖半径（米）
`azimuth`	float	否	--	天线方位角（度）
`height`	float	否	--	天线高度（米）
`prs_config`	object	否	--	OTDOA 的 PRS 配置
`tac`	integer	否	--	跟踪区域代码（用于 4G CAP 警报广播目标）
`lac`	integer	否	--	区域代码（用于 2G 和 3G CAP 警报广播目标）
`rat`	string	否	--	无线接入技术：`"4g"`、`"3g"` 或 `"2g"`

响应 **(201)**

以与 `GET /api/cells/:id` 相同的格式返回创建的小区。

错误响应

状态	原因
400	`"cell_id is required"`
400	`"latitude and longitude are required"`

PUT /api/cells/:id

更新现有小区。仅更新提供的字段。

路径参数

参数	类型	描述
`id`	string	小区标识符

请求体

任何要更新的小区字段（与 POST 相同，但所有字段都□可选的）。

响应 (200)

返回更新后的小区。

错误响应

状态	原因
404	`"Cell not found: <id>"`

DELETE /api/cells/:id

从数据库中删除小区。

路径参数

参数	类型	描述
`id`	string	小区标识符

���应 (204)

成功时返回空体。

错误响应

状态	原因
404	`"Cell not found: <id>"`

GET /api/cells/nearby

查找靠近地理点的小区。

查询参数

参数	类型	必需	默认	描述
`lat`	float	□	--	搜索中心的纬度
`lon`	float	□	--	搜索中心的经度
`radius`	float	否	`10`	搜索半径（公里）

响应 **(200)**

```
{
  "status": "ok",
  "data": [
    {
      "cell_id": "001-01-0001-01",
      "latitude": 40.7128,
      "longitude": -74.0060,
      "pci": 100,
      "earfcn": 1300,
      "distance_km": 0.523
    }
  ],
  "count": 1
}
```

结果按距离排序（最近的在前）。每个条目包括 `distance_km` 字段。

错误响应

状态	原因
400	`"lat and lon query parameters are required"`

POST /api/cells/sync

触发立即的 InfluxDB 小区同步。

请求体

不需要任何。

响应 (200)

```
{
  "status": "ok",
  "cells_synced": 128
}
```

错误响应

状态	原因
500	`"Sync failed: <reason>"`
503	`"Cell sync service unavailable"`

延迟位置（GMLC）

管理周期性和地理围栏触发的位置会话。有关会话类型和 Diameter 集成的完整详细信息，请参见 GMLC & Le 接口指南。

GET /api/deferred_location

列出所有活动的延迟位置会话。

响应 **(200):**

```
{
  "status": "ok",
  "count": 1,
  "data": [
    {
      "session_id": "a1b2c3d4-e5f6-...",
      "type": "periodic",
      "imsi": "001010000000001",
      "method": "cell",
      "client_name": "rest-api",
      "status": "active",
      "interval_ms": 60000,
      "remaining_reports": 7,
      "total_reports": 10,
      "started_at": "2026-04-09T10:00:00Z",
      "last_fix_at": "2026-04-09T10:03:00Z"
    }
  ]
}
```

POST /api/deferred_location

创建一个新的延迟位置会话。

周期性会话请求体：

```
{
  "type": "periodic",
  "imsi": "001010000000001",
  "method": "cell",
  "interval_seconds": 60,
  "count": 10
}
```

字段	类型	必需	描述
`type`	string	□	`"periodic"`
`imsi`	string	□	订阅者 IMSI
`method`	string	否	定位方法：`cell`、`ecid`、`gnss`、`otdoa`。默认：`cell`
`interval_seconds`	integer	□	修正之间的秒数
`count`	integer	□	要执行的修正总数

触发会话请求体：

```
{
  "type": "triggered",
  "imsi": "001010000000001",
  "method": "cell",
  "event_type": "entering",
  "poll_interval_seconds": 30,
  "max_reports": 0,
  "areas": [
    {
      "type": "circle",
      "center": {"lat": -33.8688, "lon": 151.2093},
      "radius_meters": 500
    }
  ]
}
```

字段	类型	必需	描述
`type`	string	□	`"triggered"`
`imsi`	string	□	订阅者 IMSI
`method`	string	否	定位方法。默认：`cell`
`event_type`	string	□	`"entering"`、`"leaving"` 或 `"being_inside"`
`poll_interval_seconds`	integer	否	位置轮询之间的秒数。默认：30
`max_reports`	integer	否	最大触发报告。`0` = 无限
`areas`	array	□	区域定义列表（圆形或多边形）

响应 **(201):**

```
{"status": "ok", "message": "Periodic session created"}
```

GET /api/deferred_location/:session_id

获取延迟会话的状态。

响应 **(200):**

```
{
  "status": "ok",
  "data": {
    "session_id": "a1b2c3d4-...",
    "type": "periodic",
    "imsi": "001010000000001",
    "status": "active",
    "remaining_reports": 7,
    "total_reports": 10
  }
}
```

DELETE /api/deferred_location/:session_id

取消活动的延迟会话。

响应 **(200):**

```
{"status": "ok", "message": "Session cancelled"}
```

状态	错误
400	缺少或无效的参数
404	会话未找到

CAP 警报

POST /api/cap

提交 CAP XML 警报进行处理。警报被解析，多边形警告区域通过小区数据库解析为 TAC/LAC，警报根据配置被排队等待批准或自动广播。

请求体

```
{
  "xml": "<alert xmlns=\"urn:oasis:names:tc:emergency:cap:1.2\">...</alert>"
}
```

参数	类型	必需	描述
`xml`	string	□	完整的 CAP v1.2 XML 警报文档

响应 **(201)**

```
{
  "status": "ok",
  "data": {
    "id": "a1b2c3d4-e5f6-...",
    "status": "pending",
    "source": "http_post",
    "received_at": "2025-01-15T10:30:00Z",
    "matched_cells": 42,
    "tacs": [100, 101],
    "lacs": [5001],
    "mcc": "001",
    "mnc": "01",
    "broadcast_params": {
      "message_id": 4370,
      "repetition_period": 30,
      "num_broadcasts": 10,
      "message_text": "Tornado Warning...",
      "event": "Tornado Warning",
      "severity": "Extreme",
      "urgency": "Immediate"
    }
  }
}
```

当 `require_approval` 为 `true` 时，`status` 字段为 `"pending"`，当自动批准时为 `"sent"`。

错误响应

状态	原因
400	`"xml field is required"`
422	解析错误详细信息

GET /api/cap

列出所有状态的警报（待处理、活动、历史）。

响应 **(200)**

```
{
  "status": "ok",
  "data": {
    "pending": [...],
    "active": [...],
    "history": [...]
  }
}
```

GET /api/cap/:id

通过 ID 获取单个警报。

路径参数

参数	类型	描述
`id`	string	警报 UUID

响应 **(200)**

返回警报对象。

错误响应

状态	原因
404	`"Alert not found: <id>"`

PUT /api/cap/:id

批准或拒绝待处理的警报。

请求体

```
{
  "action": "approve",
  "operator": "operator1"
}
```

参数	类型	必需	描述
`action`	string	□	`"approve"` 或 `"reject"`
`operator`	string	否	��计跟踪的操作员名称（默认为 `"unknown"`）

响应 **(200)**

返回更新后的警报对象。

错误响应

状态	原因
400	`"action must be 'approve' or 'reject'"`
404	`"Alert not found: <id>"`

错误响应格式

所有错误响应遵循以下结构：

```
{
  "status": "error",
  "reason": "人类可读的错误描述"
}
```

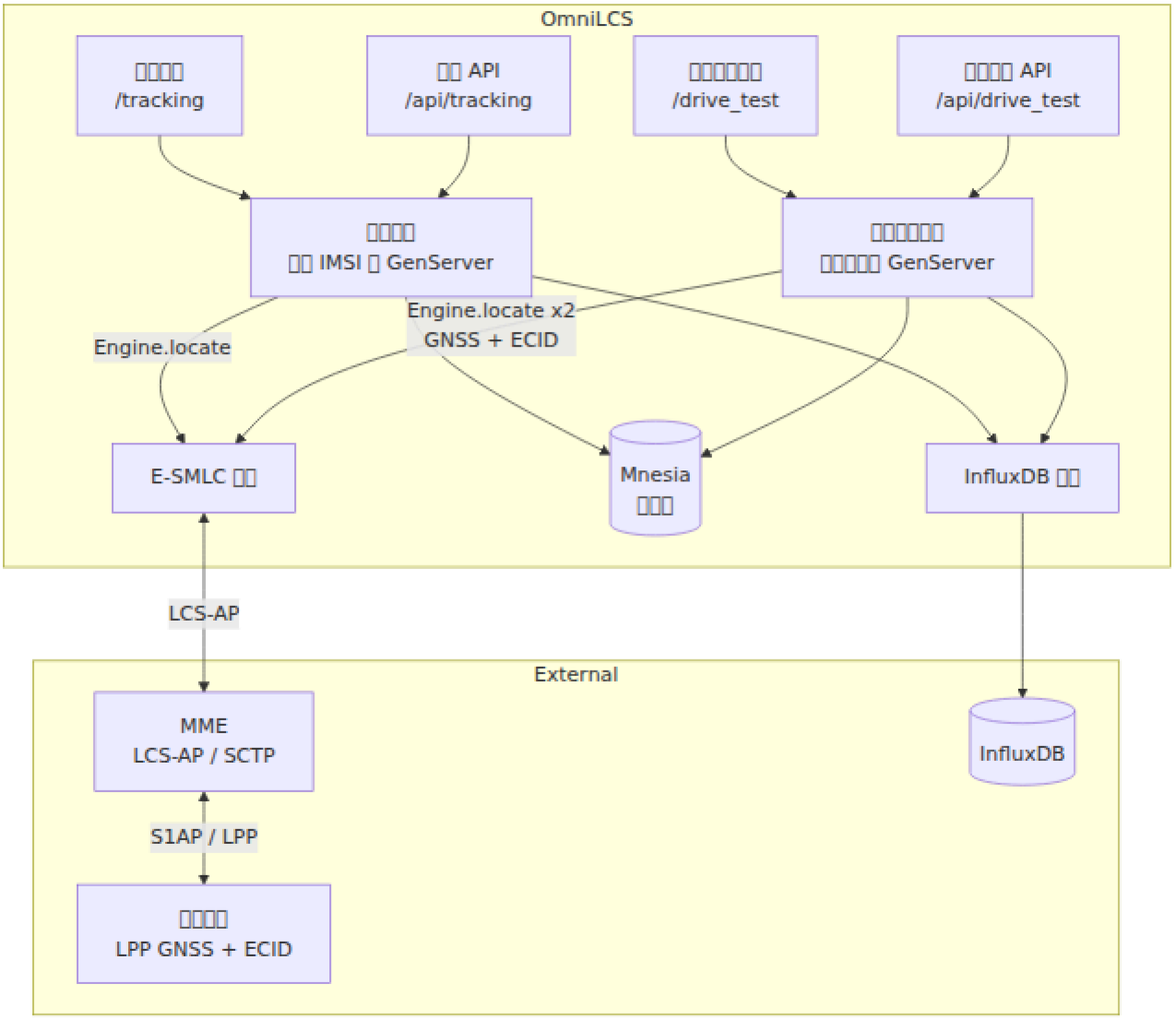
常见的 HTTP 状态代码：

代码	含义
200	成功
201	创建成功（小区创建）
204	无内容（小区删除）
400	错误请求（缺少或无效的参数）
404	未找到（未知资源）
422	无法处理的实体（有效请求但无法满足）
500	内部服务器错误
503	服务不可用
504	网关超时（定位超时）

订阅者跟踪与虚拟驾驶测试

OmniLCS 提供两种定期位置监控功能：订阅者跟踪用于单个用户设备的持续监控，虚拟驾驶测试用于结合 GPS 定位和信号质量测量的多用户设备活动。

架构



订阅者跟踪

订阅者跟踪定期使用选定的定位方法定位单个用户设备。结果保存在 Mnesia 中，并通过 Web UI 和 REST API 提供。

Web UI

在侧边栏中导航到 订阅者跟踪。输入 IMSI、轮询间隔（最小 5 秒），并选择定位方法（小区、E-CID、GNSS、OTDOA）。点击 开始观察。

活动观察以表格形式显示，包含 IMSI、方法、间隔、定位次数、设备状态（在线/离线/错误）、最后定位时间和最后结果。点击一行查看定位历史。使用 导出 **CSV** 或 导出 **KML** 按钮下载数据。

REST API

方法	路径	描述
`GET`	`/api/tracking`	列出活动跟踪会话
`POST`	`/api/tracking`	开始跟踪。主体：`{"imsi": "...", "method": "gnss", "interval": 30}`
`GET`	`/api/tracking/:imsi`	获取某个 IMSI 的跟踪历史
`DELETE`	`/api/tracking/:imsi`	停止跟踪某个 IMSI
`GET`	`/api/tracking/:imsi/export/csv`	下载 CSV 导出
`GET`	`/api/tracking/:imsi/export/kml`	下载 KML 导出

设备状态

跟踪会话处理设备在线/离线转换：

状态	条件	行为
在线	`Engine.locate` 成功	定位与坐标存储
离线	没有连接的 MME 或用户设备不可达	记录错误，继续轮询会话
错误	由于其他原因定位失败	记录错误，继续轮询会话

会话在出现错误时不会停止。它会继续以配置的间隔进行轮询，直到明确取消。

虚拟驾驶测试

虚拟驾驶测试运行多 IMSI 活动，将 **GNSS** 定位（用于 GPS 坐标）与 **E-CID** 测量（用于 RSRP、RSRQ、服务小区、PCI、EARFCN）结合。这□信号质量测量的物理驾驶测试等效。

工作原理

在每个时间间隔，活动：

1. 对于活动中的每个 IMSI（最多 4 个并发）：
    - 通过 MME 向用户设备发送 LPP **GNSS** 请求以获取 GPS 坐标
    - 向用户设备发送 LPP **E-CID** 请求以获取信号测量
    - 将结果合并为单个测量记录
2. 将每个测量存储在 Mnesia 中
3. 记录到 InfluxDB（如果已配置）
4. 通过 PubSub 将结果广播到 Web UI

Web UI

□□□□□□□□ □□□□□□□

1. □□ □□□□□□□□“CBD□□□□”□
2. □□ **IMSIs**□□□□□□□□□□□□
3. □□ □□□□□ 10 □□
4. □□ □□□□

□□

列	描述
时间	定位时间戳
IMSI	目标用户设备
纬度/经度	来自 GNSS 的 GPS 坐标
PCI	来自 E-CID 的物理小区 ID
RSRP	参考信号接收功率（用户设备报告）
RSRQ	参考信号接收质量（用户设备报告）
GNSS	GNSS 状态（正常/错误）
ECID	E-CID 状态（正常/错误）

RSRP 值以颜色编码：绿色（强），黄色（中），红色（弱）。

REST API

方法	路径	描述
`GET`	`/api/drive_test`	列出活动活动
`POST`	`/api/drive_test`	开始活动。主体：`{"name": "...", "imsis": ["..."], "interval": 30}`
`GET`	`/api/drive_test/:id`	获取活动测量。查询：`?limit=200&imsi=filter`
`DELETE`	`/api/drive_test/:id`	停止活动
`GET`	`/api/drive_test/:id/export/csv`	下载包含所有测量的 CSV
`GET`	`/api/drive_test/:id/export/kml`	下载带有信号质量颜色编码的 KML

示例 API 使用

开始活动：

```
curl -sk -X POST https://omnilcs:8445/api/drive_test \
  -H "Content-Type: application/json" \
  -d '{
    "name": "CBD Coverage Test",
    "imsis": ["001010000000001", "001010000000002"],
    "interval": 30
  }'
```

获取测量：

```
curl -sk https://omnilcs:8445/api/drive_test/<campaign_id>?
limit=100
```

导出 **KML**:

```
curl -sk
https://omnilcs:8445/api/drive_test/<campaign_id>/export/kml -o
coverage.kml
```

E-CID 测量

E-CID 方法通过 LPP（3GPP TS 36.355）直接从用户设备请求 RSRP、RSRQ 和 UE Rx/Tx 时间差。此方法无论 eNB LPPa 支持与否均有效，因为用户设备在本地执行测量。

测量字段

字段	描述	来源
`rsrp`	参考信号接收功率	用户设备测量（映射值，约 -44 到 -140 dBm）
`rsrq`	参考信号接收质量	用户设备测量（映射值，约 -3 到 -19.5 dB）
`pci`	物理小区 ID	服务小区标识符
`earfcn`	E-UTRAN 绝对无线电频率信道号	服务小区频率
`cell_global_id`	小区全球标识（PLMN + 小区 ID）	服务小区
`ue_rx_tx_time_diff`	用户设备 Rx-Tx 时间差	传播延迟估计

API 响应示例

```
{
  "status": "ok",
  "method": "ecid",
  "imsi": "001010000000001",
  "ecid_measurements": {
    "measurements": [
      {
        "pci": 373,
        "earfcn": 1825,
        "cell_global_id": {
          "cell_id": 4000,
          "plmn": {"mcc": "001", "mnc": "01"}
        },
        "rsrp": 40,
        "rsrq": 25,
        "ue_rx_tx_time_diff": 19
      }
    ]
  }
}
```

导出格式

CSV

以逗号分隔的值，带有标题行。包括所有测量列：IMSI、时间戳、坐标、信号测量和状态字段。适合导入电子表格工具或分析脚本。

KML

与 Google Earth / Google Maps 兼容。每个定位成为一个标记，包含：

- 坐标（纬度/经度/高度）
- 时间戳
- 包含 IMSI、RSRP、RSRQ、PCI 的描述
- 按 RSRP 信号强度（绿色/黄色/红色）颜色编码的图标

- 每个 IMSI 的轨迹 LineString 连接所有定位

在 Google Earth、Google Maps、QGIS 或任何 KML 兼容的 GIS 工具中打开。

InfluxDB 集成

当在 `runtime.exs` 中配置 InfluxDB 连接时，订阅者跟踪和虚拟驾驶测试都会将测量记录到 InfluxDB。

写入的测量

InfluxDB 测量	来源	字段
`subscriber_tracking`	跟踪会话	latitude, longitude, altitude, device_status
`drive_test`	驾驶测试活动	latitude, longitude, altitude, rsrp, rsrq, serving_pci, uncertainty

标签

标签	描述
`imsi`	目标用户设备 IMSI
`method`	使用的定位方法
`campaign_id`	驾驶测试活动标识符（仅限驾驶测试）

持久性

所有跟踪和驾驶测试数据存储在 Mnesia 的 disc_copies 表中，在应用程序重启后仍然有效。

Mnesia 表	键	内容
`:mnesia_tracking_history`	`{imsi, monotonic_time}`	单个跟踪定位
`:mnesia_tracking_config`	`imsi`	活动跟踪会话配置
`:mnesia_drive_test_measurements`	`{campaign_id, imsi, monotonic_time}`	驾驶测试测量
`:mnesia_drive_test_config`	`campaign_id`	活动配置

历史记录会自动修剪为每个 IMSI 1,000 条（跟踪）或每个活动 10,000 条（驾驶测试）。

OmniLCS Web 界面指南

OmniLCS 控制面板——一个通过 HTTPS 在 **443** 端口提供的实时 Web 界面。它基于 Phoenix LiveView 构建，提供无需页面重新加载的实时更新视图。所有页面每 2-3 秒自动刷新。

页面概述

导航栏按以下顺序呈现页面：

□□	□□	□□
□□□	`/dashboard`	□□□□□□□□□
□□	`/location`	□□□□□□□□
SLs □□	`/sls`	SLs (LCS-AP) □□□□
□□□□	`/cells`	□□□□□□□□□
□□□	`/map`	□□□□□□□□□□□□□□□
Diameter	`/diameter`	Diameter □□□
GMLC / Le	`/gmlc`	GMLC Le □□□□□□□□□□
□□□	`/send_broadcast`	□ 2G/3G/4G □□□□□□
□□□	`/broadcasts`	□□□□□□□□
CBC 2G	`/cbc`	2G CBSP □□
CBC 3G	`/cbc3g`	3G SABP □□
CBC 4G	`/cbc4g`	4G SBC-AP □□□□□□
CAP □□	`/cap`	CAP □□□□□□□□□□
□□□	`/application`	OTP □□□□□□□□□□□□
□□	`/configuration`	□□□□□□□□□□□
□□	`/log`	□□□□□□□□□□

仪表板

路径：`/dashboard`
刷新：每 2 秒

仪表板提供整个 OmniLCS 系统的高层次操作概述。

统计卡片

顶部有四个摘要卡片：

卡片	描述
活动会话	正在进行的位置会话数量
总请求	自启动以来处理的位置请求总数
成功率	完成请求与总请求的百分比
平均响应时间	平均定位持续时间（毫秒）

最近位置请求

显示 20 个最近位置会话的表格，列包括：

列	描述
IMSI	UE 标识符
方法	定位方法（小区、E-CID、GNSS、OTDOA、混合）
MME	发起请求的 MME 主机
时间	请求时间戳（HH:MM:SS）
状态	颜色编码的徽章（活动、完成、错误、超时）

连接的对等体

显示处于“连接”状态的 SLs (LCS-AP) 和 Diameter 对等体列表，包括：

- 对等体主机名
- 接口类型（SLs 或 Diameter）

系统状态

子系统健康指示灯：

指示器	绿色	黄色	红色
SLs 接口 (LCS-AP)	至少建立 1 个 MME SCTP 关联	--	无 MME 连接
Diameter 服务	至少 1 个对等体连接	--	无对等体连接
小区数据库	小区已加载	0 个小区	--

位置

路径：`/location`
刷新：每 2 秒

交互式位置服务测试界面，用于手动请求 UE 位置。

方法选择器

切换按钮选择定位方法：

方法	描述
小区	基于小区 ID，速度最快，精度最低
E-CID	使用 LPPa 通过 eNB 测量的增强小区 ID
GNSS	通过 LPP 的 GPS/GNSS，精度最高
OTDOA	从 PRS 测量进行多边定位
混合	结合方法并提供后备

每种方法显示一个描述面板，解释其工作原理。

请求表单

- **IMSI** 输入：输入要定位的 UE 的 IMSI（例如，`999990000000001`）
- 请求位置按钮：异步发送位置请求

请求作为后台任务运行。请求进行时，旋转指示器显示。

会话计数

摘要徽章显示：

- 总会话
- 活动会话
- 完成会话
- 错误计数
- 收到的结果

会话历史表

按创建时间（最新优先）排序的所有会话：

列	描述
IMSI	UE 标识符
方法	定位方法
MME	目标 MME 主机
时间	会话创建时间
持续时间	完成时间（毫秒或秒）
状态	活动、完成、错误、超时

完成的会话展开显示：

- 坐标：纬度和经度
- 不确定性：位置不确定性（米）
- 置信度：置信度百分比
- 来源：实际使用的定位来源
- 地图：嵌入的 OpenStreetMap 显示位置

错误会话展开显示错误原因。

通知

通知弹出框显示：

- 请求已发起（信息）
- 位置已获取（成功，带坐标）
- 位置获取失败（错误，带原因）

通知在 5 秒后自动消失。

SLs 接口

路径：`/sls`
刷新：每 2 秒

监控 E-SMLC 和连接的 MMEs 之间的 SLs (LCS-AP) 接口。显示 SCTP 关联状态和连接的 MME 对等体。

接口状态

显示 LCS 应用协议 (LCS-AP) 接口状态:

- 协议参考 (3GPP TS 29.171)
- 监听配置 (PPID 29, 端口 9082)
- 连接的 MME 对等体数量
- 绿色/红色指示器

统计

卡片	描述
连接的 MMEs	活动 SCTP 关联到 MMEs 的数量
总关联	SCTP 关联总数

MME □□□□□

□□□□□ MME SCTP □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□: `/cells`
□□: □ 2 □

□□□□□□□ ID □□□ OTDOA □□□□□□□□□□□□□□□□□□□□□□□□ Mnesia□□□□□□□□□□□□□□□□□

□□□□

□□	□□
□□□□	□□□□□□□□□□
□□□□	□□ InfluxDB □□□□□□□□□□
□ InfluxDB □□	□□□□□□ InfluxDB □□□□□
□□□□ / □□□□	□□□□□□□□□□□□□□
□□□□	□□□□□□□□□□□□□□□
□□□□	□□□□□□□□□□□□

□□

□□□ ID□PCI□EARFCN□□□□□□□□□□□□□□□

□□□□

□□□□□

列	描述
小区 ID	唯一的小区标识符
PCI	物理小区身份（LTE/NR）或主扰码（UMTS）
EARFCN	无线频率信道编号
纬度	小区纬度（十进制度）
经度	小区经度（十进制度）
RAT	无线接入技术：GSM、UMTS、LTE 或 NR
来源	数据来源徽章：华为（琥珀色，XLSX 导入）、**InfluxDB**（绿色，同步）或手动（蓝色）。徽章下方显示导入时间戳。
操作	编辑和删除按钮

导入小区

点击 导入小区 打开一个具有两步工作流程的模态：

第一步：供应商选择

选择要导入的小区数据的 NMS 供应商：

供应商	导入格式
华为	U2020 XLSX 导出或 CSV
诺基亚	NetAct RAN 导出 CSV
爱立信	ENM WCDMA 导出 CSV
中兴	UMS 小区导出 CSV
其他 **/ JSON**	JSON 数组粘贴

第二步：上传

每个供应商提供适当的上传界面。

华为 **XLSX**（推荐）：拖放或选择一个或多个华为 U2020 小区报告 `.xlsx` 文件。支持多文件上传——例如，同时上传 UMTS 和 LTE 报告。文件在上传完成后自动导入。

RAT 类型（GSM、UMTS、LTE 或 NR）根据以下内容自动检测：

- 工作簿中的表名（例如，名为 "UMTS"、"LTE"、"GSM"、"NR" 的表）
- 规划数据表中的列标题（例如，`PScrambCode` 对于 UMTS，`DlEarfcn` 对于 LTE，`NRPCI` 对于 NR，`BCCH` 对于 GSM）

解析器读取 **Sheet1** 标签（带坐标的规划数据）并提取：

RAT	导入的关键字段
GSM	小区 ID、小区名称、LAC、BCCH、BSIC、经度、纬度、天线高度、方位角
UMTS	小区 ID、小区名称、LAC、SAC、RAC、UARFCN 下行、扰码、经度、纬度、天线高度、方位角
LTE	小区 ID、小区名称、PCI、TAC、DL EARFCN、经度、纬度、天线高度、小区半径、带宽
NR	小区 ID、小区名称、NR PCI、TAC、NR DL EARFCN、经度、纬度、天线高度、小区半径、带宽

CSV（所有供应商）：上传从所选 NMS 导出的 CSV。第一行必须□列标题。根据所选供应商格式，列映射□自动的。

JSON：粘贴小区对象的 JSON 数组。必填字段：`cell_id`。可选字段：`pci`、`earfcn`、`latitude`、`longitude`、`lac`、`tac`、`rat`。

重复处理

导入与现有条目具有相同小区 ID 的小区将覆盖现有数据。这意味着重新导入相同文件□安全的，并将更新任何更改的值。

数据持久性

所有小区数据存储在 Mnesia 中，使用 `disc_copies` 存储。小区在应用程序重启后持久存在，无需重新导入或重新同步。

添加 / 编辑小区表单

表单字段以网格形式组织：

字段	描述
小区 ID	唯一标识符（编辑时禁用）
PCI	物理小区身份 (0-503)
EARFCN	无线频率信道编号
纬度	十进制度
经度	十进制度
天线端口	1、2 或 4
PRS 带宽 (RBs)	6、15、25、50、75 或 100
PRS 配置索引	0-4095
CP 长度	正常或扩展

删除确认

点击小区行上的“Del”会在删除之前显示确认模态。

小区地图

路径：`/map`

交互式地图显示来自小区数据库的所有小区，使用 Leaflet 在深色主题的 CartoDB 基础地图上。小区根据其 RAT 类型绘制为彩色标记。

标记颜色

RAT	颜色
GSM (2G)	橙色
UMTS (3G)	蓝色
LTE (4G)	绿色
NR (5G)	紫色
未知	灰色
选中	琥珀色

标记在较低缩放级别下使用 Leaflet MarkerCluster 进行聚类。点击标记会显示一个弹出窗口，显示小区的详细信息：小区 ID、名称、技术、LAC/SAC/TAC、PCI、EARFCN 和坐标。

广播区域选择

使用右上角的多边形或矩形绘图工具选择地理区域内的小区。当小区被选择时：

- 底部的选择栏显示所选小区的数量和唯一的 LAC(s)、SAC(s) 和 TAC(s)
- 出现一个 发送广播到选定小区 按钮，链接到发送广播页面，LACs、SACs 和 TACs 会预填充

发送广播表单会根据所选区域代码自动选择适当的目标网络：

- 存在 LACs → 启用 2G (CBSP)
- 存在 SACs → 启用 3G (SABP)
- 存在 TACs → 启用 4G (SBc-AP)

Diameter

路径：`/diameter`
刷新：每 2 秒

监控剩余 Diameter 接口（SLg 和其他）的 Diameter 对等体连接。E-SMLC 到 MME 的 SLs 接口使用原生 LCS-AP 通过 SCTP，并在仪表板页面单独监控。

接口状态

两个面板显示接口健康状况：

接口	描述
SLg 接口 (TS 29.172)	GMLC 到 MME 通过 DRA, 应用 ID 16777264
其他 Diameter 接口	根据配置的其他 Diameter 应用

每个面板显示一个绿色/红色指示器和描述。

对等体摘要

顶部显示连接和断开连接的对等体计数。

对等体表

列	描述
对等体	Diameter 主机身份
领域	Diameter 领域
IP 地址	传输地址 (protocol://ip:port)
状态	连接或断开连接徽章

点击对等体行展开以显示详细信息：

详细信息	描述
连接发起	OmniLCS □否发起连接
传输	SCTP 或 TCP
产品名称	远程对等体的 Diameter 产品名称
广告应用	对等体支持的应用 ID

CBC 2G

□□: `/cbc`
□□: □ 2 □

□□ 2G □□□□□□ CBSP □□□

□□□□

□□ CBSP □□□□□

- □□□□ (3GPP TS 48.049)
- □□□□□
- □□□□□□□□□
- □□/□□□□□

□□□□

□□	□□
□□□ BSCs	□□□ `:connected` □□□□□□
□□□	□□□□□□□□□□□□□□□□□□
□□□□	□□□□□□□□□□ CBSP □□□□

□□□ BSCs □□

□□□□□ CBSP □□□□□□□□□□

- □□□ IP □□□□□
- □□ ID
- □□□□□
- □□□□□□

最近 CBSP 消息面板

显示 20 条最近的 CBSP 消息的表格：

列	描述
时间	消息时间戳（HH:MM:SS）
对等体	源对等体的连接 ID
消息	可读的消息类型（例如，“WRITE REPLACE COMPLETE”）
类型	颜色编码的徽章（完成为 OK，失败为 FAIL，错误为 ERR，其他为 INFO）

消息通过对 `cbsp:connections` 和 `cbsp:messages` 主题的 PubSub 订阅接收。

CBC 3G

路径：`/cbc3g`
刷新：每 2 秒

监控通过 Iu-BC 接口的 3G 小区广播的 SABP 接口（3GPP TS 25.419）。

接口信息

显示 SABP 接口状态：

- 协议参考 (3GPP TS 25.419)
- 监听端口号
- 连接的 RNC 数量
- 绿色/红色指示器

���□□□

□□	□□
□□□ RNCs	□□□ `:connected` □ RNC □□□□□
□□□	□□□□□□□□□□□□□□□□□□
□□□□	□□□□□□□□□ SABP □□□□

□□□ RNCs □□

□□□□□ SABP □□□□□□□□□□

- □□□ IP □□□□□
- □□ ID
- □□□□□
- □□□□□□

□□ SABP □□□□

□□ 20 □□□□ SABP □□□□□□

□	□□
□□	□□□□□□HH:MM:SS□
□□□	□ RNC □□□ ID
□□	□□□□□□□□□□□“WRITE REPLACE COMPLETE”□“RESTART INDICATION”□
□□	□□□□□□□□□□ OK□□□□ FAIL□□□□ ERR□□□□ INFO□

□□□□□ `sabp:connections` □ `sabp:messages` □□□ PubSub □□□□□

□□□□

□□: `/send_broadcast`
□□: □□

□□□□□□□□□□□□□□□□□ 2G (CBSP)□3G (SABP) □ 4G (SBc-AP) □□□□□□□□□□□□□□□□□□

统一视图显示所有 2G (CBSP)、3G (SABP) 和 4G (SBc-AP) 网络的活动小区广播。广播从发送时开始跟踪，直到停止或过期。视图每 3 秒自动刷新，并在广播状态变化时实时更新。

摘要卡片

卡片	描述
总活动	所有网络的活动广播总数
2G (CBSP)	活动的 2G 广播
3G (SABP)	活动的 3G 广播
4G (SBC-AP)	活动的 4G 广播

使用任何广播行上的停止按钮在其各自网络上取消广播。

CBC 4G

路径：`/cbc4g`

刷新：每 3 秒

功能齐全的 4G 小区广播管理界面，用于编排、发送、监控和停止 SBC-AP 广播。

统计卡片

卡片	描述
连接的 MMEs	已建立的 SBC-AP SCTP 关联数量
活动广播	当前活动的 4G 广播数量
总发送	历史中的广播总数

编排广播表单

字段	描述
消息文本	广播消息内容
消息 ID	16 位 CB 消息标识符（例如，4370 表示 CMAS）
序列号	16 位序列号
MCC	移动国家代码（例如，“313”）
MNC	移动网络代码（例如，“380”）
TACs	逗号分隔的跟踪区域代码
警告类型	无、地震、海啸、地震 + 海啸、测试、其他
重复（秒）	广播重复之间的秒数
广播次数	广播消息的总次数

点击“发送广播”会对消息进行编码（GSM 7 位），构建一个写替换警告请求 PDU，并将其发送给所有连接的 MMEs。

连接的 **MMEs** 面板

显示 SBC-AP SCTP 关联列表：

- MME 主机名
- IP 地址、端口和 SCTP 关联 ID
- 连接状态徽章（已连接 / 状态名称）

活动广播面板

当前活动广播的表格：

□	□□
MSG ID	□□□□□□□□□□□□□□□□□□□□0x1112□
SERIAL	□□□□□□□□□□□□
MESSAGE	□□□□□□□□□
STATUS	□□□□□□□□□□□□□□□□
ACTION	□□□□□□□□□□□□/□□□□□□□□

□□“□□”□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□ 100 □□□

□	□□
MSG ID	□□□□□□□□□□□□□□□
SERIAL	□□□□□□□□□□□□□
MESSAGE	□□□□□□□□□
TIME	□□□□□□HH:MM:SS□
STATUS	□□□□□□□

□□□□□□□□ `cbc:state` □ `cbc:connections` □□□ PubSub □□□□□

CAP □□

□□: `/cap`
□□: □ 3 □ + □□ PubSub

来自外部警报机构的 CAP（通用警报协议）警报的操作员批准工作流程。有关配置、馈送轮询和多边形解析的完整详细信息，请参见 CAP 警报操作指南。

统计卡片

卡片	描述
待批准	等待操作员操作的警报数量（当 > 0 时高亮琥珀色）
活动广播	当前正在广播的警报数量
总处理	历史中的总警报数量
批准模式	当前设置：“手动”（require_approval: true）或“自动”

待处理警报面板

仅在 CAP 配置中 `require_approval` 为 `true` 时可见。

□	□□
TIME	��□□□□□
EVENT	□□□□□□□□□□□“□□□□□”□
SEVERITY	□□□□□□□□□□□□□□□□□□□□□
CELLS	□□□□□□□□□□□□□
STATUS	□□□□□□□□□□□
ACTIONS	□□□□□□□□□□□

□□ □□□□□□□□□□□□□□□□□□□□□□□ TACs/LACs□□□ ID □ PLMN□

□□ □□□□ SBC-AP (4G)□SABP (3G) □ CBSP (2G) □□□□□□

□□ □□□□□□□□□□□□□□□□□□

□□□□□□

□	□□
EVENT	□□□□□□
MSG ID	CB □□□□□
TACs	□□□□□□□□
STARTED	□□□□□□
STATUS	□□□□□□□□□□□□□□□□□

警报历史面板

列	描述
TIME	发送或接收时间戳
EVENT	警报事件类型
SEVERITY	警报严重性
CELLS	匹配的小区数量
TACs/LACs	目标区域代码
STATUS	已发送（绿色）、已拒绝（红色）或失败（红色）

当收到新的待处理警报时，会出现一个通知弹出框。

GMLC / Le 接口

路径：`/gmlc`
刷新：每 3 秒 + 实时 PubSub

监控和管理 GMLC Le 接口和延迟位置会话。有关完整配置和 Diameter 协议的详细信息，请参见 GMLC & Le 接口操作指南。

统计卡片

卡片	描述
活动会话	总活动的延迟位置会话
定期	活动的定期位置会话数量
地理围栏	活动的触发/地理围栏会话数量

授权 **LCS** 客户端面板

列	描述
NAME	客户端身份（与 Diameter AVPs 匹配）
TYPE	LCS 客户端类型（紧急服务、增值服务等）
ALLOWED METHODS	此客户端可以请求的定位方法
RATE LIMIT	每分钟请求数（或“无限制”）

活动延迟会话面板

列	描述
SESSION ID	截断的会话 UUID
TYPE	会话类型徽章：定期（绿色）或触发（紫色）
IMSI	订阅者 IMSI
METHOD	定位方法
CLIENT	请求 LCS 客户端的名称
PROGRESS	定期：完成/总报告。触发：剩余报告或“活动（无限制）”
LAST FIX	最近位置修正的时间
ACTIONS	取消按钮以终止会话

会话历史面板

显示最近 50 个完成或取消的会话，包括会话 ID、类型、IMSI、状态（完成/取消）和开始时间。

内置页面

这些页面由控制面板框架提供。

应用程序 (`/application`)

OTP 应用程序资源查看器，显示：

- 正在运行的应用程序及其进程树
- 每个进程的内存使用情况
- 消息队列长度

配置 (`/configuration`)

运行时配置查看器，显示：

- 所有应用程序环境变量
- 按应用程序分组

日志 (`/log`)

实时日志查看器，显示：

- 来自 `ControlPanel.Logger` 后端的日志消息
- 可按日志级别过滤
- 自动滚动并具备暂停功能