Mobile Positioning & Analytics Services
Open Positioning Framework Specification

Release Version 1.0 – Aug. 2013

MPAS Technical Committee

MPAS Technical Committee Members

Changi Airport Group	Joe CHIU
Changi Airport Group	LEONG Yuh Khee
Changi Airport Group	Mokanam GOPAL
Changi Airport Group	Jennifer LIN
Changi Airport Group	Spencer LIM
Cisco Systems	Tjie Seng NJAUW
Cisco Systems	MinSe KIM
Cisco Systems	Jerry LUCAS
IDA	Danny KOIK
IDA	TOH Chian Leng
IDA	KOH Wee Loong
StarHub	Sophia ZHOU Yu Xuan
Ruckus Wireless*	TING See Ho

**Ruckus Wireless acquired YFind Technologies in July 2013*

Contents

1. SCOPE **5**

REFERENCES **6**

2. TERMINOLOGY AND CONVENTIONS **7**

2.1 CONVENTIONS **7**

2.2 ABBREVIATIONS **7**

3. INTRODUCTION **8**

4. ARCHITECTURE MODEL **11**

4.1 OPF ARCHITECTURE DIAGRAM **11**

4.2 SSP MODULE DESCRIPTION **12**

4.2.1 DI Connector 12

4.2.2 Information Parser 12

4.2.3 Location Services 12

4.2.4 ESP/CSP Connector 13

4.2.5 SSP Connector 13

4.2.6 Storage 13

4.2.7 Authentication 14

5. SSP API **15**

5.1 API WITH DI **16**

5.1.1 Delivery Method 16

5.1.2 Message Type 16

5.1.3 Message Flow 17

5.1.4 Message Format 19

5.2 API WITH ESP/CSP **20**

5.2.1 Delivery Method 20

5.2.2 Message Type 20

5.2.3 Message Flow 21

5.2.4 Message Format 22

5.3 API WITH SSP **23**

5.3.1 Delivery Type 23

5.3.2 Message Type 23

5.3.3 Message Flow 23

5.3.4 Message Format 24

APPENDIX A. CHANGE HISTORY **25**

A.1 APPROVED VERSION HISTORY **25**

A.2 DRAFT/CANDIDATE VERSION <CURRENT VERSION> HISTORY **25**

APPENDIX B. CISCO MSE SOAP REQUEST EXAMPLES **26**

B.1 REQUEST AND RESPONSE FOR LOCATION OF MOBILE STATIONS **26**

APPENDIX C. JSON REQUEST EXAMPLES **27**

C.1 REQUEST AND RESPONSE FOR LOCATION OF MOBILE STATIONS **27**

C.2 REQUEST AND RESPONSE FOR HISTORY LOCATION OF MOBILE STATIONS **27**

C.3 REQUEST AND RESPONSE FOR MOBILES WITHIN AN AREA **28**

Figures

Figure 1: MPAS OPF Architecture 11
Figure 2: SSP Building Blocks 12
Figure 3: Message Delivery Method – DI with SSP 16
Figure 4: Message Flow – SSP request for location data of mobile stations 17
Figure 5: Message Flow – SSP request for location data of an area 18
Figure 6 Communication Type between SSP and ESP/CSP 20
Figure 7: Message Flow – ESP/SSP request for location data of mobile stations 21
Figure 8: Location service flows between SSPs 23

Tables

Table 1 Message Type between SSP and DI 16
Table 2 Message Format of API between SSP and DI 19
Table 3 Message Type between SSP and ESP/CSP 20
Table 4 Message Format of API between SSP and ESP/CSP 22
Table 5 Message Type between SSPs 23
Table 6 Message Format of API between SSPs 24

1. Scope

The scope of Open Positioning Framework (OPF) specifications document is to define the architecture and communication protocols between the OPF modules and their services. This will ensure consistency in implementing Mobile Positioning & Analytics Services (MPAS).

Reference architecture includes:

- Architecture Diagram
- Functional blocks
- Reference Points / Interfaces
- Messages of APIs
- Message flows and message format

The above architecture is based on the requirements listed in the MPAS CFC document [MPAS CFC Public Document] and is jointly established by the MPAS Technical Commitee. This OPF document is subject to continuous evolvement. Please refer to the latest version for implementation.

References

[Cisco MSE API Spec]	Cisco MSE API Specification Guide – Context Aware Service of MSE, Release 7.3
[MPAS CFC Public Document]	Mobile Positioning and Analytics Services Call-for-Collaboration Public Document, IDA May 7, 2012

2. Terminology and Conventions

2.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All main sections are normative and appendixes are informative.

2.2 Abbreviations

AP	Access Point
API	Application Programming Interface
CSP	Customer Service Provider
DI	Detection Infrastructure
DM	Digital Map
ESP	Enterprise Service Provider
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
JMS	Java Message Service
JSON	JavaScript Object Notation
LA	Location Analytics
LBE	LBS Enabler
LBS	Location-based Service
LP	Location Position
MPAS	Mobile Positioning & Analytics Services
OPF	Open Positioning Framework
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SSP	Shared Service Provider
UTC	Coordinated Universal Time
XML	Extensible Markup Language

3. Introduction

The first generation of location-based services based on cell ID positioning and GPS achieved only moderate success because of inaccurate positioning technology and lack of demand for mobile data services due to inflexible and sub-optimal user interfaces such as SMS or WAP over GPRS. Today, smart phones are typically equipped with a variety of sensors such as GPS, Wi-Fi, Bluetooth and accelerometer which allows accurate location positioning to be derived, both outdoors and indoors. This sets the stage for more accurate and effective LBS.

The purpose of MPAS is to develop, promote and provide a platform that offers accurate, easily accessible and competitively priced location positioning and analytics services to enterprises and consumers scalable across multiple sectors such as, but not limited to, Tourism, Hospitality, Retail, and F&B sectors. Location detection is a key element in MPAS. Through the provision of accurate location-based services, a range of highly personalized and location-aware application, services and analytics can be realized.

Open Positioning Framework (OPF) refers to the LBS interface framework providing open access to information and specification such as, but not limited to, Application Protocol Interfaces (APIs) and XML data format for establishing end-to-end service integration across Venue Operator's (VO) Detection Infrastructure (DI) provider, Shared Service Provider (SSPs), Enterprise Service Providers (ESPs) and Consumer Service Providers (CSPs).

SSP is the backbone of OPF that retrieves raw location information from DIs, pre-process the information, interface with other LBS/LBE for big data analysis, and provide location and analytic service to ESP/CSP. Additionally, SSP also provides anonymous location sharing to other SSPs based on mutual agreement.

DI is an essential part in OPF as it is the source of position information. DI is a general term of any detection equipment installed in venues which are capable of supporting indoor positioning services to a VO such as, but not limited to, wireless access points (APs) and cellular indoor antenna.

ESP/CSP is the direct consumer of location services. ESP refers to a company that provides services for VOs or other enterprises/businesses. Examples of ESP services are such as, but not limited to, crowd management, predictive analysis, location-based campaigning system, asset/resource tracking and operational optimization. The enterprise services, provided by ESPs, leverages one or more of the available shared services from the SSP, to provide enterprise services, applications and reports, such as, but not limited to, rental-yield optimization, heat maps, footfall analysis, asset and resource management to enterprises.

CSP refers to a company that develops and provides location-based Mobile Service (M-Service) for consumers. Examples of CSP services are such as, but not limited to, in-mall navigation services and location-based promotions, car finder, family tracker. The consumer services, provided by CSPs, leverages one or more of the available shared services from the SSP, to provide mobile services such as, but not limited to, Navigation, Finder, and Tracker Services.

Venue Operators will be able to obtain vital information and in-depth analysis on their venues and customers to optimize their business and operations. ICT companies will acquire enhanced MPAS capabilities in developing and delivering more innovative and accurate LBS applications to business owners, enterprises and consumers. Business owners will be able to optimize reach to their customers, improve customer engagement and raise service levels. They will be able to generate insights to customer's behaviour enabling new lines of business and revenue models. Consumers will enjoy personalized, engaging and time-saving experience at venues through delivery of context-aware information and services.

OPF can be used to support the following services.

- Finder Services: Services that locate a specific location.
- Tracker Services: Services that locate and monitor one or multiple Mobile Devices.
- Navigation Services: Services that locates the mobile device and provides way finding functionality to the destination.
- Geo-Fencing Services: Services that can be offered in a geographically-defined boundary - as in a radius around a venue, store or points within the venue.
- Heat Map Services: Services that capture a snapshot of a venue's crowd density and represent it in a visual format.
- Footfall Analytics Services: Services that captures human traffic flow within a venue over a defined period of time and represent it in a visual format.

Supported SSP Location Services

- Real Time location services
- Historical location detection services
- Area-based location servicess

Supported Protocols

- Application Layer Protocols: SOAP/JSON/HTTP/HTTPS/FTP
- Presentation Layer Protocols: XML/JMS
- Transmission Layer Protocols: UDP/TCP/IP

Supported DIs

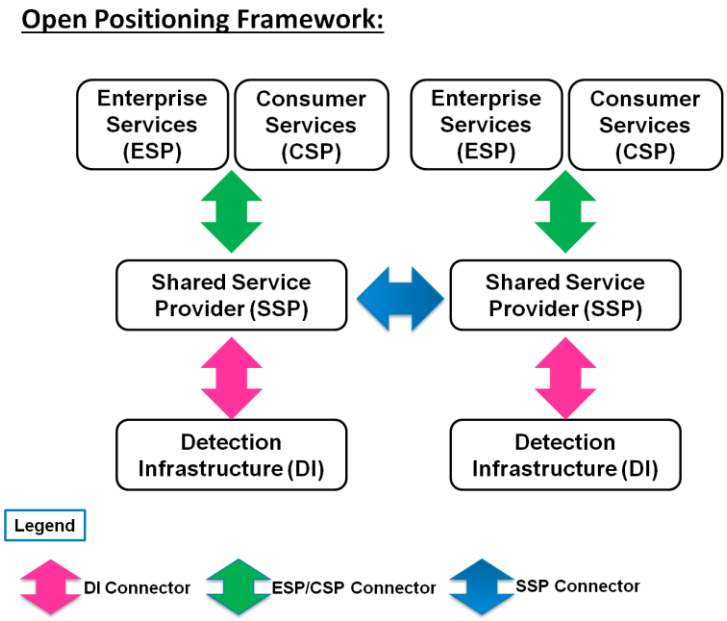
- WiFi
- Sensor

4. Architecture Model

SSP architecture and module is defined in this section.

4.1 OPF Architecture Diagram

This sub-section defines the OPF architecture (Figure 1), which comprises of DIs, SSP, and ESP/CSP. The SSP has 3 sets of connectors/interfaces DI-SSP, SSP-SSP and SSP-ESP/CSP.



SSP processes the raw location data from DIs to useful location information consumed by ESP/CSP to provide location services or analytics. A single SSP may have limited functions, so it can rely on other SSPs or even ESP/CSPs to provide more sophisticated and customized services.

Raw location information stored at SSP is only valid for 24 hours. Information older than 24 hours should be move out to a side storage requested by ESP/CSP or be purged to ensure that the SSP operates in an optimized enviornment.

Figure 1: MPAS OPF Architecture

Detection Infrastructure (DI) is a network of sensors that is able to detect the position of mobile stations. It may or may not have a control gateway that manages the position information from the sensors. SSP functions as the focal point where location information from the sensor network could be retrieved, regardless of control gateway or standalone sensor and processed by the SSP.

ESP/CSP could be a data center that serves various enterprises, a platform that host and handle multiple services, or a simple application that integrates requests from enterprise and customers. To receive SSP location service, ESP/CSP must use the messages defined in this document and protocols recommended. ESP/CSP could have a table that list down the services provided by each SSP so that it will approach the correct SSP for different location information services. ESP/CSP is required to provide SSP the source of the request, that is, from which application the request comes from. This source information will help SSP filter requests to classified logs.

4.2 SSP Module Description

Figure 2 defines typical SSP building blocks/modules that comply with OPF. Detail of each component will be introduced in the following sub-sections.

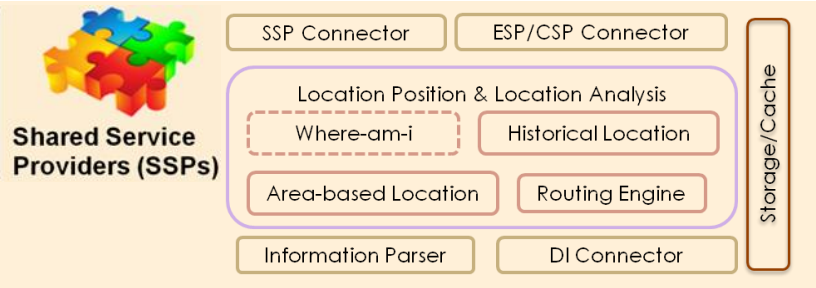


Figure 2: SSP Building Blocks

4.2.1 DI Connector

DI connector is a module that enables communications between the various types of DIs. It receives instruction from one of the four location services modules (i.e. Where-am-I, Historical location, Area-based location and Routing Engine) and triggers a request to the DI. The requested includes instantaneous location of a list of mobile stations, historic location of a mobile station, and mobile stations connected to the APs within a specified area. If there is a control point to the DI network, the DI connector will only send the request to the control point. Otherwise the DI connector is required to send the request directly to the related sensors. Regardless of DI network, the DI connector will be the single contact point for all replies from DI. The request will then be forward to Information Parser for pre-processing.

4.2.2 Information Parser

Information Parser is a translator that preprocesses the raw messages coming from DI, translates the DI information from various protocols into the OPF format.

4.2.3 Location Services

Location Services layer provides the essential location position and analytic services including real time location pinpoint, historical location retrieve, and area-based location service which provides the location of all mobile stations within a defined area.

4.2.3.1 Real Time *Where-am-I* Service

This is a real time location service that provides instantaneous position of a mobile station. If ESP/CSP request real time location of a specific mobile station with unique identity, the reply will also include the unique identity of the mobile stations. Otherwise mobile station identity is confidential to ESP/CSP/SSP.

4.2.3.2 Historical Location Service

Historical Location Service provides location data of mobile stations within a time range T1 to T2. The time range (T1 and T2) must be within 24 hrs. Similar to real time location service, unique identity of the mobile station is considered confidential and only given when provided in incoming request from ESP/CSP/SSP.

4.2.3.3 Area-Based Location Service

Area-based location service provides an aggregated users (or crowd densitity) within a period (e.g. 10min) mapped over a specified areas. This service can be used by ESP/CSP in visual representation of crowd density for an indoor area so that business owners can trigger actionable responses tha can be deployed to areas that fit the necessary conditions. To use the area-based service, ESP/CSP should provide the SSP a boundary of the area (polygon). The SSP will return the location information of all mobile stations within the specified area.

4.2.4 ESP/CSP Connector

ESP/CSP Connector uses a synchronous secured interface for external applications to connect to the platform for retrieving information that is required from their ESP/CSP. One of the key considerations for this connector is to enable easy integration for 3rd party developers, using an interface that is open-access and robust. The Platform will adopt commonly used and reliable protocol, e.g. JSON over REST. Comparing with other protocol such as SOAP (Web Service), JSON offers a more efficient and simpler implementation option, which is an ideal choice for loosely coupled, distributed system. This connector should support simple authentication/handshake with ESP/CSP/SSP.

4.2.5 SSP Connector

SSP Connector is an interface offering authentication between two or more SSPs allowing the exchange of anonymous location data.

4.2.6 Storage

Storage is a module for storing temporary data that require high throughput and high-speed access. This module will be used to store 24 hours of raw and processed data to faciliate the responses to various API calls from ESP/CSP. Upon expiry of the data (beyond 24hr), the data could be exported into file system before purging from the engine. These exported files can be SFTP to another external server(s).

4.2.7 Authentication

Mutual authentication SHALL be supported between SSPs. Server authentication SHALL be supported between SSP and ESP/CSP. Server authentication MAY be supported between SSP and DI.

5. SSP API

There are three primary types of APIs that is used to communicate with other layers: API with DI, API with ESP/CSP, and API with SSP.

The API between SSP and DI is defined for the SSP to retrieve raw location data from physical detection sensors (on-stie). A basic DI could be a WiFi mesh network. In such a scenario, the SSP will query the WiFi access point (APs) for the mobile stations associated to it via the API between SSP and DI. If WiFi has intelligent APs or WiFi network has a central control point that can perform some basic data processing, the SSP could request an advanced location data via the API between the SSP and DI. The advanced location data includes historical location of a mobile station, real-time or historical location of mobile stations within an area, floor layout, etc. If the WiFi network only connects to dumb APs (i.e. without internet connectivity), it can only provide basic location positioning related data. The basic location positioning related data will be the name or identity of the AP or Tag, the RSSI of mobile stations detected by the AP or Tag. SSP will process the raw data to provide an approximate location of the mobile station.

The API between SSP and ESP/CSP is defined for the ESP/CSP to request for location service from a SSP. As explained in previous sections, ESP/ CSP could be a server, a platform, or even a company that provides enterprise level services to businesses or consumers. ESP/CSP is the contact point of end users. It translates the positioning request from end users and query SSP via API of SSP and ESP/CSP. The positioning request could be where-am-I query, historical trace of a mobile user, people density of some popular areas, etc. SSP either provides a processed response of these positional requests to ESP/CSP, or half-processed location information for ESP/CSP to futher intergrate into more sophisticated usage. For example, where-am-I request from end user will get a direct answer of the coordinate of the mobile station from SSP-ESP/CSP API. However, a floor heat map request from end user will receive a mobile position list of the requested floor. In this scenario, ESP needs to count the number of mobiles in each interested area and plot the integrated heat map on UI. A simple ESP/CSP may just display the location response from SSP to end customer. An intelligent ESP/CSP could use the response from SSP as a location component to build up more innovative services.

API of SSPs is a communication tunnel between SSPs. This interface is required due to the functional limitation of a single SSP: the service provided by a single SSP is limited; therefore the users that can be supported by a single SSP is limited; the data storage of a single SSP is also limited. Therefore, if large deployments, bespoked analysis or location services are required, multiple SSPs may need to collaborate, exchange information and jointly develop a solution. API of SSPs is similar to the API of SSP and ESP/CSP except for the handshaking messages. Detail of the APIs will be introduced in the following sub-sections.

5.1 API with DI

API with DI offers communication methods between SSPs and DIs to exchange raw location information.

5.1.1 Delivery Method

Messages SHOULD be delivered with one the following methods.

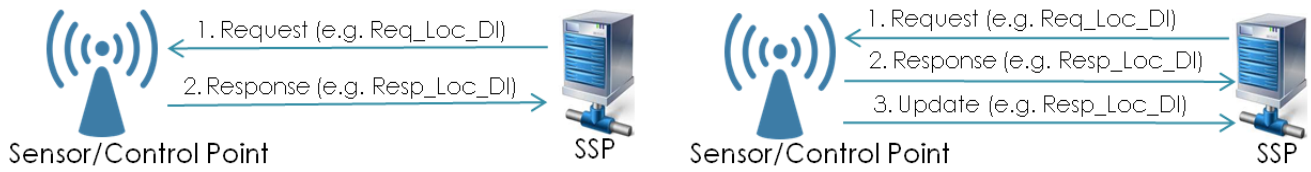


Figure 3: Message Delivery Method – DI with SSP

Figure 3 illustrates two types of delivery methods. Left method is Master-Slave model where Sensor/Control Point responses the raw location data upon request from SSP. Right method is an intelligent model that Sensor/Control Point is able to provide both response and subsequent updates for the request from SSP. Update message has the same format as response message. DI connectors from SSP MUST support at least one of these message delivery methods in Figure 3. Detail message type is explained in the following sub-sections.

5.1.2 Message Type

API between SSP and DI MUST support the messages in **Table 1**.

Table 1: Message Type between SSP and DI

Message	Req_Loc_DI	Resp_Loc_DI	Req_Area_DI	Resp_Area_DI
Sensor/Control Point-> SSP		Y		Y
SSP->Sensor/Control Point	Y		Y	
Comment	Location request	Location response	Location request of area	Location response of area

Req_Loc_DI and Resp_Loc_DI is a pair of messages for SSP to request raw location data of one or more mobile stations. Req_Area_DI and Resp_Area_DI is a pair of messages for SSP to request the number of mobile stations connected within an area.

5.1.3 Message Flow

Based on the message type between SSP and DI, there are two flows of this API. One is SSP request for location of mobile stations, and the other is SSP request for statistic data of an area.

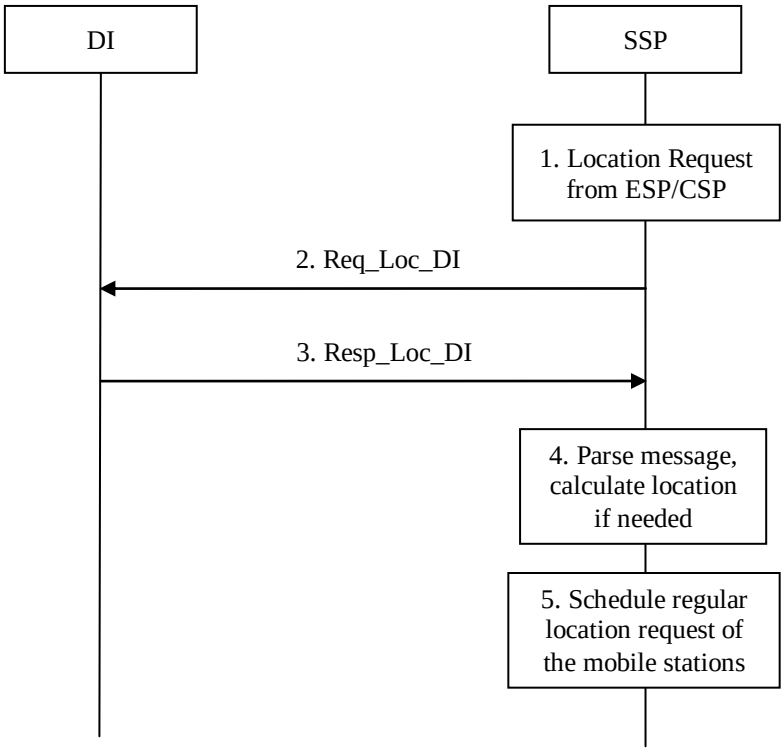


Figure 4: Message Flow – SSP request for location data of mobile stations

Figure 4 is the message flow that SSP request location information for mobile stations. It contains the following steps:

1. SSP receives the request from ESP/CSP regarding real-time location of one or more mobile stations.
2. SSP initiates a location request (Req_Loc_DI) towards DI that associated with the SSP.
3. DI fetches the location of requested mobile stations and response to SSP using Resp_Loc_DI message.
4. SSP parse the response(s) from DI. If the message contains exact (x,y), venue and floor information, no further process is required. Otherwise SSP MUST triangulate the location information and map it to the floor plan that ESP/CSP can understand.
5. SSP schedules a time interval to regularly request for location information for the requested mobile

NOTE: if DI cannot find the location of a mobile station within its infrastructure, it should return an error message to SSP. SSP then will either pass through the error message to ESP/CSP or stop the regular check for this mobile station.

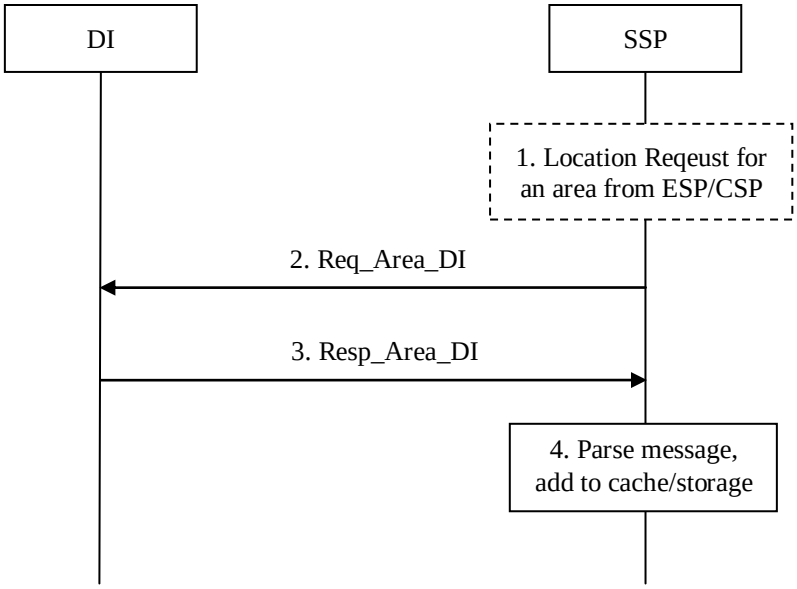


Figure 5: Message Flow – SSP request for location data of an area

Figure 5 is the message flow that SSP request area location information for mobile stations. It contains the following steps:

1. SSP receives the request from ESP/CSP regarding mobile stations within an area. This step is optional.
2. SSP initiates a location request (Req_Area_DI) towards DI that associated with the SSP.
3. DI fetches the location of requested mobile stations and response to SSP using Resp_Area_DI message.
4. SSP parse the response(s) from DI. If the message contains exact (x,y) venue and floor information, no further process is needed. Otherwise SSP MUST triangulate the location information and map it to the floor plan that ESP/CSP can understand.

NOTE: Resp_Area_Statistic_DI contains identity information of the mobile stations connected to the sensors within the area. However, the identity information in this message is not eligible to be released to ESP/CSP/SSP.

5.1.4 Message Format

This section explains the detail format of each message of API between SSP and DI. The lines in *Italic* within the quotation marks are example messages.

Table 2: Message Format of API between SSP and DI

Message	Contained Info.
Req_Loc_DI	• SSP ID • A list of ○ Identity of mobile stations *"123456* *00:E1:A2:03:A4:05* *00:E1:02:03:B4:C6"*

Resp_Loc_DI	• A list of
	  ○ ID of Mobile Station
	  ○ Sensor identity and RSSI, (x,y,floor) as alternative
	• Timestamp
	"00:01:02:03:04:05 (1.00, 2.23, 2)
	00:01:02:03:04:06 (7.60, 2.90, 6)
	10:00"
	OR
	"00:01:02:03:04:05 E0:01:D2:A1:67:22 RSSI_1
	00:01:02:03:04:05 E0:01:D2:A1:67:23 RSSI_2
	00:01:02:03:04:05 E0:01:D2:A1:67:24 RSSI_3
	10:00"

Req_Area_DI	• SSP ID • Polygon coordinate of an area or sensors within the area • T_Start • T_End *"123456* *((1.0, 2.0) (3.0, 4.0) (5.0, 6.0) (7.0,8.0) 2)* *10:00* *10:15"* OR *"123456* *E0:01:D2:A1:67:22 E0:01:D2:A1:67:23* *10:00* *10:15"*

Resp_Area_DI	• A list of
	  ○ (x,y,identity of mobile station) OR
	  ○ Sensor identity, and identity of a list of mobile stations connected to the sensor
	• Timestamp
	"1.0 2.2 00:01:02:03:04:05
	1.0 1.4 00:01:02:03:04:06
	2.1 2.4 00:01:02:03:04:07
	2.0 3.2 00:01:02:03:04:01
	10:25"
	OR
	"*E0:01:D2:A1:67:22 00:01:02:03:04:01 00:01:02:03:04:02 00:01:02:03:04:03*
	E0:01:D2:A1:67:23 00:01:02:03:04:04 00:01:02:03:04:05 00:01:02:03:04:06
	10:25 "

5.2 API with ESP/CSP

5.2.1 Delivery Method

Communication between SSP and ESP/CSP SHOULD follow the Master-Slave method as shown in Figure 6. Only when requested, SSP will response to the ESP/CSP. However, for real-time application like "Where-am-I" service, SSP MAY update ESP/CSP real-time location of a mobile station regularly until ESP/CSP close the session or time out. Update message uses the same format as Response message. This delivery method is NOT a MUST for implementation but depends on individual cases. Detail message type is explained in the following sub-sections.

Figure 6: Communication Type between SSP and ESP/CSP

5.2.2 Message Type

API between SSP and DI MUST support the messages in Table 3.

Table 3: Message Type between SSP and ESP/CSP

Message	Req_Loc_SSP	Resp_Loc_SSP	Req_Hist_Loc_SSP	Resp_Hist_Loc_SSP	Req_Area_SSP	Resp_Area_SSP
ESP/CSP-> SSP	Y		Y		Y	
SSP- >ESP/CSP		Y		Y		Y
Comment	Location request	Location response	Historical location request	Historical location response	Request of Mobiles within an area	Response of Mobiles within an area

Req_Loc_SSP and Resp_Loc_SSP is a pair of messages for ESP/CSP to request location data of one or more mobile stations from SSP. Req_Hist_Loc_SSP and Resp_Hist_Loc_SSP is a pair of messages for ESP/CSP to request the historical location of mobile stations. Historical location could be used for ESP to plot track and trace or foot fall diagram on the map. Req_Area_SSP and Resp_Area_SSP is a pair of messages to retrieve the location of all mobile stations within an area for a certain time period. SSP will provide all locations of connected mobile stations within the area defined in Req_Area_SSP. ESP could calculate the statistics based on information from Resp_Area_SSP to plot customized mobile density of the area. As a service supporting statistic usage, no mobile station credentials will be given in Resp_Area_SSP message.

5.2.3 Message Flow

Based on the message type between SSP and ESP/CSP, there is a single type of flow between SSP and ESP/CSP, which is the request-response flow shown in Figure 7.

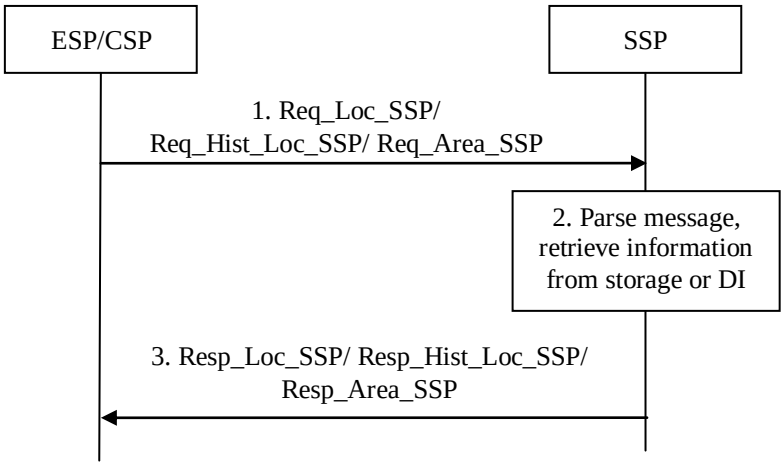


Figure 7: Message Flow – ESP/SSP request for location data of mobile stations

Figure 7 is the message flow that ESP/CSP requests location services from SSP. It contains the following steps:

1. ESP/CSP sends a request to SSP with respect to real-time, historical or area-based location services.
2. SSP parses the message from ESP/CSP and retrieve information from its storage. If required, SSP will initiate a location request (Req_Loc_DI or Req_Area_DI) towards DI to get the requested data.
3. SSP replies ESP/CSP the real-time, historical or area location services with the response message.

5.2.4 Message Format

This section explains the detail format of each message of API between SSP and ESP/CSP. The lines in *Italic* within the quotation marks are example messages.

Table 4: Message Format of API between SSP and ESP/CSP

Message	Contained Info.

Req_Loc_SSP	• Venue ID
	• A list of
	  ○ Identity of mobile stations
	  ○ Application ID
	"takashimaya
	00:01:02:03:04:05 123456
	00:01:02:03:04:06 123458
	……"

Resp_Loc_SSP	• A list of
	  ○ ID of Mobile Station
	  ○ (x,y,floor)
	• Timestamp
	"00:01:02:03:04:05 (1.00, 2.23, 2)
	00:01:02:03:04:06 (7.60, 2.90, 6)
	……
	12.15"

Req_Hist_Loc_SSP	• Venue ID • Identity of mobile station • Application ID • Start Time • End Time *"takashimaya 01:02:03:04:05 123456 11:00 14:00"*

Resp_Hist_Loc_SSP	• Identity of the requested mobile station
	• A list of
	  ○ (x,y,floor)
	  ○ Response Time
	• Timestamp
	"00:01:02:03:04:05
	1.0 2.2 2 11:02
	2.0 3.2 2 11:15
	5.2 2.8 3 12:08
	……
	20:00"

Req_Area_SSP	• Venue ID • T_Start • T_End • Polygon coordinate of an area *"Polygon 10:00 10:15 ((1.0,2.0) (3.0,4.0) (5.0,6.0) (7.0,8.0) 2)"*
Resp_Area_SSP	• A list of ○ (x,y) • Timestamp *"1.0 2.2* *2.0 3.2* *7.2 5.5* *……* *20:15"*

5.3 API with SSP

5.3.1 Delivery Type

Communication between SSPs is the same as SSP with ESP/CSP. SSP_1 needs to receive a request from SSP_2 and complete authentication process before it responses to SSP_2 with access to its location services.

5.3.2 Message Type

API between SSPs is similar to the API between SSP and ESP/CSP except for the authentication message shown in Table 5.

Table 5: Message Type between SSPs

Message	Req_Loc_SSP	Resp_Loc_SSP	Req_Hist_Loc_SSP	Resp_Hist_Loc_SSP	Req_Auc	Resp_Auc
SSP_2-> SSP_1	Y		Y		Y	
SSP_1- >SSP_2		Y		Y		Y
Comment	Location request	Location response	Historical location request	Historical location response	Authentication Information	Authentication Response

5.3.3 Message Flow

The request-response flow between SSPs is shown in Figure 8.

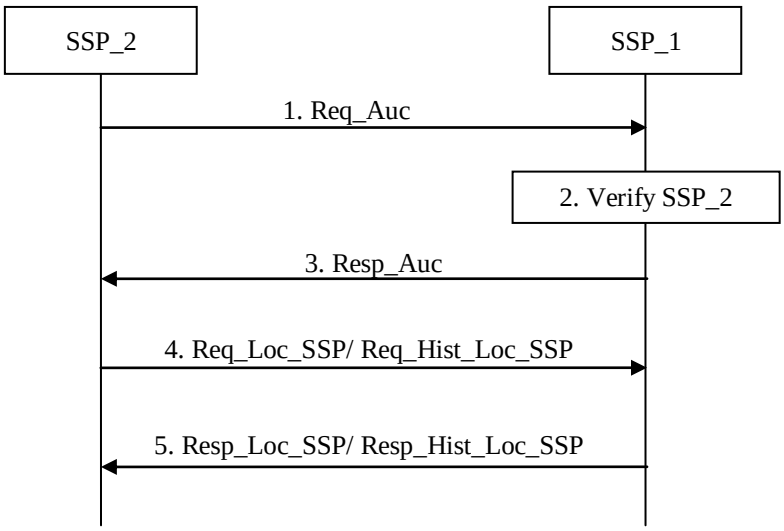


Figure 8: Location service flows between SSPs

Figure 8 is the message flow from SSP_2 requesting location services from SSP_1. It contains the following steps:

1. SSP_2 sends authentication information to SSP_1.
2. SSP_1 verify the eligibility of SSP_2.
3. SSP_1 response to SSP_2 of the success of authentication.
4. SSP_2 sends request to SSP_1 with respect to real-time, historical location services.
5. SSP_1 replies SSP_2 the real-time, historical location services with the response message.

5.3.4 Message Format

This section only defines the message format of authentication between SSPs because API between SSPs is similar to API between SSP and ESP/CSP. The lines in *Italic* within the quotation marks are example messages.

Table 6: Message Format of API between SSPs

Message	Contained Info.
Req_Auc	• SSP_ID • Company name • Access Code
Resp_Auc	• OK or ERROR

Appendix A. Change History

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	Agreed version within IDA, StarHub, YFind

A.2 Draft/Candidate Version <current version> History

Document Identifier	Date	Sections	Description
Draft Versions MPAS-OPF-Spec-v1.0	2013-08-06	All	Initial draft V1.0
			▪
			▪
			▪
			▪
			▪
			▪
			▪

Appendix B. Cisco MSE SOAP Request Examples

Cisco MSE REST/SOAP interface message could refer to **[Cisco MSE API Spec]**.

B.1 Request and Response for Location of Mobile Stations

Example of requesting individual location of a mobile station Req_Loc_DI:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<GetStationLocation xmlns=" http://cisco.com/mse/location">
<AesBusinessSession id="10510"/>
<AesMobileStation macAddress="00:01:02:03:04:05"/>
</GetStationLocation>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response of the MSE stored location of a mobile station, Resp_Loc_DI:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<Response xmlns=" http://cisco.com/mse/location">
<AesMobileStationLocation macAddress="00:01:02:03:04:05"
changedOn="122122111" floorId="10" x="12.1" y="34.6" z="2"/>
</Response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Appendix C. JSON Request Examples

C.1 Request and Response for Location of Mobile Stations

Example of Requesting Where-am-I service Req_Loc_SSP:

```
{
  "apiID": "cspext",
  "apiPwd": "!@#!@#",
  "venueID": "takashimaya",
  "macAddr": "00:0b:88:60:03:a0",
  "action": "whereAmI"
}
```

Response of Where-am-I service Resp_Loc_SSP:

```
{
  "trxId": "0001230001230",
  "statusCode": "0",
  "statusMessage": "SUCCESS",
  "Data": [
             "x": "196.1",
             "y": "202.8",
             "changedOn": "20130701231058"
  ]
}
```

C.2 Request and Response for History Location of Mobile Stations

Example of Requesting History Location service Req_Hist_Loc_SSP:

```
{
  "apiID": "cspext",
  "apiPwd": "!@#!@#",
  "venueID": "takashimaya",
  "macAddr": "00:0b:88:60:03:a0",
  "startT": "20130701100000"
  "endT": "20130701152330"
  "action": "historyLoc"
}
```

Response of History Location service Resp_Hist_Loc_SSP:

```
{
  "trxId": "0001230001230",
  "statusCode": "0",
  "statusMessage": "SUCCESS",
  "Data": [
             "x": "196.1",
             "y": "202.8",
             "changedOn": "20130701102002",
"x": "198.2",
             "y": "207.0",
```

```
            "changedOn": "20130701114606",
            "x": "190.1",
            "y": "212.8",
            "changedOn": "20130701145117"
  ]
  "timestamp": "20130718T09:59:00+08:00",
}
```


C.3 Request and Response for Mobiles within an Area

Example of Requesting Area-Based Location service Req_Area_SSP:


```
{
  "apiID": "cspext",
  "apiPwd": "!@#!@#",
  "venueID": "takashimaya",
  "startT": "20130701100000",
  "endT": "20130701101000",
  "area": [
            "x1": "196.1",
            "y1": "202.8",
"x2": "221.2",
            "y2": "207.0",
            "x3": "190.1",
            "y3": "219.8",
            "floor": "2",
 ]
  "action": "areaLoc"
}
```


Response of Area-Based Location service Resp_Area_SSP:


```
{
  "trxId": "0001230001230",
  "statusCode": "0",
  "statusMessage": "SUCCESS",
  "Data": [
            "x": "196.1",
            "y": "202.8",
"x": "198.2",
            "y": "207.0",
            "x": "190.1",
            "y": "212.8"
  ]
  "timestamp": "20130718T09:59:00+08:00",
}
```