



Cisco MSE REST API Guide, Release 8.0

Release 8.0
Aug, 2014

Last Updated: April, 2015

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-32457-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco MSE REST API Guide, Release 8.0

© 2015 Cisco Systems, Inc. All rights reserved.



Preface 1

CHAPTER 1

Introduction 1-1

Representational State Transfer 1-1

REST Resources 1-2

CHAPTER 2

Authentication 2-1

CHAPTER 3

MSE Location APIs 3-1

Maps API 3-1

Real Time Location APIs 3-2

Location History APIs 3-3

Notification APIs 3-5

CHAPTER 4

MSE API Object Descriptions 4-1

maps 4-2

mapscount 4-2

mapsinfo 4-3

mapsimage 4-3

mapsimagesource 4-4

locationclients 4-4

locationclientscount 4-5

locationtags 4-5

locationtagscount 4-6

locationrogueaps 4-6

locationrogueapscount 4-7

locationrogueclients 4-7

locationrogueclientscount 4-8

locationinterferers 4-8

locationinterfererscount 4-9

locationhistoryclients 4-9

locationhistoryclientscount 4-10

- locationhistorytags 4-10
- locationhistorytagscount 4-11
- locationhistoryrogueaps 4-11
- locationhistoryrogueapscount 4-12
- locationhistoryrogueclients 4-12
- locationhistoryrogueclientscount 4-13
- locationhistoryinterferers 4-13
- locationhistoryinterfererscount 4-14
- notifications 4-14

CHAPTER 5

Object Schema Details 5-1

- API Related Objects Schema Details for Map 5-1
- API Related Object Schema Details for Real Time and Historical Location 5-50
- API Related Object Schema Details for Notification API Object Schema Details 5-110

CHAPTER 6

Sample Examples 6-1

- Sample PresenceEventTrigger Notification Subscription 6-1
- Sample MovementEventTrigger Notification Subscription 6-2
- Sample Location Streaming NotificationTrigger Subscription 6-2



Preface

This preface describes the audience, organization, and conventions of the Cisco MSE API Guide, Release 8.0 and contains the following sections:

- [Audience](#)
- [Convention](#)
- [Assumptions](#)
- [Related Documentation](#)

Audience

This document is intended for developers who wish to implement the MSE RESTful client support on the Mobility Services Engine application. The audience is assumed to be familiar with RESTful Web services and the HTTP protocol.

Convention

This guide uses the following conventions:



Note

Means reader take note



Tip

Means the following information will help you solve a problem.



Caution

Means reader be careful. In this situation, you might do something that can result in equipment damage or loss of data.

Assumptions

HTTP Version

The Reports REST API embraces several aspects of the HTTPS standards, including some headers, HTTP verbs, and status codes. The REST API uses status codes that are only supported in the HTTP/1.1 version. Therefore, it is expected that the client consuming this API complies with these HTTP/1.1 features.

Read Only API

This REST API is used for reading location data and providing notification when specific conditions are met. It supports the HTTPS GET, PUT, and DELETE methods.

Related Documentation

Follow these steps to obtain documentation updates and related documentation for the context-aware API:

- Open your web browser and browse to the Cisco Developer Community Context-Aware API web page on Cisco.com at this URL: <http://developer.cisco.com/web/contextaware/home>.
- Double-click **Resources**.
- Click the desired link to open or download the desired component.

Cisco MSE Developer Community

For any additional questions on usage of the APIs or details of the information that can be retrieved from MSE. The Cisco MSE Developer Community Web portal is located at [Mobility Services API Developer Center](#).



Introduction

This chapter describes the Cisco Mobility Services Engine (MSE) Representational State Transfer (REST) API which is used to retrieve location information from the Mobility Services Engine.

This chapter contains the following sections:

- [Representational State Transfer, page 1-1](#)
- [REST Resources, page 1-2](#)

Representational State Transfer

Representational State Transfer (REST) is a design architecture that focuses on resources for a specific service and their representations. A resource representation is a key abstraction of information that corresponds to one specific managed element on a server. A client sends a request to a server element located at a Uniform Resource Identifier (URI) and performs operations with standard HTTPS methods, such as GET, POST, PUT, and DELETE. This provides a stateless communication between the client and server where each request acts independent of any other request and contains all necessary information to complete the request.

Mobility Services Engine provides a Representational State Transfer (REST) API. The REST API is useful for developers and administrators who aim to integrate the functionality of the MSE via the standard Hypertext Transfer Protocol (HTTPS).

The REST API is implemented as a Web service front-end component on the MSE gateway. The API allows applications to retrieve location data from the gateway using HTTP(s) GET requests using the URIs (Uniform Resource Identifier) specified in this document).

The data provided by the REST API can be formatted as XML (eXtensible Markup Language) and JSON (JavaScript Object Notation).

The benefits of the REST API are:

- **Broad client support**—Any programming language, framework, or system with support for HTTP protocol can use the API.
- **Self descriptive**—Client applications require minimal knowledge of the virtualization infrastructure as many details are discovered at runtime.
- **Resource-based model**—The resource-based REST model provides a natural way to manage a virtualization platform.

REST Resources

A REST resource is an abstraction of a piece of information, such as a single data record, a collection of records, or even dynamic real-time information. Each resource in the MSE REST API is identified by URI and is accessed using the standard HTTPS methods (GET, PUT, and DELETE).

This section contains the following topics:

- [Content-Type, page 1-2](#)
- [Accept Header Preference, page 1-2](#)
- [Character Set Encoding, page 1-3](#)

Content-Type

The report data provided by the REST API can be formatted as XML and JSON. The data format is based on the values of the incoming HTTP accept header or the “outputType” query string parameter. If no specific data format type is found in the incoming request, then the REST API defaults the response data format and content-type to “text/csv”.

The response content-type may be specified using either the HTTP request “Accept” header or the “outputType” parameter. The “outputType” parameter has precedence over the “Accept” header. If neither the “outputType” parameter nor the “Accept” header (or if the Accept header found matches */*), the content-type defaults to “text/csv”.

The following “outputType” parameter values can be used:

- For JSON: “json”
- For XML: “xml”

When using the HTTP “Accept” header, the following can be used:

- For JSON: application/json
- For XML: text/xml OR application/xml

Accept Header Preference

When the “Accept” header contains more than one accept type supported by the REST API, the following order of preference is followed:

CSV -> JSON -> XML

For example,

Accept: text/xml, text/csv, application/json -- CSV is selected

Accept: application/json, text/csv -- CSV is selected

Accept: text/xml, application/json – JSON is selected

Accept: application/xml, text/csv, application/json – CSV is selected

Accept: text/html, */*, text/xml – XML is selected

Accept: text/html, */* – default to CSV due to the */*

Accept: text/html, image/jpeg, * -- default to CSV due to *

Accept: text/html, image/jpeg, text/plain -- 406 error is returned.

Character Set Encoding

This Reports REST API defaults all the responses to the “UTF-8” charset encoding, irrespective of the requested charset sent.



Authentication

MSE APIs uses basic authentication scheme to authenticate the API request. Each API Request, contains the API access credentials in the Authorization header. The API credentials are generated by the MSE Administrator using the Prime Infrastructure user interface.

To access this page, log into Prime Infrastructure UI and choose **Services > Mobility Services >** choose the name of the mobility services engine **> Systems > Users**.



Note

The user credentials have read and write permissions associated with it. User credentials without write permissions can only perform GET operations on the REST API resources. In addition, certain APIs will only return the configurations created by the requesting user.

Authorization Header

The Authorization header is constructed as follows:

- Username and password are combined into a string "username:password".
- The resulting string literal is then encoded using Base64.
- The authorization method, a space and the string "Basic" is then put before the encoded string.

For example, if the username is 'Aladin' and the password is 'sesame open', then the header is formed as follows:

- Authorization: Basic QxhZGluOnNlc2FtIG9wZW4=

If the API request does not have the Authorization header or if the credentials are incorrect, then the MSE will send a HTTP 401 Not Authorized response code containing the authenticate header as below:

WWW-Authenticate: Basic realm="MSE API Service"





MSE Location APIs

The location APIs contains the following four categories:

- [Maps API, page 3-1](#)
- [Real Time Location APIs, page 3-2](#)
- [Location History APIs, page 3-3](#)
- [Notification Event APIs, page 3-5](#)

Maps API

The Maps API provides details about the Map data. This provides summary of all your managed systems on campuses, buildings, outdoor areas, floors, zones, access points, dimensions, map images, GPS markers, obstacles, location filter regions, and excitors.

All Map API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for Maps API. The response is determined based on the Accept HTTP header.

- XML—Use Accept: `application/xml`
- JSON—Use Accept: `application/jsonsupports`

This section contains [Resources, page 3-1](#).

Resources

- `/api/contextaware/v1/maps`
See the [maps](#) for more details.
- `/api/contextaware/v1/maps/count`
See the [mapscount](#) for more details.
- `/api/contextaware/v1/maps/info/{campusName}/{buildingName}/{floorName}`
See the [mapsinfo](#) for more details
- `/api/contextaware/v1/maps/image/{campusName}/{buildingName}/{floorName}`
See the [mapsimage](#) for more details
- `/api/contextaware/v1/maps/imagesource/{imageName}`

See the [mapsimagesource](#) for more details

Real Time Location APIs

The real time location APIs provides access to the location of devices that are currently being tracked by the MSE.

All Map API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for real time location API. The response is determined based on the Accept HTTP header.

- XML—Use `Accept: application/xml`
- JSON—Use `Accept: application/jsonsupport`

This section contains the following topics:

- [Paging, page 3-2](#)
- [Sorting, page 3-2](#)
- [Resources, page 3-2](#)

Paging

Pagination in the resources is done by specifying query Params “page” and “pageSize”. The default page size is 5000. When there is more than one page, the `nextResourceURI` will specify the URI to the next resource.

Sorting

Sorting requests in the API requests is achieved by specifying queryParams “sortBy”. By default, ascending sort order is assumed. To specify the sort order use format “sort parameter: order”. For example, to request location results by `lastLocatedTime` in descending order specify queryParams “sortBy=lastLocatedTime:desc”.

For multiple sort conditions, use multiple “sortBy” queryParams.

Resources

Real Time Location APIs for Client

- `/api/contextaware/v1/location/clients/{id} {?zones}`
See the [locationclients](#) for more details.
- `/api/contextaware/v1/location/clients/count{?}`
See the [locationclientscount](#) for more details.

Real Time Location APIs for Tag

- `/api/contextaware/v1/location/tags/{id} {?zones}`
See the [locationtags](#) for more details.
- `/api/contextaware/v1/location/tags/count{?}`
See the [locationtagscount](#) for more details.

Real Time Location APIs for Rogue AP

- `/api/contextaware/v1/location/rogueaps/{id} {zones}`
See the [locationrogueaps](#) for more details.
- `/api/contextaware/v1/location/rogueaps/count{?}`
See the [locationrogueapscount](#) for more details.

Real Time Location APIs for Rogue Client

- `/api/contextaware/v1/location/rogueclients/{id}{?zones}`
See the [locationrogueclients](#) for more details.
- `/api/contextaware/v1/location/rogueclients/count{?}`
See the [locationrogueclientscount](#) for more details.

Real Time Location APIs for Interferer

- `/api/contextaware/v1/location/interferers/{id}{?zones}`
See the [locationinterferers](#) for more details.
- `/api/contextaware/v1/location/interferers/count{?}`
See the [locationinterfererscount](#) for more details.

Location History APIs

The location history APIs provides access to the historical location of devices tracked by the MSE.

All location history API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

The following data formats are supported for location history API. The response is determined based on the Accept HTTP header.

- XML—Use `Accept: application/xml`
- JSON—Use `Accept: application/jsonsupports`

This section contains the following topics:

- [Paging, page 3-4](#)
- [Sorting, page 3-4](#)
- [Resources, page 3-4](#)

Paging

Pagination in the resources is done by specifying queryParams “page” and “pageSize”. The default page size is 5000. When there is more than one page, the nextResourceURI will specify the URI to the next resource.

Sorting

Sorting requests in the API requests is achieved by specifying queryParams “sortBy”. By default, ascending sort order is assumed. To specify the sort order use format “sort parameter: order” For example, to request location results by lastLocatedTime in descending order specify queryParams “sortBy=lastLocatedTime:desc” For multiple sort conditions, use multiple “sortBy” queryParams.

Resources

Location History APIs for Client

- /api/contextaware/v1/location/history/clients/{id}{?}
See the [locationhistoryclients](#) for more details.
- /api/contextaware/v1/location/history/clients/{?}count{?}
See the [locationhistoryclientscount](#) for more details.

Location History APIs for Tag

- /api/contextaware/v1/location/history/tags/{id}{?}
See the [locationhistorytags](#) for more details.
- /api/contextaware/v1/location/history/tags/count{?}
See the [locationhistorytagscount](#) for more details.

Location History APIs for Rogue AP

- /api/contextaware/v1/location/history/rogueaps{?}
See the [locationhistoryrogueaps](#) for more details.
- /api/contextaware/v1/location/history/rogueaps/{id}/count{?}
See the [locationhistoryrogueapscount](#) for more details.

Location History APIs for Rogue Client

- /api/contextaware/v1/location/history/rogueclients/{id}{?}
See the [locationhistoryrogueclients](#) for more details.
- /api/contextaware/v1/location/history/rogueclients/count{?}
- See the [locationhistoryrogueclientscount](#) for more details

Location History APIs for Interferer

- `/api/contextaware/v1/location/history/interferers{?}`
See the [locationhistoryinterferers](#) for more details
- `/api/contextaware/v1/location/history/interferers/{id}/count{?}`
See the [locationhistoryinterfererscount](#) for more details.

Notification Event APIs

The Notification Event APIs allows an application to register for Push Notifications from the MSE.

The following Notification events are supported and these notifications can be configured for a specific MAC address or a list of MAC addresses with a wildcard for a specific device type.

1. **Absence Event**—Triggers a notification if the subscribed devices are absent for the configured absent interval.
2. **Battery Event**—Triggers a notification if the battery level of a RFID tag is at the configured level.
3. **Exciter Event**—Triggers a notification if the RFID tag is detected by a configured exciter.
4. **Containment Event**—Triggers a notification if a device is inside or outside of a configured zone. The OUT notification is sent when a device goes OUT of IN location boundary. The IN notification is sent when a device comes IN from the OUT of location boundary.



Note To get both IN and OUT events, you need to create a notification subscription with two containment events in the `subscribedEvents`, one for the INSIDE notification and the other one for the OUTSIDE notification. The behavior has changed in MSE Release 8.0 for containment. An INSIDE event is sent only when a device moves from OUTSIDE a zone to INSIDE and vice-versa for the OUTSIDE event.

5. **Emergency Event**—Triggers a notification if the emergency or panic button on the RFID tag is pressed.
6. **MapInfoChange Event**—Triggers a notification if the map configuration on the MSE changes.
7. **Movement Event**—Triggers a notification if a device moves more than a configured distance from its old location or a configured marker.
8. **Presence Event**—Triggers a notification when a device is first detected by the MSE.
9. **Streaming Notifications**—Triggers a stream of notification for every location calculation.
10. **Association Event**—Triggers a notification when a device associates to the Wi-Fi network. The subscription should have type “AssociationEventTrigger”, “association”: true/false.

All notification event API request need an Authorization header. See the [Authentication, page 2-1](#) for more information.

To support receivers behind a high latency link, the maximum delay recommended between the MSE and the application consuming the Notification stream is 90ms. At this 90ms delay, notifications can be sent at a rate of up to 500 PER SECOND when deployed on a MSE 3365 or Large Scale Virtual MSE. This requires setting the maximum thread count to 100/101. We recommend to contact Cisco support for changing these thread settings, if required.

The following data formats are supported for notification APIs. The response is determined based on the Accept HTTP header. The transport can be either HTTPS or TCP.

- XML—Use `Accept: application/xml`
- JSON—Use `Accept: application/json`
- Protocol Buffers

Resources

- `/api/contextaware/v1/notifications/{name}`
See the [notifications](#) for more details.



MSE API Object Descriptions

This chapter contains the following sections:

- [maps](#), page 4-2
- [mapscount](#), page 4-2
- [mapsinfo](#), page 4-3
- [mapsimage](#), page 4-3
- [mapsimagesource](#), page 4-4
- [locationclients](#), page 4-4
- [locationclientscount](#), page 4-5
- [locationtags](#), page 4-5
- [locationtagscount](#), page 4-6
- [locationrogueaps](#), page 4-6
- [locationrogueapscount](#), page 4-7
- [locationrogueclients](#), page 4-7
- [locationrogueclientscount](#), page 4-8
- [locationinterferers](#), page 4-8
- [locationinterfererscount](#), page 4-9
- [locationhistoryclients](#), page 4-9
- [locationhistoryclientscount](#), page 4-10
- [locationhistorytags](#), page 4-10
- [locationhistorytagscount](#), page 4-11
- [locationhistoryrogueaps](#), page 4-11
- [locationhistoryrogueapscount](#), page 4-12
- [locationhistoryrogueclients](#), page 4-12
- [locationhistoryrogueclientscount](#), page 4-13
- [locationhistoryinterferers](#), page 4-13
- [locationhistoryinterfererscount](#), page 4-14
- [notifications](#), page 4-15

maps

The maps object returns detailed map information about campuses, buildings, floors, access points, map dimensions, regions, zones, GPS marker, image information, etc.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

None

Response Types

JSON, XML

Return Codes

- 200—Successful response
- 404—MSE does not have any Map information
- 500—Server Error

mapscount

The mapscount returns maps count specifying the number of Campuses, Buildings, and Floors known to MSE.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

None

Response Types

JSON, XML

Return Codes

- 200—Successful response
- 500—Server Error

mapsinfo

The mapsinfo object returns all the floor information associated with the campusName -> buildingName -> floorName. This includes floor dimension, Access Points and their information, GPS Markers etc.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- floorname—Name of the required floor
- buildingname—Name of the required building
- campusname—Name of the required campus

Response Types

JSON, XML

Return Codes

- 200—Successful response
- 404—No floor with the specified campusName -> buildingName -> floorName
- 500—Server Error

mapsimage

The mapsimage object returns the floor image data associated with the particular campusName -> buildingName -> floorName.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- floorname—Name of the required floor
- buildingname—Name of the required building
- campusname—Name of the required campus

Response Types

image/jpeg

Return Codes

- 200—Successful response
- 404—No image with the specified campusName -> buildingName -> floorName
- 500—Server Error

mapsimagesource

The mapsimagesource object returns the image associated with the specified image name.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- imageName—Name of the required image.

Response Types

image/jpeg

Return Codes

- 200—Successful response
- 404—No image with the specified imagename
- 500—Server Error

locationclients

The location clients object returns the current location of the wireless client for the specified device ID. The ID can be MAC address, IP address, or Username.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—Mac address, IP Address, or Username of the wireless client.
- Zones—Contains zone information for the selected location,

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there is no wireless client for the specified ID.No image with the specified imagename
- 500—Server Error

locationclientscount

The locationclientscount object returns count or location of wireless clients on the MSE. Results are filtered based on the specified query param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the wireless client.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationtags

Returns the location of Tag for the specified query condition.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC Address of the tag
- Zones—Contains zone information for the location.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no tags for the specified query conditions.
- 500—Server Error

locationtagscount

The location tags count object returns a count of Tags on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the tag.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationrogueaps

Returns the location of Rogue APs for the specified query conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC Address of the rogue AP
- Zones—Contains zone information for the specified location.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no Rogue APs for the specified query conditions.
- 500—Server Error

locationroguapscount

Returns a count of Rogue APs on MSE based on the specified Query Param conditions.

Returns a list of Location of Rogue APs for the specified query conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC Address of the rogue AP
- Zones—Contains zone information for the specified location.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationroguclients

Returns location of Rogue Client for the specified id.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC address of the rogue client.

- Zones—Contains zone information for the specified location.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there is no rogue client for the specified id.
- 500—Server Error

locationrogueclientscount

Returns a count of Rogue Clients on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the rogue client.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationinterferers

Returns the Location of Interferer for the specified id.

Returns a count of Rogue Clients on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC address, IP Address, or Username of the wireless client.
- zones—Contains zone information for this location.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there is no interferer for the specified ID.
- 500—Server Error

locationinterfererscount

Returns a count of Interferers on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the interferers.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationhistoryclients

Returns a list of historical Location records of Wireless Client for the specified id and query conditions.

**Note**

If the Cisco Mobility Services Engine (MSE) is in a Demilitarized Zone (DMZ) and is NATed to an externally routable address on the Internet, then the nextResourceURI may not have the correct IP address in the MSE API response. The workaround to handle this scenario is to have the API consumer or the Client Application to construct the nextResourceURI by replacing IP address with the correct IP address.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC address, IP address, or username of the wireless client.
- []—Defines query conditions for the wireless client

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no historical records of wireless client for the specified ID and query conditions
- 500—Server Error

locationhistoryclientscount

Returns a count of historical Location records of Wireless Clients on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the wireless client.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationhistorytags

Returns a list of the historical Location records of Tag for the specified id and query conditions

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC address of the tag
- []—Defines query conditions for the tag.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no historical records of tag for the specified ID and query conditions.
- 500—Server Error

locationhistorytagscount

Returns a count of historical location records of Tags on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the interferers.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationhistoryrogueaps

Returns a list of historical records of Location of Rogue APs for the specified query conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC Address of the Rogue AP
- []—Defines query conditions for the rogue AP.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no historical records of Rogue APs for the specified query conditions.
- 500—Server Error

locationhistoryrogueapscount

Returns a count of historical Location records of Rogue APs based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the rogue AP.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationhistoryrogueclients

Returns a list of historical Location records of Rogue Clients for the specified query conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC Address of the rogue client
- []—Defines query conditions for the rogue client.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no historical records of rogue clients for the specified query conditions.
- 500—Server Error

locationhistoryrogueclientscount

Returns a count of Historical Location records of Rogue Clients on MSE based on the specified Query Param conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the rogue client.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 500—Server Error

locationhistoryinterferers

Returns a list of historical Location records of Interferers for the specified query conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- id—MAC address of the interferer.
- []—Defines query conditions for the interferer

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no Historical Location records of Interferer for the specified query conditions.
- 500—Server Error

locationhistoryinterfererscount

Returns a count of historical Location records of Interferers for the specified id and query conditions.

HTTP Method

GET

Accepts

- application/json
- application/xml

Parameters

- []—Defines query conditions for the interferers.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404— If there are no historical records of Interferer for the specified id and query conditions
- 500—Server Error

notifications

HTTP Method

GET—Returns all the Notifications Events subscribed by the requesting user.

Accepts

- application/json
- application/xml

Parameters

- name—The name of the Notification Event subscribed.

Response Types

XML and Json

Return Codes

- 200—Successful response
- 404—If there are no Notification Events subscribed by the requesting user.
- 500—Server Error



Object Schema Details

This chapter contains the following Schema details:

- [API Related Objects Schema Details for Map, page 5-1](#)
- [Schema Description for Map API Related Objects, page 5-3](#)
- [API Related Object Schema Details for Real Time and Historical Location, page 5-50](#)
- [Schema Description for Location API Related Objects, page 5-52](#)
- [API Related Object Schema Details for Notification API Object Schema Details, page 5-110](#)
- [Schema Description for Notifications API Related Objects, page 5-112](#)

API Related Objects Schema Details for Map

The following table lists the MAP API related objects.

Table 5-1 **MAP API Related Objects**

Object Name	Description	Schema Format
MapCounts	Gives details about the number of campuses, buildings, and floors.	MapCounts.xml MapCounts.json
Maps	Gives all map information known to the MSE.	Maps.xml Maps.json
Campus	Gives the name, dimension, image information, and list of buildings in the Campus.	Campus.xml Campus.json Campus.proto
Building	Gives the name, dimension, image information, and list of floors in the building.	Building.xml Building.json Building.Proto
Floor	Gives the name, dimension, image information, access points, GPS markers, zones, obstacles, reference markers, location filter rails/regions in the Floor.	Floor.xml Floor.json Floor.proto

Table 5-1 MAP API Related Objects

Object Name	Description	Schema Format
Zone	Gives the name and image associated with the campus, building, or floor.	Zone.xml Zone.json Zone.proto
Image	Gives the name and coordinates of the zone on a floor map.	Image.xml Image.json Image.proto
Dimension	Gives the dimension of the associated entity.	Dimension.xml Dimension.json Dimension.proto
AccessPoint	Gives the name, MAC address, IP address, mode, map coordinate, and list for AP interfaces.	AccessPoint.xml AccessPoint.json ap.proto
APIInterface	Gives the band information, slot number, channel number, channel assignment, transmit power level, antenna pattern, antenna angle, and antenna gain for the AP interface.	APIInterface.xml APIInterface.json ap-intf.proto
Band	Gives the band information.	ap-intf.proto
GPSTracker	Gives the name, geo, and map coordinates for the markers.	GPSTracker.xml GPSTracker.json gps-marker.proto
Exciter	Gives the name, MAC address, range, and map coordinates of the exciter.	Exciter.xml Exciter.json Exciter.proto
Obstacle	Gives the type, attenuation, and map coordinates of the Obstacle	Obstacle.xml Obstacle.json Obstacle.proto
ReferenceMarker	Gives the name and map coordinates of the reference marker.	ReferenceMarker.xml ReferenceMarker.json Reference.proto
LocationFilterRail	Gives the width and the map coordinates of the location filter rail.	LocationFilterRail.xml LocationFilterRail.json LocationFilterRail.proto
LocationFilterRegion	Gives the region type and the map coordinates of the location filter region.	LocationFilterRegion.xml LocationFilterRegion.xml LocationFilterRegion.proto

Schema Description for Map API Related Objects

MapCounts

MapCounts.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="MapCounts" type="mapCounts"/>

  <xs:complexType name="mapCounts">
    <xs:sequence>
      <xs:element name="CampusInfo" type="campusInfo" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="totalCampuses" type="xs:int" use="required"/>
    <xs:attribute name="totalBuildings" type="xs:int" use="required"/>
    <xs:attribute name="totalFloors" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="campusInfo">
    <xs:sequence>
      <xs:element name="BuildingInfo" type="buildingInfo" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="campusName" type="xs:string"/>
    <xs:attribute name="totalBuildings" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="buildingInfo">
    <xs:sequence/>
    <xs:attribute name="buildingName" type="xs:string"/>
    <xs:attribute name="floorCount" type="xs:int" use="required"/>
  </xs:complexType>
</xs:schema>
```

MapCounts.json

```
{
  "type" : "object",
  "properties" : {
    "totalCampuses" : {
      "type" : "integer"
    },
    "totalBuildings" : {
      "type" : "integer"
    },
    "totalFloors" : {
      "type" : "integer"
    },
    "campusCounts" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "campusName" : {
            "type" : "string"
          },
          "totalBuildings" : {
            "type" : "integer"
          }
        }
      }
    }
  }
}
```



```

        "type" : "number"
    },
    "height" : {
        "type" : "number"
    },
    "offsetX" : {
        "type" : "number"
    },
    "offsetY" : {
        "type" : "number"
    },
    "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
    }
}
},
"image" : {
    "type" : "object",
    "properties" : {
        "imageName" : {
            "type" : "string"
        },
        "sourceFile" : {
            "type" : "array",
            "items" : {
                "type" : "string"
            }
        }
    }
},
"buildingList" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "objectVersion" : {
                "type" : "integer"
            },
            "name" : {
                "type" : "string"
            },
            "dimension" : {
                "type" : "object",
                "properties" : {
                    "length" : {
                        "type" : "number"
                    },
                    "width" : {
                        "type" : "number"
                    },
                    "height" : {
                        "type" : "number"
                    },
                    "offsetX" : {
                        "type" : "number"
                    },
                    "offsetY" : {
                        "type" : "number"
                    },
                    "unit" : {
                        "type" : "string",
                        "enum" : [ "FEET", "METER" ]
                    }
                }
            }
        }
    }
}

```

```

    }
  },
  "image" : {
    "type" : "object",
    "properties" : {
      "imageName" : {
        "type" : "string"
      },
      "sourceFile" : {
        "type" : "array",
        "items" : {
          "type" : "string"
        }
      }
    }
  }
},
"floorList" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "objectVersion" : {
        "type" : "integer"
      },
      "name" : {
        "type" : "string"
      },
      "dimension" : {
        "type" : "object",
        "properties" : {
          "length" : {
            "type" : "number"
          },
          "width" : {
            "type" : "number"
          },
          "height" : {
            "type" : "number"
          },
          "offsetX" : {
            "type" : "number"
          },
          "offsetY" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "floorNumber" : {
    "type" : "integer"
  },
  "image" : {
    "type" : "object",
    "properties" : {
      "imageName" : {
        "type" : "string"
      },
      "sourceFile" : {
        "type" : "array",
        "items" : {
          "type" : "string"
        }
      }
    }
  }
}

```



```
    }
  }
},
"gpsMarkers" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "geoCoordinate" : {
        "type" : "object",
        "properties" : {
          "latitude" : {
            "type" : "number"
          },
          "longitude" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "DEGREES" ]
          }
        }
      }
    }
  },
  "mapCoordinate" : {
    "type" : "object",
    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  }
}
},
"zones" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "zoneCoordinate" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            }
          }
        }
      }
    }
  }
}
```

```

        "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
        }
    }
}
},
"obstacles" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "type" : {
                "type" : "string",
                "enum" : [ "UNKNOWN", "THICK_WALL", "NORMAL_WALL",
"THICK_DOOR", "NORMAL_DOOR", "CUBICLE", "GLASS" ]
            },
            "attenuation" : {
                "type" : "number"
            },
            "obstacleCoordinates" : {
                "type" : "array",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "x" : {
                            "type" : "number"
                        },
                        "y" : {
                            "type" : "number"
                        },
                        "unit" : {
                            "type" : "string",
                            "enum" : [ "FEET", "METER" ]
                        }
                    }
                }
            }
        }
    }
},
"accessPoints" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "name" : {
                "type" : "string"
            },
            "radioMacAddress" : {
                "type" : "string"
            },
            "ethMacAddress" : {
                "type" : "string"
            },
            "ipAddress" : {
                "type" : "string"
            },
            "numOfSlots" : {
                "type" : "integer"
            }
        }
    }
},

```

```

    "mapCoordinates" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "apMode" : {
      "type" : "string",
      "enum" : [ "LOCAL", "MONITOR", "REAP", "ROGUE_DETECTOR",
"SNIFFER", "BRIDGE", "UNKNOWN" ]
    },
    "apInterfaces" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "band" : {
            "type" : "string",
            "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B"
]
          },
          "slotNumber" : {
            "type" : "integer"
          },
          "channelAssignment" : {
            "type" : "integer"
          },
          "channelNumber" : {
            "type" : "integer"
          },
          "txPowerLevel" : {
            "type" : "integer"
          },
          "antennaPattern" : {
            "type" : "string"
          },
          "antennaAngle" : {
            "type" : "number"
          },
          "antennaElevAngle" : {
            "type" : "number"
          },
          "antennaGain" : {
            "type" : "integer"
          }
        }
      }
    },
    "referenceMarkers" : {
      "type" : "array",
      "items" : {
        "type" : "object",

```

```

    "properties" : {
      "name" : {
        "type" : "string"
      },
      "mapCoordinates" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    },
    "exciters" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "name" : {
            "type" : "string"
          },
          "macAddress" : {
            "type" : "string"
          },
          "range" : {
            "type" : "number"
          },
          "mapCoordinates" : {
            "type" : "object",
            "properties" : {
              "x" : {
                "type" : "number"
              },
              "y" : {
                "type" : "number"
              },
              "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
              }
            }
          },
          "perimeter" : {
            "type" : "boolean",
            "required" : true
          }
        }
      }
    },
    "filterRegions" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "regionType" : {

```

```

        "type" : "string",
        "enum" : [ "INSIDE", "OUTSIDE" ]
    },
    "regionCoordinates" : {
        "type" : "array",
        "items" : {
            "type" : "object",
            "properties" : {
                "x" : {
                    "type" : "number"
                },
                "y" : {
                    "type" : "number"
                },
                "unit" : {
                    "type" : "string",
                    "enum" : [ "FEET", "METER" ]
                }
            }
        }
    }
},
"filterRails" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "railBoundaryWidth" : {
                "type" : "number"
            },
            "railCoordinates" : {
                "type" : "array",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "x" : {
                            "type" : "number"
                        },
                        "y" : {
                            "type" : "number"
                        },
                        "unit" : {
                            "type" : "string",
                            "enum" : [ "FEET", "METER" ]
                        }
                    }
                }
            }
        }
    }
},
"railExclusionCoordinates" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "x" : {
                "type" : "number"
            },
            "y" : {
                "type" : "number"
            },
            "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
            }
        }
    }
}

```

```
        }
      }
    }
  },
  "outdoor" : {
    "type" : "boolean",
    "required" : true
  }
}
}
}
}
}
}
```

Campus

Campus.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Campus.json

```
{
  "type" : "object",
  "properties" : {
    "objectVersion" : {
      "type" : "integer"
    },
    "name" : {
      "type" : "string"
    },
    "dimension" : {
      "type" : "object",
      "properties" : {
        "length" : {
          "type" : "number"
        },
        "width" : {
          "type" : "number"
        },
        "height" : {
          "type" : "number"
        }
      }
    },
  }
}
```

```

    "offsetX" : {
      "type" : "number"
    },
    "offsetY" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"image" : {
  "type" : "object",
  "properties" : {
    "imageName" : {
      "type" : "string"
    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  }
},
"buildingList" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "objectVersion" : {
        "type" : "integer"
      },
      "name" : {
        "type" : "string"
      },
      "dimension" : {
        "type" : "object",
        "properties" : {
          "length" : {
            "type" : "number"
          },
          "width" : {
            "type" : "number"
          },
          "height" : {
            "type" : "number"
          },
          "offsetX" : {
            "type" : "number"
          },
          "offsetY" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "image" : {
    "type" : "object",
    "properties" : {

```

```

    "imageName" : {
      "type" : "string"
    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  },
  "floorList" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "objectVersion" : {
          "type" : "integer"
        },
        "name" : {
          "type" : "string"
        },
        "dimension" : {
          "type" : "object",
          "properties" : {
            "length" : {
              "type" : "number"
            },
            "width" : {
              "type" : "number"
            },
            "height" : {
              "type" : "number"
            },
            "offsetX" : {
              "type" : "number"
            },
            "offsetY" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    },
    "floorNumber" : {
      "type" : "integer"
    },
    "image" : {
      "type" : "object",
      "properties" : {
        "imageName" : {
          "type" : "string"
        },
        "sourceFile" : {
          "type" : "array",
          "items" : {
            "type" : "string"
          }
        }
      }
    },
    "gpsMarkers" : {

```



```

"type" : "array",
"items" : {
  "type" : "object",
  "properties" : {
    "name" : {
      "type" : "string"
    },
    "geoCoordinate" : {
      "type" : "object",
      "properties" : {
        "latitude" : {
          "type" : "number"
        },
        "longitude" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "DEGREES" ]
        }
      }
    },
    "mapCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    }
  }
},
"zones" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "zoneCoordinate" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    }
  }
}

```

```

        }
      }
    }
  },
  "obstacles" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "type" : {
          "type" : "string",
          "enum" : [ "UNKNOWN", "THICK_WALL", "NORMAL_WALL", "THICK_DOOR",
"NORMAL_DOOR", "CUBICLE", "GLASS" ]
        },
        "attenuation" : {
          "type" : "number"
        },
        "obstacleCoordinates" : {
          "type" : "array",
          "items" : {
            "type" : "object",
            "properties" : {
              "x" : {
                "type" : "number"
              },
              "y" : {
                "type" : "number"
              },
              "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
              }
            }
          }
        }
      }
    }
  }
},
"accessPoints" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "radioMacAddress" : {
        "type" : "string"
      },
      "ethMacAddress" : {
        "type" : "string"
      },
      "ipAddress" : {
        "type" : "string"
      },
      "numOfSlots" : {
        "type" : "integer"
      },
      "mapCoordinates" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          }
        }
      }
    }
  }
}

```

```

    },
    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"apMode" : {
  "type" : "string",
  "enum" : [ "LOCAL", "MONITOR", "REAP", "ROGUE_DETECTOR",
"SNIFFER", "BRIDGE", "UNKNOWN" ]
},
"apInterfaces" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "band" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
      },
      "slotNumber" : {
        "type" : "integer"
      },
      "channelAssignment" : {
        "type" : "integer"
      },
      "channelNumber" : {
        "type" : "integer"
      },
      "txPowerLevel" : {
        "type" : "integer"
      },
      "antennaPattern" : {
        "type" : "string"
      },
      "antennaAngle" : {
        "type" : "number"
      },
      "antennaElevAngle" : {
        "type" : "number"
      },
      "antennaGain" : {
        "type" : "integer"
      }
    }
  }
}
}
},
"referenceMarkers" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "mapCoordinates" : {
        "type" : "object",

```

```

        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    },
  },
  "exciters" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "name" : {
          "type" : "string"
        },
        "macAddress" : {
          "type" : "string"
        },
        "range" : {
          "type" : "number"
        },
        "mapCoordinates" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    },
    "perimeter" : {
      "type" : "boolean",
      "required" : true
    }
  }
},
"filterRegions" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "regionType" : {
        "type" : "string",
        "enum" : [ "INSIDE", "OUTSIDE" ]
      },
      "regionCoordinates" : {
        "type" : "array",
        "items" : {

```

```

        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "filterRails" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "railBoundaryWidth" : {
          "type" : "number"
        },
        "railCoordinates" : {
          "type" : "array",
          "items" : {
            "type" : "object",
            "properties" : {
              "x" : {
                "type" : "number"
              },
              "y" : {
                "type" : "number"
              },
              "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
              }
            }
          }
        }
      }
    },
    "railExclusionCoordinates" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  }
}

```

```

    },
    "outdoor" : {
      "type" : "boolean",
      "required" : true
    }
  }
}
}
}
}
}
}
}
}
}
}
}
}

```

Campus.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "CampusProtos";

import "dimension.proto";
import "image.proto";
import "building.proto";

message PBCampus {

  required string name = 1;

  optional PBDimension dimension = 2;

  optional PBImage image = 3;

  optional int32 objectVersion = 4;

  repeated PBBuilding buildings = 5;

}

```

Building

Building.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Building.json

```

{
  "type" : "object",
  "properties" : {
    "objectVersion" : {
      "type" : "integer"
    }
  }
}

```

```

    },
    "name" : {
      "type" : "string"
    },
    "dimension" : {
      "type" : "object",
      "properties" : {
        "length" : {
          "type" : "number"
        },
        "width" : {
          "type" : "number"
        },
        "height" : {
          "type" : "number"
        },
        "offsetX" : {
          "type" : "number"
        },
        "offsetY" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "image" : {
      "type" : "object",
      "properties" : {
        "imageName" : {
          "type" : "string"
        },
        "sourceFile" : {
          "type" : "array",
          "items" : {
            "type" : "string"
          }
        }
      }
    },
    "floorList" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "objectVersion" : {
            "type" : "integer"
          },
          "name" : {
            "type" : "string"
          },
          "dimension" : {
            "type" : "object",
            "properties" : {
              "length" : {
                "type" : "number"
              },
              "width" : {
                "type" : "number"
              },
              "height" : {
                "type" : "number"
              }
            }
          }
        }
      }
    }
  }

```

```

    },
    "offsetX" : {
      "type" : "number"
    },
    "offsetY" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"floorNumber" : {
  "type" : "integer"
},
"image" : {
  "type" : "object",
  "properties" : {
    "imageName" : {
      "type" : "string"
    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  }
},
"gpsMarkers" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "geoCoordinate" : {
        "type" : "object",
        "properties" : {
          "latitude" : {
            "type" : "number"
          },
          "longitude" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "DEGREES" ]
          }
        }
      }
    }
  }
},
"mapCoordinate" : {
  "type" : "object",
  "properties" : {
    "x" : {
      "type" : "number"
    },
    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",

```



```

        "enum" : [ "FEET", "METER" ]
      }
    }
  }
},
"zones" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "zoneCoordinate" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    }
  }
},
"obstacles" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "type" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "THICK_WALL", "NORMAL_WALL", "THICK_DOOR",
"NORMAL_DOOR", "CUBICLE", "GLASS" ]
      },
      "attenuation" : {
        "type" : "number"
      },
      "obstacleCoordinates" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
}
},
"accessPoints" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "radioMacAddress" : {
        "type" : "string"
      },
      "ethMacAddress" : {
        "type" : "string"
      },
      "ipAddress" : {
        "type" : "string"
      },
      "numOfSlots" : {
        "type" : "integer"
      },
      "mapCoordinates" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "apMode" : {
    "type" : "string",
    "enum" : [ "LOCAL", "MONITOR", "REAP", "ROGUE_DETECTOR", "SNIFFER",
"BRIDGE", "UNKNOWN" ]
  },
  "apInterfaces" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "band" : {
          "type" : "string",
          "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
        },
        "slotNumber" : {
          "type" : "integer"
        },
        "channelAssignment" : {
          "type" : "integer"
        },
        "channelNumber" : {
          "type" : "integer"
        }
      }
    }
  },

```

```

        "txPowerLevel" : {
            "type" : "integer"
        },
        "antennaPattern" : {
            "type" : "string"
        },
        "antennaAngle" : {
            "type" : "number"
        },
        "antennaElevAngle" : {
            "type" : "number"
        },
        "antennaGain" : {
            "type" : "integer"
        }
    }
}
}
}
},
"referenceMarkers" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "name" : {
                "type" : "string"
            },
            "mapCoordinates" : {
                "type" : "object",
                "properties" : {
                    "x" : {
                        "type" : "number"
                    },
                    "y" : {
                        "type" : "number"
                    },
                    "unit" : {
                        "type" : "string",
                        "enum" : [ "FEET", "METER" ]
                    }
                }
            }
        }
    }
},
"exciters" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "name" : {
                "type" : "string"
            },
            "macAddress" : {
                "type" : "string"
            },
            "range" : {
                "type" : "number"
            },
            "mapCoordinates" : {
                "type" : "object",
                "properties" : {

```

```

        "x" : {
            "type" : "number"
        },
        "y" : {
            "type" : "number"
        },
        "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
        }
    }
},
"perimeter" : {
    "type" : "boolean",
    "required" : true
}
}
},
"filterRegions" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "regionType" : {
                "type" : "string",
                "enum" : [ "INSIDE", "OUTSIDE" ]
            },
            "regionCoordinates" : {
                "type" : "array",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "x" : {
                            "type" : "number"
                        },
                        "y" : {
                            "type" : "number"
                        },
                        "unit" : {
                            "type" : "string",
                            "enum" : [ "FEET", "METER" ]
                        }
                    }
                }
            }
        }
    }
},
"filterRails" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "railBoundaryWidth" : {
                "type" : "number"
            },
            "railCoordinates" : {
                "type" : "array",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "x" : {
                            "type" : "number"
                        }
                    }
                }
            }
        }
    }
}
}
}

```

```
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "railExclusionCoordinates" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "outdoor" : {
    "type" : "boolean",
    "required" : true
  }
}
}
```

Building.Proto

```
option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "BuildingProtos";

import "dimension.proto";
import "image.proto";
import "floor.proto";
message PBBuilding {

  required string name = 1;

  optional int32 objectVersion = 2;

  optional PBDimension dimension = 3;

  optional PBImage image = 4;

  repeated PBFloor floors = 5;
}
```

Floor

Floor.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Floor.json

```
{
  "type" : "object",
  "properties" : {
    "objectVersion" : {
      "type" : "integer"
    },
    "name" : {
      "type" : "string"
    },
    "dimension" : {
      "type" : "object",
      "properties" : {
        "length" : {
          "type" : "number"
        },
        "width" : {
          "type" : "number"
        },
        "height" : {
          "type" : "number"
        },
        "offsetX" : {
          "type" : "number"
        },
        "offsetY" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "floorNumber" : {
      "type" : "integer"
    },
    "image" : {
      "type" : "object",
      "properties" : {
        "imageName" : {
          "type" : "string"
        },
        "sourceFile" : {
          "type" : "array",
          "items" : {
```

```

        "type" : "string"
      }
    }
  },
  "gpsMarkers" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "name" : {
          "type" : "string"
        },
        "geoCoordinate" : {
          "type" : "object",
          "properties" : {
            "latitude" : {
              "type" : "number"
            },
            "longitude" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "DEGREES" ]
            }
          }
        }
      }
    },
    "mapCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    }
  }
},
"zones" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "zoneCoordinate" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
}
},
"obstacles" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "type" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "THICK_WALL", "NORMAL_WALL", "THICK_DOOR",
"NORMAL_DOOR", "CUBICLE", "GLASS" ]
      },
      "attenuation" : {
        "type" : "number"
      },
      "obstacleCoordinates" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    }
  }
}
},
"accessPoints" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "radioMacAddress" : {
        "type" : "string"
      },
      "ethMacAddress" : {
        "type" : "string"
      },
      "ipAddress" : {
        "type" : "string"
      },
      "numOfSlots" : {
        "type" : "integer"
      }
    }
  }
}
}

```



```

    },
    "mapCoordinates" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "apMode" : {
      "type" : "string",
      "enum" : [ "LOCAL", "MONITOR", "REAP", "ROGUE_DETECTOR", "SNIFFER", "BRIDGE",
"UNKNOWN" ]
    },
    "apInterfaces" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "band" : {
            "type" : "string",
            "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
          },
          "slotNumber" : {
            "type" : "integer"
          },
          "channelAssignment" : {
            "type" : "integer"
          },
          "channelNumber" : {
            "type" : "integer"
          },
          "txPowerLevel" : {
            "type" : "integer"
          },
          "antennaPattern" : {
            "type" : "string"
          },
          "antennaAngle" : {
            "type" : "number"
          },
          "antennaElevAngle" : {
            "type" : "number"
          },
          "antennaGain" : {
            "type" : "integer"
          }
        }
      }
    }
  },
  "referenceMarkers" : {
    "type" : "array",
    "items" : {
      "type" : "object",

```

```

    "properties" : {
      "name" : {
        "type" : "string"
      },
      "mapCoordinates" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "exciters" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "name" : {
          "type" : "string"
        },
        "macAddress" : {
          "type" : "string"
        },
        "range" : {
          "type" : "number"
        },
        "mapCoordinates" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    },
    "perimeter" : {
      "type" : "boolean",
      "required" : true
    }
  }
},
"filterRegions" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "regionType" : {

```

```

        "type" : "string",
        "enum" : [ "INSIDE", "OUTSIDE" ]
    },
    "regionCoordinates" : {
        "type" : "array",
        "items" : {
            "type" : "object",
            "properties" : {
                "x" : {
                    "type" : "number"
                },
                "y" : {
                    "type" : "number"
                },
                "unit" : {
                    "type" : "string",
                    "enum" : [ "FEET", "METER" ]
                }
            }
        }
    }
},
"filterRails" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "railBoundaryWidth" : {
                "type" : "number"
            },
            "railCoordinates" : {
                "type" : "array",
                "items" : {
                    "type" : "object",
                    "properties" : {
                        "x" : {
                            "type" : "number"
                        },
                        "y" : {
                            "type" : "number"
                        },
                        "unit" : {
                            "type" : "string",
                            "enum" : [ "FEET", "METER" ]
                        }
                    }
                }
            }
        }
    }
},
"railExclusionCoordinates" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "x" : {
                "type" : "number"
            },
            "y" : {
                "type" : "number"
            },
            "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
            }
        }
    }
}

```

```

        }
      }
    }
  }
},
"outdoor" : {
  "type" : "boolean",
  "required" : true
}
}
}
}

```

Floor.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "FloorProtos";

import "dimension.proto";
import "image.proto";
import "gps-marker.proto";
import "zone.proto";
import "obstacle.proto";
import "ap.proto";
import "ref-marker.proto";
import "exciter.proto";
import "loc-filter-region.proto";
import "loc-filter-rail.proto";

message PBFloor {

  required string name = 1;

  optional int32 objectVersion = 2;

  optional PBDimension dimension = 3;

  optional bool isOutdoor = 4 [default = false];

  required int32 floorNumber = 5;

  optional PBImage image = 6;

  repeated PBGPSTMarker gpsMarkers = 7;

  repeated PBZone zones = 8;

  repeated PBObstacle obstacles = 9;

  repeated PBAccessPoint accessPoints = 10;

  repeated PBReferenceMarker referenceMarkers = 11;

  repeated PBExciter exciters = 12;

  repeated PBLocationFilterRegion filterRegions = 13;

  repeated PBLocationFilterRail filterRails = 14;
}

```

Zone

Zone.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="Image" type="image"/>

  <xs:complexType name="image">
    <xs:sequence>
      <xs:element name="imageData" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="imageName" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Zone.json

```
{
  "type" : "object",
  "properties" : {
    "imageName" : {
      "type" : "string"
    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  }
}
```

Zone.proto

```
option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "ImageProtos";

message PBIImage {

  required string imageName = 1;

  optional bytes sourceFile = 2;
}
```

Image

Image.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
}
```

```

</xs:simpleType>
</xs:schema>

```

Image.json

```

{
  "type" : "object",
  "properties" : {
    "name" : {
      "type" : "string"
    },
    "zoneCoordinate" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  }
}

```

Image.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "ZoneProtos";

import "map-coordinate.proto";

message PBZone {

  required string name = 1;

  repeated PMapCoordinate zoneCoordinate = 2;
}

```

Dimension

Dimension.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns1="Dimension"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="Dimension" schemaLocation="Dimension.xsd"/>

  <xs:element name="Dimension" type="dimension"/>

  <xs:complexType name="dimension">
    <xs:sequence/>
    <xs:attribute name="length" type="xs:float" use="required"/>

```

```

<xs:attribute name="width" type="xs:float" use="required"/>
<xs:attribute name="height" type="xs:float" use="required"/>
<xs:attribute name="offsetX" type="xs:float" use="required"/>
<xs:attribute name="offsetY" type="xs:float" use="required"/>
<xs:attribute name="unit" type="ns1:Unit"/>
</xs:complexType>
</xs:schema>

```

Dimension.json

```

{
  "type" : "object",
  "properties" : {
    "length" : {
      "type" : "number"
    },
    "width" : {
      "type" : "number"
    },
    "height" : {
      "type" : "number"
    },
    "offsetX" : {
      "type" : "number"
    },
    "offsetY" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
}

```

Dimension.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "DimensionProtos";

message PBDimension {

  enum PBUUnit {
    FEET = 0;
    METER = 1;
  }

  required float length = 1;

  required float width = 2;

  optional float height = 3;

  optional float offsetX = 4;

  optional float offsetY = 5;

  optional PBUUnit unit = 6 [default = FEET];
}

```

Access Point

AccessPoint.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

AccessPoint.json

```
{
  "type" : "object",
  "properties" : {
    "name" : {
      "type" : "string"
    },
    "radioMacAddress" : {
      "type" : "string"
    },
    "ethMacAddress" : {
      "type" : "string"
    },
    "ipAddress" : {
      "type" : "string"
    },
    "numOfSlots" : {
      "type" : "integer"
    },
    "mapCoordinates" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "apMode" : {
      "type" : "string",
      "enum" : [ "LOCAL", "MONITOR", "REAP", "ROGUE_DETECTOR", "SNIFFER", "BRIDGE",
"UNKNOWN" ]
    },
    "apInterfaces" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "band" : {
            "type" : "string",
```



```

    repeated PBAPInterface apInterfaces = 8;
}

```

APIInterface

APIInterface.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="APIInterface" type="apInterface"/>

  <xs:complexType name="apInterface">
    <xs:sequence/>
    <xs:attribute name="band" type="band"/>
    <xs:attribute name="slotNumber" type="xs:int" use="required"/>
    <xs:attribute name="channelAssignment" type="xs:int" use="required"/>
    <xs:attribute name="channelNumber" type="xs:int" use="required"/>
    <xs:attribute name="txPowerLevel" type="xs:int" use="required"/>
    <xs:attribute name="antennaPattern" type="xs:string"/>
    <xs:attribute name="antennaAngle" type="xs:float" use="required"/>
    <xs:attribute name="antennaElevAngle" type="xs:float" use="required"/>
    <xs:attribute name="antennaGain" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:simpleType name="band">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="IEEE_802_11_A"/>
      <xs:enumeration value="IEEE_802_11_B"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

APIInterface.json

```

{
  "type" : "object",
  "properties" : {
    "band" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
    },
    "slotNumber" : {
      "type" : "integer"
    },
    "channelAssignment" : {
      "type" : "integer"
    },
    "channelNumber" : {
      "type" : "integer"
    },
    "txPowerLevel" : {
      "type" : "integer"
    },
    "antennaPattern" : {
      "type" : "string"
    },
    "antennaAngle" : {

```

```

        "type" : "number"
    },
    "antennaElevAngle" : {
        "type" : "number"
    },
    "antennaGain" : {
        "type" : "integer"
    }
}
}

```

ap-intf.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "ApInterfaceProtos";

import "band.proto";

message PBAPInterface {

    required PBBand band = 1;

    optional int32 slotNumber = 2;

    optional int32 channelAssignment = 3;

    optional int32 channelNumber = 4;

    optional int32 txPowerLevel = 5;

    optional string antennaPattern = 6;

    optional float antennaAngle = 7;

    optional float antennaElevAngle = 8;

    optional int32 antennaGain = 9;

}

```

Band

band.proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "BandProtos";
enum PBBand {
    IEEE_802_11_A = 0;

    IEEE_802_11_B = 1;
}

```

GPS Marker

GPSTMarker.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns2="MapCoordinate" xmlns:ns1="GeoCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

```

```

<xs:import namespace="GeoCoordinate" schemaLocation="GPSMarker.xsd"/>
<xs:import namespace="MapCoordinate" schemaLocation="GPSMarker.xsd"/>

<xs:element name="GPSMarker" type="gpsMarker"/>

<xs:element name="GeoCoordinate" type="geoCoordinate"/>
<xs:element name="MapCoordinate" type="mapCoordinate"/>

<xs:complexType name="gpsMarker">
  <xs:sequence>
    <xs:element ref="GeoCoordinate" minOccurs="0"/>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="geoCoordinate">
  <xs:sequence/>
  <xs:attribute name="latitude" type="xs:double" use="required"/>
  <xs:attribute name="longitude" type="xs:double" use="required"/>
  <xs:attribute name="unit" type="ns1:Unit"/>
</xs:complexType>

<xs:complexType name="mapCoordinate">
  <xs:sequence/>
  <xs:attribute name="x" type="xs:float" use="required"/>
  <xs:attribute name="y" type="xs:float" use="required"/>
  <xs:attribute name="unit" type="ns2:Unit"/>
</xs:complexType>
</xs:schema>

```

GPSMarker.json

```

{
  "type" : "object",
  "properties" : {
    "name" : {
      "type" : "string"
    },
    "geoCoordinate" : {
      "type" : "object",
      "properties" : {
        "latitude" : {
          "type" : "number"
        },
        "longitude" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "DEGREES" ]
        }
      }
    },
    "mapCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {

```

```

        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  }
}
}
}

```

gps-marker.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "GPSMarkerProtos";

import "map-coordinate.proto";
import "geo-coordinate.proto";

message PBGPSMarker {

    required string name = 1;

    required PBGeoCoordinate geoCoordinate = 2;

    required PBMapCoordinate mapCoordinate = 3;
}

```

Exciter

Exciter.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns1="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:import namespace="MapCoordinate" schemaLocation="Exciter.xsd"/>

    <xs:element name="Exciter" type="exciter"/>

    <xs:element name="MapCoordinate" type="mapCoordinate"/>

    <xs:complexType name="exciter">
        <xs:sequence>
            <xs:element ref="MapCoordinate" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string"/>
        <xs:attribute name="macAddress" type="xs:string"/>
        <xs:attribute name="range" type="xs:double" use="required"/>
        <xs:attribute name="isPerimeter" type="xs:boolean" use="required"/>
    </xs:complexType>

    <xs:complexType name="mapCoordinate">
        <xs:sequence/>
        <xs:attribute name="x" type="xs:float" use="required"/>
        <xs:attribute name="y" type="xs:float" use="required"/>
        <xs:attribute name="unit" type="ns1:Unit"/>
    </xs:complexType>
</xs:schema>

```

Exciter.json

```

{
  "type" : "object",
  "properties" : {
    "name" : {
      "type" : "string"
    },
    "macAddress" : {
      "type" : "string"
    },
    "range" : {
      "type" : "number"
    },
    "mapCoordinates" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "perimeter" : {
      "type" : "boolean",
      "required" : true
    }
  }
}

```

Exciter.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "ExciterProtos";

import "map-coordinate.proto";

message PBExciter {

    required string name = 1;

    optional string macAddress = 2;

    optional double range = 3;

    optional bool isPerimeter = 4;

    required PBMapCoordinate mapCoordinates = 5;
}

```

Obstacle**Obstacle.xml**

```

{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```

<xs:schema version="1.0" xmlns:ns1="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="MapCoordinate" schemaLocation="Obstacle.xsd"/>

  <xs:element name="MapCoordinate" type="mapCoordinate"/>

  <xs:element name="Obstacle" type="obstacle"/>

  <xs:complexType name="obstacle">
    <xs:sequence>
      <xs:element name="ObstacleCoordinate" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="type"/>
    <xs:attribute name="attenuation" type="xs:float" use="required"/>
  </xs:complexType>

  <xs:complexType name="mapCoordinate">
    <xs:sequence/>
    <xs:attribute name="x" type="xs:float" use="required"/>
    <xs:attribute name="y" type="xs:float" use="required"/>
    <xs:attribute name="unit" type="ns1:Unit"/>
  </xs:complexType>

  <xs:simpleType name="type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="THICK_WALL"/>
      <xs:enumeration value="NORMAL_WALL"/>
      <xs:enumeration value="THICK_DOOR"/>
      <xs:enumeration value="NORMAL_DOOR"/>
      <xs:enumeration value="CUBICLE"/>
      <xs:enumeration value="GLASS"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Obstacle.json

```

{
  "type" : "object",
  "properties" : {
    "type" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "THICK_WALL", "NORMAL_WALL", "THICK_DOOR", "NORMAL_DOOR",
"OBSTACLE", "GLASS" ]
    },
    "attenuation" : {
      "type" : "number"
    },
    "obstacleCoordinates" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {

```

```

        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  }
}
}
}
}

```

Obstacle.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "ObstacleProtos";

import "map-coordinate.proto";

message PObstacle {

  enum PBType {

    THICK_WALL = 1;

    NORMAL_WALL = 2;

    THICK_DOOR = 3;

    NORMAL_DOOR = 4;

    CUBICLE = 5;

    GLASS = 6;
  }

  optional PBType type = 1;

  optional float attenuation = 2;

  repeated PMapCoordinate obstacleCoordinates = 3;
}

```

Reference Marker

ReferenceMarker.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

ReferenceMarker.json

```

{
  "type" : "object",

```



```

"properties" : {
  "name" : {
    "type" : "string"
  },
  "mapCoordinates" : {
    "type" : "object",
    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  }
}
}
}

```

Reference.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "ReferenceMarkerProtos";

import "map-coordinate.proto";

message PBReferenceMarker {

  required string name = 1;

  required PMapCoordinate mapCoordinates = 2;
}

```

LocationFilterRail

LocationFilterRail.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns1="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="MapCoordinate" schemaLocation="LocationFilterRail.xsd"/>

  <xs:element name="LocationFilterRail" type="locationFilterRail"/>

  <xs:element name="MapCoordinate" type="mapCoordinate"/>

  <xs:complexType name="locationFilterRail">
    <xs:sequence>
      <xs:element name="RailCoordinates" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="RailExclusionCoordinates" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="railBoundaryWidth" type="xs:float" use="required"/>
  </xs:complexType>

```

```

<xs:complexType name="mapCoordinate">
  <xs:sequence/>
  <xs:attribute name="x" type="xs:float" use="required"/>
  <xs:attribute name="y" type="xs:float" use="required"/>
  <xs:attribute name="unit" type="ns1:Unit"/>
</xs:complexType>
</xs:schema>

```

LocationFilterRail.json

```

{
  "type" : "object",
  "properties" : {
    "railBoundaryWidth" : {
      "type" : "number"
    },
    "railCoordinates" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    },
    "railExclusionCoordinates" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  }
}

```

LocationFilterRail.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "LocationFilterRailsProtos";

import "map-coordinate.proto";

message PBLocationFilterRail {

```

```

optional float railBoundaryWidth = 1;

repeated PMapCoordinate railCoordinates = 2;

repeated PMapCoordinate railExclusionCoordinates = 3;
}

```

LocationFilterRegion

LocationFilterRegion.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

LocationFilterRegion.json

```

{
  "type" : "object",
  "properties" : {
    "regionType" : {
      "type" : "string",
      "enum" : [ "INSIDE", "OUTSIDE" ]
    },
    "regionCoordinates" : {
      "type" : "array",
      "items" : {
        "type" : "object",
        "properties" : {
          "x" : {
            "type" : "number"
          },
          "y" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  }
}

```

LocationFilterRegion.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "LocationFilterRegionProtos";

import "map-coordinate.proto";

```

```

message PBLocationFilterRegion {
    enum PBRegionType {
        INSIDE = 1;
        OUTSIDE = 2;
    }

    required PBRegionType regionType = 1;

    repeated PBMapCoordinate regionCoordinates = 2;
}

```

API Related Object Schema Details for Real Time and Historical Location

The below table lists the Real Time and Historical Location API.

Table 5-2 Real Time and Historical Location API Related Objects

Object Name	Description	Schema Format
WirelessClientApiQueryParams	Provides query conditions like dot11status, associated AP Mac address, SSID, map hierarchy, located before or after, paging, and sorting to retrieve the wireless client location data.	WirelessClientApiQueryParams.xml WirelessClientApiqueryparams.json
TagApiQueryParams	Provides query conditions like map hierarchy, located before or after, paging and sorting to retrieve the tag location data.	TagApiQueryParams.xml TagApiQueryParams.xml
RogueClientApiQueryParams	Provides query conditions like map hierarchy, located before or after, paging and sorting to retrieve the rogue client location data.	RogueClientApiQueryParams.xml RogueClientApiQueryParams.json
RogueApApiQueryParams	Provides query conditions like map hierarchy, located before or after, paging and sorting to retrieve the rogue AP location data.	RogueAPApiQueryParams RogueAPApiQueryParams.json
InterfererApiQueryParams	Provides query conditions like map hierarchy, located before or after, paging and sorting to retrieve the interferer location data.	InterfererApiQueryParams.xml InterfererApiQueryParams.json
DeviceCount	Gives the count of devices of the specified type and query condition.	DeviceCount.xml DeviceCount.json
Locations	This is the object returned for list queries. It contains the Location information for the requested device type. In addition, it provides paging information and location of the next resource in case of multiple pages	Locations.xml Locations.json

Table 5-2 Real Time and Historical Location API Related Objects (continued)

Object Name	Description	Schema Format
MapInfo	Gives the map location details for an associated device.	MapInfo.xml MapInfo.json Map-info.proto
MapCoordinate	Gives the map coordinates as x and y values on a floor map for an associated entity.	MapCoordinate.xml MapCoordinate.json MapCoordinate.Proto
GeoCoordinate	Gives the latitude and longitude values for an associated entity.	GeoCoordinate.xml GeoCoordinate.json geo-coordinate.proto
WirelessClientLocation	Gives the location information, IP address, username, SSID, band, associated AP MAC address and status of a wireless client.	WirelessClientLocation.xml WirelessClientLocation.json WirelessClientLocation.proto
HistoryReason	Gives the reason for which the historical location record of a device was saved.	HistoryReason.xml HistoryReason.json HistoryReason.Proto
Statistics	Gives the statistics for the location record of a device.	Statistics.xml statistics.json statistics.proto
RogueAPLocation	Gives the location information, total associated client, Containment level, type, state, SSID and zone of impact of the rogue AP.	RogueAPLocation.xml RogueAPLocation.json rogue-ap.proto
RogueClientLocation	Gives the location information, total detecting APs, containment level, state and SSID of the rogue client.	RogueClientLocation.xml RogueClientLocation.json RogueClientLocation.Proto
InterfererLocation	Gives the location information, device type, zone of impact, duty cycle, severity, affected channels and affected band for the interferer.	InterfererLocation.xml InterfererLocation.json interferer.proto
TagLocation	Gives the location information, telemetry, battery, vendor, emergency and device exciter information for a RFID Tag.	TagLocation.xml TagLocation.json tag.proto
BatteryInfo	Gives the battery information like tolerance, percentage remaining, age, etc for the RFID Tag.	BatteryInfo.xml BatteryInfo.json battery-info.proto

Table 5-2 Real Time and Historical Location API Related Objects (continued)

Object Name	Description	Schema Format
EmergencyInfo	Gives the emergency information like tamper state, reason, last received time, etc for the RFID Tag.	EmergencyInfo.xml EmergencyInfo.json emergency-info.proto
VendorData	Gives the vendor specific proprietary information as a byte blob for the RFID Tag.	VendorData.xml VendorData.json vendor-data.proto
DeviceExciter	Gives the device exciter information like MAC address, last received sequence number, timestamp, whether it is in range or perimeter and its associated floor id for the RFID Tag.	DeviceExciter.xml DeviceExciter.json device-exciter.proto
TelemetryInfo	Gives the telemetry info type, last received time and the telemetry measurement data for a RFID Tag. The telemetry measurement data can be a simple byte blob, string or of a complex type EPC telemetry data, geo telemetry data or motion telemetry data.	TelemetryInfo.xml TelemetryInfo.json tag-telemetry.proto
DefaultTelemetryMeasurementData	Gives the telemetry measurement data in string format for a RFID tag.	DefaultTelemetryMeasurementData.xml DefaultTelemetryMeasurementData.json
EPCTelemetryData	Gives the EPC data like manufacturer, product SKU and serial number for a RFID tag.	EPCTelemetryData.xml EPCTelemetryData.json
GeoTelemetryData	Gives the geo telemetry data like latitude, longitude, altitude and datum for a RFID tag.	GeoTelemetryData.xml GeoTelemetryData.json
MotionTelemetryData	Gives the motion type telemetry information for a RFID tag.	MotionTelemetryData.xml MotionTelemetryData.xml

Schema Description for Location API Related Objects

WirelessClientApiQueryParams

WirelessClientApiQueryParams.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="wirelessClientApiQueryParams">
    <xs:complexContent>
      <xs:extension base="locationApiQueryParams">
        <xs:sequence>
          <xs:element name="associatedApMac" type="xs:string" minOccurs="0"/>
          <xs:element name="dot11Status" type="clientDot11Status" minOccurs="0"/>
          <xs:element name="ssid" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="locationApiQueryParams">
    <xs:complexContent>
      <xs:extension base="apiQueryParams">
        <xs:sequence>
          <xs:element name="currentlyTracked" type="xs:boolean"/>
          <xs:element name="locatedAfterTime" type="xs:string" minOccurs="0"/>
          <xs:element name="locatedBeforeTime" type="xs:string" minOccurs="0"/>
          <xs:element name="mapHierarchy" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="apiQueryParams">
    <xs:sequence>
      <xs:element name="page" type="xs:int"/>
      <xs:element name="pageSize" type="xs:int"/>
      <xs:element name="sortBy" type="xs:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="clientDot11Status">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="ASSOCIATED"/>
      <xs:enumeration value="PROBING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

WirelessClientApiqueryparams.json

```

{
  "type" : "object",
  "properties" : {
    "sortBy" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "page" : {
      "type" : "integer"
    },
    "pageSize" : {
      "type" : "integer"
    },
    "mapHierarchy" : {
      "type" : "string"
    },
    "currentlyTracked" : {
      "type" : "boolean",
      "required" : true
    },
    "locatedAfterTime" : {
      "type" : "string"
    },
    "locatedBeforeTime" : {

```

```

        "type" : "string"
    },
    "dot11Status" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "ASSOCIATED", "PROBING" ]
    },
    "associatedApMac" : {
        "type" : "string"
    },
    "ssid" : {
        "type" : "string"
    }
}
}
}

```

TagApiQueryParams

TagApiQueryParams.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="tagApiQueryParams">
    <xs:complexContent>
      <xs:extension base="locationApiQueryParams">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="locationApiQueryParams">
    <xs:complexContent>
      <xs:extension base="apiQueryParams">
        <xs:sequence>
          <xs:element name="currentlyTracked" type="xs:boolean"/>
          <xs:element name="locatedAfterTime" type="xs:string" minOccurs="0"/>
          <xs:element name="locatedBeforeTime" type="xs:string" minOccurs="0"/>
          <xs:element name="mapHierarchy" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="apiQueryParams">
    <xs:sequence>
      <xs:element name="page" type="xs:int"/>
      <xs:element name="pageSize" type="xs:int"/>
      <xs:element name="sortBy" type="xs:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

TagApiQueryParams.json

```

{
  "type" : "object",
  "properties" : {
    "sortBy" : {

```



```

        "type" : "array",
        "items" : {
            "type" : "string"
        }
    },
    "page" : {
        "type" : "integer"
    },
    "pageSize" : {
        "type" : "integer"
    },
    "mapHierarchy" : {
        "type" : "string"
    },
    "currentlyTracked" : {
        "type" : "boolean",
        "required" : true
    },
    "locatedAfterTime" : {
        "type" : "string"
    },
    "locatedBeforeTime" : {
        "type" : "string"
    }
}
}
}

```

RogueClientApiQueryParams

RogueClientApiQueryParams.xml

```

<xs:complexType name="locationApiQueryParams">
  <xs:complexContent>
    <xs:extension base="apiQueryParams">
      <xs:sequence>
        <xs:element name="currentlyTracked" type="xs:boolean"/>
        <xs:element name="locatedAfterTime" type="xs:string" minOccurs="0"/>
        <xs:element name="locatedBeforeTime" type="xs:string" minOccurs="0"/>
        <xs:element name="mapHierarchy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="apiQueryParams">
  <xs:sequence>
    <xs:element name="page" type="xs:int"/>
    <xs:element name="pageSize" type="xs:int"/>
    <xs:element name="sortBy" type="xs:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

RogueClientApiQueryParams.json

```

{
  "type" : "object",
  "properties" : {

```

```

    "sortBy" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "page" : {
      "type" : "integer"
    },
    "pageSize" : {
      "type" : "integer"
    },
    "mapHierarchy" : {
      "type" : "string"
    },
    "currentlyTracked" : {
      "type" : "boolean",
      "required" : true
    },
    "locatedAfterTime" : {
      "type" : "string"
    },
    "locatedBeforeTime" : {
      "type" : "string"
    }
  }
}

```

RogueAPApiQueryParams

RogueAPApiQueryParams

```

{
  <xs:complexType name="locationApiQueryParams">
    <xs:complexContent>
      <xs:extension base="apiQueryParams">
        <xs:sequence>
          <xs:element name="currentlyTracked" type="xs:boolean"/>
          <xs:element name="locatedAfterTime" type="xs:string" minOccurs="0"/>
          <xs:element name="locatedBeforeTime" type="xs:string" minOccurs="0"/>
          <xs:element name="mapHierarchy" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="apiQueryParams">
    <xs:sequence>
      <xs:element name="page" type="xs:int"/>
      <xs:element name="pageSize" type="xs:int"/>
      <xs:element name="sortBy" type="xs:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

RogueAPApiQueryParams.json

```

{
  "type" : "object",
  "properties" : {

```

```

    "sortBy" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "page" : {
      "type" : "integer"
    },
    "pageSize" : {
      "type" : "integer"
    },
    "mapHierarchy" : {
      "type" : "string"
    },
    "currentlyTracked" : {
      "type" : "boolean",
      "required" : true
    },
    "locatedAfterTime" : {
      "type" : "string"
    },
    "locatedBeforeTime" : {
      "type" : "string"
    }
  }
}

```

InterfererApiQueryParams

InterfererApiQueryParams.xml

```

{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:complexType name="interfererApiQueryParams">
      <xs:complexContent>
        <xs:extension base="locationApiQueryParams">
          <xs:sequence/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="locationApiQueryParams">
      <xs:complexContent>
        <xs:extension base="apiQueryParams">
          <xs:sequence>
            <xs:element name="currentlyTracked" type="xs:boolean"/>
            <xs:element name="locatedAfterTime" type="xs:string" minOccurs="0"/>
            <xs:element name="locatedBeforeTime" type="xs:string" minOccurs="0"/>
            <xs:element name="mapHierarchy" type="xs:string" minOccurs="0"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="apiQueryParams">
      <xs:sequence>
        <xs:element name="page" type="xs:int"/>
        <xs:element name="pageSize" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
}

```

```

        <xs:element name="sortBy" type="xs:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

InterfererApiQueryParams.json

```

{
  "type" : "object",
  "properties" : {
    "sortBy" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "page" : {
      "type" : "integer"
    },
    "pageSize" : {
      "type" : "integer"
    },
    "mapHierarchy" : {
      "type" : "string"
    },
    "currentlyTracked" : {
      "type" : "boolean",
      "required" : true
    },
    "locatedAfterTime" : {
      "type" : "string"
    },
    "locatedBeforeTime" : {
      "type" : "string"
    }
  }
}

```

DeviceCount

DeviceCount.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="DeviceCount" type="deviceCount"/>

  <xs:complexType name="deviceCount">
    <xs:sequence/>
    <xs:attribute name="deviceType" type="deviceType"/>
    <xs:attribute name="deviceQueryString" type="xs:string"/>
    <xs:attribute name="count" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:simpleType name="deviceType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Unknown"/>
      <xs:enumeration value="Wired_Client"/>
      <xs:enumeration value="Tag"/>
    </xs:restriction>
  </xs:simpleType>

```

```

    <xs:enumeration value="Rogue"/>
    <xs:enumeration value="Interferer"/>
    <xs:enumeration value="WiredSwitch"/>
    <xs:enumeration value="WirelessController"/>
    <xs:enumeration value="AccessPoint"/>
    <xs:enumeration value="Wireless_Client"/>
    <xs:enumeration value="NgwcSwitch"/>
    <xs:enumeration value="Rogue_AP"/>
    <xs:enumeration value="Rogue_Client"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

DeviceCount.json

```

{
  "type" : "object",
  "properties" : {
    "deviceType" : {
      "type" : "string",
      "enum" : [ "Unknown", "Wired_Client", "Tag", "Rogue", "Interferer", "WiredSwitch",
"WirelessController", "AccessPoint", "Wireless_Client", "NgwcSwitch", "Rogue_AP",
"Rogue_Client" ]
    },
    "deviceQueryString" : {
      "type" : "string"
    },
    "count" : {
      "type" : "integer"
    }
  }
}

```

Locations

Locations.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns2="GeoCoordinate" xmlns:ns1="MapCoordinate"
xmlns:ns4="RogueClientLocation" xmlns:ns3="Dimension"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns5="RogueAPLocation">

  <xs:import namespace="MapCoordinate" schemaLocation="Locations.xsd"/>

  <xs:import namespace="GeoCoordinate" schemaLocation="Locations.xsd"/>

  <xs:import namespace="Dimension" schemaLocation="Locations.xsd"/>

  <xs:import namespace="RogueClientLocation" schemaLocation="Locations.xsd"/>

  <xs:import namespace="RogueAPLocation" schemaLocation="Locations.xsd"/>

  <xs:element name="BatteryInfo" type="batteryInfo"/>

  <xs:element name="DeviceExciter" type="deviceExciter"/>

  <xs:element name="Dimension" type="dimension"/>

  <xs:element name="EPCTelemetryData" type="epcTelemetryData"/>

```

```

<xs:element name="EmergencyInfo" type="emergencyInfo"/>
<xs:element name="GeoCoordinate" type="geoCoordinate"/>
<xs:element name="GeoTelemetryData" type="geoTelemetryData"/>
<xs:element name="Image" type="image"/>
<xs:element name="InterfererLocation" type="interfererLocation"/>
<xs:element name="Locations" type="locations"/>
<xs:element name="MapCoordinate" type="mapCoordinate"/>
<xs:element name="MapInfo" type="mapInfo"/>
<xs:element name="MotionTelemetryData" type="motionTelemetryData"/>
<xs:element name="RogueAPLocation" type="rogueAPLocation"/>
<xs:element name="RogueClientLocation" type="rogueClientLocation"/>
<xs:element name="Statistics" type="statistics"/>
<xs:element name="TagLocation" type="tagLocation"/>
<xs:element name="TelemetryData" type="defaultTelemetryMeasurementData"/>
<xs:element name="VendorData" nillable="true" type="xs:anyType"/>
<xs:element name="WiredClientLocation" type="wiredClientLocation"/>
<xs:element name="WirelessClientLocation" type="wirelessClientLocation"/>
<xs:element name="location" type="location"/>
<xs:complexType name="locations">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="location"/>
      <xs:element ref="WirelessClientLocation"/>
      <xs:element ref="TagLocation"/>
      <xs:element ref="InterfererLocation"/>
      <xs:element ref="RogueClientLocation"/>
      <xs:element ref="RogueAPLocation"/>
      <xs:element ref="WiredClientLocation"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="totalPages" type="xs:int" use="required"/>
  <xs:attribute name="currentPage" type="xs:int" use="required"/>
  <xs:attribute name="pageSize" type="xs:int" use="required"/>
  <xs:attribute name="nextResourceURI" type="xs:string"/>
</xs:complexType>
<xs:complexType name="location" abstract="true">
  <xs:sequence>
    <xs:element ref="MapInfo" minOccurs="0"/>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
    <xs:element ref="Statistics" minOccurs="0"/>
    <xs:element ref="GeoCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="macAddress" type="xs:string"/>
  <xs:attribute name="currentlyTracked" type="xs:boolean" use="required"/>
  <xs:attribute name="confidenceFactor" type="xs:float" use="required"/>

```

```

    <xs:attribute name="historyLogReason" type="historyReason" />
  </xs:complexType>

  <xs:complexType name="mapInfo">
    <xs:sequence>
      <xs:element ref="Dimension" minOccurs="0" />
      <xs:element ref="Image" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="mapHierarchyString" type="xs:string" />
    <xs:attribute name="floorRefId" type="xs:long" use="required" />
  </xs:complexType>

  <xs:complexType name="dimension">
    <xs:sequence />
    <xs:attribute name="length" type="xs:float" use="required" />
    <xs:attribute name="width" type="xs:float" use="required" />
    <xs:attribute name="height" type="xs:float" use="required" />
    <xs:attribute name="offsetX" type="xs:float" use="required" />
    <xs:attribute name="offsetY" type="xs:float" use="required" />
    <xs:attribute name="unit" type="ns3:Unit" />
  </xs:complexType>

  <xs:complexType name="image">
    <xs:sequence>
      <xs:element name="imageData" type="xs:base64Binary" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="imageName" type="xs:string" />
  </xs:complexType>

  <xs:complexType name="mapCoordinate">
    <xs:sequence />
    <xs:attribute name="x" type="xs:float" use="required" />
    <xs:attribute name="y" type="xs:float" use="required" />
    <xs:attribute name="unit" type="ns1:Unit" />
  </xs:complexType>

  <xs:complexType name="statistics">
    <xs:sequence />
    <xs:attribute name="currentServerTime" type="xs:string" />
    <xs:attribute name="firstLocatedTime" type="xs:string" />
    <xs:attribute name="lastLocatedTime" type="xs:string" />
  </xs:complexType>

  <xs:complexType name="geoCoordinate">
    <xs:sequence />
    <xs:attribute name="latitude" type="xs:double" use="required" />
    <xs:attribute name="longitude" type="xs:double" use="required" />
    <xs:attribute name="unit" type="ns2:Unit" />
  </xs:complexType>

  <xs:complexType name="wirelessClientLocation">
    <xs:complexContent>
      <xs:extension base="location">
        <xs:sequence>
          <xs:attribute name="ipAddress">
            <xs:simpleType>
              <xs:list itemType="xs:string" />
            </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="userName" type="xs:string" />
          <xs:attribute name="ssid" type="xs:string" />
          <xs:attribute name="band" type="band" />
          <xs:attribute name="apMacAddress" type="xs:string" />
          <xs:attribute name="isGuestUser" type="xs:boolean" use="required" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

        <xs:attribute name="dot11Status" type="clientDot11Status"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="tagLocation">
    <xs:complexContent>
        <xs:extension base="location">
            <xs:sequence>
                <xs:element name="TelemetryData" type="telemetryInfo" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element ref="BatteryInfo" minOccurs="0"/>
                <xs:element ref="VendorData" minOccurs="0"/>
                <xs:element ref="EmergencyInfo" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="ExciterEncounter" type="deviceExciter" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="lastBeaconTime" type="xs:string"/>
            <xs:attribute name="lastBeaconSequenceNumber" type="xs:int" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="telemetryInfo">
    <xs:choice>
        <xs:element ref="TelemetryData"/>
        <xs:element ref="EPCTelemetryData"/>
        <xs:element ref="GeoTelemetryData"/>
        <xs:element ref="MotionTelemetryData"/>
    </xs:choice>
    <xs:attribute name="type" type="telemetryType"/>
    <xs:attribute name="units" type="xs:string"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="telemetryMeasurementData" abstract="true">
    <xs:sequence>
        <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="defaultTelemetryMeasurementData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence>
                <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="measurementDataType" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="epcTelemetryData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence/>
            <xs:attribute name="header" type="xs:string"/>
            <xs:attribute name="manufacturer" type="xs:string"/>
            <xs:attribute name="productSKU" type="xs:string"/>
            <xs:attribute name="serialNo" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```



```

<xs:complexType name="geoTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence/>
      <xs:attribute name="latitude" type="xs:string"/>
      <xs:attribute name="longitude" type="xs:string"/>
      <xs:attribute name="altitude" type="xs:string"/>
      <xs:attribute name="altitudeType" type="altitudeType"/>
      <xs:attribute name="datum" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="motionTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence/>
      <xs:attribute name="motionType" type="motionType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryInfo">
  <xs:sequence/>
  <xs:attribute name="tolerance" type="xs:int"/>
  <xs:attribute name="percentRemaining" type="xs:int"/>
  <xs:attribute name="daysRemaining" type="xs:int"/>
  <xs:attribute name="batteryAge" type="xs:long"/>
  <xs:attribute name="lastReceivedTime" type="xs:string"/>
  <xs:attribute name="lastReceivedSeqNum" type="xs:int"/>
</xs:complexType>

<xs:complexType name="vendorData">
  <xs:sequence/>
  <xs:attribute name="vendorId" type="xs:int" use="required"/>
  <xs:attribute name="elementId" type="xs:short" use="required"/>
  <xs:attribute name="data" type="xs:base64Binary"/>
  <xs:attribute name="lastReceivedTime" type="xs:string"/>
  <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="emergencyInfo">
  <xs:sequence/>
  <xs:attribute name="isActive" type="xs:boolean" use="required"/>
  <xs:attribute name="reason" type="xs:string"/>
  <xs:attribute name="tamperState" type="xs:string"/>
  <xs:attribute name="tamperOpenedCount" type="xs:int" use="required"/>
  <xs:attribute name="lastReceivedTime" type="xs:string"/>
  <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
  <xs:attribute name="fragmentNumber" type="xs:byte" use="required"/>
</xs:complexType>

<xs:complexType name="deviceExciter">
  <xs:sequence/>
  <xs:attribute name="macAddress" type="xs:string"/>
  <xs:attribute name="timestamp" type="xs:string"/>
  <xs:attribute name="inRange" type="xs:boolean" use="required"/>
  <xs:attribute name="isPerimeter" type="xs:boolean" use="required"/>
  <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
  <xs:attribute name="fragmentNumber" type="xs:byte" use="required"/>
  <xs:attribute name="isActive" type="xs:boolean" use="required"/>
  <xs:attribute name="assocFloorId" type="xs:long" use="required"/>
  <xs:attribute name="index" type="xs:short" use="required"/>

```

```

</xs:complexType>

<xs:complexType name="interfererLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence/>
      <xs:attribute name="deviceType" type="interfererDeviceType"/>
      <xs:attribute name="zoneOfImpact" type="xs:float" use="required"/>
      <xs:attribute name="dutyCycle" type="xs:short" use="required"/>
      <xs:attribute name="severity" type="xs:short" use="required"/>
      <xs:attribute name="deviceKey" type="xs:string"/>
      <xs:attribute name="affectedChannels" type="xs:string"/>
      <xs:attribute name="affectedBand" type="band"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="rogueClientLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence/>
      <xs:attribute name="numOfDetectingAPs" type="xs:int" use="required"/>
      <xs:attribute name="bssid" type="xs:string"/>
      <xs:attribute name="containmentLevel" type="ns4:ContainmentLevel"/>
      <xs:attribute name="state" type="ns4:State"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="rogueAPLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence/>
      <xs:attribute name="totalClients" type="xs:int" use="required"/>
      <xs:attribute name="numOfDetectingAPs" type="xs:int" use="required"/>
      <xs:attribute name="type" type="ns5:Type"/>
      <xs:attribute name="containmentLevel" type="ns5:ContainmentLevel"/>
      <xs:attribute name="state" type="ns5:State"/>
      <xs:attribute name="maxDetectedRssi" type="xs:int" use="required"/>
      <xs:attribute name="ssid" type="xs:string"/>
      <xs:attribute name="maxDetectingAPMacAddress" type="xs:string"/>
      <xs:attribute name="maxRssiRadioType" type="xs:int" use="required"/>
      <xs:attribute name="zoneOfImpact" type="xs:float" use="required"/>
      <xs:attribute name="hiddenSSID" type="xs:int" use="required"/>
      <xs:attribute name="rogueStChBy" type="ns5:RogueState"/>
      <xs:attribute name="rogueClChBy" type="ns5:RogueClass"/>
      <xs:attribute name="classifiedRSSI" type="xs:int" use="required"/>
      <xs:attribute name="classifiedByAP" type="xs:string"/>
      <xs:attribute name="className" type="xs:string"/>
      <xs:attribute name="classification" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="wiredClientLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence>
        <xs:element name="civicAddressAsString" type="xs:string" minOccurs="0"/>
        <xs:element name="PortAssociation" type="wiredPortAssociation" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="username802Dot1X" type="xs:string"/>
      <xs:attribute name="ipAddress">

```

```

        <xs:simpleType>
          <xs:list itemType="xs:string" />
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="elin" type="xs:string" />
      <xs:attribute name="serialNumber" type="xs:string" />
      <xs:attribute name="udi" type="xs:string" />
      <xs:attribute name="vlanId" type="xs:int" use="required" />
      <xs:attribute name="vlanName" type="xs:string" />
      <xs:attribute name="switchIpAddress" type="xs:int" use="required" />
      <xs:attribute name="portNumber" type="xs:short" use="required" />
      <xs:attribute name="slotNumber" type="xs:short" use="required" />
      <xs:attribute name="state" type="associationState" />
      <xs:attribute name="portType" type="xs:byte" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="wiredPortAssociation">
  <xs:complexContent>
    <xs:extension base="aesObject">
      <xs:sequence>
        <xs:element name="duplicate" type="xs:boolean" />
        <xs:element name="moduleNumber" type="xs:short" />
        <xs:element name="portId" type="xs:long" />
        <xs:element name="portNumber" type="xs:short" />
        <xs:element name="portType" type="xs:byte" />
        <xs:element name="slotNumber" type="xs:short" />
        <xs:element name="state" type="xs:byte" />
        <xs:element name="switchIpAddress" type="xs:string" minOccurs="0" />
        <xs:element name="transitionTime" type="xs:long" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="aesObject" abstract="true">
  <xs:sequence>
    <xs:element name="changedOn" type="xs:long" />
    <xs:element name="mask" type="xs:int" />
    <xs:element name="objectId" type="xs:long" />
    <xs:element name="options" type="xs:short" />
    <xs:element name="parentId" type="xs:long" />
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="historyReason">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="FLOOR_CHANGE" />
    <xs:enumeration value="CHOKEPOINT_ENCOUNTER" />
    <xs:enumeration value="INOUT_OF_AREA" />
    <xs:enumeration value="SIGNIFICANT_MOVEMENT" />
    <xs:enumeration value="LOCP_MESSAGE" />
    <xs:enumeration value="INFORMATION_CHANGE" />
    <xs:enumeration value="DEVICE_TYPE_CHANGE" />
    <xs:enumeration value="TAG_EMERGENCY" />
    <xs:enumeration value="LOCATION_CONTRIBUTOR_CHANGE" />
    <xs:enumeration value="INTERFERER_MERGE" />
    <xs:enumeration value="INTERFERER_CLUSTER_CENTRE_CHANGE" />
    <xs:enumeration value="INTERFERER_SEVERITY_CHANGE" />
    <xs:enumeration value="STATUS_CHANGE" />
    <xs:enumeration value="SECURITY_POLICY_CHANGE" />
    <xs:enumeration value="NETWORK_STATUS_CHANGE" />
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="HASHKEY_CHANGE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="band">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="IEEE_802_11_A" />
        <xs:enumeration value="IEEE_802_11_B" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="clientDot11Status">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="ASSOCIATED" />
        <xs:enumeration value="PROBING" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="telemetryType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="GPS" />
        <xs:enumeration value="TEMPERATURE" />
        <xs:enumeration value="MOTION" />
        <xs:enumeration value="DISTANCE" />
        <xs:enumeration value="FUEL" />
        <xs:enumeration value="QUANTITY" />
        <xs:enumeration value="PRESSURE" />
        <xs:enumeration value="HUMIDITY" />
        <xs:enumeration value="STATUS" />
        <xs:enumeration value="EPC" />
        <xs:enumeration value="MOTIONPROB" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="altitudeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="AT_METERS" />
        <xs:enumeration value="AT_FLOOR" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="motionType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="NO_MOVEMENT" />
        <xs:enumeration value="ACCELERATION" />
        <xs:enumeration value="MOVEMENT" />
        <xs:enumeration value="DECELERATION" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="interfererDeviceType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="BLUETOOTH" />
        <xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_1" />
        <xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_1" />
        <xs:enumeration value="TDD_1_PHONE" />
        <xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_2" />
        <xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_2" />
    </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="MICROWAVE_OVEN_1"/>
<xs:enumeration value="DEVICE_802_11_FH"/>
<xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_3"/>
<xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_3"/>
<xs:enumeration value="MICROWAVE_OVEN_2"/>
<xs:enumeration value="BLUETOOTH_PAGING_INQUIRY_DEVICE"/>
<xs:enumeration value="GENERIC_TDD"/>
<xs:enumeration value="JAMMER"/>
<xs:enumeration value="RFID_READER"/>
<xs:enumeration value="CLASS_1_NETWORK"/>
<xs:enumeration value="CLASS_2_NETWORK"/>
<xs:enumeration value="GENERIC_WIDEBAND"/>
<xs:enumeration value="BRIDGE_ROUTER_1"/>
<xs:enumeration value="GENERIC_WAVEFORM"/>
<xs:enumeration value="GENERIC_DECT_LIKE"/>
<xs:enumeration value="VIDEO"/>
<xs:enumeration value="DEVICE_802_15_4"/>
<xs:enumeration value="DEVICE_802_11_NORMAL"/>
<xs:enumeration value="DEVICE_802_11_SPECTRALLY_INVERTED"/>
<xs:enumeration value="DEVICE_802_11_NONSTD_CHANNEL"/>
<xs:enumeration value="DEVICE_802_11_SUPERG"/>
<xs:enumeration value="RADAR"/>
<xs:enumeration value="MOTOROLA_CANOPY"/>
<xs:enumeration value="XBOX"/>
<xs:enumeration value="WIFI_MOBILE"/>
<xs:enumeration value="WIFI_FIXED"/>
<xs:enumeration value="EXALT"/>
<xs:enumeration value="CONTINUOUS_TRANSMITTER"/>
<xs:enumeration value="DECT_LIKE_NETWORK"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="associationState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CONNECTED"/>
    <xs:enumeration value="DISCONNECTED"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Locations.json

```

{
  "type" : "object",
  "properties" : {
    "totalPages" : {
      "type" : "integer"
    },
    "currentPage" : {
      "type" : "integer"
    },
    "pageSize" : {
      "type" : "integer"
    },
    "nextResourceURI" : {
      "type" : "string"
    },
    "locations" : {
      "type" : "array",
      "items" : {
        "type" : "any"
      }
    }
  }
}

```

```

    }
  }
}

```

MapInfo

MapInfo.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="Dimension"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

MapInfo.json

```

{
  "type" : "object",
  "properties" : {
    "mapHierarchyString" : {
      "type" : "string"
    },
    "floorRefId" : {
      "type" : "number"
    },
    "floorDimension" : {
      "type" : "object",
      "properties" : {
        "length" : {
          "type" : "number"
        },
        "width" : {
          "type" : "number"
        },
        "height" : {
          "type" : "number"
        },
        "offsetX" : {
          "type" : "number"
        },
        "offsetY" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "image" : {
      "type" : "object",
      "properties" : {
        "imageName" : {
          "type" : "string"
        }
      }
    }
  }
}

```

```

    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  }
}
}
}
}
}

```

Map-info.proto

```

option java_package = "com.cisco.mse.cas.maps";
option java_outer_classname = "MapInfoProtos";

import "dimension.proto";
import "image.proto";

message PMapInfo {

  optional string mapHierarchyString = 1;

  optional int64 floorRefId = 2;

  optional PBDimension dimension = 3;

  optional PBIImage image = 4;

}

```

MapCoordinate

MapCoordinate.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns1="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="MapCoordinate" schemaLocation="MapCoordinate.xsd"/>

  <xs:element name="MapCoordinate" type="mapCoordinate"/>

  <xs:complexType name="mapCoordinate">
    <xs:sequence/>
    <xs:attribute name="x" type="xs:float" use="required"/>
    <xs:attribute name="y" type="xs:float" use="required"/>
    <xs:attribute name="unit" type="ns1:Unit"/>
  </xs:complexType>
</xs:schema>

```

MapCoordinate.json

```

{
  "type" : "object",
  "properties" : {
    "x" : {
      "type" : "number"
    }
  }
}

```

```

    },
    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
}

```

MapCoordinate.Proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "MapCoordinateProtos";

message PMapCoordinate {

  enum PUnit {
    FEET = 0;
    METER = 1;
  }

  required float x = 1;
  required float y = 2;

  optional PUnit unit = 3 [default = FEET];
}

```

GeoCoordinate

GeoCoordinate.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns1="GeoCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="GeoCoordinate" schemaLocation="GeoCoordinate.xsd"/>

  <xs:element name="GeoCoordinate" type="geoCoordinate"/>

  <xs:complexType name="geoCoordinate">
    <xs:sequence/>
    <xs:attribute name="latitude" type="xs:double" use="required"/>
    <xs:attribute name="longitude" type="xs:double" use="required"/>
    <xs:attribute name="unit" type="ns1:Unit"/>
  </xs:complexType>
</xs:schema>

```

GeoCoordinate.json

```

{
  "type" : "object",
  "properties" : {
    "latitude" : {
      "type" : "number"
    },
    "longitude" : {
      "type" : "number"
    }
  }
}

```



```

    },
    "unit" : {
      "type" : "string",
      "enum" : [ "DEGREES" ]
    }
  }
}

```

geo-coordinate.proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "GeoCoordinateProtos";

message PBGeoCoordinate {

  enum PBUUnit {
    DEGREES = 0;
  }

  required double latitude = 1;

  required double longitude = 2;

  optional PBUUnit unit = 3 [default = DEGREES];
}

```

WirelessClientLocation

WirelessClientLocation.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

WirelessClientLocation.json

```

{
  "type" : "object",
  "properties" : {
    "macAddress" : {
      "type" : "string"
    },
    "mapInfo" : {
      "type" : "object",
      "properties" : {
        "mapHierarchyString" : {
          "type" : "string"
        },
        "floorRefId" : {
          "type" : "number"
        }
      }
    }
  }
}

```

```

    "floorDimension" : {
      "type" : "object",
      "properties" : {
        "length" : {
          "type" : "number"
        },
        "width" : {
          "type" : "number"
        },
        "height" : {
          "type" : "number"
        },
        "offsetX" : {
          "type" : "number"
        },
        "offsetY" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "image" : {
      "type" : "object",
      "properties" : {
        "imageName" : {
          "type" : "string"
        },
        "sourceFile" : {
          "type" : "array",
          "items" : {
            "type" : "string"
          }
        }
      }
    }
  },
  "mapCoordinate" : {
    "type" : "object",
    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  },
  "currentlyTracked" : {
    "type" : "boolean",
    "required" : true
  },
  "confidenceFactor" : {
    "type" : "number"
  },
  "statistics" : {
    "type" : "object",

```

```

    "properties" : {
      "currentServerTime" : {
        "type" : "string"
      },
      "firstLocatedTime" : {
        "type" : "string"
      },
      "lastLocatedTime" : {
        "type" : "string"
      }
    }
  },
  "historyLogReason" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "FLOOR_CHANGE", "CHOKEPOINT_ENCOUNTER", "INOUT_OF_AREA",
"SIGNIFICANT_MOVEMENT", "LOCP_MESSAGE", "INFORMATION_CHANGE", "DEVICE_TYPE_CHANGE",
"TAG_EMERGENCY", "LOCATION_CONTRIBUTOR_CHANGE", "INTERFERER_MERGE",
"INTERFERER_CLUSTER_CENTRE_CHANGE", "INTERFERER_SEVERITY_CHANGE", "STATUS_CHANGE",
"SECURITY_POLICY_CHANGE", "NETWORK_STATUS_CHANGE", "HASHKEY_CHANGE" ]
  },
  "geoCoordinate" : {
    "type" : "object",
    "properties" : {
      "latitude" : {
        "type" : "number"
      },
      "longitude" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "DEGREES" ]
      }
    }
  },
  "ipAddress" : {
    "type" : "array",
    "items" : {
      "type" : "string"
    }
  },
  "userName" : {
    "type" : "string"
  },
  "ssId" : {
    "type" : "string"
  },
  "band" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
  },
  "apMacAddress" : {
    "type" : "string"
  },
  "dot11Status" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "ASSOCIATED", "PROBING" ]
  },
  "guestUser" : {
    "type" : "boolean",
    "required" : true
  }
}
}
}

```

WirelessClientLocation.proto

```
option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "WirelessClientProtos";

import "dimension.proto";
import "image.proto";
import "map-coordinate.proto";
import "geo-coordinate.proto";
import "map-info.proto";
import "history-reason.proto";
import "statistics.proto";
import "band.proto";

message PBWirelessClientLocation {

    enum PBClientDot11Status {
        ASSOCIATED = 0;

        PROBING = 1;
    }

    required string macAddress = 1;

    optional PBMapInfo mapInfo = 2;

    optional PBMapCoordinate mapCoordinate = 3;

    optional bool currentlyTracked = 4;

    optional float confidenceFactor = 5;

    optional PBStatistics statistics = 6;

    optional PBHistoryReason historyLogReason = 7;

    optional PBGeoCoordinate geoCoordinate = 8;

    repeated string ipAddress = 9;

    optional string userName = 10;

    optional string ssId = 11;

    optional PBBand band = 12;

    optional string apMacAddress = 13;

    optional bool isGuestUser = 14;

    optional PBClientDot11Status dot11Status = 15 [default = PROBING];
}
```

HistoryReason

HistoryReason.xml

HistoryReason.json

HistoryReason.Proto

```
option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "HistoryReasonProtos";

enum PBHistoryReason {
  UNKNOWN = 0;
  FLOOR_CHANGE = 1;
  CHOKEPOINT_ENCOUNTER = 2;
  SIGNIFICANT_MOVEMENT = 4;
  INFORMATION_CHANGE = 6;
  DEVICE_TYPE_CHANGE = 7;
  TAG_EMERGENCY = 8;
  LOCATION_CONTRIBUTOR_CHANGE = 9;
  INTERFERER_MERGE = 10;
  INTERFERER_CLUSTER_CENTRE_CHANGE = 11;
  INTERFERER_SEVERITY_CHANGE = 12;
  STATUS_CHANGE = 13;
  SECURITY_POLICY_CHANGE = 14;
  NETWORK_STATUS_CHANGE = 15;
}
```

Statistics

Statistics.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="Statistics" type="statistics"/>

  <xs:complexType name="statistics">
    <xs:sequence/>
    <xs:attribute name="currentServerTime" type="xs:string"/>
    <xs:attribute name="firstLocatedTime" type="xs:string"/>
    <xs:attribute name="lastLocatedTime" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

statistics.json

```
{
  "type" : "object",
  "properties" : {
    "currentServerTime" : {
      "type" : "string"
    },
    "firstLocatedTime" : {
      "type" : "string"
    },
    "lastLocatedTime" : {
      "type" : "string"
    }
  }
}
```

```

    }
}

```

statistics.proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "StatisticsProtos";

message PBStatistics {
    optional uint64 currentServerTimeInMillis = 1;

    optional uint64 firstLocatedTimeInMillis = 2;

    optional uint64 lastLocatedTimeInMillis = 3;
}

```

RogueAPLocation

RogueAPLocation.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="RogueAPLocation"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="AP"/>
      <xs:enumeration value="ADHOC"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ContainmentLevel">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="UNASSIGNED"/>
      <xs:enumeration value="LEVEL1"/>
      <xs:enumeration value="LEVEL2"/>
      <xs:enumeration value="LEVEL3"/>
      <xs:enumeration value="LEVEL4"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="State">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="INITIALIZING"/>
      <xs:enumeration value="PENDING"/>
      <xs:enumeration value="ALERT"/>
      <xs:enumeration value="DETECTEDLRAD"/>
      <xs:enumeration value="KNOWN"/>
      <xs:enumeration value="ACKNOWLEDGE"/>
      <xs:enumeration value="CONTAINED"/>
      <xs:enumeration value="THREAT"/>
      <xs:enumeration value="CONTAINEDPENDING"/>
      <xs:enumeration value="KNOWNCONTAINED"/>
      <xs:enumeration value="TRUSTEDMISSING"/>
    </xs:restriction>
  </xs:simpleType>

```

```

<xs:simpleType name="RogueState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DEFAULT"/>
    <xs:enumeration value="MANUAL"/>
    <xs:enumeration value="AUTO"/>
    <xs:enumeration value="ROGUE_RULE"/>
    <xs:enumeration value="AWIPS"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="RogueClass">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DEFAULT"/>
    <xs:enumeration value="MANUAL"/>
    <xs:enumeration value="AUTO"/>
    <xs:enumeration value="ROGUE_RULE"/>
    <xs:enumeration value="AWIPS"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

RogueAPLocation.json

```

{
  "type" : "object",
  "properties" : {
    "macAddress" : {
      "type" : "string"
    },
    "mapInfo" : {
      "type" : "object",
      "properties" : {
        "mapHierarchyString" : {
          "type" : "string"
        },
        "floorRefId" : {
          "type" : "number"
        },
        "floorDimension" : {
          "type" : "object",
          "properties" : {
            "length" : {
              "type" : "number"
            },
            "width" : {
              "type" : "number"
            },
            "height" : {
              "type" : "number"
            },
            "offsetX" : {
              "type" : "number"
            },
            "offsetY" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        },
        "image" : {
          "type" : "object",

```

```

        "properties" : {
            "imageName" : {
                "type" : "string"
            },
            "sourceFile" : {
                "type" : "array",
                "items" : {
                    "type" : "string"
                }
            }
        }
    },
    "mapCoordinate" : {
        "type" : "object",
        "properties" : {
            "x" : {
                "type" : "number"
            },
            "y" : {
                "type" : "number"
            },
            "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
            }
        }
    },
    "currentlyTracked" : {
        "type" : "boolean",
        "required" : true
    },
    "confidenceFactor" : {
        "type" : "number"
    },
    "statistics" : {
        "type" : "object",
        "properties" : {
            "currentServerTime" : {
                "type" : "string"
            },
            "firstLocatedTime" : {
                "type" : "string"
            },
            "lastLocatedTime" : {
                "type" : "string"
            }
        }
    },
    "historyLogReason" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "FLOOR_CHANGE", "CHOKEPOINT_ENCOUNTER", "INOUT_OF_AREA",
"SIGNIFICANT_MOVEMENT", "LOCP_MESSAGE", "INFORMATION_CHANGE", "DEVICE_TYPE_CHANGE",
"TAG_EMERGENCY", "LOCATION_CONTRIBUTOR_CHANGE", "INTERFERER_MERGE",
"INTERFERER_CLUSTER_CENTRE_CHANGE", "INTERFERER_SEVERITY_CHANGE", "STATUS_CHANGE",
"SECURITY_POLICY_CHANGE", "NETWORK_STATUS_CHANGE", "HASHKEY_CHANGE" ]
    },
    "geoCoordinate" : {
        "type" : "object",
        "properties" : {
            "latitude" : {
                "type" : "number"
            }
        }
    },

```



```

    "longitude" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "DEGREES" ]
    }
  }
},
"totalClients" : {
  "type" : "integer"
},
"numOfDetectingAPs" : {
  "type" : "integer"
},
"type" : {
  "type" : "string",
  "enum" : [ "UNKNOWN", "AP", "ADHOC" ]
},
"containmentLevel" : {
  "type" : "string",
  "enum" : [ "UNKNOWN", "UNASSIGNED", "LEVEL1", "LEVEL2", "LEVEL3", "LEVEL4" ]
},
"state" : {
  "type" : "string",
  "enum" : [ "UNKNOWN", "INITIALIZING", "PENDING", "ALERT", "DETECTEDLRAD", "KNOWN",
"ACKNOWLEDGE", "CONTAINED", "THREAT", "CONTAINEDPENDING", "KNOWNCONTAINED",
"TRUSTEDMISSING" ]
},
"maxDetectedRssi" : {
  "type" : "integer"
},
"ssid" : {
  "type" : "string"
},
"maxDetectingAPMacAddress" : {
  "type" : "string"
},
"maxRssiRadioType" : {
  "type" : "integer"
},
"zoneOfImpact" : {
  "type" : "number"
},
"hiddenSSID" : {
  "type" : "integer"
},
"rogueStChBy" : {
  "type" : "string",
  "enum" : [ "DEFAULT", "MANUAL", "AUTO", "ROGUE_RULE", "AWIPS" ]
},
"rogueClChBy" : {
  "type" : "string",
  "enum" : [ "DEFAULT", "MANUAL", "AUTO", "ROGUE_RULE", "AWIPS" ]
},
"classifiedRSSI" : {
  "type" : "integer"
},
"classifiedByAP" : {
  "type" : "string"
},
"className" : {
  "type" : "string"
},

```

```

        "classification" : {
            "type" : "integer"
        }
    }
}

```

rogue-ap.proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "RogueAPLocationProtos";

```

```

import "dimension.proto";
import "image.proto";
import "map-coordinate.proto";
import "geo-coordinate.proto";
import "map-info.proto";
import "history-reason.proto";
import "statistics.proto";

```

```

message PBRogueAPLocation {

    enum PBType {
        AP = 0;
        ADHOC = 1;
    }

    enum PBContainmentLevel {
        UNASSIGNED = 0;
        LEVEL1 = 1;
        LEVEL2 = 2;
        LEVEL3 = 3;
        LEVEL4 = 4;
    }

    enum PBState {
        INITIALIZING = 0;
        PENDING = 1;
        ALERT = 2;
        DETECTEDLRAD = 3;
        KNOWN = 4;
        ACKNOWLEDGE = 5;
        CONTAINED = 6;
        THREAT = 7;
        CONTAINEDPENDING = 8;
        KNOWNCONTAINED = 9;
        TRUSTEDMISSING = 10;
    }

    required string macAddress = 1;

    optional PMapInfo mapInfo = 2;

    optional PMapCoordinate mapCoordinate = 3;

    optional bool currentlyTracked = 4;

    optional float confidenceFactor = 5;

    optional PBStatistics statistics = 6;

    optional PBHistoryReason historyLogReason = 7;

    optional PGeoCoordinate geoCoordinate = 8;

```

```

optional int32 totalClients = 9;

optional int32 numOfDetectingAPs = 10;

optional PBType type = 11;

optional PBContainmentLevel containmentLevel = 12;

optional PBState state = 13;

optional int32 maxDetectedRssi = 14;

optional string ssid = 15;

optional string maxDetectingAPMacAddress = 16;

optional int32 maxRssiRadioType = 17;

optional float zoneOfImpact = 18;
}

```

RogueClientLocation

RogueClientLocation.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

RogueClientLocation.json

```

{
  "type" : "object",
  "properties" : {
    "macAddress" : {
      "type" : "string"
    },
    "mapInfo" : {
      "type" : "object",
      "properties" : {
        "mapHierarchyString" : {
          "type" : "string"
        },
        "floorRefId" : {
          "type" : "number"
        },
        "floorDimension" : {
          "type" : "object",
          "properties" : {
            "length" : {
              "type" : "number"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "width" : {
      "type" : "number"
    },
    "height" : {
      "type" : "number"
    },
    "offsetX" : {
      "type" : "number"
    },
    "offsetY" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"image" : {
  "type" : "object",
  "properties" : {
    "imageName" : {
      "type" : "string"
    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  }
}
},
"mapCoordinate" : {
  "type" : "object",
  "properties" : {
    "x" : {
      "type" : "number"
    },
    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"currentlyTracked" : {
  "type" : "boolean",
  "required" : true
},
"confidenceFactor" : {
  "type" : "number"
},
"statistics" : {
  "type" : "object",
  "properties" : {
    "currentServerTime" : {
      "type" : "string"
    },
    "firstLocatedTime" : {

```

```

        "type" : "string"
      },
      "lastLocatedTime" : {
        "type" : "string"
      }
    }
  },
  "historyLogReason" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "FLOOR_CHANGE", "CHOKEPOINT_ENCOUNTER", "INOUT_OF_AREA",
"SIGNIFICANT_MOVEMENT", "LOCP_MESSAGE", "INFORMATION_CHANGE", "DEVICE_TYPE_CHANGE",
"TAG_EMERGENCY", "LOCATION_CONTRIBUTOR_CHANGE", "INTERFERER_MERGE",
"INTERFERER_CLUSTER_CENTRE_CHANGE", "INTERFERER_SEVERITY_CHANGE", "STATUS_CHANGE",
"SECURITY_POLICY_CHANGE", "NETWORK_STATUS_CHANGE", "HASHKEY_CHANGE" ]
  },
  "geoCoordinate" : {
    "type" : "object",
    "properties" : {
      "latitude" : {
        "type" : "number"
      },
      "longitude" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "DEGREES" ]
      }
    }
  },
  "numOfDetectingAPs" : {
    "type" : "integer"
  },
  "bssid" : {
    "type" : "string"
  },
  "containmentLevel" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "UNASSIGNED", "LEVEL1", "LEVEL2", "LEVEL3", "LEVEL4" ]
  },
  "state" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "INITIALIZING", "PENDING", "ALERT", "CONTAINED", "THREAT",
"CONTAINEDPENDING" ]
  }
}
}
}

```

RogueClientLocation.Proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "RogueClientLocationProtos";

import "dimension.proto";
import "image.proto";
import "map-coordinate.proto";
import "geo-coordinate.proto";
import "map-info.proto";
import "history-reason.proto";
import "statistics.proto";

message PB RogueClientLocation {

  enum PBContainmentLevel {

```

```

        UNASSIGNED = 0;
        LEVEL1 = 1;
        LEVEL2 = 2;
        LEVEL3 = 3;
        LEVEL4 = 4;
    }

    enum PBState {
        INITIALIZING = 0;
        PENDING = 1;
        ALERT = 2;
        CONTAINED = 6;
        THREAT = 7;
        CONTAINEDPENDING = 8;
    }

    required string macAddress = 1;

    optional PBMapInfo mapInfo = 2;

    optional PBMapCoordinate mapCoordinate = 3;

    optional bool currentlyTracked = 4;

    optional float confidenceFactor = 5;

    optional PBStatistics statistics = 6;

    optional PBHistoryReason historyLogReason = 7;

    optional PBGeoCoordinate geoCoordinate = 8;

    optional int32 numOfDetectingAPs = 9;

    optional string bssid = 10;

    optional PBContainmentLevel containmentLevel = 11;

    optional PBState state = 12;
}

```

InterfererLocation

InterfererLocation.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="GeoCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:simpleType name="Unit">
        <xs:restriction base="xs:string">
            <xs:enumeration value="DEGREES"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>

```

InterfererLocation.json

```

{
    "type" : "object",

```

```

"properties" : {
  "macAddress" : {
    "type" : "string"
  },
  "mapInfo" : {
    "type" : "object",
    "properties" : {
      "mapHierarchyString" : {
        "type" : "string"
      },
      "floorRefId" : {
        "type" : "number"
      },
      "floorDimension" : {
        "type" : "object",
        "properties" : {
          "length" : {
            "type" : "number"
          },
          "width" : {
            "type" : "number"
          },
          "height" : {
            "type" : "number"
          },
          "offsetX" : {
            "type" : "number"
          },
          "offsetY" : {
            "type" : "number"
          },
          "unit" : {
            "type" : "string",
            "enum" : [ "FEET", "METER" ]
          }
        }
      }
    }
  },
  "image" : {
    "type" : "object",
    "properties" : {
      "imageName" : {
        "type" : "string"
      },
      "sourceFile" : {
        "type" : "array",
        "items" : {
          "type" : "string"
        }
      }
    }
  }
},
"mapCoordinate" : {
  "type" : "object",
  "properties" : {
    "x" : {
      "type" : "number"
    },
    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",

```

```

        "enum" : [ "FEET", "METER" ]
      }
    },
    "currentlyTracked" : {
      "type" : "boolean",
      "required" : true
    },
    "confidenceFactor" : {
      "type" : "number"
    },
    "statistics" : {
      "type" : "object",
      "properties" : {
        "currentServerTime" : {
          "type" : "string"
        },
        "firstLocatedTime" : {
          "type" : "string"
        },
        "lastLocatedTime" : {
          "type" : "string"
        }
      }
    },
    "historyLogReason" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "FLOOR_CHANGE", "CHOKEPOINT_ENCOUNTER", "INOUT_OF_AREA",
"SIGNIFICANT_MOVEMENT", "LOCP_MESSAGE", "INFORMATION_CHANGE", "DEVICE_TYPE_CHANGE",
"TAG_EMERGENCY", "LOCATION_CONTRIBUTOR_CHANGE", "INTERFERER_MERGE",
"INTERFERER_CLUSTER_CENTRE_CHANGE", "INTERFERER_SEVERITY_CHANGE", "STATUS_CHANGE",
"SECURITY_POLICY_CHANGE", "NETWORK_STATUS_CHANGE", "HASHKEY_CHANGE" ]
    },
    "geoCoordinate" : {
      "type" : "object",
      "properties" : {
        "latitude" : {
          "type" : "number"
        },
        "longitude" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "DEGREES" ]
        }
      }
    },
    "deviceType" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "BLUETOOTH", "DECT_LIKE_NETWORK_VARIANT_1",
"DECT_LIKE_BASE_STATION_VARIANT_1", "TDD_1_PHONE", "DECT_LIKE_NETWORK_VARIANT_2",
"DECT_LIKE_BASE_STATION_VARIANT_2", "MICROWAVE_OVEN_1", "DEVICE_802_11_FH",
"DECT_LIKE_NETWORK_VARIANT_3", "DECT_LIKE_BASE_STATION_VARIANT_3", "MICROWAVE_OVEN_2",
"BLUETOOTH_PAGING_INQUIRY_DEVICE", "GENERIC_TDD", "JAMMER", "RFID_READER",
"CLASS_1_NETWORK", "CLASS_2_NETWORK", "GENERIC_WIDEBAND", "BRIDGE_ROUTER_1",
"GENERIC_WAVEFORM", "GENERIC_DECT_LIKE", "VIDEO", "DEVICE_802_15_4",
"DEVICE_802_11_NORMAL", "DEVICE_802_11_SPECTRALLY_INVERTED",
"DEVICE_802_11_NONSTD_CHANNEL", "DEVICE_802_11_SUPERG", "RADAR", "MOTOROLA_CANOPY",
"XBOX", "WIFI_MOBILE", "WIFI_FIXED", "EXALT", "CONTINUOUS_TRANSMITTER",
"DECT_LIKE_NETWORK" ]
    },
    "zoneOfImpact" : {
      "type" : "number"
    }
  }
}

```



```

    },
    "dutyCycle" : {
      "type" : "integer"
    },
    "severity" : {
      "type" : "integer"
    },
    "deviceKey" : {
      "type" : "string"
    },
    "affectedChannels" : {
      "type" : "string"
    },
    "affectedBand" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
    }
  }
}

```

interferer.proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "InterfererLocationProtos";

```

```

import "dimension.proto";
import "image.proto";
import "map-coordinate.proto";
import "geo-coordinate.proto";
import "map-info.proto";
import "history-reason.proto";
import "statistics.proto";
import "band.proto";

```

```

message PBInterfererLocation {

  enum PBInterfererDeviceType {
    BLUETOOTH = 1;
    DECT_LIKE_NETWORK_VARIANT_1 = 2;
    DECT_LIKE_BASE_STATION_VARIANT_1 = 3;
    TDD_1_PHONE = 4;
    DECT_LIKE_NETWORK_VARIANT_2 = 6;
    DECT_LIKE_BASE_STATION_VARIANT_2 = 7;
    MICROWAVE_OVEN_1 = 8;
    DEVICE_802_11_FH = 10;
    DECT_LIKE_NETWORK_VARIANT_3 = 12;
    DECT_LIKE_BASE_STATION_VARIANT_3 = 13;
    MICROWAVE_OVEN_2 = 16;
    BLUETOOTH_PAGING_INQUIRY_DEVICE = 17;
    GENERIC_TDD = 18;
    JAMMER = 19;
    RFID_READER = 20;
    CLASS_1_NETWORK = 21;
    CLASS_2_NETWORK = 22;
    GENERIC_WIDEBAND = 23;
    BRIDGE_ROUTER_1 = 24;
    GENERIC_WAVEFORM = 25;
    GENERIC_DECT_LIKE = 26;
    VIDEO = 27;
    DEVICE_802_15_4 = 28;
    DEVICE_802_11_NORMAL = 29;
    DEVICE_802_11_SPECTRALLY_INVERTED = 30;
    DEVICE_802_11_NONSTD_CHANNEL = 31;
    DEVICE_802_11_SUPERG = 32;
    RADAR = 33;
  }
}

```

```

        MOTOROLA_CANOPY = 34;
        XBOX = 35;
        WIFI_MOBILE = 36;
        WIFI_FIXED = 37;
        EXALT = 38;
        CONTINUOUS_TRANSMITTER = 256;
        DECT_LIKE_NETWORK = 512;
    }

    required string macAddress = 1;

    optional PMapInfo mapInfo = 2;

    optional PMapCoordinate mapCoordinate = 3;

    optional bool currentlyTracked = 4;

    optional float confidenceFactor = 5;

    optional PStatistics statistics = 6;

    optional PHistoryReason historyLogReason = 7;

    optional PGeoCoordinate geoCoordinate = 8;

    optional PInterfererDeviceType deviceType = 9;

    optional float zoneOfImpact = 10;

    optional int32 dutyCycle = 11;

    optional int32 severity = 12;

    optional string deviceKey = 13;

    optional string affectedChannels = 14;

    optional PBand affectedBand = 15;
}

```

TagLocation

TagLocation.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="RogueClientLocation"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="ContainmentLevel">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="UNASSIGNED"/>
      <xs:enumeration value="LEVEL1"/>
      <xs:enumeration value="LEVEL2"/>
      <xs:enumeration value="LEVEL3"/>
      <xs:enumeration value="LEVEL4"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="State">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="INITIALIZING"/>
    </xs:restriction>
  </xs:simpleType>

```

```

<xs:enumeration value="PENDING"/>
<xs:enumeration value="ALERT"/>
<xs:enumeration value="CONTAINED"/>
<xs:enumeration value="THREAT"/>
<xs:enumeration value="CONTAINEDPENDING"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>

```

TagLocation.json

```

{
  "type" : "object",
  "properties" : {
    "macAddress" : {
      "type" : "string"
    },
    "mapInfo" : {
      "type" : "object",
      "properties" : {
        "mapHierarchyString" : {
          "type" : "string"
        },
        "floorRefId" : {
          "type" : "number"
        },
        "floorDimension" : {
          "type" : "object",
          "properties" : {
            "length" : {
              "type" : "number"
            },
            "width" : {
              "type" : "number"
            },
            "height" : {
              "type" : "number"
            },
            "offsetX" : {
              "type" : "number"
            },
            "offsetY" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    },
    "image" : {
      "type" : "object",
      "properties" : {
        "imageName" : {
          "type" : "string"
        },
        "sourceFile" : {
          "type" : "array",
          "items" : {
            "type" : "string"
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "mapCoordinate" : {
    "type" : "object",
    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  },
  "currentlyTracked" : {
    "type" : "boolean",
    "required" : true
  },
  "confidenceFactor" : {
    "type" : "number"
  },
  "statistics" : {
    "type" : "object",
    "properties" : {
      "currentServerTime" : {
        "type" : "string"
      },
      "firstLocatedTime" : {
        "type" : "string"
      },
      "lastLocatedTime" : {
        "type" : "string"
      }
    }
  },
  "historyLogReason" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "FLOOR_CHANGE", "CHOKEPOINT_ENCOUNTER", "INOUT_OF_AREA",
    "SIGNIFICANT_MOVEMENT", "LOCP_MESSAGE", "INFORMATION_CHANGE", "DEVICE_TYPE_CHANGE",
    "TAG_EMERGENCY", "LOCATION_CONTRIBUTOR_CHANGE", "INTERFERER_MERGE",
    "INTERFERER_CLUSTER_CENTRE_CHANGE", "INTERFERER_SEVERITY_CHANGE", "STATUS_CHANGE",
    "SECURITY_POLICY_CHANGE", "NETWORK_STATUS_CHANGE", "HASHKEY_CHANGE" ]
  },
  "geoCoordinate" : {
    "type" : "object",
    "properties" : {
      "latitude" : {
        "type" : "number"
      },
      "longitude" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "DEGREES" ]
      }
    }
  },
  "telemetryList" : {
    "type" : "array",
    "items" : {

```

```

    "type" : "object",
    "properties" : {
      "type" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "GPS", "TEMPERATURE", "MOTION", "DISTANCE", "FUEL",
"QUANTITY", "PRESSURE", "HUMIDITY", "STATUS", "EPC", "MOTIONPROB" ]
      },
      "units" : {
        "type" : "string"
      },
      "lastReceivedTime" : {
        "type" : "string"
      },
      "lastReceivedSeqNum" : {
        "type" : "integer"
      },
      "telemetryMeasurementData" : {
        "type" : "object",
        "properties" : {
          "bytes" : {
            "type" : "array",
            "items" : {
              "type" : "string"
            }
          },
          "lengthInBytes" : {
            "type" : "integer"
          }
        }
      }
    }
  },
  "batteryInfo" : {
    "type" : "object",
    "properties" : {
      "tolerance" : {
        "type" : "integer"
      },
      "percentRemaining" : {
        "type" : "integer"
      },
      "daysRemaining" : {
        "type" : "integer"
      },
      "batteryAge" : {
        "type" : "number"
      },
      "lastReceivedTime" : {
        "type" : "string"
      },
      "lastReceivedSeqNum" : {
        "type" : "integer"
      }
    }
  },
  "vendorData" : {
    "type" : "object",
    "properties" : {
      "vendorId" : {
        "type" : "integer"
      },
      "elementId" : {
        "type" : "integer"
      }
    }
  }
}

```

```

    },
    "data" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "lastReceivedTime" : {
      "type" : "string"
    },
    "lastReceivedSeqNum" : {
      "type" : "integer"
    }
  }
},
"sortedEmergencyList" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "reason" : {
        "type" : "string"
      },
      "tamperState" : {
        "type" : "string"
      },
      "tamperOpenedCount" : {
        "type" : "integer"
      },
      "lastReceivedTime" : {
        "type" : "string"
      },
      "lastReceivedSeqNum" : {
        "type" : "integer"
      },
      "fragmentNumber" : {
        "type" : "integer"
      },
      "active" : {
        "type" : "boolean",
        "required" : true
      }
    }
  }
},
"sortedChokepointList" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "macAddress" : {
        "type" : "string"
      },
      "timestamp" : {
        "type" : "string"
      },
      "inRange" : {
        "type" : "boolean",
        "required" : true
      },
      "isPerimeter" : {
        "type" : "boolean",
        "required" : true
      }
    }
  }
},

```

```

        "lastReceivedSeqNum" : {
          "type" : "integer"
        },
        "fragmentNumber" : {
          "type" : "integer"
        },
        "assocFloorId" : {
          "type" : "number"
        },
        "index" : {
          "type" : "integer"
        },
        "active" : {
          "type" : "boolean",
          "required" : true
        }
      }
    }
  },
  "lastBeaconTime" : {
    "type" : "string"
  },
  "lastBeaconSequenceNumber" : {
    "type" : "integer"
  }
}

```

tag.proto

```

option java_package = "com.cisco.mse.cas.location";
option java_outer_classname = "TagProtos";

import "dimension.proto";
import "image.proto";
import "map-coordinate.proto";
import "geo-coordinate.proto";
import "map-info.proto";
import "history-reason.proto";
import "statistics.proto";
import "tag-telemetry.proto";
import "battery-info.proto";
import "vendor-data.proto";
import "emergency-info.proto";
import "device-exciter.proto";

message PBTagLocation {

  required string macAddress = 1;

  optional PBMapInfo mapInfo = 2;

  optional PBMapCoordinate mapCoordinate = 3;

  optional bool currentlyTracked = 4;

  optional float confidenceFactor = 5;

  optional PBStatistics statistics = 6;

  optional PBHistoryReason historyLogReason = 7;

  optional PBGeoCoordinate geoCoordinate = 8;

  repeated PBTagTelemetryInfo telemetryData = 9;
}

```

```

optional PBBatteryInfo batteryInfo = 10;

optional PBVendorData vendorData = 11;

repeated PBEmergencyInfo sortedEmergencyInfo = 12;

repeated PBDeviceExciter sortedExciterEncounter = 13;

optional uint64 lastBeaconTimeInMillis = 14;

optional int32 lastBeaconSequenceNumber = 15;
}

```

BatteryInfo

BatteryInfo.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="BatteryInfo" type="batteryInfo"/>

  <xs:complexType name="batteryInfo">
    <xs:sequence/>
    <xs:attribute name="tolerance" type="xs:int"/>
    <xs:attribute name="percentRemaining" type="xs:int"/>
    <xs:attribute name="daysRemaining" type="xs:int"/>
    <xs:attribute name="batteryAge" type="xs:long"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int"/>
  </xs:complexType>
</xs:schema>

```

BatteryInfo.json

```

{
  "type" : "object",
  "properties" : {
    "tolerance" : {
      "type" : "integer"
    },
    "percentRemaining" : {
      "type" : "integer"
    },
    "daysRemaining" : {
      "type" : "integer"
    },
    "batteryAge" : {
      "type" : "number"
    },
    "lastReceivedTime" : {
      "type" : "string"
    },
    "lastReceivedSeqNum" : {
      "type" : "integer"
    }
  }
}

```


battery-info.proto

```

option java_package = "com.cisco.mse.cas.info";
option java_outer_classname = "BatteryInfoProtos";

message PBBatteryInfo {

    optional int32 tolerance = 1;

    optional int32 percentRemaining = 2;

    optional int32 daysRemaining = 3;

    optional int64 batteryAge = 4;

    optional uint64 lastReceivedTimeInMillis = 5;

    optional int32 lastReceivedSeqNum = 6;

}

```

EmergencyInfo

EmergencyInfo.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="EmergencyInfo" type="emergencyInfo"/>

    <xs:complexType name="emergencyInfo">
        <xs:sequence/>
        <xs:attribute name="isActive" type="xs:boolean" use="required"/>
        <xs:attribute name="reason" type="xs:string"/>
        <xs:attribute name="tamperState" type="xs:string"/>
        <xs:attribute name="tamperOpenedCount" type="xs:int" use="required"/>
        <xs:attribute name="lastReceivedTime" type="xs:string"/>
        <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
        <xs:attribute name="fragmentNumber" type="xs:byte" use="required"/>
    </xs:complexType>
</xs:schema>

```

EmergencyInfo.json

```

{
  "type" : "object",
  "properties" : {
    "reason" : {
      "type" : "string"
    },
    "tamperState" : {
      "type" : "string"
    },
    "tamperOpenedCount" : {
      "type" : "integer"
    },
    "lastReceivedTime" : {
      "type" : "string"
    },
  },
}

```

```

    "lastReceivedSeqNum" : {
      "type" : "integer"
    },
    "fragmentNumber" : {
      "type" : "integer"
    },
    "active" : {
      "type" : "boolean",
      "required" : true
    }
  }
}

```

emergency-info.proto

```

option java_package = "com.cisco.mse.cas.info.tag";
option java_outer_classname = "EmergencyInfoProtos";

```

```

message PBEmergencyInfo {

    required bool isActive = 1 [default = true];

    optional string reason = 2;

    optional string tamperState = 3;

    optional int32 tamperOpenedCount = 4;

    optional uint64 lastReceivedTimeInMillis = 5;

    optional int32 lastReceivedSeqNum = 6;

    optional int32 fragmentNumber = 7;

}

```

VendorData

VendorData.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="VendorData" type="vendorData"/>

  <xs:complexType name="vendorData">
    <xs:sequence/>
    <xs:attribute name="vendorId" type="xs:int" use="required"/>
    <xs:attribute name="elementId" type="xs:short" use="required"/>
    <xs:attribute name="data" type="xs:base64Binary"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
  </xs:complexType>
</xs:schema>

```

VendorData.json

```

{
  "type" : "object",
  "properties" : {
    "vendorId" : {

```

```

        "type" : "integer"
    },
    "elementId" : {
        "type" : "integer"
    },
    "data" : {
        "type" : "array",
        "items" : {
            "type" : "string"
        }
    },
    "lastReceivedTime" : {
        "type" : "string"
    },
    "lastReceivedSeqNum" : {
        "type" : "integer"
    }
}
}
}

```

vendor-data.proto

```

option java_package = "com.cisco.mse.cas.info";
option java_outer_classname = "VendorDataProtos";
message PBVendorData {

    optional int32 vendorId = 1;

    optional int32 elementId = 2;

    optional bytes data = 3;

    optional uint64 lastReceivedTimeInMillis = 4;

    optional int32 lastReceivedSeqNum = 5;

}

```

DeviceExciter

DeviceExciter.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="DeviceExciter" type="deviceExciter"/>

    <xs:complexType name="deviceExciter">
        <xs:sequence/>
        <xs:attribute name="macAddress" type="xs:string"/>
        <xs:attribute name="timestamp" type="xs:string"/>
        <xs:attribute name="inRange" type="xs:boolean" use="required"/>
        <xs:attribute name="isPerimeter" type="xs:boolean" use="required"/>
        <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
        <xs:attribute name="fragmentNumber" type="xs:byte" use="required"/>
        <xs:attribute name="isActive" type="xs:boolean" use="required"/>
        <xs:attribute name="assocFloorId" type="xs:long" use="required"/>
        <xs:attribute name="index" type="xs:short" use="required"/>
    </xs:complexType>
</xs:schema>

```

DeviceExciter.json

```

{
  "type" : "object",
  "properties" : {
    "macAddress" : {
      "type" : "string"
    },
    "timestamp" : {
      "type" : "string"
    },
    "inRange" : {
      "type" : "boolean",
      "required" : true
    },
    "isPerimeter" : {
      "type" : "boolean",
      "required" : true
    },
    "lastReceivedSeqNum" : {
      "type" : "integer"
    },
    "fragmentNumber" : {
      "type" : "integer"
    },
    "assocFloorId" : {
      "type" : "number"
    },
    "index" : {
      "type" : "integer"
    },
    "active" : {
      "type" : "boolean",
      "required" : true
    }
  }
}

```

device-exciter.proto

```

option java_package = "com.cisco.mse.cas.info.tag";
option java_outer_classname = "DeviceExciterProtos";

message PBDeviceExciter {

    required string macAddress = 1;

    optional uint64 timeStampInMillis = 2;

    optional bool inRange = 3;

    optional bool isPerimeter = 4;

    optional int32 lastReceivedSeqNum = 5;

    optional int32 fragmentNumber = 6;

    optional bool isActive = 7 [default = true];

    optional int64 assocFloorId = 8;

    optional int32 index = 9;

}

```

TelemetryInfo

TelemetryInfo.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="EPCTelemetryData" type="epcTelemetryData"/>

  <xs:element name="GeoTelemetryData" type="geoTelemetryData"/>

  <xs:element name="MotionTelemetryData" type="motionTelemetryData"/>

  <xs:element name="TelemetryData" type="defaultTelemetryMeasurementData"/>

  <xs:element name="VendorData" type="telemetryInfo"/>

  <xs:complexType name="telemetryInfo">
    <xs:choice>
      <xs:element ref="TelemetryData"/>
      <xs:element ref="EPCTelemetryData"/>
      <xs:element ref="GeoTelemetryData"/>
      <xs:element ref="MotionTelemetryData"/>
    </xs:choice>
    <xs:attribute name="type" type="telemetryType"/>
    <xs:attribute name="units" type="xs:string"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="telemetryMeasurementData" abstract="true">
    <xs:sequence>
      <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="defaultTelemetryMeasurementData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence>
          <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="measurementDataType" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="epcTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence/>
        <xs:attribute name="header" type="xs:string"/>
        <xs:attribute name="manufacturer" type="xs:string"/>
        <xs:attribute name="productSKU" type="xs:string"/>
        <xs:attribute name="serialNo" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="geoTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">

```

```

        <xs:sequence/>
        <xs:attribute name="latitude" type="xs:string"/>
        <xs:attribute name="longitude" type="xs:string"/>
        <xs:attribute name="altitude" type="xs:string"/>
        <xs:attribute name="altitudeType" type="altitudeType"/>
        <xs:attribute name="datum" type="xs:int" use="required"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="motionTelemetryData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence/>
            <xs:attribute name="motionType" type="motionType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="telemetryType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN"/>
        <xs:enumeration value="GPS"/>
        <xs:enumeration value="TEMPERATURE"/>
        <xs:enumeration value="MOTION"/>
        <xs:enumeration value="DISTANCE"/>
        <xs:enumeration value="FUEL"/>
        <xs:enumeration value="QUANTITY"/>
        <xs:enumeration value="PRESSURE"/>
        <xs:enumeration value="HUMIDITY"/>
        <xs:enumeration value="STATUS"/>
        <xs:enumeration value="EPC"/>
        <xs:enumeration value="MOTIONPROB"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="altitudeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN"/>
        <xs:enumeration value="AT_METERS"/>
        <xs:enumeration value="AT_FLOOR"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="motionType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN"/>
        <xs:enumeration value="NO_MOVEMENT"/>
        <xs:enumeration value="ACCELERATION"/>
        <xs:enumeration value="MOVEMENT"/>
        <xs:enumeration value="DECELERATION"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

TelemetryInfo.json

```

{
  "type" : "object",
  "properties" : {
    "type" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "GPS", "TEMPERATURE", "MOTION", "DISTANCE", "FUEL",
"QUANTITY", "PRESSURE", "HUMIDITY", "STATUS", "EPC", "MOTIONPROB" ]
    }
  }
}

```

```

    },
    "units" : {
      "type" : "string"
    },
    "lastReceivedTime" : {
      "type" : "string"
    },
    "lastReceivedSeqNum" : {
      "type" : "integer"
    },
    "telemetryMeasurementData" : {
      "type" : "object",
      "properties" : {
        "bytes" : {
          "type" : "array",
          "items" : {
            "type" : "string"
          }
        },
        "lengthInBytes" : {
          "type" : "integer"
        }
      }
    }
  }
}

```

tag-telemetry.proto

```

option java_package = "com.cisco.mse.cas.info.tag";
option java_outer_classname = "TagTelemetryInfoProtos";

message PBTAGTelemetryInfo {

  enum PBTelemetryType {
    GPS = 0;
    TEMPERATURE = 1;
    MOTION = 2;
    DISTANCE = 3;
    FUEL = 4;
    QUANTITY = 5;
    PRESSURE = 6;
    HUMIDITY = 7;
    STATUS = 8;
    EPC = 9;
    MOTIONPROB = 10;
  }

  message PBDefaultTelemetryMeasurementData {
    optional string measurementData = 1;

    optional string measurementDataType = 2;
  }

  message PBEPCTelemetryData {
    optional string header = 1;

    optional string manufacturer = 2;

    optional string productSKU = 3;

    optional string serialNo = 4;
  }
}

```

```

message PBGeoTelemetryData {
    enum PBAltitudeType {
        AT_METERS = 1;
        AT_FLOOR = 2;
    }

    required string latitude = 1;
    required string longitude = 2;

    optional string altitude = 3;
    optional PBAltitudeType altitudeType = 4 [default = AT_METERS];
    optional int32 datum = 5;
}

message PBMotionTelemetryData {
    enum PBMotionType {
        NO_MOVEMENT = 1;
        ACCELERATION = 2;
        MOVEMENT = 3;
        DECELERATION = 4;
    }

    required PBMotionType motionType = 1;
}

required PBTelemetryType telemetryType = 1;
optional string units = 2;
optional uint64 lastReceivedTimeInMillis = 3;
optional int32 lastReceivedSeqNum = 4;
optional bytes dataBlob = 5;
optional PBDefaultTelemetryMeasurementData telemetryData = 6;
optional PBEPCTelemetryData epcTelemetryData = 7;
optional PBGeoTelemetryData geoTelemetryData = 8;
optional PBMotionTelemetryData motionTelemetryData = 9;
}

```

DefaultTelemetryMeasurementData

DefaultTelemetryMeasurementData.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="EPCTelemetryData" type="epcTelemetryData"/>

  <xs:element name="GeoTelemetryData" type="geoTelemetryData"/>

  <xs:element name="MotionTelemetryData" type="motionTelemetryData"/>

```



```

<xs:element name="TelemetryData" type="defaultTelemetryMeasurementData" />

<xs:complexType name="defaultTelemetryMeasurementData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence>
        <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="measurementDataType" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="telemetryMeasurementData" abstract="true">
  <xs:sequence>
    <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="epcTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence>
        <xs:attribute name="header" type="xs:string"/>
        <xs:attribute name="manufacturer" type="xs:string"/>
        <xs:attribute name="productSKU" type="xs:string"/>
        <xs:attribute name="serialNo" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="geoTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence>
        <xs:attribute name="latitude" type="xs:string"/>
        <xs:attribute name="longitude" type="xs:string"/>
        <xs:attribute name="altitude" type="xs:string"/>
        <xs:attribute name="altitudeType" type="altitudeType"/>
        <xs:attribute name="datum" type="xs:int" use="required"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="motionTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence>
        <xs:attribute name="motionType" type="motionType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="altitudeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="AT_METERS"/>
    <xs:enumeration value="AT_FLOOR"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="motionType">
  <xs:restriction base="xs:string">

```

```

    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="NO_MOVEMENT"/>
    <xs:enumeration value="ACCELERATION"/>
    <xs:enumeration value="MOVEMENT"/>
    <xs:enumeration value="DECELERATION"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

DefaultTelemetryMeasurementData.json

```

{
  "type" : "object",
  "properties" : {
    "measurementData" : {
      "type" : "string"
    },
    "measurementDataType" : {
      "type" : "string"
    },
    "bytes" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "lengthInBytes" : {
      "type" : "integer"
    }
  }
}

```

EPCTelemetryData

EPCTelemetryData.xml

```

{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="EPCTelemetryData" type="epcTelemetryData"/>

    <xs:element name="GeoTelemetryData" type="geoTelemetryData"/>

    <xs:element name="MotionTelemetryData" type="motionTelemetryData"/>

    <xs:element name="TelemetryData" type="defaultTelemetryMeasurementData"/>

    <xs:complexType name="epcTelemetryData">
      <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
          <xs:sequence>
            <xs:attribute name="header" type="xs:string"/>
            <xs:attribute name="manufacturer" type="xs:string"/>
            <xs:attribute name="productSKU" type="xs:string"/>
            <xs:attribute name="serialNo" type="xs:string"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>

    <xs:complexType name="telemetryMeasurementData" abstract="true">

```

```

    <xs:sequence>
      <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="defaultTelemetryMeasurementData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence>
          <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="measurementDataType" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="geoTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence/>
        <xs:attribute name="latitude" type="xs:string"/>
        <xs:attribute name="longitude" type="xs:string"/>
        <xs:attribute name="altitude" type="xs:string"/>
        <xs:attribute name="altitudeType" type="altitudeType"/>
        <xs:attribute name="datum" type="xs:int" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motionTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence/>
        <xs:attribute name="motionType" type="motionType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="altitudeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="AT_METERS"/>
      <xs:enumeration value="AT_FLOOR"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="motionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="NO_MOVEMENT"/>
      <xs:enumeration value="ACCELERATION"/>
      <xs:enumeration value="MOVEMENT"/>
      <xs:enumeration value="DECELERATION"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

EPCTelemetryData.json

```

{
  "type" : "object",
  "properties" : {
    "header" : {

```

```

        "type" : "string"
    },
    "manufacturer" : {
        "type" : "string"
    },
    "productSKU" : {
        "type" : "string"
    },
    "serialNo" : {
        "type" : "string"
    },
    "bytes" : {
        "type" : "array",
        "items" : {
            "type" : "string"
        }
    },
    "lengthInBytes" : {
        "type" : "integer"
    }
}
}
}

```

GeoTelemetryData

GeoTelemetryData.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="EPCTelemetryData" type="epcTelemetryData"/>

    <xs:element name="GeoTelemetryData" type="geoTelemetryData"/>

    <xs:element name="MotionTelemetryData" type="motionTelemetryData"/>

    <xs:element name="TelemetryData" type="defaultTelemetryMeasurementData"/>

    <xs:complexType name="geoTelemetryData">
        <xs:complexContent>
            <xs:extension base="telemetryMeasurementData">
                <xs:sequence/>
                <xs:attribute name="latitude" type="xs:string"/>
                <xs:attribute name="longitude" type="xs:string"/>
                <xs:attribute name="altitude" type="xs:string"/>
                <xs:attribute name="altitudeType" type="altitudeType"/>
                <xs:attribute name="datum" type="xs:int" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="telemetryMeasurementData" abstract="true">
        <xs:sequence>
            <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="defaultTelemetryMeasurementData">
        <xs:complexContent>
            <xs:extension base="telemetryMeasurementData">

```

```

        <xs:sequence>
          <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="measurementDataType" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="epcTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence>
          <xs:attribute name="header" type="xs:string"/>
          <xs:attribute name="manufacturer" type="xs:string"/>
          <xs:attribute name="productSKU" type="xs:string"/>
          <xs:attribute name="serialNo" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motionTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence>
          <xs:attribute name="motionType" type="motionType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="altitudeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="AT_METERS"/>
      <xs:enumeration value="AT_FLOOR"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="motionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="NO_MOVEMENT"/>
      <xs:enumeration value="ACCELERATION"/>
      <xs:enumeration value="MOVEMENT"/>
      <xs:enumeration value="DECELERATION"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

GeoTelemetryData.json

```

{
  "type" : "object",
  "properties" : {
    "latitude" : {
      "type" : "string"
    },
    "longitude" : {
      "type" : "string"
    },
    "altitude" : {
      "type" : "string"
    },
    "altitudeType" : {
      "type" : "string",

```

```

    "enum" : [ "UNKNOWN", "AT_METERS", "AT_FLOOR" ]
  },
  "datum" : {
    "type" : "integer"
  },
  "bytes" : {
    "type" : "array",
    "items" : {
      "type" : "string"
    }
  },
  "lengthInBytes" : {
    "type" : "integer"
  }
}
}
}

```

MotionTelemetryData

MotionTelemetryData.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="EPCTelemetryData" type="epcTelemetryData"/>

  <xs:element name="GeoTelemetryData" type="geoTelemetryData"/>

  <xs:element name="MotionTelemetryData" type="motionTelemetryData"/>

  <xs:element name="TelemetryData" type="defaultTelemetryMeasurementData"/>

  <xs:complexType name="motionTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence/>
        <xs:attribute name="motionType" type="motionType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="telemetryMeasurementData" abstract="true">
    <xs:sequence>
      <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="defaultTelemetryMeasurementData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence>
          <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="measurementDataType" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="epcTelemetryData">
    <xs:complexContent>

```

```

    <xs:extension base="telemetryMeasurementData">
      <xs:sequence/>
      <xs:attribute name="header" type="xs:string"/>
      <xs:attribute name="manufacturer" type="xs:string"/>
      <xs:attribute name="productSKU" type="xs:string"/>
      <xs:attribute name="serialNo" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="geoTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence/>
      <xs:attribute name="latitude" type="xs:string"/>
      <xs:attribute name="longitude" type="xs:string"/>
      <xs:attribute name="altitude" type="xs:string"/>
      <xs:attribute name="altitudeType" type="altitudeType"/>
      <xs:attribute name="datum" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="motionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="NO_MOVEMENT"/>
    <xs:enumeration value="ACCELERATION"/>
    <xs:enumeration value="MOVEMENT"/>
    <xs:enumeration value="DECELERATION"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="altitudeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="AT_METERS"/>
    <xs:enumeration value="AT_FLOOR"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

MotionTelemetryData.json

```

{
  "type" : "object",
  "properties" : {
    "motionType" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "NO_MOVEMENT", "ACCELERATION", "MOVEMENT", "DECELERATION" ]
    },
    "bytes" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "lengthInBytes" : {
      "type" : "integer"
    }
  }
}

```

API Related Object Schema Details for Notification API Object Schema Details

The below table lists the Notification API object schema details

Table 5-3 Notification API Related objects

Object Name	Description	Schema Format
NotificationSubscription	Top level object used to create a notification subscription. It allows to specify the name, notification type, data format and the receiver info.	NotificationSubscription.xml NotificationSubscription.json
NotificationReceiverInfo	This is top level object for specifying the transport parameters of the notification receiver.	NotificationReceiverInfo.xml NotificationReceiverInfo.json
TransportHttp	Used to specify details of a HTTP or HTTPS notification receiver.	TransportHTTP.xml TransportHTTP.json
TransportTcp	Used to specify details of a TCP based notification receiver.	TransportTCP.xml TransportTCP.json
PresenceEventTrigger	Used to specify trigger condition for a presence notification event. Presence indicates when a device is detected by the MSE either first time or after it is seen by the network again. The notification type for this trigger is EVENT_DRIVEN.	PresenceEventTrigger.xml PresenceEventTrigger.json
AbsenceEventTrigger	Used to specify trigger condition for a Absence notification Event. The absence is determined by the configured absence period. The notification type for this trigger is EVENT_DRIVEN.	AbsenceEventTrigger.xml AbsenceEventTrigger.json
MovementEventTrigger	Used to specify trigger condition for a movement notification event. Movement condition can be specified as relative distance or relative to a fixed marker on a floor Map. The notification type for this trigger is EVENT_DRIVEN.	MovementEventTrigger.xml MovementEventTrigger.xml
ContainmentEventTrigger	Used to specify trigger condition for a containment notification event. Containment can be for an in zone event or an out of zone event. The zone is defined as a hierarchy of type "Campus Name > Building Name > Floor Name > Zone Name" string. The notification type for this trigger is EVENT_DRIVEN. The attribute zoneTimeout controls the number of containment events. If a device continues to be in a zone, then there will not be repeating notifications sent out until "zoneTimeout" in minutes passes. If this is not set, notifications will be sent out multiple times even though the device is in the zone.	ContainmentEventTrigger.xml ContainmentEventTrigger.json
BatteryEventTrigger	Used to specify trigger condition for a battery notification event for a RFID tag. The notification type for this trigger is EVENT_DRIVEN.	BatteryEventTrigger.xml BatteryEventTrigger.json

Table 5-3 Notification API Related objects (continued)

Object Name	Description	Schema Format
EmergencyEventTrigger	Used to specify trigger condition for a emergency notification event for a RFID tag. The notification type for this trigger is EVENT_DRIVEN.	EmergencyEventTrigger.xml EmergencyEventTrigger.json
ExciterEventTrigger	Used to specify trigger condition for a Exciter notification event for a RFID tag. The condition contains a list of exciter mac addresses. The notification type for this trigger is EVENT_DRIVEN.	ExciterEventTrigger.xml ExciterEventTrigger.json
StreamingNotificationTrigger	Used to specify trigger condition for a Streaming Notification Event. Currently, RSSI Measurements are streamed for Tag only. The notification type for this trigger is STREAMING.	StreamingNotificationTrigger.xml StreamingNotificationTrigger.json
MapInfoChangeEventTrigger	Used to specify trigger condition for a map info change notification event. The notification type for this trigger is STREAMING.	MapInfoChangeEventTrigger.xml MapInfoChangeEventTrigger.json
AssociationEventTrigger	Used to specify trigger condition for an Association Event. The association is determined by a wireless client connecting to the Wi-Fi network.	AssociationEventTrigger.xml AssociationEventTrigger.json
AbsenceEvent	Event triggered in response to a absence event trigger subscription. Contains the associated subscription, last seen, absence duration, device Id, last known location and map hierarchy.	AbsenceEvent.xml AbsenceEvent.json AbsenceEvent.proto
PresenceEvent	Event triggered in response to a presence event trigger subscription. Contains the associated subscription, device Id, timestamp, map hierarchy and current location.	PresenceEvent.xml PresenceEvent.json presence-event.proto
MovementEvent	Event triggered in response to a movement event trigger subscription. Contains the associated subscription, move distance, reference marker if any, device Id, timestamp, map hierarchy and current location.	MovementEvent.xml MovementEvent.json movement-event.proto
ContainmentEvent	Event triggered in response to a containment event trigger subscription. Contains the associated subscription, containment boundary, contained hierarchy, device Id, timestamp, map hierarchy and current location.	ContainmentEvent.xml ContainmentEvent.json ContainmentEvent.json
MapInfoChangeEvent	Event triggered in response to a map info change trigger subscription. Contains the associated subscription, change type and the changed campus details.	MapInfoChangeEvent.xml MapInfoChangeEvent.json mapchange-event.proto
BatteryEvent	Event triggered in response to a BatteryEventTrigger Subscription. Contains the associated subscription, battery status, device Id, timestamp, map hierarchy and current location.	BatteryEvent.xml BatteryEvent.json battery-event.proto

Table 5-3 Notification API Related objects (continued)

Object Name	Description	Schema Format
EmergencyEvent	Event triggered in response to a EmergencyEventTrigger Subscription. Contains the associated subscription, emergency type, vendor id, vendor data, device Id, timestamp, map hierarchy and current location.	EmergencyEvent.xml EmergencyEvent.json EmergencyEvent.proto
ExciterEvent	Event triggered in response to a exciter event trigger subscription. Contains the associated subscription, trigger exciter MAC address and name, device Id, timestamp, map hierarchy and current location.	ExciterEvent.xml ExciterEvent.json ExciterEvent.proto
StreamingNotification	Event triggered in response to a streaming notification trigger subscription. Contains the associated subscription and the current location information. This event is triggered for every location calculated for the specified device entity.	StreamingNotification.xml StreamingNotification.json StreamingNotification.proto
TagRssiNotification	Event triggered in response to a streaming notification trigger subscription for RAW RSSI for RFID Tags. This event is triggered for every RSSI received for the subscribed device entity. Only supported in protobuf data format.	tag-rssi-protobuf ap-rssi-protobuf
AssociationEvent	Event triggered in response to a AssociationEventTrigger Subscription for association of wireless clients to the Wi-Fi network. This event is triggered for every Wi-Fi association of the subscribed device entity.	AssociationEvent.xml AssociationEvent.json AssociationEvent.proto

Schema Description for Notifications API Related Objects

NotificationSubscription

NotificationSubscription.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

  <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

  <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

  <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

  <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

```

```
<xs:element name="NotificationReceiverInfo" type="notificationReceiverInfo" />
<xs:element name="NotificationSubscription" type="notificationSubscription" />
<xs:element name="PresenceEventTrigger" type="presenceEventTrigger" />
<xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger" />
<xs:element name="TransportHttp" type="transportHttp" />
<xs:element name="TransportTcp" type="transportTcp" />
<xs:element name="transport" type="transport" />
<xs:complexType name="notificationSubscription">
  <xs:sequence>
    <xs:element ref="NotificationReceiverInfo" minOccurs="0" />
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="NotificationEventTrigger" />
      <xs:element ref="PresenceEventTrigger" />
      <xs:element ref="AbsenceEventTrigger" />
      <xs:element ref="MovementEventTrigger" />
      <xs:element ref="ContainmentEventTrigger" />
      <xs:element ref="BatteryEventTrigger" />
      <xs:element ref="EmergencyEventTrigger" />
      <xs:element ref="ExciterEventTrigger" />
      <xs:element ref="StreamingNotificationTrigger" />
      <xs:element ref="MapInfoChangeEventTrigger" />
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" />
  <xs:attribute name="notificationType" type="notificationType" />
  <xs:attribute name="dataFormat" type="dataFormat" />
  <xs:attribute name="lastModifiedTime" type="xs:string" />
</xs:complexType>
<xs:complexType name="notificationReceiverInfo">
  <xs:choice>
    <xs:element ref="transport" />
    <xs:element ref="TransportTcp" />
    <xs:element ref="TransportHttp" />
  </xs:choice>
</xs:complexType>
<xs:complexType name="transport" abstract="true">
  <xs:sequence/>
  <xs:attribute name="hostAddress" type="xs:string" />
  <xs:attribute name="port" type="xs:int" use="required" />
</xs:complexType>
<xs:complexType name="transportTcp">
  <xs:complexContent>
    <xs:extension base="transport">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="transportHttp">
  <xs:complexContent>
    <xs:extension base="transport">
      <xs:sequence/>
      <xs:attribute name="urlPath" type="xs:string" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

        <xs:attribute name="https" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="notificationEventTrigger" abstract="true">
    <xs:sequence>
      <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="eventEntity" type="eventEntity"/>
  </xs:complexType>

  <xs:complexType name="presenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="absenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="movementEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
        <xs:attribute name="referenceMarkerName" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="containmentEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="boundary" type="containmentBoundary"/>
        <xs:attribute name="zoneHierarchy" type="xs:string"/>
        <xs:attribute name="zoneTimeout" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="batteryEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="batteryStatus" type="batteryStatus"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="emergencyEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">

```

```

        <xs:sequence/>
        <xs:attribute name="type" type="type"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence>
                <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="streamType" type="streamType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="notificationType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="EVENT_DRIVEN" />
        <xs:enumeration value="STREAMING" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="dataFormat">
    <xs:restriction base="xs:string">
        <xs:enumeration value="OLD_XML" />
        <xs:enumeration value="PLAIN_TEXT" />
        <xs:enumeration value="XML" />
        <xs:enumeration value="JSON" />
        <xs:enumeration value="PROTOCOL_BUFFERS" />
        <xs:enumeration value="FILE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WIRELESS_CLIENTS" />
        <xs:enumeration value="TAGS" />
        <xs:enumeration value="ROGUE_CLIENTS" />
        <xs:enumeration value="ROGUE_APS" />
        <xs:enumeration value="INTERFERERS" />
        <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
        <xs:enumeration value="MAP_CHANGE" />
        <xs:enumeration value="MAC_ADDRESS" />
        <xs:enumeration value="ASSET_NAME" />
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="ASSET_CATEGORY"/>
        <xs:enumeration value="ASSET_GROUP"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE"/>
        <xs:enumeration value="OUTSIDE"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY"/>
        <xs:enumeration value="LOW"/>
        <xs:enumeration value="MEDIUM"/>
        <xs:enumeration value="NORMAL"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="NONE"/>
        <xs:enumeration value="ANY"/>
        <xs:enumeration value="UNKNOWN"/>
        <xs:enumeration value="PANIC_BUTTON"/>
        <xs:enumeration value="DETACHED"/>
        <xs:enumeration value="TAMPERING"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RSSI_MEASUREMENTS"/>
        <xs:enumeration value="RAW_LOCATION"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

NotificationSubscription.json

```

{
  "type" : "object",
  "properties" : {
    "name" : {
      "type" : "string"
    },
    "notificationType" : {
      "type" : "string",
      "enum" : [ "UNKNOWN", "EVENT_DRIVEN", "STREAMING" ]
    },
    "dataFormat" : {
      "type" : "string",
      "enum" : [ "OLD_XML", "PLAIN_TEXT", "XML", "JSON", "PROTOCOL_BUFFERS", "FILE" ]
    },
    "lastModifiedTime" : {
      "type" : "string"
    },
    "receiverInfo" : {
      "type" : "object",
      "properties" : {
        "transport" : {
          "type" : "object",

```

```

        "properties" : {
            "hostAddress" : {
                "type" : "string"
            },
            "port" : {
                "type" : "integer"
            }
        }
    },
    "subscribedEvents" : {
        "type" : "array",
        "items" : {
            "type" : "any"
        }
    }
}
}

```

NotificationReceiverInfo

NotificationReceiverInfo.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="NotificationReceiverInfo" type="notificationReceiverInfo"/>

    <xs:element name="TransportHttp" type="transportHttp"/>

    <xs:element name="TransportTcp" type="transportTcp"/>

    <xs:element name="transport" type="transport"/>

    <xs:complexType name="notificationReceiverInfo">
        <xs:choice>
            <xs:element ref="transport"/>
            <xs:element ref="TransportTcp"/>
            <xs:element ref="TransportHttp"/>
        </xs:choice>
    </xs:complexType>

    <xs:complexType name="transport" abstract="true">
        <xs:sequence/>
        <xs:attribute name="hostAddress" type="xs:string"/>
        <xs:attribute name="port" type="xs:int" use="required"/>
    </xs:complexType>

    <xs:complexType name="transportTcp">
        <xs:complexContent>
            <xs:extension base="transport">
                <xs:sequence/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="transportHttp">
        <xs:complexContent>
            <xs:extension base="transport">

```

```

    <xs:sequence/>
    <xs:attribute name="urlPath" type="xs:string"/>
    <xs:attribute name="https" type="xs:boolean" use="required"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>

```

NotificationReceiverInfo.json

```

{
  "type" : "object",
  "properties" : {
    "transport" : {
      "type" : "object",
      "properties" : {
        "hostAddress" : {
          "type" : "string"
        },
        "port" : {
          "type" : "integer"
        }
      }
    }
  }
}

```

TransportHTTP

TransportHTTP.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="TransportHttp" type="transportHttp"/>

  <xs:element name="TransportTcp" type="transportTcp"/>

  <xs:element name="transport" type="transport"/>

  <xs:complexType name="transportHttp">
    <xs:complexContent>
      <xs:extension base="transport">
        <xs:sequence/>
        <xs:attribute name="urlPath" type="xs:string"/>
        <xs:attribute name="https" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="transport" abstract="true">
    <xs:sequence/>
    <xs:attribute name="hostAddress" type="xs:string"/>
    <xs:attribute name="port" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="transportTcp">
    <xs:complexContent>
      <xs:extension base="transport">

```



```

        <xs:sequence/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>

```

TransportHTTP.json

```

{
  "type" : "object",
  "properties" : {
    "hostAddress" : {
      "type" : "string"
    },
    "port" : {
      "type" : "integer"
    },
    "urlPath" : {
      "type" : "string"
    },
    "https" : {
      "type" : "boolean",
      "required" : true
    }
  }
}

```

TransportTCP

TransportTCP.xml

```

{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="TransportHttp" type="transportHttp"/>

    <xs:element name="TransportTcp" type="transportTcp"/>

    <xs:element name="transport" type="transport"/>

    <xs:complexType name="transportTcp">
      <xs:complexContent>
        <xs:extension base="transport">
          <xs:sequence/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="transport" abstract="true">
      <xs:sequence/>
      <xs:attribute name="hostAddress" type="xs:string"/>
      <xs:attribute name="port" type="xs:int" use="required"/>
    </xs:complexType>

    <xs:complexType name="transportHttp">
      <xs:complexContent>
        <xs:extension base="transport">
          <xs:sequence/>
          <xs:attribute name="urlPath" type="xs:string"/>
          <xs:attribute name="https" type="xs:boolean" use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:schema>

```

```

    </xs:complexContent>
  </xs:complexType>
</xs:schema>

```

TransportTCP.json

```

{
  "type" : "object",
  "properties" : {
    "hostAddress" : {
      "type" : "string"
    },
    "port" : {
      "type" : "integer"
    }
  }
}

```

PresenceEventTrigger

PresenceEventTrigger.xml

```

{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

    <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

    <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

    <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

    <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

    <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

    <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

    <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

    <xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

    <xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

    <xs:complexType name="presenceEventTrigger">
      <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
          <xs:sequence>
            <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>

    <xs:complexType name="notificationEventTrigger" abstract="true">
      <xs:sequence>
        <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>

```

```

    <xs:attribute name="eventEntity" type="eventEntity" />
  </xs:complexType>

  <xs:complexType name="absenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="movementEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="moveDistanceInFt" type="xs:float" use="required" />
        <xs:attribute name="referenceMarkerName" type="xs:string" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="containmentEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="boundary" type="containmentBoundary" />
        <xs:attribute name="zoneHierarchy" type="xs:string" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="batteryEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="batteryStatus" type="batteryStatus" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="emergencyEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="type" type="type" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence>
          <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">

```

```

        <xs:sequence/>
        <xs:attribute name="streamType" type="streamType" />
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WIRELESS_CLIENTS" />
        <xs:enumeration value="TAGS" />
        <xs:enumeration value="ROGUE_CLIENTS" />
        <xs:enumeration value="ROGUE_APS" />
        <xs:enumeration value="INTERFERERS" />
        <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
        <xs:enumeration value="MAP_CHANGE" />
        <xs:enumeration value="MAC_ADDRESS" />
        <xs:enumeration value="ASSET_NAME" />
        <xs:enumeration value="ASSET_CATEGORY" />
        <xs:enumeration value="ASSET_GROUP" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE" />
        <xs:enumeration value="OUTSIDE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY" />
        <xs:enumeration value="LOW" />
        <xs:enumeration value="MEDIUM" />
        <xs:enumeration value="NORMAL" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="NONE" />
        <xs:enumeration value="ANY" />
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="PANIC_BUTTON" />
        <xs:enumeration value="DETACHED" />
        <xs:enumeration value="TAMPERING" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RSSI_MEASUREMENTS" />
        <xs:enumeration value="RAW_LOCATION" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

PresenceEventTrigger.json

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger" />

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger" />

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger" />

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger" />

  <xs:element name="ExciterEventTrigger" type="exciterEventTrigger" />

  <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger" />

  <xs:element name="MovementEventTrigger" type="movementEventTrigger" />

  <xs:element name="NotificationEventTrigger" type="notificationEventTrigger" />

  <xs:element name="PresenceEventTrigger" type="presenceEventTrigger" />

  <xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger" />

  <xs:complexType name="presenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence>
          <xs:attribute name="existingPresence" type="xs:boolean" use="required" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="notificationEventTrigger" abstract="true">
    <xs:sequence>
      <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="eventEntity" type="eventEntity" />
  </xs:complexType>

  <xs:complexType name="absenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence>
          <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="movementEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence>
          <xs:attribute name="moveDistanceInFt" type="xs:float" use="required" />
          <xs:attribute name="referenceMarkerName" type="xs:string" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="containmentEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">

```

```

        <xs:sequence/>
        <xs:attribute name="boundary" type="containmentBoundary"/>
        <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="batteryStatus" type="batteryStatus"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="type" type="type"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence>
                <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="streamType" type="streamType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WIRELESS_CLIENTS"/>
        <xs:enumeration value="TAGS"/>
        <xs:enumeration value="ROGUE_CLIENTS"/>
        <xs:enumeration value="ROGUE_APS"/>
        <xs:enumeration value="INTERFERERS"/>
        <xs:enumeration value="INTERFERER_DEVICE_TYPE"/>
        <xs:enumeration value="MAP_CHANGE"/>
        <xs:enumeration value="MAC_ADDRESS"/>
        <xs:enumeration value="ASSET_NAME"/>
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="ASSET_CATEGORY" />
        <xs:enumeration value="ASSET_GROUP" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE" />
        <xs:enumeration value="OUTSIDE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY" />
        <xs:enumeration value="LOW" />
        <xs:enumeration value="MEDIUM" />
        <xs:enumeration value="NORMAL" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="NONE" />
        <xs:enumeration value="ANY" />
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="PANIC_BUTTON" />
        <xs:enumeration value="DETACHED" />
        <xs:enumeration value="TAMPERING" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RSSI_MEASUREMENTS" />
        <xs:enumeration value="RAW_LOCATION" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

AbsenceEventTrigger

AbsenceEventTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger" />

    <xs:element name="BatteryEventTrigger" type="batteryEventTrigger" />

    <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger" />

    <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger" />

    <xs:element name="ExciterEventTrigger" type="exciterEventTrigger" />

    <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger" />

    <xs:element name="MovementEventTrigger" type="movementEventTrigger" />

```

```

<xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

<xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

<xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

<xs:complexType name="absenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="notificationEventTrigger" abstract="true">
  <xs:sequence>
    <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="eventEntity" type="eventEntity"/>
</xs:complexType>

<xs:complexType name="presenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="movementEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
      <xs:attribute name="referenceMarkerName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
  <xs:complexContent>

```



```

        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="type" type="type"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence>
                <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="streamType" type="streamType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WIRELESS_CLIENTS"/>
        <xs:enumeration value="TAGS"/>
        <xs:enumeration value="ROGUE_CLIENTS"/>
        <xs:enumeration value="ROGUE_APS"/>
        <xs:enumeration value="INTERFERERS"/>
        <xs:enumeration value="INTERFERER_DEVICE_TYPE"/>
        <xs:enumeration value="MAP_CHANGE"/>
        <xs:enumeration value="MAC_ADDRESS"/>
        <xs:enumeration value="ASSET_NAME"/>
        <xs:enumeration value="ASSET_CATEGORY"/>
        <xs:enumeration value="ASSET_GROUP"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE"/>
        <xs:enumeration value="OUTSIDE"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY"/>
        <xs:enumeration value="LOW"/>
        <xs:enumeration value="MEDIUM"/>
        <xs:enumeration value="NORMAL"/>
    </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="ANY"/>
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="PANIC_BUTTON"/>
      <xs:enumeration value="DETACHED"/>
      <xs:enumeration value="TAMPERING"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="RSSI_MEASUREMENTS"/>
      <xs:enumeration value="RAW_LOCATION"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

AbsenceEventTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS",
"INTERFERERS", "INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME",
"ASSET_CATEGORY", "ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "mseUdi" : {
      "type" : "string"
    },
    "geoCoordinate" : {
      "type" : "object",
      "properties" : {
        "latitude" : {

```

```

        "type" : "number"
    },
    "longitude" : {
        "type" : "number"
    },
    "unit" : {
        "type" : "string",
        "enum" : [ "DEGREES" ]
    }
}
},
"floorRefId" : {
    "type" : "number"
},
"absenceDurationInMinutes" : {
    "type" : "integer"
},
"lastSeen" : {
    "type" : "string"
},
"timestamp" : {
    "type" : "string"
}
}
}
}

```

MovementEventTrigger

MovementEventTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

    <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

    <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

    <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

    <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

    <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

    <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

    <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

    <xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

    <xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

    <xs:complexType name="movementEventTrigger">
        <xs:complexContent>
            <xs:extension base="notificationEventTrigger">
                <xs:sequence/>
                <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
                <xs:attribute name="referenceMarkerName" type="xs:string"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

```

```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="notificationEventTrigger" abstract="true">
  <xs:sequence>
    <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="eventEntity" type="eventEntity"/>
</xs:complexType>

<xs:complexType name="presenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="absenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="type" type="type"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>

```

```
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="streamType" type="streamType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
      <xs:enumeration value="WIRELESS_CLIENTS"/>
      <xs:enumeration value="TAGS"/>
      <xs:enumeration value="ROGUE_CLIENTS"/>
      <xs:enumeration value="ROGUE_APS"/>
      <xs:enumeration value="INTERFERERS"/>
      <xs:enumeration value="INTERFERER_DEVICE_TYPE"/>
      <xs:enumeration value="MAP_CHANGE"/>
      <xs:enumeration value="MAC_ADDRESS"/>
      <xs:enumeration value="ASSET_NAME"/>
      <xs:enumeration value="ASSET_CATEGORY"/>
      <xs:enumeration value="ASSET_GROUP"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
      <xs:enumeration value="INSIDE"/>
      <xs:enumeration value="OUTSIDE"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ANY"/>
      <xs:enumeration value="LOW"/>
      <xs:enumeration value="MEDIUM"/>
      <xs:enumeration value="NORMAL"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="ANY"/>
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="PANIC_BUTTON"/>
      <xs:enumeration value="DETACHED"/>
      <xs:enumeration value="TAMPERING"/>
    </xs:restriction>
  </xs:simpleType>
```

```

<xs:simpleType name="streamType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="RSSI_MEASUREMENTS"/>
    <xs:enumeration value="RAW_LOCATION"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

MovementEventTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "moveDistanceInFt" : {
      "type" : "number"
    },
    "referenceMarkerName" : {
      "type" : "string"
    }
  }
}

```

ContainmentEventTrigger

ContainmentEventTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

  <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

  <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

  <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

  <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

  <xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

```

```

<xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="notificationEventTrigger" abstract="true">
  <xs:sequence>
    <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="eventEntity" type="eventEntity"/>
</xs:complexType>

<xs:complexType name="presenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="absenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="movementEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
      <xs:attribute name="referenceMarkerName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="type" type="type"/>
    </xs:extension>
  </xs:complexContent>

```

```

</xs:complexType>

<xs:complexType name="exciterEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:attribute name="streamType" type="streamType" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="containmentBoundary">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INSIDE" />
    <xs:enumeration value="OUTSIDE" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="eventEntity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WIRELESS_CLIENTS" />
    <xs:enumeration value="TAGS" />
    <xs:enumeration value="ROGUE_CLIENTS" />
    <xs:enumeration value="ROGUE_APS" />
    <xs:enumeration value="INTERFERERS" />
    <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
    <xs:enumeration value="MAP_CHANGE" />
    <xs:enumeration value="MAC_ADDRESS" />
    <xs:enumeration value="ASSET_NAME" />
    <xs:enumeration value="ASSET_CATEGORY" />
    <xs:enumeration value="ASSET_GROUP" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ANY" />
    <xs:enumeration value="LOW" />
    <xs:enumeration value="MEDIUM" />
    <xs:enumeration value="NORMAL" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
  <xs:restriction base="xs:string">

```



```

    <xs:enumeration value="NONE"/>
    <xs:enumeration value="ANY"/>
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="PANIC_BUTTON"/>
    <xs:enumeration value="DETACHED"/>
    <xs:enumeration value="TAMPERING"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="RSSI_MEASUREMENTS"/>
    <xs:enumeration value="RAW_LOCATION"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

ContainmentEventTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "boundary" : {
      "type" : "string",
      "enum" : [ "INSIDE", "OUTSIDE" ]
    },
    "zoneHierarchy" : {
      "type" : "string"
    },
    "zoneTimeout" : {
      "type" : "string"
    }
  }
}

```

BatteryEventTrigger

BatteryEventTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

```

```

<xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

<xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

<xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

<xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

<xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

<xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:attribute name="batteryStatus" type="batteryStatus"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

<xs:complexType name="notificationEventTrigger" abstract="true">
  <xs:sequence>
    <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="eventEntity" type="eventEntity"/>
</xs:complexType>

<xs:complexType name="presenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

<xs:complexType name="absenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

<xs:complexType name="movementEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
        <xs:attribute name="referenceMarkerName" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:attribute name="boundary" type="containmentBoundary"/>
        <xs:attribute name="zoneHierarchy" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

        <xs:attribute name="zoneTimeout" type="xs:string"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="type" type="type"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence>
                <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="streamType" type="streamType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY"/>
        <xs:enumeration value="LOW"/>
        <xs:enumeration value="MEDIUM"/>
        <xs:enumeration value="NORMAL"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WIRELESS_CLIENTS"/>
        <xs:enumeration value="TAGS"/>
        <xs:enumeration value="ROGUE_CLIENTS"/>
        <xs:enumeration value="ROGUE_APS"/>
        <xs:enumeration value="INTERFERERS"/>
        <xs:enumeration value="INTERFERER_DEVICE_TYPE"/>
        <xs:enumeration value="MAP_CHANGE"/>
        <xs:enumeration value="MAC_ADDRESS"/>
        <xs:enumeration value="ASSET_NAME"/>
        <xs:enumeration value="ASSET_CATEGORY"/>
        <xs:enumeration value="ASSET_GROUP"/>
    </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
      <xs:enumeration value="INSIDE"/>
      <xs:enumeration value="OUTSIDE"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="ANY"/>
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="PANIC_BUTTON"/>
      <xs:enumeration value="DETACHED"/>
      <xs:enumeration value="TAMPERING"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="RSSI_MEASUREMENTS"/>
      <xs:enumeration value="RAW_LOCATION"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

BatteryEventTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "batteryStatus" : {
      "type" : "string",
      "enum" : [ "ANY", "LOW", "MEDIUM", "NORMAL" ]
    }
  }
}

```

EmergencyEventTrigger

EmergencyEventTrigger.xml

```

{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

```

```
<xs:element name="AbsenceEventTrigger" type="absenceEventTrigger" />
<xs:element name="BatteryEventTrigger" type="batteryEventTrigger" />
<xs:element name="ContainmentEventTrigger" type="containmentEventTrigger" />
<xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger" />
<xs:element name="ExciterEventTrigger" type="exciterEventTrigger" />
<xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger" />
<xs:element name="MovementEventTrigger" type="movementEventTrigger" />
<xs:element name="NotificationEventTrigger" type="notificationEventTrigger" />
<xs:element name="PresenceEventTrigger" type="presenceEventTrigger" />
<xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger" />

<xs:complexType name="emergencyEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="type" type="type" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="notificationEventTrigger" abstract="true">
  <xs:sequence>
    <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="eventEntity" type="eventEntity" />
</xs:complexType>

<xs:complexType name="presenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="existingPresence" type="xs:boolean" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="absenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="movementEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required" />
      <xs:attribute name="referenceMarkerName" type="xs:string" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="streamType" type="streamType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NONE"/>
    <xs:enumeration value="ANY"/>
    <xs:enumeration value="UNKNOWN"/>
    <xs:enumeration value="PANIC_BUTTON"/>
    <xs:enumeration value="DETACHED"/>
    <xs:enumeration value="TAMPERING"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="eventEntity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WIRELESS_CLIENTS"/>
    <xs:enumeration value="TAGS"/>
    <xs:enumeration value="ROGUE_CLIENTS"/>
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="ROGUE_APS" />
        <xs:enumeration value="INTERFERERS" />
        <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
        <xs:enumeration value="MAP_CHANGE" />
        <xs:enumeration value="MAC_ADDRESS" />
        <xs:enumeration value="ASSET_NAME" />
        <xs:enumeration value="ASSET_CATEGORY" />
        <xs:enumeration value="ASSET_GROUP" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE" />
        <xs:enumeration value="OUTSIDE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY" />
        <xs:enumeration value="LOW" />
        <xs:enumeration value="MEDIUM" />
        <xs:enumeration value="NORMAL" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RSSI_MEASUREMENTS" />
        <xs:enumeration value="RAW_LOCATION" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

EmergencyEventTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "type" : {
      "type" : "string",
      "enum" : [ "NONE", "ANY", "UNKNOWN", "PANIC_BUTTON", "DETACHED", "TAMPERING" ]
    }
  }
}

```

ExciterEventTrigger

ExciterEventTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

  <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

  <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

  <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

  <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

  <xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

  <xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

  <xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence>
          <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="notificationEventTrigger" abstract="true">
    <xs:sequence>
      <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="eventEntity" type="eventEntity"/>
  </xs:complexType>

  <xs:complexType name="presenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="absenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```



```
</xs:complexType>

<xs:complexType name="movementEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
      <xs:attribute name="referenceMarkerName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="type" type="type"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="streamType" type="streamType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="eventEntity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WIRELESS_CLIENTS"/>
    <xs:enumeration value="TAGS"/>
    <xs:enumeration value="ROGUE_CLIENTS"/>
    <xs:enumeration value="ROGUE_APS"/>
    <xs:enumeration value="INTERFERERS"/>
  </xs:restriction>
</xs:simpleType>
```

```

        <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
        <xs:enumeration value="MAP_CHANGE" />
        <xs:enumeration value="MAC_ADDRESS" />
        <xs:enumeration value="ASSET_NAME" />
        <xs:enumeration value="ASSET_CATEGORY" />
        <xs:enumeration value="ASSET_GROUP" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE" />
        <xs:enumeration value="OUTSIDE" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY" />
        <xs:enumeration value="LOW" />
        <xs:enumeration value="MEDIUM" />
        <xs:enumeration value="NORMAL" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="NONE" />
        <xs:enumeration value="ANY" />
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="PANIC_BUTTON" />
        <xs:enumeration value="DETACHED" />
        <xs:enumeration value="TAMPERING" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RSSI_MEASUREMENTS" />
        <xs:enumeration value="RAW_LOCATION" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

ExciterEventTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
        "INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
        "ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "triggerExciterMacs" : {
      "type" : "array",
      "items" : {

```

```

        "type" : "string"
    }
}
}
}

```

StreamingNotificationTrigger

StreamingNotificationTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

  <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

  <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

  <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

  <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

  <xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

  <xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

  <xs:complexType name="emergencyEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="type" type="type"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="notificationEventTrigger" abstract="true">
    <xs:sequence>
      <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="eventEntity" type="eventEntity"/>
  </xs:complexType>

  <xs:complexType name="presenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="absenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">

```

```

        <xs:sequence/>
        <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="movementEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
            <xs:attribute name="referenceMarkerName" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="containmentEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="boundary" type="containmentBoundary"/>
            <xs:attribute name="zoneHierarchy" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="batteryStatus" type="batteryStatus"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence>
                <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
            <xs:attribute name="streamType" type="streamType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
        <xs:extension base="notificationEventTrigger">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="type">

```

```

    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE" />
      <xs:enumeration value="ANY" />
      <xs:enumeration value="UNKNOWN" />
      <xs:enumeration value="PANIC_BUTTON" />
      <xs:enumeration value="DETACHED" />
      <xs:enumeration value="TAMPERING" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
      <xs:enumeration value="WIRELESS_CLIENTS" />
      <xs:enumeration value="TAGS" />
      <xs:enumeration value="ROGUE_CLIENTS" />
      <xs:enumeration value="ROGUE_APS" />
      <xs:enumeration value="INTERFERERS" />
      <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
      <xs:enumeration value="MAP_CHANGE" />
      <xs:enumeration value="MAC_ADDRESS" />
      <xs:enumeration value="ASSET_NAME" />
      <xs:enumeration value="ASSET_CATEGORY" />
      <xs:enumeration value="ASSET_GROUP" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
      <xs:enumeration value="INSIDE" />
      <xs:enumeration value="OUTSIDE" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ANY" />
      <xs:enumeration value="LOW" />
      <xs:enumeration value="MEDIUM" />
      <xs:enumeration value="NORMAL" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="streamType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="RSSI_MEASUREMENTS" />
      <xs:enumeration value="RAW_LOCATION" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

StreamingNotificationTrigger.json

```

{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {

```

```

        "type" : "string"
      }
    },
    "type" : {
      "type" : "string",
      "enum" : [ "NONE", "ANY", "UNKNOWN", "PANIC_BUTTON", "DETACHED", "TAMPERING" ]
    }
  }
}

```

MapInfoChangeEventTrigger

MapInfoChangeEventTrigger.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="AbsenceEventTrigger" type="absenceEventTrigger"/>

  <xs:element name="BatteryEventTrigger" type="batteryEventTrigger"/>

  <xs:element name="ContainmentEventTrigger" type="containmentEventTrigger"/>

  <xs:element name="EmergencyEventTrigger" type="emergencyEventTrigger"/>

  <xs:element name="ExciterEventTrigger" type="exciterEventTrigger"/>

  <xs:element name="MapInfoChangeEventTrigger" type="mapInfoChangeEventTrigger"/>

  <xs:element name="MovementEventTrigger" type="movementEventTrigger"/>

  <xs:element name="NotificationEventTrigger" type="notificationEventTrigger"/>

  <xs:element name="PresenceEventTrigger" type="presenceEventTrigger"/>

  <xs:element name="StreamingNotificationTrigger" type="streamingNotificationTrigger"/>

  <xs:complexType name="mapInfoChangeEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="notificationEventTrigger" abstract="true">
    <xs:sequence>
      <xs:element name="entityFilters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="eventEntity" type="eventEntity"/>
  </xs:complexType>

  <xs:complexType name="presenceEventTrigger">
    <xs:complexContent>
      <xs:extension base="notificationEventTrigger">
        <xs:sequence/>
        <xs:attribute name="existingPresence" type="xs:boolean" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

</xs:complexType>

<xs:complexType name="absenceEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="movementEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
      <xs:attribute name="referenceMarkerName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="containmentEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="zoneHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
      <xs:attribute name="type" type="type"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEventTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence>
        <xs:element name="triggerExciterMacs" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotificationTrigger">
  <xs:complexContent>
    <xs:extension base="notificationEventTrigger">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="streamType" type="streamType" />
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:simpleType name="eventEntity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WIRELESS_CLIENTS" />
    <xs:enumeration value="TAGS" />
    <xs:enumeration value="ROGUE_CLIENTS" />
    <xs:enumeration value="ROGUE_APS" />
    <xs:enumeration value="INTERFERERS" />
    <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
    <xs:enumeration value="MAP_CHANGE" />
    <xs:enumeration value="MAC_ADDRESS" />
    <xs:enumeration value="ASSET_NAME" />
    <xs:enumeration value="ASSET_CATEGORY" />
    <xs:enumeration value="ASSET_GROUP" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INSIDE" />
    <xs:enumeration value="OUTSIDE" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ANY" />
    <xs:enumeration value="LOW" />
    <xs:enumeration value="MEDIUM" />
    <xs:enumeration value="NORMAL" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NONE" />
    <xs:enumeration value="ANY" />
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="PANIC_BUTTON" />
    <xs:enumeration value="DETACHED" />
    <xs:enumeration value="TAMPERING" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="streamType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="RSSI_MEASUREMENTS" />
    <xs:enumeration value="RAW_LOCATION" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

MapInfoChangeEventTrigger.json

AssociationEventTrigger

AssociationEventTrigger.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="EmergencyEventTrigger"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="ANY"/>
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="PANIC_BUTTON"/>
      <xs:enumeration value="DETACHED"/>
      <xs:enumeration value="TAMPERING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

AssociationEventTrigger.json

```
{
  "type" : "object",
  "properties" : {
    "eventEntity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "entityFilters" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    },
    "association" : {
      "type" : "boolean",
      "required" : true
    },
    "alertInterval" : {
      "type" : "integer"
    },
    "sendIfIpAddressAbsent" : {
      "type" : "boolean",
      "required" : true
    }
  }
}
```

AbsenceEvent

AbsenceEvent.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="Dimension"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```

<xs:simpleType name="Unit">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FEET"/>
    <xs:enumeration value="METER"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

AbsenceEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "absenceDurationInMinutes" : {
      "type" : "integer"
    },
    "lastSeen" : {
      "type" : "string"
    },
    "timestamp" : {
      "type" : "string"
    }
  }
}

```

AbsenceEvent.proto

```

option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "AbsenceEventProtos";

import "event-entity.proto";
import "map-coordinate.proto";

```

```

message PBAbsenceEvent {
    required string subscriptionName = 1;
    optional uint64 timeStampInMillis = 2;
    optional PBEventEntity eventEntity = 3;
    optional string deviceId = 4;
    optional string locationMapHierarchy = 5;
    optional PMapCoordinate locationCoordinate = 6;
    optional int32 absenceDurationInMinutes = 7;
    optional uint64 lastSeenInMillis = 8;
}

```

PresenceEvent

PresenceEvent.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns2="EmergencyEvent" xmlns:ns1="MapCoordinate"
xmlns:ns4="GeoCoordinate" xmlns:ns3="Dimension"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns5="RogueClientLocation"
xmlns:ns6="RogueAPLocation">

    <xs:import namespace="MapCoordinate" schemaLocation="PresenceEvent.xsd"/>
    <xs:import namespace="EmergencyEvent" schemaLocation="PresenceEvent.xsd"/>
    <xs:import namespace="Dimension" schemaLocation="PresenceEvent.xsd"/>
    <xs:import namespace="GeoCoordinate" schemaLocation="PresenceEvent.xsd"/>
    <xs:import namespace="RogueClientLocation" schemaLocation="PresenceEvent.xsd"/>
    <xs:import namespace="RogueAPLocation" schemaLocation="PresenceEvent.xsd"/>

    <xs:element name="APInterface" type="apInterface"/>
    <xs:element name="AbsenceEvent" type="absenceEvent"/>
    <xs:element name="AccessPoint" type="accessPoint"/>
    <xs:element name="BatteryEvent" type="batteryEvent"/>
    <xs:element name="BatteryInfo" type="batteryInfo"/>
    <xs:element name="Building" type="building"/>
    <xs:element name="Campus" type="campus"/>
    <xs:element name="ContainmentEvent" type="containmentEvent"/>
    <xs:element name="DeviceExciter" type="deviceExciter"/>
    <xs:element name="Dimension" type="dimension"/>

```

```
<xs:element name="EPCTelemetryData" type="epcTelemetryData"/>
<xs:element name="EmergencyEvent" type="emergencyEvent"/>
<xs:element name="EmergencyInfo" type="emergencyInfo"/>
<xs:element name="Exciter" type="exciter"/>
<xs:element name="ExciterEvent" type="exciterEvent"/>
<xs:element name="Floor" type="floor"/>
<xs:element name="GPSThreshold" type="gpsThreshold"/>
<xs:element name="GPSThresholdEvent" type="gpsThresholdEvent"/>
<xs:element name="GeoCoordinate" type="geoCoordinate"/>
<xs:element name="GeoTelemetryData" type="geoTelemetryData"/>
<xs:element name="Image" type="image"/>
<xs:element name="InterfererLocation" type="interfererLocation"/>
<xs:element name="LocationFilterRail" type="locationFilterRail"/>
<xs:element name="LocationFilterRegion" type="locationFilterRegion"/>
<xs:element name="MapCoordinate" type="mapCoordinate"/>
<xs:element name="MapInfo" type="mapInfo"/>
<xs:element name="MapInfoChangeEvent" type="mapInfoChangeEvent"/>
<xs:element name="MotionTelemetryData" type="motionTelemetryData"/>
<xs:element name="MovementEvent" type="movementEvent"/>
<xs:element name="Obstacle" type="obstacle"/>
<xs:element name="PresenceEvent" type="presenceEvent"/>
<xs:element name="ReferenceMarker" type="referenceMarker"/>
<xs:element name="RogueAPLocation" type="rogueAPLocation"/>
<xs:element name="RogueClientLocation" type="rogueClientLocation"/>
<xs:element name="Statistics" type="statistics"/>
<xs:element name="StreamingNotification" type="streamingNotification"/>
<xs:element name="TagLocation" type="tagLocation"/>
<xs:element name="TelemetryData" type="defaultTelemetryMeasurementData"/>
<xs:element name="VendorData" nillable="true" type="xs:anyType"/>
<xs:element name="WiredClientLocation" type="wiredClientLocation"/>
<xs:element name="WirelessClientLocation" type="wirelessClientLocation"/>
<xs:element name="Zone" type="zone"/>
<xs:element name="location" type="location"/>
```

```
<xs:complexType name="presenceEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="notificationEvent" abstract="true">
  <xs:sequence>
    <xs:element name="locationCoordinate" type="mapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="subscriptionName" type="xs:string"/>
  <xs:attribute name="timeStamp" type="xs:string"/>
  <xs:attribute name="entity" type="eventEntity"/>
  <xs:attribute name="deviceId" type="xs:string"/>
  <xs:attribute name="locationMapHierarchy" type="xs:string"/>
</xs:complexType>

<xs:complexType name="mapCoordinate">
  <xs:sequence/>
  <xs:attribute name="x" type="xs:float" use="required"/>
  <xs:attribute name="y" type="xs:float" use="required"/>
  <xs:attribute name="unit" type="ns1:Unit"/>
</xs:complexType>

<xs:complexType name="absenceEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
      <xs:attribute name="lastSeen" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
      <xs:attribute name="percentRemaining" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="triggerExciterMac" type="xs:string"/>
      <xs:attribute name="triggerExciterName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="containmentEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="areaType" type="areaType"/>
      <xs:attribute name="containerHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="emergencyEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence>
        <xs:element name="vendorData" type="xs:base64Binary" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="type" type="ns2:Type"/>
      <xs:attribute name="vendorId" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="mapInfoChangeEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence>
        <xs:element ref="Campus"/>
      </xs:sequence>
      <xs:attribute name="changeType" type="changeType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="campus">
  <xs:sequence>
    <xs:element ref="Dimension" minOccurs="0"/>
    <xs:element ref="Image" minOccurs="0"/>
    <xs:element ref="Building" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="objectVersion" type="xs:int" use="required"/>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="dimension">
  <xs:sequence/>
  <xs:attribute name="length" type="xs:float" use="required"/>
  <xs:attribute name="width" type="xs:float" use="required"/>
  <xs:attribute name="height" type="xs:float" use="required"/>
  <xs:attribute name="offsetX" type="xs:float" use="required"/>
  <xs:attribute name="offsetY" type="xs:float" use="required"/>
  <xs:attribute name="unit" type="ns3:Unit"/>
</xs:complexType>

<xs:complexType name="image">
  <xs:sequence>
    <xs:element name="imageData" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="imageName" type="xs:string"/>
</xs:complexType>

<xs:complexType name="building">
  <xs:sequence>
    <xs:element ref="Dimension" minOccurs="0"/>
    <xs:element ref="Image" minOccurs="0"/>
    <xs:element ref="Floor" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="objectVersion" type="xs:int" use="required"/>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

```

```

<xs:complexType name="floor">
  <xs:sequence>
    <xs:element ref="Dimension" minOccurs="0"/>
    <xs:element ref="Image" minOccurs="0"/>
    <xs:element ref="GPSMarker" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Zone" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Obstacle" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="AccessPoint" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="ReferenceMarker" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Exciter" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="LocationFilterRegion" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="LocationFilterRail" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="objectVersion" type="xs:int" use="required"/>
  <xs:attribute name="name" type="xs:string"/>
  <xs:attribute name="isOutdoor" type="xs:boolean" use="required"/>
  <xs:attribute name="floorNumber" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="gpsMarker">
  <xs:sequence>
    <xs:element ref="GeoCoordinate" minOccurs="0"/>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="geoCoordinate">
  <xs:sequence/>
  <xs:attribute name="latitude" type="xs:double" use="required"/>
  <xs:attribute name="longitude" type="xs:double" use="required"/>
  <xs:attribute name="unit" type="ns4:Unit"/>
</xs:complexType>

<xs:complexType name="zone">
  <xs:sequence>
    <xs:element name="ZoneCoordinate" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="obstacle">
  <xs:sequence>
    <xs:element name="ObstacleCoordinate" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="type" type="type"/>
  <xs:attribute name="attenuation" type="xs:float" use="required"/>
</xs:complexType>

<xs:complexType name="accessPoint">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
    <xs:element name="ApInterface" type="apInterface" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
  <xs:attribute name="radioMacAddress" type="xs:string"/>
  <xs:attribute name="ethMacAddress" type="xs:string"/>
  <xs:attribute name="ipAddress" type="xs:string"/>
  <xs:attribute name="numOfSlots" type="xs:int" use="required"/>
  <xs:attribute name="apMode" type="apMode"/>
</xs:complexType>

```

```

<xs:complexType name="apInterface">
  <xs:sequence/>
  <xs:attribute name="band" type="band"/>
  <xs:attribute name="slotNumber" type="xs:int" use="required"/>
  <xs:attribute name="channelAssignment" type="xs:int" use="required"/>
  <xs:attribute name="channelNumber" type="xs:int" use="required"/>
  <xs:attribute name="txPowerLevel" type="xs:int" use="required"/>
  <xs:attribute name="antennaPattern" type="xs:string"/>
  <xs:attribute name="antennaAngle" type="xs:float" use="required"/>
  <xs:attribute name="antennaElevAngle" type="xs:float" use="required"/>
  <xs:attribute name="antennaGain" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="referenceMarker">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="exciter">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
  <xs:attribute name="macAddress" type="xs:string"/>
  <xs:attribute name="range" type="xs:double" use="required"/>
  <xs:attribute name="isPerimeter" type="xs:boolean" use="required"/>
</xs:complexType>

<xs:complexType name="locationFilterRegion">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="regionType" type="regionType"/>
</xs:complexType>

<xs:complexType name="locationFilterRail">
  <xs:sequence>
    <xs:element name="RailCoordinates" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="RailExclusionCoordinates" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="railBoundaryWidth" type="xs:float" use="required"/>
</xs:complexType>

<xs:complexType name="movementEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
      <xs:attribute name="referenceMarkerName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="streamingNotification">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:choice>
        <xs:element ref="location"/>
        <xs:element ref="WirelessClientLocation"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```



```

        <xs:element ref="TagLocation"/>
        <xs:element ref="InterfererLocation"/>
        <xs:element ref="RogueClientLocation"/>
        <xs:element ref="RogueAPLocation"/>
        <xs:element ref="WiredClientLocation"/>
    </xs:choice>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="location" abstract="true">
    <xs:sequence>
        <xs:element ref="MapInfo" minOccurs="0"/>
        <xs:element ref="MapCoordinate" minOccurs="0"/>
        <xs:element ref="Statistics" minOccurs="0"/>
        <xs:element ref="GeoCoordinate" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="macAddress" type="xs:string"/>
    <xs:attribute name="currentlyTracked" type="xs:boolean" use="required"/>
    <xs:attribute name="confidenceFactor" type="xs:float" use="required"/>
    <xs:attribute name="historyLogReason" type="historyReason"/>
</xs:complexType>

<xs:complexType name="mapInfo">
    <xs:sequence>
        <xs:element ref="Dimension" minOccurs="0"/>
        <xs:element ref="Image" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="mapHierarchyString" type="xs:string"/>
    <xs:attribute name="floorRefId" type="xs:long" use="required"/>
</xs:complexType>

<xs:complexType name="statistics">
    <xs:sequence/>
    <xs:attribute name="currentServerTime" type="xs:string"/>
    <xs:attribute name="firstLocatedTime" type="xs:string"/>
    <xs:attribute name="lastLocatedTime" type="xs:string"/>
</xs:complexType>

<xs:complexType name="wirelessClientLocation">
    <xs:complexContent>
        <xs:extension base="location">
            <xs:sequence/>
            <xs:attribute name="ipAddress">
                <xs:simpleType>
                    <xs:list itemType="xs:string"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="userName" type="xs:string"/>
            <xs:attribute name="ssid" type="xs:string"/>
            <xs:attribute name="band" type="band"/>
            <xs:attribute name="apMacAddress" type="xs:string"/>
            <xs:attribute name="isGuestUser" type="xs:boolean" use="required"/>
            <xs:attribute name="dot11Status" type="clientDot11Status"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="tagLocation">
    <xs:complexContent>
        <xs:extension base="location">
            <xs:sequence>
                <xs:element name="TelemetryData" type="telemetryInfo" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        <xs:element ref="BatteryInfo" minOccurs="0"/>
        <xs:element ref="VendorData" minOccurs="0"/>
        <xs:element ref="EmergencyInfo" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ExciterEncounter" type="deviceExciter" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="lastBeaconTime" type="xs:string"/>
    <xs:attribute name="lastBeaconSequenceNumber" type="xs:int" use="required"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="telemetryInfo">
    <xs:choice>
        <xs:element ref="TelemetryData"/>
        <xs:element ref="EPCTelemetryData"/>
        <xs:element ref="GeoTelemetryData"/>
        <xs:element ref="MotionTelemetryData"/>
    </xs:choice>
    <xs:attribute name="type" type="telemetryType"/>
    <xs:attribute name="units" type="xs:string"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="telemetryMeasurementData" abstract="true">
    <xs:sequence>
        <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="defaultTelemetryMeasurementData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence>
                <xs:element name="measurementData" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="measurementDataType" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="epcTelemetryData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence/>
            <xs:attribute name="header" type="xs:string"/>
            <xs:attribute name="manufacturer" type="xs:string"/>
            <xs:attribute name="productSKU" type="xs:string"/>
            <xs:attribute name="serialNo" type="xs:string"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="geoTelemetryData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence/>
            <xs:attribute name="latitude" type="xs:string"/>
            <xs:attribute name="longitude" type="xs:string"/>
            <xs:attribute name="altitude" type="xs:string"/>
            <xs:attribute name="altitudeType" type="altitudeType"/>
            <xs:attribute name="datum" type="xs:int" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motionTelemetryData">
    <xs:complexContent>
      <xs:extension base="telemetryMeasurementData">
        <xs:sequence/>
        <xs:attribute name="motionType" type="motionType" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="batteryInfo">
    <xs:sequence/>
    <xs:attribute name="tolerance" type="xs:int" />
    <xs:attribute name="percentRemaining" type="xs:int" />
    <xs:attribute name="daysRemaining" type="xs:int" />
    <xs:attribute name="batteryAge" type="xs:long" />
    <xs:attribute name="lastReceivedTime" type="xs:string" />
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" />
  </xs:complexType>

  <xs:complexType name="vendorData">
    <xs:sequence/>
    <xs:attribute name="vendorId" type="xs:int" use="required" />
    <xs:attribute name="elementId" type="xs:short" use="required" />
    <xs:attribute name="data" type="xs:base64Binary" />
    <xs:attribute name="lastReceivedTime" type="xs:string" />
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required" />
  </xs:complexType>

  <xs:complexType name="emergencyInfo">
    <xs:sequence/>
    <xs:attribute name="isActive" type="xs:boolean" use="required" />
    <xs:attribute name="reason" type="xs:string" />
    <xs:attribute name="tamperState" type="xs:string" />
    <xs:attribute name="tamperOpenedCount" type="xs:int" use="required" />
    <xs:attribute name="lastReceivedTime" type="xs:string" />
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required" />
    <xs:attribute name="fragmentNumber" type="xs:byte" use="required" />
  </xs:complexType>

  <xs:complexType name="deviceExciter">
    <xs:sequence/>
    <xs:attribute name="macAddress" type="xs:string" />
    <xs:attribute name="timestamp" type="xs:string" />
    <xs:attribute name="inRange" type="xs:boolean" use="required" />
    <xs:attribute name="isPerimeter" type="xs:boolean" use="required" />
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required" />
    <xs:attribute name="fragmentNumber" type="xs:byte" use="required" />
    <xs:attribute name="isActive" type="xs:boolean" use="required" />
    <xs:attribute name="assocFloorId" type="xs:long" use="required" />
    <xs:attribute name="index" type="xs:short" use="required" />
  </xs:complexType>

  <xs:complexType name="interfererLocation">
    <xs:complexContent>
      <xs:extension base="location">
        <xs:sequence/>
        <xs:attribute name="deviceType" type="interfererDeviceType" />
        <xs:attribute name="zoneOfImpact" type="xs:float" use="required" />
        <xs:attribute name="dutyCycle" type="xs:short" use="required" />
        <xs:attribute name="severity" type="xs:short" use="required" />
        <xs:attribute name="deviceKey" type="xs:string" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

        <xs:attribute name="affectedChannels" type="xs:string"/>
        <xs:attribute name="affectedBand" type="band"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="rogueClientLocation">
    <xs:complexContent>
        <xs:extension base="location">
            <xs:sequence/>
            <xs:attribute name="numOfDetectingAPs" type="xs:int" use="required"/>
            <xs:attribute name="bssid" type="xs:string"/>
            <xs:attribute name="containmentLevel" type="ns5:ContainmentLevel"/>
            <xs:attribute name="state" type="ns5:State"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="rogueAPLocation">
    <xs:complexContent>
        <xs:extension base="location">
            <xs:sequence/>
            <xs:attribute name="totalClients" type="xs:int" use="required"/>
            <xs:attribute name="numOfDetectingAPs" type="xs:int" use="required"/>
            <xs:attribute name="type" type="ns6:Type"/>
            <xs:attribute name="containmentLevel" type="ns6:ContainmentLevel"/>
            <xs:attribute name="state" type="ns6:State"/>
            <xs:attribute name="maxDetectedRssi" type="xs:int" use="required"/>
            <xs:attribute name="ssid" type="xs:string"/>
            <xs:attribute name="maxDetectingAPMacAddress" type="xs:string"/>
            <xs:attribute name="maxRssiRadioType" type="xs:int" use="required"/>
            <xs:attribute name="zoneOfImpact" type="xs:float" use="required"/>
            <xs:attribute name="hiddenSSID" type="xs:int" use="required"/>
            <xs:attribute name="rogueStChBy" type="ns6:RogueState"/>
            <xs:attribute name="rogueClChBy" type="ns6:RogueClass"/>
            <xs:attribute name="classifiedRSSI" type="xs:int" use="required"/>
            <xs:attribute name="classifiedByAP" type="xs:string"/>
            <xs:attribute name="className" type="xs:string"/>
            <xs:attribute name="classification" type="xs:int" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="wiredClientLocation">
    <xs:complexContent>
        <xs:extension base="location">
            <xs:sequence>
                <xs:element name="civicAddressAsString" type="xs:string" minOccurs="0"/>
                <xs:element name="PortAssociation" type="wiredPortAssociation" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="username802Dot1X" type="xs:string"/>
            <xs:attribute name="ipAddress">
                <xs:simpleType>
                    <xs:list itemType="xs:string"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="elin" type="xs:string"/>
            <xs:attribute name="serialNumber" type="xs:string"/>
            <xs:attribute name="udi" type="xs:string"/>
            <xs:attribute name="vlanId" type="xs:int" use="required"/>
            <xs:attribute name="vlanName" type="xs:string"/>
            <xs:attribute name="switchIpAddress" type="xs:int" use="required"/>
            <xs:attribute name="portNumber" type="xs:short" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="slotNumber" type="xs:short" use="required"/>
        <xs:attribute name="state" type="associationState"/>
        <xs:attribute name="portType" type="xs:byte" use="required"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="wiredPortAssociation">
    <xs:complexContent>
        <xs:extension base="aesObject">
            <xs:sequence>
                <xs:element name="duplicate" type="xs:boolean"/>
                <xs:element name="moduleNumber" type="xs:short"/>
                <xs:element name="portId" type="xs:long"/>
                <xs:element name="portNumber" type="xs:short"/>
                <xs:element name="portType" type="xs:byte"/>
                <xs:element name="slotNumber" type="xs:short"/>
                <xs:element name="state" type="xs:byte"/>
                <xs:element name="switchIpAddress" type="xs:string" minOccurs="0"/>
                <xs:element name="transitionTime" type="xs:long"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="aesObject" abstract="true">
    <xs:sequence>
        <xs:element name="changedOn" type="xs:long"/>
        <xs:element name="mask" type="xs:int"/>
        <xs:element name="objectId" type="xs:long"/>
        <xs:element name="options" type="xs:short"/>
        <xs:element name="parentId" type="xs:long"/>
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="eventEntity">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WIRELESS_CLIENTS"/>
        <xs:enumeration value="TAGS"/>
        <xs:enumeration value="ROGUE_CLIENTS"/>
        <xs:enumeration value="ROGUE_APS"/>
        <xs:enumeration value="INTERFERERS"/>
        <xs:enumeration value="INTERFERER_DEVICE_TYPE"/>
        <xs:enumeration value="MAP_CHANGE"/>
        <xs:enumeration value="MAC_ADDRESS"/>
        <xs:enumeration value="ASSET_NAME"/>
        <xs:enumeration value="ASSET_CATEGORY"/>
        <xs:enumeration value="ASSET_GROUP"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ANY"/>
        <xs:enumeration value="LOW"/>
        <xs:enumeration value="MEDIUM"/>
        <xs:enumeration value="NORMAL"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="containmentBoundary">
    <xs:restriction base="xs:string">
        <xs:enumeration value="INSIDE"/>
        <xs:enumeration value="OUTSIDE"/>
    </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="areaType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CAMPUS" />
      <xs:enumeration value="BUILDING" />
      <xs:enumeration value="FLOOR" />
      <xs:enumeration value="ZONE" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="changeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ADDED" />
      <xs:enumeration value="MODIFIED" />
      <xs:enumeration value="DELETED" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN" />
      <xs:enumeration value="THICK_WALL" />
      <xs:enumeration value="NORMAL_WALL" />
      <xs:enumeration value="THICK_DOOR" />
      <xs:enumeration value="NORMAL_DOOR" />
      <xs:enumeration value="CUBICLE" />
      <xs:enumeration value="GLASS" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="apMode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="LOCAL" />
      <xs:enumeration value="MONITOR" />
      <xs:enumeration value="REAP" />
      <xs:enumeration value="ROGUE_DETECTOR" />
      <xs:enumeration value="SNIFFER" />
      <xs:enumeration value="BRIDGE" />
      <xs:enumeration value="UNKNOWN" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="band">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN" />
      <xs:enumeration value="IEEE_802_11_A" />
      <xs:enumeration value="IEEE_802_11_B" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="regionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="INSIDE" />
      <xs:enumeration value="OUTSIDE" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="historyReason">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN" />
      <xs:enumeration value="FLOOR_CHANGE" />
    </xs:restriction>
  </xs:simpleType>

```

```

    <xs:enumeration value="CHOKEPOINT_ENCOUNTER" />
    <xs:enumeration value="INOUT_OF_AREA" />
    <xs:enumeration value="SIGNIFICANT_MOVEMENT" />
    <xs:enumeration value="LOCP_MESSAGE" />
    <xs:enumeration value="INFORMATION_CHANGE" />
    <xs:enumeration value="DEVICE_TYPE_CHANGE" />
    <xs:enumeration value="TAG_EMERGENCY" />
    <xs:enumeration value="LOCATION_CONTRIBUTOR_CHANGE" />
    <xs:enumeration value="INTERFERER_MERGE" />
    <xs:enumeration value="INTERFERER_CLUSTER_CENTRE_CHANGE" />
    <xs:enumeration value="INTERFERER_SEVERITY_CHANGE" />
    <xs:enumeration value="STATUS_CHANGE" />
    <xs:enumeration value="SECURITY_POLICY_CHANGE" />
    <xs:enumeration value="NETWORK_STATUS_CHANGE" />
    <xs:enumeration value="HASHKEY_CHANGE" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="clientDot11Status">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="ASSOCIATED" />
    <xs:enumeration value="PROBING" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="telemetryType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="GPS" />
    <xs:enumeration value="TEMPERATURE" />
    <xs:enumeration value="MOTION" />
    <xs:enumeration value="DISTANCE" />
    <xs:enumeration value="FUEL" />
    <xs:enumeration value="QUANTITY" />
    <xs:enumeration value="PRESSURE" />
    <xs:enumeration value="HUMIDITY" />
    <xs:enumeration value="STATUS" />
    <xs:enumeration value="EPC" />
    <xs:enumeration value="MOTIONPROB" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="altitudeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="AT_METERS" />
    <xs:enumeration value="AT_FLOOR" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="motionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="NO_MOVEMENT" />
    <xs:enumeration value="ACCELERATION" />
    <xs:enumeration value="MOVEMENT" />
    <xs:enumeration value="DECELERATION" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="interfererDeviceType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
  </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="BLUETOOTH"/>
<xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_1"/>
<xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_1"/>
<xs:enumeration value="TDD_1_PHONE"/>
<xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_2"/>
<xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_2"/>
<xs:enumeration value="MICROWAVE_OVEN_1"/>
<xs:enumeration value="DEVICE_802_11_FH"/>
<xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_3"/>
<xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_3"/>
<xs:enumeration value="MICROWAVE_OVEN_2"/>
<xs:enumeration value="BLUETOOTH_PAGING_INQUIRY_DEVICE"/>
<xs:enumeration value="GENERIC_TDD"/>
<xs:enumeration value="JAMMER"/>
<xs:enumeration value="RFID_READER"/>
<xs:enumeration value="CLASS_1_NETWORK"/>
<xs:enumeration value="CLASS_2_NETWORK"/>
<xs:enumeration value="GENERIC_WIDEBAND"/>
<xs:enumeration value="BRIDGE_ROUTER_1"/>
<xs:enumeration value="GENERIC_WAVEFORM"/>
<xs:enumeration value="GENERIC_DECT_LIKE"/>
<xs:enumeration value="VIDEO"/>
<xs:enumeration value="DEVICE_802_15_4"/>
<xs:enumeration value="DEVICE_802_11_NORMAL"/>
<xs:enumeration value="DEVICE_802_11_SPECTRALLY_INVERTED"/>
<xs:enumeration value="DEVICE_802_11_NONSTD_CHANNEL"/>
<xs:enumeration value="DEVICE_802_11_SUPERG"/>
<xs:enumeration value="RADAR"/>
<xs:enumeration value="MOTOROLA_CANOPY"/>
<xs:enumeration value="XBOX"/>
<xs:enumeration value="WIFI_MOBILE"/>
<xs:enumeration value="WIFI_FIXED"/>
<xs:enumeration value="EXALT"/>
<xs:enumeration value="CONTINUOUS_TRANSMITTER"/>
<xs:enumeration value="DECT_LIKE_NETWORK"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="associationState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CONNECTED"/>
    <xs:enumeration value="DISCONNECTED"/>
    <xs:enumeration value="UNKNOWN"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

PresenceEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
        "INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
        "ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    }
  }
}

```



```

    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "timestamp" : {
      "type" : "string"
    }
  }
}

```

presence-event.proto

```

option java_package = "com.cisco.mse.cas.notification.events";
option java_outer_classname = "PreseceEventProtos";

import "event-entity.proto";
import "map-coordinate.proto";
message PBPresenceEvent {
  required string subscriptionName = 1;

  optional uint64 timeStampInMillis = 2;

  optional PBEventEntity eventEntity = 3;

  optional string deviceId = 4;

  optional string locationMapHierarchy = 5;

  optional PMapCoordinate locationCoordinate = 6;
}

```

MovementEvent

MovementEvent.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="RogueClientLocation"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="ContainmentLevel">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="UNASSIGNED"/>
      <xs:enumeration value="LEVEL1"/>
      <xs:enumeration value="LEVEL2"/>
      <xs:enumeration value="LEVEL3"/>
    </xs:restriction>
  </xs:simpleType>
}

```

```

        <xs:enumeration value="LEVEL4" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="State">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNKNOWN" />
        <xs:enumeration value="INITIALIZING" />
        <xs:enumeration value="PENDING" />
        <xs:enumeration value="ALERT" />
        <xs:enumeration value="CONTAINED" />
        <xs:enumeration value="THREAT" />
        <xs:enumeration value="CONTAINEDPENDING" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

MovementEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "moveDistanceInFt" : {
      "type" : "number"
    },
    "referenceMarkerName" : {
      "type" : "string"
    },
    "timestamp" : {
      "type" : "string"
    }
  }
}

```

movement-event.proto

```

option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "MovementEventProtos";

import "event-entity.proto";
import "map-coordinate.proto";

message PBMovementEvent {
    required string subscriptionName = 1;

    optional uint64 timeStampInMillis = 2;

    optional PBEventEntity eventEntity = 3;

    optional string deviceId = 4;

    optional string locationMapHierarchy = 5;

    optional PBMAPCoordinate locationCoordinate = 6;

    optional float moveDistanceInFt = 7;

    optional string referenceMarkerName = 8;
}

```

ContainmentEvent

ContainmentEvent.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:ns2="EmergencyEvent" xmlns:ns1="MapCoordinate"
xmlns:ns4="GeoCoordinate" xmlns:ns3="Dimension"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ns5="RogueClientLocation"
xmlns:ns6="RogueAPLocation">

    <xs:import namespace="MapCoordinate" schemaLocation="ContainmentEvent.xsd"/>

    <xs:import namespace="EmergencyEvent" schemaLocation="ContainmentEvent.xsd"/>

    <xs:import namespace="Dimension" schemaLocation="ContainmentEvent.xsd"/>

    <xs:import namespace="GeoCoordinate" schemaLocation="ContainmentEvent.xsd"/>

    <xs:import namespace="RogueClientLocation" schemaLocation="ContainmentEvent.xsd"/>

    <xs:import namespace="RogueAPLocation" schemaLocation="ContainmentEvent.xsd"/>

    <xs:element name="APInterface" type="apInterface"/>

    <xs:element name="AbsenceEvent" type="absenceEvent"/>

    <xs:element name="AccessPoint" type="accessPoint"/>

    <xs:element name="BatteryEvent" type="batteryEvent"/>

    <xs:element name="BatteryInfo" type="batteryInfo"/>

    <xs:element name="Building" type="building"/>

```

```
<xs:element name="Campus" type="campus" />
<xs:element name="ContainmentEvent" type="containmentEvent" />
<xs:element name="DeviceExciter" type="deviceExciter" />
<xs:element name="Dimension" type="dimension" />
<xs:element name="EPCTelemetryData" type="epcTelemetryData" />
<xs:element name="EmergencyEvent" type="emergencyEvent" />
<xs:element name="EmergencyInfo" type="emergencyInfo" />
<xs:element name="Exciter" type="exciter" />
<xs:element name="ExciterEvent" type="exciterEvent" />
<xs:element name="Floor" type="floor" />
<xs:element name="GPSTracker" type="gpsMarker" />
<xs:element name="GeoCoordinate" type="geoCoordinate" />
<xs:element name="GeoTelemetryData" type="geoTelemetryData" />
<xs:element name="Image" type="image" />
<xs:element name="InterfererLocation" type="interfererLocation" />
<xs:element name="LocationFilterRail" type="locationFilterRail" />
<xs:element name="LocationFilterRegion" type="locationFilterRegion" />
<xs:element name="MapCoordinate" type="mapCoordinate" />
<xs:element name="MapInfo" type="mapInfo" />
<xs:element name="MapInfoChangeEvent" type="mapInfoChangeEvent" />
<xs:element name="MotionTelemetryData" type="motionTelemetryData" />
<xs:element name="MovementEvent" type="movementEvent" />
<xs:element name="Obstacle" type="obstacle" />
<xs:element name="PresenceEvent" type="presenceEvent" />
<xs:element name="ReferenceMarker" type="referenceMarker" />
<xs:element name="RogueAPLocation" type="rogueAPLocation" />
<xs:element name="RogueClientLocation" type="rogueClientLocation" />
<xs:element name="Statistics" type="statistics" />
<xs:element name="StreamingNotification" type="streamingNotification" />
<xs:element name="TagLocation" type="tagLocation" />
<xs:element name="TelemetryData" type="defaultTelemetryMeasurementData" />
<xs:element name="VendorData" nillable="true" type="xs:anyType" />
```

```
<xs:element name="WiredClientLocation" type="wiredClientLocation"/>

<xs:element name="WirelessClientLocation" type="wirelessClientLocation"/>

<xs:element name="Zone" type="zone"/>

<xs:element name="location" type="location"/>

<xs:complexType name="containmentEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="boundary" type="containmentBoundary"/>
      <xs:attribute name="areaType" type="areaType"/>
      <xs:attribute name="containerHierarchy" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="notificationEvent" abstract="true">
  <xs:sequence>
    <xs:element name="locationCoordinate" type="mapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="subscriptionName" type="xs:string"/>
  <xs:attribute name="timeStamp" type="xs:string"/>
  <xs:attribute name="entity" type="eventEntity"/>
  <xs:attribute name="deviceId" type="xs:string"/>
  <xs:attribute name="locationMapHierarchy" type="xs:string"/>
</xs:complexType>

<xs:complexType name="mapCoordinate">
  <xs:sequence/>
  <xs:attribute name="x" type="xs:float" use="required"/>
  <xs:attribute name="y" type="xs:float" use="required"/>
  <xs:attribute name="unit" type="ns1:Unit"/>
</xs:complexType>

<xs:complexType name="absenceEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="absenceDurationInMinutes" type="xs:int" use="required"/>
      <xs:attribute name="lastSeen" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="batteryStatus" type="batteryStatus"/>
      <xs:attribute name="percentRemaining" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="exciterEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="triggerExciterMac" type="xs:string"/>
      <xs:attribute name="triggerExciterName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="emergencyEvent">
    <xs:complexContent>
      <xs:extension base="notificationEvent">
        <xs:sequence>
          <xs:element name="vendorData" type="xs:base64Binary" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="type" type="ns2:Type"/>
        <xs:attribute name="vendorId" type="xs:int" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="mapInfoChangeEvent">
    <xs:complexContent>
      <xs:extension base="notificationEvent">
        <xs:sequence>
          <xs:element ref="Campus"/>
        </xs:sequence>
        <xs:attribute name="changeType" type="changeType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="campus">
    <xs:sequence>
      <xs:element ref="Dimension" minOccurs="0"/>
      <xs:element ref="Image" minOccurs="0"/>
      <xs:element ref="Building" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="objectVersion" type="xs:int" use="required"/>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="dimension">
    <xs:sequence/>
    <xs:attribute name="length" type="xs:float" use="required"/>
    <xs:attribute name="width" type="xs:float" use="required"/>
    <xs:attribute name="height" type="xs:float" use="required"/>
    <xs:attribute name="offsetX" type="xs:float" use="required"/>
    <xs:attribute name="offsetY" type="xs:float" use="required"/>
    <xs:attribute name="unit" type="ns3:Unit"/>
  </xs:complexType>

  <xs:complexType name="image">
    <xs:sequence>
      <xs:element name="imageData" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="imageName" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="building">
    <xs:sequence>
      <xs:element ref="Dimension" minOccurs="0"/>
      <xs:element ref="Image" minOccurs="0"/>
      <xs:element ref="Floor" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="objectVersion" type="xs:int" use="required"/>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="floor">

```

```

<xs:sequence>
  <xs:element ref="Dimension" minOccurs="0"/>
  <xs:element ref="Image" minOccurs="0"/>
  <xs:element ref="GPSPMarker" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="Zone" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="Obstacle" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="AccessPoint" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="ReferenceMarker" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="Exciter" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="LocationFilterRegion" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="LocationFilterRail" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="objectVersion" type="xs:int" use="required"/>
<xs:attribute name="name" type="xs:string"/>
<xs:attribute name="isOutdoor" type="xs:boolean" use="required"/>
<xs:attribute name="floorNumber" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="gpsMarker">
  <xs:sequence>
    <xs:element ref="GeoCoordinate" minOccurs="0"/>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="geoCoordinate">
  <xs:sequence/>
  <xs:attribute name="latitude" type="xs:double" use="required"/>
  <xs:attribute name="longitude" type="xs:double" use="required"/>
  <xs:attribute name="unit" type="ns4:Unit"/>
</xs:complexType>

<xs:complexType name="zone">
  <xs:sequence>
    <xs:element name="ZoneCoordinate" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="obstacle">
  <xs:sequence>
    <xs:element name="ObstacleCoordinate" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="type" type="type"/>
  <xs:attribute name="attenuation" type="xs:float" use="required"/>
</xs:complexType>

<xs:complexType name="accessPoint">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
    <xs:element name="ApInterface" type="apInterface" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
  <xs:attribute name="radioMacAddress" type="xs:string"/>
  <xs:attribute name="ethMacAddress" type="xs:string"/>
  <xs:attribute name="ipAddress" type="xs:string"/>
  <xs:attribute name="numOfSlots" type="xs:int" use="required"/>
  <xs:attribute name="apMode" type="apMode"/>
</xs:complexType>

```

```

<xs:complexType name="apInterface">
  <xs:sequence/>
  <xs:attribute name="band" type="band"/>
  <xs:attribute name="slotNumber" type="xs:int" use="required"/>
  <xs:attribute name="channelAssignment" type="xs:int" use="required"/>
  <xs:attribute name="channelNumber" type="xs:int" use="required"/>
  <xs:attribute name="txPowerLevel" type="xs:int" use="required"/>
  <xs:attribute name="antennaPattern" type="xs:string"/>
  <xs:attribute name="antennaAngle" type="xs:float" use="required"/>
  <xs:attribute name="antennaElevAngle" type="xs:float" use="required"/>
  <xs:attribute name="antennaGain" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="referenceMarker">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="exciter">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
  <xs:attribute name="macAddress" type="xs:string"/>
  <xs:attribute name="range" type="xs:double" use="required"/>
  <xs:attribute name="isPerimeter" type="xs:boolean" use="required"/>
</xs:complexType>

<xs:complexType name="locationFilterRegion">
  <xs:sequence>
    <xs:element ref="MapCoordinate" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="regionType" type="regionType"/>
</xs:complexType>

<xs:complexType name="locationFilterRail">
  <xs:sequence>
    <xs:element name="RailCoordinates" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="RailExclusionCoordinates" type="mapCoordinate" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="railBoundaryWidth" type="xs:float" use="required"/>
</xs:complexType>

<xs:complexType name="movementEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
      <xs:attribute name="moveDistanceInFt" type="xs:float" use="required"/>
      <xs:attribute name="referenceMarkerName" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="presenceEvent">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```



```

<xs:complexType name="streamingNotification">
  <xs:complexContent>
    <xs:extension base="notificationEvent">
      <xs:choice>
        <xs:element ref="location"/>
        <xs:element ref="WirelessClientLocation"/>
        <xs:element ref="TagLocation"/>
        <xs:element ref="InterfererLocation"/>
        <xs:element ref="RogueClientLocation"/>
        <xs:element ref="RogueAPLocation"/>
        <xs:element ref="WiredClientLocation"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="location" abstract="true">
  <xs:sequence>
    <xs:element ref="MapInfo" minOccurs="0"/>
    <xs:element ref="MapCoordinate" minOccurs="0"/>
    <xs:element ref="Statistics" minOccurs="0"/>
    <xs:element ref="GeoCoordinate" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="macAddress" type="xs:string"/>
  <xs:attribute name="currentlyTracked" type="xs:boolean" use="required"/>
  <xs:attribute name="confidenceFactor" type="xs:float" use="required"/>
  <xs:attribute name="historyLogReason" type="historyReason"/>
</xs:complexType>

<xs:complexType name="mapInfo">
  <xs:sequence>
    <xs:element ref="Dimension" minOccurs="0"/>
    <xs:element ref="Image" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="mapHierarchyString" type="xs:string"/>
  <xs:attribute name="floorRefId" type="xs:long" use="required"/>
</xs:complexType>

<xs:complexType name="statistics">
  <xs:sequence/>
  <xs:attribute name="currentServerTime" type="xs:string"/>
  <xs:attribute name="firstLocatedTime" type="xs:string"/>
  <xs:attribute name="lastLocatedTime" type="xs:string"/>
</xs:complexType>

<xs:complexType name="wirelessClientLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence/>
      <xs:attribute name="ipAddress">
        <xs:simpleType>
          <xs:list itemType="xs:string"/>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="userName" type="xs:string"/>
      <xs:attribute name="ssid" type="xs:string"/>
      <xs:attribute name="band" type="band"/>
      <xs:attribute name="apMacAddress" type="xs:string"/>
      <xs:attribute name="isGuestUser" type="xs:boolean" use="required"/>
      <xs:attribute name="dot11Status" type="clientDot11Status"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="tagLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence>
        <xs:element name="TelemetryData" type="telemetryInfo" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element ref="BatteryInfo" minOccurs="0" />
        <xs:element ref="VendorData" minOccurs="0" />
        <xs:element ref="EmergencyInfo" minOccurs="0" maxOccurs="unbounded" />
        <xs:element name="ExciterEncounter" type="deviceExciter" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="lastBeaconTime" type="xs:string" />
      <xs:attribute name="lastBeaconSequenceNumber" type="xs:int" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="telemetryInfo">
  <xs:choice>
    <xs:element ref="TelemetryData" />
    <xs:element ref="EPCTelemetryData" />
    <xs:element ref="GeoTelemetryData" />
    <xs:element ref="MotionTelemetryData" />
  </xs:choice>
  <xs:attribute name="type" type="telemetryType" />
  <xs:attribute name="units" type="xs:string" />
  <xs:attribute name="lastReceivedTime" type="xs:string" />
  <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required" />
</xs:complexType>

<xs:complexType name="telemetryMeasurementData" abstract="true">
  <xs:sequence>
    <xs:element name="dataBlob" type="xs:base64Binary" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="defaultTelemetryMeasurementData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence>
        <xs:element name="measurementData" type="xs:string" minOccurs="0" />
      </xs:sequence>
      <xs:attribute name="measurementDataType" type="xs:string" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="epcTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">
      <xs:sequence>
        <xs:attribute name="header" type="xs:string" />
        <xs:attribute name="manufacturer" type="xs:string" />
        <xs:attribute name="productSKU" type="xs:string" />
        <xs:attribute name="serialNo" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="geoTelemetryData">
  <xs:complexContent>
    <xs:extension base="telemetryMeasurementData">

```

```

        <xs:sequence/>
        <xs:attribute name="latitude" type="xs:string"/>
        <xs:attribute name="longitude" type="xs:string"/>
        <xs:attribute name="altitude" type="xs:string"/>
        <xs:attribute name="altitudeType" type="altitudeType"/>
        <xs:attribute name="datum" type="xs:int" use="required"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="motionTelemetryData">
    <xs:complexContent>
        <xs:extension base="telemetryMeasurementData">
            <xs:sequence/>
            <xs:attribute name="motionType" type="motionType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="batteryInfo">
    <xs:sequence/>
    <xs:attribute name="tolerance" type="xs:int"/>
    <xs:attribute name="percentRemaining" type="xs:int"/>
    <xs:attribute name="daysRemaining" type="xs:int"/>
    <xs:attribute name="batteryAge" type="xs:long"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int"/>
</xs:complexType>

<xs:complexType name="vendorData">
    <xs:sequence/>
    <xs:attribute name="vendorId" type="xs:int" use="required"/>
    <xs:attribute name="elementId" type="xs:short" use="required"/>
    <xs:attribute name="data" type="xs:base64Binary"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
</xs:complexType>

<xs:complexType name="emergencyInfo">
    <xs:sequence/>
    <xs:attribute name="isActive" type="xs:boolean" use="required"/>
    <xs:attribute name="reason" type="xs:string"/>
    <xs:attribute name="tamperState" type="xs:string"/>
    <xs:attribute name="tamperOpenedCount" type="xs:int" use="required"/>
    <xs:attribute name="lastReceivedTime" type="xs:string"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
    <xs:attribute name="fragmentNumber" type="xs:byte" use="required"/>
</xs:complexType>

<xs:complexType name="deviceExciter">
    <xs:sequence/>
    <xs:attribute name="macAddress" type="xs:string"/>
    <xs:attribute name="timestamp" type="xs:string"/>
    <xs:attribute name="inRange" type="xs:boolean" use="required"/>
    <xs:attribute name="isPerimeter" type="xs:boolean" use="required"/>
    <xs:attribute name="lastReceivedSeqNum" type="xs:int" use="required"/>
    <xs:attribute name="fragmentNumber" type="xs:byte" use="required"/>
    <xs:attribute name="isActive" type="xs:boolean" use="required"/>
    <xs:attribute name="assocFloorId" type="xs:long" use="required"/>
    <xs:attribute name="index" type="xs:short" use="required"/>
</xs:complexType>

<xs:complexType name="interfererLocation">
    <xs:complexContent>

```

```

<xs:extension base="location">
  <xs:sequence/>
  <xs:attribute name="deviceType" type="interfererDeviceType"/>
  <xs:attribute name="zoneOfImpact" type="xs:float" use="required"/>
  <xs:attribute name="dutyCycle" type="xs:short" use="required"/>
  <xs:attribute name="severity" type="xs:short" use="required"/>
  <xs:attribute name="deviceKey" type="xs:string"/>
  <xs:attribute name="affectedChannels" type="xs:string"/>
  <xs:attribute name="affectedBand" type="band"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="rogueClientLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence/>
      <xs:attribute name="numOfDetectingAPs" type="xs:int" use="required"/>
      <xs:attribute name="bssid" type="xs:string"/>
      <xs:attribute name="containmentLevel" type="ns5:ContainmentLevel"/>
      <xs:attribute name="state" type="ns5:State"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="rogueAPLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence/>
      <xs:attribute name="totalClients" type="xs:int" use="required"/>
      <xs:attribute name="numOfDetectingAPs" type="xs:int" use="required"/>
      <xs:attribute name="type" type="ns6:Type"/>
      <xs:attribute name="containmentLevel" type="ns6:ContainmentLevel"/>
      <xs:attribute name="state" type="ns6:State"/>
      <xs:attribute name="maxDetectedRssi" type="xs:int" use="required"/>
      <xs:attribute name="ssid" type="xs:string"/>
      <xs:attribute name="maxDetectingAPMacAddress" type="xs:string"/>
      <xs:attribute name="maxRssiRadioType" type="xs:int" use="required"/>
      <xs:attribute name="zoneOfImpact" type="xs:float" use="required"/>
      <xs:attribute name="hiddenSSID" type="xs:int" use="required"/>
      <xs:attribute name="rogueStChBy" type="ns6:RogueState"/>
      <xs:attribute name="rogueClChBy" type="ns6:RogueClass"/>
      <xs:attribute name="classifiedRSSI" type="xs:int" use="required"/>
      <xs:attribute name="classifiedByAP" type="xs:string"/>
      <xs:attribute name="className" type="xs:string"/>
      <xs:attribute name="classification" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="wiredClientLocation">
  <xs:complexContent>
    <xs:extension base="location">
      <xs:sequence>
        <xs:element name="civicAddressAsString" type="xs:string" minOccurs="0"/>
        <xs:element name="PortAssociation" type="wiredPortAssociation" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="username802Dot1X" type="xs:string"/>
      <xs:attribute name="ipAddress">
        <xs:simpleType>
          <xs:list itemType="xs:string"/>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    <xs:attribute name="elin" type="xs:string"/>
    <xs:attribute name="serialNumber" type="xs:string"/>
    <xs:attribute name="udi" type="xs:string"/>
    <xs:attribute name="vlanId" type="xs:int" use="required"/>
    <xs:attribute name="vlanName" type="xs:string"/>
    <xs:attribute name="switchIpAddress" type="xs:int" use="required"/>
    <xs:attribute name="portNumber" type="xs:short" use="required"/>
    <xs:attribute name="slotNumber" type="xs:short" use="required"/>
    <xs:attribute name="state" type="associationState"/>
    <xs:attribute name="portType" type="xs:byte" use="required"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="wiredPortAssociation">
  <xs:complexContent>
    <xs:extension base="aesObject">
      <xs:sequence>
        <xs:element name="duplicate" type="xs:boolean"/>
        <xs:element name="moduleNumber" type="xs:short"/>
        <xs:element name="portId" type="xs:long"/>
        <xs:element name="portNumber" type="xs:short"/>
        <xs:element name="portType" type="xs:byte"/>
        <xs:element name="slotNumber" type="xs:short"/>
        <xs:element name="state" type="xs:byte"/>
        <xs:element name="switchIpAddress" type="xs:string" minOccurs="0"/>
        <xs:element name="transitionTime" type="xs:long"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="aesObject" abstract="true">
  <xs:sequence>
    <xs:element name="changedOn" type="xs:long"/>
    <xs:element name="mask" type="xs:int"/>
    <xs:element name="objectId" type="xs:long"/>
    <xs:element name="options" type="xs:short"/>
    <xs:element name="parentId" type="xs:long"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="containmentBoundary">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INSIDE"/>
    <xs:enumeration value="OUTSIDE"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="areaType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CAMPUS"/>
    <xs:enumeration value="BUILDING"/>
    <xs:enumeration value="FLOOR"/>
    <xs:enumeration value="ZONE"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="eventEntity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="WIRELESS_CLIENTS"/>
    <xs:enumeration value="TAGS"/>
    <xs:enumeration value="ROGUE_CLIENTS"/>
    <xs:enumeration value="ROGUE_APS"/>
  </xs:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="INTERFERERS" />
    <xs:enumeration value="INTERFERER_DEVICE_TYPE" />
    <xs:enumeration value="MAP_CHANGE" />
    <xs:enumeration value="MAC_ADDRESS" />
    <xs:enumeration value="ASSET_NAME" />
    <xs:enumeration value="ASSET_CATEGORY" />
    <xs:enumeration value="ASSET_GROUP" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="batteryStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ANY" />
    <xs:enumeration value="LOW" />
    <xs:enumeration value="MEDIUM" />
    <xs:enumeration value="NORMAL" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="changeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ADDED" />
    <xs:enumeration value="MODIFIED" />
    <xs:enumeration value="DELETED" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="THICK_WALL" />
    <xs:enumeration value="NORMAL_WALL" />
    <xs:enumeration value="THICK_DOOR" />
    <xs:enumeration value="NORMAL_DOOR" />
    <xs:enumeration value="CUBICLE" />
    <xs:enumeration value="GLASS" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="apMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LOCAL" />
    <xs:enumeration value="MONITOR" />
    <xs:enumeration value="REAP" />
    <xs:enumeration value="ROGUE_DETECTOR" />
    <xs:enumeration value="SNIFFER" />
    <xs:enumeration value="BRIDGE" />
    <xs:enumeration value="UNKNOWN" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="band">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="IEEE_802_11_A" />
    <xs:enumeration value="IEEE_802_11_B" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="regionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INSIDE" />
    <xs:enumeration value="OUTSIDE" />
  </xs:restriction>

```

```
</xs:simpleType>

<xs:simpleType name="historyReason">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="FLOOR_CHANGE" />
    <xs:enumeration value="CHOKEPOINT_ENCOUNTER" />
    <xs:enumeration value="INOUT_OF_AREA" />
    <xs:enumeration value="SIGNIFICANT_MOVEMENT" />
    <xs:enumeration value="LOCP_MESSAGE" />
    <xs:enumeration value="INFORMATION_CHANGE" />
    <xs:enumeration value="DEVICE_TYPE_CHANGE" />
    <xs:enumeration value="TAG_EMERGENCY" />
    <xs:enumeration value="LOCATION_CONTRIBUTOR_CHANGE" />
    <xs:enumeration value="INTERFERER_MERGE" />
    <xs:enumeration value="INTERFERER_CLUSTER_CENTRE_CHANGE" />
    <xs:enumeration value="INTERFERER_SEVERITY_CHANGE" />
    <xs:enumeration value="STATUS_CHANGE" />
    <xs:enumeration value="SECURITY_POLICY_CHANGE" />
    <xs:enumeration value="NETWORK_STATUS_CHANGE" />
    <xs:enumeration value="HASHKEY_CHANGE" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="clientDot11Status">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="ASSOCIATED" />
    <xs:enumeration value="PROBING" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="telemetryType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="GPS" />
    <xs:enumeration value="TEMPERATURE" />
    <xs:enumeration value="MOTION" />
    <xs:enumeration value="DISTANCE" />
    <xs:enumeration value="FUEL" />
    <xs:enumeration value="QUANTITY" />
    <xs:enumeration value="PRESSURE" />
    <xs:enumeration value="HUMIDITY" />
    <xs:enumeration value="STATUS" />
    <xs:enumeration value="EPC" />
    <xs:enumeration value="MOTIONPROB" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="altitudeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="AT_METERS" />
    <xs:enumeration value="AT_FLOOR" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="motionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="NO_MOVEMENT" />
    <xs:enumeration value="ACCELERATION" />
    <xs:enumeration value="MOVEMENT" />
    <xs:enumeration value="DECELERATION" />
  </xs:restriction>
</xs:simpleType>
```

```

    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="interfererDeviceType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="BLUETOOTH"/>
      <xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_1"/>
      <xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_1"/>
      <xs:enumeration value="TDD_1_PHONE"/>
      <xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_2"/>
      <xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_2"/>
      <xs:enumeration value="MICROWAVE_OVEN_1"/>
      <xs:enumeration value="DEVICE_802_11_FH"/>
      <xs:enumeration value="DECT_LIKE_NETWORK_VARIANT_3"/>
      <xs:enumeration value="DECT_LIKE_BASE_STATION_VARIANT_3"/>
      <xs:enumeration value="MICROWAVE_OVEN_2"/>
      <xs:enumeration value="BLUETOOTH_PAGING_INQUIRY_DEVICE"/>
      <xs:enumeration value="GENERIC_TDD"/>
      <xs:enumeration value="JAMMER"/>
      <xs:enumeration value="RFID_READER"/>
      <xs:enumeration value="CLASS_1_NETWORK"/>
      <xs:enumeration value="CLASS_2_NETWORK"/>
      <xs:enumeration value="GENERIC_WIDEBAND"/>
      <xs:enumeration value="BRIDGE_ROUTER_1"/>
      <xs:enumeration value="GENERIC_WAVEFORM"/>
      <xs:enumeration value="GENERIC_DECT_LIKE"/>
      <xs:enumeration value="VIDEO"/>
      <xs:enumeration value="DEVICE_802_15_4"/>
      <xs:enumeration value="DEVICE_802_11_NORMAL"/>
      <xs:enumeration value="DEVICE_802_11_SPECTRALLY_INVERTED"/>
      <xs:enumeration value="DEVICE_802_11_NONSTD_CHANNEL"/>
      <xs:enumeration value="DEVICE_802_11_SUPERG"/>
      <xs:enumeration value="RADAR"/>
      <xs:enumeration value="MOTOROLA_CANOPY"/>
      <xs:enumeration value="XBOX"/>
      <xs:enumeration value="WIFI_MOBILE"/>
      <xs:enumeration value="WIFI_FIXED"/>
      <xs:enumeration value="EXALT"/>
      <xs:enumeration value="CONTINUOUS_TRANSMITTER"/>
      <xs:enumeration value="DECT_LIKE_NETWORK"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="associationState">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CONNECTED"/>
      <xs:enumeration value="DISCONNECTED"/>
      <xs:enumeration value="UNKNOWN"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

ContainmentEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",

```



```
    "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
  },
  "deviceId" : {
    "type" : "string"
  },
  "locationMapHierarchy" : {
    "type" : "string"
  },
  "locationCoordinate" : {
    "type" : "object",
    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  },
  "boundary" : {
    "type" : "string",
    "enum" : [ "INSIDE", "OUTSIDE" ]
  },
  "areaType" : {
    "type" : "string",
    "enum" : [ "CAMPUS", "BUILDING", "FLOOR", "ZONE" ]
  },
  "containerHierarchy" : {
    "type" : "string"
  },
  "timestamp" : {
    "type" : "string"
  }
}
}
```

containment-event.proto

```
option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "ContainmentEventProtos";
import "event-entity.proto";
import "map-coordinate.proto";

message PBContainmentEvent {

  enum PBContainmentBoundary {
    INSIDE = 0;
    OUTSIDE = 1;
  }

  enum PBAreaType {
    CAMPUS = 0;
    BUILDING = 1;
    FLOOR = 2;
    ZONE = 3;
  }
}
```

```

    required string subscriptionName = 1;

    optional uint64 timeStampInMillis = 2;

    optional PBEventEntity eventEntity = 3;

    optional string deviceId = 4;

    optional string locationMapHierarchy = 5;

    optional PMapCoordinate locationCoordinate = 6;

    optional PBContainmentBoundary boundary = 7;

    optional PBAreaType areaType = 8;

    optional string containerHierarchy = 9;
}

```

MapInfoChangeEvent

MapInfoChangeEvent.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="GeoCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DEGREES"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

MapInfoChangeEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        }
      }
    },

```

```

    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"changeType" : {
  "type" : "string",
  "enum" : [ "ADDED", "MODIFIED", "DELETED" ]
},
"changedCampus" : {
  "type" : "object",
  "properties" : {
    "objectVersion" : {
      "type" : "integer"
    },
    "name" : {
      "type" : "string"
    },
    "dimension" : {
      "type" : "object",
      "properties" : {
        "length" : {
          "type" : "number"
        },
        "width" : {
          "type" : "number"
        },
        "height" : {
          "type" : "number"
        },
        "offsetX" : {
          "type" : "number"
        },
        "offsetY" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    }
  },
"image" : {
  "type" : "object",
  "properties" : {
    "imageName" : {
      "type" : "string"
    },
    "sourceFile" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  }
},
"buildingList" : {
  "type" : "array",
  "items" : {
    "type" : "object",

```

```

"properties" : {
  "objectVersion" : {
    "type" : "integer"
  },
  "name" : {
    "type" : "string"
  },
  "dimension" : {
    "type" : "object",
    "properties" : {
      "length" : {
        "type" : "number"
      },
      "width" : {
        "type" : "number"
      },
      "height" : {
        "type" : "number"
      },
      "offsetX" : {
        "type" : "number"
      },
      "offsetY" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  },
  "image" : {
    "type" : "object",
    "properties" : {
      "imageName" : {
        "type" : "string"
      },
      "sourceFile" : {
        "type" : "array",
        "items" : {
          "type" : "string"
        }
      }
    }
  },
  "floorList" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "objectVersion" : {
          "type" : "integer"
        },
        "name" : {
          "type" : "string"
        },
        "dimension" : {
          "type" : "object",
          "properties" : {
            "length" : {
              "type" : "number"
            },
            "width" : {
              "type" : "number"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "height" : {
      "type" : "number"
    },
    "offsetX" : {
      "type" : "number"
    },
    "offsetY" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  },
  "floorNumber" : {
    "type" : "integer"
  },
  "image" : {
    "type" : "object",
    "properties" : {
      "imageName" : {
        "type" : "string"
      },
      "sourceFile" : {
        "type" : "array",
        "items" : {
          "type" : "string"
        }
      }
    }
  },
  "gpsMarkers" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "name" : {
          "type" : "string"
        },
        "geoCoordinate" : {
          "type" : "object",
          "properties" : {
            "latitude" : {
              "type" : "number"
            },
            "longitude" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "DEGREES" ]
            }
          }
        }
      }
    }
  },
  "mapCoordinate" : {
    "type" : "object",
    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      }
    }
  }
}

```

```

    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
}
},
"zones" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "zoneCoordinate" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    }
  }
},
"obstacles" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "type" : {
        "type" : "string",
        "enum" : [ "UNKNOWN", "THICK_WALL", "NORMAL_WALL",
"THICK_DOOR", "NORMAL_DOOR", "CUBICLE", "GLASS" ]
      },
      "attenuation" : {
        "type" : "number"
      },
      "obstacleCoordinates" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {

```

```

        "type" : "string",
        "enum" : [ "FEET", "METER" ]
    }
}
}
}
},
"accessPoints" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "name" : {
                "type" : "string"
            },
            "radioMacAddress" : {
                "type" : "string"
            },
            "ethMacAddress" : {
                "type" : "string"
            },
            "ipAddress" : {
                "type" : "string"
            },
            "numOfSlots" : {
                "type" : "integer"
            },
            "mapCoordinates" : {
                "type" : "object",
                "properties" : {
                    "x" : {
                        "type" : "number"
                    },
                    "y" : {
                        "type" : "number"
                    },
                    "unit" : {
                        "type" : "string",
                        "enum" : [ "FEET", "METER" ]
                    }
                }
            }
        },
        "enum" : [ "LOCAL", "MONITOR", "REAP", "ROGUE_DETECTOR",
"SNIFFER", "BRIDGE", "UNKNOWN" ]
    },
    "apInterfaces" : {
        "type" : "array",
        "items" : {
            "type" : "object",
            "properties" : {
                "band" : {
                    "type" : "string",
                    "enum" : [ "UNKNOWN", "IEEE_802_11_A", "IEEE_802_11_B" ]
                },
                "slotNumber" : {
                    "type" : "integer"
                },
                "channelAssignment" : {
                    "type" : "integer"
                }
            }
        },
    },

```

```

        "channelNumber" : {
            "type" : "integer"
        },
        "txPowerLevel" : {
            "type" : "integer"
        },
        "antennaPattern" : {
            "type" : "string"
        },
        "antennaAngle" : {
            "type" : "number"
        },
        "antennaElevAngle" : {
            "type" : "number"
        },
        "antennaGain" : {
            "type" : "integer"
        }
    }
}
}
},
"referenceMarkers" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "name" : {
                "type" : "string"
            },
            "mapCoordinates" : {
                "type" : "object",
                "properties" : {
                    "x" : {
                        "type" : "number"
                    },
                    "y" : {
                        "type" : "number"
                    },
                    "unit" : {
                        "type" : "string",
                        "enum" : [ "FEET", "METER" ]
                    }
                }
            }
        }
    }
},
"exciters" : {
    "type" : "array",
    "items" : {
        "type" : "object",
        "properties" : {
            "name" : {
                "type" : "string"
            },
            "macAddress" : {
                "type" : "string"
            },
            "range" : {
                "type" : "number"
            }
        }
    }
},

```



```

    "mapCoordinates" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "perimeter" : {
      "type" : "boolean",
      "required" : true
    }
  }
},
"filterRegions" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "regionType" : {
        "type" : "string",
        "enum" : [ "INSIDE", "OUTSIDE" ]
      },
      "regionCoordinates" : {
        "type" : "array",
        "items" : {
          "type" : "object",
          "properties" : {
            "x" : {
              "type" : "number"
            },
            "y" : {
              "type" : "number"
            },
            "unit" : {
              "type" : "string",
              "enum" : [ "FEET", "METER" ]
            }
          }
        }
      }
    }
  }
},
"filterRails" : {
  "type" : "array",
  "items" : {
    "type" : "object",
    "properties" : {
      "railBoundaryWidth" : {
        "type" : "number"
      },
      "railCoordinates" : {
        "type" : "array",
        "items" : {
          "type" : "object",

```



```

enum PBChangeType {
    ADDED = 0;
    MODIFIED = 1;
    DELETED = 2;
}

required string subscriptionName = 1;

optional uint64 timeStampInMillis = 2;

optional PBEventEntity eventEntity = 3;

optional string deviceId = 4;

optional string locationMapHierarchy = 5;

optional PBCoordinate locationCoordinate = 6;

optional PBChangeType changeType = 7;

optional PBCampus campus = 8;
}

```

BatteryEvent

BatteryEvent.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FEET"/>
      <xs:enumeration value="METER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

BatteryEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
  },
}

```

```

    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "batteryStatus" : {
      "type" : "string",
      "enum" : [ "ANY", "LOW", "MEDIUM", "NORMAL" ]
    },
    "percentRemaining" : {
      "type" : "integer"
    },
    "timestamp" : {
      "type" : "string"
    }
  }
}

```

battery-event.proto

```

option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "BatteryEventProtos";

import "event-entity.proto";
import "map-coordinate.proto";

message PBBatteryEvent {

  enum PBBatteryStatus {
    ANY = 0;
    LOW = 1;
    MEDIUM = 2;
    NORMAL = 3;
  }

  required string subscriptionName = 1;

  optional uint64 timeStampInMillis = 2;

  optional PBEventEntity eventEntity = 3;

  optional string deviceId = 4;

  optional string locationMapHierarchy = 5;

  optional PMapCoordinate locationCoordinate = 6;

  optional PBBatteryStatus batteryStatus = 7;

  optional int32 percentRemaining = 8;
}

```

EmergencyEvent

EmergencyEvent.xml

```
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="GeoCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DEGREES"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

EmergencyEvent.json

```
{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "type" : {
      "type" : "string",
      "enum" : [ "NONE", "ANY", "UNKNOWN", "PANIC_BUTTON", "DETACHED", "TAMPERING" ]
    },
    "vendorId" : {
      "type" : "integer"
    },
    "vendorData" : {
      "type" : "array",
      "items" : {
        "type" : "string"
      }
    }
  },
}
```

```

    "timestamp" : {
      "type" : "string"
    }
  }
}

```

EmergencyEvent.proto

```

option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "EmergencyEventProtos";

import "event-entity.proto";
import "map-coordinate.proto";

message PBEmergencyEvent {

  enum PBType {
    NONE = 0;
    ANY = 1;
    UNKNOWN = 2;
    PANIC_BUTTON = 3;
    DETACHED = 4;
    TAMPERING = 5;
  }

  required string subscriptionName = 1;

  optional uint64 timeStampInMillis = 2;

  optional PBEventEntity eventEntity = 3;

  optional string deviceId = 4;

  optional string locationMapHierarchy = 5;

  optional PMapCoordinate locationCoordinate = 6;

  optional PBType type = 7;

  optional int32 vendorId = 8;

  optional bytes vendorData = 9;
}

```

ExciterEvent

ExciterEvent.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="GeoCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Unit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DEGREES"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

ExciterEvent.json

```
{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        },
        "unit" : {
          "type" : "string",
          "enum" : [ "FEET", "METER" ]
        }
      }
    },
    "triggerExciterMac" : {
      "type" : "string"
    },
    "triggerExciterName" : {
      "type" : "string"
    },
    "timestamp" : {
      "type" : "string"
    }
  }
}
```

ExciterEvent.proto

```
option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "ExciterEventProtos";

import "event-entity.proto";
import "map-coordinate.proto";

message PBExciterEvent {

    required string subscriptionName = 1;

    optional uint64 timeStampInMillis = 2;

    optional PBEventEntity eventEntity = 3;
```

```

optional string deviceId = 4;

optional string locationMapHierarchy = 5;

optional PMapCoordinate locationCoordinate = 6;

optional string triggerExciterMac = 7;

optional string triggerExciterName = 8;

}

```

StreamingNotification

StreamingNotification.xml

```

{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="EmergencyEvent"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="Type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="ANY"/>
      <xs:enumeration value="UNKNOWN"/>
      <xs:enumeration value="PANIC_BUTTON"/>
      <xs:enumeration value="DETACHED"/>
      <xs:enumeration value="TAMPERING"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

StreamingNotification.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    },
    "locationCoordinate" : {
      "type" : "object",
      "properties" : {
        "x" : {
          "type" : "number"
        },
        "y" : {
          "type" : "number"
        }
      }
    }
  }
}

```



```

    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  },
  "location" : {
    "type" : "object",
    "properties" : {
      "macAddress" : {
        "type" : "string"
      },
      "mapInfo" : {
        "type" : "object",
        "properties" : {
          "mapHierarchyString" : {
            "type" : "string"
          },
          "floorRefId" : {
            "type" : "number"
          },
          "floorDimension" : {
            "type" : "object",
            "properties" : {
              "length" : {
                "type" : "number"
              },
              "width" : {
                "type" : "number"
              },
              "height" : {
                "type" : "number"
              },
              "offsetX" : {
                "type" : "number"
              },
              "offsetY" : {
                "type" : "number"
              },
              "unit" : {
                "type" : "string",
                "enum" : [ "FEET", "METER" ]
              }
            }
          }
        }
      },
      "image" : {
        "type" : "object",
        "properties" : {
          "imageName" : {
            "type" : "string"
          },
          "sourceFile" : {
            "type" : "array",
            "items" : {
              "type" : "string"
            }
          }
        }
      }
    }
  },
  "mapCoordinate" : {
    "type" : "object",

```

```

    "properties" : {
      "x" : {
        "type" : "number"
      },
      "y" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "FEET", "METER" ]
      }
    }
  },
  "currentlyTracked" : {
    "type" : "boolean",
    "required" : true
  },
  "confidenceFactor" : {
    "type" : "number"
  },
  "statistics" : {
    "type" : "object",
    "properties" : {
      "currentServerTime" : {
        "type" : "string"
      },
      "firstLocatedTime" : {
        "type" : "string"
      },
      "lastLocatedTime" : {
        "type" : "string"
      }
    }
  },
  "historyLogReason" : {
    "type" : "string",
    "enum" : [ "UNKNOWN", "FLOOR_CHANGE", "CHOKEPOINT_ENCOUNTER", "INOUT_OF_AREA",
"SIGNIFICANT_MOVEMENT", "LOCP_MESSAGE", "INFORMATION_CHANGE", "DEVICE_TYPE_CHANGE",
"TAG_EMERGENCY", "LOCATION_CONTRIBUTOR_CHANGE", "INTERFERER_MERGE",
"INTERFERER_CLUSTER_CENTRE_CHANGE", "INTERFERER_SEVERITY_CHANGE", "STATUS_CHANGE",
"SECURITY_POLICY_CHANGE", "NETWORK_STATUS_CHANGE", "HASHKEY_CHANGE" ]
  },
  "geoCoordinate" : {
    "type" : "object",
    "properties" : {
      "latitude" : {
        "type" : "number"
      },
      "longitude" : {
        "type" : "number"
      },
      "unit" : {
        "type" : "string",
        "enum" : [ "DEGREES" ]
      }
    }
  }
},
"timestamp" : {
  "type" : "string"
}
}
}

```

StreamingNotification.proto

```
option java_package = "com.cisco.mse.cas.notification.events";

option java_outer_classname = "StreamingNotificationProtos";

import "event-entity.proto";
import "wireless-client.proto";
import "tag.proto";
import "rogue-ap.proto";
import "rogue-client.proto";
import "interferer.proto";

message PBStreamingNotification {

    required string subscriptionName = 1;

    optional uint64 timeStampInMillis = 2;

    optional PBEventEntity eventEntity = 3;

    optional string deviceId = 4;

    optional PBWirelessClientLocation clientLocation = 5;

    optional PBTagLocation tagLocation = 6;

    optional PBRogueClientLocation rogueClientLocation = 7;

    optional PBRogueAPLocation rogueAPLocation = 8;

    optional PBInterfererLocation interfererLocation = 9;

}
```

TagRSSINotification**tag-rssi-proto**

```
{
option java_package = "com.cisco.mse.cas.notification.rssi";
option java_outer_classname = "TagRSSINotificationProtos";

import "ap-rssi.proto";

message PBTagRSSINotification {

    message PBCCXTagPayload {

        optional int32 timestamp = 1;

        optional int32 fragmentNumber = 2;

        optional int32 sequenceNumber = 3;

        repeated bytes data = 4;

    }

    required string macaddress = 1;

    optional int32 transmitpower = 2;

    repeated PBAPRSSIMeasurement apRssiMeasurements = 3;

}
```

```

    repeated PBCCXTagPayload ccxTagPayloadList = 4;
}

```

ap-rssi.proto

```

option java_package = "com.cisco.mse.cas.notification.rssi";
option java_outer_classname = "ApRssiProtos";
message PBAPRSSIMeasurement {

    required string apMacAddress = 1;

    required int32 ifSlotId = 2;

    required int32 bandId = 3;

    required int32 antennaId = 4;

    required int32 rssi = 5;

    optional int32 timeStamp = 6;
}

```

AssociationEvent

AssociationEvent.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="MapCoordinate"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:simpleType name="Unit">
        <xs:restriction base="xs:string">
            <xs:enumeration value="FEET"/>
            <xs:enumeration value="METER"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>

```

AssociationEvent.json

```

{
  "type" : "object",
  "properties" : {
    "subscriptionName" : {
      "type" : "string"
    },
    "entity" : {
      "type" : "string",
      "enum" : [ "WIRELESS_CLIENTS", "TAGS", "ROGUE_CLIENTS", "ROGUE_APS", "INTERFERERS",
"INTERFERER_DEVICE_TYPE", "MAP_CHANGE", "MAC_ADDRESS", "ASSET_NAME", "ASSET_CATEGORY",
"ASSET_GROUP" ]
    },
    "deviceId" : {
      "type" : "string"
    },
    "locationMapHierarchy" : {
      "type" : "string"
    }
  },
}

```

```
"locationCoordinate" : {
  "type" : "object",
  "properties" : {
    "x" : {
      "type" : "number"
    },
    "y" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "FEET", "METER" ]
    }
  }
},
"mseUdi" : {
  "type" : "string"
},
"geoCoordinate" : {
  "type" : "object",
  "properties" : {
    "latitude" : {
      "type" : "number"
    },
    "longitude" : {
      "type" : "number"
    },
    "unit" : {
      "type" : "string",
      "enum" : [ "DEGREES" ]
    }
  }
},
"floorRefId" : {
  "type" : "number"
},
"association" : {
  "type" : "boolean",
  "required" : true
},
"ipAddress" : {
  "type" : "array",
  "items" : {
    "type" : "string"
  }
},
"apMacAddress" : {
  "type" : "string"
},
"status" : {
  "type" : "integer"
},
"username" : {
  "type" : "string"
},
"timestamp" : {
  "type" : "string"
}
}
```

AssociationEvent.proto



Sample Examples

This chapter contains the following sample examples:

- [Sample PresenceEventTrigger Notification Subscription, page 6-1](#)
- [Sample MovementEventTrigger Notification Subscription, page 6-2](#)
- [Sample Location Streaming NotificationTrigger Subscription, page 6-2](#)

Sample PresenceEventTrigger Notification Subscription

The following is an example of Notification subscription to subscribe for PresenceEvent for Wireless clients in JSON format over HTTP.

```
{
  "NotificationSubscription": {
    "name": "presence-event-trigger",
    "notificationType": "EVENT_DRIVEN",
    "dataFormat": "JSON",
    "NotificationReceiverInfo": {
      "transport": {
        "type": "TransportHttp",
        "hostAddress": "1.2.3.4",
        "port": "5555"
      }
    },
    "subscribedEvents": [
      {
        "type": "PresenceEventTrigger",
        "existingPresence": "true",
        "eventEntity": "WIRELESS_CLIENTS"
      }
    ]
  }
}
```

Sample MovementEventTrigger Notification Subscription

The following is an example of NotificationSubscription to subscribe for MovementEvent for Wireless clients in XML format over HTTP.

```
{
  "NotificationSubscription": {
    "name": "movement-event-trigger",
    "notificationType": "EVENT_DRIVEN",
    "dataFormat": "XML",
    "NotificationReceiverInfo": {
      "transport": {
        "type": "TransportHttp",
        "hostAddress": "1.2.3.4",
        "port": "6666"
      }
    },
    "subscribedEvents": [
      {
        "type": "MovementEventTrigger",
        "moveDistanceInFt": "5",
        "eventEntity": "WIRELESS_CLIENTS"
      }
    ]
  }
}
```

Sample Location Streaming NotificationTrigger Subscription

The following is an example of NotificationSubscription to subscribe for Location stream of Wireless clients in Protocol Buffer format over TCP.

```
{
  "NotificationSubscription": {
    "name": "client-location-stream",
    "notificationType": "STREAMING",
    "dataFormat": "PROTOCOL_BUFFERS",
    "NotificationReceiverInfo": {
      "transport": {
        "type": "TransportHttp",
        "hostAddress": "1.2.3.4",
        "port": "7777"
      }
    },
  },
}
```



```
"subscribedEvents" : [  
  {  
    "type" : "streamingNotificationTrigger",  
    "streamType" : "RAW_LOCATION",  
    "eventEntity": "WIRELESS_CLIENTS"  
  }  
]  
}
```

