|  |  |  |
| --- | --- | --- |
| Cisco-logo-picture | Document Number | EDCS-1292424 |
| Based on Template | [EDCS-228142 Rev 1](http://wwwin-eng.cisco.com/cgi-bin/edcs/edcs_info?EDCS-228142)7 |
| Created By | Harvey Delery, Amit Hako, Ethan Hoewisch, Ramesh Singh |

CMX Mobile Application and SDK  
Software Design Specification

Reviewers

|  |  |
| --- | --- |
| Department | Name/Title |
| Development Engineering | Harvey Delery, Pritam Shah |
| Development Test Engineering | Jayshree Patel, Savitha Amarnath |
| Product Marketing | Jagdish Girimaji |
| TME | Sujit Ghosh, Darryl Sladden |

Modification History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Originator | Comments |
| 1 | 7/29/2013 | Harvey Delery  Amit Hako  Ethan Hoewisch  Ramesh Singh | Initial Version |
| 2 | 9/26/2013 | Harvey Delery  Amit Hako  Ethan Hoewisch  Ramesh Singh | Review Comments |
| 4 | 10/3/2013 | Ethan Hoewisch | Added some location challenge details |
| 5 | 10/07/2013 | Harvey Delery | Added Approver List |
| 6 | 10/30/2013 | Ethan Hoewisch | Added list of security measures/recommendations |
| 7 | 2/25/2014 | Harvey Delery | Updated with the latest Client REST APIs |

Table of Contents

[1 Problem Definition 3](#_Toc381118047)

[2 Design Considerations 4](#_Toc381118048)

[2.1 Third Party Relationships 4](#_Toc381118049)

[2.2 Security Considerations 4](#_Toc381118050)

[2.2.1 Security measures 4](#_Toc381118051)

[2.2.2 Caveats for the Location Challenge security measures 6](#_Toc381118052)

[3 Functional Structure 7](#_Toc381118053)

[3.1.1 CMX Browser Engage Interface 7](#_Toc381118054)

[3.1.2 Location, Browser Engage and CMX Cloud Connector 7](#_Toc381118055)

[3.1.3 CMX Cloud Server 7](#_Toc381118056)

[3.1.4 Google Cloud Messaging/Apple Push Notification Services 8](#_Toc381118057)

[3.1.5 CMX Mobile Application and SDK 8](#_Toc381118058)

[3.2 CMX Mobile Application Architecture 9](#_Toc381118059)

[3.2.1 Location, Browser Engage and CMX Cloud Connector 9](#_Toc381118060)

[3.2.2 CMX Cloud Server 9](#_Toc381118061)

[3.2.3 CMX Mobile Application and SDK 10](#_Toc381118062)

[4 System Flow 11](#_Toc381118063)

[4.1 Browser Engage To CMX Cloud Registration 11](#_Toc381118064)

[4.2 CMX Admin Venue Push 12](#_Toc381118065)

[4.3 CMX Mobile Application Initial Launch 13](#_Toc381118066)

[4.4 CMX Mobile Application iOS 7 Registration 14](#_Toc381118067)

[4.5 CMX Mobile Push Notification 15](#_Toc381118068)

[4.6 CMX Mobile Application Location Tracking 16](#_Toc381118069)

[4.7 Location Challenge 17](#_Toc381118070)

[5 Data Structures 17](#_Toc381118071)

[6 Description of Algorithms 17](#_Toc381118072)

[7 Interface Design 17](#_Toc381118073)

[7.1 CMX Cloud Server API for CMX Mobile Application 17](#_Toc381118074)

[7.1.1 /api/cmxmobile/v1/clients/register 17](#_Toc381118075)

[7.1.2 /api/cmxmobile/v1/clients/location/{deviceId} 18](#_Toc381118076)

[7.1.3 /api/cmxmobile/v1/venues/info/ 18](#_Toc381118077)

[7.1.4 /api/cmxmobile/v1/venues/info/{venueId} 19](#_Toc381118078)

[7.1.5 /api/cmxmobile/v1/venues/image/{venueId} 20](#_Toc381118079)

[7.1.6 /api/cmxmobile/v1/maps/info/{venueId} 20](#_Toc381118080)

[7.1.7 /api/cmxmobile/v1/maps/info/{venueId}/{floorId} 22](#_Toc381118081)

[7.1.8 /api/cmxmobile/v1/maps/image/{venueId}/{floorId} 22](#_Toc381118082)

[7.1.9 /api/cmxmobile/v1/pois/info/{venueId} 23](#_Toc381118083)

[7.1.10 /api/cmxmobile/v1/pois/info/{venueId}/{floorId} 24](#_Toc381118084)

[7.1.11 /api/cmxmobile/v1/pois/image/{venueId}/{poiId} 25](#_Toc381118085)

[7.1.12 /api/cmxmobile/v1/pois/{venueId}?search={keyWord} 25](#_Toc381118086)

[7.1.13 /api/cmxmobile/v1/routes/clients/{deviceId}?destpoi={destPoi} 26](#_Toc381118087)

[7.1.14 /api/cmxmobile/v1/banners/info/{venueId}/{floorId}/{zoneId} 27](#_Toc381118088)

[7.1.15 /api/cmxmobile/v1/banners/image/{venueId}/{floorId}/{zoneId}/{imageId} 28](#_Toc381118089)

[7.2 CMX Cloud Server API for external use 28](#_Toc381118090)

[8 End User Interface 28](#_Toc381118091)

[8.1 Prime Infrastructure 28](#_Toc381118092)

[8.2 Browser Engage 29](#_Toc381118093)

[8.2.1 Mobile Push Notification Message 29](#_Toc381118094)

[8.2.2 CMX Cloud Server 30](#_Toc381118095)

[9 Software Restrictions and Considerations 31](#_Toc381118096)

[10 Hardware Restrictions and Considerations 31](#_Toc381118097)

[11 External Restrictions and Configuration 31](#_Toc381118098)

[12 Development Testing 32](#_Toc381118099)

[12.1 Unit Testing 32](#_Toc381118100)

[12.1.1 CMX Cloud Server 32](#_Toc381118101)

[12.1.2 CMX Cloud Connector 32](#_Toc381118102)

[12.1.3 CMX Mobile Application SDK 32](#_Toc381118103)

[12.2 Integration Testing 33](#_Toc381118104)

[12.2.1 Browser Engage 33](#_Toc381118105)

[12.2.2 Location 33](#_Toc381118106)

[12.2.3 CMX Cloud Connector 33](#_Toc381118107)

[12.2.4 CMX Cloud Server 33](#_Toc381118108)

[12.2.5 CMX Mobile Application SDK 33](#_Toc381118109)

[13 Patentability Considerations 33](#_Toc381118110)

[1 References 34](#_Toc381118111)

[2 Glossary 34](#_Toc381118112)

[3 Future Consideration 34](#_Toc381118113)

[4 Review Action Items 34](#_Toc381118114)

# Problem Definition

CMX Mobile Application can be considered an extension of Browser Engage for mobile applications. Browser Engage is a key component in Mobile Concierge that enables various location based services and context aware messaging on behalf of Cisco customers to their end-users. CMX Mobile Application will allow mobile applications to have similar and additional features that Browser Engage provides for the web browser. The application will receive notifications upon entry into a venue which can allow the application to be launched. Additionally, the application will not require the device to connect to the local network. A piece of the architecture will reside in the cloud allowing the mobile device’s cellular connection to be used or the customer can host the cloud server in a local environment.

# Design Considerations

Allow the consumer to receive context-aware content without joining a venue’s WiFi network if desired. Suggesting or supplying a method to join a desired wifi network will also be provided. The CMX Cloud Server component of this architecture serves as a link between the venue’s MSE and the consumer’s device. The CMX Cloud Server should prohibit any consumer (X) from viewing the location information about any other consumer (C) without C’s consent. The CMX Cloud Server and CMX Cloud Server SDK will provide delegation hooks to determine if C has granted X permission to see C’s location. For example, this lets a Facebook connector let two friends see each other’s location.

## Third Party Relationships

Use open source code when available. Otherwise develop new code.

## Security Considerations

Goal: provide secure communication between MSE, CMX Cloud Server, and consumer devices.

Design consideration: The CMX Cloud Server will use a certificate to bootstrap TLS connections between CMX Cloud Server and MSE, and consumer device and CMX Cloud Server. The CMX Mobile SDK must reject the connection to the CMX Cloud Server if the certificate provided by the CMX Cloud Server is invalid. The MSE must reject the connection to the CMX Cloud Server if the certificate provided by the CMX Cloud Server is invalid.

Goal: Enhance consumer privacy by preventing the CMX Cloud Server from storing raw MAC addresses.

Design consideration: Devices will not use MAC Addresses to query for location. Instead they will be given a device key and always query for the location based upon the key.

### Security measures

Here's the roundup of security measures John Bashinski and I came up with.

We'll still have to do something to make sure that sensitive customer data

(like the email address or phone number of a venue owner) is kept safe,

but that's a bit more standard to do than the location data.

(Recap: Tom is the CMX App Server, Sally (or Sam or /^S.+/) is a customer

with an MSE using the CMX App Server, Bob is a user who hasn't even

downloaded the app, and Alice is a user who has downloaded the app,

registered, and opted-in.)

1. Have Tom aggressively forget about Bob's location if he

   ever learns it. If you see a MAC address that's not in

   your database of devices with the app installed, drop

   it instantly.

2. Have Tom forget Alice's location, too, as soon as he doesn't

   need it. If Alice doesn't answer a challenge within

   a reasonable time, flush her location. If Sam reports

   Alice leaving, flush her location. If Alice herself reports

   a location outside of the retailer that last reported

   her position, flush her location.

3. Require that the Alice already know something about her

   location before Tom will tell her more about it.

4. Issue a new challenge whenever Alice moves from retailer to

   retailer; don't let Alice checking in at Sam's allow tracking

   Alice at Sally's.

5. Don't give out unnecessary information about location in

   error messages, and don't log it either.

6. Possibly lie outright to Alice if she tries to use a wrong

   coarse location to acquire a fine location.

7. In general, treat location information reported to Tom as

   highly sensitive. Keep a tight boundary around it and don't

   let it out. For example, it should never be stored in any

   place that gets backed up.

8. Detect users who make too many bogus location requests, and

   disable them or alert somebody or something.

(

more details: for #6:

You can LIE to Alice. If Alice's

phone says "I'm at Sam's somewhere; please tell me exactly where", and

Sam hasn't told you Alice is there at all, then you can just tell

Alice a random location at Sam's. Weight it by the areas at Sam's

where people actually go, and perhaps skew it in the direction of the

location Alice reports. It might be possible for Mallory to detect

this with statistical fu, but it definitely raises her work factor.

You'll probably occasionally report a wrong location to a legitimate

user, but if that happens often enough to matter, it probably means

your underlying tracking is too poor to make the application useful

anyway.

)

### Caveats for the Location Challenge security measures

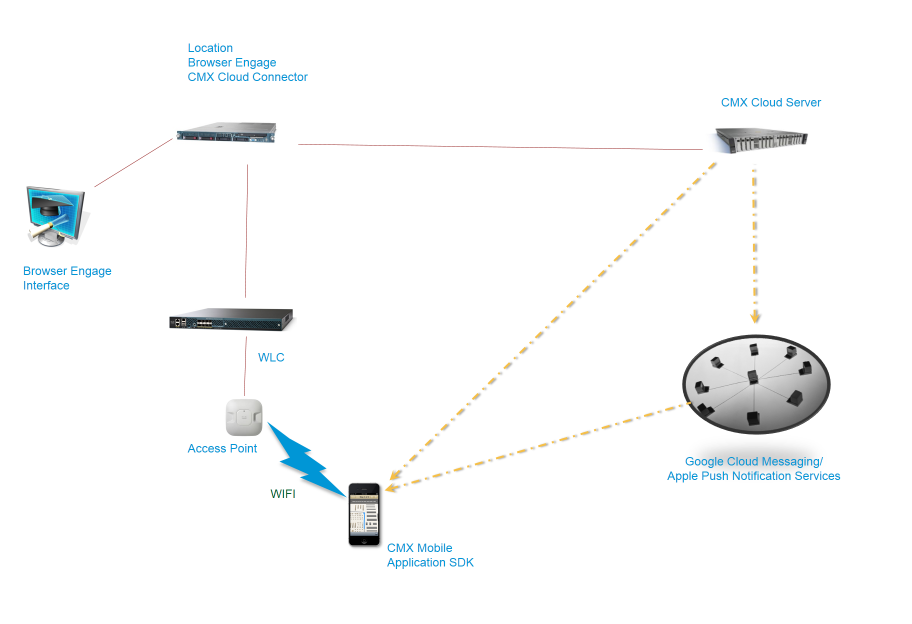
#### The GPS-coordinate-matching threshold may be tricky to configure

#### Venue owners need to accurately configure GPS markers on their floors

#### Location challenge may be unnecessary for some applications of the SDK (e.g., an automated location calibration robot)

#### If the Location Challenge proves to be unwieldy, you should fall back to using an encryption scheme such that the cloud server only stores client location data such that only the client can read it (i.e., encrypt with the client’s public key, send the encrypted blob to the client, and have the client decrypt with the client’s private key)

# Functional Structure



### CMX Browser Engage Interface

The admin interface will be extended from the existing Browser Engage interface. An authoring tool would be one extension needed in the admin interface but will not be implemented until a future release. The customer could customize the application to render HTML pages and other metadata in the mobile application. This allows the customer to brand the application without needing mobile application development experience.

### Location, Browser Engage and CMX Cloud Connector

This is the back-end server running the services for Location, Browser Engage and additional services for the CMX Mobile Application. The server resides in the customer network.

### CMX Cloud Server

This server can run in the cloud to allow access from outside the customer network or locally in the customer network. This is needed when the mobile application allows access to the device location. When the user gives access the expectation is the device will be on a public network and not on any customer specific network. The registration key will need to be stored at the time of registration to allow for notifications to the device. The CMX Cloud Server will also allow the rendering of the venue when the user is not present at the venue, but wishes to view the customer branded page on a public network. Finally, the user may not join the customer network when visiting the venue, but will still want the indoor location information shown. The location information can be transmitted across the user’s cellular connection from the CMX Cloud Server.

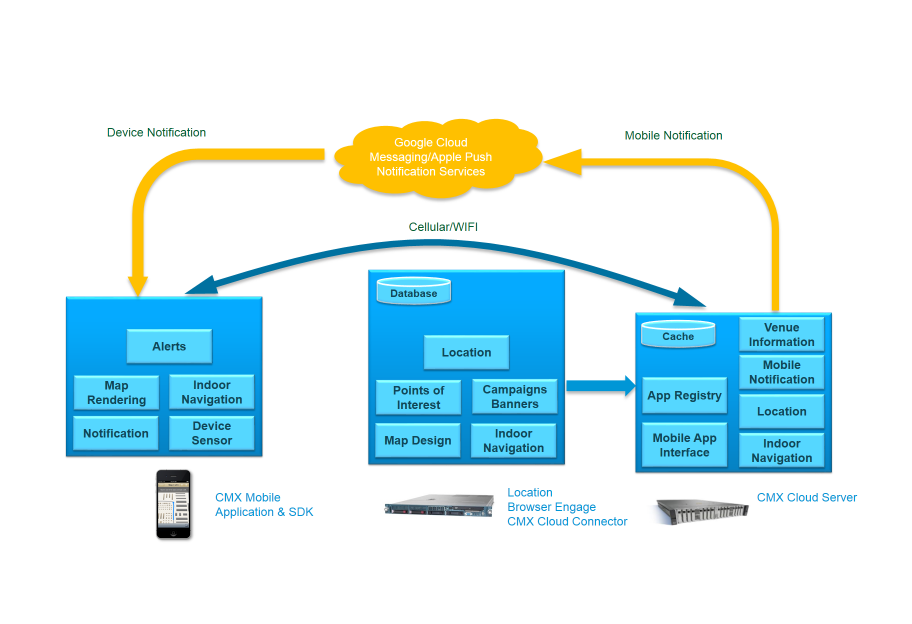
### Google Cloud Messaging/Apple Push Notification Services

A mobile device receives notification messages from Google Cloud Message for Android and Apple Push Notification Services for iOS. These services are sent the device key and the message to be sent to the device.

### CMX Mobile Application and SDK

An application and SDK will be created for the iPhone and Android platforms. The SDK will be leveraged in the mobile application which will allow customers to embed indoor location tracking into their own existing applications. The application will allow a user to choose a venue and view the customized interface for that venue. This interface was designed ahead of time in the CMX Mobile Application Admin. The application will be able to receive notifications when the application is not running. When notified, the application can be launched from the application and will include a message on how to join the customer network when first initiated. The core feature is the indoor location which will display the current location of the device on a map with any points of interests which may be nearby. The location will be updated as the user moves through the venue and can help navigate the user to some defined location.

## CMX Mobile Application Architecture



### Location, Browser Engage and CMX Cloud Connector

Browser Engage will be leveraged since many of the features can also apply to the CMX Mobile Application. Browser Engage user interface will remain the same with only a few minor changes for the mobile application. This includes adding the ability to specify the push notification message when a user enters a zone. There will also need to be configuration settings for connecting to the CMX Cloud Server which are also similar to how the current Mobile App Enablement. The configuration is done in the Cisco Prime Infrastructure product and the most significant change is to include the IP address of the server since in this release a public cloud server will not be initially supported. Mobile App Enablement require the user to provide URLs and this is not needed for CMX Cloud Server. Cisco Prime Infrastructure communicates with the CMX Cloud Connector to set the configuration. The CMX Cloud Connector is responsible for communicating all the Browser Engage settings needed by the CMX Cloud Server. The settings are retrieved from Browser Engage and sent to the CMX Cloud Server using well defined REST APIs. Cisco Prime Infrastructure will also register the location server to send location events to the CMX Cloud Server.

### CMX Cloud Server

The CMX Cloud Server is a gateway which interfaces with the MSE. The CMX Cloud Server will be deployed on the Cisco Cloud or can be hosted in on any other Cloud. The Mobile Clients will interact with the Cloud Server via the Mobile SDK where the Mobile application which uses the SDK can be a Cisco Mobile APP which will be reference implementation or any other APP.

The CMX Cloud Server will have REST based APIs to interact with the mobile client. The Cloud Server will also communicate with the MSE via REST. The Cloud Server will not be limited to act as a gateway but also a cache for faster access of data for the clients.

Redis is an in memory key-value store. Any number of Redis servers can be run on one machine/virtual machine or any number of machines in a distributed environment to form a cluster. In a distributed environment data will be distributed among the shards where each shard is a Redis server. The main purpose of this in-memory store is to allow faster access to data.

The following data will be stored in Redis

* Client location where key will be the mac-address and value will have the location information
* Point of Interest information for each floor or whole venue
* Path information. This will contain path information from one POI to another.

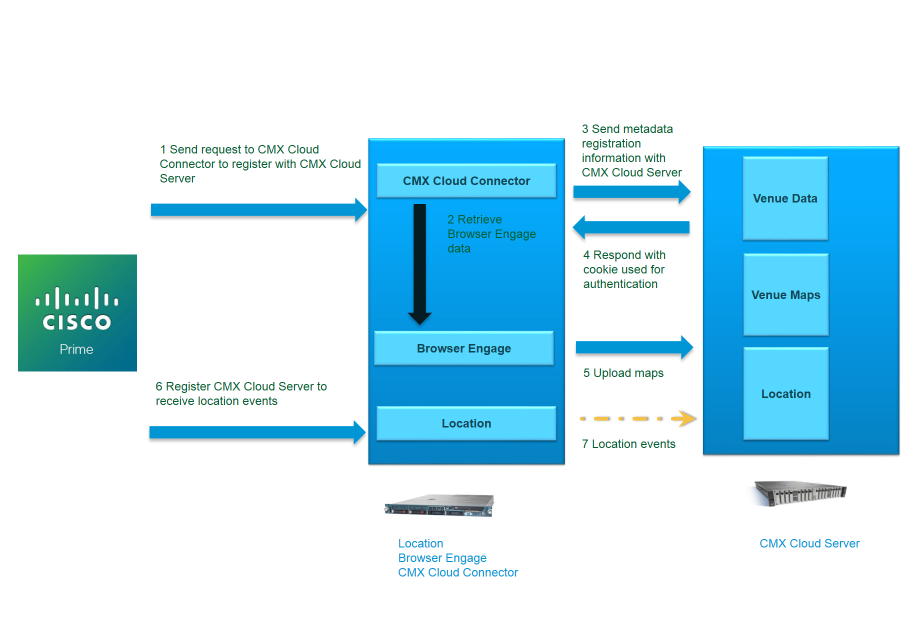
The cloud server will be managed by the MSE via the PI. Once the CMX Cloud Server is added to the MSE the setup information will be sent to it. The setup information will contain all the information about the Venue. Venue will contain floor information and floors will contain point-of-interests. Along with the Venues all the Paths will be sent. Once the floor information is sent to cloud server it will send a request to the MSE to send maps for all the floors. The Maps will be stored in disk on the cloud server. MSE will send periodic updates to the cloud server for each location change for each client. The information will be updated in one of the Redis servers. There will be a flow-control mechanism which will slow down the number of updates if the updates are sent too often. After each location update information is sent from the MSE to the cloud-server a corresponding acknowledgement will be sent back, along with this acknowledgement there will be flow-control information informing the MSE to send updates less periodically so as not to overwhelm the cloud server.

### CMX Mobile Application and SDK

The CMX Mobile communicates to the CMX Cloud Server using two different sets of APIs. The configuration, monitoring and other non-real time related functions will use REST. For the location, time sensitive information will use REST but consider other more persistent protocols.

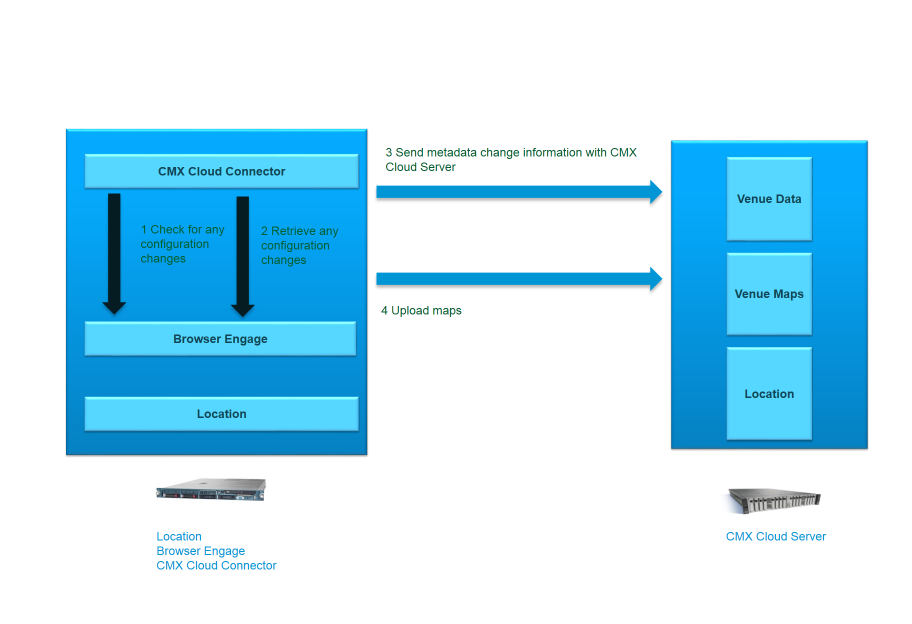
# System Flow

## Browser Engage To CMX Cloud Registration

  
  
When an administrator wants to use the mobile application then the CMX Cloud Server needs to be configured. The CMX Cloud Server is a gateway which interface for mobile clients to access.

1. From Prime Infrastructure the CMX Cloud Server is configured. PI will first contact the CMX Cloud Connector using REST to register the CMX Cloud Server. The registration will include IP address of the CMX Cloud Server and other attributes such as email, address etc.
2. CMX Cloud Connector will use REST to retrieve configuration information in Browser Engage.
3. The Browser Engage data will be formatted and a registration is done with the CMX Cloud Server
4. The CMX Cloud Server will respond with a cookie which will be used to authenticate any changes from CMX Cloud Connector.
5. The CMX Cloud Connector will then upload the images to the CMX Cloud Server
6. After the CMX Cloud Connector has successfully registered. PI will then register for location events to be sent to the CMX Cloud Server.
7. After registration the location events will be sent to the CMX Cloud Server. In each location update the CMX Cloud Server will reply and can throttle the location events or filter which MAC Addresses are tracked.

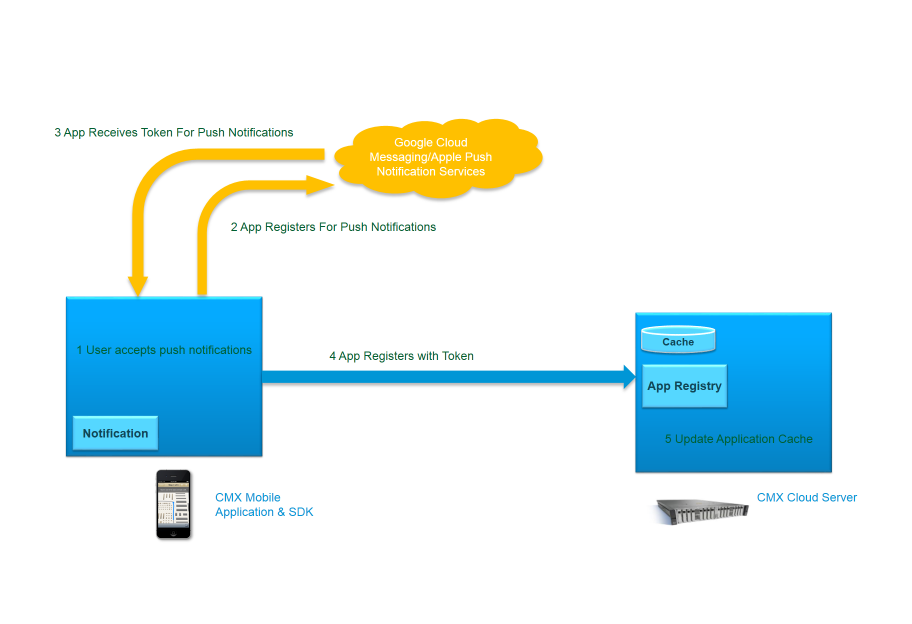
## CMX Admin Venue Push



If the administrator makes any changes in Browser Engage, those changes need to be updated in the CMX Cloud Server.

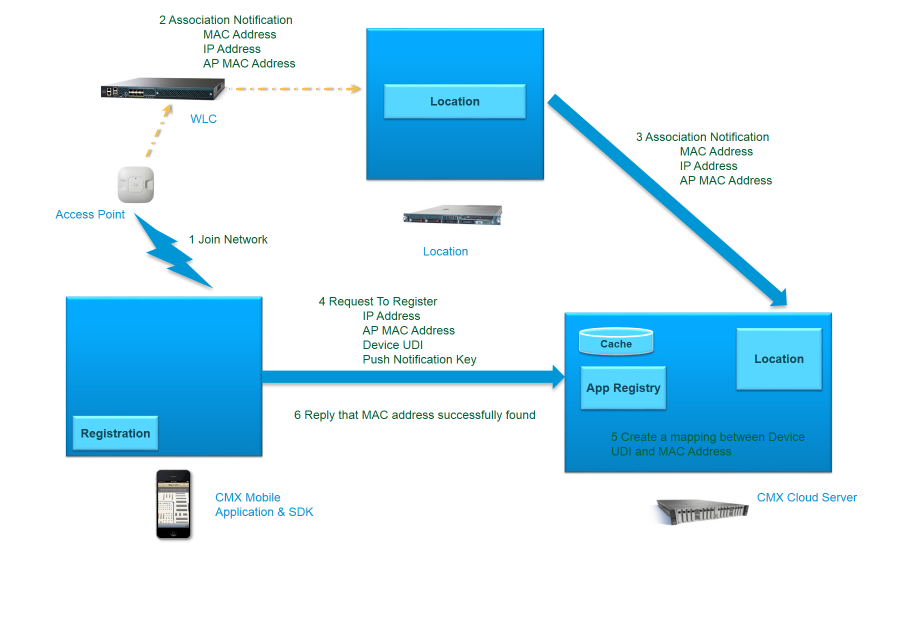
1. Either CMX Cloud Connector will query if any configuration changes have occurred or will be notified about a configuration change
2. Retrieve the changed configuration since the check may just indicate if a change has occurred
3. The configuration changes are sent to the CMX Cloud Server
4. Any new or modified maps are uploaded to the server

## CMX Mobile Application Initial Launch

  
  
When a user launches the mobile application for the first time a prompt will be displayed to allow for push notifications to the device. This will allow the application to send push notifications even if the application is not currently running. When the user accepts the following steps will be started.

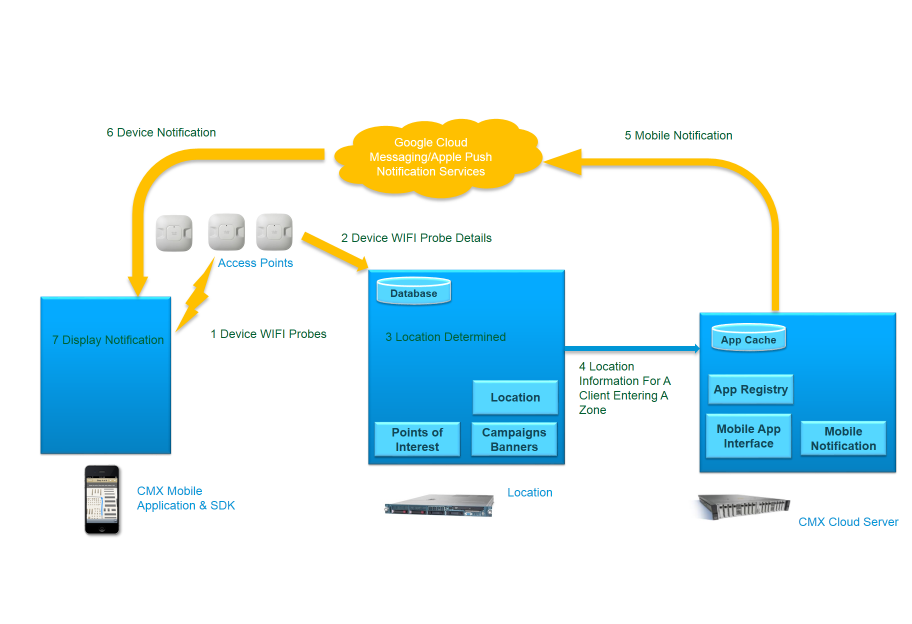
1. User accepts the push notifications to the device. A pop-up will be displayed to give the user the option to accept push notifications.
2. The application will register for push notifications from Google Cloud Messaging for Android devices and Apple Push Notification Services for iOS devices
3. The application will receive a key or token from the requesting server
4. The application will then send the key or token to the CMX Cloud Server. For Android this registration would include the MAC address of the device and a device UDI. For iOS 7 devices the MAC address will not be known unless the user is joined to a Wifi Access Point so just the device UDI will be sent. If the iOS 7 is joined then the next section describes how the device will be registered.
5. The CMX Cloud Server will then take the key and token and store it in the cache

## CMX Mobile Application iOS 7 Registration

  
  
Starting with iOS7, an application will no longer be able to retrieve the MAC address of the Wi-Fi radio. The device's Wi-Fi probing and communication with APs will continue to use the actual MAC address of the radio. The MAC address of the client is needed to send push notifications to the device. When a mobile device attempts to probe the network, the MAC address is sent while doing the probe. This can be used to identify the client and send notification when the device enters a venue or a zone. The MAC address is also needed when navigating a venue. The client will receive positioning information based upon the MAC address. Android will use this also as a verification the mapping is correct.

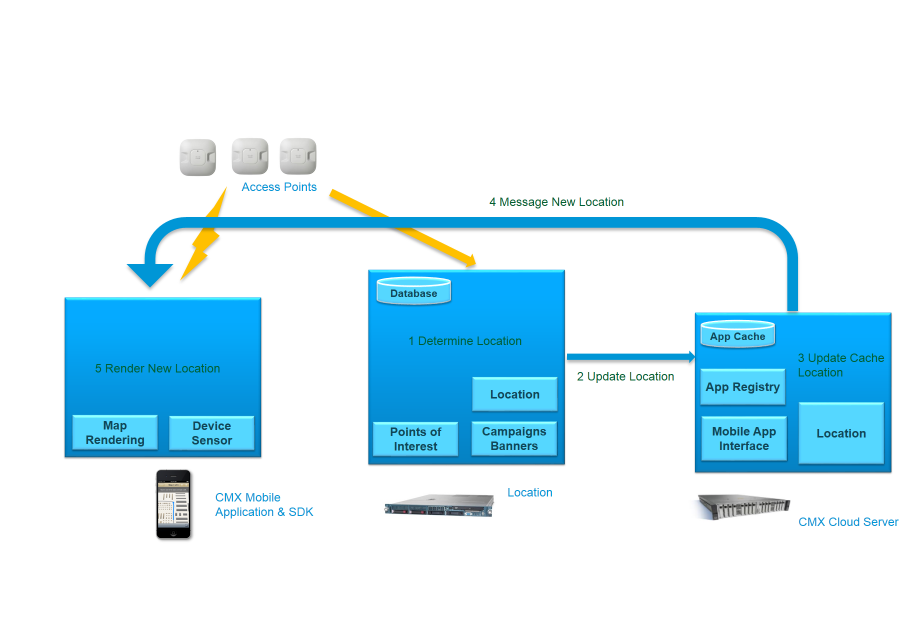
1. The device joins a wireless network monitored by MSE systems
2. The WLC will detect the new device joining the network. The WLC will send the MAC Address of the device, IP Address of the device and AP MAC Address the device joined with
3. The location service will then send a notification for this event to the CMX Cloud Server. The CMX Cloud Server will determine if a mapping already exists for a MAC address to a device UDI. If not then an application registration cache will be updated
4. The user launches the CMX Mobile Application. A registration has not been completed as of yet so a registration request is sent with the IP Address of the device, AP MAC address and device UDI. If this were the very first time the application was launched then a Push Notification Key would also be included.
5. Attempt to find the device MAC address based upon the AP MAC address and IP address of the device. If a mapping is found then an entry is created in the CMX Application registration table.
6. Reply back to the device indicating the registration was successful

## CMX Mobile Push Notification

  
Push notifications can be sent to a mobile device after registration for push notifications. The application does not have to be running for the notification to be received and viewed by the user. Also if the device is offline the notification will be sent when the device is online again. Notifications can be sent for any reason and is left to applications discretion on when and how many notifications to send. The following steps are an example of sending a notification when a user enters a venue for the first time:

1. Device probes for WIFI at a new venue or joins an AP at a venue
2. The AP will send the probe and other information to MSE as part of the location services
3. The location will be determined and CMX Mobile will determine the device is new for the venue and a notification should be sent
4. The location and mobile notification message information is sent to the CMX Gateway using the XMPP protocol
5. CMX Gateway will look up the entry in the cache table for the device key or token to be sent to Google Cloud Messaging or Apple Push Notification Services. The push notification message is formatted and sent to the Google Cloud Messaging or Apple Push Notification Services.
6. Google Cloud Messaging or Apple Push Notification Services will then send the message to the device
7. The notification is displayed on the device. The user can, by selecting the message, automatically launch the application

## CMX Mobile Application Location Tracking

  
  
One of the most important pieces of the CMX mobile application is the indoor location tracking. This gives the application the ability to track and navigate a user within a venue. The current device position will be shown on an indoor map in the CMX mobile application. This tracking will also be provided in an SDK for other applications to leverage. The steps for getting the tracking location are the following:

1. The current location is calculated by the location service. If the location is new or has changed by some distance then a location event is prepared to be sent to the CMX Gateway
2. The location update is sent to the CMX Gateway using REST.
3. The location is updated in the cache for the MAC address
4. The application may request for a location update. Whether or not the CMX Gateway sends a push notification is still to be determined. Again XMPP appears to be the desired protocol for this location update
5. The device will then take this location update information and render it on the map

## Location Challenge

Unauthorized retrieval of location information is a threat to this system. For example, an attacker may spoof her MAC address with the MAC address of her target when registering her app. When her target enters a venue reporting to the CMX Gateway, the attacker will receive a push notification, and will be able to access the x/y coordinate of her target within the venue. To prevent this, the following process will be implemented as an authentication step:

1. The location update for a particular client sent from MSE to CMX Gateway will contain GPS coordinates as estimated by the MSE.
2. The CMX Gateway will store this GPS coordinate as part of the location structure for the client.
3. When the mobile application requests its location from the CMX Gateway, the CMX Gateway checks for a valid session token with the mobile application’s request. If such a token is not present, the following happens:
   1. The CMX Gateway directs the application to send its GPS coordinates
   2. The mobile application computes its GPS coordinates as it sees fit, and replies
   3. The CMX Gateway compares the GPS coordinates from the mobile application with the GPS coordinates sent from the MSE tracking the corresponding client. If they match, the mobile application is granted a valid session token, and the mobile application may continue.
   4. If not, the mobile application is denied access to its location, and an error message may be logged.
4. When the client is detected as entering a different venue, the CMX Gateway invalidates the corresponding mobile application’s session token.

# Data Structures

Describe how the cache will be maintained in the cloud.

# Description of Algorithms

# Interface Design

## CMX Cloud Server API for CMX Mobile Application

CMX Cloud Server provides data to clients by using REST APIs. All these REST APIs require basic authentication to publish data. The format for data is JSON and all APIs are versioned to v1.

### /api/cmxmobile/v1/clients/register

**Form Parameters:**

* pushNotificationRegistrationId - The push notification token your app got when it registered for push notifications
* clientType – “android” or “ios” or “ios6”
* clientMACAddress – Client MAC Address for android devices
* apMACAddress - The MAC address of the AP
* clientIPAddress – The client IP address

**Results:**

* 4xx code if something went wrong (like the MAC address you registered with was already taken for some reason)
* 5xx code if something went wrong on the server
* 202/Accepted code if the server hasn't received the matching AssociationEvent yet
* 201/Created code if everything went right. There will also be a Set-cookie header, and the URI created will be /api/cmxmobile/v1/clients/location/{deviceId}

### /api/cmxmobile/v1/clients/location/{deviceId}

**Cookie:**

* Cookie is required from registration to authenticate device

**Parameters:**

* deviceId- This could be any ID which can be used to identify a client and can be mapped to a MAC address in the CMX Cloud Server

To get the location for a client

**Results:**

{

"deviceId": "000ab8ffe9ed",

"venueId": "782309123536",

"floorId": "730297895206518938",

"zoneId":"78230912353602"

"mapCoordinate": {

"x": "95.21",

"y": "36.11"

}

}

### /api/cmxmobile/v1/venues/info/

**Parameters:**

* None

To get all venue information

**Results:**

[

{

"venueId": "782309123536",

"streetAddress": "3625 Cisco Way, San Jose, CA 95131",

"name": "Cisco",

"locationUpdateInterval" : 2,

"imageType" : "none",

"wifiConnectionMode" : "prompt",

"preferredNetwork": [

{

"ssid": "alpha",

"password": ""

},

{

"ssid": "blizzard",

"password": ""

}

]

}

]

### /api/cmxmobile/v1/venues/info/{venueId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location

To get the venue information from a specific venue

**Results:**

{

"venueId": "782309123536",

"streetAddress": "3625 Cisco Way, San Jose, CA 95131",

"name": "Cisco",

"locationUpdateInterval" : 2,

"imageType" : "none",

"wifiConnectionMode" : "prompt",

"preferredNetwork": [

{

"ssid": "alpha",

"password": ""

},

{

"ssid": "blizzard",

"password": ""

}

]

}

### /api/cmxmobile/v1/venues/image/{venueId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location

To get the image for a specified venue

**Results:**

Venue image

### /api/cmxmobile/v1/maps/info/{venueId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location

To get the map information from a specific venue

**Results:**

[

{

"mapHierarchyString": "System Campus>SJC-14>1th Floor",

"floorId": "730297895206518931",

"venueid": "782309123536",

"dimension": {

"length": "185.8",

"width": "295.8",

"height": "10.0",

"offsetX": "0.0",

"offsetY": "0.0",

"unit": "FEET"

}

},

{

"mapHierarchyString": "System Campus>SJC-14>2nd Floor",

"floorId": "730297895206518933",

"venueid": "782309123536",

"dimension": {

"length": "185.8",

"width": "295.8",

"height": "10.0",

"offsetX": "0.0",

"offsetY": "0.0",

"unit": "FEET"

}

},

{

"mapHierarchyString": "System Campus>SJC-14>3rd Floor",

"floorId": "730297895206518935",

"venueid": "782309123536",

"dimension": {

"length": "185.8",

"width": "295.8",

"height": "10.0",

"offsetX": "0.0",

"offsetY": "0.0",

"unit": "FEET"

}

},

{

"mapHierarchyString": "System Campus>SJC-14>4th Floor",

"floorId": "730297895206518938",

"venueid": "782309123536",

"dimension": {

"length": "185.8",

"width": "295.8",

"height": "10.0",

"offsetX": "0.0",

"offsetY": "0.0",

"unit": "FEET"

}

}

]

### /api/cmxmobile/v1/maps/info/{venueId}/{floorId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* floorid - The identifier for the floor in the venue. This can be retrieved from the client location

To get the map information from a specific floor

**Results:**

{

"mapHierarchyString": "System Campus>SJC-14>4th Floor",

"floorId": "730297895206518938",

"venueid": "782309123536",

"dimension": {

"length": "185.8",

"width": "295.8",

"height": "10.0",

"offsetX": "0.0",

"offsetY": "0.0",

"unit": "FEET"

}

}

### /api/cmxmobile/v1/maps/image/{venueId}/{floorId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* floorid - The identifier for the floor in the venue. This can be retrieved from the client location

To get the map image from a specific floor

**Results:**

Map image

### /api/cmxmobile/v1/pois/info/{venueId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location

To get the points of interest information from a specific venue

**Results:**

[

{

"floorid": "730297895206518938",

"id": "7167",

"name": "Benbow",

"points": "[{\"x\":18.077657542966264,\"y\":150.684289276808},{\"x\":34.62762571610439,\"y\":150.8},{\"x\":34.50031826861871,\"y\":162.371072319202},{\"x\":18.586887332908976,\"y\":162.371072319202}]",

"venueid": "782309123536"

},

{

"floorid": "730297895206518938",

"id": "7160",

"name": "Break Room",

"points": "[{\"x\":139.16,\"y\":66.6},{\"x\":177.95,\"y\":66.6},{\"x\":178.33,\"y\":90.35},{\"x\":138.78,\"y\":90.19}]",

"venueid": "782309123536"

},

{

"floorid": "730297895206518938",

"id": "7166",

"name": "Capitola",

"points": "[{\"x\":18.26861871419478,\"y\":138.99750623441398},{\"x\":33.92743475493316,\"y\":138.76608478802993},{\"x\":34.436664544875875,\"y\":150.2214463840399},{\"x\":17.759388924252068,\"y\":150.10573566084787}]",

"venueid": "782309123536"

}

]

### /api/cmxmobile/v1/pois/info/{venueId}/{floorId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* floorid - The identifier for the floor in the venue. This can be retrieved from the client location

To get the points of interest from a specific floor

**Results:**

[

{

"floorid": "730297895206518938",

"id": "7167",

"name": "Benbow",

"points": "[{\"x\":18.077657542966264,\"y\":150.684289276808},{\"x\":34.62762571610439,\"y\":150.8},{\"x\":34.50031826861871,\"y\":162.371072319202},{\"x\":18.586887332908976,\"y\":162.371072319202}]",

"venueid": "782309123536"

},

{

"floorid": "730297895206518938",

"id": "7160",

"name": "Break Room",

"points": "[{\"x\":139.16,\"y\":66.6},{\"x\":177.95,\"y\":66.6},{\"x\":178.33,\"y\":90.35},{\"x\":138.78,\"y\":90.19}]",

"venueid": "782309123536"

},

{

"floorid": "730297895206518938",

"id": "7166",

"name": "Capitola",

"points": "[{\"x\":18.26861871419478,\"y\":138.99750623441398},{\"x\":33.92743475493316,\"y\":138.76608478802993},{\"x\":34.436664544875875,\"y\":150.2214463840399},{\"x\":17.759388924252068,\"y\":150.10573566084787}]",

"venueid": "782309123536"

}

]

### /api/cmxmobile/v1/pois/image/{venueId}/{poiId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* poiId - The identifier for the point of interest. This can be retrieved from the point of interest information

To get the image for a specified point of interest

**Results:**

Point of interest image

### /api/cmxmobile/v1/pois/{venueId}?search={keyWord}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* keyWord - Key word to be used in the search

To get the points of interest from a key word search

**Results:**

[

{

"floorid": "730297895206518938",

"id": "7163",

"name": "Newport Beach",

"points": "[{\"x\":265.62698917886695,\"y\":150.10573566084787},{\"x\":281.2858052196053,\"y\":150.10573566084787},{\"x\":281.66772756206234,\"y\":160.75112219451373},{\"x\":265.49968173138126,\"y\":160.98254364089775}]",

"venueid": "782309123536"

},

{

"floorid": "730297895206518938",

"id": "7158",

"name": "Pebble Beach",

"points": "[{\"x\":201.20942075111392,\"y\":70.61246882793017},{\"x\":219.41438574156587,\"y\":70.49675810473816},{\"x\":219.1597708465945,\"y\":97.80448877805486},{\"x\":201.46403564608528,\"y\":97.68877805486285}]",

"venueid": "782309123536"

},

{

"floorid": "730297895206518938",

"id": "7159",

"name": "Sunset Beach",

"points": "[{\"x\":166.58179503500955,\"y\":71.53815461346633},{\"x\":179.439847231063,\"y\":71.42244389027432},{\"x\":179.69446212603435,\"y\":96.30024937655861},{\"x\":166.8364099299809,\"y\":96.30024937655861}]",

"venueid": "782309123536"

}

]

### /api/cmxmobile/v1/routes/clients/{deviceId}?destpoi={destPoi}

**Cookie:**

* Cookie is required from registration to authenticate device

**Parameters:**

* deviceId- This could be any ID which can be used to identify a client and can be mapped to a MAC address in the CMX Cloud Server
* destPoi – Destination point of interest

To get the route to a point of interest from the current location of the specified device

**Results:**

[

{

"x": 95.70337364735836,

"y": 30.909226932668332

},

{

"x": 66.42266072565245,

"y": 31.14064837905238

},

{

"x": 37.14194780394653,

"y": 30.56209476309226

},

{

"x": 36.37810311903246,

"y": 16.676807980049887

}

]

### /api/cmxmobile/v1/banners/info/{venueId}/{floorId}/{zoneId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* floorid - The identifier for the floor in the venue. This can be retrieved from the client location
* zoneId – The identifier for the zone in a floor. This can be retrieved from the client location

To get all the banners for a specific zone

**Results:**

[  
 {  
 "zoneid": "78230912353603",  
 "venueid": "782309123536",  
 "id": "01",  
 "imageType" : "png",  
 "url": "https://localhost/api/cmxmobile/v1/banners/image/782309123536/730297895206518931/78230912353603/01"  
 },  
 {  
 "zoneid": "78230912353603",  
 "venueid": "782309123536",  
 "id": "02",  
 "imageType" : "png",  
 "url": "https://localhost/api/cmxmobile/v1/banners/image/782309123536/730297895206518931/78230912353603/02"  
 }  
]

### /api/cmxmobile/v1/banners/image/{venueId}/{floorId}/{zoneId}/{imageId}

**Parameters:**

* venueId - The identifier for the venue. This can be retrieved from the client location
* floorid - The identifier for the floor in the venue. This can be retrieved from the client location
* zoneId – The identifier for the zone in a floor. This can be retrieved from the client location
* imageId – The identifier for the image

To get the banner image for a specific zone

**Results:**

Banner image

## CMX Cloud Server API for external use

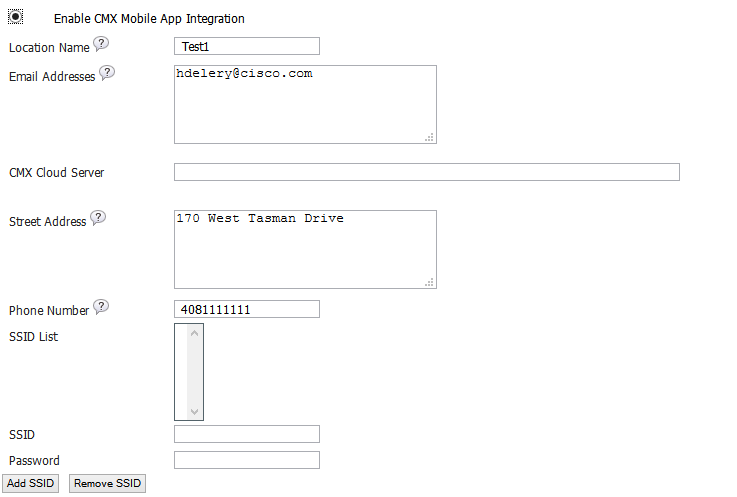
CMX Cloud Server provides REST APIs for external use. All these REST APIs require basic authentication to publish data. The format for data is JSON and all APIs are versioned to v1.

# End User Interface

## Prime Infrastructure

The user interface will leverage the existing Browser Engage interface and only adding a few minor changes in this release.

One change of note will be in the Prime Infrastructure product. The current Mobile App Enablement page will be extended to have an option to add the CMX Cloud Server.



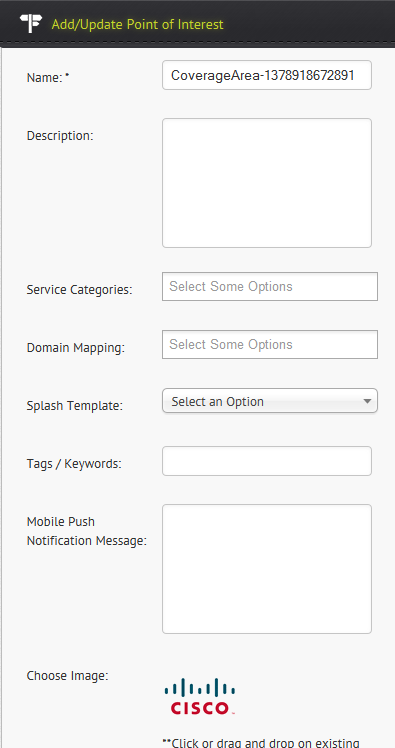
The page will have the same options for name, email address, street address, and phone number. The page will vary on the CMX Cloud Server option to enter the host name or IP address of the CMX Cloud Server. Finally the SSID list is a list of preferred networks a mobile user should join when in the venue.

## Browser Engage

Browser Engage UI changes are minimal in this release.

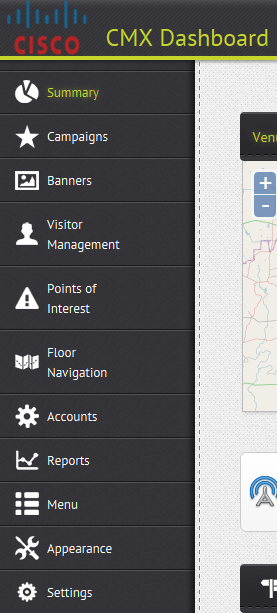
### Mobile Push Notification Message

The ability to send mobile push notifications when a user enters an area will be added in the Browser Engage screen. This notification will be sent when the user enters the zone and will not be sent again for some period of time in case the user exits and enters the zone again.



### CMX Cloud Server

A new folder will be created in Browser Engage for the CMX Cloud Server. The folder will contain monitoring capabilities to determine if the connection between Browser Engage and CMX Cloud Server is operationally up and when the last synchronization was done and how long it took to complete with the CMX Cloud Server.



# Software Restrictions and Considerations

The CMX Cloud Server will run on a separate system from Browser Engage and Location.

# Hardware Restrictions and Considerations

CMX Cloud Server will run on a Linux server with a minimum of 2 CPUs and 16G RAM

# External Restrictions and Configuration

CMX Cloud Server is dependent upon Browser Engage, Location and CMX Cloud Connector.

# Development Testing

Development will create unit test cases for the CMX Cloud Server, CMX Cloud Connector and the CMX Application SDK for Android and iOS. The Prime Infrastructure changes will not be tested with unit test cases since the changes are minimal and there is no suited framework for unit testing currently. The cost of creating unit tests in this framework is not worth the effort. The unit test cases will be reviewed by test and when possible test will develop or detail needed unit test cases. Integration tests will also be created and may require a more complex setup and are not required to run in a development environment. They are required to run in the continuous integration builds to allow for additional code coverage and additional validation with actual running system versus simulated systems used in unit testing.

## Unit Testing

Unit testing will have to be done differently for each component since each component runs on a different platform.

### CMX Cloud Server

The CMX Cloud Server will be unit tested with TestNG as the primary tool. Mockups will be done with Mockito to limit the testing requirements. The unit test cases should be run in Eclipse on a developers system with a minimum of setup. The unit test cases will be run as part of the continuous integration whenever a code review is submitted. The minimum amount of code coverage will be 60%. The unit testing will focus on the following areas:

* Browser Engage configuration information and maps interfaces
* Location notifications
* Mobile Application SDK interfaces

### CMX Cloud Connector

The CMX Cloud Connector will be unit tested with TestNG as the primary tool. Mockups will be done with Mockito to limit the testing requirements. The unit test cases should be run in Eclipse on a developers system with a minimum of setup. The unit test cases will be run as part of the continuous integration whenever a code review is submitted. The minimum amount of code coverage will be 60%. The unit testing will focus on the following areas:

* Browser Engage configuration information and maps interfaces
* CMX Cloud Server interfaces
* Prime Infrastructure configuration interfaces

### CMX Mobile Application SDK

The CMX Mobile Application SDK will need to test the CMX Cloud Server API interfaces and make sure the SDK returns the appropriate information to the application.

## Integration Testing

Integration testing will involve combining the components together and testing as a group. Some of the components will have to be simulated in order to create a reproducible test or they are too complex to setup and create a reproducible test. Node.js is the preferred tool to simulate the components needed. Also for continuous integration the component being changed should be tested with simulators to allow for more parallel tests to be executed.

### Browser Engage

Since the code is not changing directly in Browser Engage and only the configuration data on the server is needed a dedicated server will be used for some of the testing. The Browser Engage will be configured with a specific static configuration for testing. This will allow the CMX Cloud Connector to load real data from the server and validate this data on each. Also multiple tests can be run in parallel with this server since it will not change. A simulator will also be used in a future release to allow for varying configurations to be used in testing.

### Location

The simplest approach for location event notification is to use a Node.js simulator to send the events to the CMX Cloud Server. The simulator will read a configuration file and can send events at an interval or controlled so the events can be verified on the CMX Cloud Server to have accurate information.

### CMX Cloud Connector

This will always run normally. However the CMX Cloud Connector should be able to connect remote Browser Engage server to allow for multiple instances to run. When simulated Browser Engage data is needed in the integration testing environment the CMX Cloud Connector will still run. The input data to the CMX Cloud Connector will be simulated if needed but the connector will remain running unaware of the simulation.

### CMX Cloud Server

This will always run normally. Multiple CMX Cloud Servers can run on the same system to allow for parallel tests. The server will take inputs from the CMX Cloud Connector and the Node.js location event simulator. The event simulator will have leverage the CMX Mobile Application interface to verity the events.

### CMX Mobile Application SDK

Automation testing will have to be investigate further but will only be a stretch goal for this release. Testing will be done manually in this case in combination with the automated tests.

# Patentability Considerations

We have two main patentable ideas in this product:

1. Informing a network node about its MAC address when there is no established L3 communication channel between that node and the surrounding infrastructure

2. Providing an authentication mechanism for when the credentials presented during the initial registration or user creation step are easily forged (e.g., to match the userid of another entity)

Balachander Chandrasekaran has a patent idea in draft and/or pending with the Cisco patent committee for #1. [Ethan: If I recall correctly, it involves running a proxy on the AP such that certain requests from a client will be intercepted by the AP and the client's MAC address injected into the response.] Our solution, which does not involve a proxy, may be separately patentable. Our solution has also been disclosed to Phunware, who may or may not have an NDA with Cisco which would prevent the details of our solution from entering the public domain prior to Cisco filing for a patent.

The patentability of #2 is addressed by section 4.7 and CPOL #990648:

<https://wwwin-cpol.cisco.com/cpol/patent.cgi?task=PatDetails&patent_p=990648>

# References

* TBD

# Glossary

The following list describes acronyms and definitions for terms used throughout this document:

* **PI**: Prime Infrastructure

# Future Consideration

1. Allow the administrator to select which floors are published to the CMX Cloud Server
2. CMX Mobile Application Authoring Tool
3. Dashboard for statistics to be shown on the CMX Cloud Server
4. Diagnostic page on the CMX Cloud Server

# Review Action Items

End of Document