



# Alcatel-Lucent Enterprise Mobile Guest Softphone

Release - 2.0

API User Guide

# Table of Contents

---

Legal notice .....	4
Document history.....	4
Terminology .....	4
1    Introduction .....	5
1.1    Context.....	5
2    High level overview .....	6
2.1    Actors .....	6
2.2    Architecture.....	6
2.3    Actions .....	7
2.3.1    Check-in .....	7
2.3.2    Check-out.....	7
2.3.3    Call .....	7
3    Detailed action workflows .....	8
3.1    Guest check-in.....	8
3.2    Guest login on client App .....	9
3.3    Guest logout .....	10
3.4    Guest check-out .....	11
3.5    Guest initiates a call.....	12
3.6    Guest receives a call .....	13
4    SDK.....	14
4.1    Overview.....	14
4.2    Service gateway part / REST JSON API : .....	14
4.2.1    Device .....	14
4.2.2    Speed Dials .....	21
4.2.3    Help message .....	24
4.2.4    Guest Profile .....	26
4.2.5    Configuration properties.....	33
4.2.6    Authentication.....	39
4.2.7    Room Phone association .....	42
4.2.8    Resource .....	45
4.3    Sample of rest/json request : .....	48
4.3.1    Create a guest : .....	48
4.3.2    Get all guests .....	48
4.3.3    Delete a guest according to his room number .....	49
4.4    Mobile application part .....	50
4.4.1    MGS API integration into an IOS application .....	50
4.4.2    MGS API integration into an Android application .....	54
4.4.3    Methods .....	56
4.4.4    Events .....	60



4.4.5	iOS APNs VOIP certificate generation .....	62
4.4.6	Flow Diagrams .....	63

## Legal notice

Alcatel-Lucent Enterprise and the Alcatel-Lucent Enterprise logo are trademarks of Alcatel-Lucent Enterprise. All other trademarks are the property of their respective owners.

The information presented is subject to change without notice.

Alcatel-Lucent Enterprise assumes no responsibility for inaccuracies contained herein. Copyright © 2015 Alcatel-Lucent Enterprise. All rights reserved.

## Document history

Edition	Date	Changes / Comments / Details
1	23 January 2015	Creation
2	5 February 2015	Modification figures chapter 3.2
3	March 2016	Adding Property access
4	September 2016	Introducing use of Apple APNs mechanism
5	December 2016	Adding of VOICE_MAIL type for speeddial
6	February 2017	Modification in Checkin Call Flow
7	January 2018	Modification in chapter iOS "Client application creation" for xCode9 project settings
8	February 2018	Change "checkout phase" call flow

## Terminology

Acronym	Meaning
MGS	Mobile Guest Softphone
BYOD	Bring Your Own Device
MMoIP	Multi Media over IP
PABX	Private Automatic Branch Exchange
SDK	Software Development Kit
PMS	Property Management System
SIP	Session Initiation Protocol
OXE	OmniPCX Enterprise

# 1 Introduction

---

## 1.1 Context

This document details the use of the Mobile Guest Softphone's API.

The Mobile Guest Softphone (MGS) framework and its API allow Alcatel-Lucent Enterprise Application Partners to enhance their applications by adding integrated Voice and Multimedia Services.

The result is a product that is totally compliant with the Alcatel-Lucent Enterprise VoIP Communication Platform, the OmniPCX Enterprise (OXE).

By using this API, the Smartphone application becomes a new SIP extension on the OXE, dynamically assigned to the guest's suite, through the use of Hospitality features on the OXE.

The OmniPCX-Enterprise Hospitality Package is an optional feature, part of the global Alcatel-Lucent Enterprise Solution, orderable separately, and is required for MGS to function.

This first version of the MGS framework is limited to voice communication on the OXE.

For more information on the MGS, its deployment and use, please refer to the relevant Installation and Administration manuals, available on the Alcatel-Lucent Enterprise Business Portal.

Contact:[professional.services@alcatel-lucent.com](mailto:professional.services@alcatel-lucent.com)

## 2 High level overview

### 2.1 Actors

The actors involved in the operation of the MGS solution are:

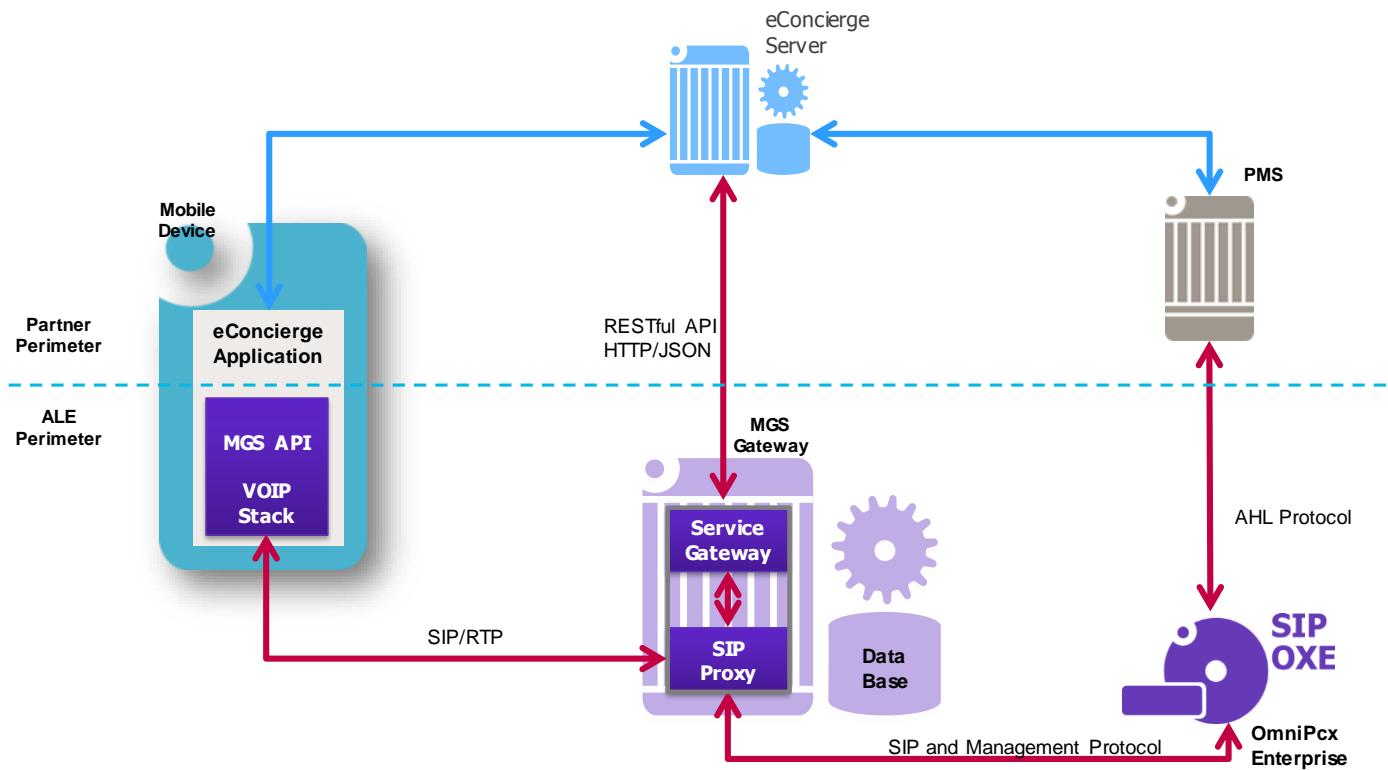
**Hotel guests**: the group of persons staying at the hotel and using the hotel app

**Hotel staff**: the group of persons in charge of the guests registering in PMS and/or in the Business Hotel App

**Hotel administrator**: the person in charge of the applications management in terms of configuration, maintenance, login/logout, on the hotel side

**IT administrator**: the person in charge of deploying, configuring and validating the whole solution

### 2.2 Architecture



The various components involved in a typical MGS setup are:

**MGS Gateway**: The server on which the MGS components are running

**Service Gateway**: The core server application used to manage users and control access to the communication system. It includes a Web Management GUI and a REST API.

These allow actions like speed dial management, guest provisioning management, QR Code generation and printing,...

**SIP Proxy**: Manages SIP communications

**SIP OmniPCX Enterprise**: The hotel's communication system, usually an Alcatel-Lucent OmniPCX Enterprise

**Premises Management System (PMS)**: The application used by the hotel staff to manage guest check-ins and check-outs

**Mobile Device**: The guest smart phone. Hotel guests will use it to run either the Mobile Guest Softphone client App, personalized as per the hotel's corporate colour scheme, or an MGS API enabled eConcierge application.

**MGS API:** API containing a VoIP stack named *Squale*

**Hotel eConcierge Server:** The Hotel's eConcierge application Server

## 2.3 Actions

Prerequisites: Prior to being able to check in or out, a guest must install the Hotel App on his/her mobile device. Please note that the telephony services will **only** be active after a check-in operation

### 2.3.1 Check-in

A typical guest check-in goes as follows:

- The receptionist checks in the guest through the PMS application.
- Upon confirmation from the OXE, the PMS Server sends a check-in request to the eConcierge application.
- The eConcierge application creates a guest on the MGS Gateway through the REST API.
- The receptionist gives the guest his account details to allow him/her to register his Mobile Device as a room extension.
- When the guest is authenticated, a license is consumed and the association between his/her device and his/her room is created.

*Note : the new SIP extension is called in parallel with the other extensions when the suite number is called.*

### 2.3.2 Check-out

A typical check-out goes as follows:

- The receptionist checks out the guest through the PMS application.
- Upon confirmation from the OXE, the PMS Server sends a checkout request to the eConcierge application.
- The eConcierge application deletes the guest on the MGS Gateway through the REST API.
- The MGS Gateway deletes the association between SIP device(s) and room number. The consumed license(s) are released.

*Note: When a guest is checking out, his communication services rights are removed. The communication in progress is ended.*

### 2.3.3 Call

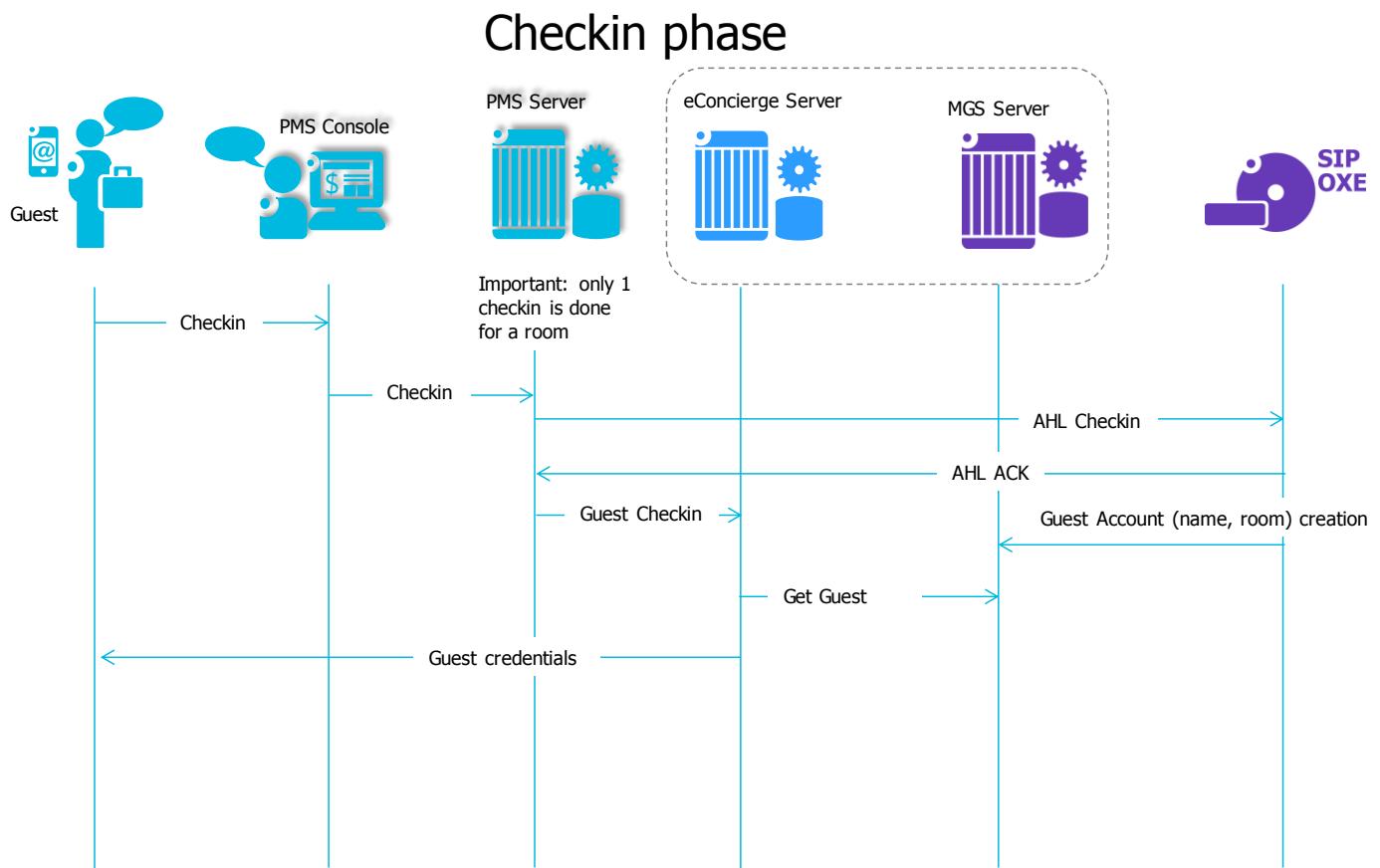
Guest needs to authenticate prior to use the “Call” functions.

A guest can perform calls, either by using the speed dial function provided by the API, or by numbers supplied by the eConcierge application.

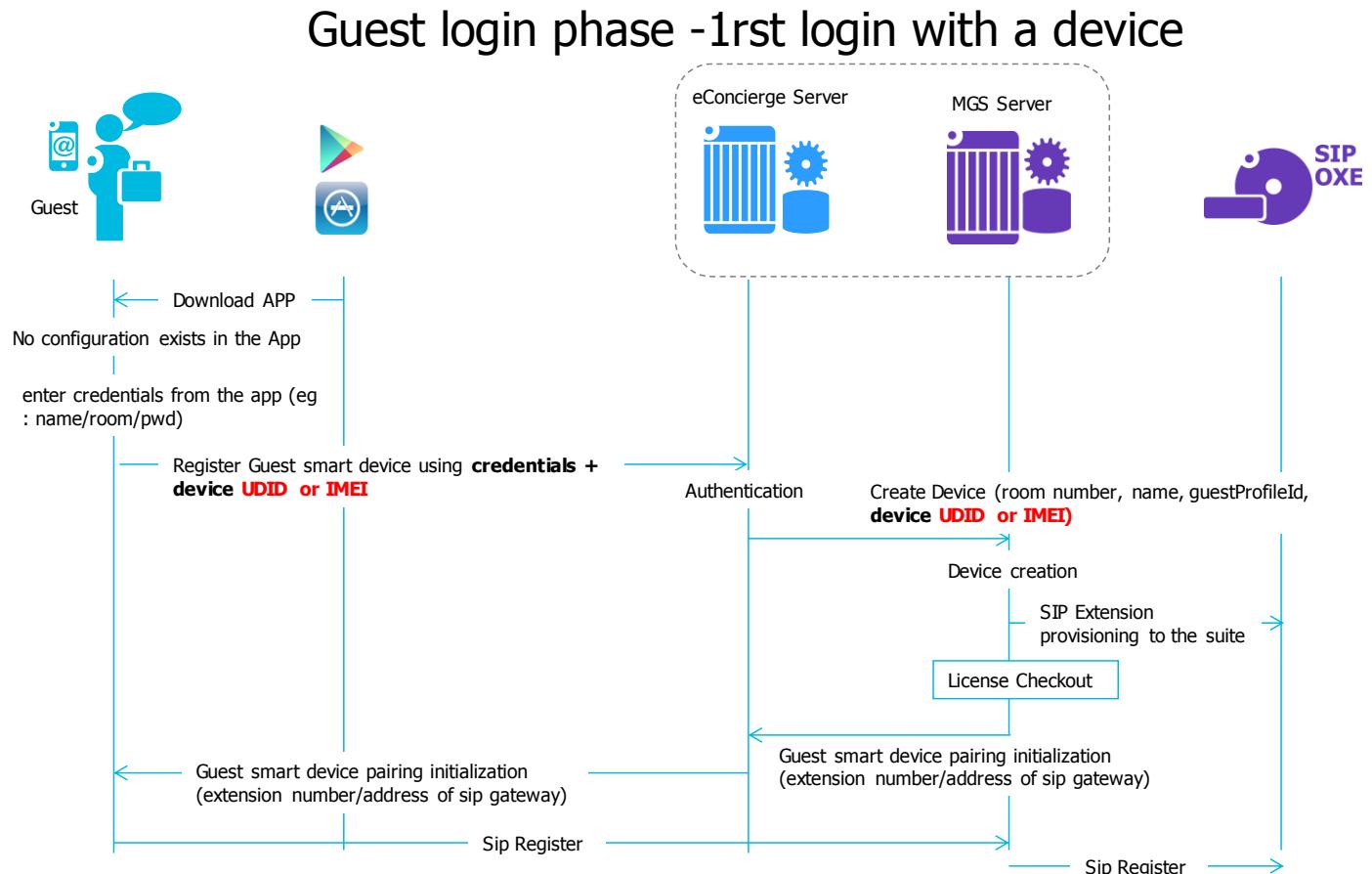
Each step of these actions is detailed in the following chapter.

## 3 Detailed action workflows

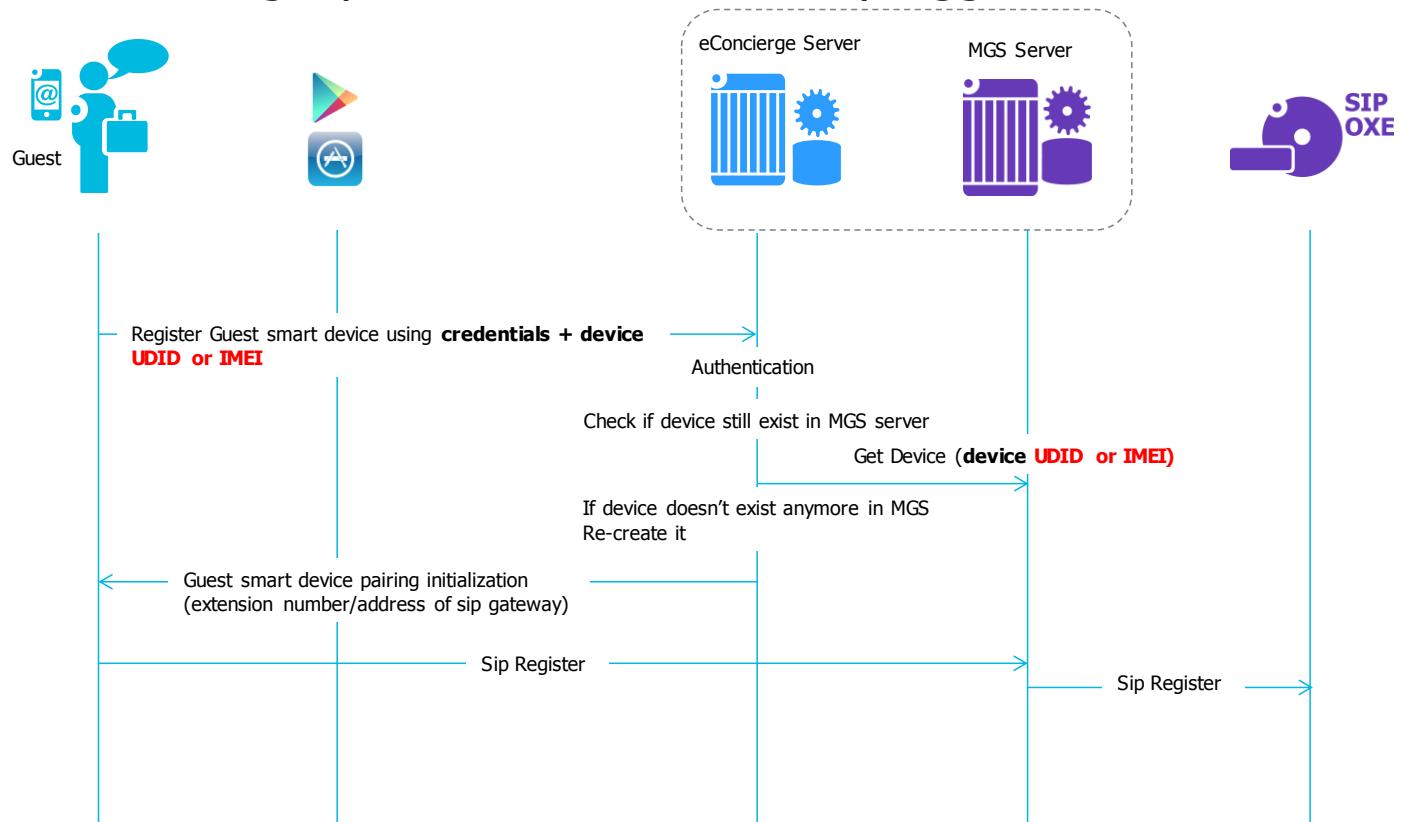
### 3.1 Guest check-in



### 3.2 Guest login on client App

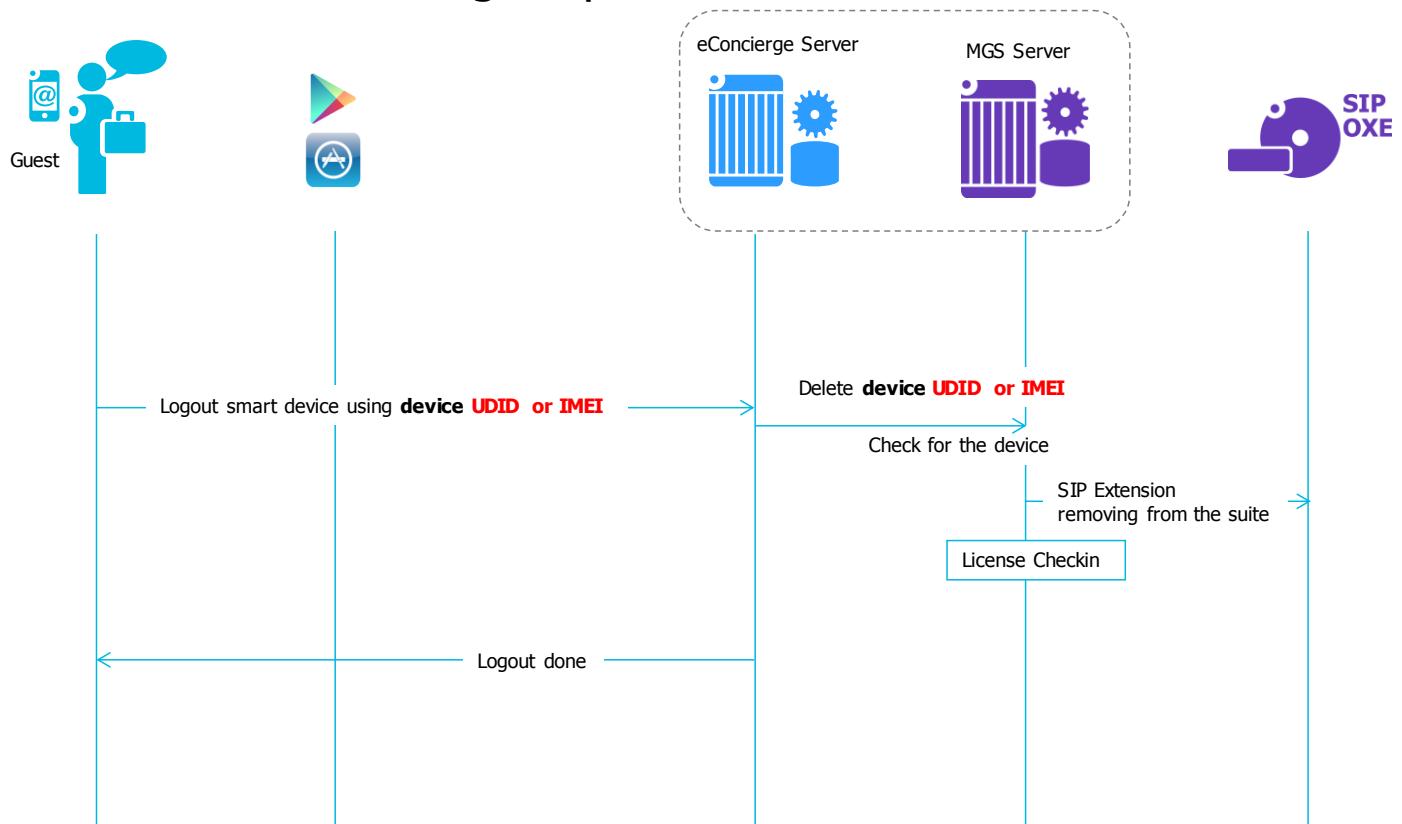


### Guest login phase – device is already logged on MGS

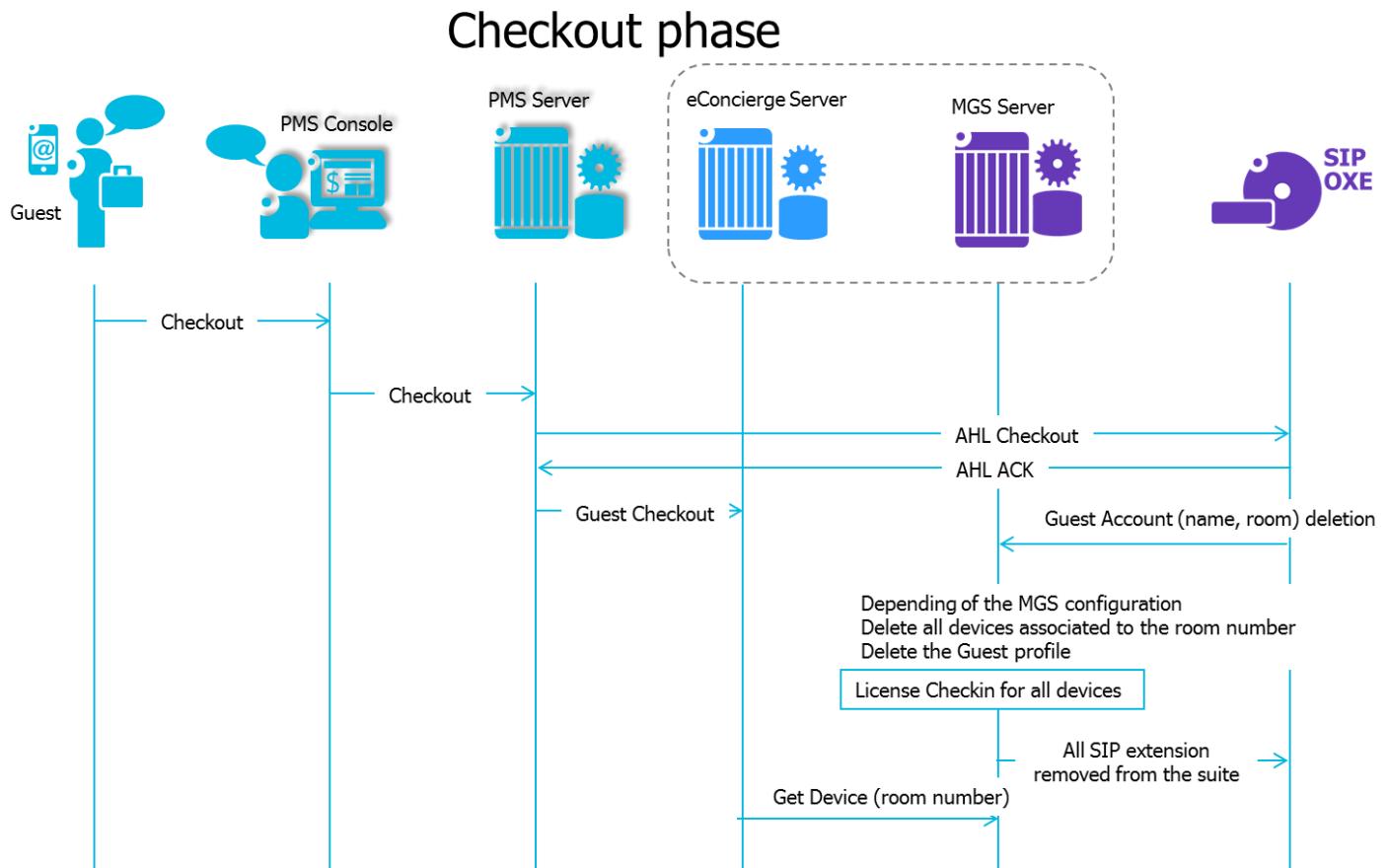


### 3.3 Guest logout

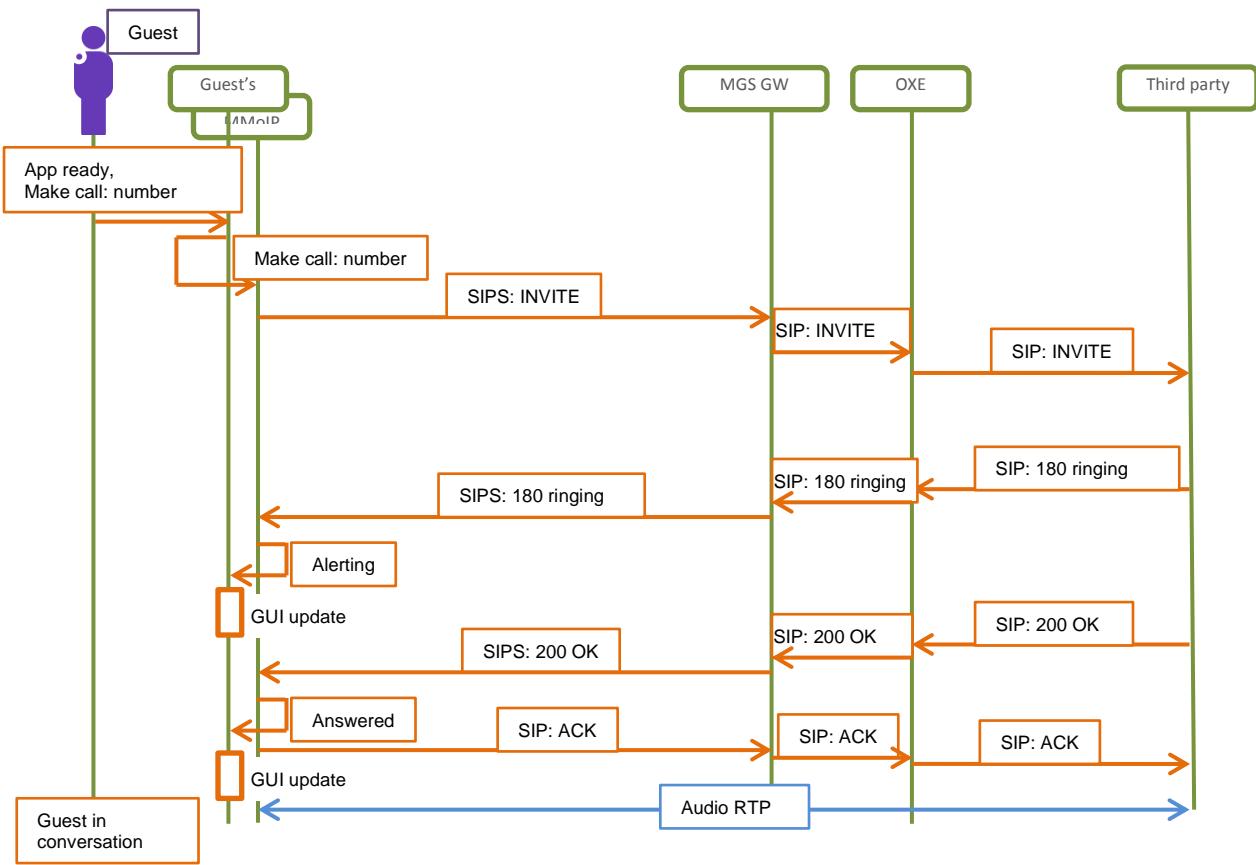
## Guest logout phase from device



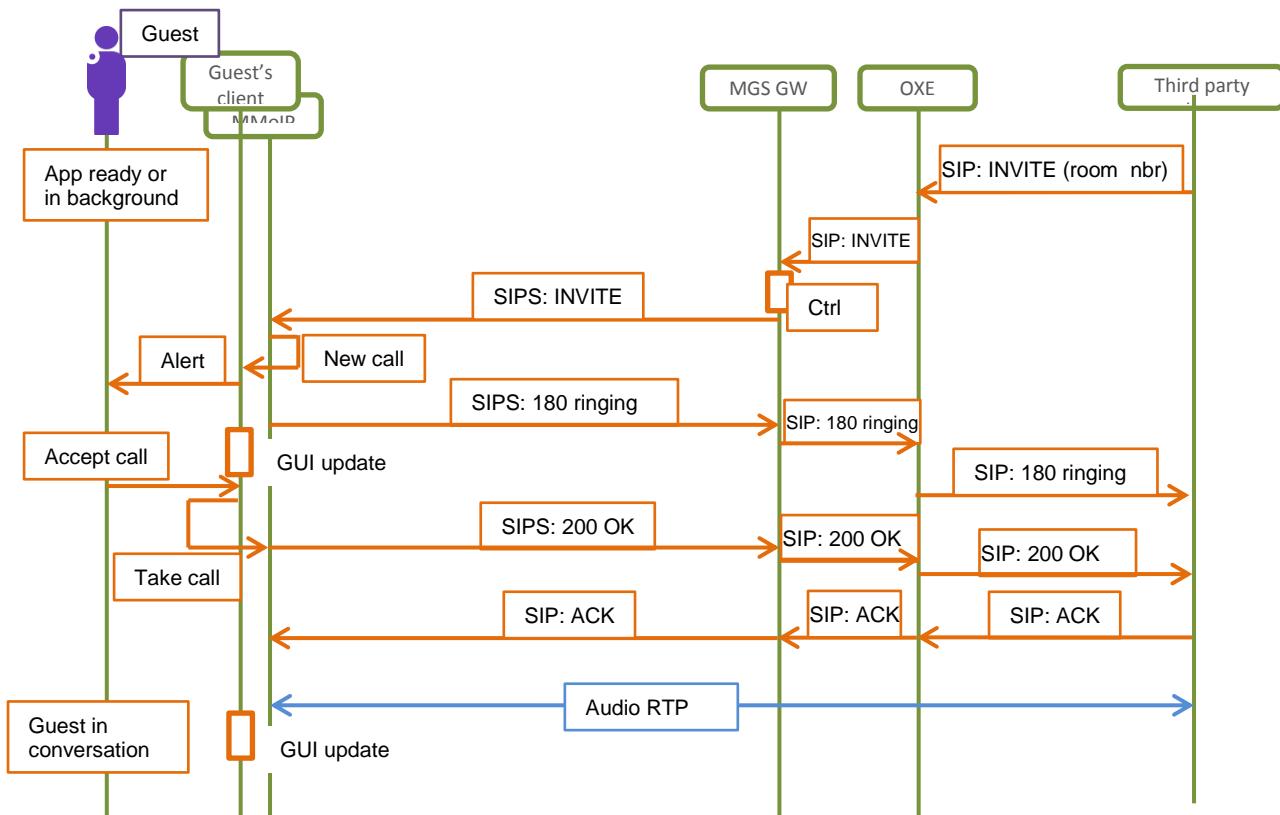
### 3.4 Guest check-out



### 3.5 Guest initiates a call



### 3.6 Guest receives a call



## 4 SDK

---

### 4.1 Overview

The SDK is divided in 2 parts:

**A REST JSON API**, exposed by the MGS Gateway to manage users and access to the communication resources that can be shared with the hotel applications (speed dials, or guest lists for instance)

It provides basic authentication using header authorization. Admin users can call functions. Users without admin rights can only do GET requests

Two mobile SDKs

**iOS : A native C++ API**, delivered as a library to be embedded in the client part of the hotel application.  
Allows device registration/de-registration and VoIP call control

Android : An android Java API, delivered as a .jar file and a native library to be embedded in the client part of the hotel application.

### 4.2 Service gateway part / REST JSON API :

This API is used to:

- Manage the user rights on communication services
- Access directories for "search-by-name feature"

In production, the API should only be made available on port 443 (https if apache configuration has been done).

For testing and development port 8080 may also be used, as in this document.

It provides access to the following objects:

- Device (named “device”)
- Speed dial (named “speedDial”)
- Help messages (see below for detail)
- Guest profile (named “guest”)
- Authentication (named “authentication”)

and Configuration parameters.

These objects are described in the paragraphs below.

When a password is requested in the Device creation it is important to use 4 random digits.

#### 4.2.1 Device

Created on the MGS gateway, when a guest registers his mobile device on the MGS ecosystem.

The MGS service gateway assigns a communication number to the user. This number will be used to register the new device and call it from another hotel device.

For this object, the requests listed below are available:

#### 4.2.1.1 Device creation

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/	
<b>Description</b> Creates a Mobile Guest Softphone Device		
<b>Licence</b> <ul style="list-style-type: none"> <li>Needs a flexlm Licence</li> </ul>		
<b>URL Parameter</b> <ul style="list-style-type: none"> <li>None</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
Device	Device(*)	
<b>Possible Responses</b>		
Code	Return	Description
200	DeviceSipForApi(**)	Returns the device created in the database with sip profile
400	Bad request	Request invalid or device invalid
500	Internal error	Internal Server Error, no phone available, Username exists already, Mcdo allocation, No more license
<b>(*) Device type</b>		
Property	Type	
deviceid	String	Mandatory
password	String (4 digits)	Mandatory
userName	String	Mandatory
roomNumber	String	Mandatory
<b>(**) DeviceForApi type</b>		
Property	Type	
id	Long	Provided by MGS Gateway
deviceid	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
userName	String	Provided by MGS Gateway
roomNumber	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway
sipProfile	SipProfile(***)	Provided by MGS Gateway
<b>(***)SipProfile type</b>		
Property	Type	
domain	String	Provided by MGS Gateway
userId	String	Provided by MGS Gateway
displayName	String	Provided by MGS Gateway
outboundProxy	String	Provided by MGS Gateway
loginName	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
contactUri	String	Provided by MGS Gateway
sbcPublicIpAddress	String	Optional, provided by MGS Gateway
sbcPort	String	Optional, provided by MGS Gateway

#### 4.2.1.2 Device Update

<b>Request Type:</b> PUT	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/	
Updates a Mobile Guest Softphone Device		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>Needs a flexlm Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>None</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
Device	Device(*)	
<b>Response</b>		
Code	Return	Description
200		
400	Bad request	The device format is invalid, request is invalid
500	Internal error	Internal Server Error, no phone available, Username already exists, Mcdo allocation, No more license
<b>(*)Device type</b>		
Property	Type	
id	Integer	Mandatory
deviceid	String	Mandatory
password	String (4 digits)	Mandatory
userName	String	Mandatory
roomNumber	String	Mandatory
sipPhoneNumber	String	Optional

#### 4.2.1.3 deviceReady

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/{deviceid}/{deviceid}/notification/deviceReady	
Create a notification to a Mobile Guest Softphone Device		
<b>Description</b>		
Tell the Mobile Guest Softphone server, the device is ready to receive a VOIP call		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>DeviceId</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200		
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Internal error	Internal Server Error.

#### 4.2.1.4 Token Registration

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/deviceld/{deviceld}/notification/register/{mode}																					
Create a Mobile Guest Softphone Device's Token																						
<b>Description</b> Tell the Mobile Guest Softphone server, the device has a new token to store.																						
<b>URL Parameter</b> <ul style="list-style-type: none"> <li>• Deviceld</li> <li>• Mode: voip or push</li> </ul>																						
<b>Request Parameter</b> <table border="1"> <thead> <tr> <th>Parameter</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Token in JSON format notificationToken: {"type":"ios", "bundleId":"applicationBunble Id", "token":"Token given by Apple after register to Pushkit framework" }</td><td></td><td></td></tr> <tr> <td>Example: {   "notificationToken":   {"type":"ios",   "bundleId":"com.com.alcatel-lucent.byod.ALE   ", "token":"e947333d345004bf37c49a76a00064b95313717c79ec07cbaab0904b819cefc3"} }</td><td></td><td></td></tr> </tbody> </table>		Parameter	Type	Description	Token in JSON format notificationToken: {"type":"ios", "bundleId":"applicationBunble Id", "token":"Token given by Apple after register to Pushkit framework" }			Example: { "notificationToken": {"type":"ios", "bundleId":"com.com.alcatel-lucent.byod.ALE ", "token":"e947333d345004bf37c49a76a00064b95313717c79ec07cbaab0904b819cefc3"} }														
Parameter	Type	Description																				
Token in JSON format notificationToken: {"type":"ios", "bundleId":"applicationBunble Id", "token":"Token given by Apple after register to Pushkit framework" }																						
Example: { "notificationToken": {"type":"ios", "bundleId":"com.com.alcatel-lucent.byod.ALE ", "token":"e947333d345004bf37c49a76a00064b95313717c79ec07cbaab0904b819cefc3"} }																						
<b>Response</b> <table border="1"> <thead> <tr> <th>Code</th><th>Return</th><th>Description</th></tr> </thead> <tbody> <tr> <td>200</td><td>Created</td><td></td></tr> <tr> <td>201</td><td>Unauthorized</td><td></td></tr> <tr> <td>401</td><td>Forbidden</td><td></td></tr> <tr> <td>403</td><td>Not Found</td><td></td></tr> <tr> <td>404</td><td>Internal error</td><td>Internal Server Error.</td></tr> <tr> <td>500</td><td></td><td></td></tr> </tbody> </table>		Code	Return	Description	200	Created		201	Unauthorized		401	Forbidden		403	Not Found		404	Internal error	Internal Server Error.	500		
Code	Return	Description																				
200	Created																					
201	Unauthorized																					
401	Forbidden																					
403	Not Found																					
404	Internal error	Internal Server Error.																				
500																						

#### 4.2.1.5 Get device list

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/									
Gets the list of all devices created in the MGS Gateway server										
<b>URL Parameter</b> <ul style="list-style-type: none"> <li>• None</li> </ul>										
<b>Request Parameters</b> <table border="1"> <thead> <tr> <th>Parameter</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td></tr> </tbody> </table>		Parameter	Type	Description						
Parameter	Type	Description								
<b>Response</b> <table border="1"> <thead> <tr> <th>Code</th><th>Return</th><th>Description</th></tr> </thead> <tbody> <tr> <td>200</td><td>DeviceForApi[](*)</td><td></td></tr> <tr> <td>404</td><td>Not found</td><td></td></tr> </tbody> </table>		Code	Return	Description	200	DeviceForApi[](*)		404	Not found	
Code	Return	Description								
200	DeviceForApi[](*)									
404	Not found									

**(\*\*) DeviceForApi type**

Property	Type	
id	Long	Provided by MGS Gateway
deviceID	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
userName	String	Provided by MGS Gateway
roomNumber	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway

**4.2.1.6 Get device by database id**

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/{id}	
Get a specific device by its database id		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String id : the database id of the device</li> </ul>		
<b>Request Parameters</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	DeviceForApi(*)	
404	Not found	Requested device was not found
<b>(*)DeviceForApi type</b>		
Property	Type	
id	Int	Provided by MGS Gateway
deviceID	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
userName	String	Provided by MGS Gateway
roomNumber	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway

**4.2.1.7 Get all devices associated to a room**

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/roomNumber/{roomNumber}	
Get all devices associated to a room		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String roomNumber : the room number of the device</li> </ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	DeviceForApi[](*)	
404	Not found	Requested device was not found
<b>(*)DeviceForApi type</b>		

Property	Type	
id	Int	Provided by MGS Gateway
deviceid	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
roomNumber	String	Provided by MGS Gateway
userName	String	Provided by MGS Gateway
sipPhoneNumberMD5	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway

#### 4.2.1.8 Get device by device id

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/device/deviceid/{deviceid}	
Get a registered device by the device id		
<b>URL parameter</b>		
<ul style="list-style-type: none"> <li>String deviceid : the identifier of the device</li> </ul>		
<b>Request parameters</b>		
Parameter	Type	description
<b>Response</b>		
Code	Return	Description
200	DeviceForApi(*)	
404	Not found	Requested device was not found
<b>(*)DeviceForApi type</b>		
Property	Type	
id	Int	Provided by MGS Gateway
deviceid	String	Provided by MGS Gateway
Password	String	Provided by MGS Gateway
roomNumber	String	Provided by MGS Gateway
username	String	Provided by MGS Gateway
sipPhoneNumberMD5	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway

#### 4.2.1.9 Delete device by database id

Request Type: DELETE	URL: IPADDRESS:8080/BYOD/api/device/{id}	
<b>Description</b>		
Delete a device by its database id		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>Returns a flexlm Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>the id of the device</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200		

#### 4.2.1.10 Delete devices by room number

<b>Request Type:</b> DELETE	<b>URL:</b> IPADDRESS:8080/BYOD/api/device/roomNumber/{roomNumber}	
Delete all devices attached to a room		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>Returns as many licenses as there are deleted devices</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>roomNumber : the room number of the device</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200		
404	Not found	Requested device was not found

#### 4.2.2 Speed Dials

Speed dials are shortcuts to access to hotel services like Restaurant, Reception, Bar, Spa...

Special speed dials are EMERGENCY and VOICE\_MAIL, they can so be displayed in various maners. Take care to have only 1 EMERGENCY and 1 VOICE\_MAIL.

##### 4.2.2.1 Speed Dial creation

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/speedDial/	
Creates a speed dial		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
speedDial	SpeedDial(*)	
<b>Response</b>		
Code	Return	Description
200		speedDial is created
400	Bad request	Format of speed dial or request is invalid
500	Internal error	Internal Server Error, speed dial already exists
<b>(*)Speed dial type</b>		
Property	Type	
name	String	Mandatory
number	String	Mandatory
externalNumber	String	optional (if empty number is used)
typeNumber	String	Mandatory (could be either SPEED_DIAL, EMERGENCY or VOICE_MAIL)
Picture	Byte[]	Optional (Icon displayed into speedDial button)

##### 4.2.2.2 Speed Dial update

<b>Request Type:</b> PUT	<b>URL:</b> IPADDRESS:8080/BYOD/api/speedDial/	
update a speed dial		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
speedDial	SpeedDial(*)	
<b>Response</b>		
Code	Return	Description
200		speedDial is created
400	Bad request	Format of speed dial or request is invalid
500	Internal error	Internal Server Error
<b>(*)Speed dial type</b>		



Property	Type	
name	String	Mandatory
number	String	Mandatory
externalNumber	String	optional (if empty number is used)
typeNumber	String	Mandatory (could be either EMERGENCY or SPEED_DIAL)
Picture	Byte[]	Optional (Icon displayed into speedDial button)

#### 4.2.2.3 Get Speed Dial list

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/speedDial/														
Get all speed dials															
<b>Licence</b>															
<ul style="list-style-type: none"> <li>No Licence</li> </ul>															
<b>URL Parameter</b>															
<ul style="list-style-type: none"> <li>none</li> </ul>															
<b>Request</b> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description											
Parameter	Type	Description													
<b>Response</b> <table border="1"> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Speeddial[](*)</td> <td></td> </tr> <tr> <td>500</td> <td>Internal error</td> <td>Internal Server Error</td> </tr> </tbody> </table>		Code	Return	Description	200	Speeddial[](*)		500	Internal error	Internal Server Error					
Code	Return	Description													
200	Speeddial[](*)														
500	Internal error	Internal Server Error													
<b>(*)Speed dial type</b> <table border="1"> <thead> <tr> <th>Property</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>String</td> <td>Mandatory</td> </tr> <tr> <td>number</td> <td>String</td> <td>Mandatory</td> </tr> <tr> <td>externalNumber</td> <td>String</td> <td>Mandatory</td> </tr> <tr> <td>typeNumber</td> <td>String</td> <td>Mandatory (could be either SPEED_DIAL, EMERGENCY or VOICE_MAIL)</td> </tr> </tbody> </table>		Property	Type	name	String	Mandatory	number	String	Mandatory	externalNumber	String	Mandatory	typeNumber	String	Mandatory (could be either SPEED_DIAL, EMERGENCY or VOICE_MAIL)
Property	Type														
name	String	Mandatory													
number	String	Mandatory													
externalNumber	String	Mandatory													
typeNumber	String	Mandatory (could be either SPEED_DIAL, EMERGENCY or VOICE_MAIL)													

#### 4.2.2.4 Get Speed Dial by name

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/speedDial/{name}	
Get a speed dial by its name		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String name : the name of the speeddial(*)</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	description
Name	String	
<b>Response</b>		
Code	Return	Description
200	speed dial(*)	
404	Not found	
500	Internal error	Internal Server Error
<b>(*)Speed dial type</b>		
Property	Type	
name	String	Mandatory
number	String	Mandatory
externalNumber	String	Mandatory
typeNumber	String	Mandatory (could be either SPEED_DIAL, EMERGENCY or VOICE_MAIL)

#### 4.2.2.5 Get Speed Dial image by name

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/speedDial/getIcon/{name}	
Get a speed dial icon from its name		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String name : the name of the speeddial</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	description
Name	String	
<b>Response</b>		
Code	Return	Description
200	Image	Icon of the speed dial
404	Not found	

#### 4.2.2.6 Get property isExternalCallAllowed

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/speedDial/isExternalCallAllowed	
Get the property isExternalCallAllowed		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>None</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	description
<b>Response</b>		
Code	Return	Description
200	Boolean	Is external call allowed

#### 4.2.2.7 Delete speed dial by name

Request Type: DELETE	URL: IPADDRESS:8080/BYOD/api/speedDial/{name}	
Delete a speed dial by its name		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String name : the name of the speeddial(*)</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	description
<b>Response</b>		
Code	Return	Description
200		
404		Not found
500		Internal error
<b>(*)Speed dial type</b>		
Property	Type	
name	String	Mandatory
number	String	Mandatory
externalNumber	String	Mandatory
typeNumber	String	Mandatory (could be either SPEED_DIAL, EMERGENCY or VOICE_MAIL)

#### 4.2.3 Help message

##### 4.2.3.1 Get help message by language

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/help/{language}
Get a help message in a specific language (2 char). If the language doesn't exist, return the help message in English. If English help doesn't exist, return 404	
Existing languages available on MGS gateway are: EN (english), FR (french), SV (swedish).	
This list of languages can be extended in future versions.	

**Licence**

- No Licence

**URL Parameter**

- Language : 2 characters

**Request Parameter**

Parameter	Type	Description
-----------	------	-------------

**Response**

Code	Return	Description
200	Help object	Help displayed into the mobile application

404	Not found
-----	-----------

**Help type**

Property	Type	
codeLanguage	String	Mandatory (language code on 2 digit)
Text	String	Mandatory (displayed as help in mobile app)
Enable	boolean	Is this language enable and selectable into the BYOD Server app

#### 4.2.4 Guest Profile

A guest profile is created when a customer checks in at the hotel desk. A guest profile contains some information about the customer (room number, name etc). If a guest profile already exists for the room number, the old one is deleted.

##### 4.2.4.1 Guest profile creation

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/	
Create a guest. A device is created when a customer, who has a guest profile, asks to connect his own device to the hotel network.		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
Guest	Guest(*)	
<b>Response</b>		
Code	Return	Description
200		The guest account is created
400		The format of guest or request is invalid
500		Internal Server Error, guest already exists
<b>(*)Guest type</b>		
Property	Type	
Id	Long	Provided by app
roomNumber	String	Mandatory
lastName	String	Mandatory
firstName	String	Mandatory

#### 4.2.4.2 Create SIP device

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/createGuestDevice/	
Create a SIP device A device is created when a customer, who has a guest profile, asks to connect his own device to the hotel network.		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• Need Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Request</b>		
<b>Parameter</b>	<b>Type</b>	<b>Description</b>
guestDeviceProfile	GuestDeviceProfile(*)	
<b>Response</b>		
<b>Code</b>	<b>Return</b>	<b>Description</b>
200	DeviceForApi(**)	The guest device is created
404		Data not found
500		Internal Server Error, guest already exists
503		No license available
<b>(*)GuestDeviceProfile type</b>		
<b>Property</b>	<b>Type</b>	
guestProfileId	Long	Mandatory
roomNumber	String	Mandatory
displayName	String	Mandatory
userName	String	Mandatory
password	String (4 digits)	Mandatory
<b>(**)DeviceForApi type</b>		
<b>Property</b>	<b>Type</b>	
id	Integer	Provided by MGS Gateway
username	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
displayName	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
sipProfile	SipProfile(***)	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway
<b>(***)SipProfile type</b>		
<b>Property</b>	<b>Type</b>	
domain	String	Provided by MGS Gateway
userId	String	Provided by MGS Gateway
displayName	String	Provided by MGS Gateway
outboundProxy	String	Provided by MGS Gateway
loginName	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
contactUri	String	Provided by MGS Gateway
sbcPublicIpAddress	String	Optional, provided by MGS Gateway
sbcPort	String	Optional, provided by MGS Gateway

#### 4.2.4.3 Create guest device

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/createGuestDeviceWithId/	
Create a sip device. A device is created when a customer, who has a guest profile but no QRCode, asks to connect his own device to the hotel network.		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• Need Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
deviceProfile	DeviceProfile(*)	
<b>Response</b>		
Code	Return	Description
200	DeviceForApi(**)	The guest device is created
404		Data not found
500		Internal Server Error, guest already exists
503		No more license available
<b>(*)DeviceProfile type</b>		
Property	Type	
guestProfileId	Long	Mandatory
roomNumber	String	Mandatory
userName	String	Mandatory
deviceid	String	Mandatory
password	String (4 digits)	Mandatory
<b>(**)DeviceForApi type</b>		
Property	Type	
Id	Integer	Provided by MGS Gateway
deviceid	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
userName	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
sipProfile	SipProfile(***)	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway
<b>(***)SipProfile type</b>		
Property	Type	
domain	String	Provided by MGS Gateway
userId	String	Provided by MGS Gateway
displayName	String	Provided by MGS Gateway
outboundProxy	String	Provided by MGS Gateway
loginName	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
contactUri	String	Provided by MGS Gateway
sbcPublicIpAddress	String	Optional, provided by MGS Gateway
sbcPort	String	Optional, provided by MGS Gateway

#### 4.2.4.4 Update SIP device

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/createGuestDeviceWithoutId/	
Update a SIP device		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• Need Licence</li> </ul>		
<b>Authorization</b>		
<ul style="list-style-type: none"> <li>• Registration without QRCode must be allowed</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
deviceProfile	DeviceProfile(*)	
<b>Response</b>		
Code	Return	Description
200	DeviceForApi(**)	The guest device is created
404		Data not found
500		Internal Server Error, guest already exists
500		Internal Server Error, No registration authorized without QRCode
503		No more license available
(*)DeviceProfile type		
Property	Type	
roomNumber	String	Mandatory
userName	String	Mandatory
deviceld	String	Mandatory
password	String (4 digits)	Mandatory
(**)DeviceForApi type		
Property	Type	
Id	Integer	Provided by MGS Gateway
deviceld	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
userName	String	Provided by MGS Gateway
sipPhoneNumber	String	Provided by MGS Gateway
sipProfile	sipProfile(***)	Provided by MGS Gateway
gatewayPublicIpAddress	String	Optional, provided by MGS Gateway
(***)SipProfile type		
Property	Type	
domain	String	Provided by MGS Gateway
userId	String	Provided by MGS Gateway
displayName	String	Provided by MGS Gateway
outboundProxy	String	Provided by MGS Gateway
loginName	String	Provided by MGS Gateway
password	String	Provided by MGS Gateway
contactUri	String	Provided by MGS Gateway
sbcPublicIpAddress	String	Optional, provided by MGS Gateway
sbcPort	String	Optional, provided by MGS Gateway

#### 4.2.4.5 Update Guest details

<b>Request Type:</b> PUT	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/	
update a guest details		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Request Parameter</b>		
<b>Parameter</b>	<b>Type</b>	
guest	Guest(*)	
<b>Response</b>		
<b>Code</b>	<b>Return</b>	
200	The guest is updated	
400	Format of guest or request is invalid	
500	Internal Server Error	
<b>(*)Guest type</b>		
<b>Property</b>	<b>Type</b>	
Id	Long	Provided by app
roomNumber	String	Mandatory
lastName	String	Mandatory
firstName	String	Mandatory

#### 4.2.4.6 Get all guest profiles

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/	
Get all guests		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> <li></li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Request Parameter</b>		
<b>Parameter</b>	<b>Type</b>	
<b>Response</b>		
<b>Code</b>	<b>Return</b>	
200	Guest[](*)	
500	Internal Server Error	
<b>(*)Guest type (into a list)</b>		
<b>Property</b>	<b>Type</b>	
Id	Long	Provided by App
roomNumber	String	Mandatory
lastName	String	Mandatory
firstName	String	Mandatory

#### 4.2.4.7 Get guest profile from id

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/{id}	
Get guest according to his id		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>Long id : guest id</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	guest	
404		Data not found
500		Internal Server Error
<b>Guest type</b>		
Property	Type	
Id	Long	Provided by app
roomNumber	String	Mandatory
lastName	String	Mandatory
firstName	String	Mandatory

#### 4.2.4.8 Get Guest profile from room number

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/roomNumber/{roomNumber}	
Get guest according to their room number		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>String roomNumber : the guest room number</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	List of the guests	
404		Data not found
500		Internal Server Error
<b>Guest type</b>		
Property	Type	
Id	Long	Provided by app
roomNumber	String	Mandatory
lastName	String	Mandatory
firstName	String	Mandatory

#### 4.2.4.9 Get registration without ID

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/registrationWithoutId/			
Get registration without Id authorization				
Returns Boolean that allow administrator to register guest without ID				
<b>Parameter</b>				
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>				
<b>URL Parameter</b>				
<ul style="list-style-type: none"> <li>• none</li> </ul>				
<b>Licence</b>				
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>				
<b>Request</b> <table> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description
Parameter	Type	Description		
<b>Response</b> <table> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> </table>		Code	Return	Description
Code	Return	Description		
<table> <tbody> <tr> <td>200</td> <td>True or false</td> <td></td> </tr> </tbody> </table>		200	True or false	
200	True or false			

#### 4.2.4.10 Delete Guest

<b>Request Type:</b> DELETE	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/{number}									
Delete a guest										
<b>URL Parameter</b>										
<ul style="list-style-type: none"> <li>• long number : the id of the guest</li> </ul>										
<b>Licence</b>										
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>										
<b>Request Parameter</b> <table> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description						
Parameter	Type	Description								
<b>Response</b> <table> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> </table>		Code	Return	Description						
Code	Return	Description								
<table> <tbody> <tr> <td>200</td> <td></td> <td></td> </tr> <tr> <td>404</td> <td></td> <td>Not found</td> </tr> <tr> <td>500</td> <td></td> <td>Internal error</td> </tr> </tbody> </table>		200			404		Not found	500		Internal error
200										
404		Not found								
500		Internal error								

#### 4.2.4.11 Delete guests from room number

<b>Request Type:</b> DELETE	<b>URL:</b> IPADDRESS:8080/BYOD/api/guest/roomNumber/{roomNumber}	
Delete a guest by roomNumber		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String roomNumber : the roomNumber of the guest</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200		
404		Not found
500		Internal error

#### 4.2.5 Configuration properties

There are some properties in configuration which values are provided for softphone application.

NB: Only user with IT role can invoke this service.

##### 4.2.5.1 Get properties

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/property	
GetAll Returns a		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	Properties values	See Update properties

##### 4.2.5.2 Update properties

<b>Request Type:</b> PUT	<b>URL:</b> IPADDRESS:8080/BYOD/api/property/
-----------------------------	--

update properties

#### Licence

- No Licence

#### URL Parameter

- none

#### Request Parameter

Parameter	Type	Description
properties	Property(*)	

#### Response

Code	Return	Description
200		Properties are updated
400		Format of properties or request is invalid
500		Internal Server Error

#### (\*)Property type

Property	Type	
sbcEnable	String	Optional
interRoomPrefix	String	Optional
gatewayPublicIpAddress	String	Optional
externalSpeedDialEnable	String	Optional
hotelChainName	String	Optional
OutgoingPrefix	String	Optional
registrationWithoutIdAllowed	String	Optional
hotelName	String	Optional
mobileToMobileCallAllowed	String	Optional
mailSubject	String	Optional
sbcPort	String	Optional
internationalPrefix	String	Optional
mailContent	String	Optional
maximumDevicesPerRoom	String	Optional
fromAddress	String	Optional
mailServerAddress	String	Optional
mailAccountLogin	String	Optional
sbcPublicIpAddress	String	Optional
displayFavoriteTab	String	Optional
mailAuthenticationRequired	String	Optional
mailAccountPassword	String	Optional
checkinOutMcdoEnable	String	Optional
checkinOutMcdolInitEnable	String	Optional

#### 4.2.5.3 Get mobileToMobileCallAllowed property

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/property/getMobileToMobileCallAllowed
Get mobileToMobileCallAllowed property Returns Boolean pproperty which allows mobile phone to call other mobile phones managed by BYOD.	
<b>Parameter</b>	
<ul style="list-style-type: none"><li>• No parameter</li></ul>	
<b>URL Parameter</b>	

- none

**Licence**

- No Licence

**Request**

Parameter	Type	Description
-----------	------	-------------

**Response**

Code	Return	Description
200	True or false	

#### 4.2.5.4 Get displayFavoriteTab property

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/property/getDisplayFavoriteTab	
Get displayFavoriteTab property		
Returns Boolean property which allows mobile phone to display view with favorites numbers.		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	True or false	

#### 4.2.5.5 Get interRoomPrefix property

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/property/getInterRoomPrefix	
Get interRoomPrefix property		
Returns String property which is the prefix to allow call between rooms.		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Licence</b>		

- No Licence

Request Parameter	Type	Description
Response Code	Return String	Description
200		The inter rooms prefix

#### 4.2.5.6 Get outgoingPrefix property

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/property/getOutgoingPrefix	
Get outgoingPrefix property Returns String property which is the prefix to allow outgoing call.		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>		
Request Parameter	Type	Description
Response Code	Return String	Description
200		The outgoing prefix

#### 4.2.5.7 Get internationalPrefix property

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/property/getInternationalPrefix	
Get internationalPrefix property Returns String property which is the prefix to allow international outgoing call.		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>		

Request Parameter	Type	Description
<b>Response Code</b>		
200	String	The international prefix

#### 4.2.5.8 Get HotelName property

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/property/getHotelName						
Get HotelName property Returns String property which is the name of the hotel.							
<b>Parameter</b>							
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>							
<b>URL Parameter</b>							
<ul style="list-style-type: none"> <li>• none</li> </ul>							
<b>Licence</b>							
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>							
<b>Request</b> <table> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description			
Parameter	Type	Description					
<b>Response</b> <table> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>String</td> <td>The name of the hotel</td> </tr> </tbody> </table>		Code	Return	Description	200	String	The name of the hotel
Code	Return	Description					
200	String	The name of the hotel					

#### 4.2.5.9 Get HotelChainName property

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/property/getHotelChainName						
Get HotelChainName property Returns String property which is the name of the hotel chain.							
<b>Parameter</b>							
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>							
<b>URL Parameter</b>							
<ul style="list-style-type: none"> <li>• none</li> </ul>							
<b>Licence</b>							
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>							
<b>Request</b> <table> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description			
Parameter	Type	Description					
<b>Response</b> <table> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>String</td> <td>The name of the hotel chain</td> </tr> </tbody> </table>		Code	Return	Description	200	String	The name of the hotel chain
Code	Return	Description					
200	String	The name of the hotel chain					

#### **4.2.5.10 Get Version property**

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/property/getVersion	
Get Version property Returns String property which is the version of the MGS server.		
<b>Parameter</b>		
<ul style="list-style-type: none"><li>• No parameter</li></ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"><li>• none</li></ul>		
<b>Licence</b>		
<ul style="list-style-type: none"><li>• No Licence</li></ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	String	The version of the MGS server

#### **4.2.6 Authentication**

This service is used to create authentication account .

NB: Only user with IT role can invoke this service.

##### **4.2.6.1 Get all authentications**

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/authentication
GetAll Returns all authentication accounts	
<b>Parameter</b>	

- No parameter

**URL Parameter**

- none

**Licence**

- No Licence

**Request**

Parameter	Type	Description
-----------	------	-------------

**Response**

Code	Return	Description
200	Authentications values	See Create Authentication

#### 4.2.6.2 Create authentication

Request Type: POST	URL: IPADDRESS:8080/BYOD/api/authentication/
Create authentication	
<b>Licence</b>	
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>	
<b>URL Parameter</b>	
<ul style="list-style-type: none"> <li>• none</li> </ul>	
<b>Request Parameter</b>	
Parameter authentication	Type Authentication(*)
<b>Response</b>	
Code 200 400 500	Return Authentication is created Format of authentication or request is invalid Internal Server Error
(*)Property type	
Property username password emailaddress role	Type String String String String
	Mandatory Mandatory Mandatory Mandatory value (USER,ADMIN)

#### 4.2.6.3 Update authentication

Request Type: PUT	URL: IPADDRESS:8080/BYOD/api/authentication/
----------------------	---

## Update authentication

### Licence

- No Licence

### URL Parameter

- none

### Request Parameter

Parameter	Type	Description
authentication	Authentication(*)	

### Response

Code	Return	Description
200		Authentication is updated
400		Format of authentication or request is invalid
500		Internal Server Error

### (\*)Property type

Property	Type	
username	String	Mandatory
password	String	Mandatory
emailaddress	String	Mandatory
role	String	Mandatory Value (USER,ADMIN)

### 4.2.6.4 Get authentication

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/authentication/{username}	
Get Returns authentication account by username		
<b>Parameter</b> <ul style="list-style-type: none"><li>• String username : the username of the authentication</li></ul>		
<b>URL Parameter</b> <ul style="list-style-type: none"><li>• none</li></ul>		
<b>Licence</b> <ul style="list-style-type: none"><li>• No Licence</li></ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	Authentication value	See Create Authentication

#### 4.2.6.5 Delete authentication

<b>Request Type:</b> DELETE	<b>URL:</b> IPADDRESS:8080/BYOD/api/authentication/{username}	
Delete an authentication by username		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>String username : the username of the authentication</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
<b>Request Parameter</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200		
404		Not found
500		Internal error

#### 4.2.7 Room Phone association

This service is used to manage room phone association.

NB: Only user with IT role can invoke this service.

##### 4.2.7.1 Get all room phone associations

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/roomPhone	
GetAll Returns all room phone associations		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>none</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>No Licence</li> </ul>		
Request Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	Room Phone association values	See Create Room Phone Association

##### 4.2.7.2 Create room phone association

<b>Request Type:</b> POST	<b>URL:</b> IPADDRESS:8080/BYOD/api/roomPhone/
------------------------------	---

## Create room phone association

### Licence

- No Licence

### URL Parameter

- none

### Request Parameter

Parameter	Type	Description
roomPhone	RoomPhone(*)	

### Response

Code	Return	Description
200		Room phone association is created
400		Format of association or request is invalid
500		Internal Server Error

### (\*)RoomPhone type

Property	Type	
roomNumber	String	Mandatory
phoneNumber	String	Mandatory

### 4.2.7.3 Update room phone association

Request Type:	URL:												
PUT IPADDRESS:8080/BYOD/api/roomPhone/													
Update room phone association													
<b>Licence</b>													
<ul style="list-style-type: none"><li>• No Licence</li></ul>													
<b>URL Parameter</b>													
<ul style="list-style-type: none"><li>• none</li></ul>													
<b>Request Parameter</b>													
<table border="1"><thead><tr><th>Parameter</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>roomPhone</td><td>RoomPhone(*)</td><td></td></tr></tbody></table>		Parameter	Type	Description	roomPhone	RoomPhone(*)							
Parameter	Type	Description											
roomPhone	RoomPhone(*)												
<b>Response</b>													
<table border="1"><thead><tr><th>Code</th><th>Return</th><th>Description</th></tr></thead><tbody><tr><td>200</td><td></td><td>Room phone association is created</td></tr><tr><td>400</td><td></td><td>Format of association or request is invalid</td></tr><tr><td>500</td><td></td><td>Internal Server Error</td></tr></tbody></table>		Code	Return	Description	200		Room phone association is created	400		Format of association or request is invalid	500		Internal Server Error
Code	Return	Description											
200		Room phone association is created											
400		Format of association or request is invalid											
500		Internal Server Error											
<b>(*)RoomPhone type</b>													
<table border="1"><thead><tr><th>Property</th><th>Type</th><th></th></tr></thead><tbody><tr><td>roomNumber</td><td>String</td><td>Mandatory</td></tr><tr><td>phoneNumber</td><td>String</td><td>Mandatory</td></tr></tbody></table>		Property	Type		roomNumber	String	Mandatory	phoneNumber	String	Mandatory			
Property	Type												
roomNumber	String	Mandatory											
phoneNumber	String	Mandatory											

#### 4.2.7.4 Get a room phone association by room number

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/roomPhone/{roomNumber}						
GetByRoomNumber Returns room phone association							
<b>Parameter</b>							
<ul style="list-style-type: none"> <li>• roomNumber : the room number of the association</li> </ul>							
<b>URL Parameter</b>							
<ul style="list-style-type: none"> <li>• none</li> </ul>							
<b>Licence</b>							
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>							
<b>Request</b> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description			
Parameter	Type	Description					
<b>Response</b> <table border="1"> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Room Phone association value</td> <td>See Create Room Phone Association</td> </tr> </tbody> </table>		Code	Return	Description	200	Room Phone association value	See Create Room Phone Association
Code	Return	Description					
200	Room Phone association value	See Create Room Phone Association					

#### 4.2.7.5 Get a room phone association by phone number

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/roomPhone/phoneNumber/{phoneNumber}						
GetByPhoneNumber Returns room phone association							
<b>Parameter</b>							
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>							
<b>URL Parameter</b>							
<ul style="list-style-type: none"> <li>• none</li> </ul>							
<b>Licence</b>							
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>							
<b>Request</b> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description			
Parameter	Type	Description					
<b>Response</b> <table border="1"> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Room Phone association value</td> <td>See Create Room Phone Association</td> </tr> </tbody> </table>		Code	Return	Description	200	Room Phone association value	See Create Room Phone Association
Code	Return	Description					
200	Room Phone association value	See Create Room Phone Association					

#### 4.2.7.6 Delete room phone association

<b>Request Type:</b> DELETE	<b>URL:</b> IPADDRESS:8080/BYOD/api/roomPhone/{roomNumber}
Delete a room phone association	
<b>URL Parameter</b>	
<ul style="list-style-type: none"> <li>• String roomNumber : the roomNumber of the association</li> </ul>	
<b>Licence</b>	

- No Licence

#### Request Parameter

Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200		
404		Not found
500		Internal error

#### 4.2.8 Resource

This service is used to manage sip resource.

NB: Only user with IT role can invoke this service.

##### 4.2.8.1 Get all resources

Request Type: GET	URL: IPADDRESS:8080/BYOD/api/resource	
GetAll Returns all resources		
<b>Parameter</b>		
<ul style="list-style-type: none"> <li>• No parameter</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>		
<b>Request</b>		
Parameter	Type	Description
<b>Response</b>		
Code	Return	Description
200	Resources values	See Create Resource
<b>(*)Resource type</b>		
Property	Type	
number	String	Mandatory
roomNumber	String	Mandatory
attached	Boolean	Mandatory
allocated	Boolean	Mandatory
AllocatedTime	Integer	Mandatory
releasedTime	Integer	Mandatory
releasedDate	Date	Mandatory
allocatedDate	Date	Mandatory

##### 4.2.8.2 Create resource

Request Type:	URL:
---------------	------

<b>POST</b>	IPADDRESS:8080/BYOD/api/resource/	
Create resource		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Request Parameter</b>		
<b>Parameter</b>	<b>Type</b>	<b>Description</b>
resource	Resource(*)	
<b>Response</b>		
<b>Code</b>	<b>Return</b>	<b>Description</b>
200		Resource is created
400		Format of resource or request is invalid
500		Internal Server Error
<b>(*)Resource type</b>		
<b>Property</b>	<b>Type</b>	
number	String	Mandatory
roomNumber	String	Mandatory

#### 4.2.8.3 Update resource

<b>Request Type:</b> PUT	<b>URL:</b> IPADDRESS:8080/BYOD/api/resource/	
Update resource		
<b>Licence</b>		
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>		
<b>URL Parameter</b>		
<ul style="list-style-type: none"> <li>• none</li> </ul>		
<b>Request Parameter</b>		
<b>Parameter</b>	<b>Type</b>	<b>Description</b>
resource	Resource(*)	
<b>Response</b>		
<b>Code</b>	<b>Return</b>	<b>Description</b>
200		Resource is updated
400		Format of resource or request is invalid
500		Internal Server Error
<b>(*)Resource type</b>		
<b>Property</b>	<b>Type</b>	
number	String	Mandatory
roomNumber	String	Mandatory

#### 4.2.8.4 Get a resource

<b>Request Type:</b> GET	<b>URL:</b> IPADDRESS:8080/BYOD/api/resource/{number}																										
Get Returns resource																											
<b>Parameter</b>																											
<ul style="list-style-type: none"> <li>• number : the number of the resource</li> </ul>																											
<b>URL Parameter</b>																											
<ul style="list-style-type: none"> <li>• none</li> </ul>																											
<b>Licence</b>																											
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>																											
<b>Request</b> <table> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description																							
Parameter	Type	Description																									
<b>Response</b> <table> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Resource value</td> <td>See Create Resource</td> </tr> </tbody> </table>		Code	Return	Description	200	Resource value	See Create Resource																				
Code	Return	Description																									
200	Resource value	See Create Resource																									
<b>(*)Resource type</b> <table> <thead> <tr> <th>Property</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>number</td> <td>String</td> <td>Mandatory</td> </tr> <tr> <td>roomNumber</td> <td>String</td> <td>Mandatory</td> </tr> <tr> <td>attached</td> <td>Boolean</td> <td>Mandatory</td> </tr> <tr> <td>allocated</td> <td>Boolean</td> <td>Mandatory</td> </tr> <tr> <td>AllocatedTime</td> <td>Integer</td> <td>Mandatory</td> </tr> <tr> <td>releasedTime</td> <td>Integer</td> <td>Mandatory</td> </tr> <tr> <td>releasedDate</td> <td>Date</td> <td>Mandatory</td> </tr> <tr> <td>allocatedDate</td> <td>Date</td> <td>Mandatory</td> </tr> </tbody> </table>		Property	Type	number	String	Mandatory	roomNumber	String	Mandatory	attached	Boolean	Mandatory	allocated	Boolean	Mandatory	AllocatedTime	Integer	Mandatory	releasedTime	Integer	Mandatory	releasedDate	Date	Mandatory	allocatedDate	Date	Mandatory
Property	Type																										
number	String	Mandatory																									
roomNumber	String	Mandatory																									
attached	Boolean	Mandatory																									
allocated	Boolean	Mandatory																									
AllocatedTime	Integer	Mandatory																									
releasedTime	Integer	Mandatory																									
releasedDate	Date	Mandatory																									
allocatedDate	Date	Mandatory																									

#### 4.2.8.5 Delete resource

<b>Request Type:</b> DELETE	<b>URL:</b> IPADDRESS:8080/BYOD/api/resource/{number}												
Delete a resource													
<b>URL Parameter</b>													
<ul style="list-style-type: none"> <li>• String number : the number of the resource</li> </ul>													
<b>Licence</b>													
<ul style="list-style-type: none"> <li>• No Licence</li> </ul>													
<b>Request Parameter</b> <table> <thead> <tr> <th>Parameter</th> <th>Type</th> <th>Description</th> </tr> </thead> </table>		Parameter	Type	Description									
Parameter	Type	Description											
<b>Response</b> <table> <thead> <tr> <th>Code</th> <th>Return</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>200</td> <td></td> <td>Not found</td> </tr> <tr> <td>404</td> <td></td> <td>Internal error</td> </tr> <tr> <td>500</td> <td></td> <td></td> </tr> </tbody> </table>		Code	Return	Description	200		Not found	404		Internal error	500		
Code	Return	Description											
200		Not found											
404		Internal error											
500													

## 4.3 Sample of rest/json request :

### 4.3.1 Create a guest :

URL : <https://172.27.134.60/BYOD/api/guest/>

type :POST

Header parameter :

- content-Type:application/json
- Authorization : Basic xxxxxxxxx

Parameter :

```
{  
    "roomNumber": "100",  
    "lastName": "Smith",  
    "firstName": "John"  
}
```

Status : 200 ok

Return : null

### 4.3.2 Get all guests

URL : <https://172.27.134.60/BYOD/api/guest/>

type :GET

Header parameter :

- content-Type:application/json
- Authorization : Basic xxxxxxxxx

Status : 200 ok

Return :

```
[  
    {  
        "id": 8356,  
        "roomNumber": "103",  
        "firstName": "",  
        "lastName": "DUPONT de NEMOURS"  
    },
```

```
{
    "id": 8930,
    "roomNumber": "101",
    "firstName": "",
    "lastName": "iPhone"
},
{
    "id": 14209,
    "roomNumber": "100",
    "firstName": "John",
    "lastName": "Smith"
}
]
```

#### **4.3.3 Delete a guest according to his room number**

URL : <https://172.27.134.60/BYOD/api/guest/roomNumber/105>

type :DELETE

Header parameter :

- content-Type:application/json
- Authorization : Basic xxxxxxxxx

Status : 200 ok

Return : null

## 4.4 Mobile application part

The API is used to:

- configure the SIP device part of the mobile's application
- register a new communication device of the ALE communication system
- unregister a device
- raise notifications to users
- manage calls

It is available for 2 operating systems, iOS and Android. The differences between these two implementations are described in the following chapters.

### 4.4.1 MGS API integration into an IOS application

Here is a brief description of how to integrate the MGS API into an iOS application.

*New implementation for iOS10 due to the exclusive usage of APNs (Apple Push Notification service) for VOIP applications.*

---

#### 4.4.1.1 Prerequisites

The client application must run on:

- IOS 9 or later
- Devices : iPhone 5 or later, iPad 3 or later , iPad mini 2 or later

It must be linked with:

- Frameworks :
  - AVFoundation
  - CFNetwork
  - Security
  - AudioToolbox
  - CoreMedia
  - CoreVideo
  - OpenGLES
  - SystemConfiguration
- Libraries :
  - libresolv
  - libiconv
  - libxml2.2
  - libSqualeByod.a      -> Squale Byod library. (Universal Architecture,arm + x86)

#### 4.4.1.2 Packaging

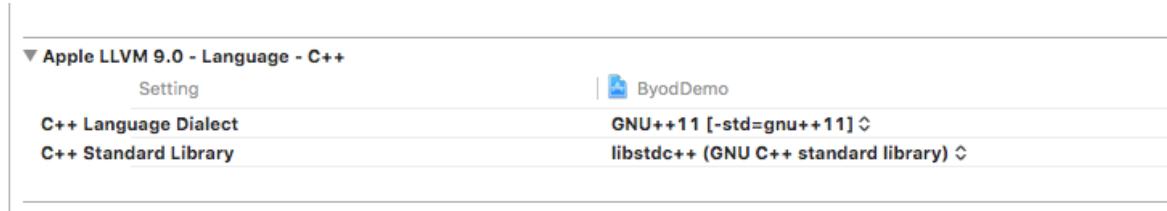
The iOS version of the API, available as a zip archive, contains the following files :

ByodDemo.zip	→ “Byod Demo” sample application source code
headers\SqualeServiceByod.h	→ MGS client API C++ header file
resources\log4cxx-ios.xml	→ MGS client Log file configuration
resources\ua.cnf	→ MGS client configuration file
resources\amusic.wav	→ MGS client ringback tone
resources\ring.wav	→ MGS client ringing tone
libSqualeByod.a	→ SIP and Multimedia library
docs\	→ documents repository

#### 4.4.1.3 Client application creation

Each client application needs to go through the following preliminary initialisation steps:

1. Audio initialization
2. Manage Token with APNs
3. Application receives a VOIP notification from APNs
4. Background mode handling
5. Resource files integration
6. Xcode project must be built with libstdc++, add build options as below, and link with libstdc++



They are described in more detail in the following chapters.

#### 4.4.1.3.1 Application Capabilities

1. In *appli-Info.plist*, item *background mode*, add:
  - App provide Voice Over IP service
  - Remote notification
  - App plays Audio
2. Enable *appli-Info.plist*, item *push notification*
3. In *appli-Info.plist*, add item *Privacy - Microphone Usage Description*.

#### 4.4.1.3.2 Audio Initialization

At init time, an audio session must be activated by the application.

```
- (void) initAudioSession{
    appAVAudioSession* audioSession = [AVAudioSession sharedInstance];
    [audioSession setCategory: AVAudioSessionCategoryPlayAndRecord error: nil];
    [audioSession setActive: YES error: nil];
}
```

#### 4.4.1.3.3 Manage Token with APNs

At init time, the application has to request for a VOIP token to APNs by using method:

- Code example to connect to pushkit

```
/ Register for VoIP notifications
- (void) voipRegistration {
    dispatch_queue_t mainQueue = dispatch_get_main_queue() ;
    // Create a push registry object
    PKPushRegistry * voipRegistry = [[PKPushRegistry alloc]

initWithQueue: mainQueue];
    // Set the registry's delegate to self
    voipRegistry.delegate = self;
    // Set the push type to VoIP
    voipRegistry.desiredPushTypes = [NSSet
setWithObject:PKPushTypeVoIP];
```

```
}
```

The VOIP token is sent back by APNs to the application via a callback

- Implement call back for token error message and notification:

```
- (void)pushRegistry:(PKPushRegistry *)registry  
didUpdatePushCredentials: (PKPushCredentials *)credentials forType:(NSString *)type  
{  
....}
```

*Notes: the APNs request from mobile can work only if port 5223 is open to internet (see prerequisites document)*

---

When application gets its VOIP token, it has to send it to the MGS server by calling the REST API

**POST /api/device/deviceld/{deviceld}/notification/register/voip**

Header parameter :

```
- content-Type:application/json  
notificationToken:  
{  
    "type":"ios",  
    "bundleId":"applicationBunble Id",  
    "token":"Token given by Apple after register to Pushkit framework"  
}
```

4.4.1.3.4 Application receives a VOIP notification from APNs

Each time a call happens on the server side, the server sends a VOIP notification to the APNs

The application is then wake up by the VOIP notification

Content of the notification:

```
{  "aps": {  
        "title": "MGS Incoming Call"  
    },  
    "fromNumber" : "2134",  
    "fromUser" : "user 2134"  
    "date": "1977-04-22T01:00:00-05:00" (in ISO 8601)  
}
```

Method to get the notification information in the Application:

```
// Handle incoming pushes  
- (void)pushRegistry:(PKPushRegistry *)registry  
didReceiveIncomingPushWithPayload:(PKPushPayload *)payload forType:(NSString *)type {  
  
    // Process the received push  
    NSLog(@"pushRegistry didReceiveIncomingPushWithPayload  
bundleId=%@", [[NSBundle mainBundle] bundleIdentifier]);
```

Regarding the data coverage (3G or Wifi), the integrated application VOIP stack may be available or not to handle the call, it is mandatory to request the squeal interface

Notes: If squale is not started, initialization procedure has to be done

1. `self.squaleService->checkReachability`
  - a. The goal of this request is to wake up the sip stack, controls are done to check if the server is reachable or not.
  - b. If the SIP stack can reach the MGS server, this request will finally result to a “`Registerok`” message in the call back (intermediates message can occurred “`registernok`”, “`registering`”...)
2. The link with MGS server is operational
  - a. “`Registerok`” message has been received by the application
  - b. Application has now to say to the server it is ready to receive Sip messages by calling the REST API

**POST /api/device/deviceId/{deviceId}/notification/deviceReady**

#### 4.4.1.3.5 Background mode handling

The application must handle background mode

1. Call `willEnterForeground` when application will enter in foreground mode

```
(void)appWillEnterForeground:(NSNotification *)notification {  
    NSLog(@"appWillEnterForeground") ;  
    if (self.squaleService) {  
        self.squaleService->willEnterForeground() ;  
    }  
}
```

2. Call `willEnterBackground` when application will enter in background mode

```
(void)applicationWillResignActive:(NSNotification *)notification {  
    NSLog(@"applicationWillResignActive") ;  
    if (self.squaleService) {  
        self.squaleService->willEnterBackground() ;  
    }  
}
```

#### 4.4.1.3.6 Resource file integration

The files “Ua.cnf”, “log4cxx-ios.xml”, “ring.wav”and “amusic.wav” must be copied under the path  
“`$AppliHome/Documents`”

## **4.4.2 MGS API integration into an Android application**

Here is a brief description of how to integrate the MGS API into an iOS application.

### **4.4.2.1 Prerequisites**

The client application must run on:

Android 4.2.2 or later

Devices tested on Nexus 4, Nexus 5, Samsung Galaxy S3, Samsung Galaxy S4, Samsung Galaxy S5

It must be linked with :

Libs :	squale-byod-service.jar
Libs\armeabi :	libsquale-service.so (arm architecture)

### **4.4.2.2 Packaging**

The Android version of the API, available as a zip archive, contains the following files :

- ByodDemo.apk → “Byod Demo” sample in debug mode.
- ByodDemo.zip → “Byod Demo” sample source code.
- README.TXT → Readme file describing the permissions to be added in “AndroidManifest.xml”
- squale-byod-service-javadoc.jar → Javadoc describing the MGS API
- libs\squale-byod-service.jar → SIP and Multimedia library Java implementation
- libs\armeabi\libsquale-service.so → SIP and Multimedia library native implementation.

### **4.4.2.3 Client application creation**

#### **4.4.2.3.1 Foreground mode handling**

In order to keep application alive, declare a service and start it in foreground mode.

Please see the sample application in ByodDemo.zip for an example on how to do this.

#### **4.4.2.3.2 Set the right permissions**

Add the following permissions to “AndroidManifest.xml” :

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.RECORD_AUDIO"></uses-permission>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"></uses-permission>
<uses-permission android:name="android.permission.BLUETOOTH" />
```

#### **4.4.2.3.3 Android 6 Doze mode**

Starting from Android 6.0 (API level 23), Android introduces two power-saving features that extend battery life for users by managing how apps behave when a device is not connected to a power source. *Doze* reduces battery consumption by deferring background CPU and network activity for apps when the device is unused for long periods of time. See [http://developer.android.com/training/monitoring-device-state/doze-standby.html#understand\\_app\\_standby](http://developer.android.com/training/monitoring-device-state/doze-standby.html#understand_app_standby)

MGS Client application has to have a permanent network access. Battery optimization must be disabled for MGS client application.

Users can manually configure the whitelist in **Settings > Battery > Battery Optimization**, or the App can provide ways for MGS app to ask users to whitelist it:

Add permission `android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS` into `AndroidManifest.xml` file.

Add sample code :

```
Intent intent = new Intent();

String packageName = context.getPackageName();

PowerManager pm = (PowerManager) context.getSystemService(Context.POWER_SERVICE);

if (pm.isIgnoringBatteryOptimizations(packageName))

    intent.setAction(Settings.ACTION_IGNORE_BATTERY_OPTIMIZATION_SETTINGS);

else {

    intent.setAction(Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS);

    intent.setData(Uri.parse("package:" + packageName));

}

context.startActivity(intent);
```

#### 4.4.3 Methods

Both mobile operating systems use the methods described in the following chapters. The sequencing is shown in chapter 4.4.6 Flow Diagrams

##### 4.4.3.1 Start Squale Service Instance

This method initializes the MGS VoIP stack and returns the `SqualeService` instance.

It must be called first before anything else.

- IOS :

```
static ISqualeService* ISqualeService::getInstance(ISqualeCallback* pCallback);  
pCallback : callback to be implemented in order to receive events.
```

```
class ISqualeCallback {  
public:  
    // Method called on call event  
    virtual void onSqualeCallEvent( EventCall* sEvent ) {} ;  
    // Method called on stack event  
    virtual void onSqualeInformation( EventInfo * sInfo ) {};  
  
    virtual ~ISqualeCallback() {};  
};
```

- Android :

```
public  
static ISqualeService SqualeServiceFactory::createSqualeService(Context context,  
ISqualeCallback callback)
```

Context : Android application context.

ISqualeCallback : callback to be implemented in order to receive events.

```
public interface ISqualeCallback {  
  
    /**  
     * Event about the squale stack status.  
     * @param stackEvt Stack event.  
     */  
    void onSipphoneInformation(IEventInfo stackEvt);  
    /**  
     * Event about squale calls.  
     * @param callEvt Call event.  
     */  
    void onCallEvent(IEventCall callEvt);  
}
```

*Call back events are described in chapter 4.4.4 Events*

##### 4.4.3.2 Stop Squale Service

This method clears all resources used by the MGS VoIP stack.

It must be called when the application terminates.

- IOS:

```
squaleService->shutdown ()
```

- Android :

```
ISquareService.destroy()
```

Specific stop service code to implement for Android:

```
buttonDestroy.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        final ISquareService squareService =  
        ByodManager.getInstance().getSquareService();  
        Intent intent = new Intent(getActivity(), ByodService.class);  
        getActivity().getApplicationContext().stopService(intent);  
  
        Timer timer = new Timer();  
        timer.schedule(new TimerTask() {  
  
            @Override  
            public void run() {  
                squareService.destroy();  
            }  
        }, 2000);  
    }  
});
```

#### 4.4.3.3 Configure Square Service

The MGS VoIP stacks must be configured before its use.

To do so, the following sequence must be followed:

1. Call `createConfiguration()` method:

```
ISquareConfiguration ISquareService.createConfiguration()
```

Returns a configuration interface to be used to provide SIP parameters.

2. Provide the required parameters. You can use the methods described in chapter 4.2 *Device* to retrieve those parameters.

The server side “SIP profile type” properties match the client side properties like this:

Server side property	Type	Mobile side property
domain	String	CfgSIPDomain:CfgSIPDomainPort
userId	String	CfgUserName
displayName	String	CfgDisplayName
outboundProxy	String	CfgOutboundProxy:CfgOutboundProxyPort
loginName	String	CfgLoginName
password	String	CfgUserPassword

<b>sbcPublicIpAddress</b>	String	SBCAddress (optional : exists if sbc is enabled on server side)
<b>sbcPort</b>	String	SBCPort (optional : exists if sbc is enabled on server side)
<b>sbcUseTls</b>	String	Always true (optional : exists if sbc is enabled on server side)

- SIP username : SIP device user identity

```
ISqualeConfiguration.setCfgUserName(java.lang.String userName)
```

- SIP displayname : SIP device display name

```
.setCfgDisplayName(java.lang.String displayName)
```

- SIP Domain : SIP domain host, this is the SIP interface PBX host (IP address or host name)

```
ISqualeConfiguration.setCfgUserDomain(java.lang.String userDomain)
```

- SIP DomainPort : SIP domain port, this is the SIP interface PBX port

```
ISqualeConfiguration.setCfgUserDomainPort(java.lang.String userDomainPort)
```

- SIP Protocol (udp/tcp/tls) ~~on iOS tcp is mandatory~~ iOS and Android use UDP.

```
ISqualeConfiguration.setCfgDirectProtocol(java.lang.String directProtocol)
```

- SIP OutboundProxy : SIP MGS gateway host (IP address or host name)

```
ISqualeConfiguration.setCfgOutboundProxy(java.lang.String outboundProxy)
```

- SIP OutboundProxyPort : SIP MGS gateway port.

```
ISqualeConfiguration.setCfgOutboundProxyPort(java.lang.String outboundProxyPort)
```

- Optional SIP Login name :

```
ISqualeConfiguration.setCfgLoginName(java.lang.String loginName)
```

- SIP user password :

```
ISqualeConfiguration.setCfgUserPassword(java.lang.String userPassword)
```

- SBC Address :

```
ISqualeConfiguration.setSBCAddress(java.lang.String sbcAddress)
Note: if SBC is not used, set SBC Address to ""
```

- SBC Port :

```
ISqualeConfiguration.setSBCPort(java.lang.String sbcPort)
```

3. Apply the configuration.

```
ISqualeConfiguration ISqualeConfiguration.applyConfiguration()
```

This method applies the parameters to the MSG Stack, which will be registered to the MGS gateway and the PBX.

The mobile will be in registered state, and ready the make/receive calls.

#### 4.4.3.4 Call control

The following methods are used to take control of calls handled on the mobile device

- Make an audio call

```
void ISqualeService.makeCallAudio(java.lang.String callee)  
callee: phone number to be dialled
```

- Accept an incoming audio call or resume a Held call

```
void ISqualeService.takeCallAudio(java.lang.String callId)  
callId: Call identifier received as part of call events, described in chapter 4.4.5.1 MGS >Call  
events
```

- Reject an incoming audio call

```
void ISqualeService.rejectCall(java.lang.String callId)  
callId: Call identifier received as part of call events, described in chapter 4.4.5.1 MGS >Call  
events
```

- Clear an audio call

```
void ISqualeService.clearCall(java.lang.String callId)  
callId: Call identifier received as part of call events, described in chapter 4.4.5.1 MGS >Call  
events
```

- Clear alls current calls

```
void ISqualeService. clearAllCalls()
```

- Hold an audio call

```
void ISqualeService.holdCall(java.lang.String callId)  
callId: Call identifier received as part of call events, described in chapter 4.4.5.1 MGS >Call  
events
```

- Send DTMF codes

```

void ISqualeService.sendDtmf(java.lang.String callId,
                               java.lang.String dtmfCodes)
callId: Call identifier received as part of call events, described in chapter 4.4.5.1 MGS >Call
events

dtmfCodes: DTMF keys to send. Valid keys are :1234567890*#

```

#### **4.4.3.5 Device control**

- Mute/Unmute microphone

```

void ISqualeService.muteMicro(boolean muted)
muted: true: muted, false: unmuted

```

- Mute/Unmute an audio call

```

void ISqualeService.muteAudioCall(java.lang.String activeCallId,
                                   boolean muted)
callId: Call identifier received as part of call events, described in chapter 4.4.5.1 MGS >Call
events

muted: true: muted, false: unmuted

```

- Disable/Enable loud Speaker

```

void ISqualeService.loudSpeaker(boolean enable)
enable: true: enabled, false: disabled

```

#### **4.4.3.6 Miscellaneous**

These functions are typically used to keep the application alive under iOS, as described in chapter 4.4.1.3.5  
*Background mode handling*

- Enable the SIP user by registering the MGS VoIP stack in its domain server.

```
void ISqualeService.registerUser()
```

- Disable the SIP user by unregistering the MGS VoIP stack from its domain server.

```
void ISqualeService.unregisterUser()
```

#### **4.4.4 Events**

The MGS API produces two kinds of events:

- MGS Stack events : EventInfo

- MGS Call events : EventCall

#### **4.4.4.1 MGS Stack events**

These events are sent when the MGS stack status changes (SIP/RTP Ok/Nok), the user is un/registered or a service call failed (makeCallAudio, takeCall, releaseCall..).

Here are the different fields:

- Raw event : raw event in xml format
- Source : SIP or RTP
- OriginValue : STACK, USER, SERVICE
- OriginState : STACK (RTP or SIP) in/out of server, Register Ok/Nok, Unregister Ok/Nok, Service Succeeded or failed (ex makeAudioCall, TakeCall...)
- ReasonSeverity : MAJOR,MINOR,WARNING
- ReasonValue:  
NETWORK,CONFIG,SERVER,INTERNAL,NORMAL,PARAMETER,AUTHENTICATION,REMOTE\_NETWORK,UNKNOWN
- ReasonService : Only if OriginValue is SERVICE (MAKECALL,RELEASE,TAKE...)
- ReasonComment : additional comment about event.

#### 4.4.5 iOS APNs VOIP certificate generation

partners have to generate a VOIP certificate associated to the partner's application BundleId.

the link below explains how to create a VOIP certificate and how to implement the VOIP Push Notification mechanism in the application.

<https://developer.apple.com/library/content/documentation/Performance/Conceptual/EnergyGuide-iOS/OptimizeVoIP.html>

From the VOIP certificate, partners have to generate a .p12 certificate. This .p12 certificate has to be uploaded on the MGS server by an Alcatel-Lucent Enterprise member or by the customer's business partner, as explained in the MGS installation document.

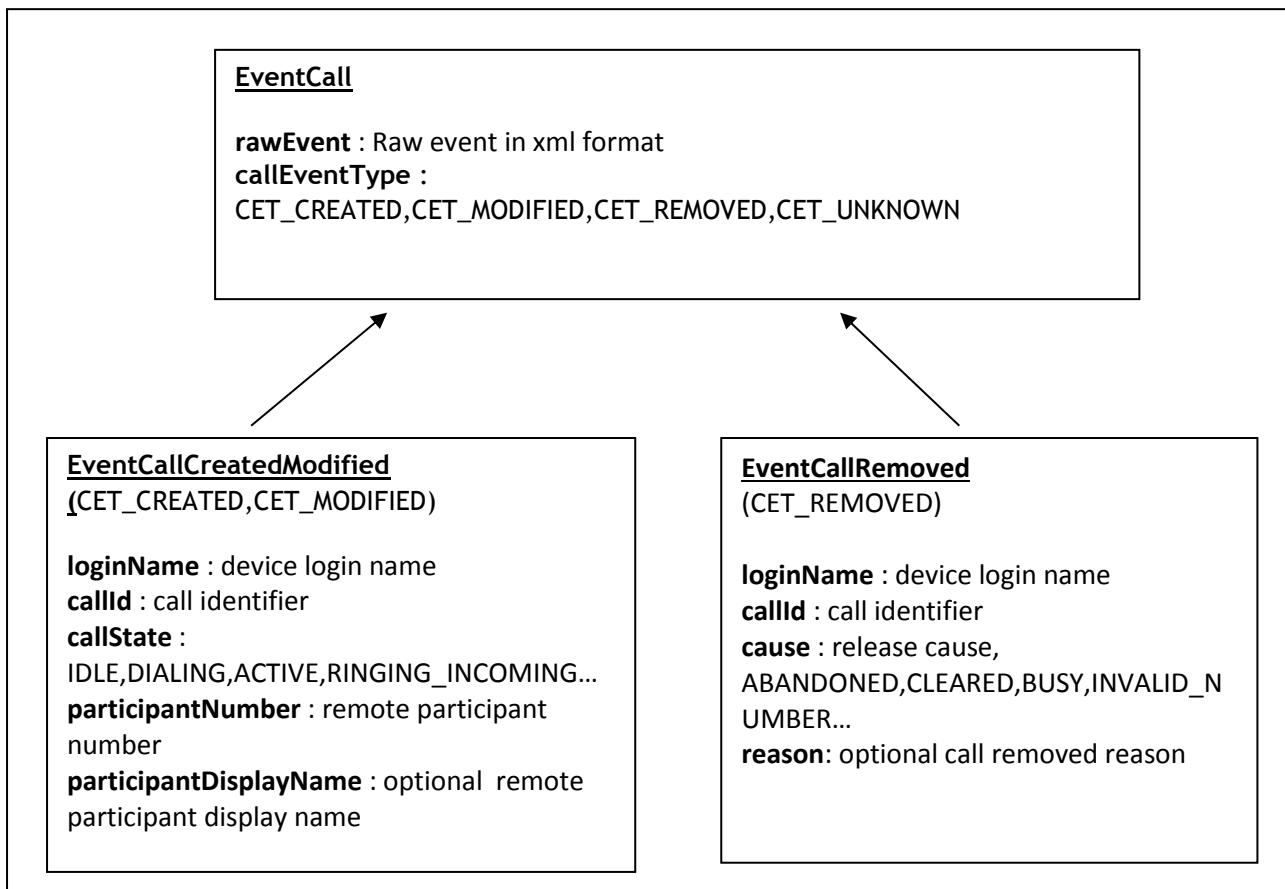
To proceed, following information are needed:

- "bundleId" is the Apple application Bundle id, ex: "com.alcatel-lucent.byod.ALE"
- "Mode" is "voip" or "push", used for APNs notification in voip or push mode.
- "file" is the certificate application name to use APNs. Ex: "ALE\_Certificates.p12"
- "passphrase" passphrase associated to P12 certificate
- "developement" optional parameter for testing purpose on Apple infrastructure (default value is false for production)

##### 4.4.5.1 MGS >Call events

This event is sent when call state change.

Call Event hierarchy

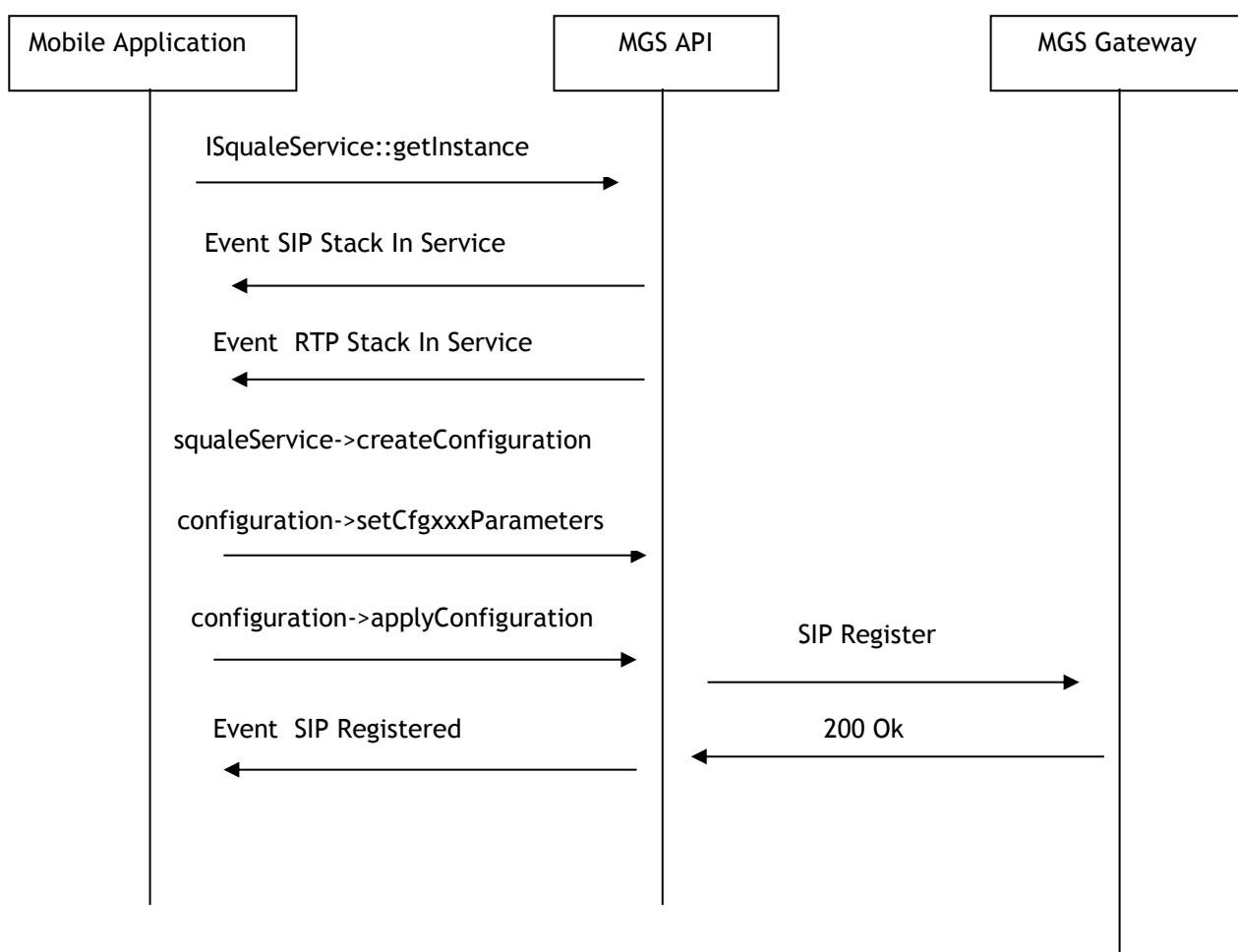


- Client Application has to handle ringing tone when an incoming event is received (RINGING\_INCOMING).
- Client application also has to play a ring back tone on outgoing call (DIALING).

#### 4.4.6 Flow Diagrams

##### 4.4.6.1 Init Mobile application

In order to properly initialise the client application, the following event sequence must be followed:



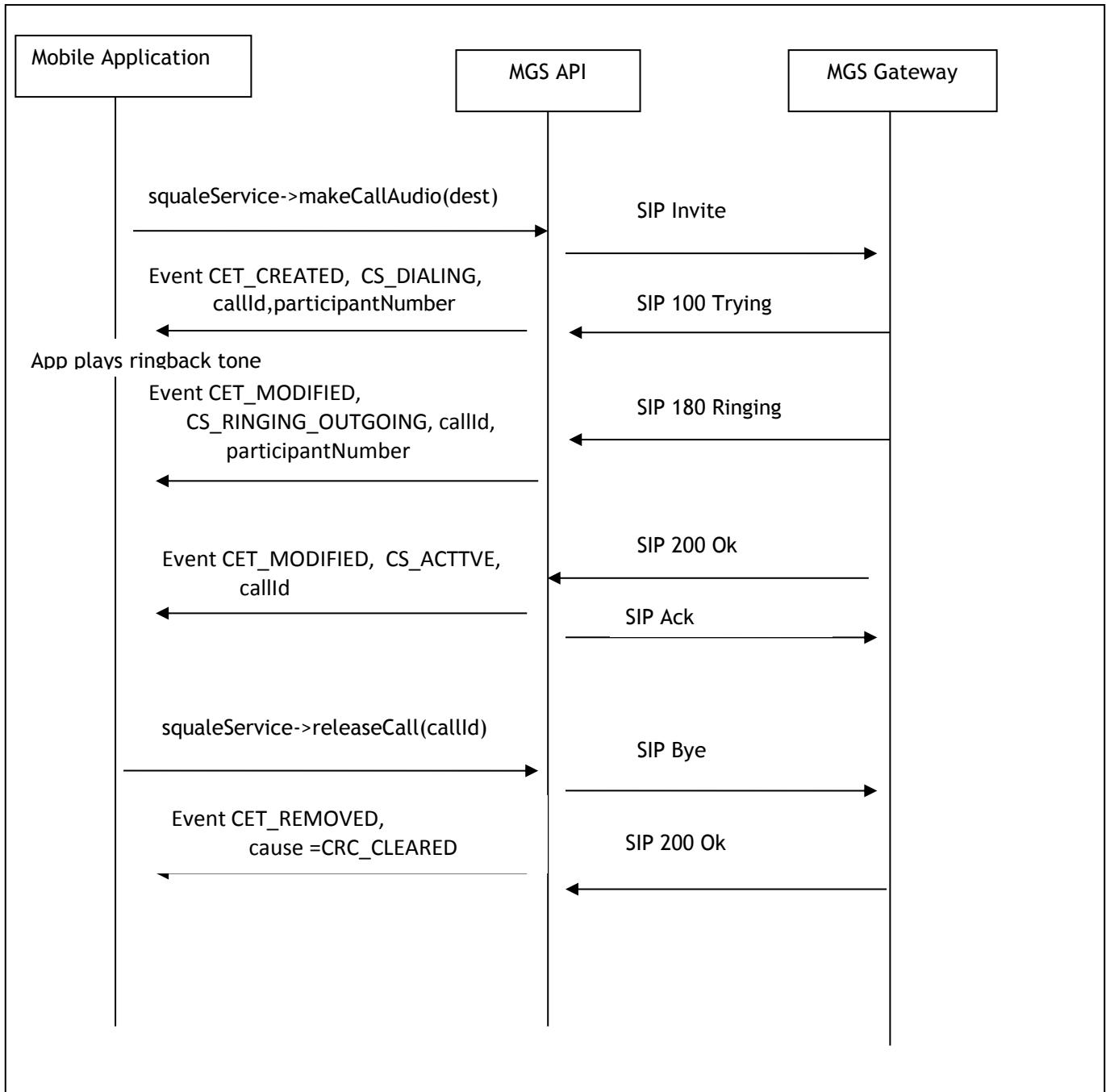
When Event SIP Registered is received , Application is ready to make and receive calls.

Source == SS\_SIP, OriginValue == SOV\_USER , OriginState == SOS\_REGISTER\_OK

Please see the sample application applicable to your operating system in ByodDemo.zip for an example on how to do this.

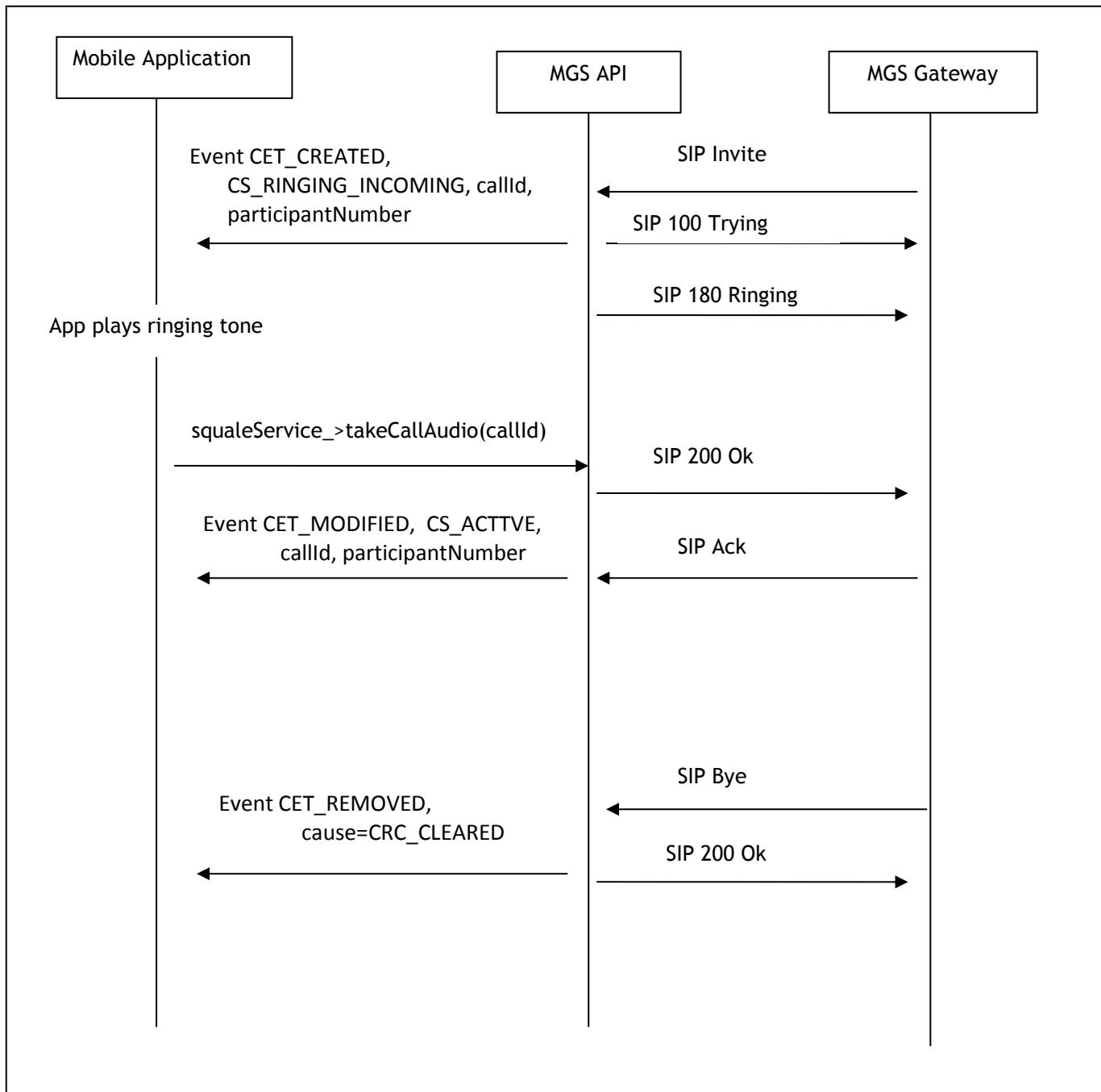
#### 4.4.6.2 Outgoing call

This call flow shows a basic outgoing call example; the call is released by mobile application.



#### 4.4.6.3 Incoming call

This call flow shows a basic incoming call example; the call is released by remote peer.



#### **4.4.6.4 Keep Alive Mechanism and Session refresh**

The Sip stack uses a keep Alive mechanism with the MGS server, by default every 40 seconds the sip stack, embedded in squale service, sends a SIP OPTION message.

In normal case the MGS server answers the OPTION message by a 200 OK

In case of restart of the MGS server or for other reason, the MGS server may reply the OPTION message by a 503, in this case the Sip Stack automatically sends a new SIP REGISTER message.

#### **4.4.6.5 Device deleted -> Option 503 + Register 404 Quit application**

If the device or the guest have been removed from the MGS Server (after an hotel Checkout for instance), the next OPTION message will be answered by a 503. Sip Stack will then send a new REGISTER message.

In this case the MGS Server will answer REGISTER message with a 404 (not found). It is then up to the application to provide a popup to the Guest saying something like “We thank you for your visit.....” and then close the application. (deactivate Squale service on iOS)