BeaconPlus SDK for iOS Documentation

V1.4

September 2015

Revision History

Date	Version	Description	Author
2014-08-01	0. 7	Initial document release	Feby
2014-08-05	1.0	Review points incorporated	Feby
2014-11-03	1.1	Error code updated	Feby
2015-03-17	1.2	Error code updated	Feby
2015-04-17	1.3	Document update	Feby
2015-09-07	1.4	Document update	Feby

Review Details

Date	Version	Author	Reviewer
2014-08-01	0.7	Feby	Nidheesh
2014-08-05	1.0	Feby	Nidheesh
2014-11-03	1.1	Feby	Nidheesh
2015-03-17	1.2	Feby	Nidheesh
2015-04-17	1.3	Feby	Nidheesh
2015-09-07	1.4	Feby	Nidheesh

Table of Contents

1 Introduction

1.1 Purpose of the document

The BeaconPlus SDK documentation provides insight into the BeaconPlus SDK and how it could be integrated to iOS Apps. For detailed API spec and other implementation details, please refer BeaconPlus API Spec (http://pinmicro.com/docs/ios-sdk/index.html).

1.2 Scope of the document

Any developer who would like to develop contextual awareness applications for iOS could use this documentation as the starting point for the same.
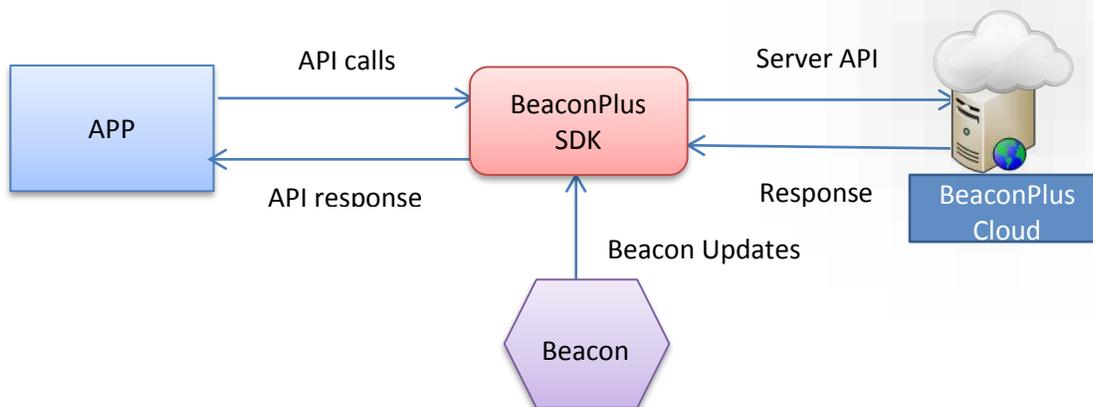
2 Acronyms

The following acronyms are used in this documentation

Acronym	Details
SDK	Software Development Kit
API	Application Programming Interface

3 BeaconPlus SDK Overview

BeaconPlus SDK is offering developers the freedom to develop contextual awareness applications. It acts as a bridge between the Client App and the BeaconPlus cloud services.

The SDK synchronizes with the cloud periodically and thereby holds the right information to project at the right time. It also handles all the internal logic for device detection and notifies the app on the 'Entry' and 'Exit' events for a specific region.

4   Setting up the SDK

This section describes how to setup the SDK for using in an iOS application.

4.1   SDK download details

BeaconPlus SDK can be downloaded from the following url:

https://pinmicro.com/blog/category/beacon-sdk/#downloads

The SDK bundle zip file consists of two parts

- BeaconPlus SDK
- Sample Application source code (GetMeShoes)
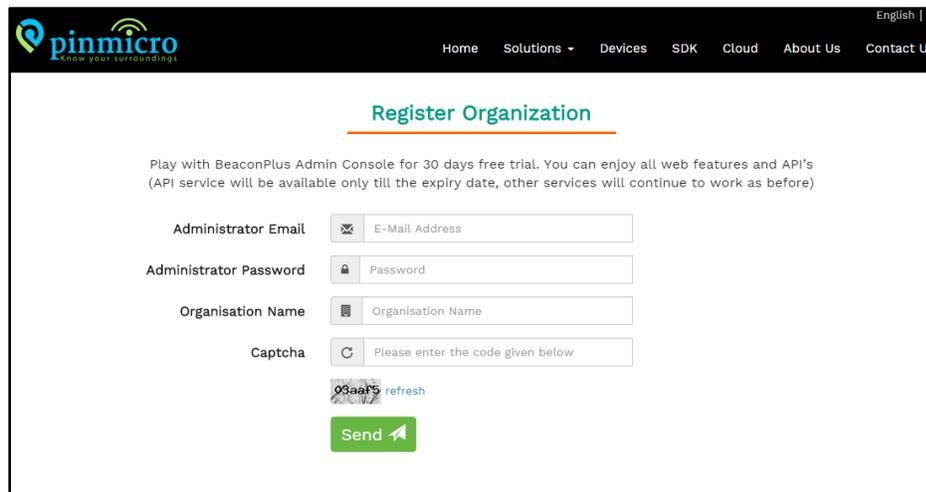
4.2   Requirements

The BeaconPlus SDK uses iBeacon protocol to identify beacons, hence requires a minimum iOS version 7.0. The hardware specifications demands that the SDK runs only on the following devices:

- iPhone 4s or later
- 3rd generation iPad or later
- iPad Mini or later
- 5th generation iPod touch or later

4.3   Registering your organization

The first and foremost step to do before using BeaconPlus SDK is to register your organization. To register, click https://pinmicro.com/register/

The link will redirect the user to the registration page as shown in the below screenshot

Fill up all the necessary details in the fields given, as shown in the above screenshot and once it is done, a confirmation mail will be send to the mail id with which user had registered.

4.4   Getting an Access Secret

An 'Access secret code' is a necessary prerequisite to build any application using BeaconPlus SDK. To get the 'Access secret Code' login to your account (https://beaconplus.pinmicro.com/authenticate/). Go to Settings > Company Details section as shown in the below screenshot.

Click on the 'Send to Mail' button to receive the code in the registered email id.

Copy the Access secret received in mail for using the SDK (shown as XXXX… in the sample mail shown below)

4.5   Setting up BeaconPlus Cloud

Before actual development starts, make sure all the necessary components are configured correctly in BeaconPlus cloud. For e.g.

1. Deployment is created
2. The application under development is added to this deployment
3. Beacons and its contents are added to deployment

For more details please refer the doc from the link below:

https://pinmicro.com/docs/BeaconPlus-Admin-UserManual.pdf

4.6   Integrating BeaconPlus SDK in your Xcode project

a) Create a new project in Xcode, name it, say, 'GetMeShoes'

b) Create a new App Id for the application and obtain the provisioning profiles for the same from Apple Developer site.

c) Copy and paste the SDK bundle 'BeaconPlusSDK' to the project folder

d) Add the SDK and header files to GetMeShoes : GetMeShoes -> 'Add files to GetMeShoes'

pinmicro
Know your surroundings
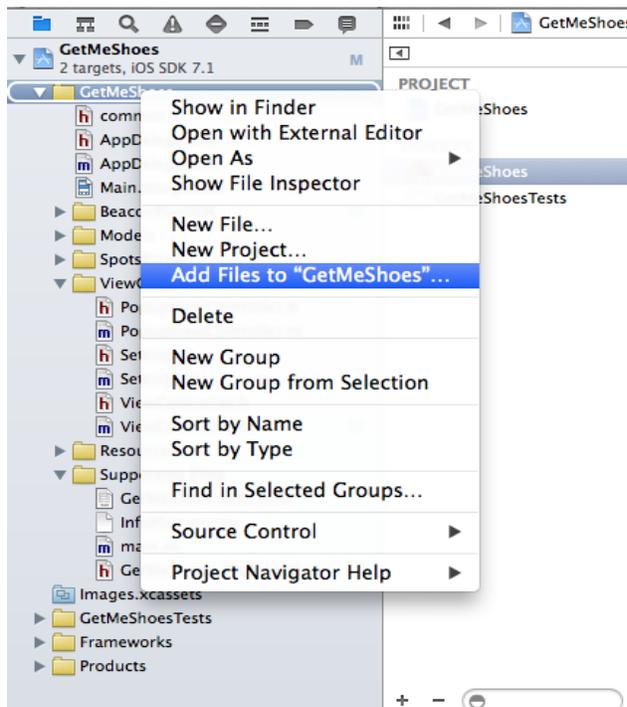
e)   Find and select BeaconPlusSDK
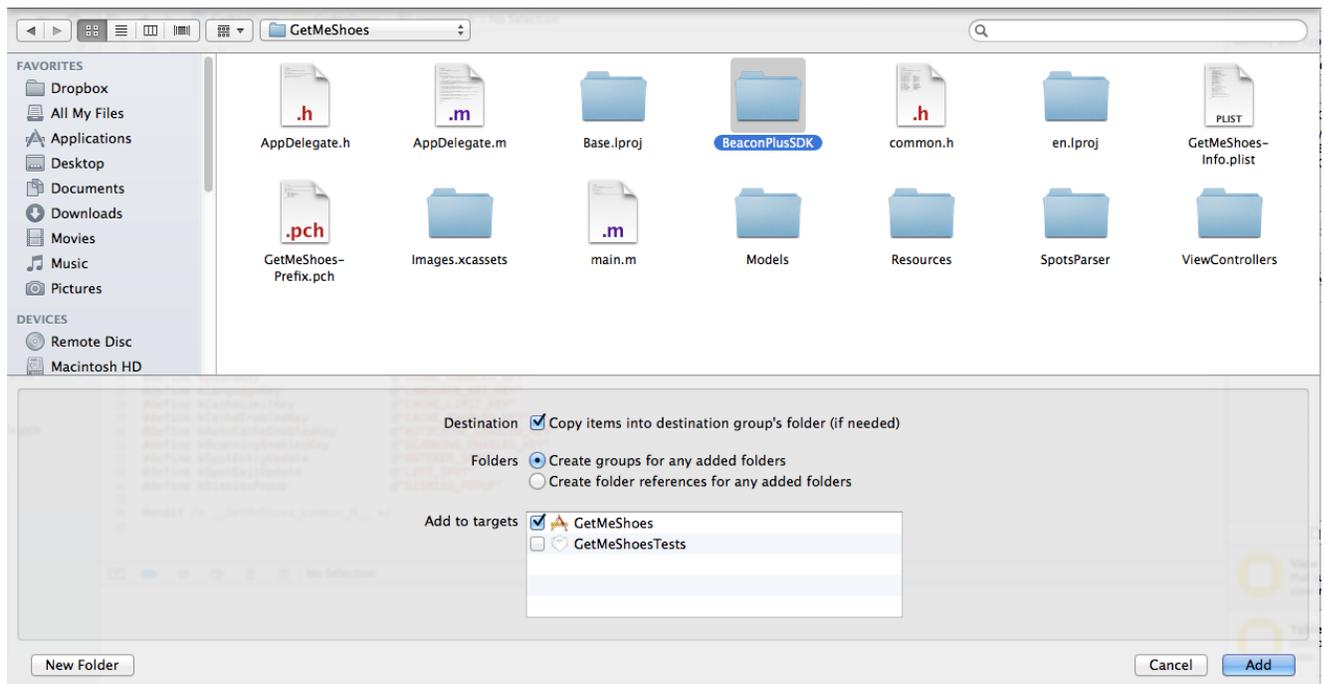
f)   Make sure that "Copy items into destination folder (if needed)" is checked

g)   Set Folders to "Create groups for any added folders"

h)   Select all targets to which you would want to add the SDK, for example "GetMeShoes" in this case.

i)   Add the following frameworks to GetMeShoes
   - libsqlite3.0.dylib
   - SystemConfiguration.framework
   - CoreLocation.framework
   - CoreMotion.framework
   - CoreBluetooth.framework

pinmicro
Know your surroundings
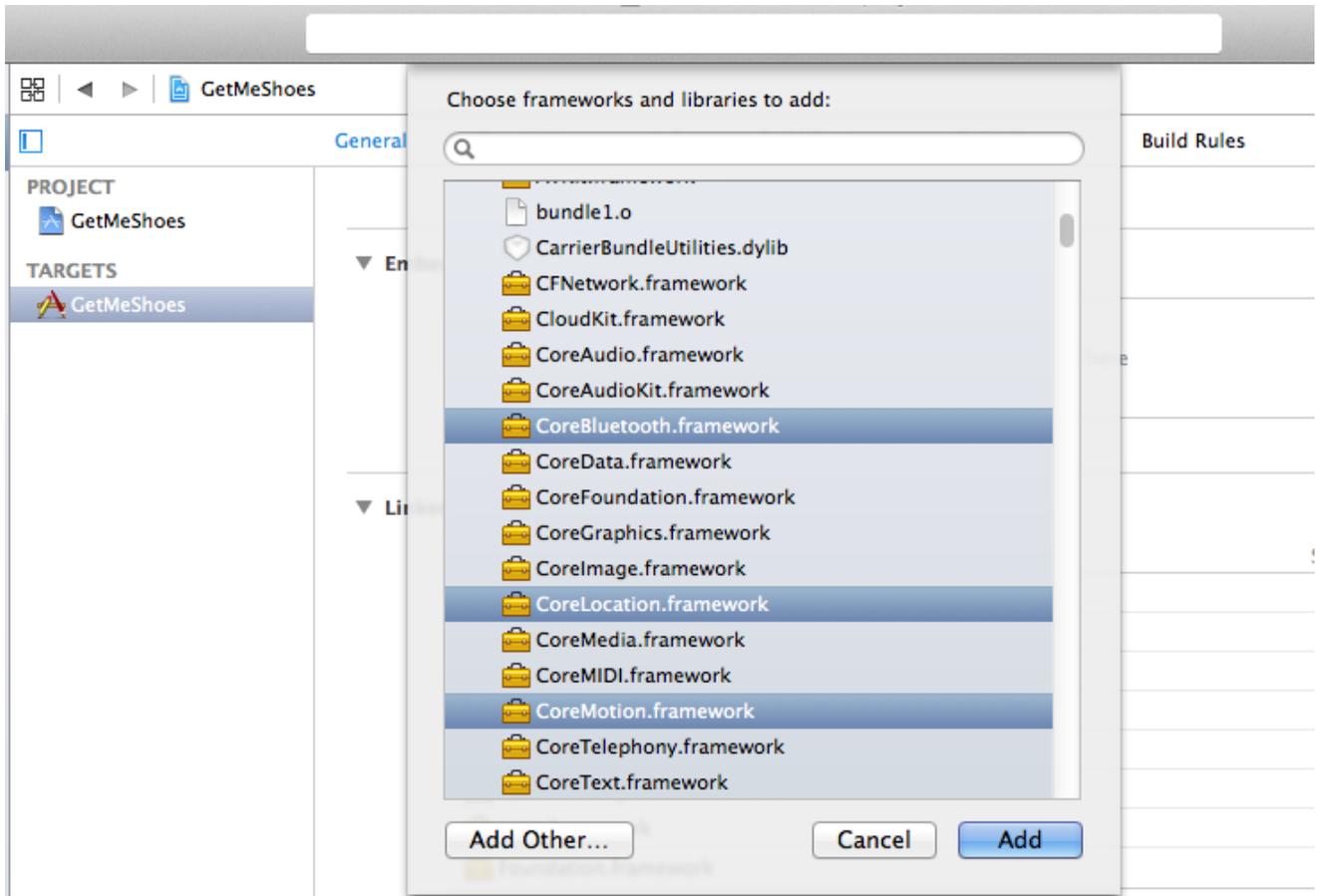
j) The 'Frameworks' folder in GetMeShoes should now look like
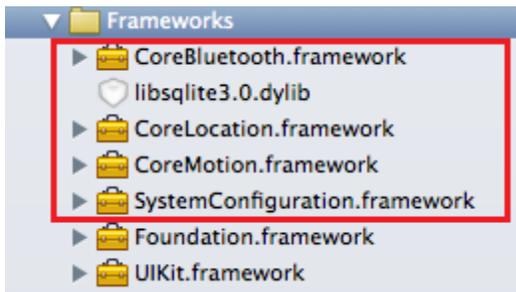
k) Verify that libBeaconPlus.a is added to the 'Link Binary With Libraries' under 'Build Phases' for the targets you want to use the SDK with
   - Select GetMeShoes in the Project Navigator
   - Select the target for which the SDK is to be enabled
   - Select the Build Phases tab
   - Open the 'Link Binary With Libraries' tab in 'Build Phases'
   - If libBeaconPlus.a is not listed, drag and drop the library/framework from your Project Navigator to 'Link Binary With Libraries'
   - Repeat Steps 2 - 5 until all targets you want to use the SDK will have the SDK linked to it

5   How to integrate the SDK in your application

a. Import BeaconPlus to the class in which you would like to implement the SDK related actions.

```
#import "BeaconPlus.h"
```

b. Add the BeaconPlusDelegate protocol.

```
HomeView: UIViewController < BeaconPlusDelegate>
```

c. Set the call back handler delegate

```
[[BeaconPlus sharedBeaconPlus] setDelegate: self];
```

d. Using the shared instance of BeaconPlus class, call the method to initialize the SDK.

```
[[BeaconPlus sharedBeaconPlus] initializeWithAccessToken: < accessToken >
                                        language: < language >
                                       serverUrl: < serverUrl >
                                     rangingMode: < mode >];
```

e. Implement the callback methods in BeaconPlusDelegate protocol

e.1. Success response

```
- (void)beaconPlusDidInitialize {
}
```

The method will be invoked when the SDK has initialized successfully.

e.2. Spot entry event

```
- (void)didEnterSpot:(Spot *)spot {
}
```

Will be called each time a new spot is detected.

e.3. Spot exit event

```
- (void)didExitSpot:(Spot *)spot {
}
```

The receiver will get a call on this method each time a previously reported spot has now gone out of the user's range.

e.4. Failure response

```
- (void)didFailWithError:(NSError *)error {
}
```

The method will be called each time an error occurs in the SDK.

pinmicro
Know your surroundings

f. Info.plist settings

Starting from iOS 8, location updates won't get enabled in an iOS application by default as it used to be in previous versions of the iOS. To achieve this, update the info.plist file with the following entries

- NSLocationWhenInUseUsageDescription
- NSLocationAlwaysUsageDescription

The info.plist file of the app should now look like

InfoDictionary version		String	6.0
Executable file		String	${EXECUTABLE_NAME}
▶ Required device capabilities		Array	(1 item)
View controller–based status bar ap...		Boolean	NO
Bundle identifier		String	com.innovaturelabs.bp
Bundle creator OS Type code		String	????
▶ Icon files (iOS 5)		Dictionary	(0 items)
Application requires iPhone environment		Boolean	YES
NSLocationAlwaysUsageDescription		String	
NSLocationWhenInUseUsageDescription		String	

g. Start scanning for beacons.

```
[[BeaconPlus sharedBeaconPlus] startScanning: &error];
```

The user may opt to scan for beacons once the SDK has been initialized. On successful commencement of the scanning process, it will send event updates (Spot entry and exit) to the didEnterSpot and didExitSpot callback methods.

6   Authentication via Access Token

The BeaconPlus SDK identifies each user via the Access Token provided to each Organization after they have created their own account in BeaconPlus cloud. The SDK then retrieves matching information for the application from the cloud and performs further tasks.

Key features:

- Access token serves as a "key" that identifies a developer and lets the system know what data to access from the cloud. Hence the developer doesn't have to implement any separate logic for login or authentication.

- An organization may use the same access token for multiple applications. The deployment information of the applications may or may not be the same and the app developer could implement the app logic to suit his requirements.
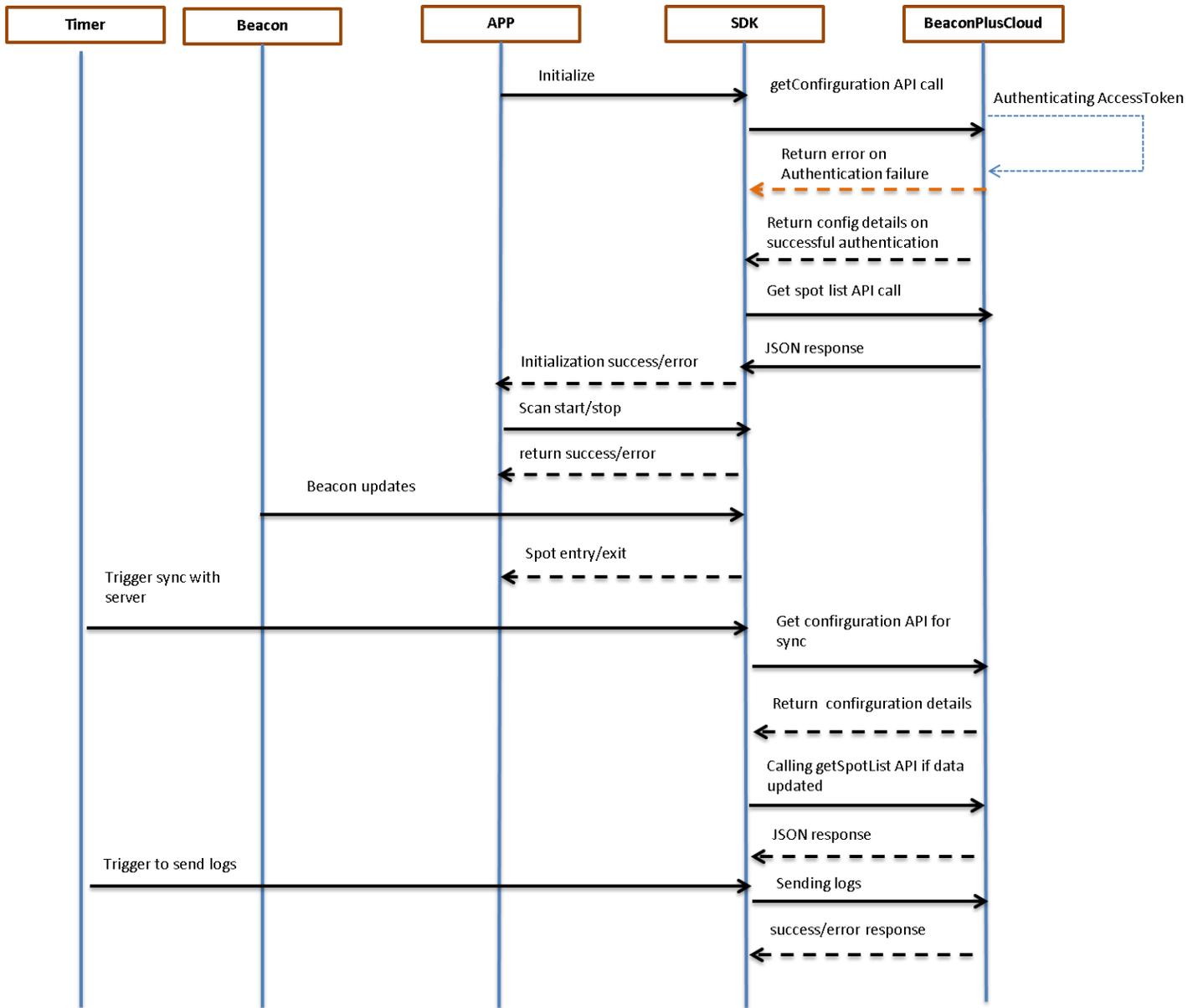
pinmicro
Know your surroundings

7   How to troubleshoot

Please go through the steps mentioned in the below document:

https://pinmicro.com/docs/sdk_help.pdf

Please feel free to contact us at support@pinmicro.com for further assistance.

8   Call flow diagram

9   Error Codes

Following are the error codes that would be returned to the app from the SDK.

Error Code	Reason
701	Network error
702	App validity has expired
703	SDK uninitialized
707	Invalid access token
709	Invalid time
710	Device unsupported
713	Caching failed, couldn't return temporary url too
714	Device types not configured
715	Scanning uninitialized
716	Regions not added
717	Devices not added
718	Location services are disabled
719	Unable to connect to server
720	Minimum cache limit not met
724	Database error occurred
729	Invalid data provided
731	Invalid device types set for scanning
732	No spots added for the application yet
733	Minimum scanning interval should be 1
734	File types not available for the specified tag name
735	Bluetooth is not enabled
736	Organization validity has expired
737	Organization blocked by super admin
738	Application blocked by super admin
739	Server error
740	Same language as that set now

10 Bibliography

Starting point for iOS development:
https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/

iBeacon reference guide: https://developer.apple.com/ibeacon/