

**Alinket Electronic Technology (Shanghai) Co,.Ltd**

**ADMS as a Driver**

**(Android Version)**

**User Manual**

**Version: 0.1**

**Date: 2015.4.16**

**Copyright**

©Alinket Technologies Corp

This document is the property of Alinket Technologies Corp and is delivered on the express condition that it not be disclosed, reproduced in whole or in part, or used for manufac­ture for anyone other than Alinket Technologies Corp without its written consent, and that no right is granted to disclose or so use any information contained in said document. This restriction does not limit the right to use information obtained from other sources.

**Trademarks**

ADMSaaD is a registered trademark of Alinket Technologies Corp. Alinket is a registered trademark of Alinket Technologies Corp. Other products and company names mentioned herein may be the trademarks of their respective owners.

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision Content** | **Revision By** | **Version** |
| 2015/4/16 | -Initialization Version | He Yi | 0.1 |
| 2015/4/17 | -Changed secure communication’s port assignments | He Yi | 0.2 |
| 2015/4/27 | -Changed APIs to public static, no need to instantiate Class Object, and give sample codes for API usage with AsyncTask | He Yi | 0.3 |
|  |  |  |  |
|  |  |  |  |

**List of Contents**

[1 General Definition 7](#_Toc417896862)

[1.1 ADMS (Alinket Device Middleware Service) 7](#_Toc417896863)

[1.2 ADMS as a Driver (Android Version) 7](#_Toc417896864)

[2 ADMSaaD Software Provisioning 7](#_Toc417896865)

[3 Secure Communication Channel 8](#_Toc417896866)

[3.1 Usage Scenario 8](#_Toc417896867)

[3.2 APIs Description 8](#_Toc417896868)

[3.3 APIs Usage 9](#_Toc417896869)

[3.4 Customer APP and Web Service Modifications 10](#_Toc417896870)

**List of Figures**

[Figure 1 ADMS Creates Communication Channels 7](#_Toc417896871)

[Figure 2 Insecure Communication Channel without ADMS 8](#_Toc417896872)

[Figure 3 Secure Communication Channel Created by ADMS 8](#_Toc417896873)

**List of Tables**

[Table 1 ADMSaaD (Android Version) APIs 9](#_Toc417896874)

[Table 2 Customer APP and Service TCP/IP Address and Ports Usage 10](#_Toc417896875)

# General Definition

## ADMS (Alinket Device Middleware Service)

ADMS is the abbreviation for Alinket Device Middleware Service, it is a set of distributed middleware software which helps all IoT “Things” (either gadgets or home appliances) to connect into IoT Cloud services.

## ADMS as a Driver (Android Version)

ADMSaaD (Android Version) is provided as an Android Module Package (.aak) and intends to be used by customer Android APP to establish communication channels with IoT Devices as well as Cloud Services:



Figure 1 ADMS Creates Communication Channels

Customer APP uses software APIs provided by ADMSaaD to establish communication channels with IoT devices and secure communication channels with cloud services (some may be implemented by customer themselves).

ADMSaaE and ADMSaaS stand for “as an Element” and “as a Service” correspondingly. They are all cooperation software modules with ADMSaaD for communication channels’ establishment.

# ADMSaaD Software Provisioning

ADMSaaD (Android Version) will be provided an Android Module Package (.aar):

|  |
| --- |
| ***AlinketIoTMiddleware-release-v<x.x.x>.aar*** |

The implemented Java package name is:

|  |
| --- |
| ***package com.alinket.iotmiddleware;*** |

Customer Android APP project can import it as a library dependency, and import the above package to use the Alinket IoT Middleware APIs.

|  |
| --- |
| ***import com.alinket.iotmiddleware.\*;***  ***……*** |

# Secure Communication Channel

## Usage Scenario

In case that customer deployed their application service (for example, a Web Service) onto the Cloud platform and requires that a secure communication channel between the APPs and the services:



Figure 2 Insecure Communication Channel without ADMS

ADMSaaD provides a more secure way to hide customer Web Service from public accessible Internet address and prevent it from being attacked:



Figure 3 Secure Communication Channel Created by ADMS

Compared the above 2 cases with and without ADMS, the latter one provides security benefits as:

* Customer Web Service is hidden from public accessible Internet address.
* All communications are encrypted by our secure communication channel.

## APIs Description



Table 1 ADMSaaD (Android Version) APIs

## APIs Usage

To create the secure communication channel with the Web Service, invoke the API directly as:

|  |
| --- |
| ***import com.alinket.iotmiddleware.\*;***  ***……***  ***AlinketIoTMiddleware.AlinketIoTMiddleware();*** |

Please note that you will not be able to invoke the API directly from an UI element (Button, etc), invoke it from an AsyncTask is a good practice, and the application is informed after connection establishment or failure in the onPostExecute() method of the AsyncTask Class, thus give a point to start the APP/Server communication.

It typically takes 20 ~ 40 seconds to successfully establish the secure communication channel or failed timeout in 190 seconds. So it is better to invoke this API in background AsyncTask on APP startup.

|  |
| --- |
| ***……In Class MainActivity:***  ***private class ConnectTask extends AsyncTask<Void, Void, Integer> {***  ***private long elapsed = 0;***  ***protected Integer doInBackground(Void... ps) {***  ***elapsed = System.currentTimeMillis();***  ***return AlinketIoTMiddleware.CreateSecureCommunication();***  ***}***  ***protected void onPostExecute(Integer connected) {***  ***elapsed = System.currentTimeMillis() - elapsed;***  ***if (connected == 0) {***  ***showbox.append(String.format("OK in %d ms.", elapsed));***  ***} else {***  ***showbox.append(String.format("Failed in %d ms.", elapsed));***  ***}***  ***}***  ***}*** |

To check for the success or failure of the creation of the secure communication channel, check the return value of the API:

|  |
| --- |
| ***Boolean connected = AlinketIoTMiddleware.IsSecureCommunicationEstablished ();***  ***……check the connected Boolean value……*** |

Notes: this API will block until the secure communication channel was established successfully, or failed due to some reason. In failure you will need to retry the above API to connect again.

|  |  |
| --- | --- |
| ***Typical wait time to success*** | *20~40 seconds* |
| ***Typical timeout*** | *190 seconds* |

To release the secure communication channel, use below API, and also create an AsyncTask to handle it in background and may implement the onPostExecute() method also depends on APP’s requirement:

|  |
| --- |
| ***……In Class MainActivity:***  ***private class DisconnectTask extends AsyncTask<Void, Void, Void> {***  ***protected Void doInBackground(Void... ps) {***  ***AlinketIoTMiddleware.ReleaseSecureCommunication();***  ***return null;***  ***}***  ***}*** |

## Customer APP and Web Service Modifications

ADMSaaD minimizes customer’s code modifications for their existing APP and Web Service and in meanwhile secures the Cloud Server from threads of attacking.

To use above secure communication mechanism customer needs to do 2 code modifications:

* **Change their APP code to create and maintain the secure communication channel.**
* **Change the APP Web Service Calls to localhost address 127.0.0.1 as below.**
* **On the Cloud Server side, deploy their Web Service to bind to localhost address 127.0.0.1 as below.**



Table Customer APP and Service TCP/IP Address and Ports Usage