



Verizon 5G Mobile Edge Compute (MEC)

VESA Edge Discovery Service

Software Development Kit (VESA EDS SDK) User Manual

Important—Please Read

Verizon Confidential & Proprietary.

© 2021 Verizon. All rights reserved.

Restricted and Controlled Distribution. Not to be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of Verizon.

All information herein is subject to change without notice. The information provided was considered accurate at the time the document(s) were developed, and Verizon disclaims and makes no guaranty or warranty, express or implied, as to the accuracy or completeness of any information contained or referenced herein.

VERIZON DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

Verizon does not guarantee or warrant the availability of the network nor the compatibility of a network with any device, service or product. Verizon disclaims liability for any damages or losses of any nature whatsoever whether direct, indirect, special or consequential resulting from the use of or reliance on any information contained or referenced herein.

Technical data contained in this document may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Verizon and Verizon logos are trademarks of Verizon. Other product and brand names may be trademarks or registered trademarks of their respective owners.

Revision history

Version	Date	Description
0.03	June 20 th , 2022	End user feedback
0.02	June 14 th , 2022	Release for portal upload
0.01	May 20 th , 2022	Initial Release to QA

Contents

1 Introduction	4
1.1 Purpose	4
1.2 Package contents	4
1.3 System requirements	4
1.3.1 Hardware	4
1.3.2 Software and tools	4
1.4 New features, fixes, and known Issues	5
2 EDS SDK RELEASE PACKAGE	6
2.1 Folder Structure	6
2.2 EDS SDK Client	6
2.2.1 EDS Sample App	6
2.2.2 EDS SDK library	6
2.2.3 EDS SDK Client APIs	7
2.4 EDS SDK Release Package Supported Features	7
3 Raspberry Pi Bring up	7
3.1 OS installation on Raspberry Pi	7
3.2 Enabling Developer Option on Raspberry Pi	8
4 EDS SDK Testing	8
4.1 Application Integration with EDS SDK Client	8
4.2 EDS SDK Client testing on AVD	9
4.3 EDS SDK Client testing on Raspberry Pi	10
5 Logging	13
5.1 EDS SDK logs	13
B References	14
B.1 Acronyms and terms	14
B.2 EDS Client Reference Logs	14

1 Introduction

1.1 Purpose

VESA EDS SDK Client provides optimal MEC endpoint to the application to reduce latency and brings real-time performance to high-bandwidth applications. To use this client, a user needs to subscribe for Verizon approved VESA services for example CVONVIF is one of the Verizon approved VESA services.

This user manual provides details about the contents of the VESA EDS SDK release package for Android OS devices, and how to use VESA EDS SDK Client APIs with the Edge sample application. This manual also shares the references to bring up the setup on a Raspberry Pi device.

1.2 Package contents

- Release Notes
- EDS SDK Client library
- EDS SDK Client API files
- EDS Sample application source code references

1.3 System requirements

1.3.1 Hardware

- Raspberry Pi android
- Activated Verizon 5G SIM

1.3.2 Software and tools

- EDS SDK Client library
- User Application code

- Android Studio

1.4 New features, fixes, and known Issues

- Refer Release_Notes.txt

2 EDS SDK Release Package

Get the EDS SDK release package from MDP (MEC Development Portal). To download the Android EDS SDK Release package, access the link below.

<https://www.verizon.com/business/5g-edge-portal>

EDS SDK Release package named as “EDSSDK_V<version>-Android” zip file.

2.1 Folder Structure

EDSSDK_V1.0.0-Android		
Name	Kind	
COPYRIGHT.txt	Plain Text Document	
EDS SDK_user manual V0.01.pdf	PDF Document	
EDS-SDK-Client	Folder	
EDS_Sample_App	Folder	
code references	Folder	
MainActivity.txt	Plain Text Document	
EDS_SDK_Client_APIs	Folder	
EDS_SDK_Client_APIs.txt	Plain Text Document	
EDS_SDK_Client_Library	Folder	
signed	Folder	
EDSSDKLib-1.0.0-debug.aar	Apple Archive	
EDSSDKLib-1.0.0-release.aar	Apple Archive	
Export_readme.txt	Plain Text Document	
LICENSE.pdf	PDF Document	
readme.txt	Plain Text Document	
Release_Notes.txt	Plain Text Document	

2.2 EDS SDK Client

This section provides the details about the VESA EDS SDK Release package contents.

2.2.1 EDS Sample App

Developers are required to develop their apps to utilize the EDS SDK Client APIs. Developers can refer to *code references/Main_activity.txt* file to understand how to integrate EDS SDK Library with the application. For more details refer to Section 4.1

2.2.2 EDS SDK library

This folder contains debug and release versions of the EDS SDK Client library. Developers can use the debug version of the library for their app development. Once Developers' applications are ready for release, they must integrate the EDS SDK Client LIBRARY release version with the application.

2.2.3 EDS SDK Client APIs

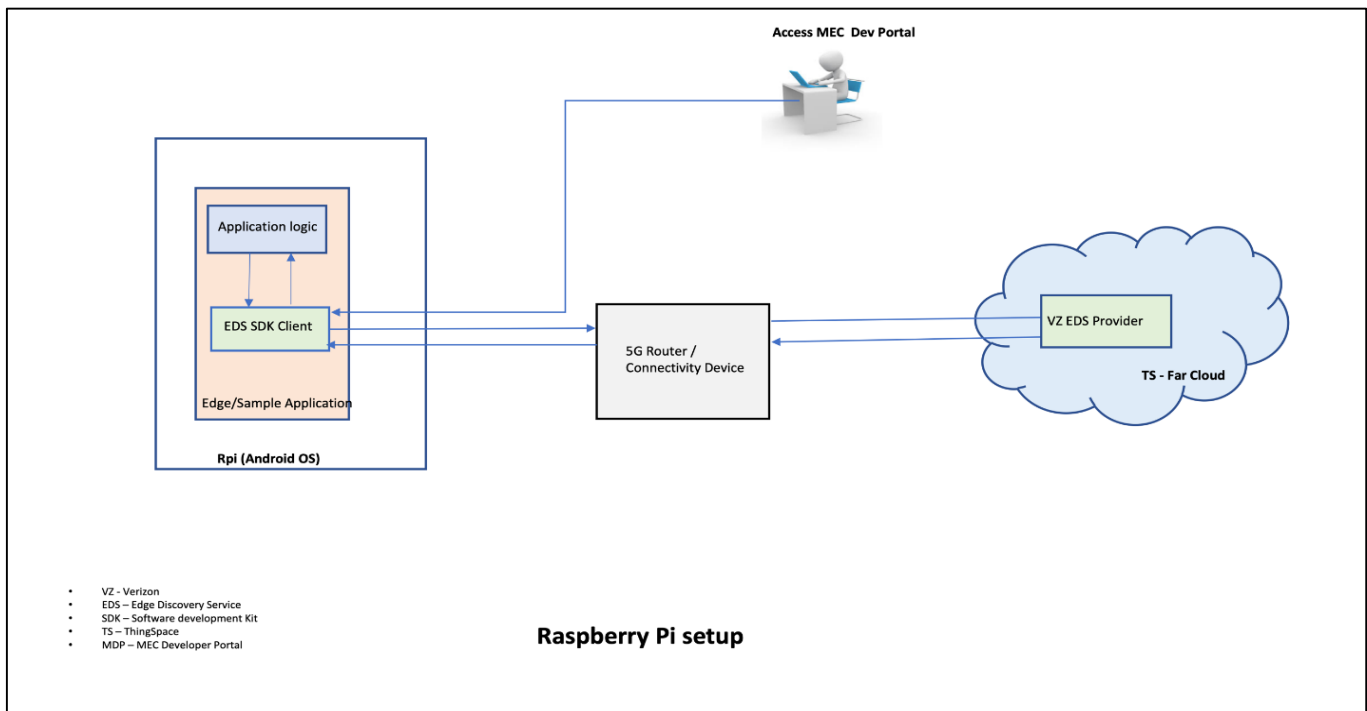
This File lists the different APIs provided by the EDS SDK Client library. The application can utilize any of these listed APIs to get the required functionalities.

2.4 EDS SDK Release Package Supported Features

For the list of supported features and known issues, refer to ReleaseNotes.txt

3 Raspberry Pi Bring up

Developers can test the EDS Sample Application or their own applications on Android OS devices. Raspberry Pi device with android OS can be used as a developer's device.



3.1 OS installation on Raspberry Pi

EDS SDK client and sample application are tested on android 11 lineage OS 18.1. To install this OS on a Raspberry Pi device, follow the step at the link below:

<https://konstakang.com/devices/rpi4/LineageOS18/>

3.2 Enabling Developer Option on Raspberry Pi

To flash the application image/apk file on a Raspberry Pi device via adb/wifi, the developer option needs to be enabled on the device. To enable it, go to

settings -> About tablet -> Click 'Build number' seven times.

Users should be able to see the device options

settings -> system -> advance -> developer options

4 EDS SDK Testing

It is the application developers can utilize EDS SDK APIs to integrate EDS SDK Client library with their application code. Users can test EDS SDK Library in debug mode while developing the app. For the application release version, it is recommended to use the EDS SDK Library release version.

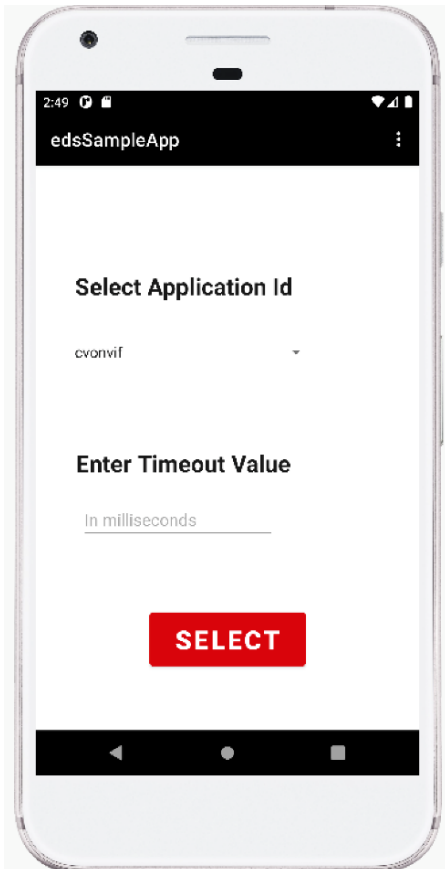
4.1 Application Integration with EDS SDK Client

EDS SDK Client is written in java language. To integrate the EDS SDK client with their application user need to follow the below steps:

1. Get the Android EDS SDK release package.
2. Open Android studio, click on File->Open the existing android application project. Give some time to sync all the files.
3. From the left pane click on the project rather than Android. Go to the <Application>/app/libs/ folder.
4. Open the EDS SDK package, Drag and drop EDSSDKLib-<version>-release.aar/ EDSSDKLib-<version>-debug.aar file to <Application>/app/libs/ folder in android studio.
5. On android studio go to the left pane and select Android rather than project. Go to Gradle Scripts and open build.gradle with name <Application>.app.
6. In <Application>.app go to dependencies body configuration add the EDS SDK library dependency by adding the two lines below –
7. implementation ("com.squareup.okhttp3:logging-interceptor:4.9.1")
8. implementation files("libs/EDSSDKLib-<version>-debug.aar") or implementation files("libs/EDSSDKLib-<version>-release.aar") according to the added library.
9. The user will see the sync option in the right top corner. Click on this option to sync the library with the application. At this moment, the user is ready to build the apk file. Go to the menu and select Build-> Build Bundles/(APKs)-> Build APK(s)
10. On android studio go to the left pane and select Android rather than project. Go to manifests folder and open AndroidManifest.xml file.
11. Search for allowBackup and replace android:allowBackup="true" to android:allowBackup="false"
12. Add tools:replace="android:theme" after android:theme="@style/Theme.<Application name>"

4.2 EDS SDK Client testing on AVD

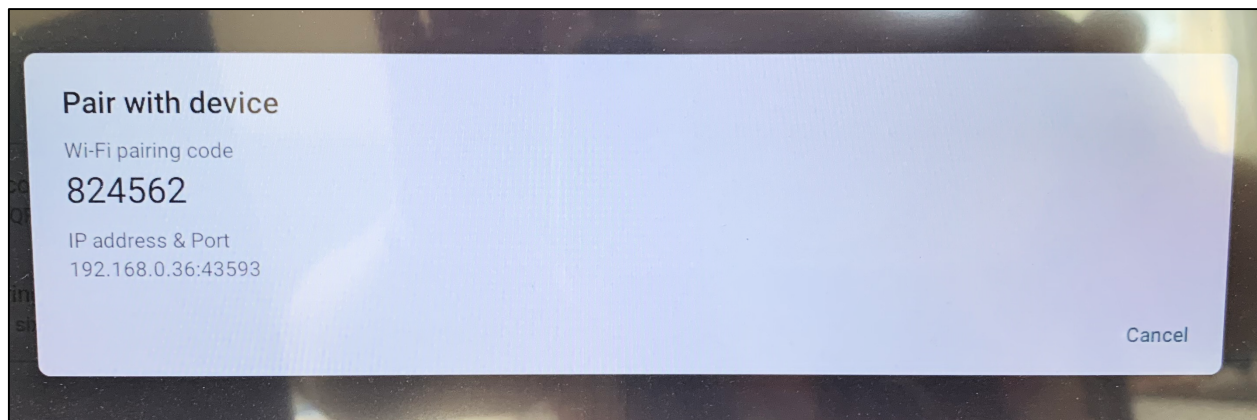
To install and run the application on the AVD (Android Virtual Device), the user can flash the apk on the emulator. Once the application is installed and running, the user can test the success/failure scenarios to test the EDS SDK Client on the emulator. Device logs can be observed from the logcat windows. Below is the screenshot of internally developed VZ sample application. Similarly, developers can install and run their compiled application apk on the AVD.



4.3 EDS SDK Client testing on Raspberry Pi

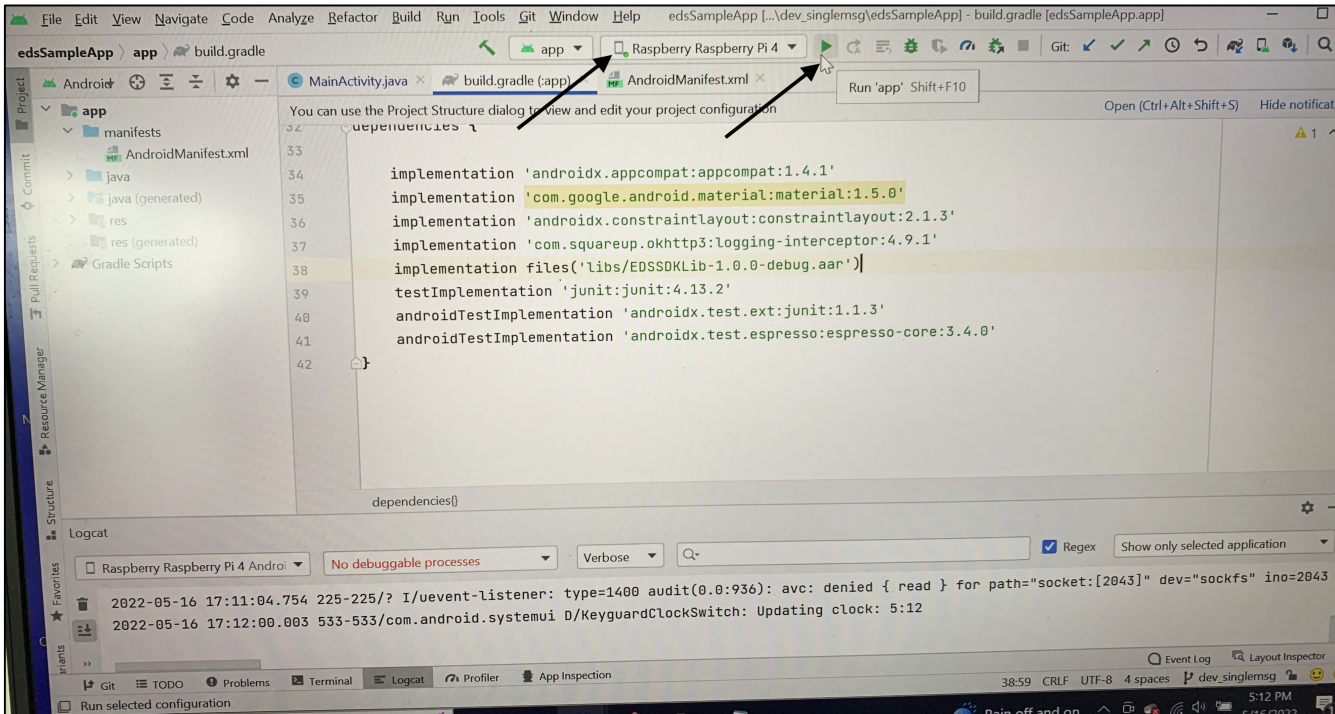
To test the EDS client on the real device, a sample application apk file need to be installed on the Raspberry Pi board. To install the apk file, the user needs to connect the Raspberry Pi board with ADB (android debug bridge). This can be achieved via a USB debug cable or over a wifi connection. To connect Rpi with ADB over wifi users must follow the below steps.

1. Remove the Ethernet cable from RPI if connected any.
2. Connect RPI and development laptop over the same wifi network.
3. On Rpi enable settings -> system -> advance -> developer options
4. On Rpi enable settings -> system -> advance -> developer options -> USB debugging on
5. On Rpi enable settings -> system -> advance -> developer options ->Wireless Debugging On
6. On Rpi settings -> system -> advance -> developer options ->Wireless Debugging->pair device with pairing code. The image below will pop up on the screen:

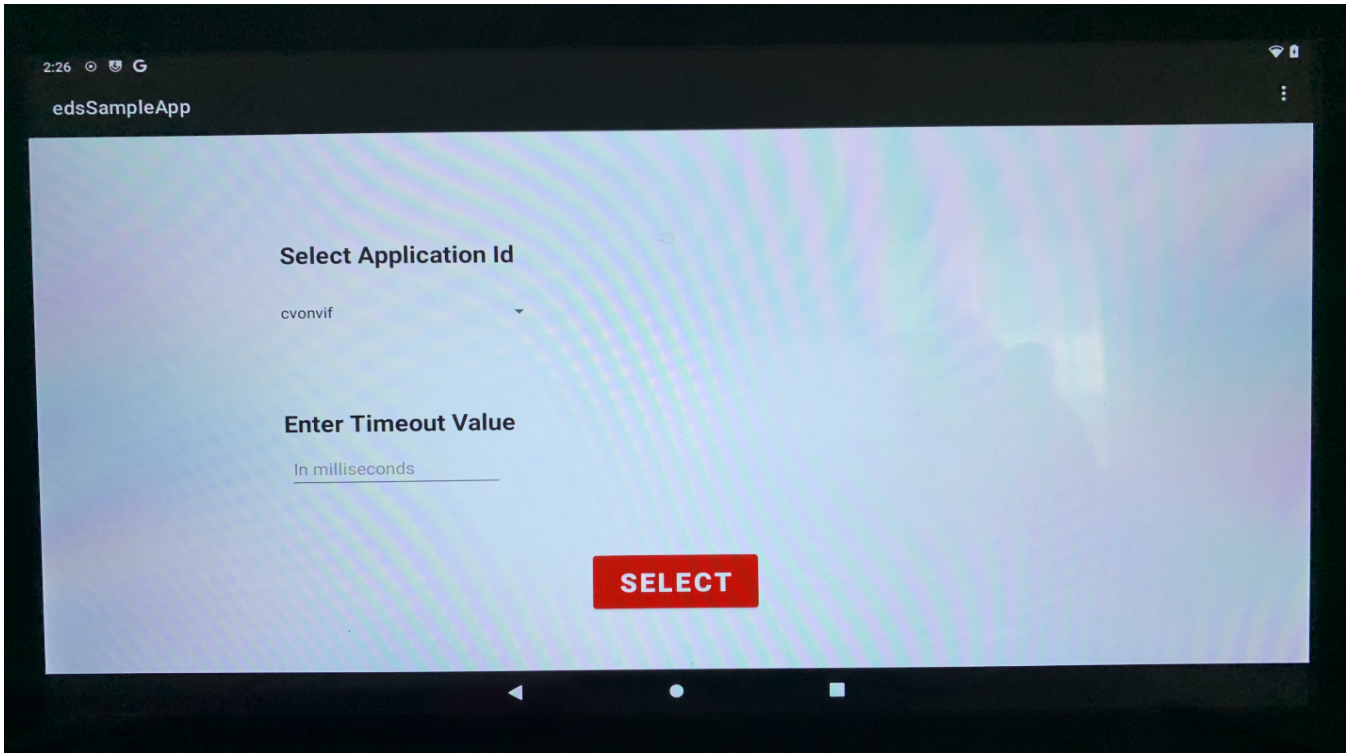


1. On the development laptop verify where the ADB is installed. For my laptop it is at `/c/users/Shweta/AppData/Local/Android/Sdk/platform-tools`. Open gitbash or any bash prompt. If not installed, then the user can install gitbash by accessing the link below:
<https://git-scm.com/>
2. On the bash prompt go to ADB path, do ls user should see adb.exe. To pair the Rpi device with windows execute the command below:
`/adb.exe pair 192.168.0.36:43593` and press enter. It will ask for a pairing code, enter the pairing code and then the User should see on Rpi that the device is paired with the development machine.

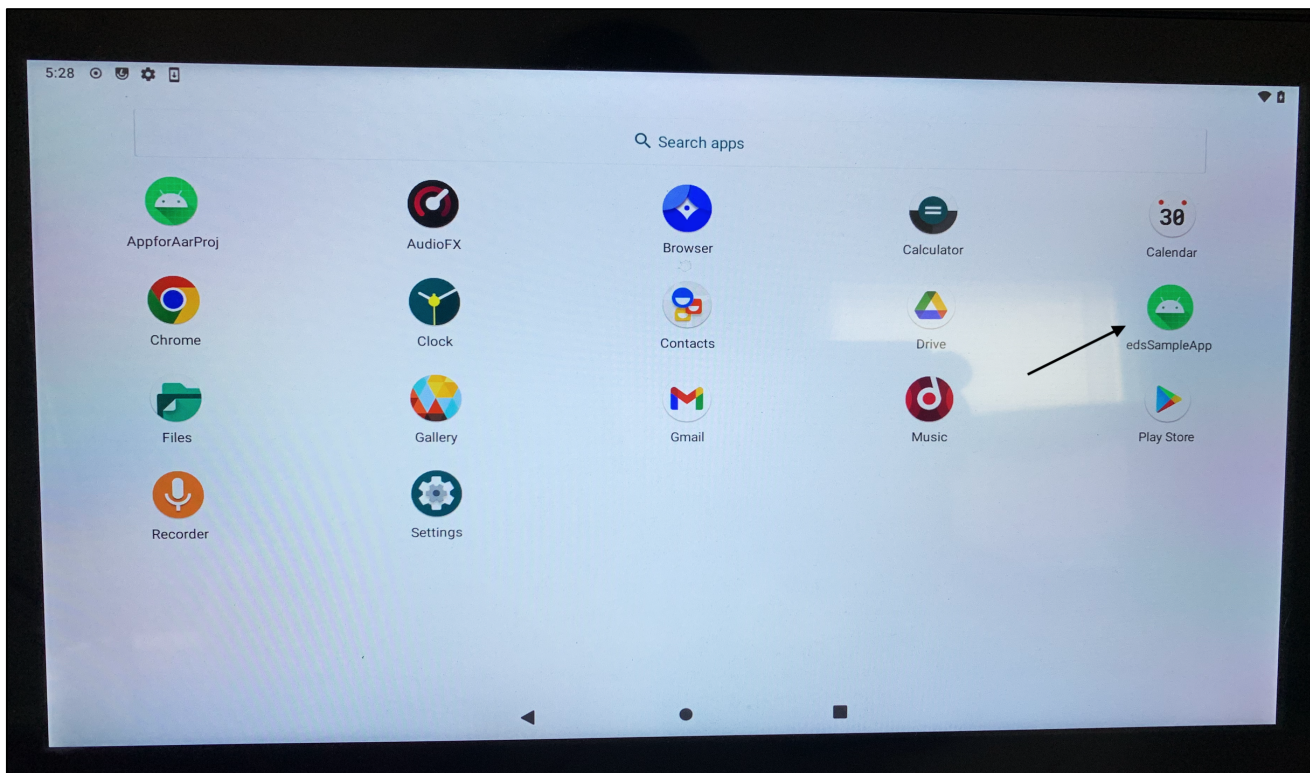
3. To connect the Rpi device with windows execute the command below:
`./adb.exe connect 192.168.0.36:43593` sometimes it displays the error that device not connected and rejected the connection. But the user should be able to see the device on the android studio device manager.
4. From the device manager select the Rpi device and click on the Run command icon to install the image. It will take a few seconds to install the app on the Rpi, the user can see the logs on logcat.



5. Once the app is installed, the user can see the App GUI on the Rpi board. Below is the screenshot of internally developed VZ sample application on RPI. Similarly, developers can install and run their compiled application apk on the RPI4.



6. To see the application icon, go to the app tray, the user will see the image captured below.



7. Similarly, developers can flash their compiled application apk file on a Raspberry Pi device.

5 Logging

5.1 EDS SDK logs

EDSSDKLib-1.0.0-debug.aar – contains all level logs

EDSSDKLib-1.0.0-release.aar – contains error and warning logs

B References

B.1 Acronyms and terms

Acronym	Definition
API	Application programming interface
RPI	Raspberry Pi
EDS	EDGE Discovery Service
VZ	Verizon
TS	ThingSpace
SDK	Software development kit

B.2 EDS Client Reference Logs

```

2022-05-18 14:45:10.788 17978-17978/com.example.edssampleapp D/edsSDKClient: edsdkLibVersion: EDSSDKLib-1.0.0
2022-05-18 14:45:10.788 17978-17978/com.example.edssampleapp D/edsSDKClient: received the ca
2022-05-18 14:45:11.066 17978-17978/com.example.edssampleapp D/edsSDKClient: EDS SDK Client Initialized
2022-05-18 14:45:11.066 17978-17978/com.example.edssampleapp D/edsSampleApp: edsSdkLibVersion: EDSSDKLib-1.0.0
2022-05-18 14:45:21.135 17978-17978/com.example.edssampleapp D/edsSampleApp: AppServStr: cvonvif
2022-05-18 14:45:21.135 17978-17978/com.example.edssampleapp D/edsSampleApp: timeout_ms: 3000
2022-05-18 14:45:21.377 17978-17978/com.example.edssampleapp D/edsSampleApp: Request sent successfully
2022-05-18 14:45:22.303 17978-18017/com.example.edssampleapp D/edsSDKClient: responseHandler
2022-05-18 14:45:22.304 17978-18017/com.example.edssampleapp D/edsSDKClient: Received
successResponse{protocol=http/1.1,code=200,message=OK,
url=edscloudeservice?deviceIp=174.115.168.67&serviceId=cvonvif&scope=no_cred}
2022-05-18 14:45:22.305 17978-18017/com.example.edssampleapp D/edsSDKClient: rspbdy:
{"edgeservices":[{"serviceid":"cvonvif","servicetype":"CV service - Cam control and
streaming","serviceaccesspoint":[{"uri":"tscloudservice.com","protocol":"wss","port":443},{"uri":"tscloudservice.com
","protocol":"rtsp","port":322}]}]}
2022-05-18 14:45:22.305 17978-18017/com.example.edssampleapp D/edsSDKClient: edgeservices_jarray.length(): 1
2022-05-18 14:45:22.305 17978-18017/com.example.edssampleapp D/edsSDKClient: srvaccesspoint_jarray.length(): 2
2022-05-18 14:45:22.306 17978-18017/com.example.edssampleapp D/edsSDKClient: num_srvaccpoint: 2
2022-05-18 14:45:22.306 17978-18017/com.example.edssampleapp D/edsSDKClient: eds_rsp: ServiceId:cvonvif SrvType:CV
service - Cam control and streaming Uri: tscloudservice.com Ipv4addr: Protocol:wss Port:443 Fqdn:
2022-05-18 14:45:22.306 17978-18017/com.example.edssampleapp D/edsSDKClient: eds_rsp: ServiceId:cvonvif SrvType:CV
service - Cam control and streaming Uri: tscloudservice.com Ipv4addr: Protocol:rtsp Port:322 Fqdn:
2022-05-18 14:45:22.307 17978-18017/com.example.edssampleapp D/edsSDKClient: edsdk_clientState:
EDSCLIENT_OPTIMAL_MEC_RES_RCVD
2022-05-18 14:45:22.307 17978-17978/com.example.edssampleapp D/edsSampleApp: onSuccessResponse msgType:
EDS_ENDPOINT_SUCCESS_RSP
2022-05-18 14:45:22.307 17978-17978/com.example.edssampleapp D/edsSampleApp: Received endpoint rsp,
numAccessPoints:2
2022-05-18 14:45:22.307 17978-17978/com.example.edssampleapp D/edsSampleApp: access_point: ServiceId:cvonvif
SrvType:CV service - Cam control and streaming Uri: tscloudservice.com Ipv4addr: Protocol:wss Port:443 Fqdn:
2022-05-18 14:45:22.307 17978-17978/com.example.edssampleapp D/edsSampleApp: access_point: ServiceId:cvonvif
SrvType:CV service - Cam control and streaming Uri: tscloudservice.com Ipv4addr: Protocol:rtsp Port:322 Fqdn:

```