



Verizon 5G Mobile Edge Compute (MEC)

Edge Discovery Service (EDS) Client Software Development Kit (SDK) - for Android Mobile Devices

Integration Guide

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 guarantee 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 warranty 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.

Table of Contents

Revision History	4
1. Introduction	5
1.1 Purpose	5
1.2 System Requirements	5
1.2.2 Hardware	5
1.2.3 Software	5
1.3 Package Contents	5
2. Prerequisite Requirements	6
2.1. Edge Service Deployment and Endpoint Configuration	6
2.2 MEC API Key	6
2.3 Google SafetyNet	6
3. EDS Client SDK Release Package	7
4. API Overview	7
5. Client SDK Integration	9
5.1 Maven Library Reference	9
6. Client SDK APIs	10
6.1 EdsClientSdkFactory	10
6.2 EdsClientSdkFactory.SdkBuilder	10
6.3 IEdsClientSdk	10
6.4 EdgeSvcTargetingCriteria	11
6.5 IEdgeDiscoveryCallback	11
6.6 IEdsSdkOptions	11
6.7 EdsServiceEndpoint	12
6.8 EdsConstants	12
7. Client SDK Utilities	12
7.1 EdsClientSdkLogWrapper	12
7.2 EdsClientSdkTaskWrapper	12
7.3 EdsNetworkUtils	12
8. References	13
9. FAQs	14

10. Appendix..... 16

10.1 SDK Sample Code 16

10.2 Java Docs 16

10.3 Error Codes..... 16

Terms and Acronyms..... 17

Revision History

Version	Date	Description
1.0	Dec, 2022	Initial release

1. Introduction

Verizon 5G Edge Services is a multi-access edge computing (MEC) platform that enables cloud servers to run closer to optimal endpoints to reduce latency and accelerate local processing. With the platform, developers can extend their cloud environments to include Verizon edge zones and consume APIs to simplify device management and connectivity to optimal edge endpoints.

1.1 Purpose

[Verizon's Edge Discovery Service](#) API enables developers to discover 5G edge platforms and register endpoints.

This document explains how developers can use an SDK instead of the REST interface to discover optimal endpoints from Android mobile client applications.

Verizon provides an 5G Edge Discovery Service (EDS) Client SDK for Android devices to simplify runtime integration with Verizon's 5G Edge service on Android devices.

Once a developer has deployed their edge services and configured within the Verizon 5G Edge MEC platform, the EDS Client SDK exposes simple APIs to applications, offering:

- Runtime Discovery of optimal endpoints use mobile's network location or region + density.
- Transparent authentication with Verizon MEC system using Device Attestation utilizing Google SafetyNet, or direct authentication using MEC access token.

1.2 System Requirements

1.2.2 Hardware

Any mobile device running the Android OS that meets specific platform requirements

1.2.3 Software

This EDS Client SDK is compatible with:

- Android devices running Android-L (Lollipop, SDK 21) or newer.
- Integration into applications with minimum SDK version 21 or newer (minTargetSdk)

Note: The SDK will only work if the device is connected to the Verizon network. (Roaming situations are supported).

1.3 Package Contents

Reference Name	File Name(s)	Location
EDS Client SDK Libraries	vzwEdsClientSdk-1.1.0-4-m2repo.zip	Bundled
EDS Client SDK Javadocs	vzwEdsClientSdk-1.1.0-4-javadoc.zip	Bundled
EDS Client SDK Sample Code	vzwEdsClientSdk-sampleCode-1.1.0-4-src.zip	Bundled
License Agreement	EDS SDK License Agreement.pdf	Bundled
Release Notes	Release_Notes.txt	Bundled
Integration guide	EDS SDK Client Integration Guide Android.docx	Bundled

2. Prerequisite Requirements

2.1. Edge Service Deployment and Endpoint Configuration

5G Edge-hosted services must be configured within the Verizon 5G Edge MEC platform. (See [Verizon 5G Edge Platform Documentation](#)).

2.2 MEC API Key

A MEC API Key must be created and configured using [Verizon MEC API Key](#) tools.

For authentication using Device Attestation (*recommended*):

With the [Verizon MEC API Key](#) tool:

1. Select **Limited Access Token** tab
2. Select **Add new key set** (or use existing key set)
3. Select **Enable Mobile Client Edge Discovery** (at bottom)
4. Select **Add Authorized Application Bundle ID**
5. Select **'android'** then add the Application's **Bundle Name** and the **SHA1 Fingerprint of the application signing certificate**

For authentication with the MEC accessToken:

A runtime MEC accessToken must be generated. Use **keyId** and **secretId** from a **MEC key set**.

1. Within the [Verizon MEC API Key](#) create a **key set** (Avoid *keyId* and *secretId*).
2. At runtime, generate an **accessToken** based on the **Obtaining an API Token** instructions in the [5G Edge Services Getting Started Guide](#) on the EDS portal.

Verizon recommends using **LIMITED ACCESS** key set and/or **accessToken** for edge discovery operations.

2.3 Google SafetyNet

For authentication using the device attestation, the application must be configured to use **SafetyNet**, Google's Android Device Attestation service.

For a Google account associated with an application

- Enable **Android Device Verification API** on the **Library** page in the **APIs and Services** section of the [Google Cloud Developers Console](#) (Independent **Google SafetyNet** sign-in required).
- Configure the API Key on the **Credentials** page in the **APIs and Services** section of the [Google Cloud Developer Console](#) (Independent **Google SafetyNet** sign-in required).
 - **NOTE:** The **Credentials** page intermittently displays the **'Create Project'** link. In other instances, the page displays the API keys creation steps. If the **API keys** option is not shown, repeat navigation using its side-menu

WARNING: Google may enforce a quota on the use of its **SafetyNet API** of 10,000 requests per day. Applications exceeding that number may be blocked. Applications may request an increased quota using the [Quota request form](#). For more information, see: [Setting up Monitoring for Quota Requests and alerting](#)

NOTICE: Google is deprecating the SafetyNet API, disallowing new registrations **January 31st 2023** and disabling fully on **June 30th 2023**. Verizon is developing an updated solution based on the replacement Google Integrity APIs.

3. EDS Client SDK Release Package

Obtain the EDS Client SDK release package from the MDP ([MEC Development Portal](#)). The EDS Client SDK package is available for download on the **Tools** page.

4. API Overview

The Verizon EDS Client SDK allows discovery of optimal edge service endpoints for your edge hosted service. The following concepts are fundamental to Edge Discovery integration:

- **Authentication** – The Verizon MEC server APIs require authentication. The EDS Client SDK supports two authentication mechanism:
 - **Attestation via Google SafetyNet** – Verizon server validates a Google SafetyNet Device Attestation response. Client Application must configure Google API Key and MEC application IDs.
 - **MEC Access Token** – Client Application responsible for obtaining MEC accessToken using MEC keyset and associate OAUTH API, which is used by EDS Client SDK for authentication.

EDS Client SDK accepts authentication specification in ***EdsClientSdkFactory.SdkBuilder***

- **Network Targeting** – Edge Discovery using device location is supported only when connected to the Verizon cellular network.
 - Client application must be aware of network operator and network connectivity state.
 - Client application must supply an active Cellular network connection as part of edge discovery
 - This connection may differ from default data network if device is connected to Wi-Fi network or in dual-SIM devices.

EDS Client SDK accepts ***targetNetwork*** as part of UE Identity targeting criteria.

EDS Client SDK Utilities ***EdsNetworkUtilities*** class contains helper functions to identify supported SIMs and obtain SIM targeted network connections

- **Edge Service Targeting** – Based on MEC capabilities, the EDS Client SDK supports multiple options for selecting the optimal Edge Service endpoint.
 - Subscriber location (based on cellular IP-based UE Location), with optional region.
 - Service Region and optional subscriber density.

EDS Client SDK accepts ***svcTargetingCriteria*** argument to discoverEdgeServices operation.

The SDK offers the following features:

- **Ease of Integration** - SDK libraries are supplied as local maven repository, allowing simple integrations and dependency resolution using Android/Gradle build tools. See [Client SDK Integration](#) for more details.
- **Factory Creation** – SDK instances and associated classes are created using a factory class and its embedded builder. The factory class will always be you starting point when creating and configuring SDK instances.

Edge Discovery Service Client Software Development Kit (EDS SDK) Integration Guide

The ***EdsClientSdkFactory*** includes several `get*()` methods, including ***getSdkBuilder()*** to obtain an ***SdkBuilder*** for construction of ***IEdsClientSdk*** instances.

- **Asynchronous Execution** – Edge discovery operations are asynchronous, with calling app providing callbacks which are executed on completion or supplied timeout is reached. A single successor error callback is guaranteed to occur unless app explicitly cancels the operation.

Applications implement the ***IEdgeDiscoveryCallback*** interface and supply to ***discoverEdgeServices()*** operation.

A Google Tasks wrapper is also provided within the `utils` library for developers who prefer the Task interface or wish to utilize the `Tasks.await()` method for simplified synchronous execution.

EDS Client SDK utilities contains ***EdsClientSdkTaskWrapper***, exposing Tasks based discovery operations.

- **Caching** – The SDK caches edge discovery results for a default of 10-min, avoiding network transactions and the associated latency. Calling applications may change the cache duration, opt to bypass the cache, and clear cache contents.
- **Logging** – The SDK supports debug logging and offer a log-wrapper to aid in debugging by ensuring all requests and responses are logged in a complete and consistent way.

EdsClientSdkFactory ***setDebugLogging()*** methods enables/disables debug logging

EDS Client SDK Utilities ***EdsClientSdkLogWrapper***.***getLogWrappedSdk()*** method will add additional request/response logging for all SDK operations and is recommended when capturing logs for investigation/diagnosis by version.

WARNING: Debug logging and log-wrapper output may contain sensitive data and should never be enabled in consumer application builds.

5. Client SDK Integration

The Client SDKs are delivered as Android Library (AAR) files within a local Maven repository.

API	Maven Dependency
EDS Client SDK	com.verizon.mec:eds-clientsdk:[1.0,)
EDS Client SDK Utilities (optional)	com.verizon.mec:eds-clientsdk-utils:[1.0,)

NOTE: Supplied Maven version matching will always take more current available build (ends with “,”)). Application may adjust as desired.

5.1 Maven Library Reference

See below build.gradle snippets to reference SDK via local maven repository. Sample may need to be adapted for the variants and local maven repository path within your specific project.

In the project build.gradle file:

- I. Define maven repository referencing a local copy of EDS maven repository folder (adjust path/naming per your project requirements/preferences).
- II. Declare dependencies on the client SDK library

```
repositories {  
    maven {  
        url uri("${projectDir}/lib/m2Repository")  
    }  
}  
  
dependencies {  
    implementation 'com.verizon.mec:eds-clientsdk:[1.0,)'  
  
    // if using utility classes (log wrapper, network utils, task wrapper  
    implementation 'com.verizon.mec:eds-clientsdk-utils:[1.0,)'  
}
```

6. Client SDK APIs

The primary SDK classes and functions are listed below. Please refer to Javadocs and Sample Code for more information.

6.1 EdsClientSdkFactory

Class: *com.verizon.mec.edsclientsdk.EdsClientSdkFactory*

Factory class used to create IEdsClientSdk and other SDK class instances, configure logging, and perform other setup and configuration tasks.

Primary methods include:

- **getSdkBuilder()** – Obtain SdkBuilder to for SDK instance creations.
- **setDebugLogging()** – Enable/Disable debug logging
- **getOperationOptions()** / **getDefaultOperationOptions()** – Obtain operations options instance.
- **getVersionInfo()** – Obtain SDK version and build information

6.2 EdsClientSdkFactory.SdkBuilder

Class: *com.verizon.mec.edsclientsdk.EdsClientSdkFactory\$SdkBuilder*

Builder class used to create IEdsClientSdk instances, obtained from EdsClientSdkFactory.getSdkBuilder.

Primary methods include:

- **build()** - Create SDK instance.
- **setIdmsAttestAuthenticator()** – Configure authentication using attestation.
- **setMecAccessTokenAuthenticator()** – Configure authentication using MEC accessToken
- **setCacheDuration()** – Change default cache duration. Default is 10-min.
- **setCallbackLooper()** – Set alternate looper to be used for success/error callbacks. Default is the main looper.

6.3 IEdsClientSdk

Class: *com.verizon.mec.edsclientsdk.api.IEdsClientSdk*

Interface defining client SDK operations. Obtained from EdsClientSdkFactory SdkBuilder. Contains primary edge discovery operation '**discoverEdgeServices()**' this asynchronous method is called with:

- **callback:** IEdgeDiscoveryCallback instance, containing applications onSuccess and onError implementation
- **svcTargetingCriteria:** EdgeSvcTargetingCriteria subclass, determining optimal edge services selection:
 - EdgeSvcNetworkAddrTargetingCriteria using device's network location (IP)
 - EdgeSvcRegionTargetingCriteria using supplied region and optional density
- **operationOptions:** IEdsSdkOptions allows caller to supply operation-specific configuration values, such as timeout. Can be null in which case default values are used, as set via setDefaultOperationOptions()

A shortcut version of discoverEdgeServices() accepts MEC svcEndpointsId and target network, performing UE location targeting (same as supplying EdgeSvcNetworkAddrTargetingCriteria to full method).

Edge Discovery Service Client Software Development Kit (EDS SDK) Integration Guide

Primary `IEdsClientSdk` methods include:

- **discoverEdgeServices()** - asynchronous edge services discovery operation, accepting callback, targetingCriteria and operation options. Returning handle to cancel operations. Guaranteed to perform onSuccess or onError callback within configured timeout.
- **setAuthenticator()** – Update authenticator set in `SdkBuilder`. Useful MEC accessToken expiration or after errors occur with errors requiring transitioning to accessToken authentication.
- **setDefaultOperationOptions()** – Convenience method to set default options for all operations.
- **clearCache()** – Clear all cached edge service results.

6.4 EdgeSvcTargetingCriteria

Classes: `com.verizon.mec.edsclientsdk.api.EdgeSvcTargetingCriteria`

`com.verizon.mec.edsclientsdk.api.EdgeSvcNetworkAddrTargetingCriteria`

`com.verizon.mec.edsclientsdk.api.EdgeSvcRegionTargetingCriteria`

Allows caller to pass edge service targeting information into `discoverEdgeServices()` operations. All criteria require a MEC serviceEndpointsId.

- **EdgeSvcNetworkAddrTargetingCriteria** used UE location and requires a Network object for an active cellular network connection in order to ensure network location is available.
- **EdgeSvcRegionTargetingCriteria** accepts region and optional minimum subscriber density. Use-cases for this targeting mechanism are unclear.

6.5 IEdgeDiscoveryCallback

Class: `com.verizon.mec.edsclientsdk.api.IEdgeDiscoveryCallback`

Asynchronous callback interface, implemented by calling application and supplied to `discoverEdgeServices` call. Caller must implement:

- **onSuccess()** - Called on successful completion of edge discovery operations, supplying a (possibly empty) set of optional edge service nodes.
- **onError()** – Called on unsuccessful completion. Includes numeric error code, error message (not for display to end user), and underlying exception, if any.

6.6 IEdsSdkOptions

Class: `com.verizon.mec.edsclientsdk.api.IEdsSdkOptions`

Operation options container class, allowing configuration of:

- **OperationTimeout** – Time after which timeout error will be returned to onError callback. Default is 30-seconds.
- **BypassCache** – Instruct SDK to bypass any cached results and force server transactions. Results will still be cached for future discovery calls. Default is to use cached results if available.

- **TransportNetwork** – Allows caller to dictate Android Network connection to use for server transactions. Android's default INTERNET connection is used by default.

6.7 EdsServiceEndpoint

Class: *com.verizon.mec.edsclientsdk.data.EdsServiceEndpoint*

EdsServiceEndpoint is a data contain class containing parsed results from Edge Discovery. These values are populated from the server response JSON, also returned in onSuccess() callback handler.

Structure and values are based on provisioned edge services data, defined in [MEC portal API specification](#).

6.8 EdsConstants

Class: *com.verizon.mec.edsclientsdk.api.EdsConstants*

Contains default configuration values, URL targets, and error code values that may be useful to calling application.

7. Client SDK Utilities

A secondary, optional, utilities library is supplied. Please refer to Javadocs and Sample Code for more information.

7.1 EdsClientSdkLogWrapper

Class: *com.verizon.mec.edsclientsdkutils.EdsClientSdkLogWrapper*

Adds logging for all request operation, parameters and results. Intended for diagnosis/troubleshooting only. Static **getLogWrappedSdk()** method accepts and returns an IEdsClientSdk instance.

Also contains **logSdkBuildInfo()** and **logHostAppInfo()** methods to trigger diagnostic logging of SDK and app version information, useful for debugging.

7.2 EdsClientSdkTaskWrapper

Class: *com.verizon.mec.edsclientsdkutils.EdsClientSdkTaskWrapper*

Exposes Google Play Service promise-based Tasks wrapper for **IEdsClientSdk discoverEdgeServices()** functions.

7.3 EdsNetworkUtils

Class: *com.verizon.mec.edsclientsdkutils.EdsNetworkUtils*

Includes methods for identifying SIMs which support edge discovery operations and establishing network connections.

Primary methods include:

- **isVerizonNetworkOperator()** – Evaluates supplied MCC/MNC to identify Verizon network operator.
- **targetVerizonNetworkSim()** – Adds Verizon SIM targeting to supplied Network Request if compatible SIM is found. Returns Boolean value indicating whether targeting criteria was added. No permissions or SDK restrictions.
- **getVerizonNetworkDataSubscriptionId()** – Return Android subscriptionId of default data SIM if that SIM supports edge services. Not supported by legacy devices. No special permissions required.

- **getVerizonNetworkDataSubscription()** – For applications requiring additional SIM information, evaluates default data SIM for compatibility. Requires phone permission and not supported by legacy devices.
- **getAllSubscriptionsOnVerizonNetwork()** – For applications supporting connections via non default SIM on DSDS devices, returns information on all candidate SIMs. Requires phone permission and not supported by legacy devices.

Primary methods include:

- **isVerizonNetworkOperator()** – Evaluates supplied MCC/MNC to identify Verizon network operator.
- **targetVerizonNetworkSim()** – Adds Verizon SIM targeting to supplied Network Request if compatible SIM is found. Returns Boolean value indicating whether targeting criteria was added. No permissions or SDK restrictions.
- **getVerizonNetworkDataSubscriptionId()** – Return Android subscriptionId of default data SIM if that SIM supports edge services. Not supported by legacy devices. No special permissions required.
- **getAllSubscriptionsOnVerizonNetwork()** – For applications supporting connections via non default SIM on DSDS devices, returns information on all candidate SIMs. Requires phone permission and not supported by legacy devices.

8. References

1. Verizon MEC Edge Discovery Services:
<https://www.verizon.com/business/5g-edge-portal/documentation/verizon-5g-edge-discovery-service.html>
2. Verizon EDS SDK License Agreement
EDS SDK License Agreement.pdf (bundled)
3. Verizon MEC Key Management
<https://www.verizon.com/business/5g-edge-portal> (sign-in/account creation required)
4. Google Play Services Registration
<https://console.cloud.google.com/projectselector2/apis/dashboard> (sign-in/account creation required)
5. Google Play Services Tasks APIs (optional)
<https://developers.google.com/android/guides/tasks>

9. FAQs

1. Which devices are supported?

The SDK supports Android devices running SDK 20 (Lollipop) or later.

2. Which SIMs and networks are supported?

Verizon Edge Discovery is available on any device connected to the Verizon network, including 5G and 4G/LTE devices. SIM roaming on Verizon network and Verizon MVNO / Wholesale carriers are supported.

3. How do we resolve issues / errors / open questions?

To submit an issue for investigation, visit [customer support](#) or call 1-800-473-0466. The submission should include:

- A detailed description and reproduction steps, including resulting `errorCode`, error message, and any exception information
- Information on device make, model and OS version
- When possible, complete logcat logs captured during operation, preferably with debug logging enabled and using `EdsClientSdkLogWrapper`.

4. What if a device is missing or uses an outdated version of Google Play Services?

Google Play services are required when using attestation-based authentication. By default, Play Services v13 or later (versions) is required to support a restriction on Google API keys, though this may be disabled in SDK creation.

If an operation requires play services and it is not available, one of the following error codes will be returned:

EdsConstants#ERROR_PLAY_SVCS_MISSING

EdsConstants#ERROR_PLAY_SVCS_OUTDATED

The application may prompt the user to install/update Play Services via the Google Playstore and proceed with edge discovery using the MEC `accessToken` (which does not require Play Services), or bypass edge discovery and use a default service endpoint.

5. What if Google SafetyNet Attestation fails for a device?

When using device attestation authentication, if the SafetyNet device attestation fails on the device the edge discovery operation will return the error code:

EdsConstants#ERROR_SAFETYNET_FAILED_ATTESTATION

If the attestation result is rejected by the Verizon server

The application chose to prompt the user to resolve the issue which caused attestation to fail. If this occurs, proceed with edge discovery using MEC `accessToken` (which does not require device attestation), bypass edge discovery and use a default service endpoint, or block access for this user/device.

6. What network should be used for communicating with edge services?

Edge Services are designed to leverage Verizon 5G connectivity. A Verizon cellular connection is required to perform edge discovery targeting of a UE location. **EdsNetworkUtils** provides helper functions for performing network and SIM checks.

Edge-hosted services are deployed by applications. Depending on the deployment, the edge service endpoint may also be available over WiFi and non-Verizon data connections.

The application may choose to interrogate the network connection quality and speed to determine the optimal network for edge service communication.

7. How should an application manage connectivity changes?

Edge-hosted services are deployed by applications. Depending on the deployment, the edge service endpoint may also be available over WiFi and non-Verizon data connections.

Depending on application endpoint accessibility, the application may choose to connect via WiFi or other connection technology should cellular connection be lost or connect to a default non-Edge-hosted endpoint.

10. Appendix

10.1 SDK Sample Code

See bundled sample code showing integration with Client SDK methods.

SEE: EDS Client SDK Release Package (Chapter. 3)

Code samples include the following:

- **EdsSdkBasicSample** – Minimal implementation using all defaults
- **EdsSdkSample** – More complete implementation showing configuration and additional SDK options
- **EdsTaskWrapperSample** – Synchronous and Asynchronous tasks example
- **EdsNetworkTargetingSample** – Usage of EdsNetworkUtils in evaluating SIMs and creating SIM-targeted Network instances.

10.2 Java Docs

Bundled ZIP contains Javadoc HTML API documentation for developers' use.

10.3 Error Codes

Error Code	Generated By	Description
100	Client SDK	General / Unknown Error
102	Client SDK	Timeout making network request
103	Client SDK	Operation timeout
113	Client SDK	Security Exception
119	Client SDK	Unexpected Runtime Exception
120	Client SDK	Malformed Server Response
121	Client SDK	SSL Certificate Error
122	Client SDK	DNS Error
123	Client SDK	Network Error
124	Client SDK	Unexpected network redirect
130	Client SDK	Invalid client parameter
131	Client SDK	Client provided network inactive
132	Client SDK	Incompatible identity Target Network for Edge Discovery (non-Cellular)
171	Client SDK	Google Play Services unavailable
172	Client SDK	Google Play Services outdated
173	Client SDK	Google SafetyNet Error
174	Client SDK	Google SafetyNet Timeout
180	Client SDK	Client Callback Exception
7xxx	Server Error	HTTP/ Network error on server Challenge (HTTP Code + 7000)
8xxx	Server Error	HTTP/Network error on server Attestation (HTTP Code + 8000)
9xxx	Server Error	HTTP/Network error on MEC Edge Discovery (HTTP Code + 9000)

Terms and Acronyms

Term	Definition
5G:	Fifth Generation wireless standard
5G Edge	Verizon MEC computing platform
accessToken	Runtime MEC API OAUTH access token based on MEC API key and secret
appld	ID associated with Verizon MEC keyset
APK:	Android Package Kit - file format for applications used on Android operating system
API:	Application Programming Interface
AAR:	Android Archive
EDS:	Edge Discovery Service
ERN	Edge Resource Number
FQND	Fully Qualified Domain Name
LTE:	Long Term Evolution (4G wireless standard providing increased network capacity and speed for IoT devices)
MAVEN	A build automation tool used primarily for Java
MDP:	MEC Developer Portal
MEC:	Multi-access Edge Compute
MVNO	Mobile Virtual Network Operator
SafetyNet	Google SafetyNet Device Attestation Service
SIM:	Subscriber Identity Module
targetNetworks	Cellular network connection used by an application when communicating with an edge service instance. Used by the EDS SDK to obtain UE Identity/
transportNetwork	Cellular or WIFI network used by EDS Client SDK to communicate with Verizon Servers.
UE:	User Equipment e.g. Mobile Device
UE-ID	A unique identifier for specific User Equipment. (Primarily an IP address, for this document).
UI:	User Interface
VZ:	Verizon