# Overview

The VehicleSimulationLib library is created for developers to use and receive vehicle crash notification in a simulated way. The library uses OpenXC library for reading vehicle data stream, and implements a simple (tentative) logic to simulate the vehicle crash situation.

Thus library is meant to provide an abstraction to developers on OpenXC interface and crash logic, and get a seamless notification on vehicle crash.

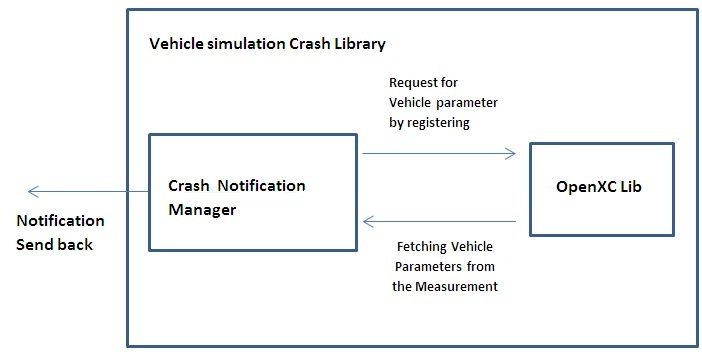
The library is Android based since uses in turn the Android based OpenXC library.

For testing the simulated crash – a c Customized Drive Trace file has been used, though developers are free to use another file which has the crash situation occurrence situation.

**On the code release folder we are providing TWO projects**

|  |  |  |
| --- | --- | --- |
| 1 | **VehicleCrashSimulationLib** | The library code for vehicle crash simulation. Can be taken and used by developers in their apps. |
| 2 | **VehicleCrashSimulationTestApp** | A simple test app written to use the **VehicleCrashSimulationLib**, and display the crash occurrence on a simple UI. |

# Architecture Overview



**Architecture highlights –**

1. Add VehicleSimulationLib library as reference library into your project.
2. Vehicle crash simulation library (CSL) uses OpenXC Lib in-turn for reading the live vehicle data stream. Pls refer the link [**http://openxcplatform.com/android/getting-started.html**](http://openxcplatform.com/android/getting-started.html)for OpenXC lib use.
3. The app that needs to use Crash simulation library, has to add this library as a reference, which is normal way of using a library in an Android app.
4. Receive notification by implementing onVehicleCrashListener interface.
5. **Crash Detection Logic**:
   1. Engine speed and vehicle speed are the parameters being used for Crash detection. We are taking the basis that if both the parameters come down drastically to 0/close to 0, then it can be perceived as a crash situation.
   2. Specifically if engine speed is greater than 1000 rpm and vehicle speed is greater than 40 kmpl and then in next read both parameters are observed zero or close to zero, then it is taken as a crash situation.
   3. In hard breaking scenario – though vehicle speed will get zero, but not the engine speed.
6. **Notification mechanism from library** –
   1. A listener (**OnVehicleCrashedListener**) is implemented to notify vehicle crash.
   2. **onVehicleCrashed ()** method is receiving crash notification, which is declared in OnVehicleCrashedListener and overridden in Test App Activity class.

# Platform requirements

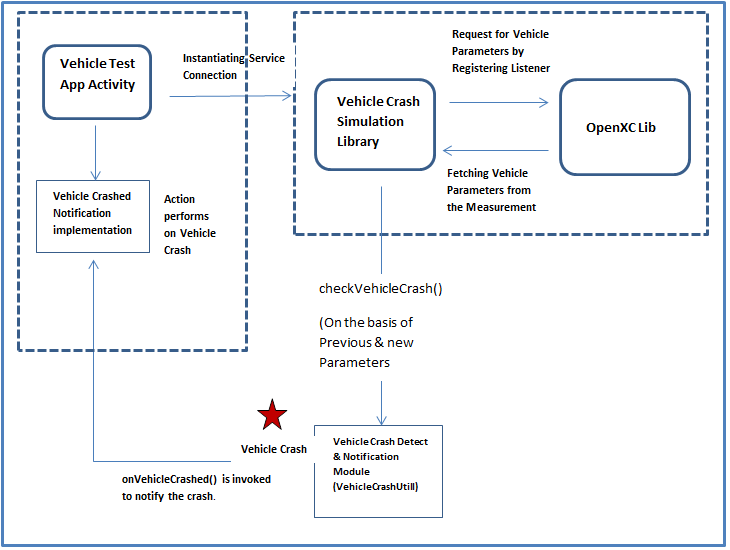
App/Library runs on Android platform (both Phone and Tab device could be used).

Platform requirements are same as that for OpenXC lib/apps.

Any version 2.3 or higher will be compatible.

# How to Develop Apps using Crash Simulation Library

Below diagram depicts an overall solution and execution flow –

****

**Key steps –**

* Add Vehicle Simulation Crash App Lib as Reference Library in the App
* Developer can get notification via method **onVehicleCrashed()**
* Write code to do afterwards action on detecting vehicle crash – like sending relevant notification to relevant recipients; uploading the crash GPS location on emergency service provider; etc.

# Testing with a Drive Trace file

Current library is using a drive trace file (Driving.JSON ) to run and detect the crash per the crash detection logic. The drive trace file as part of project in the resource folder.

Developers are free to use their own Drive trace file; however unless the file is not having the crash occurrence scenario (per the logic implemented in library), a crash will not be detected. Of course Drive trace file can be modified manually to create a crash occurrence in simulation.

**~~ end of doc ~~**