Heart Attack Risk Predictor

“ Heart Attack Risk Predictor is an mobile application designed to use On-Device Machine Learning with **TensorFlow Lite inference** ”

- Input is EHR Data and follows a ISO 13606 standard.

Two Scenarios

- Case 1: Pre-Diagnosis of a patient at Primary Health Care centre.
- Case 2: Self-Diagnosis of a Patient at Home.

Two Applications

- Heart Attack Risk Predictor [HARP] takes 22 EHR features based on Case 1.
- Heart Disease Predictor [HDP] takes 10 features based on Case 2.

Flutter for Mobile Applications

Our main objective is to develop an application that is cross-platform for mobile, desktop and web platforms.

To achieve this purpose, we used **Flutter** !

What is Flutter ?

- Flutter is an open source software development Kit (SDK) created by Google.
- Everything is a widget, and the apps are written in **Dart**, Google's own Programming Language.

Why Flutter ?

- Faster Development [Hot Reload & Restart]
- Native Performance.
- Cross Platform → Single code base for Android, iOS, Desktop and Web applications.
- Full Customization & Fast Rendering
- Expressive & Beautiful UI's and many more.

Firebase

- Firebase is a Backend-as-a-Service (BaaS) developed by Google offering variety of tools & services to developers.
- It is a NoSQL database, and stores data in JSON-like documents.

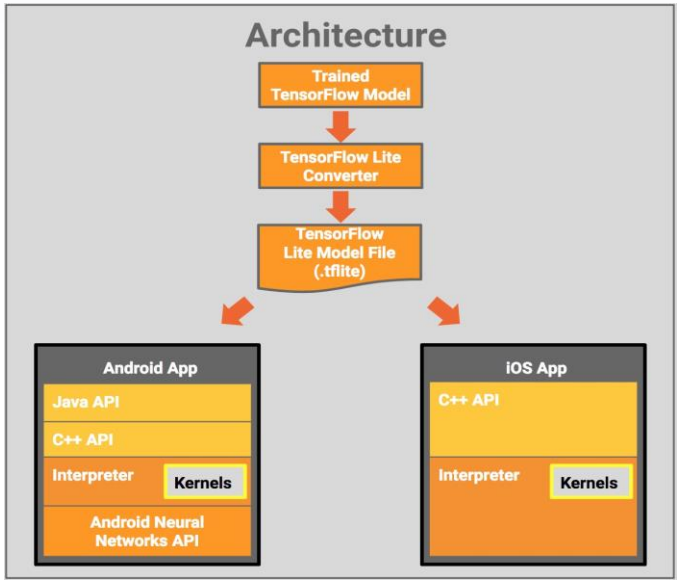
Firebase ML Kit

- Firebase offers ML Kit (Mobile SDK) to enable the power of Machine Learning on mobile devices.
- Offers powerful easy-to-use Computer Vision and Natural Language APIs.
- Some common use-cases are Text Recognition, Face detection, Barcode scanning, Image labelling, object detection etc.
- Firebase also provides a **Custom model API**.

Custom Model API

- Custom Model API allows you to use your own model.
- ML Kit supports only models created on TensorFlow framework.

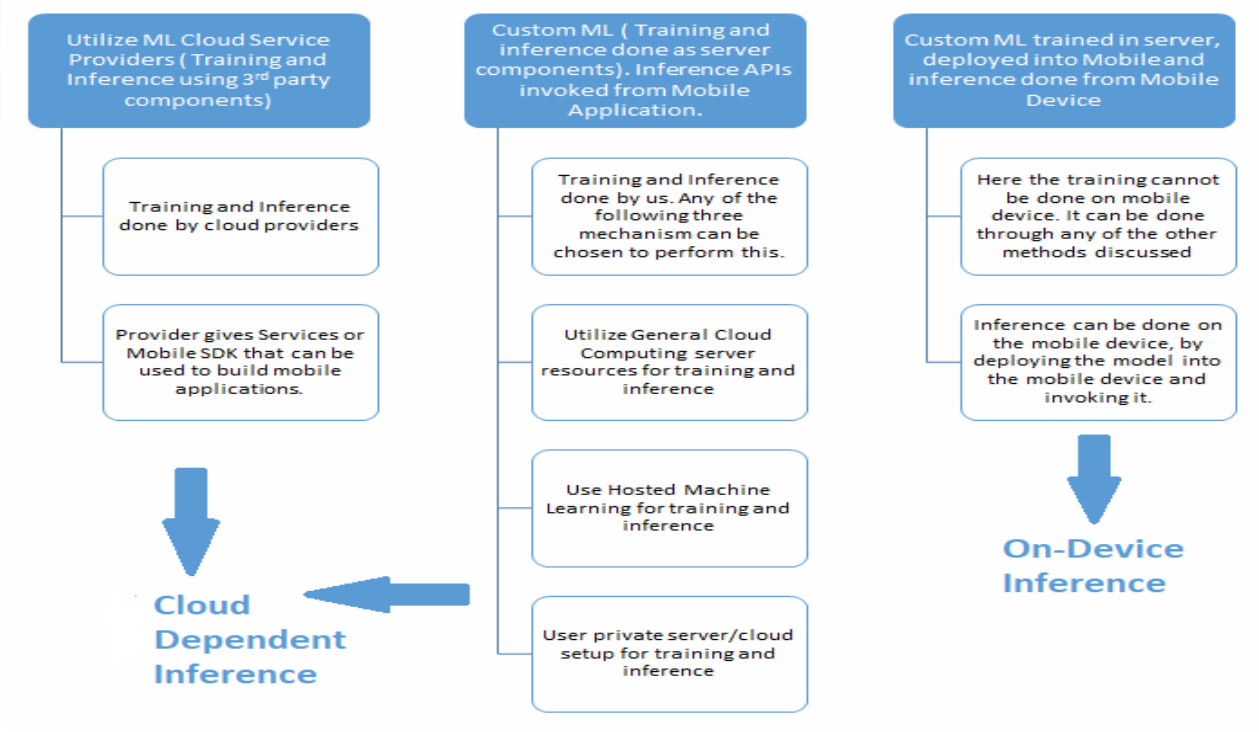
Architecture



TensorFlow Lite Inference

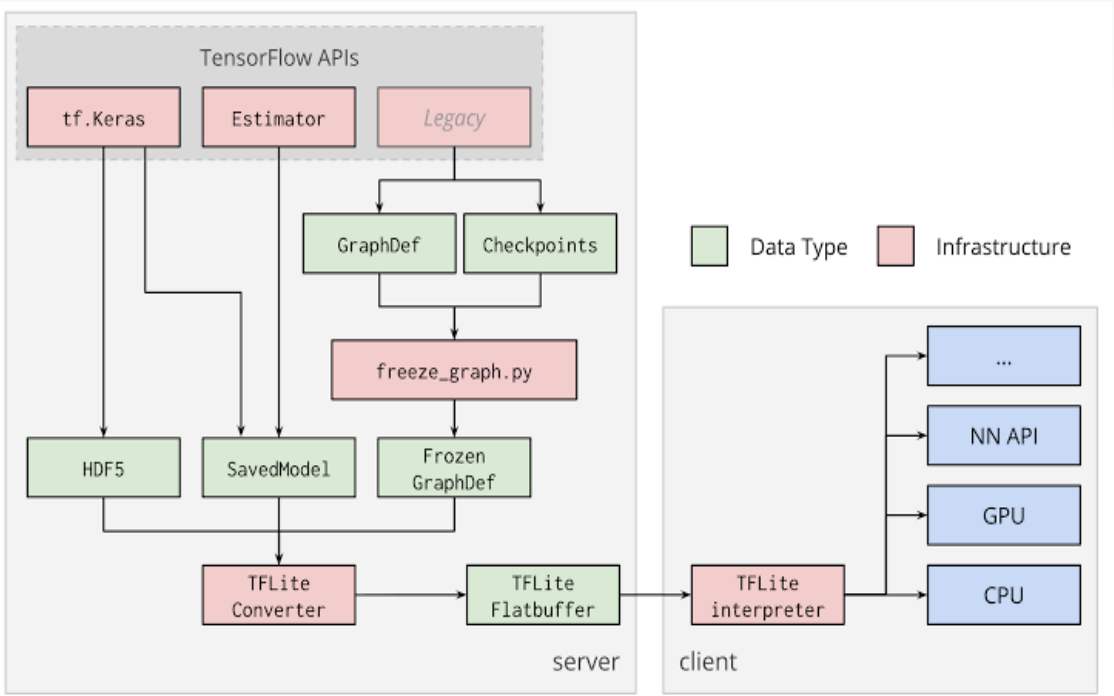
- Process of executing a TF Lite model on-device is called **Inference**.

Types of Inference



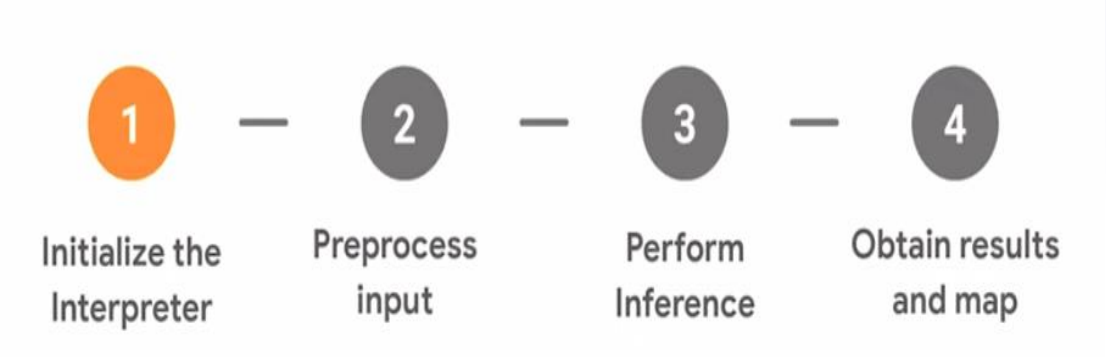
TF Lite Interpreter

- To perform inference with a TF Lite model, you must run it through an ***Interpreter.***
- Interpreter dynamically allocates memory, ensures minimal load and faster initialization with lower latency.



Steps for Inference

- Loading the TF Lite model into memory
- Build & Initialize Interpreter
- Input Tensor
- Perform Inference
- Output Tensor



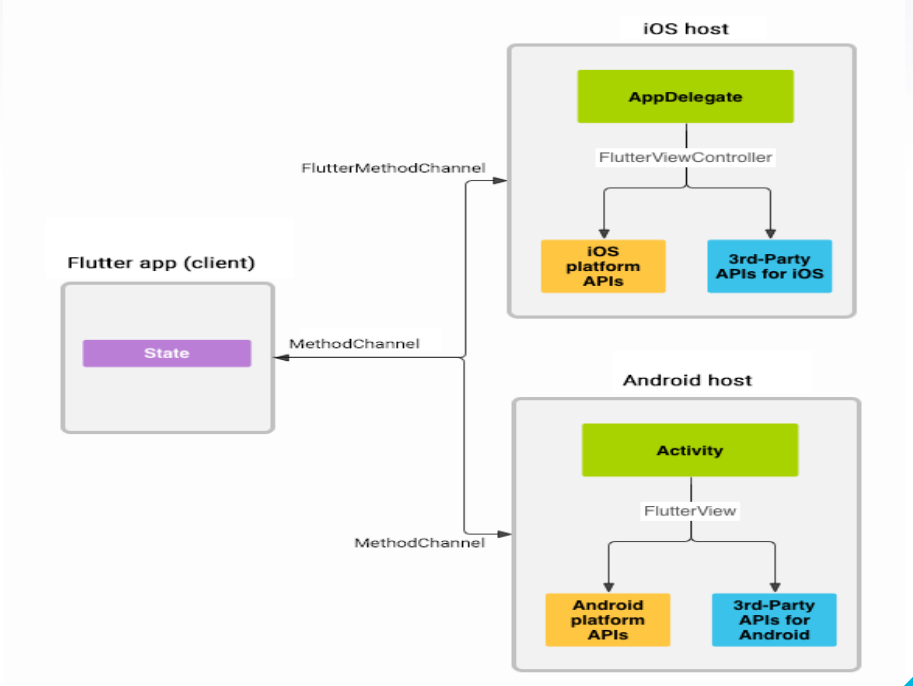
Method Channels

Interpreter supports inference using only programming languages like C++, Java, and Python.

This is a problem because our application code is written in **Dart**.

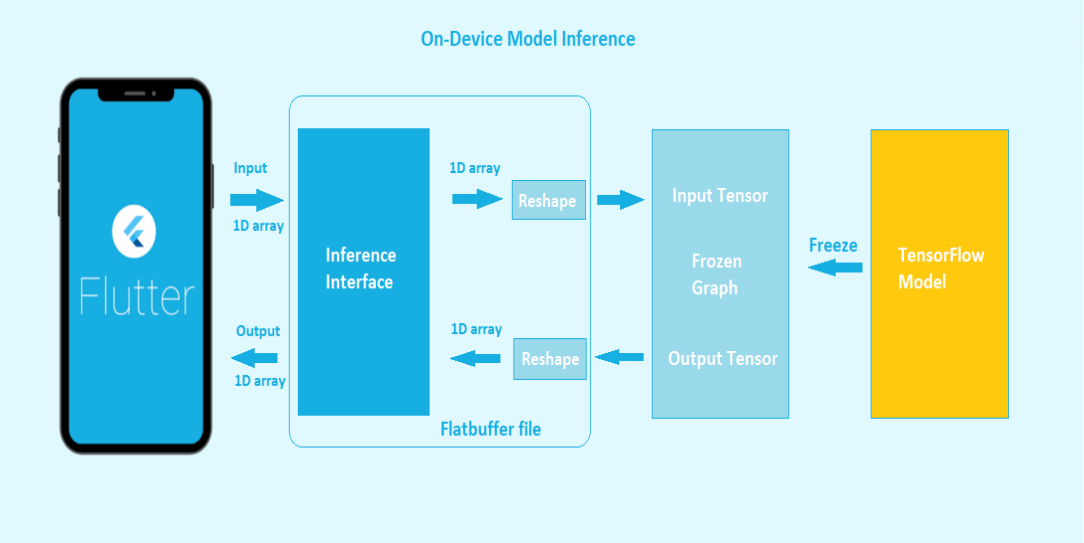
Solution → Method Channels of Flutter

- Method channels allow you to call platform-specific API's
- These method channels directly bind to TFLite C API.
- Passing messages between flutter and other hosts like Android, iOS.
- Lower Latency
- Performance acceleration using Multi-threading



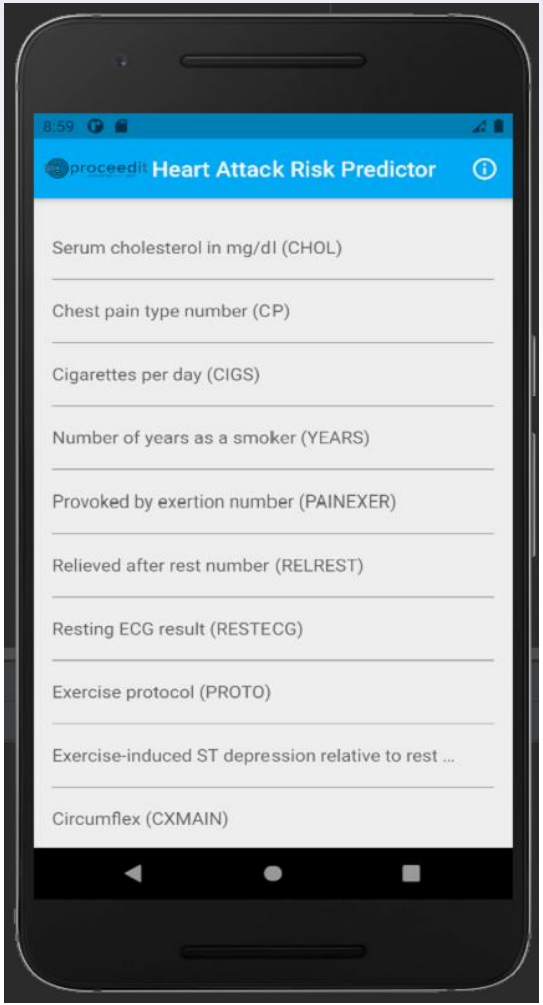
Connectivity

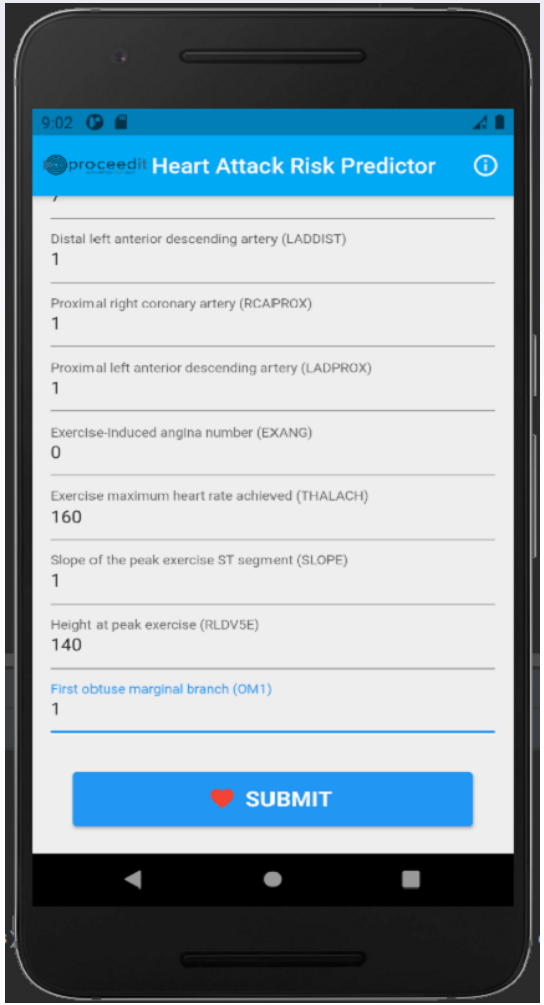
- Without Internet [On-Device Inference]



Screenshots 1

- Welcome Screen [Left]
- Inputting Data [Right]





Screenshots 2

- Results Screen 1
  Without Connection
  [Left]

- Results Screen 2
  Without Connection
  [Right]