

Improving application performance using GETX & MVVM



Presented By

Kumawat Lakhan Makhanlal (1906055)

Under the supervision of

Mr. Gourav Goyat

Senior Manager Troovi

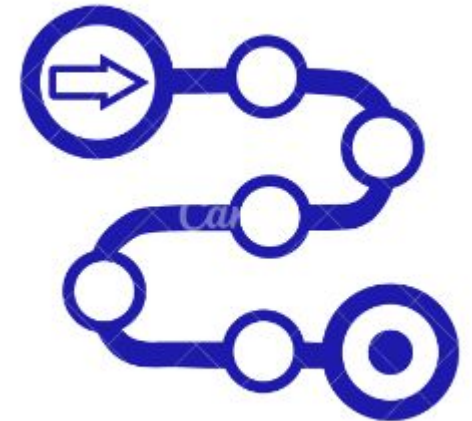
Department of Computer Science & Engineering
National Institute of Technology Patna



December 2, 2022

Outline

1. Introduction
2. Problem Statement
3. Proposed Solution
4. Implementation Results
5. Conclusion
6. References



Introduction

- This is an app based on the Indian logistics sector by digitizing the manual processes and paperwork involved in the day-to-day activities of transporters, fleet owners, and brokers.
- If you are a fleet or truck owner or you are a company managing transport services then troovi is the best fit for you , it lets you manage all your brokers , fleets owners , trucks , charges , road trips and balance all at one place , just like khata book but in transport system but with more added features .



Introduction

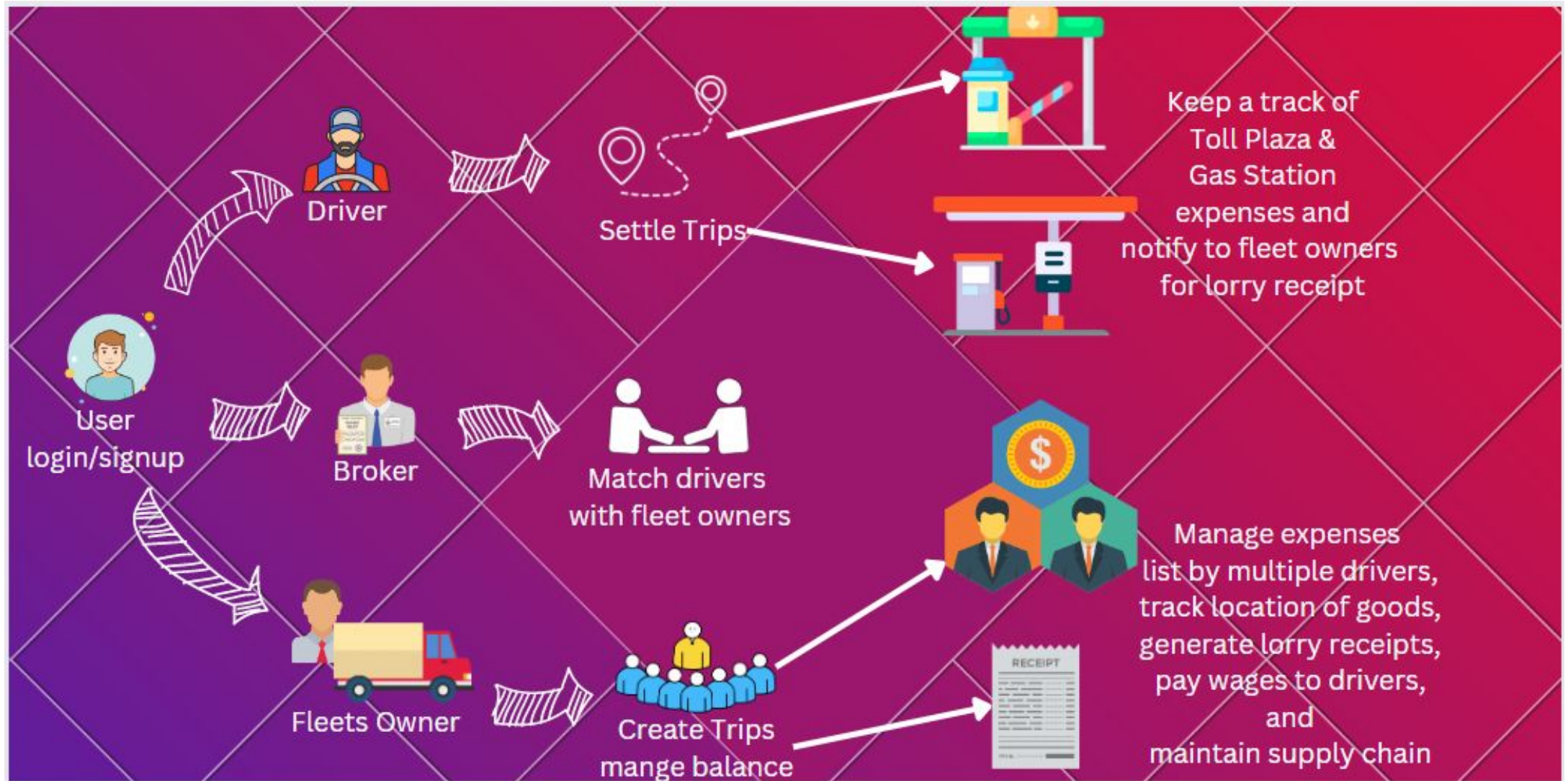


Fig 1.1 Working Diagram of Drivers,Brokers and Fleets owners

Tools & Technology Used

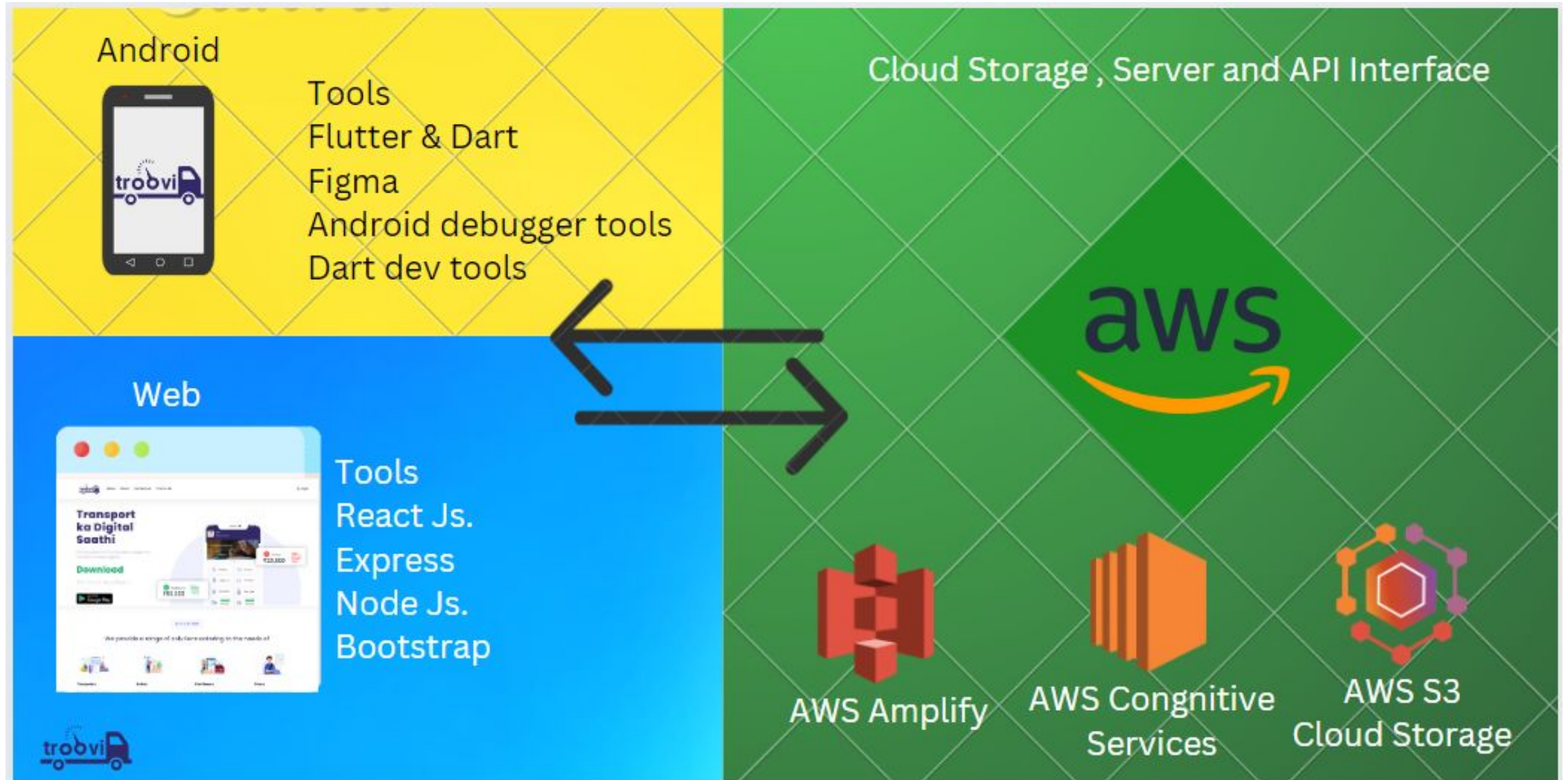


Figure 1.2 Framework Diagram

Problem Statement

Perform analysis and develop a solution for app optimization, refactor app architecture and integrate new better plugins to minimize intrinsic operations.



Proposed Solution

Use of GETX Flutter with MVVM Architecture

3.1 Architecture

Model-View- ViewModel (MVVM), Model-View-Presenter (MVP), and Model-View-Controller

The aim of these architectures is to separate the responsibilities of visualizing, processing, and data management for UI applications.

They help in Modularity

- Flexibility
- Testability



Proposed Solution

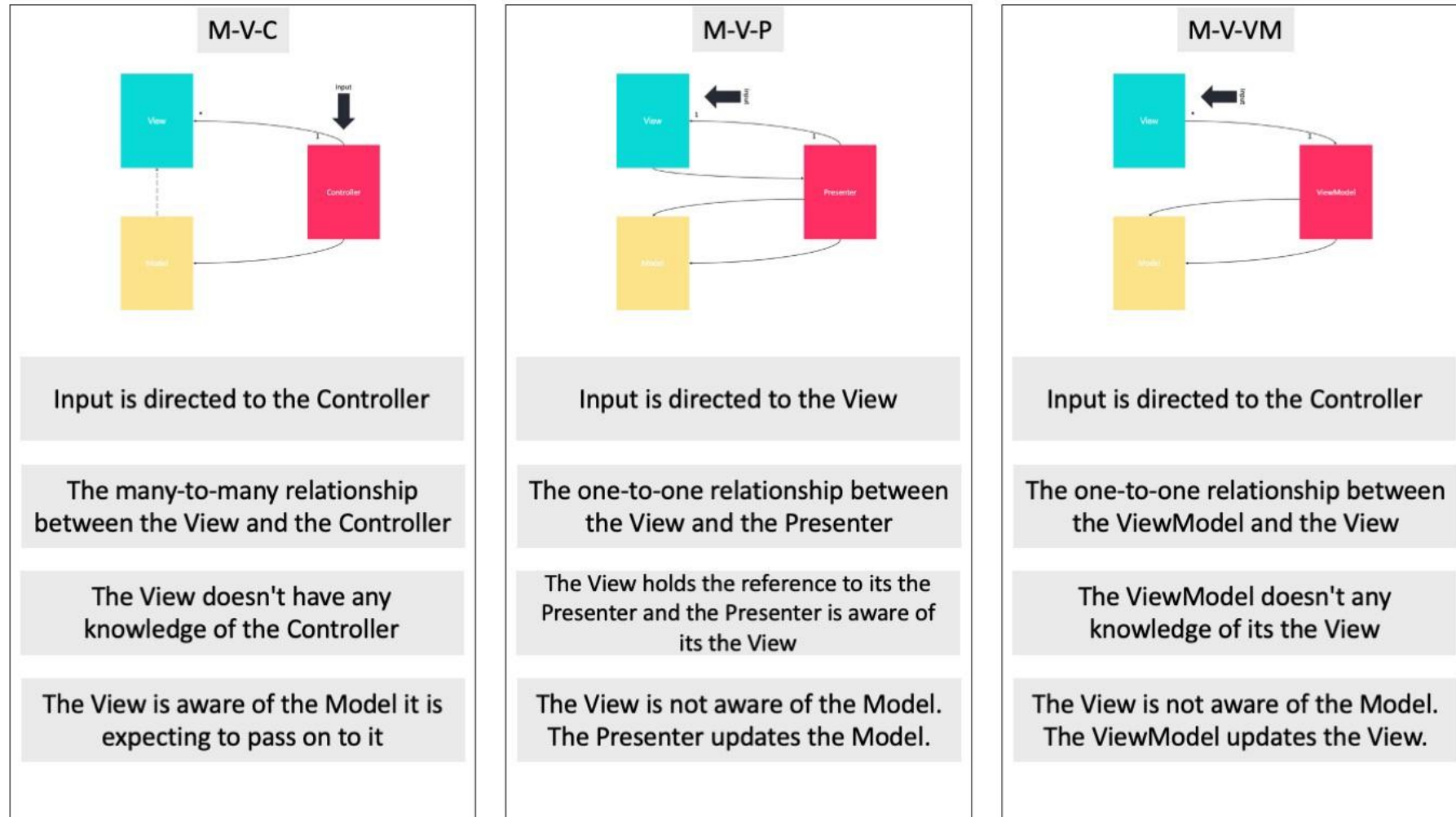


Fig 3.1 : MVC vs MVP vs MVVM Comparison

Proposed Solution

3.2 GETX Flutter

GetX is an extra-light and powerful solution for Flutter. It combines high-performance state management, intelligent dependency injection, and route management quickly and practically.

GetX has 3 basic principles which are listed below:

PERFORMANCE: GetX is focused on performance and minimum consumption of resources. GetX does not use Streams or Change Notifier.

PRODUCTIVITY: GetX uses an easy and pleasant syntax. No matter what you want to do, there is always an easier way with GetX.

ORGANIZATION: GetX allows the total decoupling of the View, presentation logic, business logic, dependency injection, and navigation.

Proposed Solution

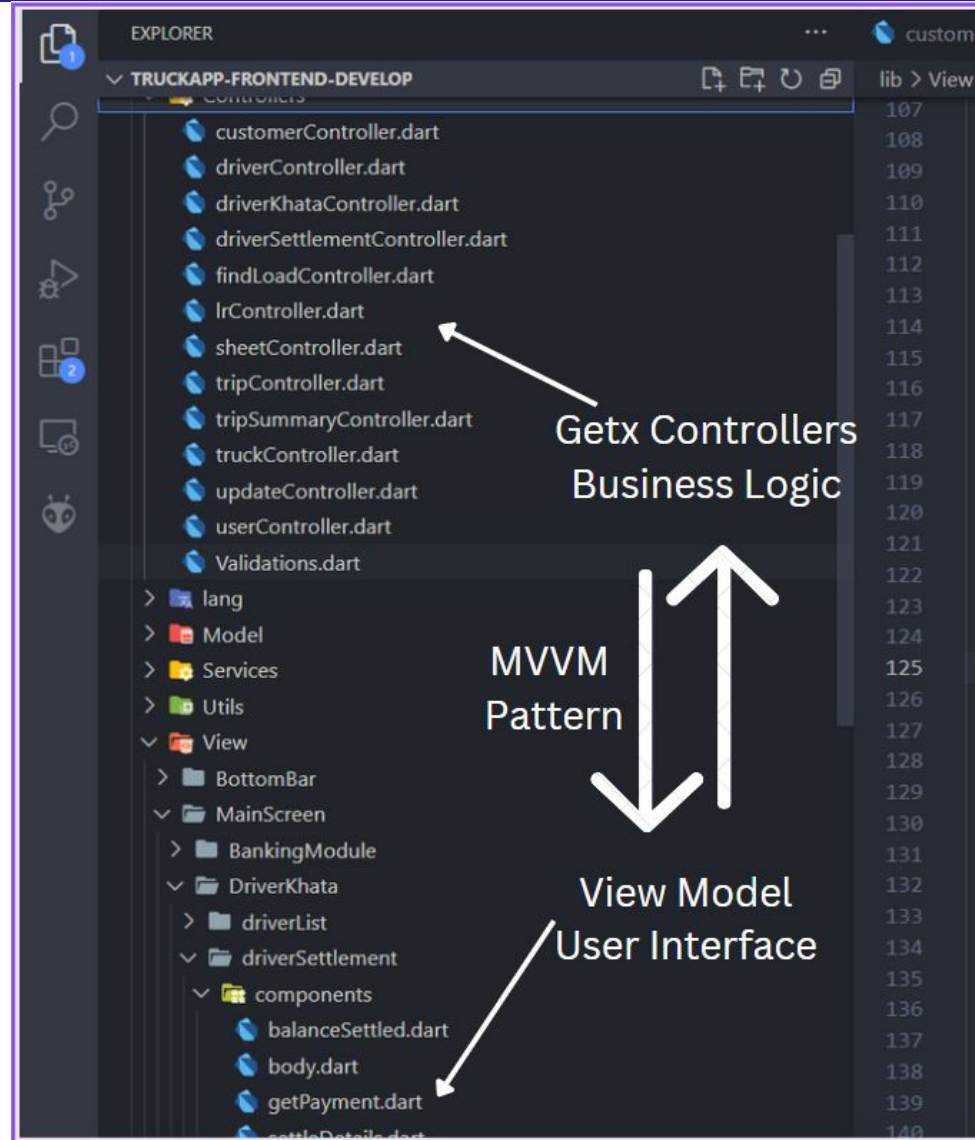
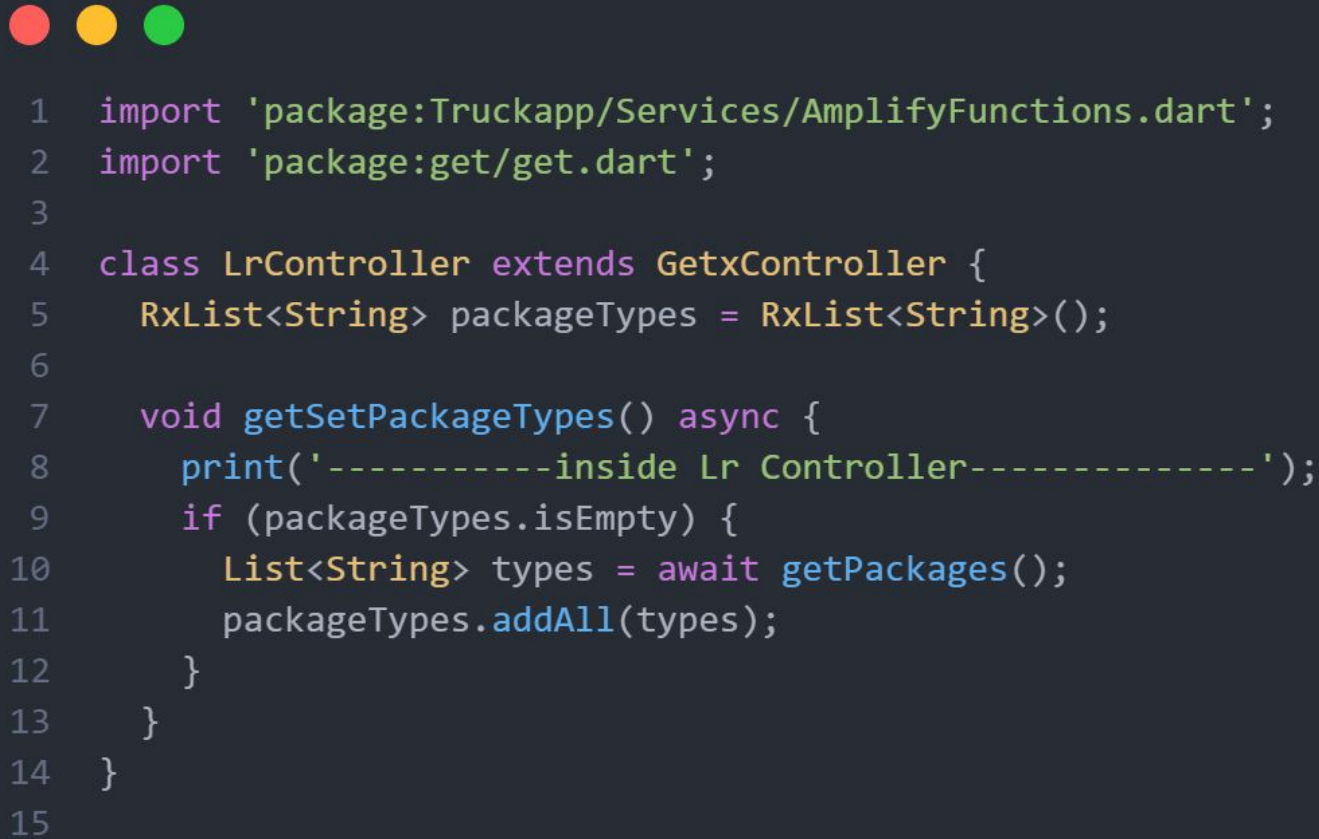


Fig 3.2 : MVVM Style pattern in folder structure and modeling

Proposed Solution



```
1  import 'package:Truckapp/Services/AmplifyFunctions.dart';
2  import 'package:get/get.dart';
3
4  class LrController extends GetxController {
5      RxList<String> packageTypes = RxList<String>();
6
7      void getSetPackageTypes() async {
8          print('-----inside Lr Controller-----');
9          if (packageTypes.isEmpty) {
10              List<String> types = await getPackages();
11              packageTypes.addAll(types);
12          }
13      }
14  }
15
```

Fig 3.3 : GETX Class Controller

Implementation Results

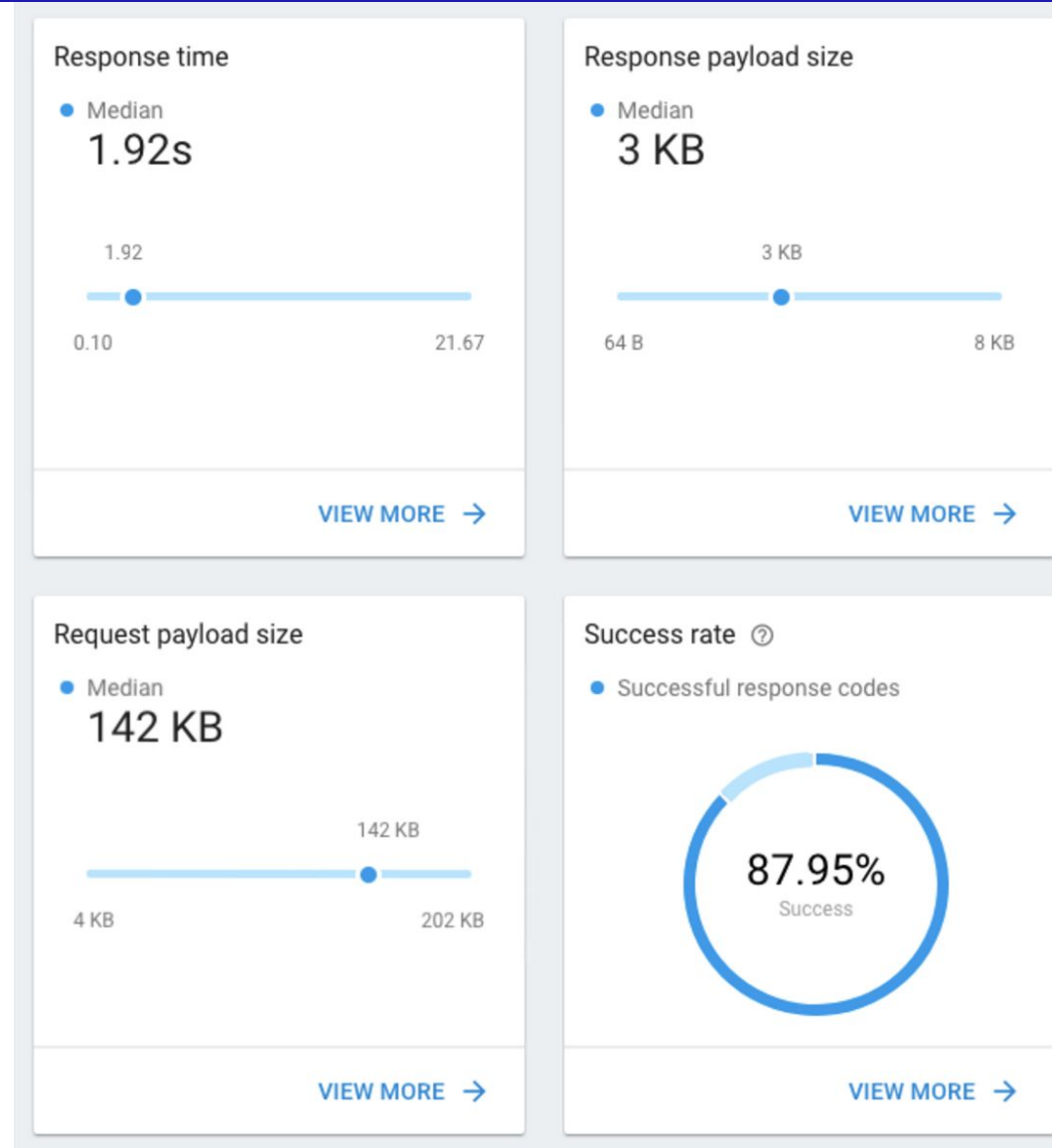


Fig 4.1 : Performance Analysis on Firebase

Implementation Results

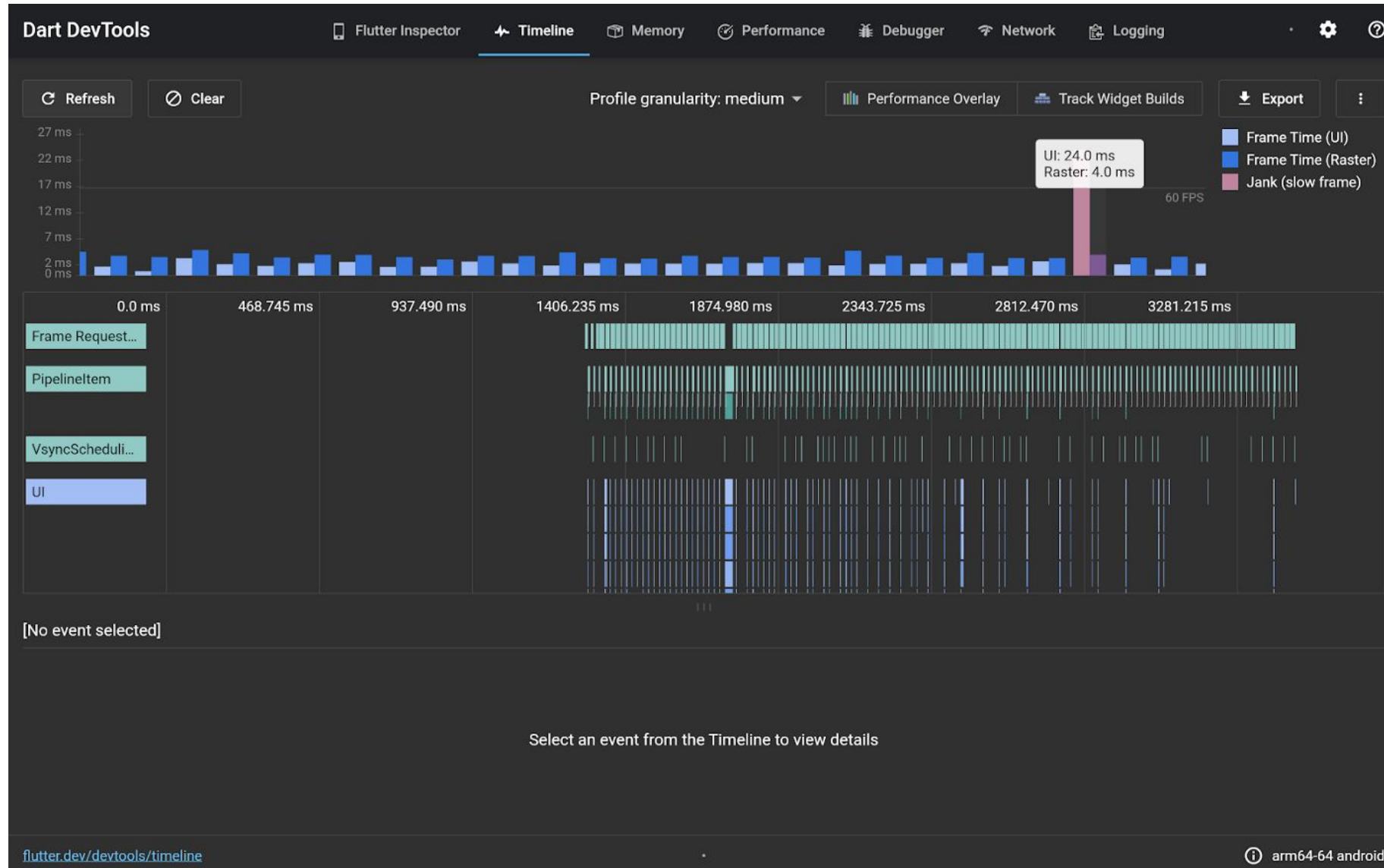


Fig 4.1 : Dart Dev Tools Performance Analysis

Conclusion

Given the influence these technologies have on our modern societies and crucial infrastructure, research and development efforts addressing the security and optimization of an application are unquestionably of paramount importance. I tried to deduce, measure and provide solutions to improve app performance and reduce the complexity. Improve app user interface and user experience.

References

- [\[1\] Jonataslow et al. designed the Getx architecture library in flutter.](#)
- [\[2\] Jitesh Mohite et al. Getting started with MVVM architecture in flutter.](#)
- [\[3\] Fanhar Tanvir et al. Flutter best practices for improving Performance.](#)

THANK
YOU