

Implementation of Hybrid Programming Approach Using Python for Cross Platforms Mobile and Web Application in Cloud Environment

¹Mihir Rajesh Panchal, ²Prinkal Hetalkumar Doshi, ³Neeta Gajanan Kadukar

^{1,2}Department of Information Technology, SVKM's Shri Bhagubhai Mafatlal Polytechnic, Mumbai, India

³Head of Information Technology, Vice Principal, Dept. of Information Technology, SVKM's Shri Bhagubhai Mafatlal Polytechnic, Mumbai, India

Abstract - Mobile and Web applications are used worldwide by individuals with drastic and emerging needs of data management and productivity. As a matter of fact, many mobile and web applications are developed in needs of individuals across the globe. These needs comprises of a number of fields like security, communication, entertainment, shopping, social media, etc. With the increasing demand of these needs and development of new fields in future, development of these mobile and web applications must fulfill them. As of now, rapid and agile development of mobile and web application along with platform independency for users is in demand. Along with that, access of these services anywhere, anytime, anyplace is also a need for end users and as and when required end user can enjoy the services. In order to provide these services technologies like Python can be integrated with Dart and frontend frameworks like Vue.js can be used in a cloud environment.

In this paper, we have ventured upon the architecture for hybrid programming of python with mobile and web application technologies in a cloud environment. Likewise, we have presented in depth comparison and layers of mobile and web application technologies. This paper provides implementation of the proposed architecture by using hybrid programming.

Keywords: Cloud Computing, Flutter, Vue.js, React.js, Dart, Python, Hybrid Programming, JavaScript, Django, DartPy.

I. INTRODUCTION

A shared pool of configurable computing resources which enables suitable software development environment on demand through a network channel is called as cloud computing. Services including storage, servers, databases, security, and networking are available through cloud computing. The transmission of these computer services through the internet, or "cloud," allows quick innovation, resource sharing that is simple and inexpensive per person.

[1]. Python is a high level ,multipurpose ,object oriented programming language designed to write more logic with less code through abundant pool of libraries. Python programming language can be implemented to make full stack web applications as Python provides bundle framework libraries like Django and Flask. These libraries have strong integrations with web APIs and cloud services which can help make an environment for full stack web applications. As of now Python has not much libraries suitable for mobile application development.

II. COMPARISON OF TECHNOLOGIES

2.1 Mobile Application Development

A mobile application often uses a network connection to speak with other external computer resources. Mobile application development is the process of developing software applications for portable electronic devices. Mobile apps are no longer limited to a single platform thanks to Flutter. The Table 1 shows the detailed comparison between the features of different types of mobile application [11][3][4]. The fields marked in red are the suitable parameters to develop a mobile application efficiently. The pace of developing secure mobile applications with less use of non available resources and capital will ensure dynamic and quality mobile applications [14]. The most advantageous type of architecture for mobile application development is Widget based Cross Platform Application development using flutter [11]. If somehow we can add the advantages of other mobile application architectures in Flutter we can develop and improvise the current flutter architecture. This can be done by integrating the use of Python Programming language in the flutter's dart environment.

As of now, the number of python developers in the market are increasing as python is an easy to learn and use programming language. As python is easy to use, non python developers can easily pick up python language and in turn will decrease the development cost of the mobile applications by

hiring diverse pool of developers. Python also an open source programming language and is much similar to dart as both languages are developed using C programming language. The security and performance of Python is respectable in comparison to other programming languages which are used in mobile application development. Integration of Python with dart will lead to a semi high security and performance level. Python's Contribution to open source is huge as it provides its developers to publish libraries on PyPi and makes a cloud based interface to create read and use libraries for all python developers.

Python when compared to Dart and Kotlin is not specifically made to develop mobile applications. Whereas Dart and Kotlin programming languages are made specific for mobile application development. Python language is much familiar to developers around the globe when compared to languages like dart and kotlin. But due to the fact that python lacks easy integration of SDKs with app development in android and ios, therefore it needs a hybrid approach to implement the advantages of python in application development. Interpreter-type languages are fundamentally not supported by iOS or Android. This explains why making Python mobile applications was uncommon in the past. But now that numerous frameworks have emerged, things are different. PyGame, PyOpenGL, Arcade, Panda3D, Java Rx libraries can be used to make the Python software run natively on mobile devices.

2.2 Web Application Development

Web application development is the construction of application programs for web use. These programs are sent to the user's device via the internet and are stored on remote servers. These applications are stored on distant servers and sent through the internet to the user's device. A web application may be viewed across a network without being downloaded. Moreover in web application development, serverless architecture enables software developers to build and sustain products without worrying about the underlying infrastructure.

The Table 2 shows the detailed comparison between the Javascript frameworks used in web application development [5]. Out of the frequently used web application frameworks vuejs and react js are the more efficient as they provide improvised functionality when compared to other frameworks [6]. The MVT of Django and Virtual DOM of Vuejs are easily compatible with each other and are efficient in performance standards [7][8].

III. HYBRID PROGRAMMING WITH DART

3.1 Flutter

Flutter is Google's mobile UI framework that provides fast app development with cross-platform support [2]. The programming language used to develop apps using the Flutter UI framework is Dart, a programming language developed by Google. Flutter apps are widget based and the widgets built are from a single codebase and are compiled directly to native arm code which can use the GPU and access platform APIs and services [2].

3.2 Integration of Dart with Python

Flutter is underused by developers who are less experienced in Dart programming language. Dart is currently in development and due to this not much features are available for the developers. Python on the other hand has a large community and tools like PyPI that offer developers to write and share libraries with each other. Integration of Dart with Python to provide functionalities and code written in python in flutter UI framework using Dart can be beneficial. Python supports some mobile application api staples like JSON responses XML processing modules etc. For Python to run in dart programming language a flutter installable library is required to be included in the environment of flutter. The library should call out python codes by using Python C-API embedding and dart:ffi to call native C APIs.

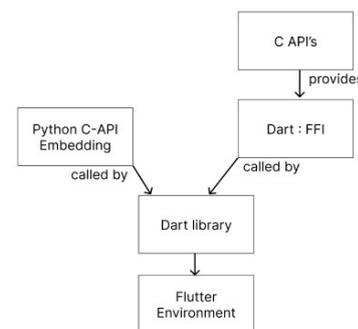


Figure 1: Integration of Dart with Python

In the diagram above, the dart library calls Python CAPI Embedding and dart:ffi that helps embed python code in extendable c/c++ format style of coding which is under stable to dart programming language's interpreters. The Dart library then can be imported in the flutter environment. Dynamic Pathing of Code can be easily done with encapsulation of dart libraries. A mixed programming language approach of Python and Dart can be applied but the configuration of python path with the dart library must be done in order to run python in dart library. This can be done by providing command line configuration while installing dart library and also via plugins with dart library. There is one limitation to this as not many

libraries are written in the mind that they will be used with the dart library. Having a separate cloud base which is similar to PyPI for uploading python code which gives functionality for dart programming must be created in order to import those libraries inside the dart code.

3.3 Cloud Integration

The python code which we can write in Flutter environment using dart is possible with the dart library. However the python code will require some additional dart library support which is not compatible with Python PyPI [12].

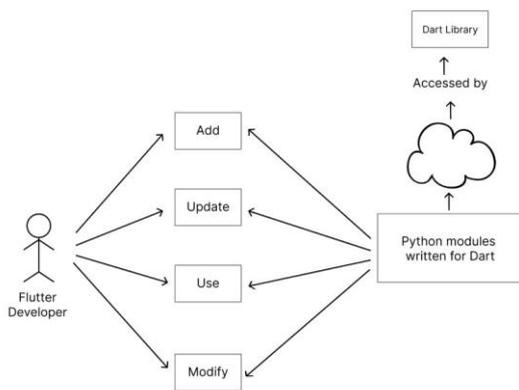


Figure 2: Cloud Integration with Dart

In order to make it compatible the python codes for specific dart functionality can be shared in a separate cloud sharing repository. Developers can share, add, modify those python modules in the repository on the cloud [10]. Also by this the python code in the cloud can directly be accessed in the dart library via cloud fetching APIs [12]. The common mobile app utilities like accessing the camera, storage etc can be easily done by importing the python modules made for dart through the dart library. By extending the Python interpreter to the dart’s environment we can access some unique functionalities of python.

3.4 Flutter Web

Flutter is used to create cross platform apps along with web development support due to dart’s convention of treating elements of code in a tree based widget system. However flutter web is still in development and has some disadvantages when compared to react native. This is because Dart is not much flexible with other web technologies like JQuery, AJAX and JS. Whereas React Native uses JavaScript to build its web application. Integrating Python/C API Embedding into Dart will allow Python to communicate with web technologies with its abundant pool of libraries while still being under the dart interpretation environment.

3.5 Data Layer

The data layer is the layer in which data storage based technologies are made use of to store user data for web application [13]. CRUD Operations are performed in request to client side by SQL and NOSQL operations. Web 3.0 uses a different storage mechanism called the blockchain wherein data is stored in a series of blocks. Data storage is structured in the means of data request so as to communicate with the application and client layer [10]. Tools like Postgres, MySQL, AWS, Azure, Digital Ocean, Polygon are used in data layer.

IV. HYBRID PROGRAMMING WITH VUE JS

4.1 Vue JS

Vue JS is a JavaScript framework for building frontend User Interfaces. Vue JS is a modern web technology to make dynamic web pages and web applications. Vue JS is used frequently by developers of open source community for creating web applications. Vue JS provides a way to build components that encapsulate data or state in your JavaScript script and then connect that state to a template. Vue.js also has great connectivity with backend technologies like Python Django, Node.js, Golang, PHP.

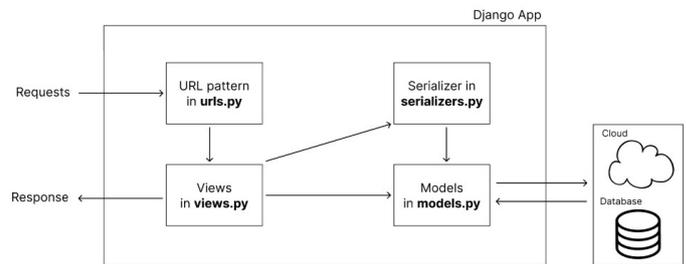


Figure 3: Analysis Vista Backend

4.2 Django MVT Integration with Vue.js

Vue.js connects the View and Model via two way data bindings [6]. The Model View Controller requires a data communicator to extract and write data for Vue JS. Backend Technology like Python Django can be used as a link communicator between the models and views of web application. Django provides a bundle of code for Vue.js to call view models via routes [8][9]. These routes are in the form of web urls fired along with some request. The Model View Template helps maintain data in the data storages with Python’s data storage connecting libraries. Data Objects are returned as a JSON format and can be rendered by Vue.js and Python Libraries dynamically. Django in return also provides security of data from breaches and rapid responses from the data storages [9].

4.3 Cloud Integration with Vue.js

Vue JS when integrated with backend technology like Python Django can be integrated with cloud technologies [10]. Django web applications can easily be deployed on cloud by additional cloud based Python libraries. Django also supports a lot of data storage tools like PostgreSQL ,MySQL and even MSSQL. These can be integrated with cloud by Django databases serverless with some configuration. Serving Containers of Cloud can be accessed by Django via cloud interface APIs through cloud maintenance tools. Scalable and Secure web applications can be created with Vue.js and Python Django.

V. IMPLEMENTATION

5.1 Flutter

1) *Frontend:* The contemporary framework used to create Flutter widgets was modelled after React. The main concept is to create your user interface (UI) using widgets. Widgets provide descriptions of how their views should appear given their settings and current state. The framework then compares this description to the previous description to figure out what minimal changes are required in the underlying render tree to get from one state to the next. As seen in Figure 4, there are two types of Flutter widgets: stateful widgets and stateless widgets. [11]. The runApp() function takes the given Widget and makes it the root of the widget tree [11]. In our application, the widget tree consists of many different widgets. For instance, we have made use of Centre widget to set the profile picture in the centre and also Text widget to set the name email id of the person. The framework forces the root widget to cover the screen, which means the person’s name and its email id ends up centred on screen. A widget’s main job is to implement a build() function, which describes the widget in terms of other, lower-level widgets. The framework builds those widgets in turn until the process bottoms out in widgets that represent the underlying RenderObject, which computes and describes the geometry of the widget [12].

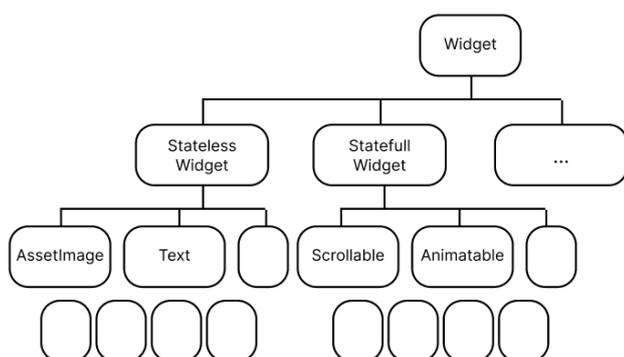


Figure 4: Widget Structure in Flutter

2) *Proposed Method:* Dart programming language provides the functionality of providing rich set of core libraries along with additional APIs available. But these libraries are bound to the dart programming language. Additionally to provide hybrid programming in dart programming language the DartPy additional API can be used to achieve hybrid programming in mobile application development. The DartPy library can be added to dart programming environment by importing it into the main.dart file. Additionally all the dependencies can be added to the project by the pubspec.yaml file. Below are the steps to follow to implement hybrid programming of Dart with Python.

- 1) Create a Flutter Project and open the terminal of your project.
- 2) Install the dartpy library by the following command: flutter pub add dartpy OR flutter pub add dartpy.
- 3) Run the pub get command and check the pubspec.yaml file for the dependency.
- 4) Import the library by adding the import statement to the dart file : import 'package:dartpy/dartpy.dart';

By following the above mentioned steps a flutter developer can import the library to implement hybrid programming of Dart with Python. After importing the library, flutter developers can use the methods in the DartPy Library by simply calling them. The dartpyc.Py Initialize(); line of code is used to initialize the python object instance in dart programming environment. The Python code can then be stored in a dart string variable. The dartstring variable is then converted to nativeutf8 by dart:ffi library. After the conversion the python code can be executed using the dartPy library. A sample code for hybrid programming of python is given below. 3) Source Code: Python is integrated with dart by the below given source code. The DartPy library is initialized and is made ready to use. The Python code is then written into a dart string.

```
void main(List<String> args) {
  dartpyc.Py_Initialize();
  final python = '''
  from time import time, ctime
  print("Today is", ctime(time()))
  ''';
  final pystring = python.toNativeUtf8();
  dartpyc.PyRun_SimpleString(pystring.cast<Int8*>());
  malloc.free(pystring);
  print(dartpyc.Py_FinalizeEx());
}
```

Figure 5: DartPy Source Code

The dart string is then converted into C strings encoded with UTF-8. The converted C string encoded with UTF-8 is then run with python interpreter through the dartpy library [15].The output from the python code is then made use by dart

for implementing mobile application features and functionality. 4) Backend: Dart can call out to Python code via the Dart:ffi function and the Python C-API embedding thanks to the dartpy library. In order to assist the Python garbage collector, you must manage the reference counts of Python objects through a somewhat low-level interface. There are many different plugins for play/pause/seek songs, fetching music metadata, and browsing music library. These plugins are built using iOS's MediaPlayer Framework and Android's MediaPlayer to fetch and play music from the user's music library.

5.2 Vue.js

AnalysisVista is a web application which helps users to create a shared team and collaborating between a repository with access to features like data analysis (.csv, .xlsx), text-to-speech (.docx, .pdf), speech-to-text (.wav, .mp3, .mp4). Users are given the ability to execute self-service data analytics over the cloud from any location, at any time, on any device. It can produce sophisticated visualisations that gather information from several database sources and transform that information into clear results.

1) *Frontend*: Technology used to create the frontend of this web application is Vue.js. The Vue.js framework offers efficient re-rendering, allowing the system to keep track of dependencies as they progress through the render. Vue.js is a framework that is favoured by the majority of software developers worldwide due to its re-rendering capability because it is its most distinctive feature. Figure 6 shows the main screens of the AnalysisVista web app – screens for creating new teams, allowing access rights to team members and uploading data for analysis.



Figure 6: Analysis Vista

2) *Backend*: AnalysisVista uses Django rest API and the MVT provided by Django to communicate between the different layers of web application architecture. The MVT also called as Model View Template provides bundle code in Django for web application to process JSON data with backend applications like the database and cloud [15].The client communicates with the data layer via the controller of MVC which is similar to MVT provided by Django [7][8]. The client does so by interacting with the frontend components made with VUE.js The Controller is used to receive request

send by client in the client layer. Request are received and are controlled by Django to map the request with the specific url in the urls.py file. The View is used to the display the dynamic content of the web page in respect to user's request. AnalysisVista showcases detailed data visualization in respect to the csv files uploaded. Jinja and Werkzeug format is used to embed data dynamically in web pages and components made in Vue.js. The Model is used to manipulate the data into backend services like database and cloud. The models are mentioned with their database characteristics in the models.py file of the Django app.

3) *Approach*: This web application's primary goal is to organise a vast amount of information in a graphical manner. As a result, we implemented an incremental and iterative strategy, in which each software build adds functionality incrementally and a workable build is provided at the end of each iteration. (see Fig. 9).

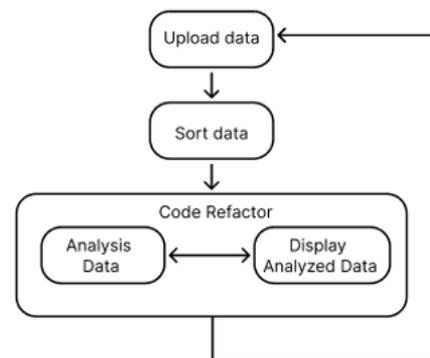


Figure 7: AnalysisVista Approach

This will allow us to address platform differences with minimal impact as they show up during iterations. Each iteration begins by collecting the data from the user and analysis of that data. Different types of analysis are performed on different file formats.

5.3 Python

Interpreter-type languages are fundamentally not supported by iOS or Android. It denotes the Python application cannot be executed natively. This explains why making Python mobile applications were uncommon in the past. But now that numerous frameworks have emerged, things are different. To make the Python software run natively on mobile devices, there are several Python GUI frameworks available. Python libraries like Django, TurboGears, Flask, Web2py give cross platform web development toolkits to develop web applications. Also python libraries like pygame, pyopengl, arcade, Panda3d give cross platform mobile game and application development kits.

VI. FUTURE SCOPE

There is a vast potential in the frameworks developed recently as they aim to advance technology into futuristic domains like Web 3.0, blockchain and AI. It is projected in the next years that these technologies will flourish even more. To make these technologies more viable to the developers of the current generation, a hybrid approach of the current generation technologies and futuristic technologies must be applied.

VII. CONCLUSION

In this paper we ventured about the hybrid approach of implementing python with mobile and web application development in a cloud environment. We also proposed architecture of hybrid programming of Dart with python along with the layers of mobile application development. In the same way we implemented hybrid programming of python with Vue.js to create complex web application using technologies of Web 3.0.

REFERENCES

- [1] Pankaj D. Rathod, Vanshita C. Shah, Chintan J. Dodia, "Venturing Cloud Surveillance in Concur to Deep Learning with Artificial Intelligence" Volume 5, Issue 12, pp 54-59, December-2021.
- [2] Yoonsik Cheon, Carlos Chavez, "Converting Android Native Apps to Flutter Cross- Platform Apps", 2021 International Conference on Computational Science and Computational Intelligence (CSCI).
- [3] Peixin Que, Xiao Guo, Maokun Zhu, "A Comprehensive Comparison Between Hybrid and Native App Paradigms" 2016 8th International Conference on Computational Intelligence and Communication Networks.
- [4] Ivano Malavolta, "Web-based Hybrid Mobile Apps: State of the Practice and Research Opportunities" 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems.
- [5] Ramandeep Kaur, Bhanu Sharma "Comparative Study for Evaluating the Usability of Web Based Applications" 2018 4th International Conference on Computing Sciences.
- [6] Cornelia Mihaela, Ovidiu Constantin, Raluca Marina, Mircea Ioan, Gy'ongyi BUJDOS'Ó and Camelia Maria DINDELEGAN, "Comparative study of some applications made in the Vue.js and React.js frameworks" 2021 16th International Conference on Engineering of Modern Electric Systems (EMES).
- [7] Mohammad Robihul Mufid, Arif Basofi, M. Udin Harun Al Rasyid, Indhi Farhandika Rochimansyah, Abdul rokhim "Design an MVC Modele using Python for Flask Framework Development". 2019 International Electronics Symposium (IES).
- [8] D.V. Annenkov, E.A. Cherkashin "Generation Technique for Django MVC Web Framework using the Stratego Transformation Language" MIPRO 2013, May 20-24, 2013, Opatija, Croatia.
- [9] Chunsheng Song, Ruiping Huo, Shuqing Wang, Changzhi Lv, "Transformer Equipment Temperature Monitoring Based on the Network Framework of Django" 2019 Chinese Automation Congress (CAC).
- [10] Liangchen Guo, Yazhong Zhang, Chang Lu, Yinan Jing, Zhenying He, X. Sean Wang "A System for Exploratory Analysis in Cloud" 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom).
- [11] Kewal Shah, Harsh Sinha, Prof. Payal Mishra "Analysis of Cross- Platform Mobile App Development Tools" Pune, India. Mar 29-31, 2019.
- [12] S. Durai, C. Shyamalakumari, Dr. T. Sujithra "Cloud Computing based Multipurpose E-Service Application using Flutter" IEEE Xplore Part Number: CFP22K25-ART; ISBN: 978-1-6654-1028-1.
- [13] Enrique Jim'enez Domingo, Javier Torres Ni'no, Angel Lagares Lemos, Miguel Lagares Lemos, Ricardo Colomo Palacios, Juan Miguel, G'omez Berb'is "CLOUDIO: A Cloud Computing-oriented Multi-Tenant Architecture for Business Information Systems" 2010 IEEE 3rd International Conference on Cloud Computing.
- [14] Stefan Bosnic, Istvan Papp and Sebastian Novak "The development of hybrid mobile applications with Apache Cordova" unpublished.
- [15] Yanuarizki Amanatullah, Charles Lim, Heru Purnomo Ipung, Arka vJuliandri "Toward Cloud Computing Reference Architecture: Cloud Service Management Perspective" International Conference on ICT for Smart Society.
- [16] Dart Py Package, "https://pub.dev/packages/dartpy", Accessed on 25- September-2022.

Citation of this Article:

Mihir Rajesh Panchal, Prinkal Hetalkumar Doshi, Neeta Gajanan Kadukar, “Implementation of Hybrid Programming Approach Using Python for Cross Platforms Mobile and Web Application in Cloud Environment” Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 8, pp 58-64, August 2023. Article DOI <https://doi.org/10.47001/IRJIET/2023.708008>
