APPLICATION ARCHITECTURE

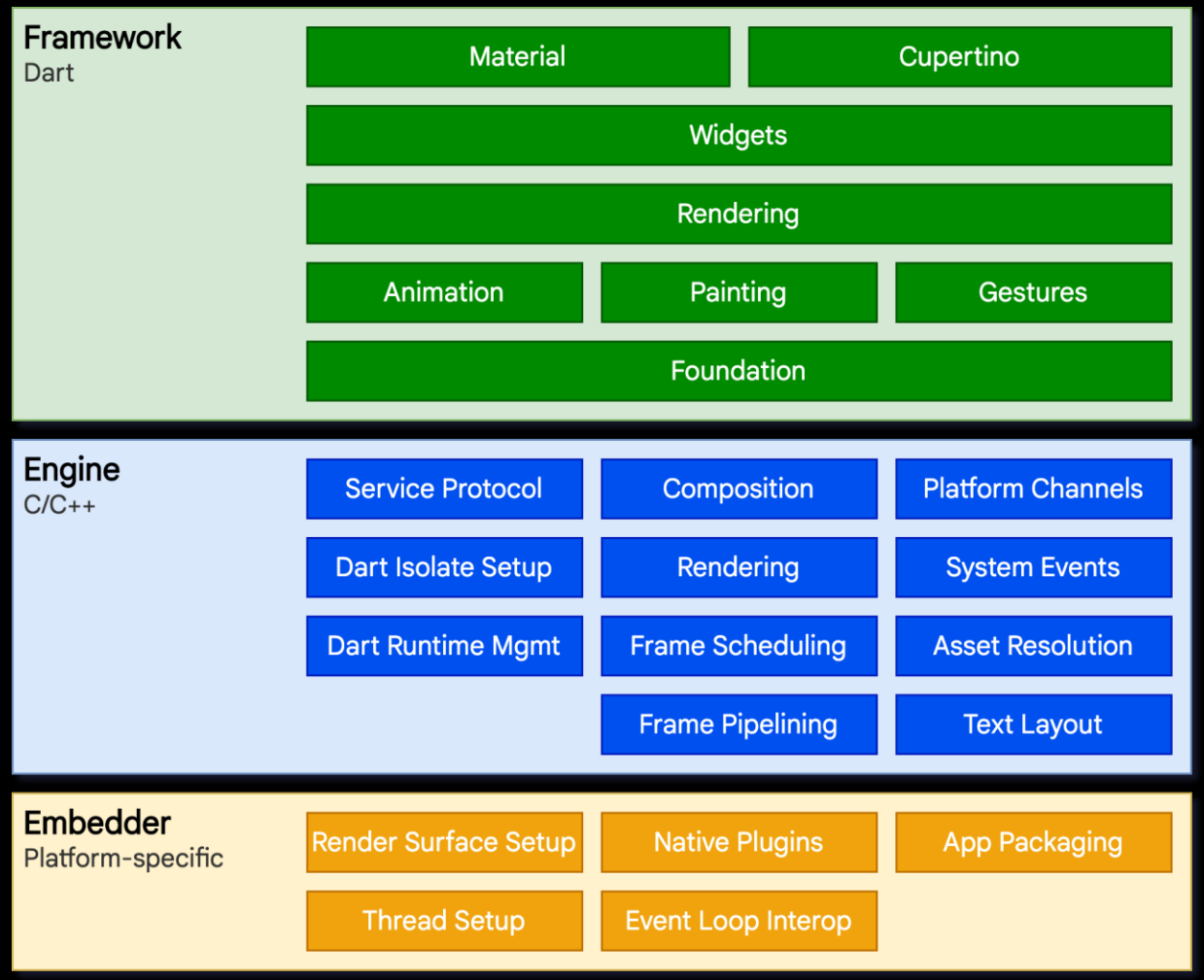
- DevBees

OVERVIEW

- Flutter architecture
- MVC
- MVP
- MVVM

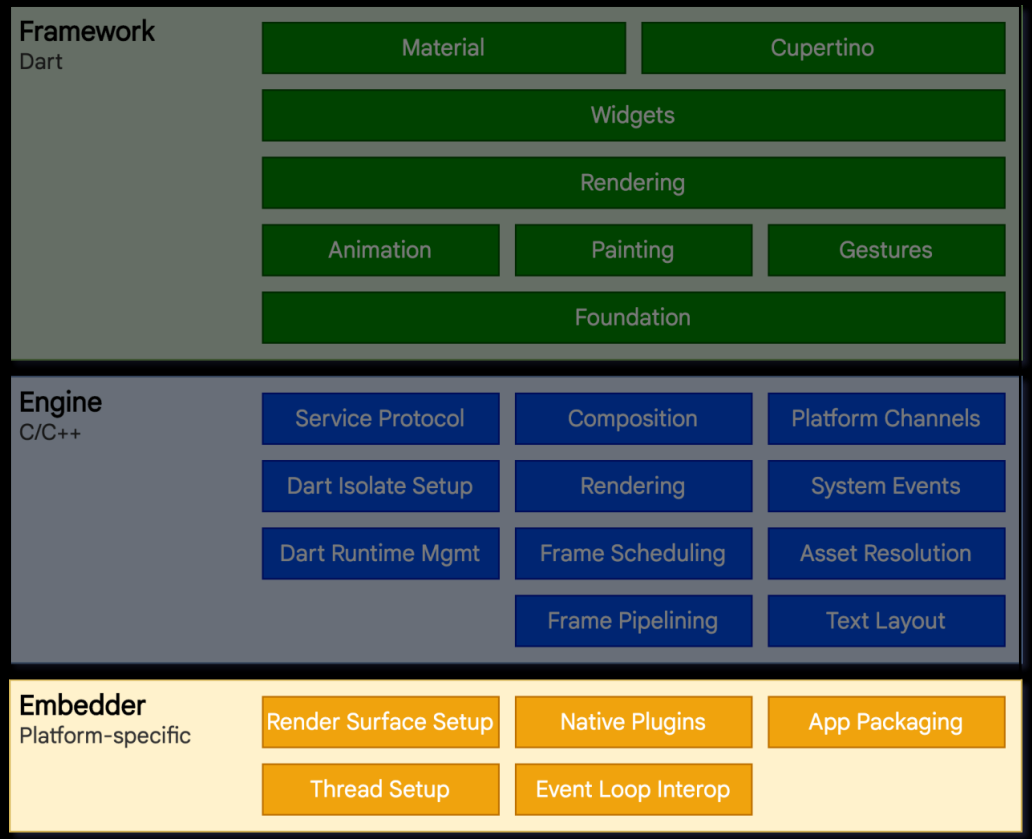
FLUTTER ARCHITECTURE

- Embedder
- Engine
- Framework



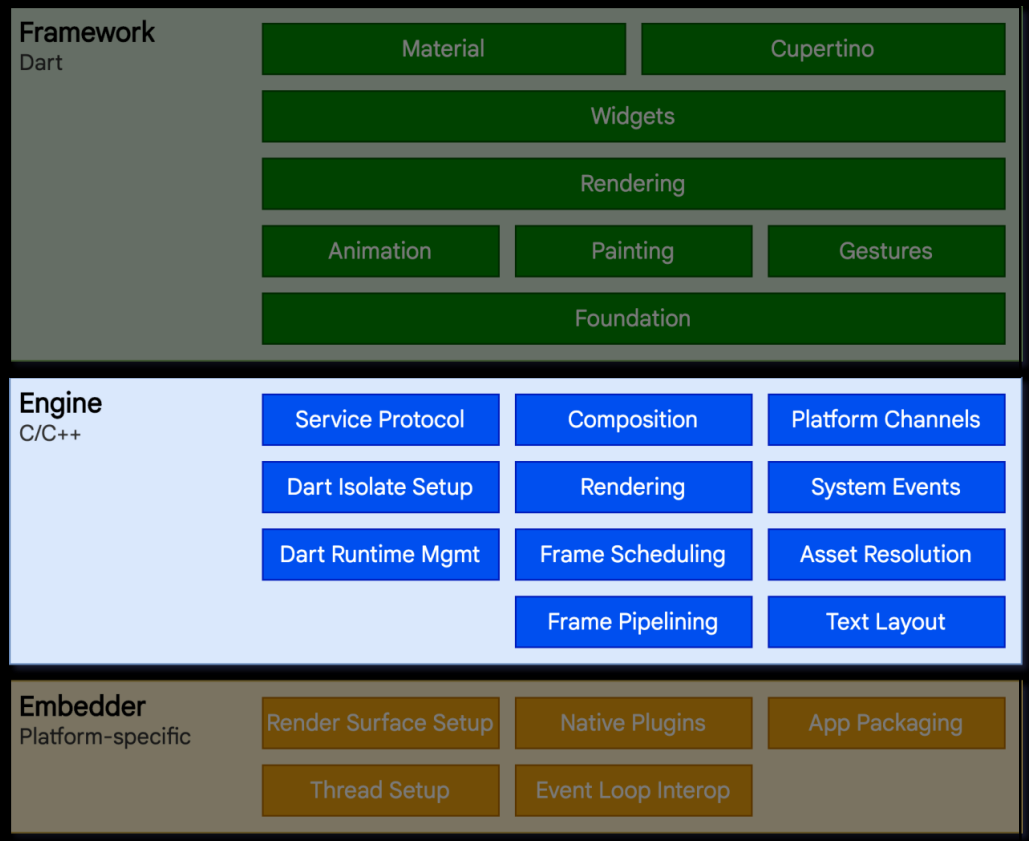
FLUTTER ARCHITECTURE

- **Embedder**
  - Lowest layer
  - Platform specific entry point
  - Co-ordinates with underlying OS to access services like rendering surfaces, accessing and inputs
  - Languages
    - Android : JAVA/C++
    - IoS & MacOS : Objective C/Objective C++
    - Windows & Linux : C++



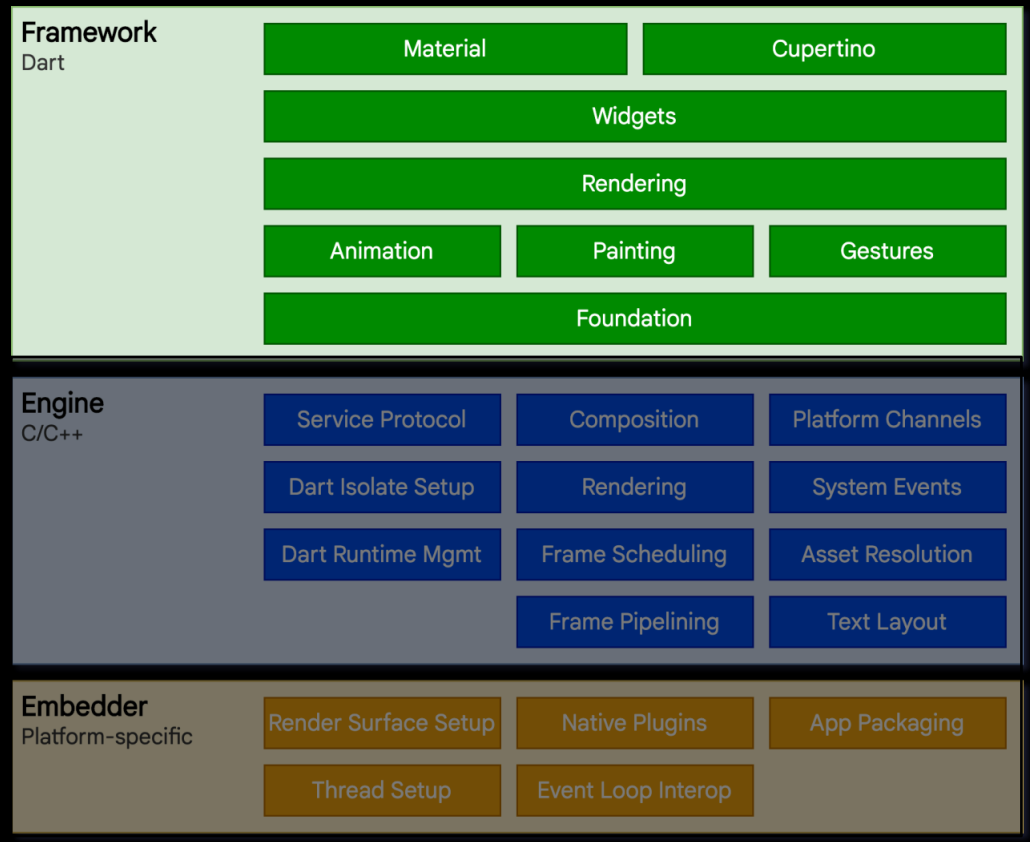
FLUTTER ARCHITECTURE

- **Engine**
  - Written in C++ language
  - Takes care of
    - Input,
    - Output,
    - Network requests
    - Rendering whenever frame needs to be painted
  - Rendering Engine : Skia
  - Revealed to framework via dart : ui



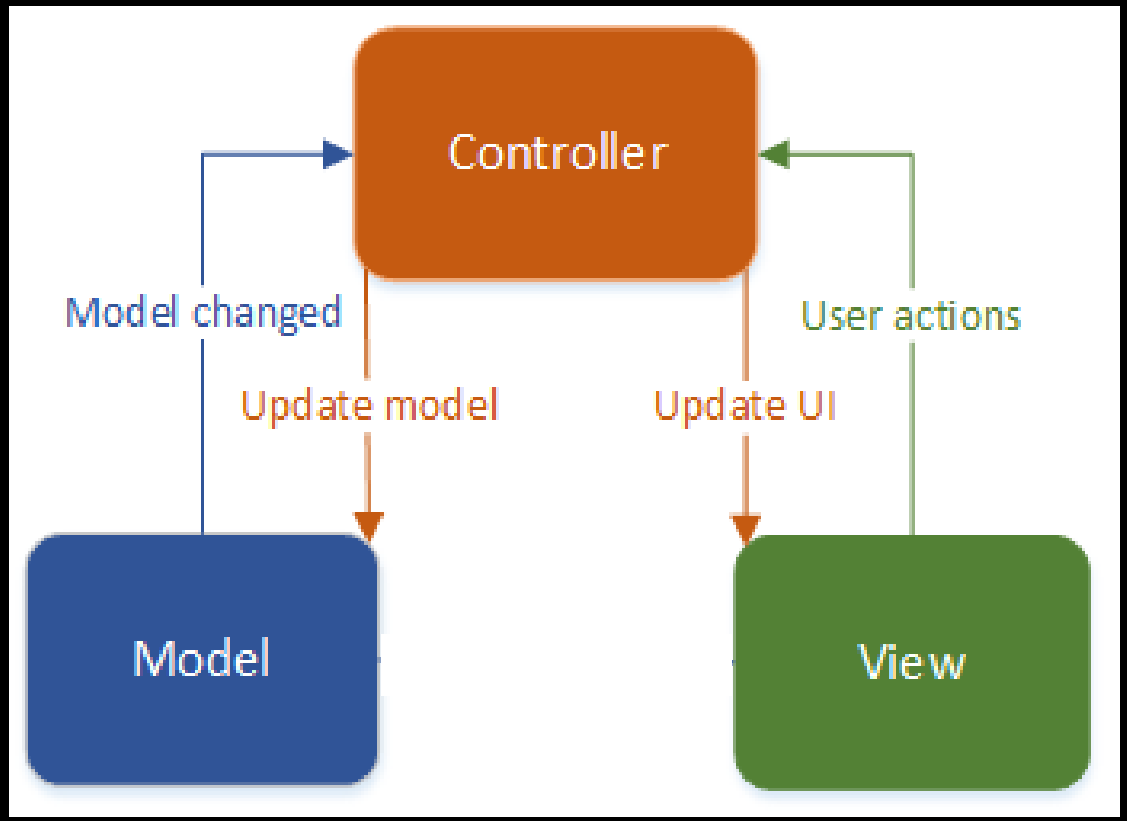
FLUTTER ARCHITECTURE

- **Framework**
  - Developers interact with Flutter
  - Modern reactive framework written in Dart
  - Comprises mainly of
    - Widgets
    - Rendering
    - Cupertino/Material
    - Animation
    - Painting
    - Gestures
  - Required for writing Flutter applications



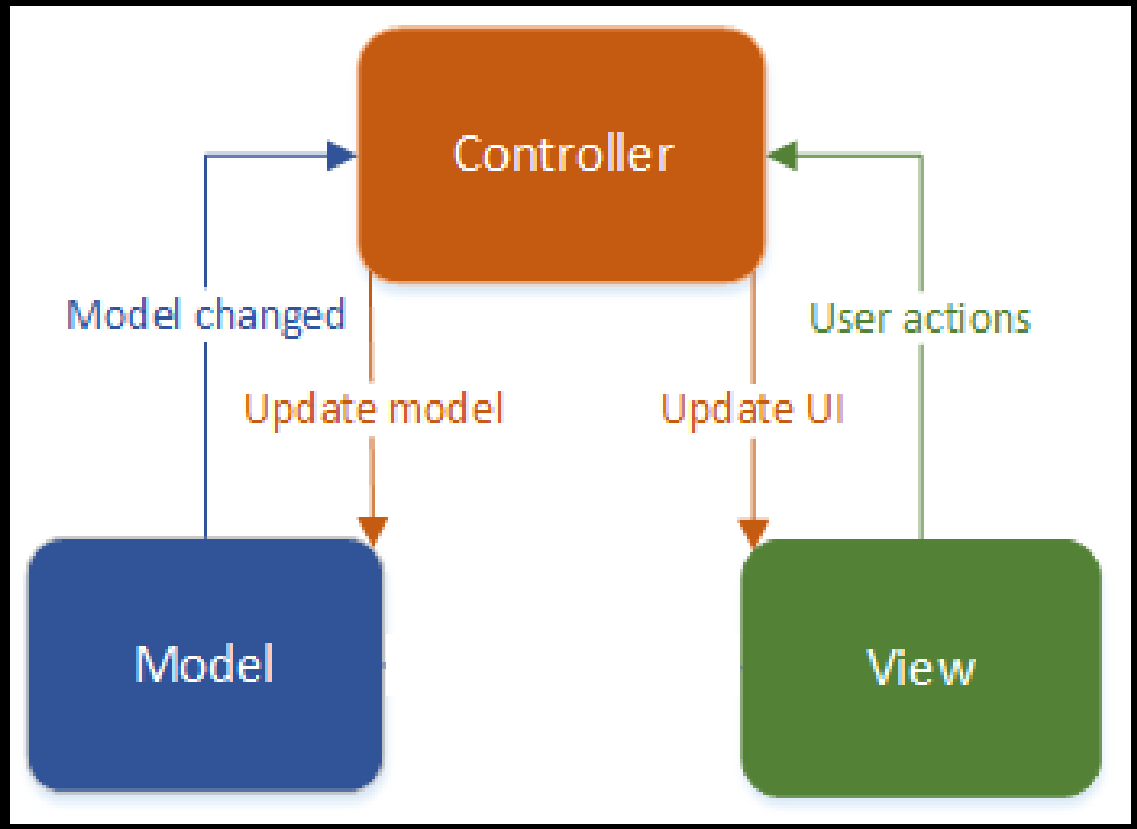
MVC

- **M**odel **V**iew **C**ontroller
- Design pattern architecture
- Model : The data
- View : The UI
- Controller : The event handler
- Controller in MVC accesses View components and so is not independent



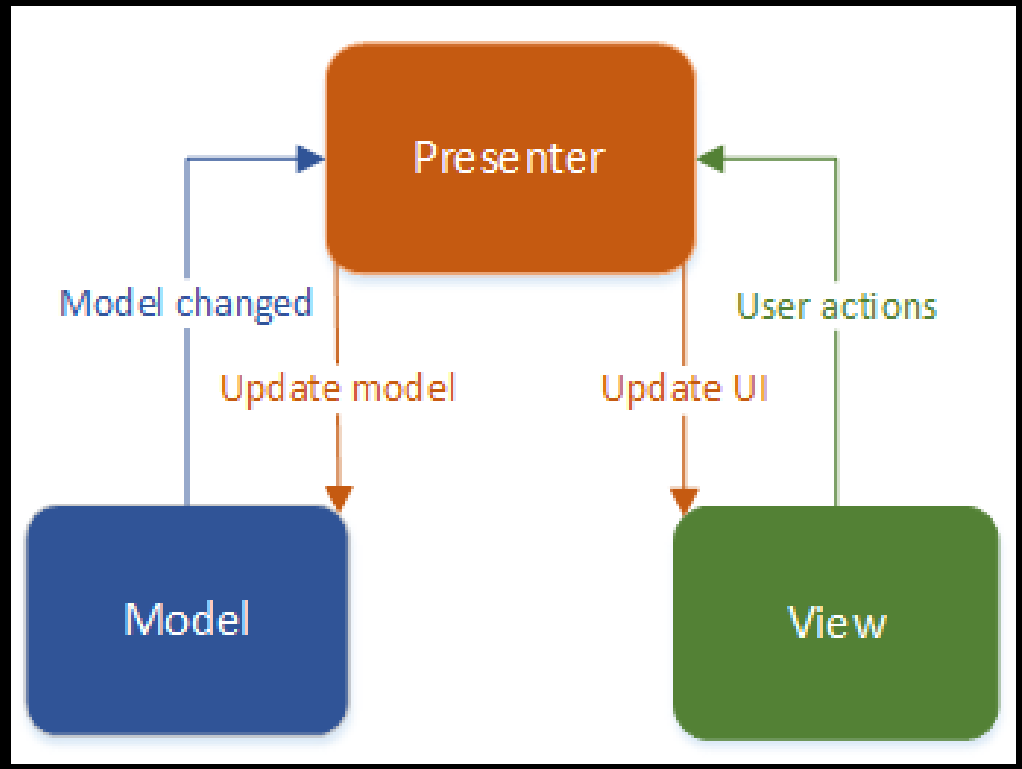
MVC in Flutter

- MVC structure in flutter can be implemented using the packages and corresponding File structures
- Packages
  - mvc_pattern: ^8.12.0
  - mvc_application: ^8.13.1
- Detailed example
  - Reference 1
  - Reference 2
  - Video Tutorial



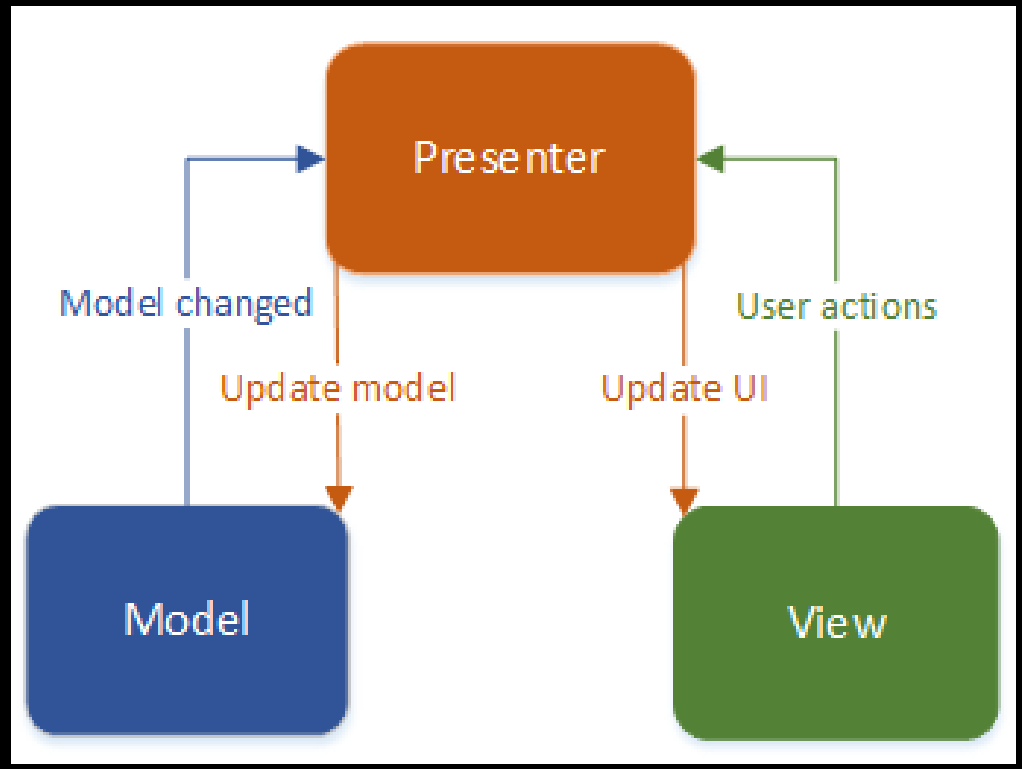
MVP

- **M**odel **V**iew **P**resenter
- Modified form of MVC
- Model : The data
- View : The UI
- Presenter : Presentation logic
- Presenter and View are completely independent and communicate via **interfaces**



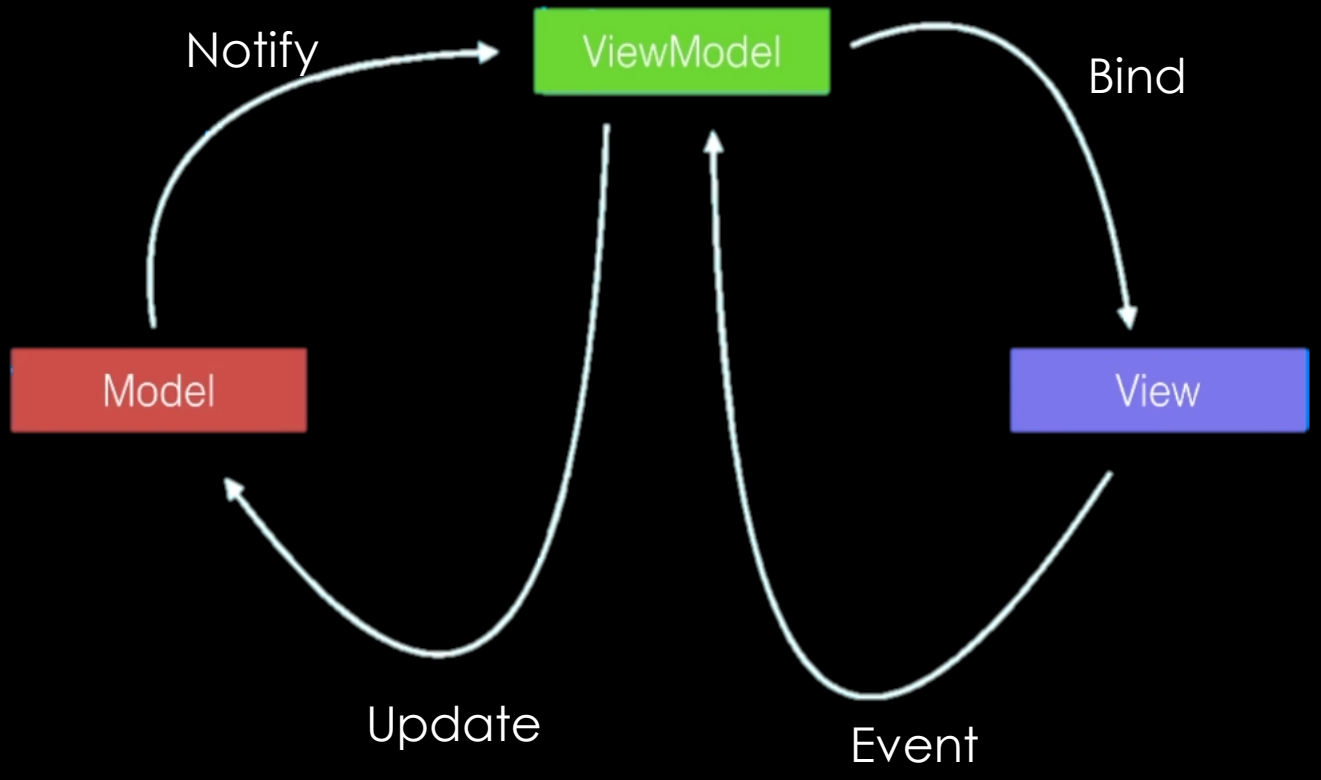
MVP in Flutter

- MVP in flutter can be implemented with or without any related packages
- For detailed referencing
  - Package
    - mvp: ^1.0.0
  - Tutorials
    - Example tutorial 1
    - Example tutorial 2
    - Video tutorial



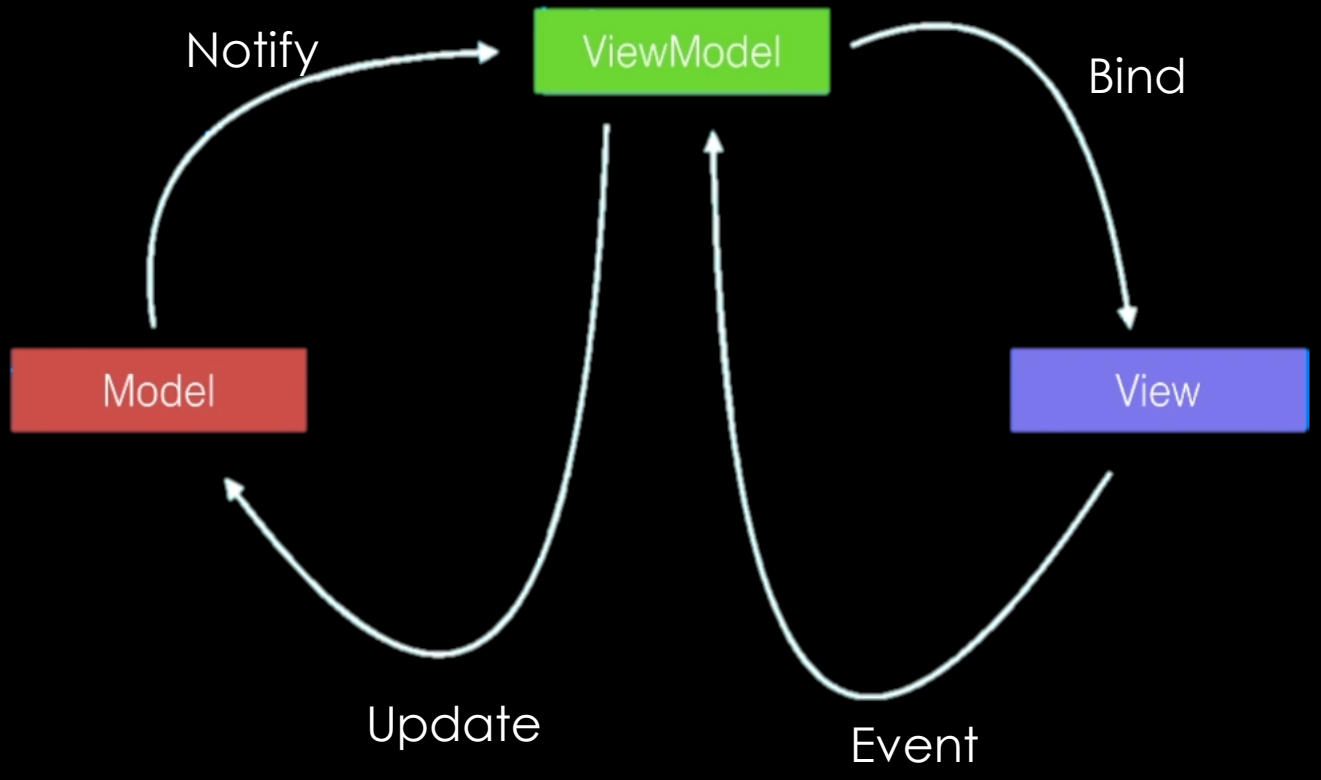
MVVM

- **M**odel **V**iew **V**iew**M**odel
- Model : The data
- View : The UI
- ViewModel : The mediator
- View and Model interact via the ViewModel



MVVM in Flutter

- MVVM can be implemented in Flutter using Provider state management or packages
- For complete detailed tutorial refer
  - Tutorials
    - Tutorial example 1
    - Tutorial Example 2
    - Tutorial Example 3
    - Video tutorial 1
    - Video tutorial 2
  - Package
    - mvvm: ^0.5.2



THANK YOU

- DevBees