

[Open in app](#)[Get started](#)

Published in Level Up Coding

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)



Anh T. Dang

[Follow](#)Sep 27, 2020 · 5 min read ★ · [Listen](#)

Save



THOUGHT PROGRAMMER

# MVC vs MVP vs MVVM

## What's different between them?



[Open in app](#)[Get started](#)Photo by [Edwin Andrade](#) on [Unsplash](#)

Nowadays we have many options when it comes to architecture design patterns. After developing many apps using Model-View- ViewModel (MVVM), Model-View-Presenter (MVP), and Model-View-Controller (MVC), I finally feel qualified to talk about the differences between them. For easier to understand, we can use a simple example like developing a screen to search books in the **BookSearch App**.

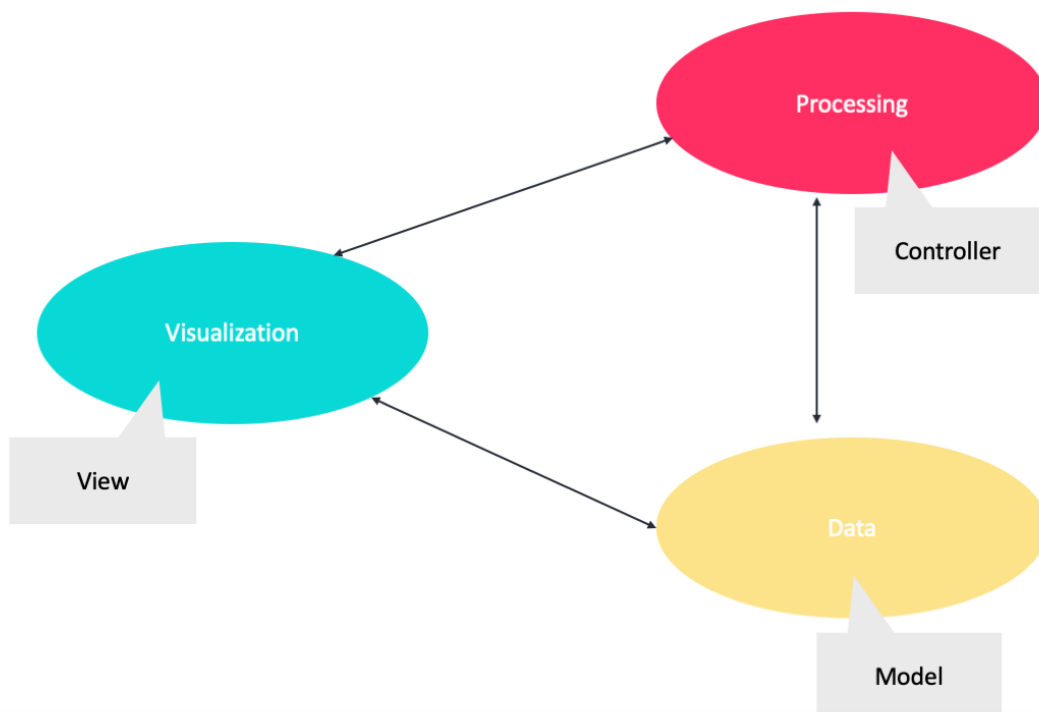
Let's begin now...!

## MV(X) essentials

First of all, we need to understand briefly MVC, MVP, and MVVM architecture before dive into them.

### Why Model-View-(C or P or VM)?

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

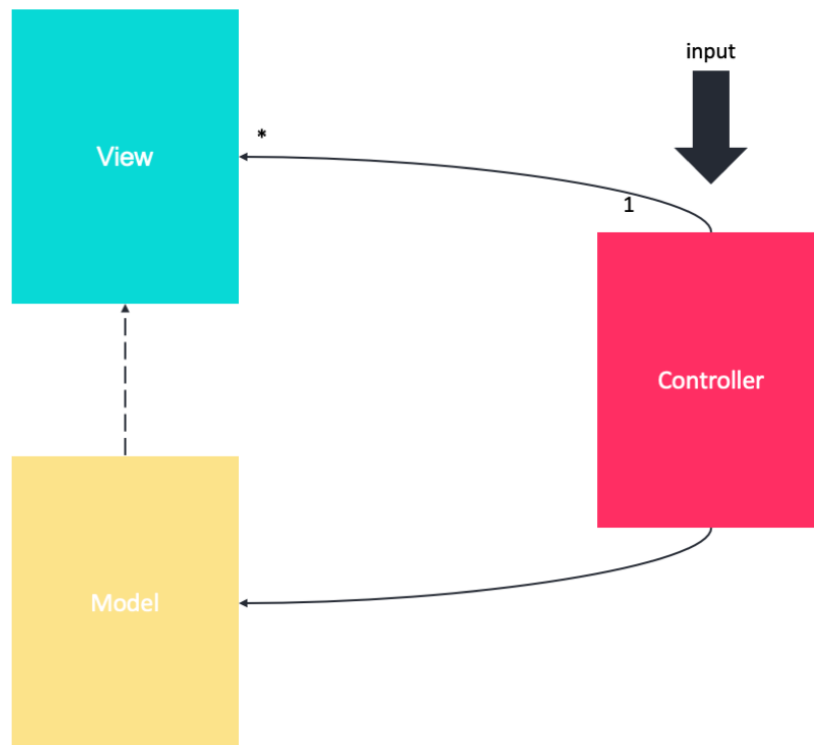


[Open in app](#)[Get started](#)

- Modularity
- Flexibility
- Testability
- Maintainability

## Model-View-Controller

Or MVC for short is a widely used design pattern for architecting software apps. The pattern was originally designed by Trygve Reenskaug during his time working on Smalltalk-80 (1979) where it was initially called Model-View-Controller-Editor. MVC went on to be described in depth in “Design Patterns: Elements of Reusable Object- Oriented Software” (The “GoF” book) in 1994, which played a role in popularizing its use. The pattern breaks an app up into three components.

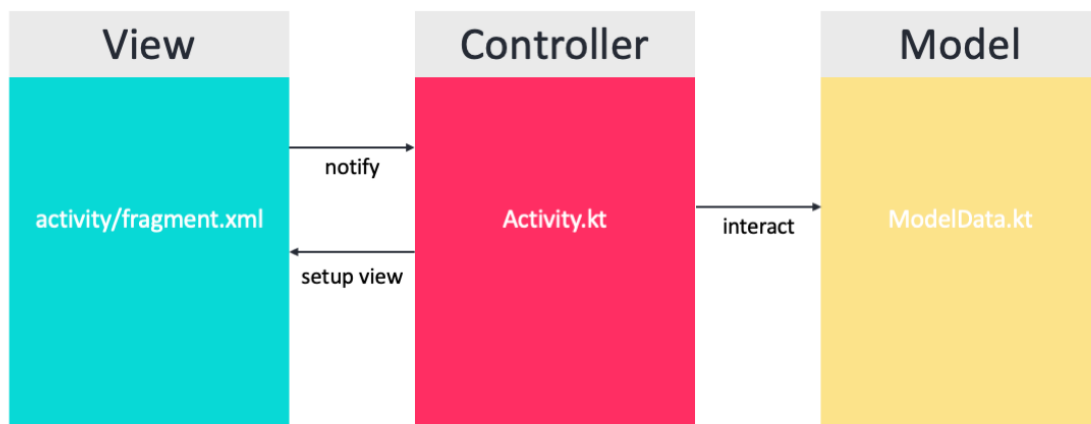


- **Model** — is responsible for the business logic of the application. It manages the state



[Open in app](#)[Get started](#)

- **View** — this component has two important tasks: presenting data to the user and handling user interaction.
- **Controller** — the view layer and the model layer are glued together by one or more controllers.

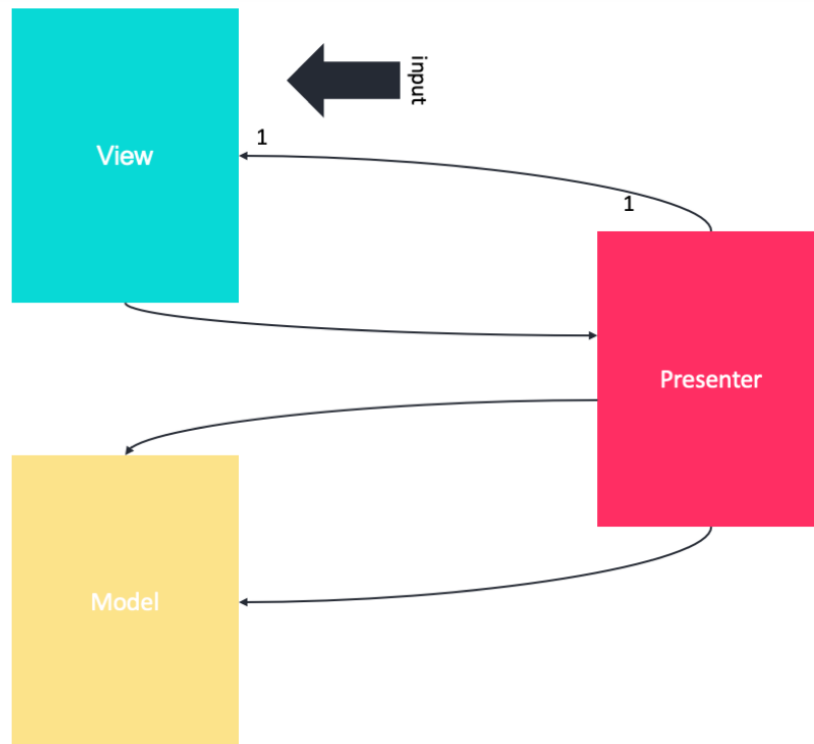


MVC on Android

## Model-View-Presenter

MVP is a derivative of the MVC design pattern which focuses on improving presentation logic. It originated at a company named Taligent in the early 1990s while they were working on a model for a C++ CommonPoint environment.

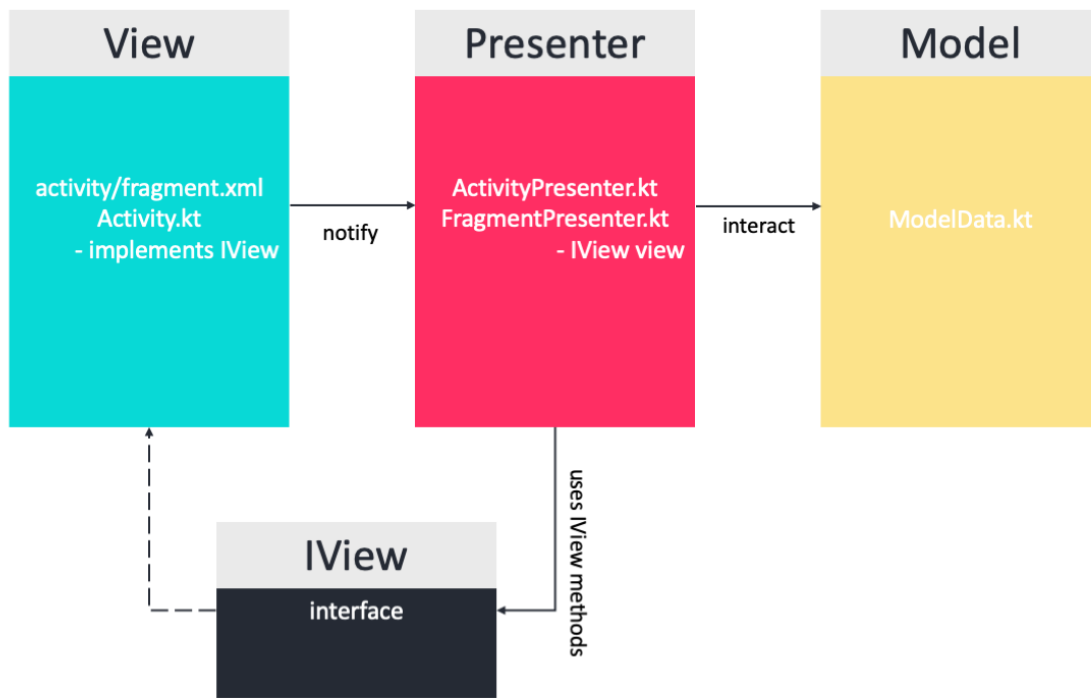


[Open in app](#)[Get started](#)

Although MVP is a derivation of MVC, they do have their slight differences.

- **Model** — The Model represents a set of classes that describes the business logic and data. It also defines business rules for data means how the data can be changed and manipulated.
- **View** — The View is used for making interactions with users like XML, Activity, fragments. It has got nothing to do with the logic that is to be implemented in the process.
- **Presenter** — The presenter gets the input from the View, processes the data with the help of the Model, and passes the results back to the View after the processing is done.



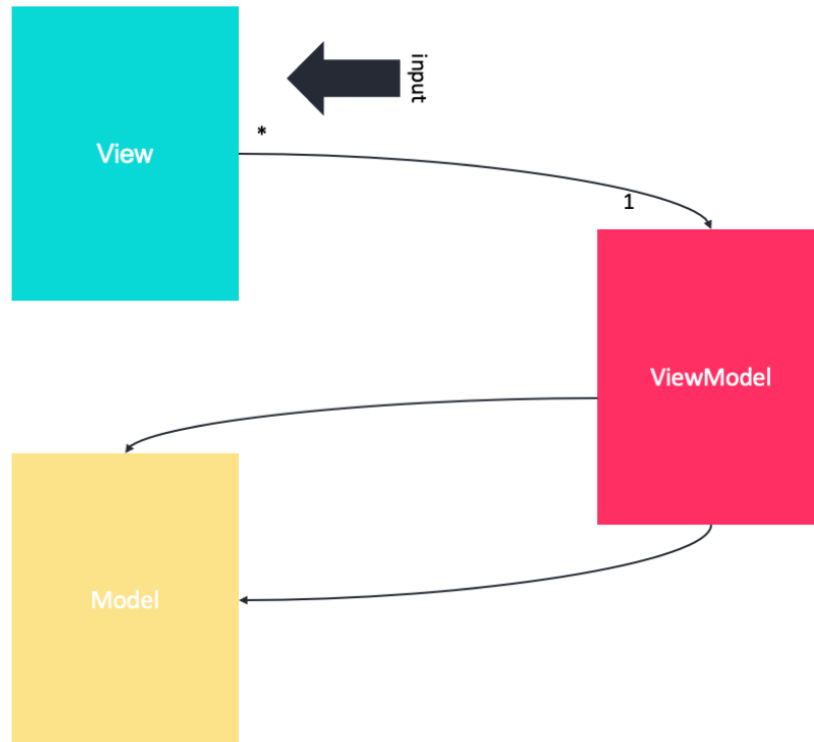
[Open in app](#)[Get started](#)

MVP on Android

## Model-View-ViewModel

MVVM was originally defined by Microsoft for use with Windows Presentation Foundation (WPF) and Silverlight, having been officially announced in 2005 by John Grossman in a blog post about Avalon (the codename for WPF). This pattern based on MVC and MVP which attempts to more clearly separate the development of UI from that of the business logic and behavior in an application.

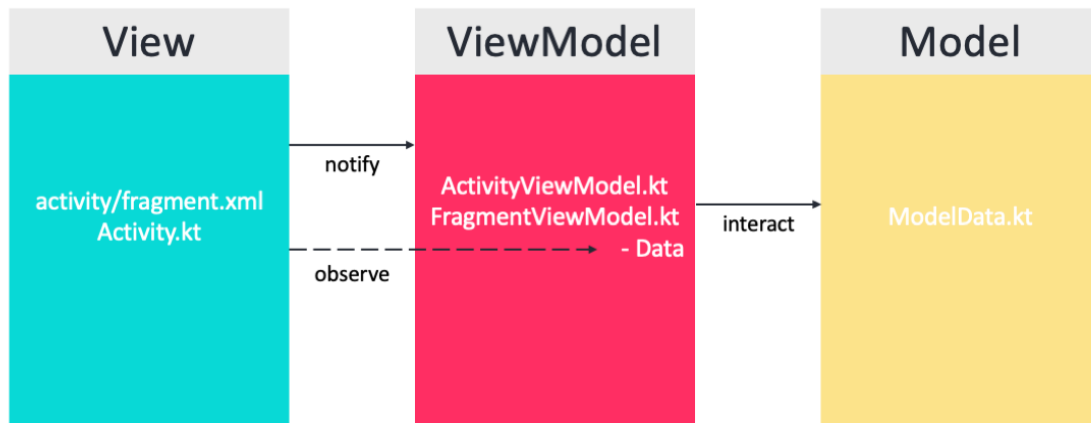


[Open in app](#)[Get started](#)

It has three main components as follows.

- **Model** — the Model used in MVVM is similar to the model used in MVC, consisting of the basic data required to run the software.
- **View** — the View is a graphical interface between the user and the design pattern, similar to the one in MVC. It displays the output of the data processed.
- **View-Model** — the View-Model is on one side an abstraction of the View and, on the other, provides a wrapper of the Model data to be linked. That is, it comprises a Model that is converted to a View, and it also contains commands that the View can use to influence the Model.

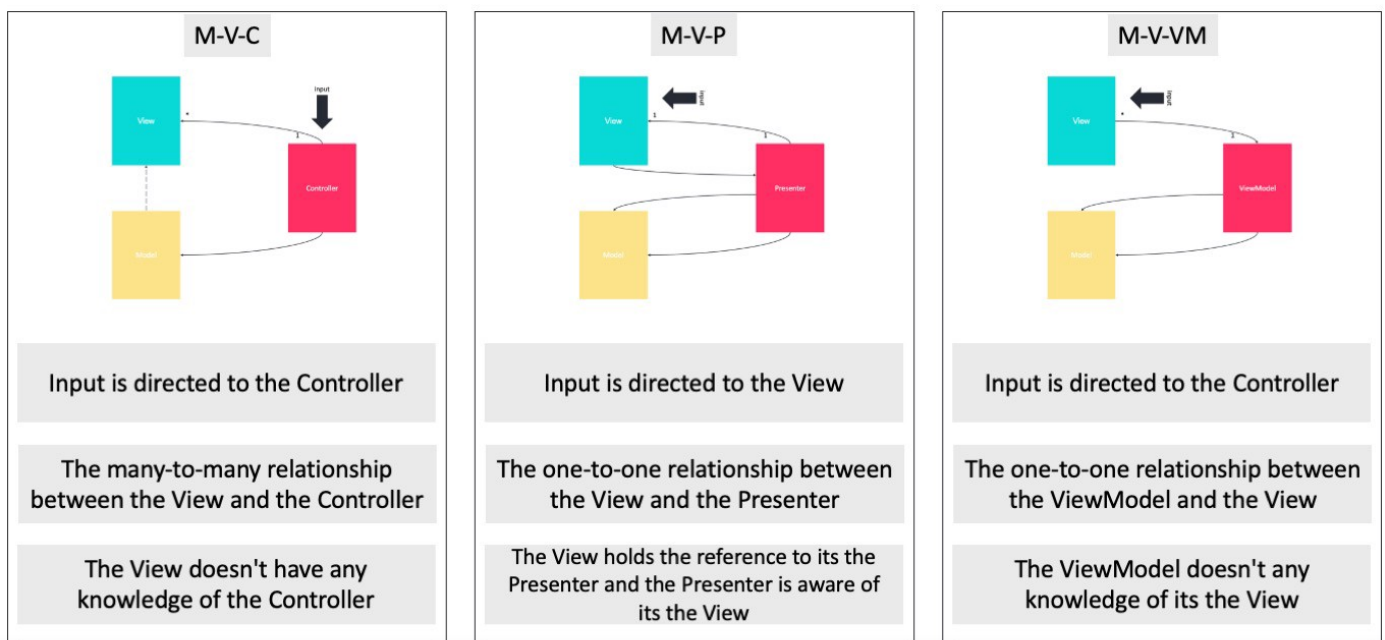



[Open in app](#)
[Get started](#)


## MVVM on Android

## MVC vs MVP vs MVVM

Let's find out MVC Vs MVP Vs MVVM, main points of differences.





[Open in app](#)[Get started](#)

**Performance Evaluation** — When we are testing the UI performance, MVP turns out to be the one with the highest reliability and lowest hindrance when it comes to rendering frames. Data binding in MVVM creates an additional overload that could drastically affect its performance when performing complex tasks.

**Compatibility** — When testing the patterns based on their compatibility, MVVM was the best of the lot due to its data binding which had a positive impact. MVP fared better than MVC, which had serious restating issues.

**Modifiability** — When we talk about design patterns, it is evident that it should be modifiable since that gives us the option of adding new features and strategies into our app.

- Based on these factors, we observed that changes are very less in MVP and MVVM, with MVVM contributing a lot towards maintainability.
- MVC tends to increase the number of changes in the majority of the cases.

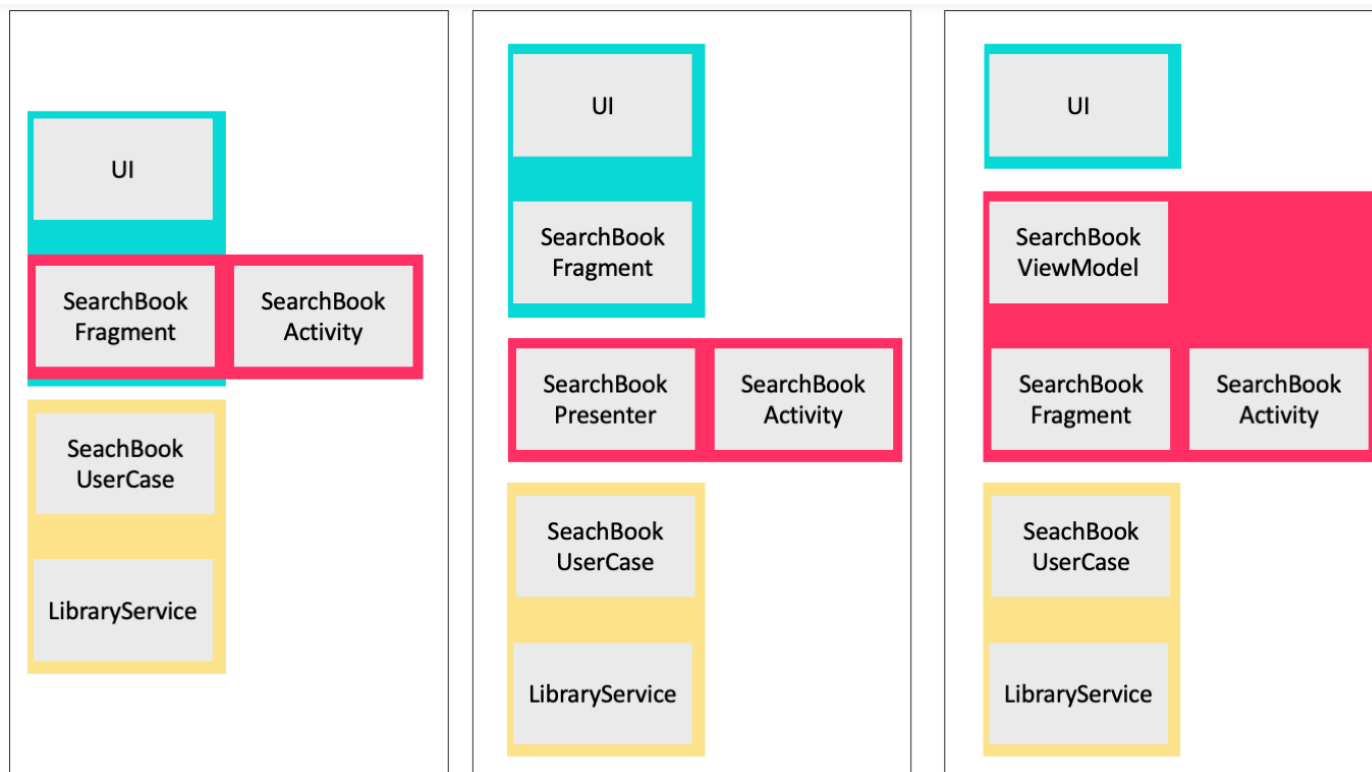
**References** — In MVC, the View doesn't have reference to the Controller while in MVP, the View has reference to the presenter and in MVVM, the View has reference to the View-Model.

**Entry Point** — For MVC, the entry point to the application is the Controller whereas, for MVP and MVVM, the entry point is the View.

## Implement the example by various architectures

You can implement the BookSearch application to understand by using one of three architecture as the following diagram.



[Open in app](#)[Get started](#)

MVC vs MVP vs MVVM implementation for the BookSearch application

Easy, right? Can you implement it yourself?

## Sign up for Top Stories

By Level Up Coding

A monthly summary of the best stories shared in Level Up Coding [Take a look.](#)

[Get this newsletter](#)





[Open in app](#)

[Get started](#)

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

