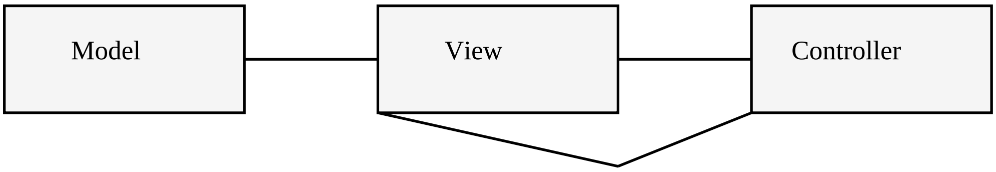
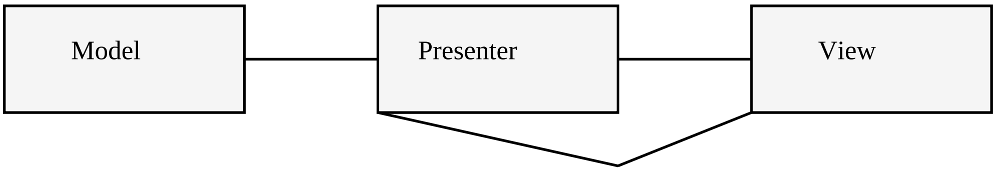
MVC (Model–View–Controller) Design Pattern – One Page Summary

Overview: Model–View–Controller (MVC) is a design pattern that separates an application into three components to improve modularity, maintainability, and scalability. Each component has a distinct responsibility and communicates through well-defined interfaces.

Core Components:
Model: Handles data and business logic.
View: Displays information to the user.
Controller: Processes user input and coordinates between Model and View.

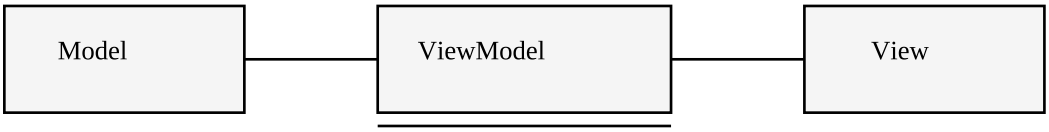


Variant 1: Model–View–Presenter (MVP)
In MVP, the Presenter contains the presentation logic and directly updates the View. The View is passive and forwards user input to the Presenter.



Appropriate Usage: MVP is suitable for desktop and enterprise applications where unit testing and strict separation between user interface and business logic are required.

Variant 2: Model–View–ViewModel (MVVM)
MVVM introduces a ViewModel that enables automatic synchronization between the View and Model through data binding mechanisms.



Appropriate Usage: MVVM is ideal for modern web and mobile applications that support data binding frameworks, such as Angular, React, or Android architectures.