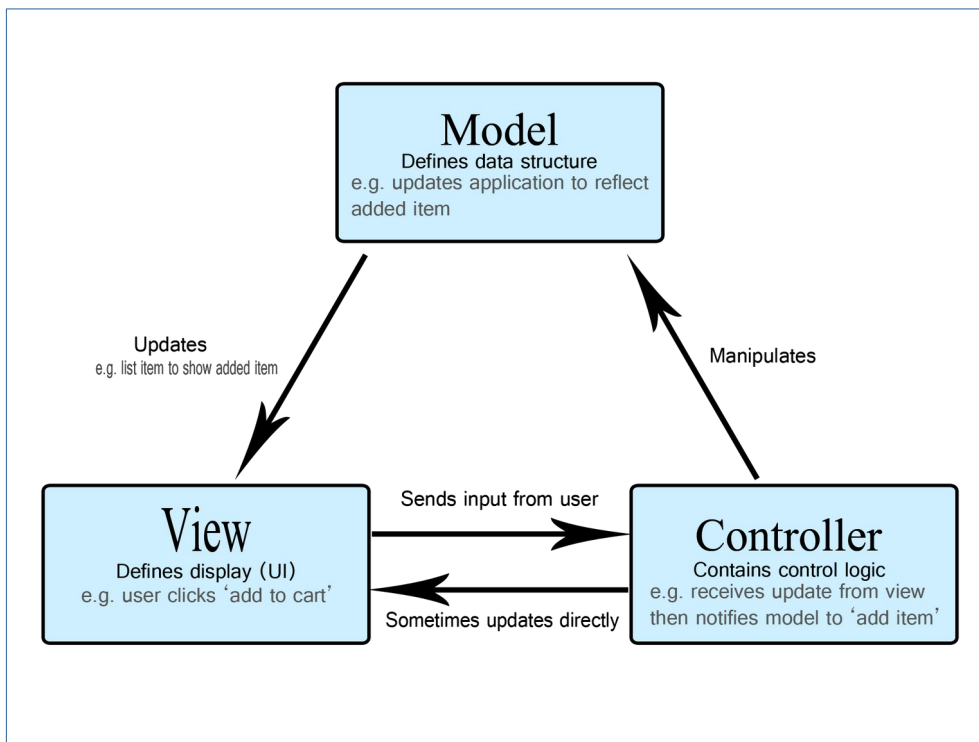


## „MVC-Designmuster“

**MVC** : steht für **Model-View-Controller**. Es ist ein Architektur- oder Designmuster, das verwendet wird, um Softwareanwendungen **in drei verbundene Komponenten** zu trennen:

- **Model (Modell) oder (domain)**: Dies ist der Teil, der die Daten und die Geschäftslogik enthält. Es kümmert sich um das Speichern, Abrufen und Verwalten von Daten.
- **View (Ansicht)**: Dies ist der Teil, der die Benutzeroberfläche darstellt, also das, was der Benutzer sieht und mit dem er interagiert.
- **Controller**: Dies ist der Teil, der die Eingaben des Benutzers verarbeitet, das Model aktualisiert und die View entsprechend anpasst.



1. **View (Ansicht)**: Die View zeigt dem Benutzer die Daten an und ermöglicht ihm, Aktionen durchzuführen (z. B. durch Klicken auf einen Button).
2. **Benutzer führt eine Aktion in der View aus**: Diese Aktion (wie z. B. ein Button-Klick) wird an den **Controller** gesendet.
3. **Controller verarbeitet die Aktion**: Der Controller empfängt die Aktion und führt die entsprechende Logik aus. Dies kann eine Aktualisierung der Daten im **Model** bedeuten.
4. **Model wird aktualisiert**: Der Controller ändert die Daten im Model basierend auf der Benutzeraktion.
5. **Model teilt der View nichts direkt mit**: Es gibt kein direktes Signal vom Model an die View. Stattdessen:
6. **Controller aktualisiert die View**: Nachdem der Controller das Model aktualisiert hat, sagt er der View, dass sie sich aktualisieren soll, basierend auf den neuen Daten im Model.

Also, die richtige Reihenfolge lautet:

1. **View:** Benutzer führt eine Aktion durch.
2. **Controller:** Die Aktion wird an den Controller gesendet.
3. **Controller:** Aktualisiert das Model.
4. **Controller:** Fragt das Model nach den neuen Daten.
5. **Controller:** Aktualisiert die View mit diesen neuen Daten.