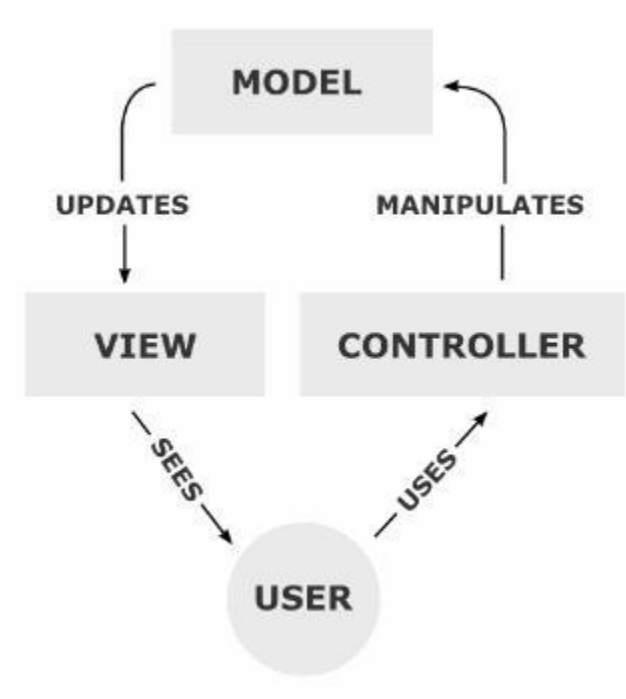
MVC Architecture

Model view controller (**MVC**) is a software architecture pattern which separates the representation of information from the user's interaction with it.



SFDC MVC:

- We can write our VIEW pages using SFDC visual force (VF pages).
- Each VF page is associated with a Controller.
- We can make use to already built Standard controllers or we can write our own controller using Apex language. Apex is OO and very much similar to JAVA.
- We can also write Model Classes using Apex.

SFDC MVC pattern contains below three modules:

Model (Database Layer): The schema and data salesforce uses to represent the system completely. In salesforce, we can say that **sObjects** are the model as every entity in salesforce is mapped to some sObject. (sObjects are any objects that can be stored in the Force.com platform database)

View (User Interface Layer): How the schema and data is represented. **Visualforce** is used to present the data to users.

Controller (Business Logic Layer): How the interface actions. **Controllers** are used to perform the actions whenever users interact with visual force.

In SFDC:

1. Visual Force pages, Page Layouts, Tabs comes under View Layer of Model View controller.
2. Workflows, Apex Classes, Triggers comes under Controller part in Model View controller.
3. Objects, Fields, Relationships comes under Model Layer of Model View Controller.

Benefits of using MVC architecture:

1. Reliability:

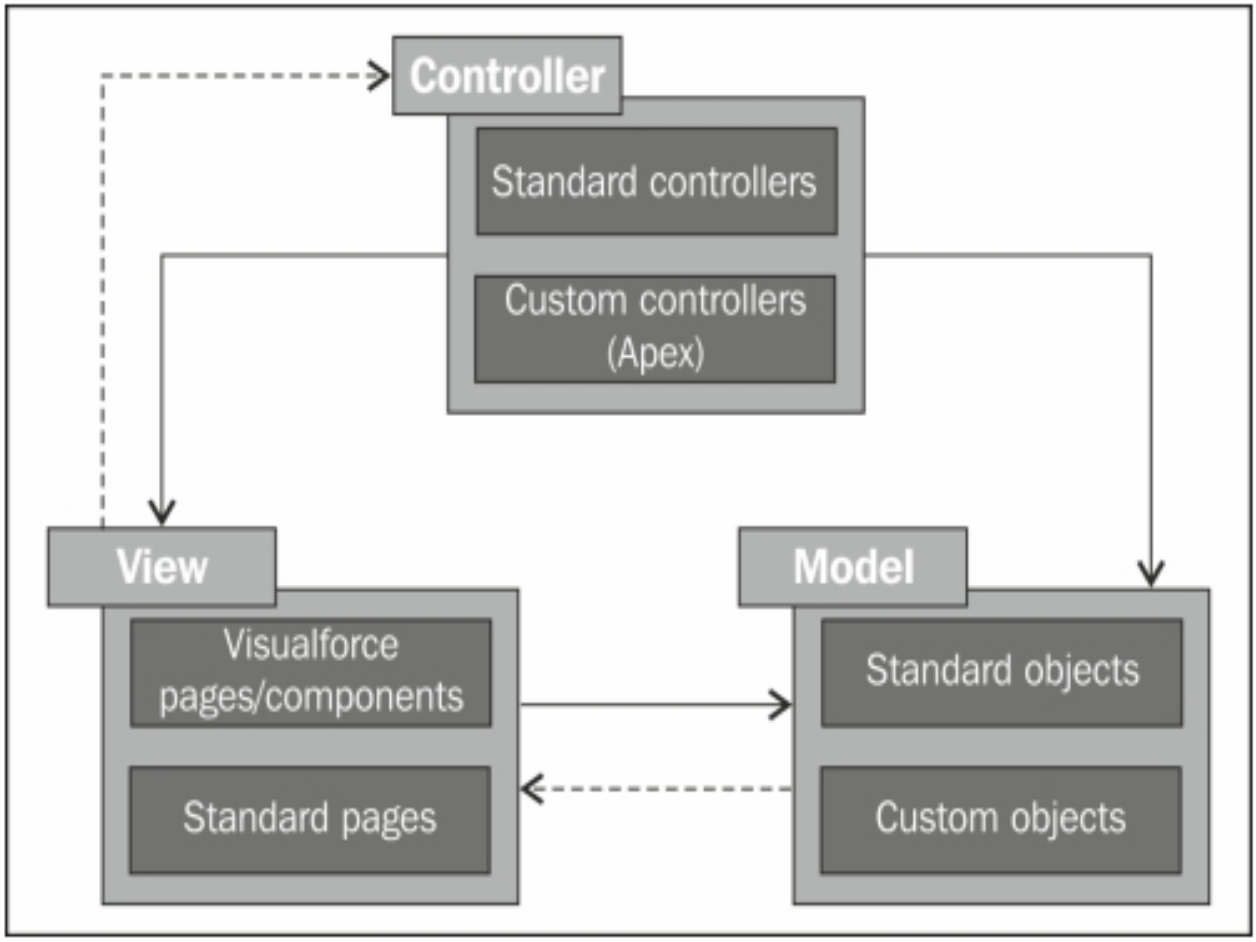
   The layers have clear separation so it allows greater reliability if we want to change any layer. For example, if can change the look and feel of an application without recompiling Model or Controller code.

2. High Reuse and adaptability:
   MVC provides multiple types of views all accessing the same code.
3. Very low development and life cycle cost:

   It makes it possible to have lower-level programmers develop and maintain the user interfaces. Development time can be significantly reduced because Controller programmers focus solely on transactions and view programmers focus solely on presentation.

4. Easy to maintain:

   The separation between layers makes it easier to maintain, test and deploy apps.



Implementation of MVC Architecture in Salesforce