



# **Model – View – Controller**

# Model – View – Controller (MVC)

- **A Software Architectural Pattern**
  - Much like a design pattern
  - It uses design patterns
- **Provides low coupling in User Interface systems.**

# MVC Key Idea

## Break application into 3 logical components

- **Model:**

- Holds the state of the application and logic/rules of how state can be updated.

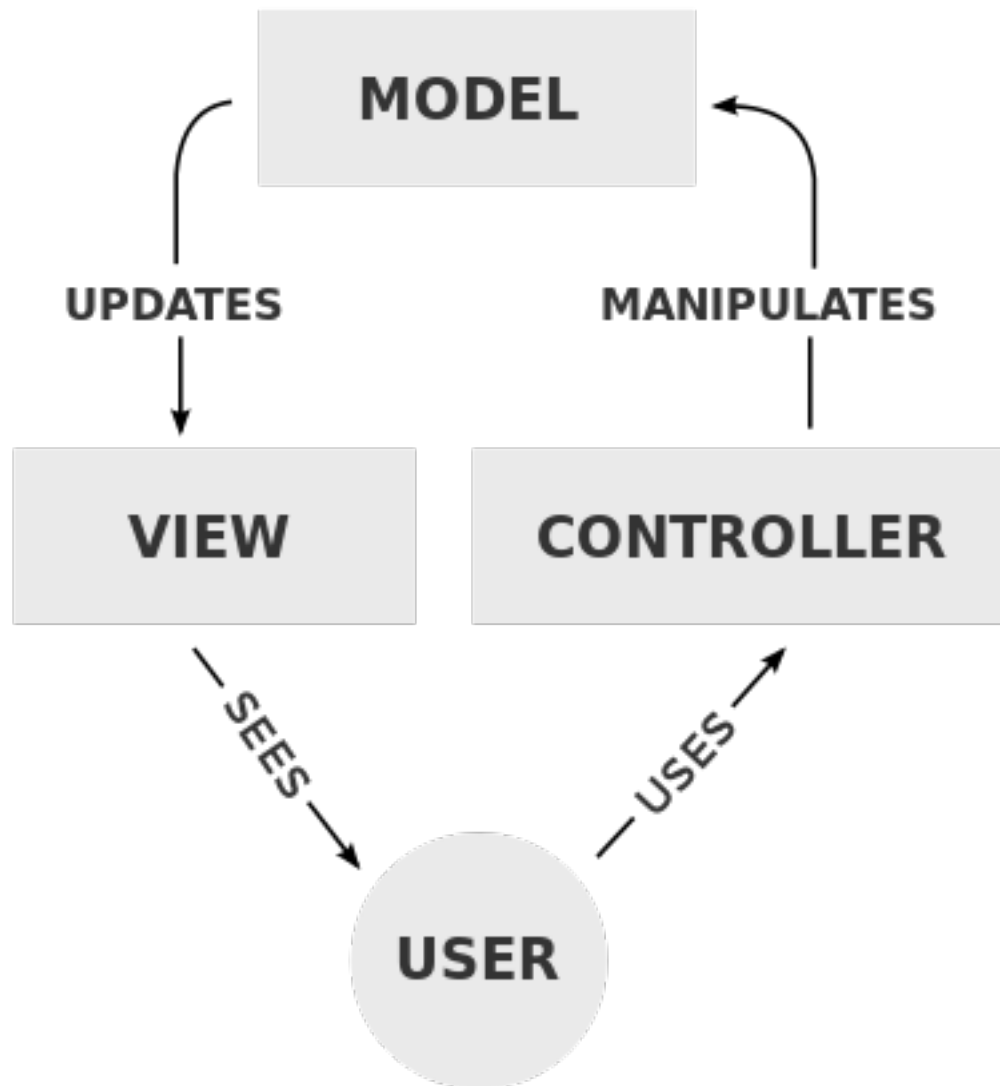
- **View:**

- Output representation of information

- **Controller:**

- Accepts user input and translates it into commands for the model or a view

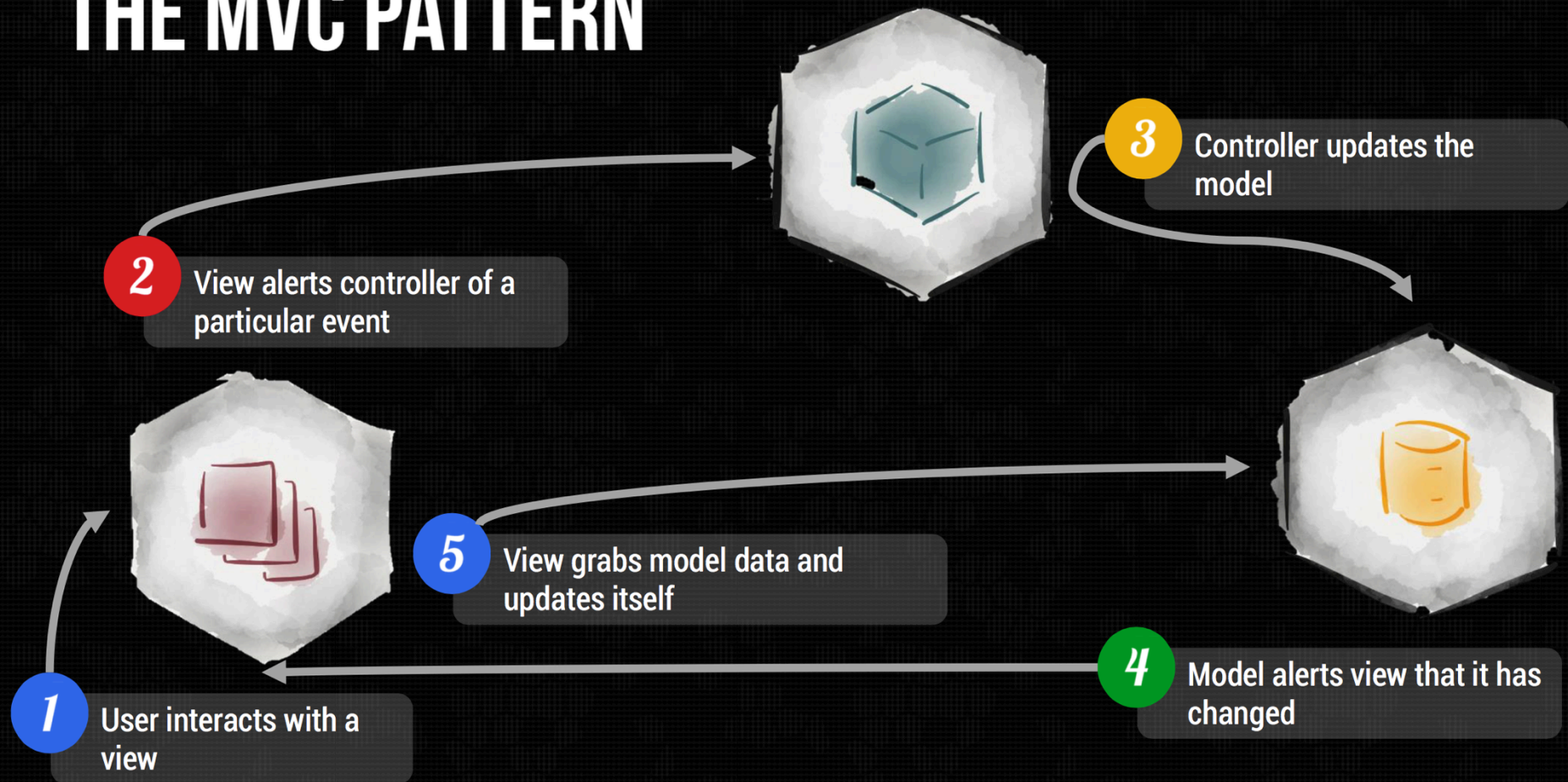
***Decouple interface from application state***



# Low Coupling from Views/Controllers to Model

- **Model code is independent from interface**
- **Can support many different views**
  - Simultaneously or one-at-a-time (e.g., summary vs. detail)
  - Use Observer pattern to attach Views to Model
- **Can support many different controllers**

# THE MVC PATTERN



# CS 126 Term Project

- **Imagine, Design, and Create your own android app.**
- **Three Requirements:**
  - Have at least two activities (in the Android sense)
    - Thursday's lecture
  - Have interactivity between App users across devices
    - Next week
  - Use one Android platform feature not explicitly taught
    - E.g., hardware features (e.g., GPS, accelerometer, gyroscope, camera, microphone, speaker), authentication (e.g., Firebase), notifications (e.g., Firebase), activities with multiple fragments, custom Views using `android.graphics.canvas`,