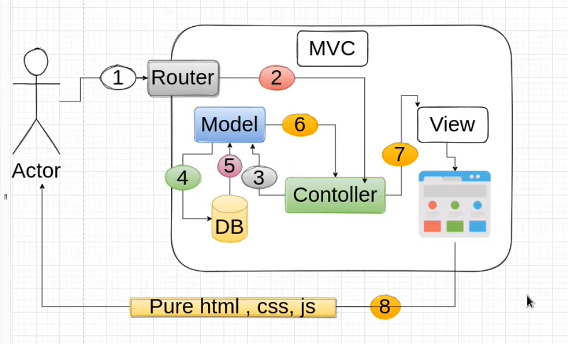
# Django Architectural Pattern:

**MVC (Model-View-Controller)** and **MVT (Model-View-Template)** are both architectural patterns used for organizing the components of an application. The two are conceptually similar but differ in how they handle certain aspects of application design.

**1. MVC (Model-View-Controller):**

This pattern is widely used in many web and desktop applications.



**Components:**

* **Model**: Manages the data, logic, and rules of the application. It directly interacts with the database.
* **View**: Displays the data to the user. It contains the UI/UX logic (HTML, CSS, etc.).
* **Controller**: Acts as an intermediary between Model and View. It processes user input, updates the model, and refreshes the view.

**Workflow:**

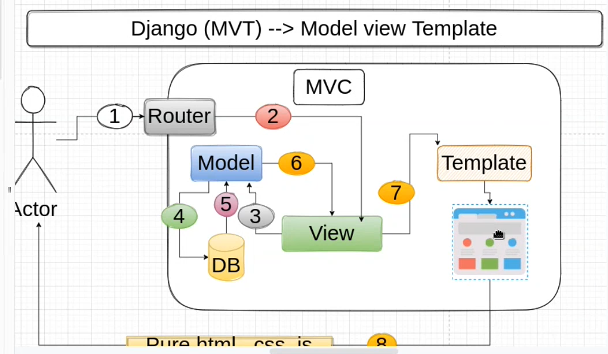
1. User interacts with the **View** (e.g., clicks a button).
2. The **Controller** handles the input and decides what to do.
3. The **Controller** fetches/updates data in the **Model**.
4. The **Model** sends updated data back to the **View** for rendering.

**Example Frameworks:**

* Ruby on Rails
* ASP.NET MVC
* AngularJS

**2. MVT (Model-View-Template):**

This is the architecture followed by Django, and it's a slight variation of MVC.



**Components:**

* **Model**: Same as MVC. It manages data and database interactions.
* **View**: Handles business logic and acts as a bridge between the Model and the Template.
* **Template**: Handles the presentation layer (HTML + Django Template Language). It renders dynamic content using data passed by the View.

**Workflow:**

1. User makes a request (e.g., navigates to a URL).
2. The **View** processes the request, fetches data from the **Model**, and passes it to the **Template**.
3. The **Template** formats the data for the user and returns the response.

**Key Difference from MVC:**

* In MVT, the **Controller** part is implicitly handled by the Django framework (via URL routing and middleware).
* The **View** in MVT does more than just display data—it contains business logic and passes data to the **Template**.

**Example Frameworks:**

* Django (Python)
* Flask (Python, although loosely based)