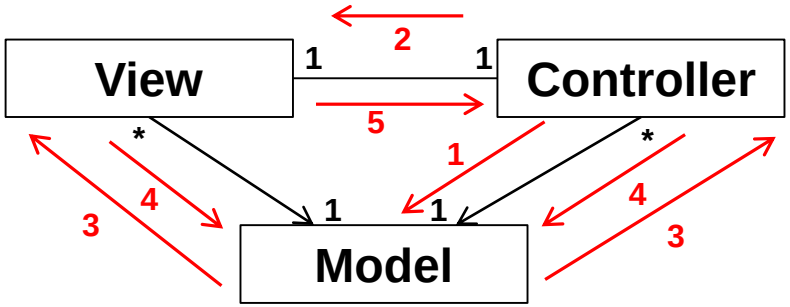
Model-View-Controller (MVC)

- **Compound Pattern**
  - A set of patterns working together to solve a recurring or general problem

- **Model-View-Controller**
  - Software architecture pattern which separates the representation of information from the user's interaction with it
    - Model: representation of application data & business logic
    - View: presentation of the data (multiple views are possible) and the source of user interaction
    - Controller: controls and interprets input, forwards it to the model

Model-View-Controller (MVC)



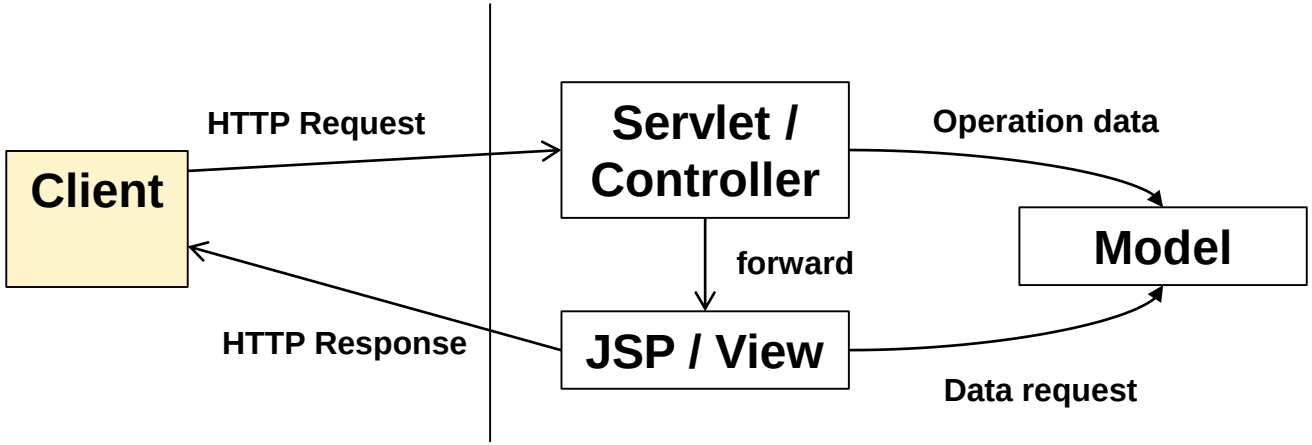
- **Interactions**
  1. Controller reacts on user input and initiates state change on the model
  2. Controller can update view directly (e.g. scrolling)
  3. Notification of state changes (may affect look (view) and feel (controller))
  4. Access to the model data
  5. View forwards interactions to the controller

Model-View-Controller (MVC)

- **Patterns**
  - Observer
    - Model = Observable
      View / Controller = Observers
    - Information of the views and the controller about state changes
  - Strategy
    - Controller represents the behavior of the view.
    - The behavior can be exchanged by replacing the strategy

Model-View-Controller (MVC)

- **MVC2**
  - MVC adapted for use with the internet



    - View is not registered as an observer in the model, it is informed about state changes from the controller