Esercitazioni Prova Finale 2019

Part 4 - MVC

Docente Gianpaolo Cugola

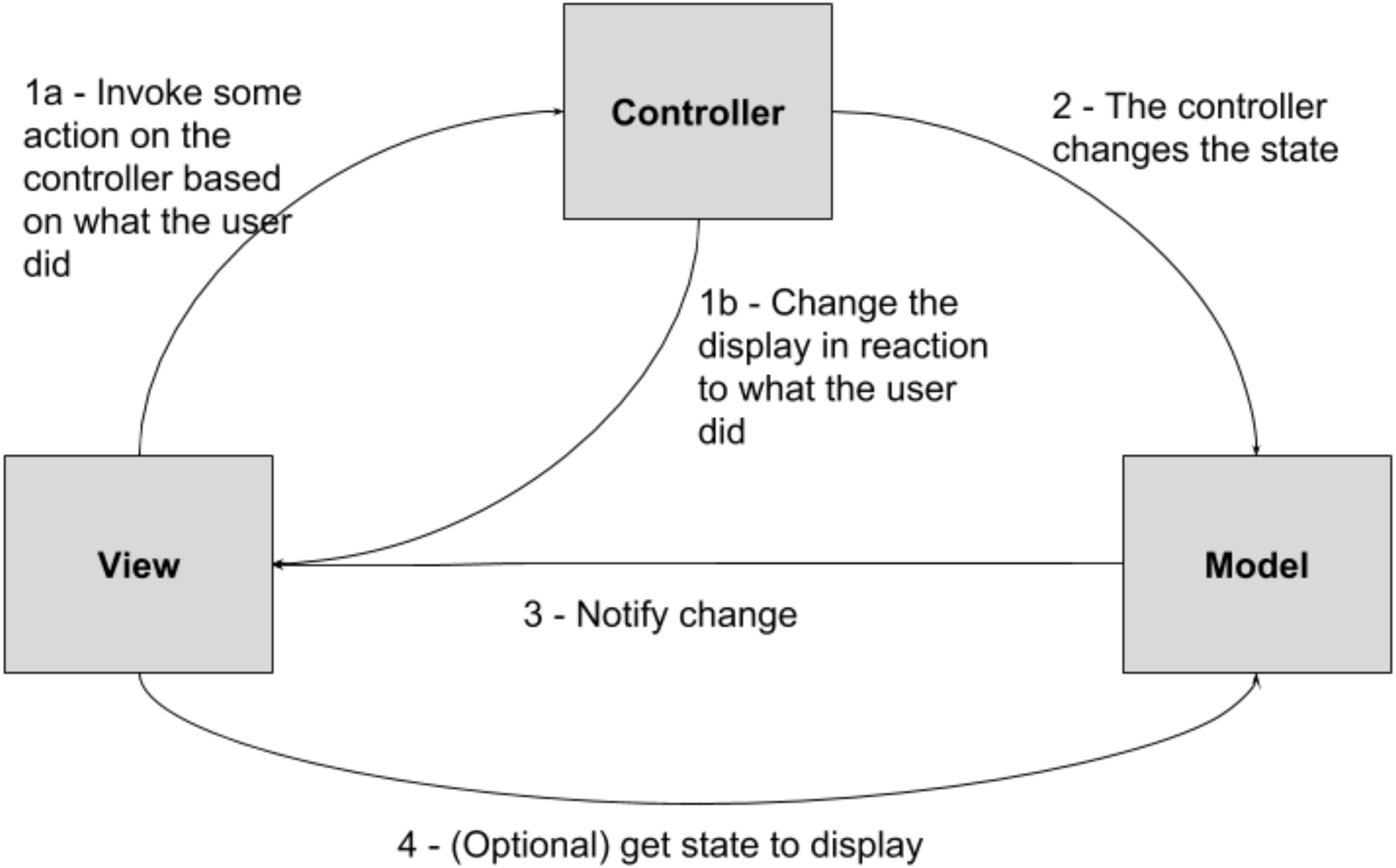
Esercitatore Mario Scrocca

Code: https://github.com/marioscrock/ingsoft-prova-finale-19

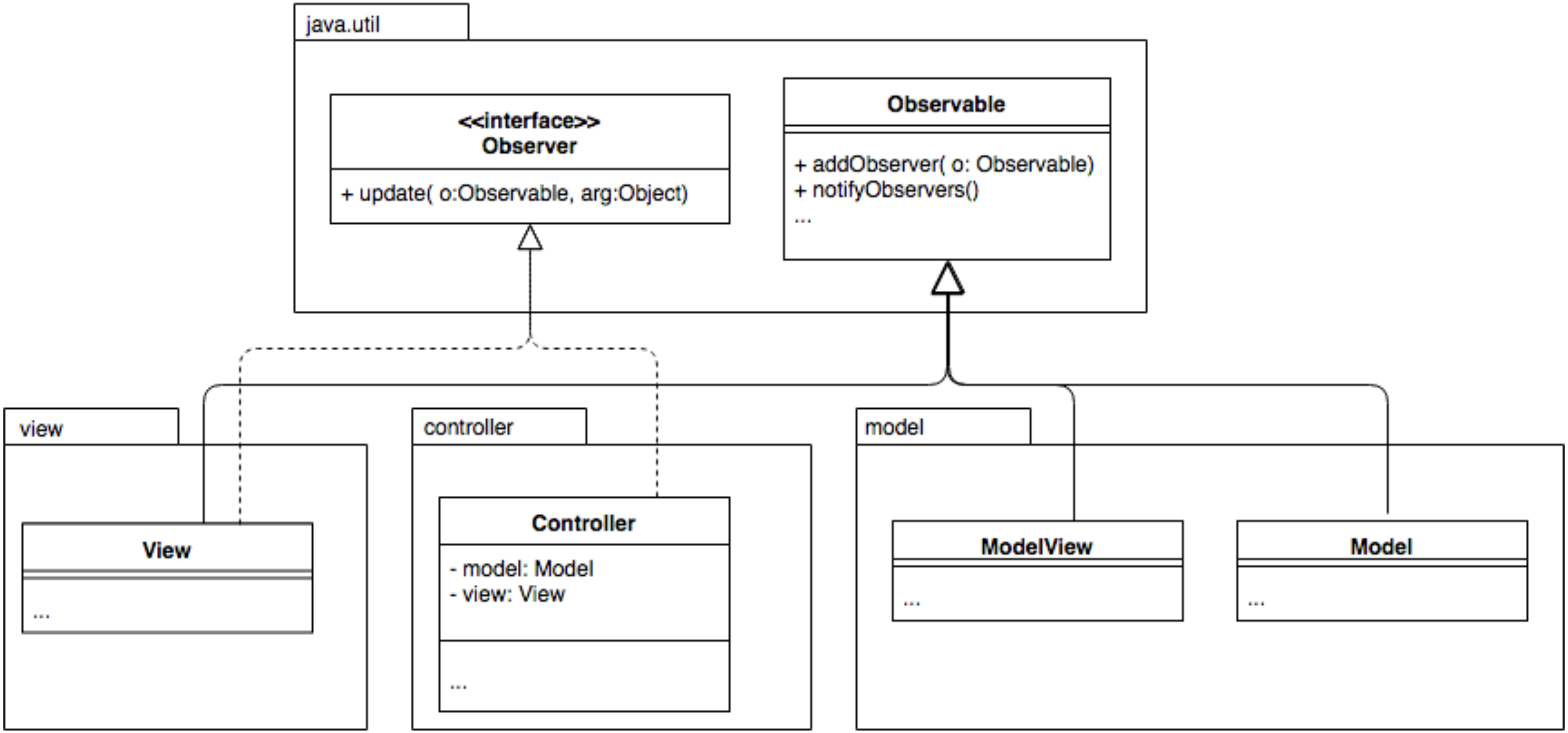
MVC

The MVC pattern identifies three entities:

- **The model**: hold all the state and application logic
- **The view**: gives a representation of the model
- **The controller**: maps the user input from the view to state changes in the model


Controller
1a - Invoke some action on the controller based on what the user did
2 - The controller changes the state
1b - Change the display in reaction to what the user did
View
Model
3 - Notify change
4 - (Optional) get state to display

Observer & Observable



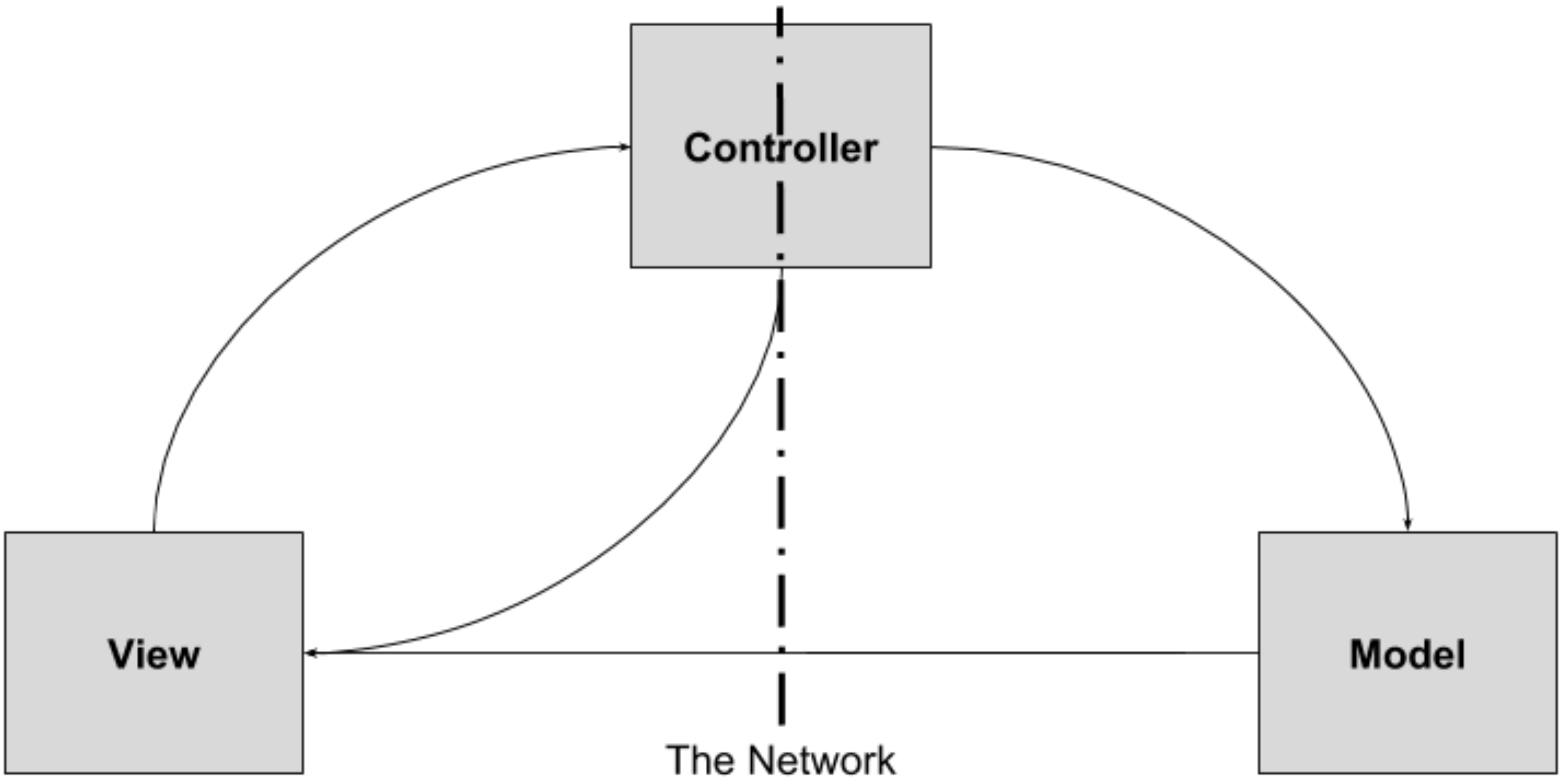
Observer & Observable

- Programming based on events
- Observable object generated events
- Observer object is notified on events
- Can be implemented with Generics to avoid duplication of code

Distributed MVC

- When the MVC is distributed, one must decide **where to split and introduce a networking layer** to make the method calls become remote.
- It seems natural to split the Controller into a client-side and a server-side part.
- By splitting, we also intercept the **edge that connects the Model and the View**.
- **Do NOT confuse MVC and NETWORK**: network management should be transparent and should abstract from the specific technology (e.g. RMI e Socket)

Distributed MVC



Distributed MVC

- Transfer only objects representing events or created specifically
- Modifiable objects of the model shouldn’t be sent to the view
- **Solutions**:
  - Limited interfaces (methods to visualize only selected information)
  - Immutable objects or cloneable()
  - Objects created specifically for the visualization