

# MVC

WEB TABANLI WINDOWS PROGRAMLAMA DERSİ



# WHAT IS MVC ?

- **MODEL**

- **REPRESENTS THE LOGIC OF THE APPLICATION**
- **THE APPLICATION OBJECT**

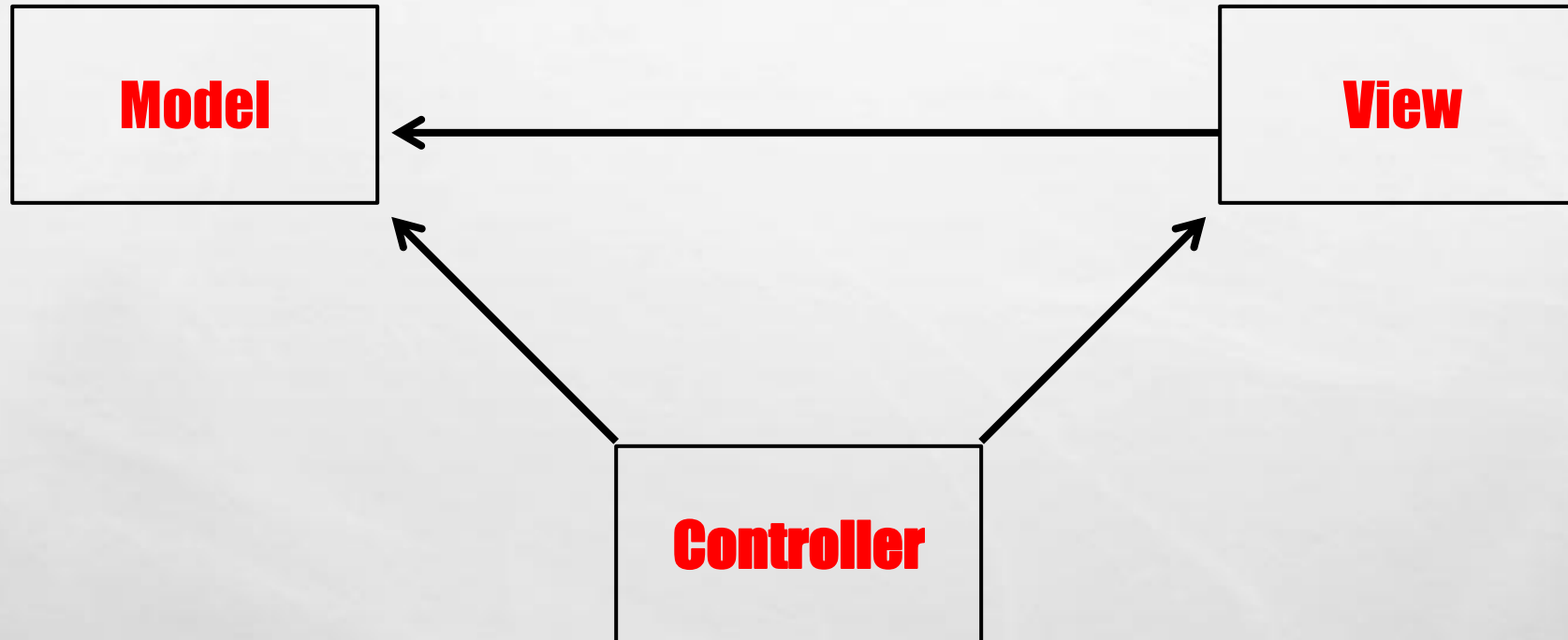
- **VIEW**

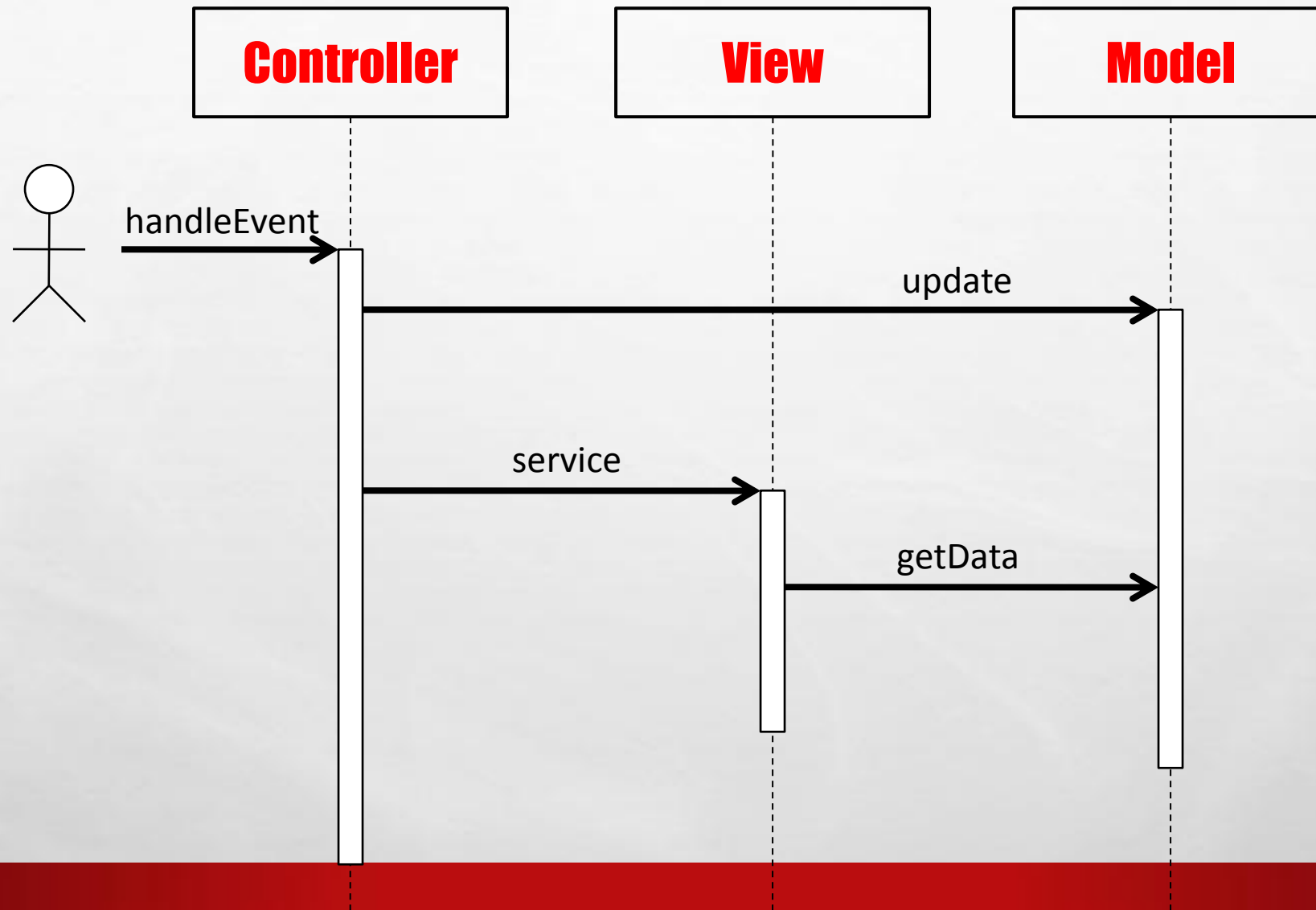
- **THE VISUAL REPRESENTATION OF THE MODEL**
- **THE SCREEN PRESENTATION**

- **CONTROLLER**

- **RESPONSIBLE FOR HANDLING ALL USER INPUT**
- **DEFINES THE WAY THE USER INTERFACE REACTS TO USER INPUT**

# MVC





# BEFORE MVC ?

- **BEFORE MVC, USER INTERFACE DESIGNS TENDED TO LUMP THESE OBJECTS TOGETHER**
- **MVC DECOUPLES THEM TO INCREASE FLEXIBILITY AND REUSE**

# SEPARATION OF CONCERNS (SOC)

- **EACH COMPONENT HAS ONE RESPONSIBILITY**
  - **SRP : SINGLE RESPONSIBILITY PRINCIPLE**
  - **DRY : DON'T REPEAT YOURSELF**
- **MORE EASILY TESTABLE**
- **HELPS WITH CONCURRENT DEVELOPMENT**

# ASP.NET MVC

- **NOT** THE WEB FORMS 4.0
- **NOT** REPLACING WEB FORMS, BUT ADDS TO IT
- IT CAN **NOT** USE WEB CONTROLS
- **NOT** A WHOLE NEW ENGINE BUT SITS ON ASP.NET ENGINE

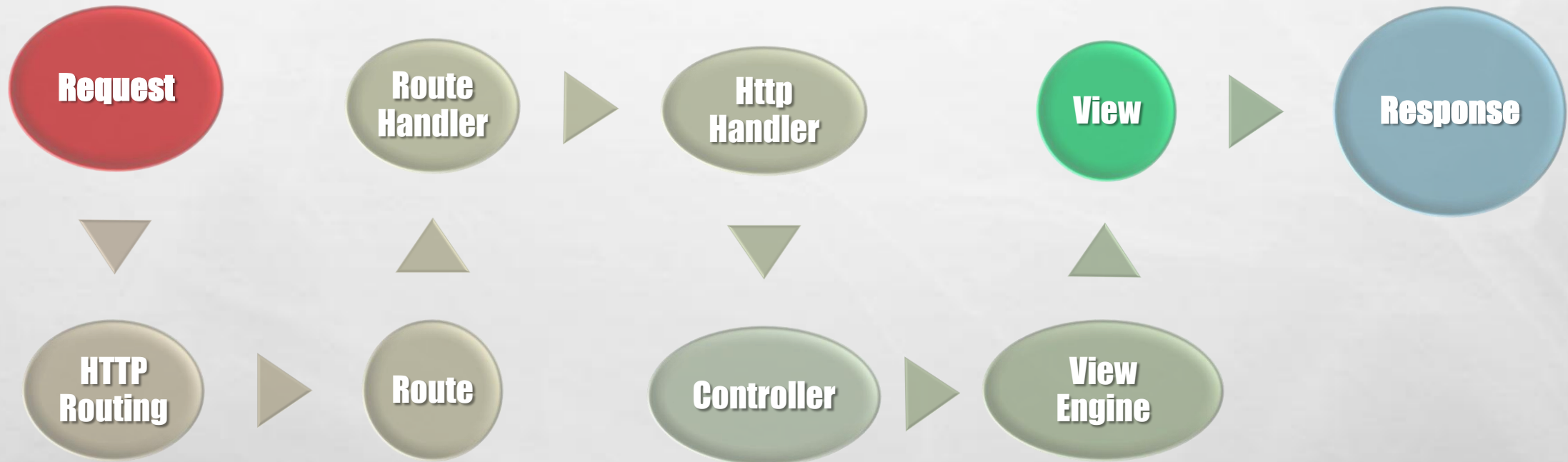


# WEBFORMS VS MVC

MVC	Web Forms
You want full control over markup	You like programming against the reusable control abstraction that encapsulate UI and logic
You want a framework that <i>enforces</i> separation of concerns	You like using the WYSWIG designer and would rather avoid angle brackets
TDD/Unit Testing is a priority for you	You like keeping logic on the server rather than hand writing Javascript
Control abstractions get in your way more than they help	Unit testing with the MVP pattern is sufficient for your needs
You like writing Javascript	



# ASP.NET MVC REQUEST FLOW



# DEMO