

# Modello MVC

Progettazione del Software II – A.A. 2014/2015

Sapienza Università di Roma

Dipartimento di Ingegneria Informatica, Automatica e Sistemistica

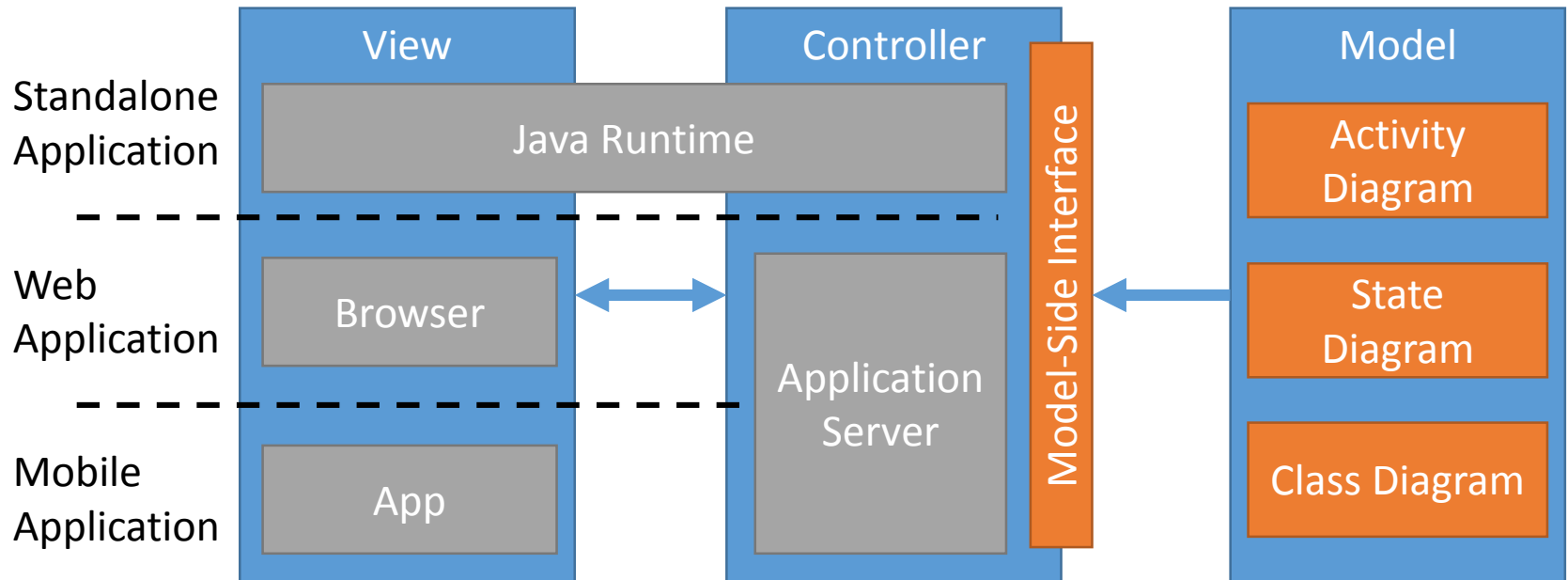
Francesco Leotta

08/05/2015

# Il modello MVC

- Basato sulla separazione dei componenti software di un applicativo
  - il **model** fornisce i metodi per accedere ai dati utili all'applicazione;
  - il **view** visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
  - il **controller** riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.
- L'interazione tra i tre componenti può seguire diverse modalità in pratica
- In questo corso utilizzeremo una modalità strict in cui il model richiama le funzionalità del controller e quest'ultimo richiama decide cosa visualizzare nella view
  - Tutto ciò che avete studiato nel corso costituisce il modello e rappresenta la «vera» applicazione
  - La view può essere a seconda delle esigenze
    - Una console
    - Una GUI
    - Una Web Application
    - ...
  - Il controller è dipendente dalla coppia <model, view> e si occupa della traduzione dell'interazione tra model e view
  - Nessuna interazione diretta tra il model e la view

# Il modello MVC che seguiremo



- L'esercitazione di oggi mostrerà lo sviluppo di una applicazione Java Standalone
- Ci occuperemo dell'implementazione del Class Diagram, dell'Activity Diagram e della Model-Side Interface
- In particolare la model-side interface verrà realizzata in Java ma altre soluzioni sono possibili
  - Il model definisce una interfaccia che il controller implementa
  - Il controller trasforma le richieste del model in operazioni interpretabili dall'interfaccia utente
  - In principio ogni componente può essere distribuito su una macchina apposita

# Suggerimenti per l'Esercitazione

- Suddividere il codice in 3 package relativi ai tre componenti
  - Il codice per il package 'view' è fornito come input
- L'interfaccia model side è già fornita
  - Analizzare la differenza tra le operazioni disponibili per la view e quelle definite all'interno dell'interfaccia per comprendere il tipo di trasformazione che il controller deve effettuare
- Utilizzare la classe Main fornita con l'esercitazione