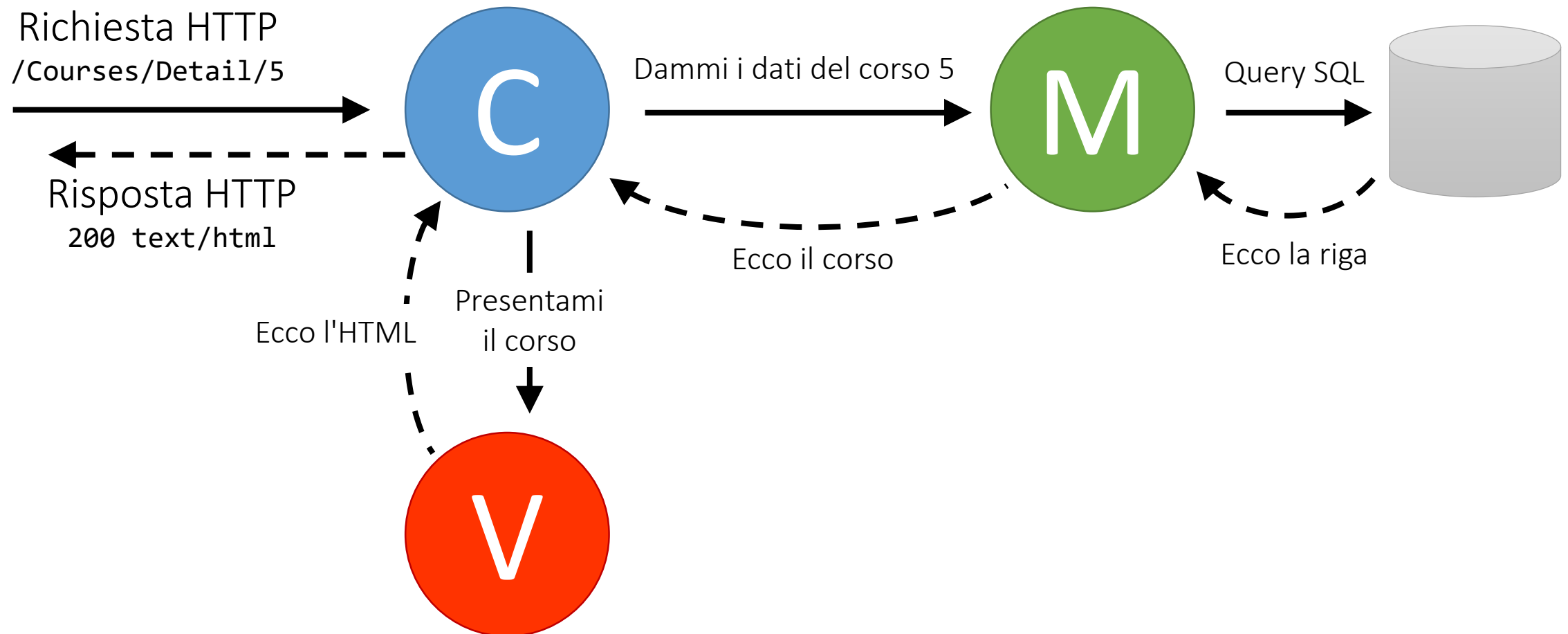


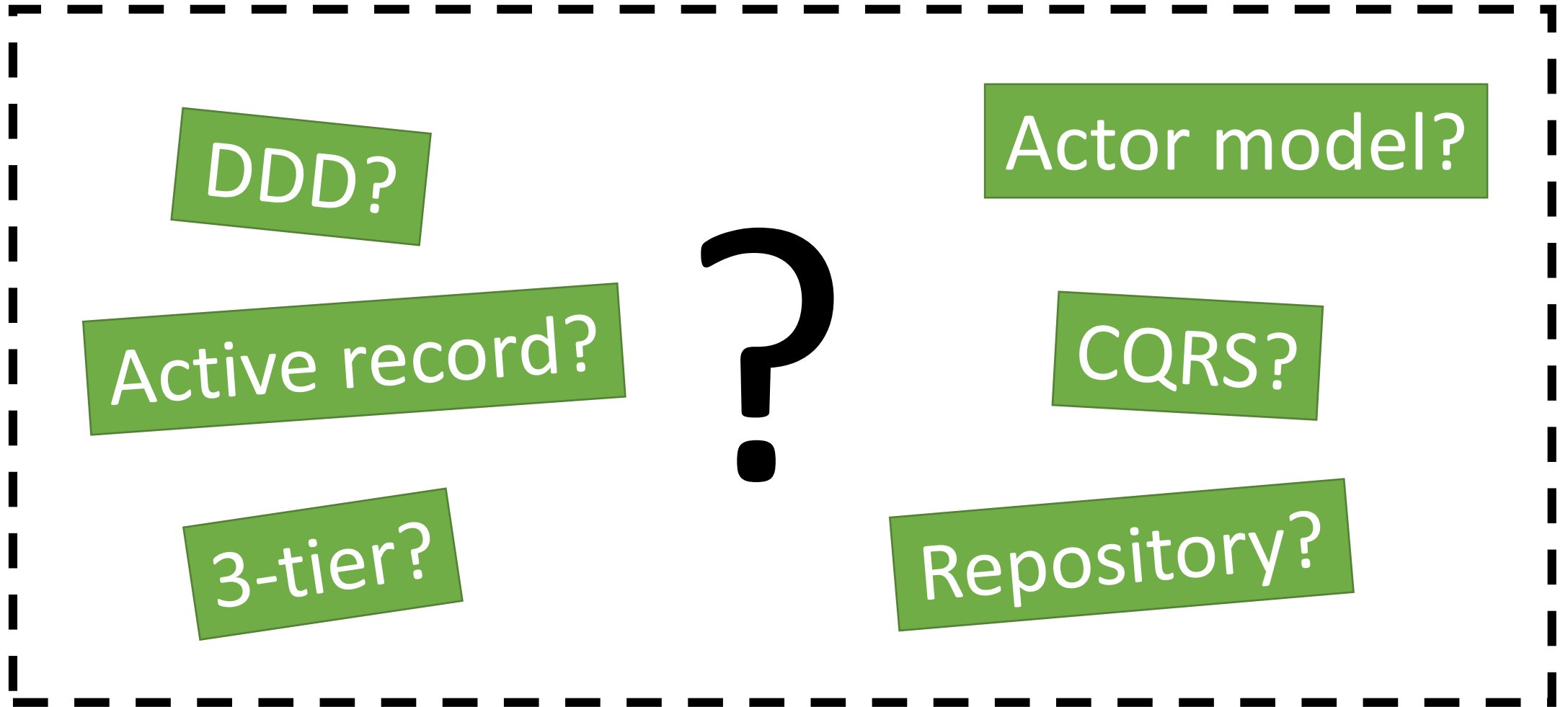
Sezione 08

ASP.NET Core MVC: il Model

# Il pattern MVC



# Quale approccio per strutturare il model?

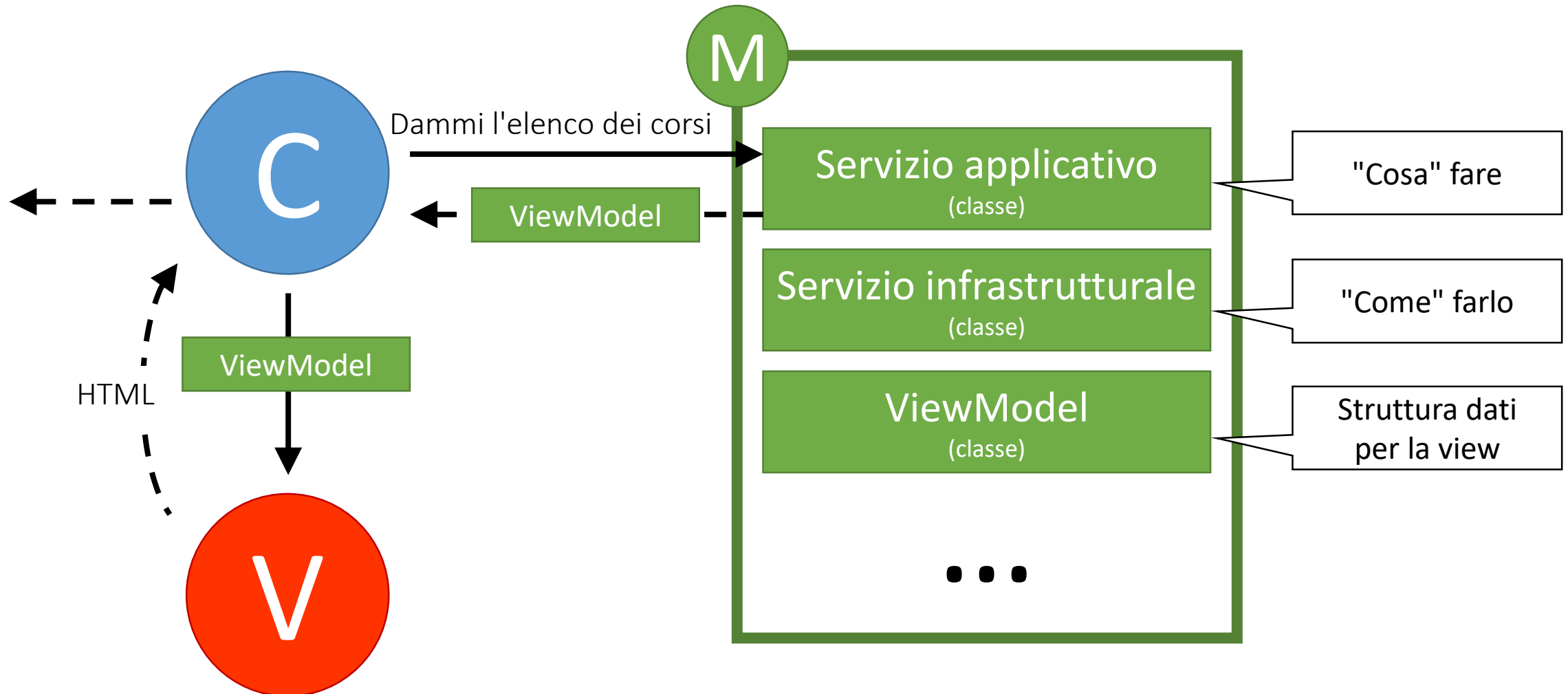


DE PRINDE







QUE TO QUE SUBE  
QUE P



# Organizzazione del model per questo corso



# Suddivisione in directory del model

-  Models
  -  Services
    -  Application
    -  Infrastructure
  -  ViewModels
  -  ...

# View fortemente tipizzate

- La view può dichiarare quale tipo di oggetto è predisposta a presentare;

In cima alla **view di contenuto** mettiamo:

```
@model List<CourseViewModel>
```

- Se servono degli *using*, mettiamoli in `/Views/_ViewImports.cshtml`

```
@using List<CourseViewModel>
```

# View fortemente tipizzate

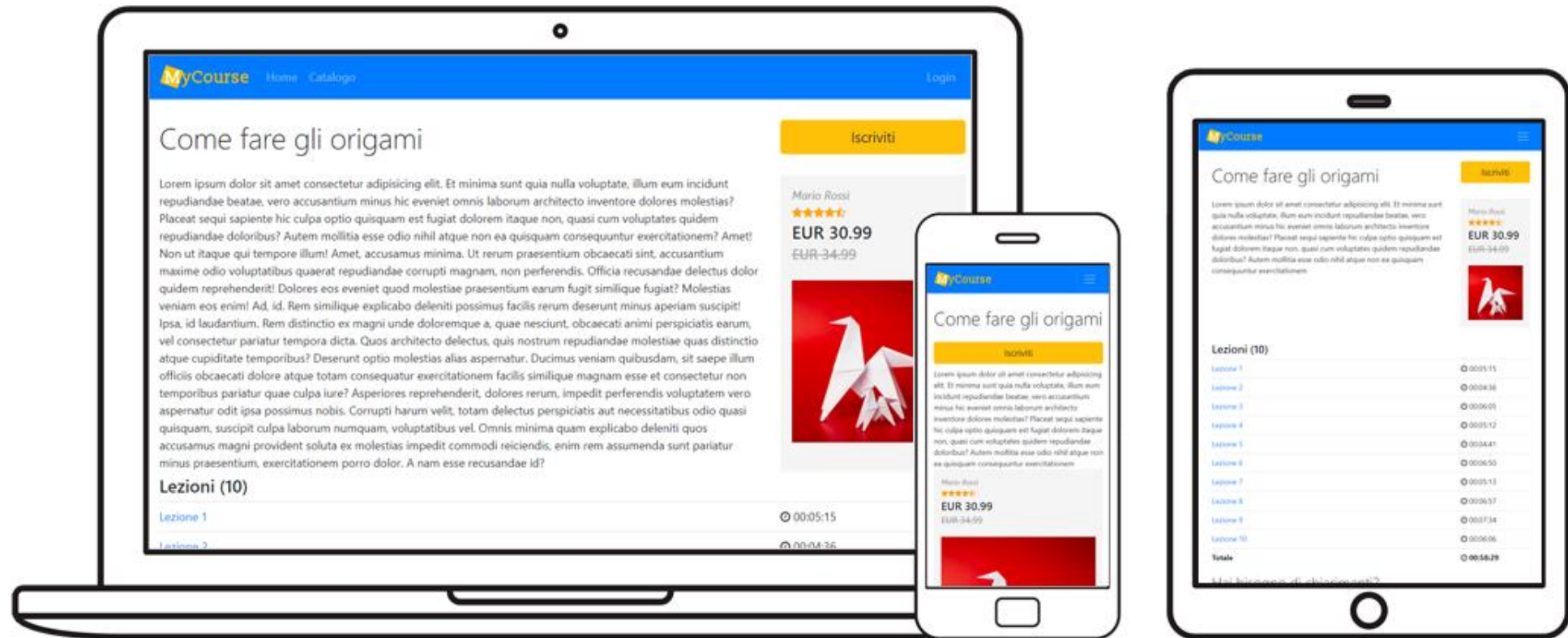
I vantaggi di rendere una view fortemente tipizzata sono:

- Possiamo usare l'espressione `@Model` per accedere alle proprietà dell'oggetto;
- Ci viene offerto l'intellisense, così c'è meno rischio di sbagliare;
- Se, a causa di un bug, il controller dovesse fornire un tipo di oggetto diverso, avremmo un errore che ci rende evidente il problema.

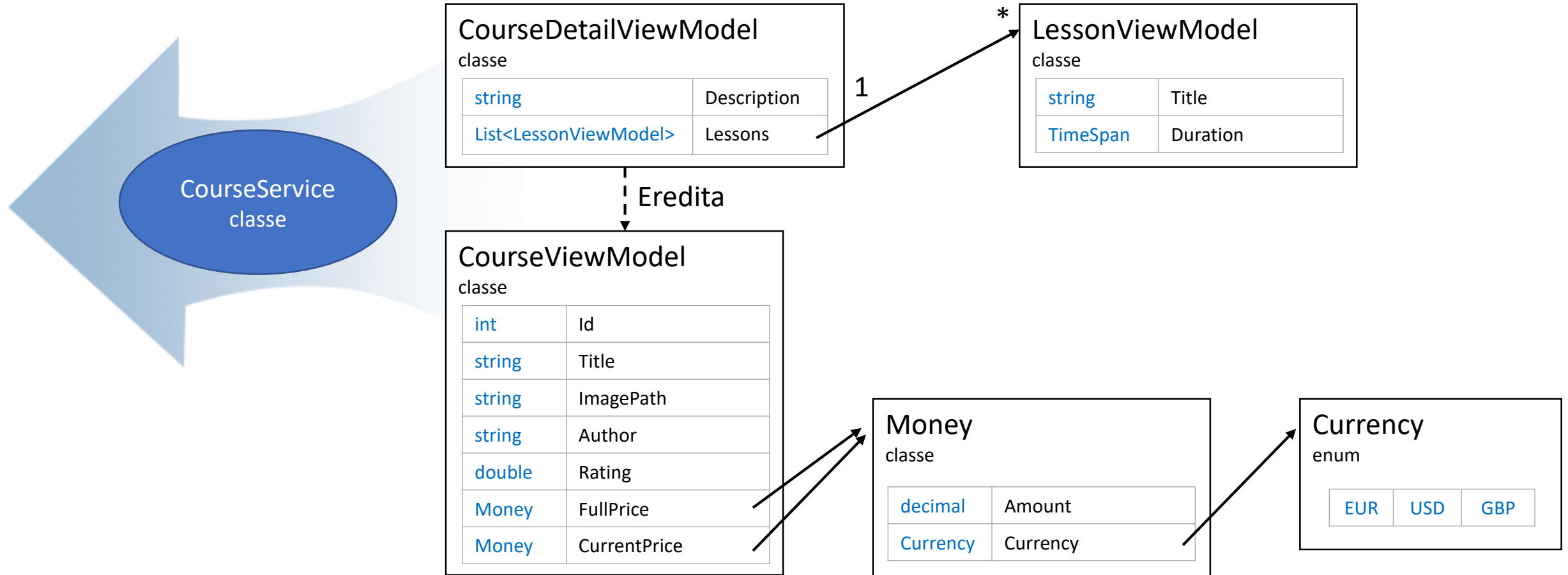


# Visualizzazione responsive

Usiamo il grid system di Bootstrap per rendere fruibili le informazioni di dettaglio anche su smartphone e tablet.



# Il nostro model



# Scambiare dati con le view

- Usiamo l'oggetto **ViewData** oppure l'oggetto **ViewBag**.
  - Servono allo stesso scopo, hanno giusto due stili diversi;
  - Da usare soprattutto per far arrivare dati alla view di layout.
- Nell'**action** o in cima alla **view di contenuto** mettiamo:  
`ViewData["Title"] = "Corso 5";`
- E poi, nella **view di layout**, stampiamo il titolo così:  
`<title>@ViewData["Title"]</title>`

# Scambiare dati con le view

- Usiamo l'oggetto **ViewData** oppure l'oggetto **ViewBag**.
  - Servono allo stesso scopo, hanno giusto due stili diversi;
  - Da usare soprattutto per far arrivare dati alla view di layout.
- Nell'**action** o in cima alla **view di contenuto** mettiamo:  
`ViewBag.Title = "Corso 5";`
- E poi, nella **view di layout**, stampiamo il titolo così:  
`<title>@ViewBag.Title</title>`