Device-specific Use-cases

Dino Esposito
http://software2cents.wordpress.com
@despos

pluralsight
hardcore dev and IT training

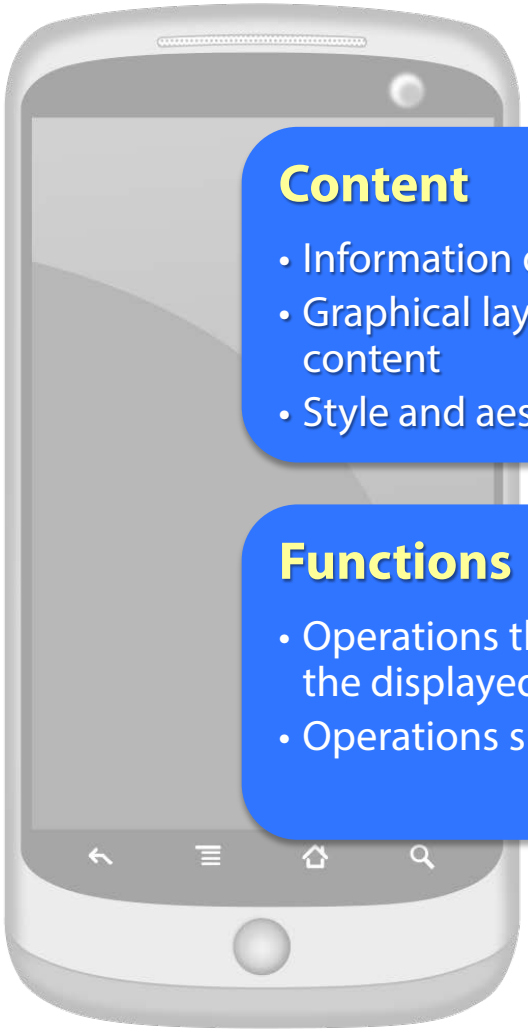
Key Points

Device-specific views in ASP.NET

Device-specific **use-cases**

Refactoring of the ASP.NET solution

Device-specific: What Does It Mean, Exactly?



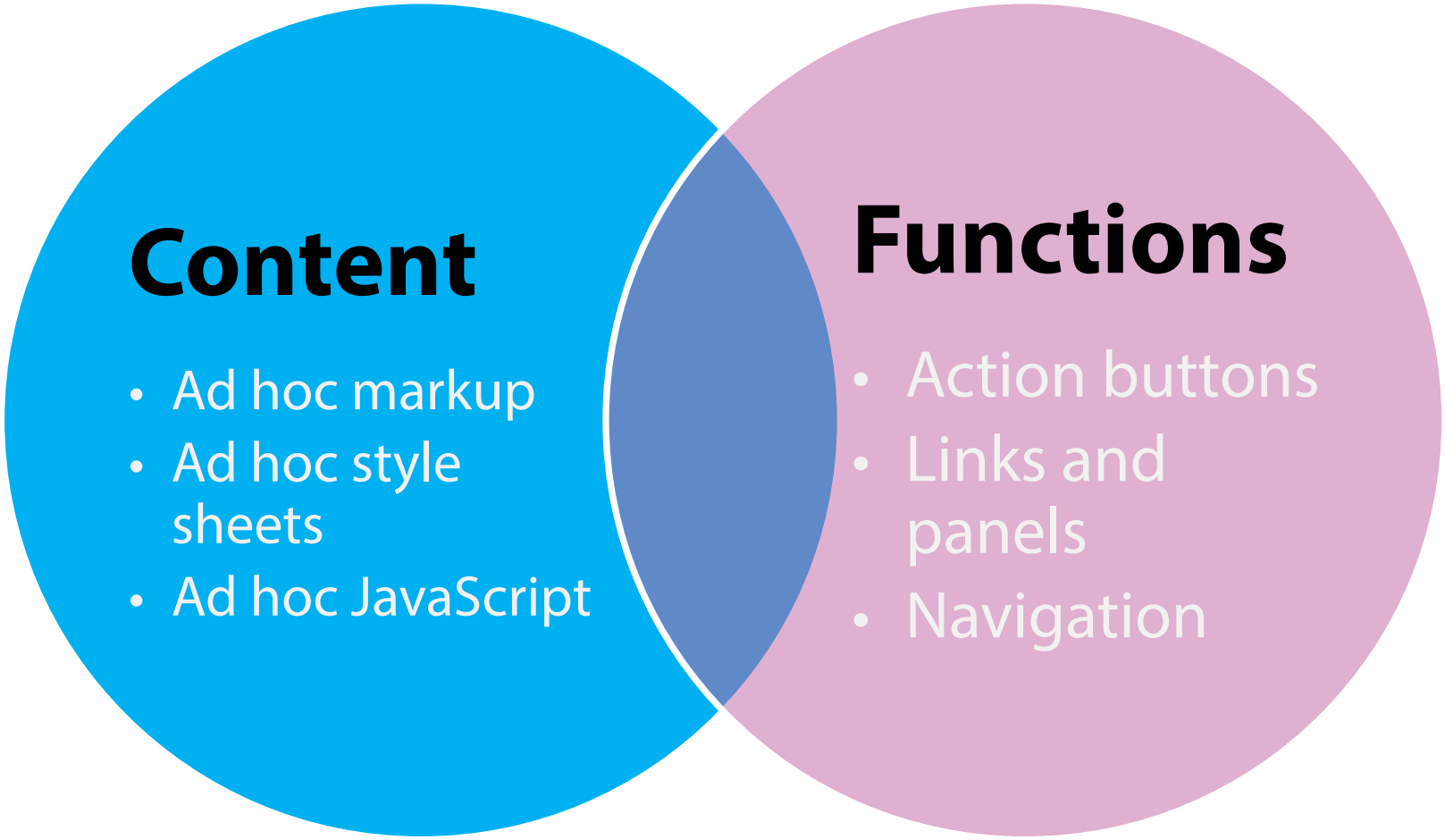
Content

- Information displayed through the view
- Graphical layout and organization of the content
- Style and aesthetics

Functions

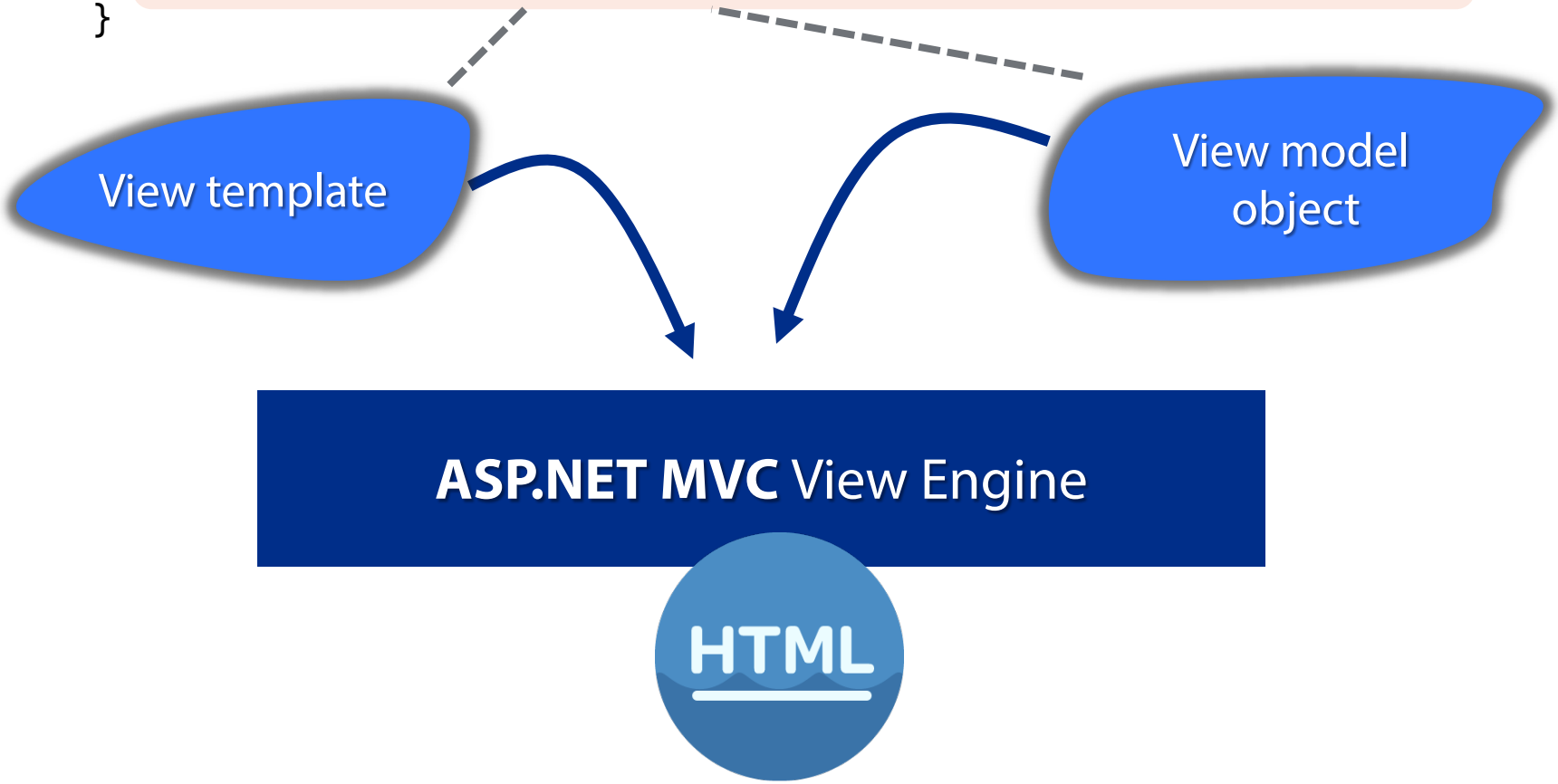
- Operations that can be performed through the displayed content
- Operations specific of the device UI

Device-specific is About…



```
public ActionResult Index()
{
    // Invoke the application layer to provide data for the view
    var model = _service.GetModelForIndex(HttpContext);

    // Invoke the view engine to generate HTML
    return View("index", model);
}
```



Going One Step Further…

- **Display modes automatically adapt template to devices**
  - Matching rule to discover the device profile
  - Suffix to identify the view template to use
- **Far-fetched scenario?**
  - Application layer implements use-cases of the application
  - Devices to (ideally) have their own use-cases
- **How to automatically adapt view model objects?**

Options

- **Per-device controllers**
  - Some of the application logic coded within controllers
- **One set of controllers and per-device application layers**

- **One comprehensive**
  - Set of controllers (all possible endpoints)
  - Set of application services (all possible methods)
  - Set of view model objects (all possible data)

DEMO About ASP.NET MVC and WURFL

Controllers

Application services

Device-specific views

Summary

- **One web site to take care of all devices is possible**
  - DDRs are the key to success
  - WURFL is one of them
- **ASP.NET MVC display modes simplify view routing**
  - Can pass the same view model to any view
  - Ignore what's not being used in the view
- **If too heavyweight, add one more layer of routing**
  - From view up to application services
  - From application services up to controllers
  - From controllers up to controller factories (via IoC)