



WARSZAWSKIE
DNI INFORMATYKI

ASP.NET WebForms vs. ASP.NET MVC

Introducing MVC to WebForms Developers



WARSZAWSKIE
DNI INFORMATYKI

Why WebForms?

- Reaction to ASP “Classic” disadvantages
 - Code EVERYTHING by hand
 - Loosely Typed, bad error handling, no memory mgmt
- Make building web apps more like desktop apps
 - Abstract away HTTP
 - Object Oriented and Event-based, rather than script
 - Strongly Typed code
- Created around 2000

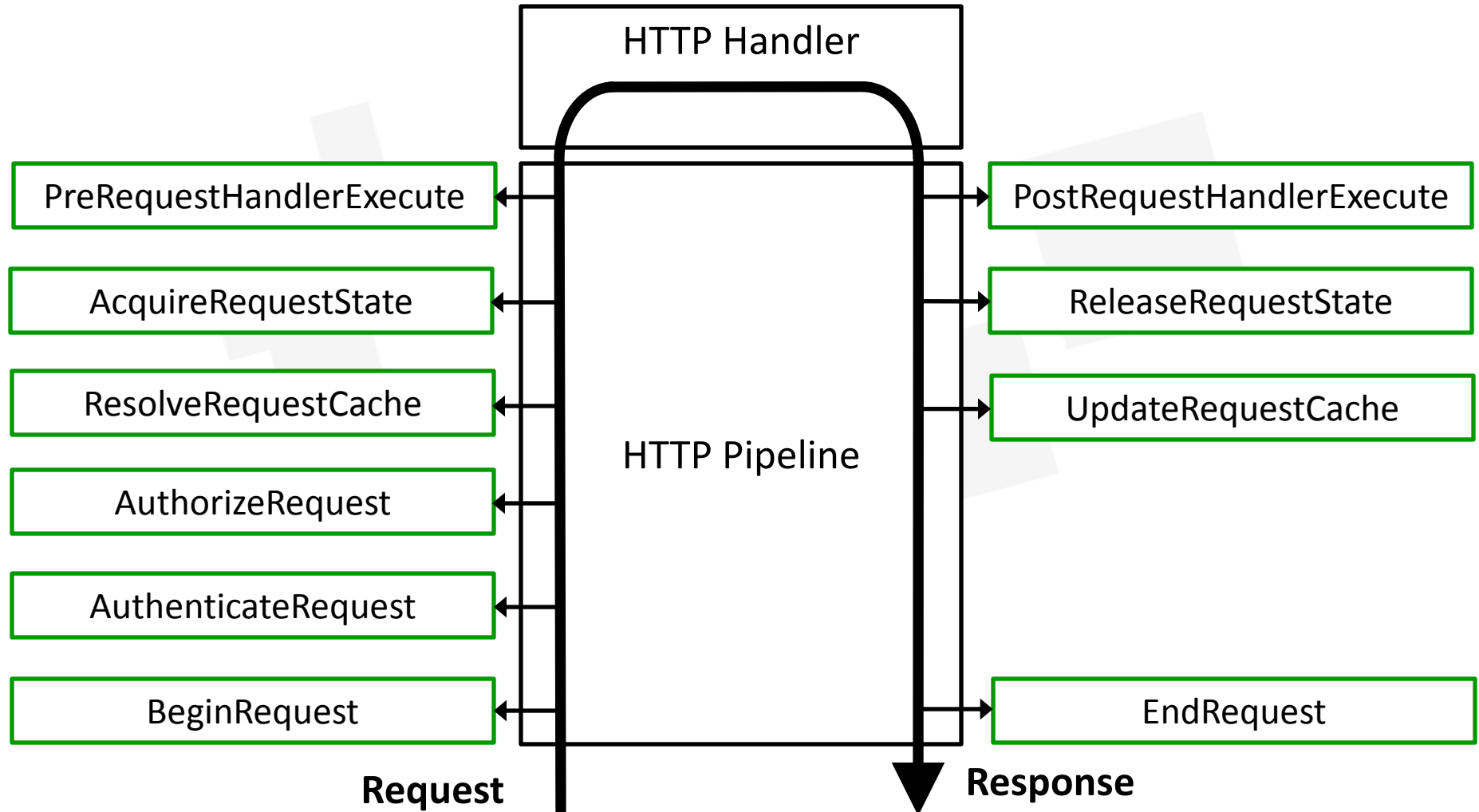


WARSZAWSKIE
DNI INFORMATYKI

ASP.NET Features

- Page & Object caching
- Authentication, Authorization & Membership
- Application State & Session State
- Master Pages & Themes
- Provider Model based functionality
- Virtual File System
- And more ...

ASP.NET Pipeline



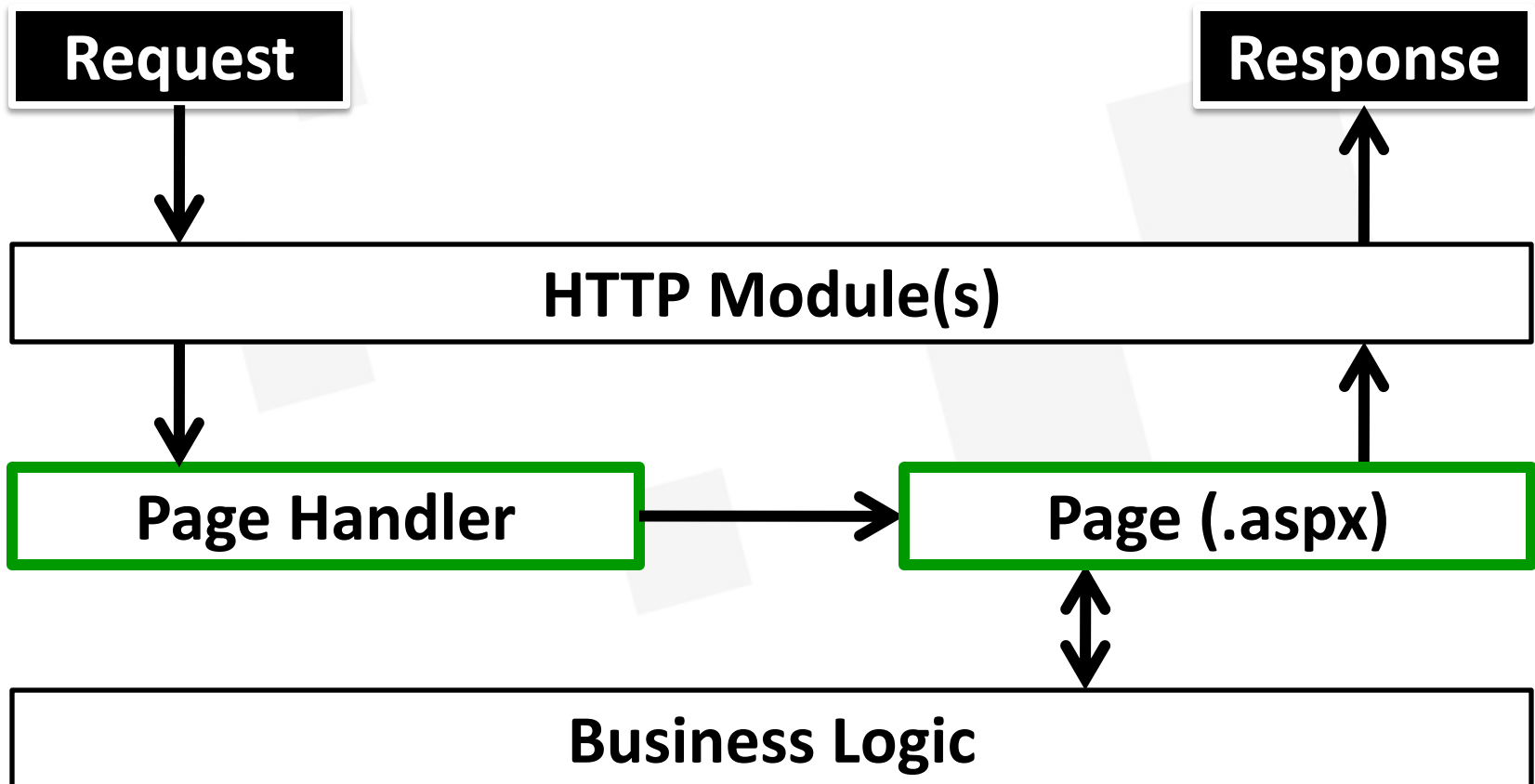


WARSZAWSKIE
DNI INFORMATYKI

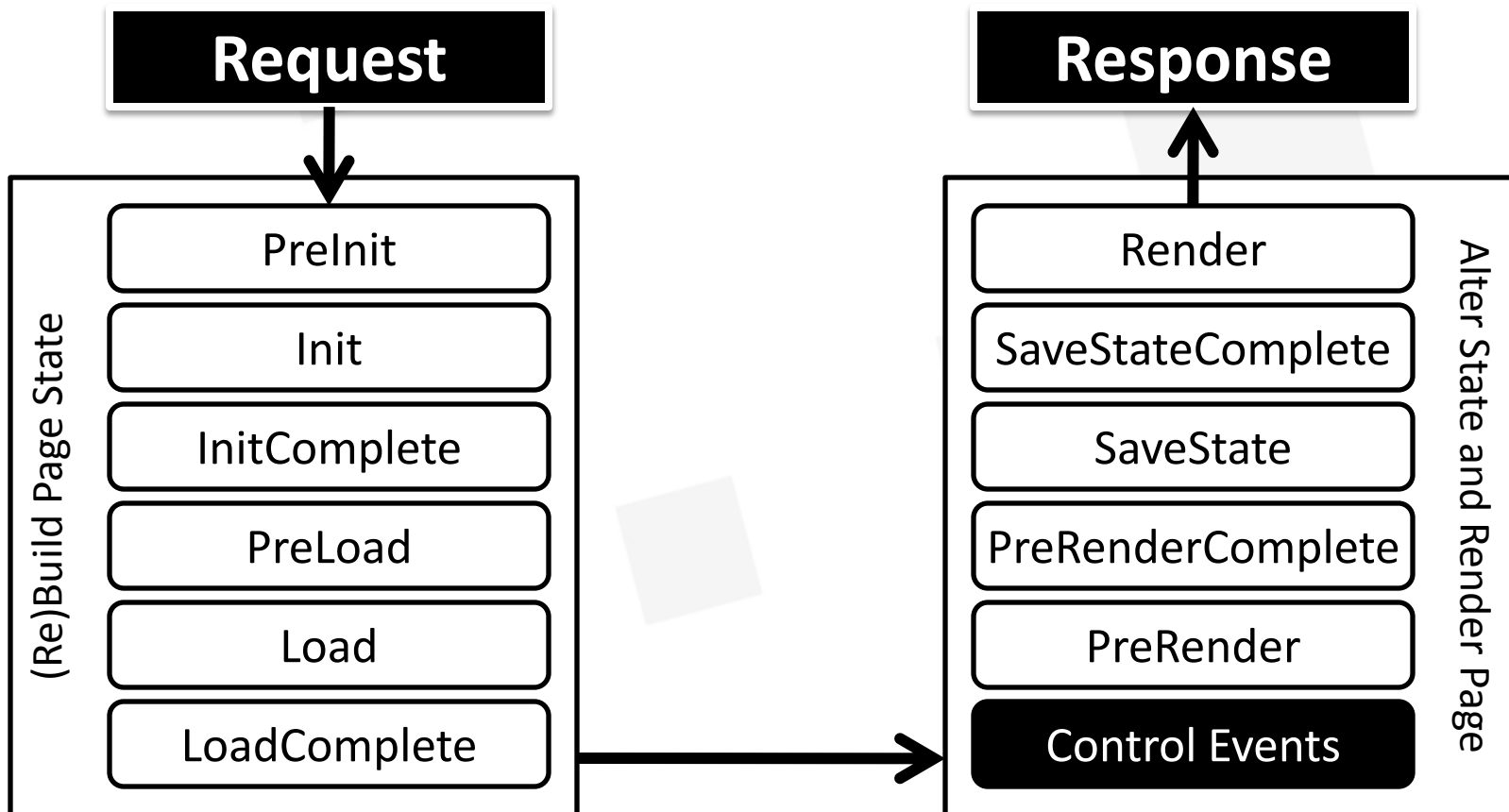
WebForms Basics

- Object oriented Page with an event based lifecycle
- Page can contain server controls
 - Interactive objects rendering part of a page
 - Can have events like Click, SelectedIndexChanged etc.
- Page contains ViewState
 - Hidden data to maintain the illusion of state
 - If managed improperly can become very big (> 1 MB)

WebForms Request Handling



WebForms Page Lifecycle





WARSZAWSKIE
DNI INFORMATYKI



DEMO

WebForms App

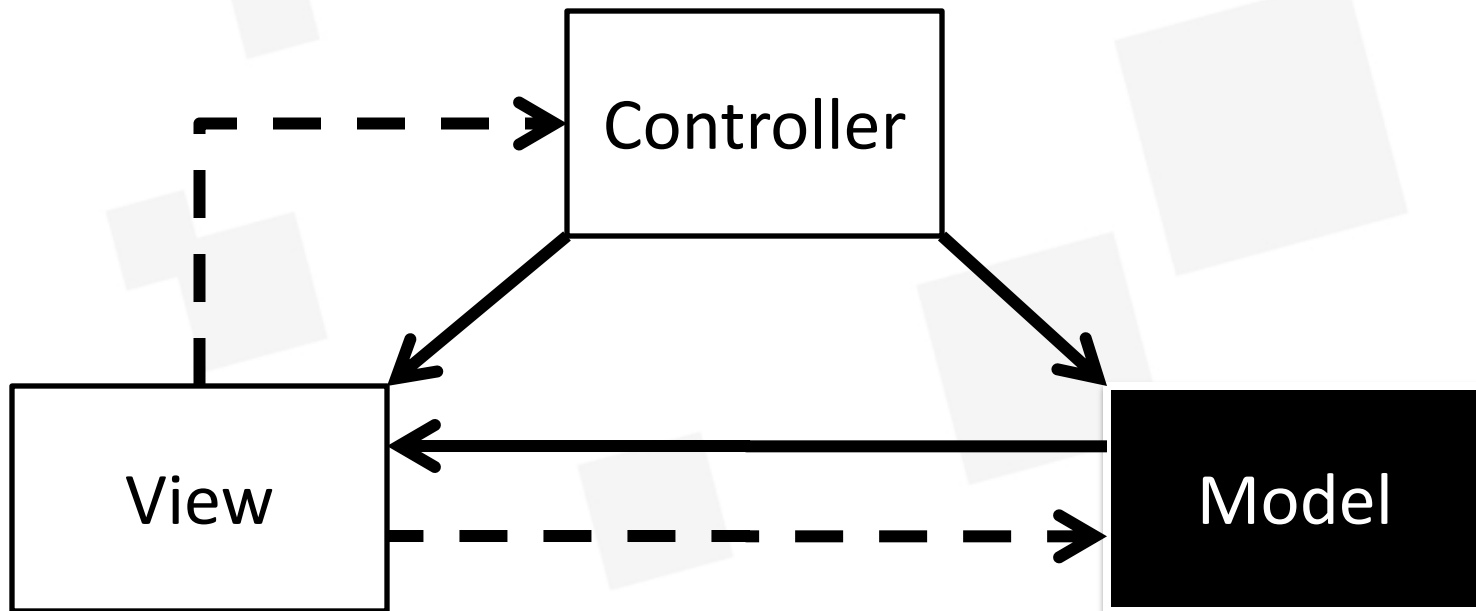


WARSZAWSKIE
DNI INFORMATYKI

MVC

- Design Pattern Invented in 1979 for Smalltalk
- Used in various technologies for the last 30 years
- Used more and more on the web
 - e.g. Struts, Ruby on Rails, Django, and now in ASP.NET
- Separates UI from logic through shared data
 - Enables different UI for same functionality
 - Makes testing UI easier

MVC Overview



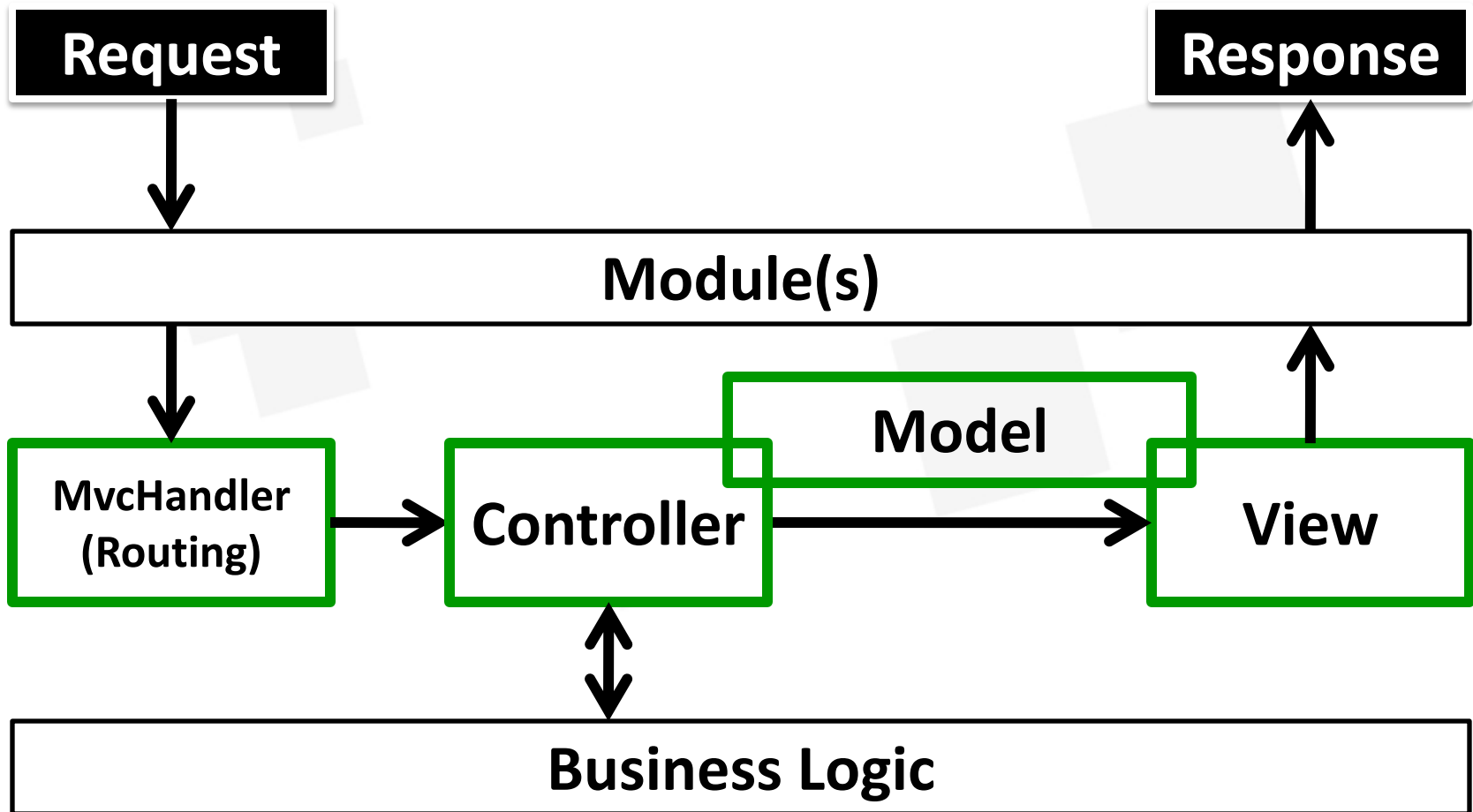


ASP.NET MVC

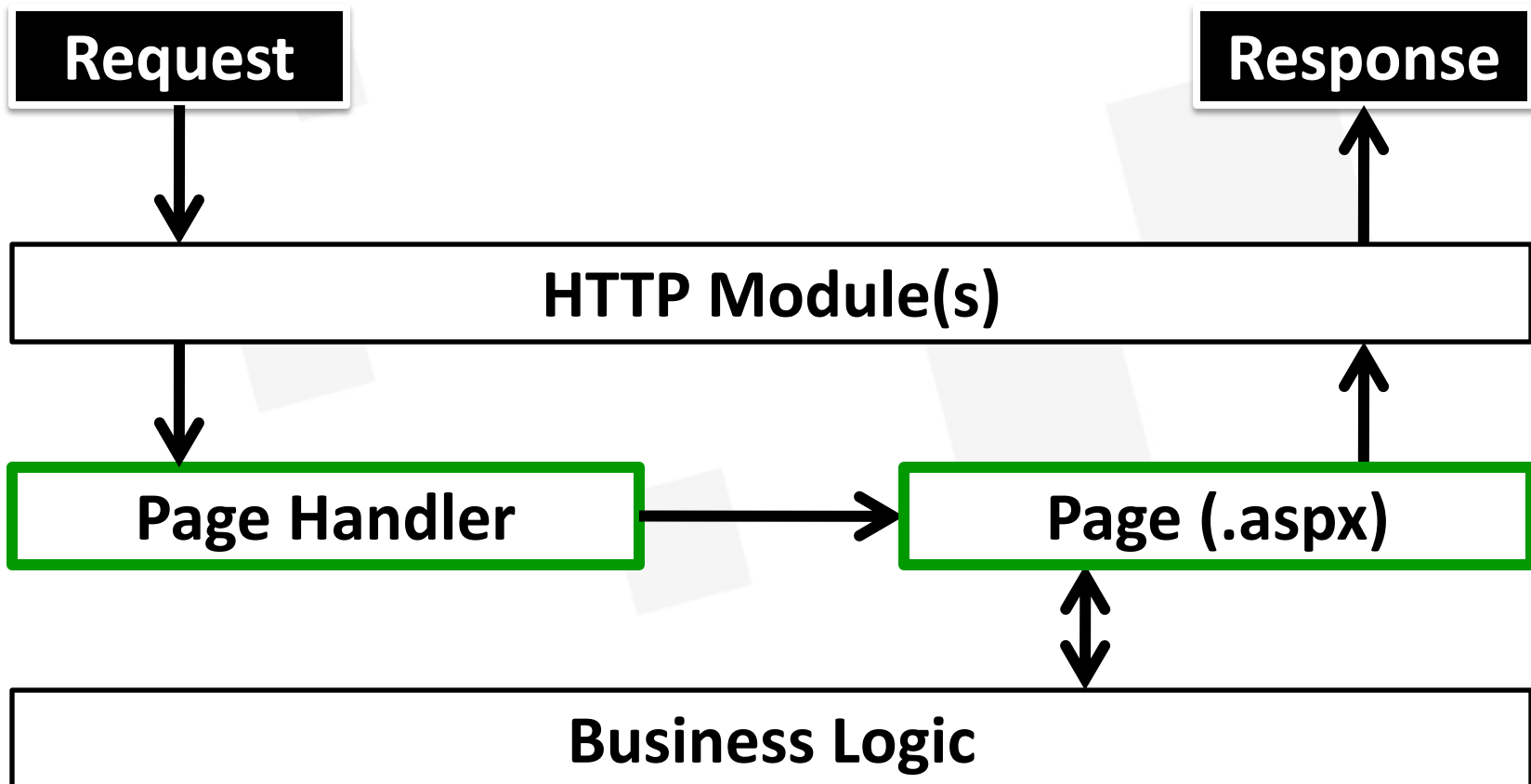
- New ASP.NET project type (WebForms is not gone!)
 - MVC 1.0 extension on .NET 3.5 SP1
 - MVC 2.0 part of .NET 4.0 and Visual Studio 2010
- Excellent for (public) web applications/sites
 - 100% control over HTML (or other output)
 - (Search Engine) Friendly URLs
 - No (illusion of) state as in WebForms
- Easy to test, comes with unit tests by default

MVC

Request Handling



WebForms Request Handling





Routing

- Routing Engine directs requests to proper controller
 - Routes must be defined at application start
 - Maps URLs to controller, action, and parameters
 - Uses naming convention: Trip → TripController
- Routes are evaluated in order, first match “wins”
- Routing implemented by System.Web.Routing
 - Available since .NET 3.5 SP1, not MVC specific



WARSZAWSKIE
DNI INFORMATYKI

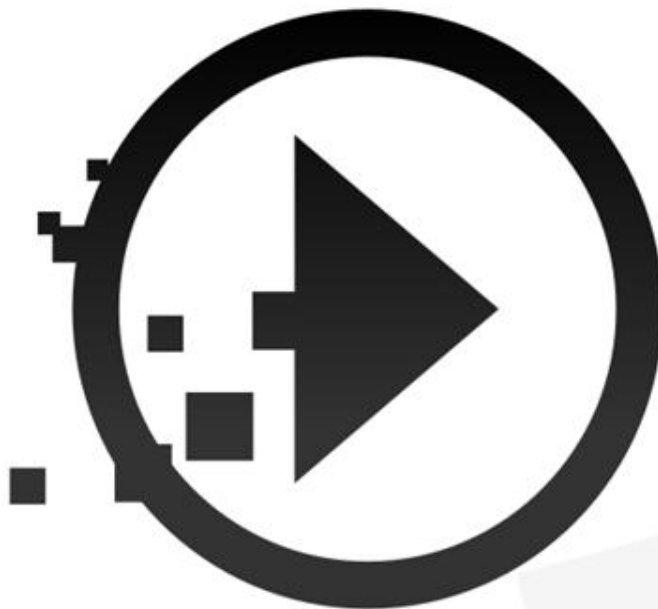
Routing

C#

```
Routes.MapRoute(  
    "Default",  
    "{controller}/{action}/{id}",  
    new { controller = "Home",  
          action = "Index",  
          id = "" });
```



WARSZAWSKIE
DNI INFORMATYKI



DEMO

Routing Demo



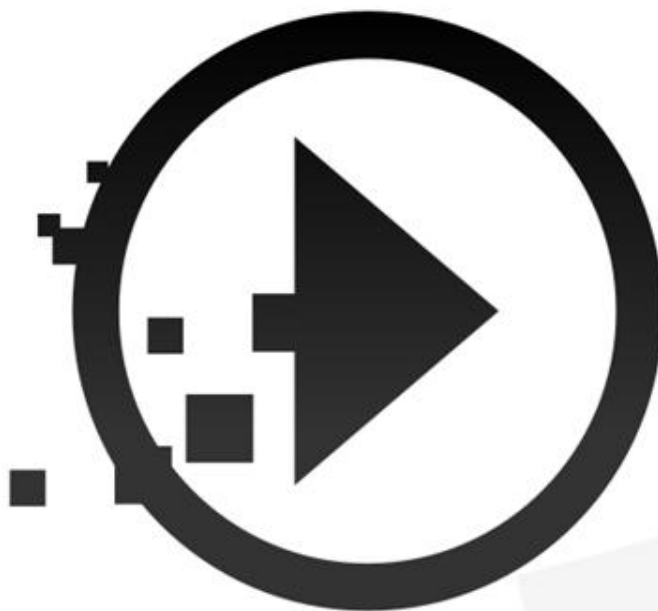
WARSZAWSKIE
DNI INFORMATYKI

MVC Controller

- Class inheriting `System.Web.Mvc.Controller`
- Public methods are Actions called from Routing
- Usually builds a Model to be used later
 - Can use ViewData (State Bag) for model/state
 - Model can also be passed directly → strongly typed
- Usually returns ActionResult telling where to go next



WARSZAWSKIE
DNI INFORMATYKI



DEMO

Controller Demo



WARSZAWSKIE
DNI INFORMATYKI

Action Filters

- Action Filters are attributes applied to actions
- Can execute code before/after action, independently of the action
 - Authorization
 - Logging
 - Exception handling
 - Output caching
- Build your own based on `ActionFilterAttribute`

Testing ASP.NET MVC Apps

- MVC removes UI dependency → easier to test
- Interfaces and abstract base classes to mock/stub
 - IViewEngine
 - HttpSessionStateBase
 - HttpContextBase
- Works with different testing frameworks
 - NUnit, XUnit, MBUnit



WARSZAWSKIE
DNI INFORMATYKI

WebForms vs. MVC

WebForms

- Intuitive model, mimics real life
- Abstracts HTTP protocol
- Page/Controls render output
- Focused on one UI type: HTML
- Hard to test UI
- Many existing apps and controls
- Good RAD platform

MVC

- Indirection less intuitive
- Direct interaction with HTTP
- View enables full output control
- Usable for many UIs
- UI testable by testing controller
- Relatively new, less available
- Helpers make life easier



WARSZAWSKIE
DNI INFORMATYKI

Summary

- MVC is a new model in ASP.NET
- Does not replace WebForms → Choice
- Benefits from pre-existing WebForms features
- Optimized for web: Clean HTML and URLs
- Separation of concerns enables cleaner code
 - Indirection must be understood by developers
 - Allows for testability and extensibility



WARSZAWSKIE
DNI INFORMATYKI

Michiel van Otegem

mail: michiel@aspnl.com
www: michiel.vanotegem.nl
