

# Migrating to ASP.NET Core

## Challenges & Opportunities

Dino Esposito

[dino.esposito@jetbrains.com](mailto:dino.esposito@jetbrains.com)

It's a relevant update ...

**FOR  
EVERYBODY?**

# Migration?

## Challenges

- MVC only
- C# only
- New API
- New hosting model
- No IIS-centrism
- New .NET FX

## Opportunities

- X-platform
- Improved API
- Same MVC paradigm
- Total control

**YOU DECIDE.**

# What We Have Now...

**CURRENT**

**.NET Framework**

**ASP.NET**

**Web Forms**

**MVC**

**Web API**

**system.web**

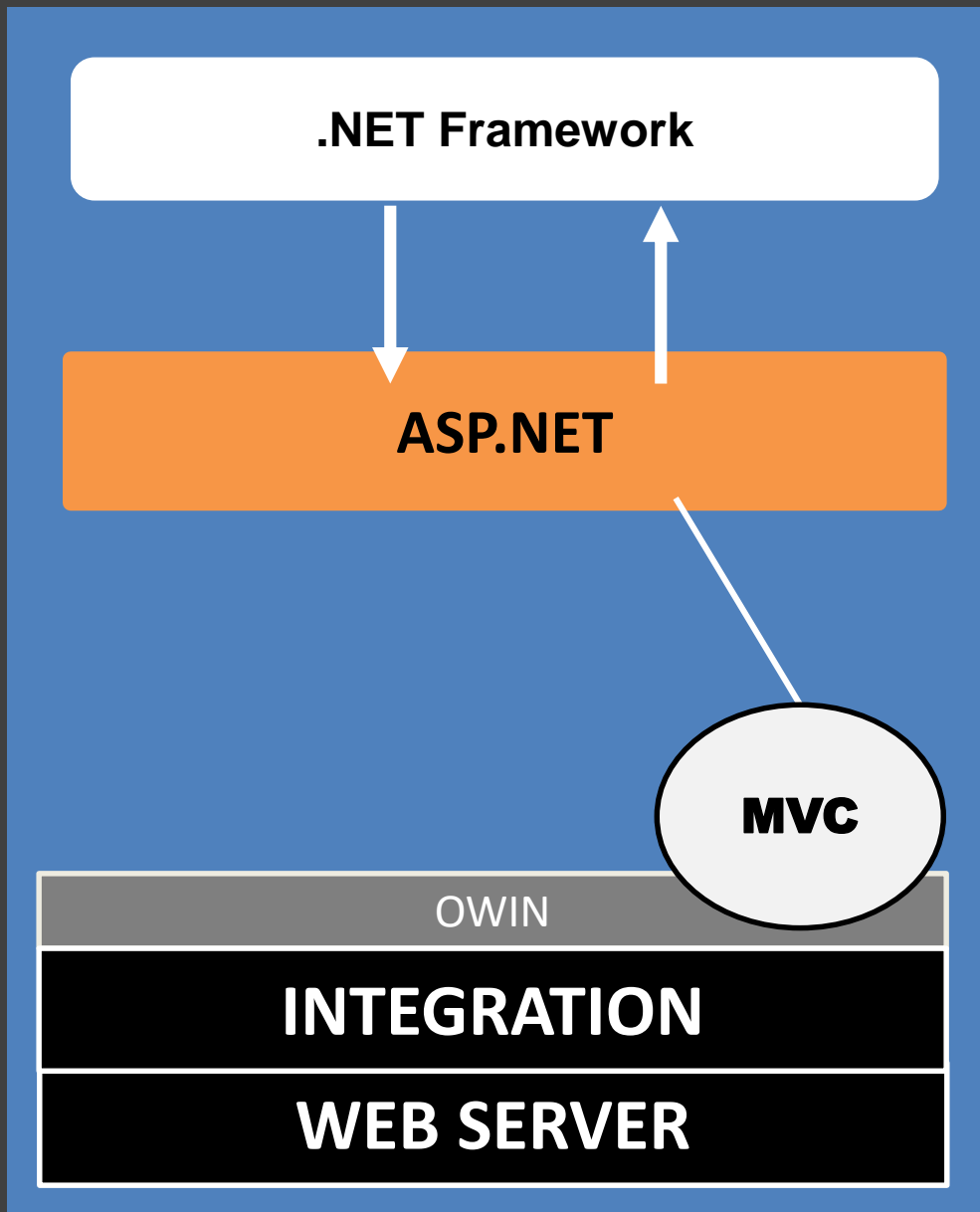
AD HOC

OWIN

**IIS**

**INTEGRATED PIPELINE**

**FUTURE**



# Changes

- **Tight coupling between ASP.NET and IIS**
  - Presented as a big win only a few years ago
- **Platform dependence**
  - Windows only
- **System.web is legacy code**
  - Devised back in 1997

# When they took **SYSTEM.WEB** apart ...





- From EMPTY to FUNCTIONAL in Visual Studio 2015

It's a complete change of platform and to  
some extent also tools and habits

# Migration Facts

- **Find out who moved the cheese**
  - ASP.NET CORE has a high cost of setup
  - Costs tend to zero over time
  
- **Find out why they moved the cheese**
  - Some good reasons
  - Some "other" reasons

# Good Reasons

- **New runtime environment**
  - Unification of ASP.NET and WebAPI pipelines
  - Old system.web dropped
  - Memory footprint
  
- **All web app services decoupled to interfaces**
  - Authentication
  - Static files management
  - Exception handling and logging



# DECISION POINT

## You and IIS

- **Having problems with IIS?**
  - Looking for Apache Server?
- **Some node.js nostalgia?**
  - Wishing to call yourself a "geek" again?

cmd.exe

```
C:\> _
```



## DECISION POINT

# You and ASP.NET

- **Having problems with Web Forms?**
  - Looked into ASP.NET MVC and/or plain SPAs?
- **Having problems with ASP.NET MVC?**
  - Looked into Web Forms and/or plain SPAs?
- **Sure it's not your code that sucks?**

# There's not much left that can be added to ASP.NET

- Battle-tested, well-known and stable
- Room for small improvements
  - Razor
  - DI
  - Azure integration
  - Pipeline customization



**What if you're planning a  
complete rewrite?**



# What's different?

- **Configuration, startup, pipeline**
  - Free IoC though 😊
  - Full rewrite of HTTP modules (including tools)
- **New ways of doing old things**
  - Core authentication (except Identity)
  - Embedded resources
  - Project management
- **Development experience**
  - Test, build, deploy, manage dependencies

# ASP.NET MVC

- **No support for Web Forms**
- **Migrating code is overall doable**
- **Switching to ASP.NET Core mindset is easy**
  - Aside a few new APIs to learn, it's all the same
  - Plus minor features in Razor

# The underlying .NET

- **.NET Framework**
  - As above
  
- **.NET Core (x-platform)**
  - EF Core
  - No SignalR
  - No 3<sup>rd</sup> party frameworks (if not ported)
  - Subset of the .NET Framework (and no VB)

# Plan moving to ASP.NET Core **especially** if you're experiencing any of the following:

- Issues with IIS and system.web
- Issues with the app on the cloud
- Performance issues in the core runtime
- ... or see a business opportunity in x-plat hosting

**Inevitably the future.**  
**But nobody knows when**  
**the future will be.**

