



.NET Conf

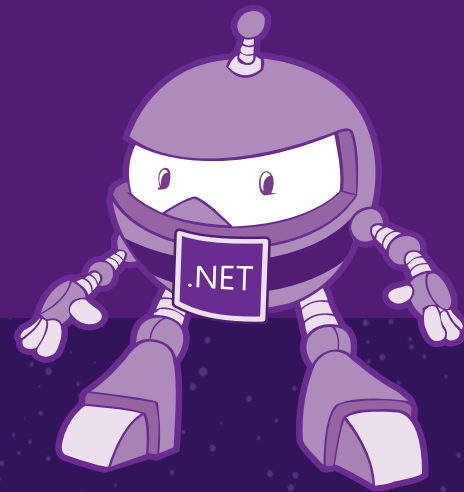
探索 .NET 新世界



Host by
STUDY4

該準備從 .NET Framework 4.x 遷移至 .NET Core 3.0 了嗎？

Gelis



關於我

吳俊毅 Gelis - FB 軟體開發之路-經營者

- 部落格 (Gelis 技術隨筆)
<http://gelis-dotnet.blogspot.tw/>
- FB 粉絲團(Gelis 的程式設計訓練營)
<https://www.facebook.com/gelis.dev.learning/?ref=bookmarks>
- FB 社團 (軟體開發之路)
<https://www.facebook.com/groups/361804473860062/?ref=ts&fref=ts>

集英信誠-資深.NET技術顧問



Agenda

- 從「傳統架構 SOA」到「商業邏輯導向」架構
- 從「傳統架構」計談 Clean Architecture 的軟體架構設計
- 談原本的 EasyArchitect Framework 設計目標
- 解決跨平台 Lib 共享問題的 (.NET Standard)
- 客製化的 Web API 伺服器服務框架
- 實作：使用 EasyArchitect Framework 打造 ApiHost 網站
- 實作：使用 .NET Standard 從無到有打造跨平台的 Web Api 框架

Agenda

- .NET Framework vs. .NET Core
- 漫談 .NET Core & ASP.NET Core
- 什麼是 .NET 5 ?
- 原有 ASP.NET MVC 5 遷移至 ASP.NET Core 3.0 的挑戰
- 相關 Migration 工具
 - .NET 可攜性分析工具 .NET Portability Analyzer
 - 使用 .NET API 分析器 PortApi CLI
- 移轉的實例：Demo
 - 使用自行開發 VS Extension & CLI 工具移轉 ASP.NET Web API to ASP.NET Core

.NET Framework vs. .NET Core

	.NET Framework	.NET Core
跨多個平台	否	是 (勝)
Open Source	否	是 (勝)
Third-Party Components 支援程度	廣	日漸增加、因為 Open Source，未來即將超越 .NET Framework
Windows GUI	Windows Form、WPF	目前無、發展中 WinForm/XAML for .NET Core (目前Windows Only)
Microservice/Docker	侷限於 Windows	優 (勝)
效能	佳	極佳、更適合企業級後台系統 / Cloud 環境 (勝)
開發成本 / 擴充	不錯、但只在 Windows 上	Write Once, Run Anywhere、內建 DI、Middleware、 延展與擴充 (勝)
部署	Install .NET Framework / MSI / GAC...	FDD / SCD / FDE (勝)
略...

為什麼要遷移至 .NET Core



.Net core現在可以做什麼？未來發展有前景嗎？

贊同51

11 條評論

分享

收藏

感



find goo

MOV AX, BX

12人讚同了該回答

我有很多程序都是.net 2.0平台上開發的asp.net和命令行程序。目前更換。

.NET Standard 2.0是支持.net 2.0絕多數，這樣可以在linux服務器運行發現穩定性比以前高多了，沒死過一次機。

我的.net程序都是量化交易程序，一般是自己用，也不會有什麼起c#開發起來速度比java快，運行速度似乎比python要快點，這類用語言表達出來而不是追求性能，因為高頻交易在國內有昂貴的，基本不可能用上，做日線上的交易因為t+1一分鐘或一個小時跳出不能掙錢，掙多錢，和你用什麼語言沒有關係（當然好的語言事半功倍的代碼化。

我覺得.net core用於移植以前的程序挺好的，開發很多程序需要mysql數據庫，這樣在linux下兼容性好，在windows下也能運行sqlite,mysql,mongodb這些通用的平台，一個成熟的量化交易平臺的增加c#在軟件工程特徵比js/php/python更有表現力，比java發。

unity3d在ios平台的火爆，c#語言大量應用於遊戲（如王者榮耀、仙劍奇俠、穿越火線、時空召喚、全民無雙....），讓微軟看到只是需要提供一個運行環境。程序發展到一定程度就會慢慢穩定。看好c#，雖然大型電商類應用推薦java，但多數小型程序推薦用c#用java代替c#的地步，你也就不差錢了，到時再換也不遲，c#在金錢的力作，現在有了.net core，在開發時注意一下兼容性，完碼更有生命力。



Gelis Wu ▸ 軟體開發之路

10月13日下午11:07 · 技術討論 ·

關於 .NET Core 的未來，許多社群開始有期待性的討論著，大部分的朋友也都看好 .NET Core 的未來發展。

雖然 .NET Core 在 Server 端端缺少像 express、spring、hadoop 這類的殺手級應用，有如 go 有 docker 和 kubernetes 支撐一樣；相較於以往微軟對開源的態度，這次社群應該比以往會更熱。未來只會越來越多支援 .NET Core 或以下可以直接稱作 .NET 5 好了。

事實上 .NET Core 的 Middleware 的架構與設計方式與 express 非常接近，與 .NET Framework 相比非常的輕量！花些時間去了解非常值得！

再加上執行速度比 Python 要快一點 <== 這不是我說的 XD

且 .NET Core 比 Java 更容易開發 <== 這也不是我說的 XD

.NET Core 在 Linux 下的相容性非常棒 <== 你還有什麼理由不學 .NET Core 呢？

再來，.NET Core 現在能做的事情包括：巨大的 .NET Standard 的函式庫的支持。

ASP.NET Core 可以用來做 Web MVC 程序，也可以寫 WEB API，支持 Self host、Server 有 Kestrel 和 WebListener。前者跨平台，後者只支持 Windows。EntityFrameworkCore，幫助你訪問數據。

現在支持 MySQL、Oracle、Sql Server、PostgreSQL、SQLite。現在甚至連 MongoDB 已經出了支持 .NET Core 的 client，rabbitmq 也出了支持 .NET Core 的 client，全新的 SignalR 也開發完成！支持 Web Socket Server。

你還有什麼理由觀望呢？XD

<https://www.zhihu.com/question/56485550>



ZHIHU.COM

.Net core现在可以做什么？未来发展有前景吗？

...以做什麼？未來發展有前景嗎？

不是好架構師

來發展。

器開發，拋卻了 AspNet 的臃腫組件，非常輕量，但現在的 rc 版本功能還沒有跟進，可以說還處於發展的初級階段，因此真正未來的情況還要看兩

台優勢，後期勢必會加入 .Net Core 的支持，未來相信 AWS（特指會加入其支持（現在 Lambda 已經支持 .Net Core 1.0），加上微軟的跨平台性，對於開發人員也非常友好，所以整體環境來說是健康發展的，未來技術大的優勢。

就是推廣力度及社區支持度，但由於 .Net 系開發者相對於 Java 來說薪資也不如 Java 系，因此未來還很難說。

把一些邏輯使用 .Net Core 來寫，但是相對標準及現在的功能真的太弱會把 .Net 標準集的功能逐漸加入，也許還會提供 libuv 的 .Net 封裝，方便

press、spring、hadoop 這類的殺手級應用，有如 go 有 docker 和相較於以往微軟對開源的態度，這次社區應該比以往會更熱。

技術的存在導致 C# 的接受程度更高，其語法糖及包管理機制也是比 Java 更工業級語言的地位也會越來越高。

pre 值得學習，值得期待；兩年後也許就是這門技術開始火熱的時候。

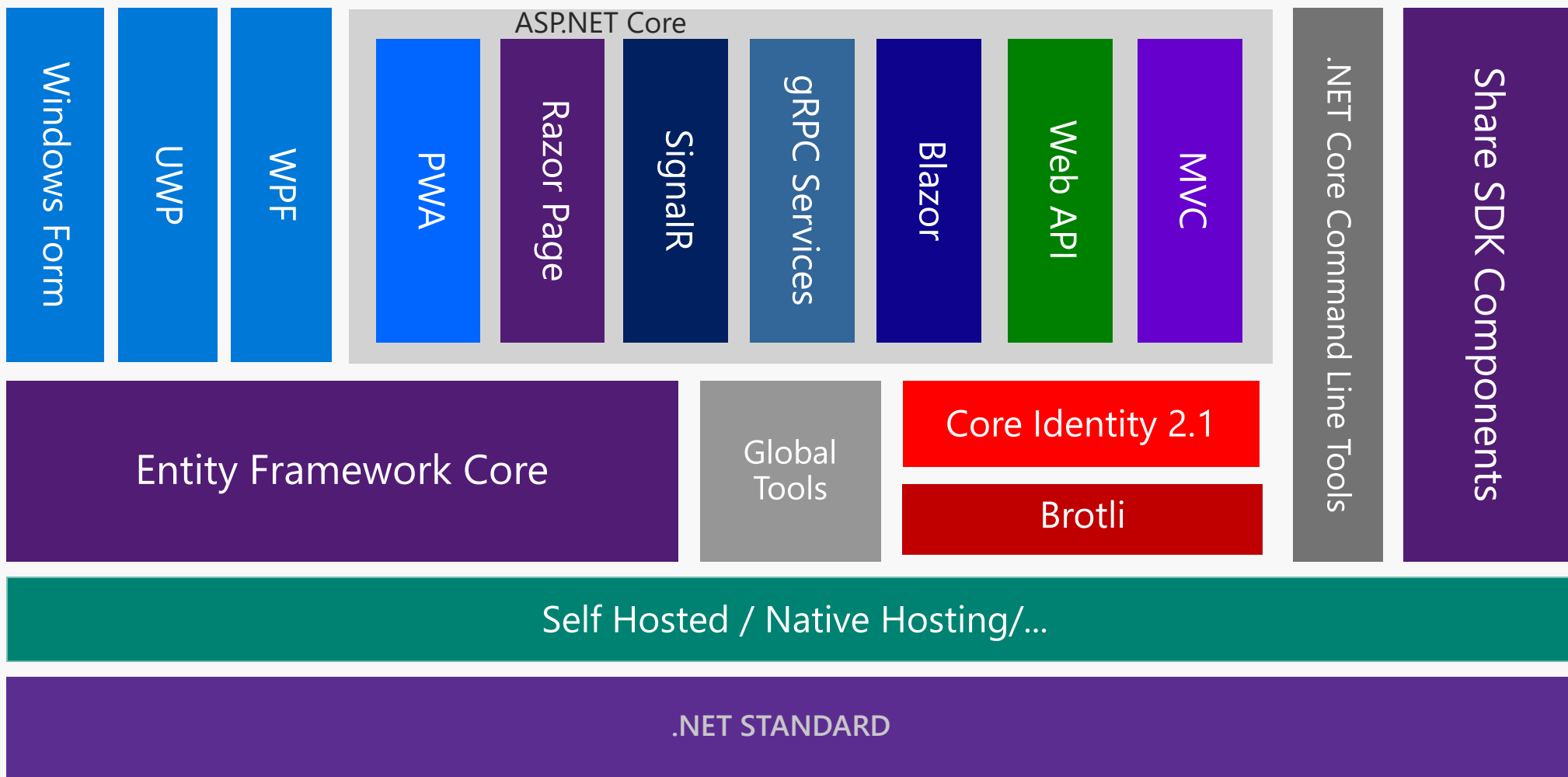
有心理準備，接受可能較低的薪資水準。

堅持下去，因為興趣是最好的老師；然後在現在的基礎上多接觸學習服務器開發，你會有收穫的！

法通，學就對了！

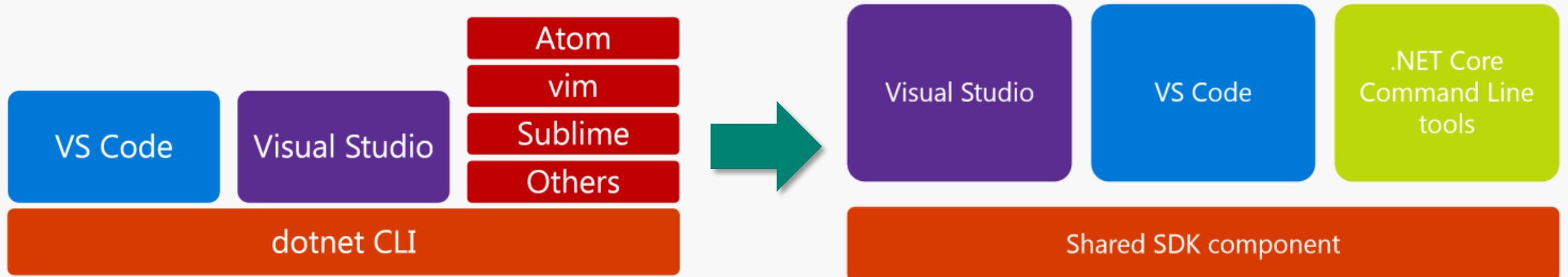
.NET Core 3.0 裡面有什麼？



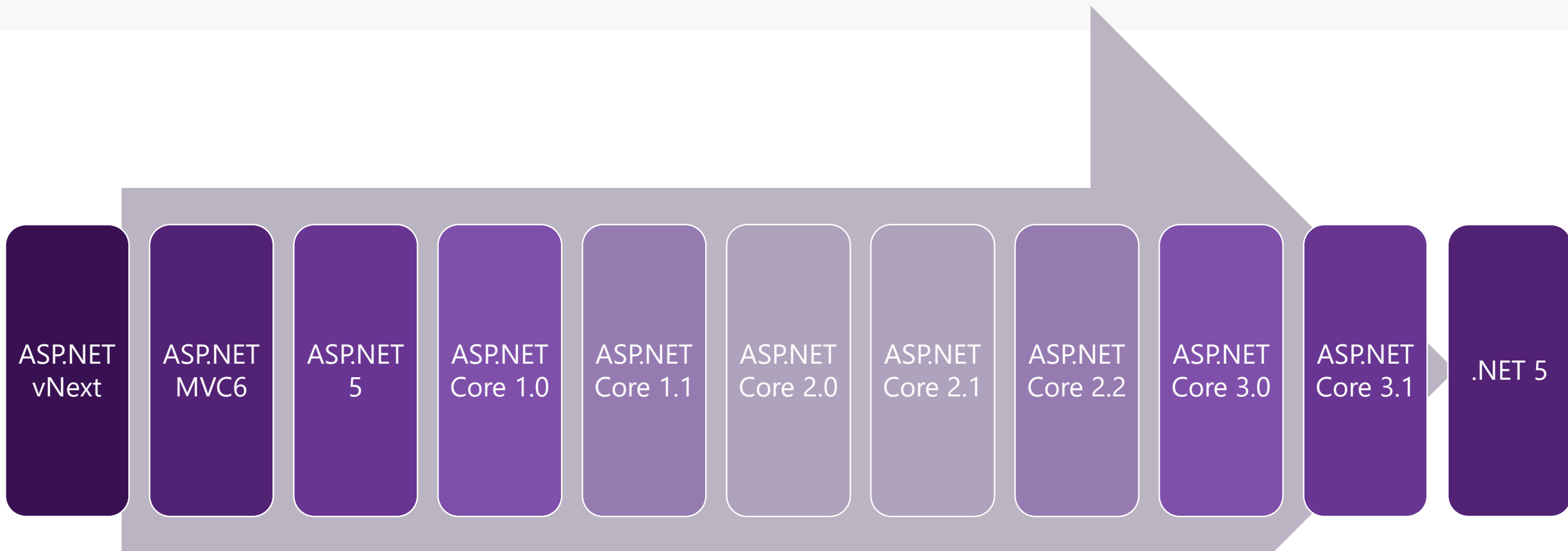


.NET Core 與 ASP.NET Core 的關係

- 2014 - ASP.NET Core 依賴 ProjectK 後來更名 DNX
- .NET Core 1.0 取代 DNX - 因此 dnvm 從此停止發展
 - 取代 Mono 成為真正 Cross-Platform 的執行環境
- **dotent CLI 時代**



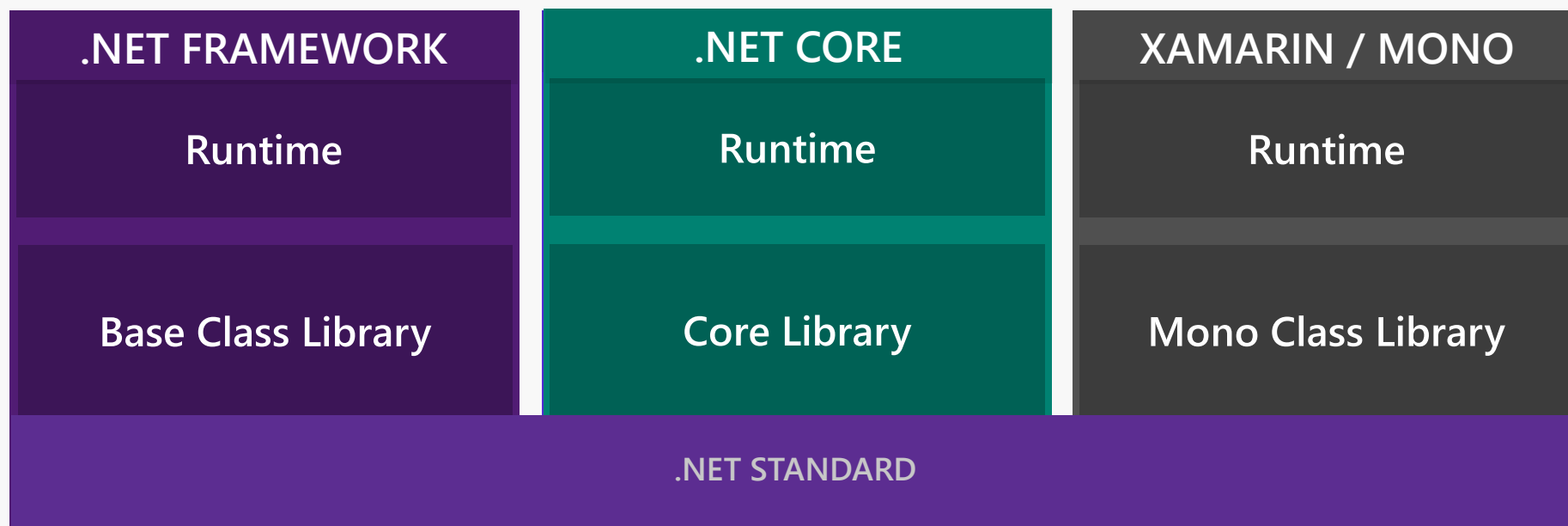
ASP.NET Core 的發展概況



什麼是 .NET 5?



.NET 5



什麼是 .NET 5 ?

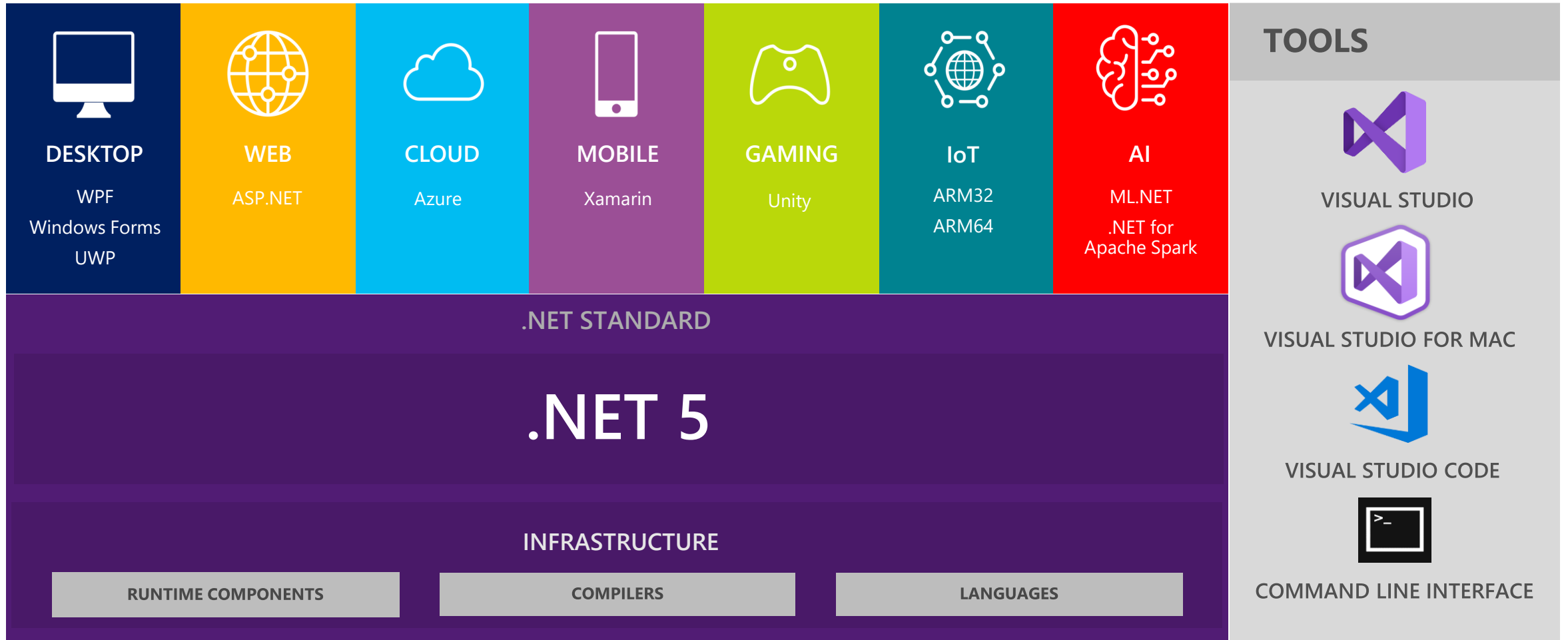
- **.NET 5 = .NET Core vNext**
- One Base Class Library (.NET Framework / .NET Core / Xamarin, Mono)
- 功能更強大的 CLI、**所有應用程式都可使用 CLI 建立**
- CoreFX 更會在 .NET (Ahead-of-Time - AOT) 的靜態編譯上持續發展
 - **程式編譯後佔用空間更小，並支持更多的作業系統**
- 更高的執行效率
- AOT 編譯、包括 Blazor 也都將使用 Mono AOT
- 所有 .NET 5 應用程序都將使用 CoreFX 框架
- C# 將與 .NET 5 同步發展，.NET 5 的開發人員都可使用最新的 C# 版本和功能

.NET Schedule



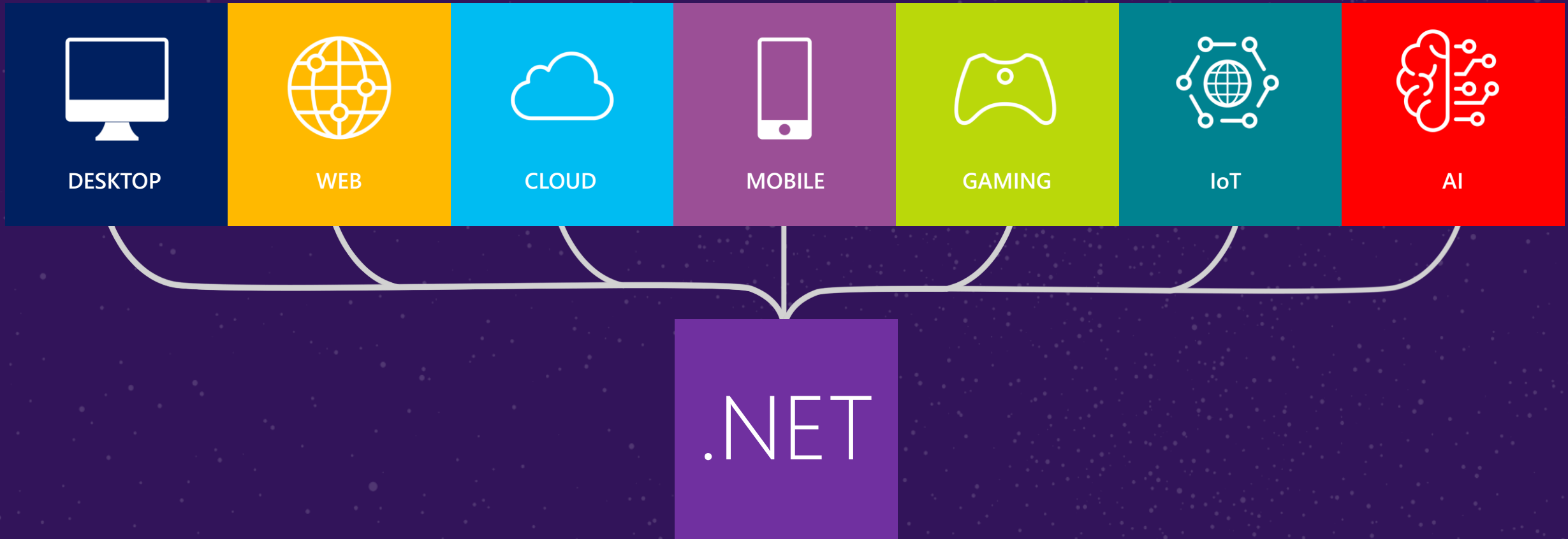
- .NET Core 3.0 released today!
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

.NET – A unified platform



Download .NET Core 3.0 Today!

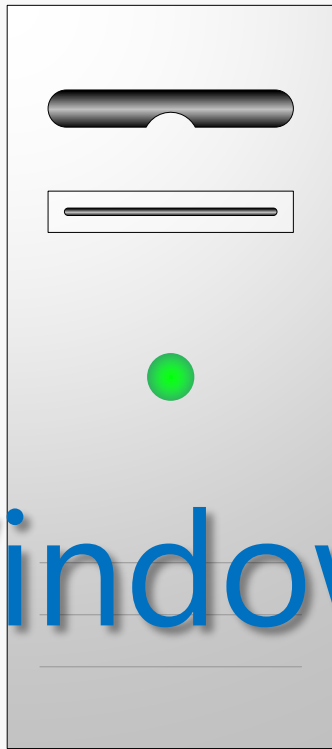
dot.net/get-core3
visualstudio.com/downloads



.NET Core 給企業帶來什麼挑戰？



不容易擴充
不容易部署
不容易微服務化
不容易 Scale-Out



Windows

.NET Framework

單體式系統

大量部署困難
移轉困難

Azure
DevOps

docker

高效能

FreeBSD[®]

fedora^f

容器化



Container

Microservices



Cloud



.NET Core

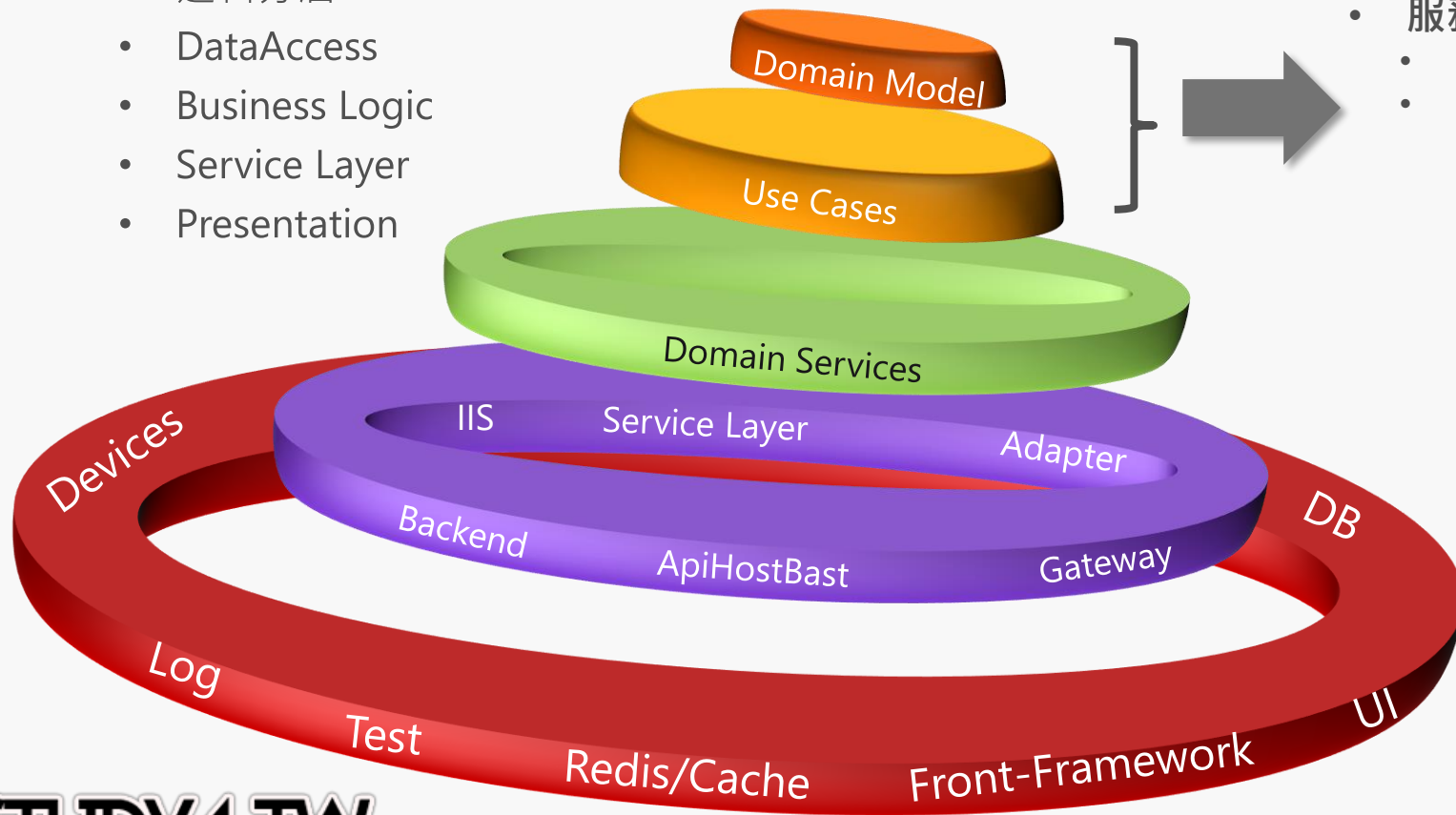
如何突破困境？

常見的軟體架構設計指南..

- Plug-in Architecture
 - Clean Architecture
- N-Layered Architecture (多層式架構)
 - N-Layerd .NET 4.0 Microsoft Architecture Guide
- Hexagonal Architecture (六角架構)
 - 由 Alistair Cockburn 於 2005 年提出
- Onion Architecture (洋蔥式架構)
 - 由 [Jeffrey Palermo](#) 於 2008 年提出
- MVC (Model View Controller)
- ...

Clean Architecture : 水平切割系統

- 與前進微服務相同手法
 - 分層 (水平切割)
 - 邏輯分層
 - DataAccess
 - Business Logic
 - Service Layer
 - Presentation



- 服務類型切割 (縱向切割)
 - 從 (服務 / 需求 / 類型) 角度來思考、可獨立運作的個體
 - 透過 Use Case 找出系統的邊界、挑出適合切割出獨立的服务區塊

但是.. 這跟移轉有什麼關係？

分層

?

?

?

?

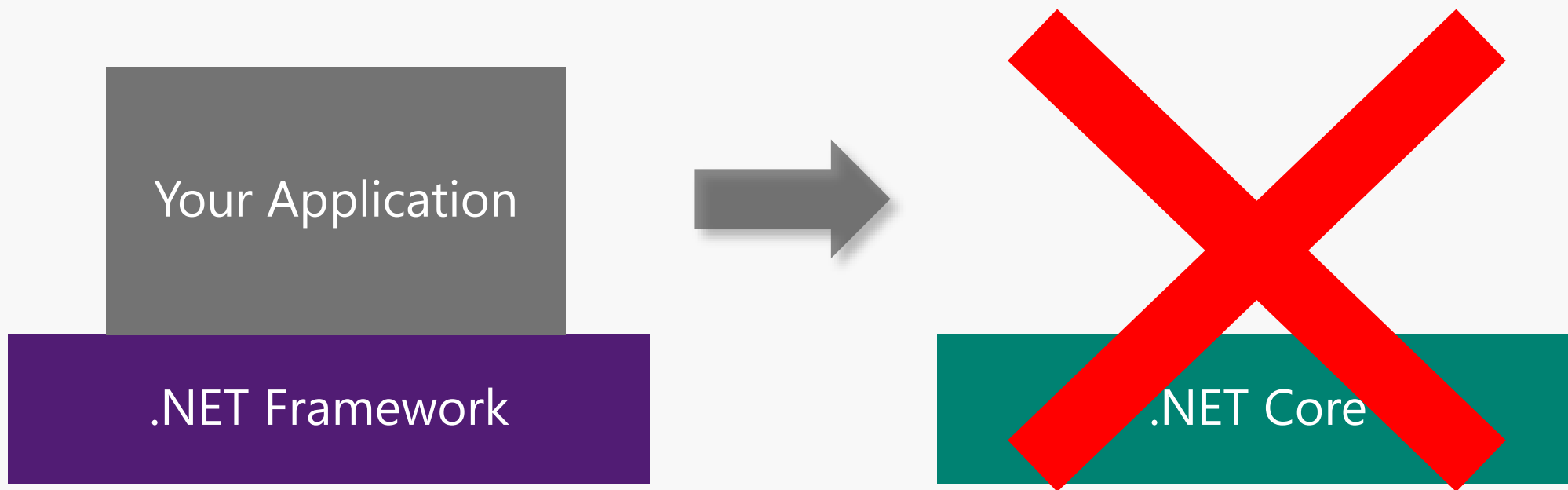
➡ 我們一直 Refactor

重構無法解決所有的問題

但重構仍然是你平常就應該要做的事情

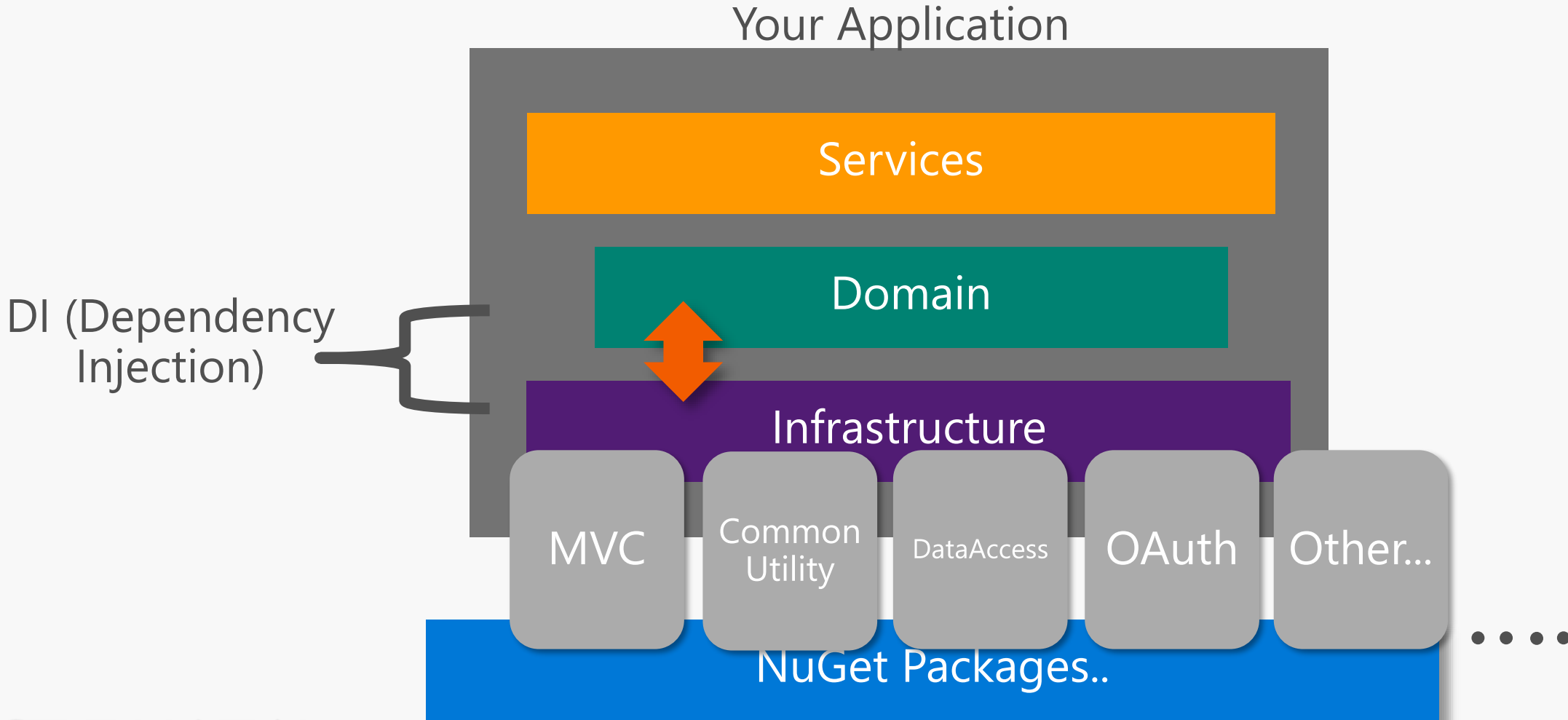
因為重構可能無法解決平台的問題

重構可能還是無法解決平台的問題



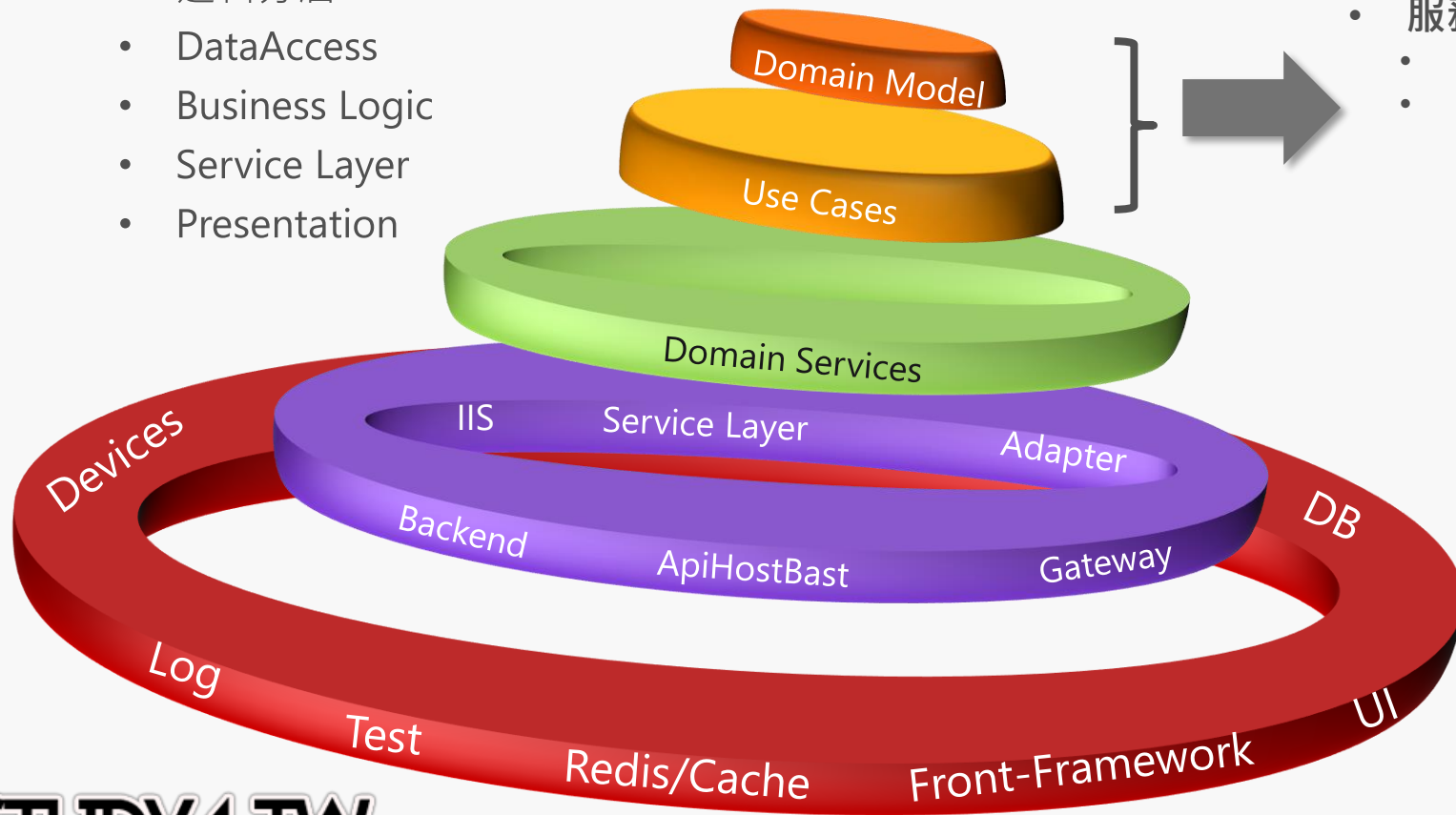
Your Application

- Infrastructure as Services
- Configuration as Services
- Domain as Services



Clean Architecture : 水平切割系統

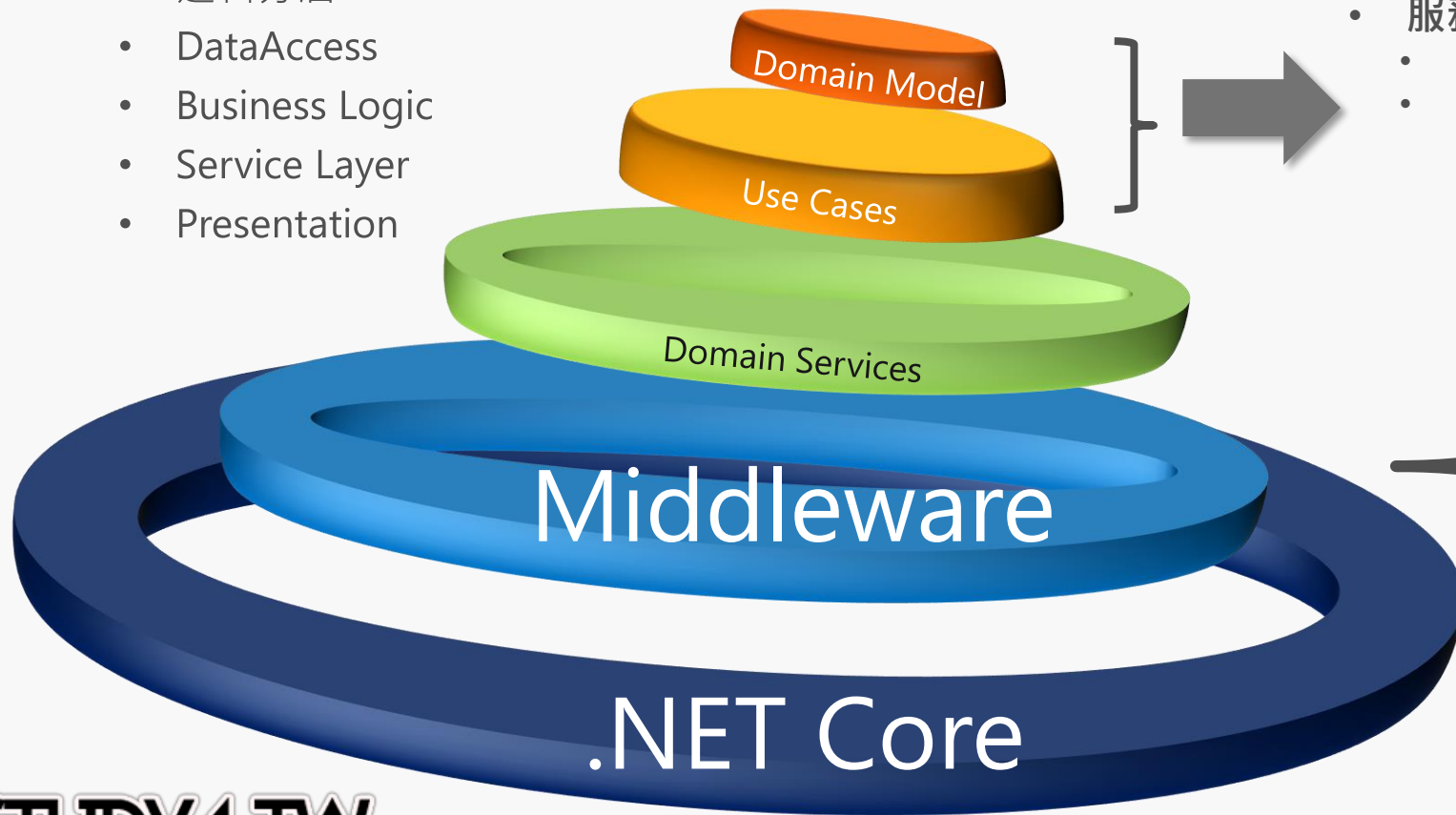
- 與前進微服務相同手法
 - 分層 (水平切割)
 - 邏輯分層
 - DataAccess
 - Business Logic
 - Service Layer
 - Presentation



- **服務類型切割 (縱向切割)**
 - 從 (服務 / 需求 / 類型) 角度來思考、可獨立運作的個體
 - 透過 Use Case 找出系統的邊界、挑出適合切割出獨立的服务區塊

Clean Architecture：水平切割系統

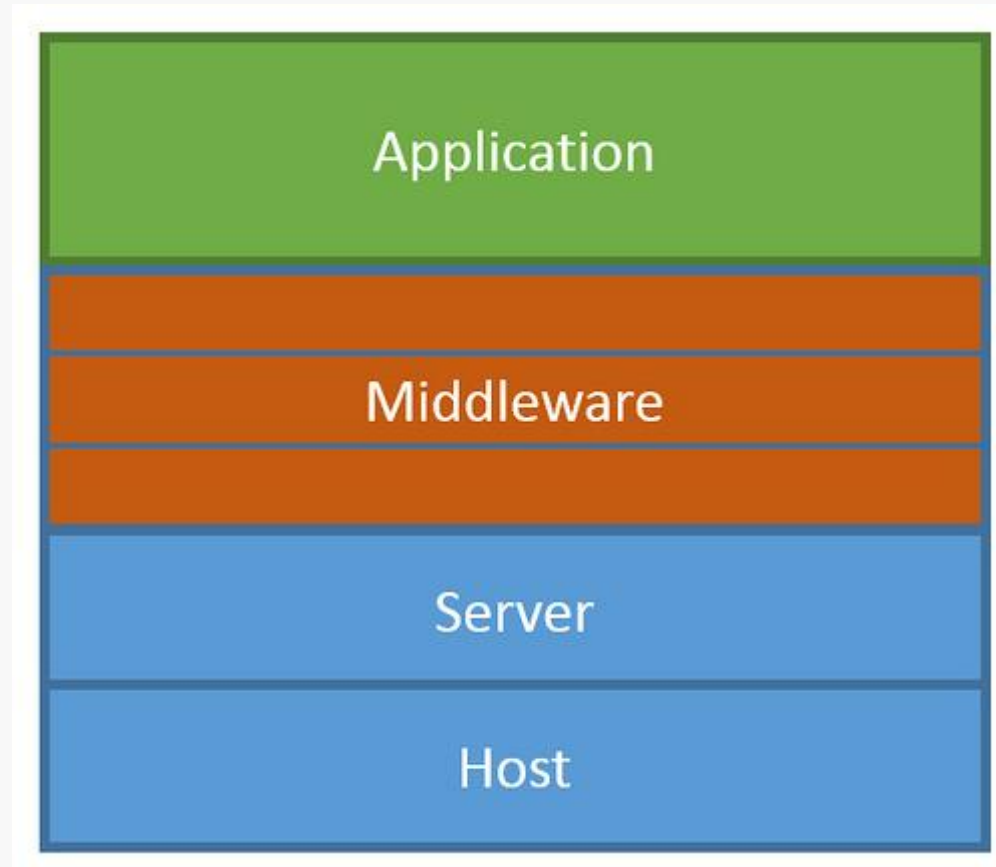
- 與前進微服務相同手法
 - 分層 (水平切割)
 - 邏輯分層
 - DataAccess
 - Business Logic
 - Service Layer
 - Presentation



- 服務類型切割 (縱向切割)
 - 從 (服務 / 需求 / 類型) 角度來思考、可獨立運作的個體
 - 透過 Use Case 找出系統的邊界、挑出適合切割出獨立的服务區塊

- DI (真正的隔離相依性)
 - 高內聚、低耦合
 - Infrastructure as Services

ASP.NET Core Middleware



所以..

企業最大的挑戰還是在...

無數的 高度耦合(未寫測試)的遺留系統、
不良的架構就像手銬一樣
【限制】住業務量發展

甚至限制住公司的發展

因為升級替換成本非常巨大

更不用說 Migration to .NET Core !



國內許多大企業現狀

即便有分層、但沒有撰寫測試...

移轉平台需要測試 + 整合測試

甚至沒有足夠的可維護性、沒有落實 CI

平常就沒撰寫測試、你跟我說要轉換平台？

相關 Migration 方法 / 工具

- .NET 可攜性分析工具 - .NET Portability Analyzer
- .NET API 分析器



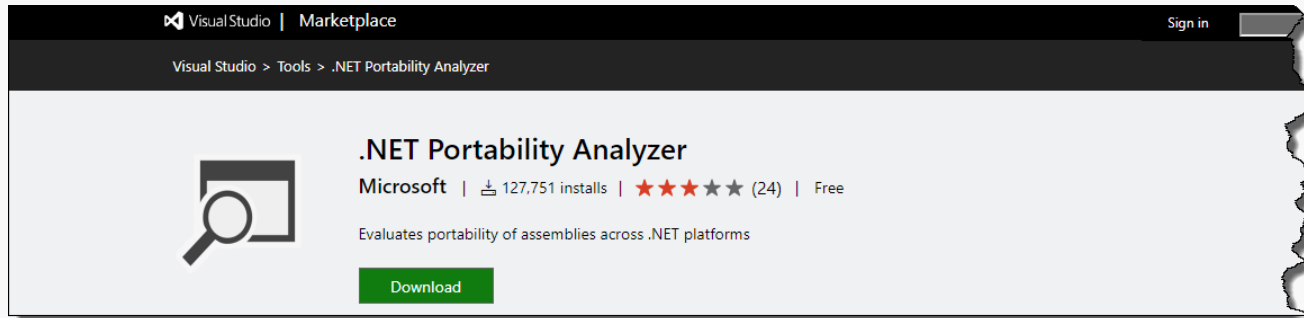
移轉步驟

- 官方：<https://docs.microsoft.com/zh-tw/aspnet/core/migration/proper-to-2x/?view=aspnetcore-3.0>)
 - 目標 Framework
 - 專案檔結構差異
 - 以 Startup 取代 Global.asax
 - 組態設定 Configuration (JSON)
 - Middleware (Startup, DI)
 - 靜態檔案
 - wwwroot
 - CSS
 - Javascript
 - Bower
 - Webpack
 - ...

.NET 可攜性分析工具 .NET Portability Analyzer

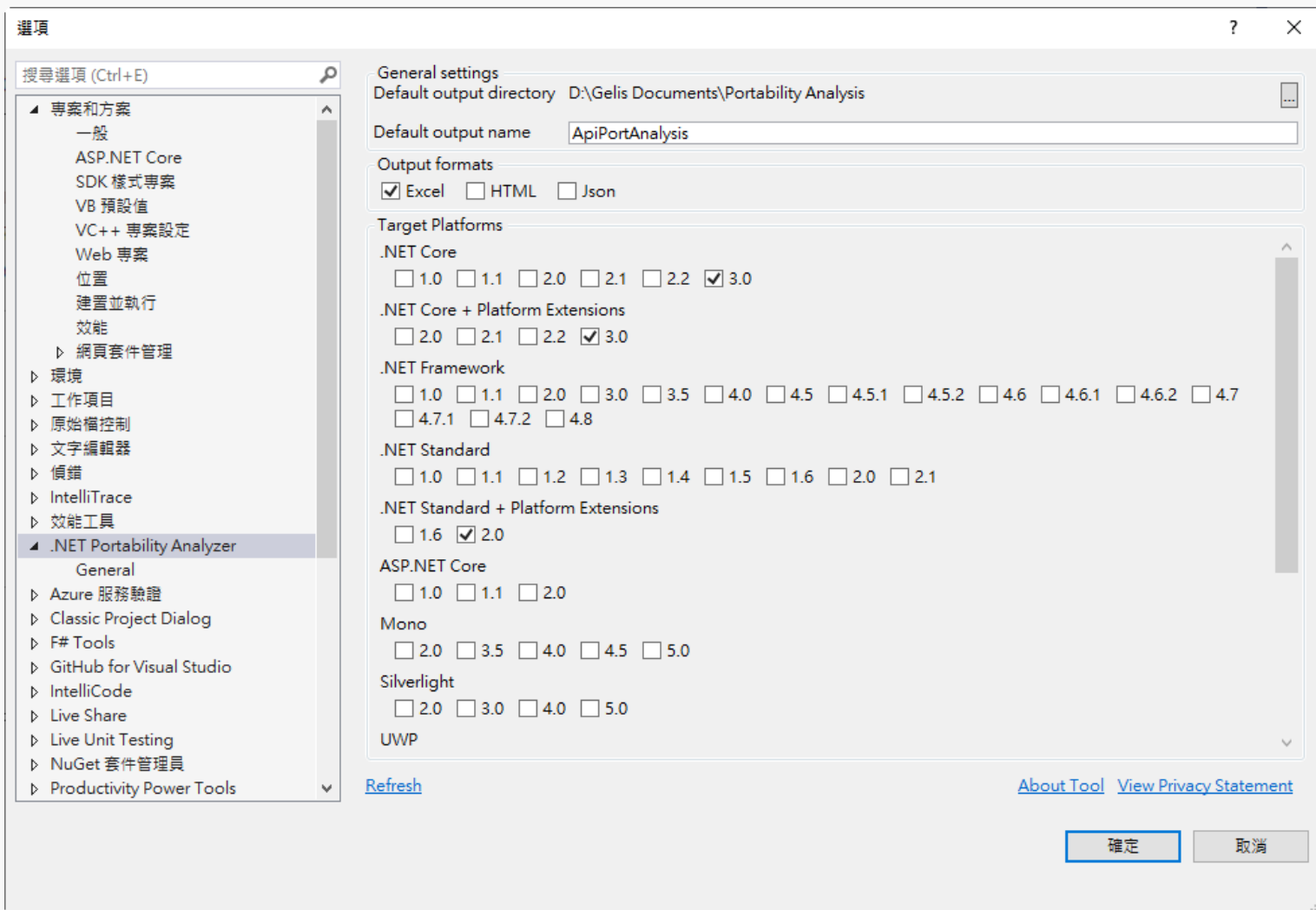
現有程式碼分析(.NET 可攜性分析工具) .NET Portability Analyzer

- .NET Portability Analyzer (Visual Studio extension)
- <https://marketplace.visualstudio.com/items?itemName=ConnieYau.NETPortabilityAnalyzer>



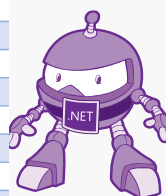
- .NET Portability Analyzer (dotnet CLI)
- <https://github.com/microsoft/dotnet-apiport>

現有程式碼分析(.NET 可攜性分析工具) .NET Portability Analyzer



現有程式碼分析(.NET 可攜性分析工具) .NET Portability Analyzer

Target type	Target member	Header for assembly name entries	.NET Core + F	.NET Core	.NET Framework, Version=v4.6.1	.NET Framework, Version=v4.6	.NET Standard + Platform	Recon
T:System.Configuration.ConfigurationManager	M:System.Configuration.ConfigurationManager	WebRedis TestApp1	Supported: 2.0+	Not supported	Supported: 2.0+	Supported: 2.0+	Supported: 2.0+	Use cc
T:System.Web.Mvc.Controller	T:System.Web.Mvc.Controller	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.Controller	M:System.Web.Mvc.Controller.#ctor	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.Controller	M:System.Web.Mvc.Controller.get_HttpContext	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.Controller	M:System.Web.Mvc.Controller.Json(System.Object)	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.Controller	M:System.Web.Mvc.Controller.View	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.Controller	M:System.Web.Mvc.Controller.View(System.Object)	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.RouteCollectionExtensions	T:System.Web.Mvc.RouteCollectionExtensions	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.RouteCollectionExtensions	M:System.Web.Mvc.RouteCollectionExtensions	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.RouteCollectionExtensions	M:System.Web.Mvc.RouteCollectionExtensions	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Net.Http.Formatting.MediaTypeFormatter	T:System.Net.Http.Formatting.MediaTypeFormatter	WebRedis TestApp1	Supported: 2.1+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:System.Net.Http.Formatting.MediaTypeFormatter	M:System.Net.Http.Formatting.MediaTypeFormatter	WebRedis TestApp1	Supported: 2.1+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:System.Net.Http.Formatting.MediaTypeFormatter	M:System.Net.Http.Formatting.MediaTypeFormatter	WebRedis TestApp1	Supported: 2.1+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:System.Web.Mvc.ViewResult	T:System.Web.Mvc.ViewResult	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Net.Http.Formatting.XmlMediaTypeFormatter	T:System.Net.Http.Formatting.XmlMediaTypeFormatter	WebRedis TestApp1	Supported: 2.1+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:System.Web.Mvc.AreaRegistration	T:System.Web.Mvc.AreaRegistration	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.AreaRegistration	M:System.Web.Mvc.AreaRegistration.RegisterArea	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.HttpContextBase	T:System.Web.HttpContextBase	WebRedis TestApp1	Not supported	Not supported	Supported: 3.5+	Supported: 3.5+	Not supported	
T:System.Web.HttpContextBase	M:System.Web.HttpContextBase.get_Application	WebRedis TestApp1	Not supported	Not supported	Supported: 3.5+	Supported: 3.5+	Not supported	
T:System.Web.Http.Routing.IHttpRoute	T:System.Web.Http.Routing.IHttpRoute	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:Newtonsoft.Json.JsonSerializerSettings	T:Newtonsoft.Json.JsonSerializerSettings	WebRedis TestApp1	Supported: 2.0+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:Newtonsoft.Json.JsonSerializerSettings	M:Newtonsoft.Json.JsonSerializerSettings.#ctor	WebRedis TestApp1	Supported: 2.0+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:Newtonsoft.Json.JsonSerializerSettings	M:Newtonsoft.Json.JsonSerializerSettings.set_Cc	WebRedis TestApp1	Supported: 2.0+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:Newtonsoft.Json.JsonSerializerSettings	M:Newtonsoft.Json.JsonSerializerSettings.set_Tv	WebRedis TestApp1	Supported: 2.0+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:System.Web.Http.RouteCollectionExtensions	T:System.Web.Http.RouteCollectionExtensions	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Http.RouteCollectionExtensions	M:System.Web.Http.RouteCollectionExtensions	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver	T:Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver	WebRedis TestApp1	Supported: 2.0+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver	M:Newtonsoft.Json.Serialization.CamelCasePropertyNamesContractResolver	WebRedis TestApp1	Supported: 2.0+	Not supported	Not supported	Not supported	Supported: 1.6+	
T:System.Runtime.Serialization.NetDataContractResolver	T:System.Runtime.Serialization.NetDataContractResolver	WebRedis TestApp1	Not supported	Not supported	Supported: 3.0+	Supported: 3.0+	Not supported	
T:System.Runtime.Serialization.NetDataContractResolver	M:System.Runtime.Serialization.NetDataContractResolver	WebRedis TestApp1	Not supported	Not supported	Supported: 3.0+	Supported: 3.0+	Not supported	
T:System.Runtime.Serialization.NetDataContractResolver	M:System.Runtime.Serialization.NetDataContractResolver	WebRedis TestApp1	Not supported	Not supported	Supported: 3.0+	Supported: 3.0+	Not supported	
T:System.Web.Mvc.FormCollection	T:System.Web.Mvc.FormCollection	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Mvc.JsonRequestBehavior	T:System.Web.Mvc.JsonRequestBehavior	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Http.GlobalConfiguration	T:System.Web.Http.GlobalConfiguration	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	
T:System.Web.Http.GlobalConfiguration	M:System.Web.Http.GlobalConfiguration.get_C	WebRedis TestApp1	Not supported	Not supported	Not supported	Not supported	Not supported	



Demo : .NET Portability Analyzer



現有程式碼分析(.NET 可攜性分析工具) .NET Portability Analyzer

- 也可使用 ApiPort CLI

```
D:\Temp\WebMvc5TestApp1\bn\Debug>ApiPort.exe analyze -f WebMvc5TestApp1.dll -t ".NET Framework, Version=4.6.1" -t ".NET Standard, Version=2.0" -r HTML -o AnalysisReport.html
```

Demo : ApiPort CLI



使用 .NET API 分析器



如何使用 .NET API 分析器？

- 「API 分析器」是以 NuGet 套件 `Microsoft.DotNet.Analyzers.Compatibility` 的形式提供
- 探索已被取代的 API
- 由 Roslyn Compiler 所提供的功能
- 必須先 Compiler 後才可使用
- 先前稱為 Windows Compatibility Pack

Windows Compatibility Pack 安裝順序

- 先安裝 `Microsoft.Windows.Compatibility`

```
Install-Package Microsoft.Windows.Compatibility -Version 3.1.0-preview1.19504.10
```

- 再安裝 `Microsoft.DotNet.Analyzers.Compatibility`

```
Install-Package Microsoft.DotNet.Analyzers.Compatibility -Version 0.2.12-alpha
```

`Microsoft.DotNet.Analyzers.Compatibility` 目前還在 alpha 階段



Demo : .NET API 分析器

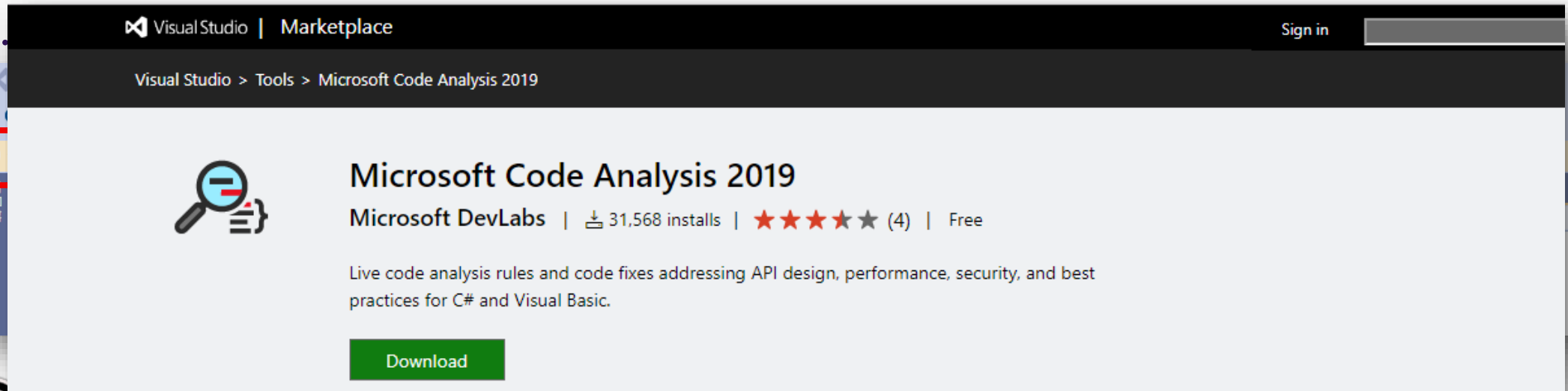


使用架構分析器 .NET Framework Analyzer




使用架構分析器 .NET Framework Analyzer

- Microsoft.CodeQuality.Analyzers
- Microsoft.CodeAnalysis.Analyzers
- Microsoft.AspNetCore.Mvc.Analyzers
- Microsoft.CodeAnalysis.FxCopAnalyzers



Visual Studio | Marketplace Sign in

Visual Studio > Tools > Microsoft Code Analysis 2019

 **Microsoft Code Analysis 2019**

Microsoft DevLabs | 31,568 installs | ★★★★★ (4) | Free

Live code analysis rules and code fixes addressing API design, performance, security, and best practices for C# and Visual Basic.

[Download](#)

使用新版的程式碼分析器

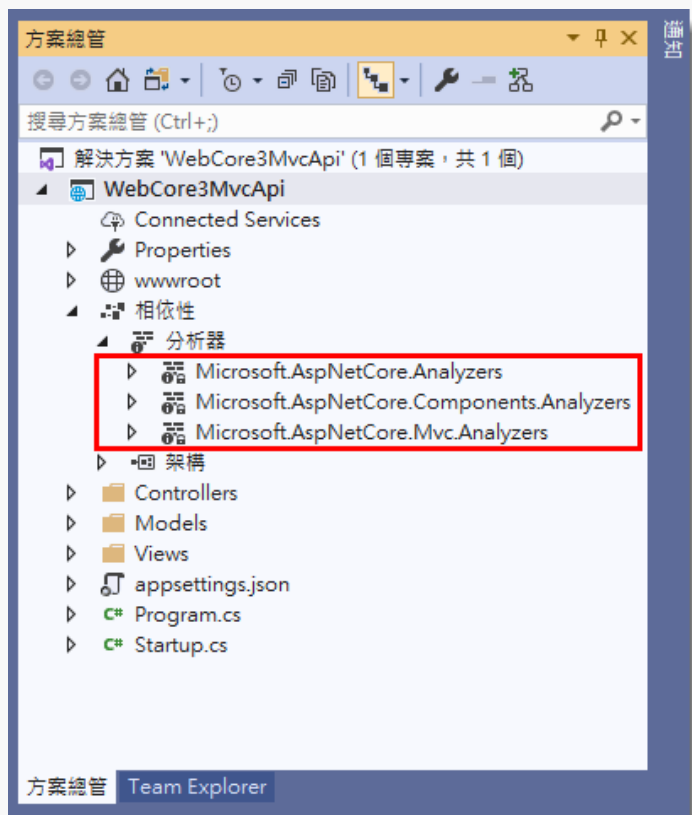
- Roslyn 分析器的來源分析會取代 managed 程式碼的舊版分析, 像是 FxCop 靜態程式碼分析
 - 舊版分析規則會慢慢改寫為 Roslyn 程式碼分析器
- 舊版分析可能無法使用在 .NET Core 或 .NET Standard 等類型專案上
- 注意：
 - VS 功能表上的【分析】 → 【執行程式碼分析】功能表選項仍然屬於舊版分析

Demo : Code Analysis 2019



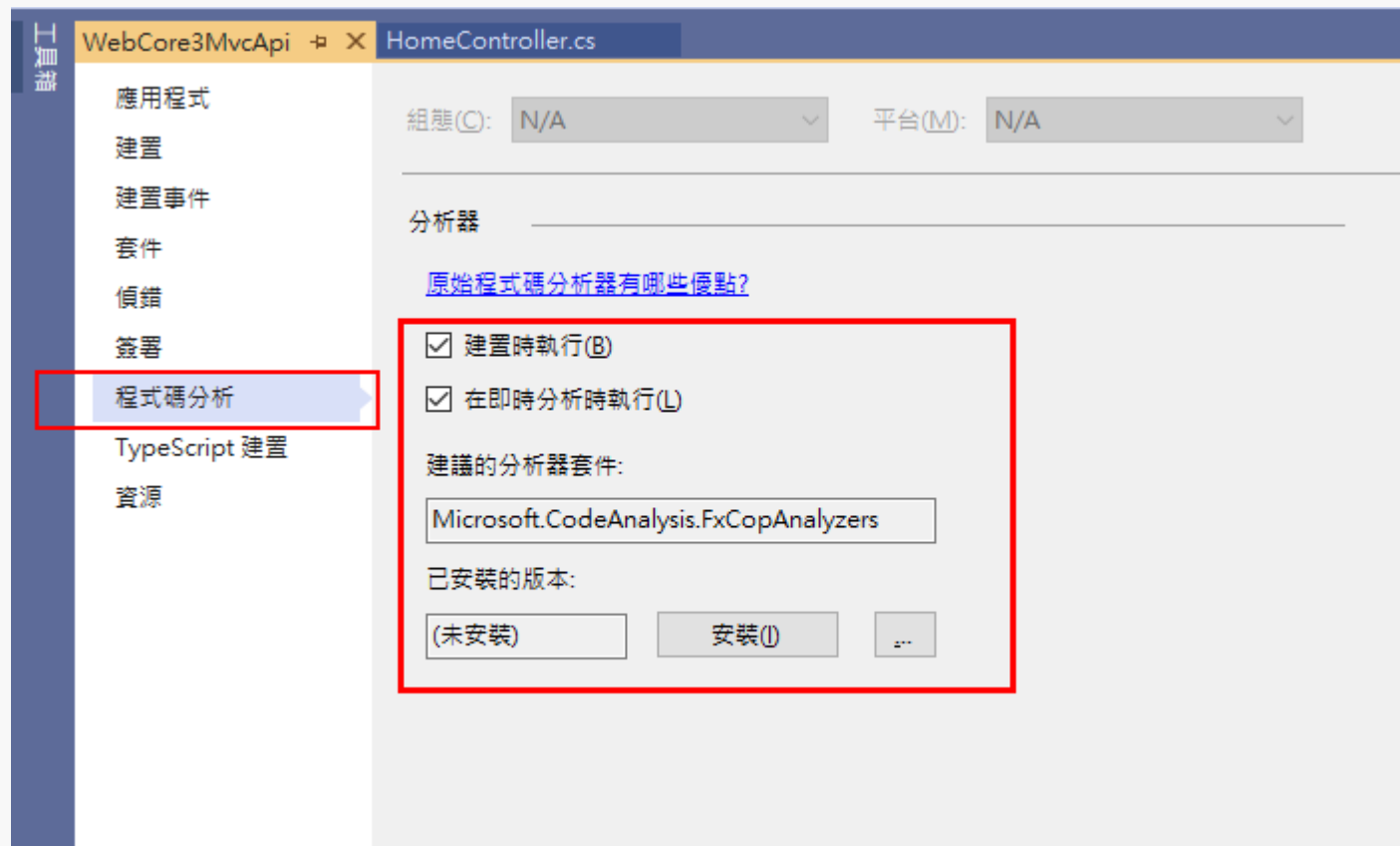
Visual Studio 2019 內建的分析器

- 分析器由 SDK 提供



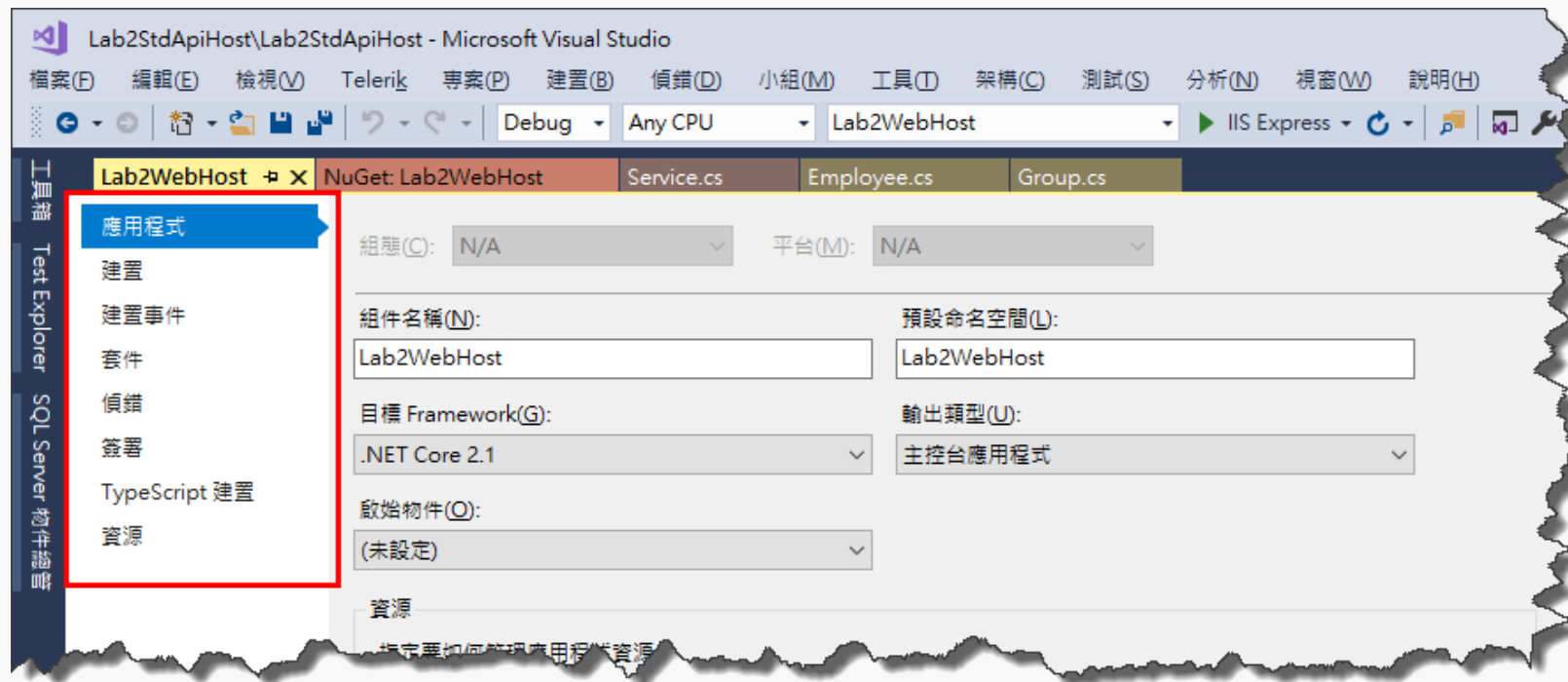
Visual Studio 2019 內建的分析器

- 也可由 NuGet 套件提供、並建置時分析



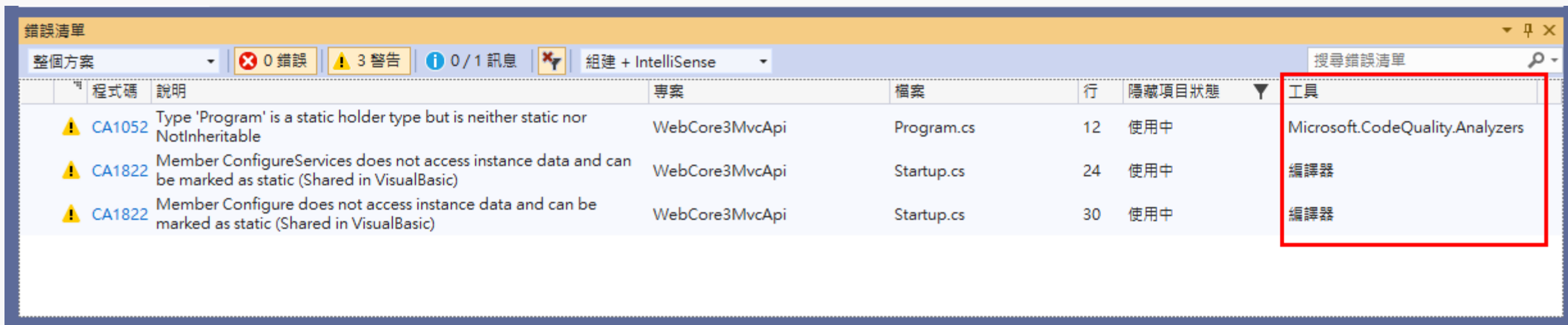
Visual Studio 2019 內建的分析器

- 也可由 NuGet 套件提供、並建置時分析
- 但注意 VS2017 沒有這個選項..



如何區分不同分析器來源？

- 錯誤清單中的『工具欄』可以區別



The screenshot shows the Visual Studio Error List window for a solution named '整個方案'. The status bar indicates 0 errors, 3 warnings, and 0/1 messages. The build configuration is '組建 + IntelliSense'. The error list contains three entries:

圖示	程式碼	說明	專案	檔案	行	隱藏項目狀態	工具
⚠	CA1052	Type 'Program' is a static holder type but is neither static nor NotInheritable	WebCore3MvcApi	Program.cs	12	使用中	Microsoft.CodeQuality.Analyzers
⚠	CA1822	Member ConfigureServices does not access instance data and can be marked as static (Shared in VisualBasic)	WebCore3MvcApi	Startup.cs	24	使用中	編譯器
⚠	CA1822	Member Configure does not access instance data and can be marked as static (Shared in VisualBasic)	WebCore3MvcApi	Startup.cs	30	使用中	編譯器

The '工具' (Tool) column is highlighted with a red box, showing the source of each error: 'Microsoft.CodeQuality.Analyzers' for CA1052 and '編譯器' (Compiler) for CA1822.

在專案內取消使用舊版分析器

- 專案檔中的 RunCodeAnalysis 設定為 false 就可避免在專案中使用舊版分析

```
.csproj ( 專案檔 )
```

```
<RunCodeAnalysis>>false</RunCodeAnalysis>
```


但？我要用 Roslyn 的分析器之前
我還是要先轉成 .NET Core 不是嗎？.....

移轉的實例：

使用自行開發 VS Extension & CLI 工具移轉

ASP.NET Web API



Manage Extensions

已安裝的

全部
控制項
範本
SDK
工具

線上
更新
漫遊擴充管理員

排序依據: 最新

搜尋 (Ctrl+E)

ConvertMVC5ToCore3
Convert ASP.MVC5 to ASP.NET Core 3.0
Preview 1 - (.NET Conf 2019)

停用(D)
解除安裝(U)

建立者: Gelis
安裝日期: 2019/11/7
版本: 1.0

動更新此延伸模組

SpecFlow for Visual Studio
SpecFlow integration for Visual Studio
TechTalk (http://www.techtalk.com)

GitHub Extension for Visual Studio
A Visual Studio Extension that brings the GitHub Flow into Visual Studio.

適用於 Visual Studio 的 Microsoft 報表專案
用於建立專業報表的 Microsoft RDL 報表設計工具、專案及精靈。此套件提供 .rptproj 類型的支援，而且專為最新版本的 Microsoft Rep...

Live Share
Real-time collaborative development from the comfort of your favorite tools.

StyleCop Check-in Policy 2019
StyleCop Checkin Policy for Visual Studio 2019.
This way, the StyleCop warnings don't have to be set as error dur...

Microsoft Code Analysis 2019 [已停用]
Live code analysis rules and code fixes addressing API design, performance, security, and best practices for C# and Visual Basic.

.NET Portability Analyzer
Evaluates portability of assemblies across .NET platforms

已排程進行安裝:
無

已排程進行更新:
無

已排程進行解除安裝:
無

Change your settings for Extensions

Close

特別感謝



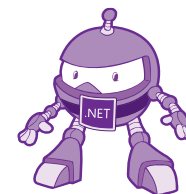
R-Ladies Taipei



多奇·數位創意



以及各位參與活動的你們



Q & A



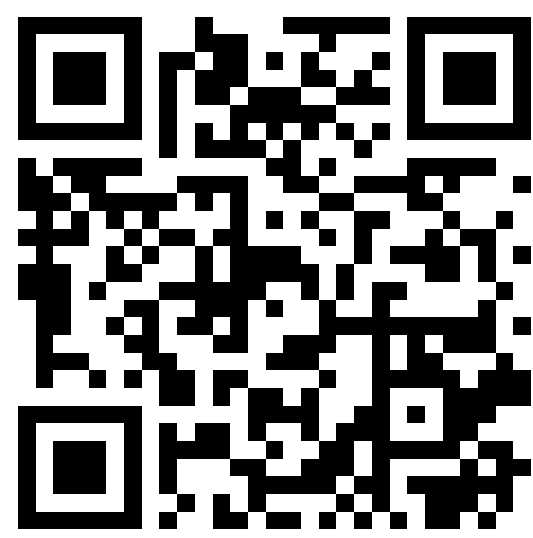
關於我



FB 社團
軟體開發之路



粉絲團
Gelis 程式設計訓練營



部落格
Gelis 技術隨筆-部落格