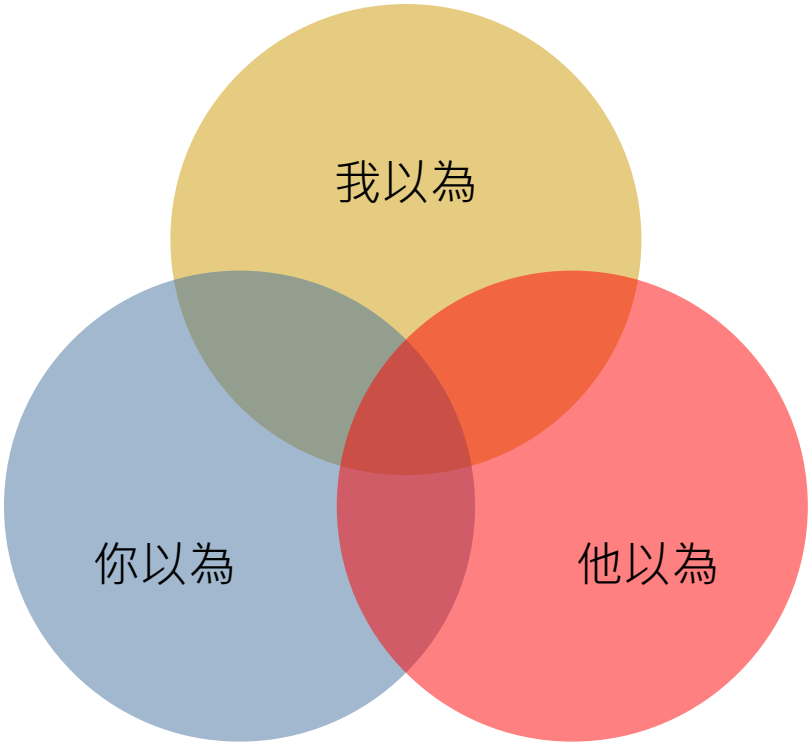
# 動手打造 Application Framework

kevin@kuicker.org

tw MVC

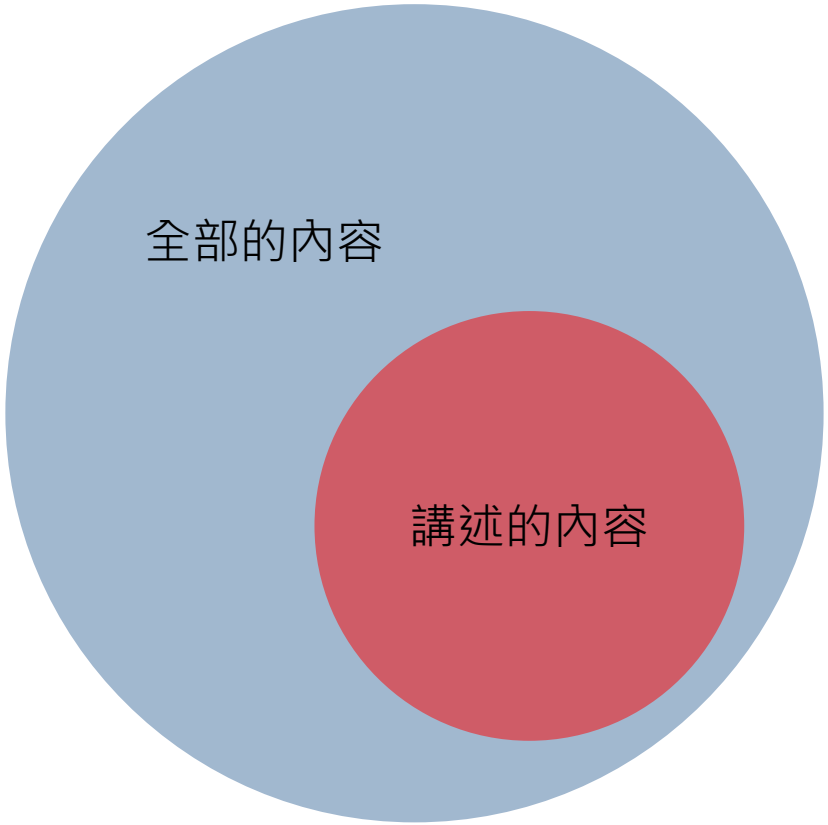
# Framework 集成

# What is the Application Framework?




- 每個人心裡想的都不一樣。
- 或許沒有一個真正的範圍！

# What I would like to say is..




- 有誰了解全部內容？
- 講我覺得重要的
- 講白話一點
- 70 內分鐘講得清楚的
- 個人的想法，與大家分享

# Why do we need Application Framework?

~~簡化~~

清晰

- 複雜的企業邏輯，不會因此變得簡單。
- Application Framework 帶來的是清晰，不是簡化。

# Library vs. Framework

| Library | Framework |
| --- | --- |
| 類別庫 | 集成 |
| You call Library. | Framework calls you. |
| Your project's Tools. | Your project's Life. |
| 一組包含在類別裡能被程式呼叫的方法，每次的呼叫將完成特定工作並回傳控制。 | 包含抽象設計與內建行為，程式需要透過子類別或是插入程式的方式，在執行期特定的時點呼叫你的程式。 |

放下既有的技術定義，先用簡單的方法了解。

# Lifecycle 系統生命周期

# Lifecycle : Running




- 執行中

專案程式執行在 Application Framework 建立的服務之上

# Lifecycle : Action



Start

Running



Stop

- 狀態

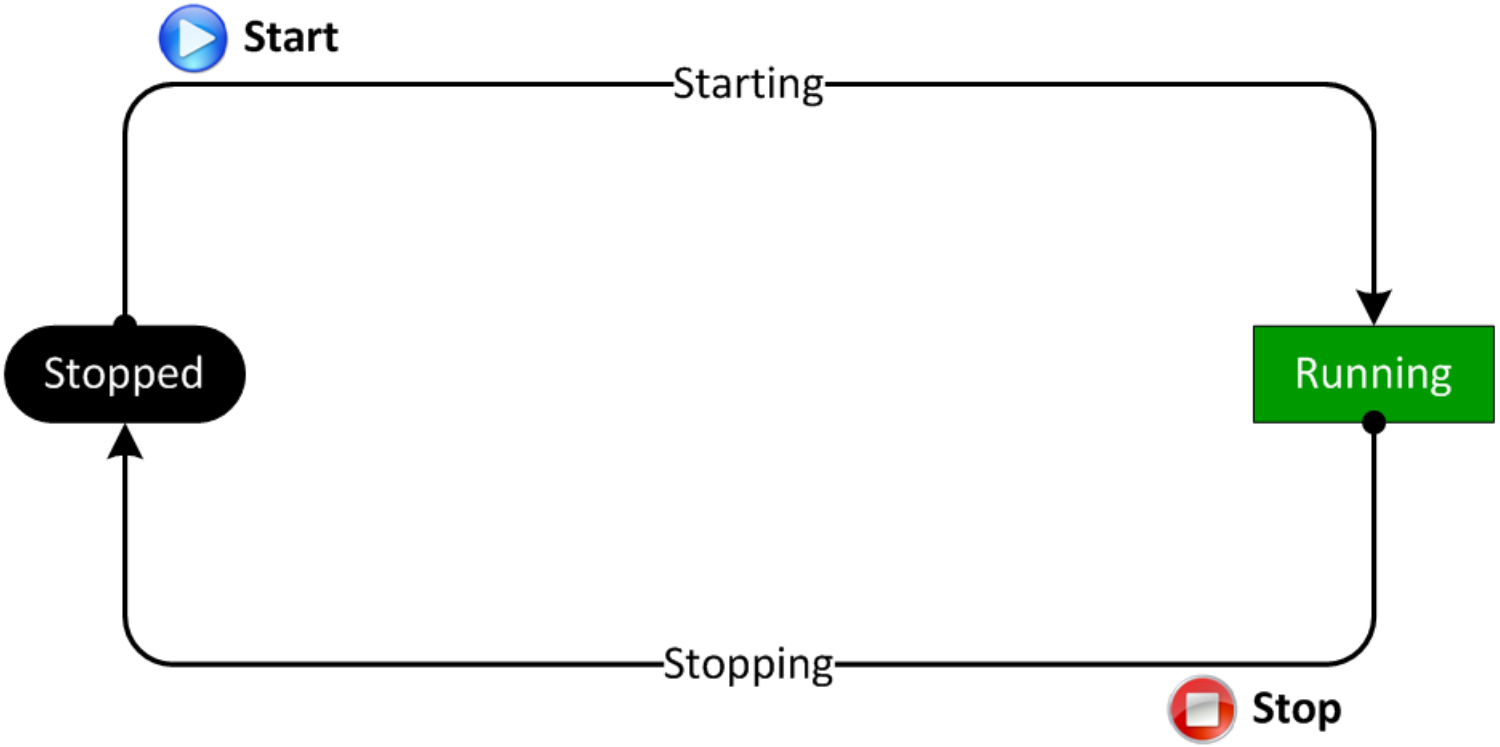
  Running

- 動作

  啟動 Start

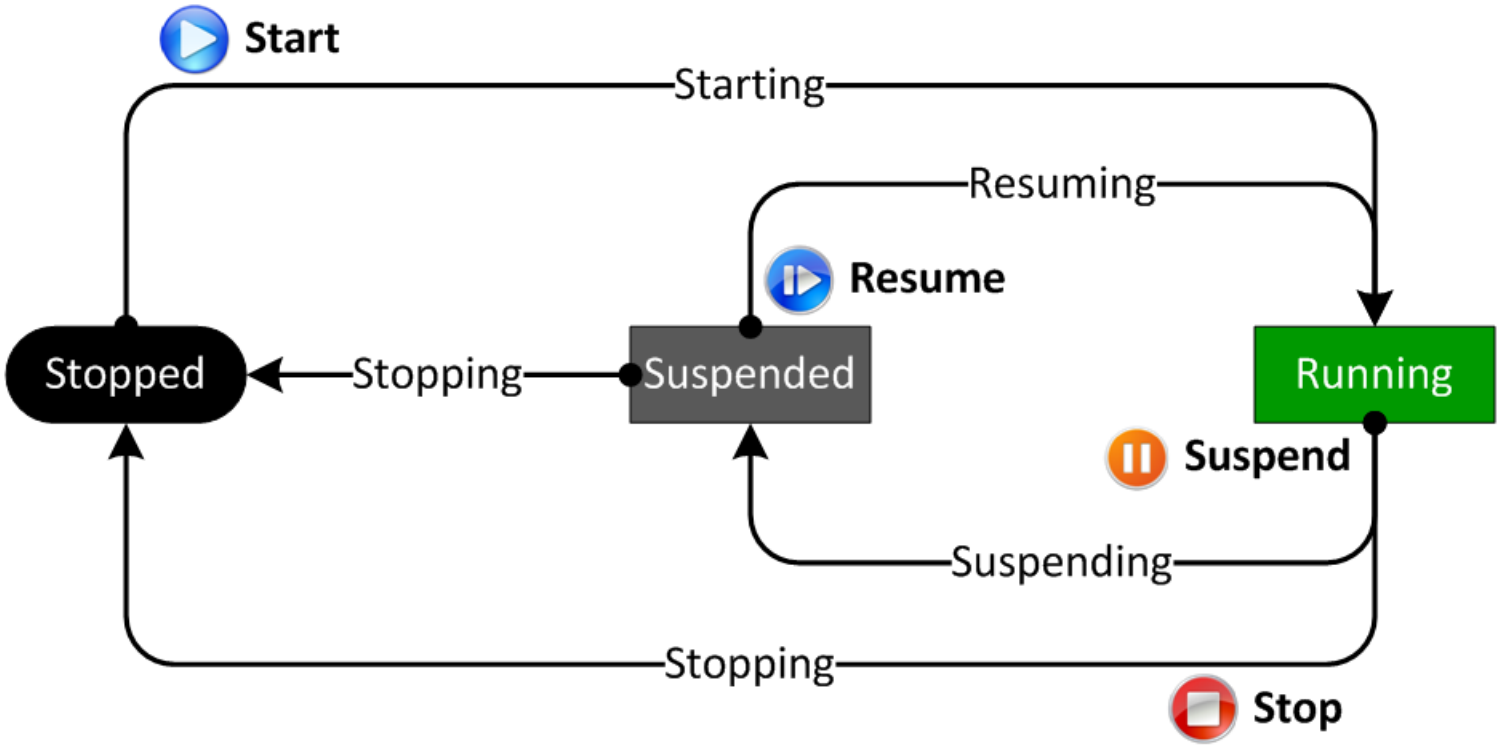
  停止 Stop

# Lifecycle : Circle




生命周期形成封閉的迴圈

# Lifecycle : Full Circle




完整的生命周期

# Lifecycle : State

- 穩定狀態

  Permanent State

  - 已停止 Stopped
  - 執行中 Running
  - 已暫停 Suspended

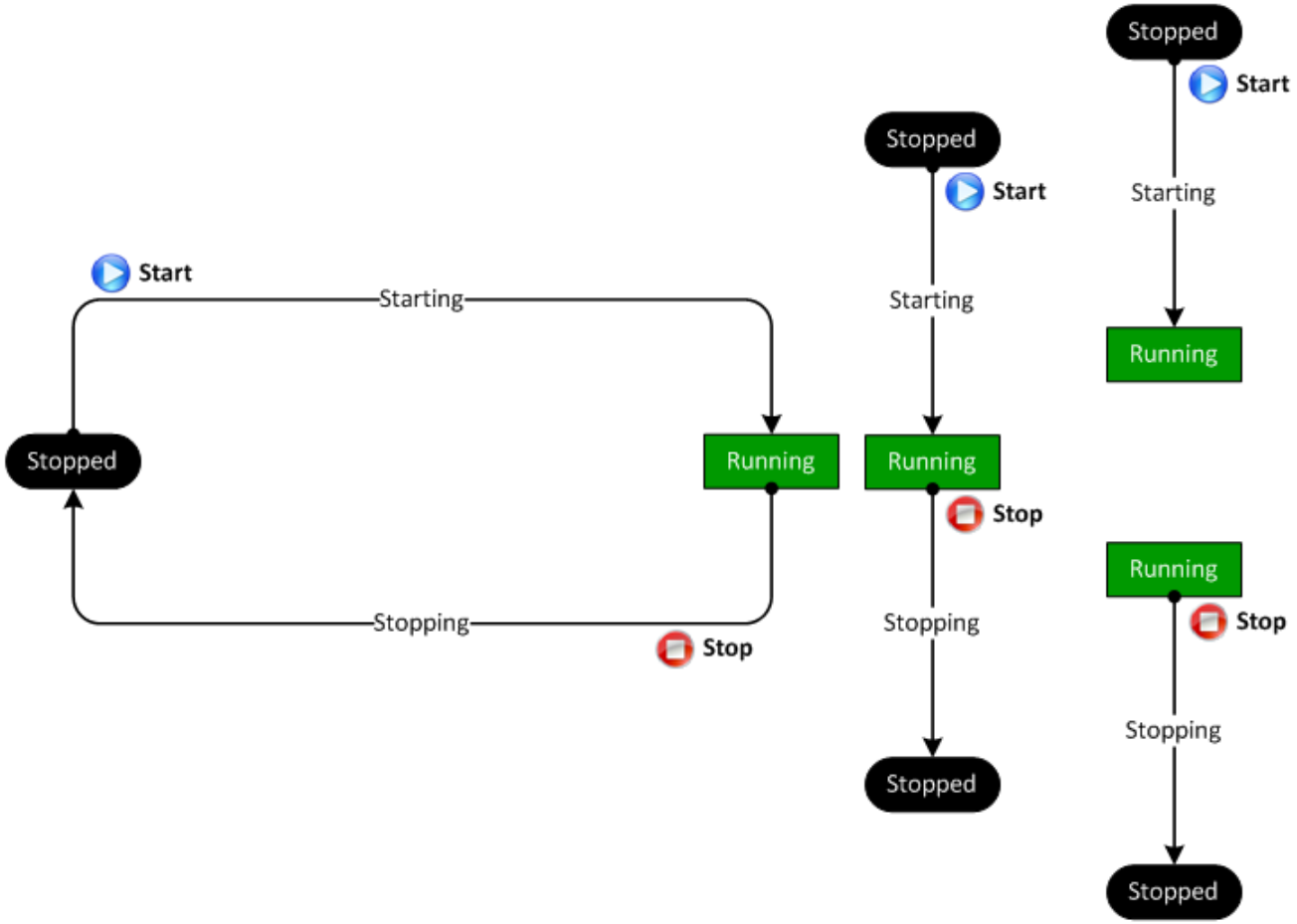
- 過渡狀態

  Transition State

  - 啟動中 Starting
  - 停止中 Stopping
  - 恢復中 Resuming
  - 暫停中 Suspending

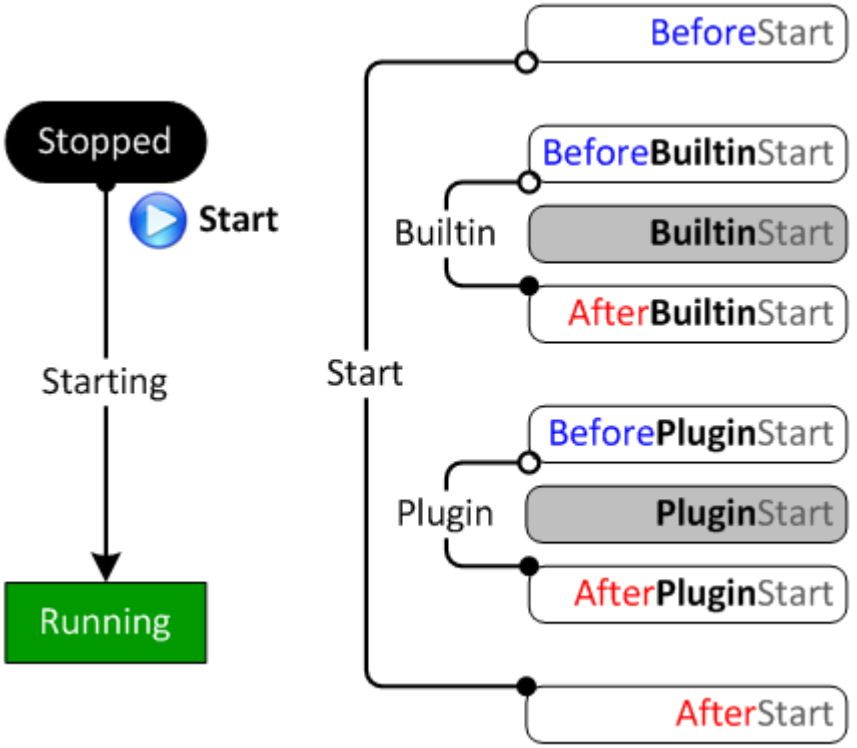
生命周期就是，系統一連串狀態的改變與處理

# Lifecycle : Sequence




由封閉的迴圈，改從循序的角度了解系統狀態的變化

# Lifecycle : Start

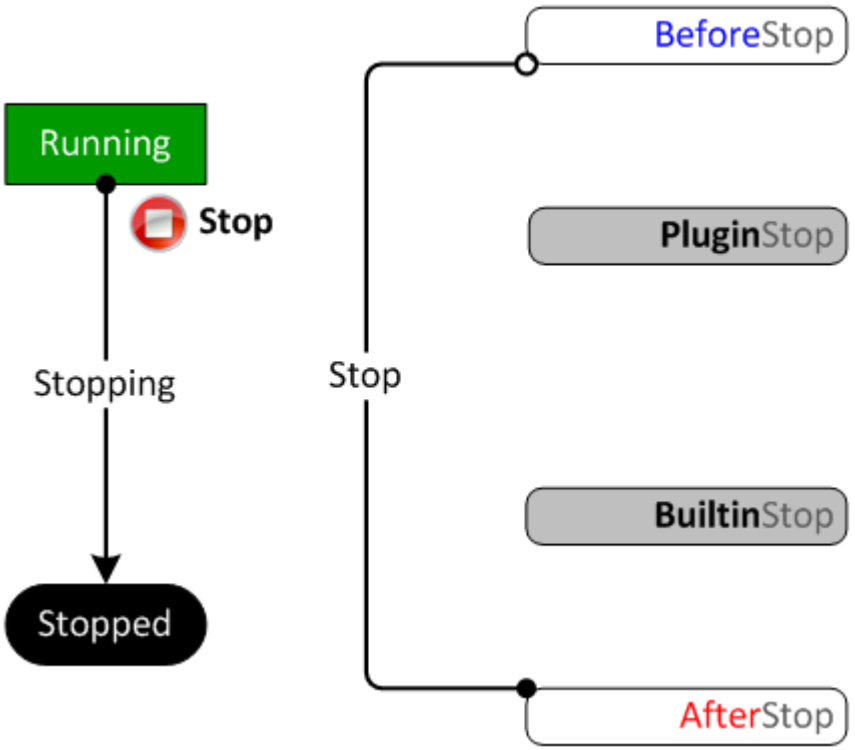



- 內建 Builtin
- 外掛 Plugin

Interceptable

Kernel Only

# Lifecycle : Stop




反序停止 Plugin, Builtin

Interceptable

Kernel Only

# Builtin & Plugin

# Builtin 內建

- 已知的被動服務
- Framework 提供每一個內建服務介面，定義服務內容
- 專案程式主動使用內建功能

- 常以靜態類別與方法提供，例如：Auditor.Success(..)
- 實作組件是否載入，不影響專案運作

# Plugin 外掛

- 未知的主動服務
- Framework 提供外掛統一介面，定義管控方式
- 專案程式被動享受外掛服務

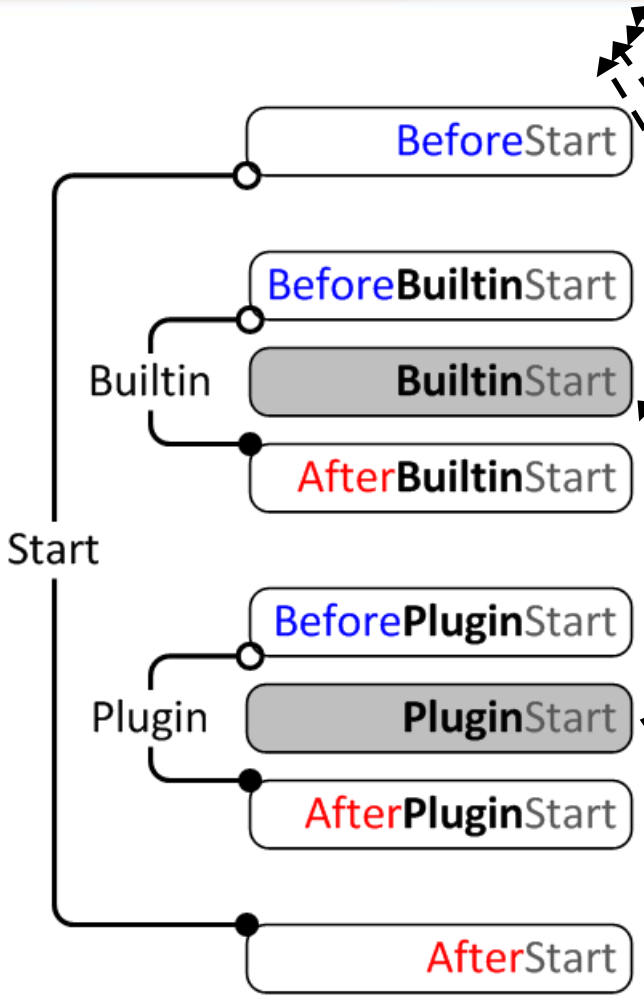
- 另開執行緒，或執行於獨有的 AppDomain 裡
- 實作組件是否載入，不影響專案運作

# Builtin vs. Plugin

| Builtin | Plugin |
| --- | --- |
| 已知的 | 未知的 |
| 被動的 | 主動的 |
| 介面定義服務內容 | 介面定義管控方式 |
| 不同介面 | 共同介面 |
| 以靜態類別與方法提供 | 執行緒或獨有的 AppDomain |
| 例如：Audit 稽核 | 例如：AD 帳號同步程式 |

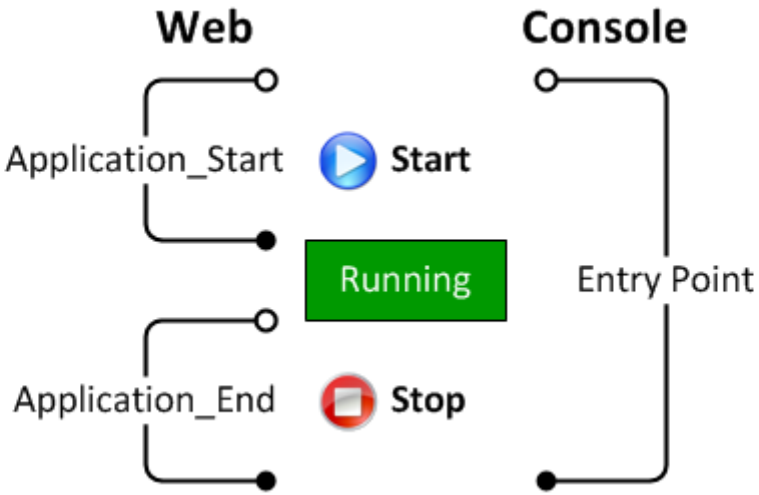
# Kuicker

# Kuicker : Execution Order




- Hosting & Kernel
- 讀取定義檔 Configurable
- 收集組件 Assembly
- 解析類別 Reflection
- 附加生命周期處理 Event
- 載入內建服務 Log, Audit
- 載入外掛服務 Pluginable

# Kuicker : Hosting & Kernel




```
Public class MvcApplication
  : HttpApplication {
  protected void Application_Start() {
    Platform.Start();
  }

  protected void Application_End() {
    Platform.Stop();
  }
}
```

```
class Program {
  static void Main(string[] args) {
    using(Kernel k = Kernel.Current) {
      // code here
    }
  }
}
```

# Kuicker : Configurable

- **1. ConfigurationSectionHandler** (*deprecated*)

  System.Configuration.IConfigurationSectionHandler

- **2. ConfigurationSection**

  System.Configuration.ConfigurationSection

# Kuicker : Assembly

- AppDomain Assemblies

```
AppDomain.CurrentDomain.GetAssemblies()
```

- APP_CODE Assemblies

```
BuildManager.CodeAssemblies.OfType<Assembly>().ToArray()
```

- Folder
  - bin >> AppDomain Assemblies
  - {special folder} >>

```
Directory.GetFiles(path)
Assembly.LoadFrom(file)
```

# Kuicker ： 熱插拔的系統

- 如果外掛組件放在 bin 裡，調整 bin 的組件將會觸發系統重啟，這是否就失去外掛的意義？

- 如果外掛組件放在特定目錄裡，讀取特定目錄裡的組件會有額外授權需求，這不利於一般化的部署。

- 我們真的需要具有熱插拔的系統嗎？

# Kuicker : Reflection

- 制定 Builtin 服務內容與 Plugin 管控方式並定義介面
- Kernel 主動收集組件裡的實作特定介面的類別
- 內建外掛執行順序採 FIFO，生命周期附加程序以核心優先
- 外掛錯誤處理，以不影響主程式為原則
- 類別提供 Self-Description 自我描述

# Kuicker : Event

- 核心事件
  - BuiltinStart, PluginStart
  - PluginStop, BuiltinStop
- 非核心事件
  - BeforeStart, AfterStop
  - BeforeBuiltinStart, AfterBuiltinStart
  - BeforePluginStart, AfterPluginStart
  - BeforeStop, AfterStop

- 實作方式
  - Event Handler
  - Type or Instance Level?
  - Attach / Detach or Implemented Interface?

# Kuicker : Log

- **Apache logging services**
  - log4j
  - log4net
  - log4php
  - log4cxx
  - Chainsaw
- **log4j Family**
  - log4plsql
  - log4js
  - log4c
  - log4javascript
  - log4perl

Log 機制的發展，逐漸向 log4j 收斂 (個人認為)

# Kuicker : Pluginable

- Send Message, Error Control, Application Domain
- Proxy, Gateway, Queue, Remoting, Socket
- Priority, Sleep
- Interrupt, Dependency
- Out-of-Process / In-Process
- Worker / Job, Load / Unload
- Synchronous / Asynchronous



demo

# Kuick >> Kuicker

# 好活動需要支持




感謝 KKTIX 贊助 twMVC 活動報名平台

# 好課程需要支持




http://skilltree.my

# 謝謝各位

http://mvc.tw


- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用姓名標示-非商業性-相同方式分享 3.0 台灣授權。閱讀本授權條款，請到 http://creativecommons.org/licenses/by-nc-sa/3.0/tw/，或寫信至Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.