

# .NET Conf

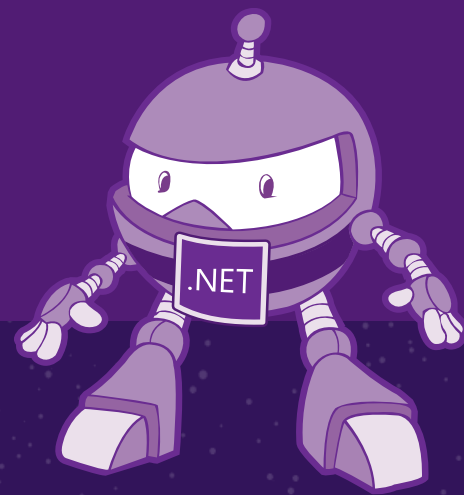
探索 .NET 新世界



Host by  
**STUDY4**

# 千年傳統 全新感受 - Rx 流

Kevin Yang



# 原由

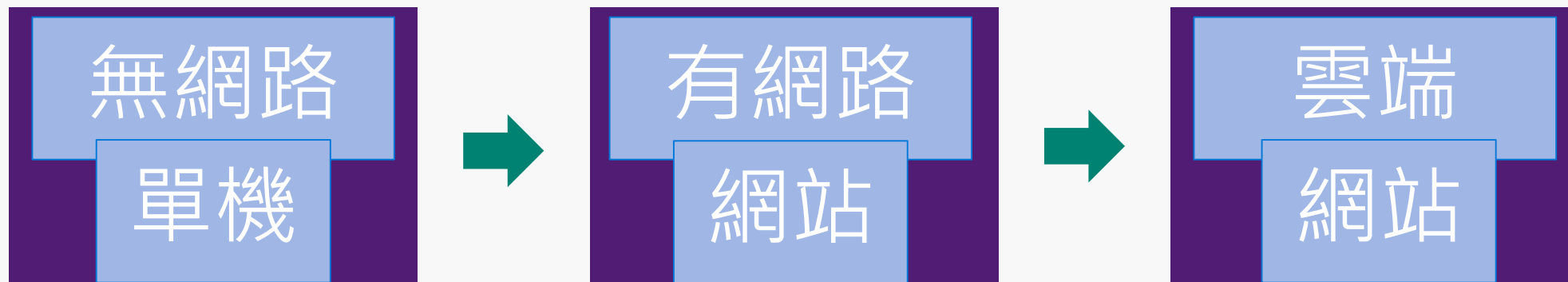


# 過往與現在

- 作業環境
- 使用人數
- 面臨考驗

# 過往與現在

- 作業環境



# 過往與現在

- 使用人數



# 過往與現在

- 面臨考驗

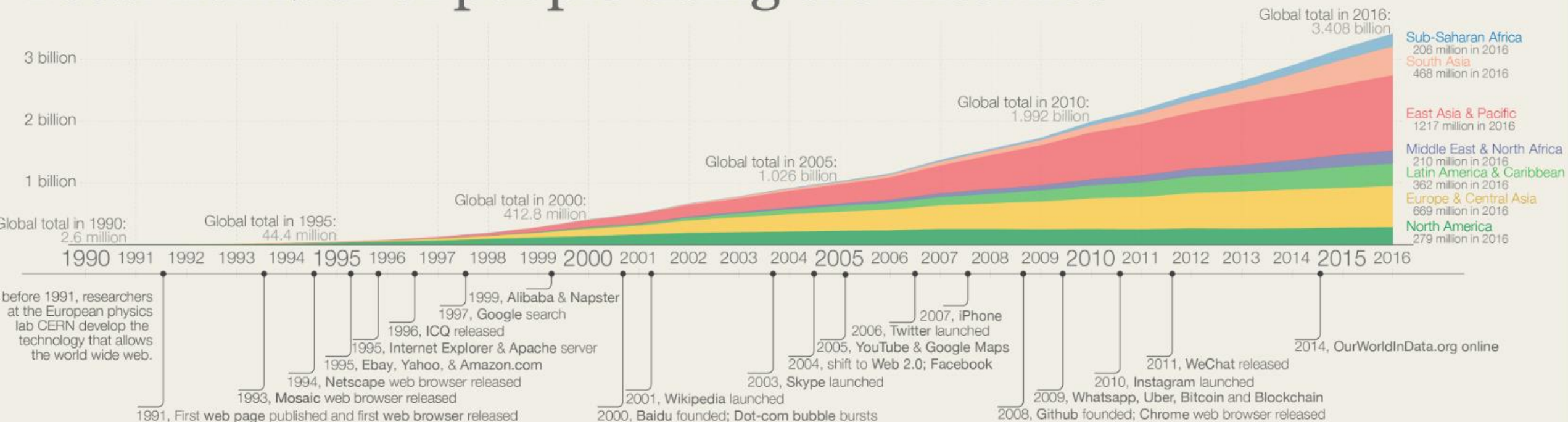
擴展

容錯

穩定

# 過往與現在

## Total number of people using the Internet



All countries for which data is available are shown in the main visualization but not all countries are labelled on the right. Internet usage includes access via a computer, mobile phone, personal digital assistant, games machine, digital TV etc. Data sources: International Telecommunication Union, World Telecommunication/ICT Development Report via the World Bank World Development Indicators.

This is a visualization from [OurWorldInData.org](https://ourworldindata.org), where you find data and research on how the world is changing.

Licensed under CC-BY-SA by the author

<https://ourworldindata.org/internet>





# Reactive Manifesto (2014)

- Responsive
- Resilient
- Elastic
- Message Driven

# Reactive Manifesto (2014)

- Responsive
  - 可用性
  - 實用性
  - 專注於提供快速、一致的回應時間

# Reactive Manifesto (2014)

- Resilient (彈性)
  - 高容錯能力
  - 冗餘 ( replication )
  - 遏制 ( containment )
  - 隔離 ( isolation )
  - 委託 ( delegation )

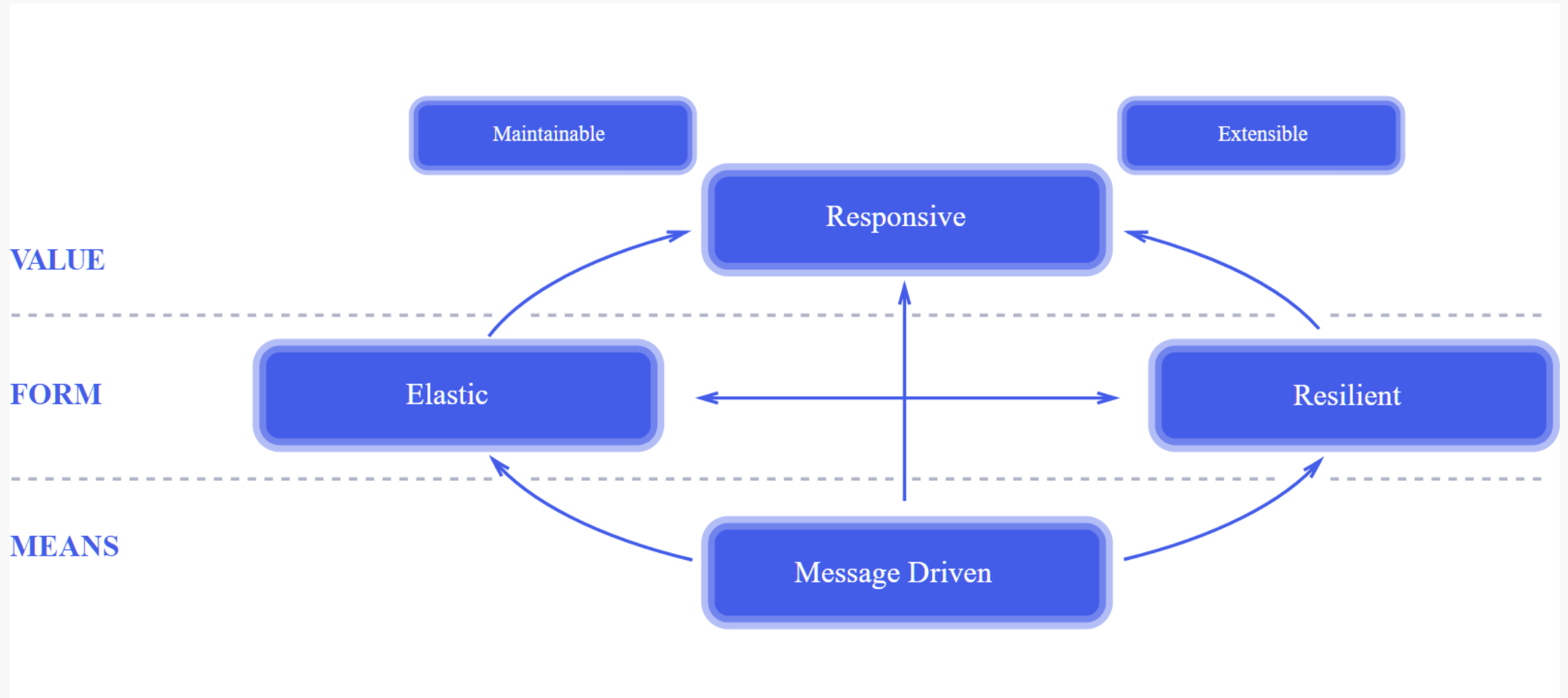
# Reactive Manifesto (2014)

- Elastic (擴展性)
  - scale up: make use of parallelism in multi-core systems
  - scale out: make use of multiple server nodes
  - important for scalability: minimize shared mutable state
  - important for scale out: Location transparency, resilience

# Reactive Manifesto (2014)

- Message Driven (Event-Driven)
  - 非同步

# Reactive Manifesto (2014)



# Reactive Extensions (Rx)





# 什麼是 Reactive Extensions (Rx)?

Rx = Observables + LINQ + Schedulers.

# 概念

- data streaming
- functional programming
- asynchronous observers

# 學習困難點

Imperative

V.S.

Functional

# Imperative - state

- Describes how the program operators
- uses statements to change state of the program
- conditional statements, loops, inheritance
- easier initial learning curve

# 範例



```
List<int> collection = new List<int> { 1, 2, 3, 4, 5 };

List<int> results = new List<int>();
foreach (var num in collection)
{
    if (num % 2 != 0)
        results.Add(num);
}
```

# Functional programming - stateless

- describe what the program should accomplish
- ideally stateless
- evaluate functions and create expressions instead of statements
- more difficult conceptually

# 範例



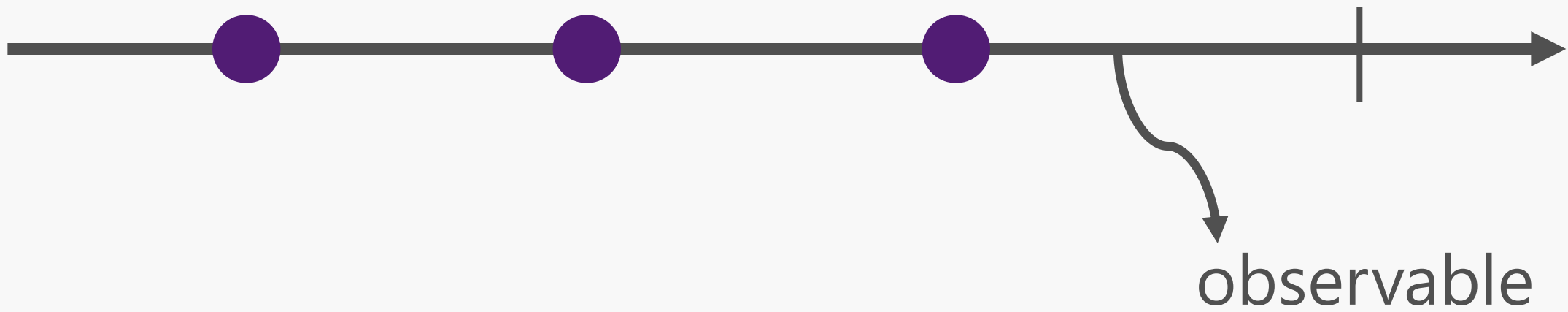
```
List<int> collection = new List<int> { 1, 2, 3, 4, 5 };  
  
var results = collection.Where( num => num % 2 != 0 );
```

# 組成介紹

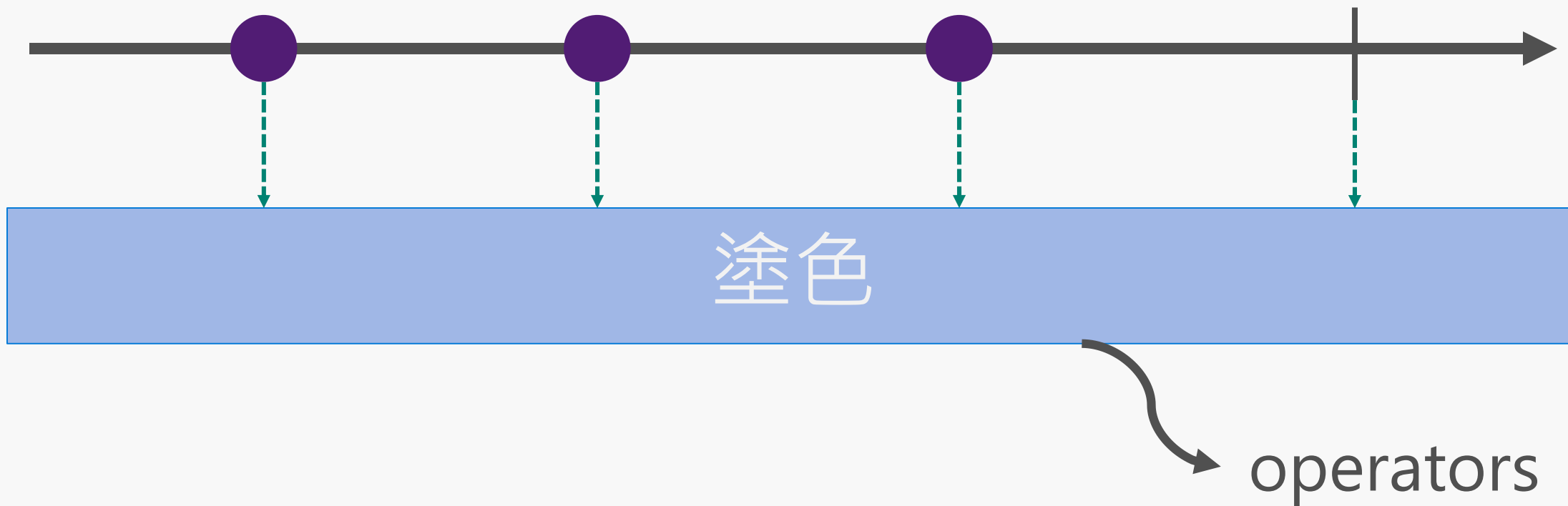
- observable
- operators
- observer
- subscription



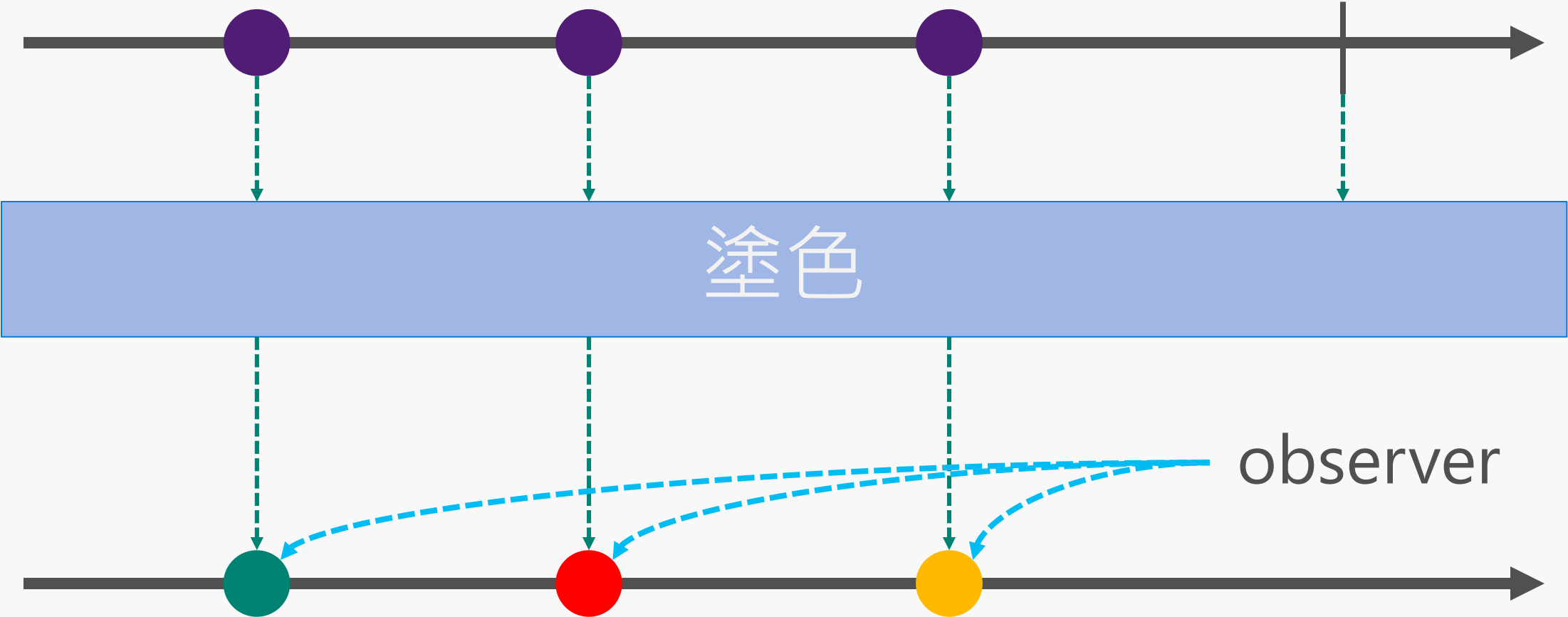
observable



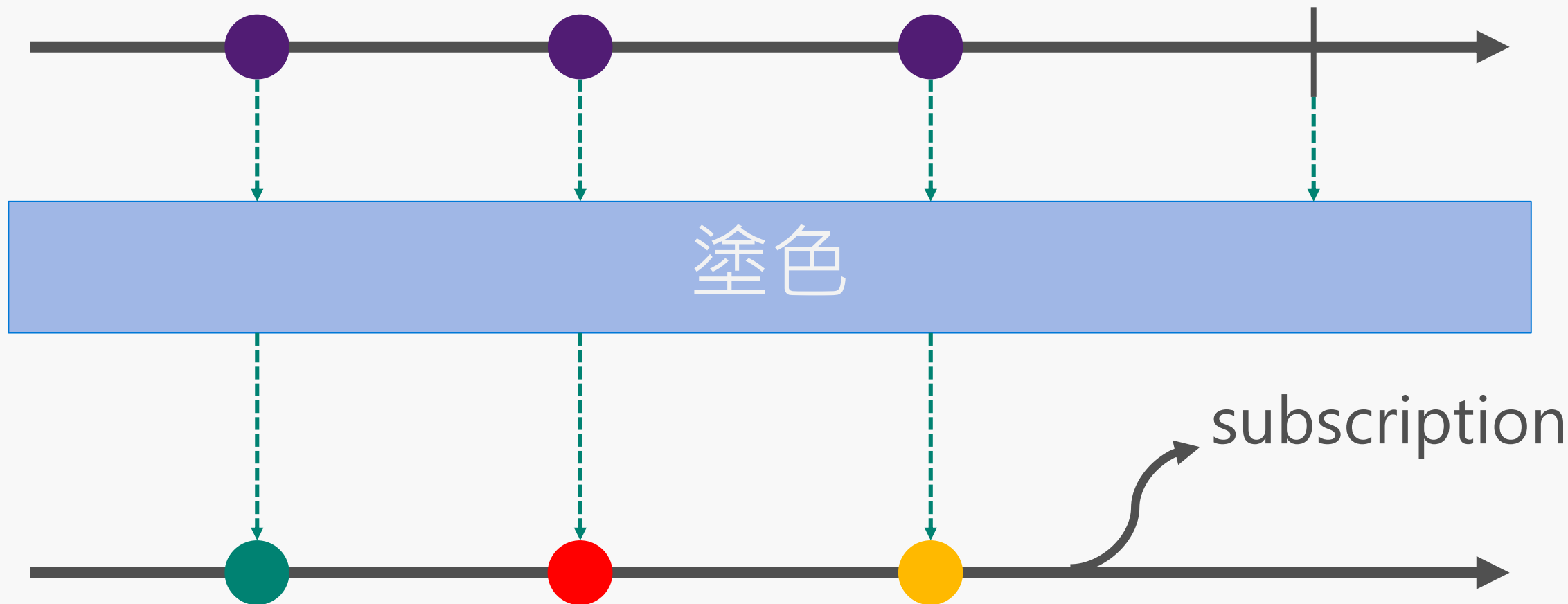
operators



observer



subscription



# 現況

# Reactive Extensions 之類的

- Rx.NET
- RxJS
- RxJava
- RxScala
- RxDart
- RxPy
- ...
- Akka.NET
- NgRx

# Demo - RxJS



# 範例程式碼

- [活動 live demo](#)
- [Processing Pipelines Series - Reactive Extensions \(Rx.NET\)](#)
- [後端工程師必備: 平行任務處理的思考練習 - PR5](#)



# 延伸閱讀

- [響應式程式開發之 .NET Core 應用](#) (2017-10-13)
- [ReactiveX](#)
- [dotnet/reactive](#)
- [Reactive Programming](#)

# 特別感謝



以及各位參與活動的你們

