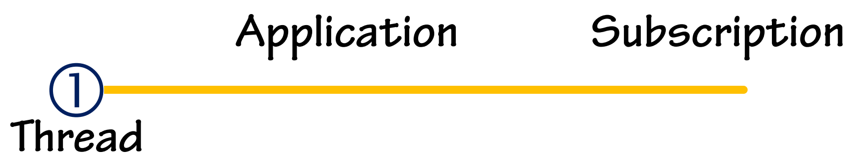
Managing Concurrency

*The left hand **does** know what the right is doing.*

Overview

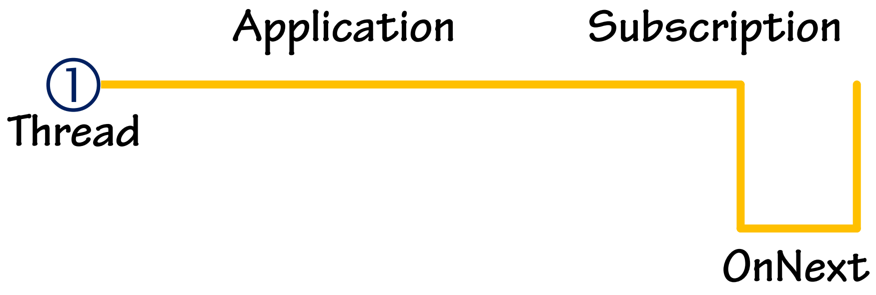
- Scheduler
- ToObservable, SubscribeOn ObserveOn



```
enumerator.MoveNext();
…
current = enumerator.Current;
```

Overview

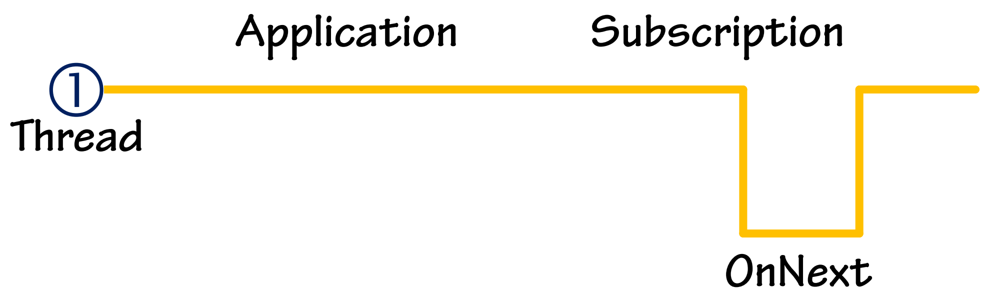
- Scheduler
- ToObservable, SubscribeOn ObserveOn



```
observer.OnNext(current);
```

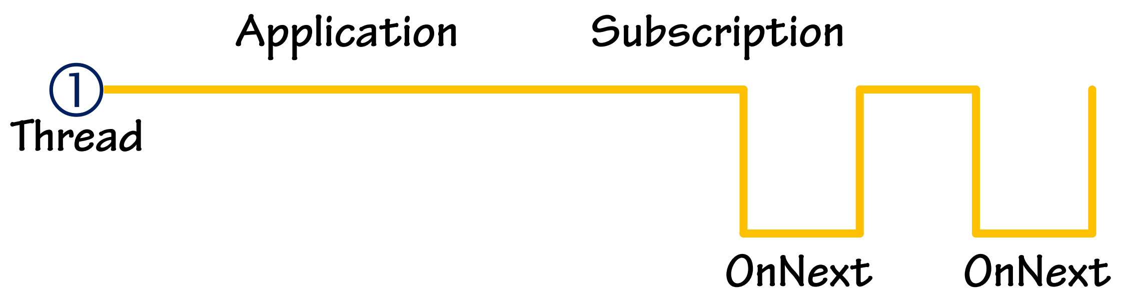
Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



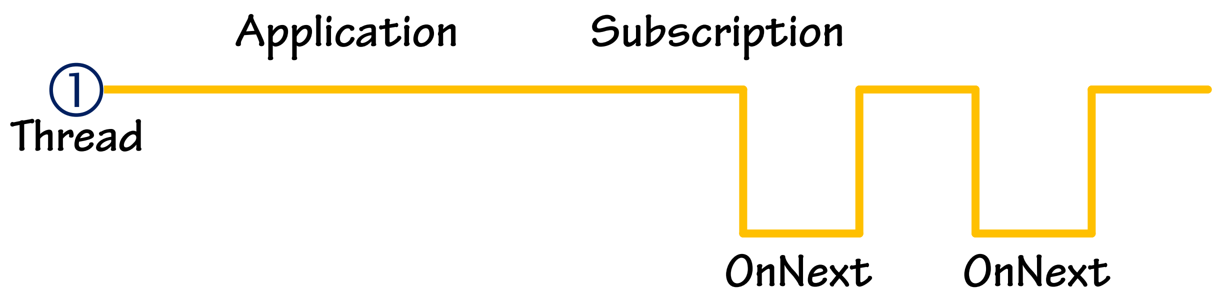
Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



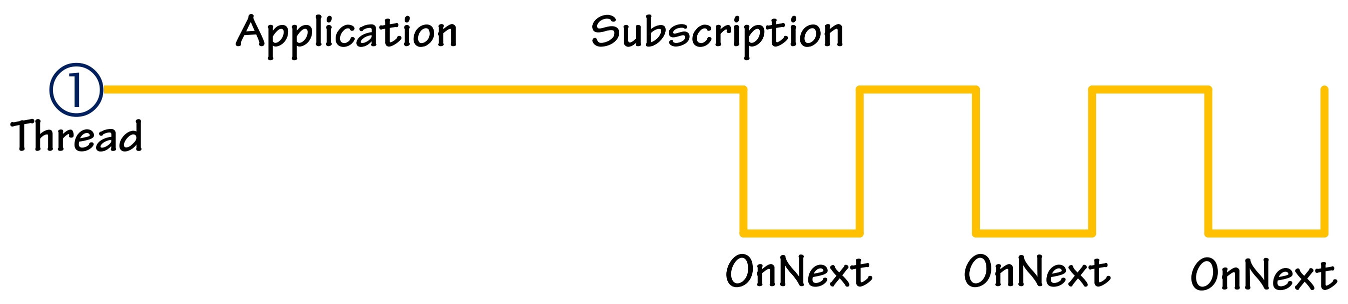
Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



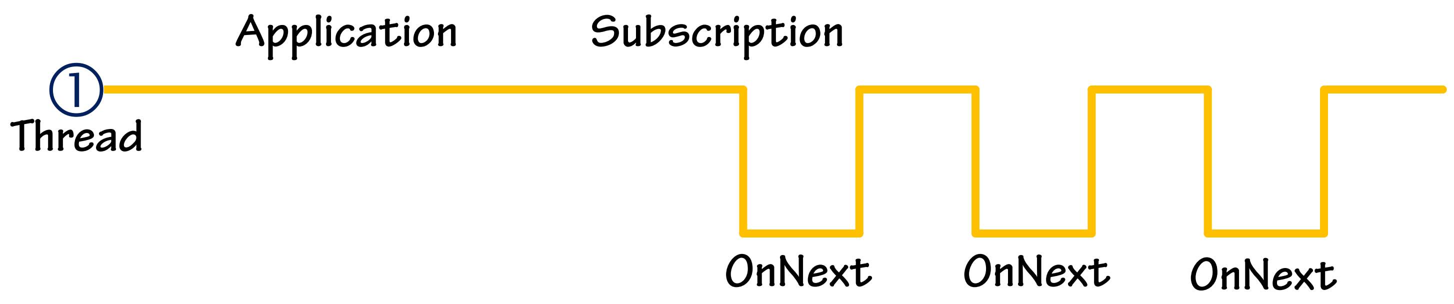
Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



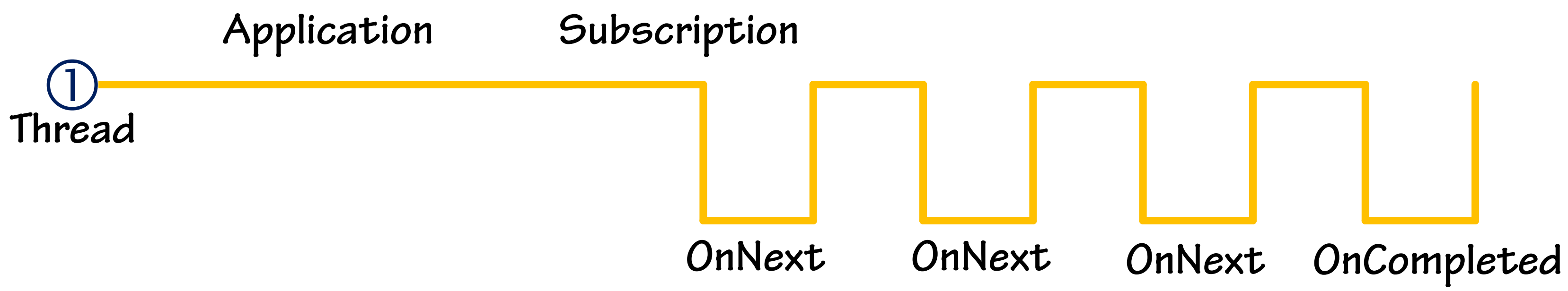
Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



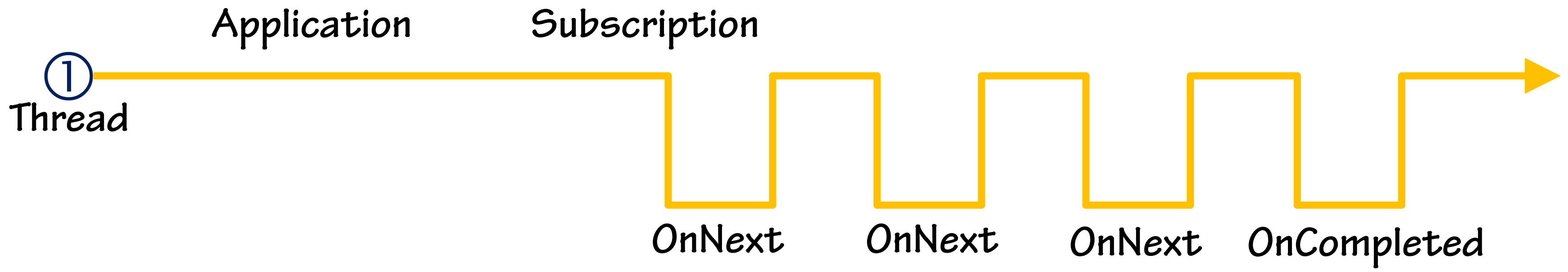
Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



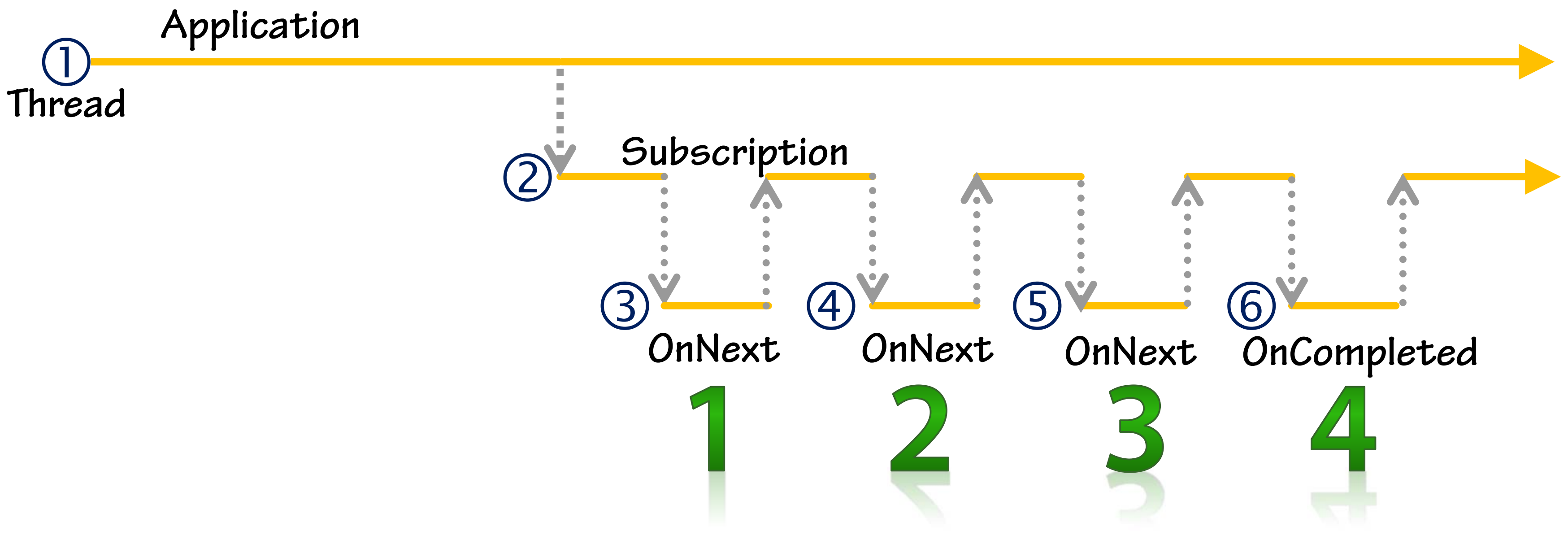
Overview

- **Scheduler**
  - **ToObservable, SubscribeOn ObserveOn**



Overview

- Scheduler
- ToObservable, SubscribeOn ObserveOn



Scheduling Delegates

- ToObservable
- SubscribeOn ObserveOn

```
Scheduler.NewThread.Schedule(MyDelegate)
```

Scheduling Delegates

- ToObservable
- SubscribeOn ObserveOn

subscription

```
enumerator.MoveNext();
…
current = enumerator.Current;
```

part of subscription

```
from number in Enumerable.Range(1,3) select number;
```

part of observation

```
observer.OnNext(current);
```

Scheduling Delegates

- ToObservable
- SubscribeOn ObserveOn

subscription

```
enumerator.MoveNext();
…
current = enumerator.Current;
```

part of subscription

```
from number in Enumerable.Range(1,3) select number;
```

part of observation

```
observer.OnNext(Process(current))
```

Scheduling Delegates

- ToObservable
- SubscribeOn ObserveOn

subscription

```
enumerator.MoveNext();
…
current = enumerator.Current;
```

part of subscription

```
from number in Enumerable.Range(1,3) select Process(number)
```

part of observation

```
observer.OnNext(current);
```

Scheduling Delegates

- ToObservable
- SubscribeOn ObserveOn

```
ToObservable(Scheduler.NewThread)

 NewThread
 ThreadPool
 TaskPool
 CurrentThread
 Immediate
 Dispatcher
```

Scheduling Delegates

- ToObservable
- SubscribeOn ObserveOn

```
SubscribeOn(Scheduler.NewThread).ObserveOn(Scheduler.Dispatcher)
```

Specific Scheduling

- ObserveOn
- SubscribeOn

Observable Lifetime

- Subscriptions cleanup automatically
- Subscriptions are disposable

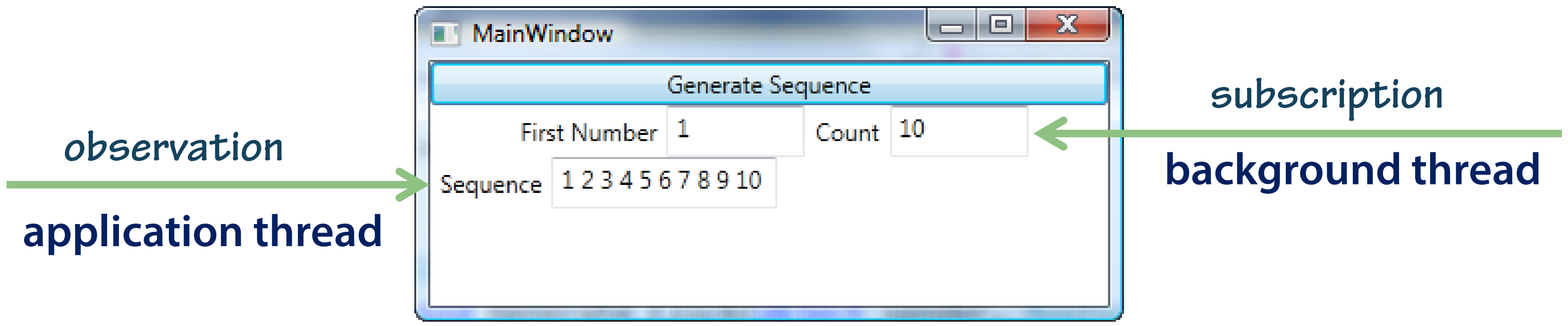
```
var subscription= observableSequence.Subscribe(Console.WriteLine);
//…
subscription.Dispose();
```

Cleanup

- Using
- Finally

Scheduler.Dispatcher

- **GUI applications**
  - WPF, Forms



First Number: 1

Count: 10

Sequence: 1 2 3 4 5 6 7 8 9 10

Summary

- Declarative
- Subscription / Observation
  - SubcribeOn / ObserveOn
- Observation Grammer / Concurrency
  - (OnNext*)?, (OnError | OnCompleted)?
- Subscription termination
- Disposable objects
- Cleanup
- WPF and Windows Forms applications