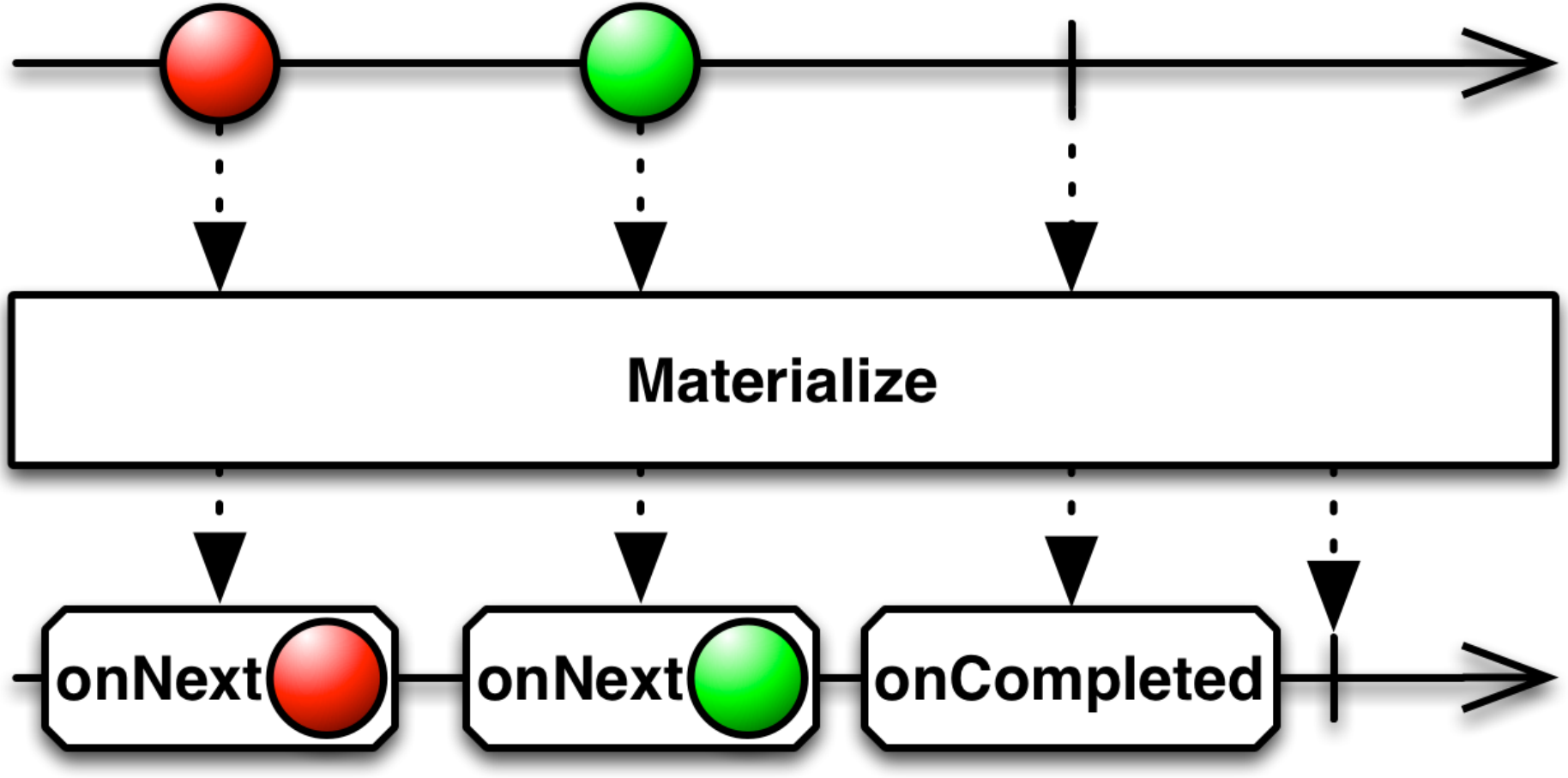
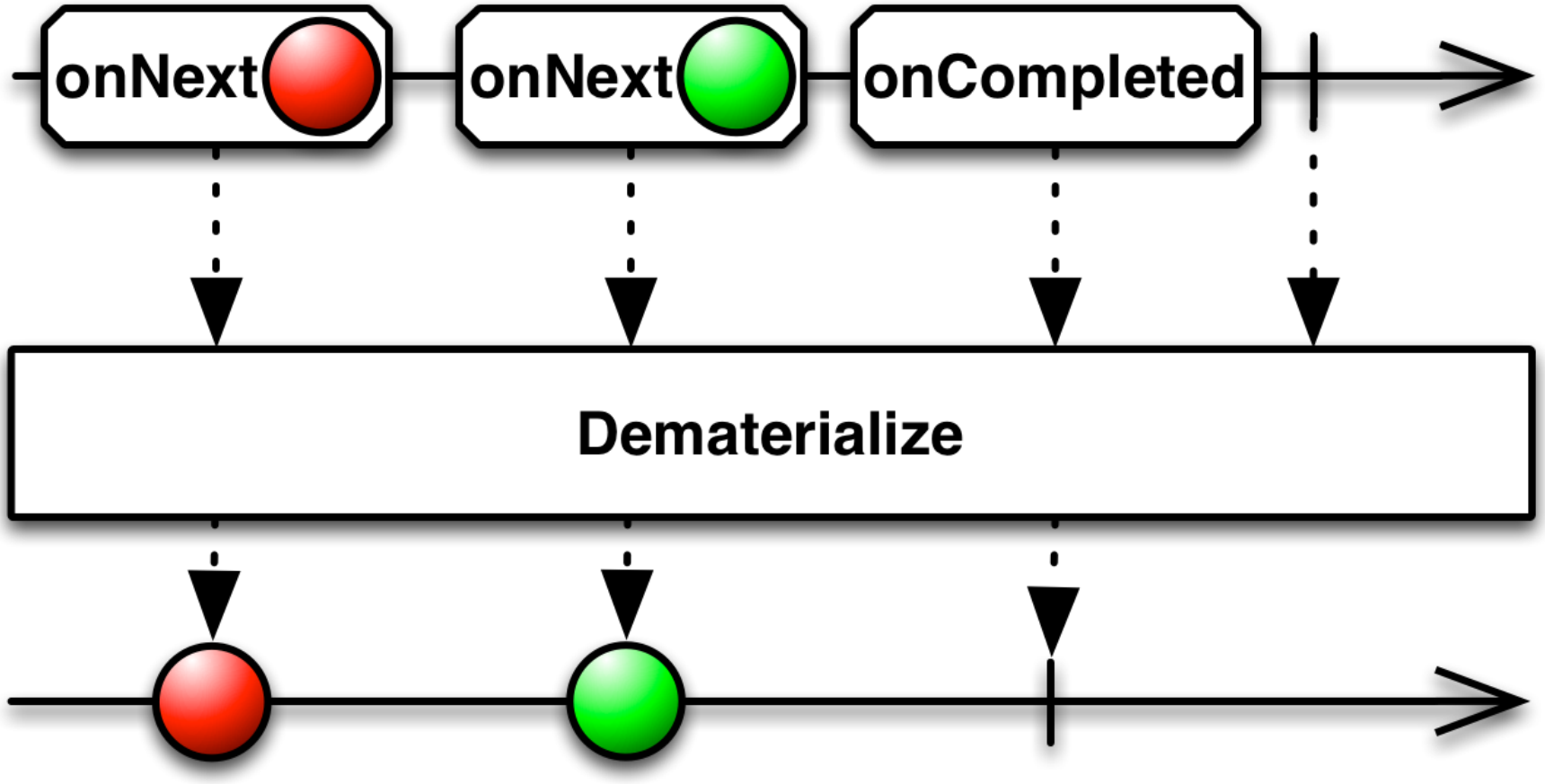
22.Materialize/Dematerialize

Materialize 示意图

Materialize 将数据项和事件通知都当做数据项发射，


Materialize
onNext
onNext
onCompleted

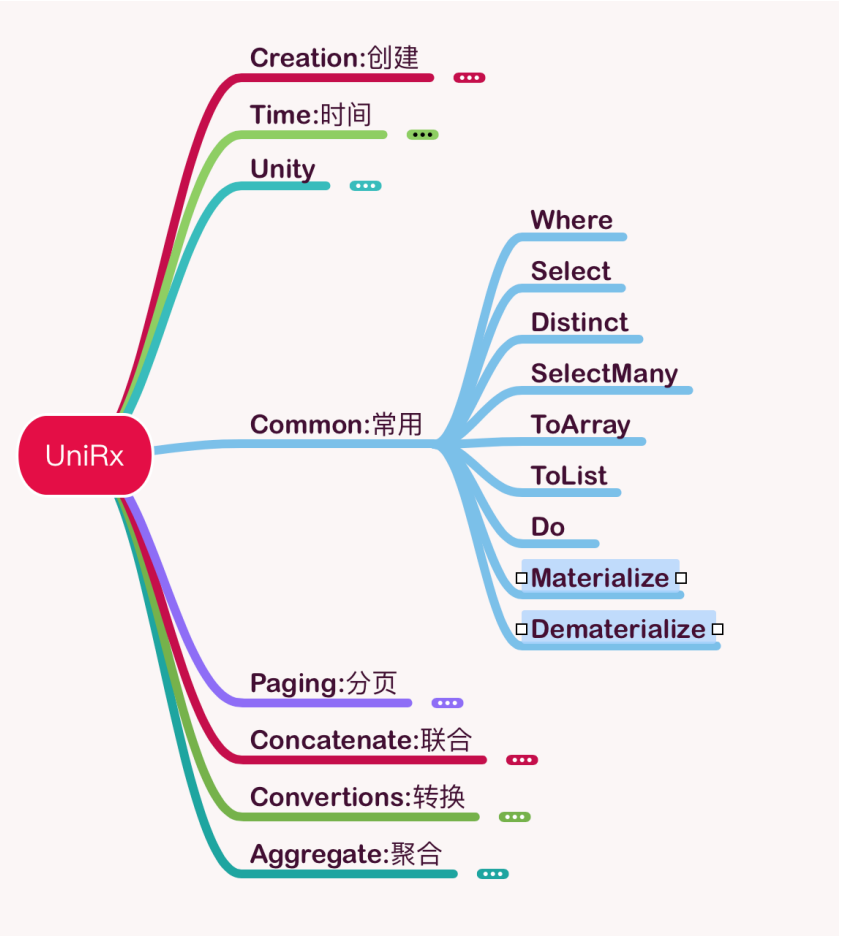
一个合法的有限的Obversable将调用它的观察者的 onNext 方法零次或多次，然后调用观察者的 onCompleted 或 onError 正好一次。Materialize 操作符将这一系列调用，包括原来的 onNext 通知和终止通知 onCompleted 或 onError 都转换为一个Observable发射的数据序列。


onNext
onNext
onCompleted
Dematerialize

Dematerialize 操作符是Materialize的逆向过程，它将Materialize转换的结果还原成它原本的形式。

dematerialize反转这个过程，将原始Observable发射的Notification对象还原成Observable的通知。

Materialize/Dematerialize 所在知识地图中的位置


UniRx
Creation:创建
Time:时间
Unity
Common:常用
Where
Select
Distinct
SelectMany
ToArray
ToList
Do
Materialize
Dematerialize
Paging:分页
Concatenate:联合
Convertions:转换
Aggregate:聚合

Materialize/Dematerialize 代码示例

```
/*************************************************************************
 * http://sikiedu.com liangxie
 *************************************************************************/

using System;
using UniRx;
using UnityEngine;

namespace UniRxLesson
{
      public class MaterializeExample : MonoBehaviour
      {
            private void Start()
            {
                  var subject = new Subject<int>();
                  var onlyExceptions = subject.Materialize()
```

```
                .Where(n => n.Exception != null)
                .Dematerialize();

            subject.Subscribe(i => Debug.LogFormat("Subscriber 1: {0}", i),
                ex => Debug.LogFormat("Subscriber 1 exception: {0}", ex.Message));

            onlyExceptions.Subscribe(i => Debug.LogFormat("Subscriber 2: {0}", i),
                ex => Debug.LogFormat("Subscriber 2 exception: {0}", ex.Message));

            subject.OnNext(123);

            subject.OnError(new Exception("Test Exception"));
        }
    }
}
```

输出结果为

```
Subscriber 1: 123
```

```
Subscriber 1 exception: Test Exception
Subscriber 2 exception: Test Exception
```

今天的内容就这些