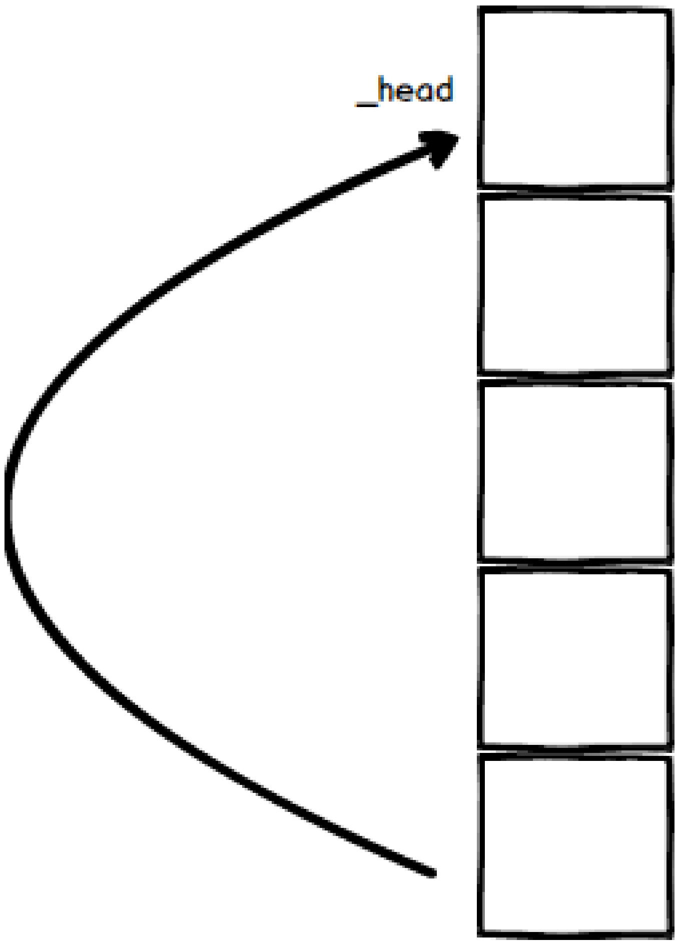
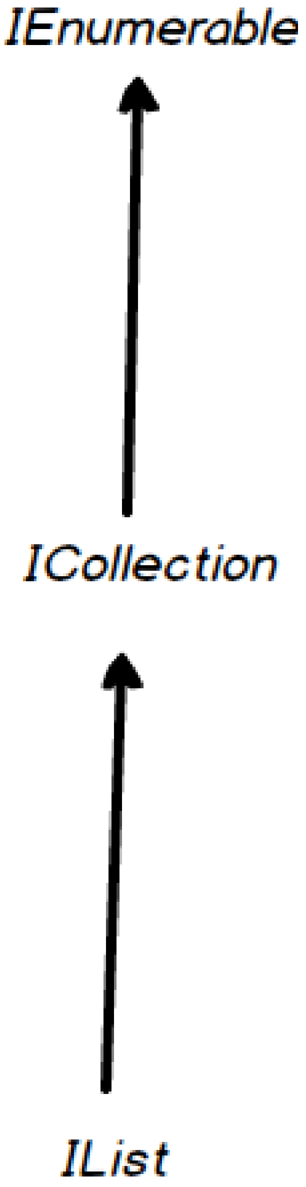
foreach(var element in elements)



IEnumerator

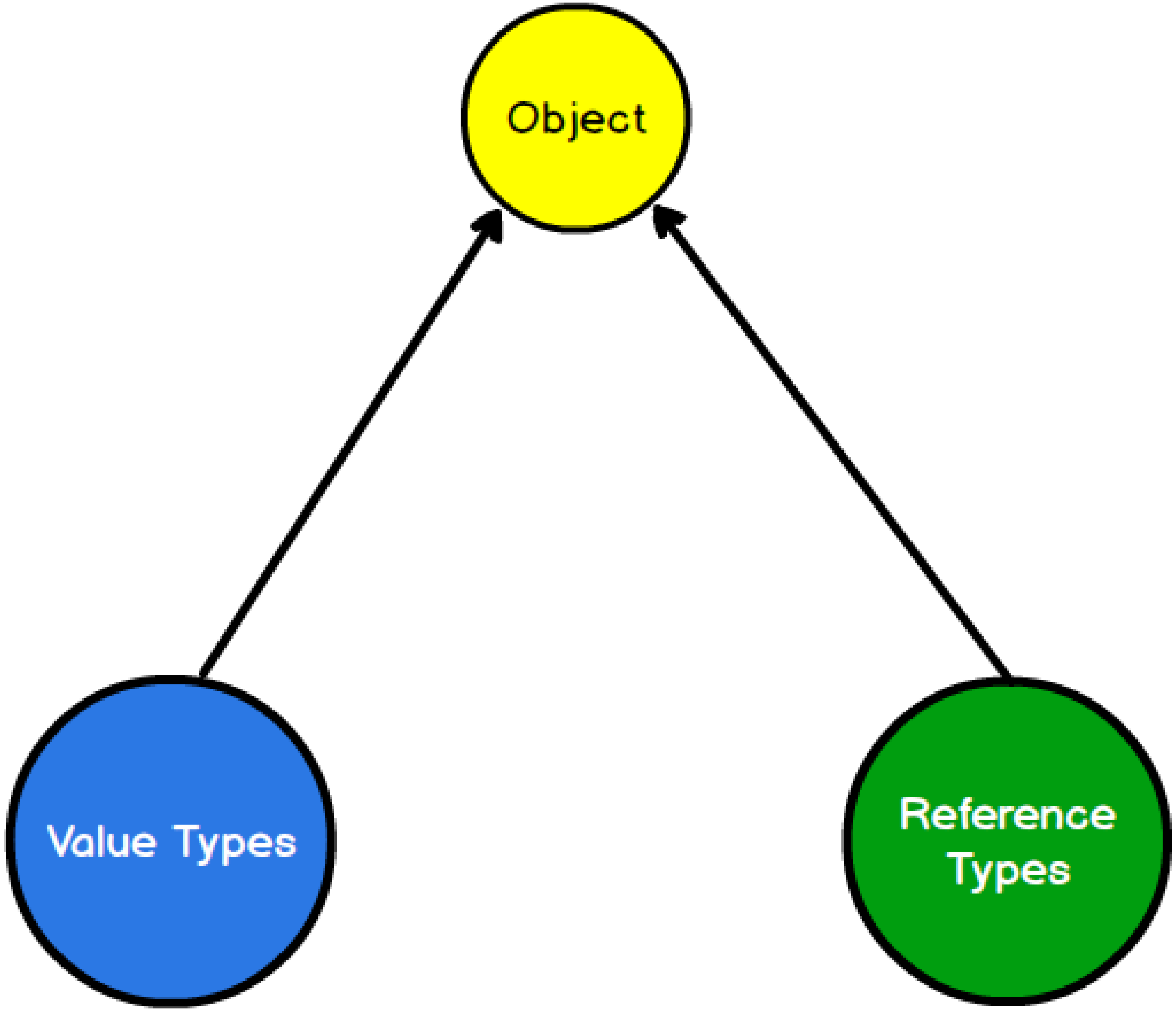
Current

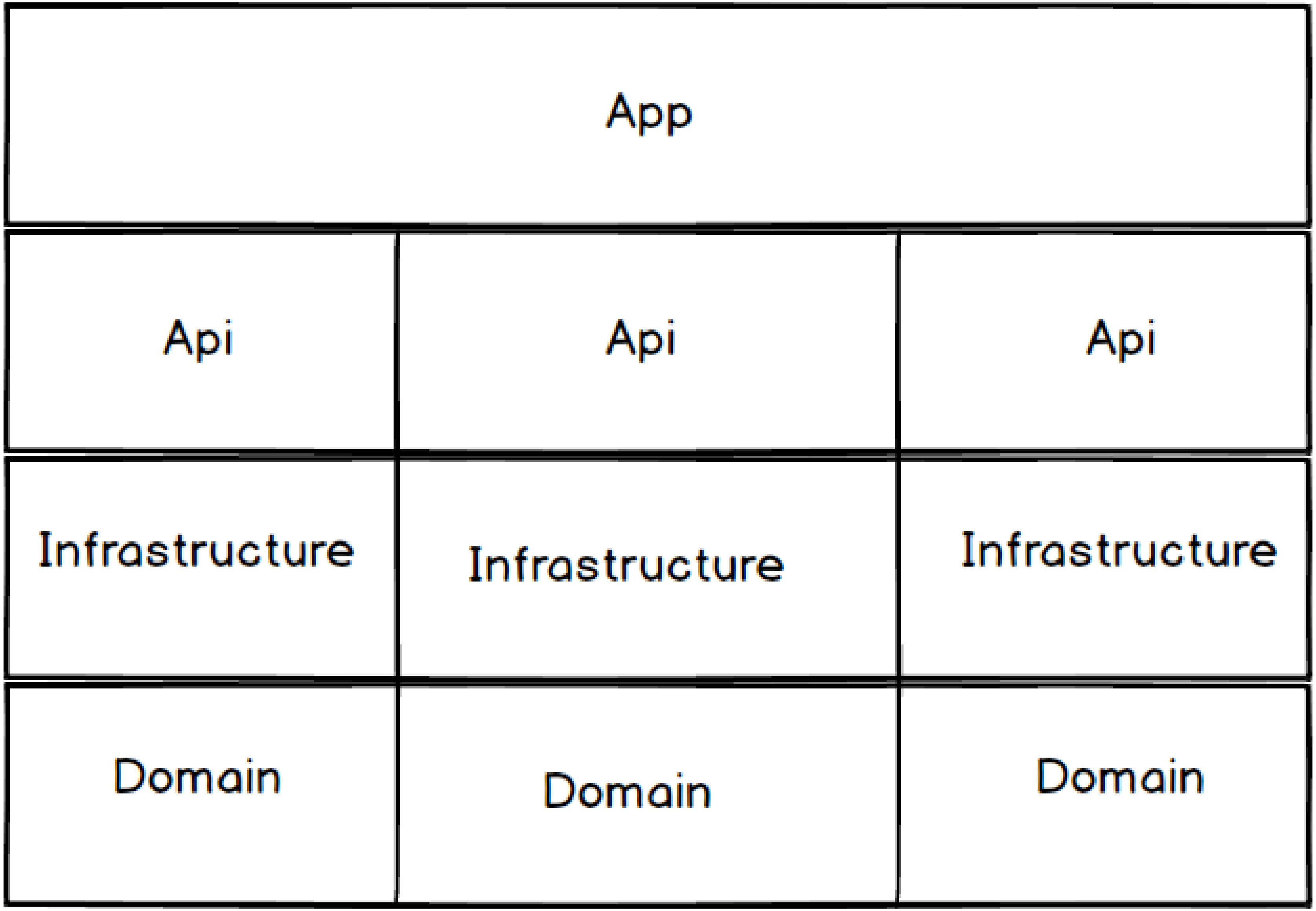


IEnumerable

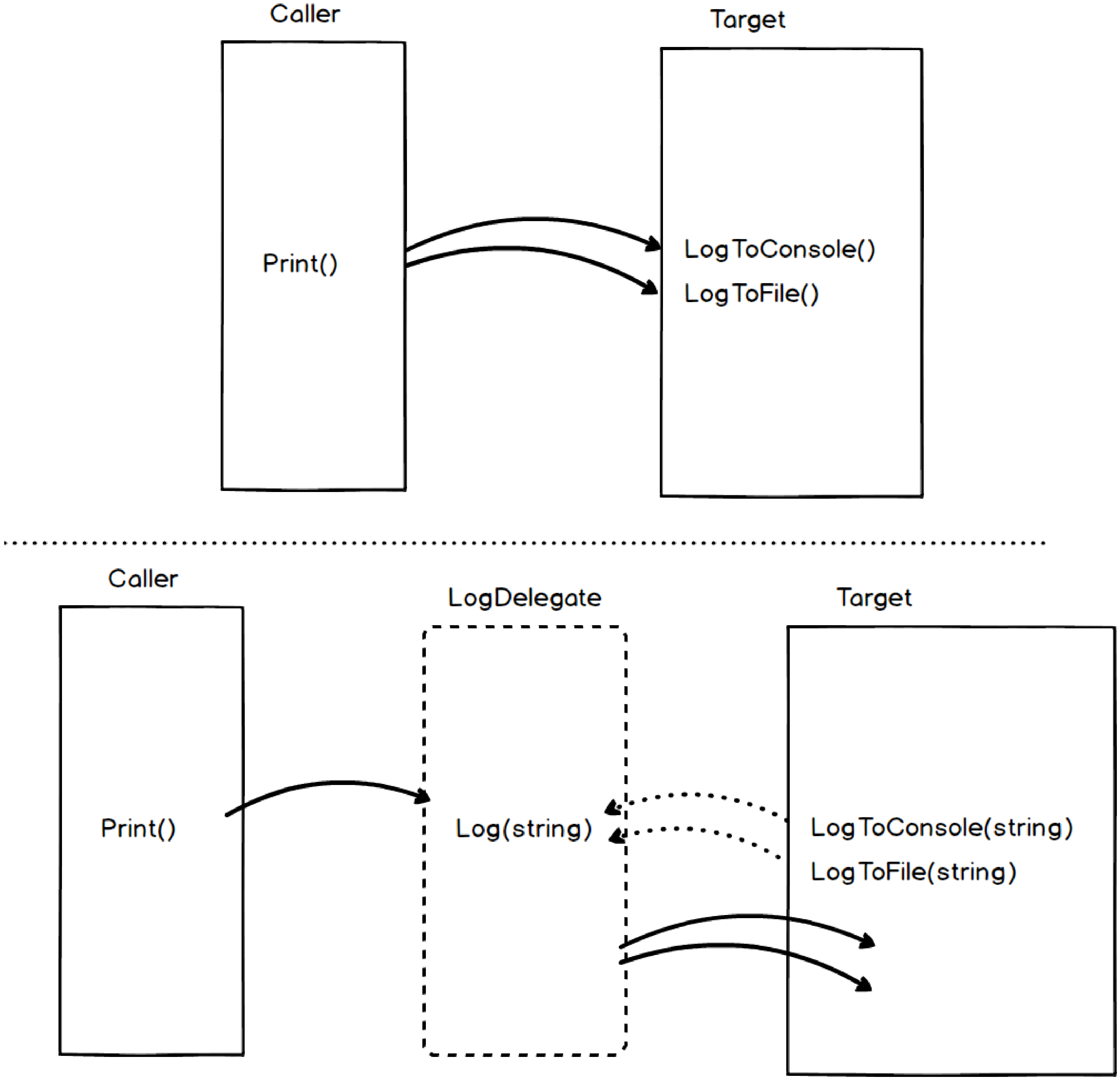
ICollection

IList


Object
Value Types
Reference Types



App		
Api	Api	Api
Infrastructure	Infrastructure	Infrastructure
Domain	Domain	Domain


Caller
Target
Print()
LogToConsole()
LogToFile()
Caller
LogDelegate
Target
Print()
Log(string)
LogToConsole(string)
LogToFile(string)

f(x) = x + 2 gdzie x należy do C

h(x) = x + 2 gdzie x należy do C

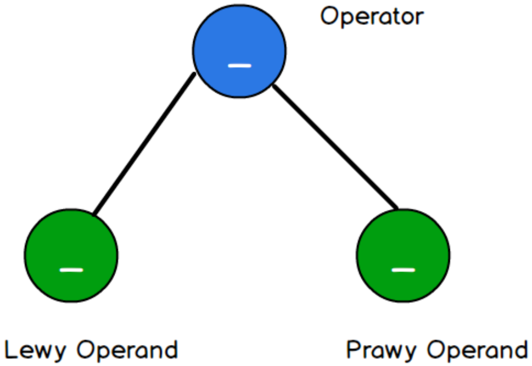
```
int F(int x) {
   return x + 2;
}
```

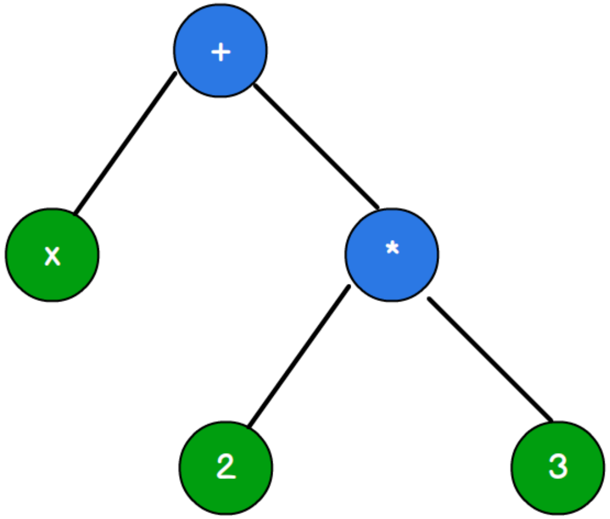
x -> x + 2 gdzie x należy do C

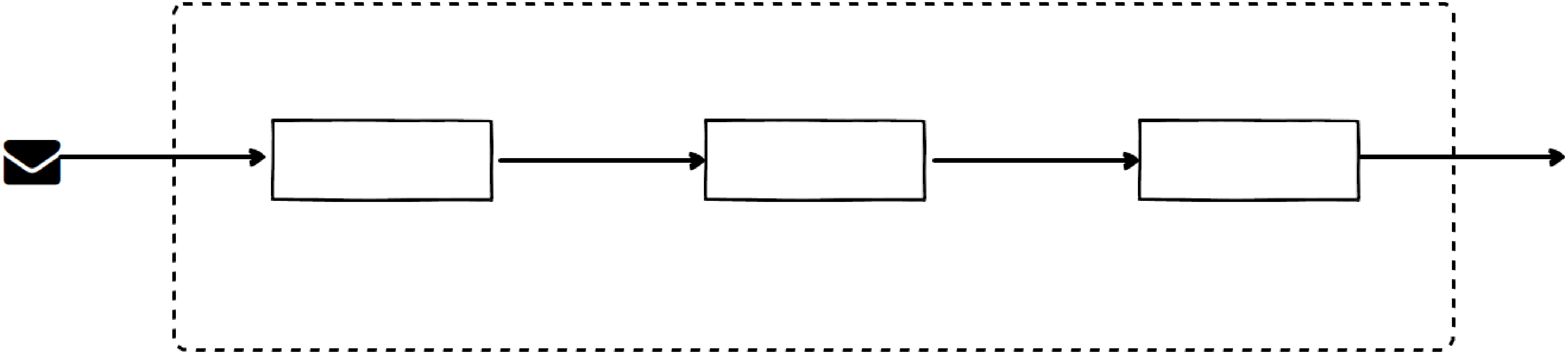
x => x + 2

x + 2 * 3

LINQ



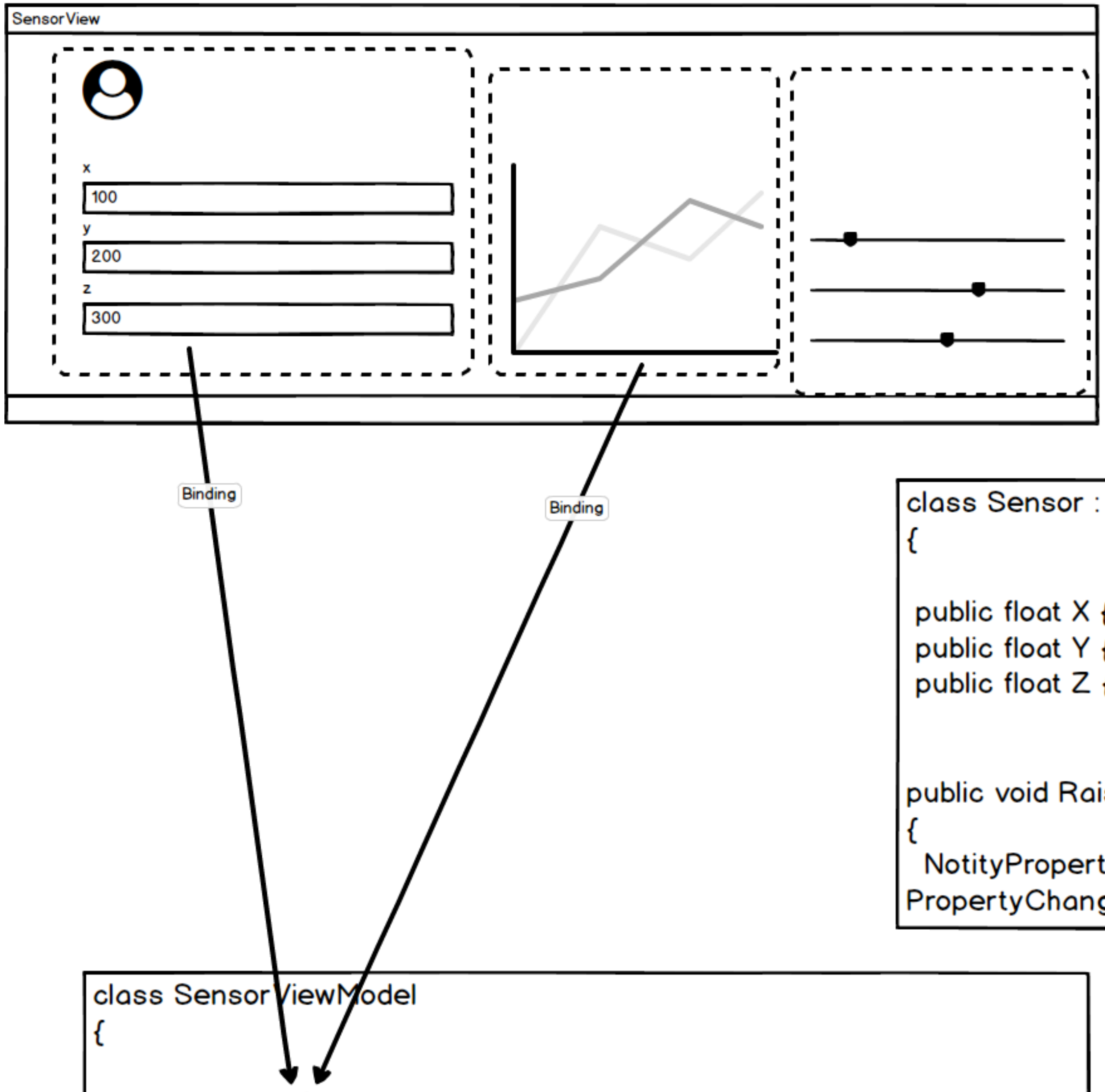




```
a = A(x);

b = B(a);

c = C(b);
```

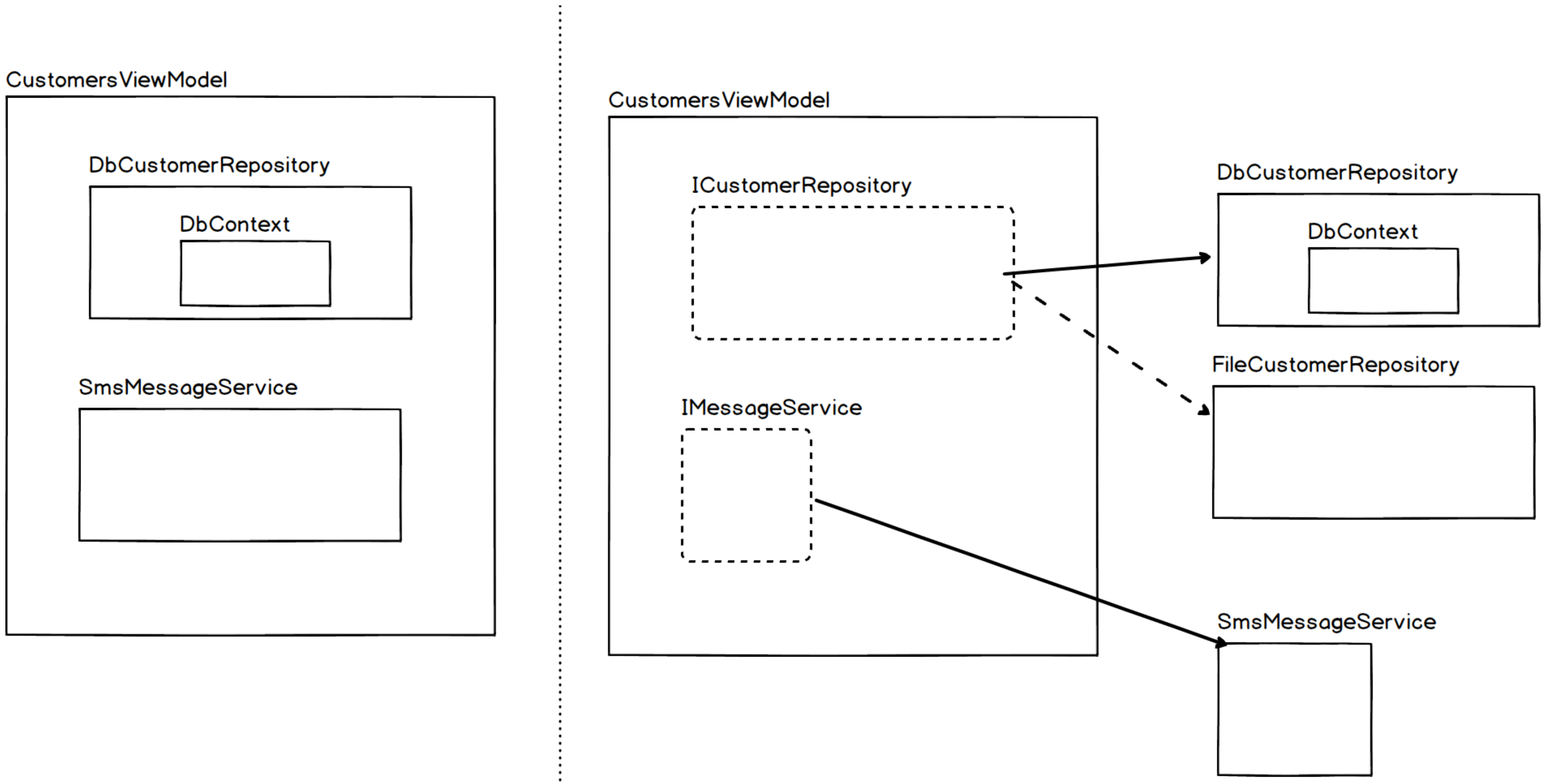
```
result = A(x).B().C()
```

Observer Design Pattern *(Event Aggregator)*

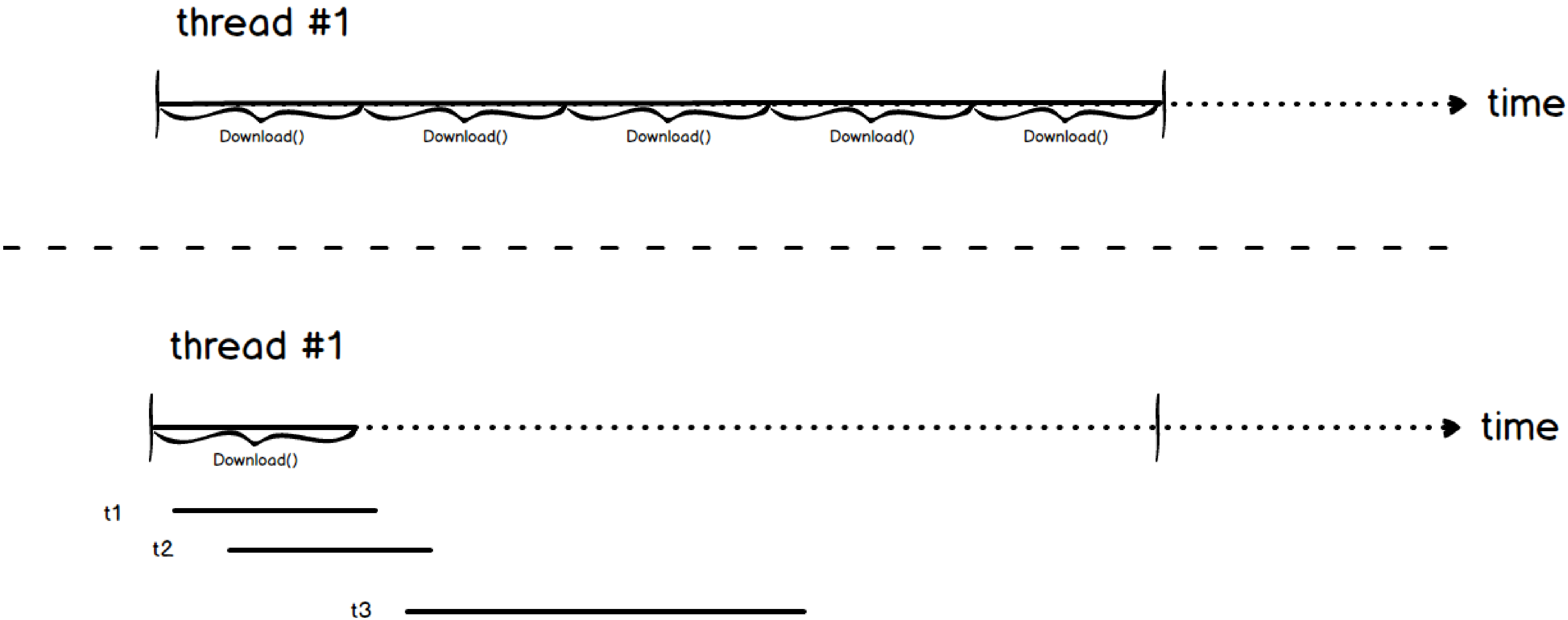


```
class Sensor : INotityPropertyChanged
{

 public float X { get; set; }
 public float Y { get; set; }
 public float Z { get; set; }


public void RaiseNotifyPropertyChanged(string propertyName)
{
  NotityPropertyChanged?.Invoke(this, new
PropertyChanged(propertyName));
```

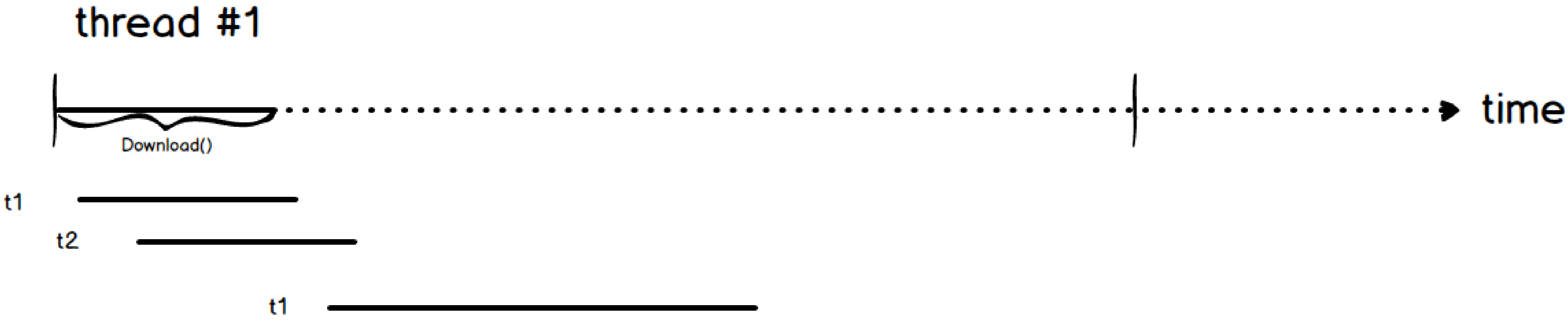
```
class SensorViewModel
{

 public Sensor { get; set; }
 public DeltaSensor { get; set; }



public double Scale {get; set;  }
```

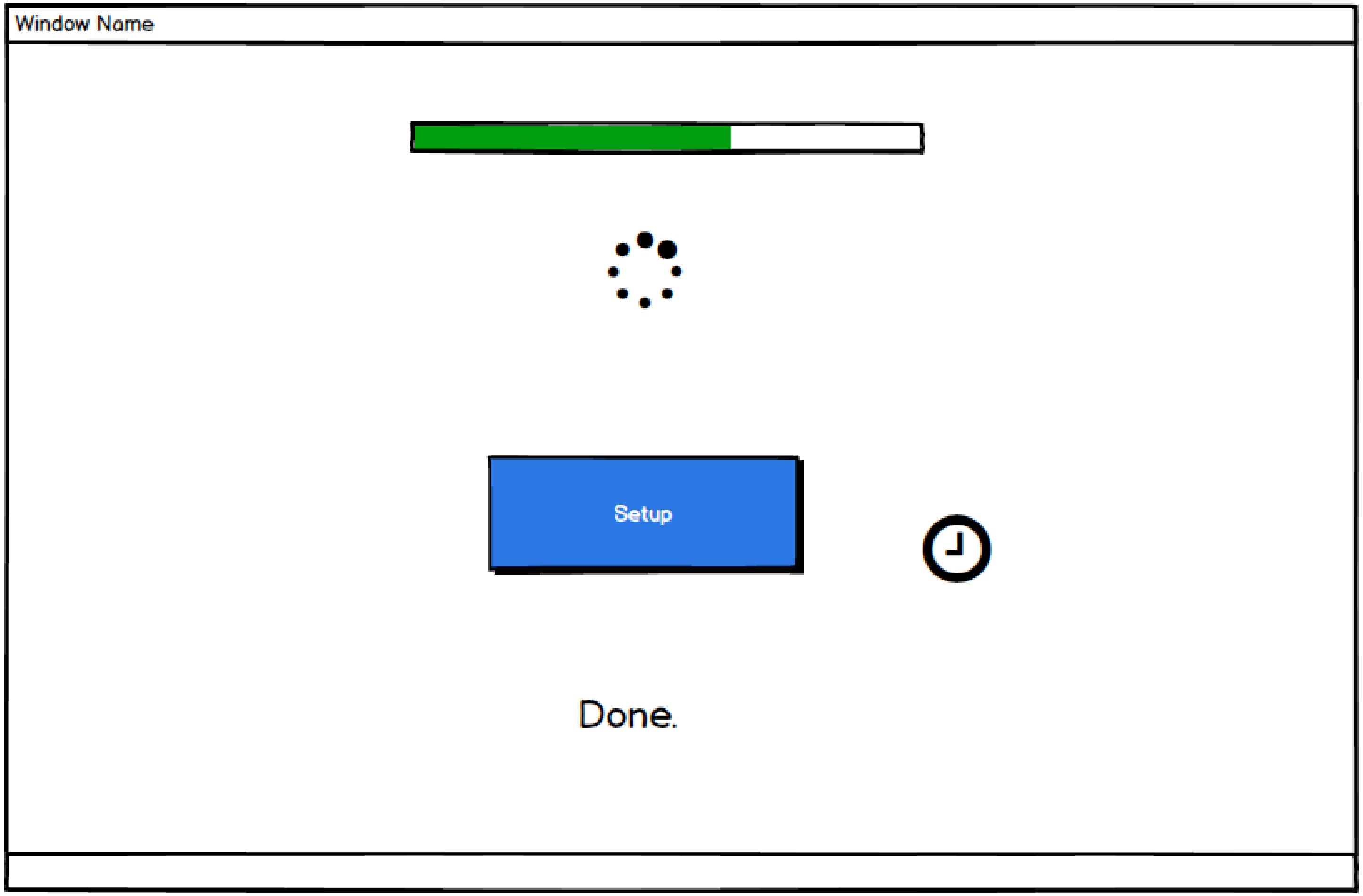


Type	TargetType	Cykl życia
IMessageService	SmsMessageService	Singleton
ICustomerRepository	DbCustomerRepository	Singleton
DbContext	DbContext	Transient
CustomersViewModel	CustomersViewModel	Transient

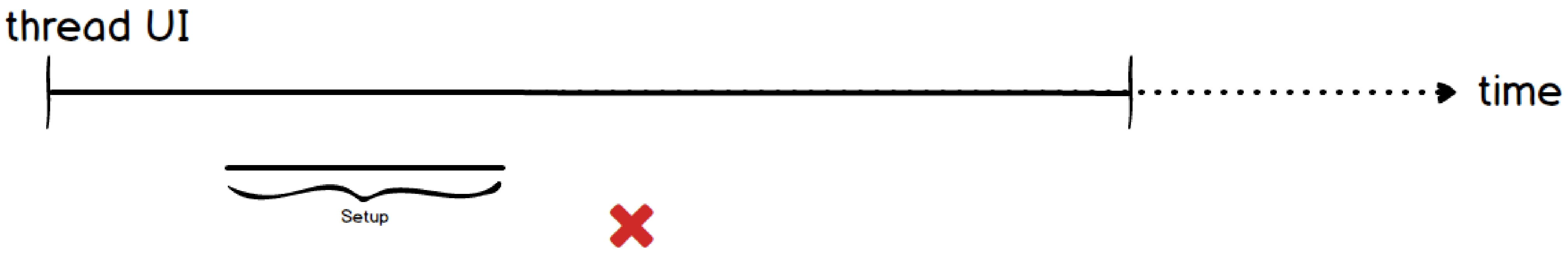

thread #1
time
Download()
Download()
Download()
Download()
Download()
thread #1
time
Download()
t1
t2
t3

Thread Pool

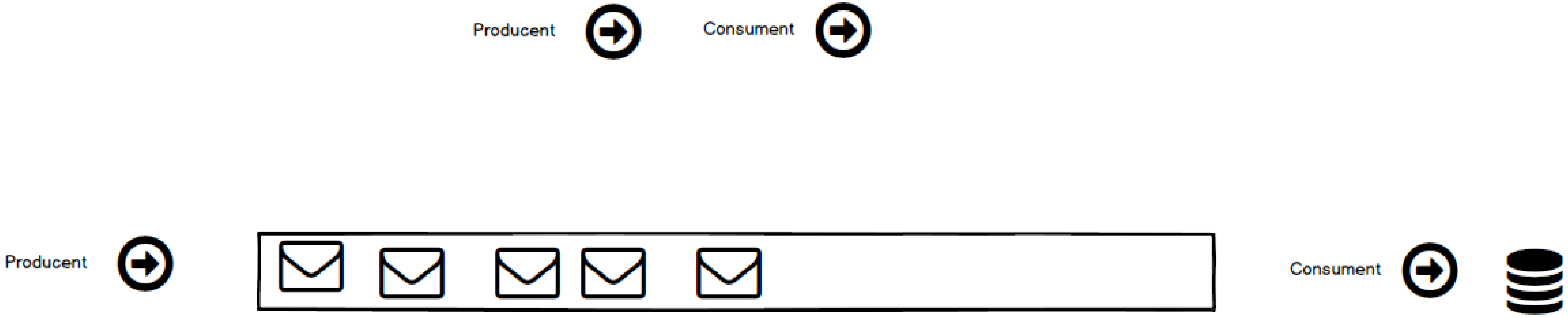




```
setupButton.Dispatcher.BeginInvoke(()=> status.Text = "Done."
```



```
status.Text = "Done."
```


Producent
Consument
Producent
Consument