IDisposable Best Practices for C# Developers

Introducing IDisposable

Elton Stoneman
geekswithblogs.net/eltonstoneman
@EltonStoneman

pluralsight
hardcore dev and IT training

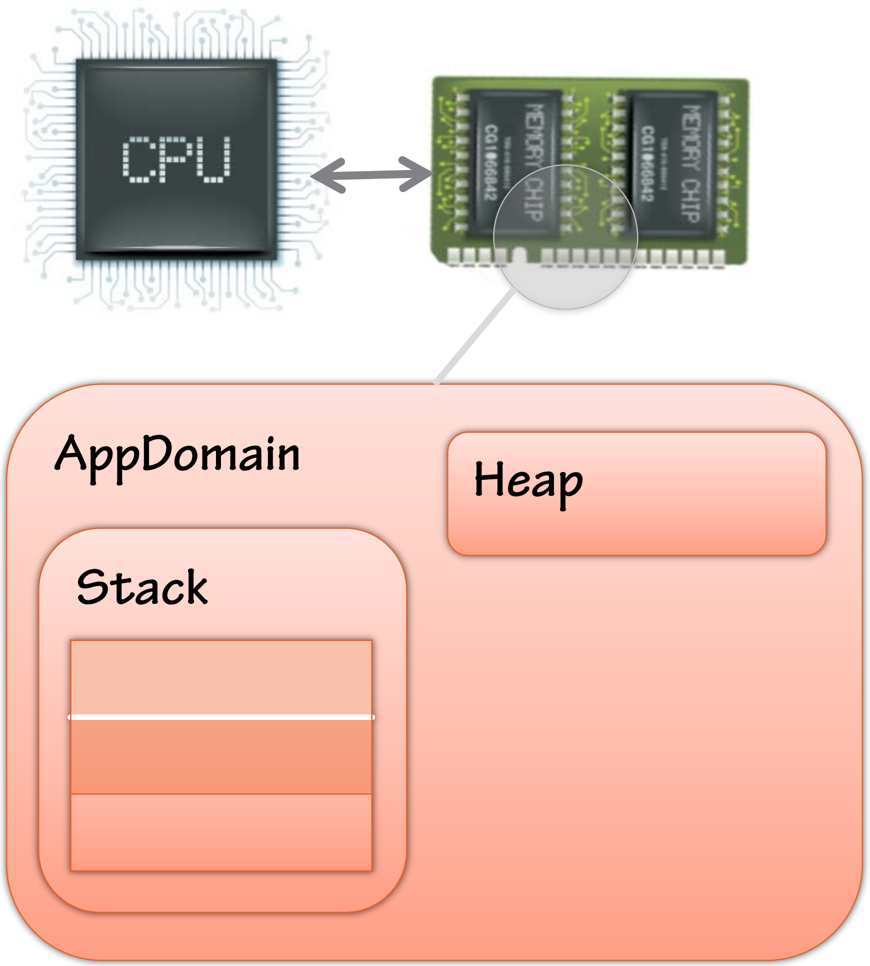
Introducing IDisposable

Memory

Databases

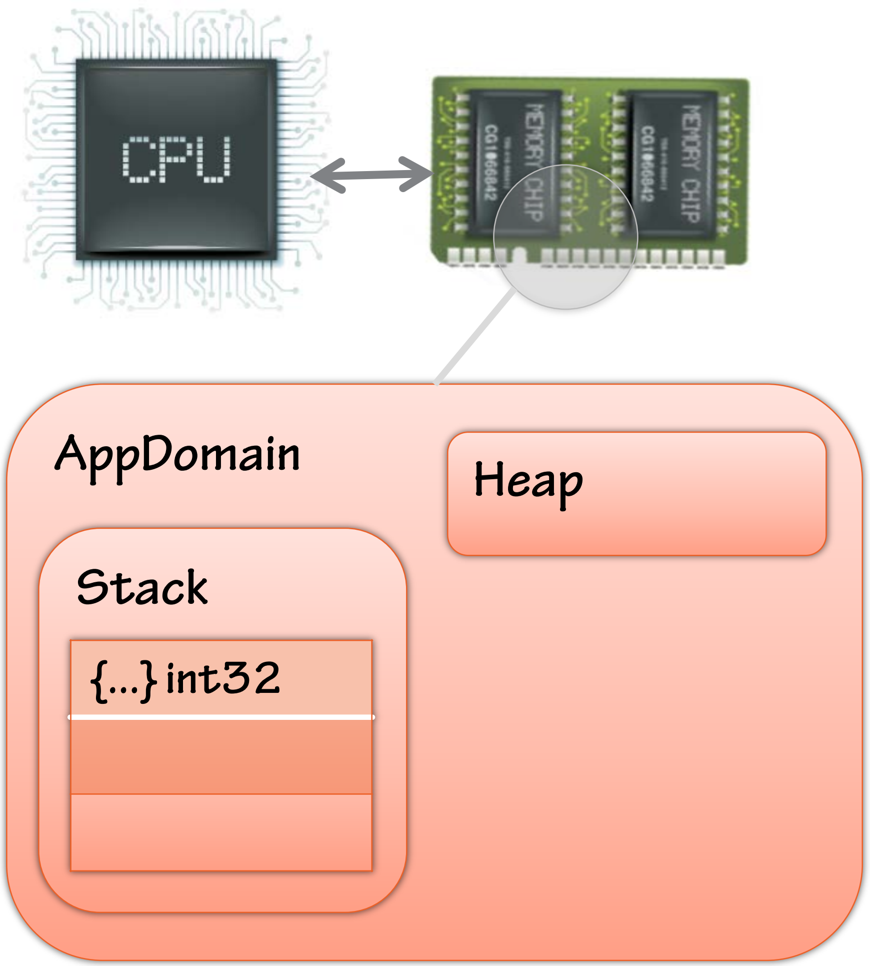
Network

Introducing IDisposable



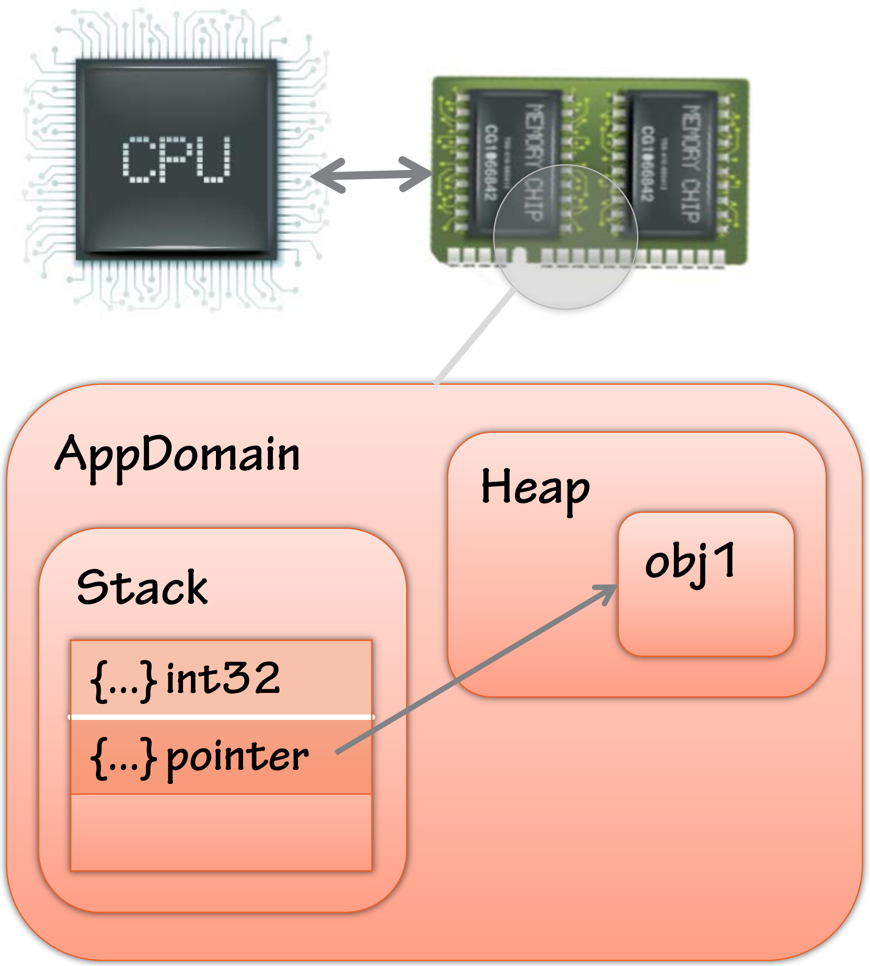
Process allocates memory

Introducing IDisposable



Process allocates memory

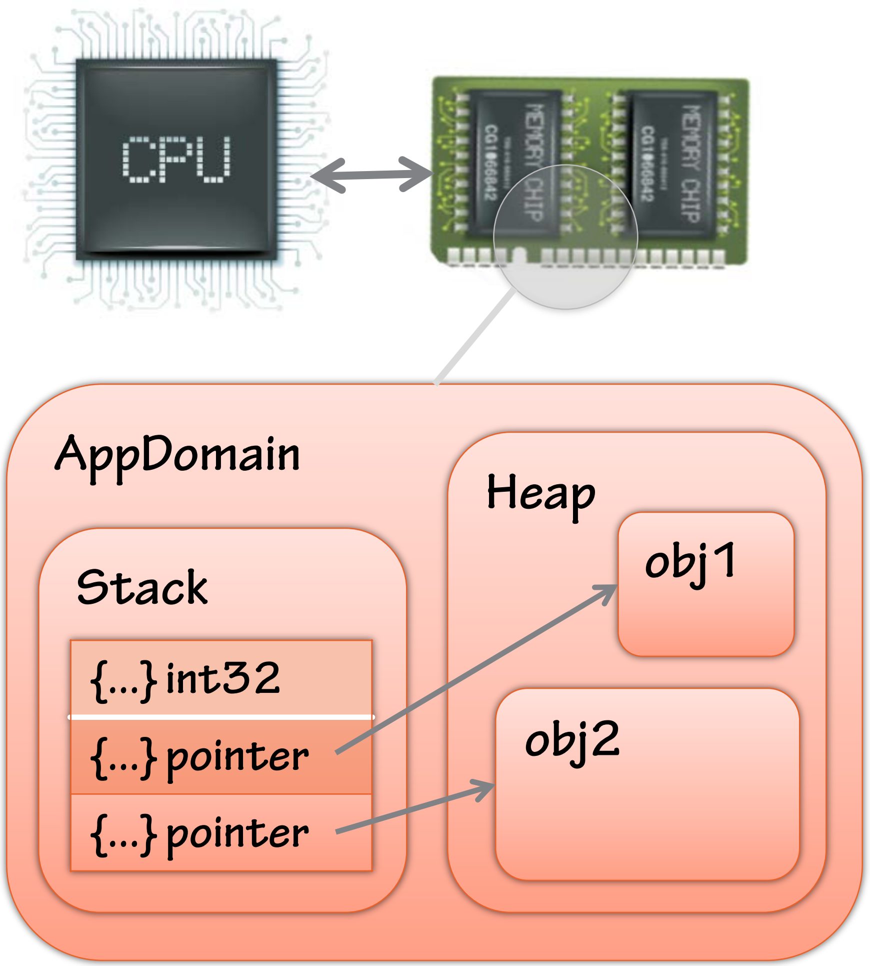
Introducing IDisposable



Process allocates memory

Runs processing

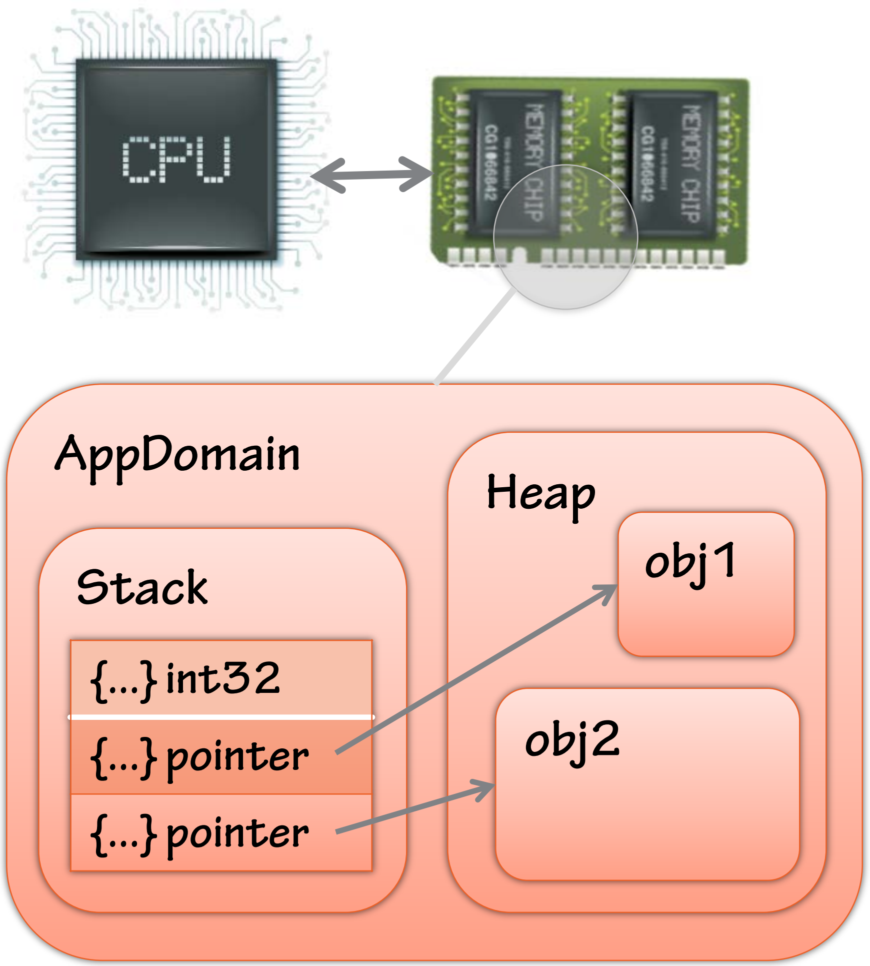
Introducing IDisposable



Process allocates memory

Runs processing

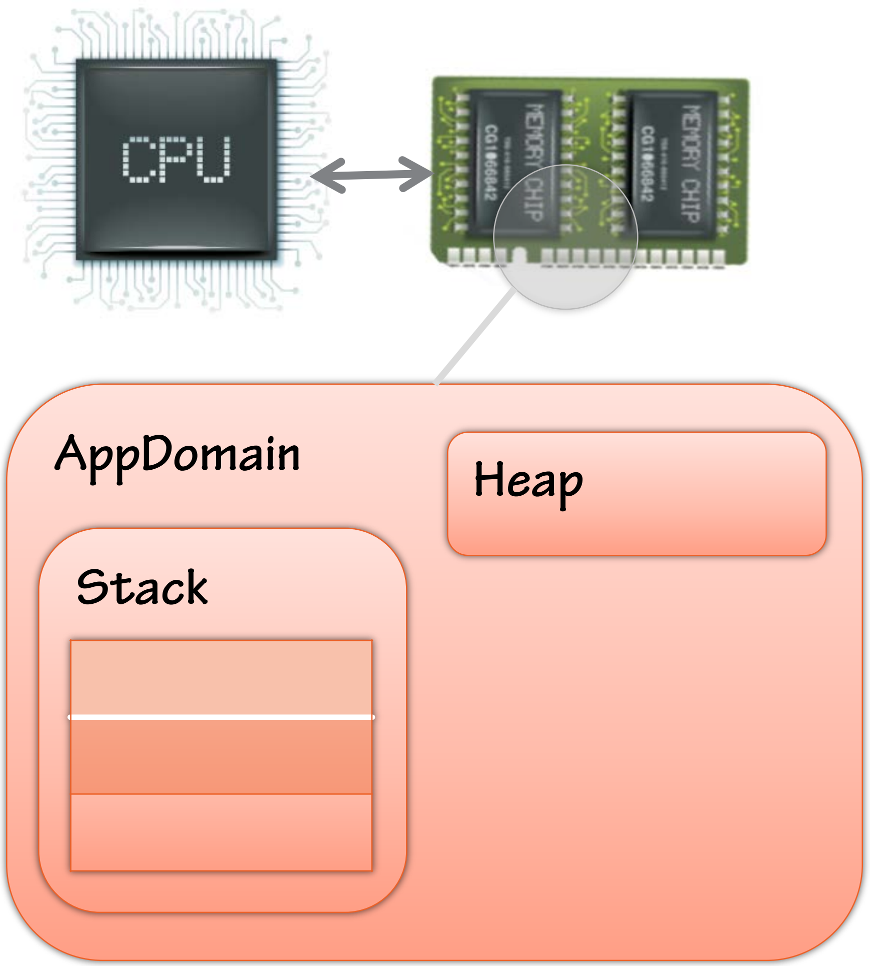
Introducing IDisposable



Process allocates memory

Runs processing

Introducing IDisposable

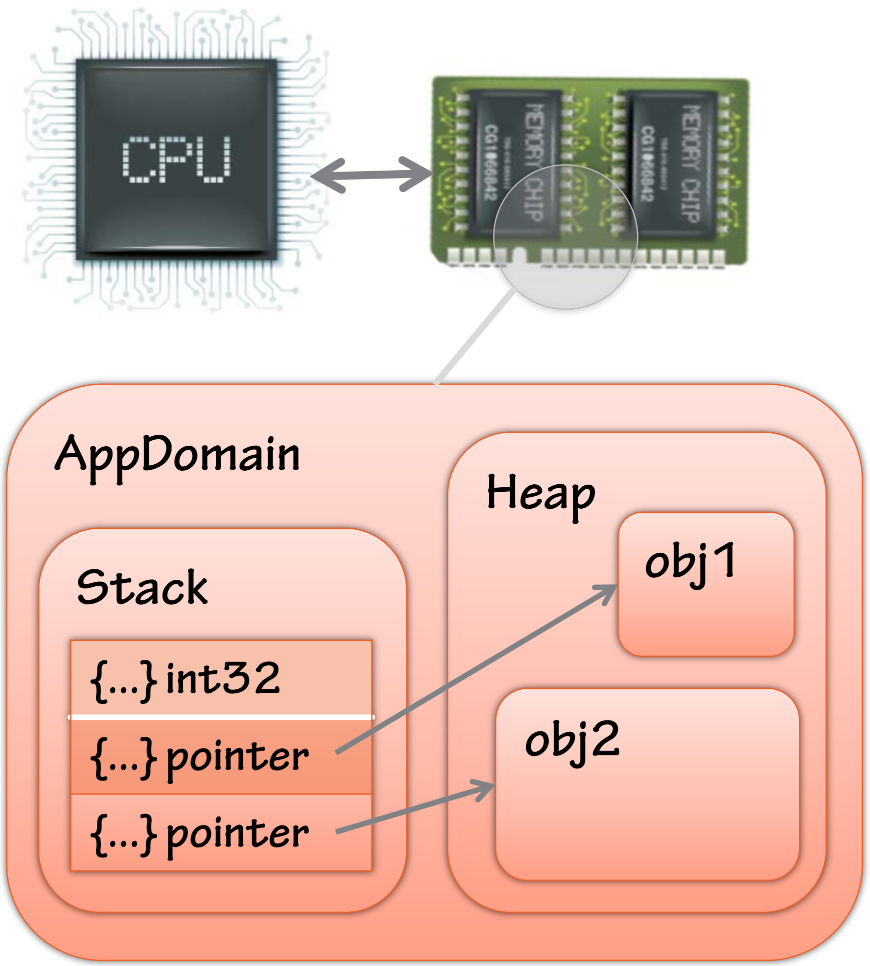


Process allocates memory

Runs processing

Releases memory

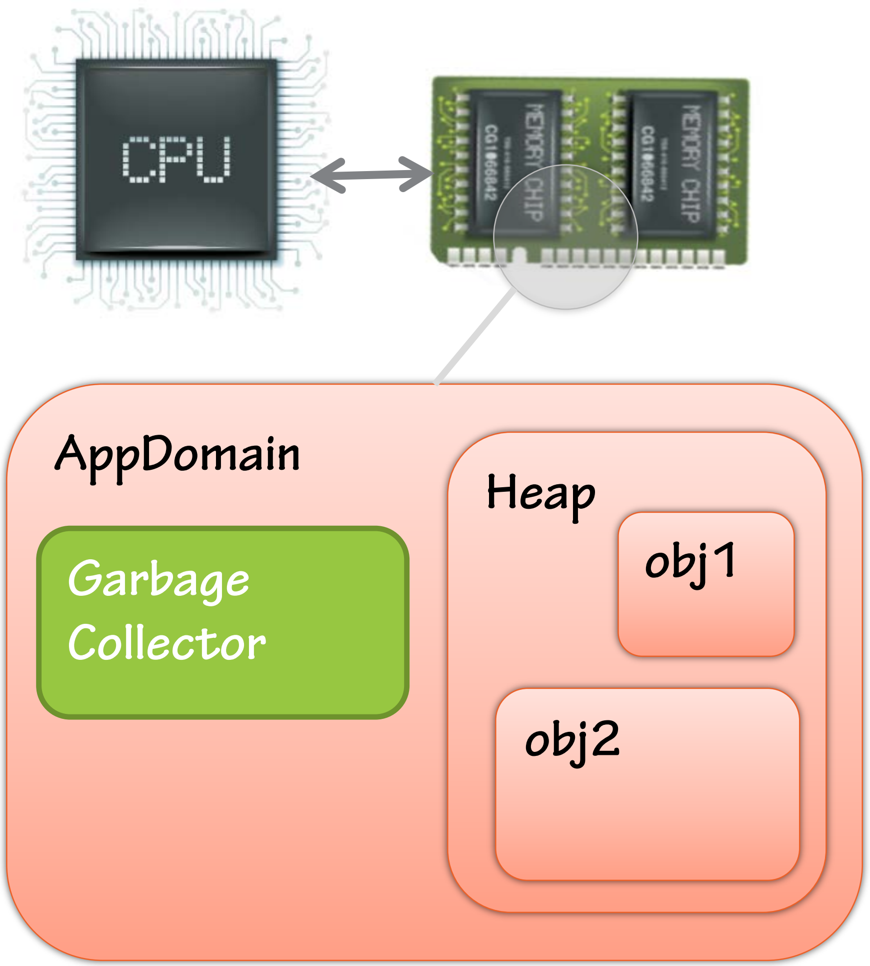
Introducing IDisposable



Process allocates memory

Runs processing

Introducing IDisposable

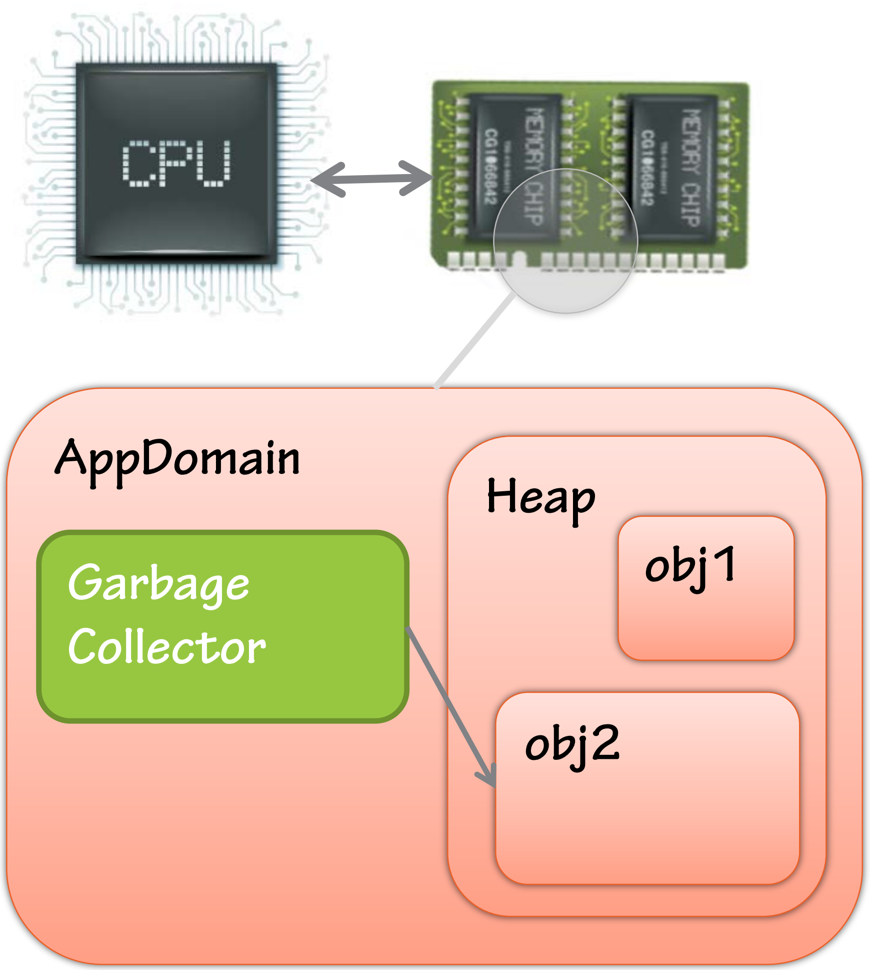


Process allocates memory

Runs processing

Garbage collector

Introducing IDisposable

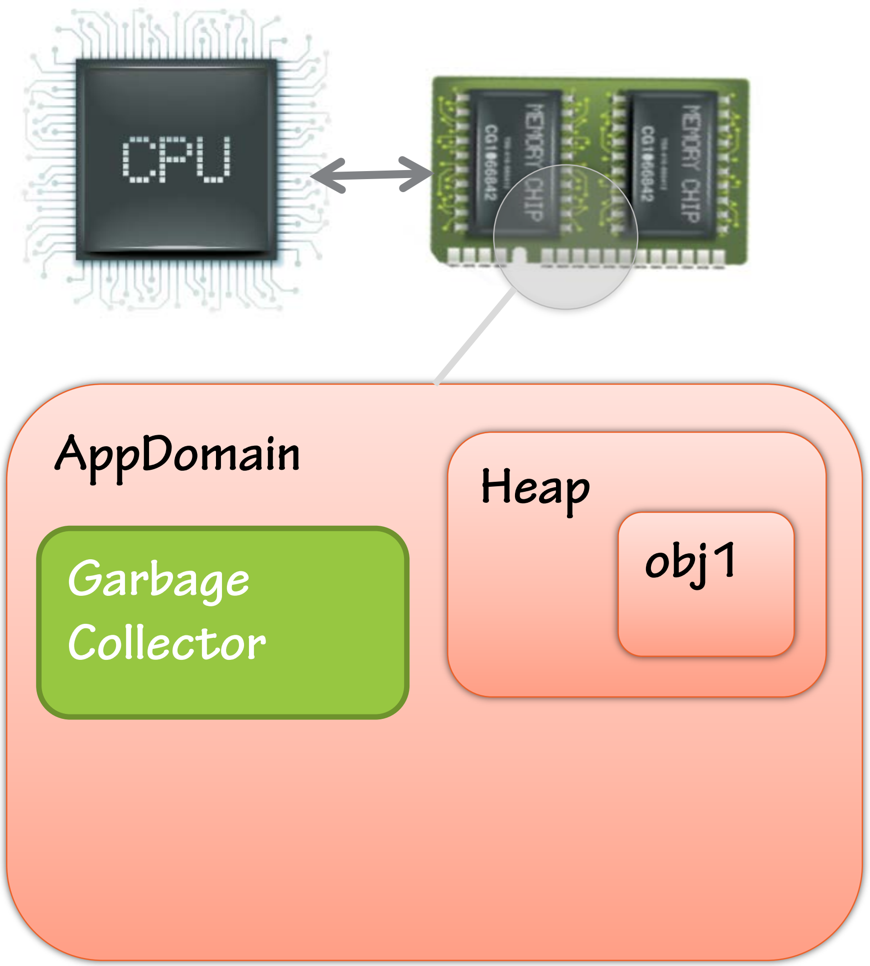


Process allocates memory

Runs processing

Garbage collector

Introducing IDisposable



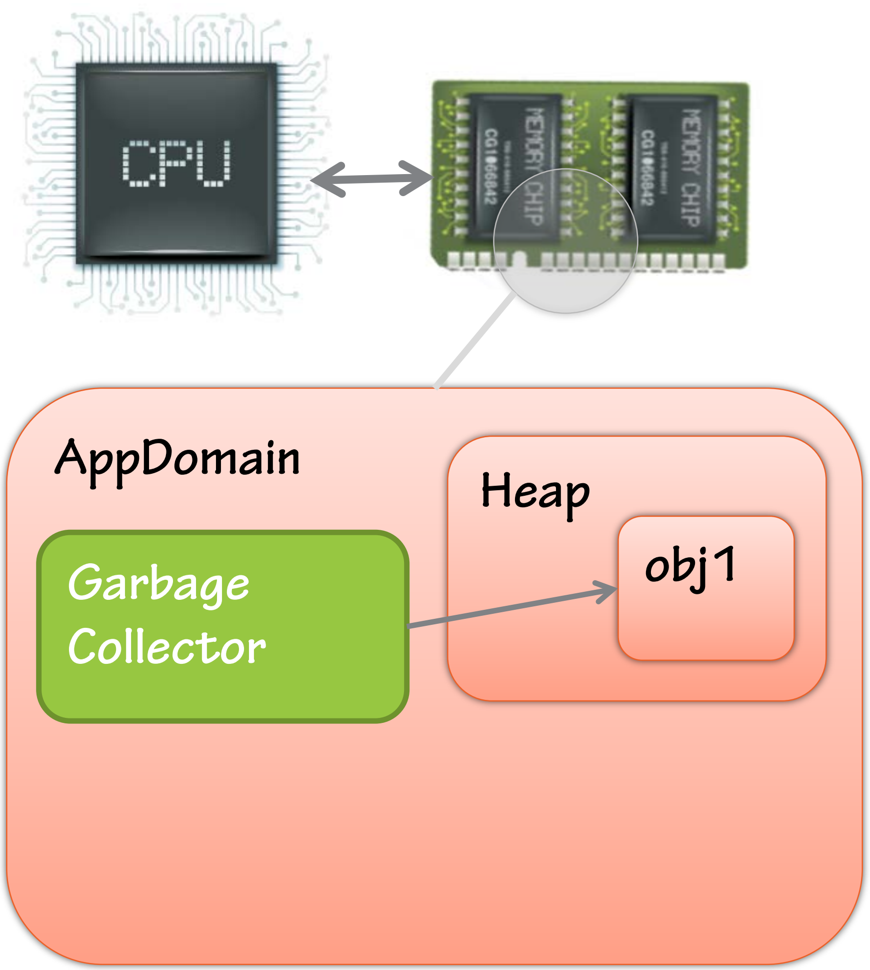
Process allocates memory

Runs processing

Garbage collector

Releases memory

Introducing IDisposable



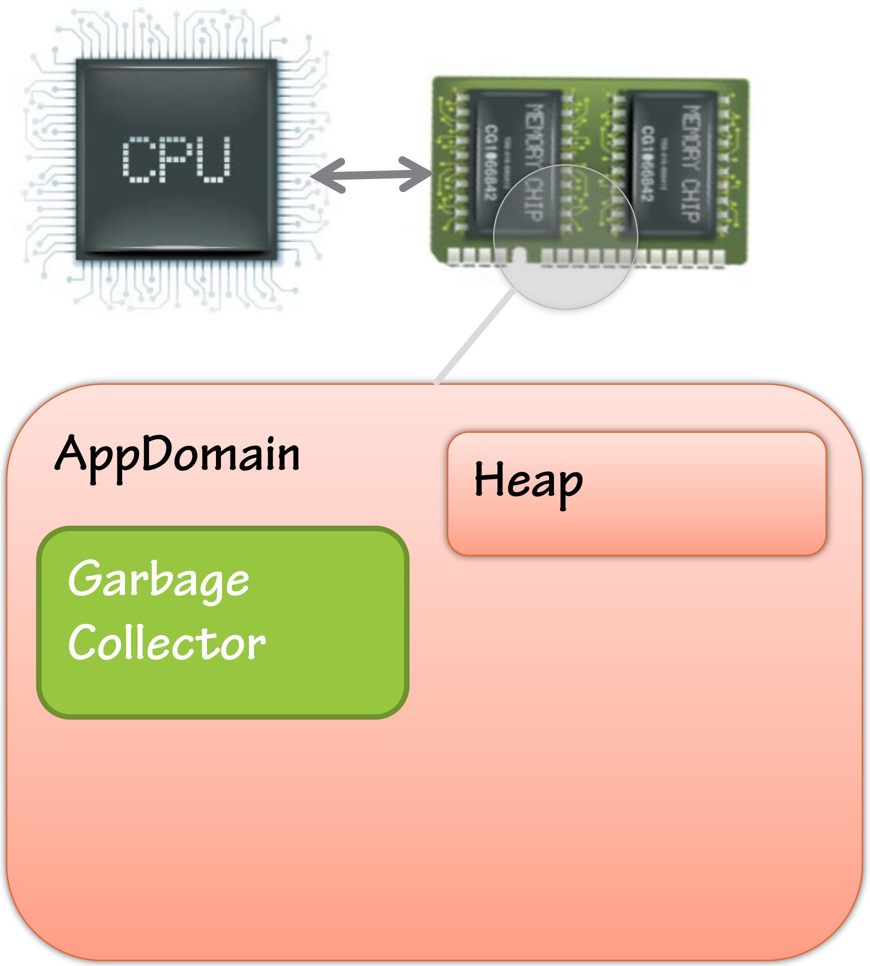
Process allocates memory

Runs processing

Garbage collector

Releases memory

Introducing IDisposable



Process allocates memory

Runs processing

Garbage collector

Releases memory

Not Using IDisposable?

6

Expanding **memory profile**

w

Hogging **external resources**

b

Functional **defects**

Introducing IDisposable

IDisposable **interface**

Using and **implementing** IDisposable

Demo with **file IO,** **SQL** and **WCF**

Course Outline

Introducing IDisposable

What Happens when the GC Runs?

What Happens if you don't Dispose?

Summary

Simple Interfaces

```
public interface IDontDoAnything
{
}
```

```
public class DoesntDoAnything : IDontDoAnything
{
}
```

Simple Interfaces

```
public interface IDoOneThing
{
    void DoTheThing();
}
```

```
public class DoesOneThng : IDoOneThing
{
    public void DoTheThing()
    {
    }
}
```

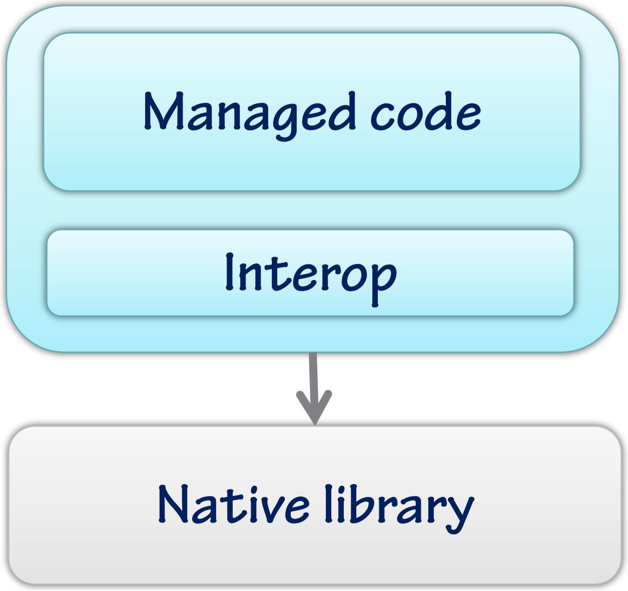
Simple Interfaces

```
namespace System
{
    public interface IDisposable
    {
        void Dispose();
    }
}
```

IDisposable

provides a mechanism for releasing unmanaged resources

Unmanaged Resources



Runtime-callable wrapper

External COM library

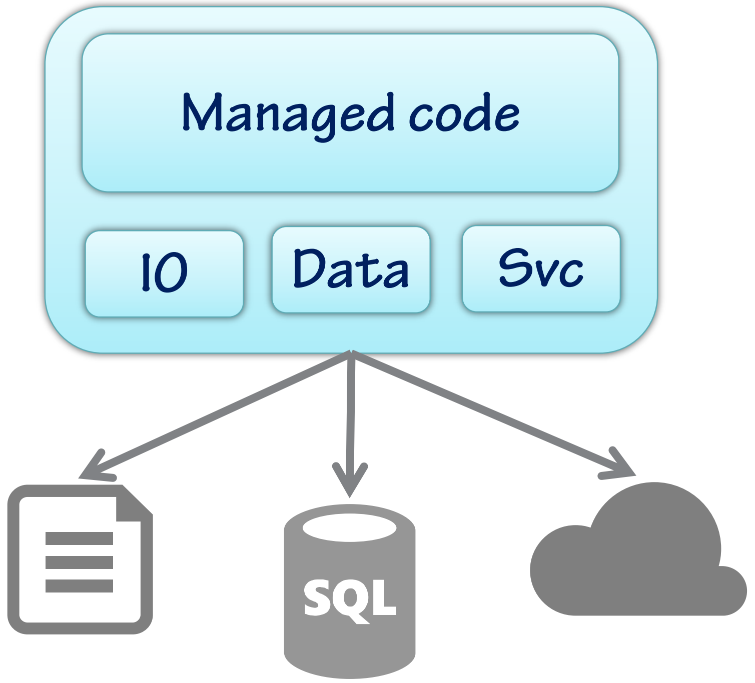
Explicitly unmanaged

DllImport

PInvoke

IntPtr

Unmanaged Resources



Managed access

External resources

Implicitly unmanaged

System.IO

System.Data

System.ServiceModel

Using IDisposable

```
public class MayUseUnmanagedResources : IDisposable
{
    public void Method() { /* ... */ }

    public void Dispose() { /* ... */ }
}
```

```
static void Main()
{
    using (var obj = new MayUseUnmanagedResources())
    {
        obj.Method();
    }
}
```

Demo 1: SqlConnection

Demo 1: SqlConnection

```
public class DatabaseState : IDisposable
{
    private SqlConnection _connection;

    /* ... */
}
```

```
using (var state = new DatabaseState())
{
      state.GetDate().Dump();
}
```

Demo 1: SqlConnection

```
public class DatabaseState : IDisposable
{
    private SqlConnection _connection;

    /* ... */

    public void Dispose()
    {
        _connection.Close();
        _connection.Dispose();
    }
}
```

Demo 1: SqlConnection

```
using (var command = _connection.CreateCommand())
{
    command.CommandText = "SELECT getdate()";
    return command.ExecuteScalar().ToString();
}
```

Best Practice #1

Dispose of IDisposable objects as soon as you can

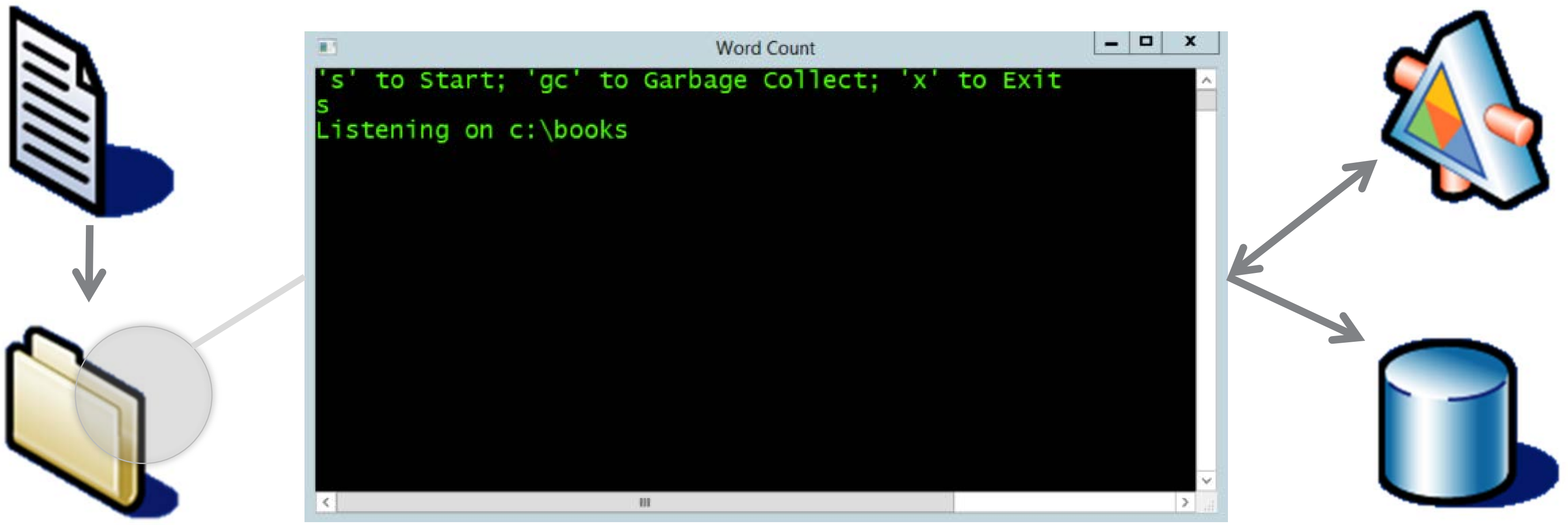
Best Practice #1

```
using (var state = new DatabaseState())
{
    state.GetDate().Dump();
}
```

```
var state = new DatabaseState();
try
{
    state.GetDate().Dump();
}
finally
{
    state.Dispose();
}
```

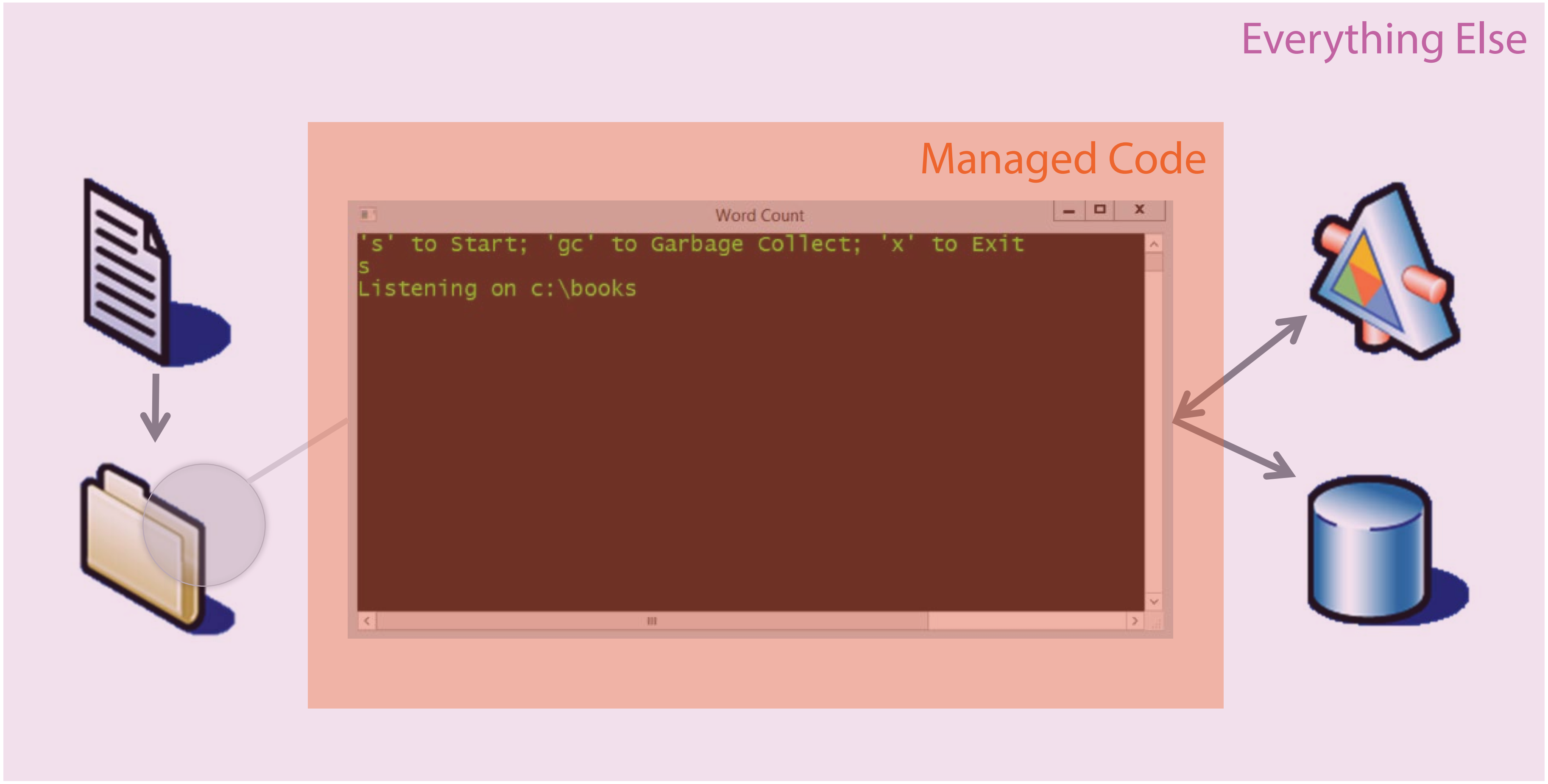
The Word Counting App

$\Sigma(\mathrm{B}) = 40{,}000$ words



```
's' to Start; 'gc' to Garbage Collect; 'x' to Exit
s
Listening on c:\books
```

The Word Counting App



```
's' to Start; 'gc' to Garbage Collect; 'x' to Exit
s
Listening on c:\books
```

Demo 2: The Word Counting App

Feature

Compute number of words in text file & store count in database

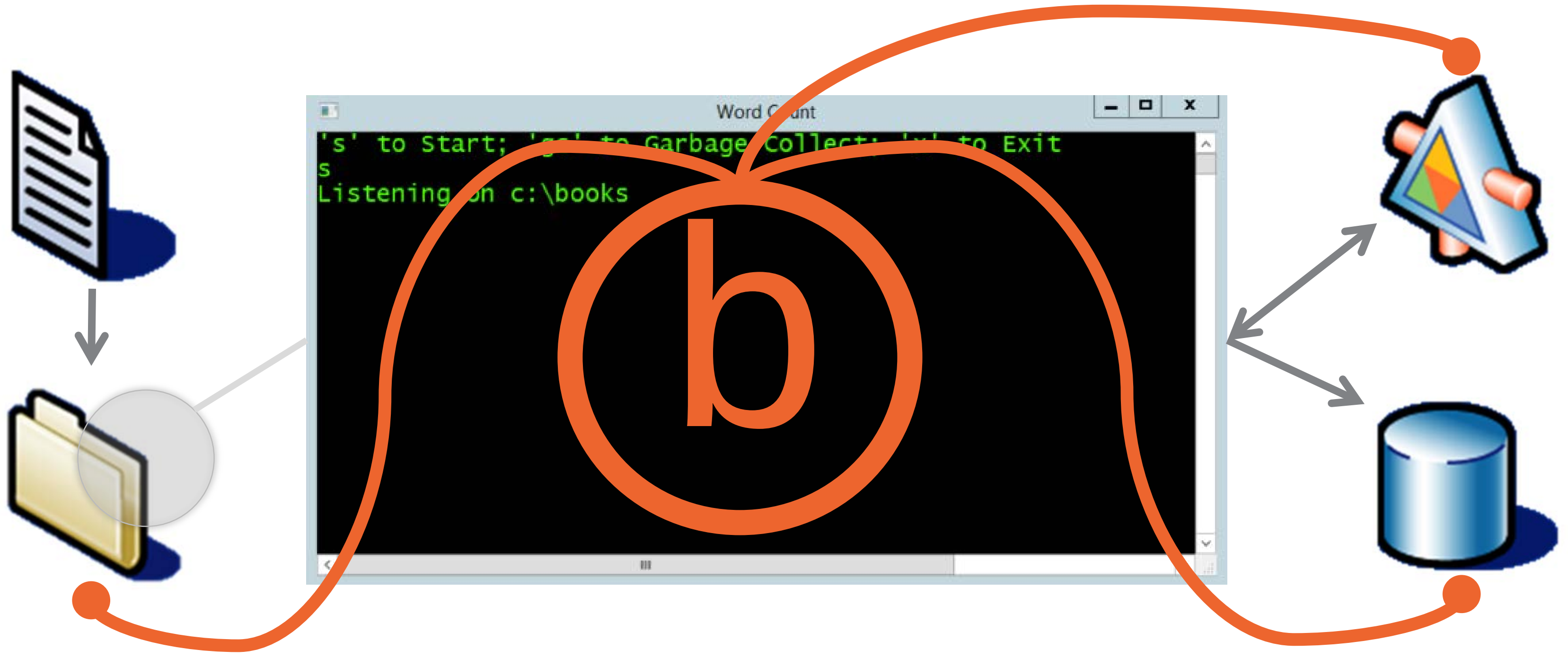
Walkthrough

Verify file drops have word count persisted

Walkthrough

App failing because of neglecting IDisposable

Demo 2: The Word Counting App



```
's' to Start; 'gc' to Garbage Collect; 'x' to Exit
s
Listening on c:\books
```

Summary

- **Introducing IDisposable** ✓
  - Interface definition
  - Meaning
- **Unmanaged resources** ✓
  - Native code– IntPtr & interop
  - ***And*** managed code
    - Which implements IDisposable
    - May or may not use unmanaged resources
- **Demo solution** ✓
  - File IO, SQL and WCF

What Happens when the GC Runs?