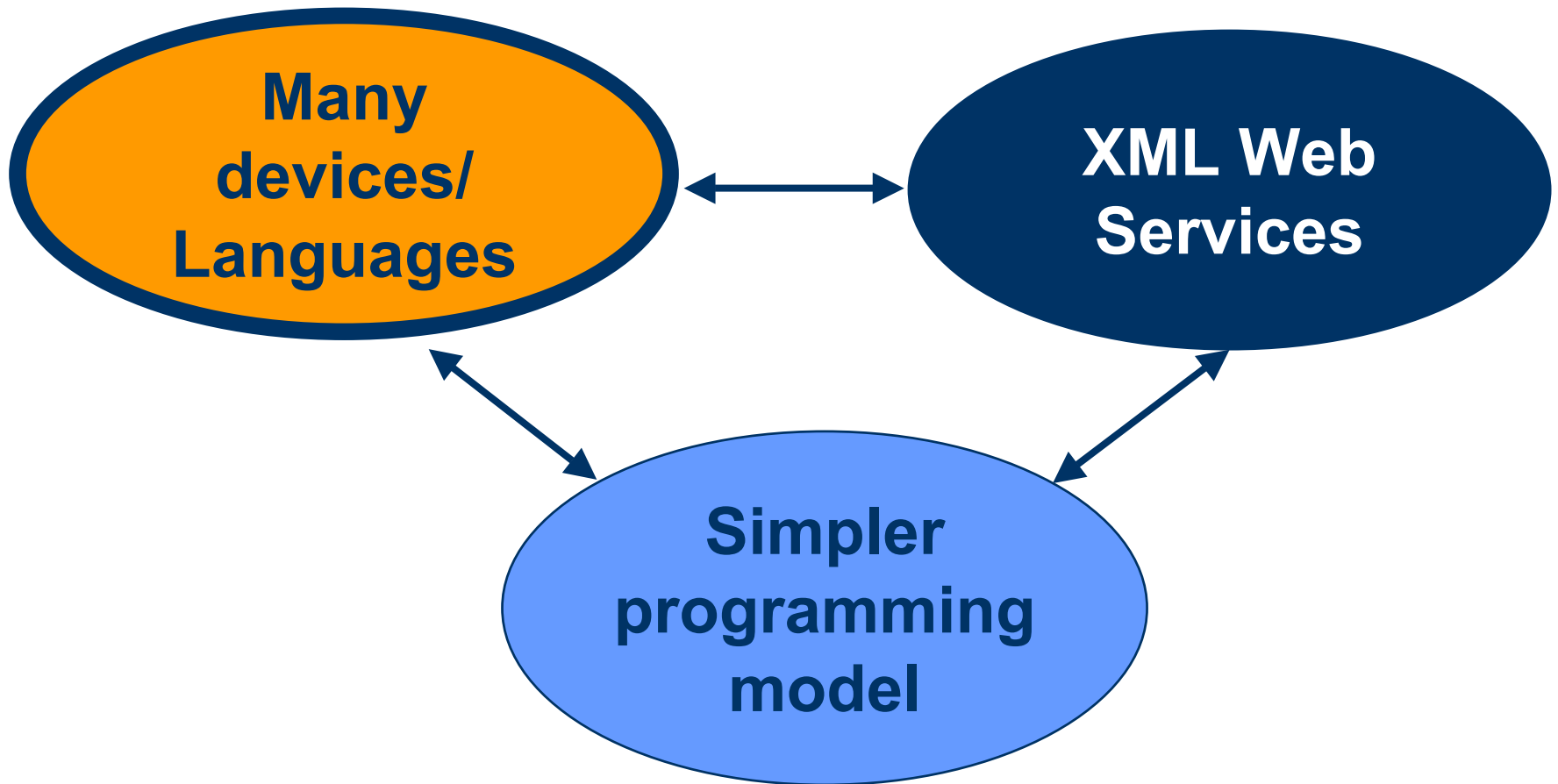




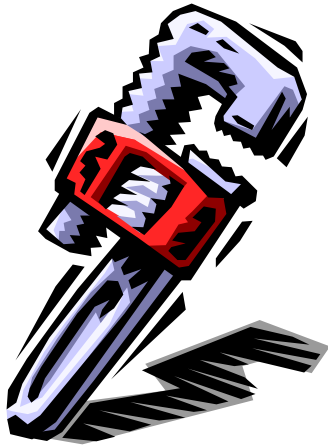
# A Programmer's view on .NET



# Three Pillars of .NET



# The Marketeer's Toolbox



**Product  
Wrench**



**Competition  
Hammer**



**Price  
Axe**

# The Programmer's Toolbox



**C# Bricks**



**COBOL  
Pictures**



**Perl  
Duct Tape**



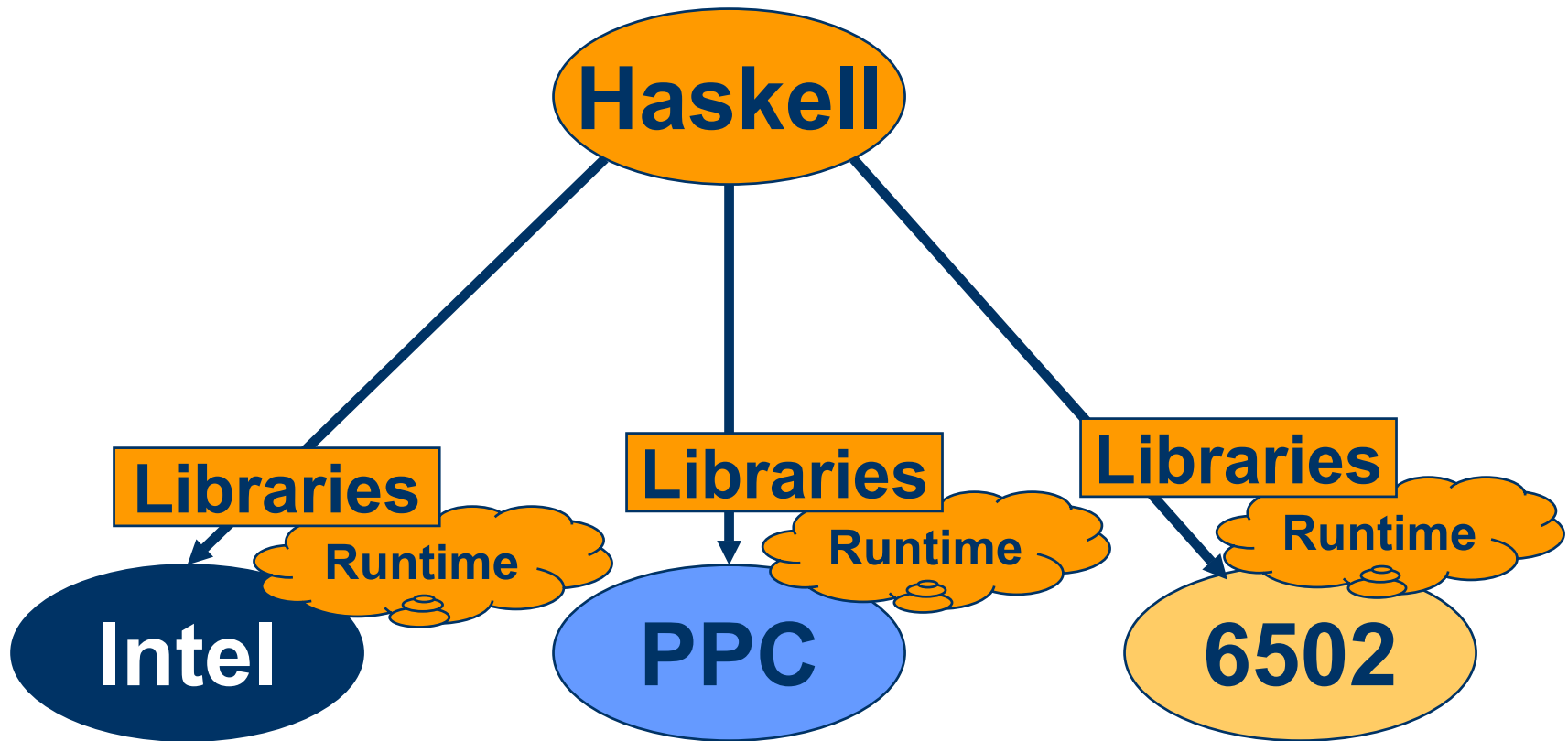
**Haskell  
Super Glue**

# dumpbin /disasm user32.dll

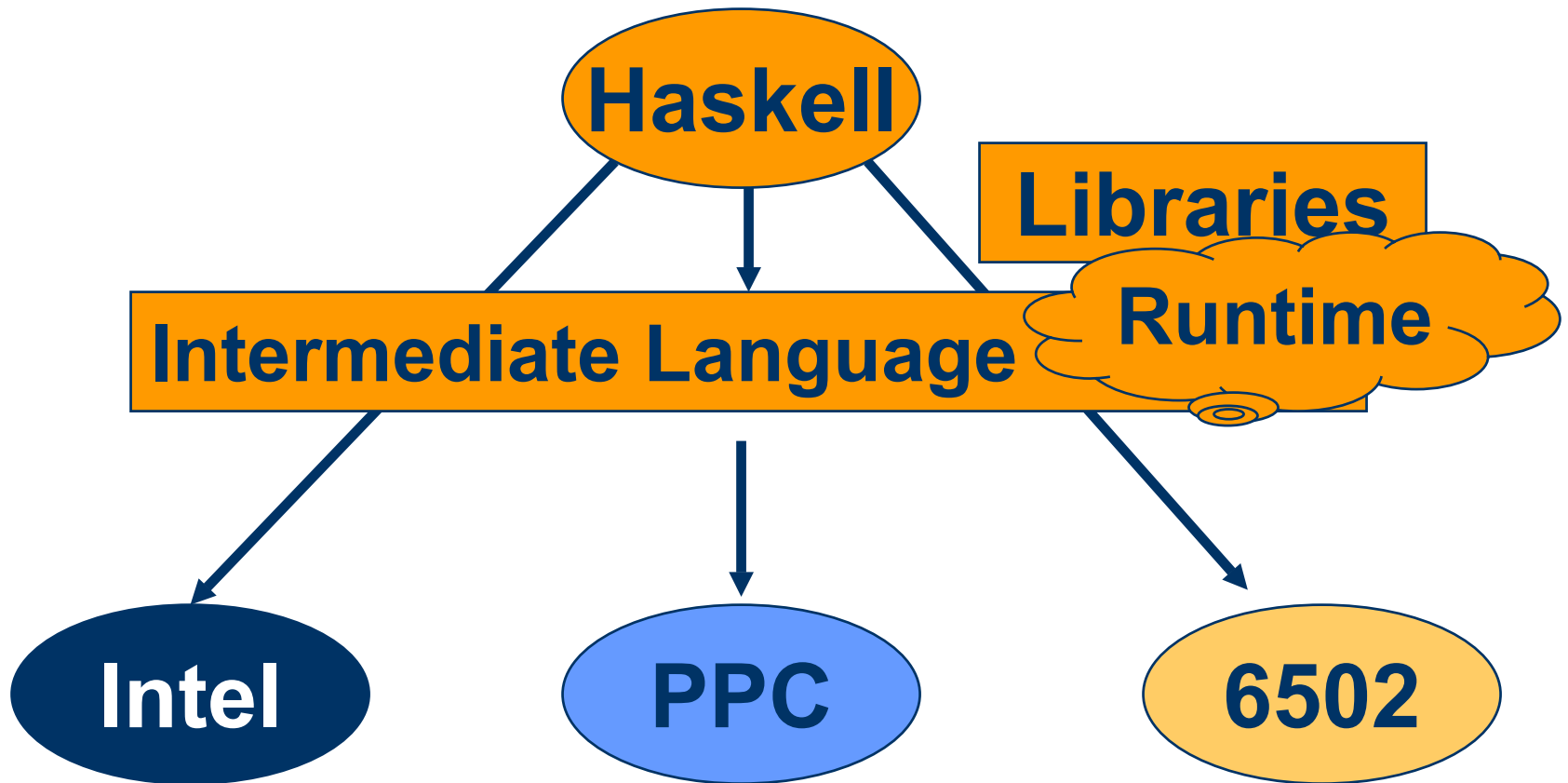
```
C:\WINNT\System32\cmd.exe
77E17D78: B8 64 11 00 00    mov     eax,1164h
77E17D7D: 8D 54 24 04      lea    edx,[esp+4]
77E17D81: CD 2E           int    2Eh
77E17D83: C2 0C 00       ret    0Ch
77E17D86: 55            push   ebp
77E17D87: 8B EC         mov    ebp,esp
77E17D89: 81 EC 08 02 00 00 sub    esp,208h
77E17D8F: 56            push   esi
77E17D90: 8B 75 10      mov    esi,[ebp+10h]
77E17D93: 85 F6        test   esi,esi
77E17D95: 74 17        je     77E17D97h
77E17D97: 8D 85 F8 FD FF FF lea    esi,[ebp+10h]
77E17D9D: 68 04 01 00 00  push  esi
77E17DA2: 50            push   esi
77E17DA3: 56            push   esi
77E17DA4: FF 15 78 12 E1 77 call   dword ptr ds:[77E11278h]
77E17DAA: 85 C0        test   eax,eax
77E17DAC: 74 21        je     77E17DCFh
77E17DAE: FF 75 18      push   dword ptr [ebp+18h]
77E17DB1: 8B C6        mov    eax,esi
77E17DB3: F7 D8        neg    eax
77E17DB5: FF 75 0C      push   dword ptr [ebp+0Ch]
77E17DB8: 8D 8D F8 FD FF FF lea    ecx,[ebp+FFFFFFF8h]
77E17DBE: 1B C0        sbb   eax,eax
77E17DC0: FF 75 08      push   dword ptr [ebp+8]
```

**Machine  
Dependent**

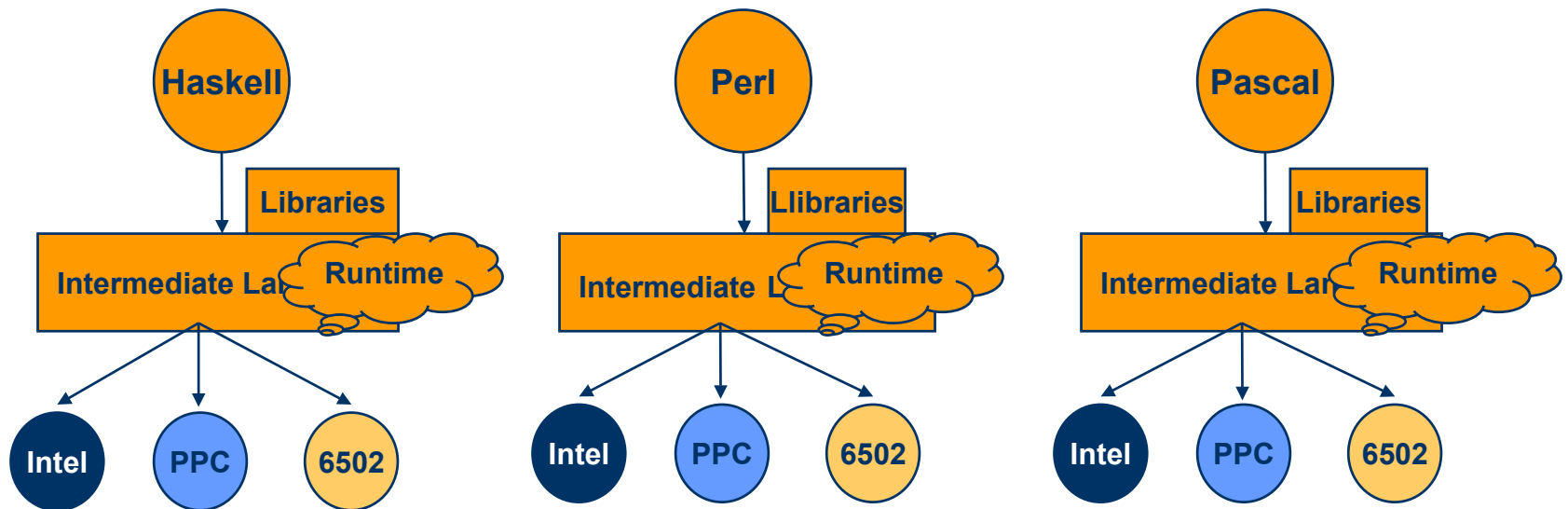
# Building a Compiler



# Building a Compiler

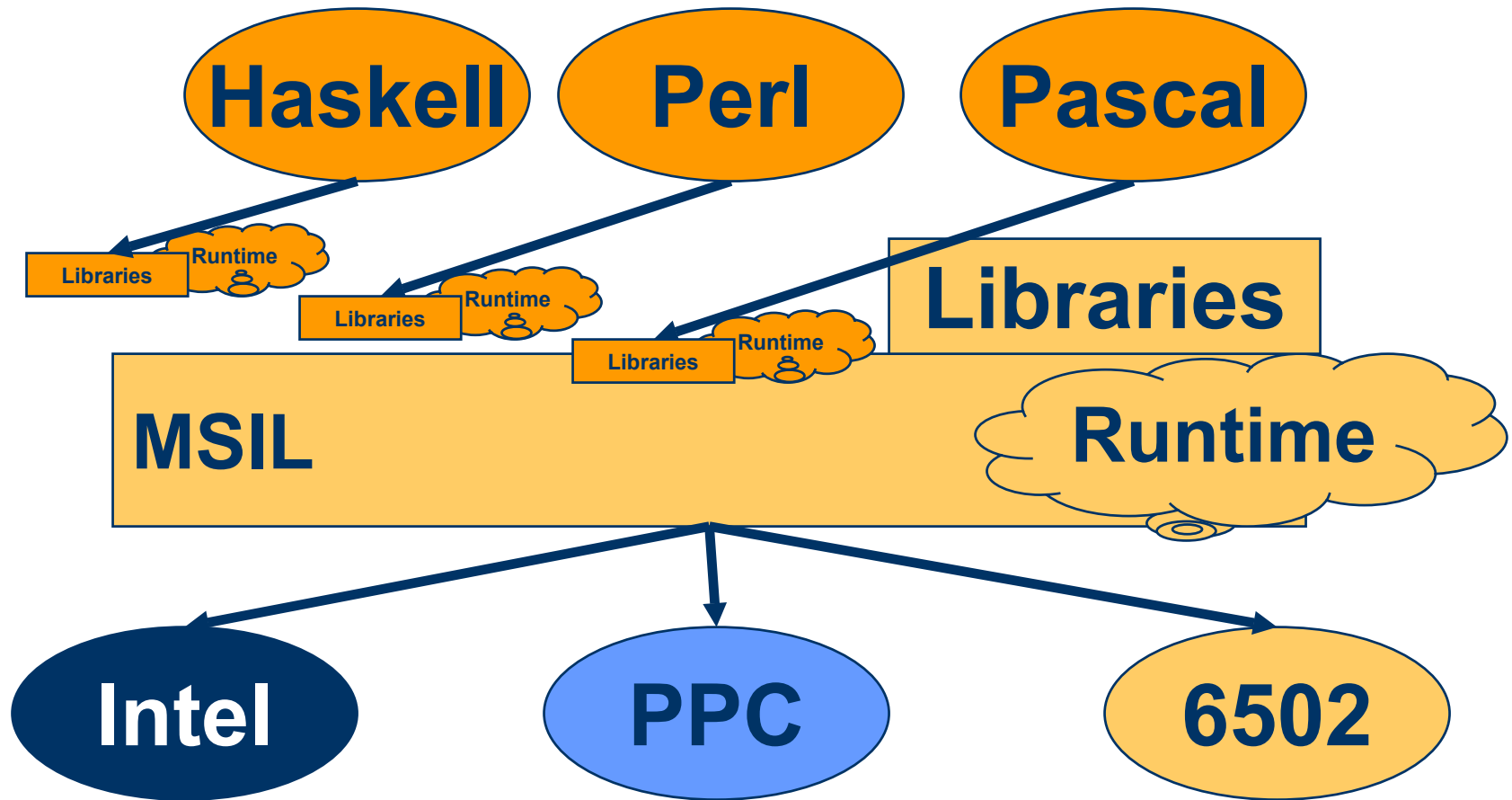


# Building a Compiler





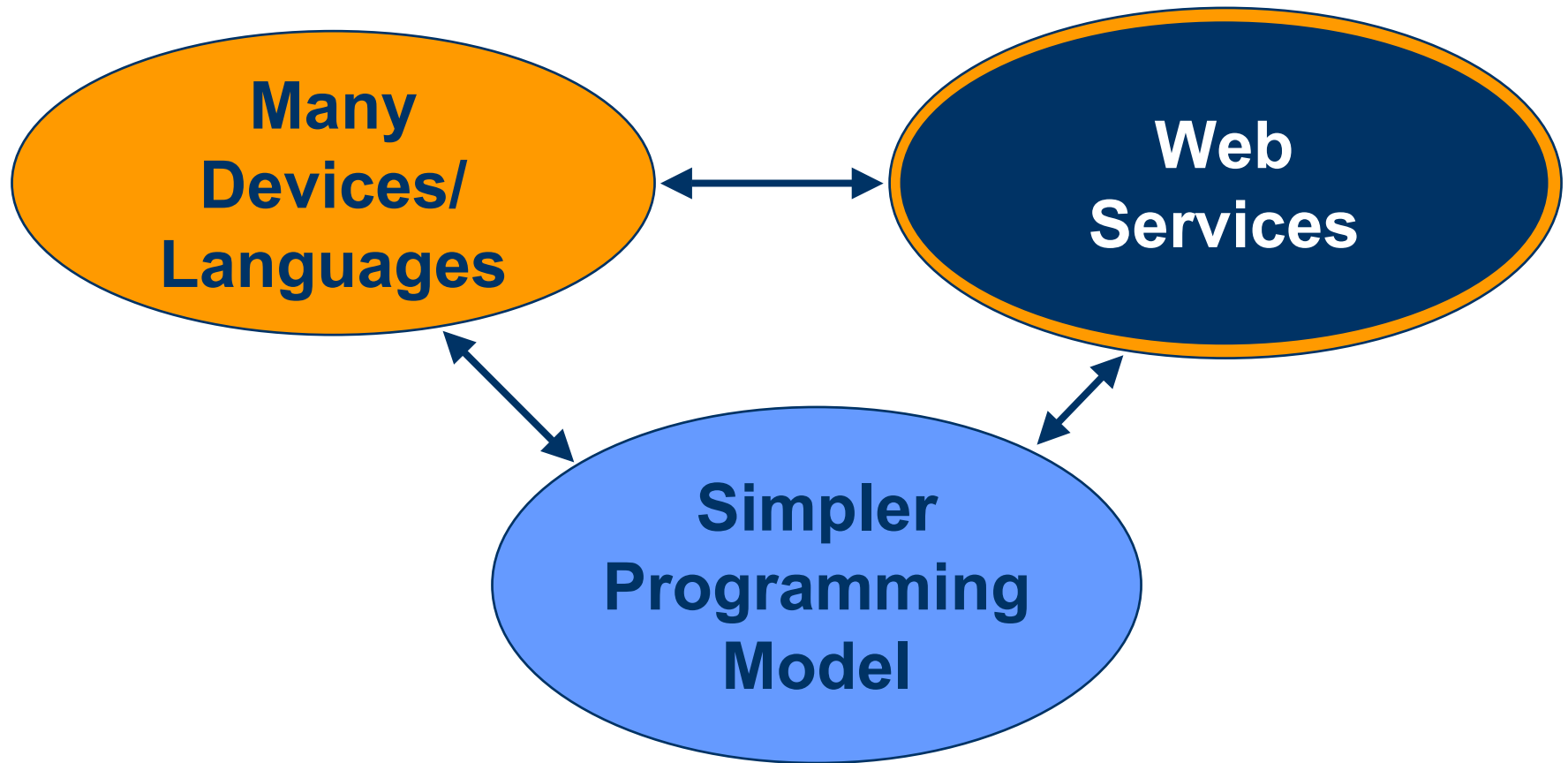
# Building a Compiler



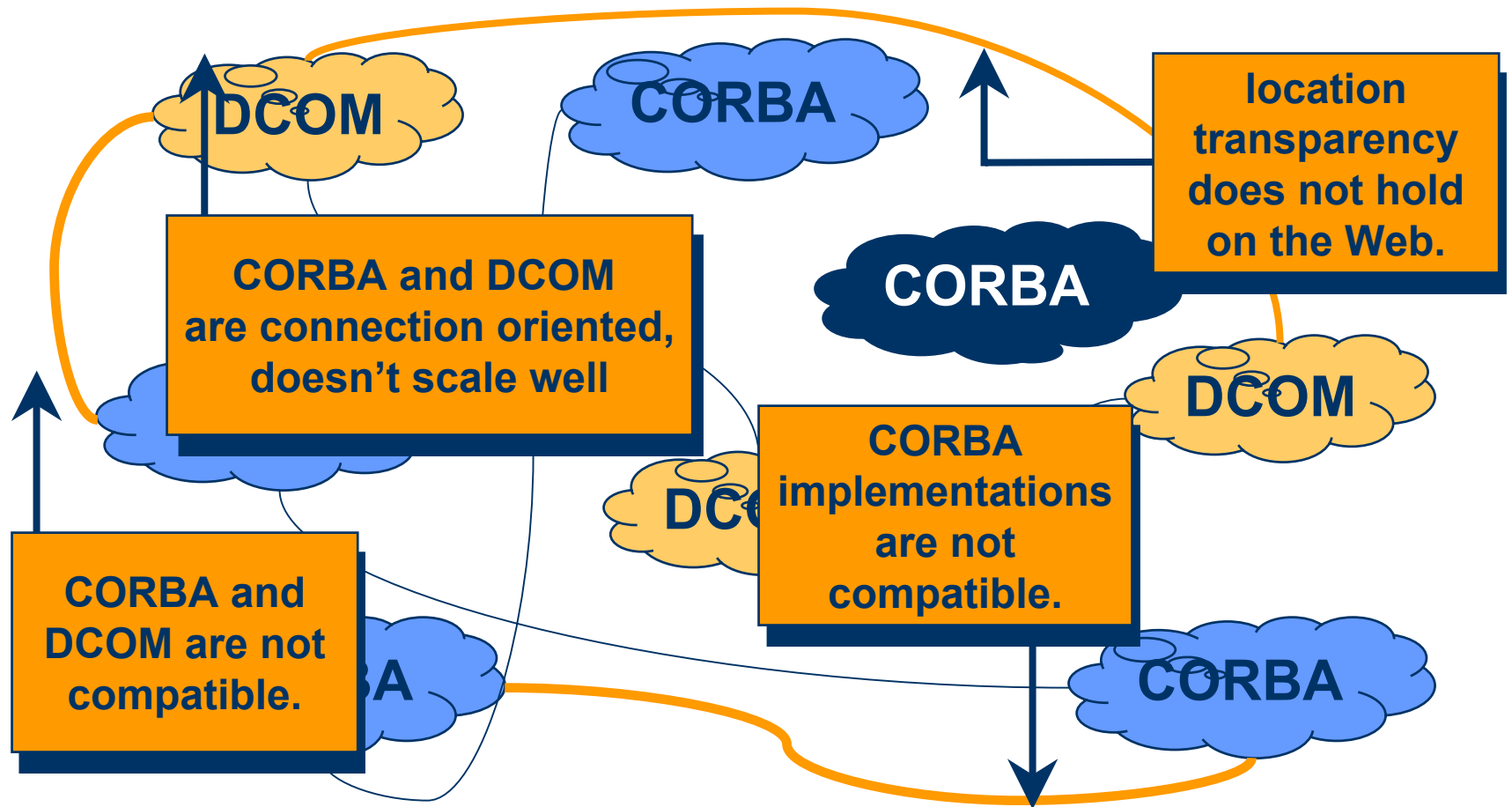
# VisualStudio.net

- ◆ "Drag and Drop" Web services development
- ◆ Fully supports .NET Framework
- ◆ Multi-language support
- ◆ Open, extensible development environment
- ◆ Easy to plug in new languages (Babel)

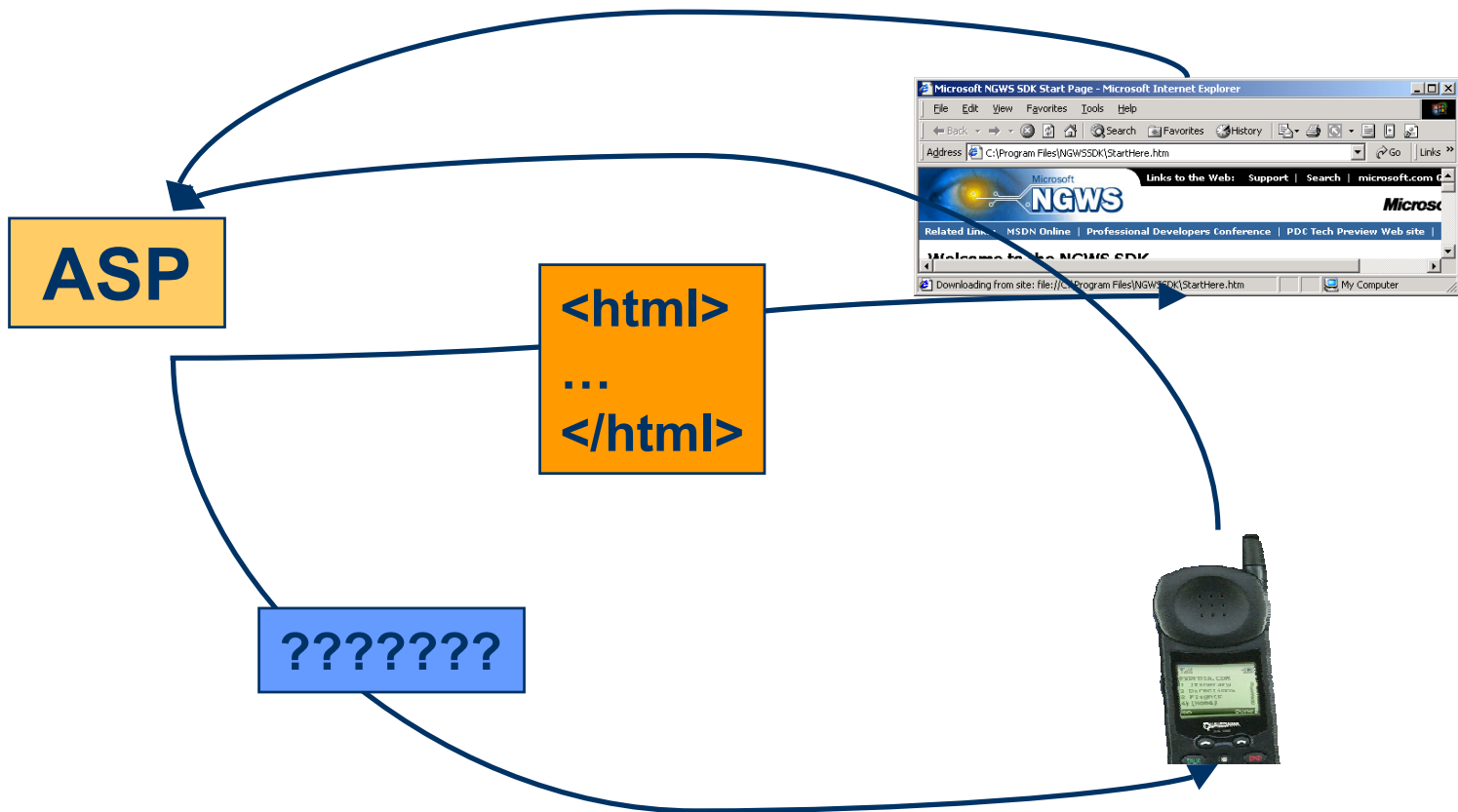
# Three Pillars of .NET



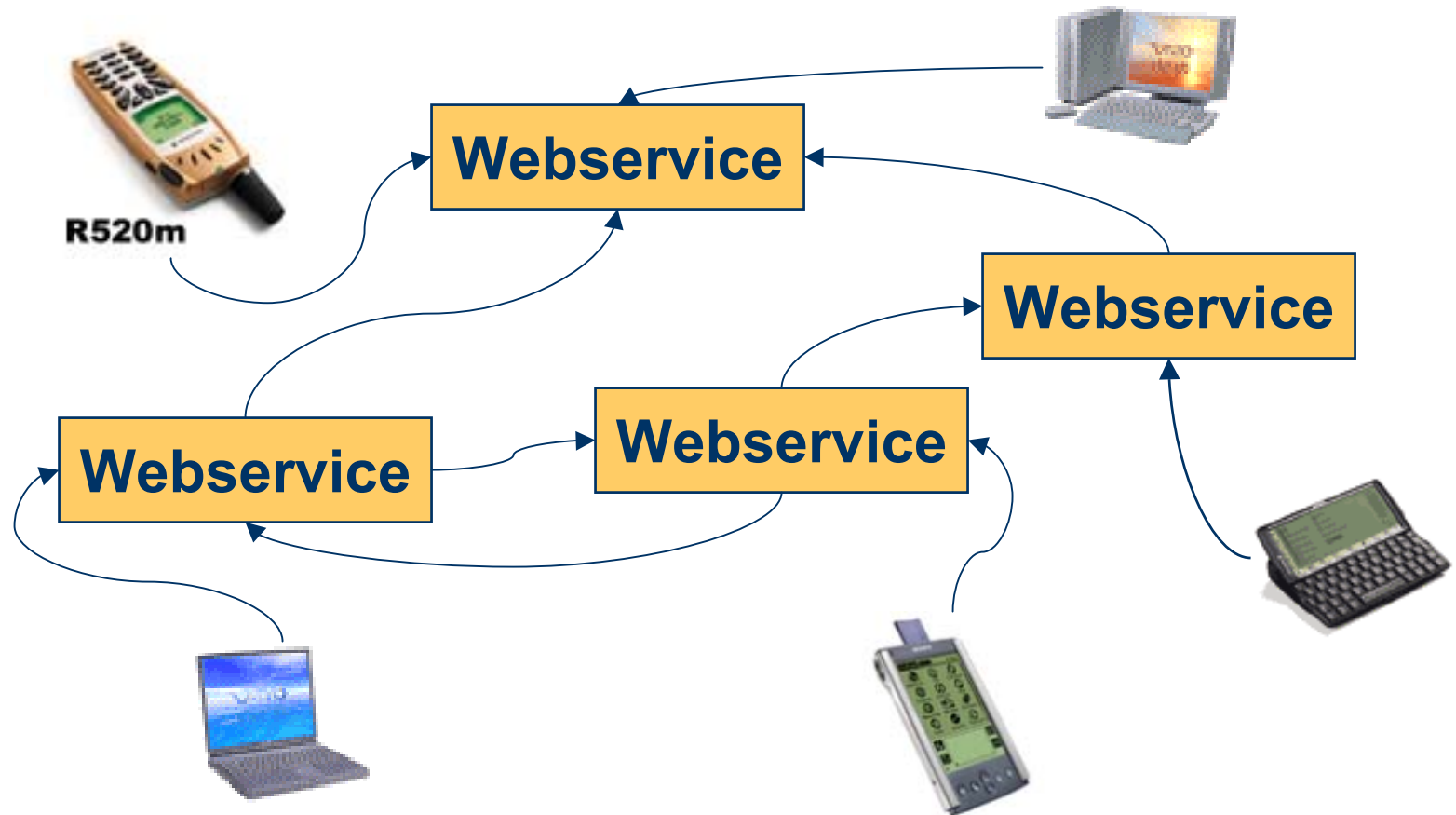
# Distributed Systems



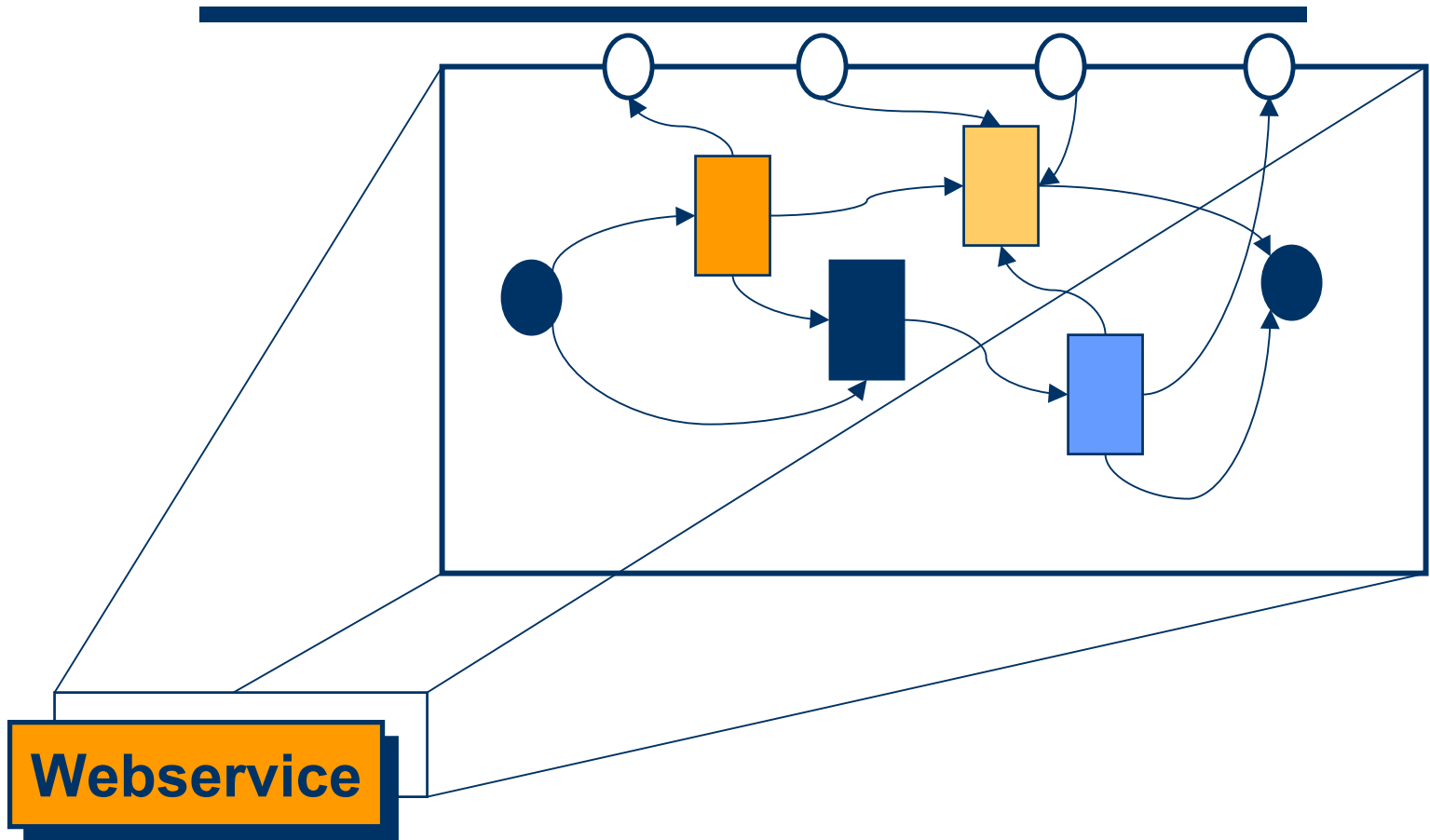
# HTTP protocol



# XML Webservice



# Orchestration



# Enterprise Servers

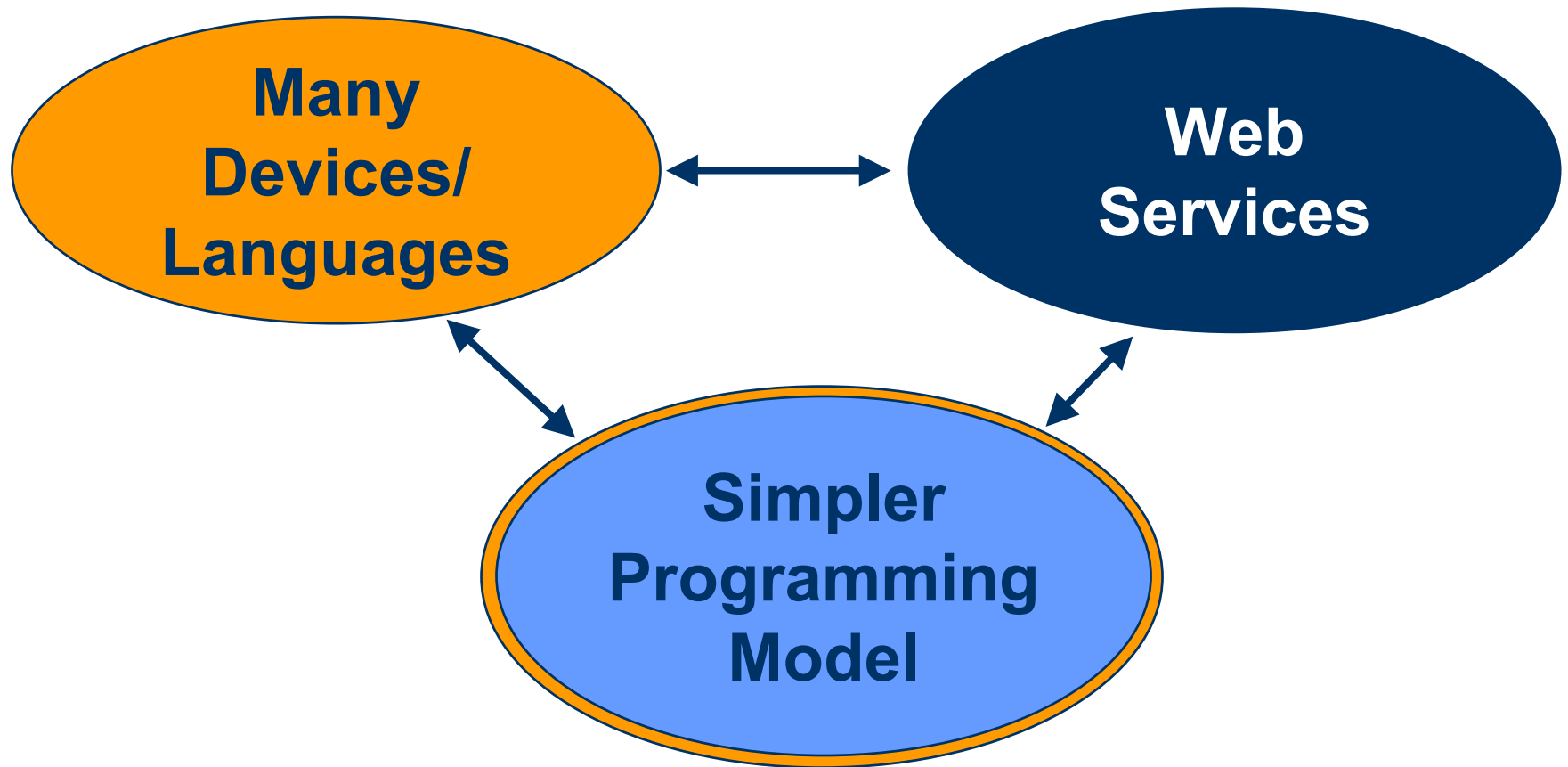
- ◆ Functionality for development and management of .NET services:
  - Application Center 2000:
    - Reliability and scalability
  - Internet Security & Acceleration Server:
    - Firewall and proxy server
  - Commerce Server:
    - Analyzing site usage
  - BizTalk Server 2000:
    - Business process orchestration and business-to-business document interchange using XML



# Enterprise Servers

- SQL Server 2000:
  - Easy-to-use database systems with native Extensible Markup Language (XML) support
- Host Integration Server 2000:
  - Integration with host systems and their data
- Mobile Information 2001 Server:
  - Integration with mobile devices

# Three Pillars of .NET



# IDL

```
.....  
void _OldSysKind();  
short MajorVersion();  
short MinorVersion();  
short AttributeMask();  
short AttributeStrings([out] SAFEARRAY(BSTR)*  
AttributeArray);  
CoClasses* CoClasses();  
Interfaces* Interfaces();  
Constants* Constants();  
Declarations* Declarations();  
short TypeInfoCount();  
void _OldGetTypeKind();  
TypeInfo* GetTypeInfo([in] VARIANT* Index);  
short GetTypeInfoNumber([in] BSTR Name);  
VARIANT_BOOL IssameLibrary([in] TypeLibInfo* checkLib);  
... .
```

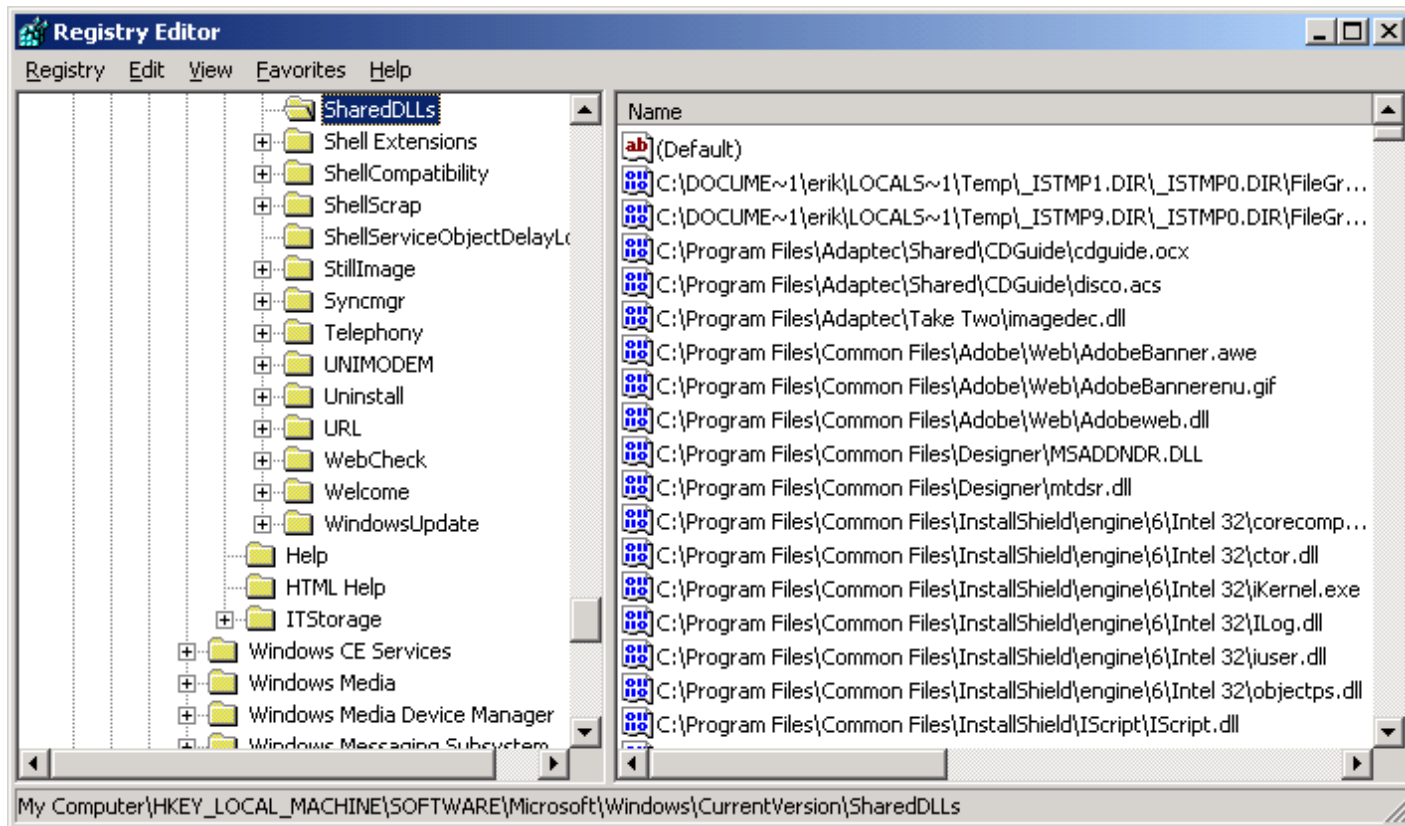
# VC++ 6

```
#import "Bar.dll"
main () {
    IFoo *b;
    CoInitialize(NULL);
    CoCreateInstance
    (__uuidof(Bar), NULL, CLSCTX_SERVER,
    __uuidof(IFoo), (void **) &b);
    .....
    .....
    .....
    .....
}
```

# C#

```
using Bar;  
  
void Main () {  
    IFoo b = new Bar();  
    .....  
    .....  
    .....  
}
```

# Don't Try This at Home!



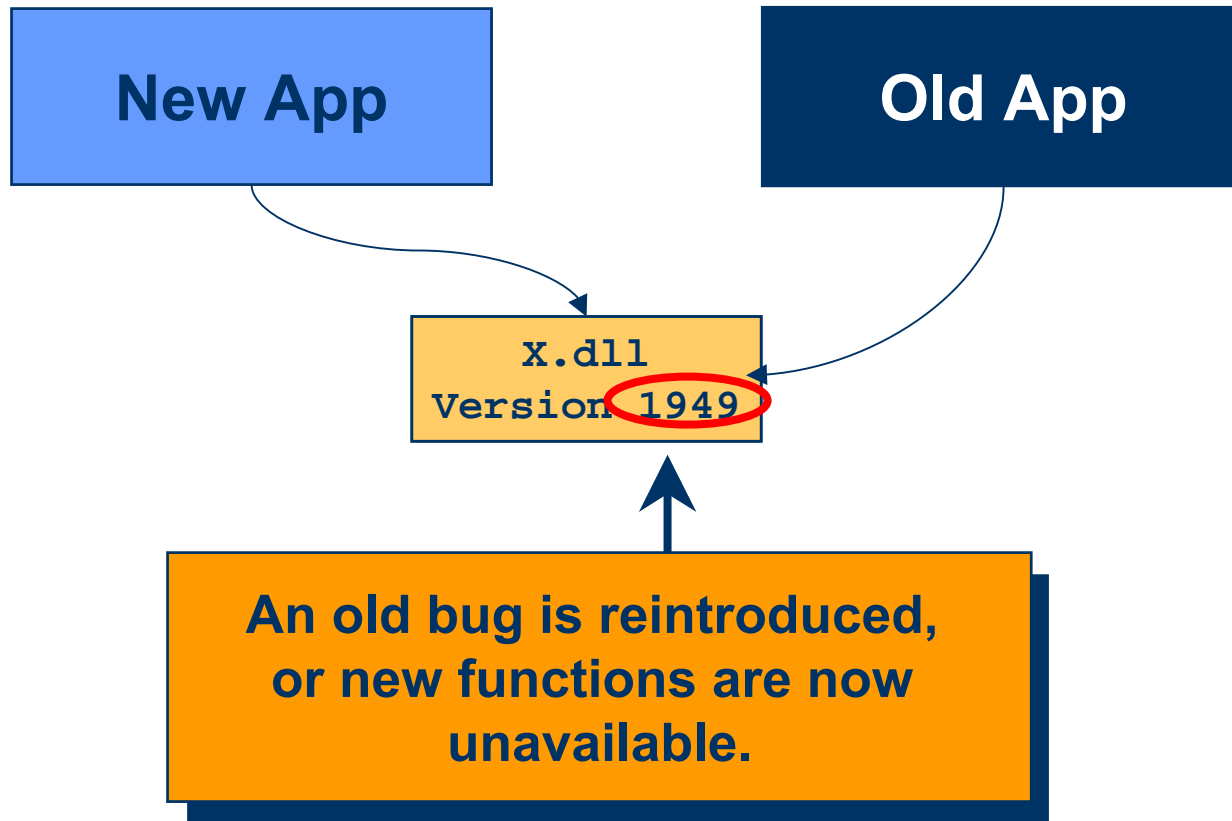
# DLL Hell

New App



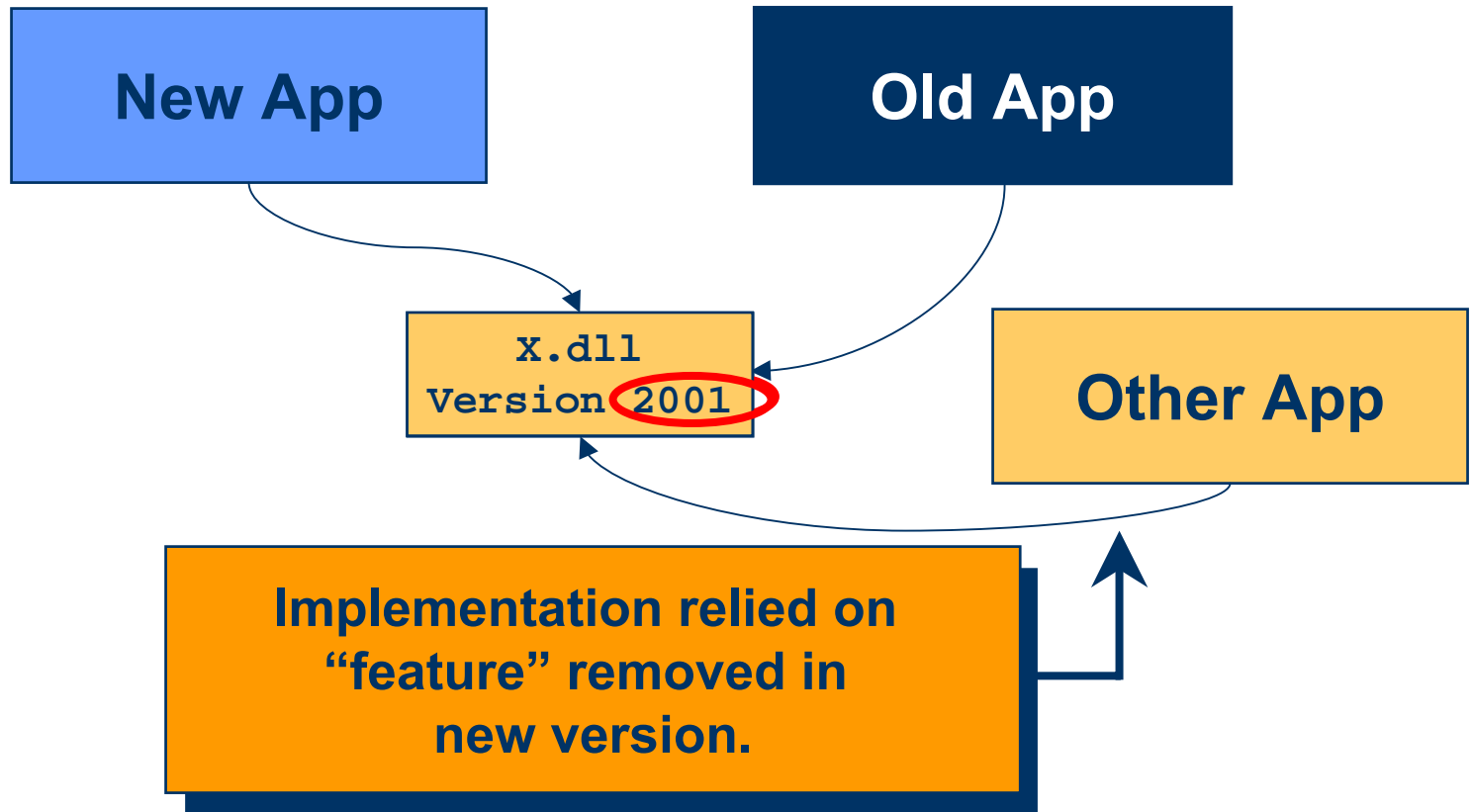
x.dll  
Version 2000

# Replacing Newer Version with Older Version





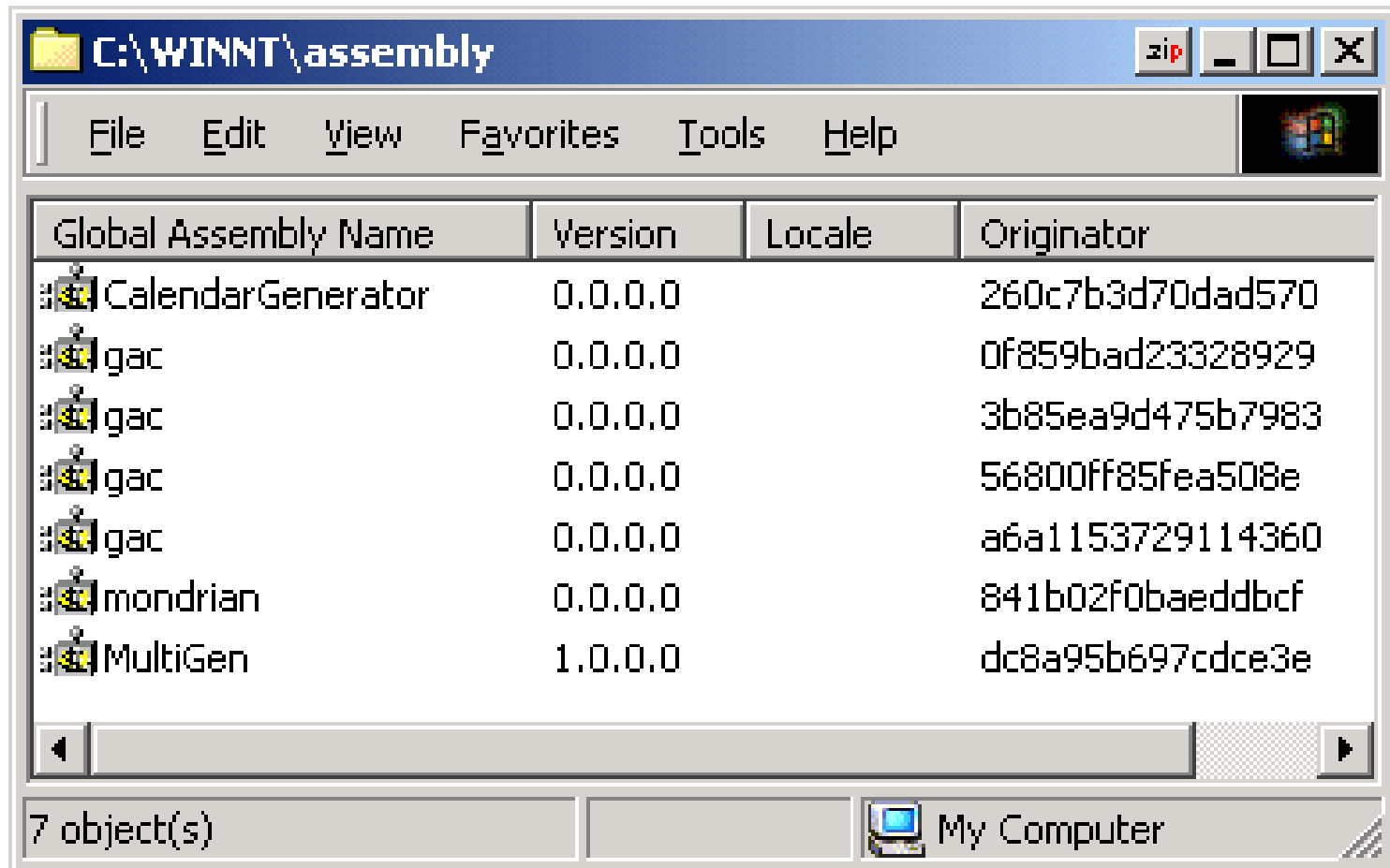
# Replacing Older Version with Newer Version



# WFP

- ◆ When a new DLL installed, W2K checks whether the changed DLL was protected.
- ◆ If the new DLL doesn't have a valid signature, W2K restores the original DLL from `winnt\system32\dllcache`.

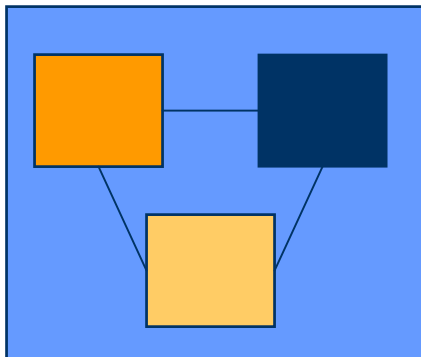
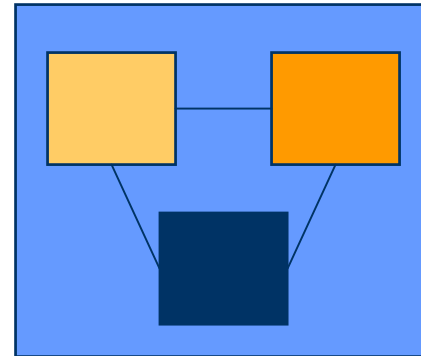
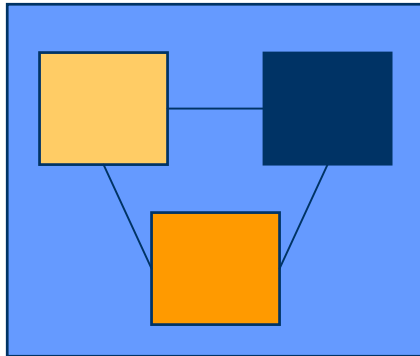
# Side-By-Side Versioning



The screenshot shows a Windows Explorer window titled "C:\WINNT\assembly" with a "zip" icon in the title bar. The menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The main pane displays a table of assembly files with columns for "Global Assembly Name", "Version", "Locale", and "Originator". The files listed are CalendarGenerator, five instances of gac, mondrian, and MultiGen. The status bar at the bottom indicates "7 object(s)" and "My Computer".

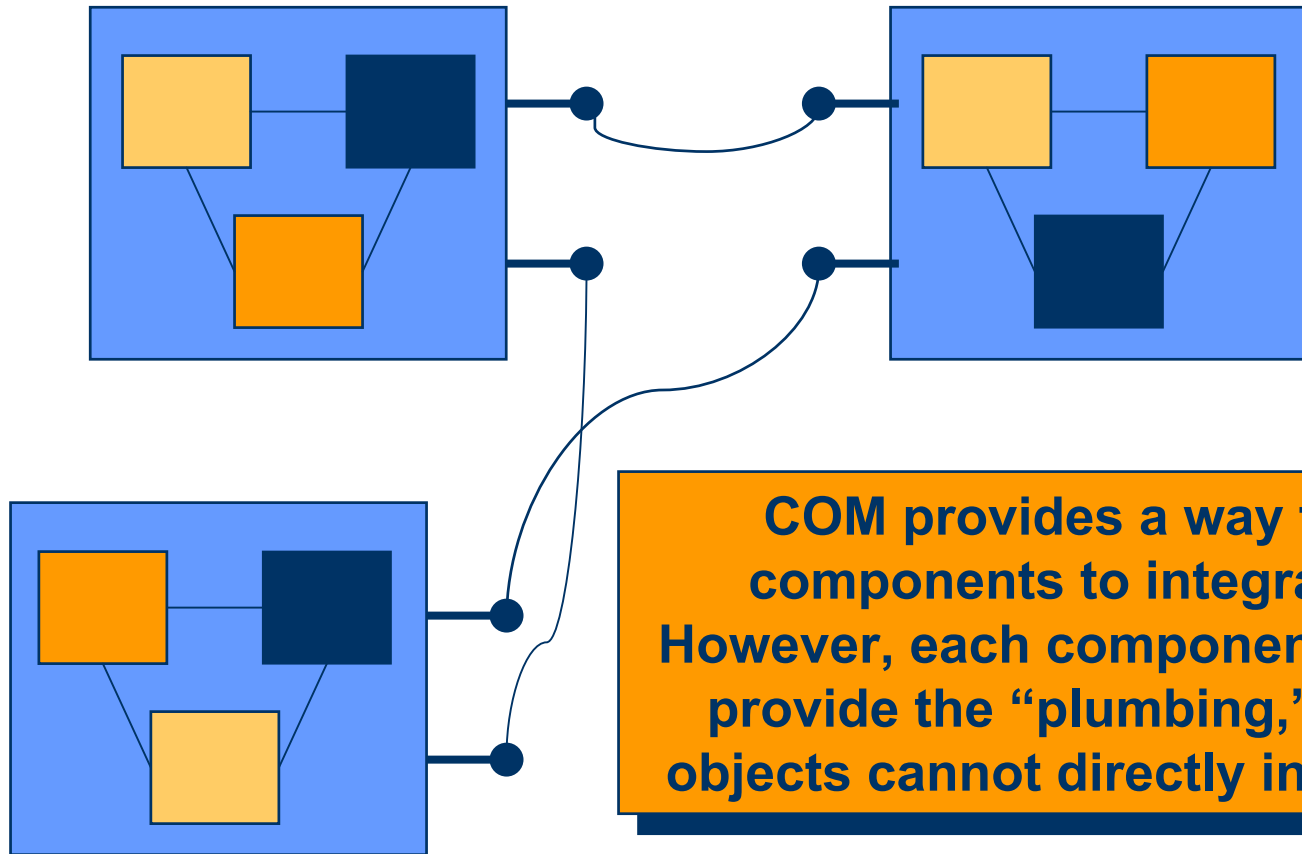
Global Assembly Name	Version	Locale	Originator
CalendarGenerator	0.0.0.0		260c7b3d70dad570
gac	0.0.0.0		0f859bad23328929
gac	0.0.0.0		3b85ea9d475b7983
gac	0.0.0.0		56800ff85fea508e
gac	0.0.0.0		a6a1153729114360
mondrian	0.0.0.0		841b02f0baeddbcf
MultiGen	1.0.0.0		dc8a95b697cdce3e

# Holy Grail of Software Reuse



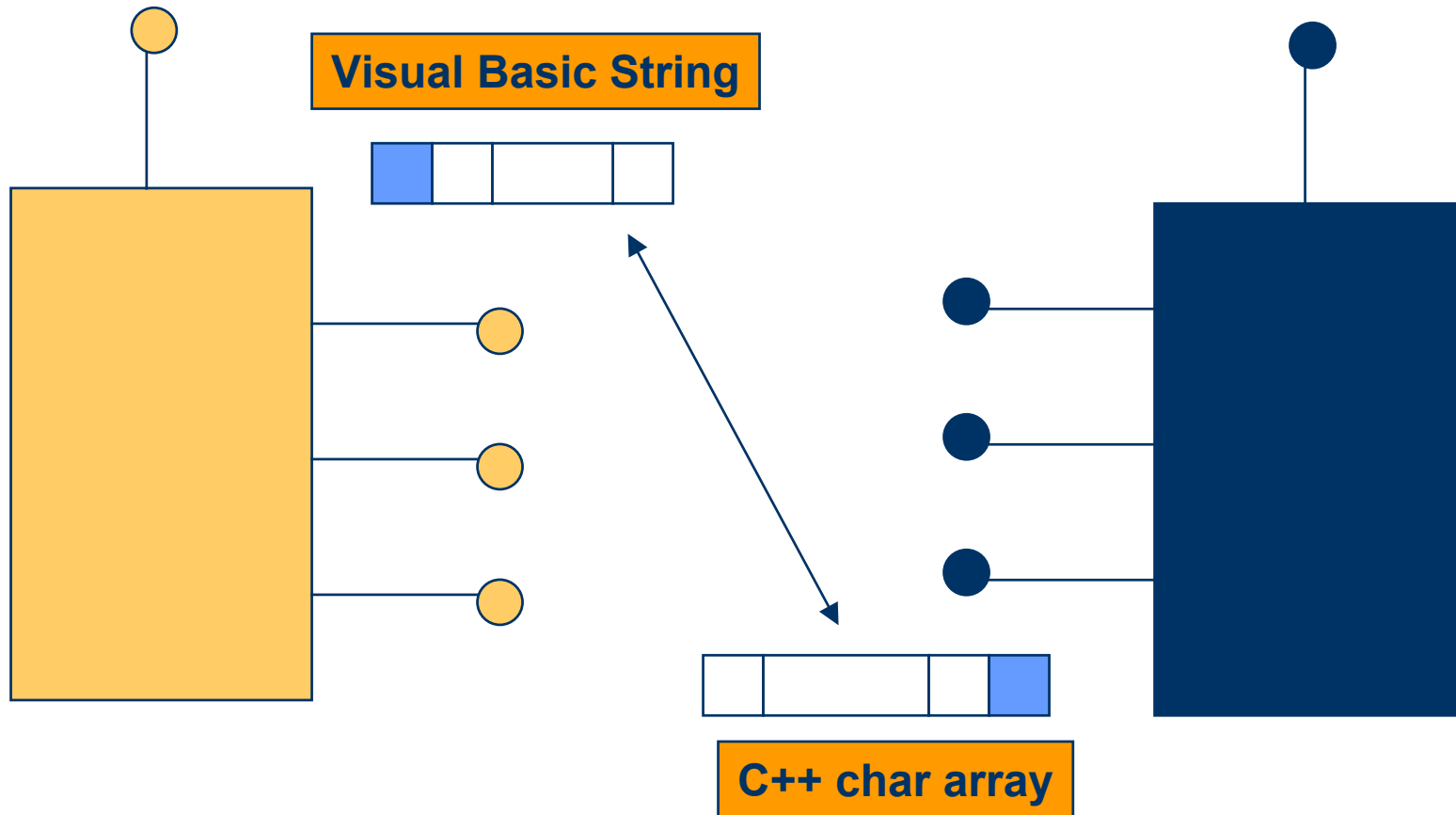
**Before COM/DLLs, applications were completely separate entities with little or no integration.**

# Holy Grail of Software Reuse

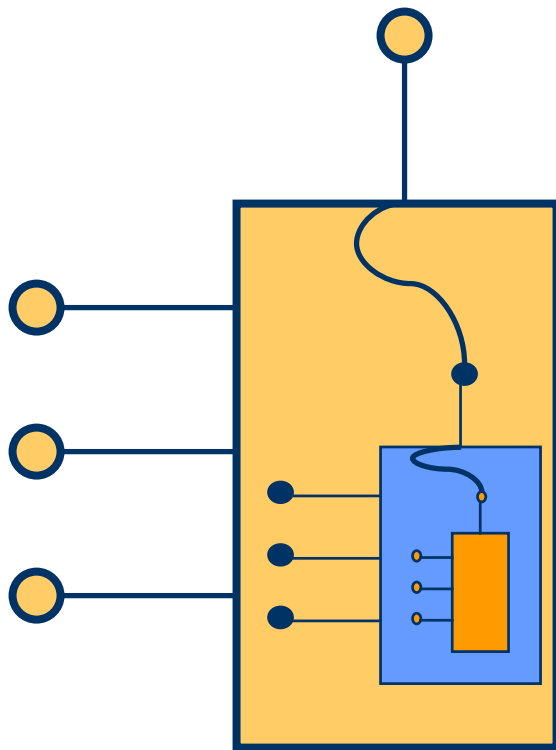


**COM provides a way for components to integrate. However, each component must provide the "plumbing," and objects cannot directly interact.**

# Type Incompatibilities



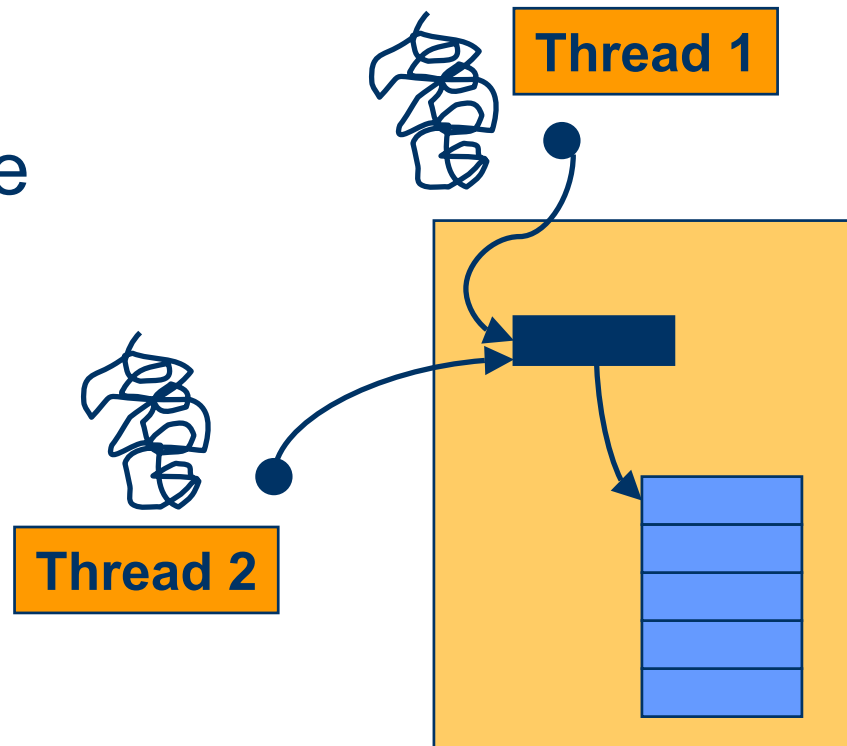
# No Reuse of Implementation



**You have to encode  
“implementation  
inheritance” explicitly  
using aggregation.**

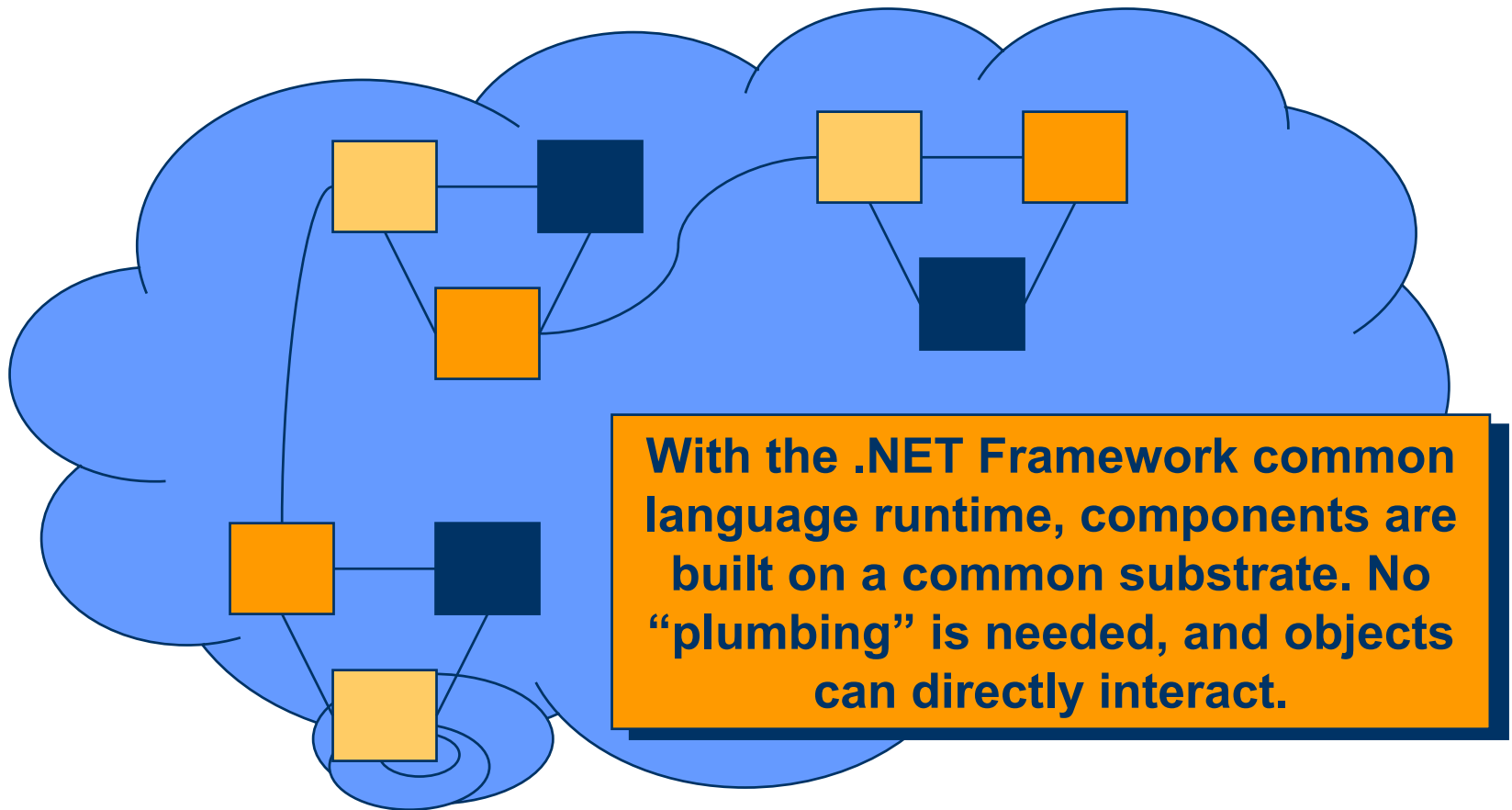
# Concurrency

- ◆ Concurrency control is not built-in (that is, coclasses do not have a classwide lock).
- ◆ Instead it has several sorts of “apartments”: MTA, STA, FTA





# Deep Integration



# Breeding Dogs

```
using System;

namespace virtualDog {
    public class Dog {
        public virtual void RollOver () {
            Console.WriteLine("Scratch my tummy.");
            Bark();
        }

        public virtual void Bark () {
            Console.WriteLine("WOOF WOOF (Dog)");
        }
    }
}
```



Thanks, Roger  
Sessions!

# Breeding Dogs



```
Imports System
Namespace VirtualDog
    Public Class Mopje : Inherits Dog
        Public overrides Sub Bark ()
            Console.WriteLine("WOEF WOEF (Mopje)")
        End Sub
    End Class
End Namespace
```

# Breeding Dogs

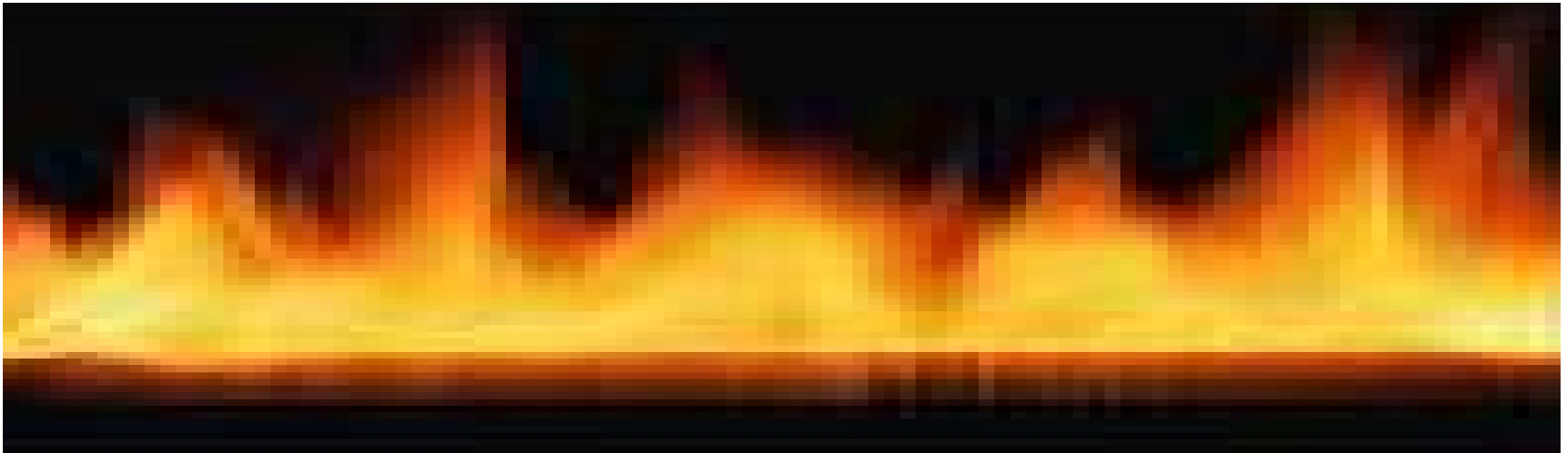
```
import VirtualDog;  
  
var d = new Dog();  
var m = new Mopje();  
  
d.RollOver();  
m.RollOver();
```



# Software Landscape

---

- ◆ Business model will change to include selling services and subscriptions and supporting non-Windows platforms.



# .NET Advantages

- ◆ Thanks to the CLI we have:
  - Cross-language implementation inheritance
  - Cross-language exception handling
  - Cross-language threading
  - Cross-language debugging
  - ASP.Net pages in any language
  - XML Webservices in any language
- ◆ Third Party Languages include:
  - COBOL, Pascal, Oberon, APL, Fortran, ...
  - Mondrian, Haskell, Mercury, Eiffel, ...