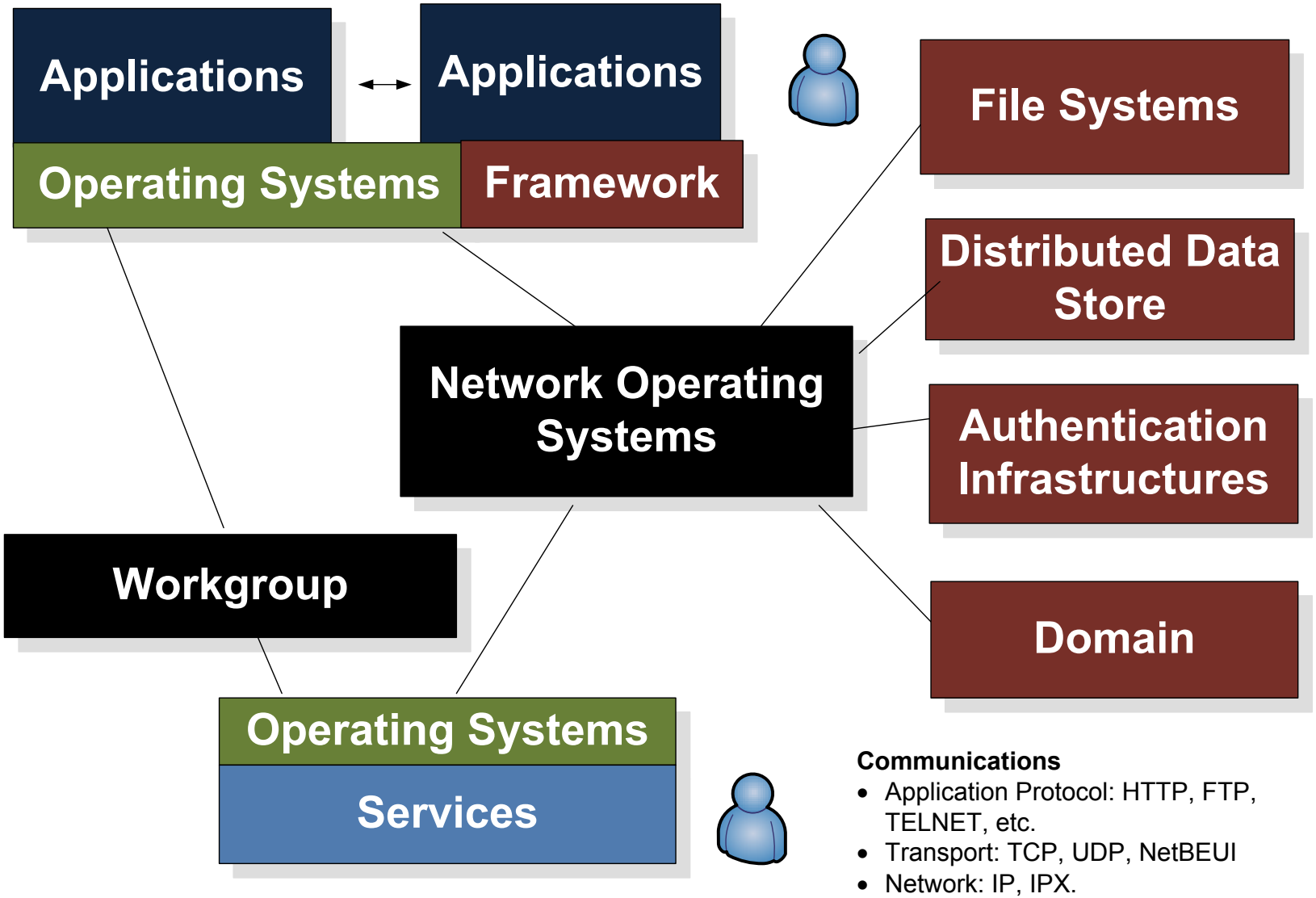
Software Security

Introduction
Best Practice
The Future
.NET Framework
Keeping Code Secure
Obfuscation
Software Security
.NET Security Model
ASP.NET
Conclusions

Software Security

Introduction



Communications

- Application Protocol: HTTP, FTP, TELNET, etc.
- Transport: TCP, UDP, NetBEUI
- Network: IP, IPX.

Lack of education in secure methods.

Lack of integration with the OS.

Lack of testing.

Lack of thought in the design process.

Lack of understanding of IP (Intellectual Property) protection.

Poor development environments.

Lack of understanding of encryption/ authentication.

Lack of support for different hardware
Programs were compiled to x86 code.

Weak integration with Internet/WWW
Code and WWW code where seen as separate entities

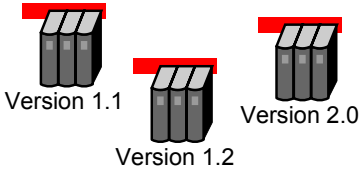
```
Code.asp
<%
     val1 = 10
     val2 = 20
     result = Cstr(val1) + Cstr(val2)
     response.write "<BR>Value is " & result
     result = val1 + val2
     response.write "<BR>Value is " & result
%>
```

Old Programs

Difficult to integrate different languages

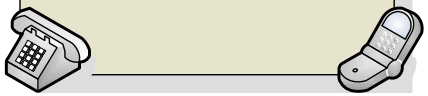
Lack of security Integration
Most programs where written with little care about security

Poor version control for system components



Poor robustness
Where applications often crash

Too much support for legacy code



Poor integration with different data sources, such as XML, databases, and text files.

Old Programs

Difficult deployments

Lack of sharing between applications

Poor pre-run checking
This is where applications often crashed on run-time

Software Security

Software Flaws

Non-intentional. These can be:

- **Validation flaws**. The code fails to check for valid input data.
- **Domain flaws**. This is where data leaks from one program to another.
- **Serialisation flaws**. This is where data changes while being passed from one program to another.
- **Identification/Authentication flaws**. This is where there is a lack of identification for processes or users.
- **Boundary condition flaws**. This is where resource access is not checked, and can thus allow an external hacker to use up resources.
- **Logic flaws**.

Bug

Trap-door

Intentional. This can either be caused by malious code (such as a Trojan or back-door programs).

Software Security

Best Practice

Software Security

Best practice

Principle of least privilege

Processes and scripts should run with the least privilege possible, to minimize damage

Never trust user input

All user input check be checked before it is used. This includes checking for correct number/string format, including vali characters.

Defence-in-depth

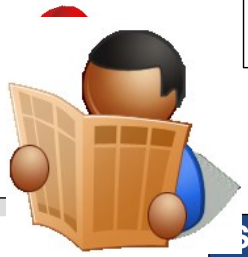
Checkpoints should be added for authentication and authorization at software interfaces, and interfaces within modules.

Use secure defaults

Sometimes developers encounter security problems in running and applications. It is important that these are fully tested before reducing the security.

Never rely on obfuscation

Obfuscation of code just makes it more difficult to determine its operation. If an intruder wants to "crac a program, then normally can, so other methods of securing the code should be employed.



If it's not used … disable it!

Any services which can be accessed can be compromised, thus, if they are not needed, they should be disabled.

Authenticate at the front-end

In terms of resources, it is often better to authentication at the front end rather than the back-end

System is only as secure as the weakest link

The overall security of a system is only as strong as its weakest link.

Never trust external systems

External systems should always be seen as a potential risk, and should never be fully trusted.

Reduce Surface Area

This should minimize the information that can be accessed from outside, and to handle errors in a graceful way.

Based on Microsoft ASP.NET Good Practice Guide

Software Security

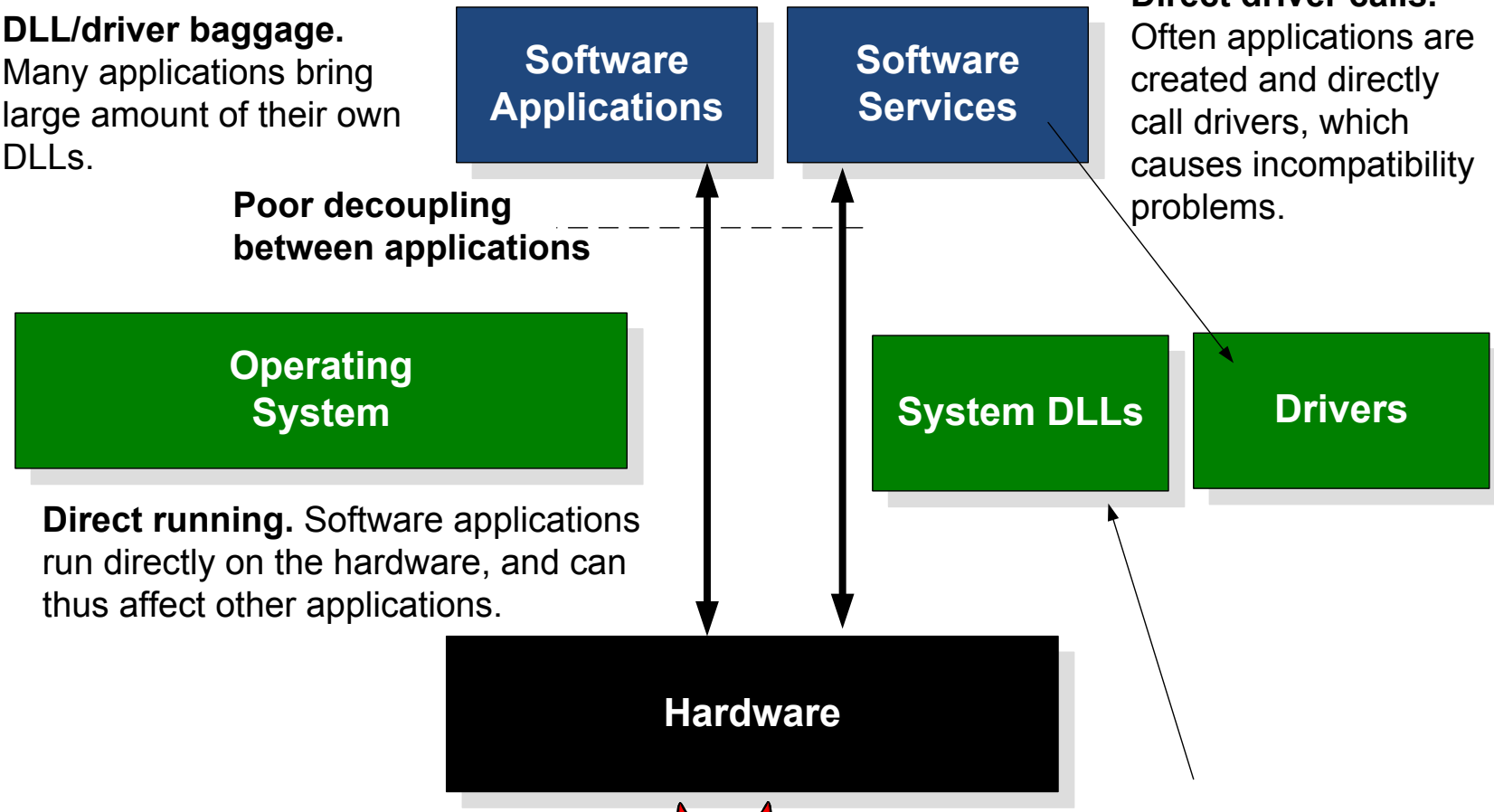
The Future

Binding to OS. Some applications are bound to the OS version, and do not work with other OSs or versions.

Bind to hardware. Often application programs are compiled to a certain hardware/architecture.

DLL/driver baggage. Many applications bring large amount of their own DLLs.

Direct driver calls. Often applications are created and directly call drivers, which causes incompatibility problems.

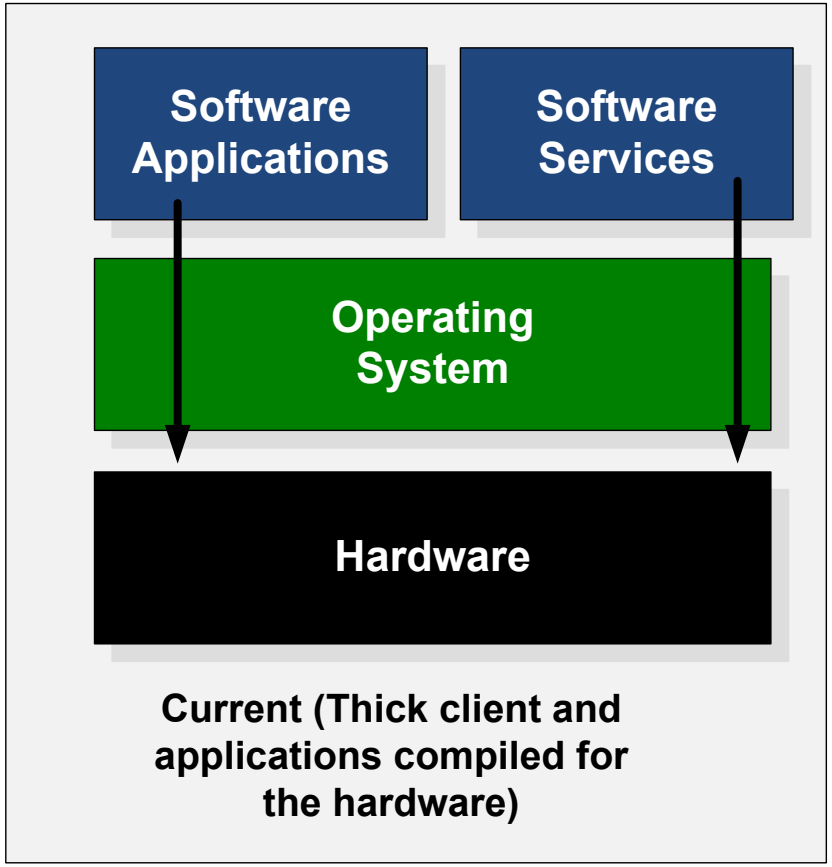


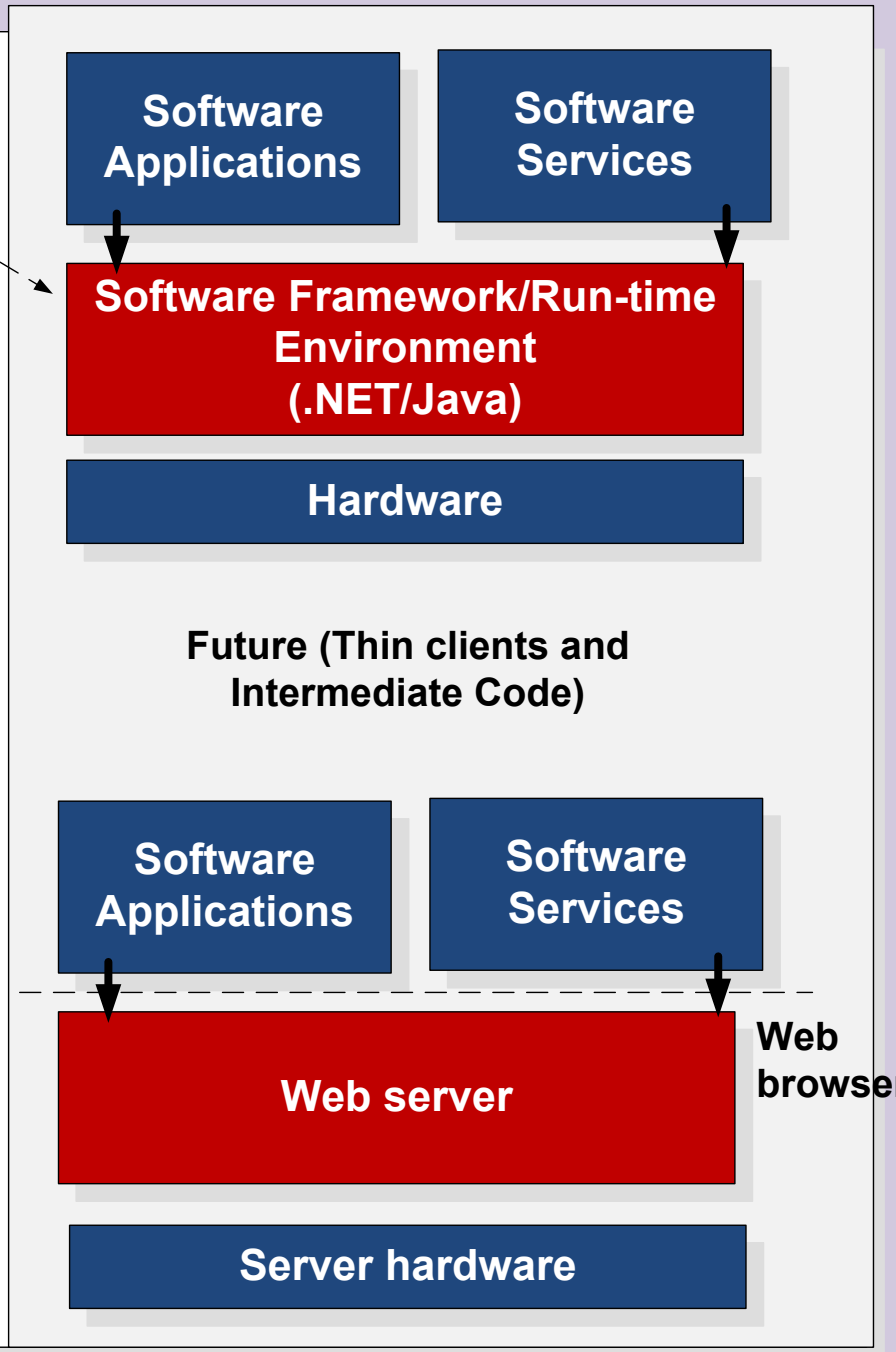
Direct running. Software applications run directly on the hardware, and can thus affect other applications.

DLL Hell. This is where DLL can be replaced with incorrect ones, such as being overwritten with malious ones or wrong version

Author: Prof Bill Buchanan

Applications are independent of the hardware and run within the framework. The framework manages their operation, to make sure they do not damage the system or other applications



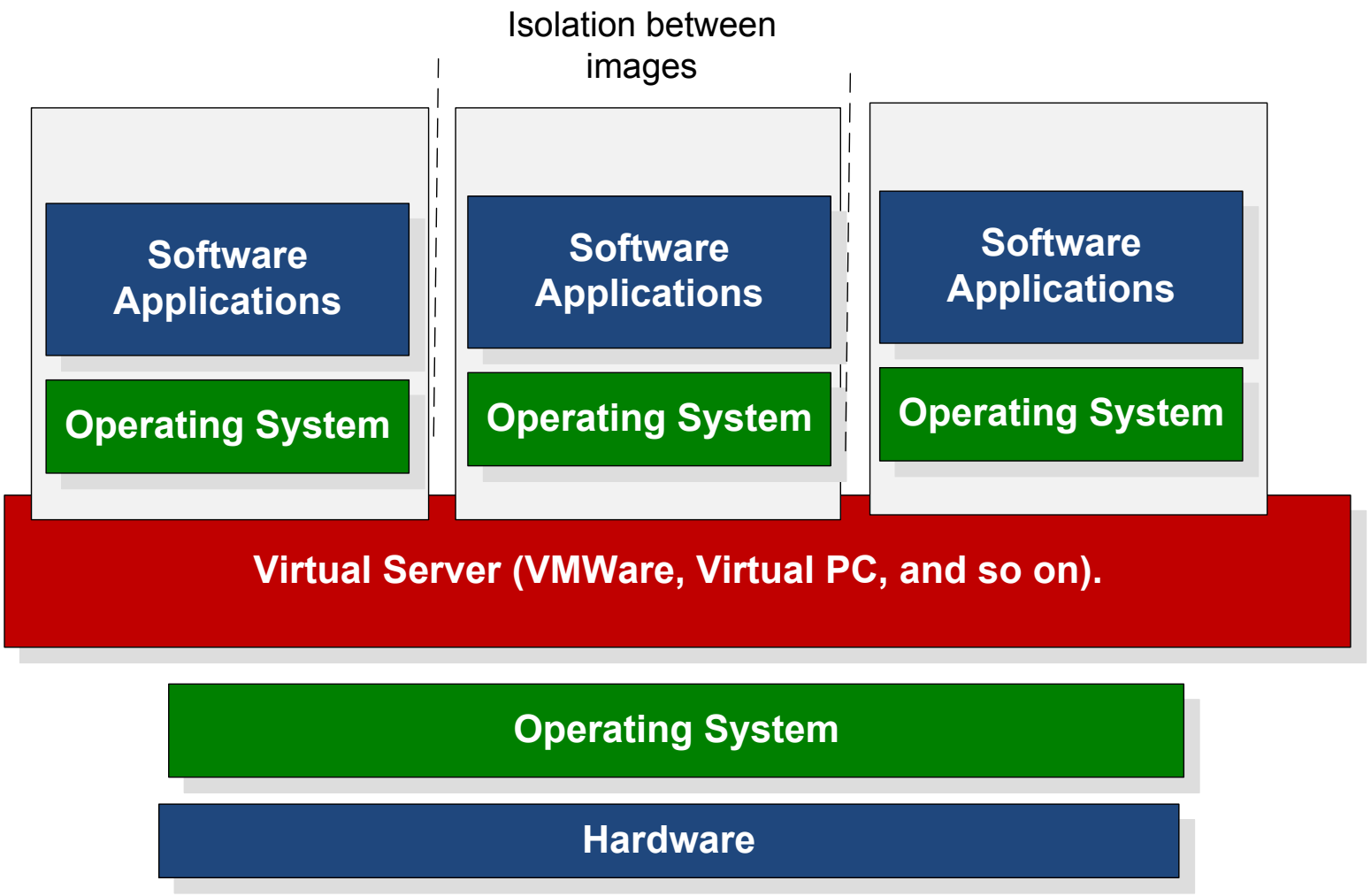


The Future

Software Security

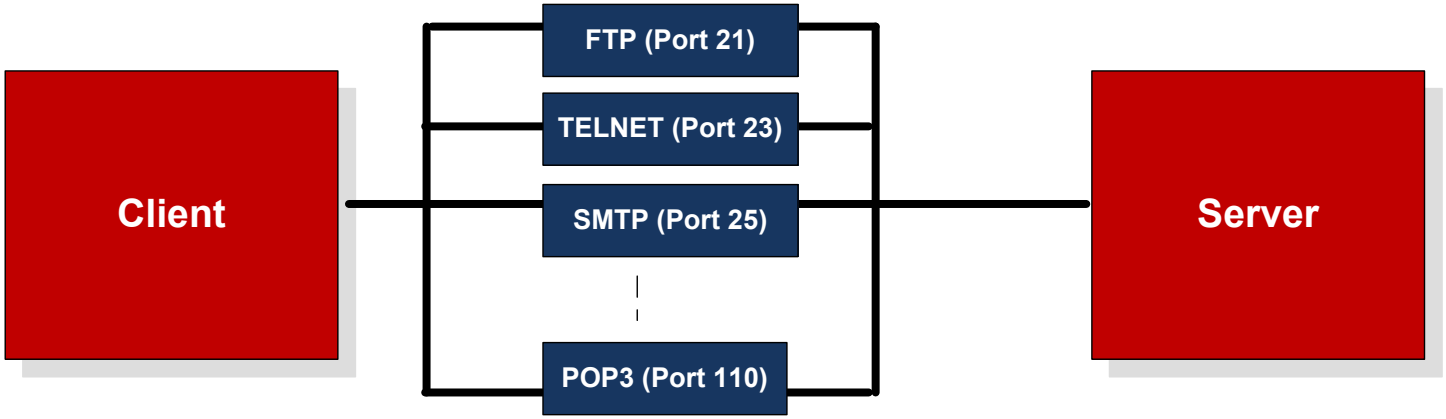
Author: Prof Bill Buchanan

The future?

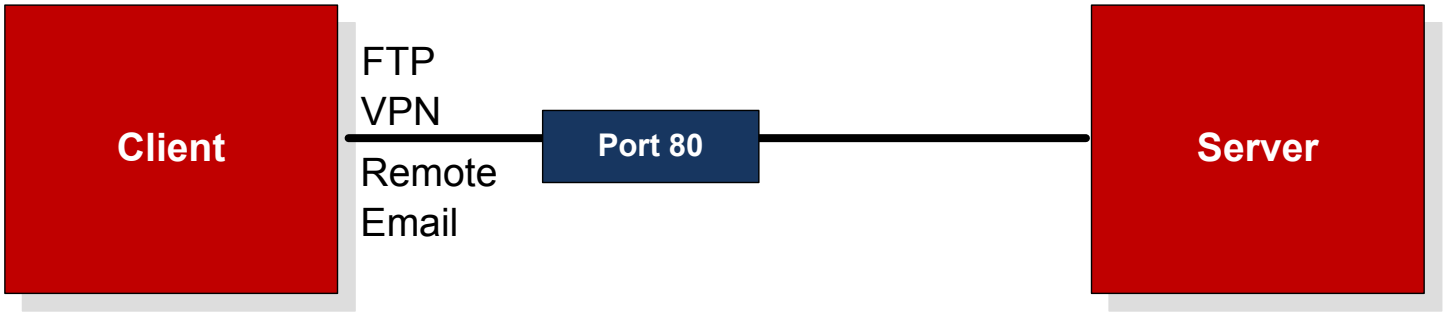


Virtualisation images allow multiple Oss to run on a single OS.

Author: Prof Bill Buchanan



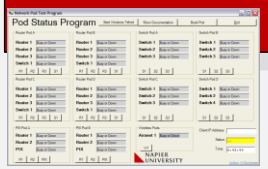
Applications use a wide range of TCP ports to communicate. Each of these ports could be blocked.



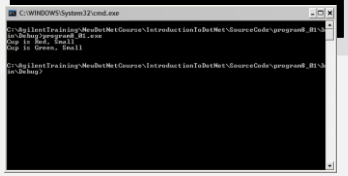
Applications communicate for a wide range of services through port 80 (Web port)… port 80 traffic is allowed through the firewall … but can cause security problems as the firewall cannot check the usage

A move to Port 80?

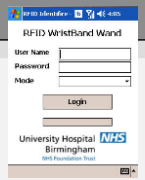
Integrated Programs (with logic, interface and data store in a Windows package)



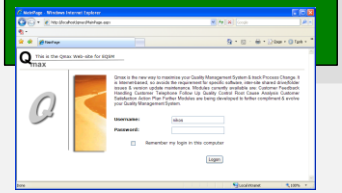
Integrated console program (with logic, interface and data store) in a console application)



Integrated mobile application (with logic, interface and data store) in a mobile application)



Integrated Web page (with logic, interface and data store in a Web package)



The Future

Software Security

Windows interface

Console interface

Mobile interface

Web interface

User services

Business logic and data store can be shared between applications

Shared business logic

Business services

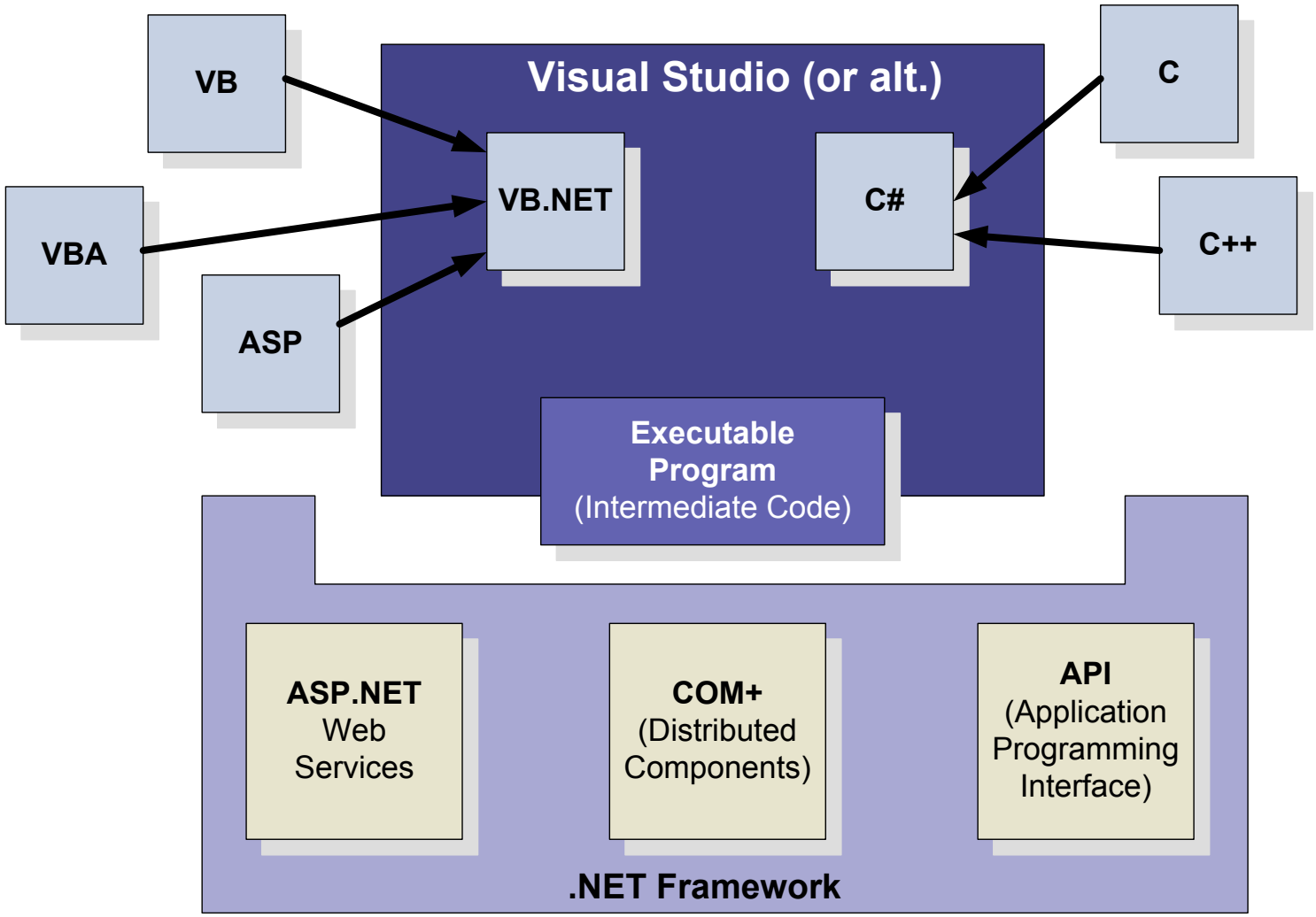
Shared data store

Data services

Three-tiered architecture

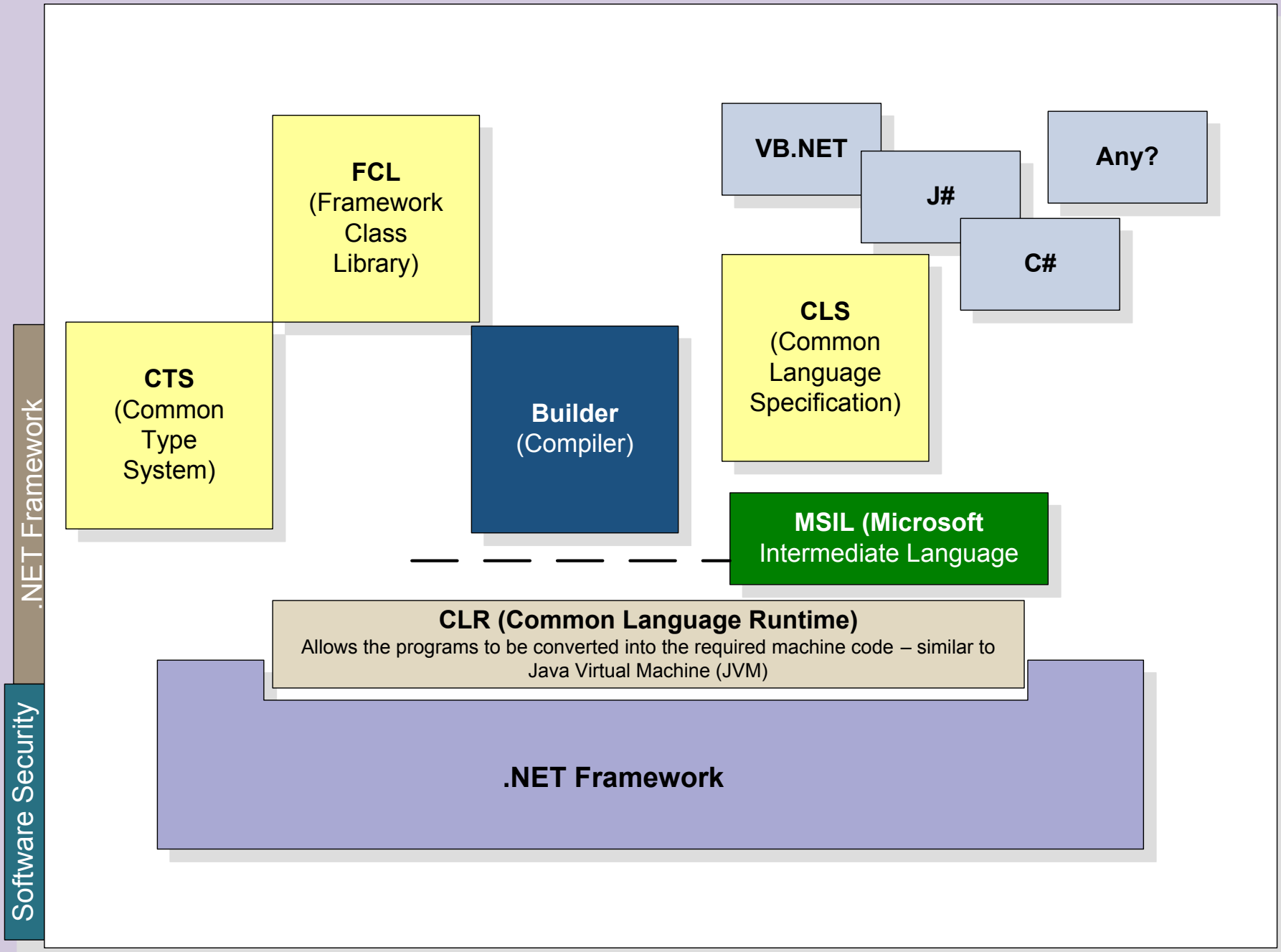
Software Security

The Future - .NET?



.NET Framework

Software Security


.NET Framework
Software Security
FCL
(Framework
Class
Library)
CTS
(Common
Type
System)
Builder
(Compiler)
VB.NET
J#
Any?
C#
CLS
(Common
Language
Specification)
MSIL (Microsoft
Intermediate Language
CLR (Common Language Runtime)
Allows the programs to be converted into the required machine code – similar to
Java Virtual Machine (JVM)
.NET Framework

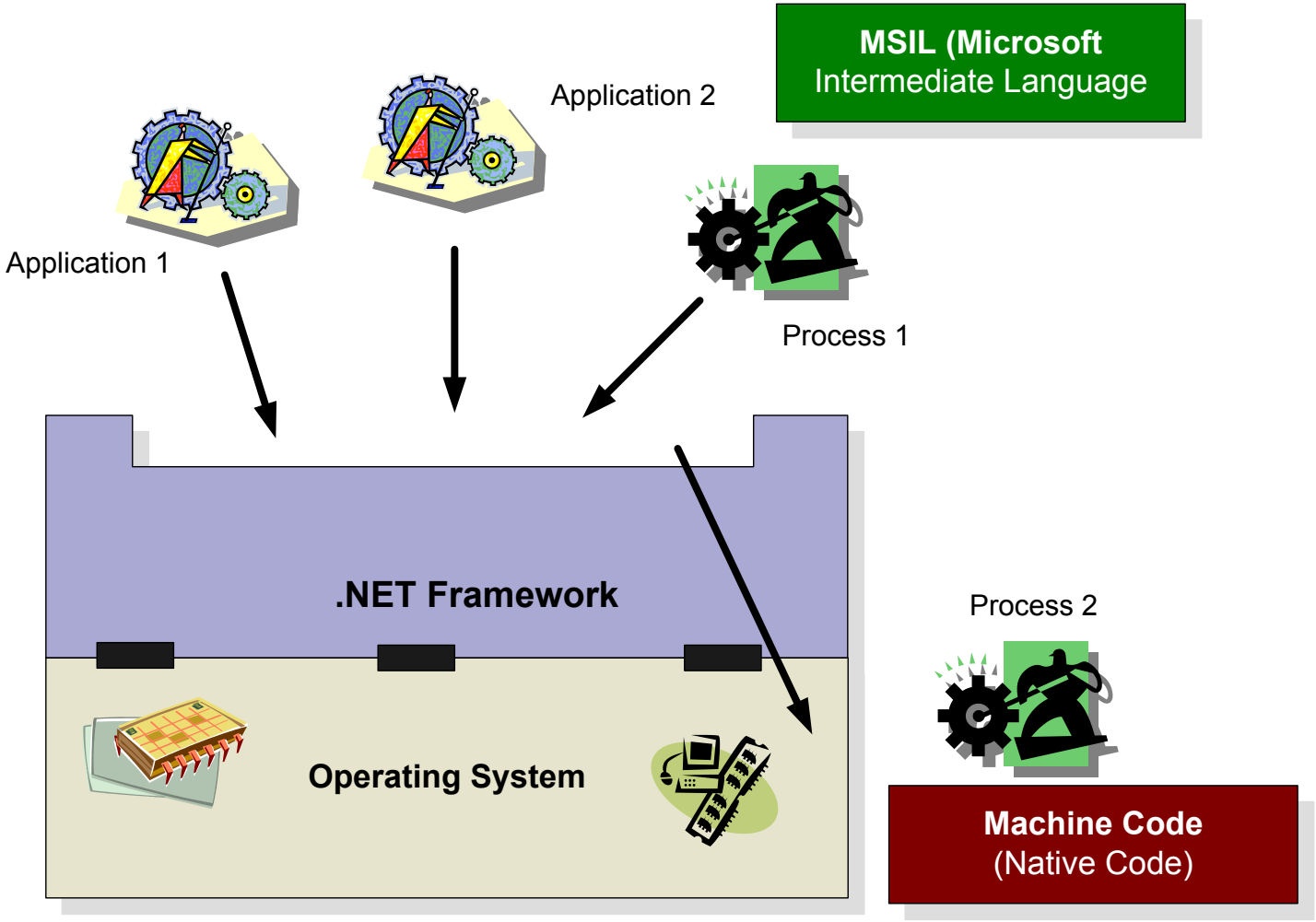
```
Volume in drive C has no label.
Volume Serial Number is 1A83-0D9D

Directory of C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322

21/02/2003  08:24             7,680 Accessibility.dll
21/02/2003  06:00            98,304 alink.dll
20/02/2003  20:19            24,576 aspnet_filter.dll
20/02/2003  20:19           253,952 aspnet_isapi.dll
20/02/2003  20:19            40,960 aspnet_rc.dll
20/02/2003  20:09            77,824 CORPerfMonExt.dll
21/02/2003  11:21           626,688 cscomp.dll
21/02/2003  08:24            12,288 cscompmgd.dll
21/02/2003  08:24            33,792 CustomMarshalers.
29/07/2002  12:11           219,136 c_g18030.dll
21/02/2003  11:21           524,288 diasymreader.dll
19/03/2003  02:52           245,760 envdte.dll
20/02/2003  20:16           798,720 EventLogMessages.
20/02/2003  20:06           282,624 fusion.dll
21/02/2003  08:24             7,168 IEExecRemote.dll
21/02/2003  08:24            32,768 IEHost.dll
21/02/2003  08:24             4,608 IIEHost.dll
21/02/2003  08:25         1,564,672 mscorcfg.dll
20/02/2003  20:09            77,824 mscordbc.dll
20/02/2003  20:09           233,472 mscordbi.dll
20/02/2003  20:09            86,016 mscorie.dll
20/02/2003  20:06           311,296 mscorjit.dll
20/02/2003  20:09            98,304 mscorld.dll
21/02/2003  08:26         2,088,960 mscorlib.dll
20/02/2003  19:43           131,072 mscormmc.dll
20/02/2003  20:06            65,536 mscorpe.dll
20/02/2003  20:09           143,360 mscorrc.dll
20/02/2003  20:09            81,920 mscorsec.dll
20/02/2003  20:09            77,824 mscorsn.dll

        20/02/2003  20:07         2,494,464 mscorsvr.dll
```

.NET Program
Uses the framework
for a whole host of classes:
encryption, Web,
networking,
forms, etc.

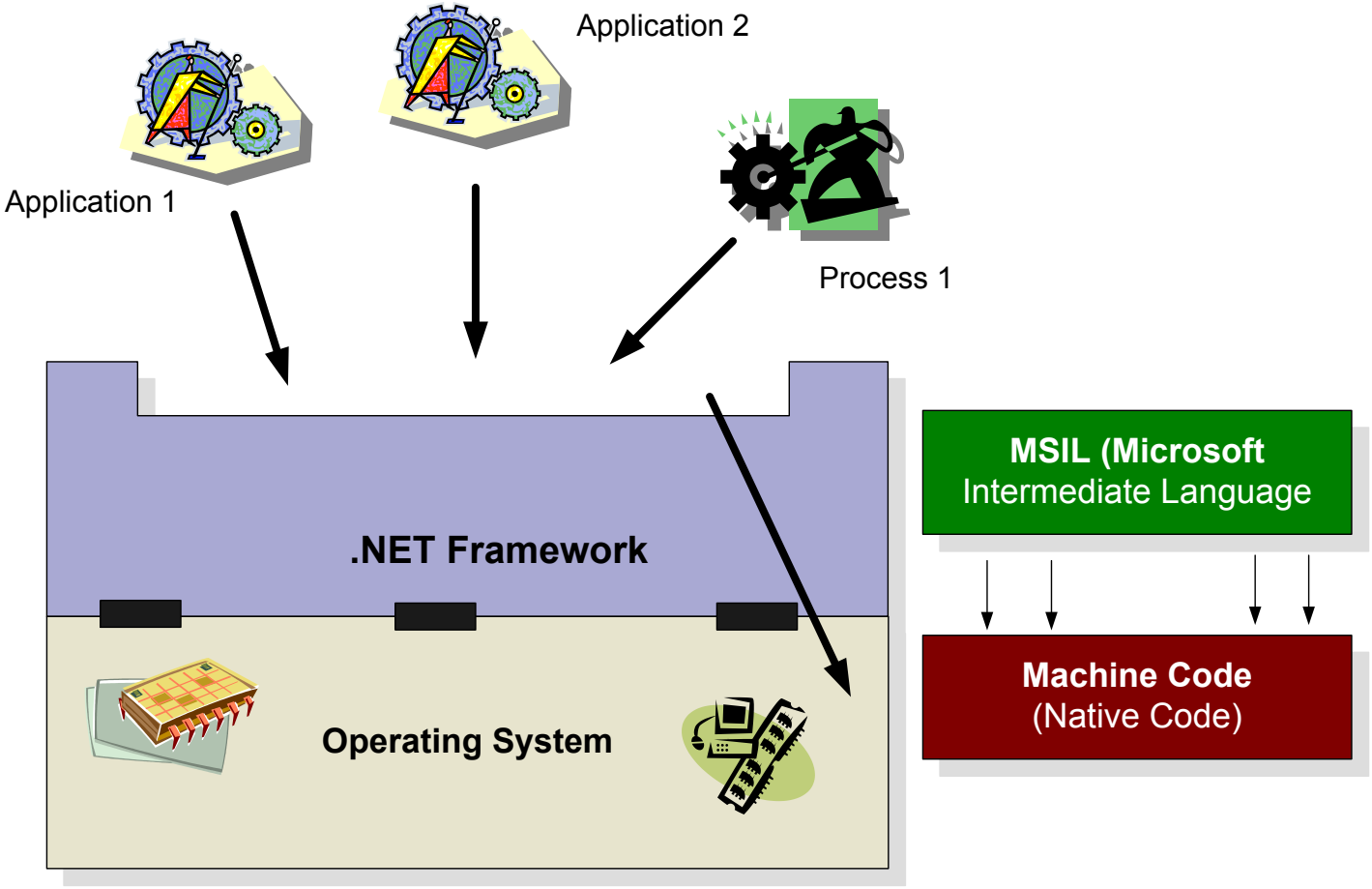
.NET Framework

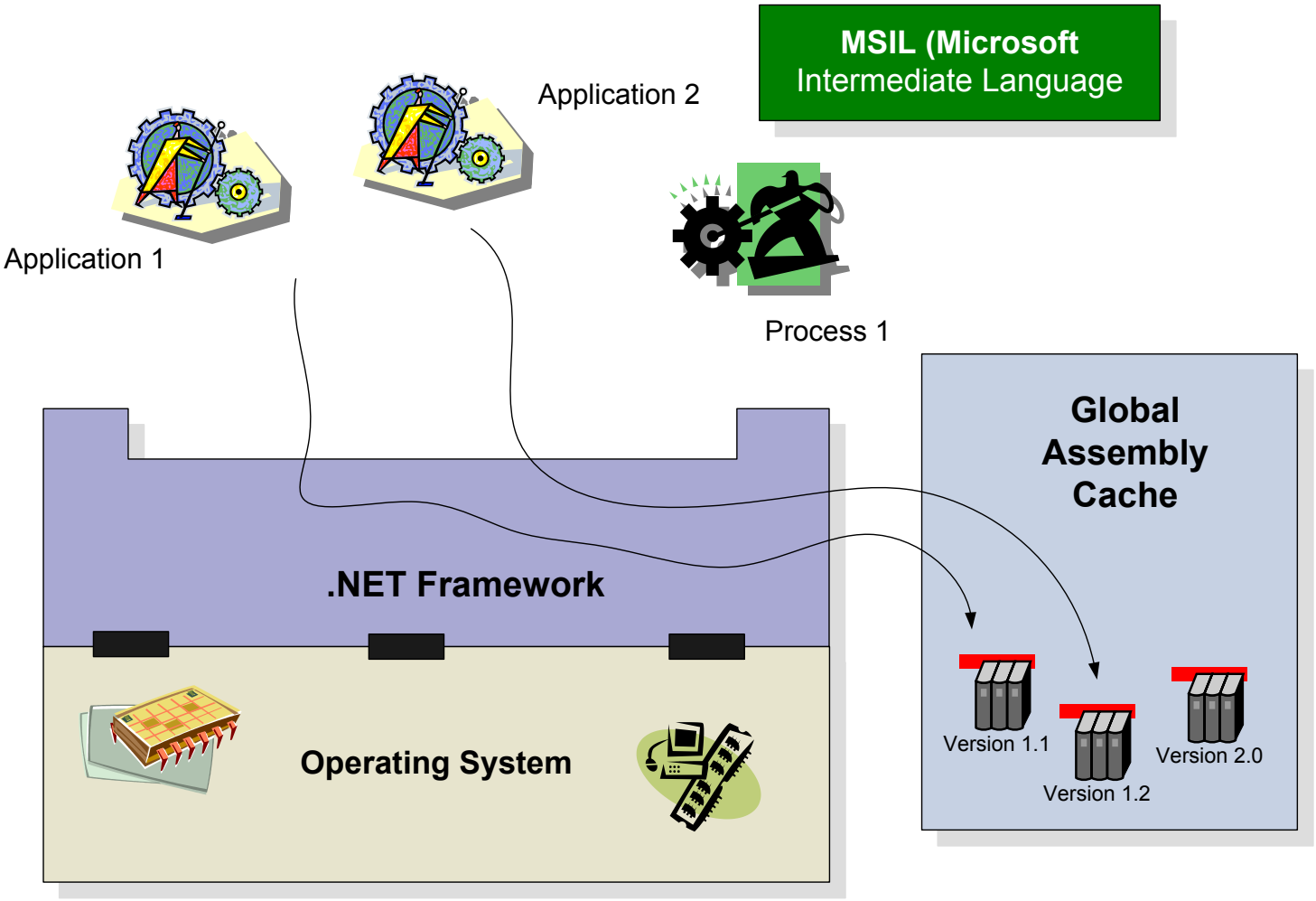
Mscorlib.dll
Arrays,
File I/O,
System,
Security,
Etc.

.NET Framework Files



Author: Prof Bill Buchanan


Application 1
Application 2
Process 1
.NET Framework
Operating System
MSIL (Microsoft
Intermediate Language
Machine Code
(Native Code)



Author: Prof Bill Buchanan

.NET Framework 2.0 Configuration

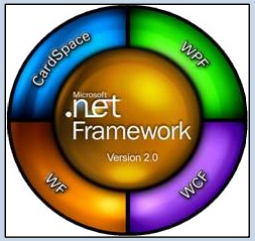
File Action View Help

- Console Root
  - .NET Framework 2.0 Con
    - My Computer
      - Assembly Cache
      - Configured Asser
      - Remoting Service
      - Runtime Security
      - Applications

Assembly Name	Version	Locale	Public Key Token
Microsoft.VisualStudio.EnterpriseTools.Cl...	9.0.0.0	neutral	b03f5f7f11d50a3a
Microsoft.VisualStudio.Shell.9.0	2.0.0.0	neutral	b03f5f7f11d50a3a
PresentationFontCache	3.0.0.0	neutral	31bf3856ad364e35
Microsoft.Build.Engine	3.5.0.0	neutral	b03f5f7f11d50a3a
System.DirectoryServices.AccountManag...	3.5.0.0	neutral	b77a5c561934e089
PresentationFramework.Classic	3.0.0.0	neutral	31bf3856ad364e35
Microsoft.VisualStudio.Modeling.Artifact...	9.0.0.0	neutral	b03f5f7f11d50a3a
Microsoft.CompactFramework.Design.P...	9.0.0.0	neutral	b03f5f7f11d50a3a
Microsoft.Nautilus.CompositionModel	1.0.0.0	neutral	31bf3856ad364e35
Microsoft.VisualStudio.CommonIDE	8.0.0.0	neutral	b03f5f7f11d50a3a
WindowsFormsIntegration	3.0.0.0	neutral	31bf3856ad364e35
Microsoft.VisualStudio.Shell.Interop.9.0	9.0.0.0	neutral	b03f5f7f11d50a3a
UIAutomationTypes	3.0.0.0	neutral	31bf3856ad364e35
Microsoft.Expression.Code	1.0.1083.0	neutral	31bf3856ad364e35
Microsoft.Expression.Project	1.0.1083.0	neutral	31bf3856ad364e35
Microsoft.Office.Tools.v9.0	9.0.0.0	neutral	b03f5f7f11d50a3a
System.ServiceModel	3.0.0.0	neutral	b77a5c561934e089
PresentationFramework	3.0.0.0	neutral	31bf3856ad364e35
System.Web.Extensions.Design	3.5.0.0	neutral	31bf3856ad364e35
Microsoft.VisualStudio.Tools.Application...	9.0.0.0	neutral	b03f5f7f11d50a3a

.NET Framework Version 3.0 (Nov 2006, WinFX)

- Windows CardSpace
- Windows Presentation Foundation
- Windows Communication Foundation
- Windows Workflow Foundation



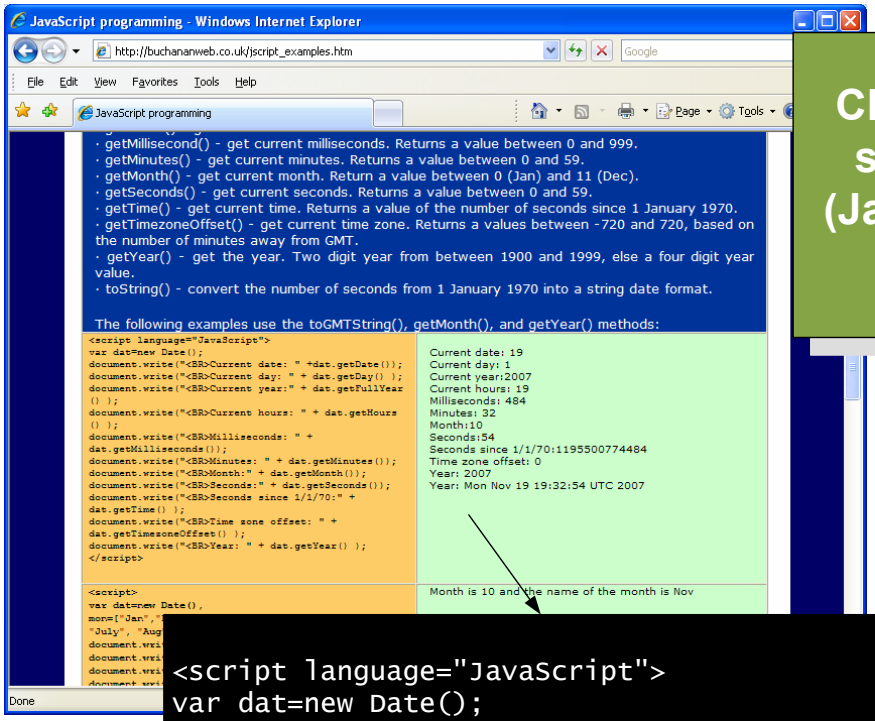
.NET Framework Version 2.0 (Nov 2005)

- Bug fixes.
- 64-bit support.
- Language support for generics.
- New controls.
- ASP.NET components.
- Bluetooth.

.NET Framework 1.0
Jan 2002

.NET Framework 1.1 (April 2003)

- Bug fixes.
- .NET Compact Framework

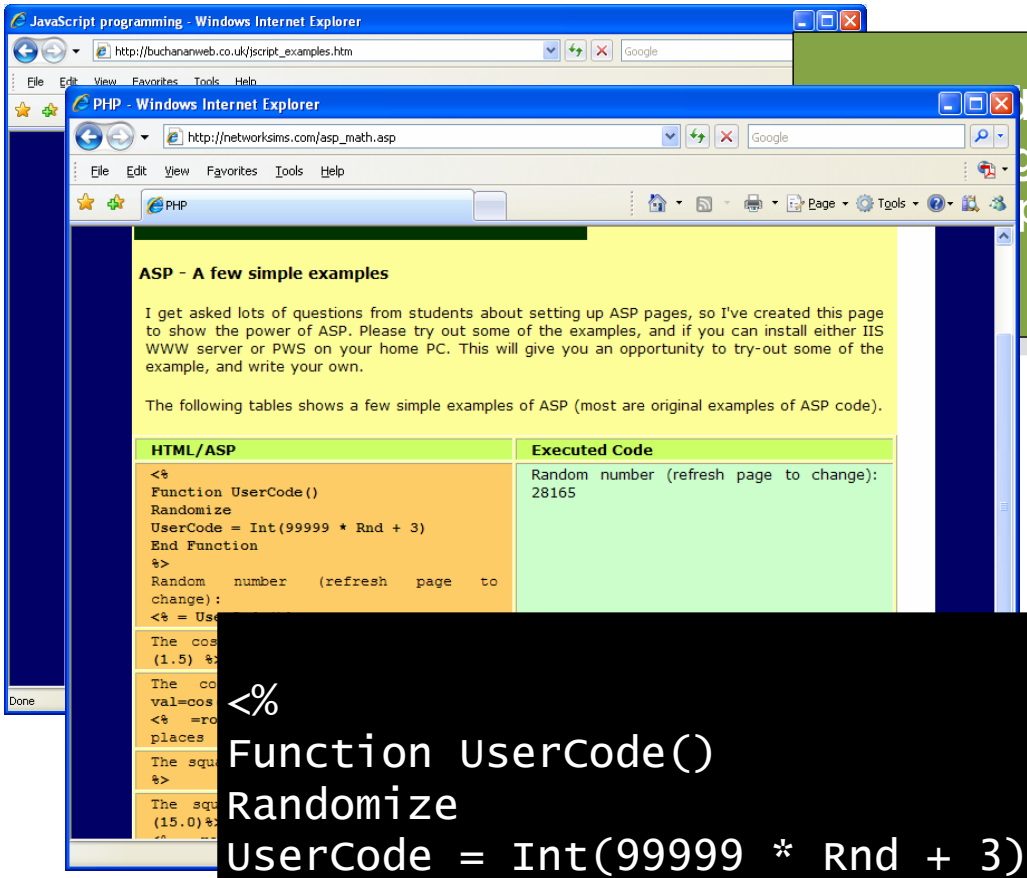


Client-side scripting (JavaScript/ Ajax)

Server-side Scripting (ASP/PHP)

Code-behind (ASPX)

```
<script language="JavaScript">
var dat=new Date();
document.write("<BR>Current date: " +dat.getDate());
document.write("<BR>Current day: " + dat.getDay() );
document.write("<BR>Current year:" + dat.getFullYear() );
document.write("<BR>Current hours: " + dat.getHours() );
document.write("<BR>Milliseconds: " + dat.getMilliseconds());
document.write("<BR>Minutes: " + dat.getMinutes());
document.write("<BR>Month:" + dat.getMonth());
document.write("<BR>Seconds:" + dat.getSeconds());
document.write("<BR>Seconds since 1/1/70:" + dat.getTime() );
document.write("<BR>Time zone offset: " +
dat.getTimezoneOffset() );
document.write("<BR>Year: " + dat.getYear() );
</script>
```




```
<%
Function UserCode()
Randomize
UserCode = Int(99999 * Rnd + 3)
End Function
%>
Random number (refresh page to
change):
<% = UserCode()%>
```

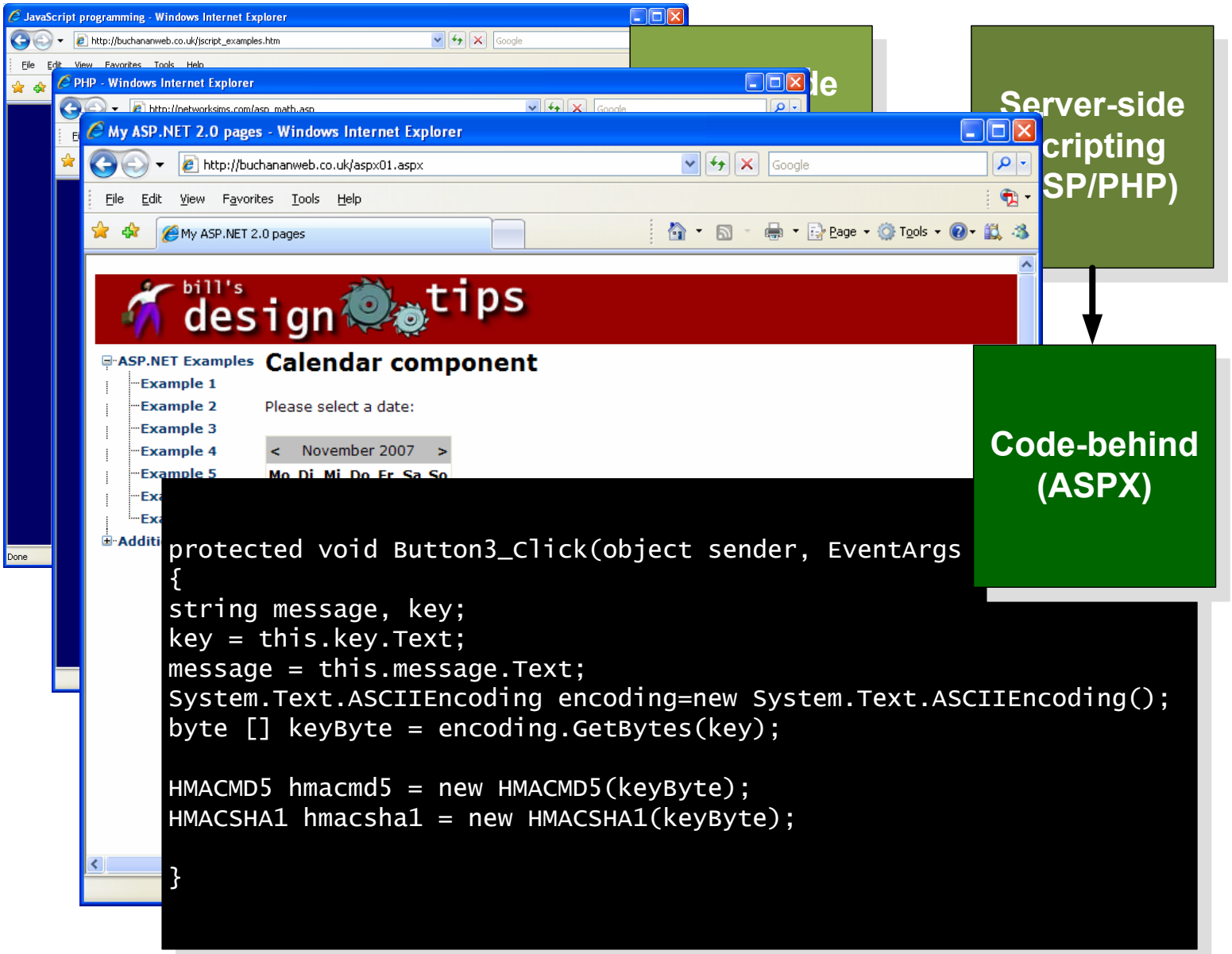
Server-side Scripting (ASP/PHP)

Code-behind (ASPX)

.NET Framework

Software Security

2nd generation: ASP/PHP/Integrated code



.NET Framework

Software Security

3rd generation: Code-behind

LEGACY

Windows ME
Windows 98
Windows NT
Windows 3.x

Server side: ASP
Components: Direct X/COM
Languages: VB, C++

Reduced LEGACY

Windows XP
Windows Server 2003
Vista
Windows Server 2008

Server side: ASPX
Web services: ASMX
Components: COM+/.NET
Frameworks: .NET 1.0, 1.1, 2.0, 3.0 and 3.5
Languages: C#, VB.NET, J#, ...

No LEGACY

Future?

Server side: ASPX
Web services: ASMX
Graphics: Presentation Foundation (XAML)
Components: .NET
Frameworks: .NET 3.0
Languages: C#, VB.NET, J#, …
Architecture: Service-based

3rd generation: Code-behind

Software Security

Keeping Code Secure

```
using System;

namespace simple
{
    class Class1
    {
        static void Main(string[] args)
        {
            string name;

            System.Console.Write("What is your name?");
            name=System.Console.ReadLine();
            System.Console.WriteLine("Hello " + name);
        }
    }
}
```

EXE

```
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Code security

Software Security

Code to EXE

```
F:\docs\src\simple\test>dir
 Volume in drive F has no label.
 Volume Serial Number is 2886-0553

 Directory of F:\docs\src\simple\test

25/07/2008  01:20    <DIR>          .
25/07/2008  01:20    <DIR>          ..
28/01/2007  17:06               437 simple.cs
               1 File(s)            437 bytes
               2 Dir(s)  113,418,530,816 bytes free

F:\docs\src\simple\test>csc simple.cs
Microsoft (R) Visual C# 2008 Compiler version 3.5.21022.8
for Microsoft (R) .NET Framework version 3.5
Copyright (C) Microsoft Corporation. All rights reserved.


F:\docs\src\simple\test>dir
 Volume in drive F has no label.
 Volume Serial Number is 2886-0553

 Directory of F:\docs\src\simple\test

15/09/2008  16:37    <DIR>          .
15/09/2008  16:37    <DIR>          ..
28/01/2007  17:06               437 simple.cs
15/09/2008  16:37             4,096 simple.exe
               2 File(s)          4,533 bytes
               2 Dir(s)  113,418,526,720 bytes free

F:\docs\src\simple\test>simple
Program to determine the square root of a value.
Enter value 1:9
Square root is:3
```

Run program

```
F:\docs\src\simple\test>exemplar simple.exe > list.cs

F:\docs\src\simple\test>dir
 Volume in drive F has no label.
 Volume Serial Number is 2886-0553

 Directory of F:\docs\src\simple\test

15/09/2008  16:38    <DIR>          .
15/09/2008  16:38    <DIR>          ..
15/09/2008  16:38               451 list.cs
28/01/2007  17:06               437 simple.cs
15/09/2008  16:37             4,096 simple.exe

F:\docs\src\simple\test>type list.cs
namespace simple {
        class simple {

                [STAThread]
                private static void Main(string[] args) {
                        double local0;
                        double local1;

                        Console.WriteLine("Program to determine the
square root of a value.");
                        Console.Write("Enter value 1:");
                        local0 = Convert.ToDouble(Console.ReadLine());
                        local1 = Math.Sqrt(local0);
                        Console.WriteLine("Square root is:" + local1);
                }

                public simple() : base() {
                }
        }
}
```

Red Gate's .NET Reflector

File View Tools Help

C#

- mscorlib
- System
- System.Xml
- System.Data
- System.Web
- System.Drawing
- System.Windows.Forms
- simple
  - simple.exe
    - References
    - { } -
    - { } simple
      - simple
        - Base Types
        - Derived Types
        - .ctor()
        - Main(String[]) : Void

private static void **Main**(string[] args);

Declaring Type: simple.simple

Assembly: simple, Version=0.0.0.0

Disassembler

```
[STAThread]
private static void Main(string[] args)
{
    Console.WriteLine("Program to determine the square root of a value.");
    Console.Write("Enter value 1:");
    Console.WriteLine("Square root is:" + Math.Sqrt(Convert.ToDouble(Console.ReadLine())));
}
```

Softwa

Disassembler

.NET Decompiler, Java Decompiler Resources - Windows Internet Explorer

http://www.preemptive.com/decompilers/

File Edit View Favorites Tools Help

.NET Decompiler, Java Decompiler Resources

Home | Products | Free Downloads | Partners | Support | Company | News | Contact Us | PreEmptive Solutions

Search

Buy Now

Free Trial | See a Demo

index

Secure Coding Quiz

Decompilers for .NET, Decompilers for Java

What does a Java decompiler or a .NET decompiler do?

Java decompilers and .NET decompilers are designed to accept an executable such as a Java class or jar file, or a .NET exe or dll file as input, and produce a compilable source file as its result.

How good are Java decompilers and .NET decompilers?

Programs in Java or the .NET framework are easy to decompile. This is simply a reality of modern, intermediate-compiled languages. Both Java and .NET mutually share the use of expressive file syntax for delivery of executable code: bytecode in the case of Java, MSIL (Microsoft Intermediate Language) for .NET. Being much higher-level than binary machine code, the intermediate files contain identifiers and algorithms that are immediately observable and ultimately understandable.

See for yourself-- download a Java or .NET decompiler below or watch a demonstration on our decompiler and obfuscation demo page.

Should I use a Java or .NET decompiler that allows me to submit code from my browser?

Be careful. Those decompiler sites might preserve the source code of the files you submitted. It is best to download a free Java or .NET decompiler and run it locally.

How do I protect my Java or .NET code from reverse engineering?

Using an obfuscator will give you peace of mind when you deploy your applications. A good .NET obfuscator or Java obfuscator can put the reverse engineering bar back to or higher than it was before the days of intermediate languages.

Free Java Decompilers

- **Jad**: a Java decompiler written in C++. Jad is command-line only, but various GUIs are available on the download page.
- **JODE**: a powerful Java decompiler written in Java.

Free .NET Decompilers

- **Reflector for .NET**: Reflector is a .NET decompiler, featuring an excellent interface and fast, effective decompilation. Free add-ins extend Reflector's functionality, allowing for various useful operations. The FileDisassembler add-in will dump an assembly decompiled with Reflector to a set of source files that can then be searched for critical strings, or loaded, edited and recompiled.
- **Anakrino**: Another .NET decompiler, although it is becoming dated.

Protect yourself from Decompilers:

Java Obuscator- Dash O

.NET Obfuscator - Dotfuscator

Get a free evaluation copy

Internet 100%

Dotfuscator

Software Security

Obfuscation

```
#include <stdio.h> main(t,_,a)char
*a;{return!0<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86,0,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-
94,-27+t,a)&&t==2?_<13? main(2,_+1,"%s %d %d\
n"):9:16:t<0?t<-72?main(_,t, "@n'+,#'/*{}w+/
w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/
n{n+,/+#n+,/#\ ;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K
w'K:'+}e#';dq#'l \ q#'+d'K#!/
+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n'){)#}w'){){nl]'/
+#n';d}rw' i;# \ ){nl]!/n{n#'; r{#w'r nc{nl]'/
#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' \
iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/
*{q#'ld;r'}{nlwb!/*de}'c \ ;;{nl'-{}rw]'/
+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ')
}+}{rl#'{n' ')# \ }'+}##(!!/") :t<-
50?_==*a?putchar(31[a]):main(-
65,_,a+1):main((*a=='/')+t,_,a+1)
:0<t?main(2,2,"%s"):*a=='/'||main(0,main(-61,*a,
"!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m
.vpbks,fxntdCeghiry"),a+1);}
```

Obfuscation

Software Security

What does this program do?

On the first day of Christmas,
my true love sent to me
A partridge in a pear tree.
On the second day of Christmas,
my true love sent to me
Two turtle doves,
And a partridge in a pear tree….

Elimination of all whitespace.
Use of conditional and list expression instead of the more familiar if-then-else statement and statement blocks.
A simple encoding of the poem's strings.
Encoding of multiple "functions" into the single function main

Identifier Renaming

This involves renaming all the classes, methods, and fields to short names, or even non-printing names

Before

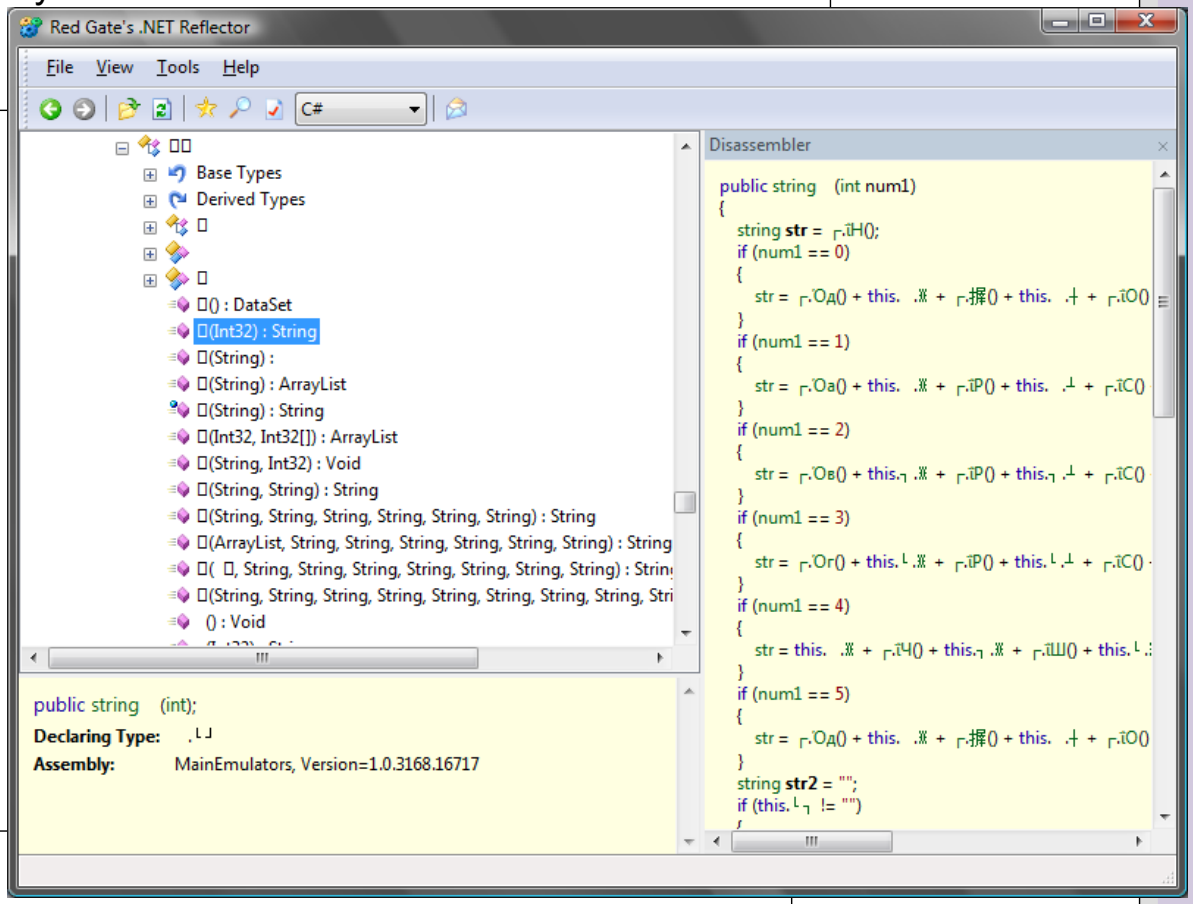
```
namespace Emulator {
        public class gen_switch {
          public ArrayList32logging59
          public ArrayList32level5Commands59
          public string32level5Name59
          public ArrayList32level69Commands59
          public string32level69Name59
          public ArrayList32level39Commands59
          public string32level39Name59
          public ArrayList32level16Commands59
          public string32level16Name59
          public ArrayList32level17Commands59
          public string32level17Name59
          public ArrayList32level25Commands59
          public string32level25Name59
```

After

```
C:\netwsims>exemplar mainemulators.exe
namespace ☺ {
        public class ☺ {
                public ArrayList32☺59
                public ArrayList32☻59
                public string32☺59
                public ArrayList32♥59
                public string32☻59
                public ArrayList32♦59
                public string32♥59
                public ArrayList32♣59
                public string32♦59
                public ArrayList32♠59
                public string32♣59
                public ArrayList32♫59
```

String Encryption

A standard way that many “crackers” work is to search for key strings within a program. String encryption is used to encrypt strings through an encryption method, so that they cannot be search for.

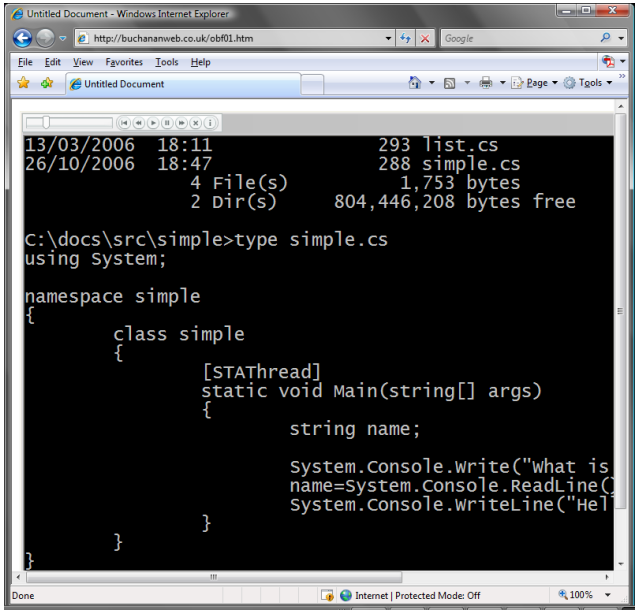


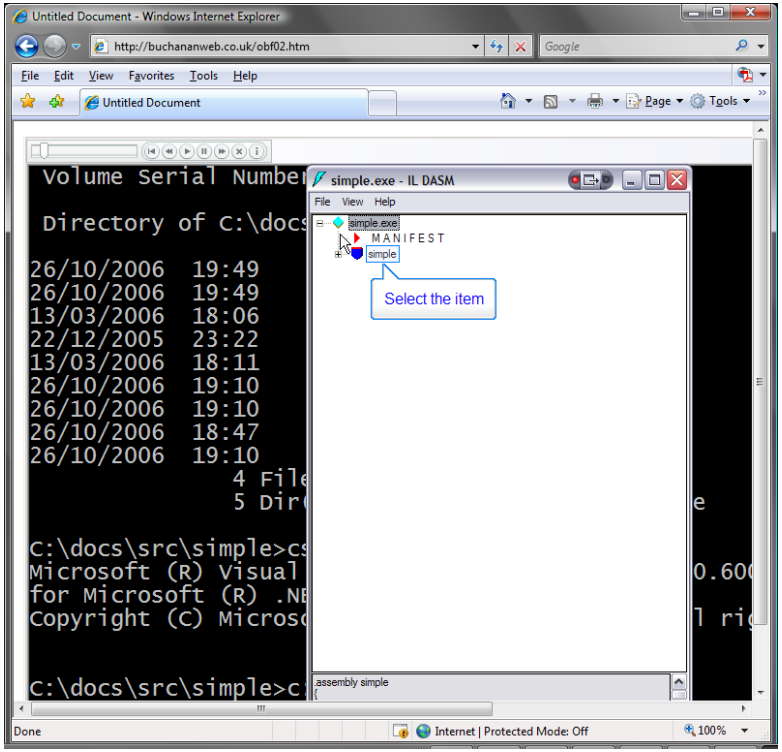
Flow obfuscation

This involves scrambling the flow of the program, so that it is difficult to determine its actual operation.

View demo of Obfuscation

http://buchananweb.co.uk/obf01.htm





View demo of Obfuscation2

http://buchananweb.co.uk/obf02.htm

```
using System;

namespace simple
{
    class Class1
    {
        static void Main(string[] args)
        {
            string name;

            System.Console.Write("What is your name?");
            name=System.Console.ReadLine();
            System.Console.WriteLine("Hello " + name);
        }
    }
}
```

EXE

```
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Obfuscation

Software Security

Example

Class1::Main : void(string[])

```
.method private hidebysig static void  Main(string[] args) cil managed
{
  .entrypoint
  .custom instance void [mscorlib]System.STAThreadAttribute::.ctor() = ( 01
  // Code size       33 (0x21)
  .maxstack  2
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Variables have been renamed

```
) cil managed

te::.ctor() = ( 01
  .maxstack  2
  .locals init (string V_0)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void class [mscorlib]System.Console::Write(string)
  IL_000a:  call       string class [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string string::Concat(string,
                                             string)
  IL_001b:  call       void class [mscorlib]System.Console::WriteLine(string
  IL_0020:  ret
} // end of method a::Main
```

Classes have been renamed

Obfuscation

Software Security

Example

Class1::Main : void(string[])

```
.method private hidebysig static void  Main(string[] args) cil managed
{
  .entrypoint
  .custom instance void [mscorlib]System.STAThreadAttribute::.ctor() = ( 01
  // Code size       33 (0x21)
  .maxstack  2
  .locals init ([0] string name)
  IL_0000:  ldstr      "What is your name\?"
  IL_0005:  call       void [mscorlib]System.Console::Write(string)
  IL_000a:  call       string [mscorlib]System.Console::ReadLine()
  IL_000f:  stloc.0
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string [mscorlib]System.String::Concat(string,
                                                              string)
  IL_001b:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0020:  ret
} // end of method Class1::Main
```

Variables have been renamed

Microsoft Visual C++ Runtime Library

Runtime Error!

Program: c:\exemplar.exe

This application has requested the Runtime to terminate it in an unusual way.
Please contact the application's support team for more information.

OK

```
  .maxstack
  .locals i
  IL_0000:
  IL_0005:
  IL_000a:
  IL_000f:
  IL_0010:  ldstr      "Hello "
  IL_0015:  ldloc.0
  IL_0016:  call       string string::Concat(string,
                                             string)
  IL_001b:  call       void class [mscorlib]System.Console::WriteLine(string
  IL_0020:  ret
} // end of method a::Main
```

Classes have been renamed

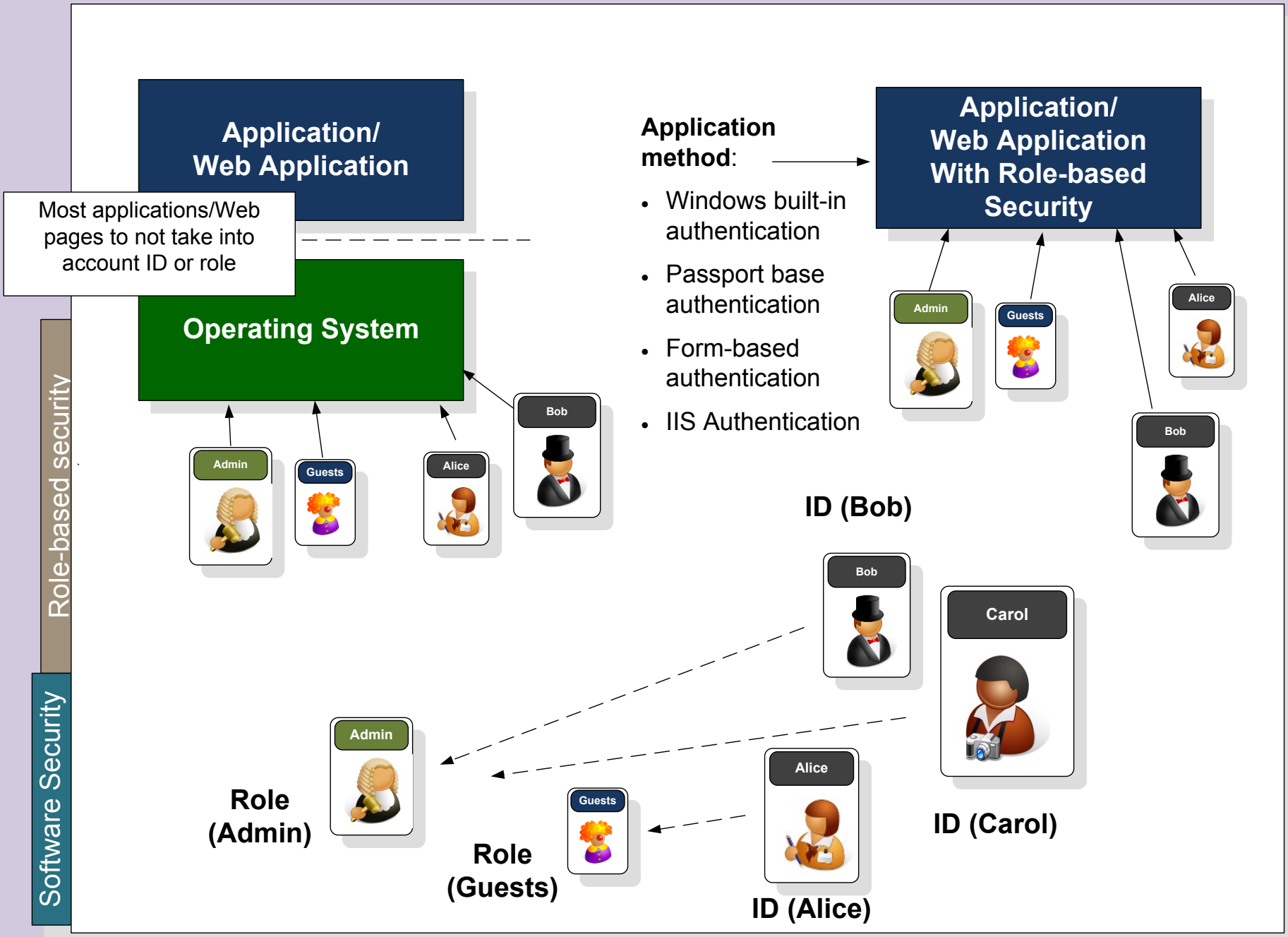
Obfuscation

Software Security

Example

Software Security

Role-based security



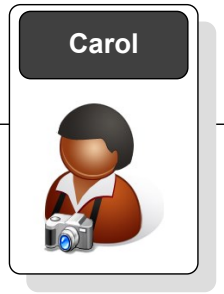
Role-based security

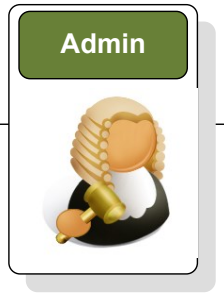
Role-based security

Software Security

```
using System;
using System.Security;
using System.Security.Principal;
namespace ConsoleApplication3
{
    class Class1
    {
        static void Main(string[] args)
        {
            WindowsIdentity myID = WindowsIdentity.GetCurrent();
            System.Console.WriteLine("Your ID: " + myID.Name);
            System.Console.WriteLine("Authentication: " +
                                                myID.AuthenticationType);
            WindowsPrincipal myPrin = new WindowsPrincipal(myID);
            if (myPrin.IsInRole(WindowsBuiltInRole.Administrator))
                System.Console.WriteLine("You're an Administrator ");
            else
                System.Console.WriteLine("You're not an Administrator");
            Console.ReadLine();
        }
    }
}
```

A major problem with software is that security is left to the operating system. With .NET, the developer can integrate role-based security into the program/Web application.





Code security

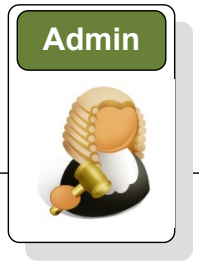
Software Security

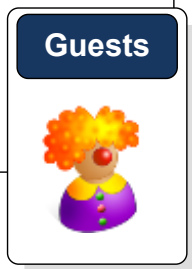
```
using System;
using System.Security;
using System.Security.Principal;
namespace ConsoleApplication3
{
    class Class1
    {
        static void Main(string[] args)
        {
            WindowsIdentity myID = WindowsIdentity.GetCurrent();
            System.Console.WriteLine("Your ID: " + myID.Name);
            System.Console.WriteLine("Authentication: " +
                                     myID.AuthenticationType);
```

```
- WindowsIdentity.GetCurrent();
ine("Your ID: " + myID.);
ine("Authentication:
onType);

in = new WindowsPrin

indowsBuiltInRole.Ad
iteLine("You're an A

iteLine("You're not
```

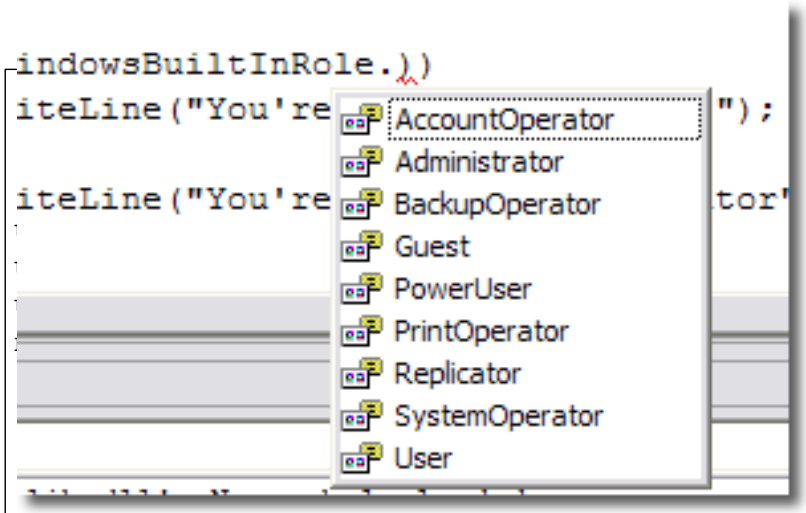
- AuthenticationType
- Equals
- GetHashCode
- GetType
- Impersonate
- IsAnonymous
- IsAuthenticated
- IsGuest
- IsSystem
- Name

```
lowsPrincipal(myID);
aRole.Administrator))
're an Administrator ");

're not an Administrator");
```

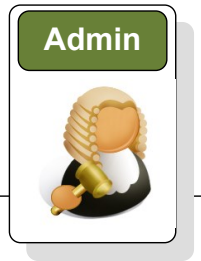


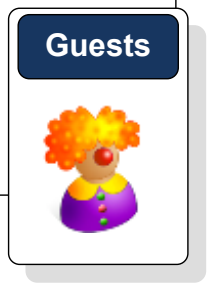


Role-based security




```
        {
            WindowsIdentity myID = WindowsIdentity.GetCurrent();
            System.Console.WriteLine("Your ID: " + myID.Name);
            System.Console.WriteLine("Authentication: " +
                                                myID.AuthenticationType);
            WindowsPrincipal myPrin = new WindowsPrincipal(myID);
            if (myPrin.IsInRole(WindowsBuiltInRole.Administrator))
                System.Console.WriteLine("You're an Administrator ");
            else
                System.Console.WriteLine("You're not an Administrator");
            Console.ReadLine();
        }
    }
}
```

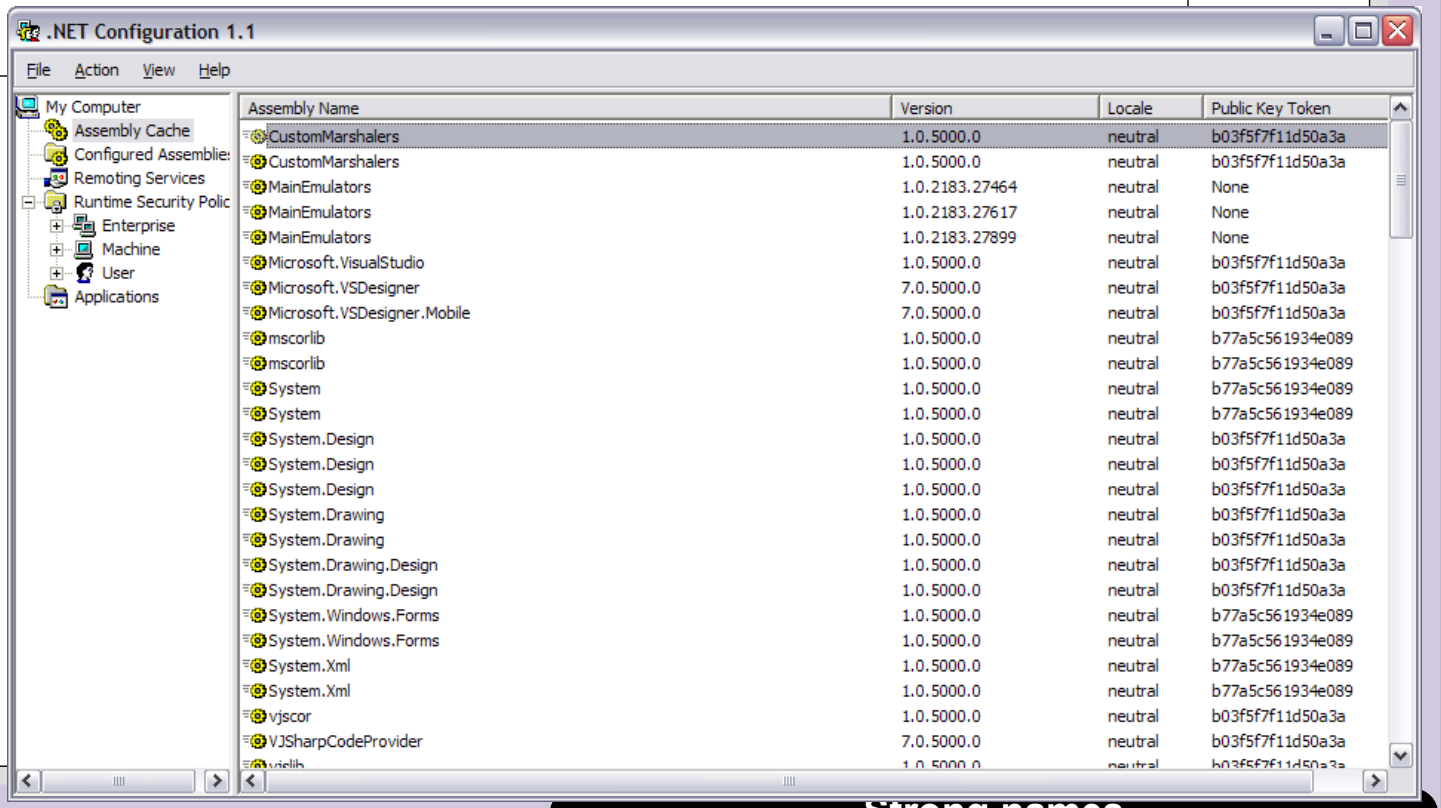




Software Security

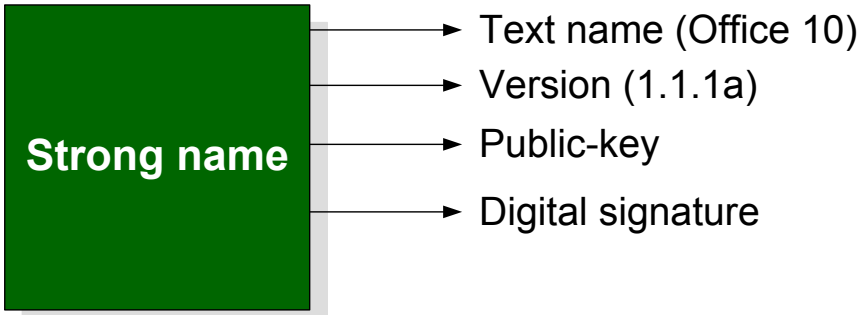
.NET Security

- Strong Name uses cryptography and digital signatures. It uses a digital signatures with asymmetric cryptography (RSA, EL Gamal), and a hash signature (MD5, SHA).
- Overcomes DLL hell and are used for versioning and authentication.
- In order to enhance security, an assembly uses an assembly strong name which normally has a text name, a public key and a digital signature. The digital signature is used to validate the assembly, and the system can thus check to see if the code has be modified in any way. If the code has been tampered with, the assembly will not load.

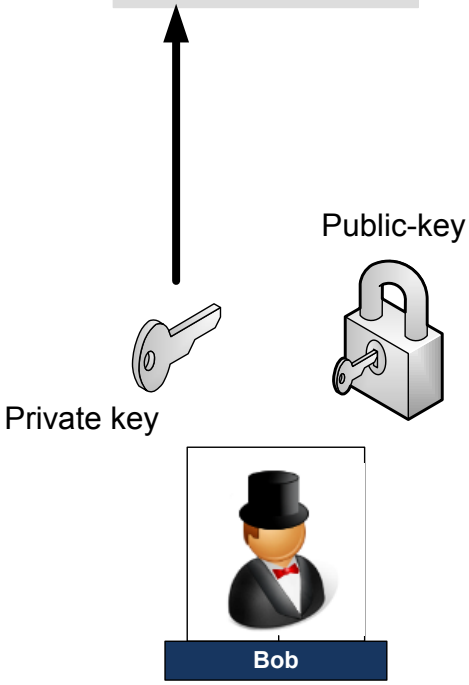


Assembly Name	Version	Locale	Public Key Token
CustomMarshalers	1.0.5000.0	neutral	b03f5f7f11d50a3a
CustomMarshalers	1.0.5000.0	neutral	b03f5f7f11d50a3a
MainEmulators	1.0.2183.27464	neutral	None
MainEmulators	1.0.2183.27617	neutral	None
MainEmulators	1.0.2183.27899	neutral	None
Microsoft.VisualStudio	1.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VSDesigner	7.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VSDesigner.Mobile	7.0.5000.0	neutral	b03f5f7f11d50a3a
mscorlib	1.0.5000.0	neutral	b77a5c561934e089
mscorlib	1.0.5000.0	neutral	b77a5c561934e089
System	1.0.5000.0	neutral	b77a5c561934e089
System	1.0.5000.0	neutral	b77a5c561934e089
System.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Drawing	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Drawing	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Drawing.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Drawing.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Windows.Forms	1.0.5000.0	neutral	b77a5c561934e089
System.Windows.Forms	1.0.5000.0	neutral	b77a5c561934e089
System.Xml	1.0.5000.0	neutral	b77a5c561934e089
System.Xml	1.0.5000.0	neutral	b77a5c561934e089
vjscor	1.0.5000.0	neutral	b03f5f7f11d50a3a
VJSharpCodeProvider	7.0.5000.0	neutral	b03f5f7f11d50a3a
vjslib	1.0.5000.0	neutral	b03f5f7f11d50a3a

Strong names



- No other name can ever exist.
- Supports different versions.
- Verifies the assembly.




```
C:\bill>sn -k bill.snk

Microsoft (R) .NET Framework Strong Name Utility
Version 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2002. All
rights reserved.

Key pair written to bill.snk

C:\bill>dir
 Volume in drive C has no label.
 Volume Serial Number is A873-2C50

 Directory of C:\bill

08/03/2007  21:15    <DIR>          .
08/03/2007  21:15    <DIR>          ..
08/03/2007  21:15               596 bill.snk
               1 File(s)          596 bytes
               2 Dir(s)   2,058,932,224 bytes free
```

.NET Security

Software Security

.NET Configuration 1.1

File Action View Help

My Computer
- Assembly Cache
- Configured Assemblie
- Remoting Services
- Runtime Security Polic
- Applications

Assembly Name	Version	Locale	Public Key Token
CustomMarshalers	1.0.5000.0	neutral	b03f5f7f11d50a3a
CustomMarshalers	1.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VisualStudio	1.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VisualStudio	1.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VSDesigner	7.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VSDesigner	7.0.5000.0	neutral	b03f5f7f11d50a3a
Microsoft.VSDesigner.Mobile	7.0.5000.0	neutral	b03f5f7f11d50a3a
mscorlib	1.0.5000.0	neutral	b77a5c561934e089
mscorlib	1.0.5000.0	neutral	b77a5c561934e089
System	1.0.5000.0	neutral	b77a5c561934e089
System	1.0.5000.0	neutral	b77a5c561934e089
System.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Design	1.0.5000.0	neutral	b03f5f7f11d50a3a
System.Drawing	1.0.5000.0	neutral	
System.Drawing	1.0.5000.0	neutral	
System.Drawing.Design	1.0.5000.0	neutral	
System.Drawing.Design	1.0.5000.0	neutral	
System.Windows.Forms	1.0.5000.0	neutral	
System.Windows.Forms	1.0.5000.0	neutral	
System.Xml	1.0.5000.0	neutral	
System.Xml	1.0.5000.0	neutral	
vjscor	1.0.5000.0	neutral	
vjscor	1.0.5000.0	neutral	
VJSharpCodeProvider	7.0.5000.0	neutral	
VJSharpCodeProvider	7.0.5000.0	neutral	
vjslib	1.0.5000.0	neutral	
vjslib	1.0.5000.0	neutral	

ne can

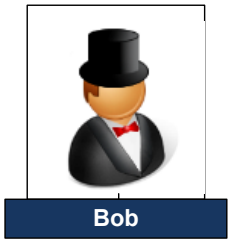
erent

ssembly.

Private key

Directory of C:

08/03/2007 21
08/03/2007 21
08/03/2007 21
1 File
2 Dir(



office Properties

General

Assembly name:	office
Modified:	
Culture:	neutral
Version:	11.0.0.0
Public key token:	71e9bce111e9429c
Codebase:	
Cache type:	Gac

OK Cancel Apply

Strong name

```
using System.EnterpriseServices;
using System.Runtime.CompilerServices;
using System.Reflection;
// Specify a name for the COM+ application.
[assembly: ApplicationName("MyMathService")]
// Specify a strong name for the assembly.
[assembly: AssemblyKeyFile("MyMathService.snk")]
namespace MyMathService
{
    [Transaction(TransactionOption.Required)]
    public class Maths: ServicedComponent
    {
        [AutoComplete]
        public int add(int a, int b)
        {
        return(a+b);
        }
    }
}
```

Software Security

ASP.NET

Web.config file is used to define the security of the Web pages

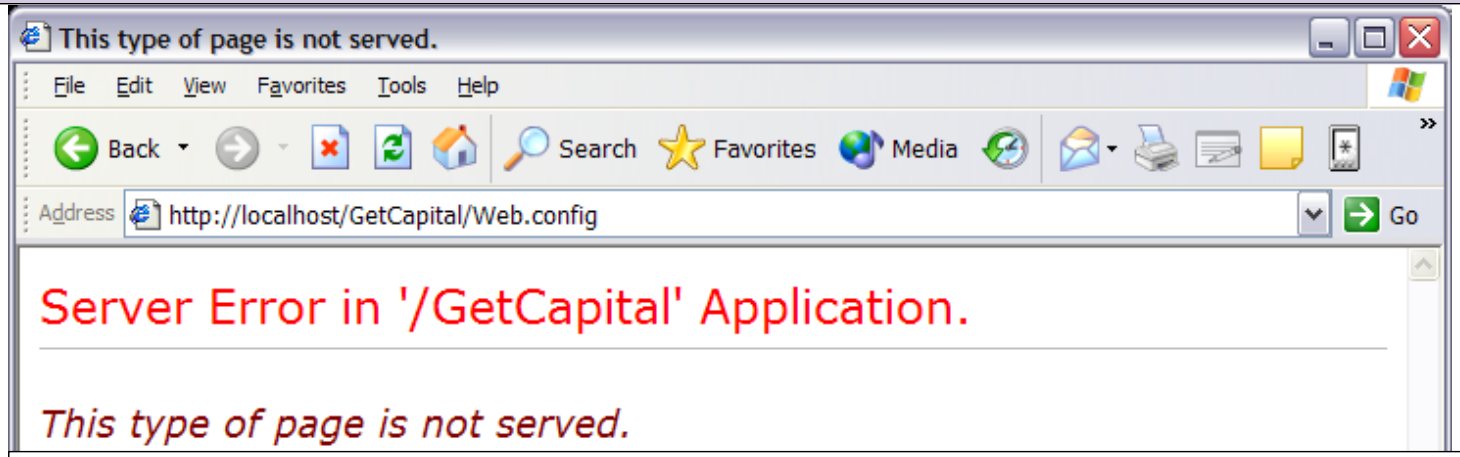
Web.config

Web pages

Web pages

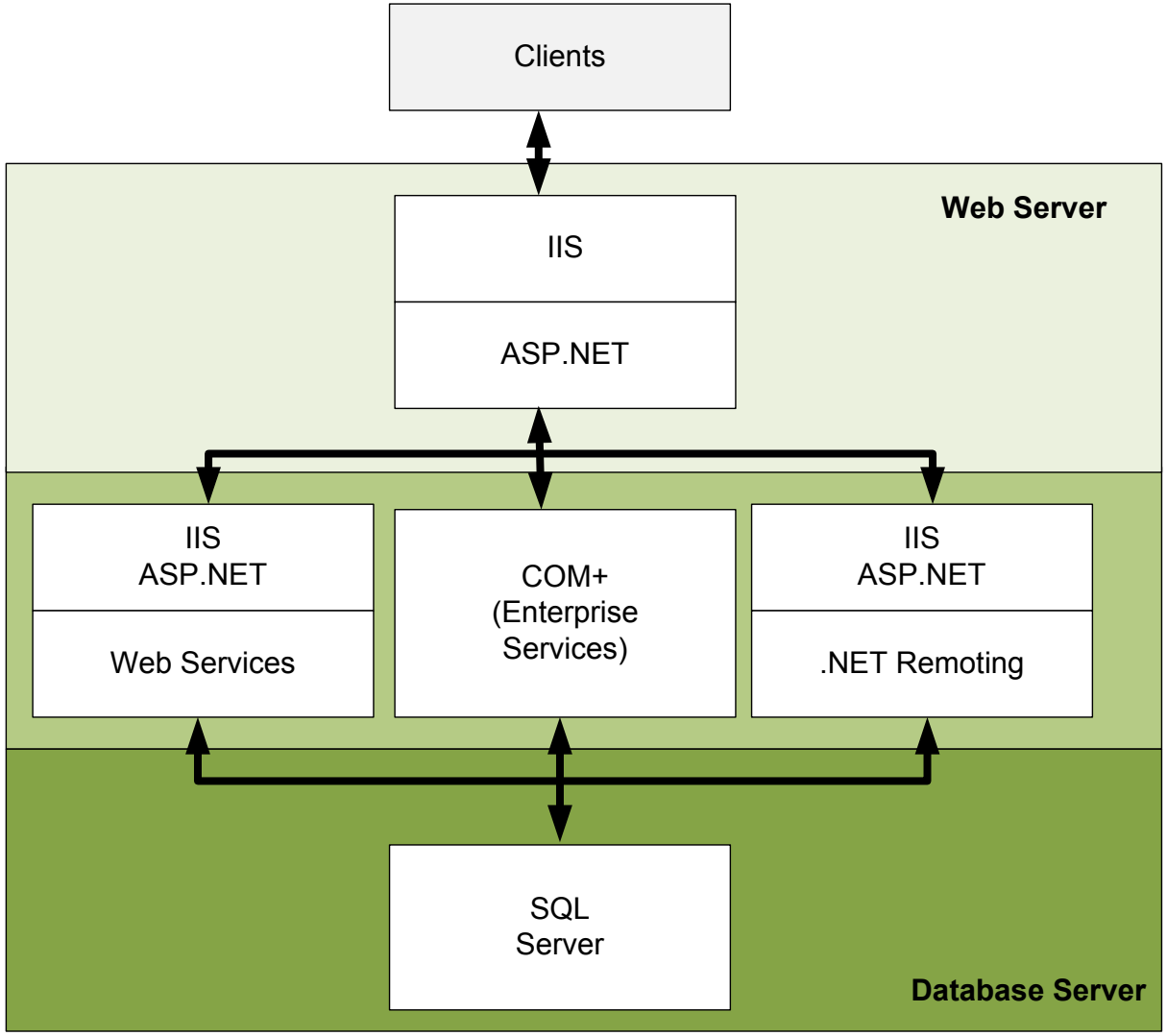
Web pages

```
<configuration>
  <system.web>

    <authorization>
      <deny users="?"/>
    </authorization>
    <!--
    <authentication mode="Forms">
    </authentication>
    -->

    <authentication mode="Forms">
      <forms name="Test" loginUrl="login.aspx" protection="All"
timeout="30" path="/">
        <credentials passwordFormat="Clear">
          <user name="fred" password="pass1"/>
          <user name="bert" password="pass1"/>
          <user name="napier" password="pass1"/>
        </credentials>
      </forms>
    </authentication>

    <compilation debug="true"/>
  </system.web>
</configuration>
```




```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <!--  DYNAMIC DEBUG COMPILATION
          Set compilation debug="true" to enable ASPX debugging.
                    Otherwise, setting this value to
          false will improve runtime performance of this application.
    -->
    <compilation
         defaultLanguage="c#"
         debug="true"
    />

    <!--  CUSTOM ERROR MESSAGES
          Set customErrors mode="On" or "RemoteOnly" to enable custom error
                    messages, "Off" to disable.
          Add <error> tags for each of the errors you want to handle.

          "On" Always display custom (friendly) messages.
          "Off" Always display detailed ASP.NET error information.
          "RemoteOnly" Display custom messages only to users not running
    -->
    <customErrors
    mode="RemoteOnly"
    />
```


Web.config



IIS
Authentication:
Anonymous, Basic, Digest, Integrated and Certifications
Authorization:
NTFS Permissions, IP Restrictions

ASP.NET
Authentication:
Windows, Forms, Password, None
Authorization:
URL Authorization, File Authorization and .NET Role

- Authentication (to identify the clients of your application)
- Authorization (to provide access controls for those clients)
- Secure communication (to ensure that messages remain private and are not altered by unauthorized parties)

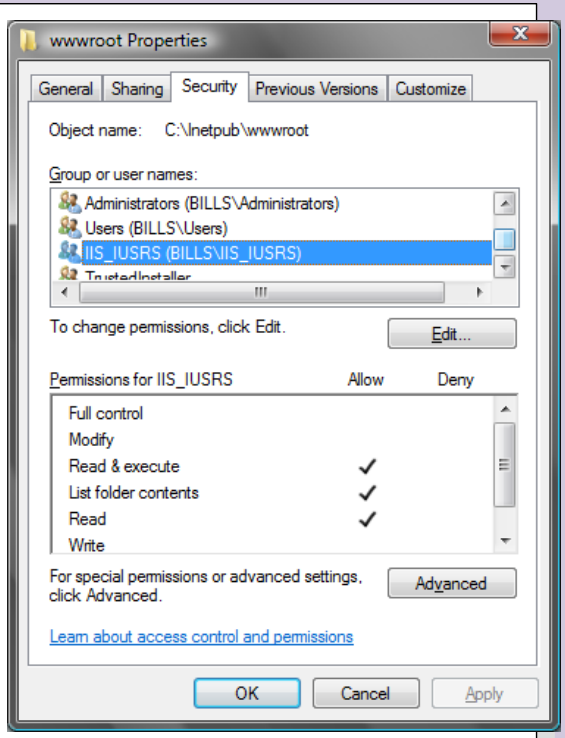
Client

Web server

Impersonation

Local Reource

```
<identity impersonate="true" />
```



For security, only one hop is allowed

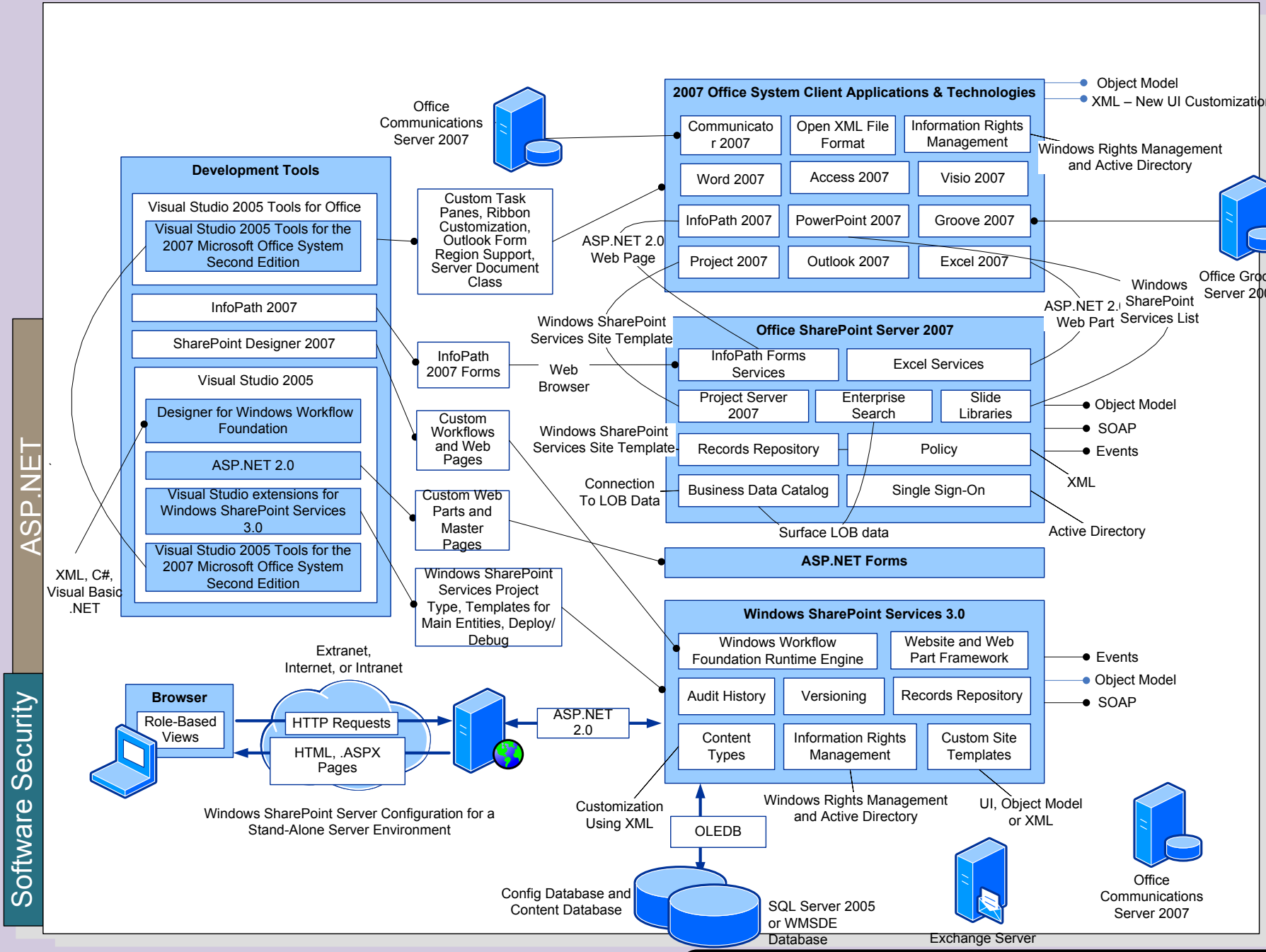
Client

Web server

Delegation

Remote Resource

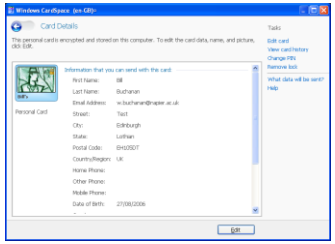
Impersonation and Delegation



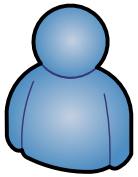
Microsoft Office SharePoint Server (MOSS)

Software Security

Kerberos



Identity



User

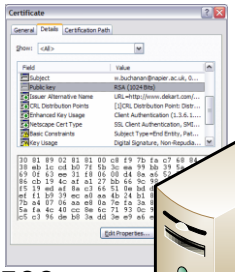
Identity selector

SAML (Security Assertion Markup Language)
Or Custom

WS-Security Policy
WS-Security

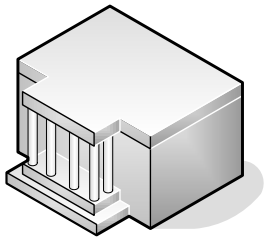
Relying Parity (RP)

Security Token Service (STS)



Identity Provider (IP)

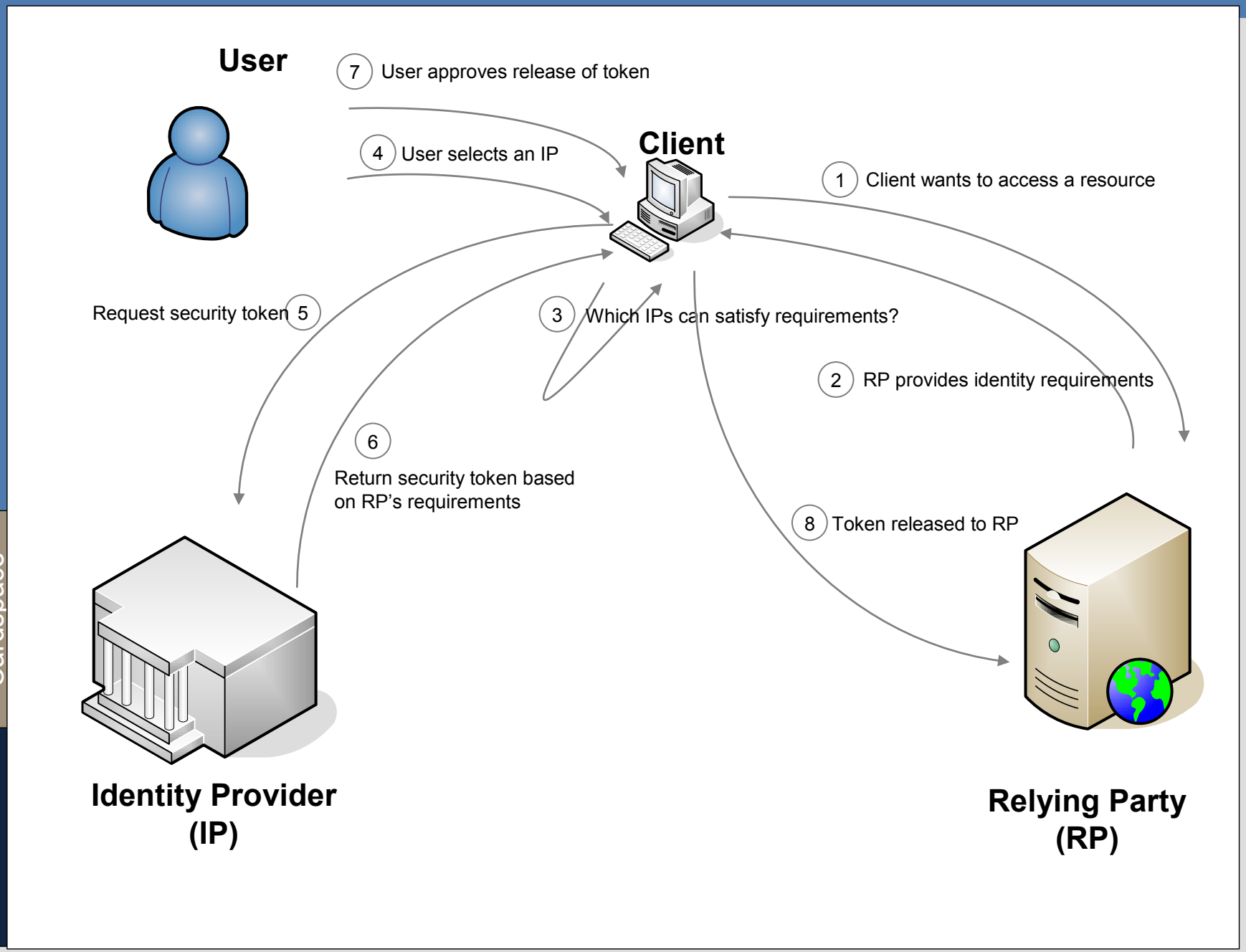
X509 Certificate



Kerberos

PKI Server
Digital Certificate Granter (Verisign)

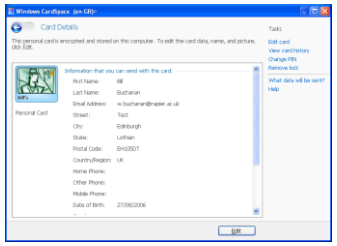
Open XML standards:
WS-*:-
WS-Trust, WS-Metadata Exchange Framework


User
7 User approves release of token
4 User selects an IP
Client
1 Client wants to access a resource
Request security token 5
3 Which IPs can satisfy requirements?
2 RP provides identity requirements
6
Return security token based on RP's requirements
8 Token released to RP
Identity Provider (IP)
Relying Party (RP)
Cardspace
Authentication

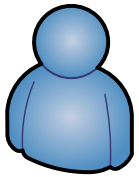
RP and IP

Cardspace

Authentication



Identity



User

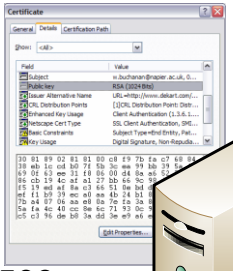
Identity selector

SAML (Security Assertion Markup Language)
Or Custom

WS-Security Policy
WS-Security

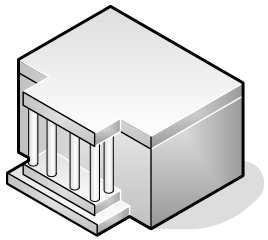
Relying Parity (RP)

Security Token Service (STS)



Identity Provider (IP)

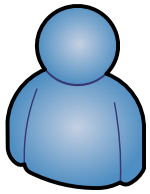
X509 Certificate

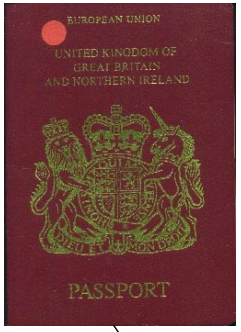


Kerberos

PKI Server
Digital Certificate Granter (Verisign)

Open XML standards:
WS-*:-
WS-Trust, WS-Metadata Exchange Framework





Token

STS (Security Token Service)

Check in

Trust

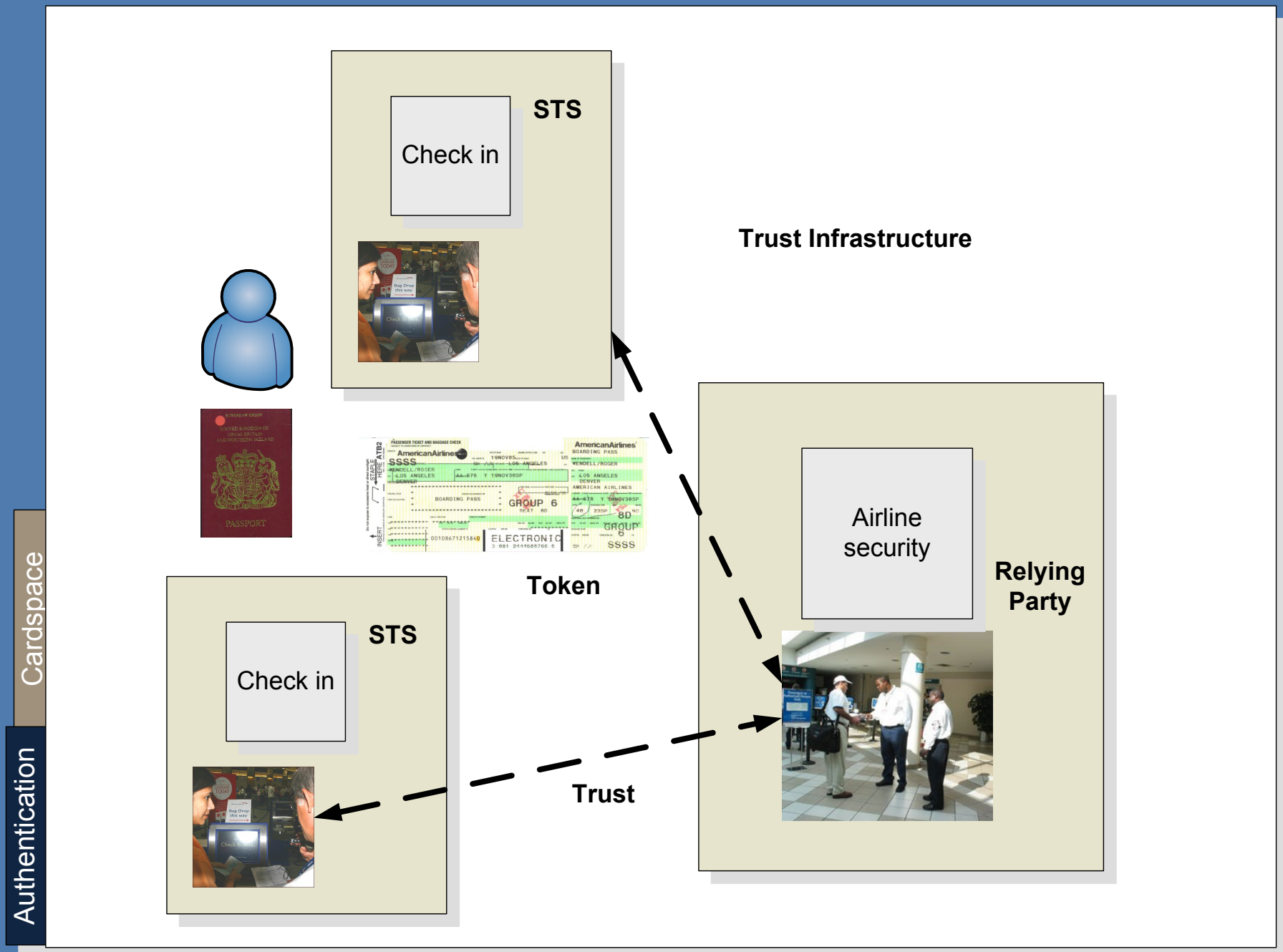
Airline security

Relying Party

Cardspace

Authentication

RP and IP

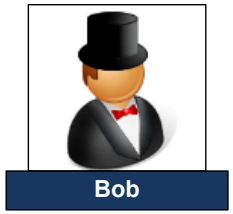

STS
Check in
Trust Infrastructure
Token
Relying Party
Airline security
STS
Check in
Trust
Cardspace
Authentication

RP and IP

AS_REQ is the initial user authentication request. This message is directed to the KDC component known as Authentication Server (AS).

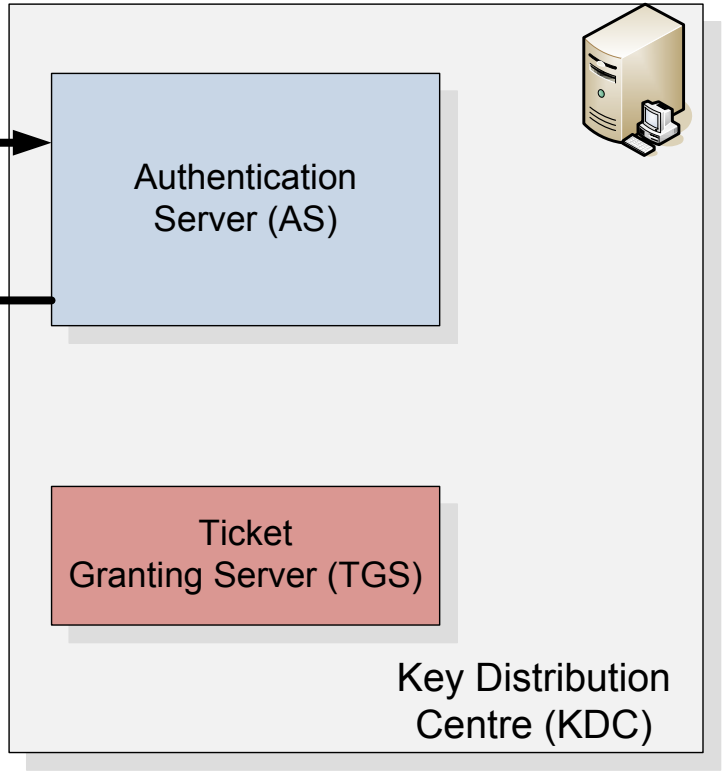
AS_REQ = ($Principal_{Client}$,$Principal_{Service}$,IP_list,Lifetime)

Eg $Principal_{Client}$ = Principal for user (such as fred@home.com), IP_list = all IP address which will use the ticket (may be null if behind NAT), lifetime = require life of the ticket.



AS_REQ

AS_REP



AS_REP. Reply for the previous request. It contains the TGT (Ticket Granting Ticket - encrypted using the TGS secret key) and the session key (encrypted using the secret key of the requesting user).

TGT = ($Principal_{Client}$,krbtgt/REALM@REALM,P_list,Timestamp,Lifetime,SK_{TGS})

AS_REP = { $Principal_{Service}$,Timestamp,Lifetime,SK_{TGS} }K_{User} { TGT }K_{TGS}

SK_{TGS} – Session key of the TGS – randomly created.

K_{TGS} – Key of TGS.

K_{user} – Secret key of Bob.

Note:

{ Message } – The curly brackets identify an encrypted message.

(Message) – The round brackets identify an non-encrypted message.

$AS\_REQ = (Principal_{Client}, Principal_{Service}, IP\_list, Lifetime)$

AS_REQ

$TGT = (Principal_{Client}, krbtgt/REALM@REALM, P\_list, Timestamp, Lifetime, SK_{TGS})$

$AS\_REP = \{ Principal_{Service}, Timestamp, Lifetime, SK_{TGS} \}K_{User} \{ TGT \}K_{TGS}$

AS_REP

Authentication Server (AS)

Bob now needs a service ticket

TGS_REQ

$Authenticator = \{ Principal_{Client} , Timestamp \}SK_{TGS}$

$TGS\_REQ = (Principal_{Service} , Lifetime , Authenticator) \{ TGT \}K_{TGS}$

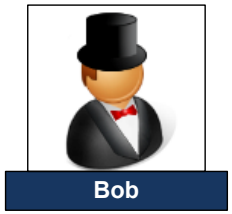
TGS_REP

Ticket Granting Server (TGS)

$T_{Service} = (Principal_{Client}, Principal_{Service}, IP\_list, Timestamp, Lifetime, SK_{Service})$

$TGS\_REP = \{Principal_{Service}, Timestamp, Lifetime, SK_{Service} \}SK_{TGS} \{ T_{Service} \}K_{Service}$

Key Distribution Centre (KDC)



SK_{TGS} – Session key of the TGS – randomly created.

K_{TGS} – Key of TGS.

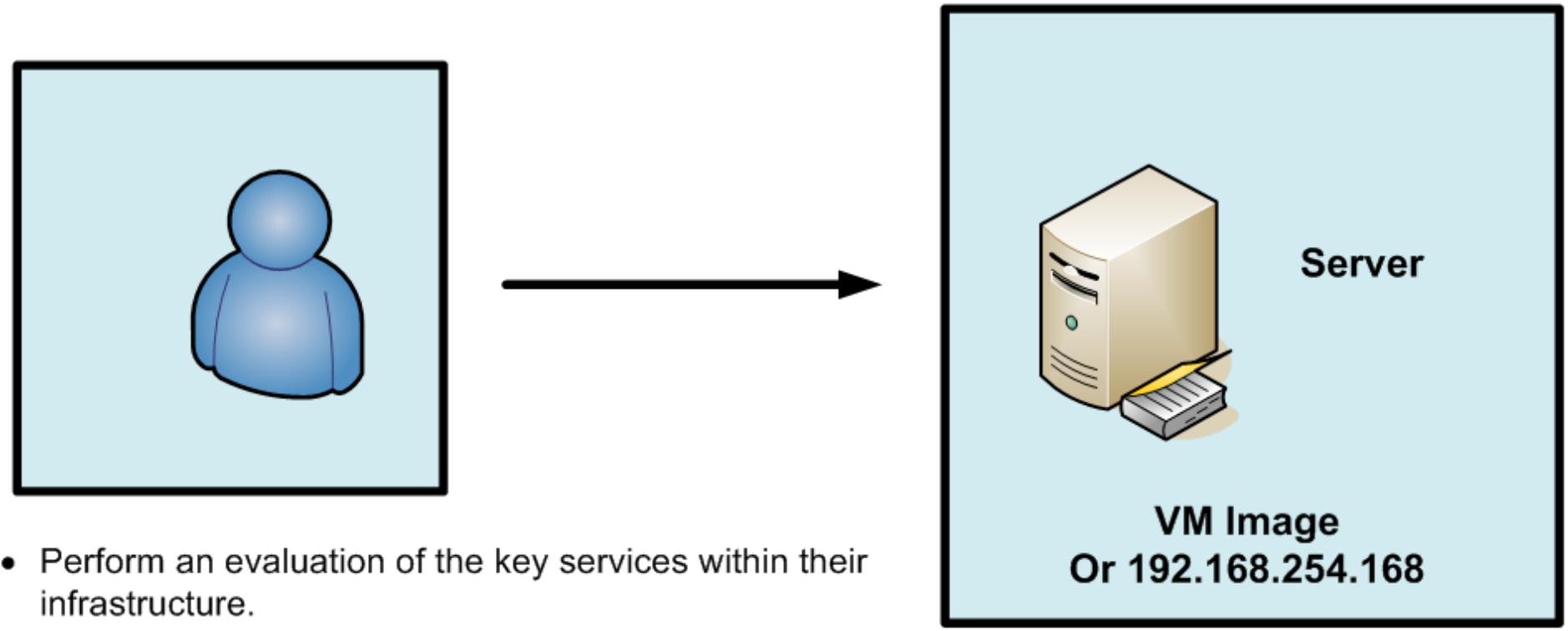
K_{user} – Secret key of Bob.

$SK_{Service}$ – Secret key of the service

Software Security

Conclusions

- The aim is for you to create a prototype of a system which outlines how the system could work. For this you should implement an agent-based system, either using: your own agent (using Winpcap and .NET or Java); a stand-alone version of Snort; or using a graphical management system which interfaces to Snort (a mixture of Snort, Winpcap and .NET). Overall the alerts should be useful, and, possibly, stored in a secure manner.



- Perform an evaluation of the key services within their infrastructure.
- Develop and implement a strategy to detect the networking scanning of their system.
- Develop and implement a strategy to detect activities which involves the login of an administrator through Telnet or FTP.
- Develop and implement a strategy to detect a malicious Bot agent (to be given).