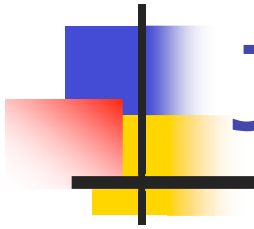


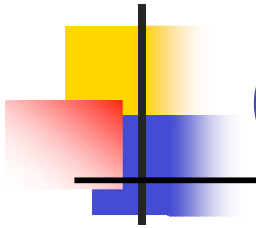
An Introduction to .NET for the J2EE Programmer

Jeroen Frijters

Sumatra Software b.v.

jeroen@sumatra.nl

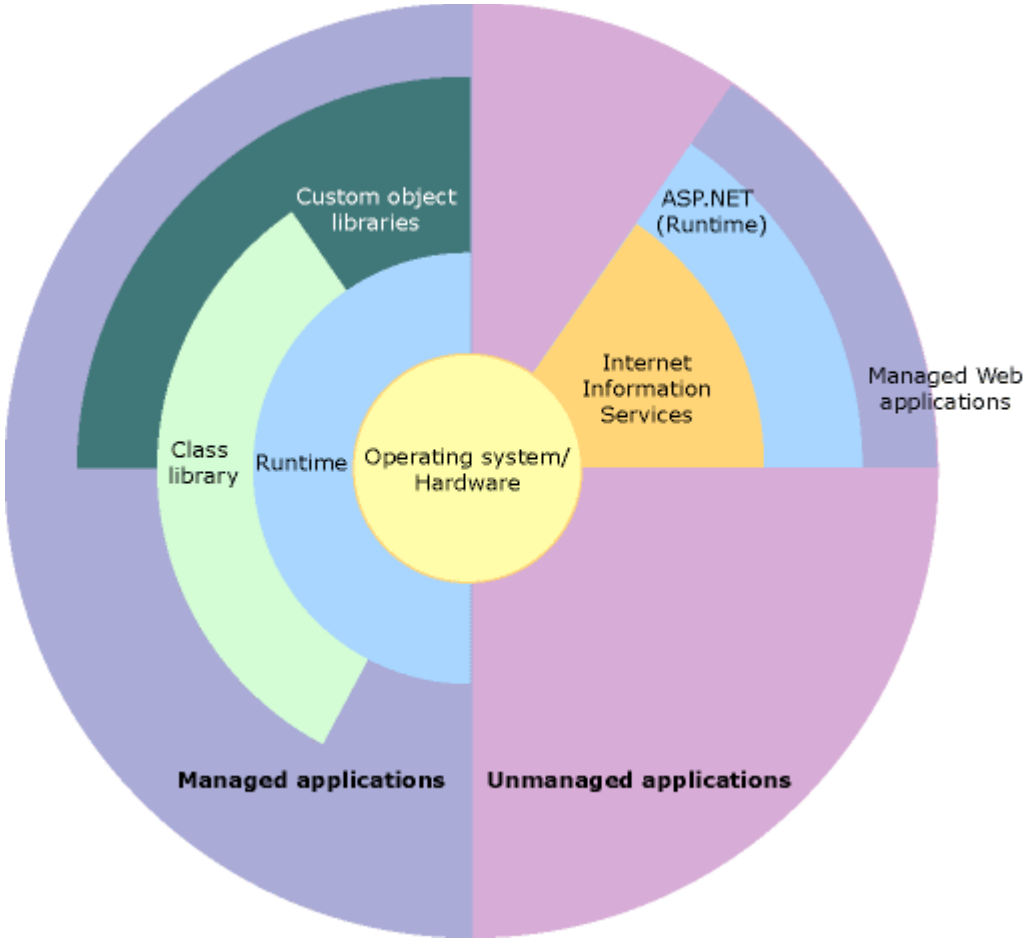
http://weblog.ikvm.net/



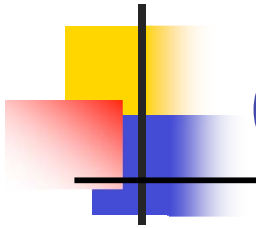
Overview

- .NET Framework overview and terminology
- A Quick Look at C#
- A detailed look at the important topics
  - Class Libraries
  - Component Model
  - Application Servers
  - Code Packaging / Deployment
  - Versioning
  - Security Model
  - Integration with Legacy Code
  - Portability

.NET Framework Overview and Terminology

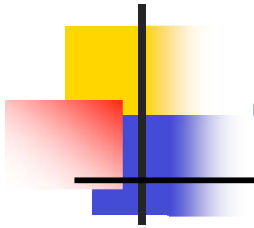


Source: Microsoft Corp.



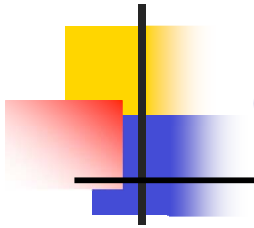
.NET Framework Overview and Terminology 2

- .NET Framework
- TLAs
  - CLI – Common Language Infrastructure
  - CLR – Common Language Runtime
  - CLS – Common Language Specification
  - CTS – Common Type System
  - VES – Virtual Execution System
  - CIL – Common Intermediate Language



Common Language Infrastructure

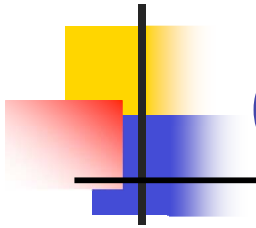
- ECMA/ISO standard
- CLR is the name for Microsoft's implementation
- Defines the Common Type System (CTS), the Virtual Execution System (VES) and the Common Language Specification (CLS)



Common Language Runtime

- Microsoft ships several languages
  - C#
  - VB.NET
  - C++ (with managed extensions)
  - JScript.NET
  - J#
  - ILasm
- Third party languages
  - Eiffel by Eiffel Software
  - COBOL by Fujitsu
  - FORTRAN by Lahey/Fujistu
  - Delphi by Borland
  - Perl by ActiveState
  - Many research/toy languages

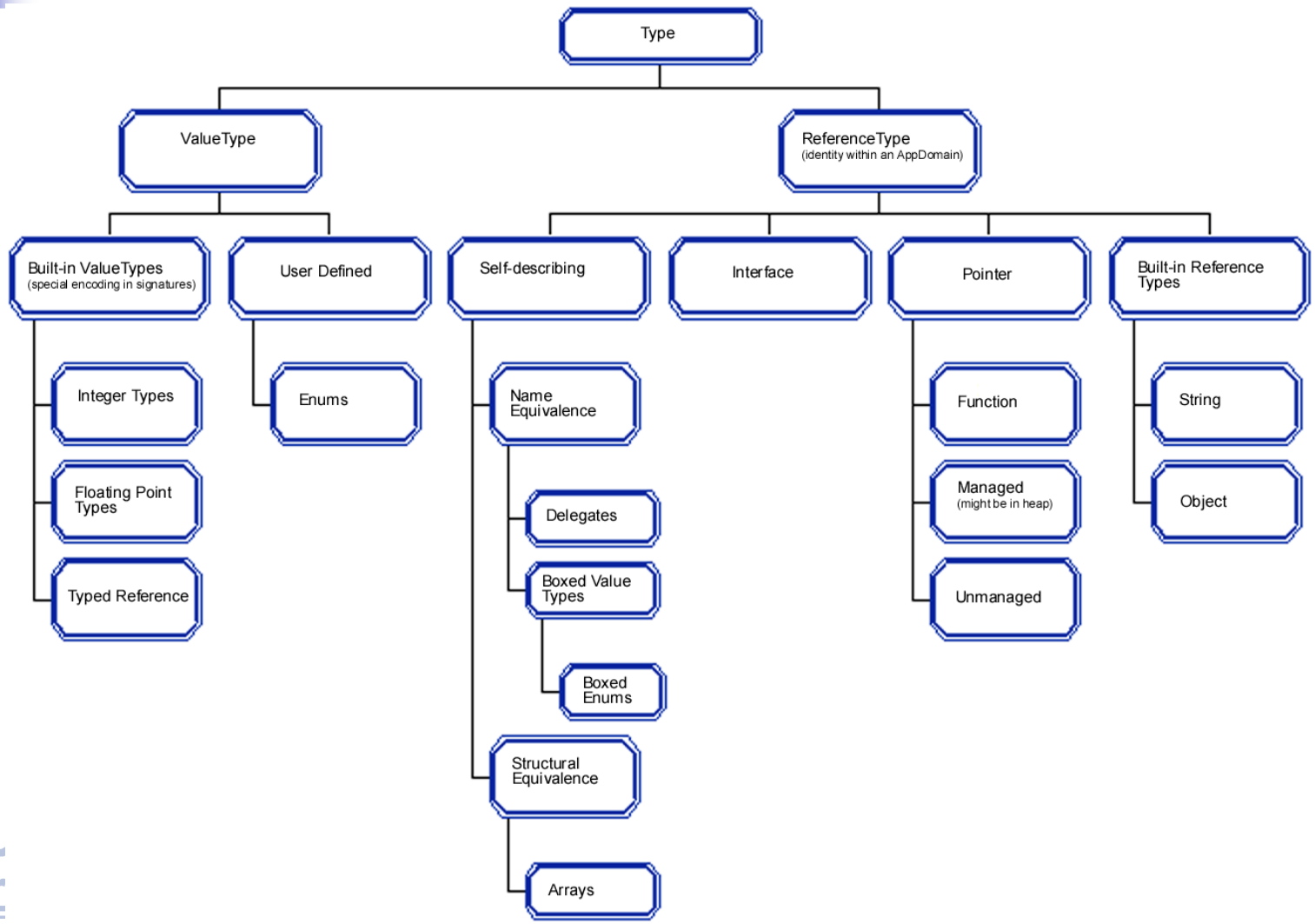
http://www.jasonbock.net/dotnetlanguages.html

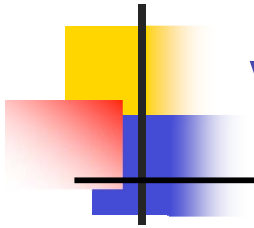


Common Language Specification

- CLS Framework
  - A library consisting of CLS-compliant code
- CLS Consumer
  - A language or tool that is designed to allow access to all of the features supplied by CLS-compliant frameworks, but not necessarily be able to produce them
- CLS Extender
  - A language or tool that is designed to allow programmers to both use and extend CLS-compliant frameworks

Common Type System

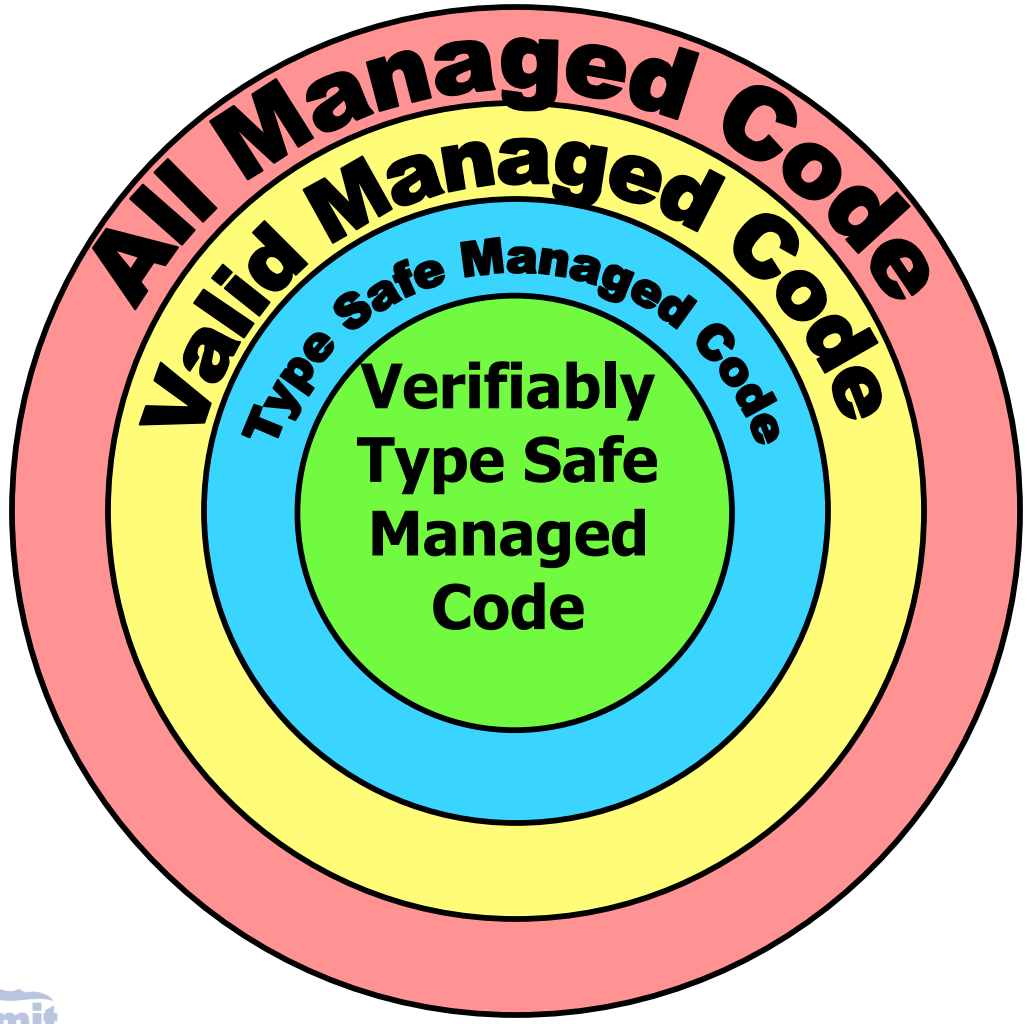




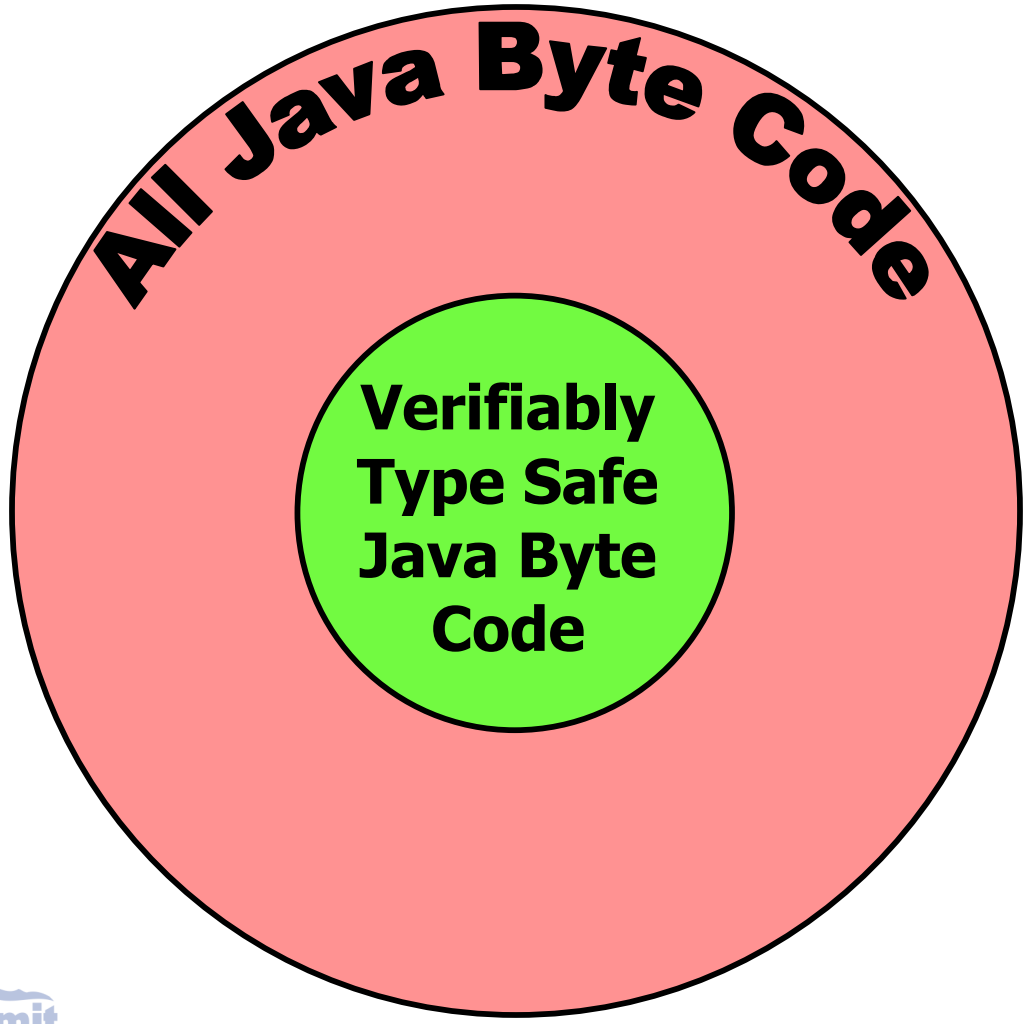
Virtual Execution System

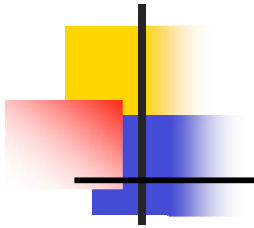
- An abstract runtime model (the equivalent of the Java Virtual Machine).
- Common Intermediate Language (CIL)
  - Not designed to be interpreted.
  - Has direct support for manipulating unmanaged data.
  - Has direct support for calling unmanaged code.
- Provides Application Domains and Contexts and remoting support across both of these boundaries.

Managed Code



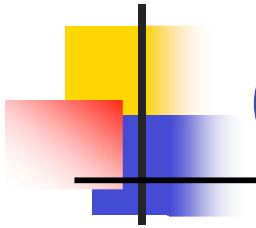
Java Byte Code





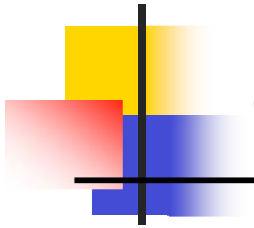
Managed Data

- Like Java, most code uses Managed Data. That is data that is managed by the runtime either on the stack or *via* the garbage collector.
- Unlike Java, trusted code can use unmanaged data (*i.e.* data outside of the control of the runtime), this can be used for interoperating with existing unmanaged code or for efficiency.



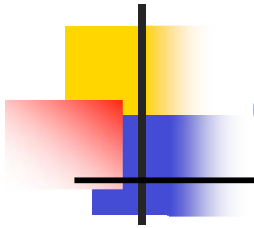
C#

- Not just a copy of Java
- Syntax very much like C++ and Java
- Fixes some Java flaws (*e.g.* return from finally)
- Additional Features (C# 1.1 *vs* Java 1.4)
  - Custom Attributes, Value Types, Unsafe Code, Delegates, Enums, Internal Access Modifier, Properties, Events, Conditional Compilation, Operator Overloading, Boxing, Variable Argument Lists, Checked Integer Arithmetic
- Java 5.0 adds some of the C# features
  - Annotations, Enums (but different), Boxing, Variable Argument Lists.



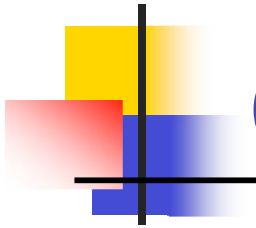
Class Libraries

- Like Java, the .NET Framework has a rich set of class libraries.
- The class library is a little bit more consistent than in Java.
- Major Parts
  - Base Framework
  - Data and XML
  - ASP.NET
  - Windows Forms



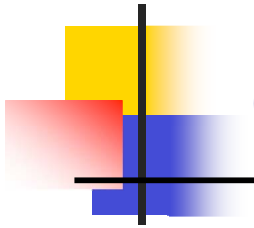
Class Libraries – Base Framework

- System
  - Collections
  - Configuration
  - Diagnostics
  - Globalization
  - IO
  - Net
  - Reflection
    - Emit
  - Resources
- System
  - Security
  - ServiceProcess
  - Text
  - Threading
  - Runtime
    - InteropServices
    - Remoting
    - Serialization



Class Libraries – Data and XML

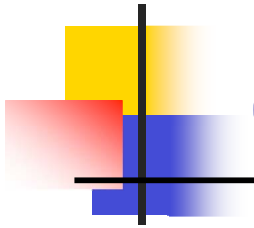
- System.Data
  - OleDb
  - Common
  - SqlClient
  - SqlTypes
- System.Xml
  - Xslt
  - XPath
  - Serialization



Class Libaries – ASP.NET

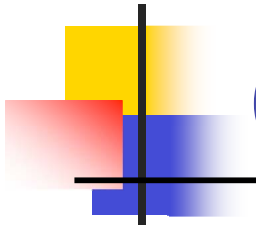
- System.Web
  - Services
    - Description
    - Discovery
    - Protocols
  - UI
    - HtmlControls
    - WebControls
  - Caching
  - Configuration

- System.Web
  - Security
  - SessionState



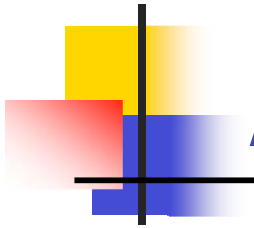
Class Libraries – Windows Forms

- System.Windows.Forms
  - Design
  - ComponentModel
- System.Drawing
  - Drawing2D
  - Imaging
  - Printing
  - Text



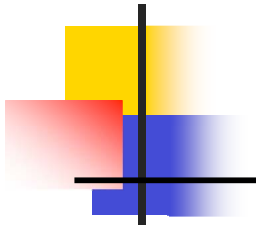
Component Model

- Very nice component model, including design time support infrastructure.
- Demonstration!



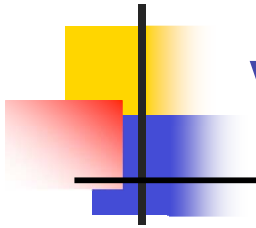
Application Servers

- ASP.NET
  - Web sites
  - Web services
  - Typically hosted by IIS, but can also be hosted inside other HTTP servers.
- COM+
- Windows Services
- Like the JVM, you can host the CLR inside your own application.



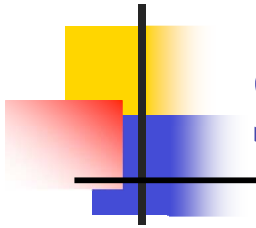
Packaging / Deployment

- Assemblies
  - Extension of the Windows executable file format (PE files).
  - Can contain both managed as well as unmanaged code.
  - GAC / NGEN
- xcopy deployment
  - ASP.NET sites can be deployed by copying a directory. DLL files can be overwritten.
  - Many client applications can be xcopy deployed as well.



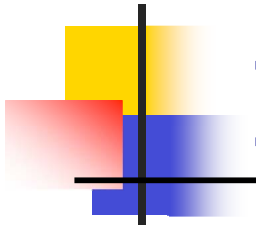
Versioning

- Assemblies can have a "strong name" and a version number.
- Publisher can set versioning policy.
- Domain/Local administrator can set versioning policy.
- Multiple versions can be loaded side-by-side.
- Demonstration!



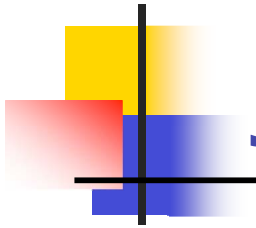
Security Model

- Code Access Security
  - Declaritive
  - Imperative
- Link Demands
- Isolated Storage
- Partially trusted code can read/write files through by using the standard file dialogs.
- Demonstration!



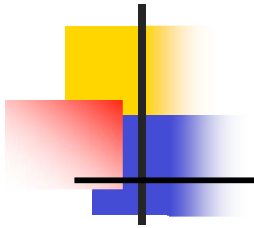
Integration with Legacy Code

- P/Invoke
  - Efficient and relatively easy way to call existing native code, without having to write glue code.
  - Demonstration!
- Managed C++ / C++/CLI
  - Very powerful way to write glue code for complex object models.
- COM interop
  - COM interfaces and objects can be used by .NET code and vice versa.



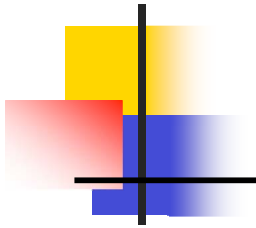
Java on the .NET Framework

- J#
  - JDK 1.1.4 only
  - Primarily a migration path for J++ users
- Third Party Tools
  - Come see my other talk!



Portability

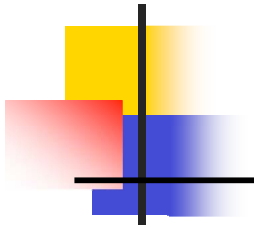
- CLR
  - Windows x86/x64/IA64
- Compact Framework
  - Windows CE
- Shared Source CLI a.k.a. Rotor
  - Windows x86, FreeBSD, Mac OS X
- Mono
  - Windows x86, GNU/Linux x86/PPC, Mac OS X
- Portable.NET
  - GNU/Linux x86



Future

- Somewhere in 2005, version 2.0 of the .NET Framework 2.0 will be released.
  - Generics
  - C# language enhancements (generics, anonymous methods, iterators)
  - Many class library enhancements
  - C++/CLI
- The next version of Windows (Longhorn) will have an integrated CLR and many operating system components will be written as managed code.

Questions



Resources

- http://www.sscli.net/
- http://www.jasonbock.net/dotnetlanguages.html
- http://www.mono-project.com/
- http://www.dotgnu.org/
- http://weblog.ikvm.net/
- Microsoft has tons of interesting weblogs: http://blogs.msdn.com/