

# Microsoft .Net Overview

PROCASE Consulting  
March 20, 2005

Gordon Yan  
gyan@ProcaseConsulting.com

# Agenda

- Web Technology Overview
- Microsoft .Net
  - Managed Execution
  - Development
  - System Integration
  - Deployment
- .Net Case Study

# Web Technology Overview

- ASP / COM
  - Was the first popular web solution
- Java
  - After ASP/COM, better architecture
- .Net
  - Get the best out from ASP/COM and Java
- Other Web Technologies
  - Oracle AS, PHP, ColdFusion

# Web Technology Overview

- Similar High Level Structure
  - Logical 3 tiers
  - Physical 2 tiers or 3 tiers
  - Similar Code & Deployment Structure

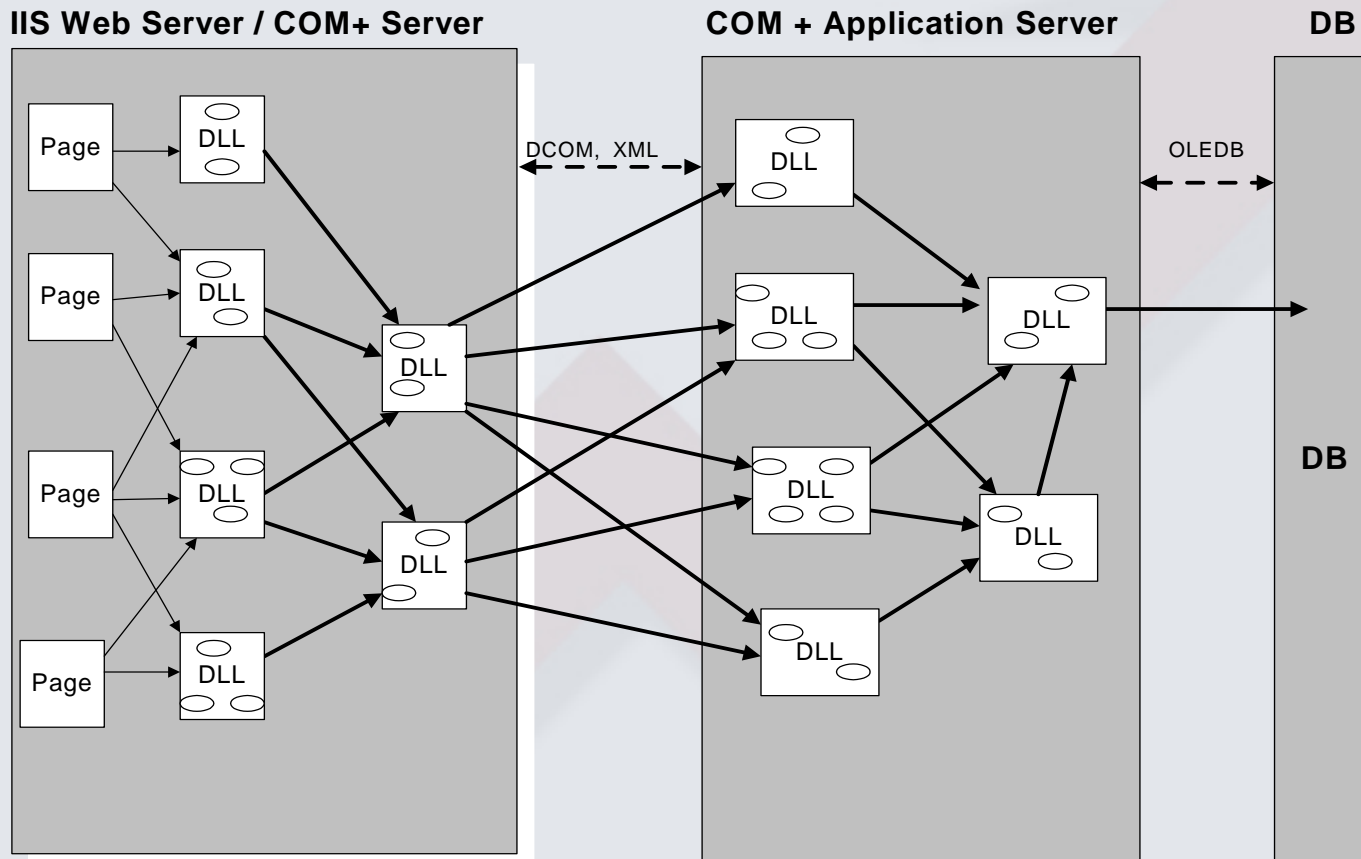
ASP	COM	DCOM	
JSP	EJB	RMI	Web Service
ASPX	Assembly	Remoting	Web Service

- Very Different Implementation

# Microsoft ASP/COM

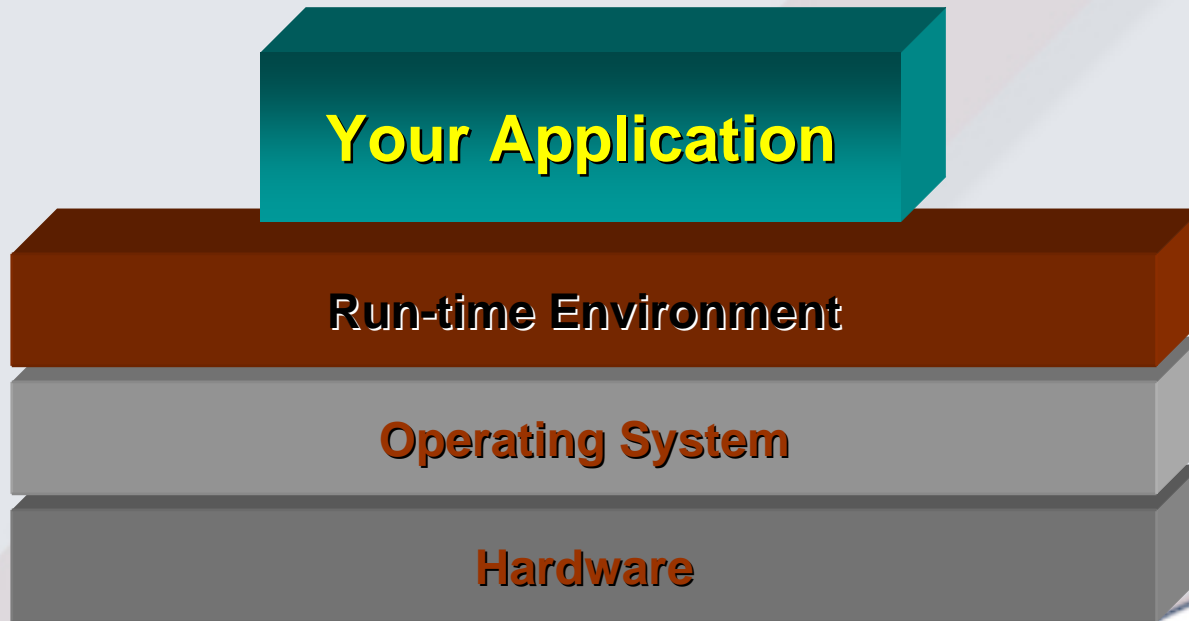
- ASP = VB Script + Java Script + HTML
  - Interpret Execution
- COM: Compiled DLL, VB or C++
  - MTS
  - COM+
- DCOM
  - TCP Communication for Remote Scenario

# PC 3 Tier ASP/COM Architecture



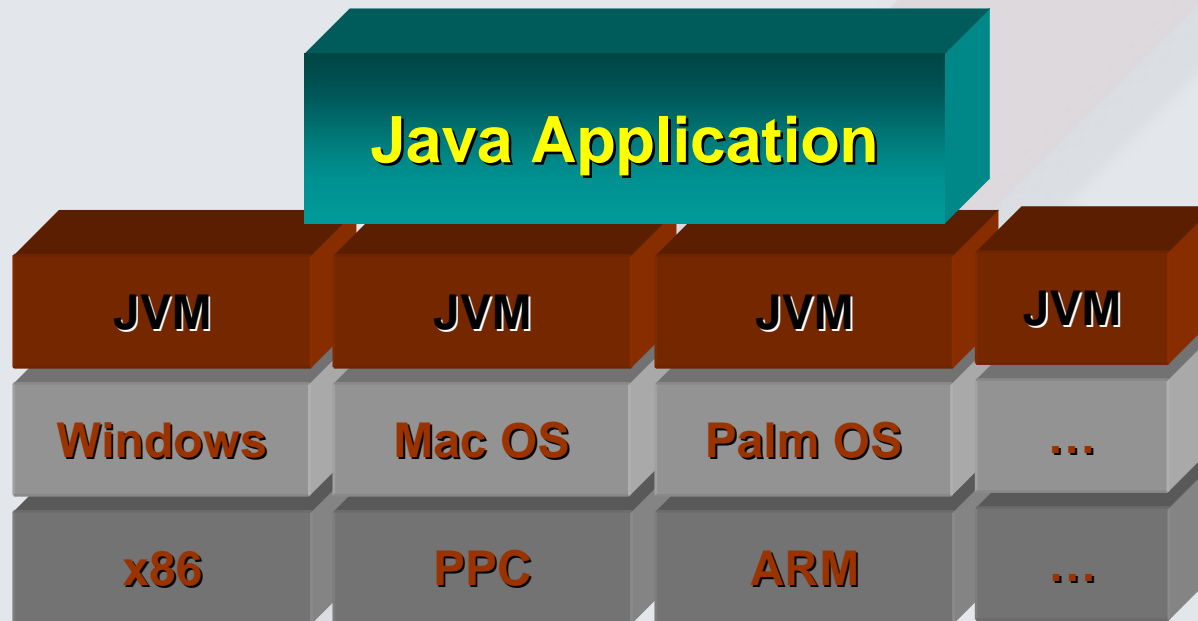
# .Net Managed Execution

- Idea:
  - modern software executes within run-time environment
  - why? portable and safer execution...



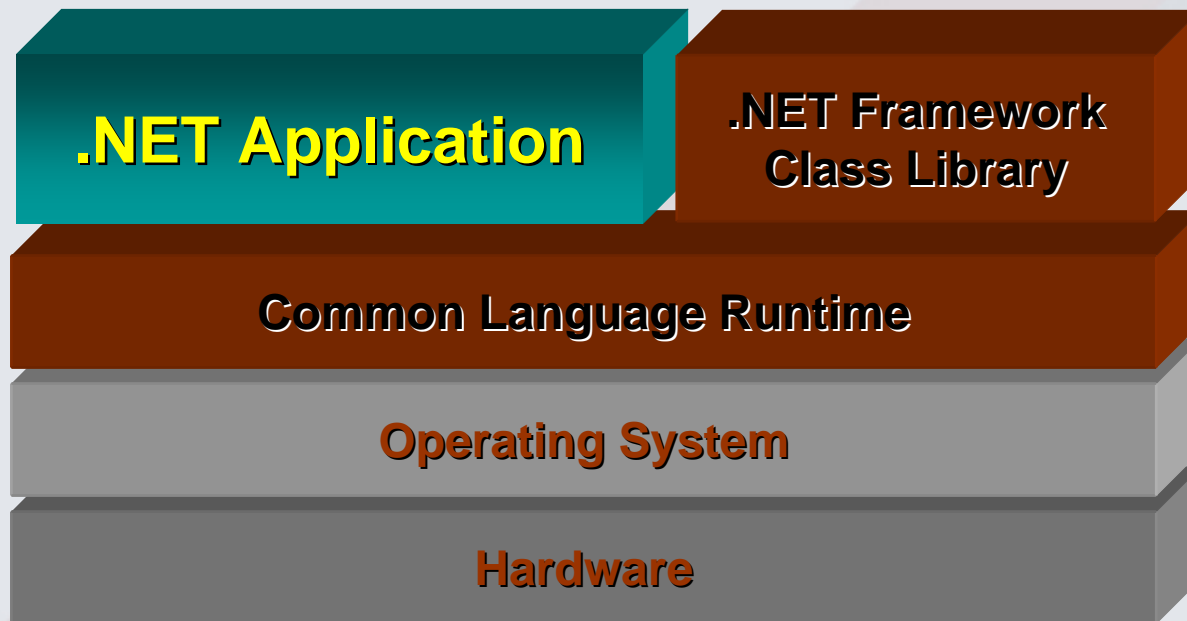
# Java JVM

- Based on run-time environment called *JVM*
  - JVM = Java Virtual Machine



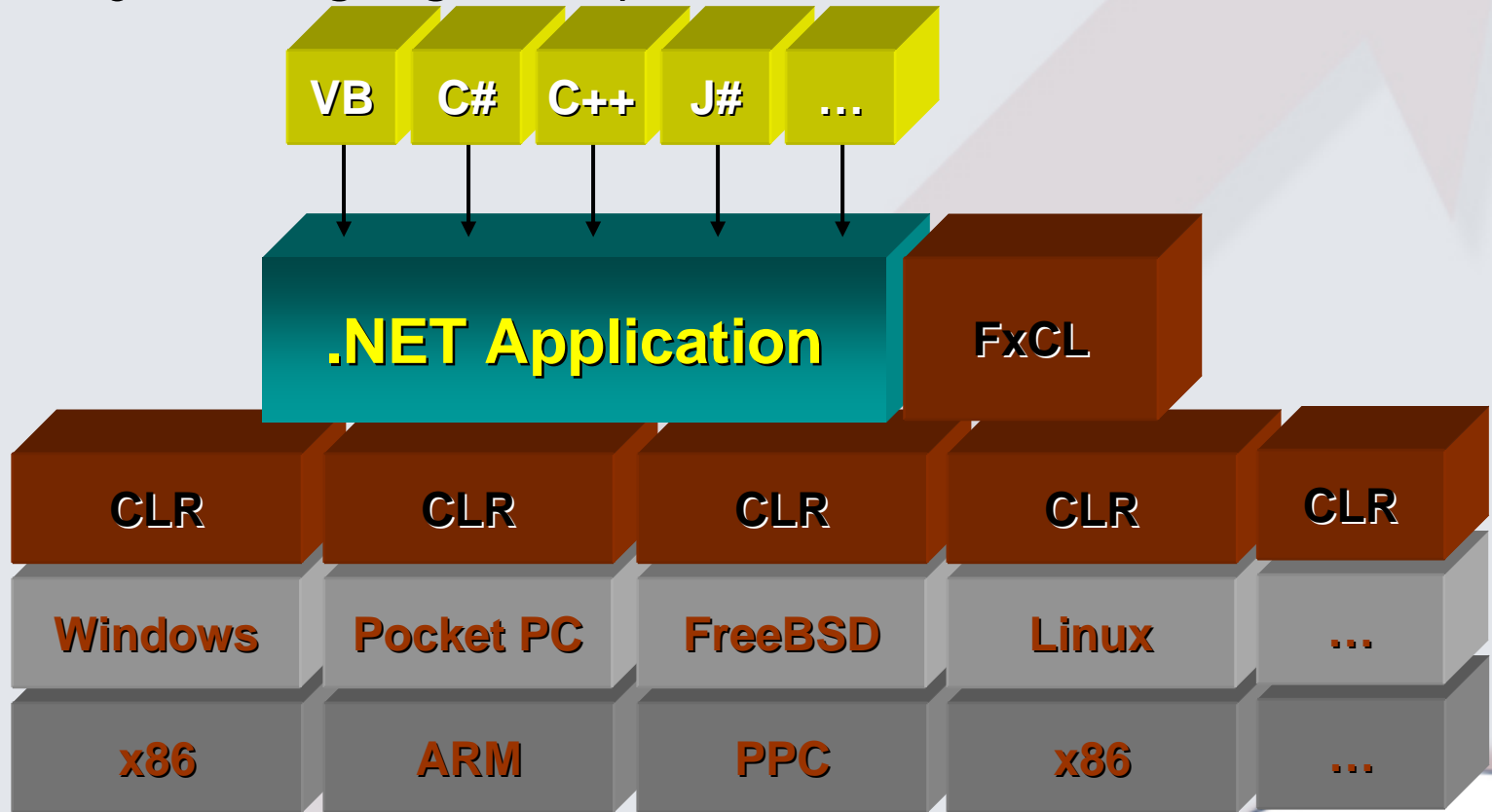
# .NET CLR

- Based on *CLR* and *FxCL*
  - CLR = Common Language Runtime
  - FxCL = Framework Class Library



# .NET CLR

- Pick your language and platform...



# .Net Development

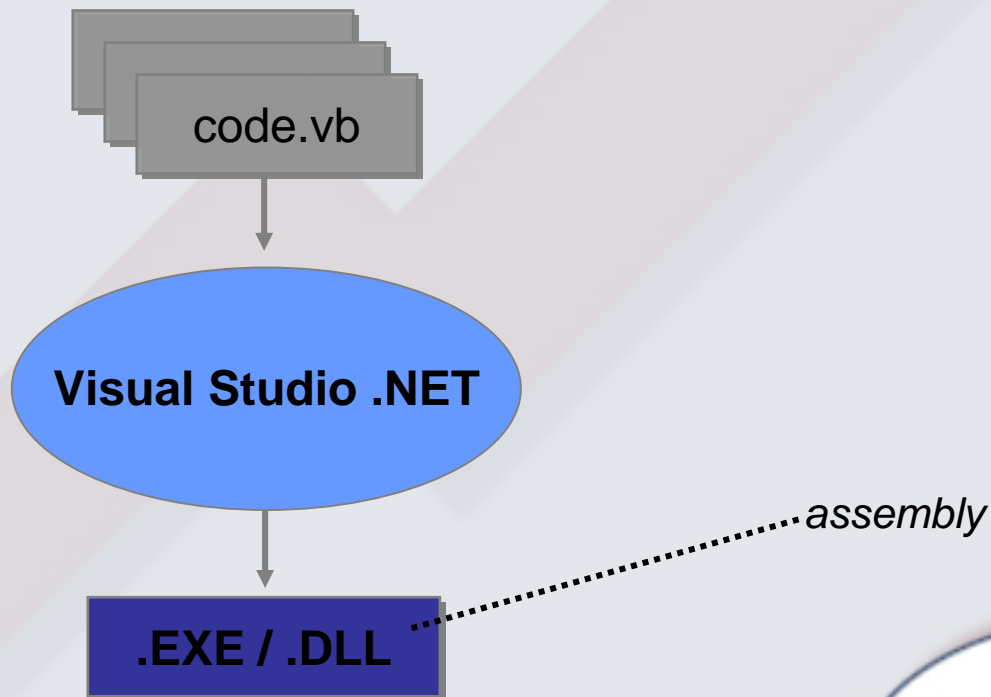
- Code behind -- Compiled code
  - No more spaghetti code, fast
- Automatic garbage collection
- IDE -- Everything is in one dev environment
- Debug is much easier
  - From page to assembly, from C# to VB.Net

# .Net Development

- Configurable application
- Work process auto recycle
  - More stable application
- Advanced session management
  - Easy to configure Web Farm
- Application blocks
  - Why build everything from scratch?

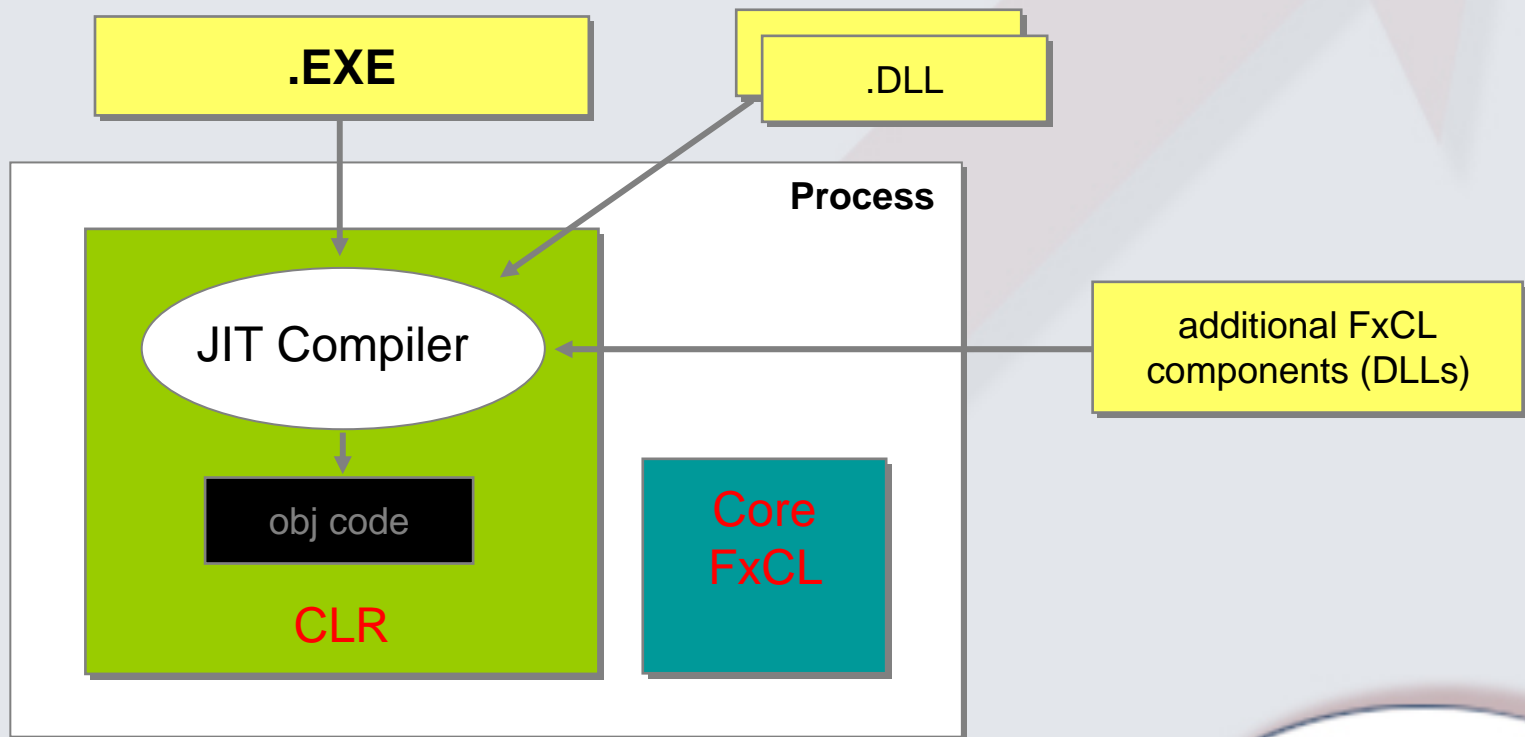
# Assemblies

- .NET components are called *assemblies*
- Unit of deployment in .NET
  - 1 assembly = 1 or more compiled source files



# Assembly Resolution

- CLR must be able to locate assemblies
  - our own + FxCL



# Assembly Resolution

- .NET figures out what version is needed
- .NET searches GAC (*Global Assembly Cache*)
- If not found and .config file is present then .NET searches where configured to
- If still not found then .NET searches folder containing .EXE
- If still not found then application terminates with error

# Implications

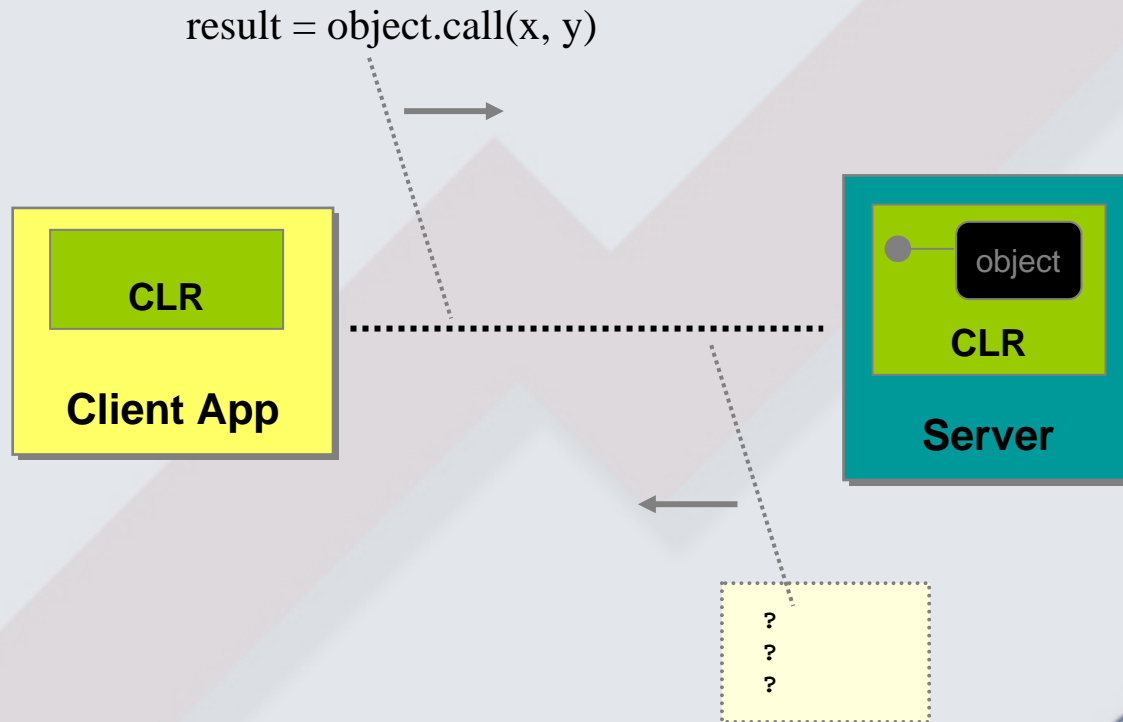
- No more registry!
  - .NET uses well-defined search path
- No more DLL hell!
  - applications never use the wrong version
  - unless you say so...
- **Config file hell?**
  - machine config, user config, app config, etc.

# .Net Application Integration

- XML web service – industry standard, platform independent
- .Net Remoting – Replace DCOM

# .Net Remoting

- App makes call, waits for proprietary data
  - .NET platform on both sides...



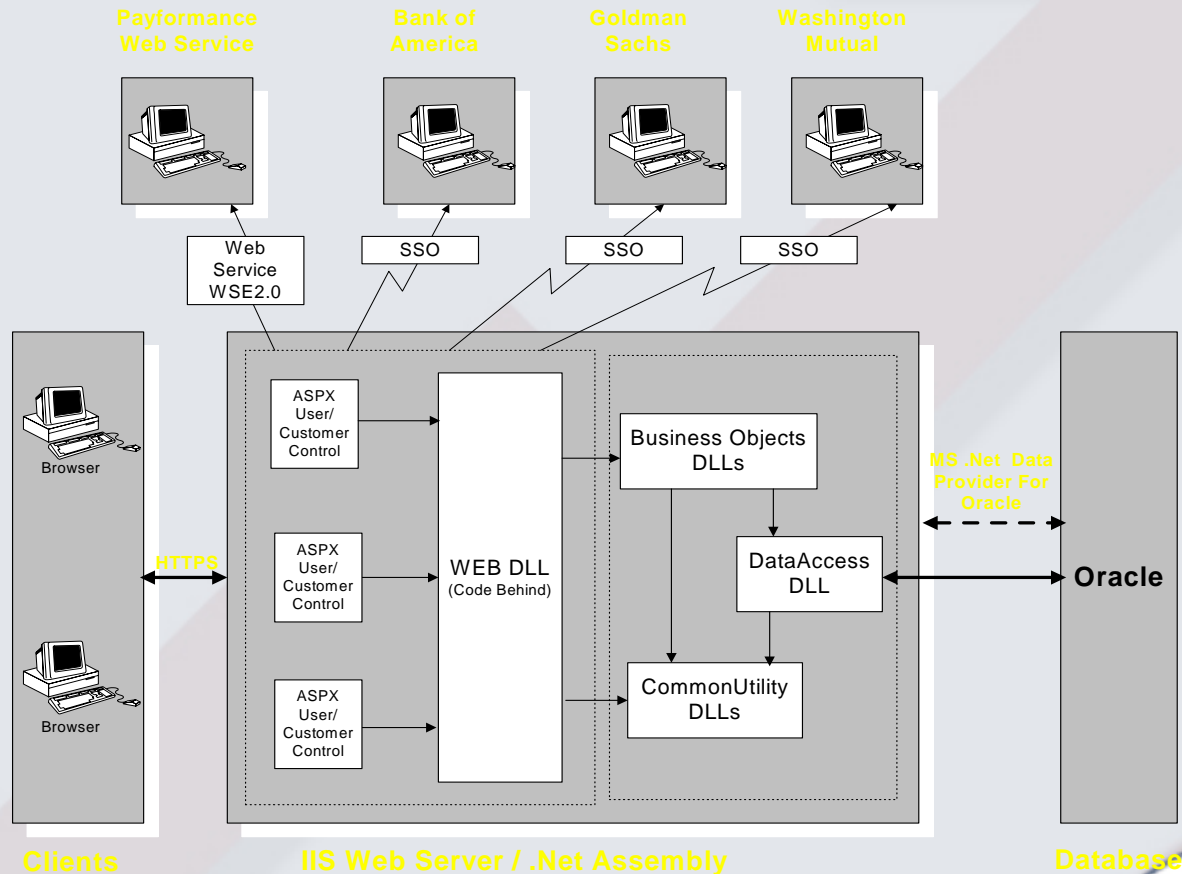
# .Net Deployment

- X-Copy Deployment
- Side By Side Deployment

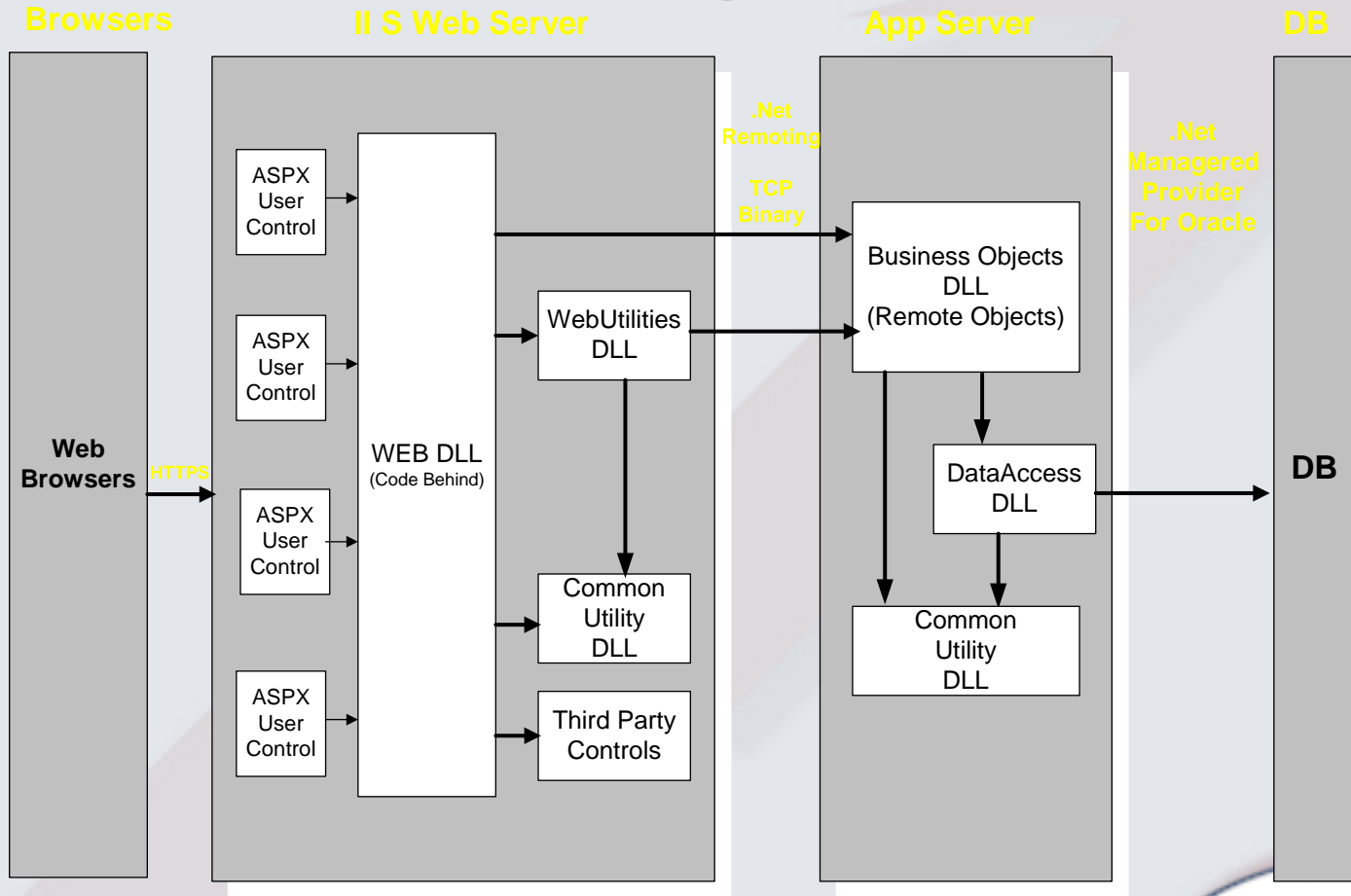
# Deployment Options

1. Install assemblies in same dir as .EXE
2. Install some in .EXE dir, remainder in GAC
  - ◆ GAC allows you to share assembly across apps
  - ◆ GAC allows you to install different versions
3. Custom deployment via .config file
4. "Zero-touch" deployment

# .Net Case Study – WW EE Site



# .Net Case Study – New EFIS



# Summary

- Portable and Safer execution
- Fast development & better performance
- Better system integration
- Easy deployment