.NET versus J2EE

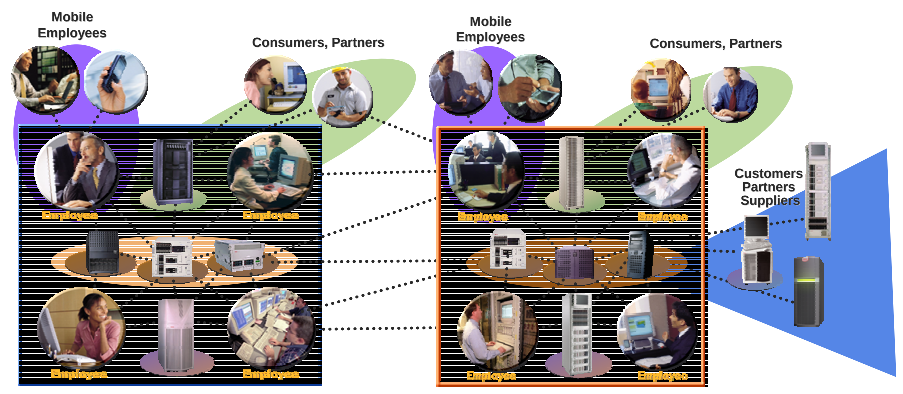
Felicia cheng
Jarred zheng
Jonathan Card
Peng Li
Xiao he

Agenda

- Background Introduction
- J2EE Structure
- .NET Structure
- J2EE vs. .NET
- Conclusions

Today's Enterprise Environment

Company A **Company B**



Challenges of Enterprise Application

- Distributed computing service
- High reliability, portability, security, and extendibility
- Integration with Existing System
- Complex customer requirements

Challenges of Enterprise Application Development

- Programming Productivity
- Time-to-Market
- Integration with Existing System
- Response to Demand
- Maintaining Security

What we need

- Web Services—implemented in an Enterprise solution Framework
- A Web Service:
  - receives a request formatted in XML from an application
  - performs a task
  - and returns an XML-formatted response.
- Web Services are delivered using open industry standards
  - Services to be described in WSDL
  - Services to located via UDDI
  - Data to be exchanged via XML
  - Protocols are HTTP and SOAP

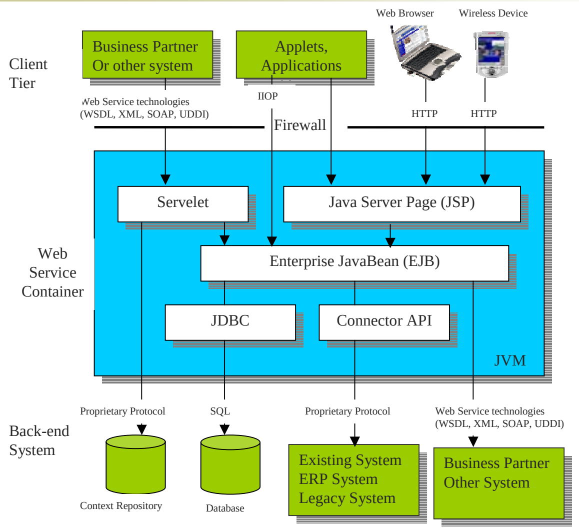
Agenda

- Background Introduction
- J2EE Structure
- .NET Structure
- J2EE vs. .NET
- Conclusions

What's J2EE

- Open and standard based platform for
- Developing, deploying and managing
- N-tier, Web-enabled, and component-based enterprise applications

J2EE Web Services model



J2EE 1.4 Contents

- J2SE
- JAX-RPC
- J2EE Management
- J2EE Deployment
- Servlet
- JSP
- EJB
- JMX, JMS, JavaMail, JACC, Connector,…

J2EE Platform Technologies

- J2EE specifies technologies to support multi-tier enterprise applications:
  - Component
  - Service
  - Communication

Component Technologies

- A component is an application-level software unit.
- J2EE supports following types of components;
  - Applets
  - EJB (Enterprise JavaBeans)
  - Web components
  - Application clients
  - Resource adapter components

EJB—Core of J2EE

- EJB hosts application-specific business logic and provides system-level services:
  - Transaction management
  - Concurrency control
  - Security
- EJB is a fundamental link between Web tier and EIS tier

EJB—Core of J2EE

- Entity Beans--An entity bean represents an object view of business data stored in persistent storage or an existing application
- Session Beans--used to implement business objects that hold client-specific business logic
- Message-Driven Beans--allows J2EE applications to receive JMS messages asynchronously

Service Technologies

- Service technologies allow applications to access a wide range of services.
  - JDBC API
  - Java Transaction API (JTA) and Service
  - Java Naming and Directory interface (JNDI)
  - J2EE Connector Architecture
  - Java API for XML Processing Technology (JAXP)

Communication Technologies

- Communication Technologies provide mechanism for communication between clients and servers and between collaborating objects hosted by different servers
  - Internet Protocols(TCP/IP, HTTP, SSL)
  - Remote Method Invocation (RMI) Protocols
  - Object Management Group Protocols
    - Java IDL
    - RMI-IIOP
  - Messaging Technologies
    - Java Message Service API
    - JavaMail API
  - Data formats--define the types of data that can be exchanged between components

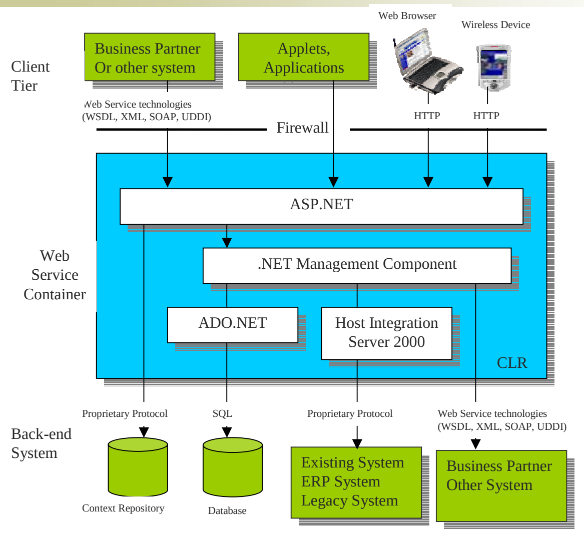
Agenda

- Background Introduction
- J2EE Structure
- .NET Structure
- J2EE vs. .NET

Microsoft .NET

- Microsoft® .NET is a set of software technologies for connecting information, people, systems, and devices. This new generation of technology is based on Web services—small building-block applications that can connect to each other as well as to other, larger applications over the Internet. ----Microsoft
- A brand name
- A set of products and technologies
  - Infrastructure
  - Tools
  - Servers
  - Services
- Microsoft core business strategy

Microsoft .NET Web Service model


Client Tier
Business Partner Or other system
Applets, Applications
Web Browser
Wireless Device
Web Service technologies (WSDL, XML, SOAP, UDDI)
Firewall
HTTP
HTTP
Web Service Container
ASP.NET
.NET Management Component
ADO.NET
Host Integration Server 2000
CLR
Back-end System
Proprietary Protocol
SQL
Proprietary Protocol
Web Service technologies (WSDL, XML, SOAP, UDDI)
Context Repository
Database
Existing System ERP System Legacy System
Business Partner Other System

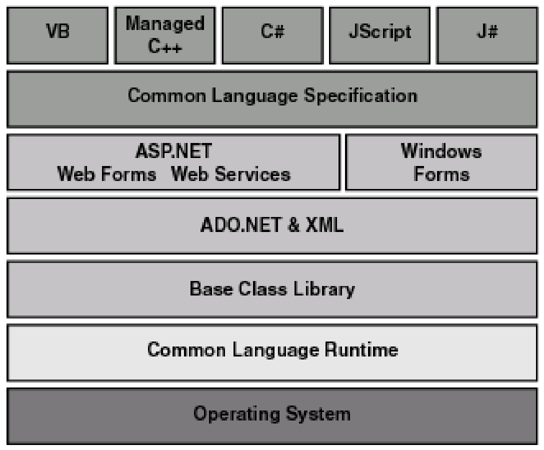
The .NET Platform Architecture

- .NET Infrastructure and Tools
  - .NET Framework
  - Visual Studio.NET
  - .NET Enterprise Servers
    - BizTalk Server 2003, SQL Server 2003, Commerce Server 2003 and more
- .NET Foundation Services
  - set of information sharing services for the Internet
    - Passport, My Services, bCentral, expedia and more
- .NET User Experience
- .NET Devices

.NET Framework

- The .NET Framework is a development and execution environment that allows different programming languages & libraries to work together seamlessly to create Windows-based applications that are easier to build, manage, deploy, and integrate with other networked systems.
  - Common Language Runtime (CLR) environment
  - Class libraries

.NET Framework



Common Language Runtime

- Responsible for run-time services
- All .NET code ultimately runs within the CLR
- Features:
  - Automatic garbage collection
  - Exception handling
  - Cross-language inheritance
  - debugging

Class Libraries

- Provide a common, consistent development interface across all languages supported by the .NET framework
  - Base classes: provide standard functionality such as input/output
  - ADO.NET classes: enable developers to interact with data accessed in the form of XML
  - XML classes: enable XML manipulation, searching, and translations
  - ASP.NET classes: support the development of Web-based application and Web services
  - Windows Forms classes: support the development of desktop-based smart client applications

Agenda

- Background Introduction
- J2EE Structure
- .NET Structure
- J2EE vs. .NET
- Conclusions

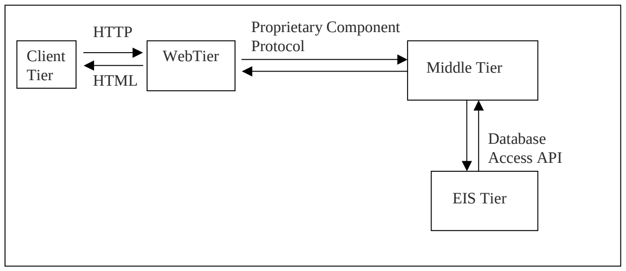
J2EE vs .NET

	J2EE	.NET
Type of Technology	Standard	Product
Middleware Vendors	50+	Microsoft and partners
Interpreter	JVM	CLR
Web GUI	JSP/Servlet	ASP.NET
Middle-Tier Components	EJB	.NET Managed Component
Database access	JDBC, SQL/J	ADO.NET
Remote Invocation	RMI-IIOP	.NET Remoting
Message Service	JMS	Message Queue
Transactions	JTA	COM+/DTC
SOAP,UDDI,WSDL	YES	YES

J2EE vs Microsoft .NET

- Both support multi-tier
- Both use the container and component idea
- Both support standards
- Both offer different tools & ways to achieve the same goal

Multi-Tier Architecture



JVM vs CLR

	JVM	CLR
Managed execution environment	X	X
Garbage Collection	X	X
Metadata and Bytecode	X	X
Platform-abstraction class library	X	X
Runtime-level security	X	X
Multi-language support	?	X
Runs across hardware platforms	X	?

J2EE vs Microsoft .NET—JVM vs CLR

- JVM designed for platform independence
  - Single language: Java
  - A separate JVM for each OS & device
- CLR designed for language independence
  - Multiple languages for development
    - C++, VB, C#, (J#)
    - APL, COBOL, Eiffel, Forth, Fortran, Haskel, SML, Mercury, Mondrian, Oberon, Pascal, Perl, Python, RPG, Scheme, SmallScript, …
  - Underlying OS: Windows

J2EE vs .NET

- Support Existing Systems
  - J2EE: J2EE Connector Architecture (JCA)
  - .NET: Host Integration Server/BizTalk Server
- Portability
  - J2EE: a standard, so it supports a variety of implementations, such as BEA, IBM, and Sun. Runs on any platform based on JRE
  - .NET: a product. Runs on Windows.
    - ECMA 334 and 335
    - The Mono Project
    - DOT GNU project

J2EE vs .NET

- Tools
  - J2EE: IBM's VisualAge for Java, Borland's JBuilder, and more.
  - .NET: Visual Studio.NET (integrated development environment)

Aganda

- Background Introduction
- J2EE Structure
- .NET Structure
- J2EE vs. .NET
- Conclusions

Conclusions: J2EE

- Sun's J2EE vision is based on a family of specifications that can be implemented by many vendors
- use of a single programming language
- offer operating system portability

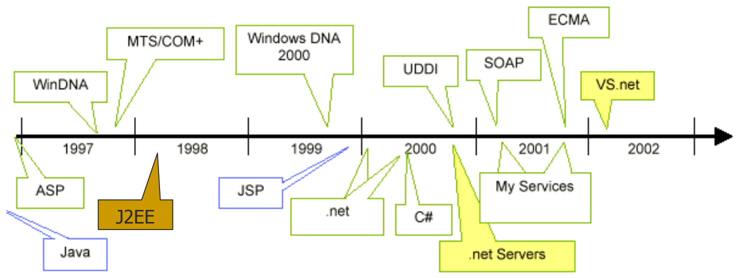
Conclusions: Micrsoft. NET

- Microsoft's .NET platform vision is a family of products, with specifications used to define points of interoperability
- limited to the Windows platform
- Support multiple programming languages
- Provide integrated developing environment

What’s your choice

- No technical superiority
- Cultural, political preferences
- Customer preference
- Vendor relations
- Skill set of your developers
- Cost

Application Platforms: Some History


WinDNA
MTS/COM+
Windows DNA 2000
UDDI
SOAP
ECMA
VS.net
1997
1998
1999
2000
2001
2002
ASP
J2EE
Java
JSP
.net
C#
My Services
.net Servers