

.Net & J2EE: War of the Frameworks

Presented to Minnesota Information Professional Society

October 15, 2002

Brian Hase – bhase@charter.net

763-458-3653



© 2002 Brian Hase

Presentation Outline

- ◆ Overview: Frameworks
- ◆ Overview: Web Services
- ◆ Overview: J2EE
- ◆ Overview: .Net
- ◆ Pros & Cons
- ◆ Market & Product Positioning
- ◆ Conclusion
- ◆ Q & A



© 2002 Brian Hase

Overview: Frameworks

- Multi-tier, enterprise applications imply integration with legacy code, legacy systems, other applications, other businesses, etc.
- Industry experience: integration is 50% of development time, e.g. "Glue is expensive."
- Frameworks target the enormous market surrounding enterprise application development particularly web services
- Benefits:
 - frameworks hide complexities such as transaction management, resource pooling, etc. from applications developers freeing them to focus on business logic or user interfaces.
 - frameworks enable proprietary platform development within an interoperable system



© 2002 Brian Hase

Overview: Frameworks

- 2 warring factions have emerged:
 - Sun has rallied the industry around its Java Centric J2EE standard; IBM, BEA, and Oracle are the biggest companies to have adopted J2EE
 - Microsoft is basing its entire product strategy around the .Net Framework
 - The current battleground salient: Capture of the Web Services market share



© 2002 Brian Hase

Overview: Frameworks

- Why the War?
 - Microsoft's dominance and (some consider) illegal or unethical practices to maintain that dominance
 - Sun's suit against Microsoft causes Microsoft to jettison Java and come up with a brand new framework
 - remarkably like J2EE with the incredible benefits a leading platform vendor can offer
 - Microsoft is big enough



Overview: Web Services

- Definition:
 - Latest abstraction of client-services provides access to an applications' functions independent of the client device or application
 - Computer processing services made available over using independent standard protocols
 - Perform discovery (where located)
 - Invocation (calling services)
 - Data passing



Overview: Web Services

- Platform and implementation independent software component that can be:
 - Described using a service description language
 - Published to a registry of services
 - Discovered through a standard mechanism at runtime or design time
 - Invoked through a declared API, usually over a network
 - Composed with other services
 - Need not necessarily exist on WWW over HTTP



© 2002 Brian Hase

Overview: Web Services

- **Uses standard protocols**
 - XML: Extensible Mark-up Language describes data vs. HTML which describes display characteristics
 - WSDL: Web Services Description Language defined in XML specifies operations in an interface, I/O to these operations, etc. (Functionally similar to IDLs used by COM or CORBA)
 - SOAP: Simple Object Access Protocol is most common defined in XML specifies a standard format for the data. Most commonly sent on top of HTTP, SOAP can be used over message queuing technology (MSMQ, MQ Series, JMS)
 - UDDI: Universal Description, Discovery and integration protocol enables applications to locate services they need



© 2002 Brian Hase

Overview: Web Services

- Not just another craze
 - Leverage existing code
 - Relatively simple to understand and implement
 - Programs, data formats, operating systems, languages need not be compatible between client and server
- Adoption rate should be fast



© 2002 Brian Hase

Overview: Web Services

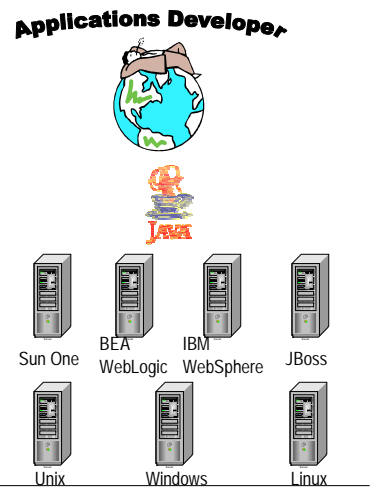
- Summary of Benefits
 - Technology transparency to data formats, OS, languages, hardware
 - Distribution of content
 - Ease of using existing code with little or no rewrites to expose it as a Web service
- One of the big issues:
 - Security & Authentication



© 2002 Brian Hase

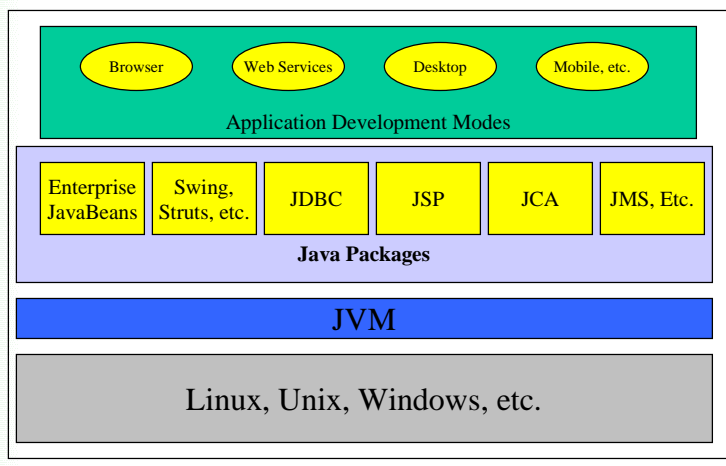
Overview: J2EE

- J2EE
 - Platform Neutral
 - Java Centric
 - Series of standards
 - Vendors: Sun, BEA, IBM, Oracle
 - Open Source platforms



© 2002 Brian Hase

Overview: J2EE



© 2002 Brian Hase

Overview: J2EE

- Defines a standard for building component-based, multi-tier enterprise applications (Sun provides a reference implementation)
- Multi-tiered distributed application model
- Standardized, modular components with a complete set of services available to these components
- Enables reuse of components
- Unified security model,
- Flexible transaction control.
- “As a single standard that can sit on top of a wide range of existing enterprise systems -- database management systems, transaction monitors, naming and directory services” – java.sun.com



© 2002 Brian Hase

Overview: J2EE

- J2EE Features
 - JDBC API - database access
 - CORBA – existing enterprise application connectivity
 - EJB – Enterprise Java Beans for transaction management
 - Java Servlets API – dynamic content delivery via server-side programming
 - Java Server Pages – dynamic content delivery via server-side scripting like ASP
 - static templates, dynamic HTML generation, and custom tags
 - JAXP – XML support via Java API for XML Parsing



© 2002 Brian Hase

Overview: J2EE

- Component Communication through standard protocols:
 - HTML
 - XML
 - HTTP
 - SSL
 - RMI
 - IIOP
- Enables vendors to come up with their own platforms and components as long as the standard is supported



© 2002 Brian Hase

Overview: J2EE

- **J2EE Standard Enterprise Services**

In addition to providing support for Enterprise JavaBeans™, Java Servlets and JavaServer Pages™ components, the Java 2 Platform, Enterprise Edition specification defines a number of standard services for use by J2EE components.
- **Java Naming and Directory Interface™ API**

Designed to standardize access to a variety of naming and directory services, the Java Naming and Directory Interface™ (JNDI) API provides a simple mechanism for J2EE components to look up other objects they require.
- **JDBC™ API**

JDBC™ API enables applications to manipulate existing data from relational databases and other data repositories. J2EE includes the latest implementation of JDBC API -- version 2.0. This new technology includes handling of SQL User-Defined Types (UDTs), rowset manipulation, connection pooling, and distributed transactions support.



© 2002 Brian Hase

Overview: J2EE

- **JavaMail™ API**

J2EE includes JavaMail™ to support applications such as e-commerce websites. The JavaMail API provides the ability to send order confirmations and other user feedback.

- **CORBA Compliance**

J2EE supports two CORBA-compliant technologies: JavaIDL and RMI-IIOP. JavaIDL enables Java applications to interact with any CORBA-compliant enterprise system. RMI-IIOP technology combines the programming ease of the Java Remote Method Invocation API (RMI) with CORBA's Internet Inter-ORB Protocol (IIOP) for easier integration of J2EE applications with legacy applications.

- **Java Transaction API**

While J2EE provides transaction support automatically, the Java Transaction API (JTA) provides a way for J2EE components and clients to manage their own transactions and for multiple components to participate in a single transaction.



© 2002 Brian Hase

Overview: J2EE

- **XML Deployment Descriptors**

J2EE defines a set of descriptors in the universal data language, XML. With its ability to support both standard and custom data types, XML makes it easier to implement customizable components and to develop custom tools.

- **Java Message Service**

The Java Message Service (JMS) API defines a standard mechanism for components to send and receive messages asynchronously, for fault-tolerant interaction. JMS is optional for J2EE release 1.0.



© 2002 Brian Hase

Overview: J2EE

Sun's Support for Vendors & Developers

Java 2 Platform, Enterprise Edition Specification

Subjected to close scrutiny and feedback from a variety of enterprise technology leaders and the industry at large, the Java 2 Platform, Enterprise Edition Specification defines a consistent yet flexible approach to implementing the platform. The J2EE™ specification enumerates the APIs to be provided with all J2EE platforms and includes full descriptions of the support levels expected for containers, clients, and components. It defines a flexible standard that can be built on either a single system or deployed across several servers, each providing a specific set of J2EE support services. This means a wide range of existing enterprise systems already in use throughout industry will be able to support the Java 2 Platform, Enterprise Edition.

J2EE Reference Implementation

Key to validating the Java 2 Platform, Enterprise Edition, the J2EE Reference Implementation provides all the specified technologies, plus a range of sample applications, tools, and documentation. This lightweight implementation of the J2EE standard is provided for two purposes. First, it provides system vendors with a standard by which to compare their own implementations. Second, it provides application developers with a way to familiarize themselves with J2EE technology as they explore commercial products for full-scale deployment of J2EE applications.



© 2002 Brian Hase

Overview: J2EE

Sun's Support for Vendors & Developers

J2EE Compatibility Test Suite

By providing a means to fully test implementations of the platform standard, the Compatibility Test Suite ensures consistent implementation across various vendor offerings. For application developers, this means full portability for enterprise applications. The suite includes tests for all classes and methods required by the J2EE specification. It also includes end-to-end tests to check that all layers of a J2EE application will interact correctly and consistently.

Sun BluePrints™ Design Guidelines for J2EE

Provided as both documentation and complete examples, the Sun BluePrints Design Guidelines for J2EE will describe and illustrate "best practices" for developing and deploying component based enterprise applications in J2EE. Topics explored will include component design and optimization, division of development labor, and allocation of technology resources.



© 2002 Brian Hase

Overview: J2EE

Sun's Support for Vendors & Developers

Sun Professional ServicesSM for J2EE developers

In addition to the J2EE deliverables, Sun offers a range of services to help enterprises adopt the J2EE standard. The J2EE Boot Camp gives your team experience implementing a distributed application with J2EE technology. The J2EE Readiness Assessment evaluates your current Java applications and provides a roadmap for migration to the J2EE standard. The J2EE Inception Service starts with your requirements to develop a plan to help initiate the J2EE application design process. And the J2EE Application Design and Deployment Service lets your team work directly with Sun architects to undertake J2EE application development.

The Enterprise Standard with a Future

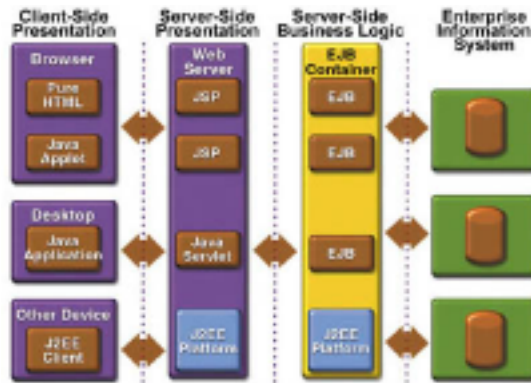
While J2EE 1.0 defines a powerful new model for implementing enterprise applications, the world of enterprise computing continues to evolve. And Sun Microsystems is already planning to evolve J2EE to keep pace, with full Java Message Service API support, additional integration of XML, the ability to interact with business objects developed in COM, and a standard API for building J2EE services on an expanding variety of existing information systems.



© 2002 Brian Hase

Overview: J2EE

Application Model



Picture Source: <http://java.sun.com/j2ee/overview2.html>



© 2002 Brian Hase

Overview: J2EE

- Basic idea:
 - J2EE will enable a business to assemble applications from a combination of standard, commercially available components and their own custom components.
- Application model:
 - divides enterprise applications into three fundamental parts: components, containers, and connectors.
 - Components:
 - are the key focus of application developers, while system vendors implement containers and connectors to conceal complexity and promote portability.



© 2002 Brian Hase

Overview: J2EE

- Containers:
 - sit between clients and components providing services to both such as transaction support – can specify much of container behavior at deployment time rather than in code
- Connectors:
 - based on JCA (Java Connector Architecture), connectors define a service API that can be plugged into various vendor platforms



© 2002 Brian Hase

Overview: J2EE

J2EE Architecture



Picture Source: <http://java.sun.com/j2ee/overview3.html>



© 2002 Brian Hase

Overview: J2EE

- EJB (Enterprise Java Bean)
 - Container specification for component model in enterprise enterprise design
 - 3 Types: session beans, entity beans, and message-driven beans.
 - A session bean represents a transient conversation with a client. When the client finishes executing, the session bean and its data are gone.



© 2002 Brian Hase

Overview: J2EE

– EJB (continued)

- Entity Bean: An entity bean represents persistent data stored in one row of a database table. If the client terminates or if the server shuts down, the underlying services ensure the entity bean data is saved.
- Message-driven Bean: A message-driven bean combines features of a session bean and a Java Message Service (JMS) message listener, allowing a business component to receive JMS messages asynchronously.

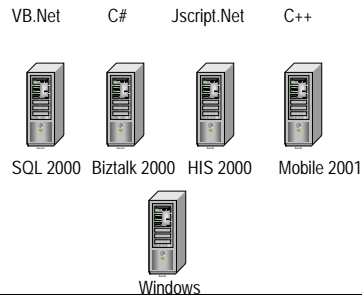


© 2002 Brian Hase

Overview: .Net

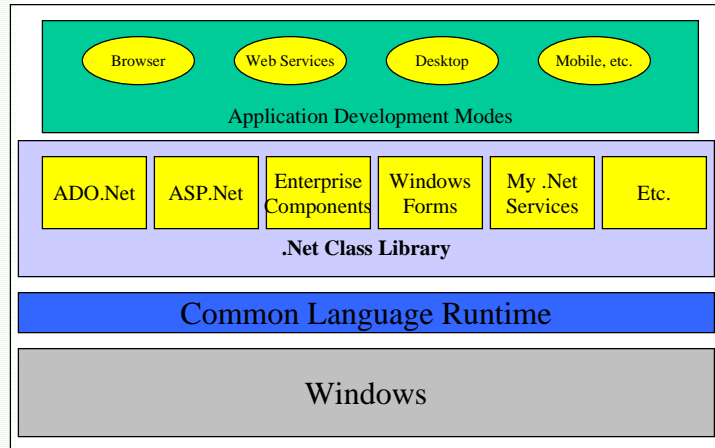
Applications Developer

- Language Neutral (VB.Net, C#, JScript.Net, C++) 3rd Party: Perl, Python, Cobol
- Evolving Visual Studio 6
- Platform Centric
- Microsoft's product strategy
- Vendors: Microsoft market share grab for web services



© 2002 Brian Hase

Overview: .Net



© 2002 Brian Hase

Overview: .Net

- .Net Framework
 - Common Language Runtime (Virtual Machine like JRE JVM)
 - .Net Framework Class Library
 - Written in C#
 - Any CLR language can create instances and call methods
 - Significant Pieces: ADO.Net, ASP.Net
 - Microsoft Intermediate Language (MSIL, Microsoft's version of Java bytecode)
 - Microsoft JIT (Just-in-time compiler)
 - Cross-Language interoperability
 - (VB.Net, C#, JScript.Net, C++) 3rd Party: Perl, Python, Cobol

© 2002 Brian Hase

Overview: .Net

- .Net Framework
 - .Net Compact Framework
 - Currently supports only Windows CE, Microsoft promising support for other platforms
 - CLR optimized for less powerful devices
 - .Net My Services (Microsoft's Web Services accessed via SOAP and defined by XML – not necessarily fundamental to .Net)
 - Passport Authentication (Kerberos)
 - .Net Alerts
 - .Net Calendar
 - .Net Contacts
 - .Net Inbox
 - .Net Documents
 - .Net Wallet
 - .Net Lists
 - .Net Profile
 - .Net Presence



© 2002 Brian Hase

Overview: .Net

- .Net Enterprise Servers

Microsoft working on these products; presumably, all will be built around the .Net framework. Originally, using the .Net label is a marketing decision – causes some confusion.

- SQL Server 2000
 - XML data exchange
 - Web-enabled data access
 - Data mining and analysis
 - Distribute database across multiple servers
 - Management automation
- Application Center 2000
 - Application sever for web applications built on MS Windows 2000 platform



© 2002 Brian Hase

Overview: .Net

- Commerce Server 2000
 - B2B & B2C
 - Profiling, Personalization
 - Integrated product catalog
- BizTalk Server 2000
 - B2B Transaction server
 - XML, SOAP, EDI
 - Security PKI, x.509, S/MIME compliant
- Internet Security and Acceleration Server (ISA)2000
 - Enterprise firewall and caching
 - VPN support



© 2002 Brian Hase

Overview: .Net

- Host Integration Server (HIS) 2000
 - Integration of host based data and applications
 - Integrate, COM+, IBM DB2, AS/400, CICS & IMS transactions
 - Security integrates with host security, W2000 Active Directory, WinNT 4 domain security
- Exchange Server 2000
 - Messaging, Collaboration
 - Integration with MSN Messenger
 - Workflow & native web application development
 - Web Storage System
- Mobile Information Server 2001
 - Platform for wireless, PDAs, mobile phones, etc.



© 2002 Brian Hase

Overview: .Net

- Software as a subscription service (HailStorm)
 - **Web based services: basics - e-mail (HotMail, instant messaging (MSN Messenger integrated with Net Meeting), automated alerts, calendars, address books, streaming media (media player), and file storage**
 - 14 initial services
 - **Goal: build a community of paying subscribers, consumers and corporate (100 million)**
 - **Integrate & link with Windows XP & Office XP**
 - **Compatible with consumer clients Linux and Mac OS**
 - **Uses .Net's MS Passport Authentication services (Relies on Kerberos) – single sign-on stored cc#s**
 - **Service/Device Independent – PC, Hand-held, cell phone**



© 2002 Brian Hase

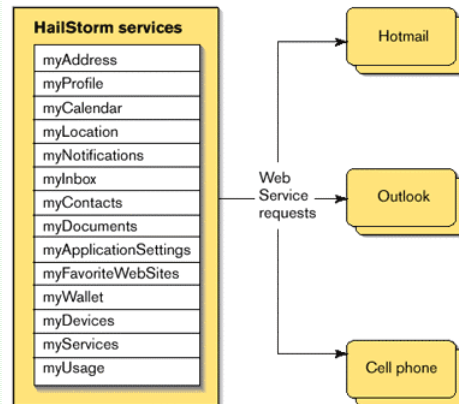
Overview: .Net

- myAddress - Electronic and geographical address for an identity
- myProfile - Name, nickname, special dates, picture
- myContacts - Electronic relationships/address book
- myLocation - Electronic and geographical location and rendezvous
- myNotifications - Notification subscription, management, and routing
- myInbox - Inbox items such as e-mail and voice mail, including existing mail systems
- myCalendar - Time and task management
- myDocuments - Raw document storage
- myApplicationSettings - Application settings
- myFavoriteWebSites - Favorite URLs and other Web identifiers
- myWallet - Receipts, payment instruments, coupons, and other transaction records
- myDevices - Device settings, capabilities
- myServices - Services provided for an identity
- myUsage - Usage report for preceding services



© 2002 Brian Hase

Overview: .Net



Accessible from
any language or
platform
integrated into
any application,
site, or device

Jon Rauschenberg, [Get Ready for HailStorm](#), .Net Magazine, March 2002

© 2002 Brian Hase



Overview: .Net

- MSIL clearly platform independent but origins of .Net framework is Windows
 - E.g. Enterprise Services must have COM+ to work leaving out Unix, Linux, Windows 98, etc.
- Will a non-Windows version of the .Net framework become a reality?
 - Microsoft promised the same for COM but it never happened (Note: CLI and C# are on the road to open source)

© 2002 Brian Hase



.Net vs. J2EE: Pros & Cons

- Web services support
 - J2EE JAXP supports e-business collaboration and web services, other 3rd party tools and support under development
 - First blush, .Net is further along. However, .Net does not support ebXML international standard. BizTalk proprietary framework has proprietary SOAP extensions making realistic deployment more iffy.
- Rapid Application Development
 - ASP.Net independence of device offers ability to render in different user interfaces without rewriting code.
 - Queued components, built-in business process management, and built-in e-commerce capabilities in .Net are superior to what is available in J2EE.
 - Visual Studio .Net is great
 - J2EE helps developer with state management services
- Legacy Integration
 - J2EE: JCA specification enabling vendors and developers to create adapters. Large number of adapters in development .Net uses Host Integration Server 2000 has limited connectivity to select systems.



© 2002 Brian Hase

.Net vs. J2EE: Pros & Cons

- Vendor solution management
 - For J2EE the additional and varied functionality offered by many vendors has a cost in portability and interoperability; need to do a lot of homework to build a system based on disparate vendors.
 - Microsoft is "the" vendor for .Net. Little problem of interoperability among their platforms. Some of the platforms are not as robust as similar J2EE solutions.
- Platform maturity
 - J2EE has the edge but still needs to mature its EJB, JCA, and Web services support
 - .Net practically everything is new. Advantage is in the ground-up design to support web services.
- Backward Compatability
 - CLR and true object-orientation of .Net poses obstacles in upgrading VB 6.0 or earlier code



© 2002 Brian Hase

.Net vs. J2EE: Pros & Cons

- Supported Languages
 - Everything in J2EE is written in Java. JVM byte-code is language neutral but for all practical purposes, J2EE is a Java platform
 - .Net supports VB.Net, C#, C++, JScript and vendor agreements may support Perl, Python, Cobol but problems:
 - Practicality of a business supporting experts in different languages, standards, training
- Portability
 - J2EE JRE available for just about any platform Win32, Unix, MainFrame
 - Industry is skeptical about Microsoft supporting other platforms
- Tools
 - Many tools available that support J2EE; functionality is good but interoperability is a problem
 - Visual Studio.Net has great productivity tools including .Net GUI components, web forms. Single vendor integration, wizards, etc. give a leg up in web services development.



© 2002 Brian Hase

.Net vs. J2EE: Pros & Cons

- Shared context
 - applies to web services as single sign-on.
 - J2EE is currently vaporware envisioning context repositories across industries to fulfill different needs
 - .Net presents Hailstorm services, a single shared context repository. It is user centric based on web services with one place to find identity information - this may be easier to implement (for Microsoft) but... industry will not apply this approach giving Microsoft dominance in managing individuals' and business identity data
- Performance/Developer
 - J2EE better supports experienced developers. New or inexperienced developers will find .Net learning curve less steep.



© 2002 Brian Hase

Market & Product Positioning

- Microsoft betting on .Net
 - Huge scope - target the enterprise & consumers
 - Includes family of familiar server applications for building, deploying and managing integrated web solutions
 - Microsoft's answer to Sun Microsystem's write-once-run-anywhere Java platform (Note)
 - Device, application, service independent
- They own the "Trifecta"
 - Operating system – Windows
 - Desktop Applications - Office
 - Portal to offer subscription services – MSN (most used Web portal worldwide)



© 2002 Brian Hase

Market & Product Positioning

- Microsoft Strategy
 - No plans for Java (Did Sun make a strategic error in not letting Microsoft "corrupt" Java?)
 - Windows/Office is a base to extend domination of personal computing world including web services
 - Government antitrust concern? Better to ask forgiveness than permission.
 - Win Developer hearts – support certain non-proprietary standards (W3C & ISO)
 - .Net shared source CLI "Rotor" (CLR, C#) to run on FreeBSD operating systems
 - Interested in other companies developing the .Net framework
- Sun Strategy
 - SunOne "services on demand" Architecture based on J2EE framework
 - Stymie Microsoft?
 - Continue to hold control over Java through JCP
 - "IBM pressures Sun to free Java" By David Berlind September 11, 2002
 - <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2879892-1,00.html>



© 2002 Brian Hase

Market & Product Positioning

- IBM
 - Open-source Java
 - Stymie Microsoft
 - Leverage and Extend Host/Mainframe investments
 - Embrace open standards
 - Beat Microsoft by aligning with Sun & Oracle but support open standards that further the agenda



© 2002 Brian Hase

Market & Product Positioning

- Open Source Community
 - Ximian Mono Project (www.ximian.com/projects/mono)
 - The Mono Project is an open development initiative sponsored by Ximian that is working to develop an open source, Unix version of the Microsoft .NET development platform. Its objective is to enable Unix developers to build and deploy cross-platform .NET Applications. The project will implement various technologies developed by Microsoft that have now been submitted to the ECMA for standardization.
 - The ".NET Initiative" is a somewhat nebulous company-wide effort by Microsoft, one part of which is a cross-platform development framework. Mono is an implementation of the development framework, but not an implementation of anything else related to the .NET Initiative, such as Passport, software-as-a-service, or corporate re-branding.



© 2002 Brian Hase

Market & Product Positioning

- DotGNU Project (www.dotgnu.org)
 - The DotGNU project was originally started in reaction to Microsoft's .NET strategy, for which it will be a complete replacement (and not just a Free Software implementation). The parts of .NET strategy about which we are most concerned are Microsoft's attempts to gain control of the internet and its users. Parts of this strategy, including the plan to unleash a "Hailstorm" have been dropped, because all the major companies which sell services to consumers have realized that this "Hailstorm" would not be good for them.



© 2002 Brian Hase

Conclusion

- Choice of framework really depends on situation
- Why is Microsoft open to .Net on other platforms?
 - managed code, VM environment
 - increased application software sales even if OS sales slip
 - US Government
- Prediction
 - .Net will continue to gain market share over J2EE on new application development
 - Unless Java is truly open-sourced by Sun, J2EE will lose momentum in the market



© 2002 Brian Hase

Questions



© 2002 Brian Hase

More Information

- Online sources
 - Understanding .Net A Tutorial and Analysis. David Chappell, Copyright 2002, Pearson Education Inc.
 - http://java.sun.com/j2ee/sdk_1.2.1/techdocs/guides/j2ee-overview/OverviewTOC.fm.html
 - <http://www.microsoft.com/net/>
 - http://internet.about.com/library/aa_netJ2EE1_080702.htm
 - <http://www.microsoft.com/net/hailstorm.asp>
 - <http://www-106.ibm.com/developerworks/>
 - <http://msdn.microsoft.com/soap/default.asp>
 - <http://www.javasoft.com>
 - <http://www.thestandard.com>
 - John Carroll "Clarifying the .Net Message" <http://zdnet.com.com/2100-1107-957998.html>



© 2002 Brian Hase