

WAS DARF ES DENN SEIN?

DIE PLATTFORMEN J2EE UND .NET HABEN VIELE GEMEINSAMKEITEN. BEI DER ENTSCHEIDUNG FÜR DIE RICHTIGE PLATTFORM SOLLTEN DIE EXISTIERENDEN SYSTEME UND DEREN INTEGRATIONSMÖGLICHKEIT GEPRÜFT UND BEWERTET WERDEN.

VON PROF. DR. MICHAEL GRÖSCHEL UND KLAUS BAUMGÄRTNER

E-Business-Anwendungen werden heutzutage auf der Basis einer Softwareplattform entwickelt. Eine solche Plattform stellt eine Vielzahl von Basisfunktionalitäten bereit, die die Erstellung von Anwendungen sehr erleichtern. Welche Plattform und welche Strategie ist nun die richtige? Diese Frage stellen sich aktuell Entscheidungsträger und das Management im IT-Umfeld. Wie so oft lautet auch hier die passende Antwort: „Es kommt ganz darauf an.“ Bei der Entscheidungsfindung gilt es, eine Reihe von Einflussfaktoren und die aktuelle Situation im Unternehmen und bei der IT zu berücksichtigen.

Bei der Integration aller Systeme werden künftig Web Services eine zentrale Rolle übernehmen.

Beim direkten Vergleich der .NET- und J2EE-Plattformen gibt es zahlreiche Gemeinsamkeiten, da beide ein Grundgerüst (Framework) für Applikationen zur Verfügung stellen, das den Entwicklern und Teams eine moderne und komponentenorientierte Software-Entwicklung ermöglicht. Beide Plattformen trennen die Entwicklungs- von der Laufzeitumgebung, unterstützen vielfältige Endgeräte (Multi-Device-Fähigkeit) sowie die aktuellen Web-Services-Standards (WSDL, SOAP, UDDI) und ermöglichen den Entwurf und die Implementierung von Mehrschichtarchitekturen.

Bei .NET handelt es sich um ein Produkt der Firma Microsoft. Auf dieser Plattform können Applikationen in meh-

rerer Programmiersprachen (z.B.: C#, C++, Visual Basic) konzipiert und implementiert werden. Aktuell gibt es für die Microsoft-.NET-Plattform jedoch nur eine auf Basis des Windows-Betriebssystems implementierte Laufzeitumgebung (Common Language Runtime, CLR). Im Gegensatz zu .NET ist J2EE kein Produkt, sondern ein von vielen Herstellern (unter anderem SUN, IBM, BEA, OpenSource) unterstützter und weiterentwickelter Standard. Der Applikationscode wird ebenfalls in einer speziellen Java-Laufzeitumgebung ausgeführt, die für mehrere Betriebssystemplattformen verfügbar ist. J2EE unterstützt dabei den Ansatz „Write once, run everywhere“.

Bevor die beiden Plattformen verglichen und bewertet werden, gilt es, noch einige Punkte zu erläutern. Die Laufzeitumgebung des .NET-Frameworks ist in hohem Maß mit der Windows-Betriebssystemfunktionalität verzahnt, wobei eine J2EE-konforme Applikation auf diversen Unix-, Windows- und Großrechnersystemen lauffähig ist. Für die Entwicklungsteams stehen bei .NET nahezu 20 Programmiersprachen zur Verfügung, unter denen insbesondere die neue Programmiersprache C# Bedeutung hat. Diese Flexibilität der Sprachwahl bei der Implementierung einer Applikation kann allerdings auch zu einem wartungsintensiven Sprachenmix führen. Im Bereich J2EE gibt es lediglich die Programmiersprache Java, für die jeweils eine spezielle Programmierbibliothek (API) vorhanden ist, um zum Beispiel Datenbank- anbindung, Integration und Datenaus-

tausch zu implementieren. Bei beiden Plattformen stehen Bibliotheken und Servicedienste zur Implementierung und Gewährleistung der Applikationssicherheit bereit. Gemein sind beiden Plattformen auch die einfache Verarbeitung von XML-Daten und -Formaten.

› EINFACHE BEDIENUNG

Ein wichtiger Faktor für die effiziente und produktive Applikationsentwicklung ist die Entwicklungsumgebung. Die hochintegrierte Microsoft-Entwicklungsumgebung „Microsoft Visual Studio .NET“ unterstützt mit vielen vorbereiteten Projekttypen und Assistenten die Codegenerierung. Durch die in Microsoft Office integrierbare und Microsoft-typische Benutzungsoberfläche ist eine hohe Produktivität und einfache Bedienung gewährleistet. Für die Entwicklung von J2EE-Applikationen stehen mehrere sehr leistungsfähige Entwicklungswerkzeuge zur Verfügung. Aufgrund der weitgehenden Standardisierung der Schnittstellen lassen sich verschiedene Tools zur Entwicklung, Versionierung und Modellierung kombiniert verwenden. Die OpenSource-Entwicklungsumgebung Eclipse hat einen sehr hohen Verbreitungsgrad.

Die Frameworks von .NET und J2EE haben mit den aktuellen Versionen einen Reifegrad erreicht, der es durchaus ermöglicht, leistungsfähige Mehrschichtapplikationen zu entwickeln. Dabei kann es beim Einsatz von J2EE dazu kommen, dass die Architektur durch eine Vielzahl von Lösungsvarianten einen gewissen

Komplexitätsgrad erreicht; dadurch ist aber gewährleistet, dass nahezu jede Anforderung abgedeckt werden kann. .NET fokussiert auf eine einfache Entwicklung durch Web Services mit dem einhergehenden Verzicht auf Flexibilität.

Das Zusammenspiel und die Interoperabilität der beiden Frameworks untereinander beziehungsweise mit anderen Systemen und EAI-Produkten ist dadurch sichergestellt, dass Standards wie XML und Web Services sowie weitere Bordmittel oder Drittproduktunterstützungen (MQ-Series) vorhanden sind.

Bei einem erforderlichen Wechsel des Betriebssystems hat die J2EE-basierte Architektur Vorteile, da mehrere Systeme für einen Wechsel zur Verfügung stehen. Die .NET-Architektur hat theoretisch ebenfalls Wechselmöglichkeiten, doch gibt es faktisch aktuell keine Alternativen zu Windows.

Beide Plattformen können aufgrund ihrer eigenen verwendeten Standards auf höhere Versionen migriert werden. Bereits existierender Code lässt sich im Java-Umfeld wesentlich leichter migrieren als bei .NET, da beispielsweise keine direkten Migrationswege für alte Visual-Basic-Programme existieren. Grundsätzlich muss jede Migration jeweils im Einzelfall analysiert und abgeschätzt werden.

Um eine ausreichend gute Performance in beiden Architekturen zu erhalten, gibt es keine Musterlösung, da die Hardwareausstattung und die Be-

triebssystemkonfiguration genauso wichtig sind, wie die Datenbank-Performance oder die verwendeten .NET- oder J2EE-Komponenten. J2EE bietet sehr viele Optimierungseinstellungen zum Preis einer höheren Komplexität. Die Skalierbarkeit einer Applikation im Umfeld von J2EE kann durch die Auswahl einer skalierbaren und hochleistungsfähigen Hardwareplattform positiv beeinflusst werden. Im Fall einer .NET-Implementierung lässt sich die Skalierung vor allem durch den Einsatz zusätzlicher, relativ günstiger Hardware erreichen. Bei J2EE existiert hier aufgrund der breiteren Auswahl an Basissystemen (Hardware und Betriebssystem) ein größeres Spektrum. Für die Integration von Drittsystemen stellen beide Frameworks eine Reihe von leistungsfähigen Erweiterungen beziehungsweise leicht integrierbare Produkte bereit. Die grundsätzliche und immer weiter entwickelte Unterstützung von Web Services und der Einsatz von XML unterstützen die Integrationsfähigkeit beider Plattformen zusätzlich. Darüber hinaus kann zum Beispiel im Bereich J2EE auf eine spezielle Integrationsarchitektur (JCA) und im Bereich Microsoft auf den Biztalk-Server zurückgegriffen werden.

Die Nutzung der .NET-Plattform impliziert eine starke Bindung an Microsoft. Diese Abhängigkeit hat positive wie negative Effekte. Die breite Auswahl an Produkten im J2EE-Bereich erscheint vorteilhafter. Eine Konzentration auf wenige ausgewählte Hersteller ist jedoch auch hier zu empfehlen, da eine einheitliche Werkzeug- und Produktumgebung Produktivitätsvorteile mit sich bringt und an einer konsequenten Weiterentwicklung und Produktstrategie partizipiert werden kann.

Ein weiterer sehr wichtiger Aspekt ist die Geschwindigkeit bei der Entwicklung und die Erweiterung von neuen Applikationen. Mittlerweile etablieren sich auch bei den J2EE-Entwicklungsumgebungen Funktionen, die den Entwicklungs-Workflow unterstützen. Diese Features sind bei .NET bereits vorhanden und etabliert.

Die klare Struktur und starke Standardisierung sowie die nicht vorhandene Möglichkeit eines Sprachenmix bei der Entwicklung erleichtern die Wartbarkeit von J2EE-Anwendungen. Beim Einsatz der .NET-Plattform ist man bei der Ermittlung der Software- und Wartungskosten an die Lizenzpolitik von Microsoft gebunden. Dagegen fallen beim Einsatz des J2EE-Frameworks keine Lizenzkosten an, da der Standard lizenzfrei ist und viele Tools und Produkte von der OpenSource-Gemeinde weiterentwickelt werden. Sobald die Anforderungen an das Gesamtsystem den Einsatz von kommerziellen Applikationsservern und Zusatzprodukten notwendig machen, fallen auch in der J2EE-Welt Lizenzkosten an. Es ist aber zu berücksichtigen, dass erfahrungsgemäß der Lizenz- und Hardwarekostenblock, gemessen an den Gesamtprojektkosten, eher gering ausfällt.

› KOEXISTENZ BEIDER PLATTFORMEN EFFIZIENT

Durch die Standardisierung aller J2EE-Programmibliotheken und die Unterstützung weiterer Internetstandards kann bei Bedarf leicht auf externe Entwicklungskapazitäten zurückgegriffen werden. Gleichzeitig wird dadurch die Portabilität und Migrationsfähigkeit der Anwendungen positiv beeinflusst.

Fazit: Bei der Entscheidung darüber, welches die richtige Plattform ist, sollten neben den genannten Kriterien im Rahmen einer unternehmensspezifischen TCO-Betrachtung auch die existierenden Systeme und deren Integrationsmöglichkeit geprüft und bewertet werden. Es ist auch möglich, dass eine Koexistenz beider Plattformen auf Dauer die effizienteste und flexibelste Lösung darstellt. Bei der Integration aller Systeme werden auch künftig Web Services eine wichtige und zentrale Rolle übernehmen. ■

Prof. Dr. Michael Gröschel, leitet den Studiengang Electronic Business der FH Heilbronn, (groeschel@taxxas.com), Klaus Baumgärtner ist Management Consultant bei der Avinci AG, (klaus.baumgaertner@avinci.de).

GLOSSAR

- .NET: Plattform und Frame von Microsoft
- J2EE: Java 2 Enterprise Edition, Spezifikation und Vorgabe für Produkthersteller
- Web Services: Über das Netzwerk und Internetprotokolle ansprechbare Softwarekomponente auf der Basis der Technologien WSDL, UDDI und SOAP
- Mehrschichtarchitektur: Konzept, die Aufgaben einer E-Business-Applikation in die Ebenen Endgerät (Client), Präsentation, Geschäftslogik und Datenhaltung/Integration zu unterteilen