

# zen Platform technical whitepaper

## Die zen Platform als strategische Unternehmensplattform

Mit der zunehmenden Verbreitung von Applikationsservern als Standardparadigma bei der Entwicklung unternehmenskritischer Anwendungen wurde ein großer Schritt hin zum effektiven Einsatz moderner serverbasierter Softwarelösungen getan. Dadurch wird die Komplexität bei der Planung und Entwicklung derartiger Software zwar weitgehend standardisiert, andererseits aber nicht ausreichend reduziert, um effektiv auf den entsprechenden mehrheitlich technischen Schnittstellen zu arbeiten.

Die schwierige Entscheidung für einen konkreten Applikationsserver definiert daher nur die technische Basis für die lösungsorientierte Anwendungsentwicklung. Sie ist nicht etwa der erste Meilenstein bei der Lösung des eigentlichen Geschäftsproblems, sondern definiert nur das methodische Gefüge. Die Entscheidung für die Architektur einer Anwendung, deren Flexibilität bezüglich Erweiterungen und Skalierbarkeit und somit deren Zukunftssicherheit muß von erfahrenen Architekten für die Java 2 Platform, Enterprise Edition (J2EE) vorgegeben werden. Der Rahmen und die benötigten Hilfsmittel zur strukturierten Lösung des Geschäftsproblems müssen der Anwendungsentwicklung so durch eine aufwendige Neuentwicklung zur Verfügung gestellt werden.

Dadurch werden nicht zuletzt die Faktoren Zeit und Kosten beeinflusst, da sogar schon vor der eigentlichen Problemlösung teure Expertenressourcen gebunden sind. Die Zeitspanne von Projektbeginn bis zur fachlichen Prüfung erster funktionstüchtiger Prototypen wird unnötig verlängert. Nicht nur in dynamischen Geschäftsbereichen hat sich das ursprüngliche Problemfeld in der Zwischenzeit oft verändert. Dies führt nicht selten zu einer Verfehlung der initialen Zielsetzung.

Die *zen Platform* der zeos informatics GmbH füllt diese umfangreiche Lücke in mehrfacher Hinsicht. Als Middleware im Applikationsserver kapselt es die komplexen technischen Schnittstellen und stellt eine vereinfachte und flexible Abstraktionsschicht zur Verfügung, auf der ein Entwickler schnell produktiv wird. Die Entwicklung der Geschäftslogik findet in der *zen Platform* auf einem sehr hohen Abstraktionsniveau statt, das den Entwickler zum einen von allen Aufgaben freistellt, die nicht primär der Problemdomäne angehören, und ihn zum anderen bei der Problemlösung selbst durch eine Vielzahl effektiver Methoden unterstützt. Die Konzentration auf das Geschäftsproblem ist so von Beginn an sogar für Entwickler ohne Expertenwissen möglich. Unabhängig vom hohen Abstraktionsniveau wird der Experte in der Anwendung des J2EE-Instrumentariums trotzdem nicht eingeschränkt.

Viele elementare Konzepte, die im Applikationsserver oft völlig vermisst oder nur fragmentarisch gelöst sind, beispielsweise dynamische Anwendungsflexibilität, Rapid Prototyping oder Lokalisierung (National Language Support), werden von der *zen Platform* nachgerüstet.

Die *zen Platform* stellt Technologien zur Verfügung, um schnell und effizient Prototypen und neue Web-Frontends, Web-Services oder auch Anwendungen mit proprietären Schnittstellen zu entwickeln oder diese zu erweitern und bringt umfangreiche Vorteile durch:

- **Visuelle Programmierung**  
Geschäftslogik für die *zen Platform* kann im *zen Developer* oder einer anderen favorisierten Umgebung entwickelt werden. Eine Anwendung wird anschließend in der graphischen Oberfläche *zen Developer* modelliert und mit der Geschäftslogik verknüpft.
- **Rapid Prototyping**  
Jede Anwendung kann schon während der Designphase visualisiert und ausgeführt werden und bietet damit die Möglichkeit zur Diskussion mit dem Fachbereich, Partnern oder Kunden, um frühzeitig Mängel aufzudecken.

- **Produktivitätssteigerung in allen Entwicklungsphasen**  
Durch die visuelle Programmierung, die Abstraktion der systemtechnischen Details und die umfangreiche Unterstützung bei der Lösung des Geschäftsproblems wird der Entwicklungsprozeß massiv beschleunigt. Auch Einsteiger werden durch die äußerst einfache Einarbeitung sehr schnell produktiv. Umfangreiche Erfahrung mit Architektur und Design ist nicht mehr notwendig. Zudem unterstützt der visuelle Debugger alle Entwicklungsphasen.
- **Flexible und dynamische Architektur**  
Eine *zen*-Anwendung kann an verschiedenen technischen und logischen klar definierten und offenen Schnittstellen beliebig erweitert werden. Die Geschäfts- und Ablauflogik kann zur Laufzeit verändert werden.
- **Skalierbarkeit**  
Jede *zen*-Anwendung skaliert ohne jegliche Modifikation von einer einfachen Servlet-Engine bis hin zum Cluster von Applikationsservern.
- **Einfache Administrierbarkeit**  
Durch die Management-Konsole können sogar verteilte Anwendungen zentral gesteuert, überwacht und rekonfiguriert werden.
- **Integration**  
Die *zen Platform* kann durch ihre Flexibilität auch als zentrale Integrationsplattform eingesetzt werden, da beliebige Anwendungstypen und Frontends unterstützt werden. Die systemimmanente Unterstützung für jegliche Art der XML-Verarbeitung vereinfacht die Anbindung von *zen*-Anwendungen an Fremd -oder Legacysysteme erheblich.

Durch den hohen Abstraktionsgrad der *zen Platform*, die damit einhergehende klare Strukturierung des Anwendungscodes, die graphische Darstellung und das visuelle Debugging wird jede Erweiterung zur einfachen Übung. Ebenso wird die Fehlerrecherche wie auch die Einarbeitung durch die Systemstrukturierung extrem vereinfacht.

## Architektur und Leistungsmerkmale der *zen Platform*

Die *zen Platform* stellt dem Entwickler auf verschiedenen Ebenen eine Vielzahl von Werkzeugen zur Verfügung, die ihn bei der effizienten Anwendungsentwicklung unterstützen. Unabhängig von der jeweiligen Anwendungsvariante, egal ob es sich um Web-Frontends, Web-Services oder Anwendungen mit beliebigen anderen Schnittstellen handelt, wird stets mit den gleichen automatisierten Methoden zur Validierung von Daten und zur Ausführung von anwendungsspezifischer Geschäftslogik gearbeitet. Bei der Implementierung steht dem Entwickler die Vielzahl von Diensten der *zen Platform* zur Verfügung, die er beliebig nutzen kann. Alle Anwendungen können über eine zentrale Management-Konsole zur Laufzeit konfiguriert, überwacht und sogar während des Betriebs erweitert oder korrigiert werden.

Jede *zen*-Anwendung wird über die graphische Oberfläche *zen Developer* in einem Repository modelliert. Die Geschäftslogik einer Anwendung wird wie üblich in Java programmiert. Im *zen Developer* wird ausgehend von der Erstellung des Workflows der Anwendung und der Modellierung der Datenstruktur festgelegt, an welchen Schnittstellen des Workflows die Geschäftslogik ausgeführt werden soll. Anschließend wird die jeweilige Geschäftslogik mit den modellierten Daten verknüpft. Die so modellierte *zen*-Anwendung wird dann von der *zen Engine* direkt ausgeführt. Die Geschäftslogik wird automatisch zur Laufzeit eingebunden und bei der Ausführung mit den jeweils passenden Daten aus dem Datenmodell versorgt.

Anders als bei heutzutage verbreiteten Vorgehensweisen ist das Anwendungsmodell mit der *zen Platform* immer konsistent mit der eigentlichen Implementierung. Die Anwendung wird durch das Repository zu jedem Zeitpunkt exakt dokumentiert.

### Visuelle Programmierung

Jede *zen*-Anwendung wird mit dem *zen Developer* entwickelt. Der *zen Developer* ist in die weit verbreitete und freie Java-Entwicklungsumgebung (IDE) Eclipse integriert. Im *zen Developer* werden neue

Anwendungen mit Hilfe von Workflows modelliert, ebenso wird hier das Datenmodell entwickelt und die anwendungsspezifische Geschäftslogik angebunden. Die Entwicklung des Java-Codes selbst kann dagegen auch in jeder beliebigen anderen IDE stattfinden.

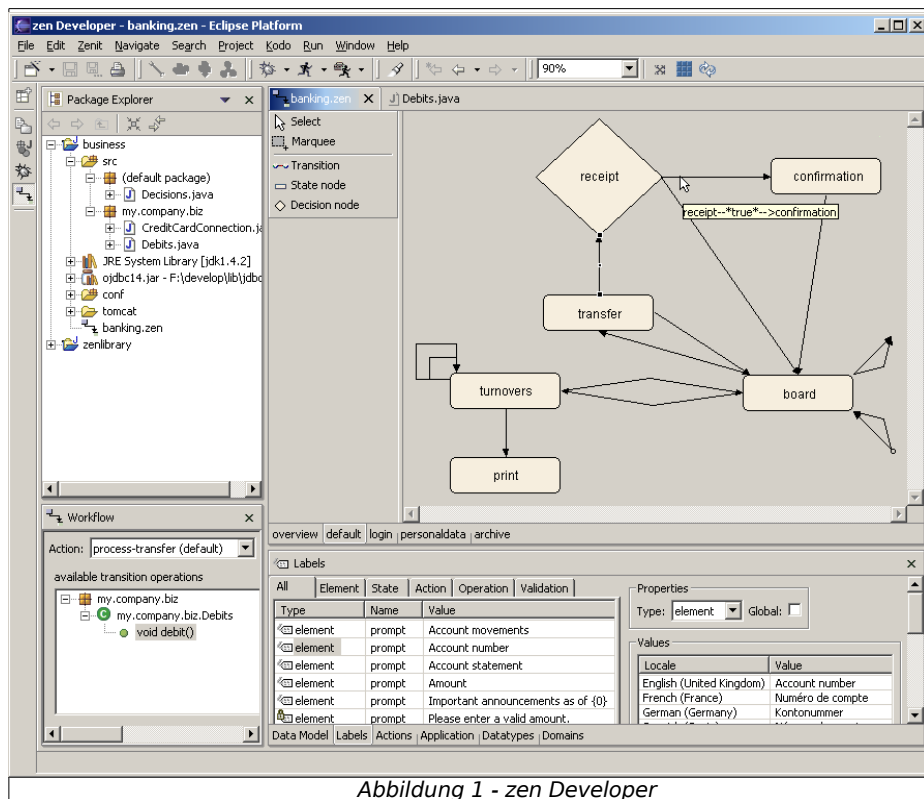


Abbildung 1 - zen Developer

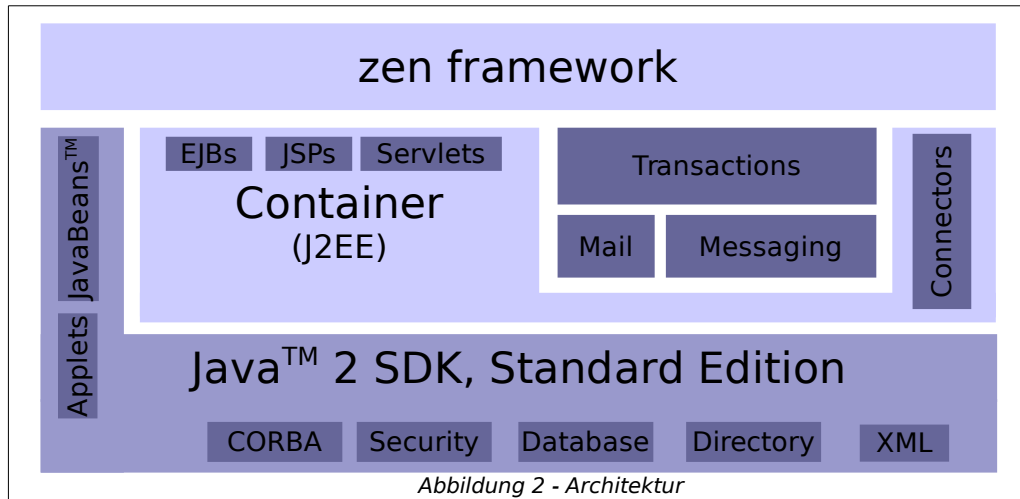
Mit einem Klick kann im zen Developer eine Prototyping-Session gestartet werden, in der ohne Erstellung spezifischer Ausgabe-Frontends die Anwendung auf Basis des jeweils aktuellen Modells in einem beliebigen Web-Browser ausgiebig getestet werden kann. Jede Anwendung kann mit dem zen Developer auch in einer graphischen Debugging-Session beobachtet und getestet werden; dabei bietet der zen Developer zu verschiedenen Zeitpunkten Einblick in den Zustand des laufenden Systems.

## Laufzeitsystem

Eine mit dem zen Developer modellierte Anwendung wird von der zen Engine ausgeführt. Die zen Engine basiert komplett auf Java und läuft daher in jeder Java Virtual Machine (VM) ab v1.3.

Zur Ausführung einer Anwendung greift die zen Engine auf die jeweilige Anwendungsdefinition im Repository zurück, die zuvor mit dem zen Developer erstellt wurde. Die Kombination zen Engine + Geschäftslogik + Repository verhält sich zur Laufzeit wie eine Standard Java-Anwendung. Eine zen Engine kann beliebig viele verschiedene zen-Anwendungen gleichzeitig ausführen.

Die zen Engine ist eine Abstraktionsschicht auf einem J2EE-Applikationsserver, die die komplexen Schnittstellen des Applikationsservers bzw. der J2EE-Spezifikation für den Anwender stark vereinfacht, optimiert und mit zusätzlichen, oft vermissten Diensten erweitert. Dadurch kann die zen Engine und damit auch jede damit entwickelte Anwendung sogar ohne Applikationsserver auf jeder beliebigen VM der Java 2 Plattform, Standard Edition (J2SE) ablaufen.

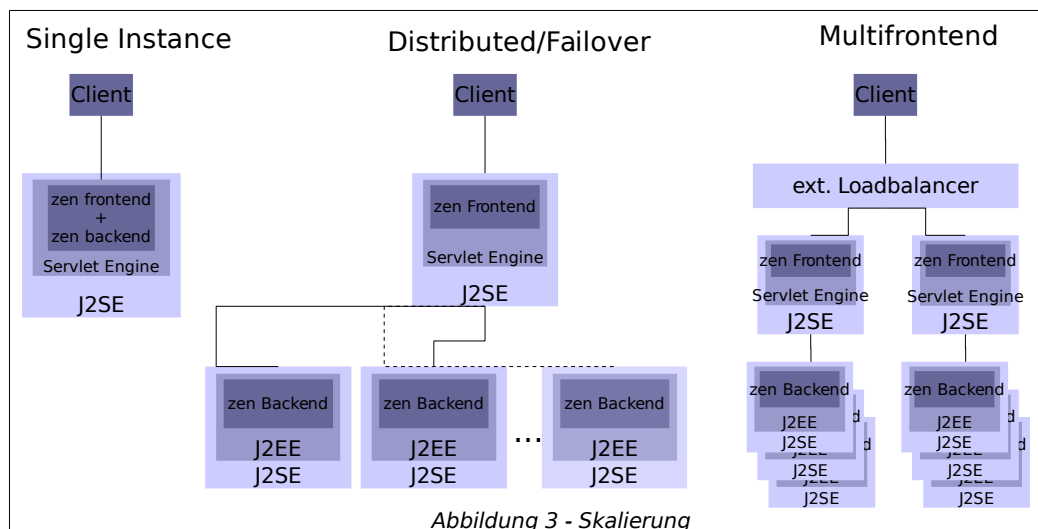


Für Anwendungen mit Web-Frontends ist eine Servlet-Engine ausreichend.

## Skalierung

Durch die Laufzeit-Flexibilität wird es möglich, mit jeder *zen*-Anwendung den jeweiligen Anforderungen entsprechend zu skalieren, ohne auch nur eine einzige Programmzeile neu zu codieren. Dieser neue Ansatz unterscheidet sich wesentlich von herkömmlicher Entwicklungsmethodik. Normalerweise ist bei wachsendem Skalierungsbedarf oft das komplette Redesign einer Anwendung erforderlich, wenn dieser beim initialen Entwurf nicht hinreichend abschätzbar war, wegen Budgetproblematik nicht berücksichtigt oder wegen einer unzureichenden Architektur nicht erfüllt werden konnte. Durch die *zen Platform* wächst die Geschäftslogik jeder Anwendung dagegen automatisch mit, da sie die abstrahierten Dienste der *zen Platform* nutzt und somit von der Umgebung, in der sie zur Laufzeit ausgeführt wird, unabhängig ist.

Auch während der Entwicklungsphase einer Anwendung kann man stark von dieser Flexibilität profitieren: Es wird keine Installation eines Applikationsservers benötigt. Das Paket aus Eclipse und *zen Developer* ist ausreichend, um auch unternehmenskritische Anwendungen zu entwickeln und zu testen.



Bei einem Clusterdeployment werden die Sessiondaten je nach Konfiguration automatisch im Cluster repliziert, so daß transparentes Failover und damit auch Hochverfügbarkeit möglich wird.

## Ablaufsteuerung

### 5-Level-Validierung

Die *zen Platform* arbeitet mit einer automatisierten fünfstufigen Datenvalidierung, die neue Eingabedaten erst in die Anwendung übernimmt, wenn sie bezüglich der Definition des Entwicklers gültig sind. In jeder Validierungsphase werden die Anfragedaten einer jeweils exakteren Prüfung unterzogen.

- (1) **Workflowvalidierung:** Das System stellt sicher, daß ein Workflow nur an definierten Einsprungpunkten betreten und anschließend konsistent durchlaufen wird.
- (2) **Strukturvalidierung:** Jede neue Anfrage an das System muß abhängig von Zustand des Workflows bezüglich der Anfragestruktur exakt der definierten Erwartung entsprechen.
- (3) **Syntaxvalidierung:** Die Eingangsdaten werden inhaltlich geprüft und auf ihre Typsicherheit hin untersucht.
- (4) **Domainvalidierung:** Daten können im Anwendungsmodell mit Domains (Wertemengen) verknüpft werden. In diesem Fall wird die Zugehörigkeit der Eingangsdaten zu den verknüpften Domains nachgeschlagen.
- (5) **Interobjekt-Validierung:** Nach einer dynamischen Abhängigkeitsanalyse werden die Anwendungsdaten durch Geschäftsregeln und -logik überprüft.

Erst nach bestandener Validierung wird eine Anfrage weiter verarbeitet. Sofern Fehler aufgetreten sind, wird die Anfrage automatisch mit modellierten Fehlerinformationen an das jeweilige Frontend zurückgeschickt. Das Frontend kann die Fehlerinformationen dann weiterverarbeiten, z.B. bei einer interaktiven Anwendung dem Benutzer visualisieren.

### Workflowgesteuerte Geschäftslogik

Der Workflow einer Anwendung kann an definierten Punkten mit der Geschäftslogik der Anwendung verknüpft werden. Wenn die *zen Engine* zur Laufzeit eine entsprechende Stelle des Workflows erreicht, wird der verknüpfte Geschäftscode mit den erforderlichen Daten aus dem Datenmodell der Anwendung ausgeführt.

### Business Rules

Zusätzlich zur workflowgesteuerten Ausführung kann Geschäftslogik in Form von Business Rules definiert werden. Eine Business Rule wird nicht mit dem Workflow, sondern nur mit dem Datenmodell verknüpft. Anders als bei workflowgesteuerter Ausführung werden Business Rules in diesem Fall immer dann aufgerufen, wenn sich die zugrundeliegenden Daten verändert haben.

## Schichtenarchitektur

### Technische Sicht

Auf der technischen Ebene ist die *zen Platform* eine Multi-Tier-Architektur. In den meisten Fällen stellt der Web-Container das Frontend des Systems dar, Validierungen und Geschäftslogik werden komplett im Backend, z.B. einem Applikationsserver, ausgeführt.

Durch die klare Trennung von Frontend und Backend wird es möglich, Anwendungen durch Austausch des Frontends an beliebig viele verschiedene und voneinander unabhängige Ausgabekanäle zu koppeln, ohne die Geschäftslogik der Anwendung dadurch zu beeinflussen. Vorhandene Ausgabeformate umfassen unter anderem XML, HTML, TEXT, WML(WAP) und PDF. Sie sind zudem beliebig erweiterbar.

## Logische Sicht

Der Workflow einer *zen*-Anwendung wird wie ein Zustandsdiagramm modelliert. Ein Workflow kann in beliebig viele Prozesse unterteilt werden. Die Datenstruktur einer Anwendung wird zunächst unabhängig vom Workflow definiert. Anschließend werden für jeden Schritt des Workflows die entsprechenden Ein- und Ausgabedaten angegeben, außerdem wird Geschäftslogik mit dem Workflow und den Daten verknüpft.

Die klar strukturierte logische Sicht ist komplett unabhängig von der technischen Sicht auf die *zen Platform*. Nicht zuletzt dadurch kann sich der Anwendungsentwickler auf die Lösung von Geschäftsproblemen konzentrieren, da umfangreiches J2EE-Expertenwissen für die Anwendungsentwicklung nicht notwendig ist.

## Dienste

Die *zen Platform* stellt dem Anwendungsentwickler eine transparente Diensteschicht zur Verfügung, die ohne Einschränkung in der Geschäftslogik verwendet werden kann. Der Zugriff und die Verwendung der einzelnen Dienste sind für den Entwickler entkoppelt vom Deployment des Gesamtsystems. Die *zen Platform* abstrahiert die Diensteschicht von der Laufzeitumgebung. Neben den Standard-Diensten, die auch in der Enterprise Java Beans-Spezifikation definiert sind, werden weitere nützliche Dienste ausgeprägt.

## Datenquellen

Neben dem Zugriff auf Datenbanken über JDBC stellt die *zen Platform* dem Entwickler auch den Datenzugriff über JDO (Java Data Objects) zur Verfügung.

## Messaging

Der Entwickler kann über den Java Message Service (JMS) direkt auf Messaging-Subsysteme zugreifen, die *zen Platform* bietet neben dem Standard zusätzlich die Möglichkeit, den Abarbeitungszustand von Nachrichten auch nach dem Versand abzufragen.

## Transaktionsunterstützung

Datenquellen können als transaktional gekennzeichnet werden, um sie automatisch über einen Transaktionsmanager (Java Transaction API) zu verwalten. Der Entwickler hat hierbei die volle Kontrolle über den kompletten Transaktionsverlauf. Die Integration mit vorhandenen Transaktionsmanagern ist möglich.

## Logging

Für jede Anwendung können eigene Logging-Kanäle definiert werden, in die der Entwickler aus der Geschäftslogik heraus schreiben kann. Jeder Logeintrag kann zudem klassifiziert werden. Die Ausgabe der einzelnen Log-Kanäle kann abhängig von der Klassifikation einfach per Konfiguration auch auf mehrere verschiedene Ziele ausgegeben werden. Neben der Dateiausgabe kann man dadurch beliebige einzelne Logeinträge beispielsweise per E-Mail zustellen lassen oder in einer Datenbank persistent machen.

## Ressourcenzugriff

Auch wenn die Enterprise Java Beans (EJB)-Spezifikation keine Unterstützung für direkten Dateizugriff bietet, kann man nicht immer darauf verzichten. Die *zen Platform* bietet daher einen Dienst, der den Zugriff auf beliebige Ressourcen J2EE-konform, transparent und sicher ermöglicht.

## **Administration**

Ein *zen-Deployment* wird zur Laufzeit über eine zentrale Administrationskonsole im Web-Browser gesteuert. Hier können alle laufenden Anwendungen überwacht werden. Ebenso ist es möglich, neue Server für ein Cluster-Deployment hinzuzufügen oder vorhandene Server abzuschalten.

Über das integrierte Monitoring kann das Laufzeitverhalten jedes Servers bis hinunter auf die Ebene einzelner Anfragen kontrolliert werden. Das Monitoring kann bei Lastproblemen automatisch Alarm-Nachrichten versenden. Auch die Konfiguration des Laufzeitsystems kann über die Konsole vorgenommen werden, beispielsweise können hier die Log-Kanäle rekonfiguriert werden.

## **Lokalisierung**

Anwendungen, die mit der *zen Plattform* erstellt werden, sind grundsätzlich lokalisierbar. Lokalisierbare Ressourcen, z.B. Texte oder URLs, werden im Repository definiert und mit einer Kombination aus Land und Sprache markiert. Zur Laufzeit selektiert die *zen Engine* abhängig von der gewählten Sprachausprägung die entsprechenden Ressourcen. Dadurch sind für neue Lokalisierungen keine Code-Anpassungen notwendig; die Übersetzung im Repository ist völlig ausreichend, um eine Anwendung in einer neuen Sprache ablaufen zu lassen.

Die Ressourcen selbst sind entweder statischer oder dynamischer Natur, wobei bei letzteren die tatsächlichen Inhalte zur Laufzeit abhängig von Sessiondaten durch die *zen Plattform* generiert wird. Die Problematik lokalisierbarer Bilder bzw. Buttons mit Textbeschriftung löst die *zen Plattform* durch ein Zusatzmodul, das lokalisierbare Buttons zur Laufzeit neu generiert; dabei kann auch die Farbe dynamisch zugesteuert werden.

## **Zusammenfassung**

Um die kurz- wie langfristige Wertschöpfung zu unterstützen, muß eine IT-Plattform in jeder Umgebung ablauffähig sein, sich nahtlos an vorhandene Systeme anbinden lassen und somit bei jedem Geschäftsproblem eine effiziente Unterstützung bieten. Die *zen Plattform* wurde auf Basis dieser elementaren Vorgaben als flexible Plattform für unternehmenskritische Anwendungen ausgelegt. Durch die visuelle Entwicklung, die klare durchdachte Systemstrukturierung, die Abstraktion von komplexen Technologien und nicht zuletzt die umfangreichen integrierten Dienste erfährt jeder Entwicklungsprozeß eine enorme Produktivitätssteigerung. Probleme werden schon in der initialen Phase durch Rapid Prototyping erkannt und kostenträchtige Fehlentwicklungen vermieden. Durch die signifikante Kosten- und Zeiteinsparung amortisiert sich die *zen Plattform* schon während der Entwicklungsphase.

Flexibilität und Produktivität in allen Entwicklungs- und Wartungsphasen erfahren damit eine neue Dimension.

Für den maximalen Erfolg in komplexen IT-Projekten bieten wir außerdem Beratung und Training durch erfahrene Experten.

Für weitere Informationen wenden Sie sich bitte an [info@zeos.de](mailto:info@zeos.de)

zeos informatics GmbH  
Gerhardstraße 1  
D-81543 München  
Tel: 089/65114244  
Fax: 089/65114238  
Web: [www.zeos.de](http://www.zeos.de)