

LAMP im Business-Umfeld

LinuxWorldExpo 2004

Frankfurt, 28.10.2004

Jan Kneschke, MySQL AB

Gaylord Aulke, 100 Days Stuttgart

Agenda

01 Einleitung / Vorstellung

02 Fragerunde

03 Was ist LAMP?

-- Pause

04 LAMP im Vergleich mit anderen Plattformen

-- Mittag

05 Cases

-- Pause

06 „Hands on“

07 Abschluss / Fazit

Einleitung / Vorstellung

Über Jan Kneschke

- Trainer und Consultant
- MySQL AB
- Seit 1995 im LAMP Umfeld tätig
- Vom Internet-Startup als Entwickler, zum SysAdmin, zum Consultant, zum Trainer bei MySQL AB
- Ist eigentlich diplomierter E-Techniker

Einleitung / Vorstellung

Über MySQL

- Das M in LAMP
- Hersteller der MySQL Datenbank
- ca. 160 Mitarbeiter weltweit
- Distributed company
- Bietet Support, Consulting und Training für alles Rund um MySQL an

Einleitung / Vorstellung

Über Gaylord Aulke

- Technische Leitung 100 DAYS
- Seit 1996 im LAMP Umfeld
- Außerdem in J2EE
- Informatiker
- Früher 4GL Entwickler / Datenbankentwickler
- New-Economy erprobt

Einleitung / Vorstellung

Über 100 DAYS

- 24 Mitarbeiter
- 2 Jahre in dieser Form am Markt
- Kunden im gehobenen Mittelstand und Großunternehmen
- Backing durch Privatinvestoren
- Focus auf Webbasierter Individualsoftwareentwicklung
- Technologieschwerpunkte: Java/J2EE, LAMP
- Focus: Agile Entwicklungsmethoden

Fragerunde

Betreiben Sie Web-Projekte?

- InHouse Entwicklungen
- Für externe Kunden
- Als freier Mitarbeiter, Projektleiter

Fragerunde

Machen Sie e-Business?

- Shop-Systeme
- Online-Werbung
- Informationsquelle für Kunden
- Kopplung von Geschäftsprozessen

Fragerunde

Sind Ihre Projekte „Mission-Critical“?

- 24/7 Verfügbarkeit?
- Maximale Ausfallzeit ?
- Rechnen Sie in Dollars/Ausfallminute
- High-Traffic?
- Mehr als 10 req/s, 100 req/s, 1000 req/s oder mehr ?

Fragerunde

Benutzen Sie LAMP?

- In Produktionsumgebungen?
- Als Entwickler?
- Kennen Sie LAMP aus der Presse?



Fragerunde

Was erwarten Sie von diesem Workshop?

Was ist LAMP?

OpenSource

„Open Source means Free Speech, not Free Beer“

- Hohe Codequalität
- Community-Support
- Weite Verbreitung
- Freie Verteilung von Wissen
- Flexibilität in Konfiguration und Verwendung

Was ist LAMP?

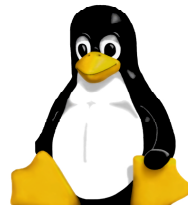
Kurze Einführung

Linux	(Betriebssystem)
Apache	(Webserver)
MySQL	(Datenbank)
PHP	(Programmiersprache)

(gelegentlich steht das P auch für auch **P**erl oder **P**ython)

Was ist LAMP?

Linux



Wir nehmen an, deswegen sind Sie alle hier.

Was ist LAMP?



DAS Open Source Projekt nach Linux selbst.

- Flexibler, stabiler Webserver
- Einfach zu konfigurieren
- Leistungsfähig und erweiterbar
- Wird auf mehr als 65% aller Server verwendet (Netcraft Oct' 04)
- Integriert sich in alle Anwendungsszenarien
- Breiter Support auf allen Ebenen

Was ist LAMP?



- Einfache und schnelle Installation
- Schnittstellen zu viele Programmiersprachen
- Freie Wahl der Plattform
- Einfaches Handling
- Wächst mit den Anforderungen mit
- Breite Installationbasis

Was ist LAMP?



- Getrieben durch PHP Group und Zend
- Hybride Programmiersprache (prozedural, objektorientiert)
- Einfach zu erlernen und einfach zu lesen
- Scriptsprache (kein Compiler-Lauf erforderlich)
- Trotzdem extrem performant und ressourcenschonend
- Zahllose Erweiterungen verfügbar
- Unterstützung für Prototyping, Entwicklung, Produktion und Pflege
- 16,946,328 Domains, 1,348,793 IP Addresses (Netcraft)

Was ist LAMP?

Warum LAMP?

- Breite Unterstützung im OpenSource und kommerziellen Umfeld
- Umfangreiches Toolset für die Fehlererkennung und Problemlösung (tcpdump, lsof, strace, ...)
- Klare Trennung zwischen den Aufgaben
- Sonderformen: WIMP, SAMP, SOJA
- Skaliert von einfachen Systemen bis zu Clustersystemen wie bei Yahoo oder Google

Plattformen-Vergleich

Im Ring stehen:
- Die Marketing-Maschine aus Redmond



ASP / .net

Plattformen-Vergleich

Im Ring stehen: - Die OpenSource-Alternative



ASP / .net



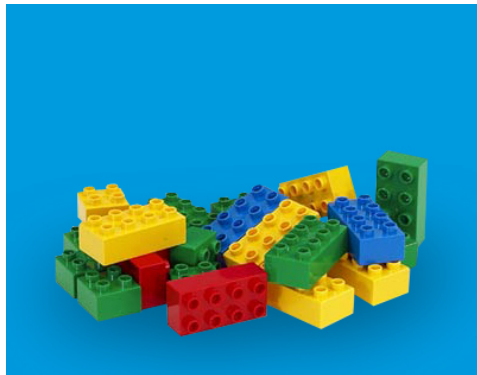
LAMP

Plattformen-Vergleich

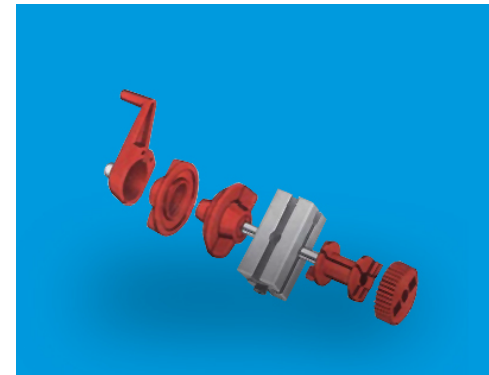
Im Ring stehen:
- Die echte Ingenieurslösung



ASP / .net



LAMP



Java / J2EE

Plattformen-Vergleich

ASP / .NET

- + Integration in die MS-System-Architektur
- + Integration in MS-Office
- + Einfache Basisinstallation, viele Tools
- Performance, Stabilität
- Sicherheit
- Single Source
- 80:20 Problematik

Plattformen-Vergleich

LAMP

- + OpenSource
- + Sehr effiziente Grundarchitektur
- + Steile Lernkurve
- + Systemunabhängig
- Bei zu hoher Komplexität unübersichtlich
- Wenig Unterstützung durch Engineering-Werkzeuge

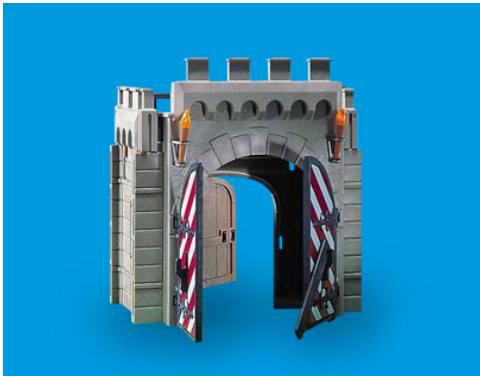
Plattformen-Vergleich

JAVA / J2EE

- + MultiSource
- + Professionelle Frameworks
- + Integrationsplattform
- + Software-Design Werkzeuge
- Komplexität
- Fehleranfälligkeit durch Komplexität

Plattformen-Vergleich

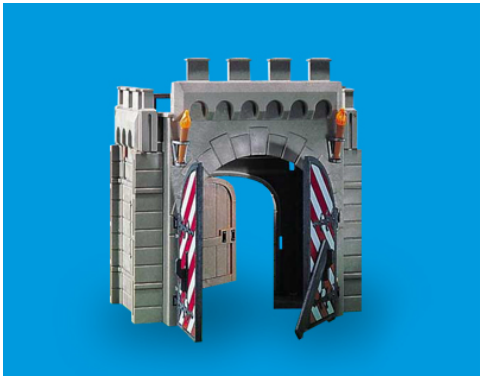
Vergleich der Lösungen:



ASP / .net

Plattformen-Vergleich

Vergleich der Lösungen:



ASP / .net



LAMP

Plattformen-Vergleich

Vergleich der Lösungen:



ASP / .net



LAMP



Java / J2EE

Plattformen-Vergleich

Unterscheidungsmerkmale

- Architektur, Technik
- Kosten (Lizenz, Entwicklung, Know-How, Betrieb, Pflege)
- Security
- Performance, Stabilität
- Risk Management

Plattformen-Vergleich

ASP / .NET: Philosophie

- Integration ins Windows-Umfeld
- Große, mächtige Komponenten
- Einfaches Arbeiten mit den Komponenten des Systems

Plattformen-Vergleich

LAMP: Philosophie

- „If anything can go wrong, it will“
- „The simpler things are, the better they work“
- Limitierte Anwendbarkeit, dadurch einfache Struktur
- Einfaches Prototyping
- Übergang von Prototyping in Richtung Endfassung schleichend möglich
- Für gängige Probleme gibt es einfache Lösungen

Plattformen-Vergleich

Java / J2EE: Philosophie

- Sauberes Software-Design nach Lehrbuch
- Legt eher Wert auf Design als auf Implementierung
- Objekt-Orientiert durch und durch

Plattformen-Vergleich

Java / J2EE: Philosophie

J2EE löst „große Probleme“

- Der Abstraktion in J2EE versucht Komplexität beherrschbar zu machen
- Frameworks und das Kapseln von Businesslogic in Beans machen komplexe Abläufe faßbar
- Für welche Projekte ist dies jedoch notwendig ?
- Oder anders gefragt: Was ist ein **großes** Problem ?

Plattformen-Vergleich

Codebeispiel: Excel Ausgabe in ASP

(Office Web Components 1.0)

```
Dim sp As OWC10.SpreadsheetClass = New SpreadsheetClass
sp.XMLURL = MapPath("test.xml")
sp.Range("B1:B4").Cells(1).Value = Convert.ToInt32(txtFld.Text)
sp.Range("B1:B4").Cells(2).Value = Convert.ToInt32(txtTerm.Text)
sp.Range("B1:B4").Cells(3).Value = Convert.ToInt32(txtRate.Text)
sp.CalculateFull()
lblMsg.Text = (sp.Range("B1:B4").Cells(4).Value * -1)
```

Plattformen-Vergleich

Codebeispiel: Excel Ausgabe in PHP

(http://pear.php.net/package/Spreadsheet_Excel_Writer)

```
<?php
require_once 'Spreadsheet/Excel/Writer.php';
$workbook = new Spreadsheet_Excel_Writer();
$worksheet =& $workbook->addWorksheet();
$worksheet->write(2, 0, "Q1");
$worksheet->write(2, 1, 0);
$workbook->send('test.xls');
$workbook->close();
?>
```

Plattformen-Vergleich

Codebeispiel: Excel Ausgabe in J2EE

(<http://www.andikhan.com/jexcelapi/>)

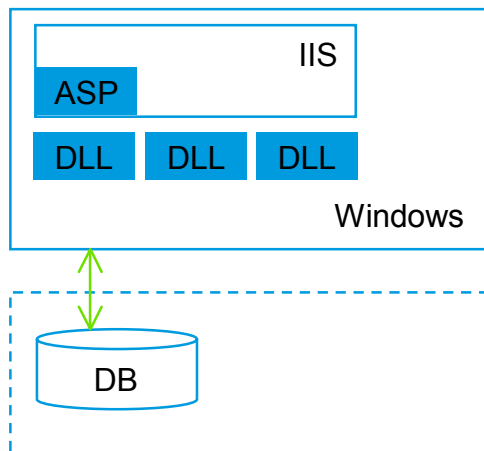
```
import java.io.File;
import java.util.Date;
import jxl.*;
import jxl.write.*;

WritableWorkbook workbook = Workbook.createWorkbook(new File("output.xls"));
WritableSheet sheet = workbook.createSheet("First Sheet", 0);
Label label = new Label(0, 2, "A label record");
sheet.addCell(label); /* to be continued */
```

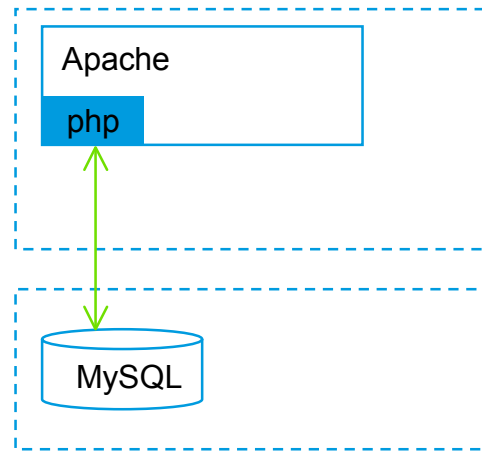
Plattformen-Vergleich

Vergleich der Architektur:

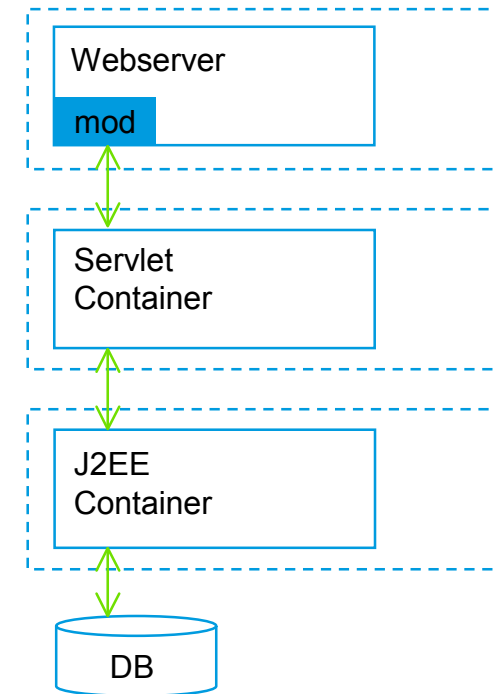
ASP / .net



php LAMP



J2EE



Plattformen-Vergleich

Know-How Aspekte

- Wie viel Know-How ist notwendig, um die Plattform zu betreiben ?
- Wie hoch sind die Investitionen für den Wissens-Aufbau und Transfer ?
- Wie viele Personen sind für gewöhnlich für den Betrieb notwendig ?

Plattformen-Vergleich

Keeping It Simple and Stupid

- Kleine, einfach kontrollierbare Komponenten
- Kombination ermöglichen komplexe Systeme
- Leitsatz: Ein Programm ist nicht dann perfekt, wenn man nichts mehr hinzufügen kann, sondern wenn man nichts mehr entfernen kann.
- Beispiel:
- # TOP10 Referer

```
$ cat access.log | awk -F '"' '{ print $4 }' | sort | uniq -c | sort -rn | head -10
```

Plattformen-Vergleich

Know-How Kosten ASP/.NET

- Komplexität wird durch graphische Interfaces verwaltbar und versteckt
- 80% wird mit sehr oberflächlichem Know-How erreicht
- Man trifft schnell auf die Probleme, die auf diesem Weg nicht lösbar sind
- Wirkliches, vertieftes Know-How ist schwer zu finden und teuer
- Zero-Administration ist wie auch das papierlose Büro oft ein Trugschluß
- Unsauber geschriebene Applikationen führen zu weitreichenden Problemen

Plattformen-Vergleich

Know-How Kosten LAMP

- 2 Personen
- Linux + Apache in der Hand des SysAdmins
- Mysql + PHP unter der Aufsicht des Entwicklers
- KISS (Keep It Simple and Stupid) ist Teil des Programmiermodells
- Erweiterung durch Integration bestehender Bibliotheken

Plattformen-Vergleich

Know-How Kosten J2EE

- 4 Personen, da 4 komplett getrennte Systeme zum Einsatz kommen
- Jedes System für sich ist zu komplex und bedarf einer eigenen Fachkraft
- Komplexe Systeme fordern mehr KnowHow bzw. haben mehr Fehlermöglichkeiten

Plattformen-Vergleich

Skalierbarkeit

- Limitiert die Leistung eines Systems die Gesamtperformance, muß die Last verteilt werden
- Lastverteilung durch mehr Leistung eines einzelnen Systems (SMP, RAID, Trunking, ...)
- Lastverteilung durch Parallelisierung der Systeme (Clustering)
- Vorteile für sauber getrennte Systeme

Plattformen-Vergleich

Skalierbarkeit: .NET

- Am einfachsten über Skalierung des Einzelsystems
- Clustering des gesamten Betriebssystems
- Abstraktion muss auf Design-Ebene der Anwendung erfolgen
- Designs nur selten auf Skalierbarkeit ausgelegt
- Designfehler lassen sich nur schwer im Nachhinein korrigieren

Plattformen-Vergleich

Skalierbarkeit: LAMP

- Freie Wahl der Architektur, erlaubt die Wahl entsprechend paralleler Systeme (z.B. Linux auf 256 CPU SGI Altix)
- Apache/PHP und MySQL können als unabhängige Cluster betrieben werden
- Applikation muss dabei das Skalieren der DB-Backends selbst abbilden
- Durch geringen Ressourcenbedarf, oft keine Skalierung erforderlich
- Die einfache Grundarchitektur ermöglicht problemloses Skalieren durch Web-LoadBalancing

Plattformen-Vergleich

Skalierbarkeit: J2EE

- Skalierbarkeit by Design
- Auftrennung der 4 Schichten erlaubt beliebige Flexibilität
- Jede einzelne Komponenten skaliert selbst
- Transparent für die Applikation

Plattformen-Vergleich

Fazit: .NET

- .NET lebt von der Integration ins Windowsumfeld
- der einfache Einstieg und die mächtigen Komponenten erlauben im gesteckten Rahmen schnelle Ergebnisse
- Insbesondere für Intranets und Extranets ist .NET daher eine interessante Option
- Im Internet-Umfeld kehren sich die Vorteile wie Direktheit und Einfachheit oft ins Gegenteil um:
- 80:20 wird sehr schnell erreicht, 95% oder 98%-Lösungen sind extrem aufwändig

Plattformen-Vergleich

Fazit: LAMP

- LAMP System erlauben schnelle Lösungen, wie es für Prototypen notwendig ist
- KISS erlaubt geringe Entwicklungs-, Wartungs- und Pflegekosten
- Fehler sind leicht zu finden
- Verständlicher und lesbarer Code erlaubt geringe Entwicklungszeit
- Das ganze Spektrum von Web-Applikationen lässt sich effizient abbilden

Plattformen-Vergleich

Fazit: J2EE

- J2EE eignet sich für Systeme, bei denen Aufgabentrennung und Redundanz/Skalierbarkeit Grundanforderungen sind
- Als Basis für eine Integrationsplattform unterschiedlicher Datenströme bestens geeignet
- Die Entwicklungszeit sollte groß genug sein, dass sich die Zeit für das Design rechnet
- Liegt die Entwicklungszeit in einer anderen Sprache unter Zeit für das Design in Java, läuft etwas falsch

Cases

LAMP Beispielprojekte

01 HypoVereinsbank

02 EurotaxGlass‘

03 Boehringer Ingelheim

Cases

HypoVereinsbank

- HypoVereinsbank AG, München
- Zweitgrößte Privatbank Deutschlands
- 60.000 Mitarbeiter
- 8,5 Millionen Kunden
- Credit Risk Control auf LAMP Basis

Cases

HypoVereinsbank

- Erster Einsatz als Dokumentenverwaltung
- „Vordok“ für Vorgangsdokumentation
- Dokumentation der Abläufe und Sicherstellung der ordnungsgemäße Abarbeitung der Kreditentscheidung
- Leistungsfähigkeit des Vorgängersystems nicht ausreichend

Cases

HypoVereinsbank

- LAMP seit 2 Jahren im Betrieb
- Hauptgründe für den Einsatz
 - Kurze Entwicklungszeiten
 - Geringe Lizenzkosten
 - Hohe Geschwindigkeit
 - Zuverlässigkeit
 - Geringe Wartungskosten

Cases

HypoVereinsbank

- „Micos“ System zur Erfassung, Steuerung und Limitierung des Sekundärrisikos aus finanzierten gewerblichen Objekten
- „Osiris“ zur Bewertung von Banken
- „GK-Rating“ zur Bewertung von Geschäftskunden im allgemeinen
- „Porthos“ zur Reduzierung von Klumpenrisiken

Cases

HypoVereinsbank

- Alle Anwendungen werden Unternehmensweit eingesetzt
- Geschäftskritische Applikationen
- 100.000 Anfragen pro Stunde
- Datenbestand ca. 125Gb, über 3 MySQL verteilt
- x86 Plattform, Red Hat Enterprise Linux, Apache, MySQL, PHP

Cases



EurotaxGlass's ist der führende Lieferant von entscheidungskritischen Informationen und Lösungen für den gesamten europäischen Automobilbereich (Beispiel: Schwacke Liste).

Seit siebzig Jahren fungiert das Unternehmen mit Hauptsitz in Freienbach, Schweiz, als unparteiisches Beobachtungsorgan, das Daten für den Automobilbereich erfasst und allen Marktteilnehmern in gedruckter oder digitaler Form on- oder offline zur Verfügung stellt.

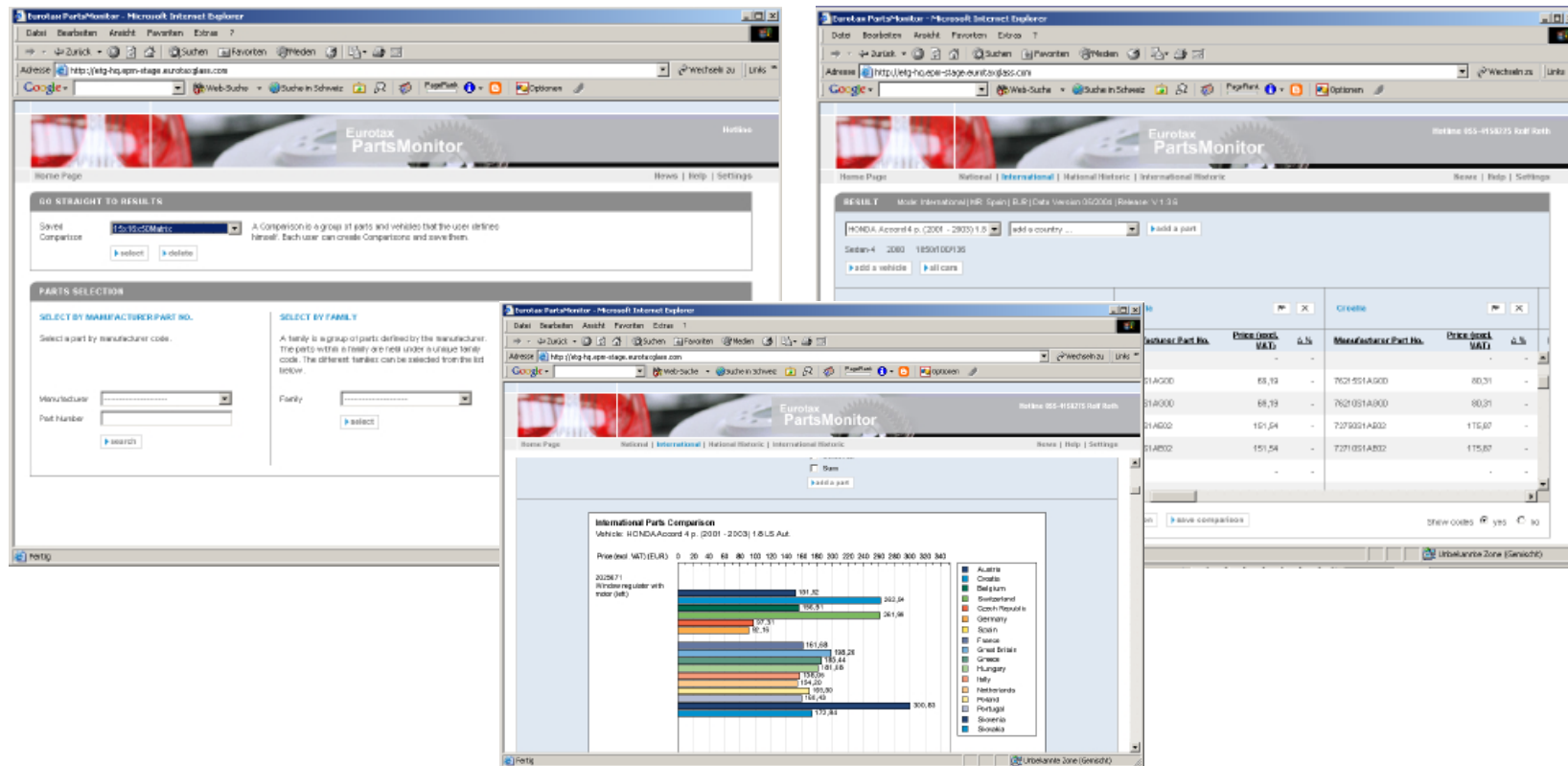
Cases



Mit dem sogenannten **"Eurotax Parts Monitor" (EPM)** sollen Automobilhersteller europaweit die Preise von insgesamt 50.000 verschiedenen Fahrzeugtypen vergleichen können und so einen entscheidenden Vorteil bei der Preisentwicklung für die eigenen Produkte erlangen.

Die Web-Applikation bietet Vergleiche für Ersatzteil-Warenkörbe, angewandt auf unterschiedliche Fahrzeugtypen, unterschiedliche Länder oder den zeitlichen Verlauf der Preisänderungen sowie technische Zeichnungen und Detailinformationen.

Cases



Frankfurt, 28.10.2004
LinuxWorldExpo LAMP Workshop

Cases



Hardware: HP Proliant 360

Datenbank: MySQL 4.0.15

Betriebssystem: SuSe Linux 8.2, Kernel Vers. 2.4.20

Webserver: Apache/1.3.27 + SSL 1.48

Programmiersprachen: PHP 4.3.1, JavaScript

Total DB size: > 80 GB and growing

Nr. of tables: 858

Nr. of rows (manufacturer): 564.350.738 part info rows (current and historical)

Nr. of vehicles/models: 166.554 different cars (national basis)

Cases



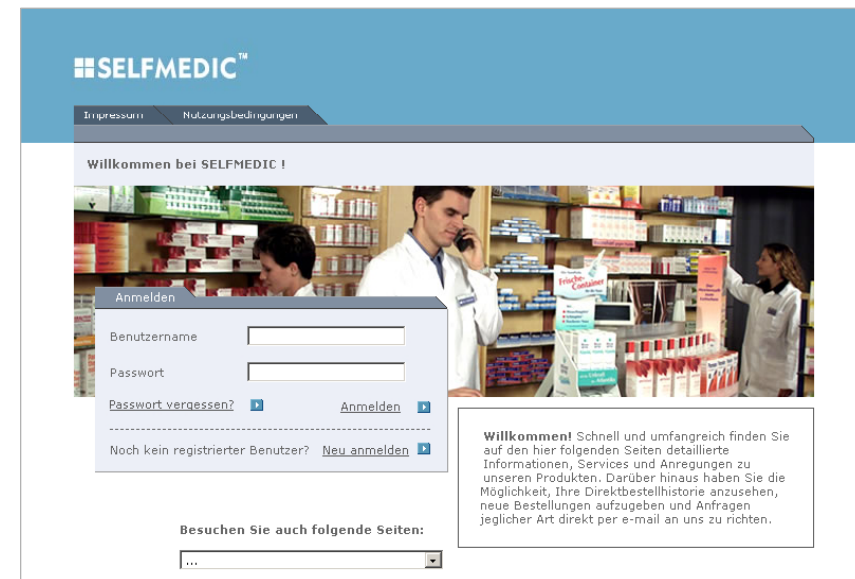
In Deutschland gehört das Familienunternehmen mit Stammsitz in Ingelheim zu den TOP 10 der Pharmabranche. Kerngeschäfte von Boehringer Ingelheim sind Humanpharma mit den Segmenten Praxis/Hospital, Biopharmazeutika, Selbstmedikation und Chemikalien sowie Tiergesundheit.

Selfmedic ist eine Marketing- und Vertriebsplattform für die Kommunikation zwischen den Apotheken und der Boehringer Ingelheim Pharma GmbH & Co. KG - von Produktbestellungen bis zur Bilddatenbank für Werbezwecke der Apotheken.

Cases



- Content-Management System
- Shop System
- Personalisiertes Portal
- Workflow
- Forum
- Bild Datenbank
- Schulungen
- eMail-Marketing
- Nutzungs Statistiken



Cases



- Betriebssystem: Windows 2000 Server
- Datenbank: ORACLE 9i
- Programmiersprachen: PHP, JavaScript
- Webserver: Apache 1.3.9 für Windows
- Versionsmanagement: Perforce SCM
- ERP-Anbindung: SAP R/3 über SAPRFC

Hands-On

Praktisches Kennenlernen

01 Installation von

- Apache
- MySQL
- PHP

02 Entwicklung einer (sehr) kleinen Anwendung

Hands-On

Apache

- Grundkonfiguration
- Wichtige Module (mod_php, mod_gzip, mod_rewrite, mod_status)
- Wichtige Parameter

Hands-On

MySQL

- Versionen, Features
- Performance Tuning
- Installation
- Integration in PHP
- Absicherung

Hands-On

PHP

- Konfiguration
- Wichtige Extensions (mysql, pcre, session)
- Wichtige Parameter
- Code-Caches

Plattformen-Vergleich

Bestandteile einer Webanwendung

- Statische Bestandteile (Layout, ...)
- Datenbankintegration, weitere externe Datenquellen
- Tabellarische Darstellung der Daten, Umformung, Personalisierung
- Internationalisierung (i18n)
- Trennung von Code, Daten und Design
- Input-Validation

Plattformen-Vergleich

Bestandteile einer Webanwendung

Inputvalidation:

- `checkdate()`, `preg_match()`, `$_GET`, `$_POST`, `$_SESSION`

Trennung Code, Design

- <http://smarty.php.net/>

Datenbankintegration

- `mysql_connect()`, `$m = new mysqli()`, `PEAR::DB`

Datenumformung

- `strftime()`, `array_split()`, `preg_replace()`

Hands-On

PHP: Forms

```
<?php
# alle POST-Variablen ausgeben
$o = "<table>\n";
foreach($_POST as $k => $v) {
    $o .= sprintf('<tr><td>%s</td><td>%s</td></tr>'. "\n",
                  $k, $v);
}
$o .= "</table>\n";
print $o;
}
?>
```

Hands-On

PHP: Output Caching

```
<?php
if (!file_exists("cache.html") ||
    file_mtime("file1.txt") > file_mtime("cache.html")) {
    /* ... generate content and send it to the client ... */
    if ($f = fopen("cache.html", "w+")) {
        fwrite($f, $content);
        fclose($f);
    }
} else {
    readfile("cache.html");
}
?>
```

Hands-On

PHP: Compression

```
<?php
# GZIP Komprimierung reduziert Traffic um ca 80%
# Content-Length erlaubt Keep-Alive, reduzierung des Overheads
ob_start("ob_gzhandler");
/* send content */
header("Content-Length: ".ob_get_length());
ob_end_flush();
?>
```

Fazit

Interessante Literatur zu LAMP

- Linux - Installation, Konfiguration, Anwendung, michael kofler, addison-wesley
- Apache: The Definitive Guide, 3rd Edition, O'Reilly, December 2002
- MySQL, Paul Dubois, Sams Developer's Library, January, 2003
- High Performance MySQL, Jeremy Zawodny, Derek J. Balling, O'Reilly 2004
- Programmieren mit PHP, Rasmus Lerdorf, O'Reilly, 2002
- Advanced PHP Programming, George Schlossnagle, Sams Developer's Library, 2004
- „The Art of Simplicity“ (The school of Niklaus Wirth), Heidelberg, dpunkt, 2000

Fazit

FIN

Wurden Ihre Erwartungen erfüllt?

jan at mysql.com

gaylord at 100days.de