



## Office-Automatisation mit PHP

International PHP Conference 2007

Thorsten Rinne



# Vorstellung

- Thorsten Rinne
- Jahrgang 1977
- Diplom-Informatiker (FH)
- Projektleiter für Banking Solutions bei der Mayflower GmbH
  - Rating- und Reportinganwendungen
  - Geschäftskritische Bankapplikationen
  - Consulting und Schulung
- Software-Entwicklung mit PHP seit 1999
- Hauptentwickler der Open Source FAQ-Managementsoftware phpMyFAQ seit 2001



# Agenda

- Einführung
- Warum Microsoft Office?
- Beispiel einer heterogenen Architektur
- Apache 2 mit PHP und mod\_ntlm
- Webservices mit PHP
- VBA
- Beispiel
- Fragen und Antworten

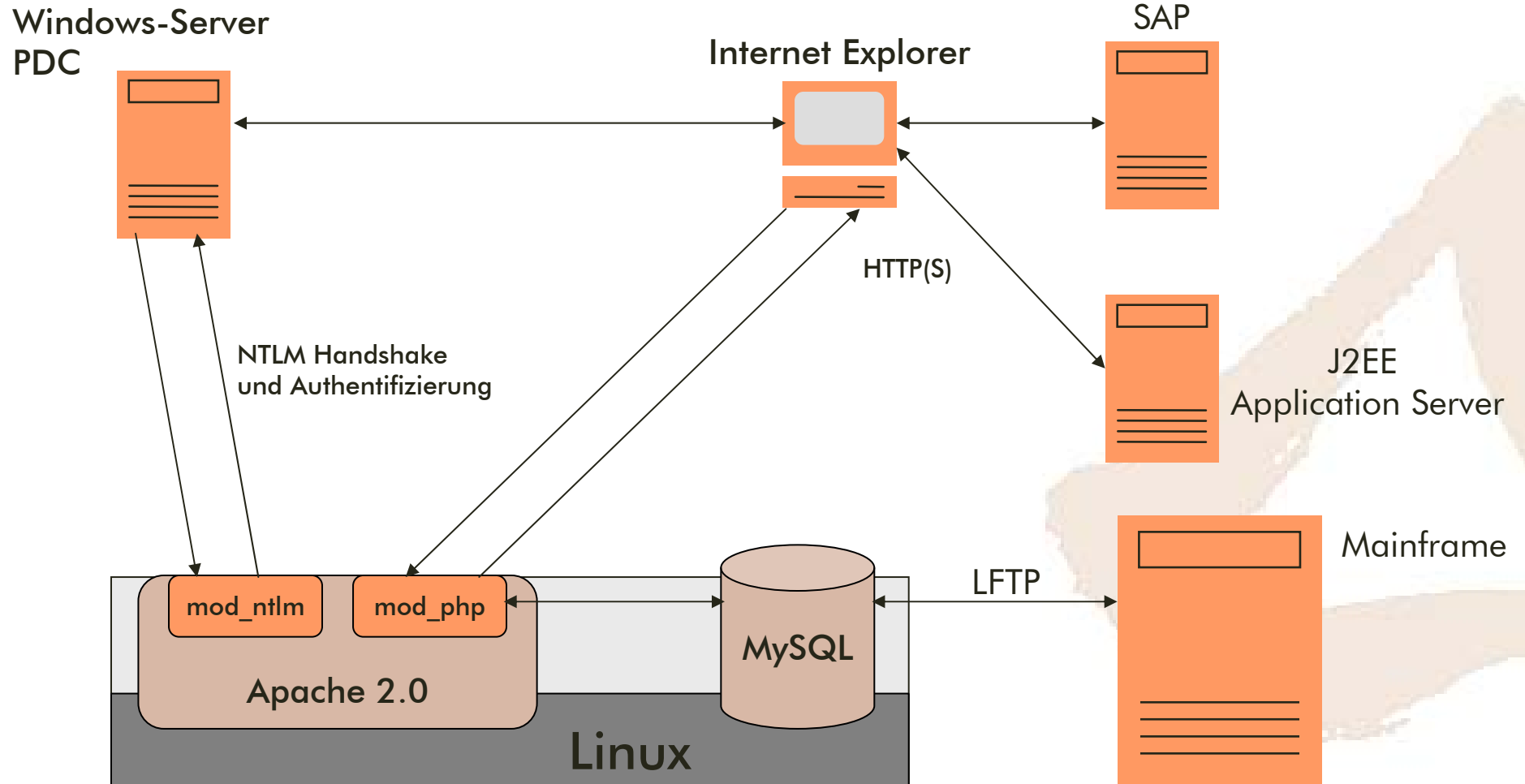


- Es gibt keine homogene Systemlandschaft in vielen Unternehmen
- Der gewöhnliche Unternehmensdesktop läuft unter Windows 2000, Windows XP oder Windows Vista mit Microsoft Office
- Die Serverlandschaft ist meist heterogen
  - Windows Server
  - Linux Server
  - Solaris / HP-UX / AIX
  - Mainframes
- Problem: Die Daten liegen auf dem Server und werden auf dem Client in einer Office-Anwendung benötigt.

# Warum Microsoft Office?

- Microsoft Office ist der De-facto-Standard mit ihren Anwendungen
  - Word (Textverarbeitung)
  - Excel (Tabellenkalkulation)
  - Powerpoint (Präsentation)
  - Access (Desktopdatenbank)
  - und weitere Anwendungen...
- Erweiterungen und Automatisierung per VBA (Visual Basic for Applications) möglich
- Anbindung per COM (Component Object Model) möglich
- Nachteile:
  - Proprietäre Dateiformate
  - Offene Dateiformate erst mit Office 2007

# Beispiel einer heterogenen Architektur



- PHP als „glue language“
  - Extensions für eigentlich jedes Problem
  - Verfügbarkeit nur auf dem Server
- Einfache und simple Implementierungen sind die Lösung
  - PHP auf dem Linuxserver
  - Authentifizierung per PDC (Windows)
  - Nutzen der vorhandenen Ressourcen auf jeder Plattform
- Nachfolgend ein „best practice“ Ansatz



# Apache 2 mit PHP und mod\_ntlm

- Apache 2.0
- PHP als Apache-Modul
  - PHP 5
  - MySQL-Extension
  - XML-Extensions
- mod\_ntlm
  - Implementierung der NTLM Authentifizierung für Apache 1.x oder 2.x
  - Nutzbar auf Unix/Linux
  - Open Source (BSD Lizenz)
  - <http://modntlm.sourceforge.net>



# Was ist NTLM?

- NTLM steht für NT LAN Manager
- Eingeführt mit Windows NT 4.0
- Fallback-Mechanismus bei Windows 2000 Server bei Kerberos
- Challenge/Response-Authentifizierungsverfahren über HTTP
- Ermöglicht Single-Sign-On auf Webservern inkl. des der Berechtigungen (Credentials)
- Support per Reverse-Engineering in Samba, Mozilla Firefox, cURL, Opera und per mod\_ntlm in Apache
- Standard unter Microsoft Windows Vista ist NTLMv2 (funktioniert nicht mit mod\_ntlm)

# Authentifizierung durch mod\_ntlm

- Authentifizierung per
  - Host: httpd.conf
  - Verzeichnis: .htaccess
- Beispiel .htaccess

```
AuthType NTLM
NTLMAuth on
NTLMAuthoritative on
NTLMServer 192.168.1.1
Require valid-user
```



- PHP 5 unterstützt
  - SOAP per ext/soap
  - XML-RPC
  - REST
- SOAP und REST sind heute die Wahl
- SOAP ist W3C Standard, aber nicht gerade schlank
- REST steht für *Representational State Transfer*



# SOAP Grundlagen

- Netzwerkprotokoll zur Datenübertragung und Durchführung von Remote Procedure Calls (RPC)
- Nutzt XML für die Repräsentation der Daten und HTTP(S) / TCP für die Übertragung
- Rechenintensiv, da XML generiert und validiert werden muss
- Protokoll ist leichtgewichtig, aber es müssen selbst für ein true/false mehrere hundert Bytes übertragen werden

# SOAP- Beispiel

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2001/12/soap-envelope">
  <s:Body>
    <m:Title xmlns:m="http://www.rinne.info/soap">
      Office-Automatisation mit PHP
    </m:Title>
  </s:Body>
</s:Envelope>
```

# REST Grundlagen

- REST steht für *Representational State Transfer*
- Reduzierung der Schnittstelle auf einheitliche Aktionen: GET, POST, PUT, DELETE
- Transportprotokoll ist HTTP(S) mit den obigen Aktionen
- Zustandsloses Client/Server-Protokoll (Cookies meist nötig)
- Universelle Syntax, jede Ressource ist per URI adressierbar
- Verwendung von HTML, XML oder Text, dadurch sehr einfach verwendbar

# REST-Beispiel

```
<?xml version="1.0"?>
<talk>
  <title>Office-Automatisation mit PHP</title>
</talk>
```

- Struktur nicht standardisiert
- Plain text möglich
- Anforderung per GET über HTTP, z.B.

GET <http://www.rinne.info/talk/phpconf/get>

# VBA: Visual Basic for Applications

- Von Microsoft entwickelte Skriptsprache
- Abgeleitet von Visual Basic
- Universell verfügbar seit Microsoft Office 2000
- Weitere kompatible Anwendungen
- Kein Support in Microsoft Office 2008 (MacOS)
- VBA
  - sehr leistungsfähig
  - interpretierte Programmiersprache
  - auf prozedurale Programmierung konzipiert
  - keine objektorientierten Konzepte wie Vererbung
  - Programmierschnittstellen gut dokumentiert

## Prüft, ob ein Blatt in Excel vorhanden ist...

```
Public Function isSheetAvailable(SheetName As String) As Boolean
    Dim Sheet As Object
    Application.ScreenUpdating = False
    For Each Sheet In ActiveWorkbook.Sheets ' check all names
        If Sheet.Name = SheetName Then
            isSheetAvailable = True
            Exit Function
        End If
    Next Sheet
    isSheetAvailable = False
End Function
```

# Welcome to the real world

- Projekt bei einem Kunden der Mayflower GmbH
- 2.000 Nutzer der Anwendung
- Webanwendung auf LAMP-Basis
  - Red Hat Enterprise Linux 4
  - Apache 2.0 mit mod\_ntlm
  - MySQL Enterprise Server 5.0
  - PHP 5.2.4 mit Suhosin
- Datenübertragung per HTTPS
- Windows Server 2003 Cluster als PDC
- Clients
  - Windows XP SP2 mit Internet Explorer 6
  - Microsoft Office XP



# Welcome to the real world!

## Ausgangslage

- 100 Millionen Datensätze in MySQL
- Zusätzlich 10 Mio personalisierte Daten
- Export per PDF vorhanden
- Kundenwunsch:

## Personalisierte Powerpoint-Folien!



# Welcome to the real world!

## Die erste Idee...

- Generierung der Powerpointfolien über
  - COM
  - PHP mit ext/com
- Nachteil:
  - PHP muss auf jedem Client verfügbar sein oder
  - PHP auf einem Windowsserver

Beides war nicht realisierbar...



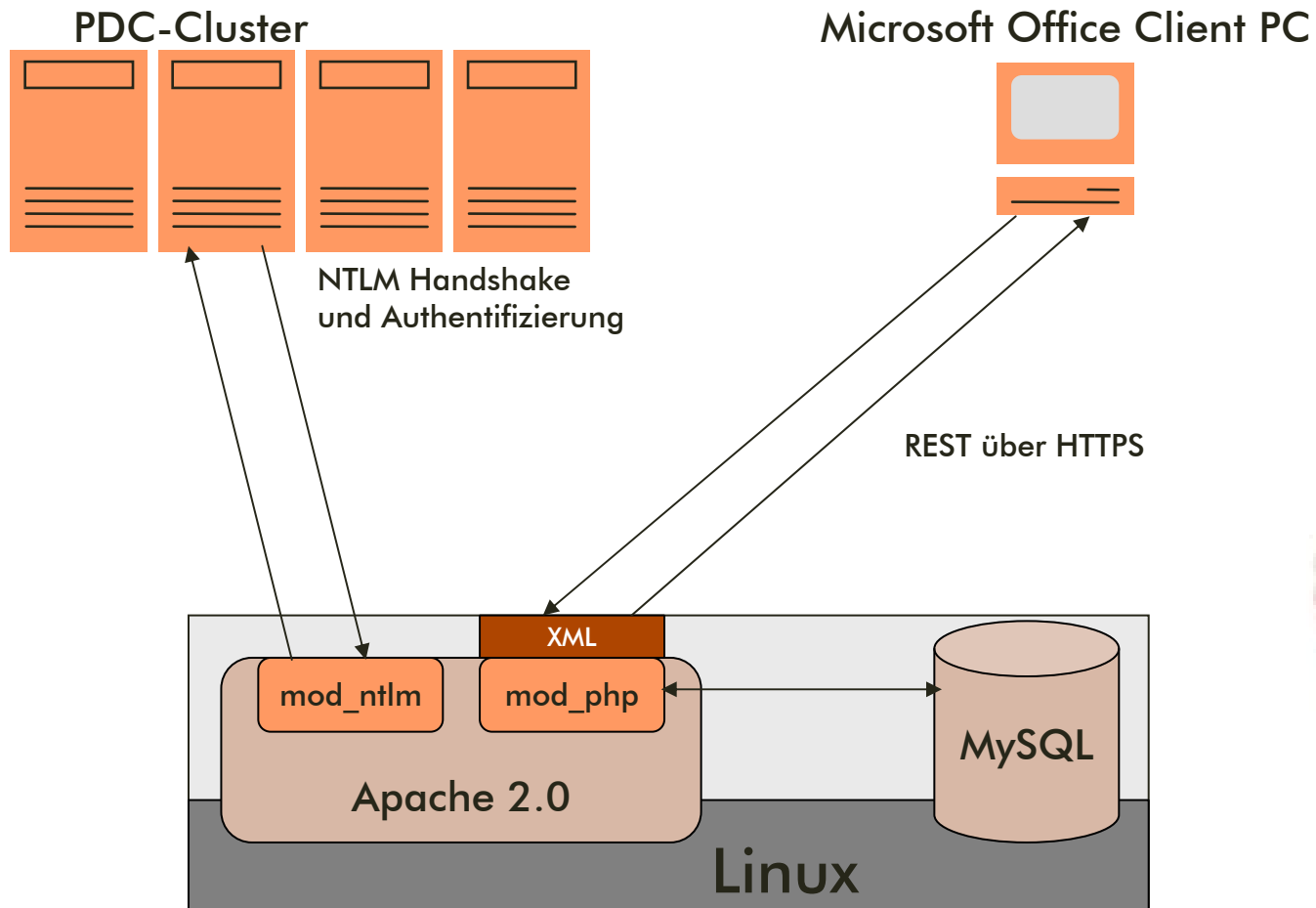
# Welcome to the real world!

## Lösung

- PHP generiert einen NTLM-geschützten Webservice per XML auf Basis von REST
- Powerpoint-Addin mit Powerpointvorlage
- Vorteile:
  - Einfache und schnelle Realisierung der serverseitigen Anwendung
  - Kein direkter Zugriff auf die MySQL-Datenbank
  - Korrekte Authentifizierung des Benutzers
  - Keine komplette Generierung eines proprietären Dateiformats
- Nachteile:
  - Download des Powerpoint-Addins
  - Bei Änderungen müssen die Anwender neue Vorlagen herunterladen

# Welcome to the real world!

## Architektur



# Welcome to the real world!

## Funktionsweise aus Anwendersicht

- Download des Powerpoint-Addins mit Vorlage
- Aufruf des Addins, evtl. Bestätigung zur Aktivierung von Makros
- Auswahl der benötigten Daten über ein Formular
- Generierung der Inhalte
- Evtl. weitere Anpassung oder Weiterverwendung der automatisch generierten Folien



# Welcome to the real world!

## Hinter den Kulissen... (I)

### Laden des XMLs über HTTP:

```
Sub load(URL As String) ⌋  
    Dim DOC As Object  
    Set DOC = loadDOMFromUrl(URL) ⌋  
    Dim parsererror As Object  
    Call ParseFromXML(DOC) ⌋  
End Sub  
  
Private Function loadDOMFromUrl(URL As String) As Object  
    Dim http As Object  
    Set http = CreateObject("MSXML2.XMLHTTP") ⌋  
    http.Open "GET", URL, False  
    On Error GoTo RequestFailed  
    http.send  
    Set loadDOMFromUrl = http.responseXML  
RequestFailed:  
    Exit Function  
End Function
```

# Welcome to the real world!

## Hinter den Kulissen... (II)

### ■ XML-Tag

```
<user_id> 4711 </user_id>
```

### ■ VBA

```
On Error GoTo domParseError  
userIdString =  
DOMDocument.getElementsByTagName("user_id").Item(0).FirstChild.nodeValue  
userId = userIdString
```

# Welcome to the real world!

## Hinter den Kulissen... (III)

### ■ Powerpoint-Platzhalter

{user\_id}

### ■ Ersetzen der Platzhalter mit VBA

```
Set TemplateRng = oTxtRng.Characters(Start:=StartRng.Start + 1,
    Length:=EndRng.Start - StartRng.Start - 1)
search = "{" & TemplateRng.Text & "}"
Replace = getTemplateValueFromVariableName(name:=TemplateRng.Text)
oTxtRng.Replace search, Replace

Private Function getTemplateValueFromVariableName(name As String) As String
    If name = "user_id" Then
        getTemplateValueFromVariableName = DataSupplyModul.GetUserId()
    End Function
```

# Welcome to the real world!

## Performance

- Etwa 0,5 bis 1 Sekunden auf dem Server
  - Authentifizierung per NTLM
  - Abfrage der Daten
  - Generierung des XMLs
  - 2x Quadcore Xeon mit 8 GB RAM
- Etwa 3 bis 4 Sekunden auf dem Client
  - Parsen des XMLs
  - Suchen und Ersetzen der Platzhalter
  - Standard PC (3 GHz P4 mit 1 GB RAM)



???





**Vielen Dank für Ihre Aufmerksamkeit**

Thorsten Rinne, Dipl.-Inf. (FH)  
Mayflower GmbH  
Mannhardtstraße 6  
D-80538 München  
+49 (89) 24 20 54 – 31  
[thorsten.rinne@mayflower.de](mailto:thorsten.rinne@mayflower.de)

