

# Zope – Ein Content Management System?

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

# Übersicht

## Teil 1:

- Was ist Zope
- Benutzung von Zope
- Erweiterungen für Zope
- Performanz von Zope
- Diskussion: Zope als Content Management System?

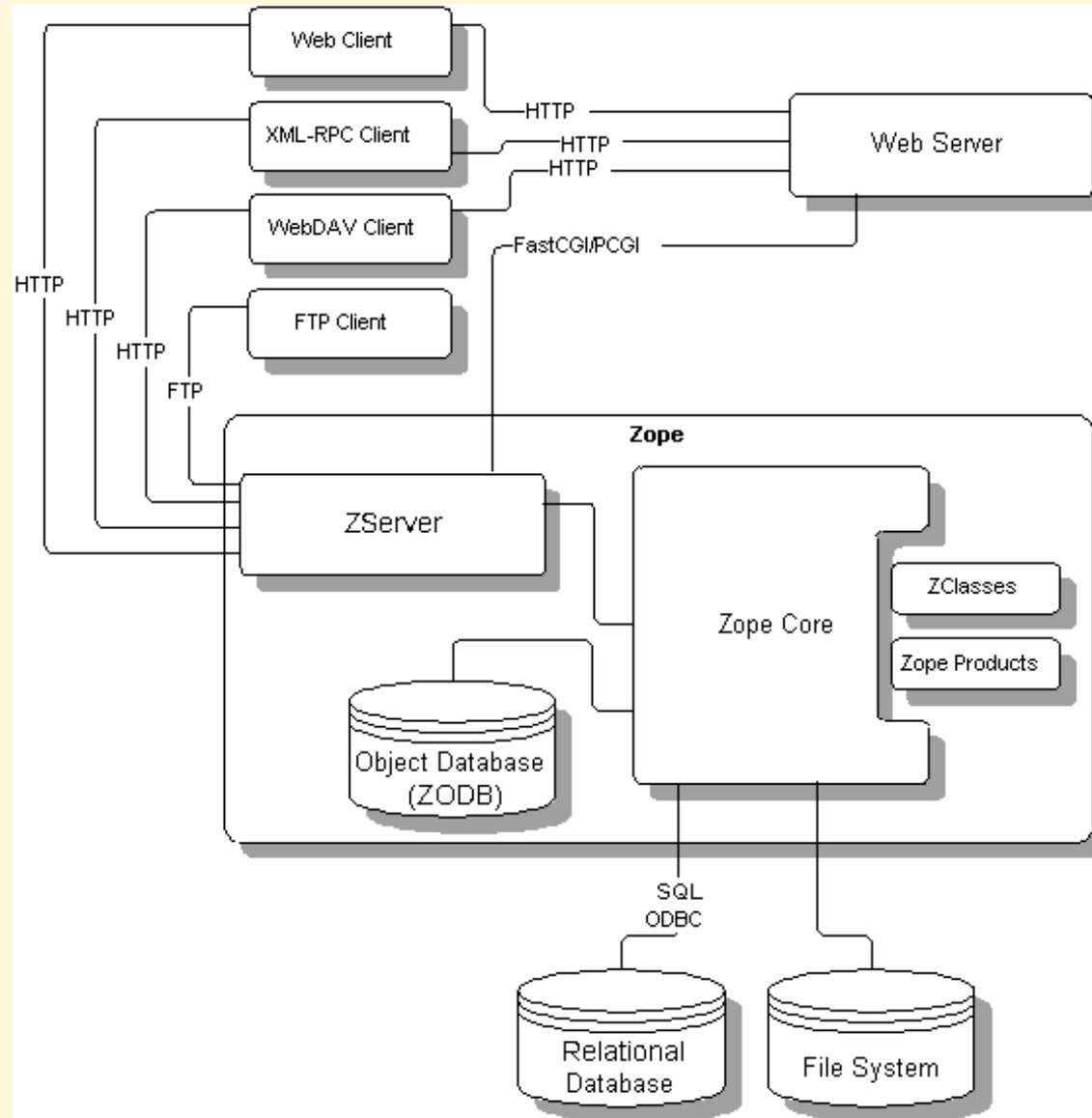
## Teil 2:

- Zope am CeBiTec
- Authentisierung via Apache

# Was ist Zope?

- *Z Object Publishing Environment*
- Zope 2: Application Server auf Basis von Python
- Web-Seiten, Templates, Verzeichnisse, Skripte, Dateien, etc. sind Objekte
- Methoden: view, edit, ...
- integrierter Web-Server (ZServer, aka **Medusa**)
- integrierte Objekt-Datenbank (ZODB), „interner“ Gebrauch
- Zope 3: kompletter Rewrite, basiert auf Komponenten

# Zope Architektur



# Zope – Woher? Wohin?

- Entwicklung seit 1996 durch **Zope Corporation**
- weitere Firmen: kommerzielle Produkte, Entwicklung, Consulting, ...
- Open Source (Zope Public License), **Zope Community**
- Anwendungen auf Basis von Zope
  - CMS: **Plone, ZMS, Silva, CPS**
  - WikiWikiWeb: **ZWiki**
  - eCommerce: **Bizar Shop**

# Voraussetzungen, Installation

- Python 2.4.2
- optional: Apache, relationale Datenbank
- einfache Installation in beliebiges Verzeichnis, z.B. `/vol/zope/`

- Instanz erzeugen:

```
/vol/zope/bin/mkzopeinstance.py
```

- Konfigurationsdatei:

```
/vol/zopeinstance/etc/zope.conf
```

- Instanz starten:

```
/vol/zopeinstance/bin/zopectl start
```

# Rundgang

The screenshot shows the 'Zope Quick Start' page in a Mozilla Firefox browser window. The address bar shows the URL 'http://doc002.cebitec.uni-bielefeld.de:8080/'. The page content includes a welcome message and a list of quick pointers for getting started with Zope.

**Zope Quick Start**

Welcome to **Zope**, a high-performance object-oriented platform for building dynamic Web applications. Here are some quick pointers to get you started:

- **Read The Fine Manual.** This document guides you through the whole process of learning Zope, from logging in for the first time to creating your own web applications.
- There is a built-in interactive **Zope Tutorial** which gets you started with some simple tasks using the Zope management interface. To use the tutorial, go to any Folder and select *Zope Tutorial* from the add list and click the *Add* button. Provide a name for the tutorial and click *Add* to begin working with the tutorial.
- **Import** and then check out the **new example Zope applications**. These examples show you simple working Zope applications that you can copy and modify.
- Go to the main **Documentation Overview** on Zope.org. Here you will find pointers to official and community contributed documentation.
- Look at the various **Mailing Lists** about Zope. The Mailing Lists are where you can get quick, accurate, friendly help from a large community of Zope users from around the world.
- Browse and search the integrated, **Online Help System** which contains documentation on the various kinds of components you'll find in Zope.
- Go directly to the **Zope Management Interface** if you'd like to start working with Zope right away. **NOTE: Some versions of Microsoft Internet Explorer, (specifically IE 5.0.1 and early versions of IE 5.5) may have problems displaying Zope management pages. If you cannot view the management pages, try upgrading your IE installation to the latest release version, or use a different browser.**
- Find out about **Zope Corporation**, the publishers of Zope.

Done

The screenshot shows the Zope Management Interface in a Mozilla Firefox browser window. The address bar shows the URL 'http://doc002.cebitec.uni-bielefeld.de:8080/manage/'. The interface is logged in as 'admin' and displays a file browser view of the 'Root Folder'.

**Zope on http://doc002.cebitec.uni-...** Logged in as **admin** Zope Quick Start

**Root Folder**

- Control\_Panel
- acl\_users
- temp\_folder
- Zope Corporation Refresh**

Folder at /

Accelerated HTTP Cache Manager Add

Type	Name	Size	Last Modified
Control_Panel	Control_Panel (Control Panel)		2006-03-15 12:59
Folder	acl_users (User Folder)		2006-03-15 12:59
Browser Id Manager	browser_id_manager (Browser Id Manager)		2006-03-15 12:59
File	error_log		2006-03-15 12:59
File	index_html	1 Kb	2006-03-15 12:59
Session Data Manager	session_data_manager (Session Data Manager)		2006-03-15 12:59
File	standard_error_message	1 Kb	2006-03-15 12:59
File	standard_html_footer	1 Kb	2006-03-15 12:59
File	standard_html_header	1 Kb	2006-03-15 12:59
File	standard_template.pt	1 Kb	2006-03-15 12:59
Folder	temp_folder		2006-03-15 12:59
Virtual Hosting	virtual_hosting		2006-03-15 12:59

Rename Cut Copy Delete Import/Export Select All

Done

# Webseiten in Zope

- zwei Template-Verfahren:
  - Document Template Markup Language (DTML)
  - Zope Page Templates (ZPT)
- DTML vergleichbar mit PHP, SSI, HTML::Template, JSP, ...
- in ZPT striktere Trennung von Layout und Logik
- ZPT-Dokument ist valides (X)HTML-Dokument
- d.h.: direkte Bearbeitung mit HTML-Editoren
- Meta-Daten als *Properties*

- `standard_html_header:`

```
<html>  
  <head><title>&dtml-title_or_id;</title></head>  
  <body>
```

- `standard_html_footer:`

```
  </body>  
</html>
```

- `boggit.html:`

```
<dtml-var standard_html_header>  
<h1>The Boggit</h1>  
<p>In a hole in the ground ...</p>  
<dtml-var standard_html_footer>
```

- `simple.pt`:

```
<html>
  <body>
    <p>This is <b tal:content="here/title">Page Title</b>!</p>
  </body>
</html>
```

- Template Attribute Language (TAL)
- Inhalt des `b`-Elements wird durch Zope ersetzt
- Platzhalter beim Preview (z.B. im HTML-Editor)

# ZPT, cont.

- komplettes Element ersetzen:

```
This is <span tal:replace="here/title">Page Title</span>!
```

- Schleifen:

```
<ul tal:repeat="article container/news/objectValues">  
  <li>  
    <a tal:attributes="href article/absolute_url"  
      tal:content="article/title">Article Title</a>  
  </li>  
</ul>
```

- Bedingungen
- Macro Expansion TAL (METAL)
- Slots

# Acquisition

- Verweise auf andere Objekte

```
<tr tal:repeat="item news">
```

- Objekt austauschbar: Python/Perl-Skript, Datenbankverbindung, ...
- hierarchische Struktur durch Verzeichnisse (*folder*)
- Objekt wird zuerst im aktuellen Verzeichnis gesucht
- danach rekursiver Aufstieg durch Eltern-Verzeichnisse
- Templates im Wurzelverzeichnis, keine expliziten Pfade
- Überschreiben von Templates durch Varianten in Unterverzeichnissen

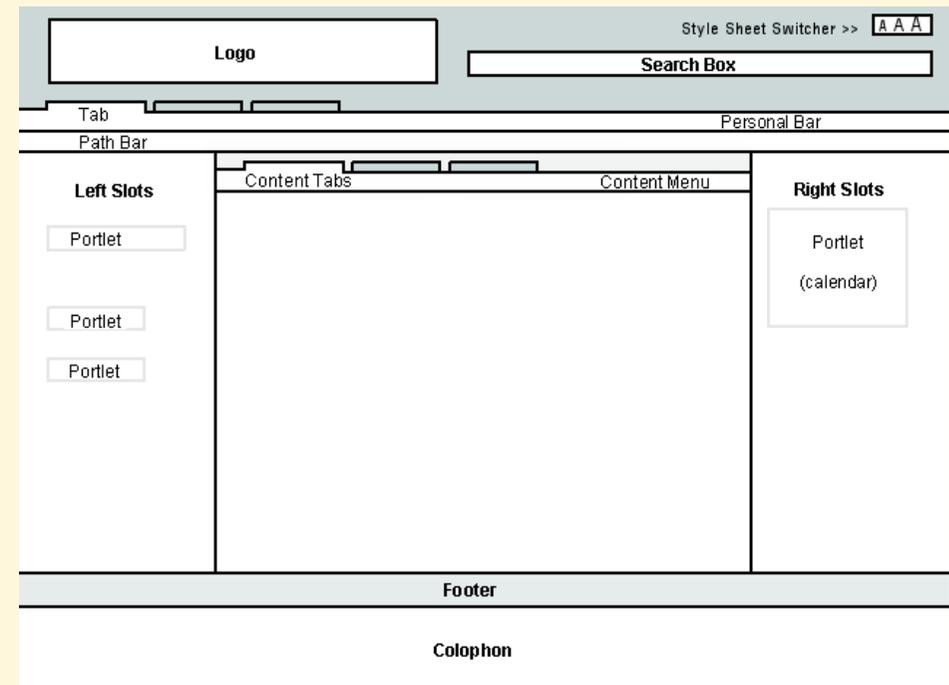
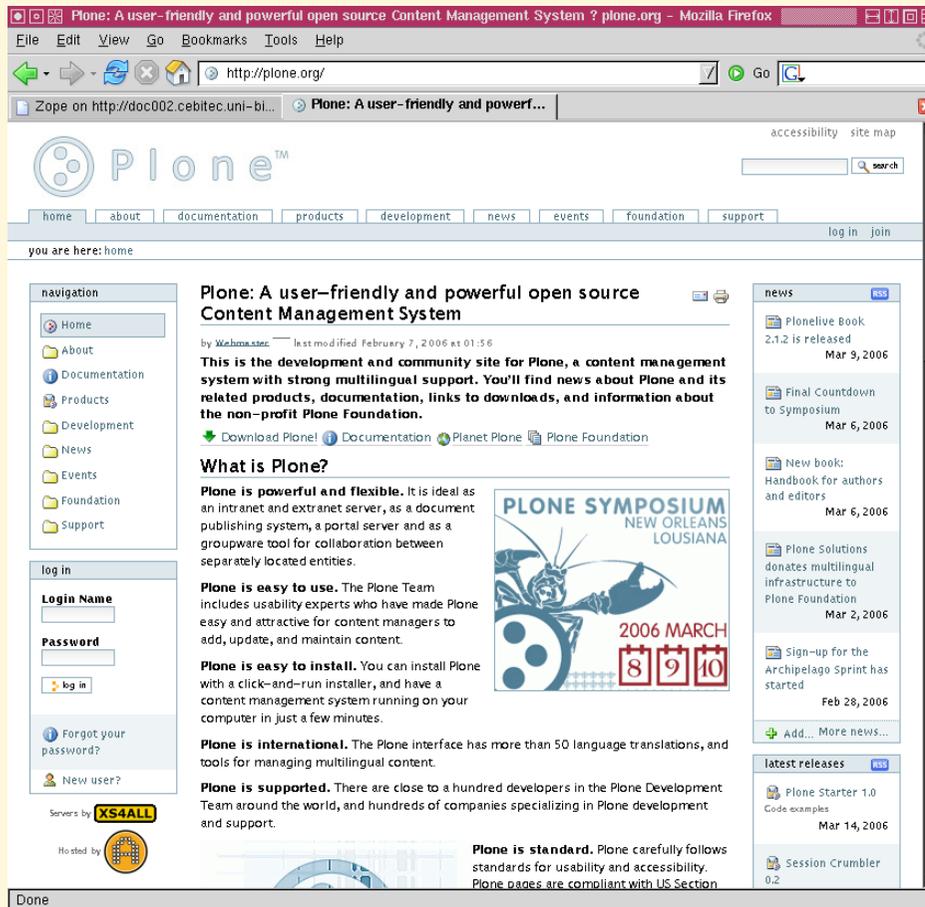
# Erweiterungen

- Erweiterungen durch *Products*
- fast alle Aspekte von Zope lassen sich verändern
- Beispiele:
  - ExternalEditor
  - HTMLDocument
  - Epoz
- Verzeichnis nach `/vol/zopeinstance/Products/` verschieben
- Zope neu starten

# Erweiterungen, cont.

- Content Management Framework (CMF)
  - weitergehende Benutzerverwaltung
  - Workflows
  - Topics, Syndication
- Plone
  - Skins, Slots
  - Portlets: Kalender, News, Events, . . .
- bescheidene Dokumentation
- Festlegung auf vorgegebene Struktur

# Plone



# Performanz

- *sehr* einfacher Vergleich mit **httperf**, aber Tendenzen erkennbar

Server	Bytes	Connections/sec	
Apache	23255	571.8	flat file
ZServer	23188	9.7	<i>dito</i>
Zope/RewriteRule	23188	167.2	<i>dito</i>
Zope/FastCGI (pipe)	23306	111.1	<i>dito</i>
Zope/FastCGI (socket)	23306	5.0	<i>dito</i>
Navigation Menu	1480	68.4	ZPT, Script (Python)
Plone	26587	3.4	
CPS	16657	3.7	
Joomla	18678	5.6	
Drupal	3838	8.2	

- zusätzliches bottleneck bei Joomla/Drupal: MySQL

# Skalierbarkeit

- Trennung von ZServer und ZODB
- Zope Enterprise Objects (ZEO)
- ein ZODB-Server, mehrere ZServer-Frontends
- Voraussetzung: identische Versionen von Zope und Produkten
- immer noch single point of failure
- aber:
  - Verteilung der I/O- und Skript-Last
  - bessere Abschottung der Business-Daten gegen Feindesland
- Verteilung der ZODB: Zope Replication Services
- kommerzielles Produkt von Zope Corp.

# To Zope or not to Zope

- Was wird *wirklich* gebraucht?
- TAL mit CGI/mod\_perl: **Petal** oder **Template::TAL**
- WYSIWYG-Editoren: **Kupu**, **Bitflux**, **FCKeditor**, ...
- WebDAV: Apache 2.0
- Integration von Objekten?
- Acquisition?
- „Vorbilder“ an der Universität Bielefeld:
  - **Zentrum für Lehrerbildung**
  - **Fakultät für Rechtswissenschaft**

# Autorisierung, Authentisierung

- Benutzerverwaltung nötig für Zuweisung von *Rollen*
- Sammlung von Benutzern in *User Folders*
- pro Verzeichnis ein User Folder möglich, sonst Acquisition
- Anbindung an externe Benutzerverwaltungen: `/etc/passwd`, LDAP, ...
- native Methode im ZServer: HTTP + Basic Authentication
- Ziel: sicherer Zugriff + Single Sign On
- aber: anonymen Zugang nicht einschränken

# Apache als Frontend

- Zope/ZServer unauffällig in Bezug auf Verwundbarkeit
- kann an geringer Verbreitung liegen
- aber: Einsatz in prominenten Websites
- Apache als Türsteher vermutlich nicht schlecht
- Bonus: Integration in legacy-Website
- Qual der Wahl
  - ProxyPass
  - RewriteRule
  - FastCGI

# URLs umschreiben

- Szenario:
  - Zope-Server: `http://zope.example.com:8080/`
  - soll unter `http://www.example.com/zope/` erscheinen
- Problem: von Zope erzeugte URLs müssen „passen“
- ProxyPass ungeeignet
- Apache-Konfiguration (eine Zeile):

```
RewriteRule ^/zope/(.*) http://zope.example.com:8080/VirtualHostBase/  
http/www.example.com:80/VirtualHostRoot/_vh_zope/$1 [L,P]
```
- auf HTTPS umbiegen: `.../https/www.example.com:443/...`
- Browser Hijacking? Cross Site Scripting?

# Authentisierung durch Apache

- Authentisierung anklemmen

```
<Location /zope>  
    AuthType Basic  
    AuthName "Zope"  
    AuthUserFile /vol/www/etc/zopepasswd  
    Require valid-user  
</Location>
```

- Digest Authentication sollte analog funktionieren
- Problem: Information (v.a. Nutzernamen) wird nicht an Zope weitergereicht

# Authentisierung durch Apache, cont.

- Weitergabe der Authentisierungsinformation durch **FastCGI**
- Kommunikation über socket oder named pipe
- FastCGI in Zope einschalten (`zope.conf`)
- Handler in Apache definieren

```
FastCgiExternalServer /vol/www/share/data/fzope -host localhost:8090
                        -pass-header Authorization
<Location /fzope>
    ...
    SetHandler fastcgi-script
</Location>
```

- Problem: FastCGI ist in Zope seit kurzem deprecated
- Benutzer muss in Zope vorher bekannt sein

# Authentisierung durch Apache, cont.

- Produkt `RemoteUserFolder`
- Option „Auto Add Users“
- Rolle „Authenticated“ oder mapping auf default-Rolle
- bekannte Benutzer (lokale Admins) können mehr Privilegien erhalten

# Anonymer Zugriff

- Server soll weiterhin anonym Zugreifbar bleiben
- Entscheidung über Authentisierung anhand von URLs schwierig

```
<LocationMatch "^/fzope/?.* /manage(_(top_frame|menu|workspace|main))?$">
```
- Idee: keine Benutzer (zumindest keine Passworte) in Zope
- anonymer Zugriff über `http://www.example.com/zope/`
  - RewriteRule oder FastCGI ohne `Auth*`-Direktiven
  - Authentisierung nicht möglich
  - Management-Funktionen nicht zugreifbar
- abgesicherter Zugriff über `http://www.example.com/fzope/`
  - FastCGI, immer Authentisierung nötig
  - Management-Funktionen zugreifbar