

# Představení nové verze aplikačního serveru Zope X3 – 3.0.0

Josef Mikloš  
josef.miklos@beneta.cz





# Program prezentace

(C) Josef Mikloš

- Představení programovacího jazyka Python
- Charakteristika Zope 2.x
- Charakteristika Zope 3.x
- Praktická ukázka



- Interpretovaný skriptovací programovací jazyk (bajtově kompilovaný)
- Vysokoúrovňový (datové typy...)
- Dynamicky – silně typovaný
- Objektově orientovaný
- Automatická správa paměti
- Úsporná syntaxe (dobrá čitelnost kódu)



- Modulární jmenné prostory
- Vyjímky
- Vlákna



- Univerzální – jak pro jednoduché skripty, tak složité aplikace (modularita)
- Vhodný pro rychlý vývoj prototypu
- Zdarma, dostupné kódy
- Rozšiřitelný pomocí C, C++
- Jython (port do Javy)



# Co je Zope?

(C) Josef Mikloš

*„Zope is an application and backend server framework that allows developers to quickly implement protocols, build applications (usually Web-based) and function as glue among other net-enabled services.“*



# Historie Zope

(C) Josef Mikloš

- 1996 Digital Creations, vývoj tří nezávislých produktů:
  - Bobo (mechanismus pro publikování objektů na webu)
- 1998 Digital Creations, vznik Zope (Z Object Publishing Environment) komerční verze -> open source, verze 1.7
- 1999 Zope Corporation, verze 2.0
- 2001 Započal vývoj verze 3



- 2004/2005

## řada 2.x

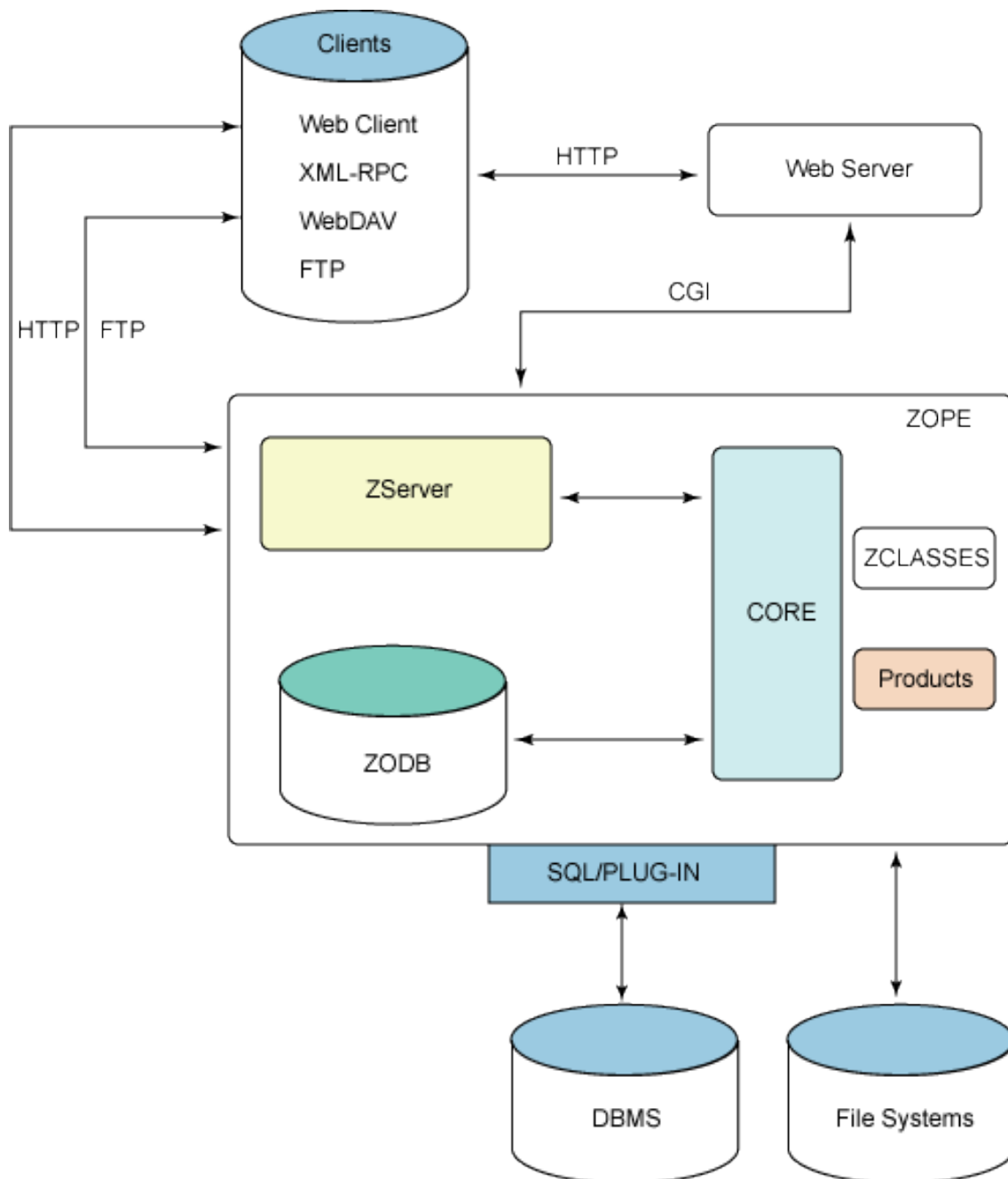
- 2.7.3 stable
- 2.7.4 rc
- 2.8.0 alfa1

## řada 3.x

- X3 3.0.0 stable

# Architektura

(C) Josef Mikloš



# ZOPE



- Perzistence objektů
- Publikování objektů na webu (URL)
- Akvizice (dynamické sdílení informací mezi objekty)
- Role a práva
- Šablony
- HTTP, FTP, WebDAV, XML-RPC
- ZEO (Zope Enterprise Objects)



- Webové rozhraní pro správu a vývoj
- Skriptování v Pythonu
- Konektory na SŘBD
- Velké množství specifických komponent - produktů



# Zope 2.x ZMI

(C) Josef Mikloš

The screenshot shows the Zope 2.x ZMI interface. The browser address bar is `http://zope-dev.beneta.cz/manage`. The user is logged in as `pepa`. The interface shows a file browser view of the root folder. The left sidebar shows a tree view of the folder structure, including `Control_Panel`, `Examples`, `Schedule`, `acl_users`, `ep_xuf`, `epoch`, `epohledavky`, `files`, `future`, `pepa`, and `temp_folder`. The main content area shows a table of files and folders with columns for `Type`, `Name`, `Size`, and `Last Modified`.

Type	Name	Size	Last Modified
<input type="checkbox"/>	Control_Panel (Control Panel)		2004-11-15 17:29
<input type="checkbox"/>	Examples (Example Applications)		2004-08-25 10:22
<input type="checkbox"/>	Schedule (Scheduled Event Catalog)		2004-12-22 08:51
<input type="checkbox"/>	SiteRoot		2004-08-20 12:20
<input type="checkbox"/>	acl_users (User Folder)		2004-08-19 17:04
<input type="checkbox"/>	browser_id_manager (Browser Id Manager)		2004-12-15 17:18
<input type="checkbox"/>	doMakej	1 Kb	2005-01-03 12:07
<input type="checkbox"/>	ep_xuf		2004-09-16 11:26
<input type="checkbox"/>	epoch (in progress, db [epohledavky])		2005-01-12 11:42
<input type="checkbox"/>	epohledavky (stable, db [epoch])		2004-12-09 10:06
<input type="checkbox"/>	error_log		2004-08-19 17:04
<input type="checkbox"/>	files		2004-09-06 08:25
<input type="checkbox"/>	future		2005-01-03 10:31
<input type="checkbox"/>	index_html	1 Kb	2004-10-05 22:25
<input type="checkbox"/>	makej	1 Kb	2004-12-22 08:31
<input type="checkbox"/>	pepa		2004-12-27 21:07
<input type="checkbox"/>	pracovniVerze (Pepova verze)		2004-12-08 11:40
<input type="checkbox"/>	session_data_manager (Session Data Manager)		2004-08-19 17:04
<input type="checkbox"/>	standard_error_message	1 Kb	2004-08-19 17:04
<input type="checkbox"/>	standard_html_footer	1 Kb	2004-08-19 17:04
<input type="checkbox"/>	standard_html_header	1 Kb	2004-09-16 09:36
<input type="checkbox"/>	standard_template.pt	1 Kb	2004-08-30 13:22
<input type="checkbox"/>	temp_folder		2005-01-10 16:38





# Proč nový Zope?

(C) Josef Mikloš

- Srozumitelný programovací model
- Menší časové nároky na zaučení
- Lepší možnost dokumentace a testování
- Podpora vícejazyčných aplikací
- Podpora workflow



- Komponentová architektura
  - Komponenta - objekt komunikující pomocí rozhraní
  - Aplikace jsou vytvářeny spojováním a skriptováním komponent
  - Aplikace mohou být konfigurovatelné záměnou komponent



# Typy komponent

(C) Josef Mikloš

- Služby
- Adaptéry
- Utility
- Továrny
- Prezentační komponenty
- Content komponenty



- Poskytují základní funkčnost pro fungování aplikačního serveru – aplikace
- Příklad: Utility Service



# Adaptéry

(C) Josef Mikloš

- Pomocí adaptérů lze přidat novou funkčnost existujícím objektům
- Adaptéry jsou vytvářeny v okamžiku potřeby
- Adaptovaný objekt není nijak dotčen
- Adapter – oriented programming



- Poskytují aplikační logiku, která se dá sdílet



- Po registraci za nás vytváří nové objekty



# Prezentační komponenty

(C) Josef Mikloš

- View
- Resources
- Layers
- Skins



# View komponenty

(C) Josef Mikloš

- Poskytují pohled na jinou komponentu z důvodu prezentace (GUI, protokol).
- Neposkytují data nebo chování mimo funkčnosti potřebné pro prezentaci.
- Komunikují s uživatelem, request + objekty



# Resources

(C) Josef Mikloš

- Obrázky
- Styly
- ...



# Layers & Skins

(C) Josef Mikloš

- View + Resources = Layers
- Layers tvoří Skin



# Content komponenty

(C) Josef Mikloš

- Zpravidla zachycují objekty problémové domény
- Většinou jsou perzistentní
- Neposkytují žádnou prezentační logiku



# ZCML - konfigurace

(C) Josef Mikloš

- ZCML (Zope Configuration Markup Language)
  - Aby aplikace mohla fungovat, musíme frameworku sdělit, jakým způsobem mezi sebou komponenty spolupracují



- XML
- Konfigurace je deklarativní, neposkytuje žádnou logiku
- Vazba mezi komponentami není zachycena ve zdrojovém kódu
- Konfigurovat může správce, ne programátor
- Konfigurační direktivy lze rozšiřovat



- V Zope X3.x je dostupný tzv. Entity-based workflows
- Definuje tok činností formou množiny stavů a přechodů mezi stavy
- Existuje nezávisle na Content komponentě
- Zachycen v XML
- Mapování WF na CC pomocí ZCML



- Schemas + Widgets
  - slouží pro generování formulářů
- Schéma obsahuje metadata o položkách (title, readonly, default, order, description atd.)
- Prezencaci schémat zajišťují Widgets
- Widget lze podle potřeby přizpůsobovat, např. specifický validátor



# Postup vývoje komponenty v Zope X3.x

(C) Josef Mikloš

- Definice rozhraní content komponenty
- Implementace rozhraní content komponenty (využití adaptérů, utilit, služeb)
- Konfigurace content komponenty
- Implementace view komponenty
- Konfigurace view komponenty
- Paralelně testování (doctesty + unittesty)
- Registrace komponenty



# Postup vývoje komponenty v Zope X3.x

(C) Josef Mikloš

- Výsledná komponenta se distribuuje jako balíček (adresář)
- **Komponenta**
  - interfaces.py
  - content.py
  - configure.zcml
  - **Browser**
    - view.pt
    - configure.zcml



- Přehlednější programovací model
- Komponentově orientovaná architektura - služby
- Vícejazyčná podpora
- WorkFlows
- Automatické generování formulářů
- Konfigurační jazyk
- Lepší možnosti ladění a testování

# Děkuji za pozornost

Josef Mikloš  
josef.miklos@beneta.cz

