



Mozilla pro vývojáře

David Majda (majda@czilla.cz)

14.4.2005

LinuxExpo 2005, Praha



Cíle této přednášky

- 1. Vnitřní architektura Mozilly**
- 2. Vývoj aplikací v Mozille**
 - XUL
 - tvorba rozšíření
- 3. Je Mozilla to pravé pro vaši aplikaci?**
 - malý test na závěr

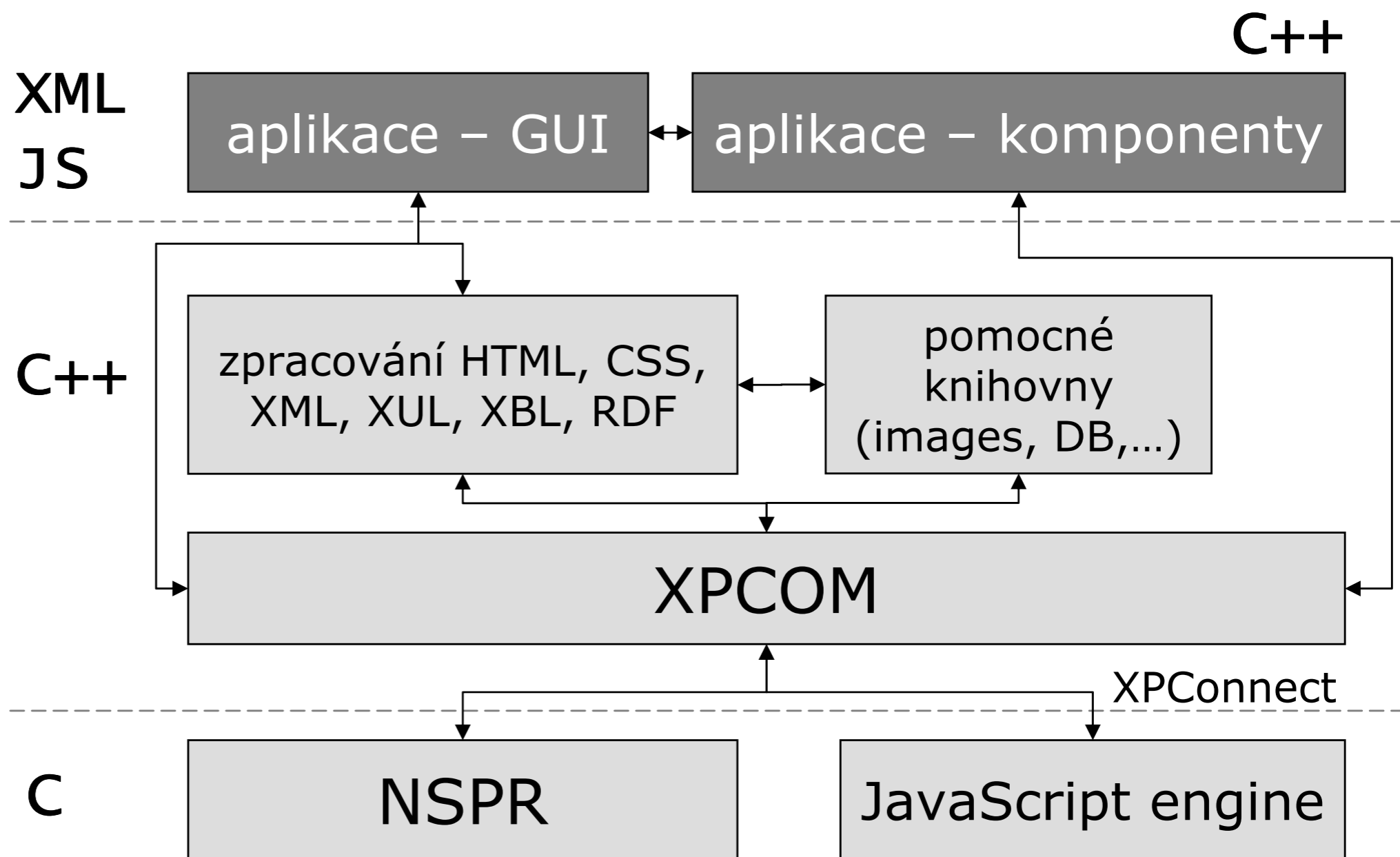


Co je vlastně Mozilla?

- 1998 – Netscape uvolňuje svůj prohlížeč jako open source (Mozilla.org)
- použitá licence MPL umožňuje začlenění i do komerčních aplikací
- rozhodnutí prohlížeč z větší části přepsat, vznik Mozilla Suite
- vzniká rozsáhlá aplikační platforma
- renderovací engine Gecko
- 2003 – vznik Mozilla Foundation
- podzim 2004 – Firefox/Thunderbird 1.0



Architektura Mozilly





NSPR (Netscape Portable Runtime)

- low-level knihovna napsaná v C
- abstrakce nad jednotlivými OS (běží ve Win32, Mac OS, 20+ unixů)
- převzata ještě ze starého Netscape Navigatoru, kde byla původně kvůli Javě
- zkompilovatelná samostatně, lze použít mimo Mozillu



NSPR – co všechno umí?

- portabilní datové typy a makra (např. 64bitovost)
- vlákna, synchronizace
- I/O nad soubory
- základní síťové funkce
- datum a čas
- správa paměti
- načítání dynamicky linkovaných knihoven (DLL/.so)



JavaScript Engine

- „interpret“ JavaScriptu 1.5 (nadmnožina standardu ECMA 262 Edition 3 - ECMAScript)
- napsal z větší části Brendan Eich, autor JavaScriptu
- napsaný v C, vysoce optimalizovaný
- nezávislý na ostatním kódu, lze zkompilovat samostatně (použití pro embedding – skriptovací jazyk pro aplikace)
- připravuje se podpora E4X

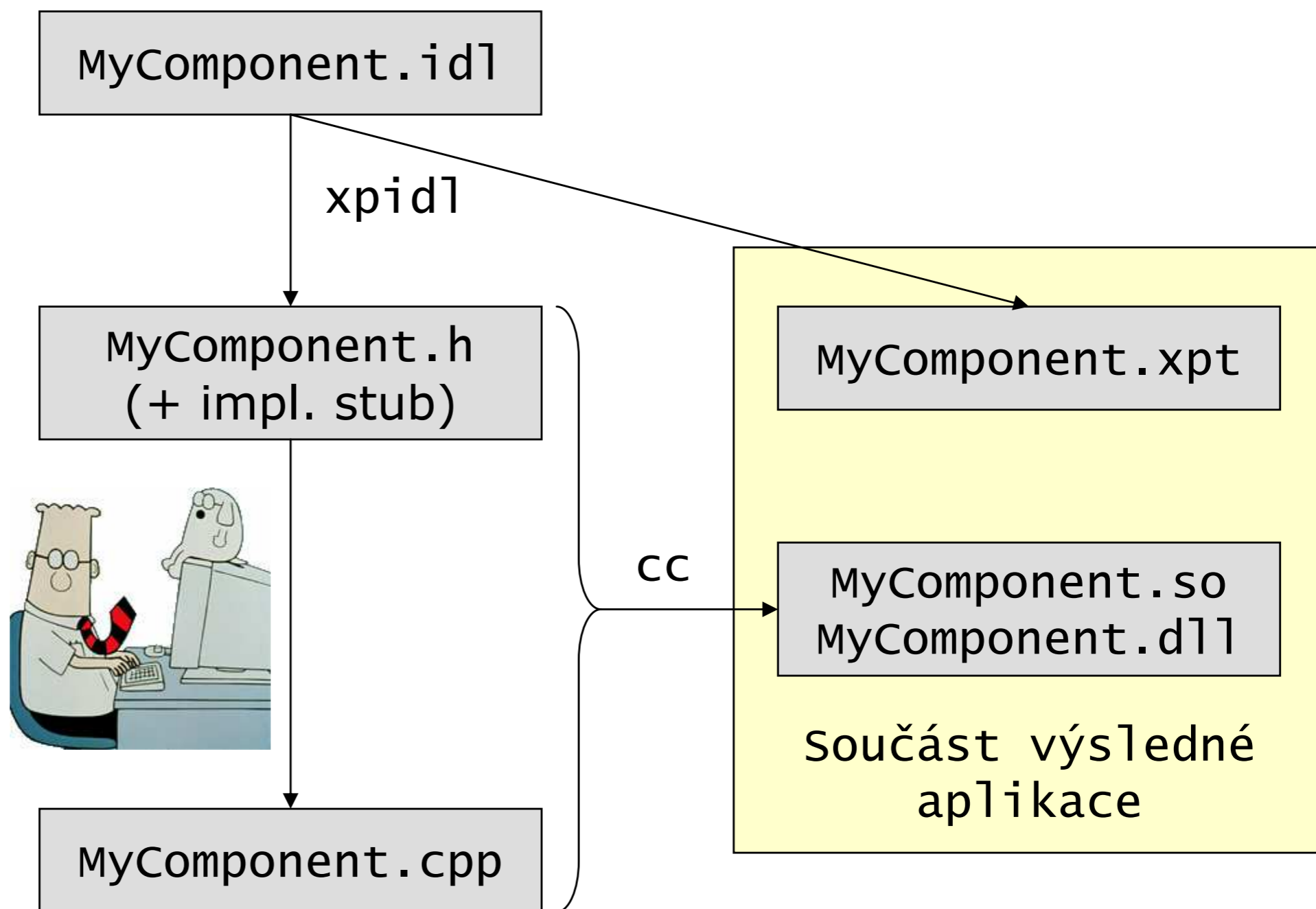


XPCOM

- **Cross Platform Component Object Model**
- obdoba Microsoft COM/DCOM nebo CORBA, ale in-process
- tvorba komponent, modularizace
- popis rozhraní komponenty v XPIDL (dialekt IDL)
- implementace typicky v C++ (generované hlavičky a šablony kódu), možno i v JavaScriptu (XPConnect)



XPCOM – vznik komponenty





XPCConnect

- umožňuje nezávislost XPCOM na jazyku
- typicky: objekt implementován v C++, volán z JavaScriptu (takto vystavěn celý DOM)
- ošetřuje volací konvence, typy parametrů,...
- řeší object lifecycle (XPCOM – reference counting, JS – garbage collector)



Gecko

- nejvyspělejší renderovací engine na světě:
 - HTML 4.01, XHTML 1.0, XML 1.0, XUL
 - CSS 1, 2, 3
 - DOM 1, 2
 - RDF
 - PNG, GIF a JPEG
- další součásti Gecko Runtime Environmentu:
 - NSPR
 - JavaScript 1.5 (kompatibilní s ECMA-262 Edition 3)
 - XPCOM core
 - Unicode, podpora L10N/I18N
 - Necko (HTTP 1.1, FTP, IMAP,...)
 - a další...





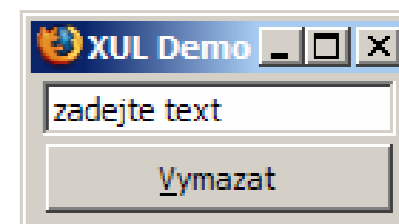
XUL (čti *zůl*) – o co jde?



- **XML User interface Language**
- Problém:
 - Mozilla multiplatformní, ale GUI na každé platformě úplně jiné
 - nutno psát obdobný kód vícekrát
- Řešení:
 - GUI zapsat deklarativně v XML
 - využít DOM a JS k „oživení“ a CSS k úpravě vzhledu
 - vše se píše jen jednou
 - vícekrát je jen zobrazovací kód
 - vše funguje „jako ve webových stránkách“ – steep learning curve



XUL - ukázka



- demo.xul

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/"
  type="text/css"?>
<window id="demo-window" title="XUL Demo"
  xmlns="http://www.mozilla.org/keymaster/
  gatekeeper/there.is.only.xul">
  <script type="application/x-javascript"
    src="demo.js" />
  <vbox>
    <textbox id="text" value="zadejte text" />
    <button label="Vymazat" oncommand="delText();" />
  </vbox>
</window>
```

- demo.js

```
function delText() {
  document.getElementById("text").value = "";
}
```



XUL – widgety (1/3)

<code><button></code>	
<code><checkbox></code>	<input checked="" type="checkbox"/> zaškrtni mě
<code><radiogroup></code> <code><radio></code>	<input type="radio"/> červená panda <input type="radio"/> opereta <input type="radio"/> modré é <input type="radio"/> dobyvatel
<code><textbox></code>	



XUL – widgety (2/3)

`<listbox>`
`<listitem>`

jablka
hrušky
švestky

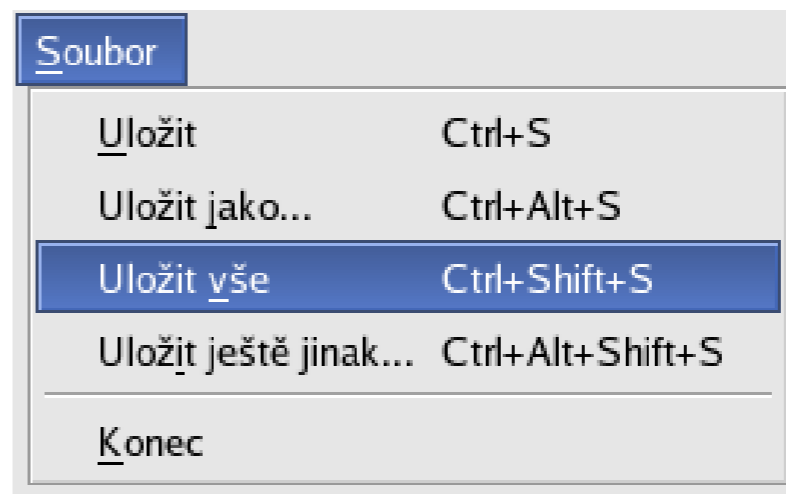
`<tree>`
`<treecols>`
`<treecol>`
`<treechildren>`
`<treeitem>`

Název	Počet	
<input type="checkbox"/> Ovoce		
jablka	3	
hrušky	7	
švestky	11	
<input type="checkbox"/> Zelenina		
mrkev	13	
hrách	17	
brambory	19	



XUL – widgety (3/3)

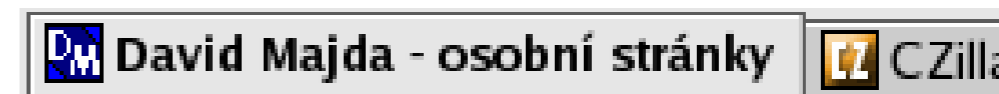
<menubar>
<menu>
<menupopup>
<menuitem>
<menuseparator>



<toolbox>
<toolbar>
<toolbarbutton>
...



<tabbox>
<tabs>
<tab>
...





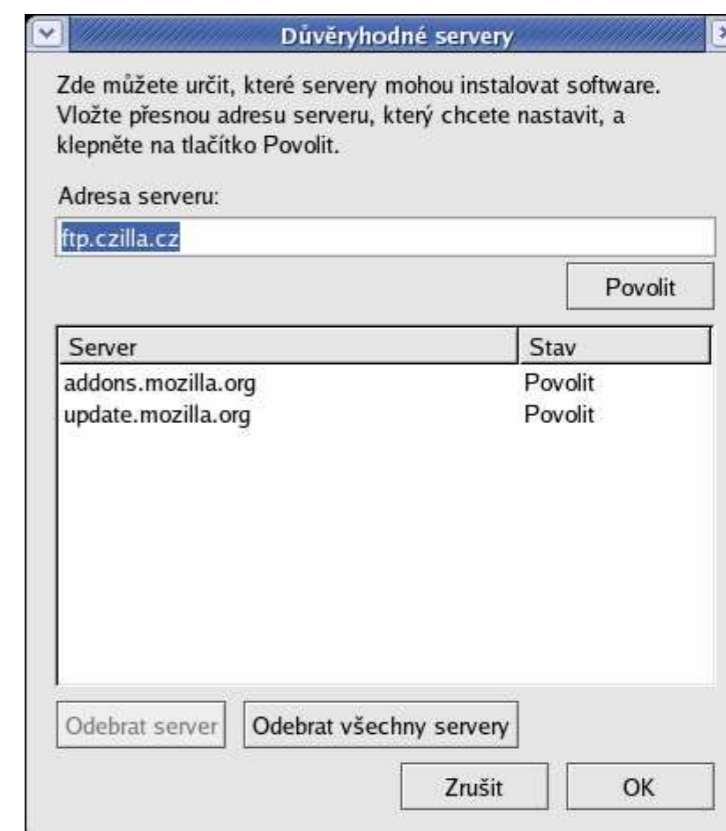
Tvorba aplikací

- **rozšíření** – nejčastější způsob psaní aplikací nad Mozillou
 - drobné odlišnosti mezi Mozilla Suite a Firefoxem/Thunderbirdem
 - v našich ukázkách se zaměříme na Firefox a Thunderbird
- **samostatná aplikace**
 - postavená nad GRE
 - netriviální věc
 - nutný build environment Mozilly
 - úpravy na úrovni C/C++



Co to je rozšíření, instalace

- XPI soubor (obyčejný ZIP)
 - install.rdf
 - content, skin, locale
- Extension Manager (Správce rozšíření)
- instalovat lze jen z povolených domén





Integrace rozšíření do aplikace

- Konvenční řešení:
 - aplikace poskytne API
 - omezené možnosti, autoři aplikace musí předvídat, do čeho bude chtít rozšíření zasahovat
- Řešení Mozilly:
 - aplikace je také v XUL
 - stačí definovat „propojovací body“, tj. určit, kam do aplikace přivěsit XML podstrom rozšíření
 - téměř neomezené možnosti úprav

```
<?xml version="1.0"?>
<overlay xmlns="...">
  <menupopup id="menu_ToolsPopup">
    <menuItem label="Boogie" accesskey="B"
      position="3" oncommand="openBoogie();" />
  </menupopup>
</overlay>
```



Test: Je Mozilla vhodná pro vaši aplikaci? (1/3)

- **Potřebujete multiplatformnost?**
+1 bod
- **Potřebujete pracovat s HTML či XML?**
+1 bod
- **Potřebujete pracovat se sítí?**
+1 bod
- **Potřebujete lokalizovatelnost?**
+1 bod
- **Chcete rychle vyvíjet a očekáváte časté změny v zadání?**
+1 bod



Test: Je Mozilla vhodná pro vaši aplikaci? (2/3)

- **Aplikace je velmi velká?** (-1 bod)
 - začne vadit JavaScript
- **Potřebujete náročnou grafiku?** (-1 bod)
- **Potřebujete hrubý výkon či nativní GUI?** (-1 bod)
 - C/C++ a knihovny na GUI (MFC, QT, wxWindows)
- **Potřebujete přístup z kteréhokoliv počítače na světě?** (-1 bod)
 - lepší je webová aplikace v HTML



Test: Je Mozilla vhodná pro vaši aplikaci? (3/3)

- **počet bodů ≥ 3**
 - zkuste popřemýšlet nad řešením založeným na Mozille
- **počet bodů < 3**
 - je potřeba zvolit jiné řešení
 - C, C++, Java, C#, Delphi, HTML/PHP...



Materiály

- Tato prezentace
 - ke stažení na www.czilla.cz
- Další materiály
 - www.xulplanet.com – průvodce a reference XUL
 - www.mozilla.org/docs – oficiální vývojářská dokumentace
 - wiki.mozilla.org – dokumentace, odkazy, plány do budoucna



Děkuji za pozornost

Dotazy?