Tvorba WebOS

(aneb aplikace v aplikaci)

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Návrh webového operačního systému

- Jádro systému
- Webtop
- Aplikace
- Offline technologie
- Datová úložiště
- Interoperabilita

PHP

- PHP 5.3, Nette Framework
- skriptovací jazyk (není kompilovaný)
- neefektivní
- HipHop - optimalizace; převod na C++ a poté kompilace!
- (Facebook - hodně hýbe s PHP)

Albireo Kernel

- autentizace, autorizace
- práce s databází
- logování
- debugging
- lokalizace
- práce se souborovým systémem
- bezpečnost operací

Webtop

- obyčejné HTML - přenositelnost
- hrátky s HTML 5, ale business sféra nepřipravena
- JavaScript, framework jQuery (a nadstavba jQuery UI)
- knihovna Albireo.js
- stará se o kontextová menu, vykreslování ploch, práci s ikonami, označování, drag and drop, AJAX, klientskou část práci s aplikacemi, procesy, API, dialogová okna
- stále se vylepšuje
- momentálně začínáme skinovat
- JSON

Aplikace

1. primitivní - html + javascript (v podstatě jen běžící šablona)
2. iframe aplikace - aplikace třetích stran, které nebyly vytvořeny přímo pro systém mohou využívat naše API pro práci se systémem (kolaborace, sdílení, ...)
3. aplikace vytvořené přímo v systému
   - vytvořit vhodné nástroje pro tvorbu
   - např. značkování pro typické desktop aplikace (menu, záložky, stromová struktura, ...)

Aplikace (2)

Dva možné směry pro vývoj:

- Vlastní "programovací jazyk"
  - velká bezpečnost
  - dovolím pouze to, co sám chci
  - musím vytvořit vhodné nástroje
  - cesta do pekel (interoperabilita = 0)
- Vyvíjejte si v čem chcete
  - dobrý koncept
  - nebezpečné
  - dobré zabezpečení v databázi
  - dobře postavené API
  - aplikace VNĚ systému! (RPC)

Offline

- Jak používat systém, když nemám Internet a všechna data mám jen online.
- offline úložistě
- umožňují přímo udržovat data ve vašem prohlížeči

Testování a logování

aneb jak na chyby uvnitř i venku

Testování

- testy se snaží ověřit kvalitu kódu
- žádná formální verifikace
- typické a limitní hodnoty
- **unit testy**
  - automatizované testování
  - "háčky" u subversion
  - starý kód funkční, napomáhají práci v týmu
- **Selenium**
  - když kliknu sem, uvidím tohle
- **Debugging**
  - v PHP bývá trošku problém ->ladicí nástroj
  - FireBug, WDT

Logování

- debugging samozřejmě jen pro vývojáře
- jinak logujeme
- php chyby
- pokusy o porušení zabezpečení
- i na "nejnižší" úrovni (triggery v databázi)
- **na chybové stavy reagujeme**
- zachytávání výjimek
- kritický a normální mód
- v produčním nasazení se musí systém postarat, aby uživatel nic nepoznal nebo aspoň byl dostatečně informován o problémech, které nastaly

Zabezpečení systému

Autentizace

- obecný uživatel v databázi, který má pouze právo číst jména a hesla
- heslo sha512
- rainbow tables -> sůl a pepř
- proč konstantní
- proč variabilní
- po přihlášení - každý vlastní DB účet
- heslo opět "obalená" hash, sha1 (o dost rychlejší, je častěji třeba a prozrazení většinou není zase tak kritické)
- MasterPassword

Autorizace

- ACL
- každý svůj DB účet
- logování změn
- využití MVC k právům
- uživatel má role
- zdroje jsou presentery
- permissions jsou akce presenterů
- u souborů trošku jinak

Podvody

- ochrana před
  - SQL injection
  - XSS (Cross-site scripting)
    - Content-aware escaping
  - CSRF (Cross-site request forgery)
  - HTTPS
  - Azure: 512b symetrické klíče, digitální podpisy
  - bezpečné šablony
  - obecně vše defaultně zabezpečené a výjimky se musejí povolovat

Dotazy?

KONTAKT: kunc@albireo.eu,
kuncajs@gmail.com