

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# Vytvoření specializovaného klienta pro podporu e-learningového portálu

DIPLOMOVÁ PRÁCE

**Marek Vantuch**

Brno, 2014

## Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Marek Vantuch

**Vedoucí práce:** Ing. Leonard Walletzký, Ph.D

## Poděkování

Chtěl bych obzvláště poděkovat Ing. Leonardu Walletzkému, Ph.D za příležitost pracovat na tomto projektu a za maximální podporu při jeho vypracování. Dále JUDr. Jindřišce Šedové, CSc. za konzultace a podporu při vývoji. Děkuji také všem svým spolupracovníkům za ohleduplnost a trpělivost nutnou s ohledem na ztížené podmínky, které byly způsobeny mou prací ze zahraničí. Speciálně Mgr. Ondrovi Maternovi za návrh celého řešení, Bc. Ivaně Haraslínové za design a jeho aplikaci a Mgr. Radkovi Foltýnovi za všeobecnou koordinaci projektu.

## Shrnutí

Tato práce popisuje implementaci internetového portálu pro fakultu Ekonomicko-správní Masarykovy Univerzity. Jsou zahrnuty dvě hlavní části, popis úprav existujícího řešení postaveného na open-source CMS systému Drupal a jeho propojení pomocí vzdálené plochy na servery ASPI. Při vývoji byl kladen důraz na udržitelnost a strukturalizaci řešení. Ačkoliv Drupal typicky ukládá nastavení v databázi, pomocí dostupných nástrojů byla tato data exportována do PHP kódu a byl nastaven proces jejich přenesení do testovacího či produkčního prostředí. Pro ulehčení byly vytvořeny skripty, které byly propojeny s „Continuous Integration“ serverem. Důraz byl kladen i na uživatelskou přívětivost a jednoduchost, kterých bylo dosaženo sjednocením platformy postavené na Drupalu a existující implementace vzdálené plochy využívající HTML5 a JavaScriptu a implementací přihlašování autorizací skrze Informační systém Masarykovy Univerzity.

## **Klíčová slova**

Guacamole, vzdálená plocha, ASPI, ESF, Drupal, Team City, Drush, Phing, CORS, WebSockets, JAVA, PHP, JavaScript

# Obsah

1	Úvod . . . . .	2
2	Analýza . . . . .	3
2.1	Existující řešení . . . . .	3
2.2	Použité technologie . . . . .	4
2.2.1	Jádro CMS a přidružené technologie . . . . .	4
2.2.2	Technologie připojení ke vzdálené ploše . . . . .	6
2.2.3	Podpora nasazení a vývoje projektu . . . . .	7
2.3	Struktura portálu v prostředí Drupalu . . . . .	8
2.3.1	Typy polí použitelné pro vlastnosti typů obsahu . . . . .	9
3	Implementace portálu v Drupalu . . . . .	11
3.1	Aktualizace Drupal 6 na Drupal 7 . . . . .	11
3.2	Odebrané moduly . . . . .	13
3.3	Nově přidávané moduly . . . . .	14
3.3.1	Typy obsahu . . . . .	17
3.3.2	Slovníky . . . . .	19
3.4	Téma vzhledu a jeho rozložení . . . . .	20
3.5	Proces přihlášení uživatelů . . . . .	22
3.6	Struktura zdrojových souborů . . . . .	23
4	Implementace připojení ke vzdálené ploše . . . . .	25
4.1	Architektura připojení ke vzdálené ploše . . . . .	25
4.2	Ostranění autentizace uživatelů a vytvoření vlastní autorizace . . . . .	27
4.3	Přesun JavaScript kódu a CSS stylů do modulu Drupalu . . . . .	27
4.4	Implementace CORS . . . . .	29
4.5	Implementace úpravy nastavení z administrace portálu . . . . .	29
5	Instalace, konfigurace a správa portálu . . . . .	30
5.1	Požadavky a příprava systému . . . . .	30
5.2	Nasazení (deployment) a jeho možnosti . . . . .	31
5.3	Instalace informačního systému (Drupal) . . . . .	32
5.4	Instalace Guacamole . . . . .	33
5.5	Administrace . . . . .	34
5.6	Výkonové optimalizace . . . . .	36
6	Organizace vývojového procesu . . . . .	38
6.1	Organizace řízení projektu . . . . .	39
7	Závěr . . . . .	41
	Reference . . . . .	48

# 1 Úvod

Tato práce se zabývá projektem, jehož cílem je vytvoření informačního systému pro studenty Ekonomicko-správní fakulty Masarykovy Univerzity. Stávající řešení v podobě webové stránky bylo postaveno na open-source projektu *Drupal*<sup>1</sup> verze 6 a celá jeho struktura byla řešena na bázi slovníků a stránek, bez rozlišení typů obsahu. Mgr. Ondřej Materna ve své práci zanalyzoval možnosti zlepšení řešení a reálné požadavky studentů. Výsledkem je návrh řešení, který se neslučoval s existujícími stránkami, které tak musely být razantně přepracovány. *Drupal* verze 6 existoval již více než pět let a dva roky existuje i jeho aktualizovaná verze 7[2]. Ten poskytuje vyšší rychlost, stabilitu i rozšíření díky širší podpoře komunity a nezávislých vývojářů. Stávající řešení je hlouběji rozebráno v kapitole 2, ve které jsou zároveň popsány technologie využití k implementaci nové verze portálu a základní architektura řešení.

Důležitou částí portálu je propojení s ASPI za pomoci klienta vzdálené plochy. Ve stávajícím řešení byli studenti nuceni používat jednu ze dvou možností připojení:

1. lokální instalace nativní aplikace ASPI na klientský počítač a její spuštění - odkazy se otevírají přímo v aplikaci
2. připojení se na vzdálenou plochu pomocí jednoho z veřejně dostupných klientů a používání nativní aplikace zde

Odkazy na stránkách se však automaticky nepřenášely na vzdálenou plochu a celkově vyžadovalo toto řešení vyšší technickou zdatnost uživatelů.

Hlavním cílem projektu je uživatelská přívětivost a proto byl při výběru řešení kladen důraz převážně na jednoduchost a minimální požadavky na klientské zařízení a uživatele. Bylo zvoleno řešení postavené na prvcích jazyka HTML5 a open-source nástroji *Guacamole*. Ten poskytuje klienta vzdálené plochy čistě skrze okno prohlížeče. Komunikace se vzdálenou plochou probíhá za pomoci sprostředkovatelského proxy serveru. Architektura je detailně popsána v kapitole 4.

Jak bylo zmíněno výše, tato práce staví na diplomové práci Mgr. Ondřeje Materny „Návrh a realizace právního portálu pro ESF MU“[8]. Její obsah je důkladně zanalyzován a z ní vyplývající poznatky uplatněny na prostředí CMS *Drupal* a jeho možnosti. Proces aktualizace a implementace funkční a základ vzhledové stránky řešení jsou popsány v kapitole 3. Vzhled samotný není předmětem této práce, nýbrž práce Bc. Ivany Harasíkové.

Z důvodu spolupráce mezi více studenty byl vytvořen základní systém pro vedení projektu, využívající prostředí systému *GitHub* a jeho stručný popis je obsažen v kapitole 6. Projekt je veřejně dostupný na adrese <https://github.com/kanei/esf-mu-portal>.

---

1. <https://drupal.org>

## 2 Analýza

Projekt byl od začátku koncipován jako aktualizace stávajícího řešení, do kterého budou promítnuty zkušenosti a znalosti nabyté za dobu jeho fungování, zkombinované s moderními technologiemi, které nebyly v době vývoje předešlé verze dostupné. Návrh klade důraz na minimalizaci možných uživatelských chyb a jednoduchost a názornost vytváření obsahu. Těchto cílů dosahuje za pomoci definování vztahů mezi obsahem a omezením možností úprav automatickým generováním prvků webu, jakými jsou například seznamy a tabulky zobrazující propojení elementů.

### 2.1 Existující řešení

Při analýze bylo nutné zohlednit stávající řešení a možnosti jeho rozšíření. Existující portál využíval *Drupal* v 6. verzi, který je již značně zastaralý a jeho rozšiřitelnost omezená, či neadekvátně komplikovaná. Řešení bylo rozšířeno řadou volně dostupných modulů, které dohromady poskytovaly komplexní strukturu nabídky menu vytvořenou namíru požadavků fakulty. Struktura byla navržena v době, kdy nebylo známo, že systém bude pro zobrazování zákonů využívat externí software ASPI, proto bylo počítáno s možností zobrazení zákonů přímo v systému.

Autentizace na server probíhala pouze za pomoci jednotného hesla, které bylo měněno ve čtvrtletních cyklech. Heslo nebylo nijak vázáno na jednotlivé uživatele a tento způsob nedovoľoval téměř žádnou kontrolu nad uživateli přistupujícími k obsahu - po získání hesla byl schopen na server přistupovat kdokoli a zabránění přístupu bylo možné pouze na bázi IP adresy uživatele.

#### Struktura menu

Navigace systému se nachází v levé části stránky a obsahuje v jediné stromové struktuře veškeré odkazy na obsah portálu.

- **[název předmětu 1]**  
rozbalí/sbalí menu
  - ★ **[název předmětu 1]**  
odkaz na stránku předmětu, obsahující pouze název a informace o spolufinancování Evropská Unie (EU)
  - ★ **Studijní text**  
rozbalí/sbalí menu
    - **Studijní text**  
odkaz na stránku studijní text, která je typicky prázdná
    - **Úvod**  
odkaz na stránku obsahující typicky úvodní text k danému předmětu



- **[studijní text 1]**  
jeden z odkazů na stránku s obsahem přednášek, či jinak tématicky oddělených informací k předmětu
  - **[studijní text 2]**
- ★ **Ostatní studijní materiály**  
tento a následující již nejsou pevně dané a vyskytují se nepravidelně pouze u některých předmětů
- ★ **Prezentace**
- ★ **Právní předpisy**
- ★ **Judikáty**
- ★ **Procvičování**
- **[název předmětu 2]**
- **[název předmětu 3]**

Z výše popsané struktury vyplývá, že její vlastnosti nesplňují požadavky moderních informačních systémů. Odkazy mají nekonzistentní chování - buď se rozbalí pod-menu s dalšími odkazy, nebo prohlížeč přejde na novou stránku. Uživatel proto nemůže jednoduše vědět, jaké chování od své akce očekávat. Struktura obsahuje duplicitu (název předmětu a odkaz na studijní text se opakují a v obou případech mají jiný význam). Dále není určená společná struktura všech předmětů a tím může lehce dojít ke zmatení uživatelů, kteří musejí pokaždé informace hledat na jiném místě (stránky *Ostatní studijní materiály*, *Prezentace*, ...). Některé stránky také místo agregace užitečných informací obsahují buď pouze banner, nebo jsou úplně prázdné (úvodní stránka předmětu) v místech.

## 2.2 Použité technologie

Základními technologiemi jsou *Drupal* a *Guacamole*, které definují vyžití dalších technologií. Základem jsou programovací jazyky *PHP*, *JAVA* a *JavaScript* a výstup je realizován pomocí *HTML* (v.4 a v.5) a *CSS* (v.2 a v.3). Ačkoliv bylo využito velké množství modulů, zmíněny budou pouze ty, které měly zásadní vliv na funkcionalitu nebo proces vývoje.

### 2.2.1 Jádru CMS a přidružené technologie

#### Drupal

<http://drupal.org>

V době psaní této práce byl *Drupal* světově třetím nejrozšířenějším CMS[1]. Založený na jazyce *PHP*, klade důraz na vývojáře a na možnosti úpravy stránek, proto často bývá označován za framework. Konkurenční CMS *WordPress* cílí na uživatelskou jednoduchost a většina stránek na něm postavných je lehce rozpoznatelná, zatímco

*Drupal* je možné změnit od základu a jeho API pro tvorbu modulů poskytuje komplexní možnost úpravy. Minimálním požadavkem pro spuštění je *PHP* verze 5.3 a i když některé jeho části jsou již implementovány objektově, celkově převládají čisté funkce s množstvím proprietárních principů řešení. Mezi ty patří například `hook_api`, poskytující přípojný body pro další moduly a tím rozšiřitelnost částí systému, nebo systém vzorů (templates), které pomocí speciální jmenné konvence umožňují měnit výpis html kódu prvků webu (viz. kapitola 3.4).

*Drupal* podporuje využití tzv. instalačních profilů. Jedná se o speciální moduly neposkytující samy o sobě žádnou funkcionalitu, ale sdružující informace o potřebných modulech a jejich iniciálním nastavení. To lze využít pro specifické distribuce, například vícejazyčná instalace, řešení pro blog, řešení orientované na výkon a podobně.

### **Kerberos** (Protocol)

<http://web.mit.edu/kerberos>

Vyvinut na *Massachusettském technologickém institutu (MIT)*, protokol *Kerberos* poskytuje autentizační funkcionalitu za pomoci symetrické kryptografie (jediný klíč je použit pro šifrování i dešifrování). Autentizace je provedena za pomoci důvěryhodného třetího serveru (v tomto případě Informační systém (IS) Masarykovy Univerzity).

### **Omega** (Drupal Theme)

<http://drupal.org/project/omega>

Pro *Drupal* existuje nespočet témat vzhledu, které se starají o formátování výstupu html kódu a s ním spojených kaskádových stylů CSS. Projekt *Omega* je zaměřen na responsivní web design a poskytuje implementaci stylů pro různá zobrazovací zařízení. Zatímco třetí verze obsahuje komplexní administrační rozhraní, ve kterém lze nastavit rozmístění a poměry mezi bloky stránky, přístup čtvrté verze se zaměřil na implementaci v kódu. Poměry mezi jednotlivými bloky jsou zapsány pomocí nástroje *SUSY* a pomocí knihovny *Breakpoints*<sup>1</sup> a vlastností `@Media-Queries` se mezi sebou přepínají.

### **SASS** (Syntactically Awesome Style Sheets)

<http://sass-lang.com/>

Díky tématu vzhledu *Omega* se nabídla možnost využití nástroje *SASS* pro generování popisu vzhledu *HTML* elementů. Namísto přímého použití CSS je kód napsán ve formátu *SASS* a poté zkompilován do CSS. Tento přístup přináší bezproblémovou kompatibilitu se všemi prohlížeči doplněnou o rozšířené možnosti definice pravidel, jako je využití proměnných, vnoření pravidel, in-line import a další. Kromě výrazného vylepšení čitelnosti a možnosti seskupování pravidel bez zvyšování zátěže na přenos dat tak lehce lze dosáhnout i snížení programátorské náročnosti a zvýšení efektivity.

1. <http://xoxco.com/projects/code/breakpoints/>

**COMPASS**<http://compass-style.org/>

Soubory *SASS* musí být kompilovány do *CSS* pro zobrazení prohlížečem a protože manuální kompilace vyžaduje opakované spouštění příkazů a je tím pádem náchylná k opomenutí, nabízí se využití programu *COMPASS*. Přestože poskytuje širokou škálu funkcionality určené pro ulehčení práce designérům, pro potřeby tohoto projektu byla využita pouze automatická konverze z *SASS* do *CSS* dle definovaných pravidel za pomoci příkazu `$ compass watch`. Po spuštění démona jsou kontrolovány všechny v souborech ve složce a automaticky regenerovány výstupní *CSS* soubory pro načtení prohlížečem.

**SUSY**<http://susy.oddbird.net/>

Další technologií využitou v tématu vzhledu je *SUSY* - implementace responsivní mřížky pro *COMPASS*. Za pomoci jednoduchých pravidel lze nadefinovat rozdílné rozvržení stránky v závislosti na rozlišení zobrazovacího zařízení. Například monitoru počítače s rozlišením vyšším než 1024 bodů můžeme postranní panel zobrazit nalevo, zatímco na mobilním zařízení můžeme text zmenšit a zobrazit v horní části stránky spolu s vypuštěním některých nedůležitých bloků. Celá stránka je může být rozdělena na počet sloupců, které mohou být dynamicky vyplňovány a pravidla lze velmi jednoduše zapisovat bez nutnosti řešení problémů s kompatibilitou mezi prohlížeči.

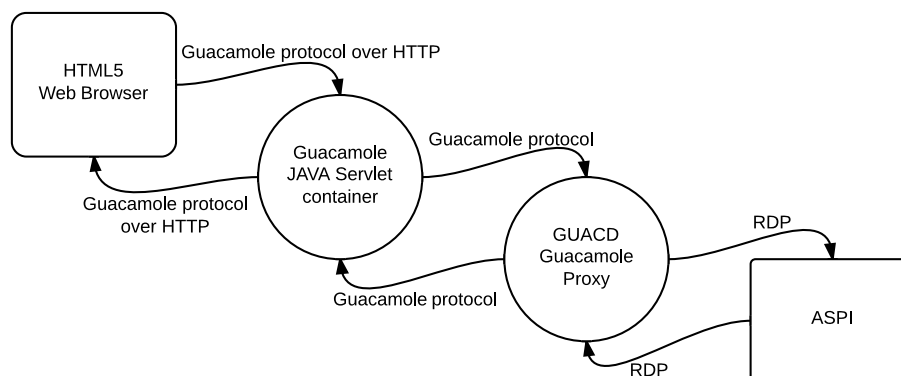
**2.2.2 Technologie připojení ke vzdálené ploše****Guacamole**<http://guac-dev.org/>

Jak je popsáno na stránkách této knihovny, jedná se o potvrzení myšlenky připojení ke vzdálené ploše skrze webový prohlížeč. Namísto potřeby instalace speciálního klienta je možné používat jakýkoliv prohlížeč podporující některé z technologií *HTML5* (např. *canvas*). Připojení probíhá skrze proxy server implementovaný v jazyce *C*, který zprostředkovává připojení a komunikaci. Ke klientovi putují již jen obrazová data a povely, obojí zakódováno v proprietárním protokolu, který umožňuje snížit odezvu obrazu a dekodovat jej v průběhu přijímání dat. Postup komunikace mezi klientským prohlížečem a vzdálenou plochou je blíže vyobrazen na diagramu 2.1.

Propojení mezi webovým prohlížečem a proxy serverem je provedeno za pomoci *AJAX* požadavků v čistém jazyce *JavaScript*, zatímco propojení z proxy dále podporuje protokoly *SSH*, *VNC* a *Remote Desktop Protocol* (*RDP*).

**CORS**<http://www.w3.org/TR/cors/>

Kvůli bezpečnostním rizikům je *JavaScriptu* běžícímu v prohlížeči zakázáno přistupovat k jiným doménám, tento přístup se nazývá *Same-origin policy* a slouží k zamezení podvodných stránek a zvýšení internetové bezpečnosti. Veškeré požadavky



**Obrázek 2.1:** Architektura komunikace nástroje Guacamole skrze vzdálenou plochu

směřující na jinou doménu tak musejí být vykonávány přímo ze serveru a prohlížeči zasílány pomocí protokolu AJAX. Tento přístup je často zbytečně náročný a v dnešní době bývá nutné toto omezení obejít. K tomu slouží Cross-origin Resource Sharing (CORS) [6], kdy za pomoci *HTML* hlaviček přidaných do komunikace mezi serverem a prohlížečem a nastavení serveru je umožněna komunikace s jinými doménami. Pokud server odešle zpět hlavičku `Access-Control-Allow-Origin: [doména]`, prohlížeč ji zanalyzuje a případně umožní komunikaci. K tomuto je nutné využít *XMLHttpRequest* (*XDomainRequest* v případě Internet Explorer (IE)) s atributem `WithCredentials`, o zbytek se postará prohlížeč se serverem a komunikace funguje stejně jako v případě klasických požadavků.

## ASPI

<http://www.systemaspi.cz/>

Systém *ASPI* poskytuje jeho uživatelům komplexní právní informace z prostředí České republiky. Velký důraz je kladen na možnost jednoduchého vyhledávání a verzování obsahu, stejně jako dostupnost skrze hlavní operační systémy. *ASPI* funguje jako aplikace instalovaná na klientském prostředí, ale nabízí také možnost vytvoření serverů dostupných k připojení skrze vzdálenou plochu, čehož je využito v případě tohoto projektu.

### 2.2.3 Podpora nasazení a vývoje projektu

#### Drush

<http://github.com/drush-ops/drush>

Pro ulehčení administračních úkonů nad instalací *Drupalu* byl komunitou vyvinut program *Drush*, poskytující textové administrační rozhraní nad *Drupalem* v terminálové konzoli. V základu jsou umožněny operace nad moduly a tématy vzhledu a každý z modulů může pomocí implementace funkcí v souboru `[název].drush.inc` rozšířit rozhraní vlastními příkazy.

**Phing**<http://www.phing.info/>

*Phing* je nástroj určený ulehčení nasazení a nastavení *PHP* projektů, fungující na bázi *XML* konfiguračního souboru. Je založený na podobné myšlence jako *Apache ANT*, který poskytuje podobnou funkcionalitu pro jazyk *JAVA*. V konfiguračním souboru lze definovat pravidla, proměnné a závislosti vedoucí k nasazení projektu, kontrolám kódu, či automatizaci jakýchkoliv dalších akcí, potřebných k správnému chodu aplikace.

**DrushTask**<https://drupal.org/project/phingdrushtask>

Tento projekt využívá možnosti vývoje vlastních operací (task) v nástroji *Phing*. Za pomoci příkazu `TaskdefTask`[3] je definovaná operace `<drush>`, poskytující rozhraní k programu *Drush*. Tím je umožněno automatizovat většinu administrátorských činností nad projektem a případně implementovat automatickou analýzu kódu a testování pomocí CI.

**Maven**<http://maven.apache.org/>

Podobně jako *Phing* pro *PHP*, *Maven* poskytuje automatizaci nasazení *JAVA* projektů (převážně). Jako základ také používá *Apache ANT*, který rozšiřuje a specializuje se na možnost znovuvyužití skriptů pro více projektů.

**Team City**<http://www.jetbrains.com/teamcity/>

Podobně jako open-source komunitou často využívaný *Jenkins* či *Hudson*, *Team City* je implementací CI serveru. V případě tohoto řešení jde o čistě komerční produkt, který je ale pro malé projekty dostupný zcela zdarma (je omezen na počet projektů). *Team City* umožňuje komplexní nastavení projektů a jednotlivých úkonů, kontrolujících zdrojový kód, nebo provádějících operace nasazení na produkčním prostředí. Je také poskytována integrace s většinou moderních CVS.

**GitHub**<https://github.com/>

Pro udržování historie kódu a pro spolupráci více autorů byla vybrána CVS technologie *Git* a online portál *GitHub*, který poskytuje jednoduchou instalaci a online správu uložení. Všechny úpravy mohou být jednoduše zobrazeny přímo v prohlížeči a provázány s úkoly vytvořenými v úkolovém managementu na stejném internetovém portálu.

### 2.3 Struktura portálu v prostředí Drupalu

Při vývoji informačních systémů jsou typicky data ukládány v databázi. To stejné platí i pro Drupal, pouze s tím rozdílem, že poskytuje administrační rozhraní schopné dynamicky vytvářet úložiště pro data. Namísto vytváření nové tabulky jsou uživatelé

**Tabulka 2.1:** Základní typy entit v Drupalu

Typ entity	Strojový název	Dostupnost polí	Rozšiřitelnost
Komentář	comment	✓	✓
Soubor	file		
Slovník	vocabulary		
Uzel	node	✓	✓
Uživatel	user	✓	✓
Záznam slovníku	term	✓	✓

schopni definovat typ obsahu, který navíc může dědit určité vlastnosti. Základním typem je Entita, poskytující možnost ukládání základních informací. Entita označuje abstraktní pohled na data a nemůže být instancionalizována, namísto toho musí nejdříve být definován typ entity, na jehož základě až může být vytvořen typ obsahu. Pro ilustraci, v jádře drupalu existuje typ entity Uzel (Node), definující obsah s vlastní URL adresou a titulkem. Ten však nemůže být vytvořen přímo a administrátor musí definovat typ obsahu (Bundle), který pojmenuje *Stránka* (Page). Tu je již možno vytvářet a následně k ní přistupovat. *Stránka* je tedy uzlem a entitou. Toto však neplatí pro všechny typy obsahu. soubory (File) a slovníky (Vocabulary) lze vytvářet přímo a nelze definovat jejich potomky[10]. Seznam všech základních entit je zobrazen v tabulce 2.1.

Pokud daný typ entity poskytuje možnost ukládání polí, je možné mu přiřadit pole, které si lze představit v analogii databáze jako sloupce tabulky. Každé pole je však reprezentováno jako tabulka v databázi a tak je při přístupu k jedné entitě v Drupalu ve skutečnosti nutno přistoupit do tolika tabulek, kolik má daná entita přiřazených polí. Zároveň jdou pole využívat skrze více typů entit a tak mohou být pole sdílena skrze celý systém, například typ obsahu *Stránka* a typ obsahu *Novinka* mohou oba poskytovat pole *Obrázek*, které bude v databázi ukládáno do jediné tabulky *field\_obrazek*. Instance obou typů obsahu dostanou přiděleno unikátní číslo v rámci všech entit, které určuje její data v databázi. Ne všechna pole musejí být zobrazena na stránce entity, nebo může pro jejich zobrazení být využito speciálních formátovacích technik, jako tabulek, či jiných výpisů. Všechna tato nastavení lze opět definovat buď v kódu, nebo skrze administrační rozhraní *Drupalu*.

### 2.3.1 Typy polí použitelné pro vlastnosti typů obsahu

V drupalu existuje několik základních typů polí, které fungují podobně jako datové typy v klasických programovacích jazycích. Narozdíl od jednoduchých datových typů je ke každému poli přiřazeno jedno či více formátování výsledného *HTML* kódu a zpravidla poskytují rozšiřující nastavení. Formátování jde dále změnit pomocí přepsání základních formátovacích funkcí, či vytvoření a přiřazení funkcí nových.

**Text** Poskytuje pole pro text s omezenou délkou bez možnosti vkládání html značek. Jeho obsah je zadáván pouze v jediném řádku a poskytuje pouze limitované

možnosti nastavení, kterými je délka vstupního pole, či výchozí hodnota.

**Dlouhý text se souhrnem** Narozdíl od typu pole *Text* může v tomto případě být text neomezeně dlouhý a obsahovat značky html kódu. Ty mohou být vkládány buď ručně a nebo být automaticky formátovány pomocí WYSIWYG editoru. Zároveň je pro text možné vyplnit souhrn, který je ve specifických případech zobrazit - jedná se například o seznam aktualit na úvodní stránce. Pokud souhrn není vyplněn, typicky se zobrazí prvních několik set znaků obsahu.

**Hodnota slovníku** V nastavení pole lze vybrat jeden z existujících slovníků (více o slovnících v kapitole 3.3.2). Jeho hodnoty jsou vypsány v jednom z dostupných formátů (radio/zaškrtačková tlačítka) a dle nastavení lze vybrat jen jedna, nebo více možností.

**Logická hodnota** Poskytuje pole pro logickou hodnotu - ano / ne. Pro hodnoty lze vybrat jednu z typicky použitelných nápisů (ano/ne, pravda/nepravda, ✓/x), nebo definovat nápisy vlastní.

**URL adresa** Kromě samotné adresy poskytuje možnost uložení dalších parametrů, podobně jako element `<a>` jazyka html. Mezi tyto parametry patří například titulky, cíl a podobně.

**Média** Pro vkládání médií do systému je využit modul Media (kapitola 3.3), který přidává komplexní řešení pro vkládání multimediálního obsahu. Je možné vkládat dokumenty, soubory pdf, zvuk, obraz, či video a všechny tyto typy obsahu jsou uživateli dostupné v jednotném formátu.

**Váha** I když by se pro určení váhy produktu dalo využít jednoduchého číselného pole, jeden z rozšiřujících modulů poskytuje speciálně využitelné pole pro výběr hodnoty a také pohledy na obsah a řadící funkce, které celou implementaci značně ulehčují.

**Odkaz na entitu** Důležitou vlastností většiny informačních systémů je možnost propojení entit. Pro modelování těchto vztahů existuje typ pole *odkaz na entitu*, u kterého je možné vybrat typy entity, na které je entita navázána a způsob propojení. Za pomoci extra modulů lze modelovat místo jednoduché asociace kompozici (při smazání rodiče jsou smazány i všechny propojené moduly). Lze také definovat, zda je možné pouze vytvářet nové podentity, či přidávat ty již existující.

## 3 Implementace portálu v Drupalu

Pro převedení stávajícího řešení na novou verzi *Drupalu* bylo nutné vykonat několik kroků. Nejprve byla data z produkčního prostředí přesunuta na testovací server, kde bylo řešení zanalyzováno co se týče implementační stránky (k běžícímu portálu nebylo možné přistupovat s administrátorským účtem). Jakmile byly stránky zanalyzovány, byl započat proces jejich aktualizace na novou verzi Drupalu při zachování veškerého obsahu. Tento postup se ukázal jako kontraproduktivní, neboť i když byl obsah stránek převeden, bylo potřeba kompletně změnit jeho strukturu spolu s konverzí na jiné typy obsahu a jiná cílovýmá pole. Všechna data tak nakonec byla přesunuta ručně.

### 3.1 Aktualizace Drupal 6 na Drupal 7

Sedmá verze Drupalu komplexně přepracovala celý kód redakčního systému a zohlednila v té době relevantní požadavky na moderní CMS. Mnoho modulů bylo přesunuto přímo do jádra systému - například podpora vytváření vlastních polí obsahu, pro což byl v *Drupalu* 6 potřeba modul *CCK* (Content Creation Kit). Množství změn je jednoduše vidět i ve velikosti zdrojových souborů, kdy verze 7 obsahuje 12.8 MB zdrojového kódu oproti 3,7 MB u verze 6.

Kromě funkcionality přidané do jádra, které by bylo možné dosáhnout i za pomoci modulů a předchozí verze, byly přepracovány i mnohé části funkcionality jádra systému [4]. Pro toto řešení jsou relevantní následující vylepšení:

- podpora implementace vyrovnávací paměti za pomoci reverzní proxy (např. Varnish)
- Entity API - unifikace elementů stránek a poskytnutí API ulehčujícího vytváření nových typů obsahu a tříd pro ulehčení práce [10]
- možnost přidávat pole ke všem entitám (uživatelé, slovníky, ...)
- podpora různých databází se stejným zdrojovým kódem v *PHP* (abstrakce nad dotazy do databáze a využití knihovny *PDO*)
- zlepšená podpora *AJAX* volání
- celkově propracovanější kód (vhodný pro vývoj dalších modulů)

V době práce na informačním systému již byl dostupný i *Drupal* verze 8, ale nebyl ještě plně stabilní. Některé jeho vlastnosti však byly zpětně implementovány jako moduly pro *Drupal* 7 a jakožto implementace moderních technologií byly zahrnuty i v tomto projektu (například *NavBar*<sup>1</sup>, který poskytuje responsivní administrační panel, filtrace modulů, či administrační téma *Ember*<sup>2</sup> postavené na *SASS*).

---

1. <https://drupal.org/project/navbar>

2. <https://drupal.org/project/ember>



**Tabulka 3.1:** Porovnání verzí modulů mezi Drupalem 6 a 7

Jméno modulu	Drupal 6	Drupal 7	stav
Backup and Migrate	2.7	2.7	✓
Colorbox	1.6	2.4	✓
CCK	2.9	jádro	✓
Custom Breadcumbs	1.5	1.x-alpha3	
DB Maintenance	1.4	1.1	✓
DHTML Menu	3.5	1.0-beta1	
Email Field	1.4	1.2	✓
File (Field Paths	1.5	1.0-beta4	
FileField	3.11	jádro	✓
Front Page	1.3	2.4	✓
ImageAPI	3.11	jádro	✓
ImageCache	2.0-rc1	jádro	✓
IMCE	2.5	1.7	✓
IMCE Wysiwyg bridge	1.1	1.0	✓
jCarousel	2.6	2.6	✓
Link	2.10	1.1	✓
Localization Update	2.10	1.1	✓
Menu Attributes	2.0-beta1	1.0-rc2	✓
Menu Block	2.4	2.3	✓
Pathauto	1.6	1.2	✓
Taxonomy Breadcrumb	1.1	-	
Token	1.19	1.5	✓
Transliteration	3.1	3.1	✓
Views	2.16	3.7	
Views Search	1.0	-	
WYSIWYG	2.4	2.2	✓

## 3.2 Odebrané moduly

Funkcionalita některých z modulů byla v sedmé verzi přesunuta přímo do jádra Drupalu a proto nebylo jejich využití již potřebné - byly využity jejich ekvivalenty z jádra. Kromě těchto byly odebrány i další moduly, kdy některé nejsou pro novou verzi vůbec dostupné. Dále byly odstraněny moduly, jejichž využití již postrádalo smysl. Z důvodu změny a zjednodušení struktury menu již nebylo potřeba mnoho modulů, které umožňovaly složitá menu předchozí implementace. Bylo odstraněno několik modulů, které nebyly na stránkách vůbec využity a zbytečně zpomalovaly běh portálu.

### **Views Search**

[https://drupal.org/project/views\\_search](https://drupal.org/project/views_search)

Rozšiřuje funkcionalitu Views a jejich obsahových filtrů, pro novou verzi již není modul dostupný a funkcionalita není ani vyžadována.

### **Front Page**

<https://drupal.org/project/front>

Slouží k nastavení výchozí stránky pro různé uživatelské role. Z důvodu plánování kompletně nové implementace domovské stránky byla tato funkcionalita prozatím odstraněna s možností opětovného zapnutí v budoucnosti dle požadavků.

### **Menu Attributes**

[https://drupal.org/project/menu\\_attributes](https://drupal.org/project/menu_attributes)

Umožňuje přiřazení atributů k položkám menu. Ty pak mohou být použity buď pro stylování, nebo navázány na JavaScript. Tato funkcionalita opět nebyla vyžadována.

### **Taxonomy Breadcrumb**

[https://drupal.org/project/taxonomy\\_breadcrumb](https://drupal.org/project/taxonomy_breadcrumb)

Generuje drobečkovou navigaci za pomoci slovníků definovaných v systému a jejich hierarchie. Hierarchie tím pádem může být velmi jednoduše upravena - toto však již není po úpravě systému potřeba s ohledem na zjednodušenou strukturu.

### **Custom Breadcrumbs**

[https://drupal.org/project/custom\\_breadcrumbs](https://drupal.org/project/custom_breadcrumbs)

Umožňuje detailní nastavení drobečkové navigace, která již není potřebná.

### **DHTML Menu**

[https://drupal.org/project/dhtml\\_menu](https://drupal.org/project/dhtml_menu)

Mění funkcionalitu menu - při prvním kliknutí se namísto přesunutí na danou stránku pouze rozbalí menu s potomky daného odkazu a až při druhém kliknutí se přejde na danou stránku - tato funkcionalita zřejmě nebyla využívána.

**jCarousel**<https://drupal.org/project/jcarousel>

Slouží k propojení se stejnojmennou JavaScriptovou knihovnou poskytující zobrazení obrázků v ovladatelném „kolotoči“. S ohledem na aktuální absenci jakékoliv grafiky byl modul shledán nepotřebným.

**Colorbox**<https://drupal.org/project/colorbox>

Podobně jako jCarousel slouží k propojení s knihovnou určenou ke zobrazování obrázků a ani tento modul nebyl aktivně využíván.

### 3.3 Nově přidané moduly

Do projektu byly zahrnuty další moduly, některé z nich rozšiřující funkcionalitu řešení, jiné za účelem zjednodušení vývoje a údržby. Další byly přidány z kvůli kompatibility s propojením se vzdálenou plochou za pomoci nástroje *Guacamole*.

**Context**<https://drupal.org/project/context>

Modul Context umožňuje definovat reakce v závislosti na splněných podmínkách prostředí portálu. Pro každou podmínku, kterou může být například přihlášená role uživatele, aktuální stránka či jazyk, lze nadefinovat změny v portálu. Ať se jedná o nastavení drobečkové navigace, změnu obsahu bloku a podobně. Podmínky lze mezi sebou vzájemně kombinovat a uzavírat do logických výrazů, stejně jako lze pro jeden logický výraz provést více operací. Pro potřeby tohoto projektu je využíván hlavně podmodul Context Layouts a další modul Context Omega<sup>3</sup>, které dohromady umožňují změnu rozložení stránek.

**Kerberos Authentication** [https://drupal.org/project/kerberos\\_authentication](https://drupal.org/project/kerberos_authentication)

Modul využívá protokol Kerberos k autentizaci uživatelů - v tomto případě za pomoci serveru Masarykova Univerzita (MUNI). Uživatelům, kteří ke připojení použijí své UČO, je automaticky vytvořen účet se stejným uživatelským jménem a mohou si poté změnit heslo specificky pro tento portál. Modul dále povoluje vypnutí autentizace pomocí účtů pouze v portálu, tato vlastnost však není pro tento projekt potřebná.

**Features**<https://drupal.org/project/features>

Jak je popsáno v kapitole 6, modul Features slouží k ulehčení vývoje a správy nastavení stránek. Základní myšlenkou je třívrstvé rozvržení systému, v první úrovni je jádro a jeho moduly, ve druhé rozšiřující moduly (tzv. Contrib) a ve třetí jsou rysy (Features), které v sobě zapouzdřují finální nastavení stránek. Ačkoliv se jedná také o moduly, jsou generovány systémem na stránce `admin/structure/features`.

---

3. [https://drupal.org/project/conext\\_omega](https://drupal.org/project/conext_omega)

Zde je možné buď vytvořit nový rys, či spravovat ty stávající. Ty mohou být v několika stavech v závislosti na aktuálním nastavení systému. Pokud bylo v systému něco změněno od chvíle, kdy byl rys vygenerován a povolen (případně pouze povolen, pokud byl přenesen z jiného serveru), je možné dané změny buď vrátit ke stavu definovanému v rysu, nebo rys aktualizovat na novou verzi.

#### **Features Extra**

[https://drupal.org/project/features\\_extra](https://drupal.org/project/features_extra)

Jelikož modul *Features* v sobě implementuje pouze základní funkcionalitu exportu typů obsahu a poskytuje API pro další moduly, některé prvky webu nejsou exportovatelné. Tento modul možnosti exportu rozšiřuje o další možnosti, ze kterých pro tento projekt je relevantní export nastavení umístění bloků.

#### **@font-your-face**

<https://drupal.org/project/fontyourface>

Pro využití řezů písem třetích stran je možné postupovat několika způsoby - buď nahrát soubor obsahující písmo přímo na server, nebo se odkazovat na server třetí strany, který daný font poskytuje (např. Google Fonts API). Modul *@font-your-face* unifikuje tyto přístupy a poskytuje jednoduché rozhraní pro výběr typů a řezů písma a jejich použití na stránkách. Buď lze vše nastavit v administračním rozhraní, nebo lze poskytnutý kód jednoduše vložit do souboru definujícího téma vzhledu - například řádek `fonts[google_fonts_api][] = "BenchNine&subset=latin-ext#300"` automaticky načte z Google Fonts API font se jménem *BenchNine* v řezu 300.

#### **Entity API**

<https://drupal.org/project/entity>

Jak je popsáno v kapitole 3.1, v předchozí verzi *Drupalu* bylo možné vytvářet pouze uzly (node), které s sebou nesou nezanedbatelnou složitost v podobě vlastní URL, revizí a dalších informací. Ve verzi 7 je již možné vytvořit i čistou entitu, u které lze nastavit všechny vlastnosti, například zda ukládá informace o autorovi a podobně. V základu je tato implementace poněkud komplikovaná a není nijak lehce dostupná administrátorům, ani vývojářům a proto byl vyvinut modul *Entity API*, poskytující unifikované rozhraní, umožňující jednoduchou definici entity na bázi zdrojového kódu (viz. [10]).

#### **Entity Creation Kit (ECK)**

<https://drupal.org/project/eck>

Zatímco modul *Entity API* poskytuje možnost definovat entity ve zdrojovém kódu, za pomoci modulu *ECK* je možno je definovat z administračního rozhraní, čímž je celý proces tvorby značně zjednodušen. Definované entity lze exportovat za pomoci modulu *Features* do kódu modulu a tím zabezpečit jejich verzování.

**Entity Reference**[https://drupal.org/project/entity\\_reference](https://drupal.org/project/entity_reference)

Spolu s modulem Inline Entity Form<sup>4</sup> poskytuje modul *Entity Reference* uživatelsky přívětivou možnost ke vzájemnému propojení entit. Každá entita (takže i všechny uzly) může odkazovat na jednu či více entit jiného, či stejného typu, které mohou být pomocí tohoto modulu přímo vytvářeny na stránce úpravy rodičovské entity.

**Media**<https://drupal.org/project/media>

CMS systém *Drupal* v základu poskytuje pouze jednoduchou funkcionalitu práce s obrázky. Zvuk, video ani dokumenty nejsou v aktuální verzi (7.x) podporovány vůbec. Zároveň uživatelé nemají mnoho možností operovat se soubory, které nahráli na server a tím pádem jsou často nuceni obsah opětovně nahrávat, čímž dochází k jeho duplikaci. Tyto a mnohé další problémy se snaží řešit projekt *Media*, poskytující uživatelům srozumitelné rozhraní a unifikaci práce s různými typy multimediálního obsahu a zároveň správu veškerého obsahu nahraného na server. Pro svůj běh vyžaduje modul File Entity<sup>5</sup>, který poskytuje možnosti práce se soubory jako entitami v systému.

**Views Bulk Operations (VBO)**[https://drupal.org/project/views\\_bulk\\_operations](https://drupal.org/project/views_bulk_operations)

Modul *Views* (<https://drupal.org/project/views>) rozšiřuje funkcionalitu o uživatelsky definovatelné pohledy na data (podobně jako views v databázi) a v dnešní době je nainstalován téměř na každé instalaci *Drupalu* (ve verzi 8 je již obsažen v jádru). Tento modul však není sám schopen poskytovat operace nad více řádky najednou, s čímž pomáhá modul *Views Bulk Operations* - ke každému řádku je přiřazeno zaškrtnutí tlačítko a nad formulářem je zobrazen seznam operací, které lze nad elementy provádět.

**Workbench**<https://drupal.org/project/workbench>

Administrace *Drupalu* je mnohdy kritizována pro její nízkou uživatelskou přívětivost a složitost, čemuž se snaží ulehčit tento modul. Přihlášenému uživateli s právem administrace či správy obsahu je umožněno přistoupit k „pracovnímu stolu“, stránkách postavených na modulu *Views* a poskytujících přehled obsahu stránek - ať už se jedná o jakýkoliv nový obsah, nebo o obsah naposledy upravený daným uživatelem. Tato funkcionalita je dále rozšířena, jak je popsáno v kapitole 6.

**Workbench Media**[https://drupal.org/project/workbench\\_media](https://drupal.org/project/workbench_media)

Zatímco modul *Workbench* poskytuje pohled na obsah stránek, *Workbench Media*, jak již název napovídá, přidává propojení s modulem *Media* a tím pádem pohled na mediální obsah daného uživatele a celého portálu.

---

4. [https://drupal.org/project/inline\\_entity\\_form](https://drupal.org/project/inline_entity_form)

5. [https://drupal.org/project/file\\_entity](https://drupal.org/project/file_entity)

### 3.3.1 Typy obsahu

Kdybychom si představili typy obsahu ve světě objektově orientovaného programování, všechny by dědily vlastnosti ze základní třídy *Entita*. Těmito vlastnostmi jsou typ, ID a několik dalších, relevantních pouze pro interní fungování redakčního systému. Z *Entity* dědí třída *Uzel*, která ji rozšiřuje o nadpis, adresu a stav. Pro lepší představu jsou vztahy popsány na diagramu 3.1.

**Uzel** Stránka mající vlastní URL a obsah.

**Položka** Entita s nadpisem použitelná k dalšímu rozšíření.

**Osoba** Entita použitelná pro osobu, obsahující pole pro vložení jména.

**Médium** Multimediální obsah - obraz, zvuk či video uložené na serveru, nebo referencované ze služby třetí strany.

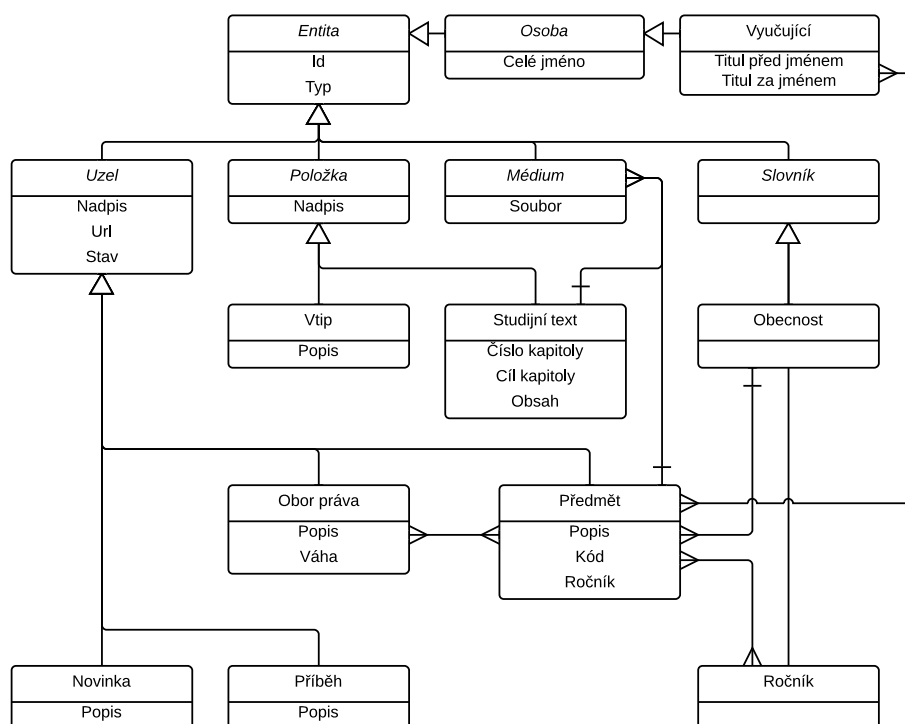
**Slovník** Záznam ve slovníku (česky také označován jako číselník), který obsahuje definovanou množinu záznamů, které se typicky přiřazují jako značky dalším typům obsahu v rámci kategorizace.

**Novinka (*uzel*)** Na stránkách jsou publikovány novinky (aktuality) týkající se buď portálu samotného, fakulty ESF, či jakýchkoliv dalších relevantních událostí. Tyto novinky jsou zobrazeny ve vysouvacím postranním panelu.

**Předmět (*uzel*)** Popisuje předmět vyučovaný v rámci fakulty ESF a tím vyjadřuje základní rozdělení celého portálu a slučuje hlavní část informací, o které se návštěvníci zajímají. Na stránce detailu předmětu je zobrazen krátký popis, za kterým následuje tabulka se seznamem studijních textů a k nim se vztahujících materiálů. Materiály, které se vztahují přímo k předmětu jsou zobrazeny samostatně a celá stránka je uzavřena seznamem oborů které s daným předmětem souvisejí.

**Obor práva (*uzel*)** Zatímco některé předměty mohou souviset pouze s jedním oborem práva, jiné se mohou dotýkat téměř všech všech. Proto je důležité tento vztah vymodelovat a uživatelům poskytnout možnost se v něm jednoduše orientovat a přecházet mezi obsahem, který je právě zajímavý. Na stránce oboru práva je zobrazen popis oboru a tabulka se seznamem předmětů, které se k němu vztahují.

**Studijní text (*uzel*)** Studijní text obsahuje informace vztahující se k jednomu celku výuky, typicky k jedné přednášce. Lze definovat zkrácený cíl přednášky, přidat k ní materiály a určit její pořadové číslo. U všech materiálů je vhodné zadat název pro správné zobrazení v tabulce detailu předmětu, na kterém se zobrazuje seznam těchto entit.



Obrázek 3.1: Diagram tříd popisující vztahy mezi typy obsahu

**Příběh (uzel)** Pro odreagování studentů je v jednom ze segmentů stránek zobrazen náhodný příběh z právního prostředí.

**Vtip (položka)** V postranním panelu lze zobrazit vždy aktuální vtip z právního prostředí, který, podobně jako příběh, slouží k odreagování studentů.

**Vyučující (osoba)** Entita popisující vyučujícího působícího na fakultě. Ke každému vyučujícímu je možné doplnit akademické tituly, které jsou při výstupu správně formátovány.

### 3.3.2 Slovníky

Pro kategorizaci a pro výběr možností bylo v požadavcích identifikováno několik slovníků. Díky vztahu k entitě (ze které záznamy slovníku dědí) může také pro své hodnoty definovat pole a tím rozšířit jejich funkcionalitu a množství informací.

#### Stupeň obecnosti

Pro řazení oborů je potřeba uložit jejich obecnost, která je vyjádřena v několika stupních, které mohou být v případě potřeby přidávány. Každé entitě může být přiřazen jeden stupeň obecnosti.

Pole

- Váha (váha)  
číselná hodnota použitelná k řazení

Hodnoty

- Velmi obecný (váha = 0)
- Středně obecný (váha = 1)
- Málo obecný (váha = 2)

#### Ročník

Pro případné budoucí úpravy systému, či filtrování, je vhodné u předmětu vyplnit i ročník, ve kterém je vyučován. S ohledem na opakování předmětů může každé entitě přiřazeno neomezené množství ročníků (i žádný).

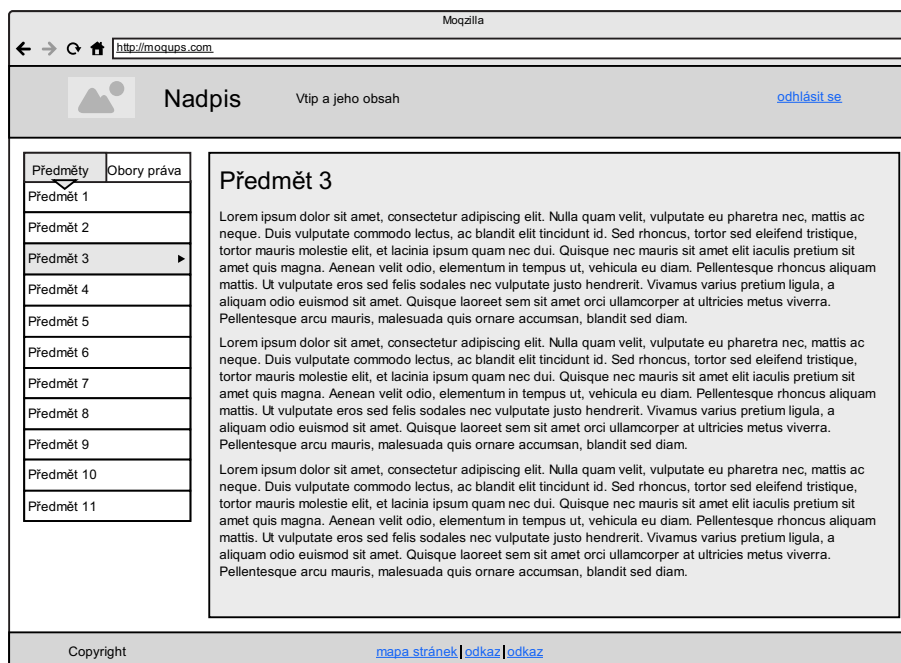
Hodnoty

- 2012/13
- 2013/14
- 2014/15



### 3.4 Téma vzhledu a jeho rozložení

Pro rozvržení stránek bylo využito tématu vzhledu *Omega* ve verzi 4 (kapitola 2.2.1). Díky využití *SUSY* je rozvržení stránky implementována přímo v kódu a v administračním rozhraní jsou určovány až pozice jednotlivých bloků *Drupalu* - každé téma vzhledu definuje množinu dostupných segmentů stránky, do kterých lze vkládat bloky s vlastním obsahem portálu.



Obrázek 3.2: Rozvržení stránky

Jak lze vidět na obrázku 3.2, je stránka rozdělena na čtyři hlavní segmenty - hlavičku, postranní panel, obsah a patičku. Pro možnosti budoucího rozšíření je navíc definován segment Postskript, který se nachází ve spodní části obsahové části stránky. Zatímco hlavička i patička zabírají celou dostupnou šíři stránky (960px), centrální segment je rozdělen mezi postranní panel a obsah - poměr mezi těmito segmenty je určen za použití knihovny *SUSY* a media queries a mění se v závislosti na šíři okna prohlížeče [5].

**Telefon** (velikost okna do 44em)

Postranní panel zabírá celou šířku a za ním následuje obsah.

**Tablet + standart** (velikost okna mezi 44em a 100em)

Postranní panel a obsah jsou zobrazeny v poměru 1 : 3.

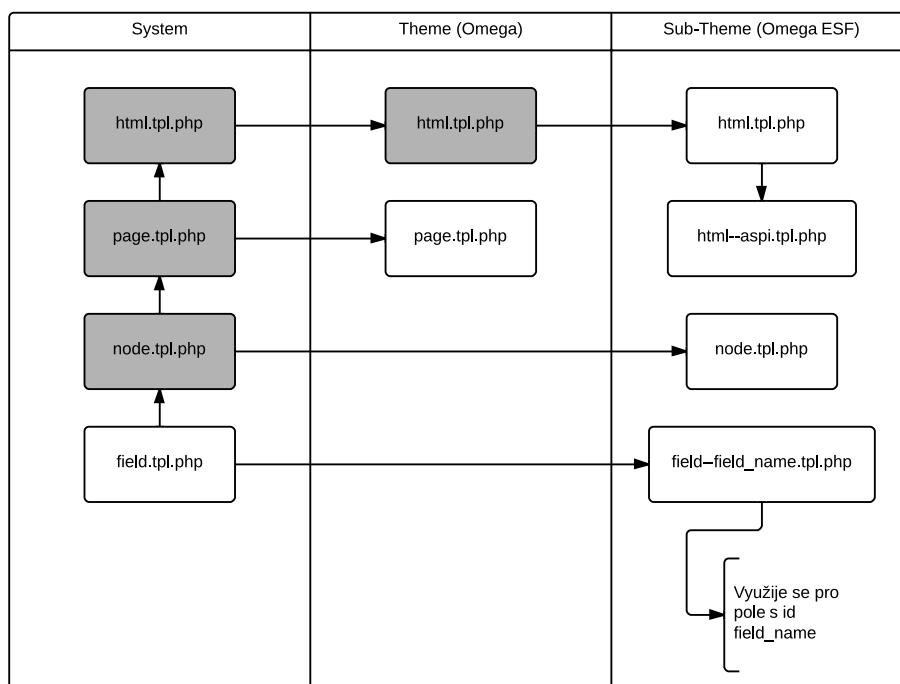
**Širokoúhlý monitor** (velikost okna nad 100em)

Postranní panel a obsah jsou zobrazeny v poměru 1 : 4.

Toto rozložení je vhodné pro využití při klasickém procházení stránkou. S ohledem na využití *Guacamole* a připojení se na vzdálenou pracovní plochu přímo z portálu bylo potřeba zajistit vhodnou strukturu html kódu stránky, který by typicky obsahoval veškerý generovaný kód. Pro oddělení těchto dvou nesourodých prostředí je využito *Layouts* - rozložení stránky za pomoci tématu *Omega*. Každé z těchto rozložení implementuje vlastní šablony (viz. dále) pro generování html kódu a tím umožňuje kompletně změnit výsledný vygenerovaný kód. Při vývoji je potřeba respektovat určitá pravidla, jako je počet a pojmenování segmentů, které musí odpovídat tématu vzhledu, ve kterém budou využita. Přepínání mezi rozloženími stránky se definují za pomoci modulu *Context* a jeho podmodulu *Context Layouts* (sekce 3.3). Zatímco pro všechny stránky portálu je využito rozložení s názvem Blue (podle barvy vzhledu stránek), v případě že se jedná o stránku implementující připojení ke vzdálené ploše (identifikace probíhá na základě URL), systém přepne na rozložení *Guacamole* (dle využití technologie).

Při vlastním vývoji vzhledu portálu byly využity technologie a rozhraní dostupné v CMS *Drupal*, ve kterých probíhá vývoj ve třech úrovních.

1. Je potřeba generovat vhodný HTML kód. K tomu slouží systém přepisování šablonovacích souborů (přípona `.tpl.php`) pro jednotlivé prvky webu. V základu je dostupná stromová struktura, začínající u celého dokumentu (`html.tpl.php`), který obsahuje veškerý obsah včetně hlavičky `<head>`, a pokračující přes stránku (`page.tpl.php`) až po jednotlivé elementy (např. `fieldset.tpl.php`). Většina těchto šablon je definována přímo v jádru systému a všechny lze jednoduchým způsobem (na základě jména souboru) přepsat. Zároveň je možné vytvářet úplně nové šablony, které lze následovně volat při vytváření nových elementů. Šablonám lze i upravit data, která do nich budou vložena a to implementací funkcí s příponami `__preprocess` a `__process` a jejich vložení do kódu tématu vzhledu.
2. Výstup html lze měnit přepsáním funkcí formátujících jednoduché elementy stránky (například seznam elementů `ul`). Typicky jsou tyto prvky generovány funkcí `theme([jméno elementu], [proměnné hodnoty])`, která může být přepsána v modulu nebo tématu vzhledu, jako `[jméno modulu]__[jméno elementu]()` - například `function omega_fieldset($variables)`. Téma vzhledu *Omega* nad tímto rozhraním navíc poskytuje možnosti všechny tyto funkce vkládat do samostatných souborů pojmenovaných dle specifických konvencí, čímž dále zpřehledňuje výsledný kód (kód výše zmíněného příkladu by byl umístěn v souboru `theme/container.theme.inc`).
3. Na HTML kód se nanášejí CSS styly. Jak je popsáno v sekci 2.2.1, je výhodnější využívat nástroj *SASS*, který je do jazyka CSS kompilován a poskytuje více možností. Vývoj probíhá dle standartů určených v tématu vzhledu *Omega*



Obrázek 3.3: Architektura vykreslení HTML kódu stránky

(viz. 2.2.1) a je definován v rámci několika *SASS* souborů, každý z nichž obsahuje určitou skupinu nastavení.

V repozitáři je již dostupná kompilovaná verze *CSS* souborů, ale v případě nutnosti je možné je znovu vygenerovat. K tomu je nutno mít nainstalován programovací jazyk *Ruby*, *COMPASS* (sekce 2.2.1) a *Bundler*. Kompilace se spustí následujícími příkazy:

1. `$ sudo gem install bundler`
2. `$ bundle install`
3. `$ compass watch`

*COMPASS* bude od této chvíle sledovat obsah složky a v případě jakýchkoliv změn automaticky aktualizovat výsledné *CSS* soubory.

### 3.5 Proces přihlášení uživatelů

Každý IS vytvořený v prostředí CMS *Drupal* má automaticky vytvořeného uživatele s ID 1, který má všechna administrační práva. Kromě tohoto uživatele mohou být v

systému vytvoří další uživatelské účty a každý z nich může být přiřazen k určité uživatelské roli (více o rolích v sekci 5.5). Jelikož má být portál dostupný širší skupině uživatelů, jmenovitě studentům MUNI, je vhodné využít pro autentizaci externí službu, v tomto případě se jedná o autentizaci skrze protokol Kerberos a server MUNI.

Uživatelé mohou pro své přihlášení využít své UČO a ihned mají přístup do systému. Je jim zobrazena nápověda, že pro přístup do ASPI je nutné zadat uživatelské jméno a heslo a zároveň jim je nabídnuta adresa URL, na které si o dané údaje mohou požádat.

### 3.6 Struktura zdrojových souborů

- **doc/**  
dokumentace projektu
  - ★ **mvantuch/**  
zdrojové kódy pro generování dokumentace projektu v
  - ★ **omaterna/**  
práce Mgr. Ondřeje Materny určená pro referenci
- **phing/**  
soubory určené k podpoře nasazení pomocí nástroje Phing
- **ref/**  
referenční soubory, design a pod.
- **src/**  
zdrojové kódy jednotlivých částí projektu
  - ★ **main/**  
aplikace Guacamole upravená pro potřeby tohoto projektu
  - ★ **modules/**  
moduly CMS Drupal vyvinuté speciálně pro tento projekt, nebo exportované za pomoci modulu Features (kapitola 3.3)
  - ★ **profiles/**  
instalační profil pro Drupal, definující veřejně dostupné moduly, potřebné k chodu
  - ★ **themes/**  
téma vzhledu projektu
- *build.xml*  
xml soubor obsahující instrukce pro nástroj Phing k operacím nad instalací Drupalu

- *pom.xml*  
xml soubor s projektem Maven, kontrolujícím Guacamole část projektu
- *esf.make*  
make skript pro příkaz Drush make
- *INSTALL.md*  
markdown soubor s instalačními instrukcemi
- *README.md*  
markdown soubor se všeobecnou nápovědou

## 4 Implementace připojení ke vzdálené ploše

Implementace připojení ke vzdálené ploše probíhala v několika fázích. Nejdříve bylo potřeba spustit řešení *Guacamole* lokálně a ujistit se, že je použitelné pro naše potřeby. Bylo také nutné zanalyzovat, jaké možnosti poskytuje a které vlastnosti jsou potřebné a které naopak přebytné (správa uživatelů, zadávání parametrů připojení a jejich skupin). Dále je implementována podpora klávesnici na obrazovce, ukládání snímků obrazovky a mnoho dalších rozšířených možností, které nebyly do základní verze vyžadovány a byly spíše kontraproduktivní s ohledem na zvýšení komplexnosti kódu a odladění celého řešení.

Jakmile byla funkčnost ověřena, bylo nutné celou klientskou část řešení přesunout do modulu pro *Drupal* a upravit pro tamnější podmínky. Bylo nutné změnit *HTML* výstup, kdy *Drupal* si v základu buduje strukturu pomocí několika šablonových souborů pro stránku, tělo a pod (viz. kapitola 3.4). Bylo nutné přidat JavaScriptové knihovny z *Guacamole*, odstranit nepotřebný kód a upravit vše k funkčnosti v prostředí s použitím CORS.

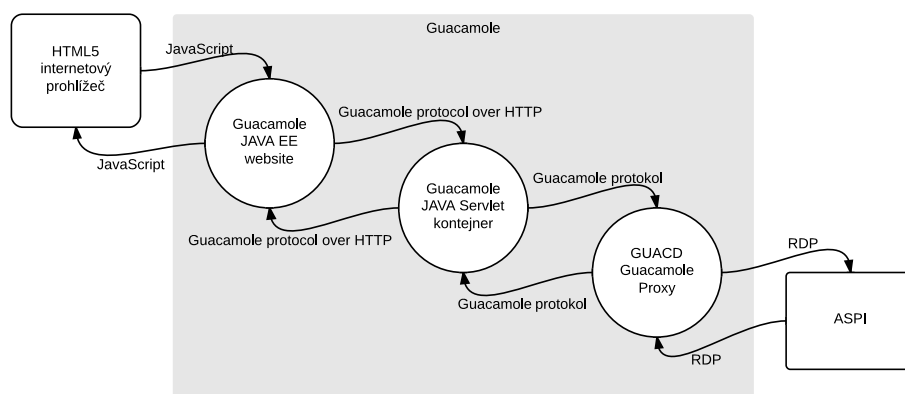
Nástroj *Guacamole* poskytuje mnohá nastavení a pro ulehčení byla do modulu přidána administrační stránka, která přímo zapisuje do konfiguračního souboru a tím eliminuje potřebu uživatelského připojení na server a manuální konfigurace.

### 4.1 Architektura připojení ke vzdálené ploše

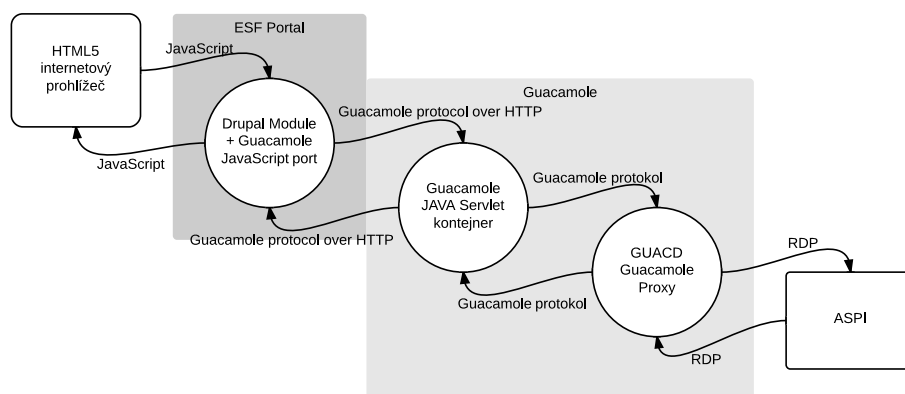
Základně poskytuje *Guacamole* komplexní řešení na propojení ke vzdálené ploše včetně stránky pro management uživatelů, kde každý z nich si může pro sebe vytvořit své vlastní připojení, u kterých se zobrazuje náhled v posledním momentu připojení a podobně. Jak je zobrazeno na diagramu 4.1, součástí *Guacamole* je i webová aplikace v jazyce *HTML*, která implementuje knihovnu *Guacamole-js* a tím i komunikaci s *JAVA Servlet*-em. Pro potřeby tohoto projektu však bylo nutné tyto JavaScriptové knihovny převzít a implementovat do modulu pro *Drupal*.

Díky skutečnosti, že *JAVA Server* a *JAVA EE* Web aplikace nemusí nutně běžet na stejném serveru, jak by tomu bylo při základním nastavení, bylo nutné změnit architekturu *AJAX* požadavků. Ty totiž musejí v typickém případě být zasílány pouze na stejnou doménu kvůli prevenci XSS. Pro vyřešení tohoto problému byla změněna implementace na CORS požadavky, k čemuž musel být uzpůsoben i *JAVA Servlet* kontejner, u kterého byly povoleny požadavky z dané domény.

Uživatel proto pracuje základně pouze s portálem postaveným na *Drupalu*, do kterého byl přidán modul implementující stránku určenou ke komunikaci s *Guacamole Servlet* kontejnerem za pomoci upravené JavaScriptové knihovny *Guacamole-js*, jak je vyobrazeno na diagramu 4.2.



**Obrázek 4.1:** Neupravená architektura a rozčlenění řešení



**Obrázek 4.2:** Architektura přizpůsobená pro portál ESF

## 4.2 Ostranění autentizace uživatelů a vytvoření vlastní autorizace

*Guacamole* poskytuje API pro správu uživatelských účtů a definic připojení. Jsou implementovány dvě metody, jedna spoléhá na uložení dat do konfiguračního souboru, druhá pracuje s *MySQL* databází. Obě rozšiřují třídu *SimpleAuthenticationProvider*, ke které přidávají potřebnou funkcionalitu.

Pro potřeby tohoto projektu je potřeba vždy pouze jediná definice připojení a pro jednotlivé uživatele se mění pouze jejich přihlašovací jméno. Proto byla vytvořena nová třída *DrupalAuthenticationProvider*, která načítá adresu serveru a port přímo z konfiguračního souboru pro všechna připojení a vrací je k dalšímu zpracování.

Pro nastavení přihlašovacích údajů je potřeba se nejdříve k serveru přihlásit. Pro minimalizaci úkonů potřebných ke každému připojení se ke vzdálené ploše a také k zjednodušení celého procesu si musí každý uživatel portálu uložit své přihlašovací údaje do uživatelského profilu na portálu, odkud jsou použity při přihlášení. K tomu je použito dvou serverových služeb - *connect* a *login*. Proces je znázorněn na diagramu 4.3.

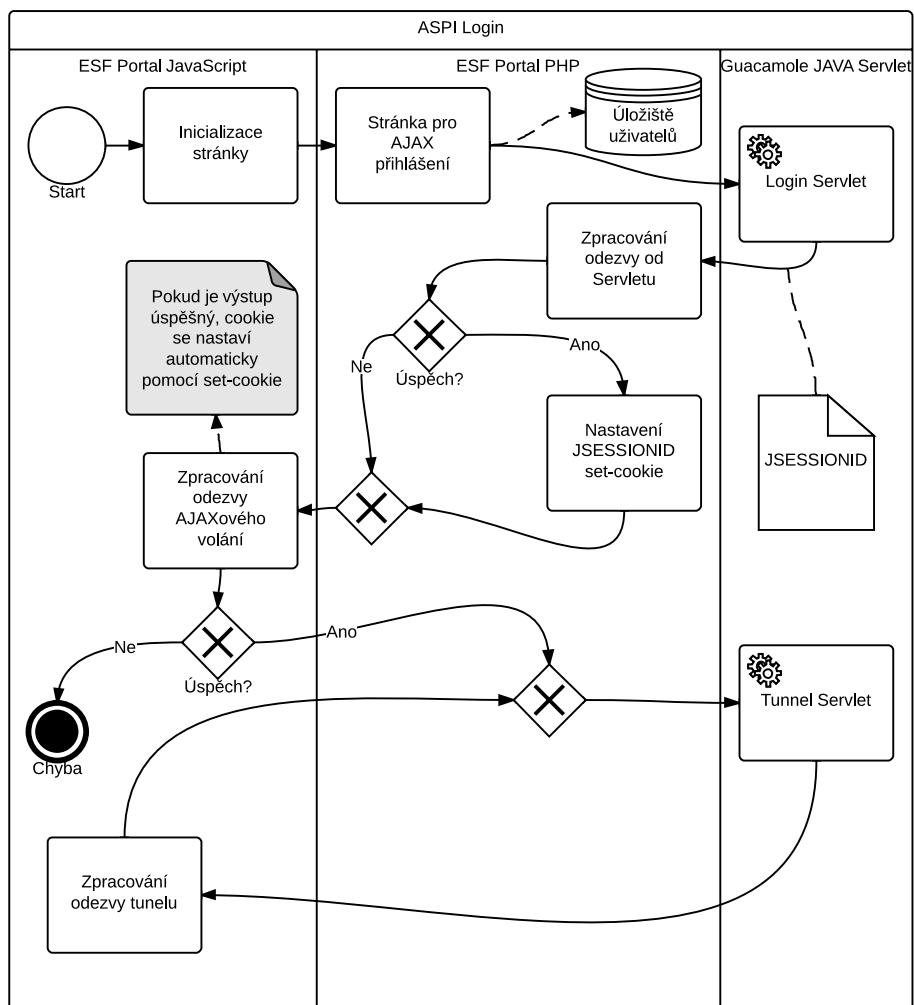
Komunikace je započata přímo v klientském prohlížeči z JavaScriptu. Po přístupu na stránku `/aspi` je nejdříve kontaktována pomocí technolie *AJAX* stránka `/aspi/ajax` na portálu, která se připojí ke službě *Login*. *Guacamole* Servlet si vytvoří nové relace a jeho identifikátor odešle zpět *PHP* kódu. Pokud vše proběhne v pořádku, je identifikátor relace odeslán pomocí `set-cookie` hlavičky zpět stránce `/aspi`, kde si informaci uloží prohlížeč jako cookie pro pozdější komunikaci.

V případě jakékoliv chyby je chybové hlášení vypsáno na obrazovce a zalogováno v systému bez jakýchkoliv dalších pokusů o připojení, uživatel je tedy nucen buď stránku obnovit, nebo kontaktovat správce stránek. Pokud však vše proběhne v pořádku, je inicializována smyčka přímých volání služby *tunnel* poskytované *Guacamole* Servlet kontejnerem za pomoci technologie *CORS*. Komunikace probíhá za pomoci vlastního protokolu, který je konvertován na grafický výstup na obrazovce uživatele. Ten je od této chvíle schopen pomocí myši a klávesnice ovládat vzdálenou plochu a tím i *ASPI*.

## 4.3 Přesun JavaScript kódu a CSS stylů do modulu Drupalu

Pro ulehčení správy relace, které by jinak muselo být synchronizováno mezi dvěma stránkami, byl veškerý kód stránky z *Guacamole* *JAVA* projektu přesunut do modulu `esf_module`. Do složky `js/lib` byly přesunuty všechny zdrojové kódy z knihovny *Guacamole-js* a byla vytvořena nová stránka `/aspi` poskytující stejnou funkcionalitu jako *JAVA* modul. Spolu s JavaScriptovým kódem bylo nutné přesunout i základní CSS soubory pro zachování formátu vykreslování okna vzdálené plochy. Jak knihovny, tak styly jsou importovány pouze na stránce `/aspi` pro snížení náročnosti běhu celé platformy, která by jinak načítala nepotřebné soubory.





Obrázek 4.3: Proces připojení ke vzdálené ploše

## 4.4 Implementace CORS

Pro možnost komunikace mezi více doménami bylo potřeba implementovat CORS (viz. sekce 2.2). Hlavní změna je u zpracování požadavků na straně serveru, k čemuž musela být do projektu přidána a nastavena knihovna k jeho zpracování. Pro *JAVA EE* byla využita knihovna CORS filter od [d]zhuvinov [s]oftware<sup>1</sup>, která tuto implementaci poskytuje ve formě knihovny. Na stránkách je i detailně popsáno nastavení a použití, kdy v případě tohoto serveru bylo nutno omezit zdroj CORS požadavků na jedinou doménu a více nebylo nutno se o bezpečnost obávat.

Druhá část implementace je umožnění cross-domain požadavků v JavaScriptu. Knihovna Guacamole-js musela být lehce přepsána v místech, kde byla použita volání XMLHttpRequest, která jsou v tomto případě volána asynchronně. Použité řešení bohužel nelze aplikovat na IE, neboť XMLHttpRequest nepodporuje asynchronní zpracování a bude nutné toto řešení ještě přepracovat do funkční podoby (změna není součástí této práce).

## 4.5 Implementace úpravy nastavení z administrace portálu

Guacamole typicky poskytuje nastavení pomocí konfiguračního souboru umístěného ve složce instalace, nebo v domácí složce uživatele spouštějícího program. Jelikož je ke změně těchto konfiguračních souborů nutné přistupovat přímo k serveru pomocí SSH, byla administrační část přepracována a zpřístupněna přímo z administrace portálu. Nastavení je rozdělené na dvě stránky:

### ESF Nastavení

*/config/esf/settings*

Zde je možné upravit základní nastavení projektu jako je URL adresa pro připojení ke *Guacamole* Servlet službám či přístup ke konfiguračnímu souboru. Ten se typicky jmenuje `guacamole.properties` a nachází se na serveru ve složce `/srv/guacamole/`. Je nutné se ujistit, že jak *Guacamole* Servlet (*Apache TomCat*) tak Ekonomicko Správní Fakulta (ESF) Portál (*Apache*) mají k souboru přístup a mohou jej upravovat.

### Nastavení vzdálené plochy

*/config/esf/remote*

Na této stránce lze za předpokladu, že lze přistupovat ke konfiguračnímu souboru, upravovat nastavení *Guacamole* a URL pro připojení k serveru vzdálené plochy - což je v našem případě ASPI. Dále je nutné nastavit port pro připojení ke *Guacd* démonu. Uživatelské jméno a heslo je převzato z nastavení každého jednotlivého uživatele.

1. <http://software.dzhuvinov.com/cors-filter.html>

## 5 Instalace, konfigurace a správa portálu

Díky využití Features (sekce 3.3) a instalačních profilů (sekce 2.2.1) je možné portál nainstalovat na jakýkoliv server bez kopírování souborů či databáze, pouze za pomoci nástrojů *Phing* a *Drush*. Je vytvořena kompletní struktura obsahu, nastavení odpovídající vzhled a vytvořeny uživatelské role. Obsah však přenášen není a je nutné jej buď importovat, nebo vytvořit znovu. Tento přístup se nejvíce hodí v případě vývoje nových prvků, či správy funkcionality připojení ke vzdálené ploše.

### 5.1 Požadavky a příprava systému

Na stroji je nutné mít nainstalovaný internetový server (*Apache*, *Lighttpd*, etc.)<sup>1</sup> s podporou *PHP* a *JAVA* Servlet kontejner *Apache Tomcat*. Systém podporuje *MySQL* a *PostgreSQL*, pro ulehčení instalace lze však využít i minimalisticky orientovanou implementaci *SQLite*, která data ukládá do jediného souboru ve složce `sites/default/files`. *Drupal* dále ke svému chodu vyžaduje knihovny `php php-gd`, `php-pdo` a `php-krb5`<sup>2</sup>. Pro usnadnění instalace je nutné mít nainstalovanou knihovnu *Phing*. Pro zkompileování systémového démona je potřeba mít nainstalovány nástroje *gcc*, *make*, *Maven* a *JRE* a knihovny *libpng-devel*, *cairo-devel* a *freerdp-devel*.

Volitelně je možné mít nainstalován *Drush* a s ním spojená nastavení prostředí systému, určená k urychlení manuálních operací. Ještě donedávna se *Drush* instaloval jako knihovna pro *PEAR*, tento způsob však byl nahrazen instalací pomocí programu *Composer*, který je tím pádem potřeba před instalací. Nejjednodušší způsob, jak jej nainstalovat, je pomocí *CURL*:

```
$ curl -sS https://getcomposer.org/installer | php
```

Jakmile je *composer* dostupný, lze již *Drush* nainstalovat v jeho 6. verzi:

```
$ composer global require drush/drush:6.*
```

Po instalaci je vhodné nastavit prostředí systému a integraci mezi nástroji *Drush* a *BASH*. Do souboru `.bashrc` v domovském adresáři vložte řádek obsahující cestu k *Drushi* a kód s nahráním souboru `.drush_bashrc`.

```
export PATH="$HOME/.composer/vendor/bin:$PATH"
```

```
if [ -f ~/.drush_bashrc ] ; then
. ~/.drush_bashrc
fi
```

Je vhodné aktualizovat `bashrc` soubor:

---

1. Technologie jsou popsány v sekci 2.2.

2. [https://github.com/ashishtilara/php\\_krb5](https://github.com/ashishtilara/php_krb5)

```
$ source ~/.composer/vendor/drush/drush/examples/  
example.bashrc
```

Nyní by měly být přístupné příkazy pod zkrácenými aliasy - například `dr` místo `drush` a `cc` namísto `drush cache-clear`. Je možné i povolit automatické doplňování příkazů, dle instrukcí na stránkách vývojového týmu. Dále je vhodné v souboru `$HOME/.drush/aliases.drushrc.php` definovat alias stránky pro usnadnění manipulace. Obsah souboru může vypadat následovně:

```
<?php  
$aliases['pravo'] = array(  
  'root' => '/srv/www/htdocs/',  
  'uri' => 'pravo.com',  
);
```

Po tomto nastavení je možné do složky přistoupit jednoduše pomocí příkazu `$ cddl @pravo` (případně `$ cd @pravo` pokud je nastavený alias pro příkaz `cd`). Všechny příkazy nástroje *Drush* je možné spouštět pro daný alias, což je výhodné v případě více prostředí (vývojové vs. produkční).

Obsah repozitáře projektu může být umístěn kdekoliv, například v domovské složce uživatele:

```
$ cd ~  
$ git clone git@github.com:kanei/esf-mu-portal.git
```

Před samotnou instalací je nutné nastavit základní vlastnosti projektu. Soubor `phing/default.build.properties` překopírujte do hlavní složky repozitáře a přejmenujte jej na `build.properties`:

```
$ cp ~/.esf-mu-portal/phing/default.build.properties  
~/esf-mu-portal/build.properties
```

Druhou možností je spuštění úvodní inicializace, jejíž součástí je i překopírování souboru v případě, že neexistuje:

```
$ phing void
```

V konfiguračním souboru je nutné v proměnné `project.dir` nastavit cílovou složku pro instalaci. Složka však nesmí existovat a uživatel spouštějící skript musí mít práva k jejímu vytvoření, protože jinak *Drush* není schopen s instalací pokračovat a skončí s chybovou hláškou. V konfiguračním souboru lze změnit další nastavení, jakými jsou administrátorský účet, url databáze a pod, ale pro základní funkcionalitu je možné tyto hodnoty ponechat nezměněné.

## 5.2 Nasazení (deployment) a jeho možnosti

Pro správu nasazení je využíván nástroj *Phing*, umožňující definování pravidel v souboru `build.xml`. Kromě tohoto souboru je také dostupný soubor

`phing/default.build.properties` obsahující výchozí nastavení projektu. Při první inicializaci jakéhokoliv cíle (`<target>`) se zkontroluje přítomnost souboru `build.properties` a v případě jeho absence se na jeho místo zkopíruje výše zmíněný výchozí soubor. Je také možné spustit příkaz `phing void` který pouze připraví tento konfigurační soubor.

V konfiguračním souboru je vhodné zadat nastavení prostředí, na kterém se bude portál instalovat (cesty k souborům, nastavení databáze a pod.) - více informací je v samotném souboru.

**void**

Pouze provede inicializaci konfiguračních souborů a ukončí se.

**guacamole** Vytvoří konfigurační soubory projektu Guacamole ve správných složkách (je zkopírován soubor `phing/default.guacamole.properties`).

**deploy** Překopíruje aktuální moduly na správné místo a zajistí správné nastavení portálu dle aktuálního nastavení (převážně využití Features).

**install** Provede kompletní instalaci portálu dle nastavení v konfiguračních souborech - překopíruje moduly a témata specifická pro toto řešení, stáhne potřebné zdrojové kódy Drupalu z internetu a nainstaluje Drupal, kterému následně nastaví správné téma vzhledu a povolí potřebné moduly.

### 5.3 Instalace informačního systému (Drupal)

Instalace probíhá za využití dávkového instalačního souboru `build.xml`, ve kterém je krom jiného popsán instalační cíl *install* (viz předchozí kapitola). Pro spuštění instalace je potřeba v terminálu spustit následující příkaz:

```
$ phing install
```

**Průběh instalace**

1. Je zkontrolována cílová složka a uživatel je případně upozorněn na nutnost jejího smazání.
2. `$ drush make` - Vytvoří se struktura stránek, při čemž se ze souboru `esf.make` (který načte podsoubor `src/profiles/esf_profile/esf_profile.make`) načte seznam všech modulů využívaných v řešení spolu s jejich požadovanou verzí. Tyto se stáhnou do cílové složky, ve které se zároveň připraví konfigurační soubory pro instalaci Drupalu.
3. Do projektového adresáře jsou překopírovány moduly, témata a instalační profily projektu, aby s nimi mohl instalační skript pracovat.

4. Při instalaci nejsou automaticky provedeny všechny úkony a některá nastavení je nutné spustit manuálně po dokončení. Jedná se mimo jiné o nastavení tématu vzhledu.
5. `$ drush site --install` - Spustí se instalace Drupalu z příkazové řádky, při které se povolí základní moduly a vytvoří databázová struktura. Je nainstalován instalační profil *esf\_profile* a s ním jsou povoleny moduly rozšiřující funkcionalitu a také rysy dokončující nastavení stránek. Celý proces trvá až několik minut a v průběhu nijak uživatele neinformuje o průběhu, je tedy důležité vyčkat až do jeho ukončení.

Po instalaci je vhodné zkontrolovat práva souborů a nastavit uživatele i skupinu na hodnoty daného internetového serveru (`apache/lighttpd/...`)

```
$ sudo chown apache esf -R
$ sudo chgrp apache esf -R
```

Toto je obzvláště důležité v případě použití *SQLite* databáze, ke které jinak nemá drupal práva zapisovat a tím pádem stránky spadnou s fatální chybou.

Pokud vše proběhlo v pořádku a server je schopen přistupovat ke všem souborům i databázi, je možné na adrese propojené se stránkami navštívit úvodní stránku (např. <http://localhost/esf>). Na stránky bylo převedeno veškeré nastavení, ale nebyl zde vytvořen žádný obsah. Ten lze buď migrovat z produkčních stránek, nebo generovat za pomoci modulu *Devel*. Na adrese <http://localhost/esf/?q=user> se lze přihlásit za pomoci administrátorského účtu (pokud nebyl změněn v konfiguračním souboru, jedná se o účet admin a heslo admin) a přistoupit k administračnímu rozhraní stránek.

## 5.4 Instalace Guacamole

Guacamole vyžaduje na straně serveru dvě komponenty: *JAVA* Servlet aplikaci a démona *Guacd*. Druhý jmenovaný není nijak upraven a lze tedy stáhnout aktuální verzi ze stránek projektu<sup>3</sup>. Po stažení stačí zkompilovat a nainstalovat projekt (za předpokladu, že jsou v systému dostupné všechny knihovny a nástroje zmíněné v kapitole 5.1):

```
$ cd guacamole-server-0.9.0
$ ./configure
$ make
$ make install
$ ldconfig
```

V této chvíli by již měl jít spustit příkaz *guacd*, který spustí na portu 4822 službu pro připojení ke vzdálené ploše. Nyní je potřeba nastavit parametry připojení.

3. V době sepsání tohoto textu byla aktuální verze 0.9, na které byl i celý projekt testován. Projekt je dostupný na adrese <http://guac-dev.org/release/release-notes-0-9-0>

Na stránce portálu, na adrese `admin/config/esf/settings` je nutno nastavit správné hodnoty a zajistit přístup ke konfiguračnímu souboru Guacamole pro *Drupal* i *Guacamole JAVA* aplikaci. Dále budou předpokládány výchozí hodnoty nastavení.

```
$ cp ~/esf-mu-portal/phing/default.guacamole.properties /srv/guacamole/guacamole.properties
$ chown apache guacamole.properties
$ chgrp apache guacamole.properties
```

Jméno a uživatelská skupina závisí na použitém operačním systému. V souboru `guacamole.properties` je nutno nastavit parametry připojení ke vzdálené ploše ASPI a informace o démonu `Guacd`, což lze udělat buď přímo v souboru, nebo za pomoci *Drupalu* na adrese `admin/config/esf/remote`.

Tím je dokončeno nastavení démona a lze přikročit k instalaci *JAVA Servlet*. Ten je potřeba zkompilovat a nahrát jako do *Apache Tomcat*.

```
$ cd ~/esf-mu-portal/
$ mvn package
$ cp ~/esf-mu-portal/target/guacamole-esf-0.4.war /srv/tomcat/webapps/
$ rctomcat restart
```

Po dokončení kompilace by měla být aplikace automaticky dostupná na adrese `localhost:8080/guacamole-esf-[číslo verze]/`. Tuto adresu je nutné zadat do políčka *Guacamole Servlet path* v portálu na adrese `admin/config/esf/settings`. Poté by se již uživatelé měli být schopni přihlásit ke vzdálené ploše na adrese `/aspi`.

## 5.5 Administrace

Správa probíhá za využití administrace dodávané v jádře *Drupalu*, rozšířené modulem *Workbench* a dalšími úpravami. Pro správu portálu jsou definovány čtyři úrovně uživatelských práv:

- **anonymní uživatel**  
Tato role je spolu s rolí *přihlášeného uživatele* definována v jádře CMS a nelze tedy měnit. Označuje uživatele, kteří se nepřihlásili do portálu. S ohledem na nutnost autentikace jsou tito uživatelé odstríženi od přístupu k jakýmkoliv datům a je jim zpřístupněna pouze stránka s přihlášením.
- **přihlášený uživatel**  
Po přihlášení uživatelé získávají přístup k veškerému hlavnímu obsahu stránek, uživatelskému nastavení a připojení ke vzdálené ploše.
- **moderátor**  
Moderátor má přístup k základní administraci stránek, obzvláště k úpravám obsahu, základním změnám provázání obsahu a administraci přístupu uživatelů.

- **administrátor**

Pro tohoto uživatele neexistuje speciální role, ale jedná se o uživatele s identifikačním číslem 1, který má právo přístupu ke všem prvkům obsahu a administrace portálu a tato práva nejdou nijak omezit.

Tato práce se zabývá pouze specifickou administrací a nezaobírá se administrací Drupalu, ve které nebyly provedeny změny výraznějšího charakteru a funguje dle standartního přístupu. Po přihlášení má uživatel s rolí moderátora dostupný administrační panel umístitelný buď v horní, nebo levé části okna prohlížeče (obrázek 5.1). Administrační panel obsahuje odkazy na administrační stránky a pracovní stůl.

### **Struktura pracovního stolu**

#### **Můj obsah**

Tato sekce nabízí pohled na stránky vytvořené či upravené právě přihlášeným uživatelem a na nejaktuálnější obsah portálu všeobecně. Všechny pohledy vedou po rozkliknutí na detailnější přehledy obsahu dané sekce.

#### **Vytvořit obsah**

V této sekci je možné vytvářet veškerý obsah, ke kterému má uživatel přístup - buď mediální obsah (obrázky, videa, dokumenty), ke kterému lze později přistupovat skrze WYSIWYG editor, nebo vlastní obsah portálu - nové předměty, stránky, obory práva viz. sekce 3.3.1.

#### **Seznam souborů**

Poskytuje náhled na všechny soubory nahrané na portál, které jsou sledovány systémem a dostupné skrze administrační rozhraní.

#### **Obory práva**

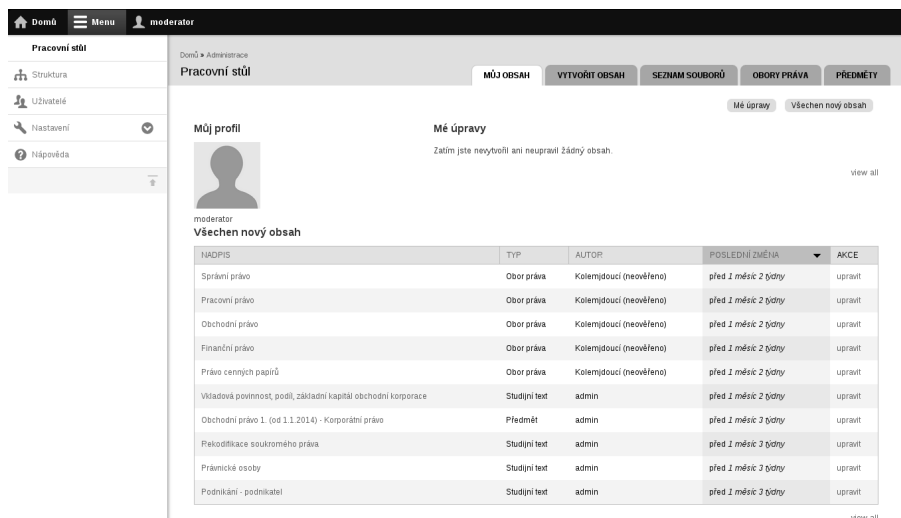
Pohled na všechny zadané obory práva, jejich základní atributy a předměty, které se k nim vztahují. S obory práva lze manipulovat pomocí rozhraní využívajícího modulu Views Bulk Operations (sekce 3.3) a tím pádem hromadnou úpravu entit.

#### **Předměty**

Podobně jako předchozí sekce poskytuje pohled na předměty, pouze s tím rozdílem, že neukazuje jejich propojení na obory práva, ale materiály, které jsou k nim přidružené.

Veškerý obsah stránek je pokryt verzováním a je možné ukládat tzv. revize obsahu, mezi kterými pak lze zobrazit výčet změn od předchozí verze. K editaci slouží WYSIWYG editor *CKeditor*, poskytující uživatelům přívětivé rozhraní s rozšířenými možnostmi úpravy textu a prvky specificky vytvořenými pro použití s CMS Drupal. Pro prohlížení souborů uložených na serveru ve specifické složce se využívá modulu *IMCE*.





Obrázek 5.1: Prostředí pracovního stolu pro uživatele s právy moderátora.

## 5.6 Výkonové optimalizace

Drupal je všeobecně znám pro svou vysokou náročnost na výpočetní výkon serveru, což je způsobeno jeho rozšiřitelností. Z tohoto důvodu je nutné používat pokročilé techniky ukládání a načítání z vyrovnávací paměti[7]. Od verze 7 je v jádru implementováno hned několik přístupů ke zvýšení výkonu:

### Registry

Protože mnoho částí systému umožňuje přepisování výstupu za pomoci jmenových konvencí, jsou soubory prohledávány pouze jednou a informace o dostupných funkcích jsou uloženy v databázi. Kvůli tomuto přístupu je nutné přebudovat registry pokaždé, když je přidán nový soubor, například typu `.tpl.php`. K této operaci slouží modul *Registry Rebuild*<sup>4</sup>. Inicializace registrů nejde vypnout a pouze v nastavení některých témat vzhledu a za pomoci modulu *Devel* lze nastavit, aby se přebudovaly při každém načtení stránky. S tím však značně klesá výkon.

### Vyrovňovací paměť

Za pomoci Cache API je možné využívat vyrovnávací paměti typicky uložené v databázi. V jádru je definováno několik tabulek pro ukládání a moduly mohou jednoduše definovat další (například modul *Views* a tabulka `cache_views`). Data z této vyrovnávací paměti jsou načítána pouze v případě, kdy je povolena nastavení výkonu systému. V konfiguračním souboru je možné definovat

4. [https://drupal.org/project/registry\\_rebuild](https://drupal.org/project/registry_rebuild)

umístění jednotlivých vyrovnávacích pamětí - další umístění mohou být přidávána rozšiřujícími moduly (například Memcache, APC, ...).

### **Agregace**

Při načítání obsahu stránek je pro každý soubor vytvořen nový požadavek, který je zaslán serveru. Pro snížení náročnosti komunikace je vhodné snížit počet souborů, čehož je typicky dosáhnuto agregací JavaScriptových a CSS souborů (v případě, že je obsah poskytován z více domén, je vhodné ponechat alespoň několik výsledných souborů, které mohou být stahovány paralelně)[9]. Základní funkcionality je rozšířena modulem *Advanced Aggregation*<sup>5</sup>.

### **Statická vyrovnávací paměť**

Nepřihlášeným uživatelům, pro které stránky vypadají vždy stejně, je možné posílat výsledný *HTML* kód z vyrovnávací paměti. Stránka však nemůže obsahovat žádný dynamický obsah. Tento problém, zároveň se servírováním stránek přihlášeným uživatelům, řeší modul *Authenticated Cache*<sup>6</sup>, který namísto dynamicky generované části stránky vytvoří pouze kontejner, do kterého je výsledný obsah přidán za pomoci *AJAX* volání. Uživatelé je stránka zobrazena prakticky okamžitě a dynamické prvky se objeví řádově rychleji, neboť pro tento způsob volání je využit okleštěný inicializační skript *Drupalu*.

---

5. <https://drupal.org/project/advagg>

6. <https://drupal.org/project/authcache>

## 6 Organizace vývojového procesu

Vývoj řešení probíhal na lokálně běžícím serveru. Pro minimalizaci nároků na výkon zařízení byl namísto typicky využívaného *HTTP* serveru *Apache* nainstalován *Lighttpd* poskytující dostačující funkcionalitu při mnohem nižších požadavcích na výkon. Pro databázi byl namísto *MySQL* použit *SQLite*, který nepotřebuje k běhu instalaci, ale celá implementace běží pouze nad jedním souborem uloženém na disku – v případě *Drupalu* ve složce s daty dané stránky.

Pro dostupnost komplexního testování i bez využití internetu byl lokálně nainstalována služba *Guacamole Proxy* (*Guacd*) a místo připojení ke vzdálenému zařízení byl virtuálně spuštěn systém *Windows Server 2008 SP2*, na kterém byla povolena vzdálená plocha pomocí protokolu *RDP*. Pro *JAVA* aplikaci byl použit server *Apache Tomcat*, stejně jako na produkčním serveru.

Protože *Drupal* je postaven z velké části na konfiguraci uložené v databázi, bylo potřeba vymyslet způsob, jak změny prováděné na lokálním stroji efektivně přenášet do produkčního prostředí. Pro uložení nastavení do konfiguračních souborů byl použit projekt *Features* (sekce 3.3), který exportuje pomocí funkcionality poskytnuté modulem *CTools*<sup>1</sup> nastavení (typy obsahu a pod.) a také pomocí modulu *Strongarm*<sup>2</sup> nastavení systému samotného. Všechna tato nastavení jsou pak uložena do modulu a mohou pak být jednoduše přenesena na jiné prostředí a aktualizována v průběhu času a s pokračujícím vývojem. Pro udržení přehlednosti byla konfigurace rozdělena na tři části:

### ESF Feature (esf\_feature)

základní nastavení systému včetně typů obsahů, vztahů mezi nimi, metody zadávání obsahu (*WYSIWYG*) a základní prvky zobrazené na stránkách

### ESF Feature UI (esf\_feature\_ui)

administrační rozhraní pro správu obsahu a nádstavba nad modulem *Workbench* (viz. sekce 3.3), poskytující náhled na jednotlivé typy obsahu stránek

1. <https://drupal.org/project/ctools>

2. <https://drupal.org/project/strongarm>

**Tabulka 6.1:** Porovnání technologií použitých na lokálním a produkčním prostředí

	Vývojové prostředí	Produkční prostředí
HTTP Server	Lighttpd	Apache
SQL databáze	SQLite	MySQL
JAVA Servlet Container	Apache Tomcat 7	Apache Tomcat 7
Vzdálená plocha	Windows Server 2008 SP2 (Oracle VM VirtualBox)	ASPI

**ESF Permissions (esf\_permissions)**

definice uživatelských rolí a jejich práv

Moduly jsou uloženy v *GIT* repozitáři a tím je zajištěno jejich verzování. V případě potřeby je možné je doručit do produkčního prostředí a nastavení systému aktualizovat. V případě bezproblémové aktualizace (v komponentech popsaných v daném modulu nebyly manuálně provedeny žádné změny) se změny provedou automaticky – jinak musí být přes rozhraní či drush manuálně určeno, která změna se má využít a zda se případně nemá přepsat daný modul.

Instalační skript a profil jsou generovány automaticky pomocí modulu *Profile Builder*<sup>3</sup>. Jeho výstupem je dávkový soubor `.make` a instalační profil. V dávkovém souboru je vypsán seznam modulů a jejich verzí, určený pro příkaz `$ drush make`, který stáhne potřebné projekty z repozitáře *Drupalu*. Po vytvoření je nutné ručně odstranit moduly `esf_*`, které *Drush* neumí automaticky stáhnout a instalace by selhala. Díky rozdělení na hlavní a doplňkový instalační skript je možné definovat url adresy ke knihovnám, které nelze automaticky doplnit a ty zároveň nejsou přepsány opětovným vygenerováním - hlavní instalační soubor je přesunut do kořenové složky a přejmenován na `esf.make`. V instalačním profilu `esf_profile` jsou obsažena základní nastavení portálu a seznam modulů, které je nutné povolit ke správné funkčnosti stránek. Ke každému modulu lze přiřadit i opravné balíčky (patch), které jsou buď automaticky dohledány a jako odkazy přidány do profilu, nebo mohou být přidány dodatečně ručně. Při instalaci jsou automaticky aplikovány na kód. Díky propojení s features se na stránky dostane i definované nastavení a struktura.

**6.1 Organizace řízení projektu**

S ohledem na nízký počet zainteresovaných osob a rozsah projektu nebyly implementovány pokročilé metody projektového řízení a byl využit jednoduchý seznam úkolů. Zdrojový kód projektu byl uložen na serveru *GitHub* a jeho funkcionalita byla použita i k tomuto účelu. Ačkoliv se nemůže rovnat s platformami specializujícími se na správu úkolů, poskytuje *GitHub* několik základních prvků - problém (issue), milník (milestone) a značku (tag). Značky lze využít pro rozlišení mezi úkolem a chybou a také důležitosti. Milníky byly využity pro jednoduché plánování a sledování pokroku.

**Označení úkolů**

Typ

**Úkol (*task*)**

úkol, který bylo třeba vykonat na projektu

**Chyba (*bug*)**

chyba nalezená na projektu, kterou bylo potřeba opravit

3. [https://drupal.org/project/profile\\_builder](https://drupal.org/project/profile_builder)

Priorita

**Nízká (0-low)**

není aktuálně důležitá a nijak nebrání správnému běhu portálu (převážně kosmetické problémy)

**Střední (1-medium)**

může bránit v určitých scénářích, existuje však způsob, jak ji obejít

**Vysoká (2-high)**

brání scénářům a ovlivňuje více než jednoho uživatele

**Kritická (3-critical)**

brání v užívání celého portálu, nebo ovlivňuje základní funkcionalitu

Milníky

**0.1 | Inicializace**

úvodní výzkum týkající se připojení ke vzdálené ploše a dostupných technologií

**0.2 | Drupal modul**

vytvoření modulu pro drupal a jeho základní funkcionalita

**0.3 | Struktura a práva**

struktura stránek, jejich obsahu a práva uživatelů k jejich použití

**0.4 | Guacamole Drupal**

přesun funkcionality z Guacamole do Drupal modulu a jeho propojení s JAVA serverovou aplikací

**0.5 | Test v praxi**

změny potřebné k umístění řešení na produkční servery a vytvoření skriptů k automatizaci tohoto procesu

**1.0 | Základní verze**

spuštění základní funkční verze

**1.1 | Produkční verze**

vyřešení všech problémů, komunikace se stranou klienta a přípravy na reálné spuštění v produkčním prostředí

**1.2 | Údržba**

první z verzí, ve kterých se bude dodávat údržba řešení

## 7 Závěr

Projekt implementace informačního portálu pro fakultu ESF byl vyvíjen za využití moderních technologií. Ačkoliv je většina nastavení CMS *Drupal* ukládána v databázi a v typických případech je jediným způsobem replikace prostředí jeho zkopírování, byly využity moduly *Features* a *Profile Builder* k vytvoření udržitelného procesu vývoje, který lze využít i do budoucna. Vývoj probíhal na odděleném prostředí (lokálně běžící server) a veškeré změny byly prováděny buď v modulu specifickém pro tento projekt (*ESF Module*), nebo byly exportovány za pomoci *Features* (*ESF Feature*). Vývoj probíhal za maximálního respektování standartů, včetně stylu kódu a komentářů.

Všechny využití moduly třetích stran byly průběžně zaznamenávány v instalačním profilu (*ESF Profile*) a díky této kombinaci je možné nainstalovat v případě nutnosti celý projekt zcela od nuly bez ovlivnění historickými chybami, které mohou vznikat například špatnými daty v databázi. Pro zjednodušení procesu aktualizace a instalace byly sepsány instalační skripty za pomoci nástrojů *Phing* a *Maven*, které byly zároveň propojeny s CI serverem *Team City*, umožňujícím průběžnou kontrolu kvality a funkčnosti kódu.

Portál byl vyvíjen s ohledem na moderní trendy v interaktivních webových aplikacích, včetně optimalizace pro dotyková zařízení a přístroje s nižšími zobrazovacími schopnostmi. Tyto požadavky byly zohledněny i při implementaci administrace, která byla zároveň vytvořena s co největším ohledem na uživatelskou přívětivost a přímocí. Do systému je možné vkládat většinu typů multimediálního obsahu s možností rozšíření do budoucna dle možných měnících se požadavků uživatelů systému (je možné například připojovat videa z *YouTube* a podobně).

Struktura portálu byla vyvinuta dle požadavků vycházejících z práce Mgr. Ondřeje Materny, které byly průběžně konzultovány s JUDr. Jindřiškou Šedovou, CSc. a upravovány dle potřeby. Výsledkem je návrh reflektující strukturu výuky a poskytující co možná nejkonzistentnější pohled na informace k předmětům a oborům práva, které tvoří základní stavební kameny systému. Na úvodní straně byl vytvořen diagram, který dále zviditelňuje propojení mezi předměty a obory práva, čímž uživatelům poskytuje jednoduchou možnost procházet pouze relevantní informace. Propojení jsou zobrazena i na stránkách obou výše zmíněných entit a umožňují rychlou navigaci mezi souvisejícími tématy.

Do portálu bylo implementováno propojení s aplikací *Guacamole* a jejím systémovým démonem *Guacd*, které dohromady umožňují připojení se ke vzdálené ploše skrze internetový prohlížeč podporující technologie *HTML5* a *CSS3*. Díky této možnosti se prudce zjednodušuje interakce uživatelů s aplikací. Uživatelé se přihlašují za pomoci údajů synchronizovaných s IS MUNI a jsou upozorněni na nutnost vyplnění přístupových údajů, potřebných k připojení ke vzdálené ploše. Případně jim je nabídnuta možnost přesměrování na stránku, která jim umožní vygenerování hesla nového. Po přihlášení již nejsou žádána žádná další hesla. Tím se uživatelská zkušenost značně vylepšuje a odpadá nutnost instalace klientů pro

připojení ke vzdálené ploše a opakujícího se zadávání hesla, která mohla být jinak pro velkou skupinu méně technicky zdatných uživatelů značně frustrující. Pro uživatele preferující připojení skrze dedikovaného klienta je tato možnost zachována.

V době vypracování této práce již portál běžel v testovacím provozu a byly opravovány poslední chyby před finálním spuštěním a zpřístupněním studentům ESF Masarykovy univerzity. Řešení propojení mezi CMS *Drupal* a nástrojem *Guacamole* je dosud nevyužívanou cestou a díky skutečnosti, že je projekt veřejně dostupný na portálu *GitHub*, byl již zdrojový kód použit i mimo toto řešení.

## Zkratky

**CORS** Cross-origin Resource Sharing. 7, 25, 27, 29

**ESF** Ekonomicko Správní Fakulta. 29, 41, 42

**EU** Evropská Unie. 3

**IE** Internet Explorer. 7, 29

**IS** Informační systém. 5, 22, 41

**MUNI** Masarykova Univerzita. 5, 14, 23, 41

**RDP** Remote Desktop Protocol. 6, 38



## Slovník

**@Media-Queries** Pravidla jazyka CSS umožňující podmínit použití vnořených pravidel dle určité podmínky (typicky rozlišení monitoru a podobně). 5

**AJAX** Asynchronní JavaScript a XML označuje způsob vývoje aplikací pomocí asynchronní komunikace mezi prohlížečem a serverem. 6, 7

**API** Aplikační rozhraní (API z anglického *Application Programming Interface*) označuje rozhraní poskytované k integraci programů třetích stran. 5, 11, 15, 27, 36

**ASPI** „Automatizovaný Systém Právních Informací“ označuje informační systém vyvíjený společností Wolters Kluwer, poskytující komplexní informace z právnických oborů. 2, 23, 27, 29, 34

**Bundler** Nástroj pro jednoduchou správu a instalaci gemů (balíčků programů programovacího jazyka Ruby). 22

**CI** Průběžná integrace (CI z anglického *Continuous Integration*) označuje souhrn nástrojů použitých k průběžné kontrole zdrojového kódu. Typicky sem patří spouštění testů, kontrola kvality kódu, statická analýza kódu a podobně. 8, 41

**CMS** Systém pro správu obsahu (CMS z anglického *Content Management System*) označuje typicky internetovou aplikaci umožňující uživatelům úpravu obsahu. 2, 4, 11, 16, 21–23, 34, 35, 41, 42

**CSS** Kaskádové styly (CSS z anglického *Cascading Style Sheets*) je jazyk určený k popisu vzhledu webových stránek. 5, 6, 21, 22, 27, 37

**CVS** Systém ke správě verzí projektu (CVS z anglického *Concurrent Version System*) slouží k ukládání historie verzí zdrojového kódu. 8

**framework** Označení pro nástroj ulehčující vývoj software, typicky obsahující podpůrné knihovny, nástroje či popisující správný postup vývoje. 4

**nasazení** Proces instalace projektu na typicky vzdálený server a spuštění případných migračních skriptů a podobně. 8, 23, 31

**open-source** Software jehož zdrojový kód je volně dostupný a dle licence i upravitelný. 2, 8

**relace** Označuje přetrvávající spojení mezi serverem a klientem. 27

**responsivní web design** Způsob stylování webových dokumentů, při kterém je brán ohled na různá rozlišení klientských zařízení (telefon, tablet, počítač). 5

**Servlet** Program v jazyce JAVA, který na straně serveru zpracovává HTTP požadavky. 25, 27, 29, 30, 33, 34

**SSH** Zabezpečený komunikační protokol (z anglického *Secure Shell* používaný v TCP/IP sítích. 29

**URL** Jednotný lokátor zdrojů (URL z anglického *Uniform Resource Locator* označuje řetězec znaků definující jedinečné umístění. 9, 15, 17, 21, 23, 29

**UČO** Unikátní číslo studenta či zaměstnance vysoké školy. 14, 23

**WYSIWYG** Zkratka anglického „*What you see is what you get*“, doslovně přeloženo jako „dostaneš to co vidíš“. Používá se pro označení editorů html kódu, které poskytují formátování pomocí tlačítek a výstup automaticky konvertují do html kódu. 10, 35

**XSS** Využití bezpečnostních chyb stránky (XSS z anglického *Cross-Site Scripting* za pomoci narušení skriptů stránek a podstrčení změněného kódu či dat. 25

## Seznam obrázků

2.1	Architektura komunikace nástroje Guacamole skrze vzdálenou plochu	7
3.1	Diagram tříd popisující vztahy mezi typy obsahu	18
3.2	Rozvržení stránky	20
3.3	Architektura vykreslení HTML kódu stránky	22
4.1	Neupravená architektura a rozčlenění řešení	26
4.2	Architektura přizpůsobená pro portál ESF	26
4.3	Proces připojení ke vzdálené ploše	28
5.1	Prostředí pracovního stolu pro uživatele s právy moderátora.	36

## Seznam tabulek

- 2.1 Základní typy entit v Drupalu 9
- 3.1 Porovnání verzí modulů mezi Drupalem 6 a 7 12
- 6.1 Porovnání technologií použitých na lokálním a produkčním prostředí 38

## Reference

- [1] Market share trends for content management systems for websites. [http://w3techs.com/technologies/history\\_overview/content\\_management](http://w3techs.com/technologies/history_overview/content_management).
- [2] Wikipedia: Drupal. <http://en.wikipedia.org/wiki/Drupal>.
- [3] Andreas Aderhold a Alex Black a Manuel Holtgrewe a Hans Lellelid a Michiel Rook a Johan Persson. Phing user guide. <http://www.phing.info/docs/guide/stable/index.html>.
- [4] Drupal Connect. 20 reasons why drupal 7 is better than drupal 6. <http://drupalconnect.com/blog/20-reasons-why-drupal-7-better-drupal-6>.
- [5] Mike Crittenden. *Responsive Theming for Drupal*. O'Reilly Media, 2014.
- [6] Monsur Hossain. Using cors. <http://www.html5rocks.com/en/tutorials/cors/>.
- [7] Nathaniel Catchpole Jeff Sheltren, Narayan Newton. *High Performance Drupal*. O'Reilly Media, 2013.
- [8] Mgr. Ondřej Materna. Návrh a realizace právního portálu pro esf mu. 2013.
- [9] Vlad Savitsky. Optimizing javascript and css-files in drupal. <http://shvetsgroup.com/blog/optimizing-javascript-and-css-files-drupal>.
- [10] Sammy Spets. *Programming Drupal 7 Entities*. Packt Publishing, 2013.