

# **Vývoj podpůrných služeb pro poskytování geoprostorových informací na bázi WMS**

Development of Support Services for Geographical  
Information Provision Based on WMS

Bc. Petr Balogh

---

Diplomová práce  
2012



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2011/2012

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr BALOGH**  
Osobní číslo: **A10479**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Vývoj podpůrných služeb pro poskytování  
geoprostorových informací na bázi WMS**

Zásady pro vypracování:

1. Vytvořte rešerši na téma využití a dostupnosti webově orientovaných služeb geoinformačních systémů.
2. Vytvořte webovou aplikaci v oblasti geoinformačních technologií využívající cloudové řešení na platformě Google AppEngine v jazyce Python.
3. Aplikace bude mimo jiné obsahovat či poskytovat tyto služby: registrace uživatelů, katalog českých WMS serverů, ověřování funkčnosti serverů za pomoci OGC standardů WMS, rozpoznávání vrstev, cachování dat, náhledy WMS serverů, apod.
4. Aplikaci publikujte na veřejně dostupném serveru.
5. Vytvořte programovou a uživatelskou dokumentaci.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. FU, Pinde a Jiulin SUN. Web GIS: principles and applications. 1st ed. Redlands, Calif: ESRI Press, 2010. ISBN 15-894-8245-X.
2. MITCHELL, Tyler. Web mapping illustrated. Sebastopol, Calif.: OReilly, c2005, 349 s. ISBN 978-059-6008-659.
3. LUTZ, Mark. Naučte se Python. 1. vyd. Praha: Grada, 2003, 339 s. ISBN 80-247-0367-X.
4. HARMS, Daryl. Začínáme programovat v jazyce Python. 1. vyd. Brno: Computer Press, 2003, 458 s. ISBN 80-722-6799-X.
5. SANDERSON, Dan. Programming Google App Engine. 1st ed. Sebastopol, CA: OReilly, c2010, 367 s. ISBN 05-965-2272-X.
6. SEVERANCE, Charles R. Using Google App Engine. 1st ed. Sebastopol, CA: OReilly, c2009, 241 s. ISBN 05-968-0069-X.
7. Open Geospatial Consortium OpenGIS [online]. 1994 [cit. 2012-02-01]. Dostupné z: <http://www.opengeospatial.org/>

Vedoucí diplomové práce:

**Ing. Michal Bližňák, Ph.D.**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

**24. února 2012**

Termín odevzdání diplomové práce:

**21. května 2012**

Ve Zlíně dne 24. února 2012



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Mgr. Roman Jasek, Ph.D.  
*ředitel ústavu*

### Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby,
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce,
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3,
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona,
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo - diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše),
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům,
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### Prohlašuji, že

- jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor,
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

18.5.2012... 

podpis diplomanta

## **ABSTRAKT**

Diplomová práce na téma „Vývoj podpůrných služeb pro poskytování geoprostorových informací na bázi WMS“ pojednává o aktuálně používaných webových službách v oblasti geoinformačních technologií, jejich využití a poskytování funkcionality při práci s prostorovými informacemi. Dále jsou v práci popsány standardy a normy, které jsou využity při poskytování geografických dat, nebo jsou ve webových službách implementovány. V praktické části se práce věnuje popisu použité technologie cloudového řešení, návrhu a implementaci aplikace, jenž slouží jako ucelený katalog dostupných WMS serverů v České republice. Cílem práce bylo vytvořit webovou aplikaci naprogramovanou v jazyce Python fungující v cloudovém řešení s názvem „Google App Engine“. Tato aplikace by měla obsahovat určité podpůrné služby pro poskytování geoprostorových informací na bázi standardu „Web Map Service“. Výsledná aplikace v podobě interaktivního katalogu může sloužit jak odborníkům v GIS, kteří hledají geografická data, tak i běžným uživatelům, kteří nahlíží do mapových dat, která nejsou běžně k dispozici. Součástí diplomové práce je zároveň i podrobná uživatelská dokumentace k vytvořené aplikaci.

*Klíčová slova:* WMS, GIS, webové služby, Google App Engine, cloud, Python

## **ABSTRACT**

Diploma thesis on „Development of Support Services for Geographical Information Provision Based on WMS“ deals with the currently used web services in the area of geoinformation technologies, their use and providing functionality for working with spatial information. This work also describes the standards which are employed for providing geographic data or are implemented within the web services. The practical part deals with the description of the technology used for cloud-based solution and describes the design and implementation of the application which serves as a compact catalog of available WMS servers in the Czech Republic. The aim was to create a web application written in Python that would run on the „Google App Engine“ cloud solution. This application should incorporate certain support services for providing geospatial information based on „Web Map Service“ standard. The resulting application in the form of interactive catalog may serve both GIS professionals who are looking for geographic data and ordinary users who just like to browse through some map data that are not usually available. The thesis also contains a detailed user documentation for the developed application.

*Keywords:* WMS, GIS, Web Services, Google App Engine, Cloud, Python

Tímto bych rád poděkoval vedoucímu práce, Ing. Michalovi Bližnákovi, Ph.D, za spolupráci, cenné rady a připomínky při našich konzultacích. Dále bych chtěl poděkovat své rodině, zaměstnavateli a přátelům, kteří mě podporovali při celém studiu.

## OBSAH

ÚVOD .....	10
<b>I TEORETICKÁ ČÁST .....</b>	<b>11</b>
<b>1 WEBOVÉ SLUŽBY .....</b>	<b>12</b>
1.1 SOAP .....	12
1.2 REST .....	13
1.2.1 Základní vlastnosti používané v architektuře v REST .....	14
<b>2 GEOINFORMAČNÍ SYSTÉMY .....</b>	<b>15</b>
2.1 HISTORIE GIS .....	15
2.2 AKTUÁLNÍ OBORY V GIS .....	16
2.3 UŽIVATELÉ .....	17
<b>3 WEBOVĚ ORIENTOVANÉ SLUŽBY V GIS .....</b>	<b>18</b>
3.1 METODOLOGIE .....	18
3.1.1 Klienti .....	20
3.1.2 Web GIS server .....	23
3.1.3 Architektura klientů v GIS .....	24
3.1.4 GIS databáze .....	24
3.2 ROZDĚLENÍ GIS SLUŽEB .....	25
3.2.1 Mapové služby .....	26
3.2.2 Objektově funkční a editační služby .....	26
3.2.3 Vyhledávací služby .....	27
3.2.4 Obrazové služby .....	27
3.2.5 Geokódové služby .....	27
3.2.6 Geoprocenční služby .....	27
3.3 TESTOVÁNÍ SLUŽEB .....	27
<b>4 INTEROPERABILITA, STANDARDY A JEJICH IMPLEMENTACE V GIS SLUŽBÁCH .....</b>	<b>29</b>
4.1 OPEN GEOSPATIAL CONSORTIUM (OGC) .....	29
4.2 OSTATNÍ STANDARDY .....	30
4.2.1 ISO/TC 211 .....	30
4.2.2 W3C .....	30
4.2.3 INSPIRE .....	30
4.3 INTEROPERABILITA PODLE ČSN ISO 19119 .....	31
4.4 WEB MAP SERVICE (WMS) .....	31
4.5 WEB FEATURE SERVICE (WFS) .....	33

4.6	WEB COVERAGE SERVICE (WCS).....	35
4.7	WEB PROCESSING SERVICE (WPS).....	36
4.8	KEYHOLE MARKUP LANGUAGE (KML).....	38
4.9	GEOGRAPHY MARKUP LANGUAGE (GML).....	39
4.10	GEORSS.....	40
<b>II</b>	<b>PRAKTICKÁ ČÁST.....</b>	<b>42</b>
<b>5</b>	<b>POUŽITÉ TECHNOLOGIE.....</b>	<b>43</b>
5.1	GOOGLE APP ENGINE.....	43
5.1.1	Aplikační prostředí.....	43
5.1.2	Zabezpečené prostředí.....	44
5.1.3	Běhové prostředí Pythonu.....	44
5.1.4	Úložiště dat.....	45
5.1.5	Kvóty a limity.....	46
5.2	OPENLAYERS.....	46
5.3	OSTATNÍ TECHNOLOGIE A KNIHOVNY TŘETÍCH STRAN.....	47
5.3.1	OWSLib.....	47
5.3.2	PagedQuery.....	48
<b>6</b>	<b>NÁVRH A ZPRACOVÁNÍ APLIKACE.....</b>	<b>49</b>
6.1	LOGICKÁ STRUKTURA APLIKACE.....	50
6.2	NÁVRH A REALIZACE UŽIVATELSKÉHO ROZHŘANÍ.....	50
6.3	USE CASE DIAGRAM.....	52
6.4	SLUŽBY APLIKACE.....	53
6.5	TŘÍDY PRO PRÁCI S WMS SERVERY A DATOVÝ MODEL.....	55
6.6	UŽIVATELÉ A ZABEZPEČENÍ APLIKACE.....	56
<b>7</b>	<b>ŘEŠENÍ PŘI TVORBĚ APLIKACE.....</b>	<b>62</b>
7.1	SOUŘADNICOVÉ A REFERENČNÍ SYSTÉMY.....	62
7.2	VYTVÁŘENÍ NÁHLEDŮ.....	62
7.3	VYHLEDÁVÁNÍ V KATALOGU.....	65
7.4	ZOBRAZOVÁNÍ LEGEND.....	65
<b>8</b>	<b>POPIS APLIKACE A JEJÍ OVLÁDÁNÍ.....</b>	<b>66</b>
8.1	ÚVODNÍ STRÁNKA.....	66
8.2	REGISTRACE.....	67
8.3	FUNKCE WMS INFO.....	67
8.4	MENU VYHLEDÁVÁNÍ.....	68
8.5	PŘIHLÁŠENÍ A ZAPOMENUTÉ HESLO.....	69

8.6	UŽIVATELSKÉ FUNKCE.....	69
8.6.1	Přidat WMS server.....	69
8.6.2	Změna osobních údajů a hesla, smazání účtu .....	70
8.7	PROHLÍŽENÍ INFORMACÍ O ULOŽENÉM WMS SERVERU .....	70
8.8	TAGY A KATEGORIE.....	71
8.9	ADMINISTRÁTORSKÉ FUNKCE.....	72
	<b>ZÁVĚR.....</b>	<b>73</b>
	<b>CONCLUSION .....</b>	<b>74</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>75</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>79</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>80</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>81</b>

## ÚVOD

Geoinformační systémy jsou počítačové systémy, které nějakým způsobem pracují s geografickými daty, nebo je pomáhají získat. Tato data mohou upravovat, následně analyzovat a podat výstupy v patřičné formě, například vizualizací. S vývojem nových technologií se geoinformační systémy dostávají čím dál více do povědomí běžných uživatelů a obsah prostorových dat tak neustále roste. Proto je nutné geodata uchovávat a následně distribuovat mezi uživatele, kteří je využívají. Pro tyto účely byla použita celosvětová síť Internet, nejrozšířenější médium sloužící k výměně informací.

Diplomová práce si klade za cíl přiblížit čtenáři standardy webových služeb, které byly vybudovány pro zobecnění komunikace mezi různými druhy softwarových produktů nezávislých na použité platformě a ukázat čtenáři další využití těchto služeb v oblasti prostorových informací a geoinformačních systémů. Těžištěm práce je naprogramování nové interaktivní webové aplikace v cloudovém prostředí, která bude mít podpůrný vliv na poskytování geoprostorových služeb na standardu „Web Map Service“.

Práce je rozdělena na dvě části, teoretickou a praktickou. Teoretická část se obecně zabývá webovými službami a popisuje protokol SOAP a architekturu REST. Následně zmíní základy geoinformačních systémů, jejich historii, využití a popis aktuálních oborů v oblasti GIS. Uvedeme webové orientované služby v GIS, kde nejprve představíme používané technologie. Čtenáře postupně provedeme metodologií, která zahrnuje informace o klientech a jejich rozdělení, webový GIS server, architekturu tenkých a tlustých klientů a úložiště geoprostorových dat. Zmíníme používané druhy služeb v GIS a ke konci teoretické části podrobně popíšeme standardy a jejich implementace v geoprostorových službách. Zaměříme se na interoperabilitu, různé organizace věnující se standardizaci a podrobněji čtenáře seznámíme s některými implementacemi konsorcia OGC.

Praktická část práce je rozdělena na čtyři kapitoly. První kapitola se zabývá technologiemi použitými při implementaci, uvádí aplikační prostředí „Google App Engine“ a další knihovny třetích stran. Druhá kapitola popisuje logickou strukturu webové aplikace, návrh a realizaci uživatelského rozhraní a třídy reprezentující služby aplikace, které byly vytvořeny v průběhu vývoje. Formou diagramů usnadňujeme čtenáři a programátorovi orientaci ve zdrojovém kódu. Třetí kapitola uvádí konkrétní příklady řešených problémů vzniklých při tvorbě aplikace WMSkatalog. Poslední kapitola popisuje základní funkce uživatelského rozhraní a slouží jako návod pro uživatele. V závěru se práce věnuje celkovému zhodnocení, poukazuje na výsledky, přínos aplikace a zmiňuje možnosti dalšího rozvoje.

# I. TEORETICKÁ ČÁST

## 1 WEBOVÉ SLUŽBY

Webové služby jsou na serveru spuštěné programy, které poskytují určitý typ služeb, jejichž obsah je konzumován či využíván klienty. Komunikace mezi těmito dvěma stranami, klientem a serverem, je zprostředkována skrze síť, převážně internet. Tento typ služeb vznikl především z důvodu zjednodušení a generalizace komunikace mezi různými druhy softwarových produktů nezávisle na použité platformě. Účelem bylo umožnit jednoduchou komunikaci, a proto se pro tyto potřeby výměny dat začalo používat XML (Extensible Markup Language).

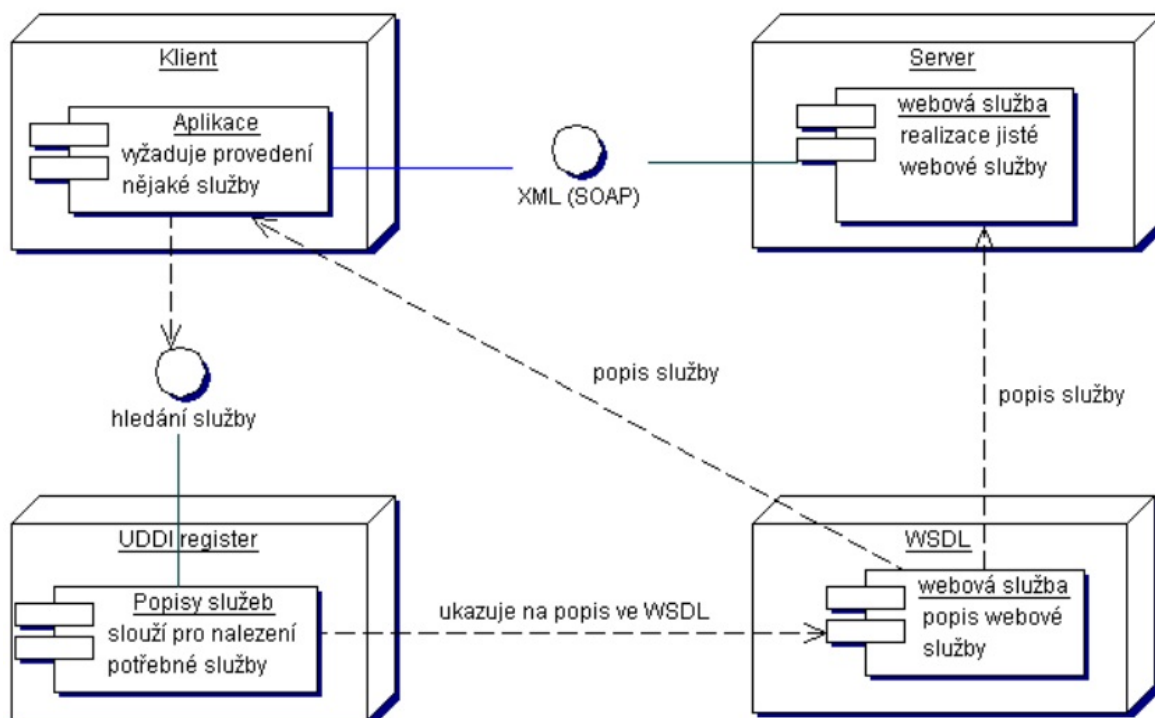
### 1.1 SOAP

Protokol, který XML začal používat pro výměnu zpráv, se nazývá SOAP, neboli Simple Object Access Protocol. Tento protokol je jedním ze stavebních pilířů webových služeb. Dále následuje WSDL (Web Services Description Language), což je formát, který využívá XML k popisu síťové služby, neboli předává informaci o tom, co služba nabízí a jakým způsobem mohou být dané funkce volány. Tyto operace a zprávy jsou popsány abstraktně a dále jsou spojeny s určitým koncovým bodem, anglicky „endpoint“, který definuje konkrétní síťový protokol, i přesný formát zpráv, který bude použit při komunikaci.

Pro komunikaci se nejčastěji používá protokol HTTP (Hypertext Transfer Protocol), který je určený hlavně pro výměnu dokumentů ve formátu HTML (HyperText Markup Language), ale díky rozšíření MIME (Multipurpose Internet Mail Extensions), je schopný přenést jakýkoli soubor. SOAP využívá pro komunikaci protokol HTTP, ten ale není jediným možným protokolem. Využívá se i ostatních známých otevřených protokolů, jako je například SMTP (Simple Mail Transfer Protocol). Výhoda použití těchto známých protokolů je také v důvěře firewallů, které pak síťovou komunikaci neblokují. Při použití HTTP protokolu se SOAP požadavek zasílá v těle HTTP požadavku a metody, které jsou k tomu používány, jsou POST a od verze SOAP 1.2 je možné využít i metodu GET. Při použití metody POST se využívá tělo požadavku pro vložení dat, kdy v hlavičce požadavku je specifikována „SOAPAction“, jež identifikuje, že se jedná o SOAP požadavek.

Další hlavní částí je UDDI registr (Universal Description, Discovery, and Integration), který se stará o poskytování informací o službách, které mohou klienti využívat. Dovoluje klientům hledat v seznamu služeb. Ti se pak mohou rozhodnout, kterou službu využijí. S UDDI registrem ovšem nekomunikují jen klienti, kteří chtějí služby využít, ale i ti, kteří chtějí své služby ostatním klientům nabízet.

Všechny výše popsané komponenty webových služeb a jejich interakci znázorňuje následující obrázek č. 1



Obr. 1. Komponenty webových služeb (zdroj: [1])

## 1.2 REST

Pro zjednodušení webových služeb se začíná čím dál více používat architektury zvané REST (Representational State Transfer). Tato architektura byla navržena pro distribuovaná heterogenní prostředí, kde se pro komunikaci využívá síť internet. Jako klienta si můžeme představit webový prohlížeč. Server nabízející službu je program napsaný v libovolném jazyce, umístěn na webovém serveru pod určitou URL (Uniform Resource Locator). Pro komunikaci se používá HTTP protokol, bez SOAP, což nám vše zjednoduší. S REST architekturou se používá při výměně dat hlavně formát JSON (JavaScript Object Notation), který je při reprezentaci stejných dat výrazně kompaktnější než XML. Tím se nám sníží i počet přenesených dat. To má ve výsledku vliv na rychlejší komunikaci a výkon služby.

U architektury REST se pro manipulaci se zdroji používají čtyři základní operace, jejichž zkratka se nazývá CRUD, což je spojení z anglických slov Create, Read, Update a Delete. REST nám tedy umožňuje provádět převážně tyto operace pro vytvoření, čtení, aktualizace a mazání zdrojů daného systému. Pro tyto účely jsou použity známé metody HTTP protokolu:

- **HTTP Post** - vytvoření zdroje,
- **HTTP Get** - čtení zdroje,

- **HTTP Put** - úpravy a vytvoření zdroje,
- **HTTP Delete** - smazání zdroje.

K identifikaci zdroje se používá URI (Uniform Resource Identifier) nebo URL (Uniform Resource Locator), kde vše představuje zdroj, a kde každý z nich má přiřazené URI (známý slogan: With REST, everything is a URL). URL adresy zdrojů jsou hierarchicky řazené, předvídatelné a jednoduché k pochopení. Programátoři tak mohou rychle a jednoduše bez čtení dokumentace vygenerovat URL, která bude směřovat k požadovanému zdroji.

Většina REST služeb se omezuje jen na metodu HTTP Get, kde všechny parametry přijímá v URL adrese. URL adresa takového požadavku pak může vypadat následovně:

```
http://up.kr-jihomoravsky.cz/WMS_JMK_UAP_B1_F/wmservice.aspx?LAYERS=C3
&TRANSPARENT=TRUE&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&STYLES=
&FORMAT=image%2Fpng&SRS=EPSG%3A102067&BBOX=-846136.36316804,
-1307499.999765,-353863.63542896,-1022499.999495&WIDTH=1425&HEIGHT=825
```

### 1.2.1 Základní vlastnosti používané v architektuře v REST

- **klient-server** - rozdělení na serverovou a klientskou část, kdy komponenta zvaná server nabízí určitý druh služeb a naslouchá tak požadavkům klienta na tyto služby,
- **bezstavovost** - klient musí vždy ve svém požadavku odeslat všechny informace o aktuálním stavu,
- **využití keše** (vyrovnávací paměti) - využívá se příznaku (cacheable/non-cacheable), který se přidává v odpovědi ze serveru. Klient toho může využít a uložit si odpověď do vyrovnávací paměti, kde při dalším požadavku na stejnou URL již znovu nemusí kontaktovat server,
- **vrstvený systém** - pro zjednodušení systému se rozdělí na hierarchické vrstvy, kde komunikace mezi nimi je omezena pouze na ty nižší.

## 2 GEOINFORMAČNÍ SYSTÉMY

Jsou počítačové systémy, které nějakým způsobem pracují s geografickými daty nebo je pomáhají získávat. Tato data mohou upravovat, následně analyzovat a podat výstupy v patřičné formě, například vizualizací. Geoinformační systém, stejně jako informační systém, můžeme rozdělit do několika částí:

- **hardware** - počítače, či výkonné výpočetní servery a jejich příslušenství,
- **software** - programy sloužící pro práci s geografickými daty,
- **data** - nesoucí prostorový vztah k povrchu Země s kterými systémy pracují,
- **lidé** - uživatelé, nebo operátoři využívající či pracující s geoinformačním systémem.

### 2.1 Historie GIS

Vývoj geoinformačních systémů souvisí s vývojem počítačů a historicky sahá již do 70. let minulého století. Ze začátku se používaly převážně pro tvorbu map na počítačových systémech a byly označovány jako Computer Aided Cartography (CAM), neboli počítačem podporovaná kartografie. V roce 1977 byly zformulovány základní úlohy těchto systémů, mezi něž patří:

- rychlejší a přesnější tvorba map,
- tvorba map se zohledněním specifických požadavků uživatelů,
- možnosti tvorby map v případech, kdy nebyl k dispozici vyškolený personál,
- možnosti experimentovat s rozdílnými grafickými reprezentacemi stejných údajů,
- možnosti tvorby a aktualizace map z údajů v digitální podobě,
- možnosti analýzy údajů, které jsou založené na syntéze statistických analýz a mapování,
- minimalizace potřeby využívání papírových map a následná minimalizace míry generalizace údajů,
- využívání map, které je problematické tvořit bez výpočetní techniky (3D mapy, stereoskopické mapy apod.).

Jakmile proběhla tato formulace z roku 1977, začaly se objevovat první programy na alespoň částečně automatizovanou tvorbu map. Zanedlouho se CAM přesunula do

polohy efektivních nástrojů na sdílení a zpřístupňování velkého množství prostorově referencovaných dat formou rozdílných informačních systémů. Tak se vyvinuly GIS. [2] GIS se začal používat v mnoha odvětvích a oblastech, kde nejde jen o technicky zaměřené obory, ale své uplatnění si našel v odvětvích, jako jsou:

- **veřejná správa** – při zpřístupňování dat z katastru nemovitostí, územní plánování, evidování různých informací, pasportizace, atd.,
- **bezpečnostní složky** – kde se jedná o zajištění a zlepšení dostupnosti služeb pro zvýšení bezpečnosti obyvatel. Lze analyzovat nejkratší možné trasy pro výjezdy záchranných vozidel s aktuálními informacemi o dopravě, atd.,
- **inženýrské sítě** – potrubí, kanalizace, elektrické sloupy, vedení, kabeláže, optické linky a vedení informací s nimi spjatými například při opravách, atd.,
- **kartografie** – tvorba mapových podkladů, digitalizace starých map, atd.,
- **přírodní zdroje** – lesnictví, zemědělství, těžba nerostných surovin, atd.,
- **armáda** – plánování akcí, mapové podklady, využití družicových a leteckých snímků, využití GPS, atd.,
- **doprava** – budování infrastruktury, mapování, přeprava zboží, GPS navigace, atd.

## 2.2 Aktuální obory v GIS

V dnešní době je GIS velice rozsáhlou množinou softwarových produktů, zabývající se více či méně komplikovanými analýzami, a práci s geodaty. Geodata můžeme popsat jako soubor informací, které jsou geograficky spojené se zemským povrchem, a tak je možné podat výsledky v uživatelsky přívětivé formě, například zobrazeny v mapě. Při práci se používají složité algoritmy, metody a postupy, které se vyvíjely a zlepšovaly postupně s časem. Patří sem především tyto obory:

- **digitální kartografie** – tvorba mapových podkladů za pomoci počítačové techniky a dalších periférií,
- **dálkový průzkum Země (DPZ)** – využití digitálních informací o nadmořské výšce, leteckých a družicových snímků ke sběru a analýze informací o objektech a jevech na zemském povrchu,
- **geomatika/geoinformatika** – definována jako vědecký a technický interdisciplinární obor zabývající se sběrem, distribucí, ukládáním, analýzou, zpracováním a prezentací geografických dat nebo geografických informací [4],

- **GPS**, neboli Global Positioning System, kde se jedná o globální družicový polohový systém, zpočátku využívaný pro vojenské účely, později širokou veřejností. Díky GPS lze určit přesnou polohu, a čas kdekoliv na Zemi,
- **digitální fotogrammetrie** je na digitálních technologiích založený obor, zabývající se zpracováním obrazových dat zemského povrchu, získaných prostřednictvím metod DPZ,
- **geostatistika** se zabývá odhady a předpovědí spojitých jevů v prostoru za použití dat jen z omezeného počtu míst v tomto prostoru, aplikuje obecné statistické principy na modelování a vyvozování závěrů o geostatistických problémech.

Geografické informační systémy hrají významnou roli v analýze dat, produkci výstupů a v řízení rozhodování všeho druhu. Poskytují nám prostředky pro vizualizaci informací, které nám umožňují provedení lepšího rozhodnutí, jak řešit závažné problémy týkající se našeho světa.

### 2.3 Uživatelé

Lidi, kteří používají GIS, můžeme rozdělit do dvou skupin uživatelů:

- **aktivní** – jsou osoby, které budují geodata, kontrolují, analyzují nebo je nějakým způsobem upravují. Jedná se o lidi, jimž problematika GIS není cizí, a většinou jsou odborníky ve svém oboru,
- **pasivní** – jsou pouhými konzumenty obsahu, který ovšem mohou využívat například při své práci. Mnohdy tento uživatel ani nebere na vědomí, že s geoinformačním systémem pracuje, protože prohlížení mapy na internetu bere v dnešní době jako samozřejmost.

GIS se díky moderní době a trendům dostává do široké veřejnosti obyvatel. Díky chytrým telefonům má dnes mnoho lidí kapesní GIS formou navigace, nebo interaktivních aplikací spolupracujících s online či offline mapovými podklady a ostatními geodaty.

Profesionálové v oboru se již zabývají složitými analýzami dat, kde jsou si vědomi, jaké algoritmy a postupy používají. Současný software se ovšem snaží být uživatelsky přívětivý, a tak plno operací je zjednodušených na doslova pár kliknutí myši a zadáním několika parametrů.

### 3 WEBOVĚ ORIENTO VANÉ SLUŽBY V GIS

Stejně jako webové služby, obecně i GIS služby jsou sestrojeny pro provádění určitých funkcí na straně serveru, které jsou ovšem spojené s využitím geoprostorových dat. Tyto funkce využívají otevřené webové protokoly jako jsou SOAP, HTTP, architektury REST s použitím XML, JSON, HTML a podobně.

Webové služby mají revoluční potenciál ve způsobu, pro jaký byly vyvinuty, jak je přístupný a jak je používáný. Webové služby v GIS poskytují prostorová data a funkcionality na WWW stránkách, nebo přímo přes internet a intranet. Aplikace využívající tyto služby pak mohou slučovat obyčejná data s prostorovými a poskytovat tak geoprostorové operace jako jsou:

- hledání adres,
- prostorové dotazy (na body, linie, polygony),
- hledání cest / navigace,
- mapování,
- lokace polohy.

GIS služby se snaží ulehčit sdílení geografických dat a integrují ostatní technologie dohromady. Příkladem mohou být kamerové systémy integrované v GIS.

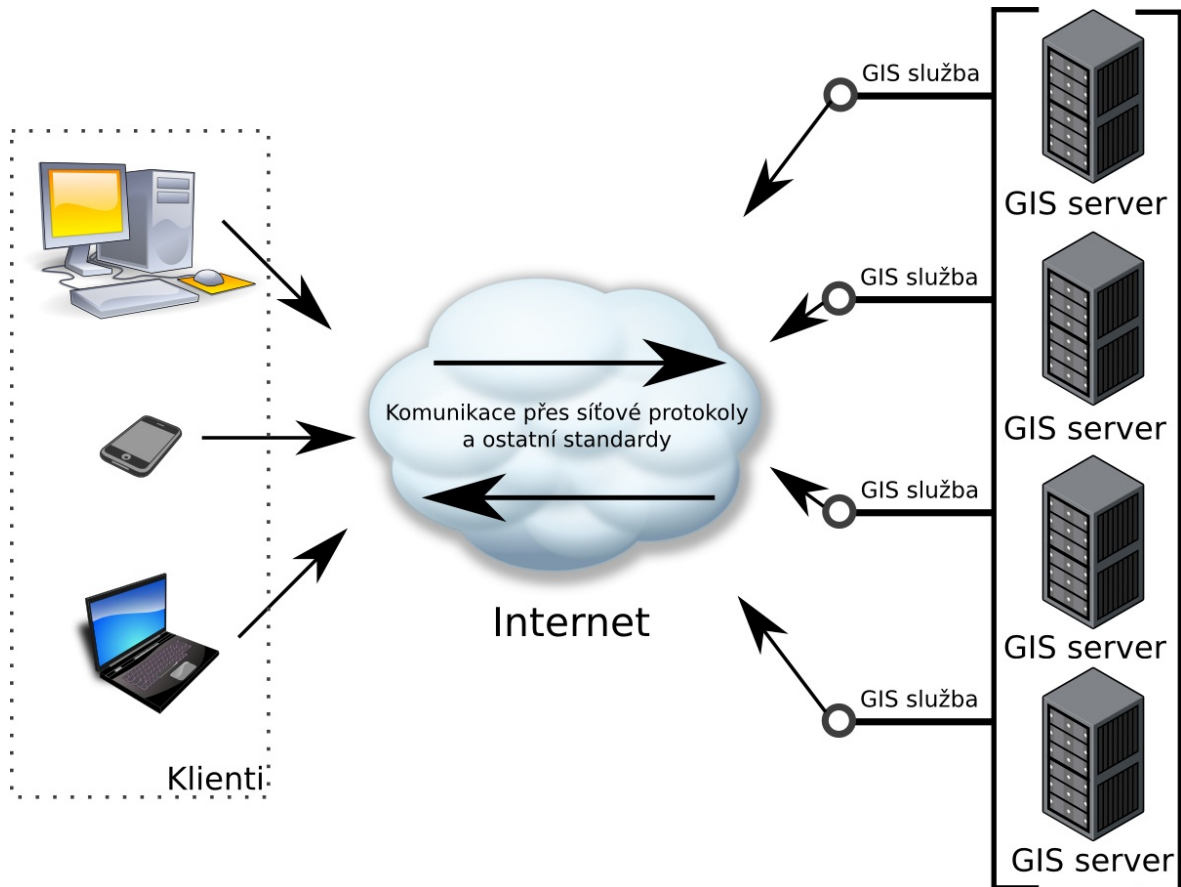
Tyto služby dále umožňují uživatelům přistupovat ke geodatům, s nimi spojenou funkcionalitou přes web, a spojit je s jejich vlastními či používanými systémy a aplikacemi bez nutnosti vlastního vývoje nebo hostování specifických nástrojů a datových sad. Tyto služby tak poskytují uživatelsky přívětivé prostředí pro mnoho aplikací, které používají prostorová data jako jsou katastry nemovitostí, územní plánování, turistika a elektronické obchodování.

Další z výhod je přinášena vývojařům webových aplikací, kteří mohou integrovat funkcionalitu GIS do svých aplikací, bez nutnosti znalostí problematik, vyvíjení vlastního systému a s absencí uchovávání a hostování mapových dat, které zabírají velkou kapacitu, o které se starají poskytovatelé služeb.

#### 3.1 Metodologie

Je potřebné se zaměřit na techniky a postupy používaných u webově orientovaných GIS služeb. Vytvořením tohoto druhu služeb vznikla platforma nezávislá na distribučním kanále pro geodata. Aplikace začaly sdílet data z odlišných zdrojů, s rozličnými formáty, které kombinují v samostatných aplikacích, aniž by bylo zřejmé, že tato data pochází z různých zdrojů či míst. Princip komunikace je znázorněn na obrázku č. 2. kde

vidíme použití různých druhů klientů, což nám znázorňuje nezávislost na platformě, kde komunikace je skrze internet a jeho protokoly se zachováním určitých standardů, jako je například XML. Na druhé straně vidíme GIS servery, jenž poskytují různé druhy služeb, které jsou nabízeny a publikovány skrze internet. Webové služby mají



Obr. 2. Komunikace klientů a serverů přes webové služby

následující vlastnosti:

- jsou součástí aplikace,
- komunikují pomocí otevřených protokolů,
- jsou samostatné a samostatně popsatelné (WSDL),
- mohou být naleznutelné přes UDDI,
- lze je použít v ostatních aplikacích,
- XML je základem webové služby.

Každá z publikovaných služeb poskytuje určité API(Application Programming Interface), které určuje jak získat přístup ke službě. Jedná se o zveřejněnou specifikaci, která

popisuje jak jiné softwarové produkty mohou přistupovat k funkcím automatizované služby. API jsou navržena tak, aby řídila doručení geodat z GIS serveru směrem ke klientovi (např. webovému prohlížeči). Data pak jsou přístupná běžnými skriptovacími jazyky, jako je například JavaScript, a ty je pak dále možné díky těmto jazykům jednoduše modifikovat či zobrazovat již na straně klienta.

### 3.1.1 Klienti

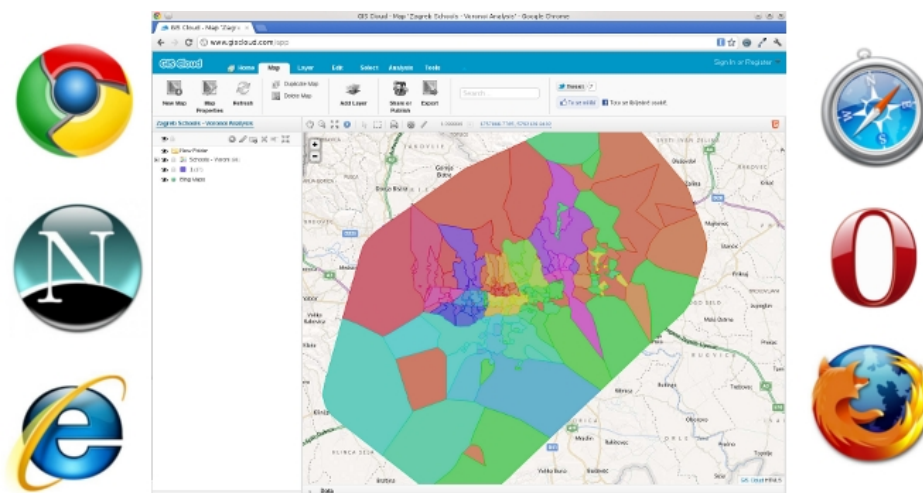
**Webový prohlížeč** Jedná se o softwarovou aplikaci sloužící k načtení a prezentaci informací z webových zdrojů. Pro mnoho uživatelů se prohlížeč jeví jako hlavní tvář webu, ale fakt je, že se jedná pouze o webového klienta, který implementuje HTTP, HTML, JavaScript a další specifikace, jenž umožňují komunikaci s webovým serverem, zobrazovat HTML stránky a interpretovat i spouštět JavaScriptový a jiný standardizovaný kód.

Internetových prohlížečů existuje celá řada, každý z nich má rozličnou podporu pro zobrazování a vykonávání HTML, JavaScriptu a dalších technologií. Proto se může stát, že stejná webová stránka bude v každém prohlížeči zobrazena jinak. To ztěžuje práci programátorům, jelikož musí optimalizovat webovou aplikaci pro alespoň nejpožívanější webové prohlížeče. Mezi nejpopulárnější můžeme řadit:

- Google Chrome,
- Mozilla Firefox,
- Internet Explorer,
- Safari,
- Opera.

#### *Programový aparát webových prohlížečů*

- JavaScript, vynalezen firmou Netscape v roce 1995, je skriptovací jazyk, který je primárně spouštěn webovým prohlížečem. S jeho použitím se stávají stránky výrazně dynamičtějšími, interaktivnějšími a uživatelsky více přívětivými.
- AJAX (Asynchronous JavaScript and XML) se stal populárním od roku 2005. Nejedná se o nový programovací jazyk, ale o seskupení existujících webových vývojařských technik sloužících k tvorbě lepších, rychlejších, a více interaktivnějších webových aplikací v internetových prohlížečích. Princip AJAXu je v asynchronní komunikaci mezi prohlížečem a webovým serverem, jenž se odehrává na pozadí. Výsledek takové komunikace je zobrazen v části aktuální webové stránky



Obr. 3. Ukázka webového klienta s logy předních prohlížečů

(nebo provedení jiné operace) bez nutnosti znovu načítání celého obsahu webové stránky.

- **Adobe Flex** je otevřeným frameworkem pro budování vysoce interaktivních, webových aplikací, které jsou podporovány ve všech hlavních webových prohlížečích. Flex může zobrazovat efektivní animace a různé druhy grafických efektů díky použití Adobe Flash Playeru, který je podporován i na různých platformách a operačních systémech. Pro popis chování aplikační logiky a uživatelského rozhraní se používá MXML (jazyk založený na XML), a pro potřeby tvorby logiky na straně klienta je použit ActionScript (silně typový, objektově orientovaný programovací jazyk).
- **Microsoft Silverlight** je programový doplněk podporován napříč různými webovými prohlížeči a operačními systémy používaný k poskytování aplikací využívající grafiku, animace, či video. Je spojen s používáním platformy .NET framework k tvorbě uživatelského rozhraní s programovacími jazyky jako jsou C# nebo VB.NET.
- **Oracle/Sun JavaFX** je dalším krokem vývoje Java aplikací obsahujících bohaté uživatelské rozhraní. Je navržen pro poskytování lehkých, hardwarově akcelerovaných aplikací s využitím plné integrace v běhovém prostředí programovacího jazyku Java (Java Runtime Environment).
- **HTML5** je důsledkem vývoje a inovace webových standardů, které čím dál více vstupuje do popředí dnešních trendů webových aplikací. Počátky vývoje sahají do roku 2004, kdy se na vývoji začali podílet W3C HTML WG (W3C HTML Working Group) a WHATWG (Web Hypertext Application Technology Working Group). Spolu s W3C spolupracují přední korporace zabývající se vývojem majo-

ritních prohlížečů, kam patří Microsoft, Apple, Mozilla a Opera. HTML5 přidává řadu nových syntaktických prvků, mezi něž patří například elementy `<video>`, `<audio>` a `<canvas>`. Nové vlastnosti umožňují jednodušeji vkládat multimédia a grafický kontext, se kterým lze dále pracovat bez použití dalších zásuvných modulů třetích stran.

Některé z výše uvedených druhů programového aparátu nám přináší možnost využití vytvořeného programového kódu nejen ve webových prohlížečích, ale s drobnými úpravami i v desktopových aplikacích, což zefektivňuje práci.

**Mobilní klienti** v poslední době čím dál více vstupují do popředí mobilní zařízení, a s ním i mobilní GIS. Díky technologickému vývoji a snížení cen se s mobilními platformami setkáváme téměř všude. Výhoda pak představuje využitelnost GIS technologií, kdekoliv, kde to potřebujeme. Mobilní klienty můžeme rozdělit do dvou kategorií:

- **závislé na prohlížeči** – jejich výkon se omezuje na možnosti prohlížeče, které se u různých platformech mohou velmi lišit. Vesměs mohou poskytovat jen jednoduché funkcionality GIS, s prostým uživatelským ovládáním,
- **nativní aplikace** – zde jsou možnosti daleko rozsáhlejší. Využívá se mnoha programovacích jazyků, především těch, které jsou primárně určeny pro danou platformu (Objective-C, .NET framework, Java, ...). Využívá se API operačních systémů, jenž poskytuje daleko větší komfort pro vývoj uživatelského prostředí na dané platformě.



Obr. 4. Mobilní klienti iPad a iPhone s aplikací iKatastr

**Desktopové aplikace** jsou softwarové produkty spustitelné přímo operačním systémem, nebo jiným subsystémem, který s ním spolupracuje. Příkladem mohou být aplikace napsané v .NET frameworku, Java a v mnoha dalších programovacích jazycích.

Tyto desktopové aplikace mohou přistupovat k webovým službám obdobně jako webový prohlížeč, ale mnohdy nejsou tak omezeny. Prohlížeč totiž dovoluje programovému kódu přístup jen do ohraničeného území, které se nazývá „sandbox“, jež limituje funkcionality přístupu k systémovým zdrojům. Další výhodou spočívá v práci s lokálními daty ve formě souborů, lokálních databází a ostatních periférií (např. GPS modul).

Jako příkladové aplikace můžeme uvést ArcGis Desktop, Qgis, Google Earth, GRASS GIS a mnoho dalších. Jejich možnosti z pouhých soběstačných aplikací pracujících převážně s lokálními daty se pomalu ubírají směrem k distribuovanému prostředí a desktopové aplikace začínají pracovat jako klienti prostorových webově orientovaných služeb. Narozdíl od tenkých klientů mohou naplno využít svůj potenciál robustního systému a dát uživateli plnou moc nad geoprostorovými daty.



Obr. 5. Desktopová aplikace ArcGis Desktop

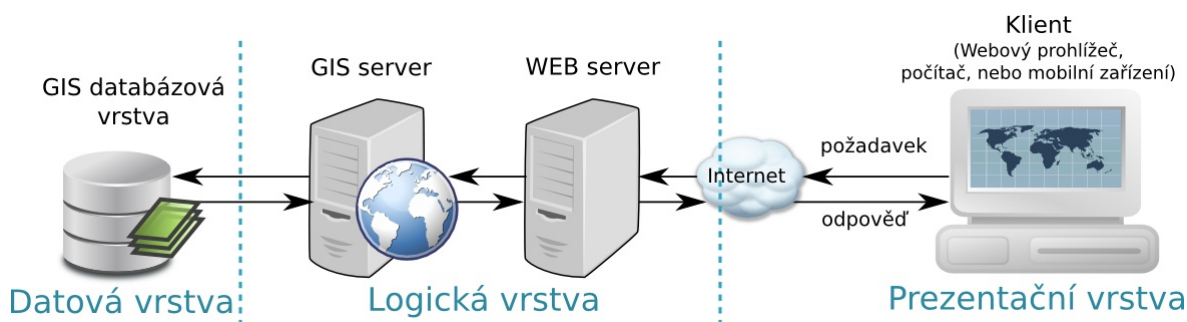
### 3.1.2 Web GIS server

Jedná se o nejdůležitější komponentu ve webově orientovaných geoinformačních systémech a službách. Jeho funkcionalitou je možnost poskytnout přizpůsobení, rozšiřitelnost a výkon důležitý pro úspěch GIS webové aplikace či služby. Architekturu webového GIS můžeme rozdělit na tři vrstvy:

- datovou,
- logickou,
- prezentační.

Na obrázku č. 6 znázorňujeme tuto architekturu a posloupnost putování dat skrze jednotlivé vrstvy. Požadavek je vyslán z klientské stanice, kterou může být webový prohlížeč, osobní počítač nebo mobilní zařízení. Putuje směrem k webovému serveru skrze internet, který předá požadavek ke zpracování GIS serveru. GIS server může dále komunikovat s databázovým GIS serverem. Výstup operace pak putuje zpět na

webový server, který se postará o doručení informací zpět ke klientskému zařízení, kde je výsledek prezentován uživateli.



Obr. 6. Zobrazení architektury webového GIS

### 3.1.3 Architektura klientů v GIS

Základní úvahou pro všechny klient-server aplikace je, jakým způsobem můžeme rozdělit zatížení mezi klientskou a serverovou část. Můžeme se rozhodnout mezi dvěma modely, které si následně popíšeme:

- **architektura tenkých klientů** – je závislá na výkonnosti serveru, kdy klientská aplikace se především stará o prezentování dat uživateli. Takový klient zasílá požadavky směrem k serveru, kde jsou pak zpracovány a následně vykonány (poskládání mapové kompozice, analýzy a další operace) a v odpovědi zaslány zpět na stranu klienta. Klientská část aplikace je ve většině případů přístupná na webové stránce, kdy pro její spuštění je vyžadována pouze přítomnost webového prohlížeče v operačním systému daného zařízení.
- **architektura tlustých klientů** – pokud aplikace využívá především výpočetní potenciál hostovaného stroje pro poskytnutí více funkcionalit, a server využívá například jen jako datovou vrstvu, jedná se o tlustého klienta. Tito klienti jsou především nativní aplikace, jenž pracují na principu požadavku serveru na data, která si následně uloží do paměti a další operace a analýzy jsou prováděny výpočetním výkonem zařízení, na kterém je aplikace spuštěna.

### 3.1.4 GIS databáze

Tyto databáze jsou úložištěm geodat pro účel GIS. Slouží k uchovávání kolekcí datových souborů rozdílných typů jako jsou vektorová data (body, linie a polygony), rastrová data (satelitní a letecké snímky), data CAD aplikací, 3D, GPS souřadnice, data z měření, atd. Rozsah může být od malé jedouživateléské databáze, až po velký datový sklad pro celou organizaci, která může být přístupována několika lidmi ve stejný okamžik.

Geodatabáze tvoří základní podporu pro webové GIS aplikace, jelikož odpověď od takové aplikace může být jen tak dobrá, jako je kvalita informací obsažených a poskytovaných geodatabází. Moderní databáze v GIS nabízí některé z následujících možností:

- uchovávání prostorových dat a jejich kolekcí v centralizovaných distribuovaných systémech,
- aplikuje sofistikované pravidla a vztahy mezi daty,
- definuje geoprostorové relační modely,
- udržuje integritu prostorových dat v konzistentní formě,
- umožňuje víceuživatelský přístup s podporou verzování,
- podporuje vlastní funkce a chování,
- poskytuje prostředky pro zabezpečení dat, zálohování, obnovy a možnosti takzvaných „rollbacků“ (s kterými jsme schopni po neúspěšném provedení operace vrátit data zpět do perzistentní podoby, ve stavu ve kterém se nacházela před spuštěním operace),
- udržovat vysoký výkon s přibývajícími daty a připojenými uživateli.

### 3.2 Rozdělení GIS služeb

Aktuální typy služeb můžeme rozdělit do několika druhů:

- **geoprocení** – z anglického „geoprocessing“, které poskytují přístup k geografickým operacím, analýzám, které jsou následně zpracovány na straně výkonného serveru s přístupnými nástroji a výsledek této operace je vrácen klientovi v předem specifikovaném formátu,
- **mapové** – poskytují přístup k obsahu mapových dokumentů,
- **3D** – poskytující data v trojrozměrném prostoru,
- **geometrické** – poskytují práci geometrickými typy objektů a umožňují jejich geometrické transformace a výpočty s nimi,
- **objektově funkční a editační** – povolují koncovým uživatelům přidávat, editovat a mazat objekty v geografických datech, které mohou být uloženy například v geodatabázi. Tento typ služeb pracuje převážně s vektorovými daty a jejich atributy,

- **obrazové** – poskytují rastrová data jako jeden obrázek nebo několik obrázků formou mozaiky,
- **geodatabázové** – umožňují přístup k datům uloženým ve speciálních geodatabázích, případně jejich synchronizaci,
- **vyhledávací** – zpřístupňují vyhledávací mechanismy v geoprostorových datech,
- **geokódové** – převádějí neprostorové informace do prostorových, především při práci s adresami,
- **katalogové** – starají se o indexaci metadat (dat o datech).

Všechny výše zmíněné služby poskytují klientům přístup ke geografickým zdrojům dat, ovšem každá z nich využívá jiné principy a postupy. Nyní si rozebereme podrobněji některé z nich.

### 3.2.1 Mapové služby

Umožňují klientům požadovat od služby mapy pro specifické území, které je definováno v požadavku. Server požadavek zpracuje a zašle klientovi data v požadovaném nebo dostupném formátu (JPEG, PNG, GIF, atd.). Ze všech výše jmenovaných geoprostorových služeb se jedná o tu nejpoužívanější. Navíc umožňují kromě zobrazování a prohlížení mapových kompozic provádět atributové dotazy nad zobrazenými daty nebo jejich reprojekci do jiného souřadnicového či projekčního systému.

Mapové služby mohou poskytovat data buď dynamicky, nebo z kešovaných dat formou dlaždic. Tyto dlaždice jsou dostupné jen v určitých měřících a s předem definovanými souřadnicemi, které musí klient znát. Velikou výhodou je pak rychlost, se kterou jsou data doručována zpět ke klientovi. Způsob použití se omezuje na mapové podklady, které jsou neměnné, použitelné především jako podkladové mapy. Dynamické mapy je potřeba vykreslit s každým příchozím požadavkem klienta, jelikož se mohou v čase či konstantě měnit, a proto je jejich předání směrem ke klientovi pomalejší než u kešovaných dat.

### 3.2.2 Objektově funkční a editační služby

Z anglického Feature editing services, kde se jedná o služby umožňující uživatelům přidávat, editovat a mazat geografické objekty, které jsou uloženy v uživatelově vzdálené geodatabázi. Tyto úkony je možné provádět například skrze webového klienta, který poskytuje sadu základních nástrojů pro kreslení, úpravu a mazání bodů, linií a polygonů, nebo rozšířené nástroje pro rozdělování či slučování polygonů a ořezávání či rozšiřování linií. Uživatelé navíc mohou u objektů upravovat a přidávat nové atributy, nebo připojit další soubory, např. digitální fotografii.

### 3.2.3 Vyhledávací služby

Slouží k indexování a vyhledávání ve zdrojích geoinformačních systémů (například datové vrstvy, tabulky, nebo v celé podnikové databázi). Toto ocení především uživatelé, kteří nemají přístup do intranetu dané organizace, a tak využívají těchto služeb k vyhledání požadovaných GIS zdrojů.

### 3.2.4 Obrazové služby

Poskytují přístup k rastrovým datům, jako jsou data získaná z leteckého nebo družicového snímkování. Podporují surová data, jejich extrakci a následné stažení ke klientovi pro možnosti častého zobrazování. Umožňují publikovat obří množství datových kolekcí bez předchozího zpracování. Za běhu podporují vytvoření mozaiky z dat a vícenásobné reprodukce obrazu z jednoho datového zdroje. Všechny tyto operace pak poskytují snadnou a rychlou distribuci velkých objemů rastrových dat do několika klientských aplikací zároveň.

### 3.2.5 Geokódové služby

Obsahují operace pro převod adres na prostorový údaj. Pro geografické znázornění bodu, kde se adresa nachází, se ve většině případů používá zeměpisné šířky a délky. Nejde o úplně triviální službu, jelikož zde nastávají komplikace při vyhledávání nejednoznačných adres nebo adres, které nejsou v databázi obsaženy. V tomto případě je jako výsledkem geokódové služby seznam podobných možností.

Existuje celá řada fungujících geokódových služeb, které poskytují firmy jako Google, Microsoft, Yahoo, Seznam, Atlas, atd. Mnohdy mají otevřené API, které můžeme dále využít ve svých aplikacích.

### 3.2.6 Geoprocenční služby

Umožňují sdílení funkcionalit a postupů výpočtu určitých druhů operací navržených člověkem, které jsou zpracovávány na straně serveru. Uživatel tedy zadá vstupní data do požadavku na službu (např. označí polygon v mapové kompozici). Služba se postará o provedení sledu operací a výsledek (např. výpočet sklonitosti terénu) je předán zpět klientovi, který se postará o jejich prezentaci.

## 3.3 Testování služeb

Jelikož služby poskytují rozsáhlé množství dat, je důležité kontrolovat, zda údaje poskytované uživatelům webové služby jsou správné a dávají pravdivé výsledky. To se provádí za pomoci testování použitelnosti služby, a provádí se za účelem zvýšení funkčnosti a spolehlivosti.

Testování výkonu je druhým pohledem na testování. Testuje se postup operací, prováděných na serveru při vykonávání určitých úloh, aby se zjistilo, jak rychlé jsou výstupy ze systému na základě určitých aspektů. Pokud se zjistí nedostatky, je třeba, aby služba byla optimalizována.

Údaje jsou mnohdy citlivé povahy a jsou vyžadovány v mnoha organizacích na celém světě pro každodenní práci. Geoinformační systémy tak přijaly pomyslnou výzvu k poskytování těchto služeb nabízejících obrazová data a funkcionality spojenou s geodaty.

Testování výkonu může sloužit různým účelům:

- může prokázat že systém splňuje určitá kritéria,
- je možné porovnat dva rozdílné systémy a zjistit tak, který si vede lépe,
- nebo můžeme změřit výkonnost určité části systému či její vliv na celkovou pracovní zátěž služby.

Tyto úkony nám pak pomáhají zjistit nedostatky, a vybudují základní informace pro programátory, kteří mají za úkol systém upravit (optimalizovat).

## 4 INTEROPERABILITA, STANDARDY A JEJICH IMPLEMENTACE V GIS SLUŽBÁCH

Současné trendy v geoinformačních systémech směřují čím dál více k užívání distribuovaných systémů na bázi webových služeb. V tomto kontextu můžeme hovořit o takzvané interoperabilitě, která ve své podstatě představuje prostředek pro integraci nesourodých dat s různými datovými formáty a odlišnými systémovými platformami, používané při jejich vzájemné komunikaci. Umožňuje vzájemnou kooperaci rozličným softwarovým produktům, které těchto možností využívají skrze standardy webových služeb.

V této kapitole budeme dále hovořit o organizaci OGC, která se nejvíce zasloužila při vytváření a definici jednotlivých standardů a služeb, které jsou používány v současném průmyslu GIS. Jmenovitě si následně popíšeme a probereme následující z nich: WMS, WFS, WCS, WPS, KML, GeoRSS a GML.

### 4.1 Open Geospatial Consortium (OGC)

Jedná se o neziskovou mezinárodní organizaci, zabývající se standardy a službami specializované na prostorová data. Je založená na dobrovolné shodě, a v době psaní tohoto textu má 447 aktivních členů skládajících se z vládních, komerčních, nevýdělečných a vědeckých organizací a obsahuje 56 OGC standardů. Organizace byla založena v roce 1994. Ve svých počátcích byla známá spíše pod názvem Open GIS Consortium. V roce 2000, kdy byla publikována specifikace WMS, získalo OGC na síle a to vedlo k rychlému zavedení webových geoprostorových služeb v geoinformačních systémech a tak kýžené interoperabilitě.

Mnoho z navržených OGC specifikací je založeno na obecné architektuře, jenž jsou popsány v dokumentech, které bývají označovány jako obecné specifikace (Abstract Specification). Tyto specifikace mají za úkol popsat základní datový model, který se stává základem pro většinu běžných rozvojových aktivit OGC. Tyto specifikace pak umožňují interoperabilitu mezi různými druhy prostorových systémů, a jsou základem, ze kterého vycházejí všechny implementační specifikace OGC.

OGC dále poskytuje takzvané referenční modely (OGC Reference Model), které stanovují rámec pro pokračující práci OGC. Popisují základy standardů, zaměřující se na jejich další vztahy mezi ostatními standardy.

Další druh dokumentů, které OGC nabízí, jsou osvědčené postupy (Best Practices Documents), které obsahují diskuzi o osvědčených postupech při používání nebo implementaci OGC přijatých dokumentů. Jsou jejich oficiálním stanoviskem a představují tak souhlas s obsahem dokumentu.

Všechny výše uvedené dokumenty a další informace jsou dostupné webu OGC viz [6].

## 4.2 Ostatní standardy

Mezi organizace, zabývající se standardizací, které napomáhají interoperabilitě patří:

### 4.2.1 ISO/TC 211

(International Organization for Standardization, Technical Commit) Jedná se organizaci, která byla založena roku 1994, a zabývá se normalizační činností v oblasti geografických informací a geomatiky. Jde o technickou komisi, která se stará o vydávání norem a technických zpráv řady ISO 19100. OGC a ISO/TC 211 mezi sebou úzce spolupracují, při vydávání mezinárodních standardů, ale OGC se stará spíše o praktická řešení, která implementuje na určitých použitelných řešeních, kdežto ISO se zabývá problematikou více abstraktněji a řeší základy standardizace. Tyto základy z norem ISO pak OGC využívá ve výše zmíněných abstraktních specifikacích. Těchto norem využívá i ČSN (Československá norma, soubor českých technických norem), která se postarala o překlad anglických norem do českého jazyka. Naší problematikou služeb se zabývají převážně dvě normy. První je ČSN ISO 19116 (Geografická informace - Polohové služby) a druhá obecnější ČSN ISO 19119 (Geografická informace - Služby).

### 4.2.2 W3C

The World Wide Web Consortium je hlavní mezinárodní standardizační organizace pro web. Byla založena roku 1994 Timem Berners-Leeem, který je považován za otce internetu. W3C má za úkol řešit problémy nekompatibility, které mohou vzniknout v internetových technologiích od různých distributorů. Mezi různé standardy W3C patří například HTML, XML, XHTML (Extensible HTML), CSS, XML schémata, SOAP a WSDL, které jsou v široké míře používány prostorovými webovými službami. W3C vynalezla a standardizovala jeden ze souborových formátů pro popis vektorové grafiky nazvaný SVG (Scalable Vector Graphic), o kterém v budoucnosti ještě mnoho uslyšíme. Je to právě on, který je jedním ze stavebních kamenů HTML verze 5. V GIS systémech má tento formát využití na zobrazování vektorových mapových podkladů v různých webových prohlížečích a mobilních, ale i desktopových klientech.

### 4.2.3 INSPIRE

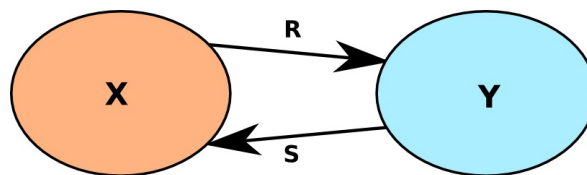
(Infrastructure for Spatial Information in Europe) je stejnojmenná směrnice Evropské komise a rady, která má cíle k vytvoření legislativního rámce potřebného pro vybudování infrastruktury prostorových informací ve státech EU. Tato směrnice byla vydána a vešla v platnost v roce 2007, kde v průběhu následujících dvou let proběhla implementace do národních legislativ jednotlivých států. Do České republiky byla převedena 23.10.2009 formou novely zákona č. 123/1998 Sb., o právu na informace o životním prostředí. Vznikl i plán, pomocí něhož by následná implementace INSPIRE měla být

proveditelná do roku 2013. Mezi cíle směrnice patří snaha o efektivní sběr, uložení a sdílení prostorových dat; možnosti bezproblémové kombinace těchto dat z odlišných zdrojů; prostorová data tvořit na státní úrovni formou státní správy; dostupnost prostorových dat za podmínek, které nebudou omezovat jejich využití; zjednodušit vyhledávání mezi prostorovými daty se zpřístupněním podmínek, za kterých mohou být data využita.

### 4.3 Interoperabilita podle ČSN ISO 19119

Interoperabilita je schopnost komunikovat, provádět programy nebo přenášet data mezi různými funkčními jednotkami tak, že uživatel nemusí mít žádnou nebo jen malou znalost konkrétních vlastností těchto jednotek.

Dvě komponenty X a Y (viz obrázek č.7) mohou interoperovat (jsou interoperabilní), jestliže X může posílat Y požadavky R na služby, založené na vzájemném porozumění požadavku R komponentami X a Y, a jestliže Y může podobně vracet X vzájemně srozumitelné odpovědi S. To znamená, že dva interoperabilní systémy mohou společně



Obr. 7. Interoperabilita

interagovat k provedení úkolů. Pro geografickou doménu platí následující popis termínu „geografická interoperabilita“:

„Geografická interoperabilita“ je schopnost informačních systémů 1) volně vyměňovat všechny druhy prostorové informace o Zemi a o objektech a jevech na, nad a pod zemským povrchem; a 2) v součinnosti zpracovávat software schopný manipulovat s takovou informací v sítích, viz [5]

### 4.4 Web Map Service (WMS)

Jedná se o webovou mapovou službu definovanou OGC 19.4.2000 ve verzi 1.0.0, a posléze standardizovaná organizací ISO jako norma ISO 19128, která dynamicky produkuje mapy z geografických informací skrze internet/intranet. Tato OGC norma definuje „mapu“ k vyobrazení prostorových informací v počítačově založených systémech formou obrazového souboru. Poskytovaná data z této služby bývají v jednom z mnoha rastrových grafických formátů, jako jsou PNG, JPEG, GIF, TIFF, BMP, ale příležitostně i vektorově založených, například dříve zmíněný SVG nebo WebCGM (Web Computer Graphics Metafile). Postupným vývojem se WMS rozvíjelo, měnily se některé specifikace a další verze nesly označení 1.0.7, 1.1.0, 1.1.1 a doposud nejnovější

1.3.0, která byla vydána 15.3.2006. Po usilovném hledání na internetu, která z dosavadních verzí je nejrozšířenější skončila výsledkem pro verzi 1.1.1 jež byla publikována 16.1.2002. Nejnovější verze WMS serverů 1.3.0 vesměs umí odpovědět i verzi starší, jako je 1.1.1, ovšem některé servery určité parametry ignorují, nebo je podávají s názvem z verze 1.1.1, ale hodnotou pro verzi 1.3.0.

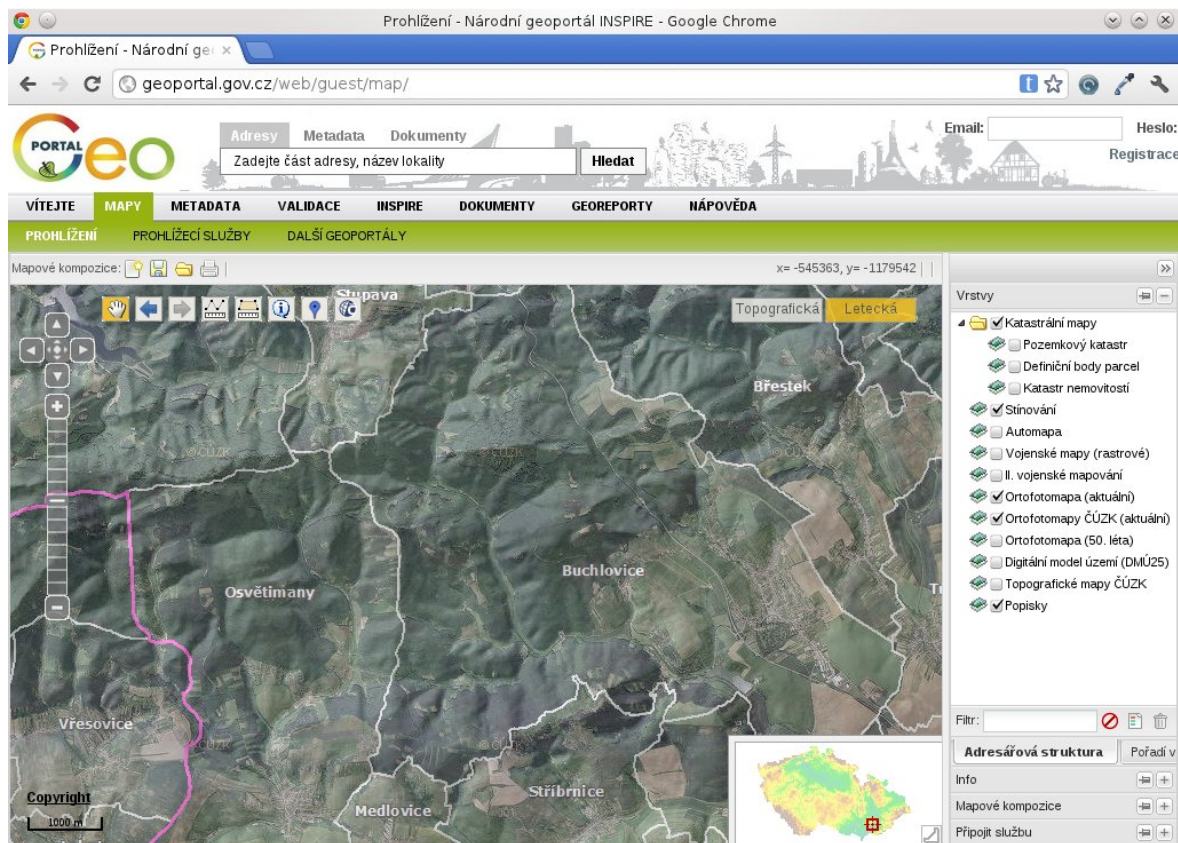
Ve WMS jsou definované tři základní operace GetCapabilities, GetMap a GetFeatureInfo (v následujícím pořadí budou dále rozepsány), které jsou určené parametrem „Request“, sloužících k vrácení:

- **metadat**, kdy služba vrací informace jak o sobě samé, tak o datech, která je schopna nabídnout. Tyto informace jsou ve formátu XML. Konkrétním příkladem těchto údajů může být titulek a popis služby, kontaktní informace o poskytovateli, podporovaných operacích, o datových vrstvách a jejich vlastnosti, jako jsou souřadnicový systém, abstrakt, klíčová slova atd.,
- **map** neboli obrazových dat geografického charakteru, která mohou být v různých grafických formátech, jenž byly zmíněny výše.
- **informací o objektu** je volitelná operace služby, která poskytuje informace o prvku, který je umístěn v mapové kompozici.

Veškeré operace služby jsou přístupné skrze URL adresu, jejichž obsah je závislý na požadované operaci. Pokud od služby požadujeme mapu, musíme v URL adrese uvést veškeré údaje, které nám identifikují danou oblast a požadovaná data, která chceme v mapě zobrazit. Těmito údaji jsou:

- **ohraničující oblast**, z anglického „boundingbox“, je oblast, která je definována souřadnicemi dvou bodů, které reprezentují levý spodní a pravý horní roh oblasti,
- **souřadnicový systém**, který definuje, v jakém souřadnicovém systému jsou uvedené souřadnice ohraničujícího boxu, a v kterém budeme požadovat výslednou mapu,
- **rozměry**, které definují šířku a výšku požadovaného obrázku udané v pixelech,
- **obrazový formát** specifikující typ souboru, ve kterém chceme data,
- **vrstvy**, jenž reprezentují různé typy mapových podkladů, které služba nabízí. Vrstev může být definováno více, server se postará o jejich poskládání do výsledné mapové kompozice, která je vrácena klientovi jako výsledek operace,
- **průhlednost** u některých obrazových formátů umožňuje transparentnost pozadí.

Jelikož se tato diplomová práce věnuje WMS serverům, další podrobnosti a postupy používané s touto službou budou zmíněny v praktické části. Čtenáře, kteří by hledali podrobný popis služby odkazujeme na [7], kde nalezne všechny potřebné informace.



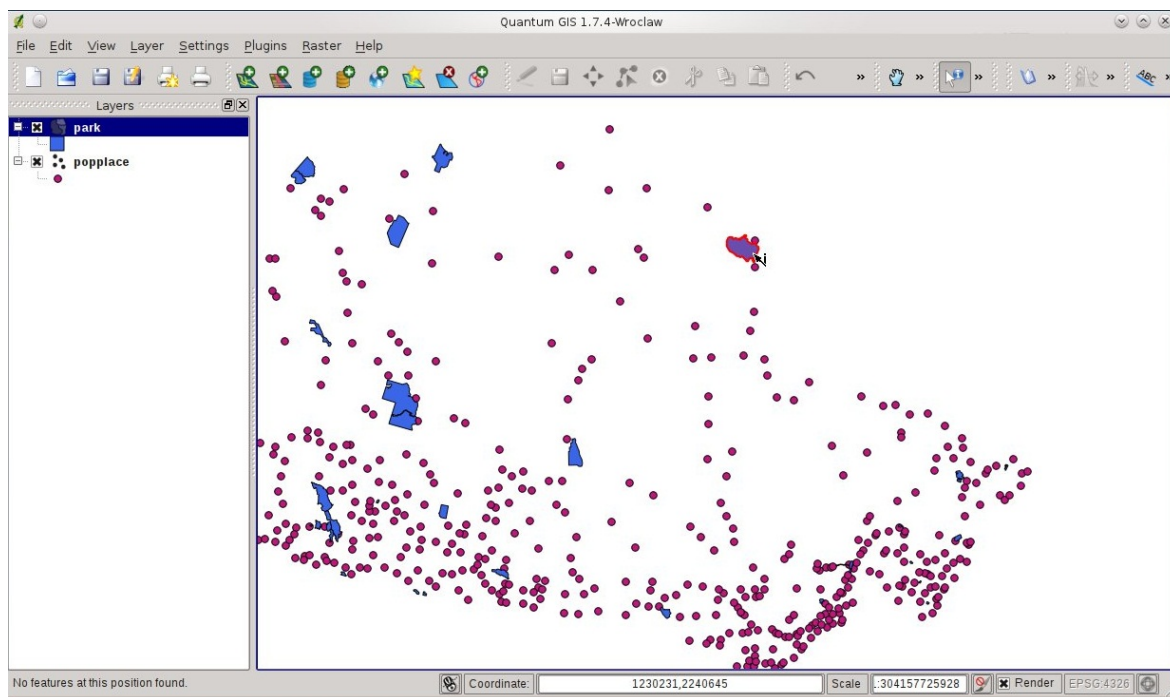
Obr. 8. Obrázek připojených dat z WMS na tenkém klientovi ve webovém prohlížeči

#### 4.5 Web Feature Service (WFS)

Jednou z dalších specifikací OGC, týkajících se webově orientovaných služeb, je WFS. Jedná se o službu, která umožňuje vytváření, editování a sdílení geografických objektů ve vektorovém formátu. Překlad slova „feature“ znamená „prvek“. V této službě však představuje spíše vektorový geoprostorový objekt. Nejedná se o klasický přístup na úrovni souborů za pomoci souborových protokolů, ale o přímý přístup ke geografickým objektům pomocí internetu/intranetu. Klientům je tak umožněno bez stažení souboru načtení jen potřebných údajů k danému území, se kterým pak mohou začít manipulovat. Dále jim jsou povoleny vykonávat operace umožňující vkládat, upravovat, mazat a provádět dotazy nad objektově prostorovými daty, nacházejícími se na serveru. Poprvé byla publikována ve verzi 1.0.0, data 17.5.2002, a tak se jedná o dva roky mladší službu než je WMS. Dnes je aktuální verze 2.0 z 2.11.2010. Také ji můžeme nalézt ve standardu organizace ISO jako normu ISO 19142. Služba umožňuje 11 druhů operací, které si stručně popíšeme:

- **GetCapabilities** žádá metadata služby. Odpověď je ve formátu XML, která popisuje obsah a možnosti služby, zahrnuje informace o objektových typech poskytovaných službou a operace, které podporují,
- **DescribeFeatureType** žádá strukturu objektového typu podporovaného WFS,
- **GetPropertyValue** operace umožňující z datového úložiště získat hodnotu z vlastnosti objektu, nebo jeho části, sloužící k nastavení identifikace objektu při používání dotazovacích výrazů,
- **GetFeature** se stará o načtení geoobjektů a jejich atributů tak, aby odpovídala filtrovacímu dotazu,
- **GetFeatureWithLock** je funkčně podobná operaci GetFeature s rozdílem, že při dotazu na tuto funkci WFS negeneruje jen odpověď jako v GetFeature, ale musí navíc zamknout objekty ve výsledné sadě. Toto omezení zřejmě slouží k aktualizaci objektů v dalších spuštěných transakcích,
- **LockFeature** Webová spojení jsou ze své podstaty nestavové a z tohoto důsledku nejsou serializovatelné transakce dochovány. Tyto problémy jsou spojené s operací aktualizace objektů. Jestliže klient načte instanci objektu, ta je pak modifikována klientem a následně poslána zpět k WFS skrze transakční požadavek o aktualizaci dat. Serializace je ztracena, jelikož není nikde zaručeno, že nemohl přijít jiný klient, co přistoupil ke stejným datům v databázi, upravil je a poslal zpět k WFS dříve než klient předešlý. Jedním ze způsobů jak serializaci uchovat je, aby přístup k datům byl možný jen skrze transakce, kdy jedna transakce získá přístup k datům a pak již žádná jiná nesmí změnit žádnou položku v datech používané při první transakci. To je umožněno skrze zámek objektu, který řídí právě tato operace LockFeature,
- **Transaction** tato operace je používána k popisu výše zmíněných transakčních operací, jako jsou vytváření, modifikace, nahrazení a mazání, které jsou aplikovány na objekty WFS,
- **CreateStoredQuery** umožňuje vytváření uložených dotazů ve WFS,
- **DropStoredQuery** umožňuje mazání uložených dotazů ve WFS,
- **ListStoredQueries** vypisuje dotazy, které jsou ve službě WFS k dispozici,
- **DescribeStoredQueries** tato operace poskytuje detailní metadata o dotazech, které jsou uloženy ve službě WFS.

Další podrobné specifikace a informace o této službě jsou dostupné v [8].



Obr. 9. Ukázka připojené služby WFS v programu Quantum GIS

#### 4.6 Web Coverage Service (WCS)

Tato služba se zabývá podporou elektronického získávání prostorových dat ve formě prostorové informace. Doslovný překlad slova „coverage“ je „pokrytí“, ale v této službě představuje výše zmíněnou prostorovou operaci, která představuje různé jevy v prostoru (proměnlivého v čase). Služba je implementací dalšího ze standardů OGC, který vznikl 27.8.2003 ve verzi 1.0.0 a nyní se nachází ve verzi 2.0.0 z data 27.10.2010.

Služba umožňuje získávat zmíněná prostorová data, nejčastěji pořízena formou satelitního či leteckého snímkování zemského povrchu, digitální údaje o nadmořské výšce a jiné jevy představující hodnoty v každém ze zaměřených bodů. Od služby WMS se liší ve dvou klíčových významech. První z nich je poskytování surových dat, zatímco WMS poskytuje již výsledné vizuální reprezentace dat. Druhým z nich se ukrývá ve formátu poskytovaných dat. WMS poskytuje rastrová data, WFS vektorová data, zatímco WCS se stará o poskytování přístupu k potenciálně detailních a objemných geoprostorových informací ve formě, která je optimalizovaná k vykreslování dat na straně klienta.

Data z WCS jsou klientovi navrácena s jejich původní sémantikou namísto obyčejného obrázku (statické mapy). Navíc služba definuje bohatý syntaktický aparát pro dotazování na tyto data, proto je možné je například interpolovat či extrapolovat atd., nejen zobrazit.

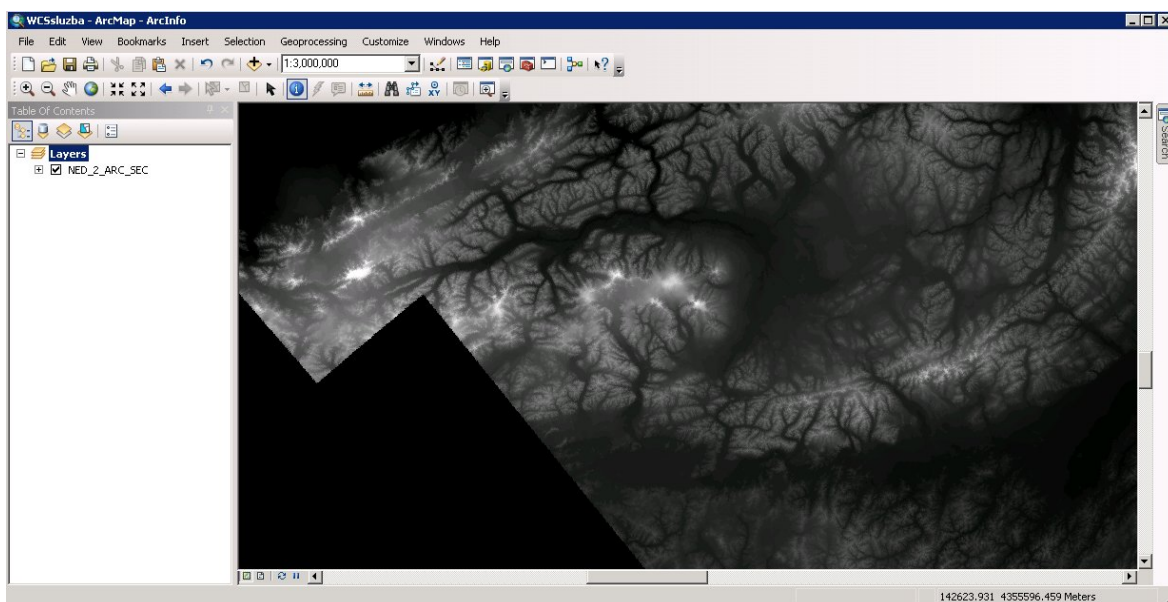
Tato služba nabízí tři operace, které si následně popíšeme:

- **GetCapabilities** operace, která dovoluje získávat informace o službě formou

metadat, jenž jsou zaslány ve formátu XML jako odpověď klientovi.

- **GetCoverage** je operací umožňující získávání podmnožin prostorových informací dané geografické oblasti za vybrané časové období s definovaným souřadnicovým systémem a požadovaným datovým formátem (GeoTiff, HDF-EOS, NITF). Služba tento požadavek zpracuje a vrací výsledná data formou odpovědi zpět klientovi.
- **DescribeCoverage** umožňuje klientovi zažádat o plný popis prostorových informací, které WCS nabízí. Je možné žádat informace jen o jednotlivých prostorových informacích, nebo jich můžeme žádat v dotazu více zároveň. Jakmile server požadavek zpracuje, podá klientovi odpověď ve formátu XML.

Další podrobné specifikace, a informace o této službě jsou dostupné v [9].



Obr. 10. Ukázka připojených dat z WCS v programu ArcGis

#### 4.7 Web Processing Service (WPS)

I další ze služeb je definovaná standardy OGC, zabývající se poskytováním pravidel pro standardizované vstupní a výstupní data pro geoprostorové služby nabízející operace (procesy) nad geodaty. Tyto operace mohou být jakýmkoliv algoritmem, výpočtem nebo modelem. Tato standardizace pak umožňuje, aby informace byly strojem čitelné a využitelné, stejně jako jsou metadata snadno čitelná a využitelná člověkem. Historie této služby se váže k datu 8.6.2007, kdy byla vydána jako specifikace OGC ve verzi 1.0.0, která je doposud aktuální.

WPS může nabízet nějaký druh funkcionality geoinformačních systémů pro klienty skrze síťovou infrastrukturu. Může obsahovat různé naprogramované výpočty, výpočetní modely, které pracují s prostorovými daty (rastrovými, vektorovými). Tyto služby mohou nabízet jak jednodušší operace, například porovnávání dvou datových vrstev ze stejného území z různých časových okamžiků nebo komplexnější rozsáhlé analýzy.

Specifikace jsou navrženy tak, aby umožnily poskytovateli těchto služeb vystavit webově přístupný proces, do kterého mohou uživatelé jednoduchým způsobem vkládat svoje prostorová data bez potřeby znalosti serializace atd. Tento webový proces může být přístupný např. přes webového klienta nebo formou API, které mohou využít i jiní klienti. Standardizuje tak celý proces, popisuje jeho vstupy a výstupy, to jak klient může žádat spuštění procesu, i jak je výsledek procesu zpracovat.

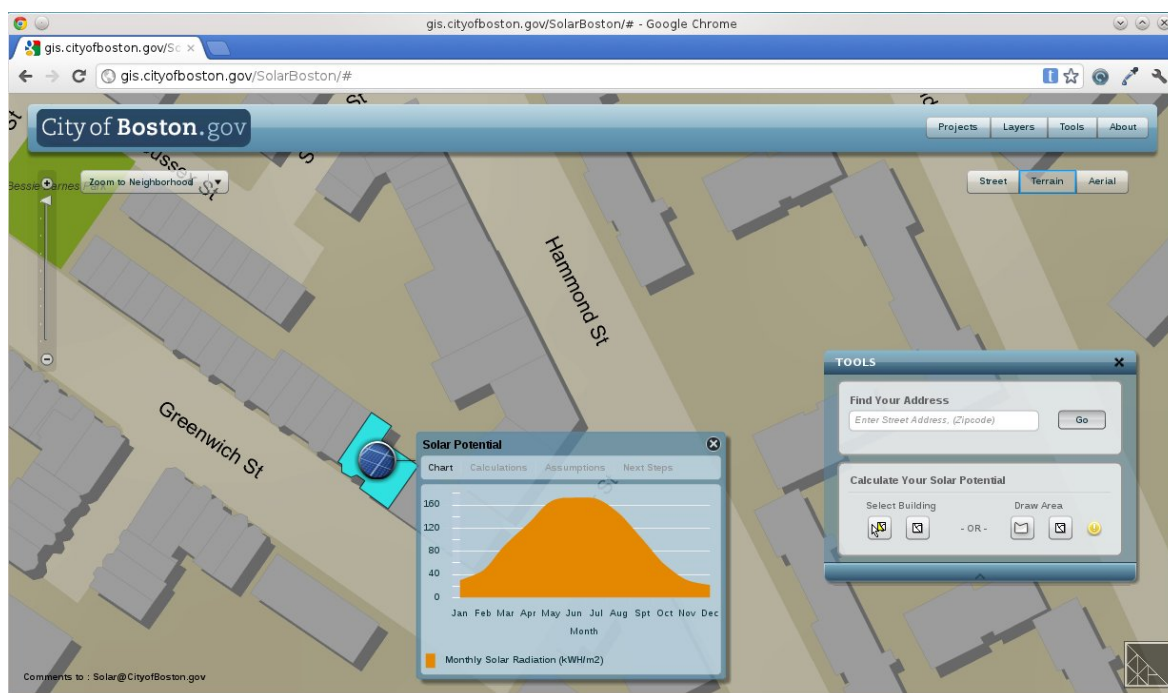
Jedná se o obecné rozhraní, které může být zaobaleno pro použití s ostatními OGC službami, které se zaměřují na práci s geoprostorovými daty a ve výsledku tak poskytnout například výkonnou webovou aplikaci s mnoha funkcionalitami.

Služba nabízí tři operace, které následně zmíníme a popíšeme jejich účel:

- **GetCapabilities** vrací metadata o službě ve formě dokumentu XML,
- **DescribeProcess** umožňuje WPS klientům žádat o celý popis jedné nebo více procesů, které mohou být spuštěny operací Execute. V popisu jsou zahrnuty informace o vstupních, výstupních parametrech a jejich formátech. Tento popis může být následně použit k vygenerování uživatelského rozhraní, které slouží ke sběru parametrů, potřebných pro následné spuštění procesu.
- **Execute** dovoluje klientům služby WPS spouštět specifikované operace, jež jsou ve službě implementovány. Při volání operace se používají hodnoty vstupních parametrů a jako výsledkem jsou výstupní parametry operace definované v popisu procesu z operace DescribeProcess. Parametry mohou být jednak vkládány přímo do volání operace, nebo mohou být předány formou odkazu na jiné umístění dostupné v síti. Jakmile se operace zpracuje a vykoná, je klientovi vrácen výsledek ve formátu XML s vloženým výsledkem, nebo je výsledek uložen na dostupné cestě v síti, kde XML odpověď je rozšířena o URL adresu pro každý výsledek uložený na serveru. Případně další možností, pokud výsledkem operace je jen jediný výstup, může být výsledek vrácen přímo bez zabalení do XML odpovědi.

WPS dále poskytuje mechanismus, kdy jen nečekáme na výstup od služby, ale můžeme sledovat průběh procesu, který je službou právě zpracováván. To může být například využito při zobrazení těchto informací uživateli ve stavové liště aplikace, kde může být vyobrazen např. procentuálně stav nebo odhadovaný čas, který zbývá do vykonání operace.

Na obrázku č. 11 je znázorněná implementace WPS služby, která je zpracována ve webovém klientovi ArcGIS for Flex. Tato ukázková aplikace umožňuje uživatelům vybrat oblast budovy, pro kterou je následně službou vypočítán solární potenciál v kWh/m<sup>2</sup> v průběhu roku, ze kterého je pak uživateli zobrazen graf. Na straně tohoto klienta jsou pak s těmito daty umožněny další výpočty na základě úpravy parametrů uživatelem, jako je využitelná plocha střechy pro solární panely v procentech, která vede k zobrazení aktuálních údajů, jen pro nastavenou využitelnost střechy. Pro více informací o WPS čtenáře odkazujeme na [10].



Obr. 11. Ukázka aplikace města Boston zpracované v ArcGIS Viewer for Flex

#### 4.8 Keyhole Markup Language (KML)

KML je gramatikou jazyka XML, používanou k reprezentaci, ukládání a modelování geodat jako jsou body, linie, polygony, ale i různé obrázky či prostorové 3D modely, které mohou být zobrazeny v různých typech aplikací (např. Google Earth či Google Maps, atd.). Určuje jak zakódovat co zobrazit a určuje jak to zobrazit.

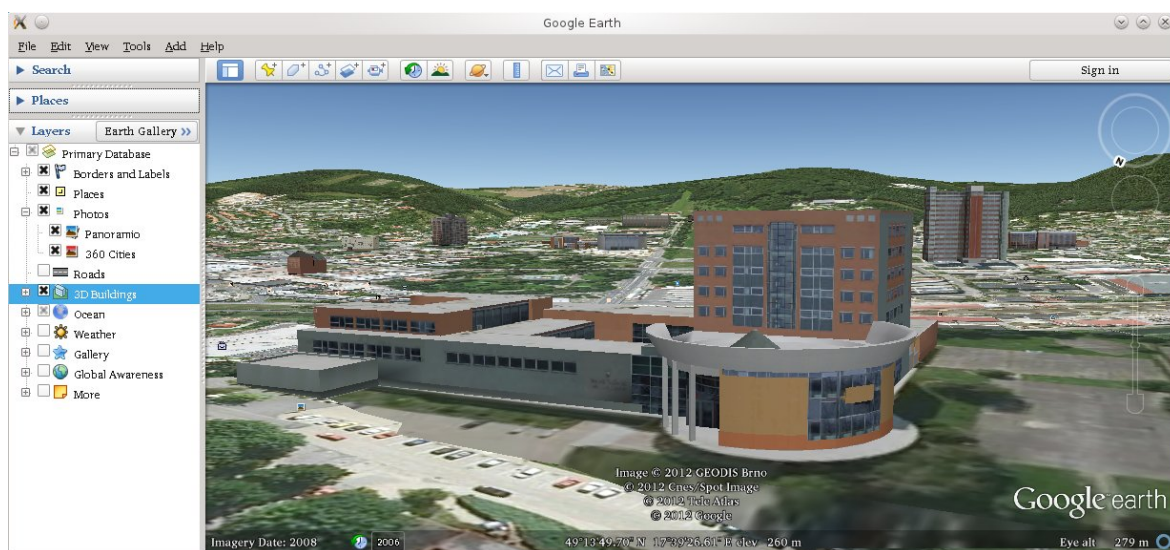
Uživatelé, kteří KML používají, jej využívají k označení různých míst, ke kterému mohou přiložit fotografii, dále např. pro zaznamenání cesty, která vznikla turistickým nebo cyklo výletem. Tyto údaje získaly oblibu díky snadnému šíření výše zmíněných informací, které jsou často sdíleny přes různé druhy sociálních sítí. Vědci je často používají k poskytování detailního mapování různých zdrojů a modelů, zaznamenávání zemětřesení, nerostných ložisek, atd. Je využíván realitními makléři, architekty a městskými úřady k účelům vytvoření vizuálních plánů. KML se nyní nachází ve verzi 2.2.0,

pod kterou byl dne 14.4.2008 přijat jako standard OGC. Pro další informace o tomto standardu, čtenáře odkazujeme na [11].

Ukázka jednoduchého zápisu KML:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <name>FAI - UTB</name>
      <description>Škola kde jsem studoval</description>
      <Point>
        <coordinates>17.6574906,49.2306014</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Na následující obrázku č. 12 je zobrazena aplikace Google Earth, s použitím KML je vyobrazen 3D model budovy FAI UTB.



Obr. 12. Ukázka KML v praxi v aplikaci Google Earth

## 4.9 Geography Markup Language (GML)

Jedním z dalších velice rozšířených standardů, definovaných OGC ve verzi 3.2.1 je právě GML. Nyní jsou rozšířené specifikace na verzi 3.3.0 z data 16.1.2012 a jeho počátky sahají ke dni 12.1.2000, kdy byl poprvé uveřejněn ve verzi 1.0. Ani organizace ISO nezahálela a GML se stal standardem s označením ISO 19136 v roce 2007.

Jedná se o gramatiku jazyka XML, sloužící k vyjádření geografických objektů v unifikovaném záznamu, jejich modelování, přenosu a ukládání. Může popisovat informace

jako jsou: geometrie prostorových objektů, jejich vlastnosti, souřadnicový referenční systém (EPSG), topologii, prostorovou informaci. V geografických obrazových souborových typech je použito právě GML pro popis uvedených informací, který je vložen přímo v souboru.

GML je definován pomocí tří XML souborů, které definují atributy, pořadí elementů, jejich počet a kde se mohou jednotlivé elementy vyskytovat. Nyní si tyto soubory ve stručnosti uvedeme a popíšeme:

- **geometry.xsd** jedná se o geometrickou složku, jenž se stará o popis vrstev a jejich atributy,
- **xlink.xsd** popisuje odkazy mezi různými elementy a dokumenty, pomocí jazyku, který se nazývá XLink,
- **feature.xsd** starající se o popis jednotlivých geometrických prvků.

Máme 5 základních prvků, které GML využívá. Patří mezi ně bod, linie, pravoúhelník, uzavřená linie a polygon. Abychom mohli následně vytvořit grafický výstup, je třeba jej transformovat do jazyka, který bude grafické entity popisovat. K tomuto účelu se používá XSLT (Extensible Stylesheet Language Transformations) s příslušným stylovým procesorem (Saxon, Xalan). Výsledným souborem pak může být například vektorový formát SVG, VML (Vector Markup Language), X3D (eXtensible 3D graphic), ale také textové informace v podobě HTTP, RTF, PDF [13].

Náhradou GML, může být pro některý druh informací např. KML, které je výrazně jednodušší. Pro podrobnější specifikace, čtenáře odkážeme na [12].

#### 4.10 GeoRSS

Aby čtenář pochopil co je GeoRSS, je důležité, aby měl povědomí o RSS (RDF Site Summary). RSS je další z formátů rodiny XML určený pro čtení novinek na webu. Uživatel, který využívá RSS, si do své čtečky (aplikace, která podporuje čtení RSS) načte „RSS feed“, který představuje zdroj (kanál). Čtečka pak umožní uživateli jednoduché zobrazení všech novinek z webových zdrojů, které má načtené, bez nutnosti prohledávání svých oblíbených webových stránek.

GeoRSS je také založený na formátu XML a slouží k rozšíření klasického RSS o přídavnou hodnotou geografické informace. Dělí se na tři následující specifikace:

- **W3C Geo** již zastaralá specifikace, které může jen popisovat body v souřadnicovém systému WGS84,
- **GeoRSS-Simple** může popisovat základní geometrii (body, linie, obdelník, polygon) a další vlastnosti (objektový typ, jméno objektu, vztahy mezi tagy, výšku

a poloměr). Slovo „Simple“ v jeho jméně napovídá, že byl navržen pro to, být jednoduchým. Souřadnicový systém, který používá je vždy WGS84,

- **GeoRSS-GML** je více rozsáhlejší než výše zmíněná specifikace „Simple“, především v možnosti využití vlastností GML, např. v podpoře jiných souřadnicových systémů než WGS84.

Více informací o tomto standardu naleznete na webu [14].

## **II. PRAKTICKÁ ČÁST**

## 5 POUŽITÉ TECHNOLOGIE

V implementované aplikaci využíváme cloudové řešení Google App Engine. Nyní si popíšeme, co GAE (Google App Engine) vlastně je.

### 5.1 Google App Engine

Jedná se o platformu pro vývoj webových aplikací a služeb v cloudovém prostředí společnosti Google. Tyto aplikace (služby) je snadné vytvořit, udržovat a jednoduše rozšiřovat jejich prostředky na základě rostoucího síťového provozu nebo potřeb zvětšení datového úložiště. S GAE se nemusí uživatel starat o žádnou údržbu serverů, stačí jen aplikaci nahrát a je okamžitě připravena sloužit uživatelům.

Stejně jako v ostatních službách se za poskytující výpočetní výkon a zázemí pro cloudové aplikace platí. Platí se především za využití prostředků, které se měří. Následně je majiteli aplikace vypočítaná cena, kterou zaplatí poskytovateli. GAE ale umožňuje spustit aplikaci v cloudovém prostředí ihned, bez jakýchkoliv startovacích poplatků. V takovém případě, jsou ale zdroje omezeny na 1 GB uložených dat a výpočetního výkonu zhruba na zobrazení 5 mil. jednoduchých webových stránek za měsíc. Při aktivování placeného režimu hostování aplikací na GAE jsou tyto limity navýšeny a provozatel aplikace pak platí jen za prostředky, které byly využity nad rámec prostředků zdarma.

GAE podporuje aplikace napsané v programovacích jazycích Python (ve verzi 2.5 a od konce února 2012 i ve verzi 2.7), Java, a nyní experimentálně i v jazyce GO.

#### 5.1.1 Aplikační prostředí

Google App Engine umožňuje snadno a efektivně vytvářet aplikace, které spolehlivě běží i při velkém zatížení s velkým množstvím dat. App Engine obsahuje následující funkcionalitu:

- poskytování dynamických webů s plnou podporou běžných webových technologií, perzistentního úložiště dat s možností dotazování, třídění a transakcemi,
- automatické škálování prostředků a tzv. „load balancing“ (vyrovnání zátěže),
- API pro autentizaci uživatelů a posílání emailů za použití Google účtů,
- podpora plnohodnotného lokálního vývojového prostředí, které simuluje GAE na vašem počítači,
- fronty úloh sloužících k výkonání práce mimo rámec webových žádostí,
- možnost spouštět naplánované úlohy v určitých časech a pravidelných intervalech.

### 5.1.2 Zabezpečené prostředí

Aplikace běží v zabezpečeném prostředí, které poskytuje limitovaný přístup k základnímu operačnímu systému. Tato omezení dovolují GAE rozdělit webové dotazy na aplikaci skrze více serverů a spouštět či zastavovat servery ke splnění požadavků. Tento „sandbox“ aplikaci izoluje jen ve vlastním bezpečném a spolehlivém prostředí, které je nezávislé na hardwaru, operačním systému a fyzickém umístění.

Následně uvedeme několik příkladů, co zahrnuje zabezpečené prostředí:

- aplikace může přistupovat jen k počítačům na internetu, prostřednictvím poskytnuté URL adresy nebo emailové služby. Ostatní počítače mohou využívat aplikaci jen skrze HTTP nebo zabezpečený HTTPS protokol,
- aplikace nemůže zapisovat do souborového systému běhového prostředí. Aplikace může číst soubory, ale pouze ty, které byly nahrány společně s aplikací. Pro tyto účely můžeme využít souborové úložiště zvané „App Engine Datastore“, vyrovnávací paměť, zvanou „Memcache“ nebo jiné služby pro uchování dat mezi požadavky,
- aplikační kód se spouští jen v reakci na webové žádosti, úlohy zařazené ve frontě nebo naplánované úlohy a musí vrátit odpověď do 60 sekund. Žádná z obslužných rutin žádosti nemůže přerůst v subprocess nebo spouštět další kód po tom, co již byla odeslána odpověď.

### 5.1.3 Běhové prostředí Pythonu

V běhovém prostředí App Engine mohou být aplikace implementovány pomocí programovacího jazyka Python a spouštěny na optimalizovaném interpreteru tohoto jazyka. App Engine zahrnuje bohaté API a nástroje pro vývoj webových aplikací, včetně API s bohatými funkcemi pro modelování dat, snadno použitelného aplikačního webového frameworku a nástrojů pro přístup k aplikaci a jejím datům. Dále lze využít širokou škálu vespělých knihoven a frameworků pro vývoj webových aplikací Pythonem jako je Django.

Implicitní verze Pythonu je použita 2.5.2, ale je možné použít také verzi 2.7.2. Běhové prostředí zahrnuje základní knihovny jazyka Python, ovšem ne všechny, protože některé nemohou běžet v zabezpečeném prostředí GAE. Pro případy volání metod, které se snaží přistoupit k otevření soketu nebo zapsat soubor, je vyvolána výjimka. Některé hlavní funkce a moduly standardních knihoven, které nejsou podporovány, byly v provozním prostředí zakázány. Kód, který se je bude snažit naimportovat, také vyvolá výjimku. Pokud proběhne naimportování některé z externích knihoven třetích stran, nesmí tyto knihovny importovat žádné z nepodporovaných modulů a musí být napsané

čistě v Pythonu, protože rozšíření psané např. v jazyce „C“ nejsou z bezpečnostních hledisek dovoleny.

Toto prostředí dále poskytuje bohaté API pro ukládání dat, práci s Google účty, zpracování HTTP (HTTPS) požadavků a emailové služby. Dále nabízí jednoduchý webově aplikační framework nazvaný „webapp“ sloužící k jednoduchému budování webových aplikací.

#### 5.1.4 Úložiště dat

GAE poskytuje službu pro distribuované (NoSQL) úložiště dat nazvané „Datastore“, které poskytuje dotazování a možnosti transakcí. Stejně, jak distribuovaný webový server roste s provozem, tak distribuované datové úložiště roste s daty. GAE nabízí dva druhy těchto úložišť:

- **Master/Slave Datastore** – jedná se již o zastaralou formu úložiště, ve kterém je jedno datové centrum navrženo pro uchovávání hlavní kopie všech dat. Data jsou zapisována na tento hlavní server a následně jsou asynchronně replikována do ostatních vedlejších datových center,
- **High Replication Datastore** – ve kterém jsou data replikována napříč více datových center za použitím algoritmu „Paxos“.

App Engine Datastore není jako tradiční relační databáze. Datové objekty nebo entity mají určitý typ a sadu vlastností. Dotazy mohou načíst entity daného druhu, třídit je a filtrovat podle hodnot jejich vlastností. Hodnoty vlastností mohou být jedním z podporovaných datových typů (int, float, bool, str, datetime atd.). Datové entity jsou tzv. „schemaless“, což znamená, že jejich struktura je poskytována programovým kódem. Pythonovské prostředí obsahuje nástroje pro použití a vynucení struktury s danou aplikací, která pak přistupuje přímo do datového úložiště s využitím tak velké části této struktury, kterou právě potřebuje. Dotazy nad tímto úložištěm dat je možno tvořit pomocí třídy zvané „GQLQuery“. Toto rozhraní používá pro definici dotazu nad daty jazyk GQL, který je podobný jazyku SQL (Structured Query Language), ale navržený pro práci s GAE Datastore.

Jednou z novějších služeb pro uchování dat nazvanou „Google Cloud SQL“ je relační SQL databáze, založená na známém „MySQL RDBMS“. Tato služba však není plně otevřená a je povolena jen určitým poskytovatelům na základě žádosti zaslané společnosti Google, která ji může nebo nemusí vyhovět.

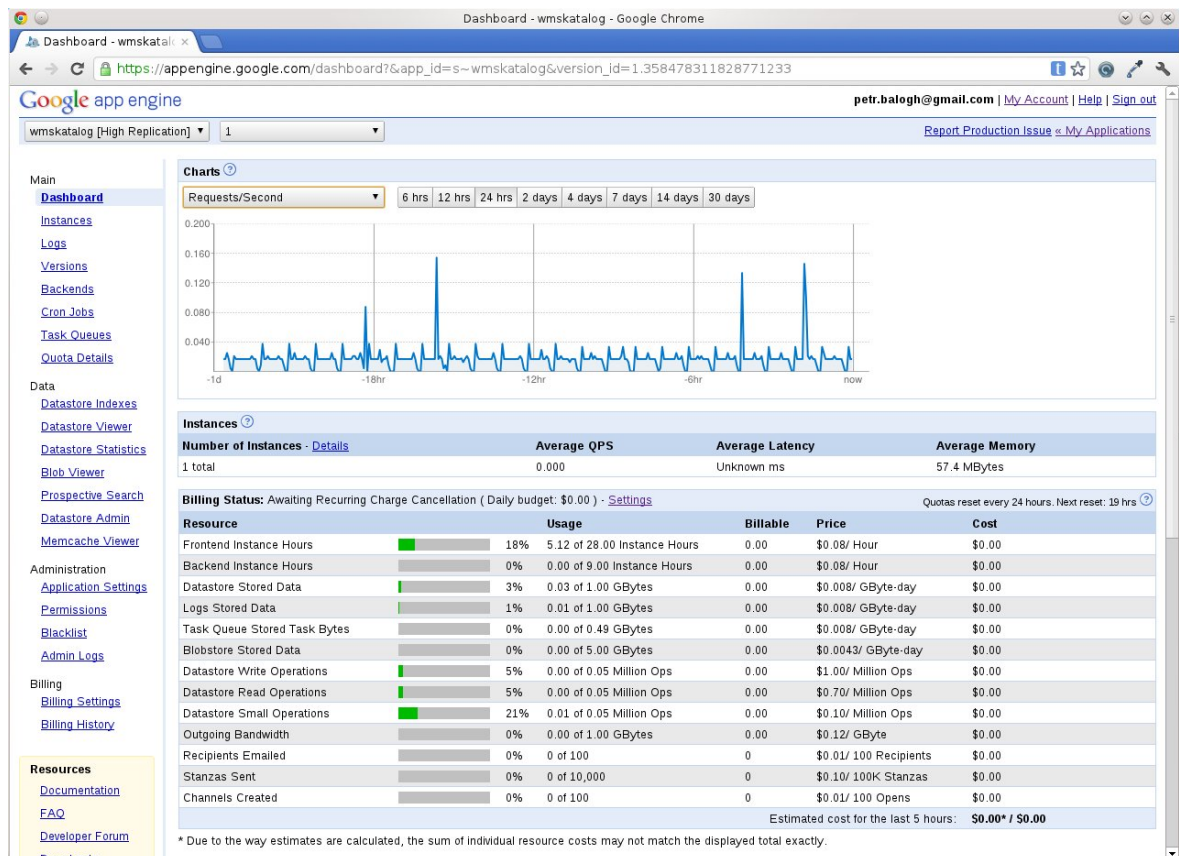
Další z možností uchování a přístupu k velkému objemu dat je „Google Cloud Storage“, který poskytuje uchovávání velkého množství dat v rámci terabajtů. Tato služba je dostupná zdarma s omezením úložiště dat na 5 GB a omezeným síťovým

provozem formou kvóty na 25/25 GB stahování/nahrávání, 30 tisíc požadavků HEAD, GET a 3000 PUT, POST, požadavků.

### 5.1.5 Kvóty a limity

Aplikace běžící na platformě GAE mohou využívat zdroje, které jsou omezeny kvótami. Aplikace tedy postupně konzumuje zdroje a jakmile dosáhne nastavené hranice, aplikace již nebude moci dále fungovat do dalšího vynulování kvót, které probíhá jednou za 24 hodin. Pokud se poskytovatel chce vyvarovat nedostupnosti své služby po vyčerpání prostředků, měl by aktivovat placený režim, ve kterém bude moci nastavit svůj maximální denní rozpočet. V případě aktivace tohoto placeného režimu je vždy účtována částka 2.10\$ za týden, viz [16].

Pro další informace a návody odkazujeme čtenáře na webové stránky [15].



Obr. 13. Ukázka nástroje pro správu aplikace hostované na GAE

## 5.2 OpenLayers

OpenLayers je knihovna napsaná v programovacím jazyce JavaScript, která slouží pro snadné vkládání dynamických map na jakoukoliv webovou stránku. Může zobrazovat

mapové dlaždice a různé popisky načtené z libovolného zdroje. OpenLayers bylo vyvíjeno pro použití všech druhů geografických informací a jejich reprezentaci ve webovém prohlížeči. Tato knihovna je dostupná zcela zdarma, vydána pod druhou klauzulí BSD (Berkley Software Distribution), licence známé jako „FreeBSD“.

OpenLayers je podporováno mnoha organizacemi po celém světě a je kompatibilní s většinou moderních webových prohlížečů. Výhodou je nezávislost na serverech, protože veškerý kód se spouští na straně klienta. Serverů se využívá pro stažení mapových podkladů, o jejichž vykreslení se stará právě OpenLayers. V této knihovně je implementováno mnoho JavaScriptových API pro tvorbu bohatých webově orientovaných aplikací v oblasti GIS. Jeho práce je obdobná jako u známých mapových prohlížečů firem Google, Microsoft, Seznam, Atlas atd., jen s rozdílem, že se jedná o software zdarma vyvíjený jako „opensource“.

Mezi základní vlastnosti OpenLayers patří:

- překrývání více mapových vrstev z různých standardů v jedné aplikaci do výsledné mapové kompozice,
- zobrazování dlaždic/obrázků z WMS, WFS, WCS a mnoho dalších standardů,
- zobrazování objektů definovaných s pomocí KML, GML, GeoRSS atd.,
- webově založená editace objektů např. služby WFS. OpenLayers obsahuje nástroje pro jednoduché editace a vytváření objektů, až po „snapping“ (přichycení k objektům) a rozdělování či slučování existujících objektů,
- použitelnost s ostatními JavaScript knihovnami (jQuery, Ext, Prototype, atd.),
- reprojekci na straně klienta,
- široká komunita uživatelů.

### 5.3 Ostatní technologie a knihovny třetích stran

Další použité technologie jen vyjmenujeme a nebudeme je dále rozepisovat, jelikož některé z nich byly popsány v teoretické části diplomové práce. Do použitých technologií patří: HTML, CSS, XML, JavaScript, AJAX, Python.

Následně zmíníme dvě knihovny třetích stran, které byly při budování výsledné aplikace použity.

#### 5.3.1 OWSLib

Jedná se o knihovnu jazyka Python, která spolupracuje s webovými službami standardu OGC a slouží k vývoji aplikací, jenž s těmito službami chtějí pracovat. Zahrnuje třídy, metody a vlastnosti pro práci se standardy WMS, WFS, WCS, ISO 19139 atd.



Obr. 14. Ukázka základní implementace knihovny OpenLayers

V této práci jsme využili především části pro spolupráci s WMS, kde je umožněno:

- načíst metadata služby z určeného serveru se zadanou URL adresou,
- získat seznam dostupných vrstev a podrobné informace pro jednotlivé vrstvy,
- vyvolat službou poskytovanou metodu „GetMap“, která umožňuje stáhnout mapový výřez se zadanými souřadnicemi a seznamem vrstev.

Pro další informace můžete navštívit stránky projektu, viz [17].

### 5.3.2 PagedQuery

Jedná se o třídu, která rozšiřuje dotazování nad GQLQuery objekty o možnosti stránkování výsledků, jež není v GAE příliš dobře řešené. Programátor může pomocí této třídy stránkovat výsledky z dotazů založených na GQL a uchovat si ukazatel, v místě kde skončil. Při dalším dotazu může pokračovat na datech následujících nebo předešlých. Pro další informace odkazujeme čtenáře na stránky projektu, viz [18].

## 6 NÁVRH A ZPRACOVÁNÍ APLIKACE

Úvod této kapitoly bychom rádi uvedli textem, který seznámí čtenáře se zadáním a výstupem praktické části diplomové práce. Podle zadání bylo úkolem vytvořit webovou aplikaci v cloudovém prostředí na platformě Google App Engine, kterou jsme popsali v části 5.1. Tato aplikace by měla sloužit k vytvoření uceleného seznamu (katalogu) dostupnosti webových služeb na standardu WMS v České republice. Tento katalog bude sloužit jako aktuální přehled pro ty, kteří budou chtít nalézt mapové podklady s určitou tematikou nebo pro určité zájmové území.

Aplikace má umožnit uživatelům registraci do systému a vybudovat tak komunitu lidí, kteří se budou starat o přidávání nových WMS serverů do tohoto katalogu. Úkolem takového seznamu WMS služeb je načíst potřebné informace formou metadat ze služby WMS do datového úložiště naší aplikace. Tyto údaje je třeba dále zpřístupnit uživateli v přehledné formě s přidanou hodnotou. Touto přidanou hodnotou máme na mysli funkcionalitu, která bude poskytnuta uživateli pracujícím s katalogem a následně mu ulehčí vyhledávání či prohlížení potřebných dat.

V zadání bylo vypsáno několik takových funkcionalit (služeb), jež by výsledná aplikace měla poskytovat. Tyto funkcionality si následně vyjmenujeme a popíšeme, co by měly umožňovat či poskytovat:

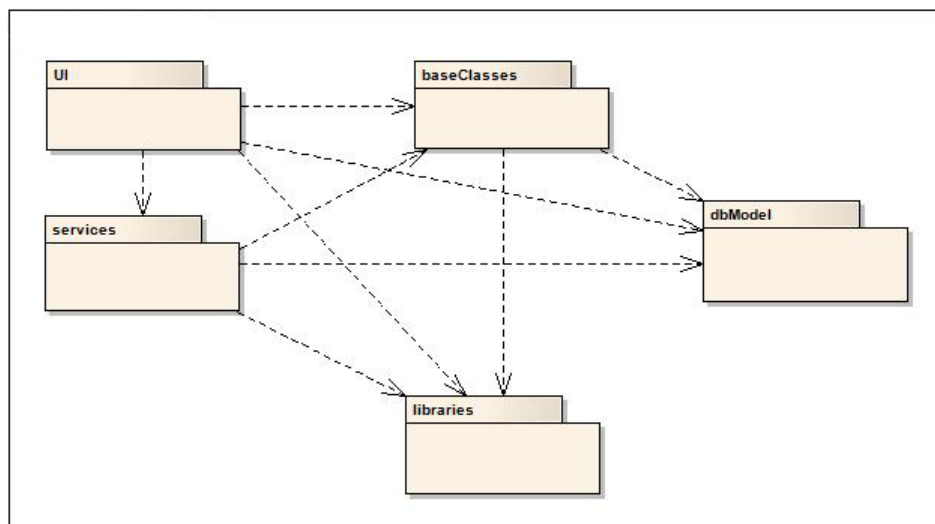
- **registrace uživatelů**, která bude moci být využita při vybudování výše zmíněné komunity lidí, používající GIS s mapovými podklady ze služeb WMS,
- **ověřování funkčnosti WMS serverů**, kde se znalostí OGC můžeme ověřovat, jestli dané servery fungují a umožnit tak informovanost uživatele, zda je daný server např. mimo provoz nebo v provozu. Tyto údaje je možné uchovávat v databázi a v budoucnu využít k statistikám dostupnosti služeb na daném serveru atd.,
- **katalog českých WMS serverů**, jež bude uchovávat údaje a další informace o WMS serverech, které umožňují uživateli do dat nahlédnout v přehledné formě,
- **rozpoznávání vrstev** je jednou z dalších funkcionalit, která je definovaná v zadání. Její práce spočívá v získání a uložení informací o jednotlivých mapových vrstvách, které budou součástí informací o WMS serveru,
- **cachování dat**, jež by mělo umožnit nějakým způsobem cachovat data z WMS serverů,
- **náhledy WMS serverů**, které by měly uživateli umožnit prohlížení dat z vybraného serveru formou náhledu nebo dynamické mapy.

## 6.1 Logická struktura aplikace

V návrhu a postupném vývoji aplikace byl zdrojový kód rozdělen do několika částí (balíčků) pro přehlednost v projektu, které byly následně pojmenovány. Nyní si popíšeme, co v daných částech nalezneme:

- **baseClasses** – zde se nachází základní třídy aplikace, které se následně používají v dalších modulech,
- **dbModel** – místo pro datovou vrstvu aplikace, směřujeme sem všechny třídy starající se o definici datové struktury a práci s databází „GAE Datastore“,
- **services** – sem směřujeme všechny třídy starající se o vykonávání požadavků spuštěné uživatelem, či jinou částí aplikace,
- **staticFiles** – adresář, jehož obsah je přístupný uživatelům přes definovanou URL adresu,
- **UI** – v tomto balíčku definujeme veškeré náležitosti spojené s uživatelským rozhraním přes html stránky, kaskádové styly, až po programový kód starající se o vytvoření toho, co uživatel vidí.

Grafické znázornění a interakci mezi jednotlivými balíčky znázorňuje následující diagram balíčků na obrázku č. 15.



Obr. 15. Diagram balíčků

## 6.2 Návrh a realizace uživatelského rozhraní

Z počátku vývoje aplikace bylo zapotřebí navrhnout design a následně vytvořit nástroje pro zobrazování a generování uživatelského rozhraní. Pro účely tvorby grafického designu jsme použili opensource nástroje pro tvorbu rastrové (Gimp) a vektorové grafiky

(Inkscape). Jakmile byla grafika hotová, proběhlo vytvoření HTML šablony s použitím kaskádových stylů CSS. Pro psaní HTML a CSS kódu jsme použili editor s názvem Komodo Edit, který je opět zdarma jako opensource projekt.

Jakmile byla šablona vytvořena, mohl začít vývoj webové aplikace v prostředí GAE s jazykem Python. Jako vývojové prostředí bylo zvoleno IDE (Integrated Development Environment) Eclipse v nejnovější verzi s označením Indigo, do kterého bylo doinstalováno rozšíření pro podporu programování v jazyce Python s názvem PyDev. Další nutností pro začátek vývoje bylo stažení a nainstalování vývojového balíku Google App Engine SDK for Python a pluginu do prostředí eclipse pro pohodlnější práci s GAE.

V GAE se pro zpracování dotazu na URL používá definice vlastních tříd, které dědí z třídy s názvem „RequestHandler“, obsažené ve webovém frameworku „webapp“. Ve vlastní třídě se pak definuje metoda „get“ nebo „post“ (případně obě zároveň), jenž umožňují zachycení a zpracování těchto HTTP požadavků. Jelikož chceme mít pro každou stránku určité společné vlastnosti jako je template, menu, atd., byl vytvořen modul s názvem „MyPage.py“, který obsahuje importy všech potřebných modulů a hlavně nastavení systému pro využití šablon s názvem „jinja2“.

V modulu „MyPage“ je definována třída s názvem „myRequestHandler“, jejíž rodič je již zmíněná třída RequestHandler. V naší třídě myRequestHandler je několik metod a proměnných, starajících se o ulehčení práce při vytváření dalších stránek ve webové aplikaci. Nyní si vyjmenujme ty nejdůležitější, a popíšeme si, co představují nebo co provádějí:

- **\_\_init\_\_()** – konstruktor třídy, v kterém nastavujeme defaultní hodnoty proměnným a voláme potřebné metody v určeném pořadí pro vytvoření odpovědi, která bude generovat HTML kód výsledné stránky,
- **init()** – tato metoda je volána na posledním řádku konstruktoru a v rodičovské třídě je implementována jako prázdná metoda, která vrací hodnotu None. Slouží k implementaci vlastního kódu v potomku této třídy, představující jednotlivé webové stránky,
- **setStyles()** – slouží k nastavení kaskádových stylů pro jednotlivé stránky,
- **setJs()** – slouží k importu souborů s JavaScriptovým kódem pro jednotlivé stránky,
- **menuJoin()** – metoda určená k seskládání všech definovaných menu, které mají být na stránce zobrazeny,
- **renderTemplate()** – tato metoda se postará o vykeslení celkového HTML obsahu, který je pak odeslán do webového prohlížeče klienta,

- **get()** – v této metodě je zpracován GET požadavek protokolu HTTP, který se stará o volání metody `renderTemplate()`,
- **styles** – je seznam, do kterého se v potomkovy třídy přidávají názvy souborů kaskádových stylů, které mají být pro danou stránku zavedeny,
- **js** – obdobně jako „styles“ jedná se o seznam, sloužící pro přidávání JavaScriptových souborů,
- **contentFile** – proměnná typu string, která definuje název souboru se šablonou obsahu (bez koncovky html),
- **dataContent** – je slovník, definující obsah, který bude vstupovat do systému pro vykreslení šablony obsahu,
- **permis** – slouží k definici oprávnění, pod kterými může být daný obsah zobrazen,
- **requireLogged** – určuje, jestli pro zobrazení dané stránky musí být uživatel přihlášen,
- **user** – obsahuje instanci třídy user. V případě, že je uživatel přihlášen, zároveň i všechny údaje o něm.

Pro přehled dostupných stránek v aplikaci slouží následující diagram tříd na obrázku č. 16. Dále je zde znázorněna asociace s třídou user, přes kterou jsou definovány uživatelé a omezení spojená s oprávněním a uživatelskými účty.

### 6.3 Use Case diagram

Pro zobrazení dynamické (funkční) struktury systému z pohledu uživatele byl vytvořen diagram případu užití, viz obrázek č. 17. Tento diagram zachycuje základní případy užití se třemi druhy aktorů, mezi něž patří:

- **neregistrovaný uživatel** – návštěvník webové aplikace, který může procházet katalogem, vyhledávat údaje, ale nemůže v systému nic přidávat ani upravovat,
- **přihlášený uživatel** – tento typ uživatele má rozšířené možnosti, mezi něž patří především přidávání nových WMS serverů a práce s přiřazováním tagů a kategorií k jednotlivým vrstvám a serverům,
- **administrátor** – tento druh uživatele má nejsilnější typ oprávnění, narozdíl od uživatele může spravovat WMS servery, kategorie a tagy, kde má např. oprávnění pro vymazání těchto údajů ze systému.

## 6.4 Služby aplikace

V naší webové aplikaci je hodně využíván JavaScript přes knihovnu jQuery, díky které jednoduše voláme určité služby naší aplikace a jejich výsledek následně zobrazíme uživateli na stránce bez potřeby znovu načítání celého obsahu webové stránky. Naše služby jsme logicky rozdělili do dvou Python modulů, nacházejících se v balíčku s názvem „services“. Pod pojmem služba myslíme třídu, která umožňuje zpracovat požadavek GET nebo POST protokolu HTTP a následně vrátit odpověď. V našich službách je odpověď formou vygenerovaného HTML kódu nebo ve formátu JSON.

Jelikož všechny naše služby budou mít mnohé společné, vytvořili jsme rodičovskou třídu s názvem `myServices`, která je opět potomkem třídy „`RequestHandler`“. V této třídě se implementuje ověření přihlášení a oprávnění. Dále je zde definovaná metoda pro standardizovaný výstup formátu JSON. Následně rozepíšeme hlavní součásti této třídy:

- **logged** – jde o proměnnou boolovského typu, která obsahuje pravdivostní hodnotu „True“, jestliže je uživatel přihlášen. Tato proměnná se používá v potomcích této třídy v podmínce, jestli má být kód vykonán,
- **message** – proměnná, sloužící k nastavení obsahu zprávy, která bude obsažena v odpovědi klientovi,
- **success** – slouží k nastavení boolovské hodnoty, která bude obsažena v odpovědi klientovi. Značí, zda daná operace ve službě proběhla úspěšně nebo nikoliv,
- **permis** – je list, obsahující seznam oprávnění, které uživatel potřebuje k vykonání dané služby,
- **user** – obsahuje instanci třídy „`user`“. Jestliže je uživatel přihlášen, tak jsou přístupné i všechny jeho údaje,
- **jsonForReturn** – je slovník, který obsahuje data pro vygenerování odpovědi ve formátu JSON,
- **doJsonResponse** – využívá metody „`getJsonResponse`“, která přidá do slovníku „`jsonForReturn`“ hodnoty „`success`“ a „`message`“. Ze slovníku je vygenerován formát JSON, a následně vytvořena odpověď, která je zaslána klientovi.

Nyní zmíníme dva moduly, které slouží pro logickou strukturu služeb naší aplikace a popíšeme si jaký typ služeb obsahují:

- **services.py** – je zde definována rodičovská třída „`myServices`“ a všechny obecné služby týkající se aplikace samotné. Patří sem například `logOn` a `logOff` služby, které se starají o přihlášení a odhlášení uživatele,

- **wms.py** – zde jsou zahrnuty všechny služby, které nějak souvisí s WMS servery. Patří sem služby pro přidání WMS serveru, zobrazení informace o WMS serveru na základě zadané URL adresy, správu tagů a kategorií atd.

Na následujících dvou diagramech tříd viz obr. č. 18 a 19 jsou znázorněny třídy (služby) ze dvou výše jmenovaných modulů.

Nyní si v krátkosti popíšeme základní služby aplikace:

- **userRegistrationService** – stará se o registraci uživatelů, vytvoření nového účtu v systému,
- **logOnService** – umožňuje přihlášení do systému, vytvoří cookies, která udržuje uživatele přihlášeného atd.,
- **logOffService** – odhlašuje uživatele ze systému, odstraňuje cookies atd.,
- **removeUserService** – se postará o odstranění uživatelského účtu ze systému,
- **editUserService** – umožňuje změnu uživatelských údajů,
- **forgottenPasswordService** – umožní vygenerování nového hesla a jeho zaslání na emailovou adresu uživatele,
- **wmsService** – služba umožňující vracet informace ve vygenerované podobě HTML o WMS serveru na základě URL adresy. Druhou možností této služby je přidání (uložení) nového WMS serveru do katalogu.,
- **delWms** – postará se o odstranění WMS serveru z katalogu,
- **wmsServerChecker** – služba sloužící ke kontrole dostupnosti WMS služby, používá se automatickým spouštěním a následným zařazením této služby do fronty pro každý WMS server v katalogu,
- **saveLayer** – používá se pro uložení informací o vrstvě WMS serveru. Operace se zpracovává frontou, kdy každá úloha ve frontě přidá jednu vrstvu. Pokud tomu tak není, nastanou problémy s timeoutem, protože server se 160 vrstvami nestihne být zpracovaný v časovém intervalu jedné minuty,
- **thumbnails** – další služba pro zpracování náhledů vrstev WMS serveru, zpracovává se také s pomocí fronty, jelikož docházelo opět k timeoutům,
- **getThumbnail** – vrací náhled k dané vrstvě jako obrázek, umožňuje náhled vrátit ve dvou velikostech,
- **addTag** – služba umožňující vytvořit nový tag, který může být zároveň přiřazen patřičné vrstvě nebo WMS serveru,

- **delTag** – umožňuje mazání tagu a všech vazeb na něj,
- **assignTag** – přiřazuje tag k WMS serveru nebo vrstvě,
- **unassignTag** – odstraňuje přiřazený tag z WMS serveru nebo vrstvy,
- **getAllTags** – vrací seznam všech dostupných tagů v systému ve formátu JSON,
- **getTagsForWmsOrLayer** – vrací seznam všech přiřazených tagů k WMS serveru nebo vrstvě ve formátu JSON,
- **addCategory** – vytváří novou kategorii,
- **assignCategory** – přiřazuje kategorii k WMS serveru,
- **unassignCategory** – zruší přiřazení patřičné kategorie z WMS serveru,
- **delCategory** – smaže kategorii ze systému a všechny vazby na ni,
- **getCategoriesForWms** – vrací kategorie přiřazené k WMS serveru ve formátu JSON,
- **getAllCategories** – vrací všechny dostupné kategorie v systému ve formátu JSON.

## 6.5 Třídy pro práci s WMS servery a datový model

Jelikož pro práci s WMS servery byla použita existující externí knihovna „OWSLib“, potřebovali jsme vybudovat základní třídu, jenž bude dělat prostředníka mezi naší aplikační vrstvou a zmíněnou knihovnou. Dále by tato pomocná třída měla umožňovat práci s naší datovou vrstvou, implementovanou v balíku „dbModel“. V balíčku „baseClasses“ byl pro tyto účely vytvořen modul s názvem myWms.py, který obsahuje dvě třídy sloužící k výše zmíněným účelům s jmény wmsServer a wmsLayer. Tyto třídy jsou znázorněny na obrázku č. 20, v diagramu tříd, kde je znázorněn import balíčků „dbModel“ a „libraries“.

Databázový model v GAE je navržen pomocí tříd, které jsou potomky základní třídy „db.Model“. V tomto modelu je umožněno definovat vlastní metody a jiné atributy pro práci s daty. Přístup k jednotlivým datům je možný pomocí metody „get“, která očekává jako parametr unikátní klíč. Metoda „get“ vrací instanci naší třídy datového modelu, kde všechny proměnné obsahují hodnoty, které jsou uloženy v datovém úložišti GAE. K těmto datům je možné přistupovat i pomocí dotazů, které vrací seznam všech objektů s danými parametry. Výhodou je, že v každém objektu můžeme volat definované metody, které nám pak vrací data např. v upravené formě. My jsme toho využili a několik metod jsme si definovali právě v našem datovém modelu.

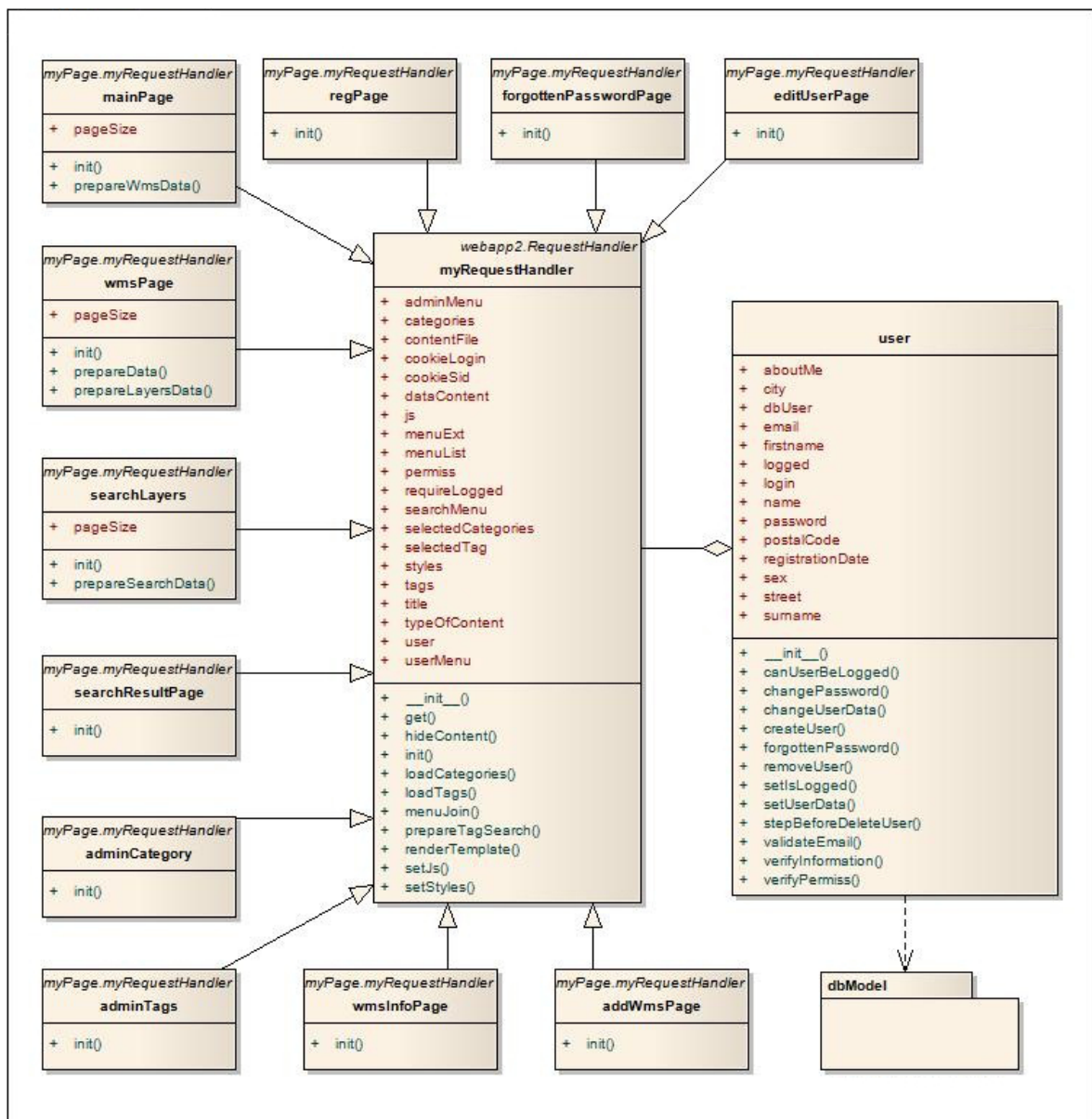
Nyní si vypíšeme názvy tříd, které reprezentují naše datové modely a to ty, které jsou implementovány v modulu `wmsDbModel.py`, jenž je součástí balíčku `dbModel`. Ve zkratce uvedeme, co popisují nebo obsahují:

- `wmsServers` – WMS servery,
- `wmsLayers` – vrstvy WMS serverů,
- `wmsBBoxes` – boundingboxy vrstev,
- `userTags` – uživatelské tagy,
- `joinTagsTable` – přiřazení tagu vrstvě nebo WMS serveru,
- `wmsLayerThumbnails` – náhledy vrstev,
- `categories` – kategorie,
- `joinCategoriesTable` – přiřazení kategorie jednotlivým WMS serverům,
- `wmsAvailabilityLog` – záznamy o dostupnosti.

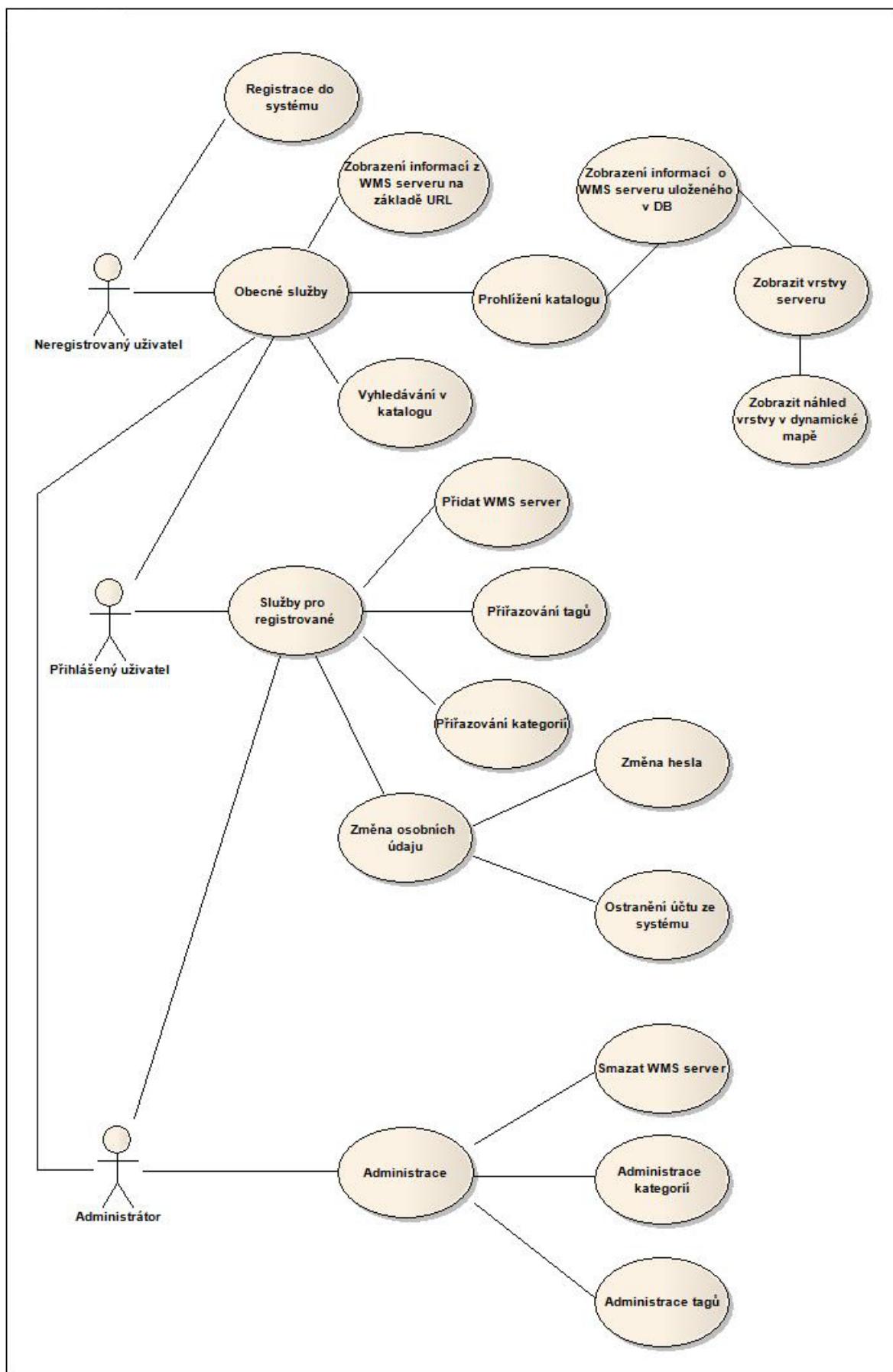
## 6.6 Uživatelé a zabezpečení aplikace

GAE umožňuje využití přihlašování přes existující Google účty. V naší aplikaci jsme vytvořili vlastní mechanismus, který definuje vlastní uživatele. Pro tyto účely byla vytvořena třída „user“, která je využívána skoro ve všech výše zmíněných třídách, a přes kterou probíhají veškeré kontroly přihlášení, čerpání informací o uživateli, atd. Vyobrazení třídy je viditelné např. v diagramu tříd uživatelského rozhraní viz obr. č. 16. V datovém modelu v balíčku `dbModel` je využito modulu `mainSec.py`, který obsahuje třídy reprezentující datové modely pro základní údaje o uživateli a zabezpečení. Následně si tyto třídy vypíšeme a opět krátce zmíníme, co obsahují nebo popisují:

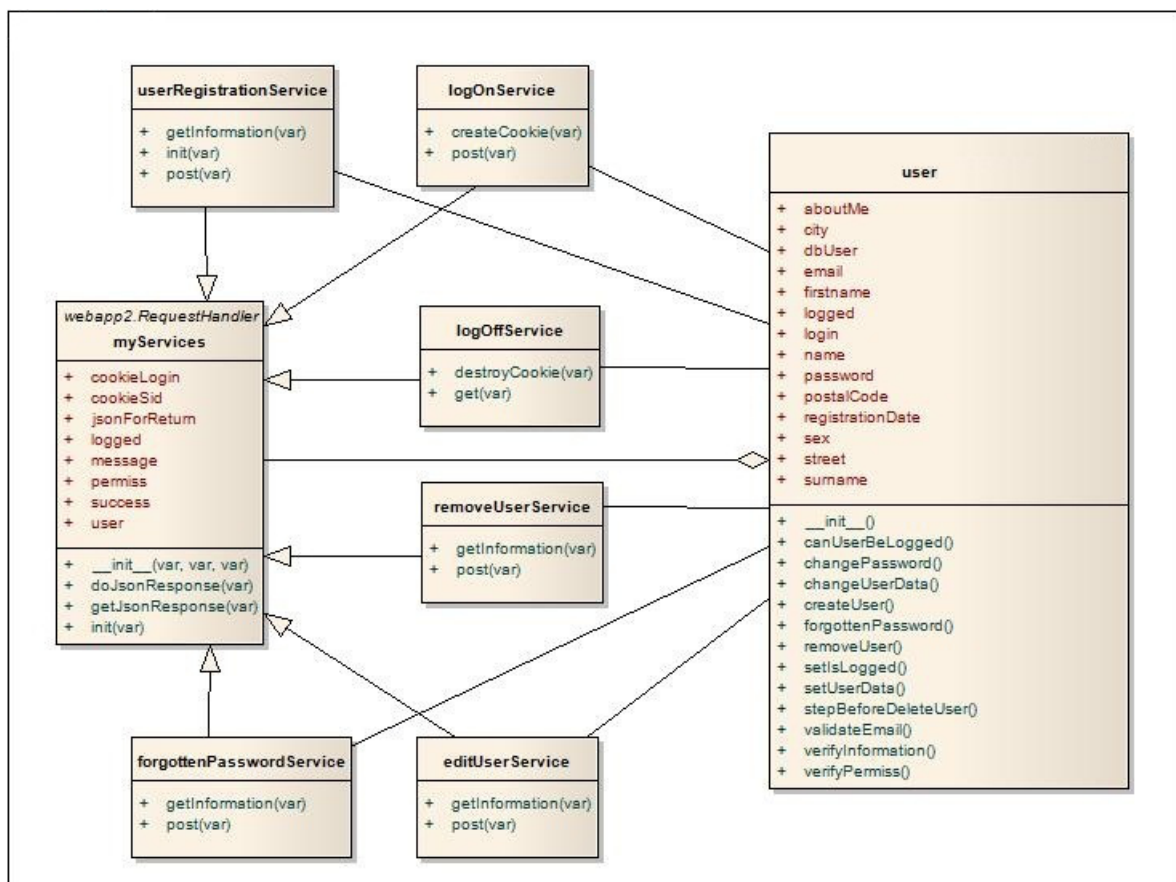
- `users` – uživatelské účty,
- `loggedUsers` – informace o přihlášení,
- `permissions` – oprávnění,
- `permissionsJoinTable` – přiřazení oprávnění pro uživatele.



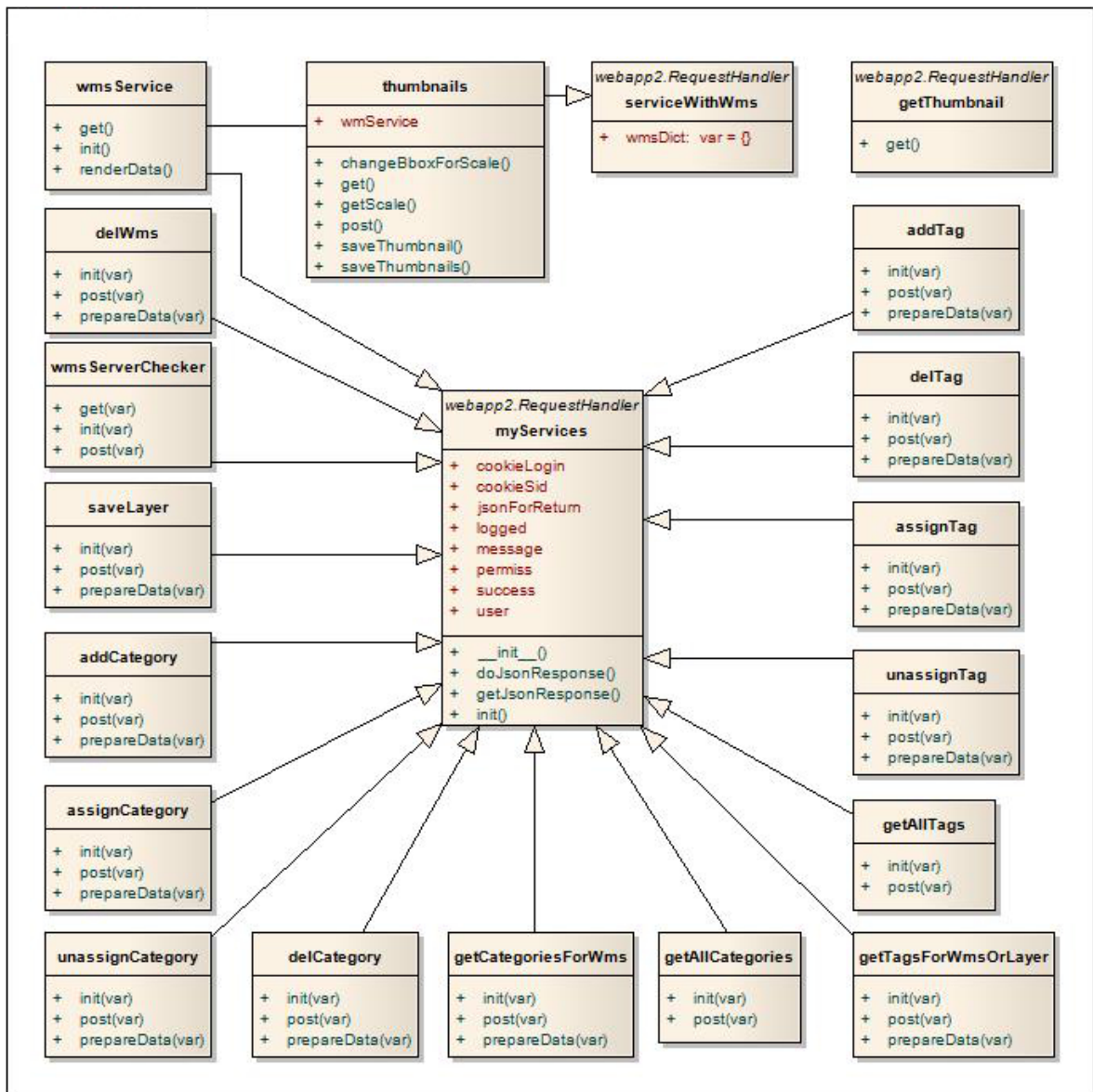
Obr. 16. Diagram tříd uživatelského rozhraní



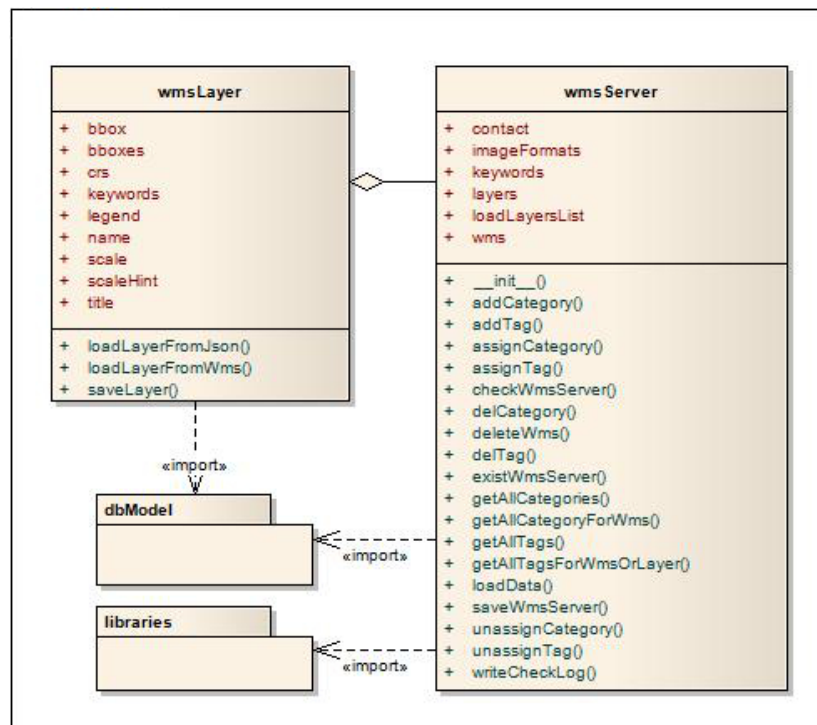
Obr. 17. Use Case diagram



Obr. 18. Diagram tříd služeb aplikace z modulu service.py v balíčku services



Obr. 19. Diagram tříd služeb aplikace z modulu wms.py v balíčku services



Obr. 20. Diagram tříd pro práci s WMS servery z balíčku „baseClasses“

## 7 ŘEŠENÍ PŘI TVORBĚ APLIKACE

V průběhu vývoje jsme narazili hned na několik problémů, které komplikovaly vývoj celé aplikace. Jelikož aplikační doména geoinformačních systémů a WMS serverů je poměrně rozsáhlá, bylo zapotřebí do ní proniknout a naučit se plno potřebných údajů. Jednalo se především o pochopení základního principu přístupu ke geografickým datům, souřadnicovému systému, práci s různými typy souřadnicových systémů, různé výpočty atd. Nyní si některé z těchto problematik popíšeme.

### 7.1 Souřadnicové a referenční systémy

V dnešní době se pro různé účely používá mnoho referenčních a souřadnicových systémů, ve kterých je umožněno pracovat se souřadnicemi. Většina z nich je používána a vhodná jen pro určité geografické lokace (určité státy), naopak některé se snaží popsat souřadnice celé Země.

Pro shromáždění parametrů, které identifikují jednotlivé souřadnicové systémy byla vytvořena databáze EPSG (European Petroleum Survey Group). Tato databáze má za úkol jednoznačně identifikovat souřadnicový systém a definovat transformace a konverze mezi jednotlivými CRS (Coordinate Reference System).

Transformace souřadnic je obecně proces, při kterém dochází k přechodu od jedné soustavy souřadnic ke druhé. Tento přechod se dá vyjádřit pomocí transformační rovnice. Pro tento proces transformace bylo vytvořeno několik nástrojů a knihoven, které se hojně využívají programátory nebo uživateli. EPSG zpřístupňuje takzvané transformační matice, jenž obsahují údaje o posunutí počátku, otočení souřadnicových os o určitý úhel a změnu měřítka, viz [19].

Nejnámější knihovnou pro tyto transformace je PROJ.4. Tuto knihovnu můžeme implementovat v programovacím jazyku Python, ale v GAE ji importovat nelze, jelikož obsahuje součásti, které jsou naprogramovány v jazyce „C“, což GAE nepodporuje.

### 7.2 Vytváření náhledů

V této části bychom chtěli popsat postup, který byl použit pro vytvoření náhledů. Úvodem zmíníme souřadnicové a referenční systémy, které jsou v České republice nejčastěji používány na WMS serverech. Patří sem: S-JTSK (nejpoužívanější v ČR), S-42, WGS 84 / UTM zone 34N, WGS 84 / UTM zone 33N a WGS 84 / geographic. Některé vrstvy na serveru mohou být dostupné jen v některých souřadnicových systémech, ale většina serverů má vždy dostupný geografický systém WGS 84 a k němu lokální, např. S-JTSK. Nevýhoda zobrazení dat v geografickém souřadnicovém systému WGS 84 spočívá v deformaci obrazového výstupu.

V naší aplikaci využíváme dvou druhů náhledů na data vrstev z WMS serveru. Prv-

ním z nich je snaha o vytvoření náhledu při ukládání WMS serveru do databáze. Při tvorbě náhledů bylo zjištěno, že WMS servery mnohdy neudávají informaci o boundingboxu (ohraničujícím území) v různých souřadnicových systémech, ve kterém data nabízí. Jsme tedy odkázáni jen na systém WGS 84, který jsme schopni vždy načíst z elementu <LatLonBoundingBox> dané vrstvy.

Pokud bychom chtěli získat data z vrstvy v jiném souřadnicovém systému, museli bychom převést souřadnice z WGS 84 do systému S-JTSK nebo jiného, a dále si tato data ze serveru vyžádat formou obrázku (např. PNG). Na transformaci mezi souřadnicovými systémy jsme nemohli použít knihovnu PROJ.4 a vývoj vlastního nástroje by byl velice komplikovaný. Jelikož většina serverů podporuje datový výstup v CRS WGS 84, rozhodli jsme se použít tento formát jako výchozí pro tvorbu náhledů. Pro tak malý obrázek není důležité, že je obraz mírně natažený, ale důležité je, aby uživatel měl přehled o tom, jak data v dané vrstvě vypadají.

Dalším faktorem pro rozhodnutí o zvolení souřadnicového systému WGS 84 byl výpočet měřítka. Měřítka nám udává poměrnou velikost délky na mapě ku skutečné velikosti. Jelikož chceme získat náhled dat a některé vrstvy jsou viditelné jen od některých měřítek, je zapotřebí vypočítat měřítko náhledu a porovnat, jestli je v rozmezí maximálního a minimálního měřítka. Pokud tomu tak není, je zapotřebí upravit měřítko a přepočítat boundingbox pro dotaz, který bude v odpovídajícím rozmezí maximálního a minimálního měřítka.

Nyní si představíme dvě metody, které byly sestrojeny pro výpočet měřítka a změnu boundingboxu pro dané měřítko fungující pro souřadnicový systém WGS 84.

```
def getScale(width, longitude):
    mainScale = (2.0*math.pi*6378137.0)/(width/96*0.0254)
    percentLongitude = (longitude/360)
    scale = mainScale*percentLongitude
    return int(scale)
```

Metoda „getScale“ přijímá dva parametry. První z nich je „width“, který udává šířku náhledu v pixelech. Druhým je „longitude“ a představuje nám šířku v souřadnicovém systému WGS 84 (rozdíl krajních souřadnic boundingboxu). Proměnná „mainScale“ znázorňuje poměr skutečné velikosti obvodu elipsoidu ku šířce plochy zobrazené na monitoru. Číslo 6378137 představuje velikost (v metrech) hlavní poloosy elipsoidu WGS 84, který nahrazuje zeměkoulí. V první závorce je vypočítaný obvod elipsoidu, v druhé závorce je následně převedena šířka v pixelech na metry.

Proměnná „percentLongitude“ udává procentuální část naší šířky z parametru „longitude“. Jelikož WGS 84 je udáváno v zeměpisných souřadnicích (zeměpisná šířka a délka), víme, že celková délka má 360 stupňů. Jakmile jsme zjistili, kolik procent z celkového obvodu elipsoidu zabírá naše šířka výřezu, můžeme již vypočítat měřítko vyná-

sobením „mainScale“ a „percentLongitude“.

Druhá z metod nám převádí boundingbox na nový, jenž bude mít výsledné měřítko v definovaném měřítku. Samozřejmě, že výsledná oblast bude jiná, ale my potřebujeme zobrazit uživateli náhled dat, proto nezáleží odkud pochází. Naopak potřebujeme, aby nám WMS server vrátil nějaká data. Pokud budeme požadovat data, která nebudou vyhovovat měřítkovému omezení vrstvy, tak pravděpodobně od serveru dostaneme prázdný obrázek. Tuto metodu už si nebudeme podrobně popisovat, jelikož využívá principy z metody pro výpočet měřítko. Podotkneme pouze, že výsledný boundingbox je vždy vycentrován na střed území původního boundingboxu.

```
def changeBboxForScale(bbox, xsize, ysize, width, scaleNew):
    mainScale = (2.0*math.pi*6378137.0)/(width/96*0.0254)
    percentLongitude = float(scaleNew) / float(mainScale)
    longitude = percentLongitude*360
    scaleRatio = (longitude/(xsize/100))/100
    newXsize= xsize*scaleRatio
    newYsize= ysize*scaleRatio
    xReduce = (xsize - newXsize)/2.0
    yReduce = (ysize - newYsize)/2.0
    newBbox = (float(bbox[0])+xReduce, float(bbox[1])+yReduce, float(
        bbox[2])-xReduce, float(bbox[3])-yReduce)
    return newBbox
```

V případě, kdy není možné vygenerovat výřez kde jsou obsažené data, nemá smysl ukládat obrázek, který je prázdný (bílý nebo průhledný). Pro tyto účely jsme využili obrazovou knihovnu implementovanou v GAE a použili metodu pro výpočet histogramu. Tato metoda vrací seznam obsahující další tři seznamy o 256 číselných hodnotách. Každý seznam reprezentuje jednu z barev RGB (červenou, zelenou, modrou) a každá číselná hodnota představuje počet výskytů jedné z 256 barevných hodnot dané barvy. Zjistili jsme, že pokud se jedná o průhledný nebo bílý obrázek jsou všechny hodnoty seznamů rovny nule mimo krajních hodnot pole, kde nenulová hodnota ve všech seznamech je buď ta nejvíc vlevo nebo vpravo. Poté nebylo problém vytvořit podmínku, která zjistí, zda-li je obrázek prázdný a není jej třeba ukládat.

Druhý způsob náhledů na data WMS serveru je pomocí dynamické mapy implementované ve webové stránce. K tomu využíváme již zmíněnou knihovnu „OpenLayers“. Díky knihovně Proj4js, což je implementace knihovny PROJ.4 v jazyce JavaScript, bylo možné transformovat souřadnice z WGS 84 do S-JTSK a následně zobrazit data primárně v tomto projekčním systému. Výsledná projekce není zakreslená jako s použitím WGS 84, což má ve výsledku vliv na kvalitu a skutečnost zobrazených dat. V této dynamické mapě se uživatel může pohybovat po území, které definuje boundingbox dané vrstvy a může přibližovat či oddalovat data pomocí myši nebo ovládacích prvků umístěných v mapě.

### 7.3 Vyhledávání v katalogu

Ve snaze vytvořit vyhledávání v datech, uložených v aplikaci, jsme narazili na nedostatek v GAE. Zjistili jsme že v GAE Datastore nelze vyhledávat jen část textu v „db.StringProperty“, kterou používáme pro ukládání textových informací typu string. Nalezli jsme na internetu několik projektů, které by umožňovali toto vyhledávání, ale nakonec jsme se rozhodli použít pro tyto účely fulltextového vyhledávání projekt společnosti Google s názvem „custom search“, který umožňuje prohledat celou stránku a vrátit nalezené výsledky tak, jak jsme zvyklí z vyhledávače Google.

Jelikož tento typ vyhledávání pracuje podobně jako vyhledávač Google a indexuje stránky s rozdílem, že prohledává jen konkrétní doménu, bylo zapotřebí vytvořit XML sitemap, která bude obsahovat adresy na všechny WMS servery, a kategorie v aplikaci, abychom docílili efektivního zaindexování všech stránek. Pokud tedy vyhledávací robot podá dotaz na soubor „sitemap.xml“, naše aplikace vygeneruje XML soubor, v němž bude obsah stanovený pro XML sitemap se všemi výše zmíněnými adresami.

Druhý typ vyhledávání jsme implementovali na základě uživatelských tagů a třídění dle kategorií, budeme popisovat v kapitole s názvem „Popis aplikace a její ovládání“.

### 7.4 Zobrazování legend

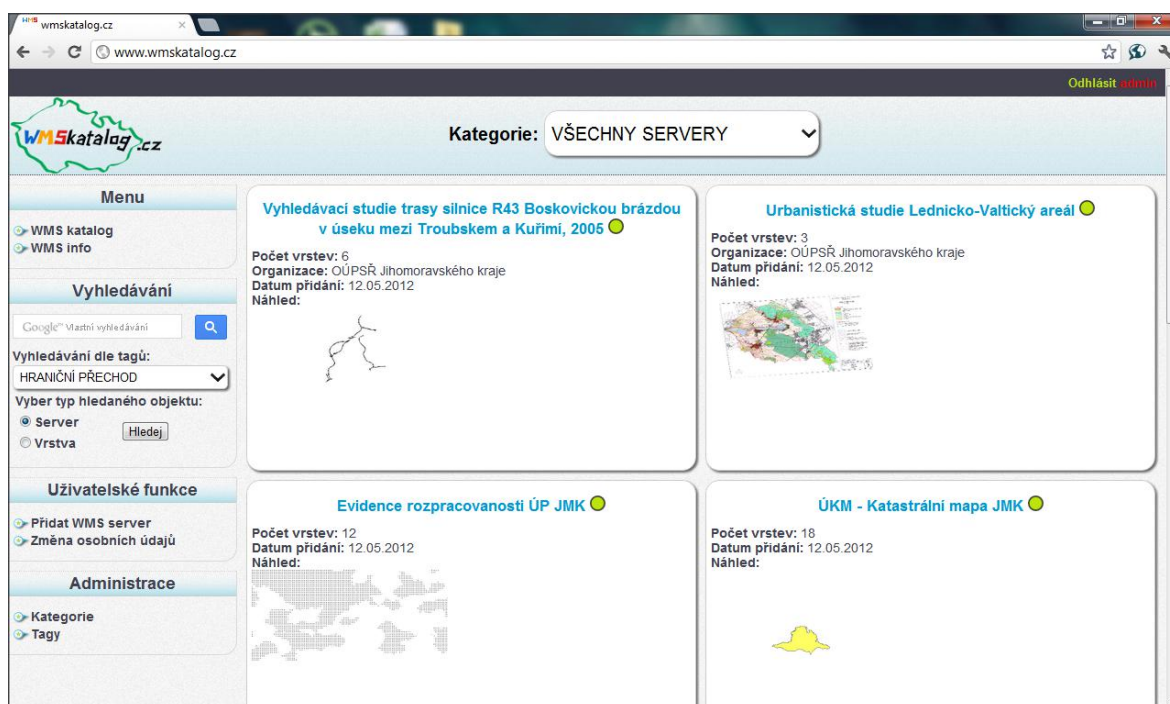
Další problém nastal při zobrazování legend vrstev. Pokud byl některý z vytíženějších serverů naráz požádán o vrácení všech legend k zobrazeným vrstvám, nestihl někdy tyto legendy vrátit a na stránce následně nebyly zobrazeny. Pro tyto účely jsme použili plugin do jQuery s jménem „Lazy Load“, což volně přeloženo znamená „líné načítání“. Implementace tohoto pluginu nám zajišťuje, že obrázky legend se budou načítat, jakmile začne uživatel scrollovat po stránce směrem dolů. V momentě, kdy bude spodní část zobrazené stránky těsně před místem pro legendu, začne se obrázek legendy ze serveru načítat. Pokud uživatel nescrolluje až na konec stránky, nebudou se zbytečně načítat všechny legendy. Tento stejný plugin jsme zároveň použili i při zobrazování náhledů uložených v databázi naší aplikace.

## 8 POPIS APLIKACE A JEJÍ OVLÁDÁNÍ

V této kapitole si zrekapitulujeme veškerou funkcionalitu aplikace ze strany uživatele a popíšeme uživateli, jak s aplikací, která byla pojmenována „WMSkatalog“, pracovat. WMSkatalog je veřejně dostupný na webové adrese s URL <http://www.wmskatalog.cz>. Uživatel tuto adresu může vložit do adresového pole svého prohlížeče a začít s WMSkatalogem pracovat.

### 8.1 Úvodní stránka

Úvodní stránka aplikace nabízí 10 nejnověji přidaných WMS serverů do katalogu. Doporučené rozlišení pro prohlížení je o šířce obrazovky 1280 pixelů, kdy jsou vedle sebe zobrazeny vždy dvě dlaždice obsahující název WMS serveru, počet vrstev a některé další dostupné informace. Na konci je zobrazen jeden z náhledů vrstvy daného WMS serveru. Šířka obrazovky však není omezujícím faktorem, protože layout byl konstruován tak, aby se šířka dlaždic přizpůsobila velikosti prohlížeče. Jestliže uživatel bude mít velký širokoúhlý monitor, zobrazí se na jeho šířku více dlaždic. Jakmile by uživatel začal okno prohlížeče zmenšovat, naprogramovaný JavaScriptový kód tuto událost zachytí a začne přizpůsobovat velikosti dlaždic. Jestliže zjistí, že již není místo pro poslední sloupec dlaždic, bude tento sloupec odstraněn a velikost dlaždice bude rozšířena. Tento JavaScriptový kód umí samozřejmě pracovat i v opačném případě, když bude okno webového prohlížeče zvětšováno. Úvodní stránka je znázorněna na obrázku č. 21.



Obr. 21. Pohled na úvodní stránku aplikace WMSkatalog

Nyní si popíšeme vizuální rozdělení aplikace. Po levé straně se nachází uživatelské menu, jehož obsah je závislý na tom, jestli je uživatel přihlášen a dále na jeho oprávnění. Vpravo od menu se nachází obsahová část aplikace, ve které se zobrazují veškeré informace o WMS serverech a dalších částech aplikace. Tato obsahová část může zobrazovat již zmíněné dlaždice přizpůsobivé velikosti okna nebo obsahuje jen jednu dlaždici, ve které je vypsán obsah dané stránky.

Ve horní části aplikace se nachází černá lišta sloužící k přihlášení nebo identifikaci uživatele. V případě, že uživatel není přihlášen, jsou v pravé části lišty zobrazena textová pole pro login a heslo s tlačítkem k přihlášení. Vlevo od těchto textových polí je odkaz „Registrace“, přes který uživatel vstoupí na stránku, kde si může založit přístupový účet do WMSkatalogu a stát se tak registrovaným uživatelem.

## 8.2 Registrace

Pro registraci do systému je nutné kliknout na červený odkaz „Registrace“ nacházející se v horní části stránky. Po kliknutí se zobrazí stránka sloužící k registraci, kde jsou textová pole pro přihlašovací jméno, email, heslo, ověření hesla, jméno, příjmení, ulici, město, PSČ, tlačítka pro výběr pohlaví a textové pole pro informace o uživateli. Uživatel nebude moci registraci dokončit, pokud nebudou vyplněna textová pole s červenými popisky. Jedná se o tato textová pole: přihlašovací jméno, email, heslo a ověření hesla. Přihlašovací jméno nesmí být již v systému obsažené, jinak bude uživatel na tuto skutečnost upozorněn a bude tento údaj muset změnit. Email bude ověřen regulárním výrazem pro kontrolu správného zadání emailové adresy, a pokud nebude vyhovovat, uživatel opět nebude připuštěn k dokončení registrace. Další podmínkou je, aby se shodovala textová pole pro heslo a ověření hesla. V případě že tomu tak nebude, bude uživatel upozorněn na nekorektní zadání. Další z omezujících podmínek je minimální délka hesla, která činí minimálně 6 znaků. Jestliže tomu tak nebude, uživatel bude upozorněn, aby toto zadání napravil.

## 8.3 Funkce WMS info

Pokud uživatel klikne v menu na odkaz s názvem „WMS info“, zobrazí se mu stránka s textovým polem pro URL adresu WMS serveru a tlačítkem s nápisem „Zobraz informace“, které slouží k odeslání požadavku o zobrazení informací o WMS serveru. Tato funkce v podstatě volá WMS operaci „GetCapabilities“ na službu se zadanou URL adresou. Pokud je WMS služba na serveru pod danou URL adresou dostupná, je výsledné XML zpracováno a uživateli zobrazeno v příjemnější podobě rozepsaných informací, které jsou lépe čitelné než XML. Pokud je adresa WMS serveru zadána špatně nebo nastala chyba při zpracování údajů, je o tom uživatel informován chybovým hlášením. Mezi informace vypsané o WMS serveru patří: název serveru, počet vrstev, které

obsahuje, názvy vrstev (formou odkazu, který po stisknutí scrolluje stránku na danou vrstvu), podporované výstupní formáty, informace o omezení a kontaktní údaje. Dále jsou vypsané informace o dostupných vrstvách, které služba nabízí. Mezi údaje o vrstvě patří: jméno vrstvy, titulek, boundingbox v souřadnicovém systému WGS 84, ostatní boundingboxy, koordinační systémy, legenda atd. Pokud daná vrstva disponuje i ostatními boundingboxy, jsou tyto informace implicitně skryté a po stisku na odkaz s nápisem „Ukázat“ se tyto informace objeví.

#### 8.4 Menu vyhledávání

Toto menu, které je možné vidět např. na obrázku č. 21, slouží k vyhledávání údajů v katalogu WMS serverů. První možností je použití fulltextového vyhledávače Google, kde do textového pole s logem Google a nápisem „Vlastní vyhledávání“ napíšeme údaj, který chceme vyhledat (např. slovo „doprava“). Po kliknutí na modrou ikonu s lupou se vyhledávač pokusí na WMSkatalogu daný údaj nalézt a podá výstup s výsledky hledání. Tento výstup pro hledané slovo „doprava“ je vyobrazen na obrázku č. 22.

**Vyhledávání na webu**

doprava

Přibližný počet výsledků: 9 (0,11 s)

provazováno na Google™ Vlastní vyhledávání

**Reklamy Google**

**Doprava a Spedice**  
Denně až 300 tis nabídek nákladů a volných vozů z celé Evropy!  
[www.timocom.cz/Burza-Nakladu](http://www.timocom.cz/Burza-Nakladu)

**Dopravní databáze Trans**  
Bezplatné vymáhání pohledávek Dodatočné možnosti databáze  
[www.trans.eu/](http://www.trans.eu/)

**Doprava**  
Vnitrostátní **doprava** a spedice. Vyzvedneme zásilku kdekoliv v ČR!  
[www.mtmtransport.cz/](http://www.mtmtransport.cz/)

**Autobusova doprava**  
Vnitrostátní a mezinárodní **doprava**. Nej kvalitnější služby a komfort.  
[www.abdop.cz/](http://www.abdop.cz/)

**Šikovní program DOPRAVKA**  
Prodejní a fakturační program pro spediční firmy, přepravce, prodejce  
[www.dopravka.cz/](http://www.dopravka.cz/)

**Doprava**  
Využijte skvělých služeb v oblasti **dopravy!** Profesionalita především.  
[www.azd.cz/](http://www.azd.cz/)

**wmskatalog.cz - Wms server: ÚAP – A.3 Výkres záměrů na ...**  
[www.wmskatalog.cz/ui/wms/?wms...page=3](http://www.wmskatalog.cz/ui/wms/?wms...page=3)  
26. duben 2012 ... Titulek: [D] **Doprava**. Uživatelské tagy: Zobrazováno pouze v měřítku: Min: 1: 10000, Max: 1:10000000. Vypočítáno ze scalehint wms verze ...

**wmskatalog.cz - Wms server: Prověřovací územní studie v oblasti ...**  
[www.wmskatalog.cz/ui/wms/?wms...](http://www.wmskatalog.cz/ui/wms/?wms...)  
26. duben 2012 ... Vrstvy na serveru: **Doprava**, **Doprava**, **Doprava**, Funkční plochy, Funkční plochy, Funkční plochy, Výkres – Urbanist. řešení, Výkres ...

**wmskatalog.cz - Wms server: ÚAP – B.1.1 Územní závady a ohrožení**  
[www.wmskatalog.cz/ui/wms/?wms...](http://www.wmskatalog.cz/ui/wms/?wms...)  
26. duben 2012 ... Titulek: [D] **Doprava**. Uživatelské tagy: Zobrazováno pouze v měřítku: Min: 1: 10000, Max: 1:10000000. Vypočítáno ze scalehint wms verze 1.1.1. ...

**wmskatalog.cz - Wms server: ÚAP – B.1.2 Střety záměrů na ...**  
[www.wmskatalog.cz/ui/wms/?wms...](http://www.wmskatalog.cz/ui/wms/?wms...)  
26. duben 2012 ... Vrstvy na serveru: [C] Územní členění, [D] **Doprava**, [D] **Doprava**, [G] Geologie, [J] Zvláštní zájmy, [L] Lesy, NATURA 2000 – evr. význ.

**wmskatalog.cz - Wms server: Územní studie aglomeračních vazeb ...**  
[www.wmskatalog.cz/ui/wms/?wms...](http://www.wmskatalog.cz/ui/wms/?wms...)  
26. duben 2012 ... Vrstvy na serveru: Brněnská aglomerace, Dojíždka do Brna > 25%, **Doprava** – stav, Dostupnost centra Brna do 40 min, Hranice, Model A ...

**wmskatalog.cz - Wms server: JMK – Územní prognóza**

Obr. 22. Fulltextové vyhledávání na WMSkatalogu přes „Google Custom Search“

Další typ vyhledávání je pomocí uživatelských tagů, které budou následně popsány později. Uživatel si ve formuláři vybere jeden z hledaných tagů, následně vybere, jestli chce hledat mezi servery nebo vrstvami a stiskne tlačítko s nápisem „Hledej“. Aplikace pak zobrazí nalezené výsledky (vrstvy nebo servery) s přiřazeným tagem.

## 8.5 Přihlášení a zapomenuté heslo

Jestliže je uživatel již zaregistrovaný, může se do aplikace přihlásit. Po vyplnění přihlašovacího jména a hesla v pravém horním rohu může uživatel stisknout tlačítko s nápisem „Přihlásit“. Aplikace ověří, zda jsou pole s uživatelským jménem a heslem vyplněna. Pokud tomu tak není, uživatel bude upozorněn, aby tyto přihlašovací údaje vyplnil. Jestliže uživatel udělá chybu v přihlašovacím jméně nebo hesle, a tyto špatné údaje odešle, aplikace zjistí nekorektnost údajů. Následně je uživatel upozorněn na nesprávné přihlašovací údaje s varováním, že bude přesměrován na stránku pro zapomenuté heslo. Na této stránce budou zobrazeny dvě textová pole pro zadání přihlašovacího jména, emailové adresy a tlačítko s nápisem „Odeslat“. Po vyplnění údajů a stisku tlačítka se údaje v aplikaci ověří a vygeneruje se nové heslo, které je odesláno uživateli na jeho emailovou adresu.

Jestliže uživatel zadal správné přihlašovací údaje při přihlašování do systému, bude přesměrován na úvodní stránku a objeví se mu rošířené menu o uživatelské funkce.

## 8.6 Uživatelské funkce

V aplikaci jsou zatím přístupné dvě funkce, které se zpřístupní s přihlášením uživatele do aplikace WMSkatalog. Nyní si tyto funkce popíšeme.

### 8.6.1 Přidat WMS server

Jednou z rozšířených funkcí registrovaného uživatele je možnost přidávat do WMSkatalogu nové WMS servery. Po stisku odkazu „Přidat WMS server“ nacházejícího se v menu s názvem „Uživatelské funkce“, je uživateli zobrazena stránka velice podobné stránce pro funkci „WMS info“. Také se zde nachází textové pole pro URL adresu WMS serveru, kterou uživatel musí zadat. Po zadání adresy by měl stisknout tlačítko s nápisem „Ulož“. Po stisku tlačítka server ověří, zda v systému není již obsažena tato adresa, a pokud není, proběhne pokus o přidání WMS serveru do katalogu. Jestliže se přidání serveru zdařilo, bude uživatel o této události informován vypsáním textu s informací o úspěšném přidání serveru (jestliže se nepodařilo údaje z WMS serveru s zadanou URL adresou načíst, bude o tom uživatel informován chybovým hlášením). Pokud si uživatel bude chtít informace o uloženém serveru prohlédnout, může kliknout na logo aplikace WMS katalog, která ho přesměruje na hlavní stránku, kde jsou vypsány nejnověji přidané servery, nebo může využít odkazu v menu s textem „WMS katalog“. Tato operace uložení trvá v závislosti na počtu vrstev na serveru, nikdy by však neměla přesáhnout čas jedné minuty. Aplikace WMSkatalog v dalším čase pokusí vytvořit náhledy vrstev, ato v intervalu jednoho náhledu za minutu (tento interval se může kdykoliv změnit v konfiguraci fronty administrátorem aplikace).

### 8.6.2 Změna osobních údajů a hesla, smazání účtu

Pro potřebu změny osobních údajů nebo hesla slouží stránka, na kterou se uživatel dostane s pomocí stisknutí odkazu v menu s nápisem „Změna osobních údajů“. Na této stránce je zobrazen formulář, který vypadá obdobně jako formulář ze stránky pro registraci s tím rozdílem, že má již vyplněné zadané údaje. Tyto údaje můžeme změnit a následně po stisknutí tlačítka s nápisem „Změň“ se údaje změní. Pokud si uživatel přeje změnit heslo, vyplní údaje „Heslo“ a „Ovření hesla“, které se musí shodovat.

V této části aplikace je také možné účet v aplikaci WMSkatalog odstranit kliknutím na odkaz „Chci smazat účet“. Po kliknutí na odkaz je potřeba tuto operaci potvrdit. Následně při potvrzení je účet trvale odstraněn ze systému.

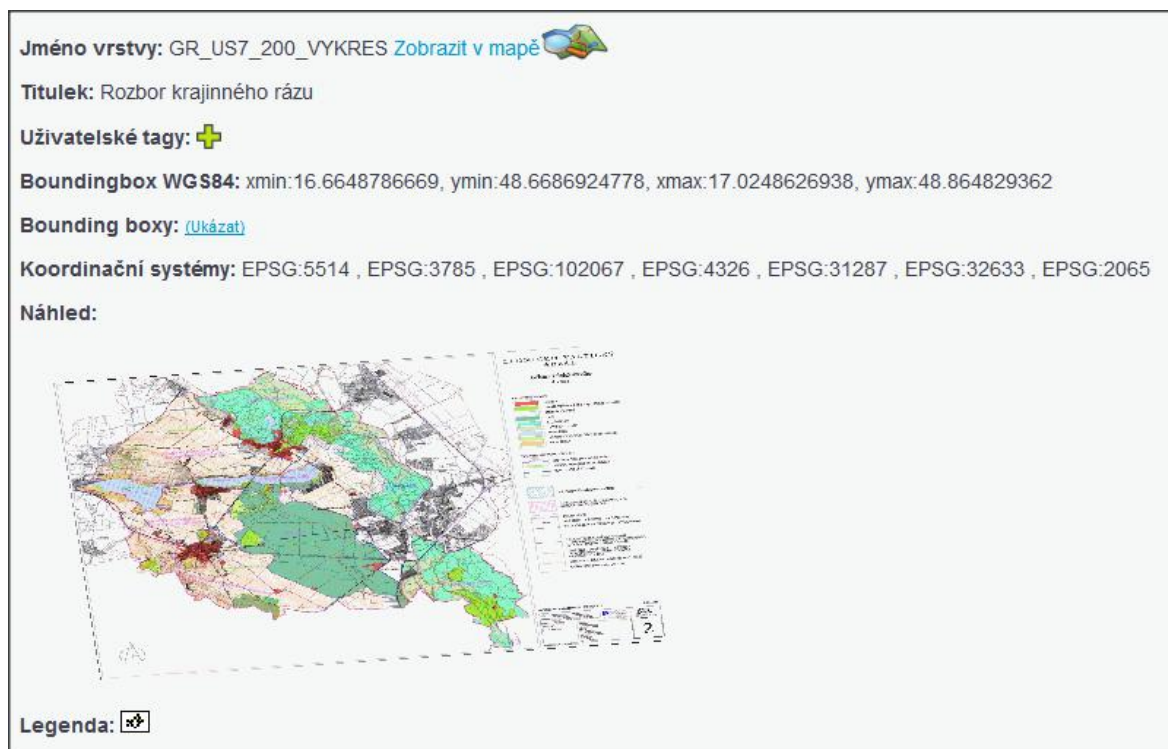
## 8.7 Prohlížení informací o uloženém WMS serveru

Pokud si uživatel bude chtít zobrazovat informace o WMS serveru a jeho vrstvách, musí kliknout na název WMS serveru v dlaždici s informacemi o daném serveru. Další možnost, jak se dostat na informace o WMS serveru, je za pomoci vyhledávání, které bylo již zmíněno výše. Tyto informace zobrazené o WMS serveru jsou obdobné jako ve funkci „WMS info“ s rozdílem, že data jsou zobrazována z databáze aplikace WMSkatalog. Na jednu stránku se zobrazuje maximálně 10 vrstev. Pokud jich má server více, ve spodní části stránky se objeví navigační odkaz s nápisem „další“ (případně „předchozí“).

Jelikož je zobrazeno maximálně 10 vrstev zároveň, seznam vrstev, který je uveden ve vrchní části stránky s informacemi o WMS serveru, neztrácí funkci prokliku, který zobrazí a sjede na patřičnou část stránky, kde se nachází informace o vrstvě. Jestliže se vrstva nachází na jiné stránce, než je stránka aktuální, bude uživatel přesměrován na novou stránku, kde je informace o vrstvě obsažena.

Pokud se podařilo vytvořit pro vrstvu náhled, bude tento náhled zobrazen v informacích o vrstvě. Ukázka informací o vrstvě je znázorněna na obrázku č. 23.

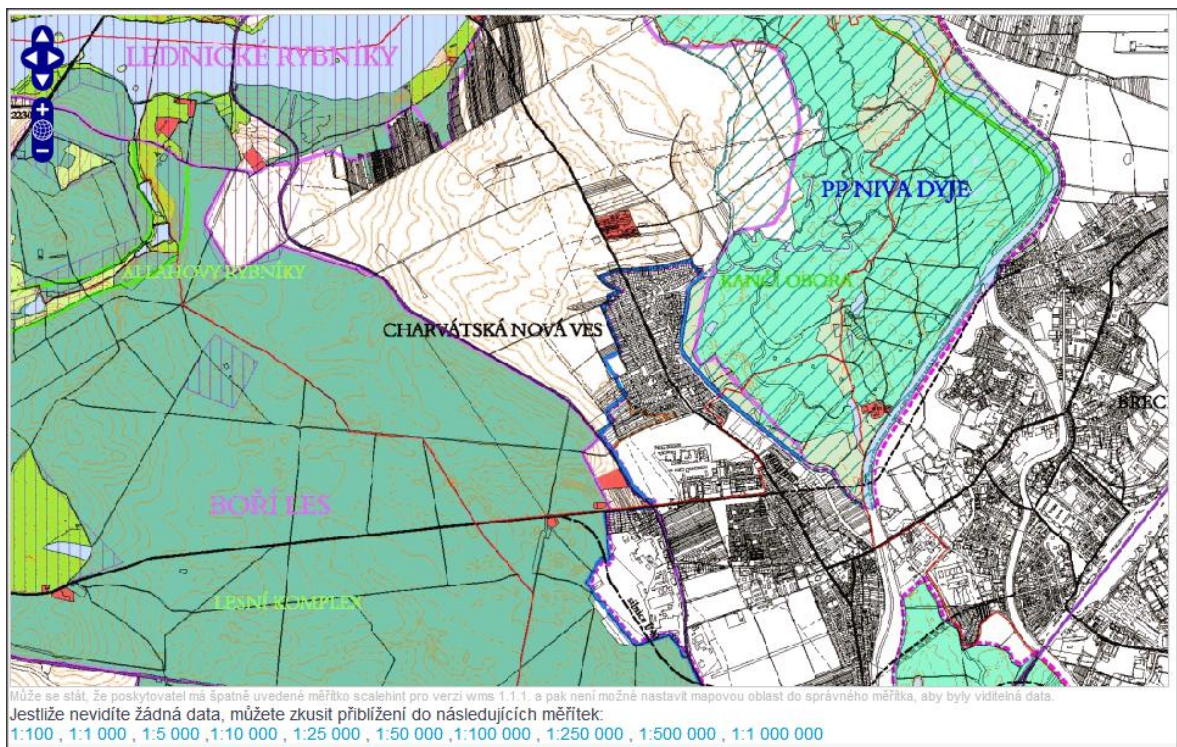
Jestliže uživatel klikne na ikonu mapy, která je zobrazena na obrázku č. 23, nebo na nápis „Zobrazit v mapě“, aplikace WMSkatalog vygeneruje mapu, ve které se pokusí zobrazit data vrstvy z daného WMS serveru. Pokud je WMS služba špatně nastavena, nebo data v dané oblasti nejsou, může se stát, že se v mapě nic nezobrazí. V takovém případě může uživatel zkusit zvětšení do jiného měřítka pomocí kolečka myši, ovládacích prvků na mapě, nebo do přesných měřítek uvedených pod mapou formou interaktivních odkazů. Ukázka zobrazení vrstvy v dynamické mapě je znázorněna na obrázku č. 24.



Obr. 23. Ukázka zobrazení informace o vrstvě WMS serveru

## 8.8 Tagy a kategorie

Uživatelské tagy mohou přidávat k jednotlivým vrstvám a WMS serverům pouze přihlášení uživatelé. Vedle nadpisu „Kategorie“ a „Uživatelské tagy“, se nachází zelená ikona ve tvaru plus. Po stisknutí této ikony se objeví formulář, ve kterém vybereme ze seznamu tagů nebo kategorií, kterou chceme k dané vrstvě či WMS serveru přiřadit. Následně stiskneme tlačítko s nápisem „Přiřad““. U tagů může uživatel navíc přidat vlastní tag, jestliže jej v seznamu již existujících tagů nenalezl. Může toho docílit kliknutím na odkaz s nápisem „Přidat nový tag“. Kliknutí na odkaz aplikace vygeneruje nové textové pole, do kterého může uživatel zapsat nový tag, který ještě není v systému obsažen. Po stisknutí klávesy „Entrer“ bude požadavek o přidání odeslán do aplikace. Následně proběhne kontrola, zda daný tag již není obsažen v databázi. Pokud tomu tak není, bude tag uložen do databáze a přiřazen k aktuální vrstvě nebo WMS serveru. Odebrání kategorie či tagu z WMS serveru nebo vrstvy probíhá za pomoci stisku červené ikony ve tvaru mínus. Tato ikona se nachází vedle seznamu přiřazených tagů nebo kategorií. Po stisknutí ikony se objeví formulář se seznamem přiřazených tagů nebo kategorií. V tomto formuláři vyberete požadovaný tag nebo kategorii, kterou chcete odebrat a následně stisknete tlačítko s nápisem „Odeber““. Po stisku se požadavek zpracuje a následně se tag případně z kategorie odebere.



Obr. 24. Ukázka zobrazení vrstvy v dynamické mapě

## 8.9 Administrátorské funkce

Mazání serverů je funkce, která je dostupná pouze administrátorovi aplikace. Tato funkce je přístupná přes odkaz „Smazat server“, ve výpisu informací o vybraném WMS serveru. Po stisku tohoto odkazu se server odstraní z aplikace WMSkatalog.

Poslední z funkcionalit je správa kategorií a tagů. V těchto částech aplikace přístupných přes příslušné odkazy v menu „Administrace“, které vidí jen administrátor, se dostaneme do seznamu tagů nebo kategorií. Zde je možné jednotlivé tagy či kategorie smazat nebo vytvořit nové.

## ZÁVĚR

Webových služeb pro poskytování geoprostorových informací skrze distribuované prostředí sítě internet je několik, ovšem nejrozšířenější a nejpoužívanější je služba „Web Map Service“ definována konsorciem OGC, na které je založena praktická část této práce. Prvním cílem práce bylo vytvoření přehledu dostupných webově orientovaných služeb v GIS. Práce popisuje aktuální problematiku v této oblasti a jednotlivé služby, které se v dnešní době nejvíce využívají pro poskytování geoprostorové informace skrze webové služby. Druhým cílem bylo vytvoření nové webové aplikace poskytující ucelené informace o dostupných WMS službách v České republice. Nově implementovaná aplikace s názvem „WMSkatalog“ zpracovává informace z veřejných WMS serverů a poskytuje ucelený interaktivní seznam (katalog) pro přehledné vyhledávání v mapových podkladech, které je možné dále zpracovávat nástroji v GIS.

Aplikace disponuje interaktivními náhledy mapových vrstev umožňující jejich snadné prohlížení a přehled o datech, které WMS servery nabízejí. WMSkatalog umožňuje uživatelům registraci a přihlášení do systému, jenž dovoluje využívat další funkcionality, mezi něž patří přidávání nových WMS serverů a přiřazování uživatelských tagů a kategorií. Cílem této snahy je vytvoření komunity, která bude katalog neustále aktualizovat.

S ohledem na druh nabízených dat a korektnost metadat služeb o standardu WMS bylo zjištěno, že nelze vždy docílit zobrazení dat některých vrstev jasným programovým řešením. To zapříčiní buď vytvoření náhledu anebo zobrazení těchto dat v mapě.

Jednou z funkcionalit obsažených v zadání práce bylo kešování dat z WMS serverů. Tato funkcionalita by umožňovala uložení obrazových dlaždic do úložiště aplikace a tyto dlaždice s pevně danými souřadnicemi by byly přístupné klientovi. Bylo zjištěno, že jediná datová vrstva například pro celou Českou republiku, která by byla takto kešovaná ve všech úrovních přiblížení, by zabírala více než 100 GB dat. Kvůli vysokým cenám za cloudové služby spojené s úložištěm a prostředky využitými při vytváření keše a toku dat mezi klientem a aplikací při využívání kešovaných dat, je nereálné takovouto službu provozovat.

Výsledná aplikace v podobě interaktivního katalogu může sloužit jak odborníkům v GIS, kteří hledají geografická data, tak i běžným uživatelům, kteří nahlíží do mapových dat, která nejsou běžně k dispozici. V katalogu mohou tyto uživatelé snadno vyhledávat díky implementovanému fulltextovému vyhledávání, nebo mohou využít kategorie a uživatelské tagy, které rozšiřují možnost nalezení požadovaných dat.

Projekt WMSkatalog lze dále rozvíjet, a tak je vybudován dobrý základ, na kterém bude možné implementovat další funkcionality v systému, které budou moci využívat již uložená data.

## CONCLUSION

Diploma thesis on „Development of Support Services for Geographical Information Provision Based on WMS“ deals with the currently used web services in the area of geoinformation technologies, their use and providing functionality for working with spatial information. This work also describes the standards which are employed for providing geographic data or are implemented within the web services. The practical part deals with the description of the technology used for cloud-based solution and describes the design and implementation of the application which serves as a compact catalog of available WMS servers in the Czech Republic. The aim was to create a web application written in Python that would run on the „Google App Engine“ cloud solution. This application should incorporate certain support services for providing geospatial information based on „Web Map Service“ standard. The resulting application in the form of interactive catalog may serve both GIS professionals who are looking for geographic data and ordinary users who just like to browse through some map data that are not usually available. The thesis also contains a detailed user documentation for the developed application.

There are several web services for providing geospatial information through a distributed network (internet) environment, but the most widespread and most commonly used is the „Web Map Service“ developed by the Open Geospatial Consortium (OGC). This standard protocol forms the basis of the practical part of this work. The first objective was to create a list of available web-based GIS services. This work describes the current issues in this area and the services that are currently most used for providing geospatial information through web services. The second objective was to create a new web application providing comprehensive information about the available WMS services in the Czech Republic. The newly developed application called „WMSkatalog“ processes information from public WMS servers and provides a complete interactive list (catalog) for a clear searching through the map data that can be further processed by GIS tools.

The application offers interactive previews of map layers for easy browsing and getting an overview of the data provided by different WMS servers. WMSkatalog allows users to register and log into the system which enables using other functionalities, including adding new WMS servers and assigning user tags and categories. The aim is to create a community of people who will constantly update the catalog.

With regard to the type of offered data and correctness of the metadata within the WMS services it has been found that it is not always possible to display some layers correctly through a clear software solution. This results in either building a preview or displaying such data in a map.

One of the functionalities which were part of the work assignment was caching data

from WMS servers. This functionality would allow saving the image tiles into the application storage, and these tiles with fixed coordinates would be accessible to the client. It was found that a single data layer covering the entire Czech Republic which would be cached at all zoom levels would occupy more than 100 GB of data. Due to high prices for cloud services offering sufficient data storage, caching and streaming the data between the client and the application it is impossible to run such a service.

The resulting application may serve both GIS professionals who are looking for geographic data and ordinary users who just like to browse through some map data that are not usually available. These users may easily search in the catalog using the embedded full-text search, or they may use categories and user tags which extends the possibility of finding the required data.

WMSkatalog project can be further developed, providing a basis for implementing additional functionalities which may work with the already stored data.

## SEZNAM POUŽITÉ LITERATURY

- [1] RICHTA, Karel. *Standardy pro Webové služby WSDL, UDDI*. In: Standardy pro Webové služby WSDL, UDDI [online]. 2003 [cit. 2012-04-29]. Dostupné z: <http://www.ksi.mff.cuni.cz/~richta>
- [2] HLÁSNÝ, Tomáš. *Geografické informační systémy - Priestorové analýzy*. Banská Bystrica: Čižmárová - Partner, Poniky, 2007. ISBN 978-80-8093-029-5
- [3] *Sborník příspěvků z 18. konference studentů v matematice na školách VŠTEZ: Geostatistika v R-projektu* [online]. Praha: Vydavatelství ČVUT v Praze, 2010 [cit. 2012-05-05]. Dostupné z: <http://mat.fsv.cvut.cz/komisevstez/18sk/files/sbornik.pdf>
- [4] ŠÍMA, J. Geomatics and Geoinformatics in Modern Information Society - Projection of New Trends into their Curricula at the University of West Bohemia in Pilsen. In *Scientia Est Potentia*. Prague: Czech Technical University, 2007. s. 147-154. ISBN: 978-80-01-03718-8
- [5] ČSN ISO 19119. *Geografická informace - Služby*. Praha: ČESKÝ NORMALIZAČNÍ INSTITUT, 2006
- [6] Open Geospatial Consortium, Inc.<sup>®</sup>. *Home page* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/>
- [7] Open Geospatial Consortium, Inc.<sup>®</sup>. *Web Map Service (WMS)* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/standards/wms>
- [8] Open Geospatial Consortium, Inc.<sup>®</sup>. *Web Feature Service (WFS)* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/standards/wfs>
- [9] Open Geospatial Consortium, Inc.<sup>®</sup>. *Web Coverage Service (WCS)* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/standards/wcs>
- [10] Open Geospatial Consortium, Inc.<sup>®</sup>. *Web Processing Service (WPS)* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/standards/wps>
- [11] Open Geospatial Consortium, Inc.<sup>®</sup>. *Keyhole Markup Language (KML)* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/standards/kml>

- [12] Open Geospatial Consortium, Inc.<sup>®</sup>. *Geography Markup Language (GML)* [online]. c1994-2012 [cit. 2012-05-04]. Dostupné z: <http://www.opengeospatial.org/standards/gml>
- [13] Geography Markup Language (GML). *WMS Web Map Service* [online]. 2008 [cit. 2012-05-07]. Dostupné z: <http://geo3.fsv.cvut.cz/wms/index.php?clanek=gml>
- [14] Main Page. *GeoRSS* [online]. 2011 [cit. 2012-05-07]. Dostupné z: <http://www.georss.org/>
- [15] *Google App Engine* [online]. 2012 [cit. 2012-05-08]. Dostupné z: <https://developers.google.com/appengine/>
- [16] What Is Google App Engine? *Google App Engine* [online]. 2012 [cit. 2012-05-08]. Dostupné z: *OWSLib svn trunk documentation* [online]. 2010, 2012 [cit. 2012-05-08]. Dostupné z: <https://developers.google.com/appengine/docs/whatisgoogleappengine>
- [17] *OWSLib svn trunk documentation* [online]. 2010, 2012 [cit. 2012-05-08]. Dostupné z: <http://owslib.sourceforge.net>
- [18] PagedQuery: Home Page. *Google Code* [online]. 2012 [cit. 2012-05-08]. Dostupné z: <http://code.google.com/p/he3-appengine-lib/wiki/PagedQuery>
- [19] FAJT, Jaromír. GEOMETRICKÉ TRANSFORMACE V GIS. *Fakulta aplikovaných věd: katedra matematiky oddělení geomatiky* [online]. 2005 [cit. 2012-05-11]. Dostupné z: <http://gis.zcu.cz/studium/ugi/referaty/05/GeometrickeTransformace/>
- [20] SEVERANCE, Charles R. *Using Google App Engine*. 1st ed. Sebastopol, CA: O'Reilly, c2009, 241 s. ISBN 05-968-0069-X
- [21] SANDERSON, Dan. *Programming Google App Engine*. 1st ed. Sebastopol, CA: O'Reilly, c2010, 367 s. ISBN 05-965-2272-X
- [22] MITCHELL, Tyler. *Web mapping illustrated*. Sebastopol, Calif.: O'Reilly, c2005, 349 s. ISBN 978-059-6008-659
- [23] HARMS, Daryl. *Začínáme programovat v jazyce Python*. 1. vyd. Brno: Computer Press, 2003, 458 s. ISBN 80-722-6799-X
- [24] LUTZ, Mark. *Naučte se Python*. 1. vyd. Praha: Grada, 2003, 339 s. ISBN 80-247-0367-X

- [25] FU, Pinde a Jiulin SUN. *Web GIS: principles and applications*. 1st ed. Redlands, Calif: ESRI Press, 2010. ISBN 15-894-8245-X

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

XML	Extensible Markup Language
WSDL	Web Services Description Language
HTTP	Hypertext Transfer Protocol
HTML	Hyper Text Markup Language
MIME	Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
UDDI	Universal Description, Discovery, and Integration
SOAP	Simple Object Access Protocol
REST	Representational State Transfer
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
URI	Uniform Resource Identifier
CAM	Computer Aided Cartography
DPZ	Dálkový průzkum Země
API	Application Programming Interface
AJAX	Asynchronous JavaScript and XML
WHATWG	Web Hypertext Application Technology Working Group
OGC	Open Geospatial Consortium
ISO	International Organization for Standardization
ČSN	Československá norma
XHTML	Extensible HTML
SVG	Scalable Vector Graphic
INSPIRE	Infrastructure for Spatial Information in Europe
WMS	Web Map Service
WebCGM	Web Computer Graphics Metafile
WFS	Web Feature Service
WCS	Web Coverage Service
WPS	Web Processing Service
KML	Keyhole Markup Language
GML	Geography Markup Language
EPSG	European Petroleum Survey Group
VML	Vector Markup Language
RSS	RDF Site Summary
GAE	Google App Engine
SQL	Structured Query Language
IDE	Integrated Development Environment
CRS	Coordinate Reference System

## SEZNAM OBRÁZKŮ

Obr. 1. Komponenty webových služeb (zdroj: [1]) .....	13
Obr. 2. Komunikace klientů a serverů přes webové služby .....	19
Obr. 3. Ukázka webového klienta s logy předních prohlížečů .....	21
Obr. 4. Mobilní klienti iPad a iPhone s aplikací iKatastr .....	22
Obr. 5. Desktopová aplikace ArcGis Desktop .....	23
Obr. 6. Zobrazení architektury webového GIS .....	24
Obr. 7. Interoperabilita .....	31
Obr. 8. Obrázek připojených dat z WMS na tenkém klientovi ve webovém prohlížeči .....	33
Obr. 9. Ukázka připojené služby WFS v programu Quantum GIS .....	35
Obr. 10. Ukázka připojených dat z WCS v programu ArcGis .....	36
Obr. 11. Ukázka aplikace města Boston zpracované v ArcGIS Viewer for Flex .....	38
Obr. 12. Ukázka KML v praxi v aplikaci Google Earth .....	39
Obr. 13. Ukázka nástroje pro správu aplikace hostované na GAE .....	46
Obr. 14. Ukázka základní implementace knihovny OpenLayers .....	48
Obr. 15. Diagram balíčků .....	50
Obr. 16. Diagram tříd uživatelského rozhraní .....	57
Obr. 17. Use Case diagram .....	58
Obr. 18. Diagram tříd služeb aplikace z modulu service.py v balíčku services .....	59
Obr. 19. Diagram tříd služeb aplikace z modulu wms.py v balíčku services .....	60
Obr. 20. Diagram tříd pro práci s WMS servery z balíčku „baseClasses“ .....	61
Obr. 21. Pohled na úvodní stránku aplikace WMSkatalog .....	66
Obr. 22. Fulltextové vyhledávání na WMSkatalogu přes „Google Custom Search“ ..	68
Obr. 23. Ukázka zobrazení informace o vrstvě WMS serveru .....	71
Obr. 24. Ukázka zobrazení vrstvy v dynamické mapě .....	72

## SEZNAM PŘÍLOH

- P I.      Obsah přiloženého CD
- P II.     Průvodce nasazením aplikace v prostředí GAE

## **PŘÍLOHA P I. OBSAH PŘILOŽENÉHO CD**

1. Text diplomové práce v souboru **fulltext.pdf**
2. Zdrojové kódy aplikace WMSkatalog v adresáři **SRC**
3. SDK pro vývoj aplikací na platformě Google App Engine v adresáři **GAE**

## PŘÍLOHA P II. PRŮVODCE NASAZENÍM APLIKACE V PROSTŘEDÍ GAE

1. Nainstalovat Google App Engine SDK pro daný OS.
2. Nainstalovat Python verze 2.7.
3. Založit účet v GAE na adrese <http://appengine.google.com/>.
4. Vytvořit aplikaci ve webové administraci GAE.
5. V textovém editoru otevřít soubor „app.yaml“, který se nachází ve zdrojovém kódu aplikace. V tomto souboru přepsat na prvním řádku řetězec „wmskatalog“ na název aplikace vytvořené v kroku č. 4.
6. Otevřít příkazový řádek, změnit aktuální pracovní adresář na adresář se zdrojovými kódy, kde se nachází soubor s názvem „app.yaml“.
7. Spustit příkaz:  

```
python /Cesta_K_SDK_Google_App_Engine/appcfg.py update ./.
```
8. Spuštěný Python skript si vyžádá přihlašovací údaje pro GAE účet a po zadání správném zadání nahraje aplikaci do prostředí GAE.
9. Pomocí webového prohlížeči přistoupíme na URL adresu aplikace <http://NazevAplikaceZKroku4.appspot.com/install>. Tímto docílíme instalaci aplikace.
10. Instalace zajistí vytvoření uživatele „admin“ s heslem „123321“.
11. Aplikace je připravena k používání.