

Implementace cloud API pro sdílení souborů do frameworku QCubed

Bc. David Mališka



Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David Mališka**

Osobní číslo: **A12423**

Studijní program: **N3902 Inženýrská informatika**

Studijní obor: **Počítačové a komunikační systémy**

Forma studia: **prezenční**

Téma práce: **Implementace cloud API pro sdílení souborů do frameworku QCubed**

Téma anglicky: **The Implementation of Cloud API for File-sharing in the QCubed Framework**

Zásady pro vypracování:

1. Prostudujte dostupné cloudové služby pro sdílení souborů (Google Drive, Skydrive, Dropbox) a zpracujte literární rešerši na toto téma.
2. Srovnáním jednotlivých služeb vyberte vhodného kandidáta pro implementaci do vývojového frameworku QCubed.
3. Navrhněte a realizujte třídy pro framework QCubed, jež budou umožňovat přihlášení ke cloudové službě, výpis souborů, upload, mazání a sdílení souborů s ostatními uživateli služby.
4. Zpracujte dokumentaci k vytvořeným třídám.
5. Na ukázkové webové aplikaci demonstруйте funkcionalitu vytvořených tříd a nové možnosti sdílení souborů pomocí cloudové služby ve frameworku QCubed.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. QCubed [online]. [cit. 2014-01-31]. Dostupné z: <http://qcu.be>.
2. LAVIN, Peter. PHP – objektově orientované: koncepty, techniky a kód. 1. vyd. Praha: Grada, 2009. ISBN 978-80-247-2137-8.
3. JQuery: kuchařka programátora. Vyd. 1. Brno: Computer Press, 2010. ISBN 978-80-251-3152-7.
4. VELTE, Anthony T. Cloud computing: praktický průvodce. Vyd. 1. Brno: Computer Press, 2011, 436 s. ISBN 978-80-251-3333-0.
5. Google Disk. [online]. [cit. 2014-01-31]. Dostupné z: <http://www.google.com/drive/about.html>.
6. Dropbox. [online]. [cit. 2014-01-31]. Dostupné z: <https://www.dropbox.com/about>.

Vedoucí diplomové práce:

Ing. Radek Vala

Ústav informatiky a umělé inteligence

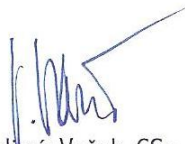
Datum zadání diplomové práce:

7. února 2014

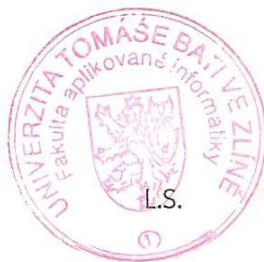
Termín odevzdání diplomové práce:

27. května 2014

Ve Zlíně dne 7. února 2014



prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Karel Vlček, CSc.
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.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Cílem diplomové práce je implementace cloud API pro sdílení souborů do frameworku QCubed. Teoretická část vysvětluje pojem Cloud computing a možnosti jeho využití. Dále je provedena studie vybraných cloudových úložišť a jejich vlastností. V praktické části jsou realizovány třídy pro implementaci Google Drive API do PHP frameworku QCubed. Součástí je také ukázková webová aplikace, která umožňuje přihlášení a odhlášení uživatele, zobrazení souborů nacházejících se na cloudovém disku, procházení složek, nahrání nového souboru, sdílení souboru s ostatními uživateli, stažení, odstranění a obnovení souboru.

Klíčová slova: Cloud computing, framework QCubed, PHP, Google Drive API, webová aplikace

ABSTRACT

The aim of the master's thesis is the implementation of cloud API for file-sharing in the QCubed framework. The theoretical part explains the term Cloud computing and its possible use. Furthermore is creating a study of selected cloud storages and their properties. In the practical part are realized classes to implement Google Drive API in PHP framework QCubed. It also includes sample web application that allows user login and logoff, view the files on the cloud drive, browse folders, upload new files, share files with other users, download, delete, and restore the file.

Keywords: Cloud computing, framework QCubed, PHP, Google Drive API, web application

Rád bych poděkoval panu Ing. Radku Valovi za cenné rady, profesionální přístup a spolupráci při psaní této diplomové práce. Dále děkuji rodině a přátelům za podporu v průběhu studia.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 CLOUD COMPUTING	11
1.1 HISTORIE.....	12
1.2 CHARAKTERISTIKY CLOUD COMPUTINGU	12
1.3 MODELÝ SLUŽEB.....	13
1.3.1 SaaS (Software as a Service).....	14
1.3.2 PaaS (Platform as a Service)	14
1.3.3 IaaS (Infrastructure as a Service)	14
1.3.4 Další modely služeb	15
1.4 MODELÝ NAsAZENÍ.....	16
1.4.1 Soukromý cloud	16
1.4.2 Komunitní cloud.....	17
1.4.3 Veřejný cloud	17
1.4.4 Hybridní cloud	18
1.5 VÝHODY CLOUD COMPUTINGU	19
1.6 NEVÝHODY CLOUD COMPUTINGU	19
2 CLOUDOVÁ ÚLOŽIŠTĚ	21
2.1 GOOGLE DRIVE	22
2.2 MICROSOFT ONEDRIVE.....	23
2.3 DROPBOX	25
2.4 SUGARSYNC.....	27
2.5 COPY.COM.....	28
2.6 WUALA	29
2.7 DALŠÍ ÚLOŽIŠTĚ	30
2.7.1 Box	30
2.7.2 Cubby	30
2.7.3 Capsa.cz	31
3 QCUBED	32
3.1 ZÁKLADNÍ ČÁSTI.....	32
3.1.1 Generátor kódu.....	32
3.1.2 Knihovna uživatelského rozhraní.....	32
3.2 MVC (MODEL-VIEW-CONTROLLER)	32
II PRAKTICKÁ ČÁST	34
4 POROVNÁNÍ ÚLOŽIŠŤ.....	35
5 IMPLEMENTACE GOOGLE DRIVE DO FRAMEWORKU QCUBED	38

5.1	PŘIHLÁŠENÍ KE SLUŽBĚ	38
5.2	KOMUNIKACE S GOOGLE DRIVE	39
5.3	NAHRÁNÍ SOUBORŮ	40
5.4	OBSAH DISKU	43
5.5	PRÁCE SE SOUBORY	50
5.6	TYP INTERNETOVÉHO MÉDIA	55
6	UKÁZKOVÁ WEBOVÁ APLIKACE	57
6.1	NÁVRH APLIKACE.....	57
6.1.1	Funkční požadavky	57
6.1.2	Nefunkční požadavky.....	57
6.2	REALIZACE APLIKACE	57
6.3	POPIS A FUNKCIONALITA	59
6.3.1	Přihlášení.....	59
6.3.2	Nahrání souboru	60
6.3.3	Sdílení souboru.....	61
6.3.4	Odstranění a obnovení souboru.....	61
6.3.5	Volba jazyka.....	62
6.3.6	Stažení souboru	63
6.3.7	Odhlášení.....	63
	ZÁVĚR	64
	SEZNAM POUŽITÉ LITERATURY.....	65
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	68
	SEZNAM OBRÁZKŮ	70
	SEZNAM TABULEK.....	71
	SEZNAM PŘÍLOH.....	72

ÚVOD

Řada uživatelů PC využívá denně online služby jako například e-mail, chat nebo sociální síť. Využívání těchto služeb na Internetu se souhrnně označuje jako Cloud computing. Pro toto slovní spojení existuje mnoho definic a jen těžko se hledá výstižný český překlad. Slovo „cloud“ znamená oblak a význam slova „computing“ je práce s počítačem. Spojením těchto slov vzniká pomyslná „kancelář v oblacích“. Do cloudu se lze připojit prakticky s jakýmkoliv zařízením, které má přístup na Internet. Uživatel tak nemusí zůstat připoután pouze ke svému stolnímu počítači, ale ke slovu přicházejí chytré telefony, tablety a notebooky.

Základním atributem všech služeb Cloud computingu by měla být schopnost poskytovat je „na vyžádání“ ve formě a objemu, které jsou adekvátní okamžitému požadavku zákazníka. Proto se poskytovatelé cloudových služeb většinou snaží zabezpečit automatizované zřizování prostřednictvím tzv. samoobslužných portálů bez potřeby komplikované komunikace se zákaznickým centrem. Pravděpodobně nejatraktivnějšími vlastnostmi služeb Cloud computingu je jejich škálovatelnost a flexibilita. Uživatel má možnost kdykoliv si alokovat dodatečnou úložnou kapacitu a výpočetní zdroje podle aktuální potřeby a později je flexibilně opět vrátit.

Právě poskytování flexibilní úložné kapacity je velmi perspektivním odvětvím. Objem digitálních dat neustále roste a ukládání dat do cloudového úložiště přináší mnoho výhod jako například vysokou dostupnost, lepší a častější zálohování oproti disku domácího počítače, verzování (možnost vrátit se ke starší verzi dokumentu), obnovu smazaných dat, nebo spolupráci s více uživateli (sdílené složky), či snadné sdílení souborů přes web. Jelikož se zvětšuje počet uživatelů využívající této služby, roste také počet provozovatelů a cena za poskytovanou kapacitu klesá. Některé společnosti nabízejí základní úložný prostor zdarma, popřípadě k vyzkoušení na omezenou časovou dobu. Cloudový disk se tak stává dostupný nejen pro firmy, ale i pro domácnosti.

Cílem diplomové práce je prozkoumat dostupné poskytovatele cloudových úložišť a vybrat vhodného kandidáta pro implementaci do vývojového frameworku QCubed. Následně provést realizaci tříd umožňujících práci s cloudovým diskem a jejich funkcionalitu ověřit na webové aplikaci.

I. TEORETICKÁ ČÁST

1 CLOUD COMPUTING

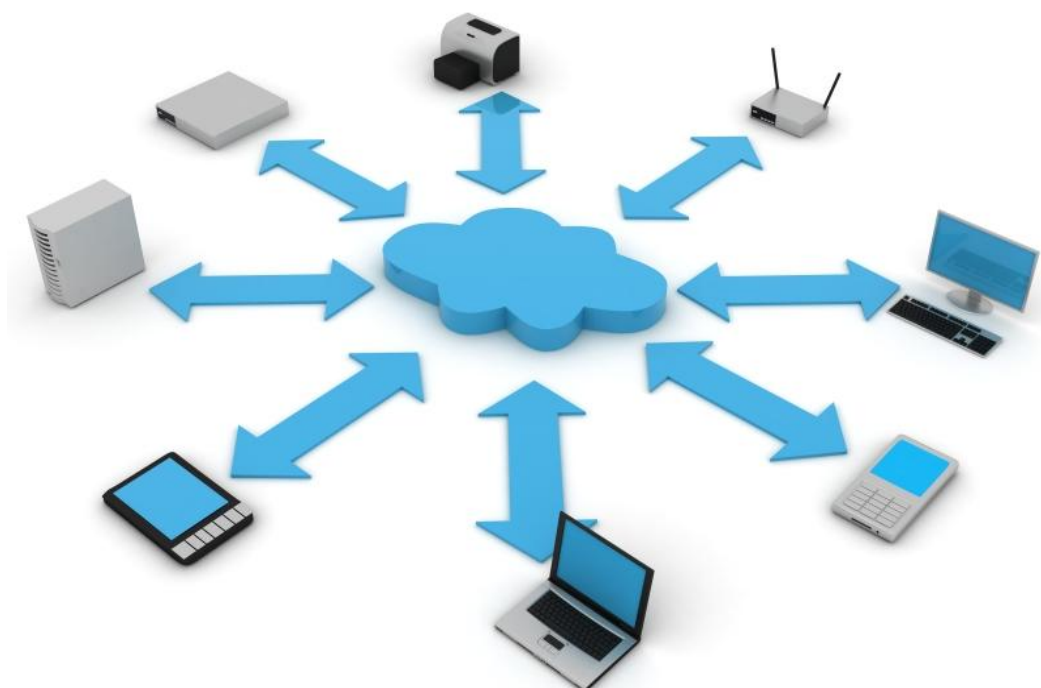
Cloud computing je v současné době často používané slovní spojení, ke kterému existuje mnoho definic. Americký národní standardizační institut (NITS) schválil definici CC v roce 2011, která zní:

„Cloud computing je model umožňující všudypřítomné, pohodlné, na vyžádání (on-demand) síťové připojení nastavitelných počítačových zdrojů (např. sítí, serverů, úložných zařízení, aplikací a služeb), které lze rychle spustit s minimálním úsilím věnovaným správě nebo interakci s poskytovatelem služeb. Tento cloudový model je složen z pěti základních charakteristik, tří modelů služeb a čtyř distribučních modelů.“ [2]

Z českých zdrojů stojí jistě za zmínku dvě definice:

„Cloud computing je nový způsob využívání zdrojů (zejména hardware, software) v IT, vycházející z možnosti jejich sdílení mezi aplikacemi a odstranění přímé vazby aplikační logiky na fyzické komponenty (virtualizace).“ [3]

„Cloud computing je jedno z velmi oblíbených marketingových spojení v IT průmyslu. Cloud computing je sdílení hardwarových i softwarových prostředků pomocí sítě. Právě tento princip dal systému název, v diagramu se podobá mraku (anglicky cloud).“ [4]



Obrázek 1: Cloud computing [10]

1.1 Historie

První myšlenku o fungování Cloud computingu vyslovil americký informatik John McCarthy. V roce 1960 prohlásil, že by se v budoucnosti mohla počítačová technika organizovat podobně jako veřejná služba. Jeho idea spočívala ve sdílení počítačových technologií s podobnou logikou jako například sdílení elektrické energie. Elektrickou energii potřebuje řada domácností a firem, které mají zakoupeny nejrůznější elektrické spotřebiče. Málokterá z nich si ale kvůli tomu pořizuje vlastní elektrárnu. Mnohem častější je model, kdy jednu elektrárnu využívají až desetitisíce odběratelů, kteří se k ní připojují vzdáleně – pomocí elektrorozvodné sítě. Díky tomuto přes padesát let starému srovnání počítačů, elektřiny a podobných služeb souhrnně nazývaných v angličtině utility, se Cloud computing někdy označuje jako Utility computing. [5] [6]

První krok k revoluci udělaly telekomunikační organizace, které v 90. letech jako první učinily posun od datových okruhů typu point-to-point do virtuální privátní sítě (VPN). Díky optimalizaci využití zdrojů tak mohly svoji práci uskutečňovat efektivněji a levněji. Samotný termín Cloud computing byl poprvé použit na přednášce profesora Ramnath Chellappa v roce 1997, který ho definoval jako výpočetní paradigma, kde se hranice výpočetního výkonu budou stanovovat spíše z ekonomického hlediska, než z hlediska technických omezení. [5] [6]

První reálný cloud computing vnikl v roce 1999, kdy firma Salesforce.com představila koncept poskytování podnikových aplikací přes webové stránky. Po ní spustila firma Amazon v roce 2002 Amazon Web Services, dále následoval v roce 2006 Google s produktem Google Docs a Amazon Elastic Compute Cloud (EC2). Firma Eucalyptus vytvořila v roce 2008 první open source API pro nasazení privátních cloudů a o pár měsíců později OpenNebula vybudovala open source software pro nasazení privátních a hybridních cloudů. O rok později se přidal také Microsoft se službou Windows Azure.

V současnosti existují stovky poskytovatelů cloud computingu, mezi kterými je i česká firma Capsa.cz fungující od roku 2013. [1] [5]

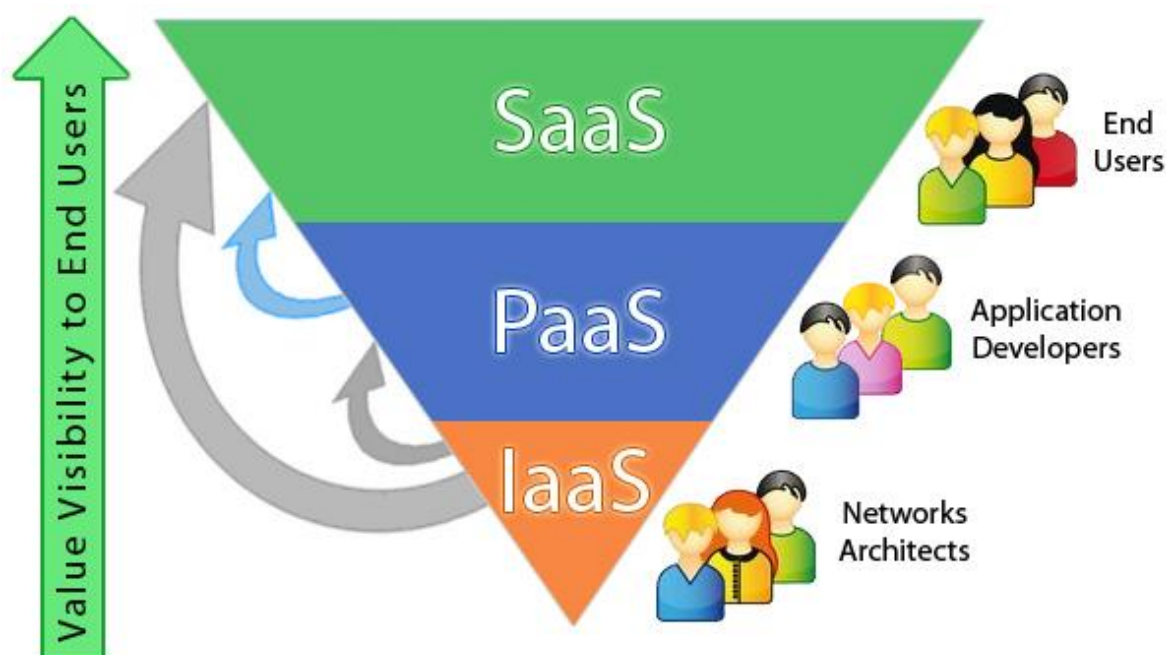
1.2 Charakteristiky Cloud computingu

- On-demand self-service (dostupné na vyžádání, samoobslužné) – služby jsou pro koncového uživatele dostupné dle jeho vlastní potřeby automaticky, bez nutnosti lidské interakce.

- Broadband network access (široký síťový přístup) – služby jsou přístupné přes počítačovou síť z tlustých i tenkých klientů (např. mobilní telefon, tablety, notebooky a PC).
- Resource pooling (sdílení zdrojů) – zdroje jsou sdíleny více zákazníky s různými fyzickými a virtuálními prostředky, dynamicky přiřazovány dle poptávky.
- Rapid elasticity (rychlé přizpůsobení) – schopnost automaticky rychle reagovat na zvýšenou poptávku po zdrojích působí z pohledu zákazníka jako neomezená kapacita zdrojů.
- Measure service (měřená služba) – schopnost monitorovat, řídit a hlásit množství spotřebovaných prostředků poskytovateli i zákazníkovi (např. volná kapacita úložiště, počet aktivních uživatelů, apod.). [2]

1.3 Modely služeb

Modely služeb se zabývají pronájmem výpočetního času a prostředků (hardware i software) poskytovatele cloudového řešení. Zdroje se jeví z hlediska běžného uživatele jako nekonečné, jelikož je možné prostředky dynamicky měnit. Zde je rozdíl oproti běžnému webhostingu, navíc účtování se provádí po menších časových jednotkách. Cloudové technologie umí pružně reagovat na potřebu skokového nárůstu zátěže a zajistit větší přenosové pásmo po dobu zvýšené zátěže. [6] [7]



Obrázek 2: Modely služeb a typy uživatelů [11]

1.3.1 SaaS (Software as a Service)

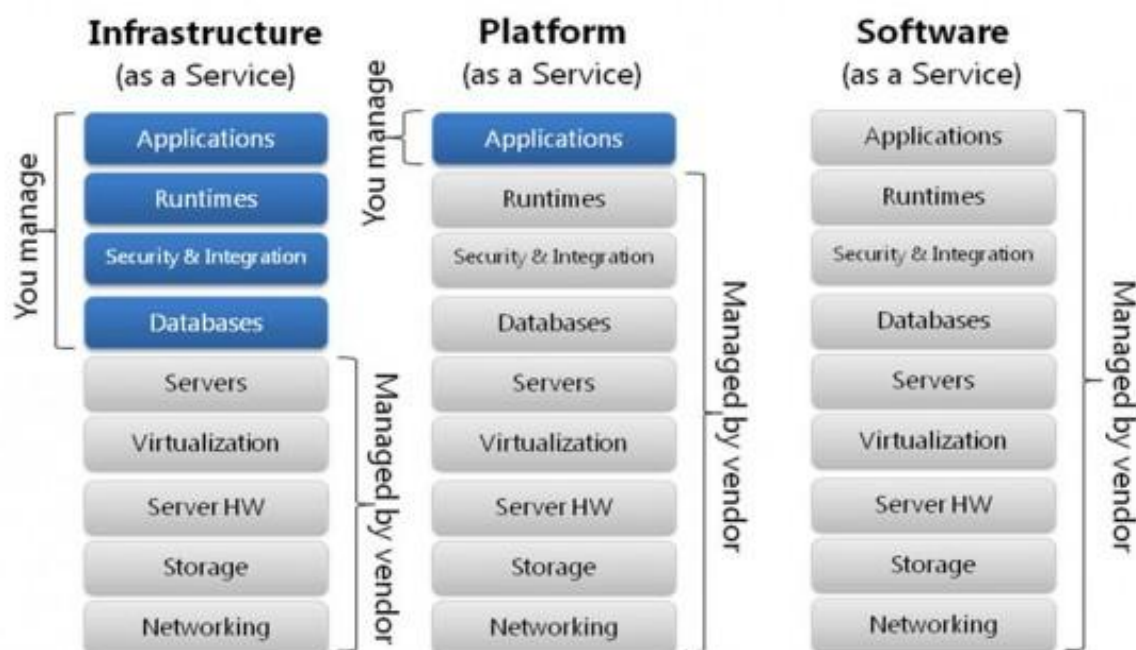
Software jako služba – aplikace je nabízena jako služba zákazníkům, kteří k ní přistupují přes Internet. Uživatel využívá software na dálku jako službu. Důvodem vzniku bylo snížení nákladů na software, např. není potřeba koupě licencí programů. Správu a podporu aplikace zajišťuje poskytovatel. Nevýhodou však může být rozhodnutí poskytovatele změnit aplikaci. Příkladem je sada aplikací Google Apps.

1.3.2 PaaS (Platform as a Service)

Platforma jako služba – poskytovatelé nabízí zákazníkům aplikace, které však nemají sloužit koncovým uživatelům, ale vývojářům. Cílem je poskytnout prostředky pro vývoj, testování, implementaci a hostování aplikací přes Internet, bez možnosti stažení softwaru. U poskytovatele této služby lze vytvořit, hostovat a udržovat aplikaci s tím, že zákazník má k dispozici i integraci webových služeb, databázi, zajištěnou bezpečnost, úložiště, škálovatelnost, správu verzí, apod. Nevýhodou této služby je závislost na konkrétním poskytovateli a jeho konkrétním řešení. Přenos vytvořené aplikace od jednoho poskytovatele k druhému nemusí být možný, a pokud poskytovatel ukončí svou činnost, zákazník ztratí aplikaci i data. Příkladem jsou Google App Engine a Force.com.

1.3.3 IaaS (Infrastructure as a Service)

Infrastruktura jako služba – poskytovatelé v tomto modelu služeb nabízejí infrastrukturu (počítače, síťová zařízení, úložné místo, atd.), ke kterým se přistupuje přes síť z PC jednotlivých uživatelů. Tato varianta je vhodná pro uživatele, kteří vlastní software, ale nechtějí se starat o hardware. Za služby se obvykle platí na základě užitku, takže poskytovatelé je účtují podle spotřebovaných prostředků (procesor, paměť, úložiště atd.). Veškeré parametry lze sledovat pomocí jednoduchých nástrojů a měnit jejich vlastnosti pomocí virtualizace. Příkladem jsou Amazon Web Services a Windows Azure.



Obrázek 3: Modely služeb [7]

1.3.4 Další modely služeb

Kromě zmíněných tří základních modelů služeb se využívají také další modely. Příkladem mohou být:

- STaaS (Storage as a Service) – úložiště jako služba. Poskytovatel pronajímá datový prostor, který může sloužit také jako řešení problému zálohování dat mimo pracoviště.
- SECaaS (Security as a Service) – bezpečnost jako služba. Poskytovatel nabízí síťové zabezpečení jako například anti-virus, anti-spyware nebo detekce narušení bezpečnosti.
- DaaS (Data as a Service) – data jako služba. Umožňuje zákazníkovi využívat data od poskytovatele.
- TEaaS (Test Environment as a Service) – testovací prostředí jako služba. Umožňuje klientovi testovat software.
- APIaaS (Application Programming Interfaces as a Service) – API jako služba. Poskytuje rozhraní pro programování aplikací.
- BaaS (Backend as a Service) – backend jako služba. Poskytuje vývojářům webové a mobilní aplikace.

- AaaS (Analytics as a Service) – analýza jako služba. Poskytuje zákazníkovi analytické nástroje.

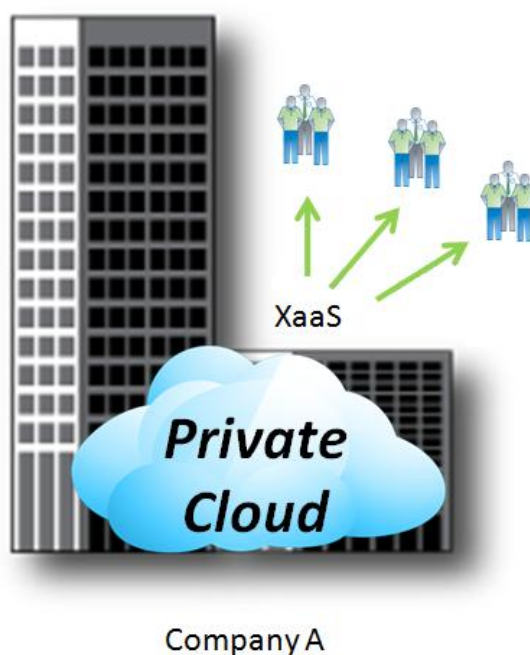
Z výše jmenovaných modelů služeb je velmi zajímavý model Storage as a Service, kdy velká společnost pronajme prostor ve své infrastruktuře menší firmě, nebo jednotlivci. Hlavní výhodou STaaS je úspora nákladů – personální a hardwarové. Dále je také hodně využívána k zajištění dlouhodobého uschování dat. Příklad provozovatelů: GoogleDrive, Microsoft OneDrive a Dropbox. [8] [9]

1.4 Modely nasazení

Modely nasazení popisují, jak je cloud poskytován a dělí se do čtyř základních kategorií.

1.4.1 Soukromý cloud

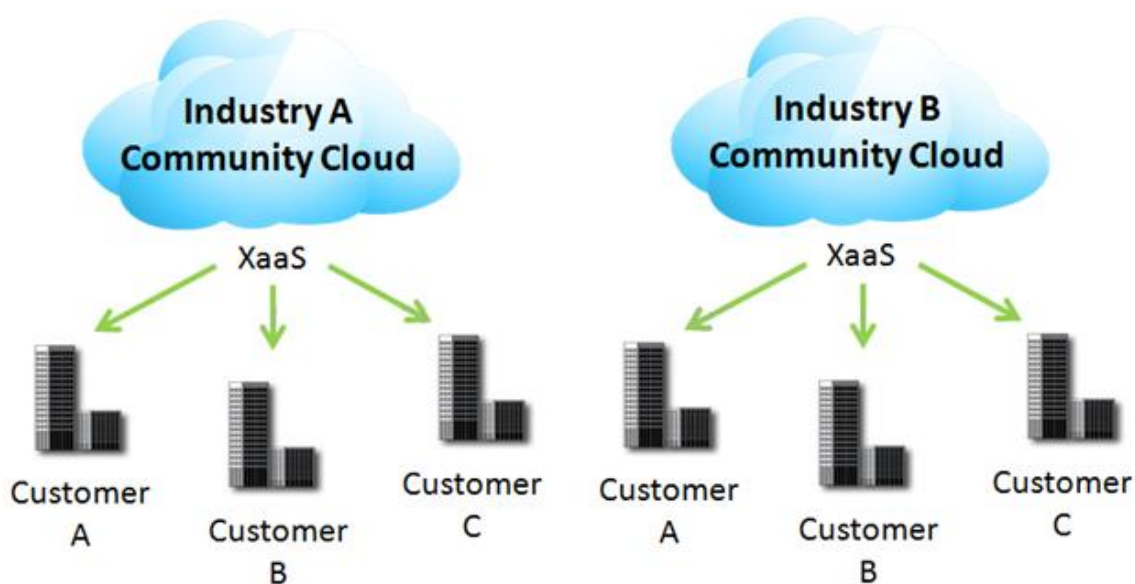
Soukromý cloud je navrhnut a využíván v rámci nějaké organizace. Hostován a spravován může být samotnou organizací nebo třetí stranou. Důvodem vzniku je obava společnosti o bezpečnost svých dat, popřípadě o místo uložení dat. U tohoto konceptu zaniká výhoda klasického Cloud computingu – nižší náklady na provoz, které pramení z pronájmu služeb více nájemníkům. Výhodou však jsou možnosti přizpůsobení potřebám zákazníka, které u veřejných cloudových služeb nejsou možné. Další uplatnění lze nalézt ve veřejné správě. [12]



Obrázek 4: Grafické zobrazení soukromého cloudu [12]

1.4.2 Komunitní cloud

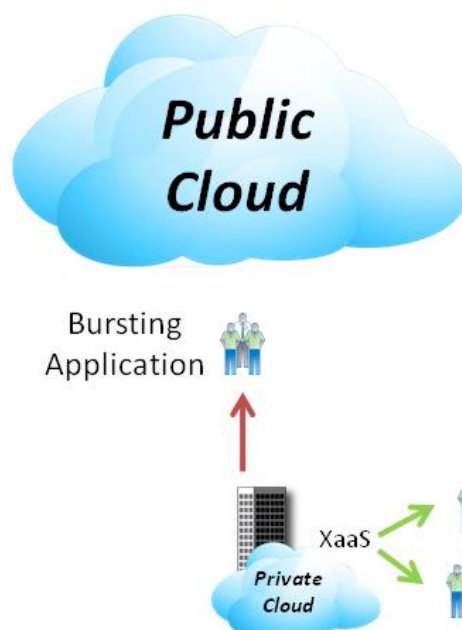
Komunitní cloud se hodí pro organizace, které mají například stejný obor podnikání nebo stejnou bezpečnostní politiku. Potřebná infrastruktura cloudu je zde sdílena těmito organizacemi a veškeré náklady jsou tak rozděleny mezi tyto organizace. Velmi výhodný je pro malé firmy, kterým nejen že pomůže efektivně snížit náklady, ale také poskytne vysoce kvalitní IT služby větší společnosti. [12]



Obrázek 5: Grafické zobrazení komunitního cloudu [12]

1.4.3 Veřejný cloud

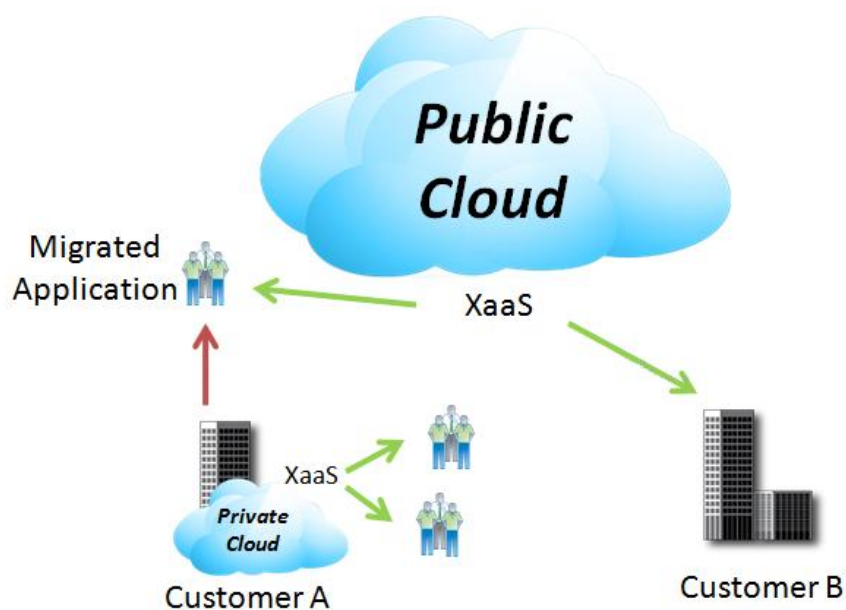
Veřejný cloud je model, který používá v podstatě každý, kdo využívá Internet. Tato koncepce poskytuje službu široké veřejnosti a to zpravidla ve stejném rozsahu, tzn. se stejnou funkcionalitou – všichni mají stejné podmínky. Někdy se jedná o služby, které by lidi ani nezařadili do Cloud computingu. Jde například o portál seznam.cz, nebo datové úložiště uloz.to apod. [12]



Obrázek 6: Grafické zobrazení veřejného cloudu [12]

1.4.4 Hybridní cloud

Hybridní cloud spojuje soukromý a veřejný cloud. Obvykle je realizován dvěma způsoby: dodavatel má privátní cloud a vytváří partnerství s poskytovatelem veřejného cloudu nebo poskytovatel veřejného cloudu tvoří partnerství s dodavatelem, který poskytuje privátní cloud platformy. Navenek toto spojení vystupuje jako jeden cloud, avšak oba modely jsou propojeny pomocí standardizačních technologií. [12]



Obrázek 7: Grafické zobrazení hybridního cloudu [12]

1.5 Výhody Cloud computingu

- Snížení nákladů – úspora financí za nákup softwaru a hardwaru. Dále také snížení provozních nákladů (např. energie na provoz serveru) a mezd pracovníků, starajících se o chod infrastruktury.
- Škálovatelnost – možnost dynamicky měnit kapacitu služeb v reálném čase za pomoci webového formuláře.
- Dostupnost – poskytovatel zpravidla garantuje velmi vysokou dostupnost, které dosáhne seskupením několika serverů.
- Jednoduchost – využití přehledných webových formulářů.
- Žádná fyzická správa a údržba.
- Rychlost uvedení do provozu – služby jsou dostupné prakticky ihned po registraci, popř. zaplacení požadované částky.
- Mobilita – ke službě lze přistoupit z různých míst a různých zařízení (např. mobilní telefon, tablety, notebooky a PC).
- Sdílení prostředků – lepší rozdělení výkonu. Uživatel platí pouze za výkon, který potřebuje.
- Bezpečnost – data bývají několikanásobně zálohovaná a velké data centrum je zpravidla lépe zabezpečeno, než jeden počítač.
- Nejnovější aktualizace – poskytovatel sám aktualizuje funkce a uživatel se nemusí o nic starat.
- Vysoká kapacita pro uschování dat. [1] [6] [13]

1.6 Nevýhody Cloud computingu

- Výpadek nebo bankrot – poskytovatel může mít výpadek, nebo úplně zbankrotovat. Selhat ovšem může také osobní pevný disk.
- Zneužití dat – data mohou být zneužita nepovolanými osobami. Lze tomu zabránit šifrováním dat.
- Putování dat Internetem – veškerá data jsou posílána přes Internet a mohou být zachycena třetí stranou. Opět je zde řešením šifrování.
- Nevýhodné platby – v některých případech mohou měsíční nebo roční platby přesáhnout cenu jednorázového nákupu klasického hardwaru nebo softwaru.

- Závislost na Internetovém připojení – při špatném signálu může software běžet pomaleji. Úplný výpadek Internetového připojení může zamezit možnosti pracovat.
- Méně funkcí – i přes rychlý vývoj může mít online aplikace méně funkcí než klasická desktopová verze. [1] [6]

2 CLOUDOVÁ ÚLOŽIŠTĚ

Cloudové úložiště je typ modelu služeb Storage as a Service (česky úložiště jako služba), při kterém velká společnost pronajímá datový prostor ve své infrastruktuře menší firmě, nebo jednotlivci. V dnešní době existují stovky provozovatelů cloudových úložišť. Někteří z nich mají specifické zaměření ukládání dat (např. e-mailové zprávy, fotografie, videa apod.), jiní nabízejí uložení všech forem digitálních dat. [9] [15]



Obrázek 8: Cloudové úložiště [14]

Při ukládání dat odešle zákazník kopie svých souborů pomocí webového rozhraní (popř. nainstalované ulity) přes Internet do datového centra. Tyto centra větších poskytovatelů se skládají ze stovek serverů, které jsou připojeny k Internetu s vysokou konektivitou a garantovanou dostupností – ta je zajištěna geografickým rozmístěním serverů. Aby se zamezilo ztrátě nebo poškození dat, jsou informace ukládány redundantně na více serverů. Pokud si zákazník přeje získat uložená data zpět, přistupuje k datovému centru opět pomocí webového rozhraní (popř. nainstalované ulity) přes Internet. Server následně odešle soubory zpět klientovi, nebo umožňuje přístup a manipulaci s daty přímo v datovém úložišti. [15]

2.1 Google Drive

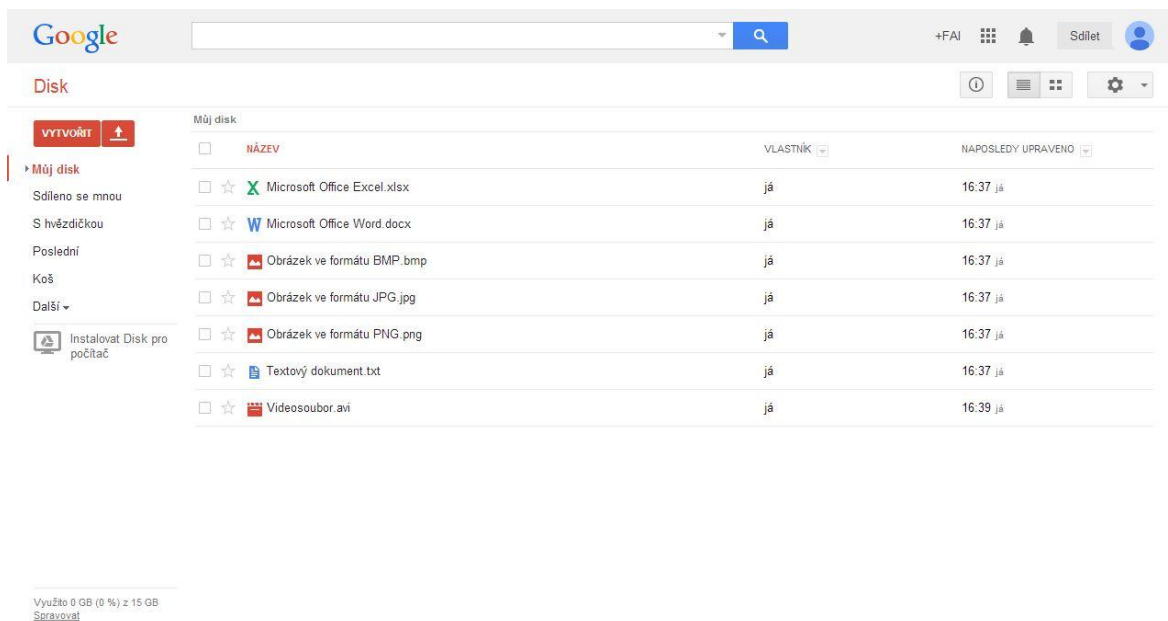
Datové úložiště od společnosti Google (do češtiny přeloženo jako Disk Google) vzniklo 24. dubna 2012. K tomuto úložišti lze přistupovat díky webovému rozhraní prostřednictvím libovolného moderního webového prohlížeče. Webové rozhraní úložiště Google Drive navazuje na službu Google Docs, jejíž kancelářské webové aplikace měly také své vlastní internetové úložiště a webové prostředí pro jeho správu. Dále spojuje tři nejčastěji používané služby firmy Google: Gmail, Google+ a Picassa Web Albums. [16] [17]



Obrázek 9: Google Drive logo [16]

Uživatelé nabízí zdarma kapacitu 15 GB, kterou lze rozšířit až na 30 TB s ročním paušálem. Maximální velikost souboru, kterou lze uložit, je omezena na 10 GB. Zajímavostí je, že ne všechny soubory se započítávají do celkové kapacity – např. nativní formáty Google Docs (Gdoc, Gslides, Gsheet), fotky v Google+, které mají menší rozlišení než 2048 x 2048 pixelů a videa kratší než 15 minut se nepočítají do limitu.

Prostřednictvím webového rozhraní lze spravovat soubory v úložišti, sdílet je s dalšími uživateli nebo je veřejně publikovat. K dispozici je webový prohlížeč obrázků a přehrávač multimédií podporující vybrané frekventované formáty souborů. Nové soubory lze do webového úložiště jednoduše nahrávat jejich přetažením myši do okna webového prohlížeče. Soubory je možné také stahovat. Jak jednotlivě, tak hromadně, kdy jsou zabaleny do jednoho ZIP archivu pro lepší manipulaci s nimi. [16] [17]



Obrázek 10: Webové rozhraní GoogleDrive

Přímo ve webovém prohlížeči lze také vytvářet nové soubory a ukládat je do úložiště díky sadě webových aplikací Google Docs. Ta obsahuje textový editor, tabulkový kalkulač, nástroj pro práci s prezentacemi a editor kreseb – zde se nejedná o klasický grafický editor, s jehož pomocí by bylo možné upravovat fotky, ale spíše nástroj pro tvorbu a úpravu kancelářské grafiky. Díky Google Docs lze též prohlížet a upravovat dokumenty, tabulky a prezentace přítomné v úložišti. Navíc díky službě Google Cloud Print lze vytisknout dokument, tabulku či prezentaci na dálku přes Internet na tiskárně. [29] [30]

Mimo přístupu přes webové rozhraní nabízí toto úložiště instalaci klienta pro operační systémy Windows a OS X a pro mobilní platformy Android a iOS. Ostatní platformy oficiálně podporovány nejsou, ale je možné nalézt neoficiální klienty, popřípadě si vytvořit vlastní skript díky veřejnému API.

Nevýhodou je chybějící ochrana souborů heslem, vzájemná synchronizace složek a volitelné omezení přenosové rychlosti, aby při přenášení většího objemu dat nedocházelo k plnému využití internetového připojení. [16] [17]

2.2 Microsoft OneDrive

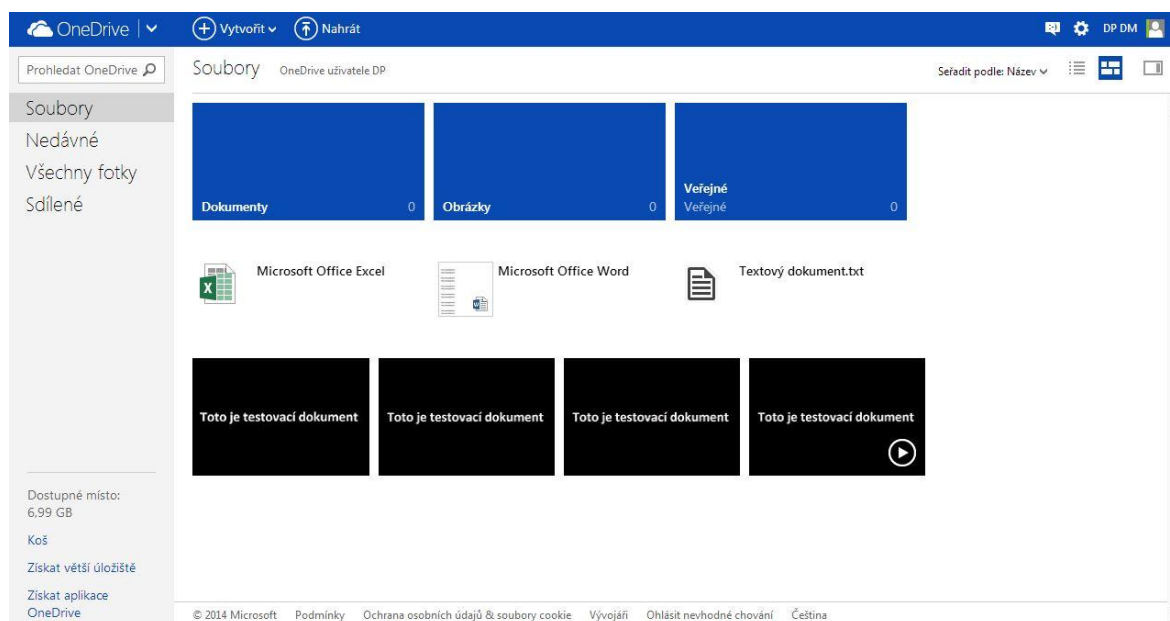
Datové úložiště od firmy Microsoft vzniklo 1. srpna 2007 pod názvem Windows Live Folders. Hned po 8 dnech bylo přejmenováno na Windows Live SkyDrive a pod tímto názvem vydrželo až do června 2013, kdy se firma Microsoft dostala do soudního sporu s Britskou televizí společností BSkyB – British Sky Broadcasting Group plc. Ve sporu šlo

o slovo Sky (česky nebe), které porušovalo ochrannou známku. Firmy nakonec dopěly k dohodě a od 19. února je datové úložiště používáno pod názvem OneDrive. [19] [20]



Obrázek 11: OneDrive logo [19]

K úložišti lze přistupovat díky webovému rozhraní přes webový prohlížeč a jako jedno z mála je optimalizované pro klasické ovládání myší a klávesnicí a také pro dotykové ovládání. Při registraci nabízí zdarma kapacitu 7 GB s možností rozšíření o 200 GB s ročním paušálem. Bezplatně lze kapacitu navýšit o 500 MB za každého registrovaného přítele do maximálního limitu 5 GB, nebo o 3 GB při povolení automatického nahrávání fotek z mobilních zařízení využívající OneDrive. Soubory v koši se nepočítají do celkové kapacity, avšak jsou zachovány pouze po dobu 3 – 30 dnů. Poté jsou nenávratně smazány. [19] [20]



Obrázek 12: Webové rozhraní Microsoft OneDrive

Další možností přístupu je klientská aplikace OneDrive běžící na operačních systémech Windows a OS X a na mobilních platformách Android, iOS, Windows Phone a Xbox.

Mimo klasického cloudového úložiště podporuje OneDrive všechny programy z kancelářské sady Microsoft Office 2013 a manipulaci s nimi. Dále programy Movie Maker a Photo Gallery ze sady Windows Essentials podporují ukládání videí a fotek do úložiště OneDrive, odkud je lze volitelně sdílet s přáteli popřípadě prohlížet a přehrávat na televizní obrazovce díky herní konzoli Xbox 360. Při využití e-mailové služby Outlook lze při odesílání objemných souborů spolupracovat s OneDrive – soubor se neposílá celý v příloze, ale vygeneruje se odkaz ke stažení z úložiště. Přímou ve webovém prohlížeči lze pracovat s dokumenty pomocí webové kancelářské sady Office Web Apps, která přináší zjednodušené verze textového editoru Word, tabulkového kalkulátoru Excel, programu pro práci s prezentacemi PowerPoint a OneNote pro správu poznámek. [29] [30]

Zajímavostí je funkce zvaná fetch, která umožní uživateli na dálku přes webový prohlížeč přistupovat ke všem datům uloženým v počítači, na kterém je nainstalována ulita OneDrive. Takto lze přistupovat i k datům na externím zařízení připojeným přes USB nebo k datům na médiu v optické mechanice. Podmínkou je pochopitelně zapnutý PC a přístup k Internetu.

Nevýhodou OneDrive je, že nepodporuje stále hodně rozšířený operační systém Windows XP nebo Linux. Dále také chybí možnost synchronizace různých složek, popřípadě různých disků a volitelné omezení přenosové rychlosti. [19] [20]

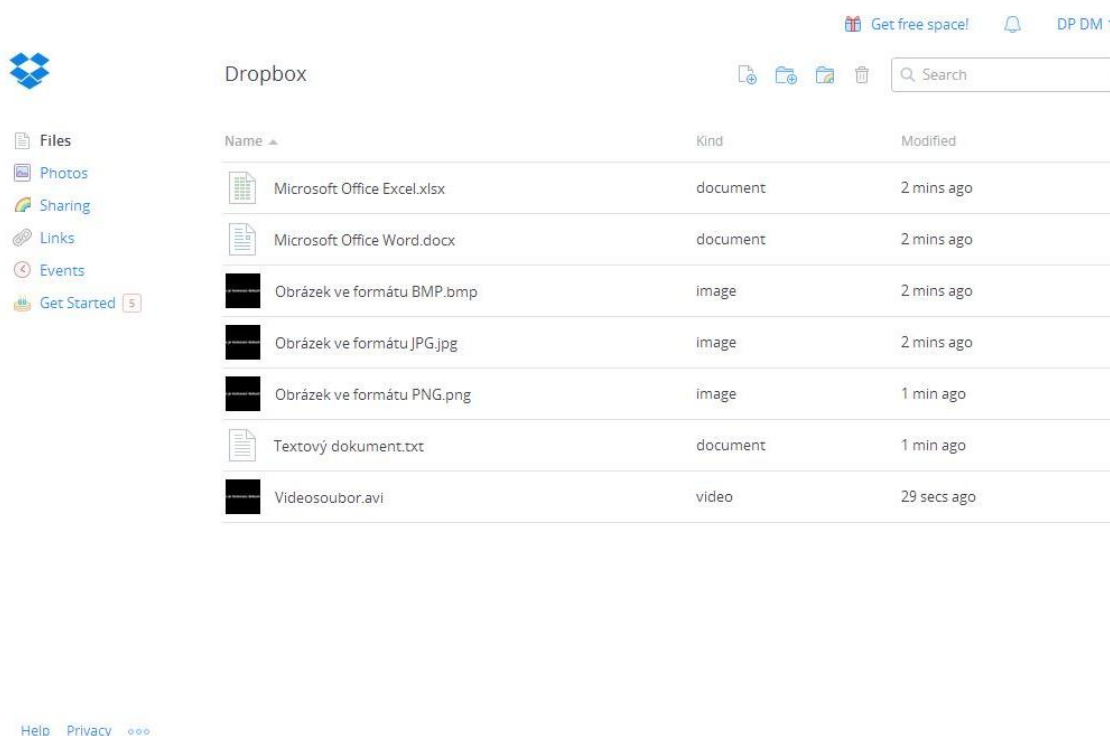
2.3 Dropbox

Dropbox je jedním z prvních cloudových úložišť, které vytvořil Drew Houston při studiu vysoké školy v roce 2007. Na rozdíl od konkurence se výhradně specializuje na synchronizaci dat mezi různými počítači. K tomu používá primárně ulitu podporující operační systémy Windows, OS X a Linux. Z mobilních systémů podporuje Android a iOS. Po registraci nabízí uživateli zdarma 2 GB datového prostoru, které lze rozšířit několika způsoby. Kromě standardního dokoupení 100 GB s měsíčním paušálem, je zde možnost získání 500 MB za každého nově registrovaného přítele a to až do limitu 16 GB. Další možností je také spárování účtu s Twitterem nebo Facebookem za které získá uživatel 1 GB. [22]



Obrázek 13: Dropbox logo [22]

Přístup k úložišti je umožněn mimo klienta také pomocí webového rozhraní, které je velmi jednoduché a mimo manipulace se soubory obsahuje pouze nástroj k prohlížení a publikování obrázků. Naopak synchronizace je velmi propracovaná a umožňuje synchronizaci dat prostřednictvím místní počítačové sítě (LAN). Data tedy nemusí proudit přes Internet, pokud jsou počítače ve stejné síti, což vede ke značnému zrychlení. Nespornou výhodou je také možnost pozastavit synchronizaci, popřípadě nastavit limit odesílání i přijímání dat. [22] [29]



Obrázek 14: Webové rozhraní Dropbox

Mezi další funkce oproti ostatním konkurentům patří automatická synchronizace z připojených paměťových medií po jejich připojení k počítači, nebo možnost zabezpečení pomocí čtyřmístného hesla. Toto heslo je označováno jako PIN, který odstraňuje povinnost

neustálého zadávání dlouhého uživatelského jména a hesla. Navíc služba nabízí možnost, kdy při několikatém chybném zadání tohoto hesla automaticky smaže data ze zařízení, které pocházejí z Dropboxu.

Nevýhodou je malá kapacita pro základní uživatele, velmi strohé webové rozhraní a nemožnost synchronizace různých složek, popřípadě různých disků. [22]

2.4 SugarSync

Datové úložiště SugarSync vzniklo v roce 2008, v ČR není však mnoho rozšířené i přesto že je nainstalováno na noteboocích firmy Lenovo. Předchůdcem této služby byla firma Sharpcast a konkrétně produkt Sharpcast Photos, sloužící k synchronizaci fotografií z PC a mobilních zařízení. SugarSync nenabízí žádnou stálou datovou kapacitu zdarma. Registrovaný uživatel si může vyzkoušet 30-ti denní zkušební verzi, poté jsou k dispozici placené varianty s kapacitou 60 GB až 1 TB. [23]



Obrázek 15: SugarSync logo [23]

Hlavní odlišností od konkurence je synchronizace dat. SugarSync nevytváří pouze jednu speciální složku na pevném disku, ale umožňuje synchronizovat kterékoli již existující složky, popřípadě i na různých discích. Jednoduchý přístup je zajištěn díky tomu, že se služba namapuje jako virtuální pevný disk. Dále SugarSync podporuje omezení rychlosti přenosu souborů, avšak pouze upload a disponuje také zajímavou funkcí obnovy nechtěně smazaných souborů. Další výhody jsou například volitelné uzamykání pomocí čtyřmístného PINu, nebo anti-theft ochrana, která umožňuje při odcizení přenosného zařízení vyslat příkaz, který smaže veškerá data pocházející z SugarSync nacházející se na odcizeném zařízení. [23] [29]

Synchronizační klient podporuje systémy Windows a OS X, chybí však Linux a česká jazyková verze. Z mobilních platforem jsou k dispozici verze pro Android, iOS,

BlackBerry, Symbian a Windows Phone. SugarSync také spolupracuje s firmou Samsung a je k dispozici ulita pro Samsung SmartTV.

Nevýhodou je velmi jednoduché webové rozhraní, pouze se základním prohlížečem fotografií a již zmiňovaná absence bezplatné kapacity pro data. [23]

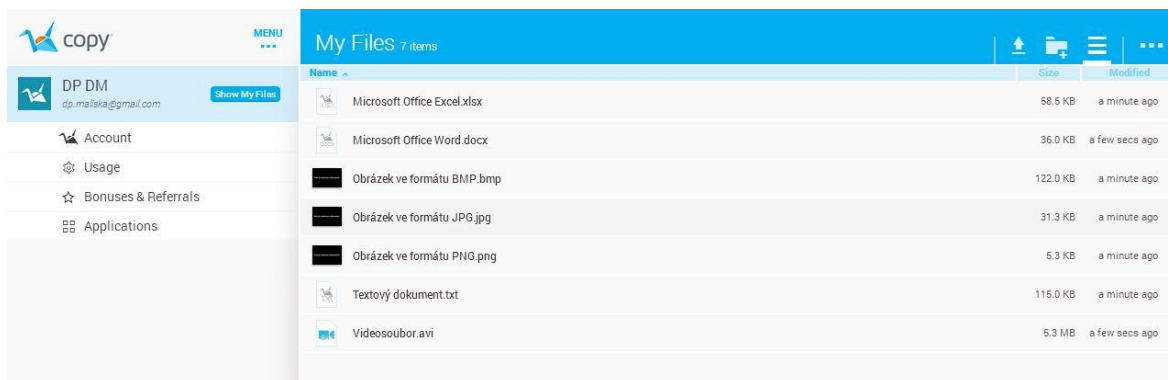
2.5 Copy.com

Copy.com je datové úložiště od firmy Barracuda Networks, fungující od ledna 2013. Jelikož se jedná o poměrně novou službu, snaží se nalákat uživatele velmi vysokými úložnými kapacitami. Nově registrovanému zákazníkovi nabízí bezplatnou kapacitu 15 GB, kterou lze rozšířit o 5 GB při instalaci některé z ulit. Další gigabity lze získat referenčním programem, při kterém uživatelé obdrží 5 GB za nově registrovaného klienta. Klasickým měsíčním paušálem lze kapacitu úložiště navýšit o 500 GB. [24]



Obrázek 16: Copy.com logo [24]

Synchronizační ulita je k dispozici pro platformy Windows, OS X i Linux a podporuje synchronizaci více složek. Další funkce jsou automatické zálohování přenosných médií, omezení rychlosti stahování a odesílání dat, pozastavení synchronizace nebo volitelná synchronizace v lokální síti. Z mobilních zařízení jsou podporovány Android, iOS a Windows Phone. [29]



Obrázek 17: Webové rozhraní Copy.com

Zajímavostí této služby je tzv. férové sdílení, které při sdílení dat s dalšími uživateli přináší úsporu volného prostoru ve vlastním úložišti. Kapacita složky sdílené s dalšími uživateli je rozpočítána mezi jednotlivé uživatele a jejich úložiště. Naopak nevýhodou je velmi strohý webový klient. [24]

2.6 Wuala

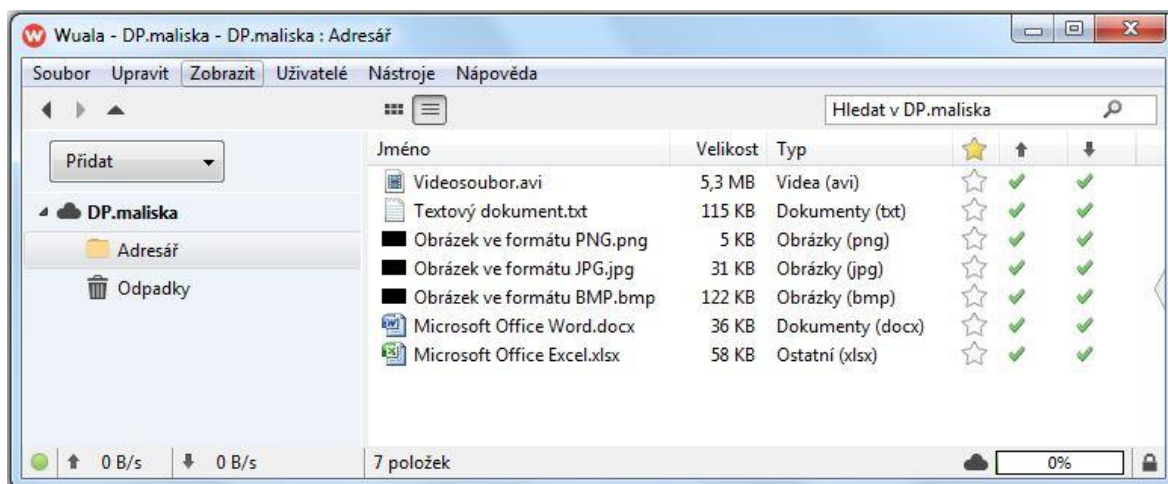
Datové úložiště od společnosti LaCie, které bylo vytvořeno v roce 2008, nabízí bezplatně kapacitu 5 GB s možností rozšíření až 2 TB s měsíčním paušálem. Hlavním odlišením od ostatních služeb je zabezpečení dat. Veškeré soubory jsou šifrovány na straně klienta pomocí symetrické blokové šifry AES s délkou klíče 256 bitů. Při přenosu sítí i při uložení v úložišti jsou data po celou dobu šifrována. Dešifrování probíhá opět u koncového uživatele, šifrovací heslo tedy neputuje sítí. I přes tento způsob zabezpečení je zachována funkce streamování videa či hudby – není tedy nutné čekat na kompletní stažení multimediálního souboru. [25]



Obrázek 18: Wuala logo [25]

Z důvodů bezpečnosti úložiště nenabízí webové rozhraní, proto se jej snaží nahradit aplikací bez nutnosti instalace. Aplikace je napsána v objektově orientovaném programovacím jazyku Java stejně jako klient podporující operační systémy Windows, OS X a Linux. Wuala nabízí kromě klasické synchronizace (včetně synchronizace různých složek, omezení rychlosti stahování a nahrávání dat) také možnost zálohování – data jsou

odesílána jednosměrně do cloudového úložiště. Z mobilních zařízení jsou podporovány systémy Android a iOS, z ostatních zařízení není žádná možnost se k úložišti připojit kvůli již zmiňované bezpečnosti. Pro tyto přenosná zařízení nabízí Wuala funkce jako správa souborů v cloudovém úložišti, automatické zálohování nebo ochrana PINem. [25] [29]



Obrázek 19: Java aplikace Wuala

2.7 Další úložiště

2.7.1 Box

- Kapacita zdarma: 10 GB
- Maximální kapacita: neomezeno
- Podporované operační systémy: Windows, OS X
- Mobilní platformy: Android, iOS, Windows Phone, BlackBerry
- Dostupná funkcionalita: synchronizační utility Box Sync, webové rozhraní, spolupráce s Google Docs, Microsoft Office a Microsoft Outlook, možnost zabezpečení s pomocí PINu
- Chybějící funkce: synchronizace více složek a přes místní lokální síť, omezení přenosové rychlosti, automatická synchronizace [26] [29]

2.7.2 Cubby

- Kapacita zdarma: 5 GB, referenční program 20 GB
- Maximální kapacita: 2,5 TB
- Podporované operační systémy: Windows, OS X
- Mobilní platformy: Android, iOS

- Dostupná funkcionalita: šifrování na straně klienta s využitím AES-256, DirectSync™ pro synchronizaci mezi zařízeními bez využití cloudové kapacity, synchronizace více složek
- Chybějící funkce: omezení přenosové rychlosti, automatická synchronizace [27] [29]

2.7.3 Capsa.cz

Pro porovnání se zahraničními službami je zde české cloudové úložiště Capsa.cz fungující od roku 2013.

- Kapacita zdarma: 5 GB, s registrací přes MojeID celkem 12 GB
- Maximální kapacita: 8 TB
- Podporované operační systémy: Windows
- Mobilní platformy: žádné
- Dostupná funkcionalita: přístup k datům pomocí FTP, pomocí klienta easyCapsa nebo přes webové rozhraní, synchronizace více složek, připojení úložiště jako disk, propojení se službou Úschovna.cz [28] [29]

3 QCUBED

QCubeD je open-source framework vytvořený v PHP5 sloužící pro rychlý vývoj aplikací. Je založen na softwarové architektuře MVC (Model-view-controller) a jeho cílem je šetřit čas vývojářů zjednodušením rutinních operací a odstraněním opakovaných úkolů. Uživatel může plně využít funkce nabízené frameworkem, upravovat je a také libovolně kombinovat s vlastním PHP kódem a SQL dotazy. Mezi podporované databáze patří MySQL, PostgreSQL, MS SQL, Oracle, Informix, SQLite. [31]

3.1 Základní části

Framework se skládá ze dvou základních částí: generátor kódu (The Code Generator) a knihovna uživatelského rozhraní (User Interface Library).

3.1.1 Generátor kódu

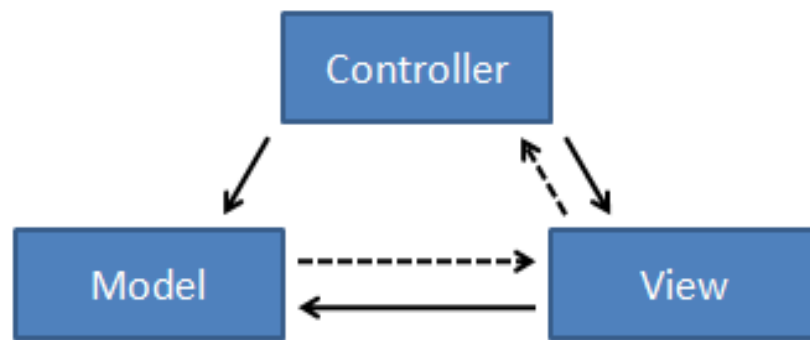
Generátor kódu vytváří PHP třídy na základě databáze. Využívá při tom ORM (objektově relační mapování), kdy každá tabulka odpovídá jedné generované třídě a její atributy odpovídají členským proměnným třídy. Tyto třídy zároveň obsahují základní CRUD metody (create, restore, update a delete) pro práci s databází a navíc také metody pro vytváření asociací a získávání dat pomocí indexů. Zároveň je vytvořen pro každou třídu uživatelský návrh (Drafts) s předem vytvořenými formuláři pro výpis, tvorbu, mazání a editaci záznamů v tabulce. Tyto formuláře si může uživatel dále upravovat nebo vytvořit své vlastní.

3.1.2 Knihovna uživatelského rozhraní

Knihovna uživatelského rozhraní označovaná jako QForm pomáhá vytvořit pomocí standardního HTML soubor s rozvržením stránky pojmenovaný jako view. Ovládací prvky a logika jsou pak umístěny v souboru označovaném jako controller. [31]

3.2 MVC (Model-view-controller)

Model-view-controller (česky Model-pohled-řadič) je softwarová architektura rozdělující datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent. Výhoda je patrná při změnách některé z komponent, která jen minimálně ovlivňuje zbývající dvě.



Obrázek 20: Diagram Model-view-controller [32]

- Model – obsahuje základní logickou strukturu dat v softwarové aplikaci.
- View – kolekce tříd reprezentující prvky v uživatelském rozhraní (textová pole, tlačítka, apod.).
- Controller – obsahuje třídy spojující model a pohled, mezi kterými umožňuje komunikaci. [32]

II. PRAKTICKÁ ČÁST

4 POROVNÁNÍ ÚLOŽIŠŤ

Pro většinu uživatelů je nejdůležitější vlastností internetového úložiště jeho kapacita. Pokud se jedná o poskytovanou kapacitu zdarma, kterou obdrží zákazník ihned po registraci, nejvíce nabízí GoogleDrive a Copy.com. V případě dokoupení datového prostoru uživatelé vyhledávají co nejpříznivější cenu, kterou nabízejí společnosti GoogleDrive a Microsoft OneDrive. Velký rozdíl je v maximálním možném limitu úložiště, který je nastaven v případě Microsoft OneDrive na 200 GB a GoogleDrive na 30 TB. Porovnání kapacit a cen všech zmíněných úložišť je uvedeno v následující tabulce.

Tabulka 1: Porovnání kapacit a cen datových úložišť

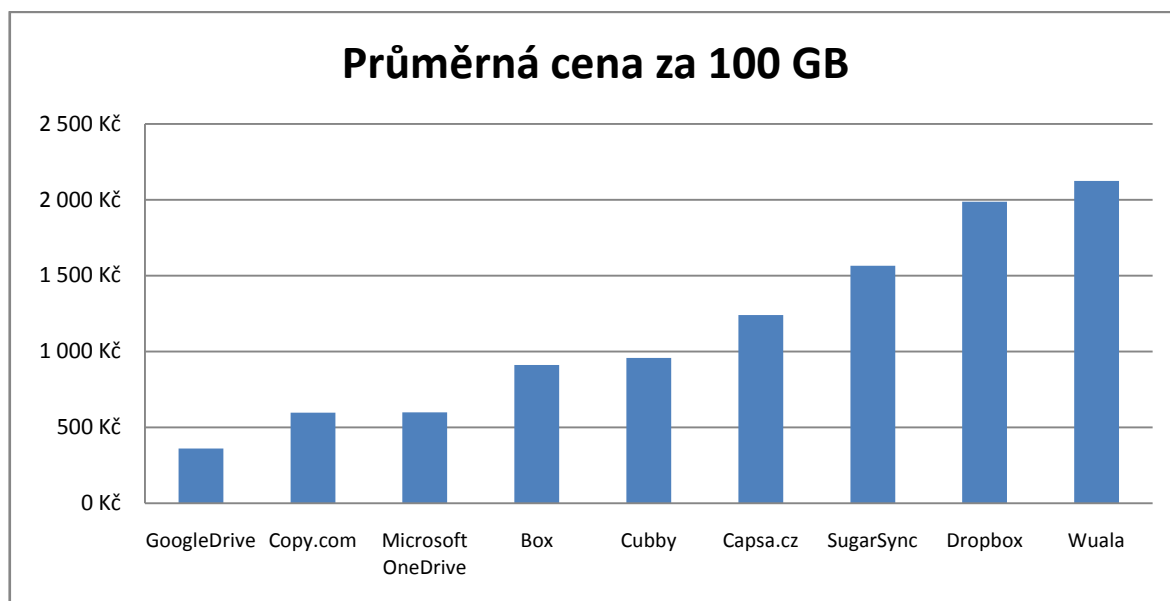
Název úložiště	Kapacita zdarma	Kapacita s bonusy zdarma	Maximální kapacita	Roční cena za 100 GB	Roční cena za 500 GB	Roční cena za 1 TB
GoogleDrive	15 GB	15 GB	30 TB	480 Kč	-	2 400 Kč
Microsoft OneDrive	7 GB	15 GB	200 GB	600 Kč	-	-
Dropbox	2 GB	18 GB	500 GB	1 980 Kč	9 980 Kč	-
SugarSync	neposkytuje	neposkytuje ¹	500 GB	2 000 Kč	7 980 Kč	11 000 Kč
Copy.com	15 GB	neomezeno ²	500 GB	-	2 980 Kč	-
Wuala	5 GB	5 GB	2 TB	2 180 Kč	10 980 Kč	19 980 Kč
Box	10 GB	10 GB	neomezeno ³	960 Kč	-	8 640 Kč
Cubby	5 GB	25 GB	2,5 TB	960 Kč	4 790 Kč	9 580 Kč
Capsa.cz	5 GB	12 GB	8 TB	1 200 Kč	6 600 Kč	12 000 Kč

¹SugarSync poskytuje pouze 30-ti denní zkušební verzi zdarma.

²Copy.com dočasně neomezuje referenční program.

³Box nabízí individuální kalkulaci ceny a maximální kapacity.

Přehlednější porovnání cen je zobrazeno v následujícím grafu. Jedná se o průměrnou cenu v korunách za 100 GB kapacity na jeden rok. Společnosti jsou seřazeny od nejlevnější po nejdražší.



Graf 1: Průměrná cena za 100 GB

Dalším kritériem pro výběr cloudového úložiště je podpora platform pro synchronizační utilitu. Zatím co operační systém Windows a OS X je většinou podporován, Linux je spíše výjimkou. Z mobilních platform jsou nejčastější Android a iOS. Následující tabulka zobrazuje nejnižší podporovanou verzi daného systému.

Tabulka 2: Porovnání podporovaných platform

Název úložiště	Windows	OS X	Linux	Android	iOS	Windows Phone	BlackBerry
GoogleDrive	Windows XP	10.6	-	2.1	5.0	-	-
Microsoft OneDrive	Windows 7	10.7	-	2.3	5.0	7.5	-
Dropbox	Windows XP	10.4	Ubuntu 7.10	2.1	5.0	-	4.5
SugarSync	Windows XP	10.6	-	2.2	5.0	5	4.5
Copy.com	Windows XP	10.7	-	2.1	5.0	8	-
Wuala	Windows XP	10.6	Debian 5	2.2	4.3	-	-
Box	Windows XP	10.7	-	2.2	5.0	7.5	-
Cubby	Windows XP	10.7	-	2.3	5.0	-	-
Capsa.cz	Windows XP	-	-	-	-	-	-

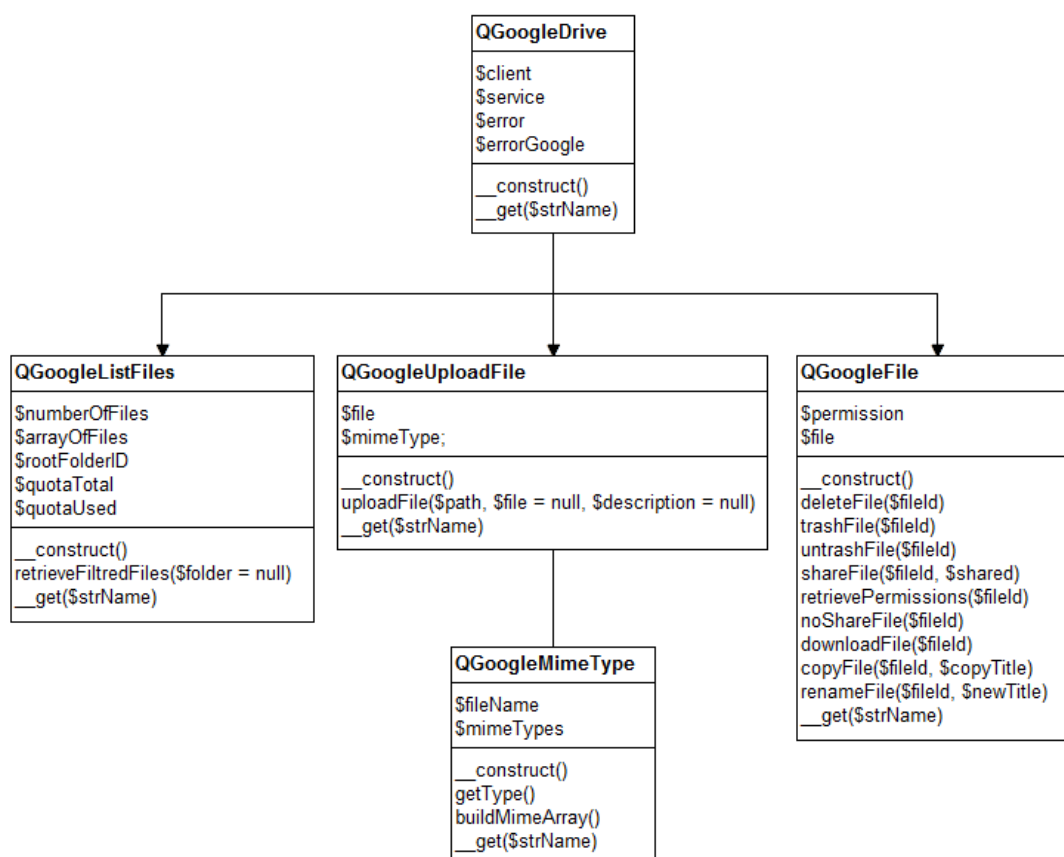
Porovnáním jednotlivých služeb bylo vybráno jako nejvhodnější úložiště pro další aplikaci Google Drive. Hlavními důvody jsou:

- Nadprůměrná kapacita zdarma.
- Nejnižší roční paušál s možností rozšíření kapacity až na 30 TB.
- Propojení s Gmail a Google+.
- Implementace kancelářských webových aplikací Google Docs a služby Google Cloud Print.
- Veřejné Google Drive API s kvalitní dokumentací.
- Dobrá podpora vývojářů s vysokým množstvím rad a návodů.
- Velký potenciál různých produktů společnosti Google.

5 IMPLEMENTACE GOOGLE DRIVE DO FRAMEWORKU QCUBED

Pro implementaci Google Drive do frameworku QCubed je využito veřejné PHP API od společnosti Google, které vyžaduje PHP 5.2.1 nebo vyšší. Dále je pro jeho funkčnost vyžadováno PHP JSON extension a PHP CURL extension. Toto API je volně dostupné na stránkách společnosti Google v sekci pro vývojáře. Použitá verze frameworku QCubed je 2.2.2, kterou lze stáhnout na stránkách QCubed.

Následující diagram zobrazuje nově vytvořené třídy, jejich atributy a metody.



Obrázek 21: Diagram vytvořených tříd

5.1 Přihlášení ke službě

Ideálním způsobem přihlášení k disku by bylo vytvoření třídy, která by tuto činnost zajišťovala. Společnost Google však neustále mění podporovaný způsob autorizace a autentizace k jejím produktům, proto je velmi obtížné takovou třídu vytvořit. Pro API verze 1 bylo podporováno přihlášení s využitím „ClientLogin“, které bylo později nahrazeno „AuthSub“. S API verzí 2 přešel Google na přihlášení pomocí „O Auth 1.0“,

které v průběhu aktualizoval na „O Auth 2.0“. Tento typ je funkční i v současnosti, avšak společnost již oznámila, že jej v brzké době přestane podporovat. V současné době se tedy jeví jako nejperspektivnější způsob autorizace a autentizace tlačítko „Google+ Sign-in“, proto je také dále použito.

5.2 Komunikace s Google Drive

Základní objekty sloužící ke komunikaci s Google Drive vytváří třída QGoogleDrive.

Ukázka zdrojového kódu souboru QGoogleDrive.class.php:

```
1 <?php
8 require_once('google-api-php-
    client/src/Google_Client.php');
9 require_once('google-api-php-
    client/src/contrib/Google_DriveService.php');
19 class QGoogleDrive {
25     protected $client;
28     protected $service;
31     protected $error;
34     protected $errorGoogle;
44     public function __construct() {
45         try {
46             $this->client = new Google_Client();
47             $this->client->setUseObjects(true);
48             $this->client-
>setAccessToken($_SESSION['accessToken']);
49             $this->service = new
Google_DriveService($this->client);
50         } catch (Google_Exception $eG) {
51             $this->errorGoogle = $eG->getMessage();
52         } catch (Exception $e) {
53             $this->error = $e->getMessage();
54         }
55     }
67     public function __get($strName) {
```

```
68         switch ($strName) {
69             case 'Client': return $this->client;
70             case 'Service': return $this->service;
71             case 'Error': return $this->error;
72             case 'ErrorGoogle': return $this->errorGoogle;
74             default:
75                 return null;
76         }
77     }
78 }
79 ?>
```

Nejprve je nutné načíst soubory `Google_Client.php` a `Google_DriveService.php` (řádky 8 a 9), které jsou použity při vytváření nových instancí objektů. V konstruktoru se vytváří instance třídy `Google_Client` (řádek 46), které je poté nastaven přístupový token. Dále je vytvořena instance třídy `Google_DriveService`, s parametrem `Google_Client` (řádek 49). Tělo konstruktoru je ošetřeno blokem pro výjimky typu `Google_Exception`, které mohou nastat například v těchto případech: token nelze dekodovat, špatný formát tokenu, časový limit tokenu vypršel, chybějící časový limit tokenu apod.

5.3 Nahrání souborů

K nahrávání souborů na Google Drive slouží třída `QGoogleUploadFile`, která dědí metody a atributy z rodičovské třídy `QGoogleDrive`. Ukázka zdrojového kódu souboru `QGoogleUploadFile.class.php`:

```
1 <?php
16 class QGoogleUploadFile extends QGoogleDrive {
22     protected $file;
25     protected $mimeType;
35     public function __construct() {
36         parent::__construct();
37         try {
38             $this->file = new Google_DriveFile();
39         } catch (Google_Exception $eG) {
40             $this->errorGoogle = $eG->getMessage();
```



```
41         } catch (Exception $e) {
42             $this->error = $e->getMessage();
43         }
44     }
45
46     public function uploadFile($path, $file = null,
47         $description = null) {
48         $arrayFileName = array();
49         try {
50             if ($file == null) {
51                 $dir = dir($path);
52                 while ($this->file = $dir->read()) {
53                     if ($this->file != '.' && $this->file
54                         != '..') {
55                         $arrayFileName[] = $this->file;
56                     }
57                 }
58                 $dir->close();
59             }
60             else {
61                 $arrayFileName[] = $file;
62             }
63             foreach ($arrayFileName as $fileName) {
64                 if(!$this->file) $this->file = new
65                 Google_DriveFile();
66                 $filePath = $path.'/'.$fileName;
67                 $fileMimeType = new
68                 QGoogleMimeType($fileName);
69                 $this->mimeType = $fileMimeType-
70                 >GetType();
71                 $this->file->setTitle($fileName);
72                 $this->file-
73                 >setDescription($description);
74                 $this->file->setMimeType($this-
75                 >mimeType);
```

```
84             $this->service->files->insert(
85                 $this->file,
86                 array(
87                     'data' =>
88                     file_get_contents($filePath),
89                     'mimeType' => $this->mimeType
90                 )
91             );
92         } catch (Google_Exception $eG) {
93             $this->errorGoogle = $eG->getMessage();
94         } catch (Exception $e) {
95             $this->error = $e->getMessage();
96         }
97     }
110     public function __get($strName) {
111         switch ($strName) {
112             case 'File': return $this->file;
113             case 'MimeType': return $this->mimeType;
114             default:
115                 try {
116                     return parent::__get($strName);
117                 } catch (QCallerException $objExc) {
118                     $objExc->IncrementOffset();
119                     throw $objExc;
120                 }
121             }
122         }
123     }
124 }
125 ?>
```

V konstruktoru se vytváří instance třídy `Google_DriveFile` (řádek 38) která reprezentuje soubor a jeho atributy, tak jak jsou uloženy na Google Drive. Metoda `uploadFile` (řádek 56), která provede nahrání požadovaných souborů má tři parametry:

- `path` – cesta ke složce, kde se nachází soubory

- file – název souboru včetně přípony (nepovinný)
- description – popis souboru (nepovinný)

Pokud není parametr file definován, je vytvořeno pole souborů z celé složky (řádky 60 – 68). V opačném případě je pole tvořeno pouze jedním souborem (řádek 71). Při nahrávání souborů je nutné specifikovat typ internetového média, který je zjištěn z přípony souboru za pomoci třídy QGoogleMimeType (řádky 78 a 80). Dále je nastaveno jméno souboru a volitelně popis. Pokud je nahráváno více souborů, je nutné neustále vytvářet nové instance třídy Google_DriveFile (řádek 75), protože při dokončení metody insert (řádek 84) stávající instance zaniká.

5.4 Obsah disku

Třída QGoogleListFiles dědí metody a atributy z rodičovské třídy QGoogleDrive a slouží k získání seznamu souborů nacházejících se na Google Drive. Ukázka zdrojového kódu souboru QGoogleListFiles.class.php:

```
1 <?php
22 class QGoogleListFiles extends QGoogleDrive {
28     protected $numberOfFiles;
31     protected $arrayOfFiles;
34     protected $rootFolderID;
37     protected $quotaTotal;
40     protected $quotaUsed;
50     public function __construct() {
51         parent::__construct();
53         $this->arrayOfFiles = array();
54         $pageToken = NULL;
55         do {
56             $parameters = array();
57             try {
58                 if ($pageToken) {
59                     $parameters['pageToken'] =
$pageToken;
60             }
```

```

61             $files = $this->service->files->
>listFiles($parameters);
62             $this->arrayOfFiles = array_merge($this->
>arrayOfFiles, $files->getItems());
63             $pageToken = $files->getNextPageToken();
64             } catch (Google_Exception $eG) {
65                 $this->errorGoogle = $eG->getMessage();
66             } catch (Exception $e) {
67                 $this->error = $e->getMessage();
68             }
69         } while ($pageToken);
71     try {
72         $aboutService = $this->service->about->get();
73         $this->rootFolderID = $aboutService->
>getRootFolderId();
74         $this->quotaTotal = $aboutService->
>getQuotaBytesTotal();
75         $this->quotaUsed = $aboutService->
>getQuotaBytesUsed();
76     } catch (Google_Exception $eG) {
77         $this->errorGoogle = $eG->getMessage();
78     } catch (Exception $e) {
79         $this->error = $e->getMessage();
80     }
81 }
90 public function retrieveFiltredFiles($folder = null)
{
91     switch ($folder) {
92         case 'trash':
93             $trash = "true";
94             $shared = "";
95             $folderId = "";
96             break;
97         case 'main':

```

```
98             $trash = "";
99             $shared = "";
100            $folderId = $this->rootFolderID;
101            break;
102        case 'shared':
103            $trash = "";
104            $shared = "true";
105            $folderId = "";
106            break;
108        default:
109            $trash = "";
110            $shared = "";
111            $folderId = "";
112            break;
113    }
114    $arrayFull = array();
115    foreach ($this->arrayOfFiles as $file) {
116        $arrayColumn = array();
117        try {
118            $arrayRow = json_encode($file);
119            $arrayRow = json_decode($arrayRow, true);
120            $tempShared =
121                json_encode($arrayRow["shared"]);
122            $tempLabels =
123                json_encode($arrayRow["labels"]);
124            $tempLabels = json_decode($tempLabels,
125                true);
126            $tempTrashed =
127                json_encode($tempLabels["trashed"],
128                JSON_UNESCAPED_UNICODE);
129            $tempMimeType =
130                json_encode($arrayRow["mimeType"],
131                JSON_UNESCAPED_UNICODE);
```

```
125             $tempMimeType =
                stripslashes(str_replace('"', "'", $tempMimeType));
126             $tempParents =
                json_encode($arrayRow["parents"]);
127             $tempParents = json_decode($tempParents,
                true);
128             $tempParents =
                json_encode($tempParents[0]);
129             $tempParents = json_decode($tempParents,
                true);
130             $tempParentID =
                json_encode($tempParents["id"], JSON_UNESCAPED_UNICODE);
131             $tempParentID =
                stripslashes(str_replace('"', "'", $tempParentID));
132             if((!$folder || $shared == $tempShared ||
                $trash == $tempTrashed || ($tempTrashed == "false" &&
                $folderId == $tempParentID)) && $tempMimeType !=
                "application/vnd.google-apps.folder") {
133                 $tempTitle =
                json_encode($arrayRow["title"], JSON_UNESCAPED_UNICODE);
134                 $tempTitle =
                stripslashes(str_replace('"', "'", $tempTitle));
135                 if($tempTitle != "null") {
136                     array_push($arrayColumn,
                $tempTitle);
138                     $tempCreatedDate =
                json_encode($arrayRow["createdDate"],
                JSON_UNESCAPED_UNICODE);
139                     $tempCreatedDate =
                stripslashes(str_replace('"', "'", $tempCreatedDate));
140                     $createdDate = new
                QDateTime($tempCreatedDate);
141                     $createdDate->AddHours(2);
```

```
142             array_push($arrayColumnn,
    $createdDate->qFormat('hhh:mm D.M.YYYY'));
144             $tempModifiedDate =
    json_encode($arrayRow["modifiedDate"],
    JSON_UNESCAPED_UNICODE);
145             $tempModifiedDate =
    stripslashes(str_replace('"', "'", $tempModifiedDate));
146             $modifiedDate = new
    QDateTime($tempModifiedDate);
147             $modifiedDate->AddHours(2);
148             array_push($arrayColumnn,
    $modifiedDate->qFormat('hhh:mm D.M.YYYY'));
150             $tempDescription =
    json_encode($arrayRow["description"],
    JSON_UNESCAPED_UNICODE);
151             $tempDescription =
    stripslashes(str_replace('"', "'", $tempDescription));
152             $tempDescription =
    str_replace('null', '-', $tempDescription);
153             array_push($arrayColumnn,
    $tempDescription);
155             $tempId =
    json_encode($arrayRow["id"], JSON_UNESCAPED_UNICODE);
156             $tempId =
    stripslashes(str_replace('"', "'", $tempId));
157             array_push($arrayColumnn,
    $tempId);
159             $tempShared =
    json_encode($arrayRow["shared"], JSON_UNESCAPED_UNICODE);
160             $tempShared =
    stripslashes(str_replace('"', "'", $tempShared));
161             $tempShared = str_replace('true',
    QApplication::Translate("Yes"), $tempShared);
```

```
162             $tempShared =
                str_replace('false', QApplication::Translate("No"),
                $tempShared);
163             array_push($arrayColumn,
                $tempShared);
165             $tempIconLink =
                json_encode($arrayRow["iconLink"],
                JSON_UNESCAPED_UNICODE);
166             $tempIconLink =
                stripslashes(str_replace('"', "'", $tempIconLink));
167             array_push($arrayColumn,
                $tempIconLink);
169             $tempFileExtension =
                json_encode($arrayRow["fileExtension"],
                JSON_UNESCAPED_UNICODE);
170             $tempFileExtension =
                stripslashes(str_replace('"', "'", $tempFileExtension));
171             array_push($arrayColumn,
                $tempFileExtension);
173             $tempFileSize =
                json_encode($arrayRow["fileSize"],
                JSON_UNESCAPED_UNICODE);
174             $tempFileSize =
                stripslashes(str_replace('"', "'", $tempFileSize));
175             array_push($arrayColumn,
                $tempFileSize);
177             array_push($arrayFull,
                $arrayColumn);
178         }
179     }
180     } catch (Google_Exception $eG) {
181         $this->errorGoogle = $eG->getMessage();
182     } catch (Exception $e) {
183         $this->error = $e->getMessage();
```



```
184         }
185     }
186     $this->numberOfFiles = count($arrayFull);
187     return $arrayFull;
188 }
200 public function __get($strName) {
201     switch ($strName) {
202         case 'NumberOfFiles': return $this-
>numberOfFiles;
203         case 'ArrayOfFiles': return $this-
>arrayOfFiles;
204         case 'RootFolderID': return $this-
>rootFolderID;
205         case 'QuotaTotal': return $this->quotaTotal;
206         case 'QuotaUsed': return $this->quotaUsed;
208         default:
209             try {
210                 return parent::__get($strName);
211             } catch (QCallerException $objExc) {
212                 $objExc->IncrementOffset();
213                 throw $objExc;
214             }
215     }
216 }
217 }
218 ?>
```

Konstruktor této třídy (řádky 50 - 69) uloží do proměnné `arrayOfFiles` seznam souborů (včetně veškerých detailů) nacházejících se na Google Drive. Dále je do proměnných `quotaTotal` a `quotaUsed` uložen řetězec nesoucí informaci o využívané a celkové kapacitě disku v bajtech (řádky 71 – 80). V proměnné `rootFolderID` se nachází ID hlavní složky.

Data jsou po stažení uložena v poli `arrayOfFiles` ve formátu `Google_DriveFile`. Na pole řetězců je převádí metoda `retrieveFiltredFiles` (řádky 90 - 188) s využitím JSON funkcí.

Parametr této metody umožňuje filtrovat soubory dle umístění (např. hlavní složka, sdílené, koš apod.). Výsledné sloupce pole obsahují:

- jméno souboru
- datum vytvoření
- datum poslední úpravy
- popis
- ID souboru
- informaci o sdílení
- odkaz na miniaturní ikonu
- příponu souboru
- velikost souboru v bajtech

5.5 Práce se soubory

K práci se soubory nacházející se na Google Drive slouží třída QGoogleFile, která dědí metody a atributy z rodičovské třídy QGoogleDrive. Ukázka zdrojového kódu souboru QGoogleFile.class.php:

```
1 <?php
16 class QGoogleFile extends QGoogleDrive {
22     protected $permission;
25     protected $file;
35     public function __construct() {
36         parent::__construct();
37         try {
38             $this->permission = new Google_Permission();
39             $this->file = new Google_DriveFile();
40         } catch (Google_Exception $eG) {
41             $this->errorGoogle = $eG->getMessage();
42         } catch (Exception $e) {
43             $this->error = $e->getMessage();
44         }
45     }
54     public function deleteFile($fileId) {
55         try {
```

```
56         $this->service->files->delete($fileId);
57     } catch (Google_Exception $eG) {
58         $this->errorGoogle = $eG->getMessage();
59     } catch (Exception $e) {
60         $this->error = $e->getMessage();;
61     }
62 }
71 public function trashFile($fileId) {
72     try {
73         $this->service->files->trash($fileId);
74     } catch (Google_Exception $eG) {
75         $this->errorGoogle = $eG->getMessage();
76     } catch (Exception $e) {
77         $this->error = $e->getMessage();
78     }
79 }
88 public function untrashFile($fileId) {
89     try {
90         $this->service->files->untrash($fileId);
91     } catch (Google_Exception $eG) {
92         $this->errorGoogle = $eG->getMessage();
93     } catch (Exception $e) {
94         $this->error = $e->getMessage();
95     }
96 }
106 public function shareFile($fileId, $shared) {
107     try {
108         if ($shared == 'public') {
109             $this->permission->setValue('user');
110             $this->permission->setType('anyone');
111             $this->permission->setRole('reader');
112         } elseif ($shared == 'anyone') {
113             $this->permission->setValue('user');
114             $this->permission->setType('anyone');
```

```
115             $this->permission->setRole('reader');
116             $this->permission->setWithLink(true);
117         } else {
118             $this->permission->setValue($shared);
119             $this->permission->setType('user');
120             $this->permission->setRole('reader');
121         }
122         $this->service->permissions->insert($fileId,
    $this->permission);
123     } catch (Google_Exception $eG) {
124         $this->errorGoogle = $eG->getMessage();
125     } catch (Exception $e) {
126         $this->error = $e->getMessage();
127     }
128 }
137 public function retrievePermissions($fileId) {
138     $result = null;
139     try {
140         $permissions = $this->service->permissions-
    >listPermissions($fileId);
141         $result = $permissions->getItems();
142     } catch (Google_Exception $eG) {
143         $this->errorGoogle = $eG->getMessage();
144     } catch (Exception $e) {
145         $this->error = $e->getMessage();
146     }
147     return $result;
148 }
157 public function noShareFile($fileId) {
158     $arrayPermissionsId = array();
159     try {
160         $permissionsList = $this-
    >retrievePermissions($fileId);
162         array_shift($permissionsList);
```

```
163         foreach ($permissionsList as $permission) {
164             $arrayRow = json_encode($permission);
165             $arrayRow = json_decode($arrayRow, true);
166             $temp = json_encode($arrayRow["id"]);
167             $temp = stripslashes(str_replace('"', '',
    $temp));
168             array_push($arrayPermissionsId, $temp);
169         }
170         foreach ($arrayPermissionsId as
    $permissionId) {
171             $this->service->permissions-
    >delete($fileId, $permissionId);
172         }
173     } catch (Google_Exception $eG) {
174         $this->errorGoogle = $eG->getMessage();
175     } catch (Exception $e) {
176         $this->error = $e->getMessage();
177     }
178 }
187 public function downloadFile($fileId) {
188     $result = null;
189     try {
190         $this->file = $this->service->files-
    >get($fileId);
191         $result = $this->file->getDownloadUrl();
192     } catch (Google_Exception $eG) {
193         $this->errorGoogle = $eG->getMessage();
194     } catch (Exception $e) {
195         $this->error = $e->getMessage();
196     }
197     return $result;
198 }
208 public function copyFile($fileId, $copyTitle) {
209     try {
```

```
210         $this->file->setTitle($copyTitle);
211         $this->service->files->copy($fileId, $this-
>file);
212     } catch (Google_Exception $eG) {
213         $this->errorGoogle = $eG->getMessage();
214     } catch (Exception $e) {
215         $this->error = $e->getMessage();
216     }
217 }
227 public function renameFile($fileId, $newTitle) {
228     try {
229         $this->file->setTitle($newTitle);
230         $this->service->files->patch($fileId, $this-
>file, array(
231             'fields' => 'title'
232         ));
233     } catch (Google_Exception $eG) {
234         $this->errorGoogle = $eG->getMessage();
235     } catch (Exception $e) {
236         $this->error = $e->getMessage();
237     }
238 }
250 public function __get($strName) {
251     switch ($strName) {
252         case 'Permission': return $this->permission;
253         case 'File': return $this->file;
255         default:
256             try {
257                 return parent::__get($strName);
258             } catch (QCallerException $objExc) {
259                 $objExc->IncrementOffset();
260                 throw $objExc;
261             }
262     }
```

```
263     }
```

```
264 }
```

```
265 ?>
```

V konstruktoru (řádky 35 – 45) se vytváří instance třídy `Google_Permission`, která slouží k přidávání přístupových práv souborů a instance třídy `Google_DriveFile` sloužící k práci se soubory. Dále tato třída obsahuje metody:

- `deleteFile` (řádek 54) – smazání souboru (soubor je odstraněn a nelze jej obnovit), argumentem je ID souboru
- `trashFile` (řádek 71) – odstranění souboru (soubor je přesunut do koše), argumentem je ID souboru
- `untrashFile` (řádek 88) – obnovení souboru (soubor je přesunut z koše do původní složky), argumentem je ID souboru
- `shareFile` (řádek 106) – sdílení souboru s ostatními uživateli (přidání nového přístupového práva), argumenty jsou ID souboru a řetězec určující typ sdílení, který může mít hodnotu:
 - `public` – kdokoli na Internetu může vyhledat soubor a přistupovat k němu
 - `anyone` – kdokoli s odkazem může přistupovat k souboru
 - konkrétní email uživatele, který může přistupovat k souboru
- `retrievePermissions` (řádek 137) – seznam přístupových práv konkrétních uživatelů, argumentem je ID souboru
- `noShareFile` (řádek 157) – odstranění sdílení souboru s ostatními uživateli (smazání všech přístupových práv ostatních uživatelů), argumentem je ID souboru
- `downloadFile` (řádek 187) – URL odkaz ke stažení souboru, argumentem je ID souboru
- `copyFile` (řádek 208) – kopie existujícího souboru, argumenty jsou ID souboru a název nového souboru
- `renameFile` (řádek 227) – přejmenování souboru, argumenty jsou ID souboru a nový název souboru

5.6 Typ internetového média

Při uložení souborů na disk je důležitým parametrem typ internetového média (MIME – Multipurpose Internet Mail Extensions, česky víceúčelová rozšíření internetové pošty).

Jedná se o řetězec složený ze dvou částí: typu a podtypu (např. obrázku JPEG odpovídá řetězec: image/jpeg).

K získání správného typu slouží třída QGoogleMimeType, která se nachází v souboru QGoogleMimeType.class.php. Její metoda getType rozezná typ dle přípony souboru. Pokud je typ souboru podporován Google Drive, metoda vrací odpovídající řetězec, v opačném případě je vrácen řetězec application/octet-stream, který reprezentuje obecná binární data.

6 UKÁZKOVÁ WEBOVÁ APLIKACE

6.1 Návrh aplikace

Základním požadavkem je vytvoření ukázkové webové aplikace, která ověří funkčnost nově vytvořených tříd.

6.1.1 Funkční požadavky

- Přihlášení uživatele pomocí jeho uživatelského jména a hesla.
- Přehledné zobrazení seznamu souborů nacházejících se na cloudovém disku.
- Přepínání mezi složkami.
- Nahrání nového souboru na cloudový disk.
- Sdílení souboru s ostatními uživateli.
- Stažení souboru.
- Odstranění a obnovení souboru.
- Odhlášení uživatele.

6.1.2 Nefunkční požadavky

- Při tvorbě aplikace je využit PHP framework QCubed verze 2.2.2.
- Pro práci s Google Drive jsou využity nově vytvořené třídy.
- Aplikace musí být jednoduchá, intuitivní a snadno ovladatelná.
- Optimalizace pro prohlížeče: Google Chrome, Internet Explorer, Mozilla Firefox a Opera.

6.2 Realizace aplikace

Přihlášení uživatele je realizováno v souboru login.php. Po úspěšném přihlášení je uživatel přesměrován na domovskou stránku, která je definována v souboru home.php. V tomto souboru se nachází třída HomeForm, která je jádrem celé aplikace. Zde je uvedeno vysvětlení proměnných této třídy:

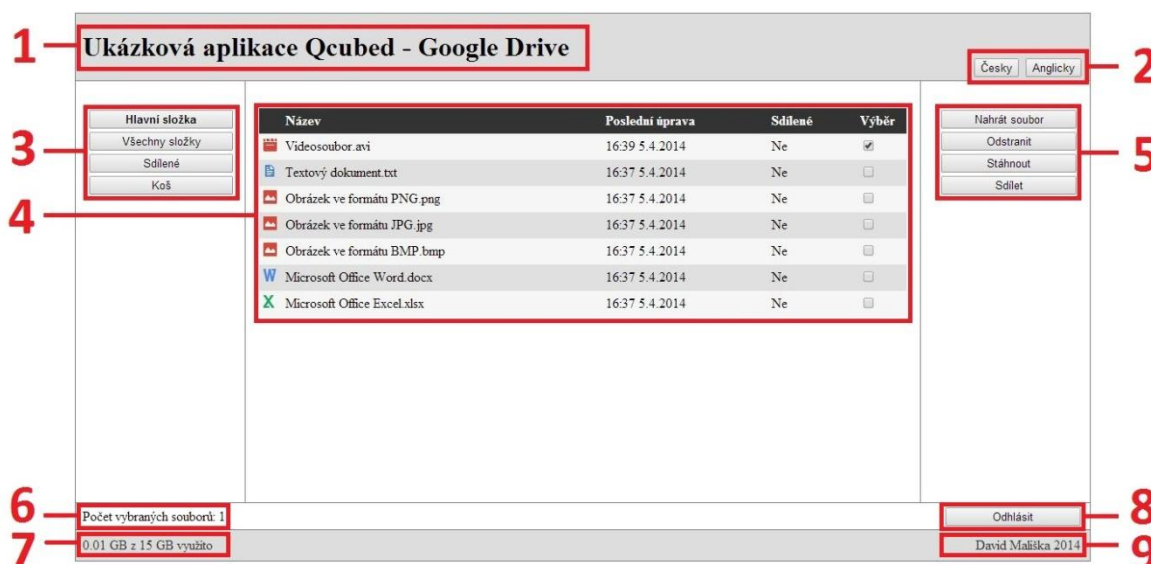
- btnMainFolder, btnAllFolders, btnSharedFolder, btnTrash – instance třídy QPushButton slouží k procházení složek, po kliku volají metodu btnFolder_Click()
- idCurrentFolder – proměnná ukládající id aktuální otevřené složky
- flUploadFile – instance nově vytvořené třídy QFileUpload slouží k vytvoření dialogu, ve kterém je možno zvolit soubor k nahrání na disk

- `btnDeleteFile`, `btnTrashFile`, `btnUntrashFile`, `btnDownloadFile`, `btnShareFile`, `btnNoShareFile` – instance třídy `QPushButton` slouží k práci se soubory (trvalé smazání souboru, přesunutí souboru do koše, obnovení souboru z koše, stažení souboru, sdílení a nesdílení souboru)
- `btnLogout` – instance třídy `QPushButton` slouží k odhlášení uživatele, po kliku volá metodu `btnLogout_Click()`
- `lblStatus`, `lblQuota` – instance třídy `QLabel` slouží k informování uživatele o aktuálním stavu aplikace a využití kapacity disku
- `selectedIdArray` – pole obsahující ID vybraných souborů
- `listFiles` – instance nově vytvořené třídy `QGoogleListFiles` slouží k práci se seznamem souborů na disku
- `tblListFiles` – instance třídy `QDataGrid` poskytuje přehledné zobrazení seznamu souborů, včetně mini ikony a výběru souboru
- `file` – instance nově vytvořené třídy `QGoogleFile` slouží pro práci se souborem
- `btnCz`, `btnEn` – instance třídy `QPushButton` umožňuje uživateli výběr jazyka, po kliku volá metodu `btnLanguage_Click`
- `dlgShareFile` – instance třídy `QDialog` slouží k nastavení sdílení, po zavření volá metodu `dlgShareFile_Click()`
- `dlgLinkToShare` – instance třídy `QDialog` slouží k zobrazení odkazu ke sdílení v dialogovém okně
- `dlgInsertEmail` – instance nově vytvořené třídy `QInsertEmailDialog` slouží k zadání e-mailu uživatele, kterému chceme přidat přístupová práva k souboru
- `objDefaultWaitIcon` – instance třídy `QWaitIcon` slouží pro přepsání původního ukazatele procesu načítání

Metody třídy `HomeForm` jsou popsány komentáři ve zdrojovém kódu. Pro plnou funkčnost aplikace byly vytvořeny dvě nové třídy `QFileUpload` a `QInsertEmailDialog`. `QFileUpload` slouží k nahrávání souborů a je alternativou ke stávající třídě `QFileAsset`, která je určena především pro nahrávání obrázků. `QInsertEmailDialog` je potomkem `QDialog` a vytváří dialogové okno pro zadání e-mailu uživatele.

6.3 Popis a funkcionalita

Pro ukázkovou aplikaci byl zvolen jednoduchý design neobsahující rušivé prvky, aby byla zajištěna co nejvyšší přehlednost. Layout popisuje následující obrázek:

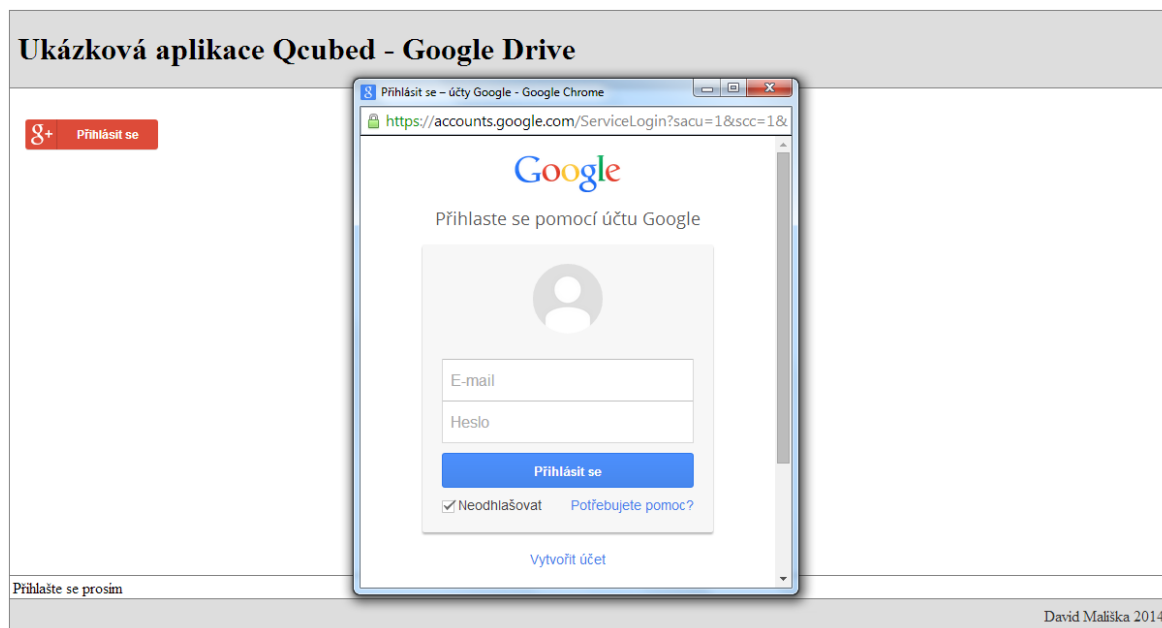


Obrázek 22: Ukázková aplikace (layout)

- 1 – hlavička
- 2 – přepínání jazykových verzí (Čeština a Angličtina)
- 3 – navigační menu pro procházení jednotlivých složek
- 4 – výpis souborů obsahující mini ikonu, název souboru, datum poslední úpravy, informaci o sdílení a zatrhávací políčko pro výběr
- 5 – nabídka s přípustnými operacemi
- 6 – stavový řádek
- 7 – využitá kapacita
- 8 – tlačítko pro odhlášení
- 9 – autor

6.3.1 Přihlášení

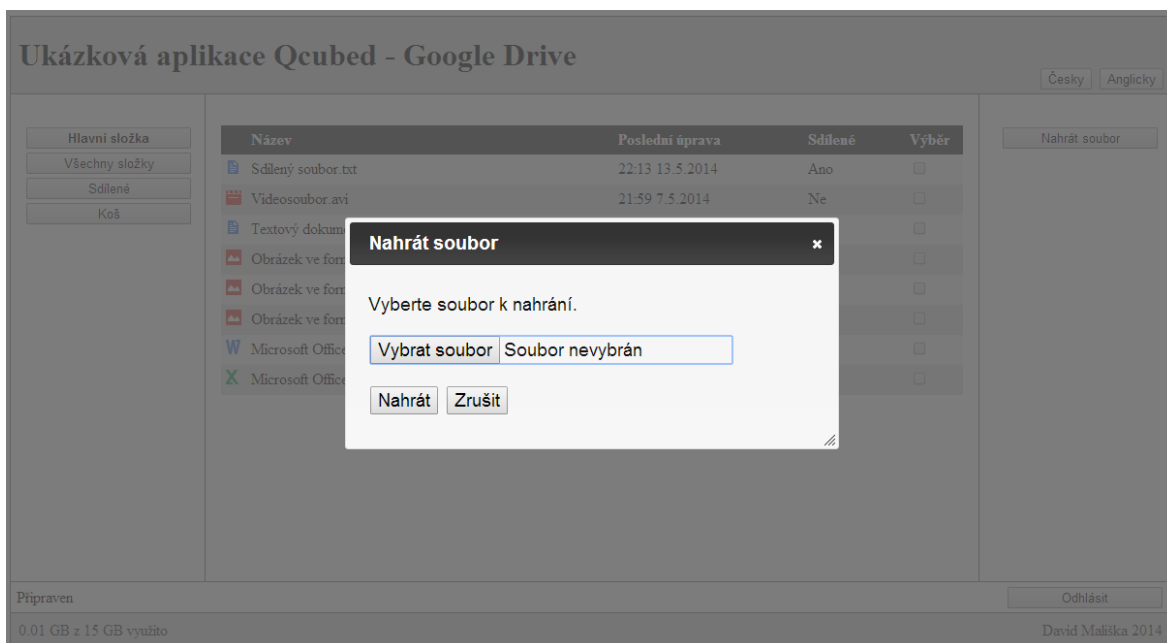
Po spuštění aplikace se zobrazí přihlašovací stránka s tlačítkem Google+, které při kliknutí otevírá přihlašovací nabídku k účtům Google. Uživatel se zde může přihlásit vyplněním uživatelského jména (e-mailu) a hesla, nebo si může založit nový účet. Po úspěšném přihlášení je zobrazen obsah jeho disku.



Obrázek 23: Ukázková aplikace (přihlášení)

6.3.2 Nahrání souboru

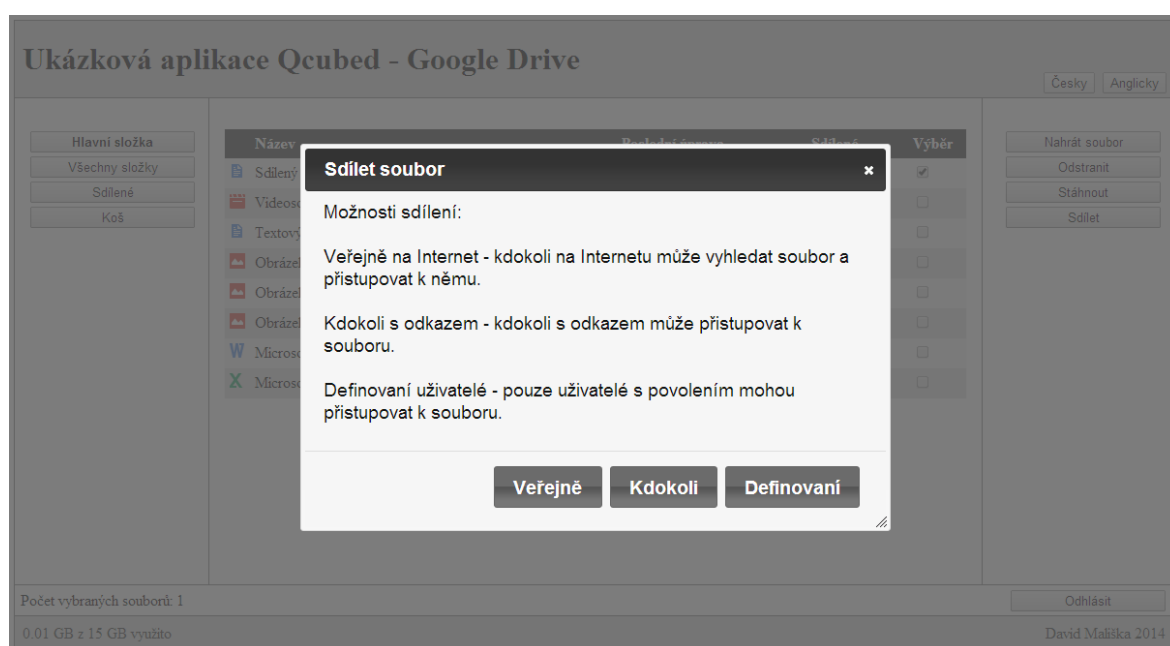
Po stisku tlačítka nahrát soubor, se otevře nabídka pro výběr souboru. Vybrán může být jakýkoli typ souboru – pokud není typ souboru podporován Google Drive, je tento soubor uložen jako obecná binární data.



Obrázek 24: Ukázková aplikace (nahrání souboru)

6.3.3 Sdílení souboru

Ukázková aplikace nabízí různé možnosti sdílení souborů. Po kliknutí na volbu sdílet, je zobrazena nabídka sdílení. Pokud si uživatel vybere některou z prvních dvou možností („Veřejně na Internet“ nebo „Kdokoli s odkazem“), je vygenerován a zobrazen odkaz, pomocí kterého mohou ostatní uživatelé přistupovat k souboru. Při výběru poslední možnosti („Definování uživatelé“) je nutno dále zadat e-mail uživatele, kterému chceme povolit přístup. Tomuto uživateli je automaticky zaslán e-mail s informacemi o sdílení (jméno sdílejícího a název souboru). Se souborem poté může pracovat na svém Google disku.



Obrázek 25: Ukázková aplikace (sdílení souboru)

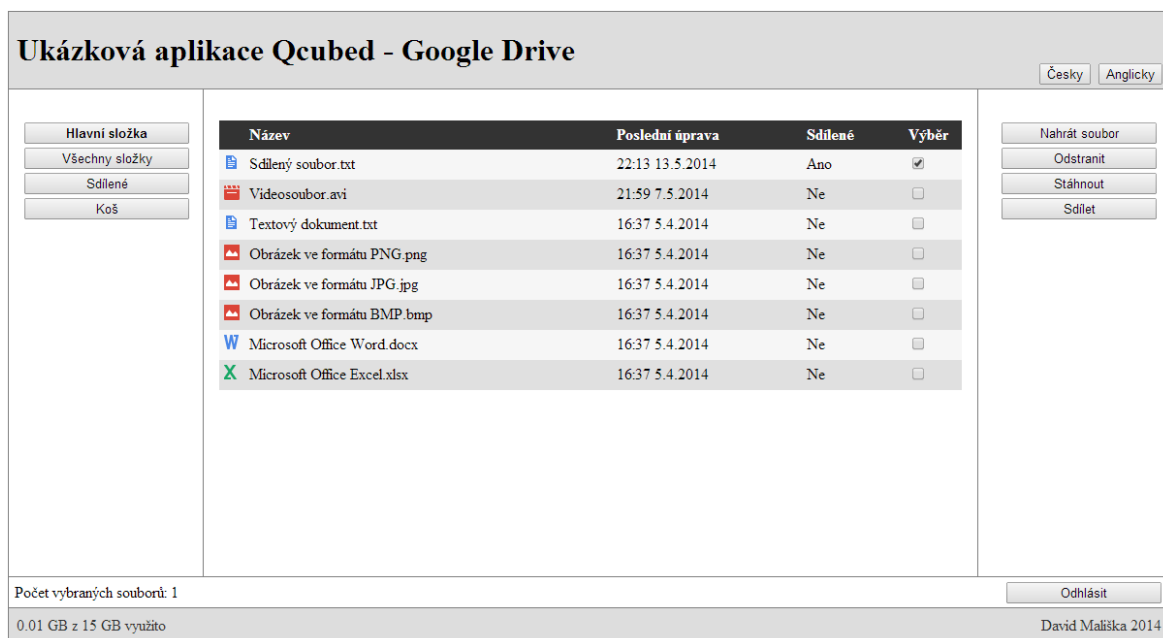
6.3.4 Odstranění a obnovení souboru

Po kliknutí na volbu odstranit, je soubor přesunut do koše. Odtud může být přesunut zpět (tlačítko obnovit), nebo úplně odstraněn z disku (tlačítko trvale smazat).

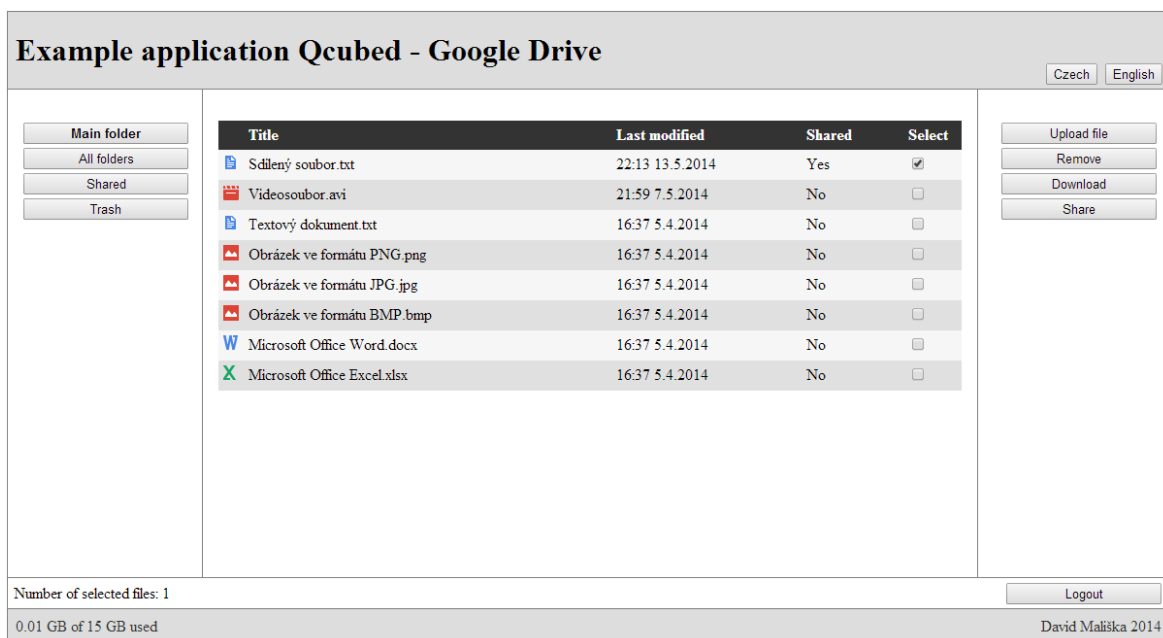
Je zde rozdíl např. oproti Gmailu – soubor v koši není automaticky odstraněn po 30-ti dnech od „vyhození“, ale setrvává zde do další akce uživatele.

6.3.5 Volba jazyka

Ukázková aplikace nabízí výběr mezi dvěma jazyky – Angličtinou a Češtinou. Volbu lze provést v pravém horním rohu. Přeloženo je všechno, kromě názvů souborů a jejich obsahu.



Obrázek 26: Ukázková aplikace (Česká verze)



Obrázek 27: Ukázková aplikace (Anglická verze)

6.3.6 Stažení souboru

Po výběru souboru a kliknutí na tlačítko stažení souboru je zahájeno stahování. Volba umístění uložení závisí na nastavení používaného webového prohlížeče.

6.3.7 Odhlášení

Po kliknutí na tlačítko odhlásit je uživatel odhlášen z aplikace (včetně všech ostatních aplikací společnosti Google jako např. Gmail, YouTube, Google+ apod.) a zobrazí se přihlašovací stránka.

ZÁVĚR

Hlavním cílem této diplomové práce bylo vybrat a implementovat vhodné cloudové úložiště do PHP frameworku QCubed a následně ověřit funkčnost pomocí ukázkové webové aplikace.

Teoretická část se zabývá vysvětlením pojmu Cloud computing a možnostmi jeho využití. Dále jsou popsána cloudová úložiště Google Drive, Microsoft OneDrive, Dropbox, SugarSync, Copy.com, Wuala, Box, Cubby a Capsa.cz. Hlavní pohled je věnován dostupné kapacitě a nabízeným službám.

V praktické části bylo srovnáním těchto poskytovatelů vybráno úložiště Google Drive pro implementaci do PHP frameworku QCubed. Tento open-source framework je založen na softwarové architektuře Model-view-controller a využívá technologií PHP, HTML, CSS a Javascript. S pomocí veřejného API byly vytvořeny nové třídy, umožňující práci s cloudovým diskem od společnosti Google. Do budoucna je možné třídy dále rozšiřovat dle potřeby programátora. Pro lepší orientaci ve zdrojovém kódu slouží přiložená dokumentace.

Ukázková webová aplikace má jednoduchý design a je snadno ovladatelná. Uživatelé umožňuje přihlášení pomocí uživatelského jména (e-mailu) a hesla, odhlášení, zobrazení souborů nacházejících se na cloudovém disku, procházení složek, nahrání nového souboru na cloudový disk, sdílení souboru s ostatními uživateli, stažení, odstranění a obnovení souboru. V levém dolním rohu navíc zobrazuje informaci o využití a celkové kapacitě disku.

SEZNAM POUŽITÉ LITERATURY

- [1] VELTE, Anthony T., Toby J. VELTE a Robert C. ELSENPETER. Cloud Computing: Praktický průvodce. Vyd. 1. Brno: Computer Press, 2011, 344 s. ISBN 978-80-251-3333-0.
- [2] Special Publication 800-145. The NIST Definition of Cloud Computing. U.S.A: National Institute of Standards and Technology, 2011.
- [3] Cloud.cz: server o Cloud computingu. [online]. [cit. 2014-03-05]. Dostupné z: <http://cloud.cz/cloud.html>
- [4] Cloud Computing. [online]. [cit. 2014-03-10]. Dostupné z: <http://cloudcomputing.cz/index.html>
- [5] A History of Cloud Computing. CloudTweaks: PLUGGING INTO THE CLOUD [online]. 9.2.2011 [cit. 2014-03-20]. Dostupné z: <http://cloudtweaks.com/2011/02/a-history-of-cloud-computing/>
- [6] Cloud computing – historie a budoucnost. DD connect: Dimension Data Magazín [online]. 2012 [cit. 2014-03-09]. Dostupné z: <http://www.ddconnect.cz/brezen-2012/datova-centra.html>
- [7] The Differences between IaaS, SaaS, and PaaS. SmartFile: The online file platform [online]. [cit. 2014-03-20]. Dostupné z: <http://www.smartfile.com/blog/the-differences-between-iaas-saas-and-paas/>
- [8] STaaS, SECaaS and 6 Other Cloud Services You Didn't Know Existed. Orange Articles [online]. 2012 [cit. 2014-03-25]. Dostupné z: <http://orangearticles.blogspot.cz/2012/11/staas-secaas-and-6-other-cloud-services.html>
- [9] Storage as a Service (SaaS). Search Storage [online]. 2009 [cit. 2014-03-25]. Dostupné z: <http://searchstorage.techtarget.com/definition/Storage-as-a-Service-SaaS>
- [10] Hands on Science and Enviromental Education. [online]. [cit. 2014-03-20]. Dostupné z: <http://www.hsci2012.org/cloud-computing-and-how-its-changing-our-world/>

- [11] Cloud Computing Outlook: IaaS, PaaS and SaaS. QArea [online]. [cit. 2014-03-25]. Dostupné z: <http://www.qarea.com/articles/cloud-computing-outlook-iaas-paas-and-saas>
- [12] Cloud Types. DefineTheCloud: Cloud Computing [online]. 2010 [cit. 2014-03-26]. Dostupné z: <http://www.definethecloud.net/cloud-types/>
- [13] Výhody Cloud Computing. Cloud Computing. [online]. [cit. 2014-03-10]. Dostupné z: <http://www.cloudcomputing.cz/vyhody.html>
- [14] How it Works. JustCloud.com [online]. 2014 [cit. 2014-03-26]. Dostupné z: <http://www.justcloud.com/how-it-works>
- [15] How Cloud Storage Works. STRICKLAND, Jonathan. HowStuffWorks [online]. 2014 [cit. 2014-04-02]. Dostupné z: <http://computer.howstuffworks.com/cloud-computing/cloud-storage.htm>
- [16] Google Disk About. Google [online]. 2014 [cit. 2014-04-05]. Dostupné z: <http://www.google.com/intl/cs/drive/about.html>
- [17] Google Drive Hepl. Google [online]. 2014 [cit. 2014-04-05]. Dostupné z: <https://support.google.com/drive/?hl=cs#topic=14940>
- [18] Google Drive. Google [online]. [cit. 2014-04-05]. Dostupné z: <https://drive.google.com>
- [19] OneDrive. Microsoft [online]. 2014 [cit. 2014-04-05]. Dostupné z: <https://onedrive.live.com/about/en-us/>
- [20] OneDrive Blog. Microsoft [online]. 2014 [cit. 2014-04-05]. Dostupné z: <http://blog.onedrive.com/>
- [21] OneDrive. Microsoft [online]. [cit. 2014-04-05]. Dostupné z: <https://onedrive.live.com/>
- [22] Dropbox [online]. [cit. 2014-04-05]. Dostupné z: <https://www.dropbox.com/>
- [23] SugarSync [online]. [cit. 2014-04-05]. Dostupné z: <https://www.sugarsync.com/>
- [24] Copy.com [online]. [cit. 2014-04-05]. Dostupné z: <https://www.copy.com/>
- [25] Wuala [online]. [cit. 2014-04-05]. Dostupné z: <http://www.wuala.com/>
- [26] Box [online]. [cit. 2014-04-05]. Dostupné z: <https://www.box.com/>
- [27] Cubby [online]. [cit. 2014-04-05]. Dostupné z: <https://www.cubby.com/>
- [28] Capsa.cz [online]. [cit. 2014-04-05]. Dostupné z: <http://www.capsa.cz/>

- [29] Úložiště dat na internetu: k datům odkudkoli. Svět hardware [online]. 2013 [cit. 2014-04-06]. Dostupné z: <http://www.svethardware.cz/uloziste-dat-na-internetu-k-datum-odkudkoli/>
- [30] Cloudová kancelář: Google vs. Microsoft. Interval.cz [online]. 2013 [cit. 2014-04-06]. Dostupné z: <http://interval.cz/clanky/cloudova-kancelar-google-vs-microsoft/>
- [31] QCubed. [online]. 2014 [cit. 2014-04-07]. Dostupné z: <http://qcu.be/>
- [32] QCubed Documentation - The MVC Architecture. Component Integration [online]. 2014 [cit. 2014-04-07]. Dostupné z: <http://www.c-integration.com/blog/showpost.php/68-qcubed-documentation-the-mvc-architecture>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AaaS	Analytics as a Service.
AES	Advanced Encryption Standard.
API	Rozhraní pro programování aplikací (Application Programming Interface).
APIaaS	Application Programming Interfaces as a Service.
BaaS	Backend as a Service.
CC	Cloud computing.
CSS	Kaskádové styly (Cascading Style Sheets).
DaaS	Data as a Service.
FTP	Protokol pro přenos souborů (File Transfer Protocol).
GB	Jednotka objemu dat (gigabyte).
HTML	Značkovací jazyk pro hypertext (HyperText Markup Language).
IaaS	Infrastructure as a Service.
ID	Identifikační číslo.
IT	Informační technologie (information technology).
JSON	Formát pro výměnu dat (JavaScript Object Notation).
kB	Jednotka objemu dat (kilobyte).
Kč	Koruny české.
LAN	Lokální počítačová síť (Local Area Network).
MB	Jednotka objemu dat (megabyte).
MIME	Typ internetového média (Multipurpose Internet Mail Extensions).
MVC	Softwarová architektura Model-pohled-řadič (Model-view-controller).
NITS	The National Institutes of Technology.
PaaS	Platform as a Service.
PC	Osobní počítač (personal computer).

PHP	Skriptovací programovací jazyk (Hypertext Preprocessor).
PIN	Osobní identifikační číslo (personal identification number).
SaaS	Software as a Service.
SECaaS	Security as a Service.
STaaS	Storage as a Service.
TB	Jednotka objemu dat (terabyte).
TEaaS	Test Environment as a Service.
TV	Televize.
URL	Řetězec identifikace souborů na Internetu (Uniform Resource Locator).
VPN	Virtuální privátní síť (virtual private network).

SEZNAM OBRÁZKŮ

Obrázek 1: Cloud computing [10]	11
Obrázek 2: Modely služeb a typy uživatelů [11]	13
Obrázek 3: Modely služeb [7]	15
Obrázek 4: Grafické zobrazení soukromého cloudu [12]	16
Obrázek 5: Grafické zobrazení komunitního cloudu [12]	17
Obrázek 6: Grafické zobrazení veřejného cloudu [12]	18
Obrázek 7: Grafické zobrazení hybridního cloudu [12]	18
Obrázek 8: Cloudové úložiště [14]	21
Obrázek 9: Google Drive logo [16]	22
Obrázek 10: Webové rozhraní GoogleDrive	23
Obrázek 11: OneDrive logo [19]	24
Obrázek 12: Webové rozhraní Microsoft OneDrive	24
Obrázek 13: Dropbox logo [22]	26
Obrázek 14: Webové rozhraní Dropbox	26
Obrázek 15: SugarSync logo [23]	27
Obrázek 16: Copy.com logo [24]	28
Obrázek 17: Webové rozhraní Copy.com	29
Obrázek 18: Wuala logo [25]	29
Obrázek 19: Java aplikace Wuala	30
Obrázek 20: Diagram Model-view-controller [32]	33
Obrázek 21: Diagram vytvořených tříd	38
Obrázek 22: Ukázková aplikace (layout)	59
Obrázek 23: Ukázková aplikace (přihlášení)	60
Obrázek 24: Ukázková aplikace (nahrání souboru)	60
Obrázek 25: Ukázková aplikace (sdílení souboru)	61
Obrázek 26: Ukázková aplikace (Česká verze)	62
Obrázek 27: Ukázková aplikace (Anglická verze)	62

SEZNAM TABULEK

Tabulka 1: Porovnání kapacit a cen datových úložišť	35
Tabulka 2: Porovnání podporovaných platforem	36

SEZNAM PŘÍLOH

P I Přihlašovací údaje

P II CD s diplomovou prací, dokumentací k vytvořeným třídám a zdrojovými kódy

PŘÍLOHA P I: PŘIHLAŠOVACÍ ÚDAJE

Pro ověření funkčnosti ukázkové webové aplikace byl vytvořen Google účet s následujícími přihlašovacími údaji:

- Uživatelské jméno (e-mail): dp.maliska@gmail.com
- Heslo: diplomovaprace2014