

# Zálohování do cloudových služeb

Bc. Roman Jahoda

---

Diplomová práce  
2017



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2016/2017

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Roman Jahoda**  
Osobní číslo: **A12749**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **prezenční**

Téma práce: **Zálohování do cloudových služeb**

Téma anglicky: **Backing-up to Cloud Applications**

Zásady pro vypracování:

1. Seznamte se s problematikou cloudových úložišť.
2. Vypracujte stručný rozbor technologií, které budou použity k vývoji aplikace.
3. Proveďte analýzu požadavků a uživatelských cílů na zvolené řešení.
4. Navrhněte vhodné řešení aplikace.
5. Realizujte funkční prototyp navržené aplikace.
6. Věnujte pozornost zabezpečení aplikace.
7. Testujte výslednou aplikaci a vyhodnoťte ji z pohledu výkonnosti.



Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

1. TROELSEN, Andrew W. Pro C# 5.0 and the .NET 4.5 framework. Sixth edition. New York, 233 Spring Street: Apress, 2012, lxvii, 1487 pages. ISBN 978-143-0242-338.
2. NATHAN, Adam. WPF 4.5 unleashed. 800 East 96th Street, Indianapolis, Indiana 46240 USA: Sams Publishing, 2013, viii, 845 pages. Unleashed. ISBN 06-723-3697-9.
3. FARLEY, Marc. Rethinking enterprise storage: a hybrid cloud model. Redmond, Washington 98052-6399: Microsoft Press, 2013, pages cm. ISBN 978-073-5679-603.
4. DAVIES, Alex a Allen JONES. Async in C# 5.0. Sebastopol, Calif.: O'Reilly Media, c2012, x, 92 p. ISBN 14-493-3716-3.
5. NELSON, Steven. Pro data backup and recovery. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2011, xiii, 280 p. Expert's voice in data management. ISBN 14-302-2662-5.
6. FREEMAN, Eric a Elisabeth FREEMAN. Head first design patterns. 1st ed. Sebastopol: O'Reilly, 2004, xxxvi, 638 s. ISBN 05-960-0712-4.

Vedoucí diplomové práce: Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce: 20. července 2017

Termín odevzdání diplomové práce: 29. srpna 2017

Ve Zlíně dne 20. července 2017

doc. Mgr. Milan Adámek, Ph.D.  
děkan



prof. Mgr. Roman Jašek, Ph.D.  
ředitel ústavu

Poděkování patří Ing. Petru Šilhavému, Ph.D, za odborné vedení diplomové práce. V neposlední řadě bych chtěl poděkovat své rodině za podporu v průběhu vypracování této práce.

### **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 připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- 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/bakalářské 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**

Práce obsahuje rešerši na téma zálohování do cloudových služeb. Součástí práce je analýza použitých technologií, popis návrhu a popis implementace aplikace. Výsledkem práce je funkční aplikace pro zálohování do cloudových služeb.

Klíčová slova: *záloha, cloud, analýza, .NET, MVVM, WPF, úložiště, DI*

## **ABSTRACT**

A thesis contains research of backup to cloud services. Part of this work is to analyze used technologies, describe application design and implementation. The result is functional cloud backup application.

Keywords: *backup, cloud, analysis, .NET, MVVM, WPF, storage, DI*

# OBSAH

<b>ÚVOD</b> .....	<b>10</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 ÚVOD DO PROBLEMATIKY</b> .....	<b>12</b>
1.1 ZÁLOHOVÁNÍ DAT .....	12
1.1.1 Typy záloh.....	12
1.1.1.1 Úplná záloha .....	13
1.1.1.2 Diferenciální záloha.....	13
1.1.1.3 Inkrementální záloha.....	14
1.2 CLOUDOVÁ ÚLOŽIŠTĚ.....	15
1.2.1 Veřejný cloud .....	15
1.2.2 Privátní cloud .....	15
1.2.3 Hybridní cloud .....	16
1.2.4 Historie a současnost cloudu.....	16
1.2.5 Přehled současných osobních cloudových úložišť.....	17
1.2.5.1 DropBox.....	17
1.2.5.2 Google Drive.....	19
1.2.5.3 One Drive.....	20
1.2.5.4 iCloud Drive .....	21
1.2.5.5 Amazon Simple Storage Service (Amazon S3).....	22
1.3 EXISTUJÍCÍ ŘEŠENÍ.....	24
1.3.1 MultCloud .....	24
1.3.2 Duplicati 2.0.....	25
<b>ANALÝZA TECHNOLOGIÍ</b> .....	<b>27</b>
1.4 .NET FRAMEWORK .....	27
1.5 WINDOWS PRESENTATION FRAMEWORK.....	28
1.5.1 DataBinding .....	28
1.5.2 MVVM.....	30
1.5.2.1 Model.....	30
1.5.2.2 ViewModel .....	30
1.5.2.3 View.....	31
<b>II PRAKTICKÁ ČÁST</b> .....	<b>32</b>
<b>2 ANALÝZA POŽADAVKŮ A UŽIVATELSKÝCH CÍLŮ NA ZVOLENÉ ŘEŠENÍ</b> .....	<b>33</b>
2.1 CÍLE PROJEKTU.....	33
2.2 KRITÉRIA ÚSPĚŠNOSTI.....	33
2.3 PŘÍPADY UŽITÍ.....	34
2.3.1 UC01: Vytvořit zálohovací plán .....	34
2.3.2 UC02: Editovat zálohovací plán .....	36
2.3.3 UC03: Smazat zálohovací plán .....	36
2.3.4 UC04: Přidat nové úložiště .....	37
2.3.5 UC05: Autorizovat přístup (DropBox) .....	38
2.3.6 UC06: Autorizovat přístup (GDrive) .....	39
2.3.7 UC06: Prohlížet vytížení úložišť.....	40
2.3.8 UC07: Prohlížet zálohy.....	40

2.3.9	UC08: Obnovit data ze zálohy .....	41
2.3.10	Algoritmus zálohování .....	42
2.4	WIREFRAME HLAVNÍHO OKNA APLIKACE .....	43
<b>3</b>	<b>SOFTWAREVÉ ŘEŠENÍ .....</b>	<b>45</b>
3.1	POPIS STRUKTURY APLIKACE.....	45
3.1.1	Projekt CloudBackupManager.Shell.....	46
3.1.2	Projekty CloudBackupManager.DataCore a CloudBackupManager.Data .....	46
3.1.3	Projekt CloudBackupManager.Infrastructure .....	47
3.1.4	Projekt CloudBackupManager.Backup.....	48
3.1.5	Projekt CloudBackupManager.ProviderForDropbox .....	48
3.1.6	Projekt CloudBackupManager.ProviderForGDrive.....	49
<b>4</b>	<b>POPIS APLIKACE .....</b>	<b>50</b>
4.1	BACKUPPLANDETAILEDVIEW.....	50
4.1.1	Backup Plan Settings.....	51
4.1.2	Backup paths settings .....	51
4.1.3	Time Settings .....	53
4.2	MAINVIEW.....	54
4.2.1	Záložka Info .....	55
4.2.2	Záložka Backups .....	56
4.2.3	Záložka Providers.....	57
4.2.4	Záložka Log .....	58
4.2.5	Záložka OperationQueue .....	58
4.3	NEWPROVIDERDIALOGVIEW .....	58
<b>5</b>	<b>POPIS IMPLEMENTACE.....</b>	<b>61</b>
5.1	POUŽITÉ KNIHOVNY .....	61
5.1.1	PRISM.....	61
5.1.2	MahApps.Metro .....	61
5.1.3	DropBox.Net .....	61
5.2	ZÁKLADNÍ LOGIKA APLIKACE.....	61
5.2.1	Dependency injection.....	61
5.2.2	Popis implementace třídy BootStraper.....	63
5.2.3	Popis implementace třídy NotifyPropertyChanged .....	64
5.3	POPIS IMPLEMENTACE TŘÍDY BACKUPPLAN.....	65
5.4	POPIS IMPLEMENTACE SLUŽBY BACKUPPLANSERVICE .....	66
5.5	IMPLEMENTACE SLUŽBY BACKUPPROVIDERMANAGER .....	67
5.5.1	IBackupProviderService .....	68
5.5.2	IBackupProvider .....	69
5.6	POPIS IMPLEMENTACE TŘÍDY BACKUPSERVICE.....	71
5.7	ZABEZPEČENÍ.....	71
5.7.1	OAuth 2.0.....	71
5.7.2	Řešení zabezpečení citlivých údajů .....	72
<b>6</b>	<b>ZHODNOCENÍ PROGRAMU Z HLEDISKA VÝKONNOSTI .....</b>	<b>74</b>



6.1	TESTOVACÍ SESTAVA:.....	75
6.2	TESTOVACÍ VZORKY DAT .....	75
6.3	TEST LOKÁLNÍHO ZÁLOHOVÁNÍ VZORKU 1 .....	75
6.4	TEST LOKÁLNÍHO ZÁLOHOVÁNÍ VZORKU 2 .....	76
6.5	TEST LOKÁLNÍHO ZÁLOHOVÁNÍ VZORKU 2 A NASTAVENOU INKREMENTÁLNÍ ZÁLOHOU.....	76
6.6	TEST ZÁLOHOVÁNÍ VZORKU 3 NA SLUŽBU DROPBOX .....	77
6.7	TEST ZÁLOHOVÁNÍ NA SLUŽBU DROPBOX SE VZORKEM 4 .....	77
6.8	TEST ZÁLOHOVÁNÍ VZORKU 3 NA SLUŽBU GOOGLE DRIVE .....	78
	<b>ZÁVĚR .....</b>	<b>79</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>80</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>83</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>84</b>
	<b>SEZNAM ZDROJOVÝCH KÓDŮ .....</b>	<b>86</b>
	<b>SEZNAM TABULEK.....</b>	<b>87</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>88</b>

## ÚVOD

Pojem cloud je v poslední době velmi moderní pojem, který zaštiťuje širokou škálu služeb a technologií. Jednou z těchto služeb je veřejné cloudové úložiště, které je určené ke sdílení, zálohování a hlavně možnosti přístupu ke svým datům prakticky odkudkoliv, kde má uživatel přístup k internetu. Konkurence v tomto odvětví je velká a většina společností již nabízí široké veřejnosti tarif zdarma, který se pohybuje kolem 2GB. Díky této skutečnosti si mohou i běžní uživatelé efektivně chránit svá cenná data, aniž by museli kupovat externí disky, či jiná média.

Cílem této práce je vytvořit aplikaci pro operační systém Windows, která umožní uživateli jednoduše zálohovat data na některé z cloudových úložišť a umožnit snadnou budoucí rozšiřitelnost o novou funkcionalitu. Součástí je také funkcionalita rovnoměrného rozdělování záloh podle vytížení jednotlivých úložišť.

Teoretická část této práce obeznámí čtenáře s riziky ztráty dat, vysvětlí, co to vlastně ta záloha je a jak se provádí. V další části bude vysvětlen pojem cloudové úložiště a představeny jednotlivé možnosti, které se v dnešní době nabízí. Také budou představeny existující řešení.

Problematika zálohování a cloudových služeb zahrnuje tak širokou škálu informací, že je nebylo možné zanést do kontextu této práce. Vzhledem k tomu se teoretická část soustředí na základní informace o dané problematice tak, aby čtenář bez předchozích znalostí pochopil praktické chování aplikace a byl ji tak schopen používat.

Součástí práce je i rozbor technologií, které byly použity k vývoji aplikace, zahrnující převážně studium komponent platformy .Net, spolu s návrhovými vzory a doporučenými postupy vývoje aplikace cílené na technologii WPF, které byly při implementaci programu prakticky použity.

Praktická část práce obsahuje stanovené cíle a požadavky na cílovou aplikaci, spolu s komplexním popisem výsledného řešení. Tento popis obsahuje rozbor modulů, ze kterých se výsledná aplikace skládá a praktické příklady zajímavých částí programu. Dále praktická část pokračuje popisem uživatelského rozhraní aplikace a vysvětlí způsob použití jednotlivých dialogů tak, aby tato část mohla sloužit jako uživatelská příručka.

V poslední části práce se nachází testování a vyhodnocení programu z hlediska výkonnosti.

## **I. TEORETICKÁ ČÁST**

## 1 ÚVOD DO PROBLEMATIKY

V dnešní době informačních technologií prostupuje digitalizace do našich životů čím dál tím více. Fotky, dokumenty, filmy a data obecně, jsou již dnes převážně ukládány v elektronické podobě. Bohužel je zde celá škála situací, které ohrožují naše data.

- Technické selhání: Porucha úložného zařízení, na kterém jsou data umístěna.
- Viry: Škodlivé programy, které mohou způsobit poškození, nebo ztrátu dat.
- Chyba programového vybavení: Chyba v programu může poškodit data, s kterými manipuluje.
- Chyba uživatele: Uživatel nechtěně zaviní ztrátu, nebo poškození dat. Vztahuje se sem i poškození, nebo ztráta samotného zařízení. [1]

Pro tyto případy je data potřeba zálohovat na jiné úložné zařízení, než na kterém se data nachází. Dříve se i drobná data musela zálohovat na externí pevné disky, flash disky, nebo DVD disky, avšak dnes jsou možnosti podstatně flexibilnější díky cloudovým úložištím. [2]

### 1.1 Zálohování dat

Zálohování je proces, při kterém se vytváří kopie otisku dat v určitý moment. Provádí se za účelem ochrany vybraných dat před jejich ztrátou. Tento otisk je poté uchováván po určitou dobu užitečnosti.[3]

Nejčastější příčinou ztráty dat je selhání lidského faktoru, kdy se uživatel dopustí nevědomky chyby, která zaviní poškození dat. Data ovšem mohou být poškozena i průnikem zvenčí, který se v mnoha případech dá také zařadit do selhání lidského faktoru.[1] Jako příklad se může uvést kryptovirus, který si uživatel zanesl do počítače nedbalým otevřením přílohy emailu. Po inicializaci kryptovirus zašifruje vybraná data a zanechá textový soubor, který uživatele vydírá k zaplacení určité částky.[4]

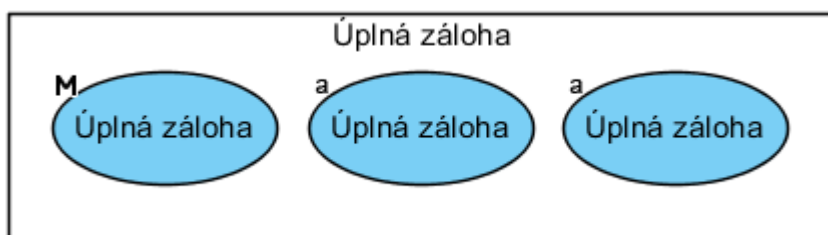
#### 1.1.1 Typy záloh

Pro zálohování existuje několik různých metod. Pro zvolení té správné metody je třeba se zaměřit na to, jak často potřebujeme zálohovat a jakou dostupnou kapacitu máme k dispozici. Stejně tak je potřeba zvážit, jak rychle je nutné data obnovit v případě ztráty a poškození. Zvolený typ zálohy spolu s vybraným médiem kam se záloha bude ukládat ovlivňuje rychlost zálohy, stejně jako obnovy, protože některé typy záloh jsou závislé na předchozích iteracích záloh.[2][3]

V praxi je možné navrhnout různé typy záloh a jejich kombinace. V další části bude popis typů záloh jen obecného charakteru, protože konkrétní implementace se může lišit operačními systémy, které zálohování nabízejí, ale také i speciálními zálohovacími aplikacemi, které jsou určeny pro zálohu uživatelských, či firemních dat.[6]

### 1.1.1.1 Úplná záloha

Tento typ zálohy označuje zálohu všech zvolených dat. Může se jednat o celý obraz disku, včetně operačního systému, nebo jen o určitou adresářovou strukturu, který obsahuje cenná data. Tento typ zálohy je časově a technicky nejnáročnějším řešením pro zálohu dat, které v sobě nese nevýhodu v podobě časové náročnosti vytvoření zálohy a velkou náročnost na dostupnou kapacitu. Proto se tento typ zálohy provádí ve firemní praxi s většími časovými rozestupy, než u jiných typů záloh. [3]

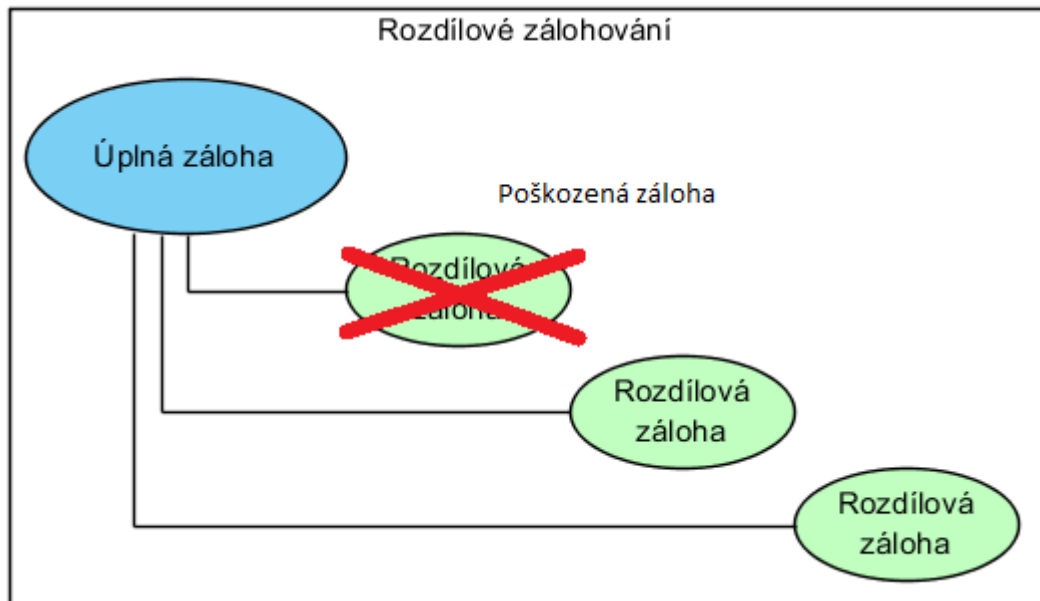


Obrázek 1 Schéma úplného zálohování

### 1.1.1.2 Diferenciální záloha

Do diferenciální zálohy se zálohují data, která se změnila od poslední úplné zálohy. Pokud úplná záloha ještě neexistuje, je potřeba ji udělat. V případě ztráty dat je pro obnovení nutné mít k dispozici předešlou úplnou zálohu a požadovanou zálohu diferenciální. Výhoda této metody je rychlá obnova a menší závislost na předchozích zálohách, než u zálohy inkrementální, která bude popsána níže. Případné poškození některé z diferenciálních záloh neovlivní jiné diferenciální zálohy. Diferenciální záloha se dá považovat za kompromis mezi zálohou úplnou a zálohou inkrementální.

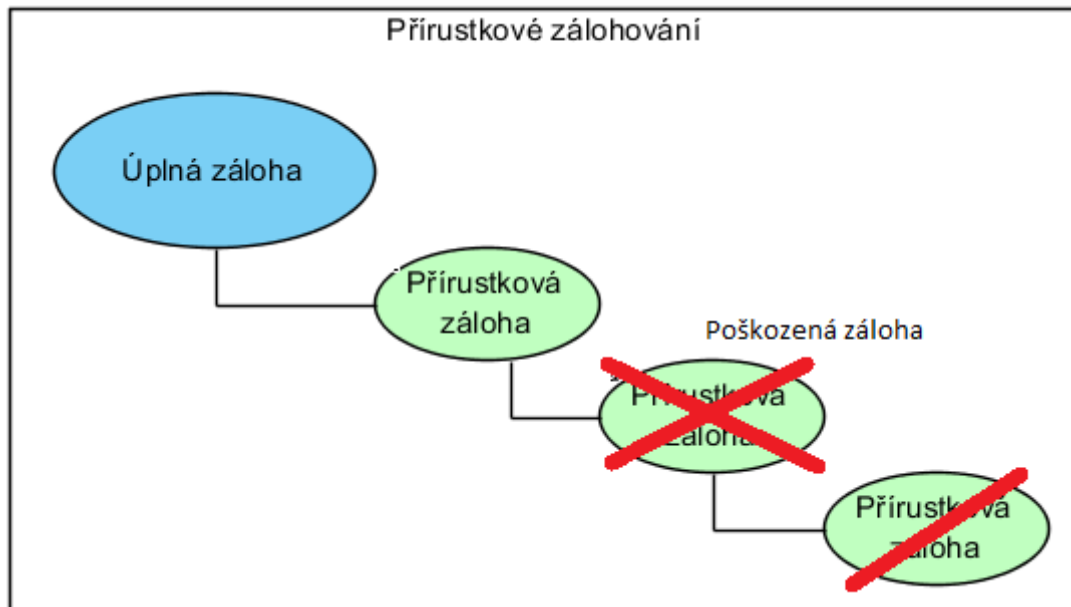
Nevýhodou tohoto typu zálohy je postupem času narůstající velikost diferenciální zálohy od zálohy úplné, kde se můžeme dostat do stavu, kdy už je diferenciální záloha větší než poslední úplná záloha. Z tohoto důvodu je potřeba vytvářet pravidelně i zálohy úplné. [3][6]



Obrázek 2 Schéma rozdílového zálohování

### 1.1.1.3 Inkrementální záloha

Tato metoda zálohování je stejně jako diferenciální záloha závislá na záloze úplné, ale navíc je závislá i na všech předchozích inkrementálních zálohách. Tento typ nabízí nejrychlejší metodu zálohování a nejnižší nároky na velikost úložného prostoru, avšak na úkor rychlosti obnovy a závislosti na předchozích zálohách. V případě obnovy je potřeba obnovit všechny zálohy od úplné, až do požadované zálohy inkrementální. Tento způsob obnovy má nevýhodu v tom, že v případě poškození některé ze zálohy, jsou všechny inkrementální zálohy v řetězci za ní neplatné. Z tohoto důvodu je, stejně jako u záloh diferenciálních, nezbytné dělat pravidelně zálohy úplné. [3]



Obrázek 3 Schéma přírůstkového zálohování

## 1.2 Cloudová úložiště

Cloudové úložiště je druh cloudové služby, ve které jsou vzdáleně spravovány data tak, aby byly přístupné uživatelům přes nějakou síť, typicky internet. Ve výsledku to znamená, že svá data uložíme na server, který je dostupný z internetu. Obecně se cloudové služby dají dělit na 3 hlavní modely nasazení, které budou popsány v následujících kapitolách. [5]

### 1.2.1 Veřejný cloud

Veřejná cloudová úložiště poskytují takzvané multi-tenant úložiště, které je vhodné pro nestrukturovaná data. Multi-tenant je architektura, kde je jedna instance softwaru, která obsluhuje vícero uživatelů, avšak zajišťuje jejich vzájemné oddělení, aby si uživatelé nemohli navzájem přistupovat k soukromým informacím a prostředkům. Tato architektura je výhodná hlavně v tom, že náklady na vývoj a údržbu jsou sdílené. V cloud computingu je tento výraz ještě širší vzhledem k tomu, že nové modely služeb využívají virtualizaci a vzdálený přístup. [5]

### 1.2.2 Privátní cloud

Privátní cloudové úložiště je dedikované prostředí, které je schováno za firemním firewallem. Tento druh je vhodný hlavně pro klienty, kteří požadují větší kontrolu a flexibilitu nad svými daty. [5]

### 1.2.3 Hybridní cloud

Tento typ je kombinace veřejného a privátního cloudu, který je používán primárně ve firemním prostředí. V praxi to znamená, že veřejná a privátní část je propojena, zpravidla dedikovanou linkou. Ve veřejné části se nachází data, které jsou pro firmu méně důležitá. V privátní části se uchovávají kritická a často používaná data. Pro funkčnost tohoto typu cloudu je potřeba hybridní prostředí, které nabízí například firma Microsoft se svým produktem Microsoft Azure. Mezi další společnosti, které nabízí tuto technologii patří například HP, Rackspace, VMware a další. [7]

### 1.2.4 Historie a současnost cloudu

Myšlenka cloud computingu vznikla již v polovině dvacátého století, kdy profesor z americké univerzity MIT John McCarthy v roce 1961 prezentoval myšlenku sdílení počítačových prostředků jako služby. Například stejně tak jak sdílíme elektřinu, kdy běžné domácnosti a firmy nemají vlastní elektrárnu, ale jsou připojené do elektrorozvodné sítě, ze které odebírají spolu s ostatními odběrateli potřebné množství elektrické energie. Tato myšlenka předběhla svou dobu vzhledem k tomu, že zahrnovala softwarovou a hardwarovou virtualizaci, která v roce 1961 ještě neexistovala. Elektráren v elektrorozvodné síti je hned několik a jsou mezi sebou propojené, takže v případě výpadku jedné z elektráren mohou zátěž přebrat ostatní elektrárny, které v případě, že stačí zvládat dodatečnou zátěž, tak cíloví odběratelé výpadek ani nepocítí.[5]

- Elektrárna – Datové centrum
- Elektrorozvodná síť – Internet
- Odběratel – Uživatel
- Spotřebiče – Notebooky, Telefony, Počítače

Pojem „Cloud computing“ se objevil až v roce 1997 v přednášce Ramnanta Chellapa, který jej definoval jako nové paradigma, kde hranice computingu budou racionalizovány spíše ekonomickými důvody, než samotnými technickými limity. Slovo a vyjádření si Cloud vypůjčil od telekomunikací, kde se koncové stanice zobrazují jako krabičky připojené do oblaku s označením internet.[5]

V současnosti cloudové služby generují zisk v řádech miliard dolarů a do budoucna se předpokládá, že se tento zisk bude nadále zvyšovat.[15] Největší model cloud computingu momentálně zastupuje Software as a Service (SaaS). Tento segment generuje



více než polovinu zisku z celého Cloud computingu. [16] Segment je založen na tom, že poskytuje klientovi vzdálený přístup k vytvořeným aplikacím, nebo databázím, zatímco infrastrukturu spravuje poskytovatel služby. Díky eliminaci nutnosti instalace a provozu aplikací na vlastních zařízeních se ze SaaS stal velmi oblíbený způsob provozu aplikací. SaaS také umožňuje redukovat klientům náklady na nákup softwaru, jelikož licence on-demand mají většinou nižší ceny a zároveň odpadnou firmě ostatní náklady, jako například hardware, licence na servery a další dodatečné náklady. [21]

Další model se označuje jako Platform as a Service (PaaS), což představuje službu, která poskytuje kompletní prostředky pro vývoj a údržbu vlastních aplikací, které jsou dostupné prostřednictvím internetu. Konkrétní aplikace si již musí klient vyvinout sám, na rozdíl od výše popsaného modelu SaaS, kde si klient využívá (pronajímá) aplikace napsané poskytovatelem služby. Tento koncept není v současné době tak moc využíván, jako ostatní koncepty. [21]

V neposlední řadě je zde Infrastructure as a Service (IaaS), kde poskytovatel služby poskytne zákazníkovi svoji infrastrukturu, neboli hostuje servery, úložiště, software a další komponenty namísto uživatele. Poskytovatel se také stará o údržbu, zálohování a další úlohy, které by jinak musel dělat klient sám. Vzhledem k tomu, že infrastruktura patří poskytovateli, tak může být monitorování, management systému obtížný a stejně jako u ostatních modelů, pokud poskytovatele zastihne výpadek, na který není připravený, tak může být postihnuta i produktivita klienta. [21]

### **1.2.5 Přehled současných osobních cloudových úložišť**

V následujících bodech budou popsány současné cloudové úložiště spolu s tabulkami ceníků služeb.

#### **1.2.5.1 DropBox**

Dropbox je jedno z nejznámějších cloudových úložišť, používané pro sdílení a zálohu dokumentů a dat. Je známý svojí jednoduchostí a spolehlivostí. Pro využití služby si stačí založit účet. Pro založení účtu není potřeba zakládat nový email, jako je třeba v případě Google Drivu, pro jehož použití je nutné mít Google účet, který zastřešuje všechny služby, nejen úložiště. DropBox úložiště poskytuje tuto službu samostatně. Pokud již uživatel vlastní účet od Googlu, tak jej může využít pro přihlašování, aniž by si musel zakládat další účet pro DropBox.



Obrázek 4 Logo služby DropBox

Silnou stránkou této služby je intuitivní uživatelské rozhraní. Uživatel může pár kliky získat odkaz pro sdílení, nebo zaslat odkaz souboru přímo na email, aniž by musel zkopírovat odkaz a pak ho odeslat ze svého emailového klienta.

Zabezpečení této služby je stejně jako u většiny velkých hráčů na vysoké úrovni. Pro zabezpečení používá 256 bitovou AES šifru. Data jsou zašifrována přímo i na serverech, takže si naše data nikdo neprohlíží. Nabízí také dvou krokovou verifikaci, která když je aktivována, tak vyžaduje pro každé nové zařízení, přes které se uživatel na službu přihlásí šesti místný číselný kód, který uživatel opíše z autorizační aplikace, kterou si nainstaluje na telefon, nebo také může přijít formou SMS. Nevýhoda tohoto způsobu autorizace spočívá v tom, že bez zapnutého telefonu se nemáme jak autorizovat, na což by si uživatelé měli dávat pozor vzhledem ke krátkým výdržím dnešních chytrých telefonů. Dále je také možné využít takzvaný USB key, což je malá USB klíčenka, kterou při autorizaci jednoduše vložíme do počítače a dokončíme přihlášení. Výhodou je rychlá autorizace a hlavně toto zařízení nepotřebuje baterii.

Služba v základu nabízí bezplatný účet s 2 GB volného místa. Pokud uživateli účet zdarma nestačí, tak má možnost zakoupit placenou variant. K dispozici jich je hned několik a kromě dalšího volného místa nabízí i dodatečné služby, jako je správa skupin, integrace s Office 365 a další.

Tabulka 1 Ceník služby Dropbox

Aktuální ceník služby DropBox		
Typ účtu	Dostupná kapacita	Cena
Dropbox Basic	2 GB	Zdarma
Dropbox Plus	1TB	9.99€ měsíčně, nebo 99 € ročně
Dropbox Bussines Standart	2TB	10€ měsíčně za uživatele
Dropbox Bussines Advanced	Neomezeně	15€ měsíčně za uživatele
Dropbox Bussines Enterprise	Na dohodě	Na dohodě

### 1.2.5.2 Google Drive

Služba Google Drive je cloudové úložiště, pro jehož používání musíme mít založený Google Email. Založením emailu získáváme účet, který zastřešuje celou škálu služeb jako jsou mapy, cloudové úložiště, youtube, google play, správa dokumentů a další. Pomocí tohoto účtu se můžeme přihlašovat i do cloudového úložiště DropBox a dalších stránek, které podporují přihlašování pomocí Google Účtu.

Toto úložiště je hlavně určeno pro ukládání dokumentů vytvořené v Google Docs. Je nutné podotknout, že Google jako první vytvořil cloudový ekvivalent osvědčených Microsoft Office. Společnost Microsoft tyto služby zavedla do svých prohlížečů až po zavedeném řešení společnosti Google. Do úložiště samozřejmě můžeme ukládat jakékoliv soubory, které nemají více než 5TB a nepřesahují limit účtu.



Obrázek 5 Logo služby Google Drive

Rozhraní Google Drive disponuje intuitivním uživatelským rozhráním, které umožňuje jednoduché sdílení souborů, nebo dokonce i celých složek. Můžete jinému uživateli přidělit práva do své složky, a on ji pak uvidí jako by se nacházela přímo v jeho úložišti.

Pro zabezpečení používá 128 bitovou AES šifru a pro přenos dat standart TSL 2.0. Stejně jako DropBox nabízí dodatečnou ochranu pomocí dvoukrokové verifikace. Google vyhrazuje právo na prohlížení a analýzu všeho co na něj nahrajeme. Pokud by tohle někomu vadilo, tak se musí poohlédnout po jiné službě anebo data zašifrovat ještě před nahráním na úložiště.

Oproti DropBoxu dostaneme zdarma kapacitu 15 GB, což je několikanásobně více a hlavně lákavá nabídka pro založení Google Účtu. [20]

Tabulka 2 Ceník služby Google Drive

Aktuální ceník služby Google Drive	
Kapacita	Cena
15GB	Zdarma
100 GB	59,99 Kč měsíčně
1TB	299,99 Kč měsíčně
10TB	2999,99 Kč měsíčně
20TB	5999,99 Kč měsíčně
30TB	8999,99 Kč měsíčně

### 1.2.5.3 One Drive

Pro využití této služby musíme mít Microsoft Účet. Tato cloudová služba je pro uživatele operačního systému Microsoft Windows nejjednodušší řešení, vzhledem k tomu, že tato služba je zabudována ve všech novějších verzích tohoto operačního systému. Pokud se do počítače přihlásíme pod svým Microsoft účtem, tak se nám úložiště automaticky zobrazí v prohlížeči souborů. U ostatních úložišť musíme pro tuto funkcionalitu instalovat dodatečné aplikace. Do adresáře úložiště jednoduše přetáhneme soubor, který si přejeme nahrát, a aplikace na pozadí soubor synchronizuje s úložištěm. Tato funkce funguje oboustranně, takže pokud soubor nahrajeme z jiného zařízení, tak se nám automaticky synchronizuje do dané složky. U souboru je zobrazen i příznak, zda je soubor již nahrán na zařízení, a nebo je momentálně dostupný pouze na cloudovém úložišti.



Obrázek 6 Logo služby One Drive

Jak již bylo zmíněno One Drive je úzce spjatý s operačními systémy Microsoft Windows, ale jsou dostupné i oficiální aplikace pro ostatní mobilní a desktopové operační systémy kromě operačního systému Linux, kde je k dostání například neoficiální aplikace OneDrive by Skilion.

Pokud si uživatel předplácí balík Microsoft Office 356, tak k němu dostane 1TB úložného prostoru.

Tabulka 3 Ceník služby DropBox

Aktuální ceník služby DropBox		
Tarif	Dostupná kapacita	Cena
Office 365 Home with Premium OneDrive Features	5TB (1TB pro 5 uživatelů)	99.99\$ ročně
Office 365 Personal with Premium OneDrive Features	1TB	69.99\$ ročně
Storage only OneDrive 50 GB	50GB	1.99\$ měsíčně
Storage only OneDrive Basic 5 GB	5GB	Zdarma
Storage + Office 365 Office 365 Business Premium	1TB	150\$ ročně za 1 uživatele
Storage only OneDrive for Business Plan 1	Neomezeně, začíná se s 5 TB místa a pokud uživatel potřebuje víc tak musí kontaktovat support a požádat o dodatečné místo.	120\$ ročně za 1 uživatele
Storage only OneDrive for Business Plan 2	1TB	60\$ ročně za 1 uživatele

#### 1.2.5.4 iCloud Drive

iCloud Drive je cloudové úložiště společnosti Apple, které stejně jako ostatní úložiště umí synchronizovat soubory napříč všemi zařízeními. Zařízení společnosti Apple mají podporu tohoto úložiště integrovanou v operačním systému obdobně jako tomu je u OneDrive. Při založení účtu iCloud dostane uživatel 5GB zdarma a v případě potřeby si může zvýšit kapacitu až na 2TB, za určité měsíční poplatky, které budou vypsány v tabulce níže.

Na zařízeních s operačním systémem Mac OS, lze využít funkci synchronizace plochy a dokumentů, které automaticky synchronizuje všechny soubory a složky na ploše a ve složce Dokumenty na cloud. Pokud z plochy soubor přesuneme do jiné složky, tak se z cloudového úložiště smaže, a uvolní tak kapacitu. Díky této funkcionalitě máme vždy automaticky zálohovaná data a zároveň přístupná ze všech ostatních zařízení.

Je dostupná i oficiální aplikace pro operační systém Windows. Tato aplikace nám umožní synchronizovat emaily, dokumenty, kontakty, kalendáře a hlavně soubory. Stejně

tak se objeví úložiště v záložce oblíbené, obdobně jako tomu je u ostatních úložišť.



Obrázek 7 Logo služby iCloud Drive

Aktuální ceník služby iCloud Drive	
Kapacita	Cena
5 GB	Zdarma
50 GB	25 Kč měsíčně
200 GB	79 Kč měsíčně
2 TB	249 Kč měsíčně

Tabulka 4 Ceník služby iCloud [19]

#### 1.2.5.5 Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service se od výše uvedených cloudových služeb ve velké míře liší. Je cílena hlavně pro firmy a vývojáře webových aplikací, což nám ale nebrání použít tuto službu pro jiné účely.

Je zde zcela rozdílná politika účtování služby. K dispozici máme neomezený prostor a platíme měsíční poplatky pouze za to, co využíváme. Každý nový uživatel dostane 5GB prostoru zdarma na jeden rok, pokud uživatel tento, nebo jiný limit překročí, tak již začne platit měsíčně rozdíl. Například pokud bychom využili 6GB prostoru tak zaplatíme stále jen 1GB za měsíc, protože zbylých 5 GB je zdarma. Konfigurací a možných služeb je opravdu velké množství, vzhledem k tomu můžeme platit 3\$ měsíčně, nebo i 300000\$ v závislosti na tom kolik služeb využíváme, kolik dat máme umístěno na úložišti a také jak často k těmto datům přistupujeme a manipulujeme s nimi.

Platíme za:

- Využitý prostor
- Stažená data (Upload na úložiště se neplatí)
- PUT/COPY/POST/LIST requesty
- GET a ostatní requesty
- Ostatní služby a funkce (Duplikace, Akcerelace rychlosti Up/Down, atd.)

Ze všeho nejdříve je nutné založit účet na Amazon Web Services. Registrace je bezplatná, ale je nutné mít k dispozici platební kartu a telefon, na který vám při ověření identity zavolá skryté číslo a vyzve uživatele k zadání pinu, který se nachází ve formuláři registrace.

Abychom mohli využívat úložiště, tak si musíme založit takzvaný „bucket“, což si můžeme představit jako kontejner, do kterého ukládáme svoje data. Těchto kontejnerů můžeme libovolné množství a musíme určit, kde se budou nacházet. Je vhodné je umístit do lokací, které jsou k uživateli nejbližší, abychom dosáhli co nejlepší latence. Je důležité poznamenat, že za transfer dat mezi buckety se také platí, pokud nejsou ve stejném regionu. K dispozici je také celá škála služeb, které toto cloudové úložiště odlišuje od konkurence, např.:

- Funkce DevPay, která zajišťuje autentikační a platební infrastrukturu. Například účtování zákazníka jednou měsíčně za přístup k produktu a datům, které jsou uloženy na Amazon S3. Amazon si jednou měsíčně strhne poplatky za transakci a rozdíl nám vrátí. Poté si zvlášť strhne částku, kterou klient spotřeboval v rámci používání (Přenosy dat, Get requesty atd.) produktu a procentuální poplatek. Pokud zákazník neplatí, tak Amazon zákazníkovi odepře přístup k produktu.
- Funkce, která umožňuje účtovat zákazníka za stažení dat z bucketu. Zákazník platí za požadavek a stažení dat. Majitel bucketu platí za skladování dat.
- Podpora BitTorrentu.
- Hostování statických webových stránek
- Správa životního cyklů dat v bucketu. Můžeme například nastavit, že se data starší než 30 dnů smažou, nebo přesunou na jiné místo do archivu.

Amazon S3 samozřejmě podporuje sdílení souborů přes link a ostatní běžné operace, stejně jako ostatní cloudová úložiště. Pro většinu operačních systémů existuje klient, který umožňuje procházení, upload, download. Jak již bylo výše zmíněno, tak je podporován i BitTorrent protokol. Pro každý veřejně přístupný soubor můžeme vygenerovat takzvaný

torrent soubor (\*.torrent), který se stáhne do uživatelského počítače, a ten pak může sdílet s ostatními. [14]

Jak již bylo řečeno, Amazon S3 je určený hlavně pro vývojáře a firmy. Díky možnostem které nabízí, stojí na předních příčkách na trhu s cloudovými úložišti, ale není zcela vhodný pro běžného uživatele, vzhledem k velkému množství funkcí, nastavení a variabilní cenové politice, která může běžného uživatele zahltit.

### 1.3 Existující řešení

V následujících bodech budou popsány existující řešení zálohovacích programů, které umí zálohovat do více různých cloudových služeb.

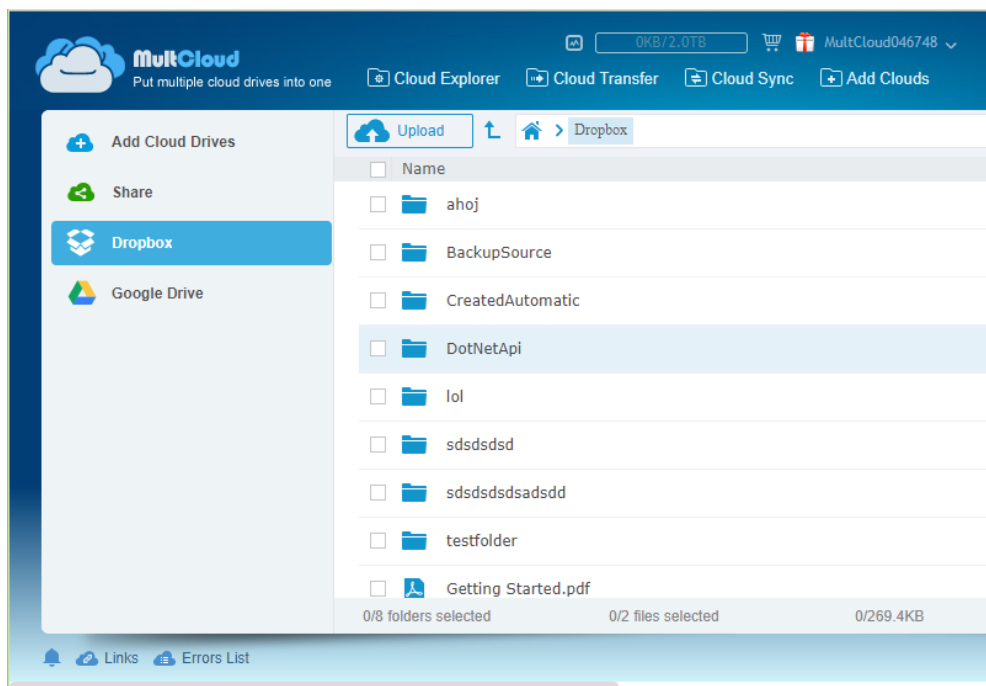
#### 1.3.1 MultCloud

Tato webová aplikace umožňuje správu a prohlížení několika různých cloudových úložišť z jednoho uživatelského rozhraní. V současné podobě podporuje až 30 různých druhů úložišť včetně FTP, stejně tak jako možnost přidat více účtů z jednoho úložiště. Přidané účty se zobrazí v seznamu a uživatel mezi nimi může proklikávat. Uživatelské rozhraní aplikace je na špičkové úrovni, jednoduché a intuitivní.

Aplikace umožňuje transfer souborů napříč úložišti, stejně tak i jejich synchronizaci v rámci složek, například pokud změním soubor v jednom úložišti, tak se soubor přehraje i do spárované složky na ostatních úložištích, lze si zvolit i režim synchronizace. Kromě jednotlivých složek lze synchronizovat celá úložiště. Součástí je i možnost nastavit čas, nebo interval, kdy se mají soubory přehrát. Po dokončení je uživateli zaslán email, který potvrzuje dokončení.

Veškeré přenášení dat a jiné operace, jsou prováděny ze serveru společnosti, což sebou nese různé výhody a nevýhody. Nelze automaticky zálohovat soubory a složky, které jsou umístěné na uživatelském notebooku, či PC. Uživatel musí otevřít prohlížeč a soubory manuálně nahrát. Nebo použít jiný program pro synchronizaci do jedné ze složek ve vybrané cloudové službě a v MultCloud nastavenou periodickou zálohu na daný adresář, která zazalohuje data do ostatních cloudových služeb.





Obrázek 8 MultCloud uživatelské rozhraní

Výhoda toho přístupu je, že se pro přenos používá internetové připojení společnosti MultCloud, které je určitě rychlejší, než připojení běžného uživatele. Vzhledem k tomu, že aplikace je webová, tak běží na všech zařízeních, které podporují moderní webový prohlížeč.

K dispozici je dostupná jak placená, tak i neplacená varianta účtu. Mile mě překvapila možnost si vyzkoušet službu bez nutnosti registrace, služba vám vygeneruje náhodné přihlašovací údaje, které si uživatel může zapsat a použít při dalším přihlášení. Tento účet lze poté dodatečně spárovat s emailem.

### 1.3.2 Duplicati 2.0

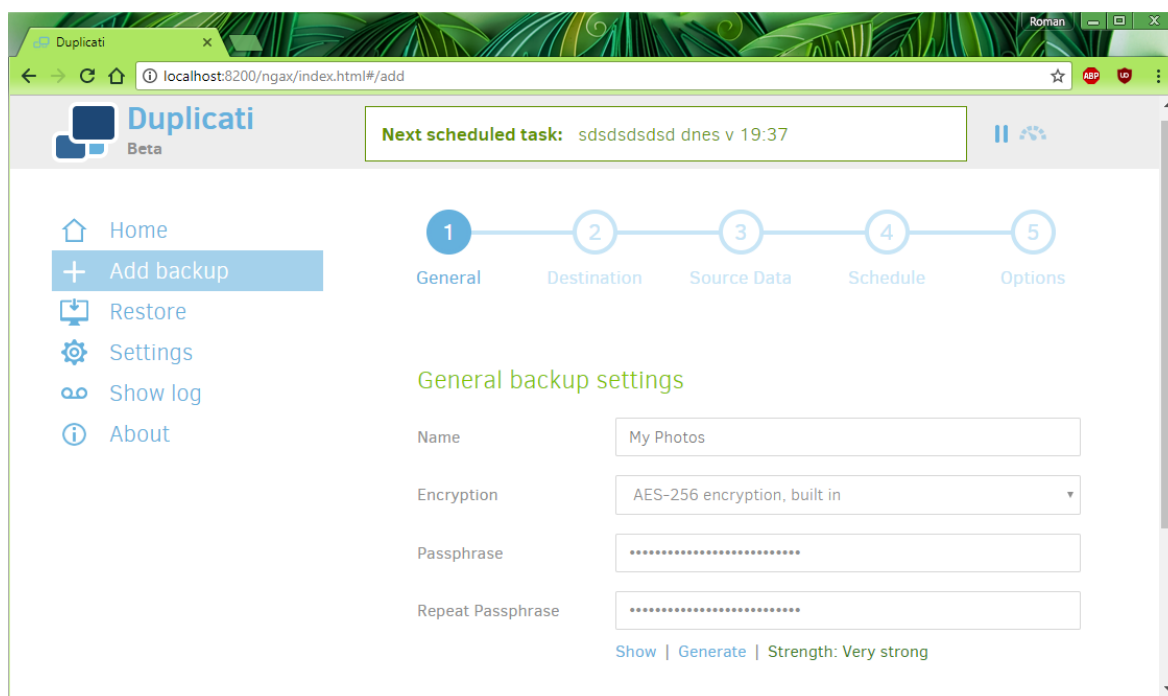
Duplicati je opensourcový zálohovací program, který je možné využít i pro komerční využití. Zálohování je možné provádět na širokou škálu cloudových služeb a podporuje protokoly jako je FTP, SFTP. Aplikace má klasický instalátor, ale uživatelské rozhraní se spouští v defaultním webovém prohlížeči. Na pozadí jinak běží služba aplikace, kterou můžeme najít v task tray.

Aplikace umožňuje vytvořit zálohovací plán, u kterého lze nastavit jednoduchou periodu opakování jako je například:

- Každý 1 den v 13:00, povolené dny: pondělí, pátek
- Každý 1 měsíc v 13:00, povolené dny: pondělí, středa

Oproti konkurenci neumožňuje nastavit rozšířené časové nastavení, jako jsou třeba týdny v měsíci, dny v měsíci, nebo měsíce v roce:

V rámci jednoho zálohovacího plánu můžeme zálohovat zdrojové soubory a složky pouze na jednu cílovou destinaci. Takže pokud bychom chtěli zálohovat data na více úložišť, tak musíme vytvořit stejný počet plánů s duplicitním nastavením kromě cílové cesty.



Obrázek 9 Rozhraní Duplicati

Silnou stránkou Duplicati je podpora šifrování záloh 256 bitovou šifrou AES, ještě než se data odešlou na cílové úložiště. Program také umí detekovat duplicitní soubory v záloze a zálohuje je pouze jednou, ve spojení s kompresí jako je 7z nebo Zip se může ušetřit spousta místa na úložišti. Podporováno je i pokračování v přerušném zálohování, které mohlo nastat z důvodu výpadku internetu, nebo jiných okolností. Verze 2.0 je ve fázi Beta verze, takže se můžeme v budoucnu těšit na rozšířenou funkcionalitu a vyladění možných chyb.[17]

## ANALÝZA TECHNOLOGIÍ

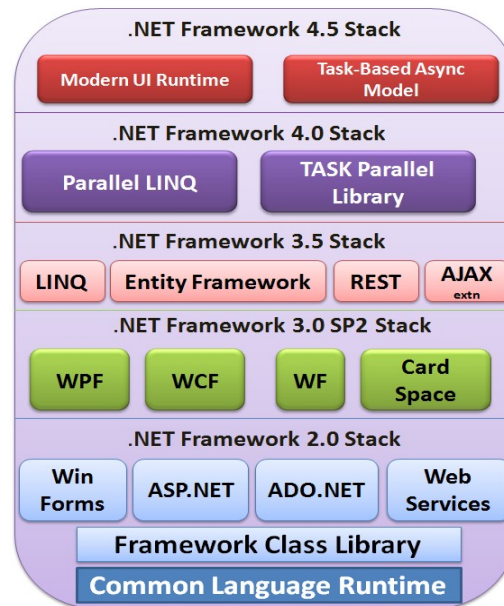
### 1.4 .NET Framework

.NET Framework je softwarová platforma, která je určena pro vývoj a běh různých druhů aplikací. Pomocí této platformy můžeme vyvíjet nejen klasické webové a desktopové aplikace pro Windows, ale i aplikace pro mobilní telefony a služby.

Hlavní částí frameworku je rozsáhlá knihovna tříd a Common Language Runtime, což je běhové prostředí a praktická implementace standardu CLI, který popisuje sadu postupů a specifikací, jak implementovat běhové prostředí a národu generovaného kódu tak, aby umožňoval použití pro více programovacích jazyků na mnoha různých platformách bez nutnosti přepisovat naprogramované aplikace pro různé architektury. Jako další příklad praktické implementace CLI lze uvést Mono CLR, který spadá do opensourcové multiplatformní vývojové platformy Mono.

Výhodou vývoje v .Net Frameworku je rozsáhlost knihoven, relativní jednoduchost a efektivita vývoje. Díky rozsáhlému použití a provázání této platformy s komerční praxí je k dispozici propracovaná dokumentace, spolu s neustálým vývojem nových verzí této platformy.

Nevýhodou této platformy je slabší výkon oproti nativnímu kódu, který je způsoben správou paměti pomocí Garbage Collectoru a tím, že kompilátor překládá výslednou aplikaci do mezikódu (MSIL). Při následném spuštění aplikace výše zmíněný CLR, mezikód interpretuje do binární podoby za běhu aplikace.[8][10]



Obrázek 10 Diagram .Net Frameworku [12]

## 1.5 Windows Presentation Framework

Windows Presentation Framework (WPF) je grafický subsystém .Net Frameworku, který umožňuje programátorovi relativně snadno vytvářet uživatelský interface pro aplikace směřované na operační systém Windows. Mezi hlavní výhody tohoto frameworku patří možnost snadno oddělit definici prezentační logiky od aplikační logiky. Takovéto oddělení má výhodu v tom, že při změnách v aplikační logice se nemusí přepisovat GUI za předpokladu, že interface zůstal nezměněn. Další předností je možnost paralelní práce návrhářů a vývojářů aplikace, tak aby si nezasahovali vzájemně do práce.

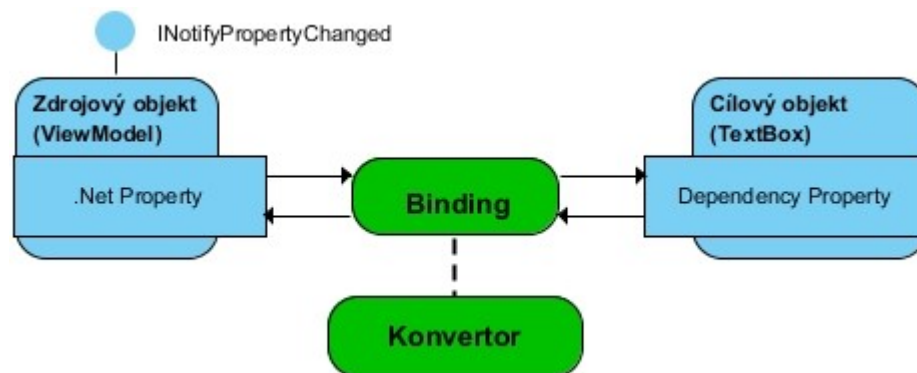
GUI se definuje pomocí značkovacího jazyka XAML, který je založený na XML. Pomocí tohoto jazyka definujeme nejen uživatelské komponenty, ale také jejich styl a chování. Při vývoji ve WPF se doporučuje používat návrhový vzor MVVM, který bude popsán níže.

Součástí WPF je také DataBinding, což je mechanismus, který umožňuje synchronizaci dat mezi datovým zdrojem a GUI. [8][9]

### 1.5.1 DataBinding

Jak již bylo řečeno DataBinding je mechanismus, který umožňuje synchronizovat data mezi GUI (View) a datovým modelem (ViewModel, Model). Předpoklad pro úspěšný DataBinding je ten, že je cílový objekt nastaven jako DataContext komponenty a cílová

vlastnost ve View musí být DependencyProperty. Ve WPF můžeme tento binding specifikovat na cílovou vlastnost vizuálního prvku pomocí syntaxe {Binding}. Tímto způsobem můžeme provést vazbu na jakýkoliv objekt a pokud DataContext implementuje INotifyPropertyChanged, je komponenta notifikována o změnách dat, takže se komponenta dozví, že má například překreslit text “Petr” na “Pavel”. V případě potřeby je možné definovat konvertory, které mohou převádět data na jiný datový typ, popřípadě formát. Tyto konvertory musí implementovat interface IValueConverter, nebo IMultiValueConverter. Díky těmto konvertorům si můžeme snadno přizpůsobovat data pro potřeby View, aniž bychom museli dopisovat funkcionalitu do ViewModelu, nebo View. Například, když zdrojový datový objekt obsahuje čas v Unixovém formátu, ale ve View by jsme ho chtěli zobrazit jako klasické datum, tak si vytvoříme konvertor, který dokáže data mezi těmito formáty převádět.[8][9][11]

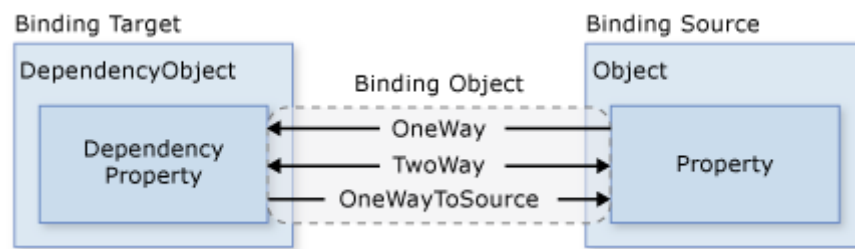


Obrázek 11 Schéma DataBindingu [11]

DataBinding má mnoho parametrů, které ovlivňují jeho chování. Mezi nejdůležitější patří parametr Mode, který určuje směr synchronizace informací. Synchronizační módy jsou následující:

- OneWay- Synchronizace probíhá pouze z datového modelu do View. Tento mód se používá u statických komponent, které pouze zobrazují data, která se mohou na pozadí měnit.
- TwoWay- Obousměrná synchronizace, která se používá u komponent se kterými uživatel nějakým způsobem interaguje a také se stav může změnit na základě nějaké jiné události. Jako příklad si uživatel musí zvolit typ platby.
- OneTime- Stejně chování jako u OneWay, ale synchronizace proběhne jen jednou na začátku. Měl by se používat jako výchozí mód pro data, která se nemění.

- OneWayToSource- Protiklad k OneWay. Pouze View synchronizuje data do datového modelu. [8]



Obrázek 12 Schéma různých typů DataBindingu [8]

### 1.5.2 MVVM

Model-View-ViewModel je návrhový vzor, který má za cíl usnadnit vývojářům rozdělení aplikační vrstvy od prezentační. Při správném použití se i při velmi rozsáhlé aplikaci dají velmi snadno rozšiřovat, modifikovat, či dokonce znovupoužít jednotlivé view.

Jak již z názvu vyplývá, tento vzor se skládá z několika komponent, které jsou od sebe odděleny, a proto umožňují těžit z následujících výhod:

- Změna interní implementace komponenty, aniž by ovlivnila jiné.
- Komponenty jsou na sobě nezávislé.
- Vývojáři mohou na komponentách pracovat paralelně za předpokladu, že je dopředu známý interface komponent.
- Při vytváření unit testů se dají snadno izolovat komponenty.
- Možnost vytvořit více View pro jeden ViewModel a dle potřeby je zaměňovat bez zásahu do ViewModelu.[8]

#### 1.5.2.1 Model

Model reprezentuje data, se kterými pracujeme. Klíčovým prvkem je ten, že model by měl obsahovat pouze informace, ale ne chování a služby, které s těmito informacemi manipulují za účelem zobrazení. Například by model neměl být zodpovědný za formátování textu tak, aby na obrazovce vypadal dobře. [8]

#### 1.5.2.2 ViewModel

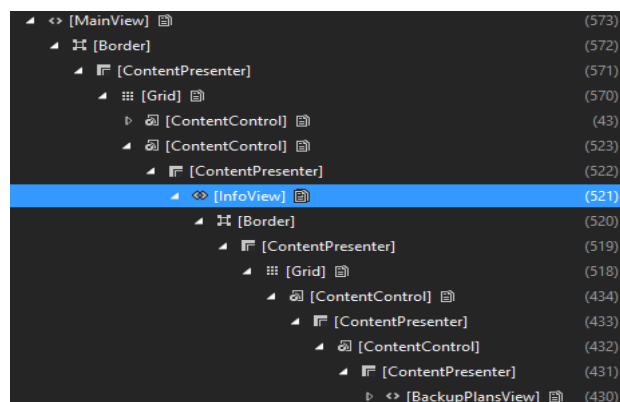
ViewModel slouží jako prostředník mezi View a Modelem. Jeho hlavní zodpovědnost leží v poskytnutí potřebných dat z modelu do View. Ve většině případů ViewModel

implementuje interface `IPropertyChanged`, který obsahuje pouze jediný prvek, a to event `PropertyChanged`, pomocí kterého může View notifikovat o změnách dat, na které View může zareagovat překreslením informace na aktuální hodnotu. V některých případech může ViewModel poskytnout View i samotný Model, za předpokladu, že model implementuje interface `IPropertyChanged`, nebo se data v Modelu již měnit nebudou.

ViewModel se také stará o obsluhu událostí z View, které by se měly správně propagovat přes interface `ICommand`, který můžeme stejně jako vlastnost `DataContext` nabídnout a v případě potřeby i nabídnout parametr ve View, například objekt kontaktu v seznamu na který uživatel vykonal dvojklik. [8]

### 1.5.2.3 View

View je odpovědný za definici vzhledu, struktury a rozložení toho, co se uživateli zobrazuje na obrazovce. V ideálním případě je celý View definovaný v Xamlu, s limitovaným kódem na pozadí, který neobsahuje business logiku aplikace. View si můžeme představit jako stránku, ve stromové struktuře, která v sobě může obsahovat několik dalších stránek. V praxi se View většinou podědí od třídy `UserControl` a poté nadefinuje vizuální struktura v XAMLu.



Obrázek 13 Ukázka vizuální stromové struktury

Jak již bylo zmíněno, data se do View primárně synchronizují pomocí `DataBindingu` a pokud je mód synchronizace nastaven jako `OneWay`, nebo `TwoWay`, tak se ještě odposlouchává událost `PropertyChanged`, kterou by `DataContext` View měl vyvolat při každé změně bindovaných dat. View tyto data aktualizuje automaticky pomocí reflexe v závislosti na nastaveném módu.[8]

## **II. PRAKTICKÁ ČÁST**



## 2 ANALÝZA POŽADAVKŮ A UŽIVATELSKÝCH CÍLŮ NA ZVOLENÉ ŘEŠENÍ

Tato kapitola se věnuje specifikaci požadavků aplikace. V zásadě se jedná o stručné analytické rozpracování požadavků, cílů a kritérií úspěšnosti implementace prototypu aplikace.

### 2.1 Cíle projektu

Hlavním cílem tohoto projektu je vyvinout aplikaci, které umožní uživateli zálohovat vybraná data na různá úložiště, bez ohledu na to, jestli je umístěno na cloudu, lokálním úložišti, nebo úložišti jiného druhu. Aplikace by také měla mít mód zálohování, který umožní automaticky rovnoměrně rozdělovat zálohované data na více úložišť v závislosti na jejich vytížení a celkové kapacitě.

Toto zadání můžeme rozložit na několik dílčích cílů:

- Implementovat proces zálohování.
- Implementovat rozhraní tak, aby se daly snadno doprogramovat další typy úložišť.
- Implementovat logiku aplikace tak, aby nezatěžovala systém.
- Implementovat rozhraní pro uživatele.
- Implementovat stabilní a spolehlivou aplikaci.
- Implementovat mód rovnoměrného rozdělování dat, podle vytížení jednotlivých úložišť.

Hlavní cílová skupina, která bude tuto aplikaci používat, je uživatel bez větších technických znalostí a proto by rozhraní aplikace mělo být co nejvíce intuitivní vzhledem k požadavkům, které by uživatel od aplikace očekával.

### 2.2 Kritéria úspěšnosti

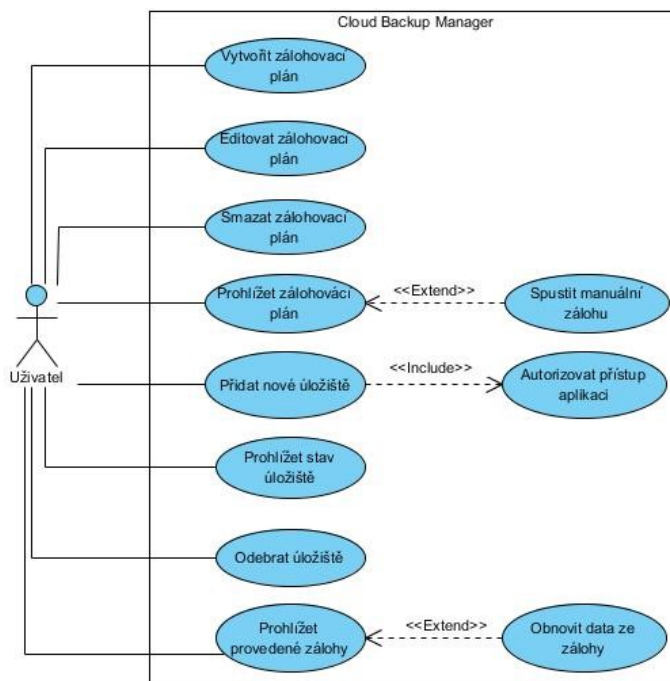
V rámci základní specifikace projektu je vhodné stanovit také kritéria úspěšnosti celého projektu. Tato kritéria neslouží pouze pro samotné zhodnocení výsledného řešení projektu, ale slouží i jako významná podpůrná informace pro praktický návrh a implementaci projektu, jelikož je z nich možné pochopit, jak uživatel nahlíží na celý projekt a jaké funkční požadavky jsou pro něj více, či méně důležité.

Základní kritéria úspěšnosti projektu jsou stanovena následovně:

- Aplikace by měla běžet na pozadí a nevyrušovat uživatele pokud to nebude nutné pro úspěšné provedení zálohy.
- Aplikace by měla být stabilní a uživatel by měl mít možnost dohledat chybové stavy, které aplikace nemůže ovlivnit (výpadek internetu při běhu zálohy atd.).
- Uživatelský interface by měl být jednoduchý na pochopení a umožnit uživateli snadnou navigaci.
- Propojení úložišť, by mělo být zpracováno ve formě pluginů, tak aby se daly snadno dopisovat další typy možných úložišť.
- Vzhledem k tomu, že by aplikace měla jít spustit i na slabších počítačích, neměla by příliš zatěžovat systémové prostředky.

## 2.3 Případy užití

V rámci práce byly navrženy případy užití, které definují funkcionalitu, kterou by měla aplikace umět.



Zdrojový kód 1 Use case diagram aplikace

### 2.3.1 UC01: Vytvořit zálohovací plán

#### *Aktéři*

Uživatel.

**Podmínky**

Uživatel inicializuje obrazovku pro vytvoření nového zálohovacího plánu.

**Podmínky pro dokončení**

Aplikace uložila zálohovací plán.

Uživatel uvidí zálohovací plán v seznamu zálohovacích plánů na hlavní obrazovce aplikace.

**Základní tok**

1. Uživatel inicializuje obrazovku kliknutím na tlačítko Vytvořit zálohovací plán
2. Aplikace zobrazí obrazovku pro vytvoření zálohovacího plánu
3. Na záložce Backup Plan Setting uživatel vyplní základní údaje (Jméno zálohovacího plánu, typ zálohování).
4. Uživatel přejde na záložku Backup Path Settings.
5. Uživatel si v sekci Files and folders to backup zvolí soubory a složky, které chce v rámci zálohovacího plánu zálohovat.
6. Uživatel si dále v sekci Backup folders zvolí poskytovatele zálohování a klikne na tlačítko Add folder.
7. Aplikace zobrazí prohlížeč úložiště, ve kterém je možné zvolit složku, kam se budou data zálohovat.
8. Uživatel si zvolí cílovou složku.
9. Aplikace přidá cílovou složku do seznamu v sekci Backup folders.
10. Uživatel opakuje akci 6. – 8. do doby, než jsou všechny požadované destinace v seznamu.
11. Uživatel si zvolí, zda chce použít režim rovnoměrného rozdělování dat.
12. Uživatel přejde na záložku Time Settings.
13. Uživatel zvolí požadované časové nastavení a doplní všechny požadované údaje.
14. Uživatel uloží zálohovací plán.
15. Aplikace přejde na hlavní obrazovku.

**Alternativní tok**

14A1: Uživatelem zadaná nastavení jsou neplatná, nebo je zadáno duplicitní jméno zálohovacího plánu.

1. Aplikace informuje uživatele o tom, které údaje jsou nesprávné.
2. Uživatel doplní/upraví údaje.

3. Příklad užití pokračuje v kroku 14. základního toku.

### **2.3.2 UC02: Editovat zálohovací plán**

#### ***Akteři***

Uživatel

#### ***Podmínky***

Uživatel inicializuje obrazovku pro editaci zálohovacího plánu.

#### ***Podmínky pro dokončení***

Aplikace uložila změny zálohovacího plánu.

#### ***Základní tok***

1. Uživatel inicializuje obrazovku pro editaci zálohovacího plánu.
2. Aplikace zobrazí obrazovku pro editaci zálohovacího plánu.
3. Uživatel upraví údaje a nastavení zálohovacího plánu.
4. Uživatel uloží zálohovací plán.
5. Aplikace přejde na hlavní obrazovku.

#### ***Alternativní tok***

4A1: Uživatelem zadaná nastavení jsou neplatná, nebo je zadáno duplicitní jméno zálohovacího plánu.

1. Aplikace informuje uživatele o tom, které údaje jsou nesprávné.
2. Uživatel doplní/upraví požadované nastavení.
3. Příklad užití pokračuje v kroku 4. základního toku.

### **2.3.3 UC03: Smazat zálohovací plán**

#### ***Akteři***

Uživatel

DropBox

#### ***Podmínky***

Uživatel se nachází na hlavní obrazovce a má zobrazený seznam zálohovacích plánů.

#### ***Podmínky pro dokončení***

Aplikace smazala zálohovací plán a všechna data na něj navázaná, kromě samotných záloh, které zůstanou zachovány.

#### ***Základní tok***

1. Uživatel si v seznamu zálohovacích plánů zvolí zálohovací plán, který si přeje smazat.
2. Uživatel klikne u příslušného plánu na tlačítko smazat.
3. Aplikace se uživatele dotáže, zda chce zvolený zálohovací plán smazat.
4. Uživatel akci potvrdí.
5. Aplikace smaže zálohovací plán a všechna data, která k němu patří. Zálohovaná data na úložištích zůstanou zachována.
6. Zálohovací plán se nenávratně odebere ze seznamu.

#### ***Alternativní tok***

4A1: Uživatel si smazání zálohovacího plánu rozmyslí.

1. Uživatel akci nepotvrdí kliknutím na tlačítko No.
2. Zálohovací plán zůstane zachován.
3. Konec alternativního toku.

### **2.3.4 UC04: Přidat nové úložiště**

#### ***Akteři***

Uživatel

#### ***Podmínky***

Aplikace má inicializované pluginy pro úložiště.

Uživatel inicializuje obrazovku pro přidání nového úložiště.

#### ***Podmínky pro dokončení***

Aplikace obdržela verifikaci přidělení oprávnění ze strany úložiště.

#### ***Základní tok***

1. Uživatel inicializuje obrazovku pro přidání nového úložiště.
2. Aplikace zobrazí průvodce přidání nového úložiště.
3. Uživatel pojmenuje úložiště
4. Uživatel si zvolí, jaký typ úložiště si přeje přidat.
5. Uživatel pokračuje kliknutím na tlačítko Další.

6. Uživatel autorizuje aplikaci přístup k úložišti, v závislosti na tom jaký typ úložiště si přeje přidat.
7. Aplikace obdrží autorizační údaje.
8. Aplikace přejde na hlavní obrazovku.

#### **Alternativní tok**

5A1: Uživateli se nepodaří autorizace, nebo ji stornuje.

Aplikace se navrátí do bodu 4 hlavního toku.

### **2.3.5 UC05: Autorizovat přístup (DropBox)**

#### ***Aktéři***

Uživatel

Služba DropBox

#### ***Podmínky***

Tok se nachází v bodu 6 hlavního toku UC04.

Uživatel má založený účet ve službě DropBox.

#### ***Podmínky pro dokončení***

Uživatel se úspěšně přihlásí.

Uživatel přidělí aplikaci požadované oprávnění.

Aplikace obdržela verifikaci přidělení oprávnění ze strany úložiště.

#### ***Základní tok***

1. Aplikace zobrazí webový přihlašovací formulář pro službu DropBox.
2. Uživatel se přihlásí.
3. Uživateli se zobrazí dialog, kde jsou vypsána požadovaná oprávnění k přístupu na úložiště.
4. Uživatel oprávnění potvrdí.
5. Služba DropBox přidělí oprávnění aplikaci.
6. Aplikace obdrží přihlašovací token od služby DropBox.
7. Aplikace s přihlašovacím tokenem pokračuje do bodu 8 základního toku UC04

#### **Alternativní tok**

2A1: Uživateli se nepodaří přihlášení, nebo přihlášení stornuje.

1. Tok se navrátí do bodu 4 základního toku UC04.

4A1: Uživatel aplikaci nechce přidělit požadované oprávnění.

1. Uživatel zavře webový formulář.
2. Aplikace se navrátí do bodu 4 hlavního toku UC04.

### **2.3.6 UC06: Autorizovat přístup (GDrive)**

#### ***Akteři***

Uživatel

Služba Google

#### ***Podmínky***

Tok se nachází v bodu 6 hlavního toku UC04.

Uživatel má založený účet ve službě Google.

#### ***Podmínky pro dokončení***

Uživatel se úspěšně přihlásí.

Uživatel přidělí aplikaci požadované oprávnění.

Aplikace obdržela verifikaci přidělení oprávnění ze strany úložiště.

#### ***Základní tok***

1. Aplikace zobrazí webový přihlašovací formulář pro službu DropBox.
2. Uživatel se přihlásí.
3. Uživateli se zobrazí dialog, kde jsou vypsána požadovaná oprávnění k přístupu na úložiště.
4. Uživatel oprávnění potvrdí.
5. Služba Google přidělí oprávnění aplikaci.
6. Aplikace obdrží přihlašovací token od služby Google.
7. Aplikace s přihlašovacím tokenem pokračuje do bodu 8 základního toku UC04

#### **Alternativní tok**

2A1: Uživateli se nepodaří přihlášení, nebo přihlášení stornuje.

1. Aplikace se navrátí do bodu 4 základního toku UC04.

4A1: Uživatel aplikaci nechce přidělit požadované oprávnění.

1. Uživatel zavře webový formulář.
2. Aplikace se navrátí do bodu 4 základního toku UC04.

### **2.3.7 UC06: Prohlížet vytížení úložišť**

#### ***Aktéři***

Uživatel

#### ***Podmínky***

Uživatel inicializuje obrazovku přehled úložišť.

#### ***Podmínky pro dokončení***

Aplikace obsahuje alespoň jedno úložiště.

#### ***Základní tok***

1. Uživatel inicializuje obrazovku přehled úložišť.
2. Aplikace zobrazí obrazovku se seznamem všech dostupných úložišť.
3. Aplikace aktualizuje aktuální vytížení pro všechna úložiště.
4. Uživatel prohlíží vytížení jednotlivých úložišť.

#### ***Alternativní tok***

2A1: Aplikaci se nepodaří získat aktuální stavy vytížení pro některá, nebo všechny úložiště.

1. Aplikace u daných položek ponechá poslední známé vytížení a zobrazí příznak, který dává uživateli najevo, že zobrazený stav není aktuální.
2. Tok se navrátí do bodu 4 základního toku.

### **2.3.8 UC07: Prohlížet zálohy**

#### ***Aktéři***

Uživatel

#### ***Podmínky***

Uživatel inicializuje obrazovku zálohy.

#### ***Podmínky pro dokončení***

Uživatel má vybraný zálohovací plán.



Existuje alespoň jedna záloha pro vybraný zálohovací plán.

#### ***Základní tok***

1. Uživatel si zvolí zálohovací plán.
2. Uživatel inicializuje obrazovku přehled zálohovacích plánů.
3. Aplikace zobrazí obrazovku s přehledem zálohovacích plánů.
4. Aplikace načte všechny provedené zálohy pro zvolený zálohovací plán.
5. Aplikace seřadí zálohy sestupně podle data uskutečnění zálohy.
6. Aplikace zobrazí seřazené zálohy v seznamu.
7. Uživatel prohlíží zálohy

### **2.3.9 UC08: Obnovit data ze zálohy**

#### ***Akteři***

Uživatel

#### ***Podmínky***

Uživatel inicializuje obrazovku přehled úložišť.

#### ***Podmínky pro dokončení***

Aplikace obsahuje alespoň jedno úložiště.

#### ***Základní tok***

5. Uživatel inicializuje obrazovku přehled úložišť.
6. Aplikace zobrazí obrazovku se seznamem všech dostupných úložišť.
7. Aplikace aktualizuje aktuální vytížení pro všechna úložiště.
8. Uživatel prohlíží vytížení jednotlivých úložišť.

#### **Alternativní tok**

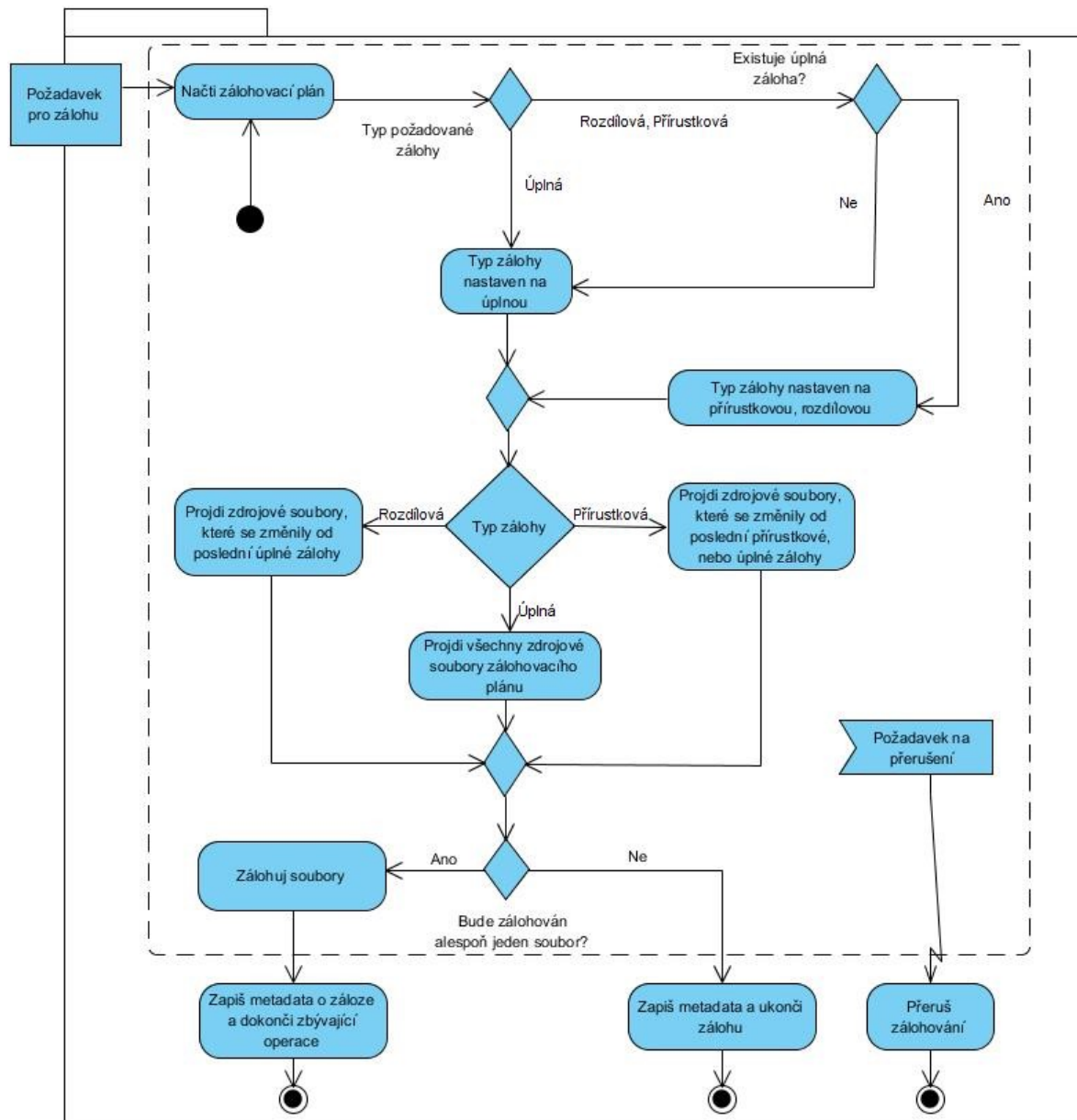
2A1: Aplikaci se nepodaří získat aktuální stavy vytížení pro některá, nebo všechny úložiště

Aplikace u daných položek ponechá poslední známé vytížení a zobrazí příznak, který dává uživateli najevo, že zobrazený stav není aktuální.

Tok se navrátí do bodu 4 základního toku.

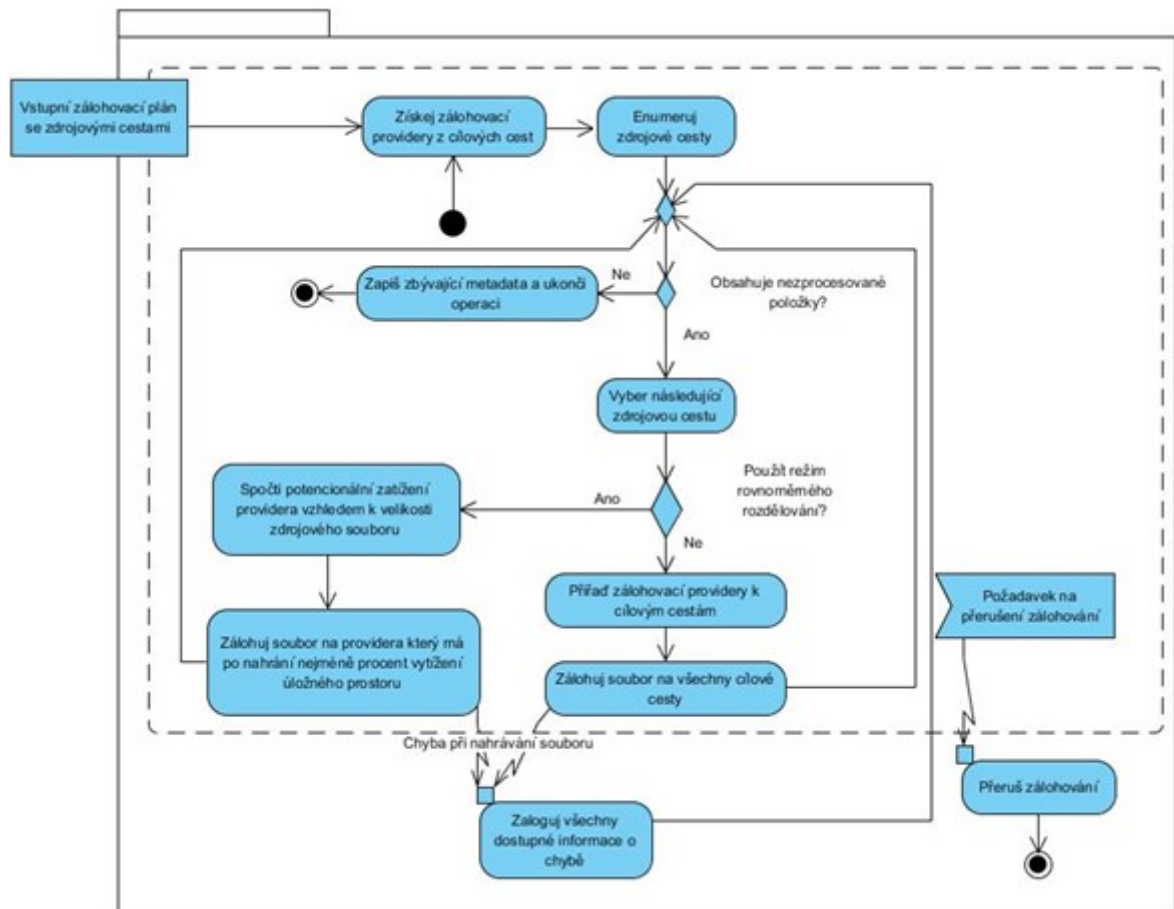
### 2.3.10 Algoritmus zálohování

Byl také navrhnout algoritmus zálohování, který je s drobnými úpravami prakticky použit v aplikaci. Ze všeho nejdříve byl navrhnout obecný aktivitní diagram algoritmu pro zpracování zálohovacího plánu a jeho zálohy.



Obrázek 14 Aktivitní diagram obecného procesu zálohování

V další fázi byl navrhnout zpřesňující diagram pro samotný proces zálohování, který rozšiřuje operaci Zálohuj soubory, z předchozího aktivitního diagramu. Je zde také vyobrazen algoritmus rovnoměrného rozdělování dat.



Obrázek 15 Aktivitní diagram zálohování

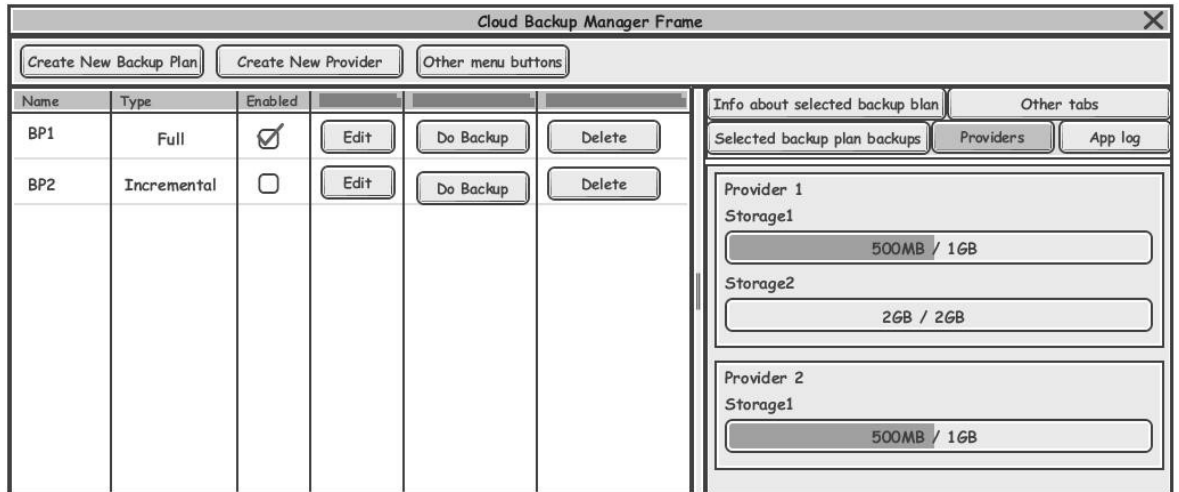
## 2.4 Wireframe hlavního okna aplikace

Před tvorbou uživatelského rohraní byla navrhnutá skica hlavního okna aplikace, která definuje rozložení funkčních prvků. Okno bylo vizuálně rozděleno na tři části.

První částí je horní menu, kde se nachází hlavní funkční tlačítka. Dále bylo počítáno s tím, že si do menu mohou pluginy přidat vlastní ovládací prvky.

V levé části hlavního okna se nachází seznam vytvořených zálohovacích plánů, který uživateli zobrazuje základní informace o všech plánech a ovládací prvky pro editaci/smazání/spuštění zálohy.

V pravé části se nachází TabControl, do kterého lze dynamicky přidávat záložky ze všech modulů a pluginů. Součástí byl i počáteční návrh možných záložek a návrh seznamu úložišť.



Obrázek 16 Wireframe hlavního okna aplikace

## 3 SOFTWAREVÉ ŘEŠENÍ

### 3.1 Popis struktury aplikace

Aplikace je napsána modulárně a skládá se z mnoha projektů. Pro tento modulární přístup byl použit framework PRISM, který k tomuto přístupu tvorby aplikace přímo nabádá[13]. Hlavní částí je projekt CloudBackupManager.Shell, který obsahuje hlavní okno aplikace a třídu Bootstraper, jejíž instance se vytvoří ihned po startu aplikace. Bootstraper inicializuje základní služby a části, jako je UnityContainer, RegionManager a EventAggregator, spolu se všemi moduly, které implementují interface IModule a jsou zdefinované v katalogu modulů.

Aplikace se konkrétně skládá z následujících modulů:

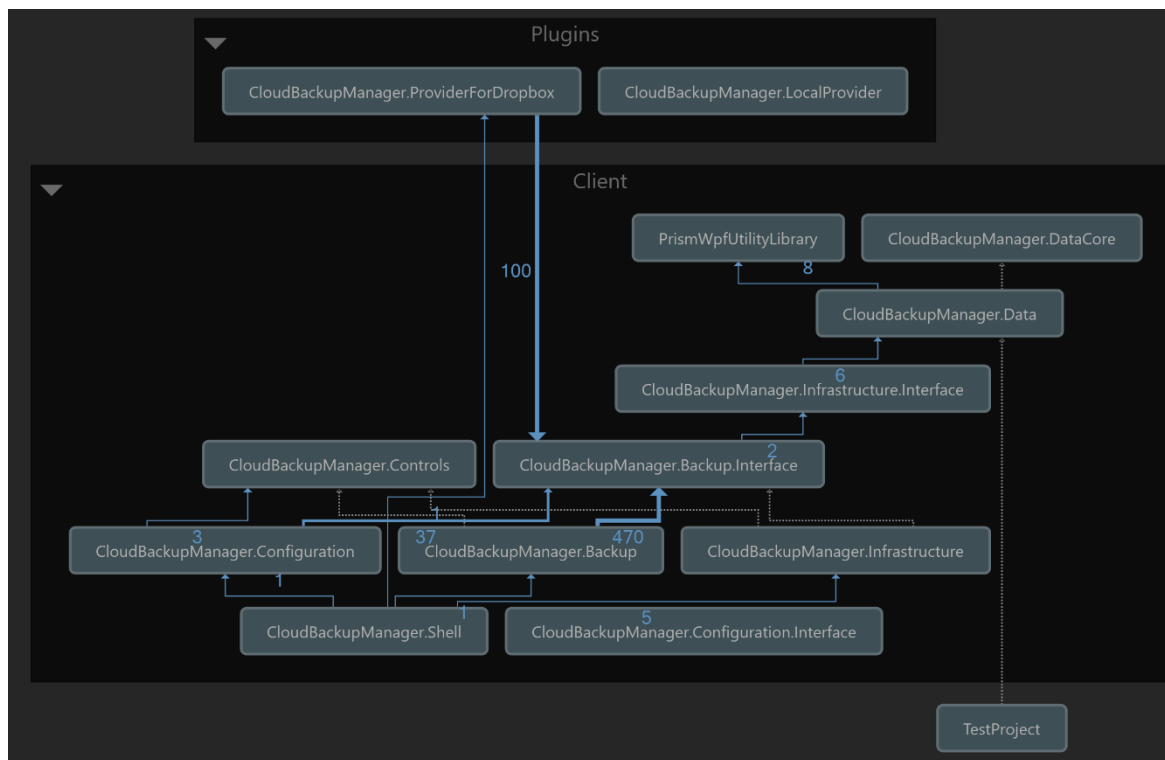
- CloudBackupManager.Backup
- CloudBackupManager.Configuration
- CloudBackupManager.Controls
- CloudBackupManager.Data
- CloudBackupManager.DataCore
- CloudBackupManager.Infrastructure
- CloudBackupManager.Shell
- CloudBackupManager.ProviderForDropbox

Součástí jsou i pomocné knihovny .Interface, ve kterých se nachází pouze rozhraní a třídy potřebné pro práci s danými rozhraními modulů.

- CloudBackupManager.Backup.Interface
- CloudBackupManager.Configuration.Interface
- CloudBackupManager.Infrastructure.Interface

Výhodou takového rozdělení je:

- Obrana proti cyklickým referencím.
- Moduly jsou na sobě navzájem nezávislé.
- Možnost kompletní záměny modulů, za předpokladu, že budou správně implementovány patřičné rozhraní a načteny všechny potřebné služby do DI kontejneru.



Obrázek 17 Diagram vztahů projektů (modulů)

### 3.1.1 Projekt CloudBackupManager.Shell

V tomto projektu se nachází hlavní okno aplikace a třída Bootstraper, který se stará o inicializaci všech potřebných modulů a služeb aplikace.

### 3.1.2 Projekty CloudBackupManager.DataCore a CloudBackupManager.Data

Projekt CloudBackupManager.DataCore obsahuje třídy potřebné pro správu databáze aplikace. Projekt s databází pracuje pomocí Entity Frameworku, což je takzvané ORM (objektově relační mapování). Pro databázi byla zvolena databáze SQL Server Compact, která pro svůj chod nepotřebuje na cílovém počítači nic instalovat, potřebné knihovny pro práci s touto databází si aplikace nese s sebou.

Struktura databáze se vytváří přístupem Code First, kdy se nejdříve nadefinují datové třídy a poté se pomocí speciálního příkazu vytvoří migrace databáze. Při každé migraci se zanesou rozdíly od posledního stavu. Díky tomuto přístupu si aplikace dokáže sama aktualizovat starší databázi na novější verzi, což umožňuje snadnou distribuci novějších verzí bez obavy ztráty, nebo nekompatibility existujících dat. Ostatní moduly mohou poté s touto databází komunikovat přes třídu CloudBackupManagerContext, kterou tento projekt obsahuje.

Datové modely tabulek databáze se ihned převádí do klientských datových objektů a používají se pouze pro ukládání a načítání dat. Tento přístup umožňuje v budoucnu snadnější přechod na jiné databáze.

---

```
namespace CloudBackupManager.DataCore.Models
{
    public class BackupPlan
    {
        [Key]
        public int Id { get; set; }
        public string Name { get; set; }
        public int? TimePlanId { get; set; }
        public bool Enabled { get; set; }
        [ForeignKey("TimePlanId")]
        public TimePlan TimePlan { get; set; }
        public int BackupPlanType { get; set; }
        public byte[] SourcePaths { get; set; }
        public byte[] TargetPaths { get; set; }
        public int? BackupChainLength { get; set; }
    }
}

namespace CloudBackupManager.Data.DataObjects
{
    [Serializable]
    public class BackupPlan : ICopyable<BackupPlan>
    {
        public BackupPlanType BackupPlanType { get; set; }
        public TimePlan TimePlan { get; set; }
        public int Id { get; set; }
        public string Name { get; set; }
        public List<PathElement> SourcePaths { get; set; }
        public List<PathElement> TargetPaths { get; set; }
        public bool Enabled { get; set; }
        public int? BackupChainLength { get; set; }
        . . .
    }
}
```

---

Zdrojový kód 2 Ukázka databázové a klientské třídy

### 3.1.3 Projekt CloudBackupManager.Infrastructure

V tomto projektu se nachází základní služby pro práci s menu a zálohovacími plány. Struktura třídy, která reprezentuje zálohovací plán, je uvedena výše. Tato třída také obsahuje časový plán, který v sobě nese všechny potřebné informace o periodě opakování zálohy. Správu zálohovacích plánů má na starosti singleton, služba BackupPlanService, kterou si mohou ostatní moduly vyžádat přes Unity IoC Container, do kterého je služba při inicializaci modulu vložena pod interfacem IBackupPlanService.

---

```
public interface IBackupPlanService : INotifyPropertyChanged
{
    IEnumerable<BackupPlan> GetBackupPlans();
    BackupPlan GetBackupPlan(int id);
    bool EditBackupPlan(BackupPlan backupPlan);
    bool DeleteBackupPlan(BackupPlan backupPlan);
    event PropertyChangedEventHandler PropertyChanged;
    BackupPlan SelectedBackupPlan { get; set; }
    BackupPlan GetNewBackupPlan();
    event EventHandler BackupPlansChanged;
}
```

---

Zdrojový kód 3 IBackupPlanService

### 3.1.4 Projekt CloudBackupManager.Backup

V tomto projektu se nachází služby, které mají na starosti správu zálohovacích providerů a samotný proces zálohování. Služba BackupProviderManager spravuje zálohovací providery, které mohou pluginy za běhu aplikace vkládat.

Služba BackupService obsahuje logiku pro samotné zálohování a ukládání metadat o zpracovaných operacích. Stejně jako u ostatních služeb je dostupná pro všechny moduly přes IoC Container pod interfacem IBackupService. Služba v krátkých časových intervalech kontroluje všechny aktivní zálohovací plány a kontroluje jejich časové nastavení. Pokud je čas pro zálohu, služba zařadí zálohovací plán do fronty, která se postupně odbavuje.

---

```
public interface IBackupService
{
    IReadOnlyNotifyCollection<OperationTaskSource> BackupQueue { get; }
    void AddToQueue(int backupPlanId);
    bool EditBackup(CloudBackupManeger.Data.DataObjects.Backup backup);
    bool DeleteBackup(CloudBackupManeger.Data.DataObjects.Backup backup);
    IEnumerable<CloudBackupManeger.Data.DataObjects.Backup> GetBackups();
    IEnumerable<CloudBackupManeger.Data.DataObjects.Backup> GetBackups(int
backupPlanId);
    CloudBackupManeger.Data.DataObjects.Backup GetNewBackup();
    DateTime? CalculateNextBackupTime(TimePlan timePlan);
    void AddToRestoreQueue(int backupId);
}
```

---

Zdrojový kód 4 IBackupService

### 3.1.5 Projekt CloudBackupManager.ProviderForDropbox

Tento projekt slouží jako plugin, který se inicializuje při startu aplikace. Obsahuje inicializaci tříd, které jsou nutné pro komunikaci s cloudovým úložištěm DropBox.



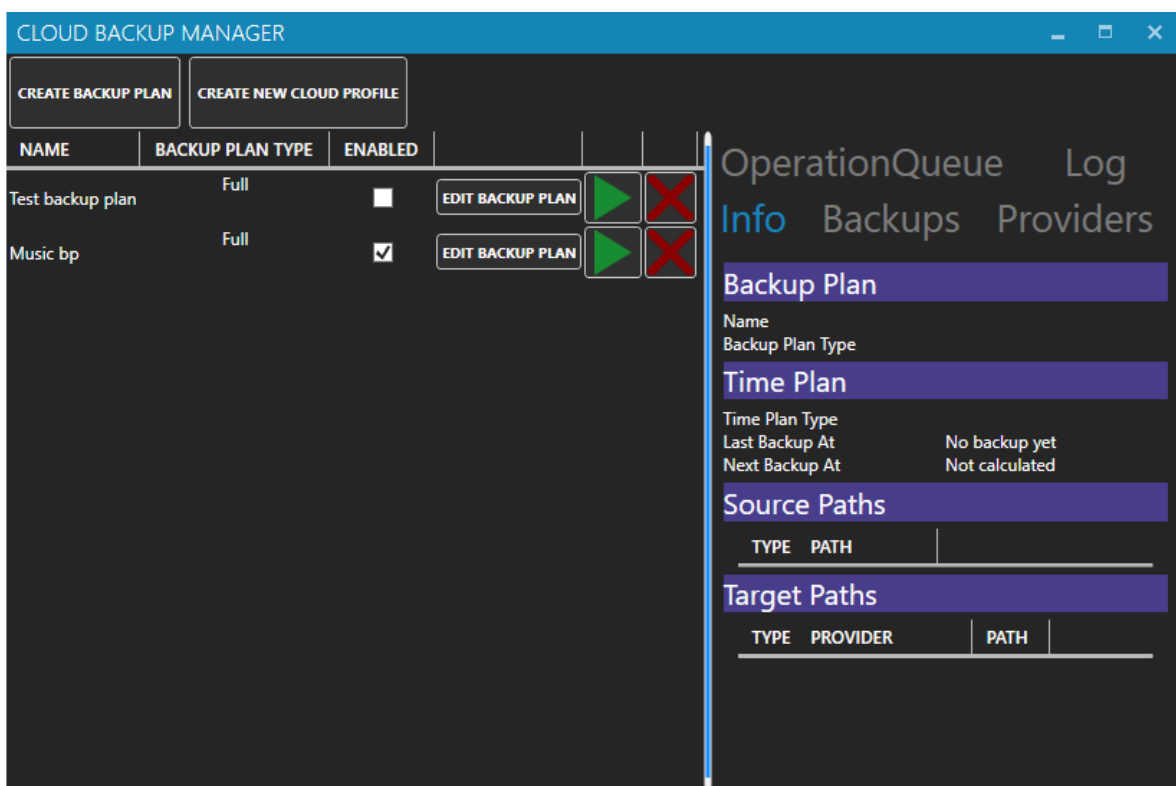
### **3.1.6 Projekt CloudBackupManager.ProviderForGDrive**

Tento projekt také slouží jako plugin, který se inicializuje při startu aplikace. Obsahuje třídy nutné pro komunikaci s cloudovým úložiště Google Drive.

## 4 POPIS APLIKACE

V rámci této sekce bude popsáno uživatelské rozhraní s jejich jednotlivými náhledy. Budou také popsány způsoby implementace jednotlivých prvků a příklady užití. V následujících kapitolách budou popsány konkrétně tyto View:

1. BackupPlanDetailView – Detail zálohovacího plánu.
2. MainView – View, které je umístěné v rootu okna aplikace a skládá se z několika dalších View.
3. NewProviderDialogView – View pro přidání nového zálohovacího providera.



Obrázek 18 Hlavní okno aplikace

### 4.1 BackupPlanDetailView

Tento dialog slouží pro editaci zálohovacího plánu. Při kliknutí na volbu Create Backup Plan se vytvoří nový zálohovací plán, který se nachází pouze v paměti do doby, než uživatel klikne na tlačítko save, čímž se záznam uloží do databáze a stává se platným. Při editaci již existujícího zálohovacího plánu se položka v databázi přepíše.

Dialog obsahuje několik záložek:

1. Backup plan settings

2. Backup paths settings

3. Time settings

#### 4.1.1 Backup Plan Settings

V této záložce se nachází základní volby pro zálohovací plán, kde si uživatel může nastavit základní chování. Všechna možná nastavení této záložky budou popsána v následující tabulce.

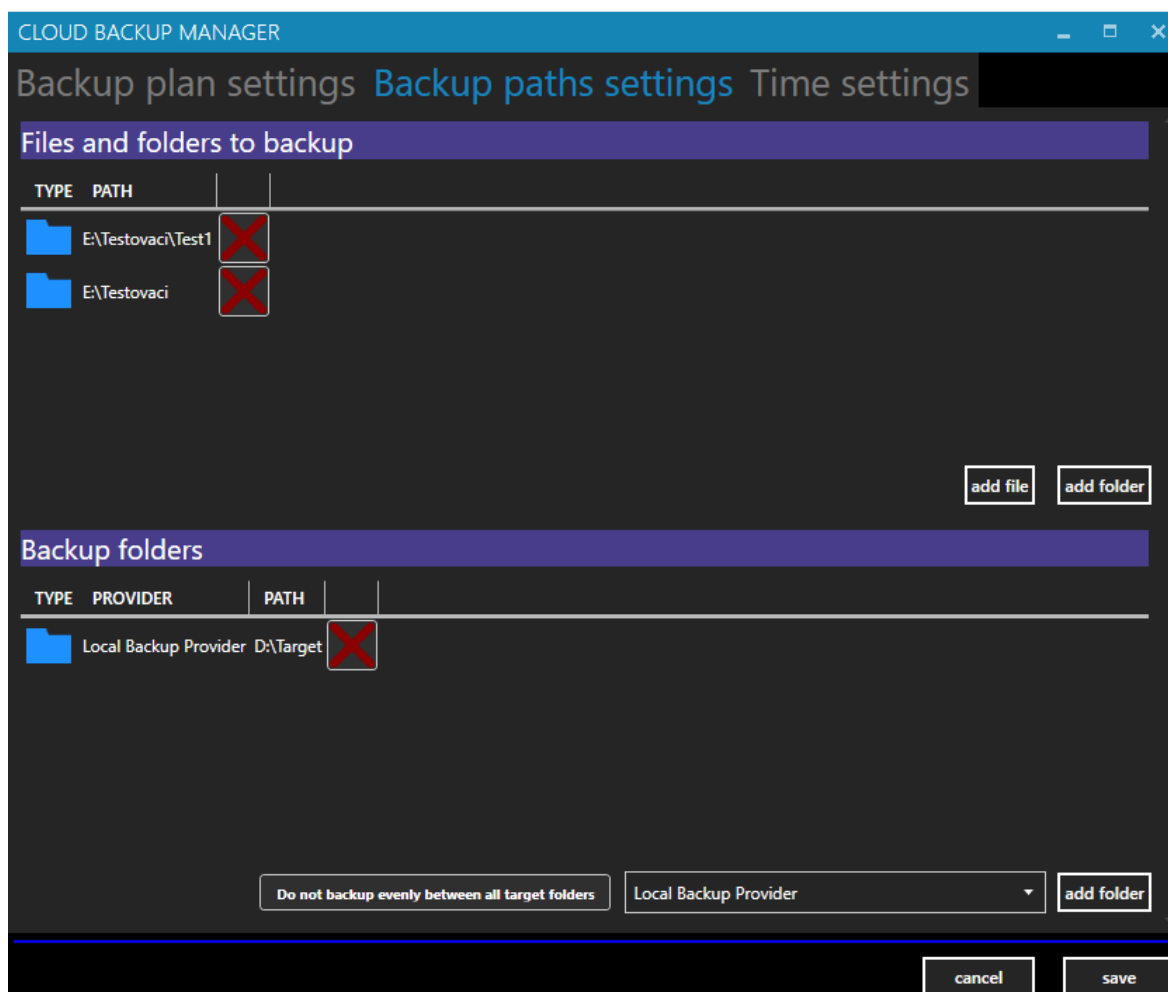
Tabulka 5 Popis nastavení záložky Backup plan settings

Název	Typ kontrolky	Volba	Popis
<b>Name</b>	TextBox		Zde si uživatel libovolně pojmenuje zálohovací plán.
<b>Backup Type</b>	ComboBox		Možnost volby typu zálohování.
		Full	Úplná záloha
		Incremental	Inkrementální záloha
		Differential	Diferenciální záloha
<b>Backup chain length</b>	NumericUpDown		Nastavení je možné zvolit pouze, pokud je typ zálohovacího plánu nastavený na Incremental, nebo Differential. Určuje po jakém počtu záloh se vytvoří záloha úplná.
<b>Status</b>	ToggleButton	Enabled/Disabled	Možnost aktivace a deaktivace plánu. U deaktivovaného plánu je možné provést zálohu pouze manuálně.

#### 4.1.2 Backup paths settings

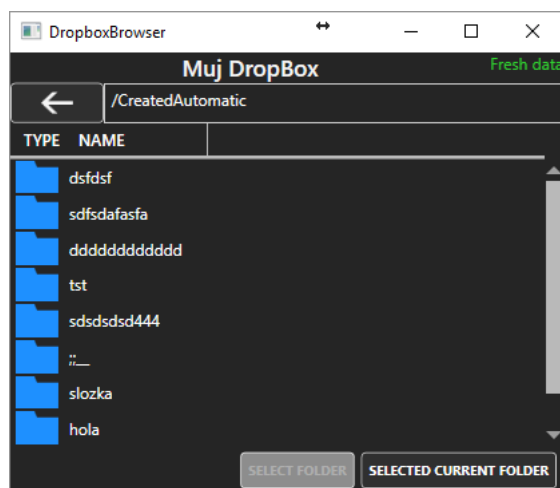
V této záložce se nachází 2 sekce. V první sekci „Files and folder to backup“ si uživatel zvolí soubory a složky, které chce zálohovat. Uživatel může zálohovat pouze data, ke kterým

může přistoupit přes operační systém. V další sekci „Backup folders“, si může uživatel zvolit, kam se budou zálohovaná data kopírovat.



Obrázek 19 Backup path settings

Pro cílové destinace si musí uživatel zvolit takzvaného poskytovatele služby(providera), z comboboxu, který se nachází vedle tlačítka add folder. Po volbě providera a kliknutím na tlačítko add folder, se uživateli zvolí patřičný dialog pro výběr složky. Dialogy se mohou lišit v závislosti na tom, jaký provider je zvolen. Při výběru lokálního providera se například zobrazí nativní dialog pro výběr složky. Při výběru providera pro DropBox se zobrazí dialog, který byl navržen, a implementován speciálně pro službu DropBox. Uživatel si zde ještě může zvolit, zda chce použít režim rovnoměrného rozdělování kliknutím na tlačítko, které se nachází nalevo od seznamu providerů.



Obrázek 20 Dialog pro výběr složky v cloudovém úložišti DropBox.

#### 4.1.3 Time Settings

Každý zálohovací plán má na sebe navázaný časový plán, který reprezentuje způsob periodického zálohování. Ze všeho nejdříve si musí uživatel zvolit typ požadovaného zálohovacího plánu, který zpřístupní ostatní volby, které jsou závislé na zvoleném typu. Jediný typ, který nemá dodatečné volby je Manuální typ, kde se záloha provádí na podnět uživatele, z přehledu zálohovacích plánů. Veškeré volby u jednotlivých typů budou popsány v následující tabulce.

Tabulka 6 Tabulka nastavení Time Settings

Typ časového plánu	Volba	Popis
<b>Daily</b>		Denní časový plán.
	Time	Čas, ve kterém se bude záloha opakovat.
<b>Weekly</b>		Týdenní časový plán.
	Time	Čas, ve kterém se bude záloha opakovat.
	Days in week	Dny v týdnu, kdy se bude záloha provádět.
<b>Monthly</b>		Měsíční zálohovací plán.
	Time	Čas, ve kterém se bude záloha opakovat.

	Days in month	Dny v měsíci, kdy se bude záloha provádět. Vzhledem k tomu, že jednotlivé měsíce mají různou délku, tak je dostupná i volba Last, pro poslední den v jakémkoliv měsíci.
<b>Yearly</b>		Roční zálohovací plán.
	Time	Čas, ve kterém se bude záloha opakovat.
	Months in year	Měsíce v roce, kdy se bude záloha provádět.
	Days in month	Dny v měsíci, kdy se bude záloha provádět.
<b>Manual</b>		Manuální časový plán, uživatel si musí manuálně vyžádat zálohu v programu, přes tlačítko v přehledu zálohovacích plánů.

## 4.2 MainView

Toto view se skládá z menu a dvou sekcí. Menu obsahuje dvě tlačítka.

1. Create Backup Plan – Vyvolá dialog pro vytvoření nového zálohovacího plánu.
2. Create New Cloud Profile – Vyvolá dialog pro vytvoření nového cloudového profilu.

V levé části se nachází BackupPlansView, které obsahuje seznam všech vytvořených zálohovacích plánů. Řádek každého zálohovacího plánu obsahuje základní informace a tři tlačítka.

1. Edit Backup Plan – Otevře dialog pro editaci zvoleného plánu.
2. Force Backup – Zařadí zvolený plán do fronty záloh. Toto tlačítko vynutí zálohu, v případě potřeby a je to jediný způsob jak spustit zálohovací plán, který má nastavený manuální typ časového plánu.
3. Delete Backup – Slouží pro odstranění zálohovacího plánu. Před samotným smazáním se ještě dotáže, zda uživatel chce akci opravdu provést.

NAME	BACKUP PLAN TYPE	ENABLED			
Zaloha1	Full	<input checked="" type="checkbox"/>	EDIT BACKUP PLAN		
Testovací plan	Incremental	<input checked="" type="checkbox"/>	EDIT BACKUP PLAN		
xxxxxx	Full	<input checked="" type="checkbox"/>	EDIT BACKUP PLAN		
dropboxtest	Incremental	<input checked="" type="checkbox"/>	EDIT BACKUP PLAN		

Obrázek 21 BackupPlansView

V pravé části je umístěn TabControl se čtyřmi záložkami, které budou popsány v následujících sekcích.

#### 4.2.1 Záložka Info

V této záložce se nachází podrobné informace pro zvolený zálohovací plán. Najdeme zde i informaci o tom, kdy se pro zálohovací plán spustí další záloha. Pokud v této době aplikace neběží, tak se zálohovací plán zařadí do zálohovací fronty ihned při příštím spuštění aplikace.

The screenshot displays the 'Info Backups Providers Log' window with the following sections:

- Backup Plan:**
  - Name: Zaloha1
  - Backup Plan Type: Full
- Time Plan:**
  - Time Plan Type: Daily
  - Time: 14:54:22
  - Last Backup At: 13-05-2017 14:54:25
  - Next Backup At: 14-05-2017 14:54:22
- Source Paths:**

TYPE	PATH
Folder	C:\Users\Roman\Desktop\SourceFolder\TestFolder
Folder	C:\Users\Roman\Desktop\SourceFolder\1344367826_x3-albion-prelude-trailer_
File	C:\Users\Roman\Desktop\SourceFolder\TestFolder\Space.jpg
- Target Paths:**

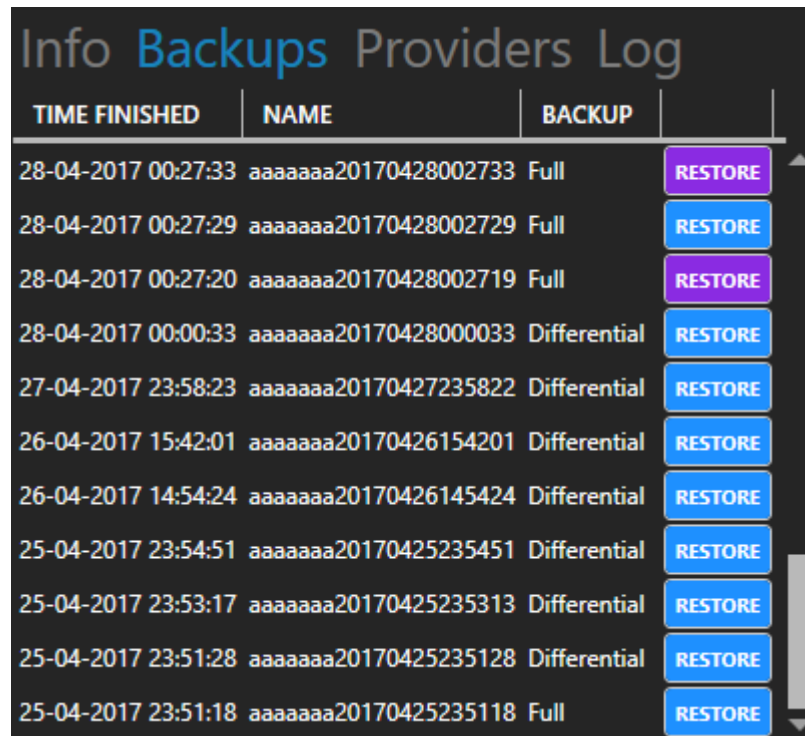
TYPE	PROVIDER	PATH
Folder	LocalBackup	C:\Users\Roman\Desktop\TargetBackupFolder

Obrázek 22 Záložka info

#### 4.2.2 Záložka Backups

Nachází se zde všechny zálohy pro zvolený zálohovací plán. Tyto zálohy jsou seřazené chronologicky, aby uživatel mohl snadno najít požadovaný bod obnovení. Kromě základních informací o záloze, se na každém řádku nachází tlačítko s nápisem „RESTORE“. Na následujícím obrázku si lze všimnout střídajících se barev. Každá barevná sekvence označuje zálohy, které spadají do stejné skupiny. Tyto skupiny se týkají hlavně záloh inkrementálních, které jsou na sobě vzájemně závislé.



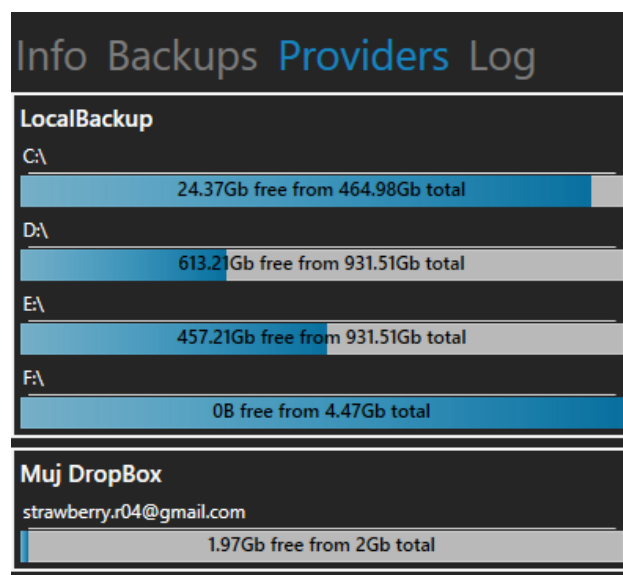


TIME FINISHED	NAME	BACKUP	
28-04-2017 00:27:33	aaaaaaa20170428002733	Full	RESTORE
28-04-2017 00:27:29	aaaaaaa20170428002729	Full	RESTORE
28-04-2017 00:27:20	aaaaaaa20170428002719	Full	RESTORE
28-04-2017 00:00:33	aaaaaaa20170428000033	Differential	RESTORE
27-04-2017 23:58:23	aaaaaaa20170427235822	Differential	RESTORE
26-04-2017 15:42:01	aaaaaaa20170426154201	Differential	RESTORE
26-04-2017 14:54:24	aaaaaaa20170426145424	Differential	RESTORE
25-04-2017 23:54:51	aaaaaaa20170425235451	Differential	RESTORE
25-04-2017 23:53:17	aaaaaaa20170425235313	Differential	RESTORE
25-04-2017 23:51:28	aaaaaaa20170425235128	Differential	RESTORE
25-04-2017 23:51:18	aaaaaaa20170425235118	Full	RESTORE

Obrázek 23 Záložka Backups

#### 4.2.3 Záložka Providers

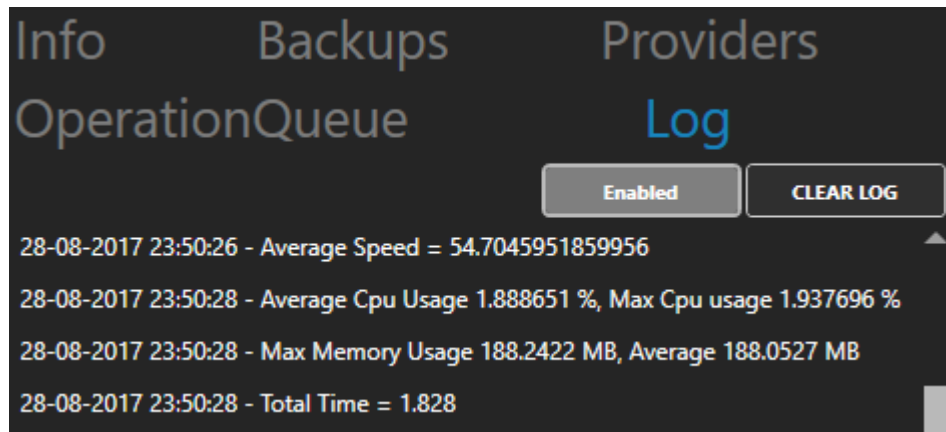
V této záložce se nachází seznam úložišť jednotlivých účtů, u kterých jsou uvedeny i kapacity všech úložišť.



Obrázek 24 Záložka Providers

#### 4.2.4 Záložka Log

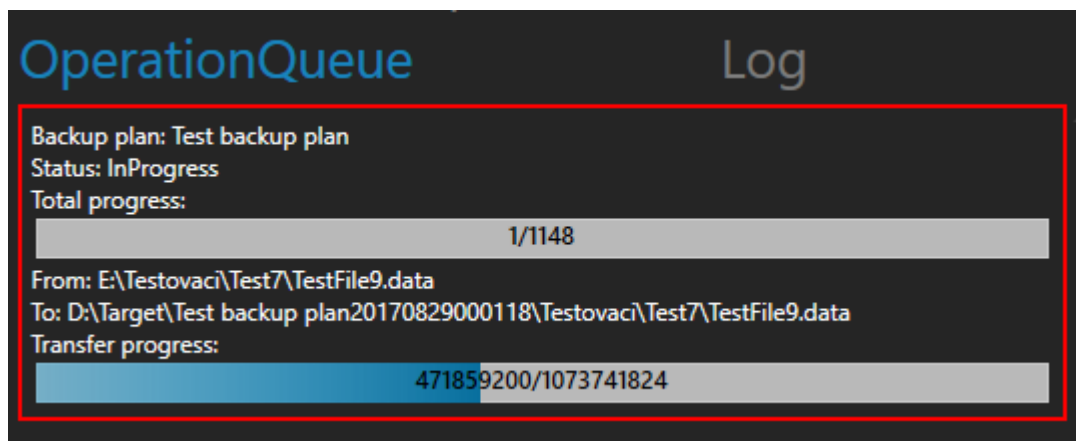
V této záložce se zobrazují zprávy z provozu aplikace. Uživatel zde může nejen sledovat průběh zálohování, ale i chybové informace a stav editace jednotlivých položek.



Obrázek 25 Záložka Log

#### 4.2.5 Záložka OperationQueue

V této záložce můžeme prohlížet zálohy, které čekají ve frontě pro zpracování. U právě vytvářené zálohy můžeme také sledovat stav postupu.

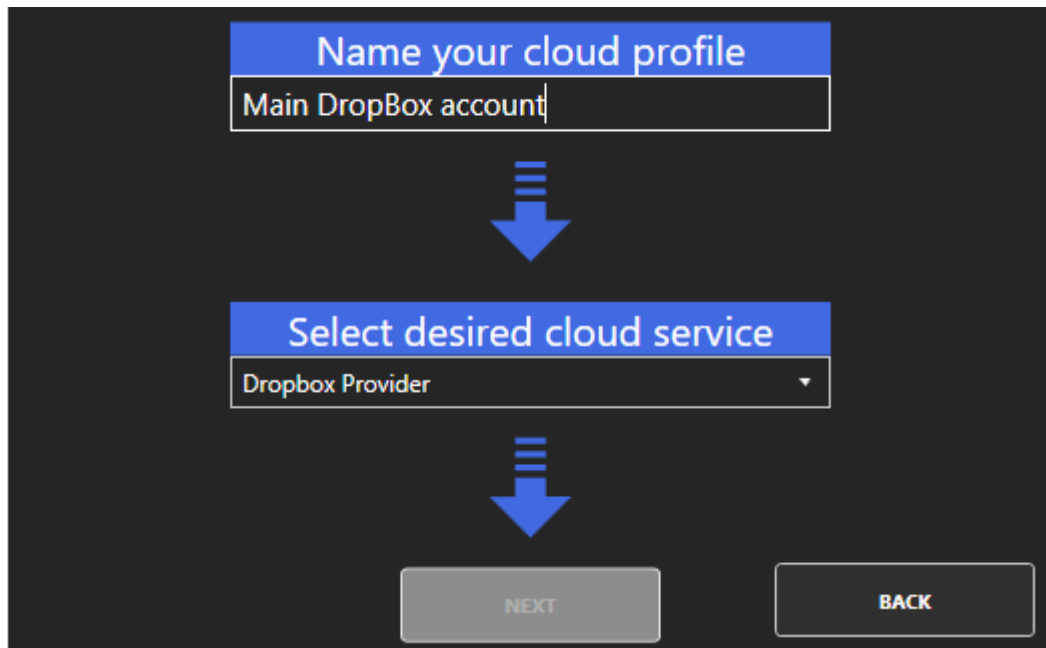


Obrázek 26 Záložka OperationQueue

### 4.3 NewProviderDialogView

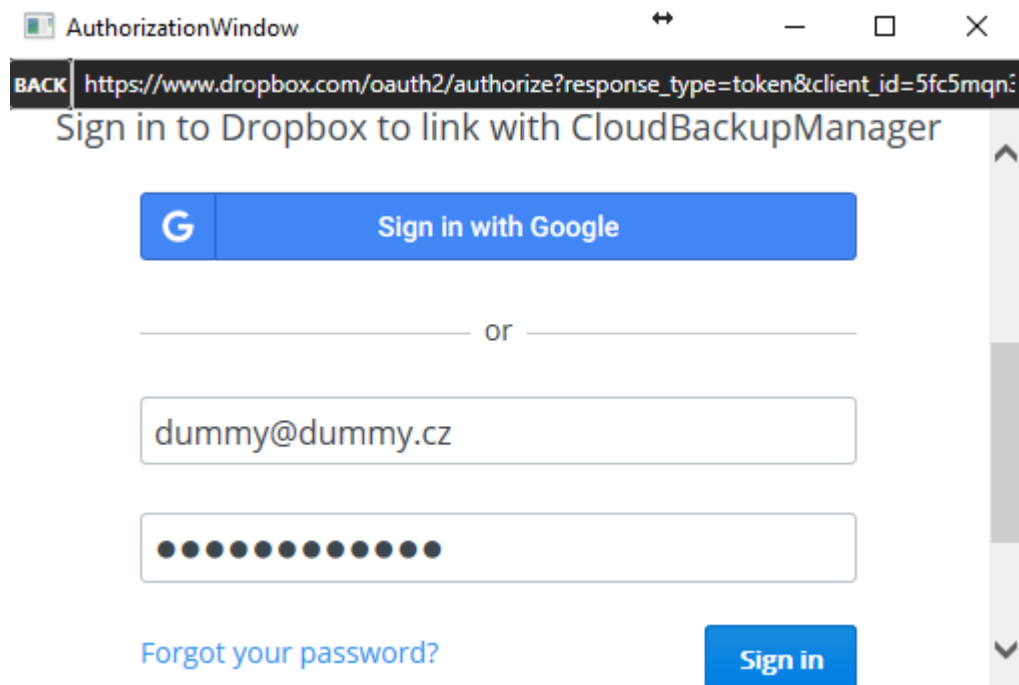
V tomto dialogu si uživatel může vytvářet zálohovací providery. Pro vytvoření nového providera stačí splnit pár kroků, které budou v následujících odstavcích popsány.

Ze všeho nejdříve musí uživatel svého nového providera pojmenovat, poté si zvolí požadovaný typ a klikne na tlačítko další.



Obrázek 27 NewProviderDialogView

Další postup se může lišit v závislosti na zvoleném provideru, ale většinou spočívá v tom, že se uživatel přihlásí pod svým účtem na požadovanou cloudovou službu a potvrdí aplikaci oprávnění.







Obrázek 28 Autorizační okno pro službu Dropbox

Aplikace si neuchovává použité přihlašovací údaje, ale jen uloží takzvaný token, který má určitou platnost a v žádném případě z něj nelze zjistit přihlašovací údaje uživatele. Uživatel má možnost kdykoliv odejmout přístup aplikaci, z webové správy účtu služby Dropbox.

#### Apps linked

You've given these apps access to your personal Dropbox.

App name	Publisher	Access type	
 CloudBackupManager	CloudBackupManager	Full Dropbox 	×
 Microsoft Office Online	Microsoft	Selected files 	×

Obrázek 29 Správa aplikací ve webovém rozhraní služby Dropbox

## 5 POPIS IMPLEMENTACE

### 5.1 Použité knihovny

#### 5.1.1 PRISM

Tento Framework obsahuje pomocné nástroje, které usnadňují vývoj modulárních aplikací ve WPF, UWP a Xamarin Forms. Jsou zde například praktické implementace návrhových vzorů, jako MVVM, DI, Event Aggregator, Commands a další.

#### 5.1.2 MahApps.Metro

Tato knihovna obsahuje GUI kontrolky a styly, které se inspirojí vzhledem Metra z operačního systému Windows 8 a výše. Nalezneme zde i třídu MetroWindow, která rozšiřuje standardní Window o mnoho funkcí.

#### 5.1.3 DropBox.Net

SDK, které obsahuje nástroje usnadňující integraci aplikace s Dropbox API v2.

### 5.2 Základní logika aplikace

V této části práce budou popsány struktury a mechanismy, které jsou využívány pro chod aplikace. Konkrétně se jedná o komunikaci mezi jednotlivými moduly, jejich službami, sdílení dat, a mechanismy propagování stavových informací mezi View a ViewModelem. V každé části bude přiložený výňatek z kódu, který bude demonstrovat praktickou implementaci popisované části.

#### 5.2.1 Dependency injection

Napříč aplikací je používán návrhový vzor dependency injection, konkrétně mix constructor injection a interface injection. Pro řešení většiny závislostí důležitých částí programu se používá UnityContainer, do kterého se inicializují sdílené služby pod jejich patřičným rozhráním. Instance tohoto kontejneru se vytvoří při startu aplikace v bootstraperu, a automaticky se s ní řeší všechny závislosti inicializovaných modulů. Pokud je vyžadován objekt, který implementuje rozhraní IUnityContainer, tak kontejner vrátí svoji instanci.

---

```
public InfrastructureModule(IUnityContainer container, IRegionManager
regionManager, IEventAggregator eventAggregator)
{
    Container = container;
    RegionManager = regionManager;
}
```

---

---

```
        EventAggregator = eventAggregator;  
    }
```

---

#### Zdrojový kód 5 Dependency injection

Všechny služby jsou registrovány jako singleton, což umožňuje jednoduché sdílení dat a služeb napříč všemi moduly aplikace. V následujících odstavcích bude představena ukázka praktické inicializace služby BackupPlanService.

---

```
public class BackupPlanService : LocalServiceBase, IBackupPlanService { . . . }
```

---

#### Zdrojový kód 6 Ukázka Backup Plan Service Singleton

Tato třída implementuje rozhraní IBackupPlanService, pod kterým bude vystupovat v rámci celé aplikace, což znamená, že si v rámci celé aplikace budeme z kontejneru žádat objekt, který implementuje rozhraní IBackupPlanService, aniž bychom se museli starat o konkrétní typ, jeho konstrukci atd.. Tato metoda předání zodpovědnosti je užitečná v tom, že můžeme změnit parametry konstruktoru za jiné známe závislosti, nebo i nahradit BackupPlanService za jiný typ, který výše uvedené rozhraní implementuje, a nemusíme v ostatních částech kódu dělat žádné dodatečné úpravy. Tato registrace probíhá při inicializaci aplikace v inicializační metodě modulu. Objekt typu ContainerControlledLifetimeManager, který se vkládá jako parametr do metody RegisterType, říká kontejneru, že si má následující instanci ponechat a zacházet s ní jako se singletonem. Bez tohoto parametru by kontejner vytvářel pokaždé novou instanci typu BackupPlanService.

---

```
public class InfrastructureModule : IModule  
{  
    . . .  
    public void Initialize()  
    {  
        Container.RegisterType<IBackupPlanService, BackupPlanService>(new  
        ContainerControlledLifetimeManager());  
    }  
    . . .  
}
```

---

#### Zdrojový kód 7 InfrastructureModule

V dalším kroku si můžeme nechat od kontejneru generovat objekty požadovaných typů pomocí metody Resolve. Prakticky to funguje tak, že pokud kontejner řeší konkrétní typ, tak rekurzivně projde a volá metodu Resolve na všechny parametry konstruktoru, které mohou mít také nějaké závislosti. Toto volání se opakuje, dokud není zkonstruovaný celý strom, až do bodu, kdy je možné vrátit požadovaný objekt. Pokud má metoda Resolve vrátit objekt, který implementuje nějaké rozhraní, tak je potřeba, aby toto rozhraní bylo v

kontejneru předem spárované s konkrétním typem, který toto rozhraní implementuje. Pokud se řešený typ nachází v kontejneru jako singleton, tak je navrácen okamžitě bez dalšího procházení vnitřních závislostí.

---

```
IUnityContainer container = unityContainer;
BackupPlanDetailViewModel viewModel =
container.Resolve<BackupPlanDetailViewModel>();

IBackupPlanService canResolveInterfaceToo =
container.Resolve<IBackupPlanService>();
```

---

Zdrojový kód 8 Ukázka práce s Unity Kontejnerem

### 5.2.2 Popis implementace třídy BootStraper

Tato třída dědí od UnityStraper, která se nachází v knihovně PRISM a má na starosti úvodní konfiguraci a registraci modulů za startu aplikace, tomuto procesu se říká také bootstrapping. Nejdříve bylo nutné překrýt virtuální metody z bazové třídy pomocí klíčového slova `override` a v těchto metodách inicializovat a vytvořit hlavní okno aplikace, které je reprezentováno třídou `Shell`, která obsahuje bezparametrický konstruktor, proto ji může ještě prázdný unity kontejner inicializovat.

---

```
class BootStrapper : UnityBootstrapper
{
    protected override DependencyObject CreateShell()
    {
        return this.Container.Resolve<Shell>();
    }

    protected override void InitializeShell()
    {
        base.InitializeShell();

        App.Current.MainWindow = (Window)Shell;
        App.Current.MainWindow.Show();
    }
}
```

---

Zdrojový kód 9 Bootstrapper

V další části je nutné naskládat do katalogu modulů jednotlivé moduly, které chceme při startu inicializovat. Pořadí v jakém moduly naskládáme do katalogu nemá vliv na pořadí v inicializaci, moduly se inicializují v pořadí podle toho, jaké mají vzájemné závislosti. Tyto závislosti lze definovat pomocí atributů v třídě projektu (modulu), který implementuje rozhraní `IModule`. Toto rozhraní obsahuje pouze jednu bezparametrickou metodu `Initialize`, která slouží jako vstupní bod pro inicializaci daného modulu.

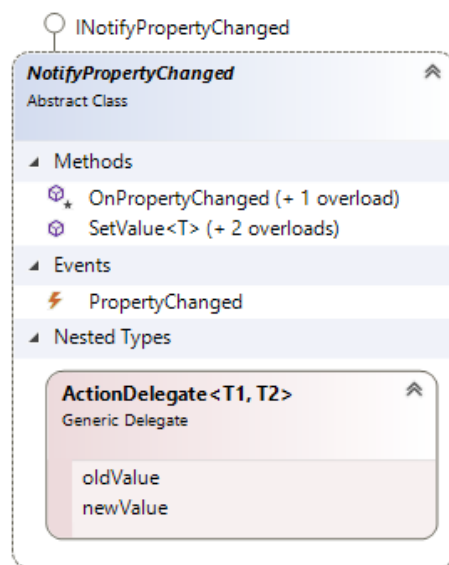
```
protected override void ConfigureModuleCatalog()
{
    Type infrastructureModule = typeof(Infrastructure.InfrastructureModule);
    ModuleCatalog.AddModule(
        new ModuleInfo()
        {
            ModuleName = infrastructureModule.Name,
            ModuleType = infrastructureModule.AssemblyQualifiedName
        });

    Type configurationModule = typeof(Configuration.ConfigurationModule);
    ModuleCatalog.AddModule(
        new ModuleInfo()
        {
            ModuleName = configurationModule.Name,
            ModuleType = configurationModule.AssemblyQualifiedName
        });
}
```

Zdrojový kód 10 Katalog modulů

### 5.2.3 Popis implementace třídy NotifyPropertyChanged

Tato třída implementuje rozhraní INotifyPropertyChanged a snaží se maximálně ulehčit správné notifikování o změnách dat.



Obrázek 30 Diagram třídy  
NotifyPropertyChanged

Umožňuje například nastavení hodnoty pomocí použití bázové metody `SetValue` v setru nastavované vlastnosti, kde se v implementaci metody automaticky kontroluje, zda není nastavovaná hodnota rovna té staré. V případě nerovnosti se přiřadí nová hodnota a automaticky vyvolá událost o změně vlastnosti. Tato událost se samozřejmě vyvolá pouze za předpokladu, že má alespoň jednoho subscripera. Do parametrů je možné i vložit instanci delegáta (reference na metodu), která má stejnou signaturu jako `ActionDelegate`. Do



parametrů referencované metody se po změně dosadí nová i stará hodnota, na které můžeme podle potřeby zareagovat. Například můžeme zakázat použití nějaké funkce v uživatelském rozhraní, pokud nová hodnota není validní.

---

```
public abstract class NotifyPropertyChanged : INotifyPropertyChanged
{
    public delegate void ActionDelegate<T1, T2>(T1 oldValue, T2 newValue);

    public void SetValue<T>(ref T field, T value, ActionDelegate<T, T>
action, [CallerMemberName] string propertyName = null)
    {
        var comparer = EqualityComparer<T>.Default;
        if (!comparer.Equals(field, value))
        {
            var oldValue = field;
            field = value;
            action?.Invoke(oldValue, value);
            OnPropertyChanged(propertyName);
        }
    }
}
```

---

#### Zdrojový kód 11 Ukázka třídy NotifyPropertyChanged

Třída obsahuje i přetížení metody OnPropertyChanged, které umožňuje řetěžit vlastnosti, pro které potřebujeme vyvolat událost o změně.

---

```
private ulong availableSpace;

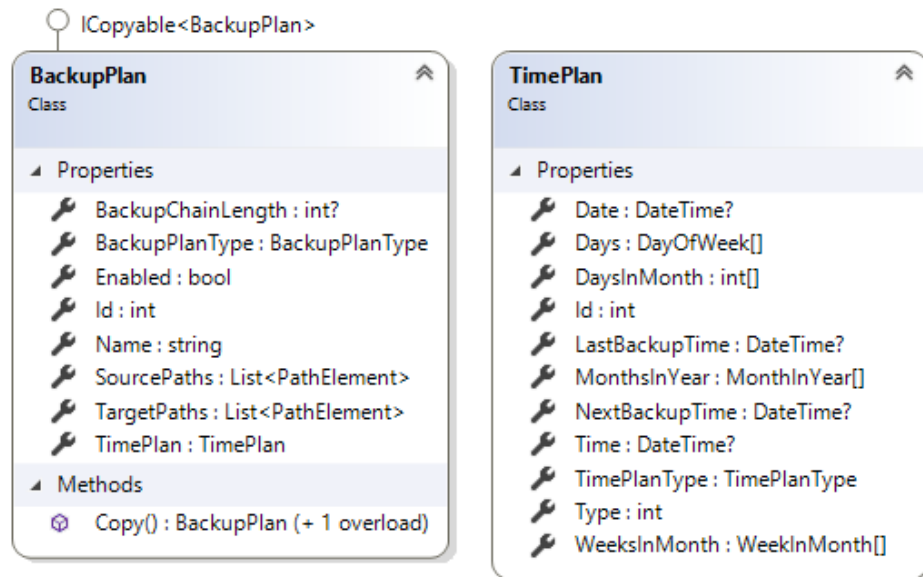
public ulong AvailableSpace
{
    get { return availableSpace; }
    set => SetValue(ref availableSpace, value, () =>
OnPropertyChanged(nameof(PercentUsed), nameof(SpaceUsageText)));
}
```

---

#### Zdrojový kód 12 Řetězení vlastností

### 5.3 Popis implementace třídy BackupPlan

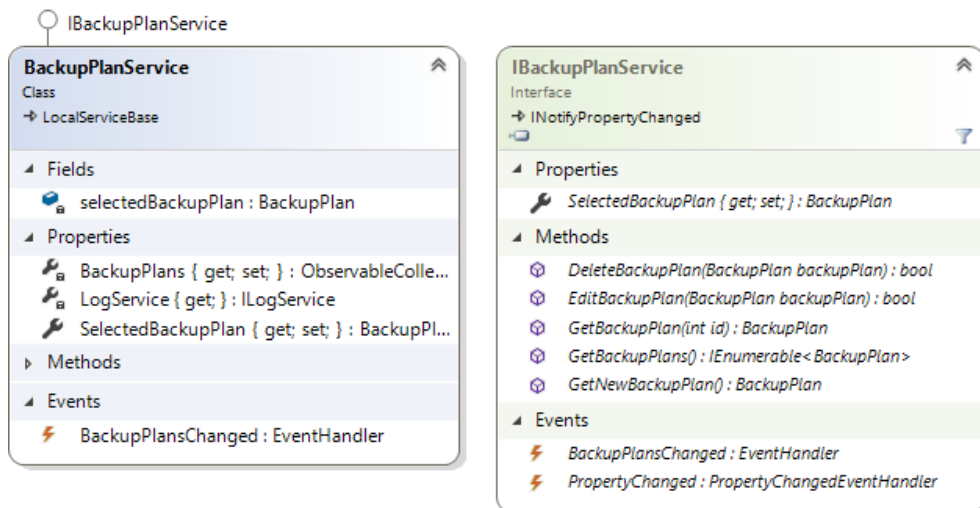
Tato třída reprezentuje zálohovací plán, který obsahuje data, které ovlivňují chování a průběh samotné zálohy. Každý plán také obsahuje vlastnost TimePlan, která obsahuje nezbytná data pro kalkulaci času periodického zálohování.



Obrázek 31 Diagram tříd BackupPlan a TimePlan

#### 5.4 Popis implementace služby BackupPlanService

Tato služba implementuje rozhraní IBackupPlanService a slouží jako centrální bod pro čtení a manipulaci s objekty zálohovacích plánů. Je zde zapouzdřena veškerá logika ukládání, čtení a konstrukce nových objektů typu BackupPlan, takže například přechod na jiný typ ukládání těchto dat neovlivní ostatní části programu.



Obrázek 32 Class Diagram třídy BackupPlanService a rozhraní IBackupPlanService

Vnitřní implementace je vláknově bezpečná, což umožňuje editaci a čtení záznamů paralelně, aniž bychom se museli bát nekonzistence dat a obsahuje také cache zálohovacích plánů, která je synchronizována s databází, což umožňuje rychlejší předávání záznamů a menší zátěž na CPU.

Inicializace probíhá po startu programu, kdy se konkrétní typ služby spáruje s jejím rozhraním v rámci DI kontejneru jako singleton. Poté tuto instanci mohou obdržet všechny části programu, které splňují následující body.

1. Projekt má referenci na knihovnu `CloudBackupManager.Infrastructure.Interface`.
2. Třída, ve které se má implementovat práce s touto službou má v konstruktoru vloženou závislost `IBackupPlanService`.
3. Instance třídy bude zkonstruována pomocí kontejneru, který do závislosti (parametru) správně dosadí instanci `BackupPlanService`.

---

```
var vm = Container.Resolve<BackupPlanDetailViewModel>();
. . . .
public class BackupPlanDetailViewModel : ViewModelBase
{
    private readonly IBackupPlanService BackupPlanService;
    //konstruktor

    public BackupPlanDetailViewModel(IBackupPlanService backupPlanService, . . .
    {
        BackupPlanService = backupPlanService;
```

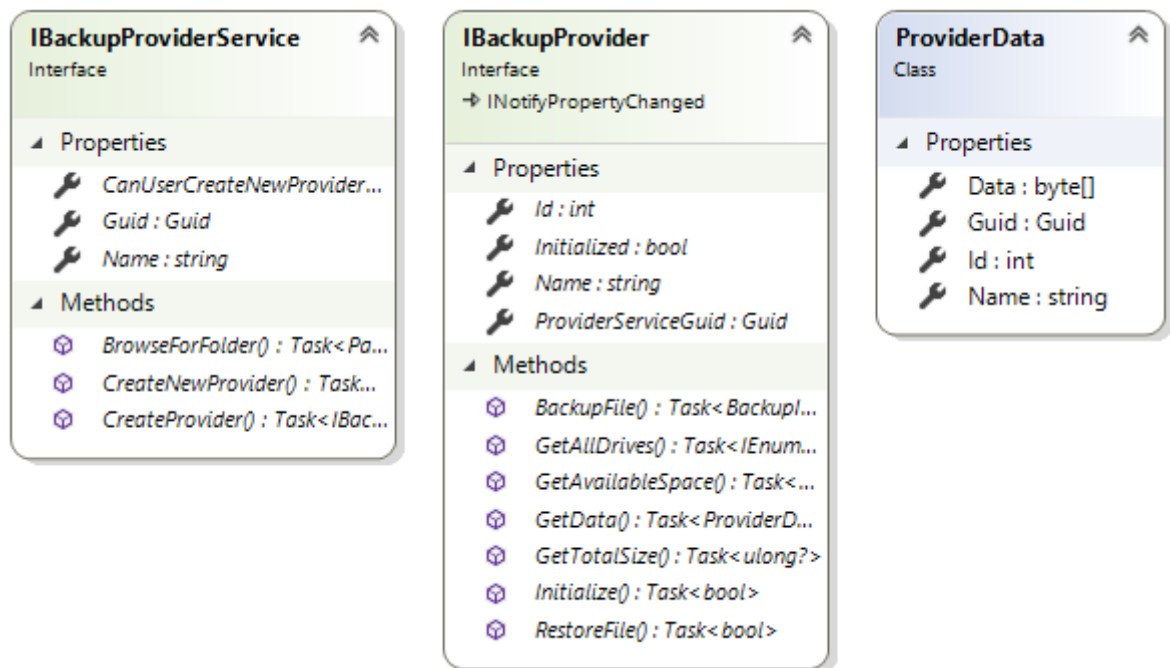
---

Zdrojový kód 13 Inicializace pomocí kontejneru

## 5.5 Implementace služby `BackupProviderManager`

Tato služba implementuje rozhraní `IBackupProviderManger` a slouží jako správce konkrétních zálohovacích služeb. Jsou zde spravovány tři typy objektů.

1. Objekt implementující rozhraní `IBackupProviderService`.
2. Objekt implementující rozhraní `IBackupProvider`.
3. Objekt typu `ProviderData`.



Obrázek 33 Diagram tříd spravovaný službou BackupProviderManager

### 5.5.1 IBackupProviderService

Objekty, které implementují tento interface reprezentují centrální službu pro konkrétní typ úložiště (Lokální úložiště, Dropbox úložiště). Pomocí asynchronní metody `CreateNewProvider` v této službě je možné vytvářet nové zálohovací poskytovatele, kteří se starají o proces přenosu zálohovaných dat na cílové úložiště účtu.

Například, aby služba mohla vytvořit poskytovatele zálohování pro Dropbox, musí nejdříve zobrazit uživateli přihlašovací okno, které obsahuje prohlížeč se zobrazenou přihlašovací stránkou, do které se musí uživatel přihlásit na požadovaný účet. Po přihlášení a autorizaci potřebných oprávnění se v url navrátí přihlašovací token, který bude poskytovatel používat pro komunikaci s účtem služby Dropbox.

---

```
internal class DropBoxBackupProviderService : IBackupProviderService
{
    public async Task<IBackupProvider> CreateNewProvider()
    {
        var token = await DropBoxHelper.GetAcessToken();
        if (token != null)
        {
            ClientData cd = new ClientData()
            {
                AccessToken = token.Serialize<string>()
            };
            return new DropboxProvider(new ProviderData())
        }
    }
}
```

---

---

```
        {
            Id = 0,
            Data = cd.Serialize(),
            Guid = AssignedGuid,
            Name = String.Empty
        }, Container);
    }
    return null;
}
```

---

#### Zdrojový kód 14 Vytvoření poskytovatele pro Dropbox

V následující ukázce kódu je předveden postup inicializace služby DropBoxBackupProviderService která implementuje rozhraní IBackupProviderService do správce zálohovacích služeb.

---

```
public class DropboxModule : IModule
{
    public void Initialize()
    {
        Container.Resolve<IBackupProviderManager>().AddBackupProviderService(
            new DropBoxBackupProviderService(Container));

        Container.Resolve<IBackupProviderManager>().InitProviders(
            Guid.Parse(DropBoxBackupProviderService.AssignedGuid.ToString()))
        .ConfigureAwait(false).GetAwaiter().GetResult();

        var provider = Container.Resolve<IBackupProviderManager>()
            .Providers.FirstOrDefault(q => q is DropboxProvider);
    }
}
```

---

#### Zdrojový kód 15 Inicializace Dropbox modulu

### 5.5.2 IBackupProvider

Objekt implementující rozhraní IBackupProvider reprezentuje poskytovatele zálohování, který má na starost komunikaci s konkrétním datovým účtem. Například poskytovatel pro Dropbox zajišťuje komunikaci s úložišti účtu, do kterého se při vytváření poskytovatele uživatel přihlásil a potvrdil patřičná oprávnění. Jeden účet může mít i více úložišť, což platí například pro poskytovatele lokálního zálohování.

Každý poskytovatel generuje objekt typu ProviderData, který je nutný pro zpětné zrekonstruování poskytovatele do posledního stavu. Například pokud poskytovatel potřebuje pro svou práci přihlašovací token, tak ho uloží právě do tohoto objektu. Data se mohou poté libovolně ukládat do databáze pomocí služby implementující rozhraní IBackupProviderManager. Díky tomuto mechanismu je možné po restartu počítače

poskytovatele znovu obnovit, aniž by po uživateli bylo vyžadováno opětovné přihlášení do svého účtu.

---

```
public class BackupProviderManager : LocalServiceBase,
Interface.Services.IBackupProviderManager
{

    public async Task<bool> SaveProviderState(IBackupProvider backupProvider)
    {
        var data = await backupProvider.GetData();
        EditProviderData(data);
        return true;
    }
}
```

---

#### Zdrojový kód 16 Uložení stavu providera

Kromě komunikace se poskytovatel také stará o samotnou zálohu a obnovení konkrétního souboru. Pro spuštění zálohy stačí zavolat parametrickou metodu BackupFile. Níže najdeme ukázkou metody, která se nachází ve třídě DropboxProvider.

---

```
public async override Task<BackupItem> BackupFile(OperationTaskSource source,
RelativeFilePath sourceFilePath, PathElement targetPath)
{
    var fi = new FileInfo(sourceFilePath.FullPath);
    var uploadFolderPath = targetPath + "/" + source.BackupPlan.Name +
    $"{source.OperationStart.Value.ToString("yyyyMMddHHmmss")}";
    var targetFilePath = uploadFolderPath + "/" + fi.Name;

    using (var httpClient = GetHttpClient())
    {
        using (var client = await GetDropBoxClient(httpClient))
        {
            try
            {
                await ChunkUpload(client,sourceFilePath.FullPath,
                targetFilePath);
                return new BackupItem()
                {
                    ProviderId = this.Id,
                    PathFrom = sourceFilePath.FullPath,
                    DateEdit = fi.LastWriteTime,
                    PathTo = targetFilePath
                };
            }
            catch (Exception e)
            {
                Container.Resolve<ILogService>().AddLog(e.ToString());
                return null;
            }
        }
    }
}
```

---

#### Zdrojový kód 17 zálohování souboru na službu Dropbox

## 5.6 Popis implementace třídy BackupService

Tato služba má na starosti logistiku zálohování. Po inicializaci služby se založí Task, který běží na separátním vlákně a periodicky provádí postupně několik operací.

1. Zkontroluje, zda se v pořadníku záloh pro obnovení nenachází záloha, kterou si uživatel vyžádal obnovit. Pokud ano, okamžitě se bude snažit provést operaci obnovení dat ze zálohy.
2. Zkontroluje, zda se v pořadníku zálohovacích plánů nenachází nějaký plán. Pokud ano, okamžitě se bude snažit provést zálohu.
3. Zkontroluje naplánované časy záloh jednotlivých zálohovacích plánů. Je-li naplánovaný čas pro zálohu menší nebo roven času momentálnímu, tak se zálohovací plán přesune do pořadníku.
4. Naplňuje budoucí časy záloh zálohovacích plánů.

## 5.7 Zabezpečení

### 5.7.1 OAuth 2.0

Jak již bylo nastíněno v předchozích kapitolách, autorizace na cloudová úložiště probíhá pomocí protokolu OAuth 2.0. Vývojář musí nejdříve zaregistrovat svoji aplikaci na příslušném cloudovém úložišti a obdržet identifikační údaje aplikace. Tyto údaje jsou následující:

- App Key
- App Secret

Aby mohla aplikace přistupovat k datům na účtu uživatele, musí jí uživatel autorizovat potřebná přístupová práva. Veškerá komunikace probíhá přes protokol https, takže je veškerá komunikace mezi aplikací a službou zašifrovaná. Přidělení probíhá tak, že aplikace zobrazí přihlašovací formulář cloudové služby ve webovém prohlížeči. Do parametrů autorizačního url, který přesměruje uživatele na přihlašovací formulář je vložený i výše zmíněný AppKey, který identifikuje aplikaci, která požaduje autorizaci. Součástí je i takzvané Redirect Uri, které specifikuje, kam bude uživatel přesměrován po dokončení autorizace, například na nějakou stránku která uživateli sdělí, že autorizace aplikace byla úspěšně provedena. Výsledné url pro službu Dropbox vypadá například následovně:

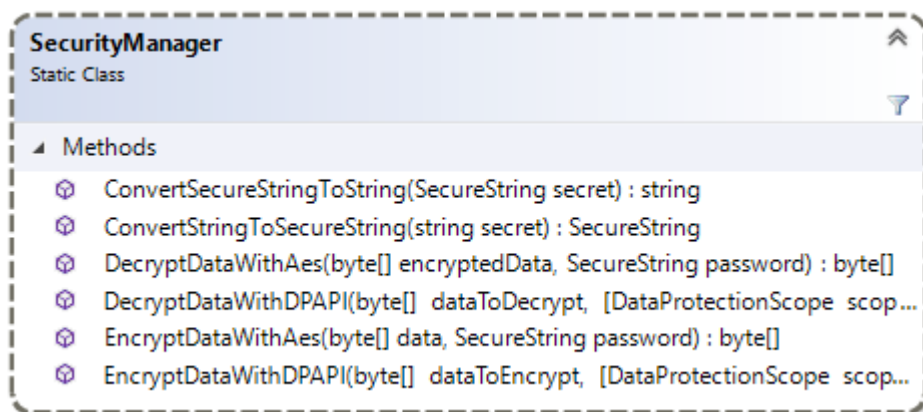
[https://www.dropbox.com/oauth2/authorize?response\\_type=token&client\\_id=XXXXXX XX&redirect\\_uri=https:%2F%2Flocalhost%2Fauthorize&state=3c45f2b4ee9143808968508056229d2a](https://www.dropbox.com/oauth2/authorize?response_type=token&client_id=XXXXXX XX&redirect_uri=https:%2F%2Flocalhost%2Fauthorize&state=3c45f2b4ee9143808968508056229d2a)

#### Zdrojový kód 18 Příklad autorizačního url služby Dropbox

Po dokončení autorizace aplikace obdrží v návratovém url autorizační kód. Tento autorizační kód poté aplikace u služby vymění za přístupový token, který aplikace zašifruje a uloží. Pomocí tohoto tokenu se aplikace autorizuje při každém požadavku na službu. Výsledkem je, že se aplikace nedostane k přihlašovacím údajům uživatele, takže nenese zodpovědnost za jejich zabezpečení a skladování. Aplikace musí pouze zabezpečit svoje identifikační údaje a přihlašovací token. Uživatel může aplikaci kdykoliv dodatečně odepřít přístup z webového rozhraní služby. V tomto případě aplikace ztratí veškerý přístup k datům na službě a celý proces se autorizace by se musel zopakovat.

#### 5.7.2 Řešení zabezpečení citlivých údajů

V rámci zabezpečení přístupových tokenů a ostatních citlivých údajů, byla vytvořena statická třída `SecurityManager`, která se nachází v projektu `CloudBackupManager.Infrastructure.Interface`. Tato třída obsahuje metody šifrování/dešifrování, kterými jsou zabezpečeny všechny citlivé údaje v rámci celé aplikace.



Obrázek 34 Class diagram třídy `SecurityManager`

Třída nabízí 2 různé způsoby šifrování a dešifrování. Nejvíce důležitou metodou je šifrování pomocí služby Data Protection Application Programming Interface (DPAPI), které umožňuje zabezpečení na úrovni operačního systému. Knihovna `.Net` obsahuje třídu `ProtectedData`, která slouží jako wrapper kolem této služby a umožňuje velmi jednoduché



použití. Při šifrování dat nezadáme heslo, pouze předáme data, která chceme šifrovat a DPAPI zašifruje data vzhledem k rozsahu který specifikujeme.

Možné rozsahy jsou následující:

- `CurrentUser` – Pouze proces spuštěný pod daným uživatelem může dešifrovat zašifrovaná data.
- `LocalMachine` – Všechny procesy běžící na daném počítači mohou dešifrovat zašifrovaná data.

Dále tato třída poskytuje možnost zabezpečení pomocí symetrické šifry AES s velikostí klíče 256 bitů. Metody lze také kombinovat tak, že nejdříve data zašifrujeme symetrickou šifrou AES a poté použitý klíč zašifrujeme pomocí DPAPI.

---

```
public async Task<IBackupProvider> CreateNewProvider()
{
    var token = await DropBoxHelper.GetAcessToken();
    if (token != null)
    {
        ClientData cd = new ClientData()
        {
            AccessToken = SecurityManager.EncryptDataWithDPAPI(token.Serialize())
        };

        return new DropboxProvider(new ProviderData()
        {
            Id = 0,
            Data = cd.Serialize(),
            Guid = AssignedGuid,
        }, Container);
    }

    return null;
}
```

---

Zdrojový kód 19 Ukázka použití třídy SecurityManager

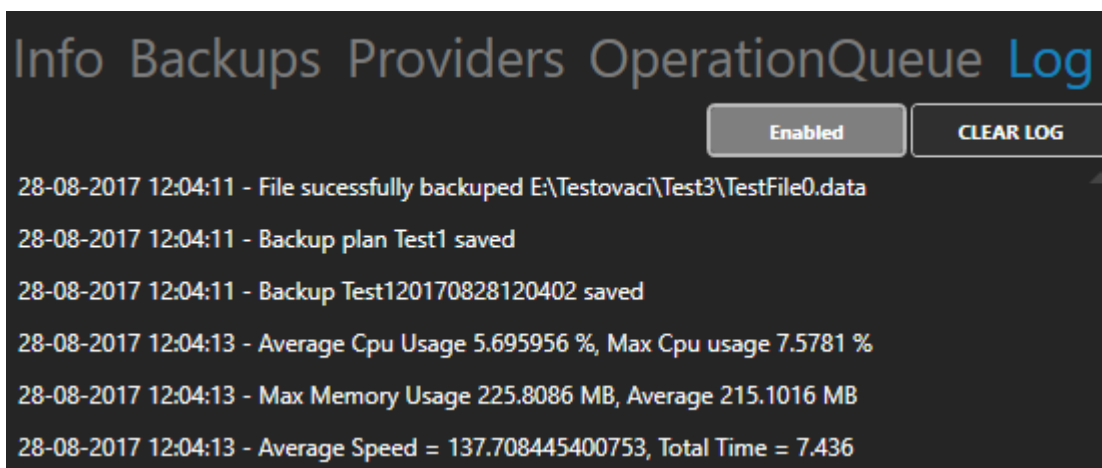
## 6 ZHODNOCENÍ PROGRAMU Z HLEDISKA VÝKONNOSTI

Aplikace byla také podrobena výkonostnímu testování. Jednotlivé procesy v aplikaci jsou paralelizované, takže uživatelské rozhraní je responzivní i během procesu zálohování a jiných operací, které běží na pozadí.

V klidovém stavu aplikace zabírá kolem 120 MB, v závislosti na tom, kolik zálohovacích plánů a poskytovatelů má uživatel vytvořených a kolik položek se nachází v logu aplikace. Pro čtení datových struktur se čte a zpracovává každá hodnota samostatně, aby se v paměti neskladovala celá struktura.

Aplikace se snaží neplytvat systémovými zdroji a ani při záloze velkých dat nespotřebuje příliš paměti. Vytížení procesoru někdy stouplo až na 30 procent, v případě zálohy velkého množství malých souborů. Výsledky testů je možné najít v tabulkách níže. Zálohování na více cloudových úložišť nemá na výkonost aplikace žádný vliv, vzhledem k tomu, že se jednotlivé požadavky na zálohu řadí do fronty.

Získávání údajů probíhalo přímo v aplikaci přes třídu System.Diagnostics.PerformanceCounter. Aplikace se jednou za sekundu cyklicky dotazovala na vytížení prostředků a po provedení zálohy spočítala základní údaje, které vypisovala do logu.



Obrázek 35 Ukázka logu aplikace

## 6.1 Testovací sestava:

- Procesor - Intel Core i7-3770k (3.50 GHz, 4 jádra)
- Paměť - 16 GB
- Specifikace Disků – 2x Westen Digital Black (1TB, Čtení ~120MB/s, Zápis ~65 MB/s)

Rychlost připojení na internet v době testování (Mb/s): 7.04 DOWN / 0.41 UP

## 6.2 Testovací vzorky dat

Ze všeho nejdříve byly vygenerované testovací vzorky dat. Obsah vytvořených dat je náhodně generován zvlášť naprogramovanou aplikací.

- Vzorek 1 - 1 soubor o velikosti 1024MB
- Vzorek 2 - 1024 souborů o velikosti 1MB
- Vzorek 3 - 10 souborů o velikosti 1MB
- Vzorek 4 - 1 soubor o velikosti 10MB

## 6.3 Test lokálního zálohování vzorku 1

V tomto testu se provedla úplná záloha vzorku 2 z jednoho lokálního disku WD Black na druhý. Výsledky ukázaly, že zálohování jednoho velkého souboru je velmi rychlé a vytěžuje málo systémové prostředky.

Tabulka 7 Výsledek testu lokálního zálohování se vzorkem 2

	Čas trvání v sekundách	Průměrná rychlost v MB/s	CPU průměr (%)	CPU max (%)	Paměť průměr (MB)	Paměť max (MB)
1	10,912	93,8	6	9,30	213,00	221,00
2	7,4	139	8,4	10,50	264,00	254,00
3	7,385	139	3,2	6,00	193,00	204,00
4	10,734	95,4	3,2	8,40	206,00	214,00
5	7,436	137,7	5,7	7,60	215,00	226,00
Průměr	8,773	109,0	5,3	8,4	218,2	223,8
Směrodatná odchylka	1,872170078	24,896653	2,184033	1,704	27,01296	18,7936

## 6.4 Test lokálního zálohování vzorku 2

Pro tento test byl použit vzorek 2, který obsahuje spoustu malých souborů. Součet velikosti všech těchto souborů je roven velikosti vzorku 1, který obsahuje jeden velký soubor. Byla vytvořena úplná záloha, stejně jako u přechozího testu. Výsledek testu ukazuje, že zálohování velkého množství malých souborů má velký dopad na rychlost a vytížení systémových prostředků, kromě využití paměti počítače, která byla jen o trošku vyšší než u vzorku 2. Zálohování trvalo více než 2 krát déle než v předchozím případě a průměrné využití CPU, bylo téměř 4x vyšší.

Tabulka 8 Výsledek testu lokálního zálohování se vzorkem 4

	Čas trvání v sekundách	Průměrná rychlost v MB/s	CPU průměr (%)	CPU max (%)	Paměť průměr (MB)	Paměť max (MB)
1	28,144	36,4	16,15	23,92	196,00	225,00
2	24,996	41,0	17,07	27,01	218,00	244,00
3	23,688	43,2	17,45	23,69	233,00	252,00
4	23,412	43,7	19,75	34,23	237,00	256,00
5	23,237	44,1	18,35	25,84	238,00	255,00
Průměr	24,695	41,7	17,8	26,9	224,4	246,4
Směrodatná odchylka	2,04768792	3,191421	1,366777	4,30238	17,78482	12,8569

## 6.5 Test lokálního zálohování vzorku 2 a nastavenou inkrementální zálohou

Pro tento test je opět použit vzorek 4. Před samotným testem se provedla úplná záloha testovacích dat, aby se mohla otestovat vytížení zálohou inkrementální. Před každým spuštěním se provedla změna na polovině (512) souborů z testovacího vzorku.

Tabulka 9 Výsledky testů inkrementálního lokálního zálohování se vzorkem 2

	Čas trvání v sekundách	Průměrná rychlost v MB/s	CPU průměr (%)	CPU max (%)	Paměť průměr (MB)	Paměť Max (MB)
1	12,521	43,6	18,12	22,99	252,45	269,00
2	12,73	42,8	20,07	26,23	242,00	265,00

3	12,25	44,8	19,79	29,81	248,00	285,00
4	12,85	42,4	18,81	25,52	241,75	265,00
5	12,788	42,6	19,26	30,33	246,00	271,00
Průměr	12,588	43,2	19,2	27,0	246,0	271,0
Směrodatná odchylna	0,26302139	0,9584467	0,778877	3,075936	4,462539	8,246211

Z výsledné tabulky si lze všimnout, že čas trvání nám naznačuje, že se skutečně zálohuje pouze polovina změněných souborů. Nároky na paměťový prostor jsou v tomto případě poněkud vyšší. Je to způsobeno tím, že aplikace načítá metadata o předchozích zálohách do paměti, pokud součet velikostí metadat nepřesáhne velikost 100MB. V případě větší velikosti se s metadatami pracuje na disku. V tomto případě měl jeden soubor s metadaty popisující 512 souborů velikost 287 KB.

## 6.6 Test zálohování vzorku 3 na službu Dropbox

Test využívá testovací vzorek 3. Z tabulky je zřejmé, že upload internetového připojení zbrzdí zálohování. Díky tomuto aplikace nebere téměř žádné systémové prostředky, kromě paměti, jejíž využití je i tak podstatně nižší než u případů rychlého zálohování na lokální médium.

Tabulka 10 Výsledky testu zálohování vzorku 3 na službu Dropbox

	Čas trvání v sekundách	Průměrná rychlost v MB/s	CPU průměr (%)	CPU max (%)	Paměť průměr (MB)	Paměť Max (MB)
1	226,517	0,04	2,34	20,26	163,00	167,00
2	220,061	0,05	1,69	10,82	182,00	184,00
3	221,846	0,05	2,08	11,77	186,00	190,00
4	220,579	0,05	2,08	21,74	189,00	193,00
5	233,549	0,04	1,89	12,77	175,00	178,00
Průměr	224,510	0,04	2,02	15,47	179,0	182,4
Směrodatná odchylna	5,658507471	0,0027386	0,242549	5,12	10,36822	10,3586

## 6.7 Test zálohování na službu Dropbox se vzorkem 4

Zálohování vzorku 4 proběhlo očekávaně o něco rychleji, vzhledem k tomu, že pro zálohu bylo potřeba méně dotazů na cloudovou službu. Upload na službu DropBox probíhá po 5MB

blocích. Stejně jako u předchozího testu je využití systémových prostředků velmi malé, kvůli nízké přenosové rychlosti.

Tabulka 11 Výsledky zálohování na službu Dropbox se vzorkem 4

	Čas trvání v sekundách	Průměrná rychlost v MB/s	CPU průměr (%)	CPU max (%)	Paměť průměr (MB)	Paměť Max (MB)
1	202,807	0,05	4,72	20,77	218,00	222,00
2	221,415	0,05	2,18	14,91	236,00	237,00
3	207,934	0,05	1,048	12,18	153,00	156,00
4	197,91	0,05	1,65	1,64	219,00	221,00
5	201,899	0,05	1,548	10,81	202,00	204,00
Průměr	206,393	0,05	2,23	12,06	205,6	208,0
Směrodatná odchylka	9,124854602	0,0004472	1,449245	6,967	31,76948	31,3289

## 6.8 Test zálohování vzorku 3 na službu Google Drive

Výsledky jsou podobné zálohování vzorku 3 na službu Dropbox, avšak doba zálohování proběhla ve všech testech o pár sekund rychleji a stejně jako u předchozích měření je vytížení procesoru a paměti velmi nízké.

Tabulka 12 Výsledek zálohování vzorku 3 na službu GoogleDrive

	Čas trvání v sekundách	Průměrná rychlost v MB/s	CPU průměr (%)	CPU max (%)	Paměť průměr (MB)	Paměť Max (MB)
1	212,486	0,05	1,56	10,81	164,00	170,00
2	211,995	0,05	3,11	23,07	164,00	167,00
3	209,185	0,05	3,17	18,78	171,00	174,00
4	217,667	0,04	3,92	29,62	182,00	185,00
5	211,603	0,05	2,88	16,30	182,00	186,00
Průměr	212,587	0,05	2,93	19,72	172,6	176,4
Směrodatná odchylka	3,110824842	0,0031305	0,8587025	7,092	9,044335	8,67756

## ZÁVĚR

Tato práce obsahuje základní poznatky problematiky zálohování a cloudových úložišť. Byly popsány rizika ztráty dat, stručný popis procesu zálohování a jednotlivé typy záloh. Dále byly popsány použité technologie a návrhové vzory, které byly použity při praktickém vývoji aplikace a vysvětlen pojem cloudové úložiště, spolu s popisem základních typů cloudových úložišť.

V rámci praktické části práce byla realizována aplikace, která umí zálohovat jak lokálně, tak i do cloudového úložiště Dropbox a Google Drive. Při tvorbě aplikace byla provedena základní analýza požadavků a stanovení cílů projektu.

Dále praktická část obsahuje stručný popis projektů a závislostí, ze kterých se výsledná aplikace skládá, spolu s popisem uživatelského rozhraní aplikace, který se dá použít i jako uživatelský manuál. Součástí je i popis praktické implementace klíčových částí, spolu s ukázkami kódu. Tato část je nezbytná pro správné pochopení vnitřní funkcionality aplikace.

V poslední části práce je aplikace otestována z hlediska výkonnosti nad sadou vygenerovaných dat, které simulují i určité extrémy, například velké množství drobných dat. Veškeré výsledky měření jsou zpracovány do tabulek a popsány.

Práce na tomto projektu byla přínosnou zkušeností, zejména při praktické implementaci struktur aplikace a návrhových vzorů, mezi které hlavně patří návrhový vzor MVVM a Dependency Injection. Tyto návrhové vzory byly prakticky použity napříč celou aplikací. Dále jsem získal zkušenosti s vývojem pluginu pro platformu Dropbox a Google Drive. V rámci práce byl také vypracován návod, který popisuje jak vytvořit plugin a načíst ho do aplikace. Tento návod je umístěn na příloženém DVD.

Hlavním přínosem naprogramované aplikace je možnost periodické zálohy na cloudová úložiště a režim rovnoměrného zálohování, který umožňuje přerozdělování dat podle vytížení jednotlivých úložišť. Vzhledem k tomu, že je aplikace napsaná modulárně, tak je možné ji v budoucnosti nadále rozšiřovat o novou funkcionalitu a moduly.

**SEZNAM POUŽITÉ LITERATURY**

- [1] ŠKODOVÁ, Jana. Nejčastější důvody ztrát firemních dat a jak se jich vyvarovat. *CIO* [online]. 2017, , 3 [cit. 2017-05-19]. Dostupné z: <http://businessworld.cz/bezpecnost/nejcastejsi-duvody-ztrat-firemnich-dat-a-jak-se-jich-vyvarovat-13412-p14594>
- [2] JUNEK, Pavel. *Zálohování a archivace dat v datových centrech v podnikovém prostředí* [online]. Ústí nad Labem, 2013 [cit. 2017-05-19]. Dostupné z: [http://www.diplomovaprace.cz/2013/9/Bakalarska\\_prace\\_Pavel\\_Junek.pdf](http://www.diplomovaprace.cz/2013/9/Bakalarska_prace_Pavel_Junek.pdf).  
Bakalářská práce. Univerzita Jana Evangelisty Purkyně. Vedoucí práce Ing. Toni Koluch, Ph.D.
- [3] NELSON, Steven. *Pro data backup and recovery securing your information in the terabyte age* [online]. Berkeley, CA: Apress, 2011 [cit. 2017-05-19]. ISBN 978-143-0226-635.
- [4] LISKA, Allan a Timothy GALLO. *Ransomware: Defending Against Digital Extortion* [online]. 2016. Sebastopol, California: O'Reilly, 2016 [cit. 2017-05-19]. ISBN 978-1-4919-6782-9. Dostupné z: <http://shop.oreilly.com/product/0636920054290.do?sortby=publicationDate>
- [5] MICZKA, Marian. *Zálohování dat a cloudová uložení* [online]. Brno, 2015, 65 s. [cit. 2017-05-19]. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=106243](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=106243).  
Vysoké učení technické v Brně, Fakulta podnikatelská. Vedoucí práce Ing. Jiří Kříž, Ph.D.
- [6] MIČAN, Jiří. *Zálohování dat* [online]. Praha, 2015, 65 s. [cit. 2017-05-19]. Dostupné z: [https://is.bivs.cz/th/22761/bivs\\_b/bakalarska\\_prace\\_zalohovani\\_dat\\_\\_Mican\\_Jiri.pdf](https://is.bivs.cz/th/22761/bivs_b/bakalarska_prace_zalohovani_dat__Mican_Jiri.pdf).  
Bakalářská práce. Bankovní institut vysoká škola Praha, Katedra informatiky a kvantitativních metod. Vedoucí práce Ing. Jiří Kříž, Ph.D.
- [7] Hybridní cloud. In: ManagementMania.com [online]. Wilmington (DE) 2011-2017, 10.04.2017 [cit. 19.05.2017]. Dostupné z: <https://managementmania.com/cs/hybridni-cloud>



- [8] Microsoft API and Reference Catalog: MSDN. *Microsoft* [online]. Redmond: Microsoft, c2017 [cit. 2017-05-19]. Dostupné z: <https://msdn.microsoft.com/en-us/library>
- [9] NATHAN, Adam. *WPF 4.5 unleashed*. 1. Indianapolis: Sams Publishing, 2013. Unleashed. ISBN 06-723-3697-9.
- [10] TROELSEN, Andrew. *Pro C# and the .NET 4.5 framework*. 6. edition. Berkeley, Calif: APress, 2012. ISBN 14-302-4233-7.
- [11] WPF Tutorial | Data Binding Overview [online]. Christian Moser, 2011 [cit. 2017-05-19]. Dostupné z: <http://www.wpftutorial.net/DataBindingOverview.html>
- [12] Components of .Net Framework, CLR, CTS, CLS, Base Class Library. *DeveloperIn* [online]. c2016 [cit. 2017-05-19]. Dostupné z: <http://www.developerin.net/a/39-Intro-to-.Net-FrameWork/23-Components-of-.Net-Framework>
- [13] Developer's Guide to Microsoft Prism Library 5.0 for WPF. *Microsoft Developer Network* [online]. Redmond: Microsoft, 2014 [cit. 2017-08-28]. Dostupné z: <https://msdn.microsoft.com/en-us/library/gg406140.aspx>
- [14] Amazon S3. *Amazon Web Services* [online]. Seattle: Amazon, c2017 [cit. 2017-08-29]. Dostupné z: <https://aws.amazon.com/s3/>
- [15] IBM Cloud revenue jumps by 33 percent. *IBM* [online]. New York: IBM, 2017 [cit. 2017-08-29]. Dostupné z: <https://www.ibm.com/blogs/cloud-computing/2017/04/ibm-cloud-revenue-jumps-33-percent/>
- [16] Cloud Computing - Statistics & Facts. *Statista* [online]. Hamburg: statista, 2017 [cit. 2017-08-29]. Dostupné z: <https://www.statista.com/topics/1695/cloud-computing/>
- [17] *Duplicati* [online]. Duplicati, 2016 [cit. 2017-08-29]. Dostupné z: <https://www.duplicati.com/>
- [18] *Duplicati* [online]. Duplicati, 2016 [cit. 2017-08-29]. Dostupné z: <https://www.duplicati.com/>
- [19] *iCloud storage plans and pricing* [online]. Cupertino: Apple, 2017 [cit. 2017-08-29]. Dostupné z: <https://support.apple.com/en-us/HT201238>
- [20] Google Drive. *Google* [online]. California: Google, c2017 [cit. 2017-08-29]. Dostupné z: <https://www.google.com/drive/>

- [21] What is IaaS? *Microsoft Azure* [online]. Redmont: Microsoft, c2017 [cit. 2017-08-29]. Dostupné z: <https://azure.microsoft.com/en-us/overview/what-is-iaas/>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

GUI	Graphical User Inteface
WPF	Windows Presentation Foundation.
MVVM	Model-View-ViewModel
XAML	Extensible Application Markup Language
CLR	Common Language Runtime
XML	Extensible Markup Language
CLI	Common Language Infrastructure
CPU	Central Processing Unit
DI	Dependency Injection
DIC	Dependency Injection Container
DVD	Digital Versatile Disc

**SEZNAM OBRÁZKŮ**

Obrázek 1 Schéma úplného zálohování.....	13
Obrázek 2 Schéma rozdílového zálohování.....	14
Obrázek 3 Schéma přírůstkového zálohování .....	15
Obrázek 4 Logo služby DropBox .....	18
Obrázek 5 Logo služby Google Drive .....	19
Obrázek 6 Logo služby One Drive .....	20
Obrázek 7 Logo služby iCloud Drive .....	22
Obrázek 8 MultCloud uživatelské rozhraní .....	25
Obrázek 9 Rozhraní Duplicati .....	26
Obrázek 10 Diagram .Net Frameworku [12] .....	28
Obrázek 11 Schéma DataBindingu [11] .....	29
Obrázek 12 Schéma různých typů DataBindingu [8] .....	30
Obrázek 13 Ukázka vizuální stromové struktury .....	31
Obrázek 14 Aktivitní diagram obecného procesu zálohování .....	42
Obrázek 15 Aktivitní diagram zálohování.....	43
Obrázek 16 Wireframe hlavního okna aplikace .....	44
Obrázek 17 Diagram vztahů projektů (modulů).....	46
Obrázek 18 Hlavní okno aplikace.....	50
Obrázek 19 Backup path settings.....	52
Obrázek 20 Dialog pro výběr složky v cloudovém úložišti DropBox.....	53
Obrázek 21 BackupPlansView .....	55
Obrázek 22 Záložka info.....	56
Obrázek 23 Záložka Backups .....	57
Obrázek 24 Záložka Providers.....	57
Obrázek 25 Záložka Log.....	58
Obrázek 26 Záložka OperationQueue.....	58
Obrázek 27 NewProviderDialogView .....	59
Obrázek 28 Autorizační okno pro službu Dropbox .....	59
Obrázek 29 Správa aplikací ve webovém rozhraní služby Dropbox.....	60
Obrázek 30 Diagram třídy NotifyPropertyChanged.....	64
Obrázek 31 Diagram tříd BackupPlan a TimePlan.....	66

---

Obrázek 32 Class Diagram třídy BackupPlanService a rozhraní IBackupPlanService .....	66
Obrázek 33 Diagram tříd spravovaný službou BackupProviderManager .....	68
Obrázek 34 Class diagram třídy SecurityManager .....	72
Obrázek 35 Ukázka logu aplikace .....	74

**SEZNAM ZDROJOVÝCH KÓDŮ**

Zdrojový kód 1 Use case diagram aplikace .....	34
Zdrojový kód 2 Ukázka databázové a klientské třídy.....	47
Zdrojový kód 3 IBackupPlanService .....	48
Zdrojový kód 4 IBackupService .....	48
Zdrojový kód 5 Dependency injection.....	62
Zdrojový kód 6 Ukázka Backup Plan Service Singleton.....	62
Zdrojový kód 7 InfrastructureModule .....	62
Zdrojový kód 8 Ukázka práce s Unity Kontejnerem .....	63
Zdrojový kód 9 Bootstrapper .....	63
Zdrojový kód 10 Katalog modulů.....	64
Zdrojový kód 11 Ukázka třídy NotifyPropertyChanged .....	65
Zdrojový kód 12 Řetězení vlastností .....	65
Zdrojový kód 13 Inicializace pomocí kontejneru .....	67
Zdrojový kód 14 Vytvoření poskytovatele pro Dropbox.....	69
Zdrojový kód 15 Inicializace Dropbox modulu.....	69
Zdrojový kód 16 Uložení stavu providera .....	70
Zdrojový kód 17 zálohování souboru na službu Dropbox.....	70
Zdrojový kód 18 Příklad autorizačního url služby Dropbox .....	72
Zdrojový kód 19 Ukázka použití třídy SecurityManager .....	73

**SEZNAM TABULEK**

Tabulka 1 Ceník služby Dropbox .....	18
Tabulka 2 Ceník služby Google Drive .....	20
Tabulka 3 Ceník služby DropBox .....	21
Tabulka 4 Ceník služby iCloud [19].....	22
Tabulka 5 Popis nastavení záložky Backup plan settings.....	51
Tabulka 6 Tabulka nastavení Time Settings.....	53
Tabulka 7 Výsledek testu lokálního zálohování se vzorkem 2.....	75
Tabulka 8 Výsledek testu lokálního zálohování se vzorkem 4.....	76
Tabulka 9 Výsledky testů inkrementálního lokálního zálohování se vzorkem 2 .....	76
Tabulka 10 Výsledky testu zálohování vzorku 3 na službu Dropbox .....	77
Tabulka 11 Výsledky zálohování na službu Dropbox se vzorkem 4.....	78
Tabulka 12 Výsledek zálohování vzorku 3 na službu GoogleDrive .....	78

## SEZNAM PŘÍLOH

PŘÍLOHA P I: OBSAH PŘILOŽENÉHO DVD



## **PŘÍLOHA P I: OBSAH PŘILOŽENÉHO DVD**

Struktura adresářů přiloženého DVD je následující:

Solution – obsahuje kompletní solution pro Visual Studio 2017

App – obsahuje vytvořenou aplikaci

NuGet – obsahuje vytvořený NuGet

Tutorial – obsahuje návod na vytvoření pluginu pro aplikaci