



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**APLIKACE PRO ŘÍZENÝ PŘÍSTUP KE VZDÁLENÝM
DOKUMENTŮM PRO GNU/LINUX**

APPLICATION FOR CONTROLLED ACCESS TO REMOTE DOCUMENTS FOR GNU/LINUX

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN BERNARD

VEDOUCÍ PRÁCE

SUPERVISOR

RNDr. MAREK RYCHLÝ, Ph.D.

BRNO 2020

Zadání bakalářské práce



Student: **Bernard Jan**

Program: Informační technologie

Název: **Aplikace pro řízený přístup ke vzdáleným dokumentům pro GNU/Linux**
Application for Controlled Access to Remote Documents for GNU/Linux

Kategorie: Operační systémy

Zadání:

1. Seznamte se s požadavky týkající se zabezpečení přístupu k dokumentům v projektu Validované datové úložiště (VDU). Prozkoumejte možnosti virtuálních souborových systémů a integrace aplikací do desktopového prostředí v operačním systému GNU/Linux.
2. Navrhněte klientskou aplikaci pro VDU, která se připojí k úložišti a bude zpřístupňovat obsah úložiště pod zadaným přístupovým klíčem jako soubor virtuálního souborového systému s řízeným přístupem a správou verzí. Navrhněte automatické testy požadovaných vlastností takové aplikace.
3. Po konzultaci s vedoucím navrženou aplikaci implementujte včetně automatických testů.
4. Výsledek popište, vyhodnoťte a zveřejněte jako open-source.

Literatura:

- Interní dokumentace projektu Validované datové úložiště.
- VANGOOR, Bharath Kumar Reddy; TARASOV, Vasily; ZADOK, Erez. To FUSE or Not to FUSE: Performance of User-Space File Systems. In: 15th USENIX Conference on File and Storage Technologies (FAST 17). 2017, s. 59-72. Dostupné z: [https://www.usenix.org/conference/fast17/technical-sessions/presentation/vangoor]
- BURIHABWA, Dorian, et al. SGX-FS: Hardening a File System in User-Space with Intel SGX. In: 2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, 2018, s. 67-72. Dostupné z: [https://doi.org/10.1109/CloudCom2018.2018.00027]
- xdg-utils. *freedesktop.org* [online], 2019 [cit. 2020-10-26]. Dostupné z: [https://freedesktop.org/wiki/Software/xdg-utils/]

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a rozpracovaný bod 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rychlý Marek, RNDr., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 27. října 2020

Abstrakt

Práce se zaměřuje na synchronizaci a životní cyklus souborů stažených do uživatelského počítače ze vzdáleného validovaného datového uložště. Z analýzy trhu vyplynulo, že na trhu je velký výběr aplikací, ale rozdíly mezi nimi jsou relativně velké. Na základě požadavků a stanovené komunikace s validovaným datovým uložštěm byla navržena a implementována aplikace pro systém GNU/Linux. Testování aplikace probíhalo na dvou vybraných Linuxových distribucích s mock serverem zastupující vzdálené uložště.

Abstract

The aim of this thesis is synchronization and life cycle of files downloaded to user's computer from validated data storage. Based on market analysis there is large selection of applications in market but there are relatively large differences along them. Application was designed and implemented for GNU/Linux according to requirements and given validated data storage communication. The application was tested on two selected Linux distributions with mock server representing remote storage.

Klíčová slova

Validované datové uložště, VDU, Linux, souborový systém, HTTP, FUSE, C++, synchronizace, soubor, dokument, oprávnění

Keywords

Validated data storage, VDS, Linux, filesystem, HTTP, FUSE, C++, synchronization, file, document, permission

Citace

BERNARD, Jan. *Aplikace pro řízený přístup ke vzdáleným dokumentům pro GNU/Linux*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce RNDr. Marek Rychlý, Ph.D.

Aplikace pro řízený přístup ke vzdáleným dokumentům pro GNU/Linux

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana RNDr. Marka Rychlého, Ph.D. a uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Jan Bernard
6. května 2021

Poděkování

Rád bych poděkoval vedoucímu práce, doktorovi Markovi Rychlému, za vynikající vedení práce a jeho podnětné poznatky a rady. Také bych rád poděkoval všem svým kolegům, se kterými jsem procházel celé bakalářské studium. A v neposlední řadě děkuji všem svým přátelům a rodině za podporu a pochopení.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 3 |
| 2 | Motivace a existující řešení | 4 |
| 2.1 | Současná řešení | 4 |
| 2.2 | Google Drive | 5 |
| 2.3 | Dropbox | 5 |
| 2.4 | pCloud | 5 |
| 2.5 | Syncthing | 6 |
| 2.6 | Shrnutí | 6 |
| 3 | Návrh řešení a použité technologie | 7 |
| 3.1 | Případy užití | 7 |
| 3.2 | Definice REST API | 8 |
| 3.3 | Architektura | 9 |
| 3.3.1 | Diagram tříd | 10 |
| 3.4 | Technologie | 11 |
| 3.4.1 | GNU/Linux | 11 |
| 3.4.2 | C++/C | 12 |
| 3.4.3 | HTTP | 12 |
| 3.4.4 | OpenAPI | 13 |
| 3.4.5 | Mock server | 13 |
| 3.4.6 | Souborový systém v uživatelském prostoru (FUSE) | 13 |
| 3.4.7 | DBus | 14 |
| 3.4.8 | Keyring | 14 |
| 4 | Popis implementace a nasazení aplikace | 15 |
| 4.1 | Definice API | 15 |
| 4.2 | Mock server | 15 |
| 4.3 | Vyžívané cesty/soubory v Linuxové adresářové struktuře | 15 |
| 4.3.1 | Konfigurační soubor | 15 |
| 4.3.2 | SQLite databáze | 15 |
| 4.4 | Autentifikace | 15 |
| 4.5 | Práce se soubory | 15 |
| 4.5.1 | Otevření souboru v příslušné aplikaci | 15 |
| 4.6 | Nasazení aplikace | 15 |
| 4.6.1 | DPKG | 15 |
| 4.6.2 | RPM | 15 |
| 4.6.3 | Sestavení ze zdrojových souborů | 15 |

| | | |
|----------|--------------------------------|-----------|
| 5 | Testovací sada | 16 |
| 5.1 | Google Test | 16 |
| 6 | Závěr | 17 |
| | Literatura | 18 |
| A | Obsah přiloženého média | 20 |

Kapitola 1

Úvod

V době vysokorychlostního a stabilního internetu je využívání vzdálených uložišť běžnou záležitostí. Pro většinu uživatelů je to nástroj pro jednoduchou synchronizaci dat napříč několika zařízeními od mobilního telefonu po stolní počítač. Pokud vezmeme v úvahu pouze velká datová centra, jedná se pravděpodobně o nejlevnější a nejspolehlivější způsob uchování dat, protože tyto centra mají velkou kapacitu úložného prostoru a také zálohy na softwarové i hardwarové úrovni. Pod pojmem vzdálené uložení si nemusíme představit jen velká datová centra se stovkami serverů, může se jednat o relativně malé uložení ve firmě nebo dokonce o osobní/domácí řešení. Pro využívání osobních/firemních uložení se lidé uchylují v případech, je-li třeba uložit osobní nebo určitým způsobem citlivá data, která by nemohla být poskytnuta třetí straně. Nabízená řešení také nemusí poskytovat potřebnou funkcionalitu nebo jejich finanční model nesplňuje zákaznická kritéria.

V průběhu let byly vyvinuty protokoly řešící sdílení souborů jako FTP, NFS a mnohé další. Sdílení souborů je komplexní záležitostí a obsahuje velké množství parametrů. Každý protokol se zaměřuje jen na určité parametry jako zabezpečení a spolehlivost přenosu, oprávnění pro jednotlivé uživatele a soubory, synchronizace modifikovaných souborů apod nebo pojme daný parametr odlišným způsobem. V dnešní době je většina komerčně využívaných aplikací vystavěna na proprietárních protokolech nebo na protokolech umožňující obecnější použití. Jedním z nejpoužívanějších aplikačních protokolů bude Hypertext Transfer Protocol neboli HTTP.

Cílem této práce je prozkoumat aktuální nabídku a možnosti na poli vzdálených uložení a navrhnout aplikaci pro řízený přístup ke vzdáleným dokumentům pro platformu GNU/Linux. Aplikace bude zaměřena na řízený životní cyklus souborů s vynucenými oprávněními pro jednotlivé soubory daná vzdáleným uložení. Komunikace skrze HTTP protokol byla definována externím zadavatelem. Zdrojové kódy jsou dostupné na veřejném Github repozitáři.¹

¹<https://github.com/PlayerBerny12/VUT-IBT-Code>

Kapitola 2

Motivace a existující řešení

Popularita a využití vzdálených uložišť roste, protože lidé využívají více zařízení a potřebují mezi nimi jednoduchou synchronizaci nebo na jednom souboru potřebuje spolupracovat více lidí. Dalším podnětem využívání vzdálených uložišť je většího množství dat a potřeba tyto data efektivně sdílet a bezpečně uložit. Postupně se digitalizují další systémy například ve státní správě nebo dalších podnikatelských sektorech.

Na trhu je velké množství služeb zaměřujících se na různé klientské potřeby. Diverzita nabídky je až překvapivě velká. Většina se zaměřuje na ukládání a sdílení dat ve svém ekosystému, které jsou určené pro široké použití. Pokud jsou požadavky specifické, tak možnost vlastní konfigurace není možná. Pro částečnou úpravu nebo automatizaci procesů lze využít poskytované API, ve většině případů se jedná o variantu REST API. Poslední možností je vystavení obdobné služby z několika existujících aplikací nebo vytvoření nové.

V kontextu této práce jsou požadavky následující:

- Dodržování oprávnění i na klientském systému
 - read-only
 - write-only
 - read/write
- Životní cyklus souboru (stažení)
 - stažení
 - případná modifikace
 - smazání souboru po vypršení platnosti
- Autentifikace
 - uživatelské jméno a klientský certifikát
 - uživatelské jméno a heslo

2.1 Současná řešení

Pro bližší analýzu byl vybrán vzorek služeb a programů obsahující velké ekosystémy služeb po open source aplikace. Zkoumáno bylo několik parametrů, jako poskytovaná funkcionality pro jednotlivce a firmy, cena, integrace s ostatními systémy a aplikacemi apod. Obecně

nelze jednoznačně určit nejlepší uložisko, ale je možné poukázat na rozdílné vlastnosti a vyzdvihnout kladné. Koncový uživatel se následně může rozhodnout dle svých potřeb.

2.2 Google Drive

Pro synchronizaci obsahu z Google Drive na osobní počítač se využívá aplikace pojmenovaná Google File Stream. Je možné ji provozovat pouze na systémech Windows a Mac OS.[6] Pro Linuxové distribuce lze využít například neoficiální open source aplikaci Google Drive OCamlFUSE¹ využívající technologii FUSE, která bude popsána v následující kapitole 3.4.6. Google Disk poskytuje plně funkční webové rozhraní, ve kterém je možné dokumenty přímo upravovat bez nutnosti stahování. Problém nenastal ani v případě konkurenčních Microsoft Office dokumentů.

Google Workspace (dříve Google Suite) je možné pořídit v několika balíčcích. Například balíček Business Standard nabízí 2 TB uložisko za 10,40 EUR měsíčně. Balíček obsahuje další služby jako firemní email a schůzky až o 150 účastnících na Google Meet.[10]

Každý uživatel má vlastní disk s omezenou kapacitou podle balíčku. Na disku lze vytvářet standardní složkovou hierarchii a je možné sdílet jednotlivé soubory nebo obsah složek s ostatními Google uživateli. Verze Enterprise umožňuje vytváření dalších disků, na které je možné přiřadit seznam uživatelů. Každý uživatel má nastavenou jednu z 6 rolí, které vymezují jeho možnosti. K dispozici je API, která pokrývá veškeré možnosti webového rozhraní jako vytvoření souboru, sdílení atd.[9]

2.3 Dropbox

Dropbox má oficiální balíčky své aplikace pro Ubuntu a Fedoru, případně je možné si aplikaci zkompilovat ze zdrojových souborů. Podpora Windows a Mac OS je samozřejmostí. Z práce „Personal Cloud Storage Benchmarks and Comparison“ lze vyčíst, že Dropbox se zaměřuje na efektivitu přenosu a minimální zatížení sítě. Využívá menší množství TCP spojení oproti jeho konkurentům a také stejně jako Google Drive komprimuje data před odesláním.[2] Toto řešení nemusí dosahovat nejvyšší výkonosti, ale nezatěžuje tolik infrastrukturu.

Webové rozhraní je více orientováno na správu souborů a jejich historii. Nabízí jednoduché obnovení smazaných souborů nebo návrat k předchozí verzi. Nenabízí tolik možností jako Google Drive, ale upravovat dokumenty ve webovém prohlížeči lze také. Na výběr je mezi Microsoft Office Online nebo Google Worksapce. Sdílení souborů a složek je možné i s uživateli, kteří nemají Dropbox účet.

Balíček Plus za 9,99 EUR měsíčně dává uživateli přístup ke 2 TB uložisti. Firemní balíčky obsahují rozšíření pro lepší správu uživatelů, vytváření skupin a admin panel/konzole. Dropbox má také API nabízející dostatečnou funkcionalitu pro případnou integraci s jinými systémy apod.[7]

2.4 pCloud

Poskytovatel pCloud podporuje nejpoužívanější platformy, a to včetně mobilních. Za 9,99 EUR měsíčně dostane uživatel 2 TB uložisko. Jako jediný z poskytovatelů v analýze nabízí

¹<https://github.com/astrada/google-drive-ocamlfuse>

balíček s jednorázovou platbou za cenu 350 EUR. Jedná se o identický balíček, pouze forma platby se liší. Všechna data jsou přenášena přes šifrovaný kanál a všechny kopie souborů na pěti různých serverech jsou šifrovaná 256bitovým AES klíčem.[15]

Webové rozhraní je jednoduché, avšak oproti konkurentům má méně funkcí. Lze zobrazit náhled dokumentů, ale upravovat je nelze. Každý soubor má tzv. revize a je možné obnovit obsah souboru na vybranou revizi. Revize jsou uchovány po dobu jednoho roku. Soubory je možné sdílet i s uživateli, kteří nemají účet u pCloudu.

Pro firemní zákazníky je nabízený rozšířený management uživatelů a monitoring s logy aktivity jednotlivých uživatelů.

2.5 Syncthing

Open source aplikace Syncthing synchronizuje soubory peer-to-peer. Nejedná se o čistou peer-to-peer architekturu, protože jeden z uzlů může být nastaven jako server a veškeré data budou synchronizována vůči tomuto uzlu. Syncthing je možné provozovat na většině dnešních systémů jako Windows, Linux, BSD a Mac OS.[17]

Aplikace poskytuje webové rozhraní, které je určeno pouze pro nastavení parametrů jako discovery protokol pro objevování ostatních uzlů, jaké složky mají být synchronizovány nebo kolik verzí jednotlivých souborů má být uchováváno a mnoho dalšího. Velká přizpůsobivost umožňuje upravit fungování aplikace vlastním potřebám, na druhou stranu bude náročné udržovat systém s větším množstvím uživatelů vystavěný na této aplikaci. Jedná se tedy spíše o domácí řešení.

2.6 Shrnutí

Trendy na poli vzdálených uložišť určují velké firmy jako Google nebo Microsoft. Důkazem tohoto jevu je například integrace aplikací obou těchto firem do webového rozhraní Dropboxu. Aplikace One Drive od Microsoftu nebyla blíže analyzována, protože se jedná o službu pouze pro platformu Windows.

Všechny služby mají webové rozhraní, ke kterému je aplikace synchronizující obsah uložště občas braná jako doplněk. Sdílení nebo obnova předchozí verze souboru je vždy možná ve webové aplikaci, ale ne vždy jsou tyto akce dostupné i na desktopové verzi.

Aplikace na jejíž základě je vytvořena tato práce nebude synchronizovat celé uložště, ale bude pracovat jen s jednotlivými soubory. Uživatel na vyžádání stáhne právě jeden soubor, se kterým bude chtít uživatel v danou dobu pracovat (zobrazit nebo upravit obsah). Synchronizace celého uložště nebo obsahu jedné složky nebude možná. Jedná se spíše o náhradu možnosti zobrazovat a upravovat soubor přímo ve webovém prohlížeči jako to umožňuje například Google Drive.

Kapitola 3

Návrh řešení a použité technologie

Prvním krokem navrhovací fáze bylo seznámení se zadáním a požadavky. Externí zadavatel dodal podrobnou dokumentaci REST API pro validované datové uložště, která byla stěžejním dokumentem při návrhu. Jednalo se formální textový popis, který nebyl v žádném standardizovaném formátu. API obsahuje funkce pro ověření spojení, autentifikaci a práci se soubory. VDU nebylo při vzniku této práce k dispozici a proto byl pro potřeby vývoje vytvořen mock server na základě přepisu poskytnuté dokumentace do standardizovaného formátu OpenAPI.

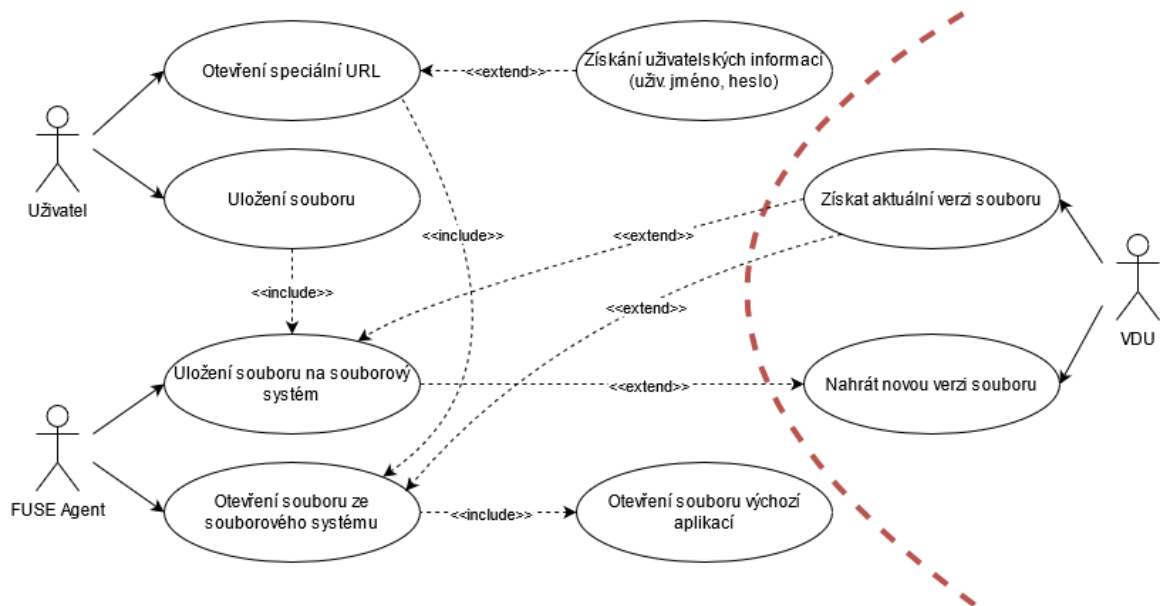
Pro vynucení práv jednotlivých souborů je standartní souborový systém nevyhovující, protože po stažení souboru do počítače uživatele ztrácí VDU kontrolu nad tímto souborem. Po konzultaci s vedoucím práce, dokterem Markem Rychlým, byla zvolena technologie FUSE neboli Filesystem in userspace. FUSE umožňuje vytvořit virtuální souborový systém a implementovat jednotlivé operace jako čtení, zápis atd. Tím je možné dosáhnout dostatečnou kontrolu nad oprávněními a obecně životním cyklem souboru v uživatelské počítači. Znalý uživatel by byl schopen tento systém obejít a přistoupit k souborům z hostovaného souborového systému. K takové operaci jsou potřeba oprávnění superuživatele `root`.

3.1 Případy užití

Ze zadání bylo možné přímo vytvořit diagram případu užití, pouze jedna věc nebyla specifikovaná. Webová aplikace musí předat desktopové přístupový token, pomocí kterého lze udělat potřebné API volání na VDU. V úvahu přišlo stažení „fake“ souboru obsahující pouze přístupový token. Druhou možností bylo přesměrování na speciální URL z VDU. Prohlížeč by na základě přesměrování rozpoznal, že má otevřít určenou aplikaci a předat ji token jako jeden z parametrů volání. Varainta se speciální URL působí na uživatele mnohem ucelenějším dojmem a nevytváří na souorovém systém dále nepotřebné soubory.

Diagram případu užití zachycuje celý systém jako celek a zobrazuje jednotlivé funkce a jejich návaznosti, které je třeba dále analyzovat a specifikovat v dalších fázích návrhu. V diagramu 3.1 vystupují tři aktéři a to uživatel, FUSE agent/démon a VDU. Implementace VDU není obsahem této práce, proto je oddělena červenou přerušovanou čarou. V diagramu není zanesená například autentifikace, na které závisí veškerá komunikace s VDU. Pro přehlednost byla tato část vypuštěna.

Všechny akce jsou spuštěny uživatelem a většina z nich vyvolá reakci FUSE démona. Jedná se například o uložení, otevření nebo přejmenování souboru ve virtuálním souborovém systému. Speciální případ nastává při otevření URL s vlastním protokolem, který má zavolat



Obrázek 3.1: Diagram případu užití

aplikaci, jež je k danému protokolu v systému přiřazena. Příkladem takového protokolu může být `mailto`, využívaný často u webových stránek. Přesměrování na adresu s tímto protokolem otevře správce elektronické pošty.

K námi definovanému protokolu pojmenovanému `vdu` je třeba mít v systému přiřazenou aplikaci, která s přístupovým tokenem uloženým v URL (`vdu://<přístupový token>`) stáhne žádaný soubor z VDU na virtuální souborový systém. Na základě této znalosti byly uskutečněny rozhodnutí při návrhu architektury.

3.2 Definice REST API

API obsahuje sedm metod pro ověření dostupnosti, autentifikaci a práci se soubory. Každá funkce má sadu možných návratových hodnot indikující jestli dané volání bylo úspěšné nebo proč dané volání selhalo.

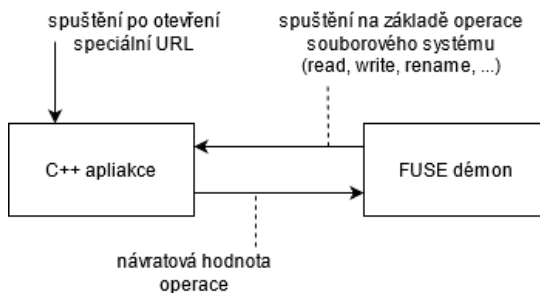
- `/ping` -
 - `GET /ping` - pro ověření dostupnosti VDU
- `/auth`
 - `GET /auth/key` - obnova autentifikačního tokenu
 - `POST /auth/key` - získání autentifikačního tokenu
 - `DELETE /auth/key` - zneplatnění autentifikačního tokenu
- `/file`
 - `GET /file/<file-access-token>` - stažení obsahu souboru
 - `POST /file/<file-access-token>` - nahrání obsahu souboru
 - `DELETE /file/<file-access-token>` - zneplatnění přístupového tokenu

Standardizace definice REST API byla vytvořena v OpenAPI standardu, pro který je dostupných mnoho nástrojů. Jedním z nich je open source nástroj Swagger, pomocí kterého můžete generovat html dokumentaci. Vygenerované rozhraní umožňuje i jednoduchou formu testování API. Pomocí standartizované definice bylo také možné rychle vytvořit mock server ve webové aplikaci Postman.

3.3 Architektura

Zvažovány byly dvě implementační varianty, jedna verze pracoval pouze s FUSE démonem a druhá rozdělila funkčnost do samostatné aplikace a FUSE démona. První variantu by bylo možné zpracovat dvěma obdobnými způsoby vedoucí ke stejnému řešení. FUSE démon je napsaný v jazyce C, takže jedno z nabízejících se řešení by byla implementace celé aplikace jedním jazyce. Druhým řešením by bylo vytvoření C++ knihovny, která by měla interface pro jazyk C. Tudíž by vytvořená knihovna mohla být konzumována FUSE démonem. Nejedná se však o nijak kriticky náročnou aplikaci na výkon, takže by varianta s C++ knihovnou byla upřednostňovaná, protože standartní knihovna jazyka C++ a jeho konstrukce umožňují jednodušší vývoj.

Rozdělení na dvě samostatné aplikace z nichž jedna je FUSE démon by mohlo přinést problémy v udržitelnosti takového systému. Pokud by jedna z aplikací nefungovala, celý systém by nemohl pracovat korektně. Hlavní aplikace by obsluhovala komunikaci s VDU a byla by vždy spuštěna voláním z FUSE démona nebo nepřímo uživatelem. Se znalostí získanou při návrhu diagramu případů užití víme, že je třeba mít registrovanou aplikaci obsluhující otevření speciální URL. I kdybychom vzali místo URL soubor se speciální koncovkou (např. *.vdu) obsahující přístupový token je stále třeba mít přiřazenou aplikaci pro obsluhu tentokrát otvírání souboru. Démon běžící na pozadí po celou dobu běhu systému nemůže být přiřazen jako aplikace obsluhující takové události. Na základě těchto okolností byla vybrána varianta se dvěma oddělenými aplikacemi.



Obrázek 3.2: Architektura

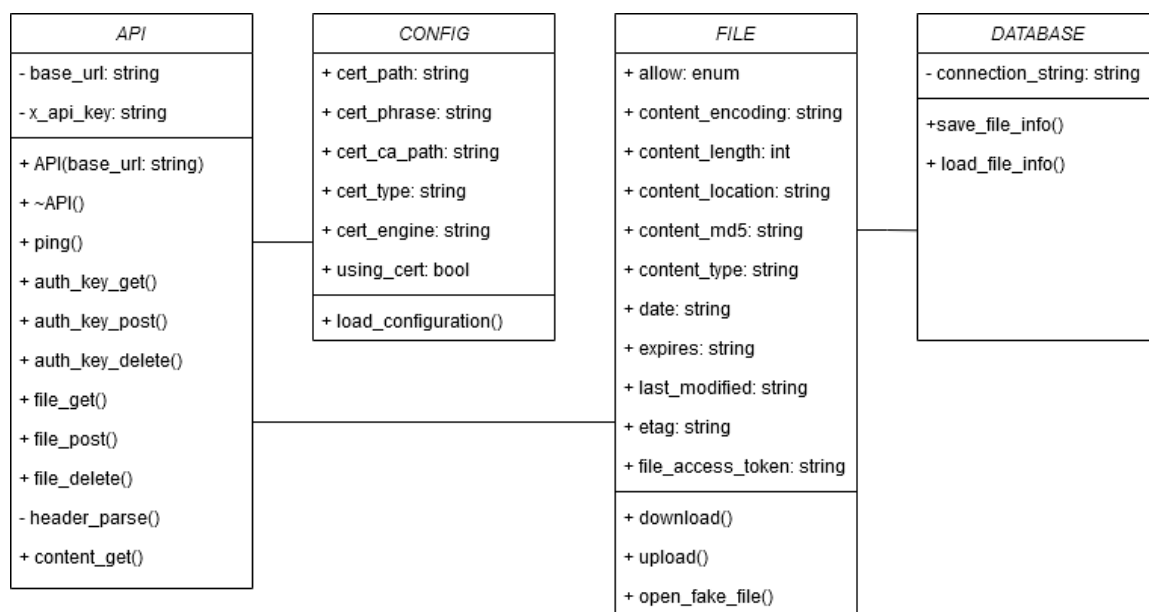
Pokud FUSE démon má vykonat akci, pro kterou nemá dostatečné informace, spustí jako další subprocess druhou hlavní aplikaci a počká na návratovou hodnotu, na základě které se rozhodne, jakým způsobem onu akci dokončí. Může se jednat o situace jako: je soubor určený jen pro četní nebo i zápis, je daný soubor stále validní (nevypršel datum expirace) apod.

V této variantě je FUSE démon velmi minimalistický a zbytek funkcionality je implementován v druhé aplikaci, která běží jen pokud je k tomu vyzvaná, narozdíl od neustále běžícího démona. Démon tedy po celou dobu běhu zabírá méně místa, protože nemusí mít načtené v paměti všechny potřebné knihovny. Tyto knihovny mohou být v paměti i

tak namapované, protože je mohou využívat jiné aplikace. Démon musí být spuštěný s oprávněními superuživatele `root` a menší množství kódu běžícího s vysokými oprávněními implikuje menší náchylnost na bezpečnostní chyby. Hlavní aplikace může běžet jen s právy přihlášeného uživatele.

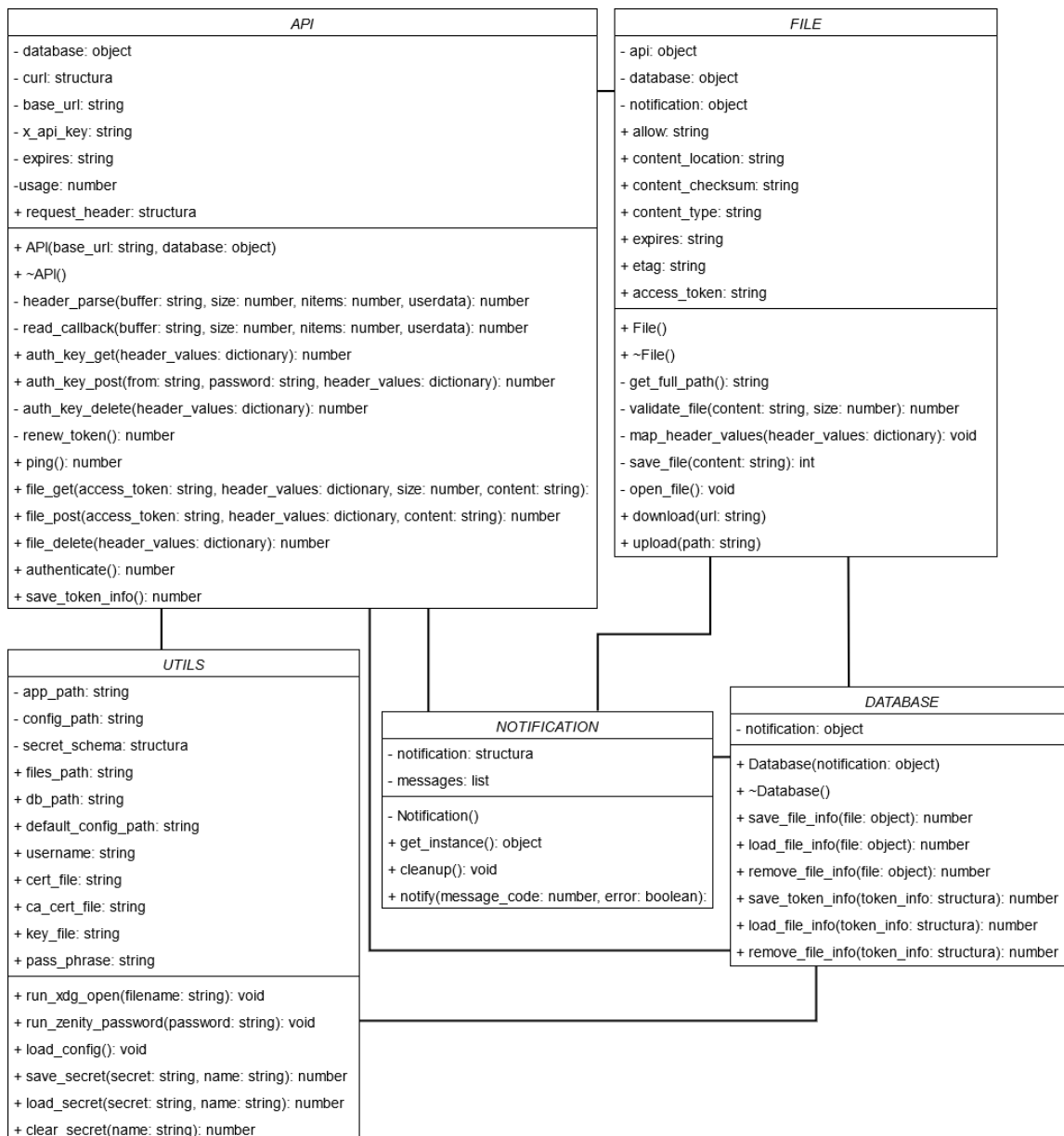
3.3.1 Diagram tříd

Aby implementace probíhala s možná nejmenším množstvím komplikací a nebylo třeba již uvažovat nad dalšími rozhodnutími, byl vytvořen diagram tříd 3.3, který se pokouší podrobně vizualizovat třídy s atributy a metodami, jež je třeba implementovat. Jedná se pouze o návrh, který se v průběhu vývoje bude měnit, ale pokud je návrh kvalitní a programátoři se jej sanží dodržet, neměl by být výsledek zásadně rozdílný.



Obrázek 3.3: Diagram tříd - návrh

Na obrázku 3.4 je možné vidět diagram tříd naimplementovaného programu. Na první pohled vypadá velmi rozdílně, po bližším prozkoumání lze vidět jednu novou třídu oproti návrhu. Jedná se o třídu zobrazující notifikace pro uživatele, která nemá vliv na hlavní funkcionalitu. Dále přibylo několik podpůrných funkcí a také přibíli některé atributy, případně se přejmenovali. Při implementaci nenastaly žádné zásadní potíže způsobené chybným nebo nepřesným návrhem, tudíž lze návrh označit za úspěšný.



Obrázek 3.4: Diagram tříd - implementace

3.4 Technologie

V této sekci budou popsány jednotlivé technologie, které ovlivnili tuto práci. Pro samotný návrh stačí elementární znalosti daných technologií, ale pro implementaci a jejich správné využití je třeba jejich bližší pochopení.

3.4.1 GNU/Linux

GNU je rekurzivní zkratka pro „GNU’s Not Unix!“. Jedná se o projekt, který měl za cíl vystavět nový operační systém kompatibilní s Unixem. Všechn kód vytvořený pod tímto

projektem je tzv. free software (svobodný software). Jedná se o filosofii vývoje software jejíž parafrázovaná definice je následující. Svobodný (free) software se vztahuje ke svobodě, ne k ceně. Slovo svobodný (free) by mělo být bráno ve významu svobodný projev (free speech) nikoliv pivo zdarma (free beer) [5].

V projektu GNU postupně vznikali jednotlivé softwareové balíky, které měli ve výsledku poskládat nový operační systém. Pod projektem GNU vnikli aplikace jako překladač GCC, textový formátovač TeX, terminálový shell Bash a mnohé další. Na počátku 90. let byl systém téměř hotový, jediné co scházelo bylo jádro. Ve stejném období dokončoval Linus Torvalds své jádro operačního systému nazývané Linux.[20][16]

Tak vznikla myšlenka tyto dva projekty spojit a vnikl GNU/Linux. Balíčky projektu GNU umožňovali práci se systémem postaveným nad Linuxovým jádrem. Všechny dnešní distribuce jako jsou Debian, Fedora a další jsou distribuce systému GNU/Linux. Linux je název pouze jádra pro operační systém, který zevšeobecněl a je tímto názvem chybně označováno celý systém. Projekt GNU dodnes vyvíjí své jádro pojmenované GNU Hurd, se kterým by vznikl operační systém GNU, což byl počáteční cíl projektu.[16]

3.4.2 C++/C

Jazyk C patří do skupiny dlouho používaných programovacích jazyků. Se svojí bohatou historií sahající až do 70. let 20. století velmi ovlivnil další nástupnické procedurální jazyky jako C++, C#, Java atd. V roce 1989 byl jazyk standardizován jako ANSI C a následně přišli ISO standardy označované jako C99, C11, C17. Patří mezi vysokoúrovňové programovací jazyky, přestože se jedná o relativně jednoduchý programovací jazyk obsahující malé množství konstrukcí.[11]

Vývoj C++ je oproti C dynamičtější, což můžeme vidět na jednotlivých ISO standardech C++98, C++03, C++11, C++14, C++17 a C++20. Jazyk C++ se v poslední dekádě velmi proměnil a období od standardu C++11 je označováno za dobu „moderního C++“. V posledním standardu přibyla podpora konceptů nebo rozsahů.[12]

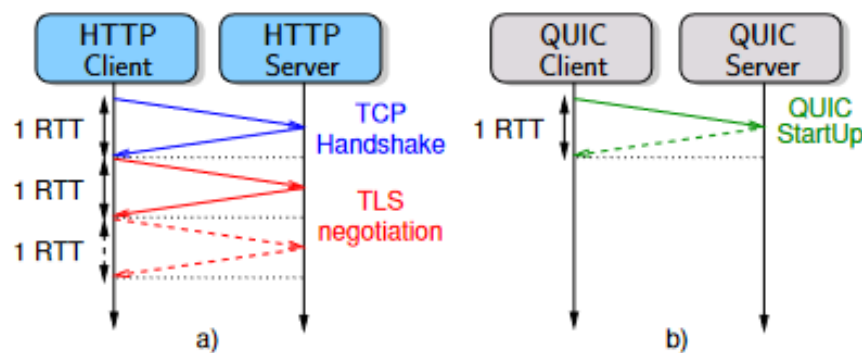
Při využití překladačů GCC nebo G++ lze využít tzv. GNU rozšiřující standardy. Poté je možné využít některé konstrukce, které nejsou specifikované v oficiálním ISO standardu. V případě jazyka C se jedná o standardy pojmenované GNU99, GNU11 a GNU17, v případě C++ se jedná o GNU++11 a jemu podobné.

3.4.3 HTTP

Hypertext Transfer Protokol je protokol aplikační vrstvy TCP/IP stacku. Byl vytvořen pro přenášení hypertextových souborů v 90. letech 20. století a dnes se jedná o nejrozšířenější aplikační protokol. Poslední stabilní verze protokolu je označena jako HTTP/2 vydaná v roce 2015. Aktuálně je ve fázi specifikace nástupce pojmenovaný HTTP/3.

HTTP v čisté formě se dnes moc nepoužívá, ale využívá se ve spojení s TLS. Při komunikaci se nejprve vytvoří TCP spojení a poté šifrovaný kanál, skrze který je HTTP bezpečně přenášeno. Toto řešení není nejefektivnější, protože vznikalo postupně a spojovali se technologie, které nebyly od počátku navrženy pro tento účel. Ve většině případů uživatel nepozná co se děje pod povrchem, ale takovéto spojení například trpí dlouhou časovou prodlevou, než se ustálí pro přenášení dat. Tento problém má vyřešit právě nová verze HTTP.[4]

Nová verze využívá nový protokol QUIC, který využívá protokol UDP namísto TCP jako tomu bylo u předchozích verzích HTTP. Na obrázku 3.5 je možné vidět, že navázání komunikace je ze tří výměn informací zredukováno jen na jednu. Formát hlavičky by měl zůstat nezměněný a tudíž je zpětně kompatibilní.



Obrázek 3.5: Porovnání HTTP verze 2 (a) a 3 (b). Převzato z „HTTP over UDP: An Experimental Investigation of QUIC“[4]

Pro tento projekt je důležitá práce s datumy a jejich formáty v HTTP hlavičce. RFC7231 specifikuje tři formáty datumů, z nichž dva jsou označené jako zastaralé, ale jsou stále validní.[8]

- Upřednostňovaný formát
 - Sun, 06 Nov 1994 08:49:37 GMT (IMF-fixdate formát)
- Zastaralé formáty
 - Sunday, 06-Nov-94 08:49:37 GMT (RFC 850 formát)
 - Sun Nov 6 08:49:37 1994 (ANSI C asctime() formát)

3.4.4 OpenAPI

Jedná se o standard pro definování metod REST API, který má v čase psaní této práce poslední verzi 3.0.3. Zápis je možný ve dvou jazycích a to JSON a YAML. Oba je možné strojově zpracovat a přitom jsou pro člověka stále čitelné.

Specifikace je velmi obsáhlá a umožňuje popsat API velmi detailně a přitom efektivně. Například je možné vytvářet šablony datových struktur, které jsou používány jako vstupní nebo naopak návratová hodnoty jednotlivých funkcí. V definici je pak možné na tuto šablonu odkázat. Struktura souboru je hierarchická a skládá se z jednotlivých objektů. Každý objekt má povinné a volitelné atributy. Definovat tedy lze hlavičky požadavku, URL query parametry, návratové kódy a k nim přiřazená návratové hlavičky a odpovědi. Toto byl jen stručný minimální výpis, co je možné pomocí tohoto standardu definovat.[14]

3.4.5 Mock server

- testovací vzor mock serveru
[1]

3.4.6 Souborový systém v uživatelském prostoru (FUSE)

- co je to fuse - jak funguje (démon, blokové zařízení, ...) - operace, které je možné implementovat - minimálně potřebné operace pro fungování fuse

[18] [3]

3.4.7 DBus

- co to je - jaké systémové služby je možné přes to ovládat

3.4.8 Keyring

- co to je - jak to funguje - implementace (gnom-keyring, kde-wallet, linux-keyring) - odlišnosti implementací

[13]

Kapitola 4

Popis implementace a nasazení aplikace

- Seznam použitých knihoven

4.1 Definice API

4.2 Mock server

4.3 Vyžívané cesty/soubory v Linuxové adresářové struktuře

4.3.1 Konfigurační soubor

4.3.2 SQLite databáze

4.4 Autentifikace

4.5 Práce se soubory

4.5.1 Otevření souboru v příslušné aplikaci

[19]

4.6 Nasazení aplikace

4.6.1 DPKG

4.6.2 RPM

4.6.3 Sestavení ze zdrojových souborů

Kapitola 5

Testovací sada

- Testy pro funkce třídy API

5.1 Google Test

Kapitola 6

Závěr

Literatura

- [1] BLOOM, J. *MockServer* [online]. Mock Server, 2020 [cit. 2021-04-28]. Dostupné z: <https://www.mock-server.com/#what-is-mockserver>.
- [2] BOCCHI, E., DRAGO, I. a MELLIA, M. Personal Cloud Storage Benchmarks and Comparison. *IEEE Transactions on Cloud Computing*. 2017, sv. 5, č. 4, s. 751–764. DOI: 10.1109/TCC.2015.2427191. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/7096995>.
- [3] BURIHABWA, D., FELBER, P., MERCIER, H. a SCHIAVONI, V. SGX-FS: Hardening a File System in User-Space with Intel SGX. In: *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. 2018, s. 67–72. Dostupné z: <https://ieeexplore.ieee.org/document/8590996>.
- [4] CARLUCCI, G., DE CICCIO, L. a MASCOLO, S. HTTP over UDP: An Experimental Investigation of QUIC. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. Association for Computing Machinery, 2015, s. 609–614. DOI: 10.1145/2695664.2695706. ISBN 9781450331968. Dostupné z: <https://doi.org/10.1145/2695664.2695706>.
- [5] *Co je to svobodný software?* [online]. gnu.org, říjen 2019 [cit. 2021-05-03]. Dostupné z: <https://www.gnu.org/philosophy/free-sw.html>.
- [6] *Deploy Google Drive for desktop* [online]. Google, 2021 [cit. 2021-04-21]. Dostupné z: <https://support.google.com/a/answer/7491144?hl=en>.
- [7] *Dropbox* [online]. Dropbox, 2021 [cit. 2021-04-21]. Dostupné z: <https://www.dropbox.com/?landing=dbv2>.
- [8] FIELDING, R. a RESCHKE, J. *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content* [Requests for Comments]. RFC Editor, červen 2014 [cit. 2021-04-25]. DOI: 10.17487/RFC7231. Dostupné z: <https://www.rfc-editor.org/info/rfc7231>.
- [9] *Google Drive for developers - API Reference* [online]. Google, 2021 [cit. 2021-04-21]. Dostupné z: <https://developers.google.com/drive/api/v3/reference>.
- [10] *Google Workspace* [online]. Google, 2021 [cit. 2021-04-21]. Dostupné z: <https://workspace.google.com/>.
- [11] *History of C* [online]. cppreference.com, březen 2021 [cit. 2021-05-04]. Dostupné z: <https://en.cppreference.com/w/c/language/history>.
- [12] *History of C++* [online]. cppreference.com, březen 2021 [cit. 2021-05-04]. Dostupné z: <https://en.cppreference.com/w/cpp/language/history>.

- [13] *What is: Linux keyring, gnome-keyring, Secret Service, and D-Bus* [online]. RTFM: Linux, DevOps, and system administration, 12. července 2019 [cit. 2021-05-02]. Dostupné z: <https://rtfm.co.ua/en/what-is-linux-keyring-gnome-keyring-secret-service-and-d-bus/>.
- [14] *OpenAPI Specification* [online]. SmartBear Software, 2021 [cit. 2021-05-05]. Dostupné z: <https://swagger.io/specification/>.
- [15] *PCloud* [online]. pCloud, 2021 [cit. 2021-04-21]. Dostupné z: <https://www.pcloud.com/eu>.
- [16] STALLMAN, R. *Linux a systém GNU* [online]. gnu.org, červenec 2020 [cit. 2021-05-03]. Dostupné z: <https://www.gnu.org/gnu/linux-and-gnu.cs.html>.
- [17] *Syncthing* [online]. Syncthing, 2021 [cit. 2021-04-21]. Dostupné z: <https://syncthing.net/>.
- [18] VANGOOR, B. K. R., TARASOV, V. a ZADOK, E. To FUSE or Not to FUSE: Performance of User-Space File Systems. In: *15th USENIX Conference on File and Storage Technologies (FAST 17)*. 2017, s. 59–72. Dostupné z: <https://www.usenix.org/conference/fast17/technical-sessions/presentation/vangoor>.
- [19] *Xdg-utils* [online]. freedesktop.org, 2019 [cit. 2021-04-20]. Dostupné z: <https://freedesktop.org/wiki/Software/xdg-utils/>.
- [20] YALTA, A. T. a LUCCHETTI, R. The GNU/Linux platform and freedom respecting software for economists. *Journal of Applied Econometrics*. 2008, s. 279–286. DOI: <https://doi.org/10.1002/jae.990>. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.990>.

Příloha A

Obsah přiloženého média