VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021 Erik Chovanec

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ŘÍZENÍ PŘÍSTUPU K DATŮM V CLOUDU
CLOUD-BASED DATA ACCESS CONTROL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Erik Chovanec
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. Jan Hajný, Ph.D.
SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Erik Chovanec ***ID:*** 211793
Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Řízení přístupu k datům v cloudu

POKYNY PRO VYPRACOVÁNÍ:

Téma je zaměřeno na analýzu současných open-source nástrojů pro pokročilé řízení přístupu ke cloudovým úložištím a na realizaci aplikace, která chrání důvěrnost a integritu dat během jejich přenosu i archivace v cloudu. Výstupem projektu bude vlastní webová aplikace, která bude schopna data bezpečně přenést na cloudové úložiště, zajistit jejich dlouhodobou archivaci a zajistit řízení přístupu pro více uživatelů, včetně řízení přístupu na základě skupinových oprávnění. Aplikace bude otestována v praktickém nasazení, zdokumentována a podrobena zátěžovým testům.

DOPORUČENÁ LITERATURA:

[1] MENEZES, Alfred, Paul C. VAN OORSCHOT a Scott A. VANSTONE. Handbook of applied cryptography. Boca Raton: CRC Press, c1997. Discrete mathematics and its applications. ISBN 0-8493-8523-7.

[2] Github: Cryptomator [online]. 2020 [cit. 2020-09-04]. Dostupné z: https://github.com/cryptomator/cryptomator

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 31.5.2021

Vedoucí práce: doc. Ing. Jan Hajný, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Táto práca sa zaoberá problematikou cloudových úložísk. Jej cieľom je realizácia cloudového úložiska schopného poskytovať požadované funkcionality, ktorými sú napríklad riadenie prístupu viacerých užívateľov, skupinové práva, dlhodobá archivácia súborov, bezpečný prenos dát na cloudové úložisko, ochrana integrity a dôvernosti dát počas prenosu aj ich uloženia. K dosiahnutiu tohto cieľa sa vyhodnocuje vhodnosť použitia služieb poskytovateľov verejných cloudových úložísk. Na základe vyhodnotenia sa prišlo k záveru, že tieto služby nie sú vhodné pre ukladanie veľmi citlivých dát. Ďalej sa porovnávajú výhody spojené s použitím softvéru šíreného pod licenciami typu open-source a free-software. Aj napriek vysokým počiatočným nákladom na infraštruktúru a nutnosti vynaložiť veľké úsilie k prevádzke vlastnej infraštruktúry s cloudovým úložiskom, vychádza lepšia možnosť prevádzkovať vlastnú infraštruktúru. Najmä v prípade využívania takéhoto úložiska k ukladaniu citlivých údajov. Ďalej sa tu vykonáva extenzívna analýza existujúcich open-source a free-software riešení. Vo finále je vybratá aplikácia Nextcloud, nakoľko poskytuje väčšinu potrebnej funkcionality. Práca sa venuje inštalácií a ukážke funkcionalít Nextcloudu. Sú tu vykonané veľmi základné záťažové testy, ktoré hovoria hlavne o efektivite odbavovať úspešne paralelné požiadavky. V poslednej časti sa rieši implementácia chýbajúcej funkcionality Nextcloudu, ktorá je nutná k dosiahnutiu zadania. Touto funkcionalitou je dlhodobá archivácia súborov. Pod týmto pojmom sa myslí možnosť archivovať súbory a zabezpečiť dôkazy o zachovaní dôvernosti a integrity týchto súborov počas dlhej doby existencie. Obsahuje teoretický návrh, ktorý sa snaží dosiahnuť tento cieľ. Následne je popísaná implementácia daného návrhu vo forme doplnkového systému. Archivácia samotných súborov je vyvolávaná upravenou aplikáciou, ktorá je ako rozšírenie Nextcloud serveru. Na základe tohto vyvolania sa spustí proces archivácie. Archivácia vytvorí kópiu archivovaného súboru na oddelenom úložisku. Získa dôkazy o jeho integrite v čase uloženia a v pravidelných časových intervaloch tieto dôkazy posilňuje získaním nového. Tieto časové intervaly sú v rokoch. Počas dlhej doby môžu kryptografické protokoly degradovať na kvalite. Tomuto javu sa snaží tento koncept zabrániť.

KĽÚČOVÉ SLOVÁ

cloudové úložisko, riadenie prístupu, licencie, šifrovanie, archivácia, archivačný systém

ABSTRACT

This thesis is dealing with problematic of cloud storage. The goal of this thesis is the implementation of cloud storage, that will be able to provide the required functionality. These functionalities are access control of multiple users, group rights, long-term file archiving, secure data transmission to cloud storage, protection of data integrity and confidentiality during transmission and storage. For achieving this goal, there is an evaluation of suitability using public cloud providers. Based on this evaluation, there was a conclusion, that these services are not suitable for storing highly sensitive data. Next is an evaluation of advantages associated with using software that is licensed under a free or open-source software license. Even though there are high start-up costs on infrastructure and there is a necessity to make a huge effort to run custom infrastructure with cloud storage, it is a much better option. Especially if it will be used for storing sensitive data. The following chapter describes an extensive analysis of existing open-source and free-software solutions. In the final part of this chapter, the application Nextcloud has been selected since it provides the most of required functionalities. The thesis also contains an installation of Nextcloud and shows its main functions. There are basic load tests, which are telling us about the efficiency of successfully dealing with parallel requests. In the last section of this thesis, we are dealing with the implementation of missing functionality in Nextcloud. This functionality is necessary to achieve our goal, which is long term archiving of files. This term means functionality that will enable the possibility to archive files with evidence about preservation integrity and confidentiality of archived files during a long period of their existence. This part of the thesis contains a theoretical design that aims to accomplish this goal. Description of the implementation follows after theoretical design. This functionality is implemented in form of a dedicated system. Archiving of files is triggered by a modified application, used in the Nextcloud server. This process will create a copy of the file to be archived on a separate hard drive. It will obtain evidence about integrity when storing the file and it will strengthen this evidence by obtaining a new one in regular periods. This happens on yearly basis. The quality of cryptographic protocols can degrade after a long period. This concept is trying to prevent this phenomenon.

KEYWORDS

cloud storage, access control, license, encryption, archiving, archivation system

CHOVANEC, Erik. *Řízení přístupu k datům v cloudu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 94 s. Bakalárska práca. Vedúci práce: doc. Ing. JAN HAJNÝ, Ph.D.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora:	Erik Chovanec
VUT ID autora:	211793
Typ práce:	Bakalárska práca
Akademický rok:	2020/21
Téma závěrečnej práce:	Řízení přístupu k datům v cloudu

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Rád by som poďakoval vedúcemu tejto práce pánovi doc. Ing. Jánovi Hajnému Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

This thesis was created as part of the key activity KA6 - Individual teaching and involvement of students of bachelor's and master's degree programs in research within the project OP RDE Creating double-degree doctoral study program Electronics and Information Technology and creating doctoral study program Information Security reg. no. CZ.02.2.69/0.0/0.0/16_018/0002575

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

V tejto práci sa rieši realizácia zabezpečeného cloudového úložiska s riadeným prístupom k dátam v cloude. Z dôvodu denne sa zvyšujúcej informatizácie spoločnosti, ľudia disponujú stále väčším objemom dát. Tie veľakrát obsahujú citlivé informácie, ktoré môžu byť nekalím spôsobom zneužité. Dáta v dnešnej spoločnosti majú stále väčšiu hodnotu nielen pre firmy či jednotlivcov, ale aj pre hackerov. Ľudia, ktorí sa neoprávnene zmocnia takýchto dát ich môžu predať alebo nimi vydierať majiteľov týchto dát. Toto vedie k trendu zvyšujúcej sa kyberkriminalite. Kyberkriminalita vedie k zvýšeniu dopytu ľudí aj firiem po službách, ktoré im vedia poskytnúť zvýšenie zabezpečenia týchto dát. S rastúcim množstvom dát sa vyskytuje aj potreba ich niekde ukladať. Popularita cloudových technológií a ich podiel na trhu veľmi rýchlo rastie, nakoľko ponúkajú mnoho benefitov oproti lokálnemu ukladaniu dát. Takisto nastáva otázka, či vôbec existujú vhodné úložiská pre ukladanie citlivých dát a či existujú úložiská, kde je možné zabezpečiť integritu a dôvernosť nahraných súborov počas dlhej doby uloženia.

Hlavným cieľom práce je zrealizovať zabezpečené cloudové úložisko, ktoré bude chrániť dáta pred ich neoprávneným získaním, úpravou alebo poškodením. Tieto dáta musia byť chránené počas celej doby existencie. To znamená, že musia byť chránené ako počas prenosu na cloudové úložisko, tak aj počas doby existencie na tomto úložisku. Úložisko okrem bezpečnostných funkcií musí umožňovať dlhodobé archivovanie dát a možnosti zdieľania zabezpečených dát rôznym oprávneným subjektom. Požadovanými vlastnosťami je teda autentizácia užívateľa, zabezpečený prenos dát na úložisko, šifrované uloženie dát na úložisku, dlhodobá archivácia dát a zabezpečenie riadeného prístupu k týmto dátam.

V práci sa nachádza teoretický úvod k častým pojmom spojeným s cloudovými úložiskami. Ďalej sa skúma otázka ohľadom vhodnosti využívania služieb verejných cloudových poskytovateľov k ukladaniu veľmi citlivých dát. Tieto zistené poznatky sa porovnávajú s využívaním produktov šírených pod licenciami typu open-source a free softver. Súčasťou dosiahnutia cieľov práce je nutná analýza možných riešení za použitia existujúcich produktov, ktoré chránia dáta pri uložení na externé cloudové úložisko ako aj produktov, ktoré samotné poskytujú službu cloudového úložiska a zabezpečeného prístupu k nemu. Analýza sa venuje predovšetkým produktom, ktoré sú poskytované pod licenciami typu open source alebo free software. Po dôkladnej analýze možných riešení sa v testovacom prostredí zrealizuje to najvhodnejšie. Posledná časť práce sa zaoberá dosiahnutím nutnej funkcionality cloudového systému, ktorú s najväčšou pravdepodobnosťou vhodne neimplementuje žiadne zo známych cloudových úložísk.

1 DEFINÍCIE POJMOV

V tejto kapitole sa preberú niektoré základné teoretické pojmy k realizácii stanovených cieľov.

1.1 Cloudové úložisko

Cloudové úložiská ponúkajú jednoduchý spôsob ukladania a presúvania dát. Umožňujú jednotlivým osobám ako aj firmám držať ich dáta uložené u poskytovateľa cloudovej služby. Vďaka tomu majú možnosť ich získať kedykoľvek a kdekoľvek. Cloudové úložiská môžu tiež slúžiť na dlhodobú archiváciu dát pre dáta, ku ktorým netreba často pristupovať. Častejšie sa však využíva k spolupráci medzi rôznymi skupinami ľudí, ako povedzme tímov v spoločnosti. Cloudové úložiská dovoľujú klientskemu zariadeniu ako počítaču, tabletu alebo smartfónu získať alebo poslať dáta na vzdialený dátový server. Dáta sú zvyčajne ukladané na viacerých serveroch, aby sa zabezpečila prístupnosť dát aj tam, kde by bol náhodou jeden server vypadnutý.

Cloudové úložiská väčšinou fungujú na princípe virtualizovanej infraštruktúry, ktorá poskytuje takmer okamžitú elasticitu a škálovateľnosť. Cloudové dáta sú väčšinou uložené v logických blokoch, ktoré sú fyzicky rozložené medzi rôznymi servermi. Servery môžu byť lokalizované na rôznych miestach. Často sa vyžívajú datacentrá, ktoré sú poskytované a spravované treťou stranou. Cloudové služby využívajú RESTFUL API, čo je protokol zabezpečujúci ukladanie objektov a súborov s ich metadátami. Ako jediný objekt má pridelené ID. Keď užívateľ potrebuje získať dáta zo serveru, vyšle požiadavku z ID daného objektu, na základe ktorej sa vyskladá obsah so všetkými metadátami, autentizáciou a zabezpečením.

V posledných rokoch poskytovatelia objektových úložísk pridali funkcionality súborových systémov. Prístupový bod cloudového úložiska vie poskytovať emuláciu súborového systému ako grafický prístup k ich objektovému ukladaniu. Toto často umožňuje aplikáciám sprístupniť dáta aj bez podpory objektového protokolu. Všetky aplikácie pre zálohovanie dát podporujú protokol objektového ukladania [1].

1.1.1 Typy cloudových úložísk

V tejto kapitole sa budeme venovať trom hlavným modelom prístupu ku cloudovému úložisku. Konkrétne sú to verejný, privátny a hybridný cloud.

Verejný cloud

Ako z názvu plynie, je to cloud, ktorý slúži pre poskytovanie služieb širokej verejnosti. Sú prístupné z internetu a založené na modeli „pay as you go“. Model znamená,

že si človek zaplatí za využívanie určitej služby a nemôže ju využiť nad rámec toho, čo si zaplatil. V tomto prípade užívateľ nemusí disponovať žiadnou infraštruktúrou k prevádzke cloudu, a tak nad ňou nemá ani žiadnu kontrolu a dáta sa môžu nachádzať kdekoľvek [2].

Privátny cloud

Hlavným rozdielom medzi verejným a privátnym cloudom je v prístupe k nemu. Pri verejnom sa k nemu vie dostať akákoľvek osoba, ktorá si zaň zaplatí. Privátny cloud slúži väčšinou v rámci nejakej organizácie a je možné sa naň dostať len zo siete organizácie. Organizácia musí mať potrebnú infraštruktúru na prevádzkovanie takéhoto cloudového úložiska, čo zvyšuje jeho cenu a náročnosť na údržbu. Hlavným benefitom je, že daný spôsob riešenia poskytuje väčšiu kontrolu nad dátami, ich bezpečnosťou a celkovými možnosťami prispôsobenia služby [2].

Hybridný cloud

Jedná sa o kombináciu predošlých dvoch riešení, kde spoločnosť rozdelí, ktoré dáta bude kde udržiavať. Napríklad citlivé dáta, kde je dôležitá ich bezpečnosť, bude držať na privátnom cloude a menej citlivé dáta na verejnom. Toto riešenie má výhody oboch vyššie spomenutých systémov a je cenovo efektívne. Hybridné cloudové riešenie ale nie je najjednoduchšie a prináša náročnosť na implementáciu. Je potrebné zabezpečiť integráciu verejného a privátneho úložiska a orchestráciu dát medzi nimi [1].

1.2 Zabezpečené dáta

Zabezpečené dáta sú dáta, na ktoré boli aplikované opatrenia, ktoré by mali zabrániť neoprávnenému prístupu k informáciám. Informácie obsahujú jej modifikácie alebo poškodenia. K zabezpečeniu dát nám slúži bezpečnostný model, ktorému sa hovorí „CIA triáda“. Hlavné prvky tohto bezpečnostného modelu sú nasledovné:

- confidentiality (dôvernosť) - dostupnosť len overenými užívateľmi,
- integrity (integrita) – zabezpečenie, že informácia je spoľahlivá a neupravená,
- availability (prístupnosť) - zabezpečenie prístupnosti k dátam, ak je to potrebné.

Tieto prvky je potrebné zabezpečiť tak, aby dáta mohli byť považované za zabezpečené. To je možné docieliť správnym použitím kryptografických protokolov. Záleží od cieľov a povahy organizácie ako zabezpečí dané prvky. Je možné uprednostňovať niektoré prvky na úkor iných, napríklad dôvernosť a integritu na úkor rýchlosti prístupu k dátam.

1.3 Archivácia dát

Archivovanie dát je proces, kde sa dáta presunú z hlavného úložiska na vedľajšie. Vedľajšie úložiská sú spravidla cenovo efektívne, pretože nepotrebujú mať veľkú rýchlosť zápisu a čítania. Archivácia dát sa deje v bode, keď sa dáta nejakú dobu aktívne nepoužívajú. Organizácie si vedia sami určiť kritéria pre dáta, ktoré budú archivovať. Samotná archivácia sa často deje automatizovane. Dáta v archívoch ostanú aj naďalej prístupné. Systém archivácie dát zvyšuje efektivitu využitia zdrojov danej organizácie tým, že hlavné úložisko dát zbavíme nadbytočnej záťaže od dát, ku ktorým sa často nepristupuje a nezapisuje sa do nich nič. Jednoduchšie sa robia aj zálohy systému. Systém, ktorý archivuje dáta musí byť schopný ochrániť integritu dát po dobu ich existencie na ňom [3].

1.4 Riadený prístup

Riadený prístup v informačných systémoch je bezpečnostná technika, ktorá zabezpečuje, že sa k dátam a funkcionalitám v systéme dostanú len oprávnené osoby. Je to jeden z najdôležitejších prvkov zabezpečenia systémov a je nevyhnutnou súčasťou všetkých organizácií. Pod pojem *riadenie prístupu* sa všeobecne zahŕňa fyzický a logický riadený prístup. Fyzický riadený prístup sa stará o to, aby zabezpečil kontrolu nad fyzickými prostriedkami. Mal by zabezpečiť ochranu priestorov v organizácii proti prístupu neoprávneným osobám, ktoré by mohli narušiť bezpečnosť systému. Logický riadený prístup rieši zabezpečenie prístupu k počítačovým sieťam, súborovým systémom a hlavne k dátam. To zahŕňa proces autentizácie a autorizácie. Jedno bez druhého nemôže fungovať. Autentizácia overuje, či je to tá daná osoba, za ktorú sa vydáva. Aby to bolo možné zabezpečiť bezpečnostným systémom, je nutné zistiť, či má daná osoba oprávnenia k tomu, aby sa do systému dostala. Na to slúži autorizácia. Organizácie mávajú častejšie väčšie problémy s autorizáciou oproti autentizácii. Nastáva to kvôli tomu, že autentifikácia užívateľa môže fungovať na základe biometrie alebo viacfaktorovej autentizácii. Samozrejme to neplatí, pokiaľ organizácie nepoužívajú tieto metódy a nechávajú autentizáciu len pomocou hesla, ktoré je volené zamestnancom v organizácii. Ľudia v organizáciách často nemajú znalosti o dôležitosti zvolenia správneho hesla a potrebujú si ho hlavne zapamätať, čo vytvára zraniteľnosť v systémoch pomocou slovníkových útokov. Problém so zabezpečením autorizácie je často v tom, že administrátori, ktorí spravujú organizáciu zabudnú zrušiť oprávnenia pre osoby, ktoré by už naďalej nemali mať prístup k organizácii. Jedná sa napríklad o bývalých zamestnancov a podobne. Ďalším častým problémom môže byť prílišná komplikovanosť systému k správe prístupových práv osôb, čo môže viesť k bezpečnostným dieram v systéme [4, 5].

Všeobecný model

Existuje niekoľko modelov riadenia prístupu a rôzne technológie podporujú rôzne množstvo týchto modelov. Vo všeobecnosti riadený prístup funguje tak, že:

- subjekt si vyžiada prístup k objektu (systém, databáza, súbor a podobne), aby na tomto objekte vykonal nejaké akcie,
- autorizačná služba skontroluje právomoci, ktoré má subjekt vo vzťahu k danému objektu a na základe prístupového pravidla,
- následne na základe predošlého kroku sa subjektu povolí alebo zamietne prístup k pristupovanému objektu [6].

DAC model

DAC (discretionary access control) model dovoľuje obmedziť prístup k objektom na základe identity subjektu alebo skupín subjektov. V tomto modeli má vlastník zdroja právo rozhodnúť o prístupových právach pre všetkých užívateľov systému. Užívateľ môže oprávniť iných užívateľov k prístupu k zdroju a podľa toho, aké veľké oprávnenia im dá, ich oni môžu posúvať ďalším užívateľom. Systém je často kritizovaný za to, že tu chýba centrálny prvok riadenia prístupových práv [6].

MAC model

MAC (mandatory access control) model kontroluje prístupové práva na základe bezpečnostných levelov, ktoré sú pridelené jednotlivým užívateľom a objektom v systéme. V tomto modeli systém priamo centralizovane spravuje užívateľove práva na základe bezpečnostných informácií, ktoré sú pridelené jednotlivým užívateľom a objektom. Systém sa často používa vo vládnych a armádnych systémoch [6].

RBAC model

RBAC (role-based access control) model funguje na princípe pridávania rolí k užívateľovom. Rola reprezentuje funkciu v organizácii. Systémový správca definuje právomoci k danej roli, ktorú ma pridelenú užívateľ. Užívateľ dostane len toľko právomocí, koľko potrebuje [6].

X-BAC modely

Jedná sa o modely, ktoré sú len rozšíreniami RBAC modelu. Vznikajú v dôsledku špecifických požiadaviek rôznych aplikácií. Nachádza sa tu mnoho typov modelov. Napríklad model ABAC, ktorý rieši prístupové práva na základe rôznych atribútov užívateľov, systémov a prostredí [6, 7].

2 ANALÝZA MOŽNÝCH RIEŠENÍ

V tejto kapitole budú prebraté možné riešenia zabezpečeného úložiska a stanovené kritéria k výberu existujúcich riešení. Nachádza sa tu vykonaná analýza existujúcich produktov poskytujúcich aspoň čiastočnú funkcionalitu, ktorú budeme vyžadovať k realizácii cieľov v tejto práci.

2.1 Vhodnosť proprietárnych produktov k riešeniu zabezpečeného úložiska

Jedno z kritérií k výberu produktu na realizáciu zabezpečeného úložiska je typ licencie, pod ktorou je daná aplikácia poskytovaná. Je to z dôvodu, že licencia hovorí, za akých podmienok môžeme daný software používať. Táto kapitola sa venuje proprietárnym produktom a ich vhodnosti k použitiu.

Je niekoľko veľkých poskytovateľov cloudových úložísk ako Google Drive, Microsoft One Drive, Dropbox a podobne. Sú to verejné cloudové úložiská, ktoré ponúkajú do rôznej miery riešenia nielen pre jednotlivcov, ale aj pre organizácie. Sú pomerne cenovo efektívne, pretože ich poskytovateľ sa stará o všetku infraštruktúru a jej údržbu. Nie vždy ale platí, že tieto služby sú lacnejšie. Výhodou oproti riešeniam, ktoré si organizácia prevádzkuje sama je obstarávacia cena, nakoľko netreba budovať rozsiahlu infraštruktúru. Takisto spomínané služby poskytujú veľké množstvo funkcionalít. Treba dať pozor, či daný poskytovateľ za danú službu neúčtuje dodatočné poplatky, nakoľko tie vedia cenu za úložisko veľmi predražiť.

Toto je veľmi lákavé riešenie až do bodu, kým sa k tomu nezaráta ochrana súkromia a bezpečnosť uložených dát na cloudových úložiskách. Tu nastávajú hlavné a podstatné nevýhody proprietárnych riešení. Existuje niekoľko dôvodov, ktoré prispievajú k zvýšeným obavám o bezpečnosť takýchto riešení. Medzi jeden z hlavných dôvodov patrí to, že zdrojový kód proprietárnych produktov nie je verejne dostupný a teda nie je možné overiť, či v systéme nie sú podstatné bezpečnostné chyby. Chyby by sa mohli vyskytovať nevedome od poskytovateľa alebo vedome ako takzvaný backdoor (metóda tajných vchodov/zadné vrátka).

V súčasnosti mnoho z týchto poskytovateľov sídli v Spojených Štátoch Amerických, ktoré majú vybudovanú sieť sledovacích programov pomocou svojich agentúr ako NSA. Tieto programy boli v minulosti zneužívané protizákonne a porušovali nimi právo ochrany súkromia [8]. Spoločnosti ako Google, Microsoft a mnoho ďalších boli na zozname spoločností, ktoré poskytovali priamy prístup pre sledovací program Prism. Poskytovali im všetky dáta, ktorými disponovali, aj keď to verejne popreli [9].

Neraz majú spoločnosti v podmienkach používania, že si môžu nárokovať na dáta, ktoré patria organizácii v mene zlepšenia poskytovania služieb [10]. Medzi ďalšiu z nevýhod patrí to, že celú infraštruktúru spravuje poskytovateľ. Organizácia, ktorá si službu prenajíma má omnoho menšiu kontrolu nad poskytovanou službou. Nevie zabezpečiť fyzickú ani logickú kontrolu prístupu. Ak by nastala situácia, kedy by sa prenajímateľ nedozvedel o kybernetickom útoku na poskytovateľa, následkom by mohol byť únik alebo iná kompromitácia dát. V prípade, že by sa o tom dozvedel, nemusel by sa dopátrať o aký veľký rozsah poškodenia dát sa jedná.

Z toho nám vyplýva nasledujúci záver. Pokiaľ nie je prístupný kód produktu, nie je možné zaručiť najvyššiu bezpečnosť pri ochrane dát daným produktom. Takéto riešenia nie je možné odporučiť ako súčasť vládneho systému a ani ako súčasť systému súkromnej spoločnosti na ukladanie citlivých dát v dôsledku možného konfliktu záujmov s poskytovateľom služby.

2.2 Výhody open source produktov k riešeniu zabezpečeného úložiska

V predchádzajúcej kapitole sa dospelo k záveru, že proprietárne produkty sú nevhodné alebo minimálne menej vhodné. Na druhej strane je takzvaný free alebo open source softvér. Na to, aby mohol byť program považovaný za free softvér podľa oficiálnej stránky musí spĺňať:

„Sloboda spúšťať program, a to za akýmkoľvek účelom (sloboda 0). Sloboda študovať ako daný program funguje, vykonávať v ňom zmeny a prispôsobovať ho tak svojim požiadavkám (sloboda 1). Prístup k zdrojovému kódu je k tomu nevyhnutnou podmienkou. Sloboda voľne šíriť kópie a pomáhať tak svojmu okoliu (sloboda 2). Sloboda zlepšovať daný program a tieto zlepšenia (vrátane upravených alebo úplne nových verzií) ďalej zverejňovať, aby tieto zlepšenia mohli byť prínosom pre celú spoločnosť (sloboda 3). Prístup k zdrojovému kódu je k tomu nevyhnutnou podmienkou [11]."

Každý free software je zároveň open source. Nie vždy to tak však je aj opačne [12]. Hlavným rozdielom medzi nimi je, že pri open source je nutné ho po úprave kódu programu ďalej zverejniť pod tou istou licenciou.

Výhody free alebo open source produktov nie sú v tom, že sú lepšie navrhnuté. Produkty majú svoj zdrojový kód zverejnený, čo rieši hlavnú vadu proprietárnych systémov. Prináša to rôzne výhody. Na kód sa môže pozrieť veľké množstvo nezávislých ľudí a expertov, ktorí môžu objaviť chyby a zároveň navrhnúť riešenia. Aj keď program nemusí byť lepšie navrhnutý, môžu sa pridať rôzni nezávislí ľudia, ktorí prispejú k jeho vylepšeniu a overeniu jeho bezpečnosti. V prípade, že sa nájde

bezpečnostná chyba v takomto produkte, aktualizácie na riešenie problému sú dostupné veľmi rýchlo. To platí hlavne v prípade, ak má produkt veľkú popularitu a väčšie množstvo prispievateľov. Ďalšia z výhod je, že ak si danú službu organizácia prevádzkuje sama, má plnú kontrolu nad systémom. To znamená kontrolu nad fyzickým aj logickým prístupom k službe. V tom sú okrem dát zahrnuté aj logovacie súbory a prístup ku kódu samotnému. Logovacie súbory sú veľmi dôležite, pretože na základe nich vieme detekovať, opraviť chyby alebo monitorovať zmeny v systéme. Následne pomocou zdrojového kódu vieme opraviť zle fungujúce funkcie programu alebo pridať nové. Táto spomenutá výhoda je za cenu vyšších počiatočných nákladov na prevádzku, nakoľko treba zabezpečiť infraštruktúru a správcu systému. Vráti sa to na nižších mesačných nákladoch a hlavne na tom, že ďalšie úkony s dátami nie sú finančne účtované, na rozdiel od niektorých proprietárnych riešení.

Z tohto zhrnutia vyplýva, že pre realizáciu zabezpečeného úložiska pre účely tejto práce sú omnoho vhodnejšie produkty, ktoré sú šírené ako free alebo open source softvér. Cieľom práce je zrealizovať bezpečné úložisko v čo najväčšej miere. Na základe zistených informácií sa budú analyzovať len produkty šírené pod free alebo open source licenciami.

2.3 Analýza vybraných produktov

2.3.1 Kritériá k porovnaniu produktov

K tomu, aby sa mohli vyvodiť porovnania možných riešení zabezpečeného úložiska sa definujú kritériá, na ktoré sa budeme pozerať pri výbere možných riešení. Kritériá vznikli na základe prieskumu trhu s existujúcimi produktami. Kritériá sú nasledovné:

- *Typ licencie* - jedná sa o licenciu, pod ktorou prevádzkovateľ alebo vlastník práv k produktu poskytuje daný produkt. V tomto prípade, ako bolo vyššie spomenuté, sa budeme zameriavať predovšetkým na produkty, ktoré sú zverejnené pod free alebo open source licenciou.
- *Známe bezpečnostné chyby* - zaujíma nás, či má produkt nejaké známe bezpečnostné chyby, ktoré nie sú zatiaľ opravené, poprípade ako dlho trvala oprava predošlých - ak nejaké boli.
- *Pokračujúci vývoj projektu* - je dôležité nepoužiť zastaralé technológie. Najideálnejšie je, pokiaľ je produkt stále podporovaný a oficiálne vo vývoji. Nie vždy by sa dali označiť produkty po ukončení podpory a vývoju ako zlé či zastaralé. Ich najväčším problémom môžu byť neohlásené a neopravené bezpečnostné chyby. V budúcnosti by to znamenalo zvýšenú náročnosť na spravovanie takéhoto systému.

- *Self-Hosting alebo samohostenie* - pod touto vlastnosťou máme na mysli, či je daný nastroj možné poskytovať s vlastnou infraštruktúrou alebo je nutné použiť externé cloudové úložisko. Je preferované ak má produkt túto možnosť.
- *Miera implementácie riadeného prístupu* - dôležité je určiť aké možnosti autentizácie a autorizácie má implementovaný daný produkt. Dôležitá je aj funkcia možnosti zdieľania súborov rôznym ľudom, skupinám a možnosti nastavenia právomocí k prístupu, či manipulovaniu so súborom.
- *Možnosti dlhodobej archivácie dát* - jedna zo žiadaných vlastností pri realizácii je schopnosť produktu poskytnúť možnosti na riešenie dlhodobej archivácie dát a hlavne zabezpečiť ich integritu počas tejto doby.
- *Zabezpečenie dôvernosti a integrity* - kľúčové vlastnosti dát, ktoré musia byť schopné takto zabezpečené úložisko poskytnúť.
- *Prístup k službe* - zhodnotenie kompatibility prístupových klientov. Možnosť prístupu z webového prehliadača, existencia desktopových klientov a ich kompatibilita pre jednotlivé operačné systémy ako Windows, Mac OS alebo Unix.
- *Šifrovanie na strane klienta* - táto možnosť je veľmi podstatná. Dáta musia byť zašifrované a bezpečne prenesené na cloudové úložisko ešte pred začatím prenosu.
- *Šifrovanie metadát a súborových štruktúr* - metadáta obsahujú dôležité informácie ako veľkosť súboru alebo dátum a čas vytvorenia tohto súboru. Takisto štruktúry priečinkov, v ktorých sú súbory uložené je vhodné utajiť. Informácie v neutajenej forme by mohli napovedať neoprávnenej osobe o aké dáta sa jedná.
- *Použité šifrovacie algoritmy* - použité šifrovacie algoritmy v produkte musia byť považované za bezpečné organizáciami ako NIST.
- *Cena* - aj napriek tomu, že v porovnaní sa práca venuje produktom, ktoré sú z celej alebo veľkej časti produkty šírene pod open source alebo free software licenciami, väčšina poskytuje možnosť platenej licencie pre organizácie. V týchto licenciách môžu byť výhody. Za ne sa považuje online podpora, konzultácie s developermi, možná úprava softvéru a odstránenie povinnosti ho ďalej zverejniť a podobne. Preto je nutné brať do úvahy aj tento faktor.
- *Výkonnosť* - vzhľadom na možnosť ukladania veľkého množstva dát je potrebné, aby aplikácia nebola veľmi náročná na zdroje v infraštruktúrach. V tejto fáze analýzy vlastností bude hodnotená výkonnosť len na základe verejne dostupných informácií.
- *Obmedzenie používania* - produkty môžu mať rôzne obmedzenie pri používaní, či už v dôsledku licencie alebo implementácie. Môže sa vyskytnúť napríklad obmedzenie počtu alebo veľkosti súborov, ktoré je tam možné uložiť.
- *Bonusové možnosti* - každý produkt môže mať nejaké funkcionality, ktoré nás

v našom prípade primárne nezaujímajú, ale poskytujú nejakú funkcionalitu, ktorá by mohla byť prospešná. Nemusí sa však jednať len o funkcionalitu softvéru. Môže sa jednať aj o bonusy služieb od poskytovateľa.

2.3.2 Selekcia produktov k podrobnejšej analýze

V tejto kapitole je niekoľko tabuliek, v ktorých budú stručne analyzované vybrané produkty. Výber bol uskutočnený na základe toho, či sú schopné poskytnúť nejakú z funkcionalít potrebnú pre realizáciu zabezpečeného úložiska. Tieto produkty sú už vyfiltrované na základe kritéria - musia byť poskytované pod licenciami typu open source alebo free software, čo znamená, že ich kód je verejný. Je tu vykonaná prvotná analýza, pomocou ktorej sa vyselektovali produkty do podrobnejšej analýzy. V nasledovnej analýze nás budú zaujímať hlavne parametre z predošlej kapitoly:

- typ licencie,
- známe bezpečnostné chyby,
- pokračujúci vývoj projektu,
- self-hosting (samohostenie),
- implementácia riadeného prístupu,
- možnosti dlhodobej archivácie dát,
- zabezpečenie dôvernosti a integrity,
- prístup k službe,
- šifrovanie na strane klienta.

Podľa získaných informácií sa vyhodnotí vhodnosť produktov pre účely tejto práce a na základe toho postúpia do podrobnejšieho výberu, kde sa budú riešiť všetky spomínané parametre.

Vzhľadom na rýchlo postupujúci vývoj dnešnej doby denne pribúdajú nové aplikácie a aj možné bezpečnostné hrozby. Z toho vyplýva, že toto porovnanie nemusí byť aktuálne. Výber produktov je vytvorený na jeseň roku 2020.

Prvotná analýza

V tabuľkách 2.1, 2.2, 2.3, ktoré sa nachádzajú na nasledujúcich stranách je možné vidieť analýzu produktov.

Tab. 2.1: Analýza produktov 1

Produkt	**Pydio Cells**	**Syncany**	**VeraCrypt**	**CryFS**	**Syncthing**
Typ licencie	GNU AGPL v3	MIT License	TrueCrypt License Version 3.0	GNU AGPL v3	Mozilla Public License 2.0
Známe bezpečnostné chyby	Nie	Nie, Posledné vydanie v roku 2017	Nie	Nie, Je nebezpečené pri prístupe k súboru z viacerých klientov	Nie
Pokračujúci vývoj projektu	Áno	Nie	Áno	Áno	Áno
Self-Hosting	Áno	Nie	Nie	Nie	Nie
Implementácia riadeného prístupu	Áno	Áno	Áno	Nie	Áno
Možnosti dlhodobej archivácie dát	Nie	Nie	Nie	Nie	Nie
Zabezpečenie dôvernosti	Áno	Áno	Áno	Áno	Áno
Zabezpečenie integrity	Nie	Áno	Nie	Áno	Nie
Prístup k službe	Web Windows Mac Unix Andoroid IOS	Windows Mac Unix	Windows Mac Unix	Mac Unix	Web Windows Mac Unix Andoroid
Šifrovanie na strane klienta	Nie	Áno	Áno	Áno	Nie
Vhodnosť pre podrobnejšiu analýzu	Nie	Nie	Nie	Nie	Nie

Tab. 2.2: Analýza produktov 2

Produkt	**Nextcloud**	**ownCloud**	**Seafile**	**MinIO**	**CryptPad**
Typ licencie	GNU AGPLv3	GNU AGPLv3	GNU AGPLv3 (server core)	GNU AGPL v3	GNU AGPL v3
Známe bezpečnostné chyby	Nie	Nie	Áno	Nie	Nie
Pokračujúci vývoj projektu	Áno	Áno	Áno	Áno	Áno
Self-Hosting	Áno	Áno	Áno	Áno	Áno
Implementácia riadeného prístupu	Áno	Áno	Áno	Áno	Áno
Možnosti dlhodobej archivácie dát	Nie	Nie	Nie	Nie	Nie
Zabezpečenie dôvernosti	Áno	Áno	Áno	Áno	Áno
Zabezpečenie integrity	Áno	Áno	Áno	Áno	Nie
Prístup k službe	Web Windows Mac Unix Android IOS	Web Windows Mac Unix Android IOS	Web Windows Mac Unix Android IOS	Web	Web
Šifrovanie na strane klienta	Áno	Áno	Áno	Áno	Áno
Vhodnosť pre podrobnejšiu analýzu	Áno	Áno	Áno	Nie	Nie

Tab. 2.3: Analýza produktov 3

Produkt	**eCryptFS**	**EncFS**	**Cryptomator**	**TrueCrypt**
Typ licencie	GNU AGPL v3	LGPL	GNU AGPL v3	TrueCrypt License Version 3.0
Známe bezpečnostné chyby	Nie	Áno	Nie	Áno
Pokračujúci vývoj projektu	Nie	Nie	Áno	Nie
Self-Hosting	Nie	Nie	Nie	Nie
Implementácia riadeného prístupu	Nie	Nie	Nie	Nie
Možnosti dlhodobej archivácie dát	Nie	Nie	Nie	Nie
Zabezpečenie dôvernosti	Áno	Áno	Áno	Áno
Zabezpečenie integrity	Nie	Nie	Nie	Nie
Prístup k službe	Unix	Windows Mac Unix	Windows Mac Unix Android	Windows Mac Unix
Šifrovanie na strane klienta	Áno	Áno	Áno	Áno
Vhodnosť pre podrobnejšiu analýzu	Nie	Nie	Nie	Nie

Stručné vyhodnotenie analýzy

V tejto podkapitole sa nachádza krátke vyhodnotenie a odôvodnenie výsledného rozhodnutia k produktom. Z tejto stručnej analýzy sa zistilo, že väčšina vybraných produktov nebola vhodná k realizácii zabezpečeného úložiska pre väčšiu organizáciu. Nakoľko v kritériách bolo spomenuté, že self-hosting nie je nevyhnutný, tak väčšina aplikácií ako Cryptomator alebo VeryCrypt neimplementovala žiadnu možnosť riadenia prístupu k súborom alebo zdieľanie týchto súborov medzi rôznymi užívateľmi. Na základe toho museli byť tieto produkty vylúčené. Väčšina z nich, pokiaľ v nich nie sú bezpečnostné chyby, sú vhodné len pre osobné účely. Pre prípady, kedy chce užívateľ používať nejaký z verejných cloudových úložísk a zároveň si zachovať súkromné dáta v bezpečí. Pri zdieľaní takto zašifrovaných dát musia užívatelia čeliť problematike zdieľania kľúčov k dešifrovaniu uložených informácií.

V prípade Syncthing aplikácie, ktorá je vhodná pre zdieľanie súborov medzi užívateľmi, sa jedná o peer to peer (sieť so vzájomným sprístupňovaním) synchronizáciu súborov medzi zariadeniami. Nie je možné pomocou nej vytvoriť zabezpečené úložisko.

CryptPad neumožňuje vkladanie súborov. Je tu možné písať a pracovať s rôznymi textovými súbormi, ktoré sa pred odoslaním na server zašifrujú a zašifrované uložia. Dorobenie možnosti vkladania súborov by bolo zdĺhavé a iné produkty to už poskytujú. Z tohto dôvodu je produkt vylúčený.

MinIO nie je aplikácia vhodná pre bežných koncových užívateľov, nakoľko poskytuje možnosti zabezpečeného úložiska, ale nie sú tu žiadne užívateľsky prívetivé prístupy k službe. K službe sa pristupuje cez príkazový riadok, knižnicu v programovacom jazyku na to určenú alebo veľmi obmedzene cez webový klient, ktorý slúži prevažne na kontrolu či táto služba funguje.

Seafile je vhodný vďaka funkcionalite, ktorú poskytuje, ale na základe bezpečnostnej chyby v jednej z ich klientskych aplikácií je vhodné produkt vylúčiť z ďalšieho porovnávania. Táto chyba je z júla 2020 a podľa dostupných informácií nebola stále opravená (27.5.2021) [13].

Pydio Cells poskytoval takmer všetku potrebnú funkcionalitu okrem šifrovania na strane klienta. Je to jedna z najpodstatnejších funkcionalít, ktorá bola potrebná a nakoľko ju iné produkty už majú zabudovanú je taktiež nutné produkt vylúčiť.

Možnosti dlhodobej archivácie podľa našich požiadaviek neponúka žiadny z produktov. Nextcloud aj ownCloud poskytujú možnosť manuálneho aj automatizovaného manažmentu súborov. V podrobnejšej analýze sa teda bude porovnávať Nextcloud a ownCloud, nakoľko len tieto dve riešenia ponúkajú všetku funkcionalitu potrebnú k realizácii cieľov tejto práce.

2.4 Podrobná analýza vybraných produktov

V nasledujúcej kapitole sa porovnajú vybrané produkty z predošlej stručnej analýzy. V podrobnej analýze sa zoberú do úvahy všetky spomenuté kritéria.

2.4.1 Nextcloud

Nextcloud je software poskytujúci privátne alebo hybridné cloudové úložisko. Je to alternatívou k proprietárnym úložiskám ako je Dropbox, Google Drive a podobne.

Licencie

Nextcloud je poskytovaný pod licenciou GNU AGPLv3. Tento produkt aj všetky jeho časti sú plne poskytované pod touto licenciou. Jedná sa o open source licenciu, ktorá umožňuje voľné používanie, čo znamená komerčné aj osobné použitie, distribúciu, modifikáciu či patentové používanie. Pod touto licenciou sa neposkytuje žiadna záruka ani zodpovednosť. Podmienky používania hovoria, že musí byť dostupné upovedomenie o tejto licencii, upovedomenie o vykonaných zmenách a zverejnenie zdrojového kódu v prípade zmien. Ďalšia distribúcia musí byť pod tou istou licenciou [14].

Známe bezpečnostné chyby

Nextcloud aktuálne nemá žiadne známe bezpečnostné chyby. Od vydania bolo objavených a nahlásených 99 bezpečnostných chýb v ich produktoch, ktoré boli ohodnotené pomocou CVE ID. Sú v nich zarátané všetky súčasti ako server, tak aj klientske aplikácie. Tieto údaje boli získané z oficialnych stránok organizacie NIST. Všetky objavené chyby boli náležite opravené. Nextcloud udržiava podrobný záznam zmien a opráv v kóde. Na nahlásené chyby reaguje veľmi rýchlo. Má bonusový program za nahlásenie nových chýb. Za ich nahlásenie je podľa závažnosti poskytovaná finančná odmena.

Pokračujúci vývoj projektu

Nextcloud bol vydaný v roku 2016 a je to pomerne nový produkt. Jeho popularita rastie a vývoj naďalej pokračuje. Má pomerne veľký tým developerov pracujúcich na projekte a mnoho prispievateľov k nemu.

Self-Hosting

Nextcloud je nutné hosťovať na vlastnej infraštruktúre a podporuje aj možnosť pripojenia externých úložísk.

Implementácia riadeného prístupu

Nachádza sa tu mnoho možností riadeného prístupu. Je možné nastavovať jednotlivým užívateľom oprávnenia k súborom. Vkladať užívateľov do skupín a na základe toho definovať oprávnenia k prístupu. Je možné zdieľať súbory rôznym ľudom či skupinám. Je tu zabezpečená transparencia v tom, kto má prístup k súborom. Administrátor má práva vidieť súbory užívateľov pokiaľ nie sú zašifrované. Všetko čo administrátor spraví je uložené v takzvaných denníkoch [15]. Existuje tu aj integrovaná možnosť použitia autentizácie užívateľov pomocou dvojfaktorovej autentifikácie, LDAP, SAML, Active Directory a Kerberos [16].

Možnosti dlhodobej archivácie dát

Nextcloud neposkytuje priamo funkcionalitu dlhodobej archivácie dát. Je tu ale dostupná možnosť označovania súborov štítkami, na základe ktorých je možné nastaviť iné prístupové práva týmto súborom, premiestniť ich do špecifickej zložky, vymazať alebo ich zadržať. Nextcloud je dizajnovaný aby spĺňal regulácie ako GDPR [17].

Zabezpečenie dôvernosti a integrity

Nextcloud má bezpečnosť na veľmi vysokej úrovni a s použitím šifrovania na strane klienta a serveru plne chráni dôvernosť a integritu dát. Nextcloud používa zabezpečené spojenie pomocou HTTPS a dáta sú chránené aj v prenose pomocou TLS protokolu [18, 20]. Najnovšia verzia protokolu 1.3 je podporovaná od verzie klientov 2.6.0 [21].

Prístup k službe

Prístup k službe je možný pomocou webovej stránky alebo klientskych aplikácií, ktoré sú implementované na mnoho operačných systémov ako pre desktopové systémy, tak aj mobilné zariadenia.

Šifrovanie na strane klienta

Možnosť šifrovania na strane klienta obsahujú klientske aplikácie. Znamená to, že dáta sa zašifrujú ešte pred prenosom na server. Server nemôže vidieť aké dáta to sú. Aj napriek šifrovaniu je možné zdieľať súbory s inými užívateľmi. Je to doplnok k už existujúcej možnosti šifrovania na strane serveru [18].

Šifrovanie metadát a súborových štruktúr

Nextcloud pri zapnutom šifrovaní na strane klienta šifruje všetky metadáta súboru spolu s názvom. V prípade potreby nahratia celej súborovej štruktúry je možné ju zašifrovanú nahrať.

Použité šifrovacie algoritmy

Dokumentácia k funkcionalite šifrovania na strane klienta je menej podrobná, čo sa týka použitých algoritmov, oproti šifrovaniu na strane servera. Na šifrovanie súborov pri šifrovaní na strane klienta sa najskôr pomocou certifikátov X.509 a asymetrickej kryptografie vytvorí a synchronizuje identita klienta. Následne sa vygenerujú metadáta a kľúč k nim, ktorým sa následne zašifrujú. Kľuč k metadátam je zašifrovaný všetkými verejnými kľúčmi užívateľov, ktorí tam majú prístup. Tie sa uložia na serveri a následne sa vygeneruje 128 bitový kľúč pre šifrovanie zvoleného súboru. Ten je potom použitý pomocou šifrovacieho algoritmu AES v móde GCM. Následne sa vygeneruje náhodný identifikátor, pod ktorým sú dáta uložené na serveri [18].

K symetrickému šifrovaniu súborov na strane serveru sa používa šifrovací algoritmus AES v móde CTR. Kľúč k symetrickému šifrovaniu pozostáva z 256 náhodných bitov. Pri šifrovaní súborov pomocou AES sa používa inicializačný vektor o veľkosti 128 bitov. Súbor je pred šifrovaním ešte rozložený do blokov o veľkosti 6072 bytov, posledný blok môže byť menší.

K asymetrickej kryptografii sa používa algoritmus RSA. Verejný kľúč je používaný k šifrovaniu zdieľaných kľúčov a privátny k dešifrovaniu týchto kľúčov. Privátny kľúč sa ukladá šifrovaný rovnako pomocou AES 256 v móde CTR. Kľúč pre šifrovanie privátneho kľúča je derivovaný pomocou algoritmu SHA256-PBKDF2 spolu s použitím soli a 100 000 opakovaní. Soľ je generovaná z premenných programu a pomocou algoritmu SHA256. Pre podpísanie zašifrovaného privátneho kľúča sa používa SHA256-HMAC [19].

Cena

Samotný produkt je zadarmo. Pre organizácie, ktoré potrebujú kritické zabezpečenie funkčnosti tu je možnosť plateného odberu a podpory pod názvom Nextcloud Enterprise. Táto podpora je poskytovaná v troch cenových kategóriách označovanými ako základ, štandard a prémium. Je možné získať špeciálnu ponuku a verejný či školský sektor môžu dostať výrazné zľavy. Nakoľko je veľké množstvo služieb, ktoré poskytujú, nebudeme ich tu vypisovať všetky, ale len v skratke ich popíšeme. Rozdiely medzi kategóriami sú hlavne v počte služieb pri podpore prevádzky tohto úložiska ako aj jej rozsahu. Odbery sú dostupné od 50 užívateľov vyššie.

Ceny sú:

- základ: 50 užívateľov – 1900€/rok,
- štandard: 50 užívateľov – 3400€/rok,
- prémium: 50 užívateľov – 4900€ /rok [22].

Výkonnosť

Nextcloud je vytvorený pre veľké spoločnosti a je schopný poskytnúť potrebný výkon. Sú v ňom možnosti nastavenia k optimalizácií výkonnosti a aj škálovateľnosť. Výkonnosť a náročnosť na zdroje bude vyhodnotená v prípade realizácie nakoľko nie sú dostupné žiadne dáta.

Obmedzenie používania

Základné nastavenie obmedzujúce maximálnu veľkosť nahrávaných súborov je 512MB. Toto obmedzenie je možné zmeniť. Jediným limitom je teda operačný a súborový systém, na ktorom služba pracuje. V prípade nahrávania do úložiska sú tu limity vo webových aplikáciách ako Internet Explorer [23].

Bonusové možnosti

Nextcloud poskytuje mnoho rozširujúcich aplikácií, ktorých podpora môže byť zapnutá alebo vypnutá. Nachádza sa tam mnoho textových editorov pre ľahšiu spoluprácu v tímoch, ako aj bezpečnostné rozšírenia, napríklad súborový antivírus. Nextcloud podporuje aj podpisovanie kódu pre všetky hlavné vydania a aplikácie [24].

2.4.2 OwnCloud

OwnCloud je rovnako ako Nextcloud software poskytujúci cloudové úložisko. OwnCloud bol založený Frankom Karlitschekom, ktorý sa v roku 2016 rozhodol odčleniť od ownCloudu a vytvoril tým Nextcloud. Veľké množstvo prispievateľov vývoju tohto produktu odišlo spolu s ním [25].

Licencie

OwnCloud štandardná edícia je takisto šírená pod licenciou AGPLv3. Jadro kódu softwaru je licencované pod ownCloud Contributor Licence Agreement (CLA). Vďaka tomuto je možné poskytovať ownCloud pod duálnymi licenciami. To znamená, že pri odbere Enterprise verzie, ktorú ownCloud poskytuje, zákazník dostáva rovnaké jadro kódu ako pri štandardnej edícii. OwnCloud Enterprise edícia je šírená pod ownCloud komerčnou licenciou, ktorá už ale nie je open source. V tomto prípade

ownCloud tvrdí, že zákazníci majú plný prístup k zdrojovým kódom a akékoľvek modifikácie kódu si môžu ponechať súkromné bez povinnosti zverejňovania tohto upraveného kódu [26].

Známe bezpečnostné chyby

K tomuto produktu bolo nahlásených od svojho vzniku 139 bezpečnostných chýb, ktorým bolo priradené CVE ID. Posledné priradené bolo z roku 2017, ale zverejnené vo februári 2020. V tomto bode nie sú známe žiadne neopravené bezpečnostné chyby, ktoré by boli ohodnotené. Tieto dáta boli ziskané zo stránok organizácie NIST. Aktualizácie a opravy vychádzajú pomerne často a sú rovnako ako pri Nextcloude dobre zdokumentované v poznámkach k vydaniu. OwnCloud má na svojich stránkach návod ako nahlásiť bezpečnostnú chybu v produkte a rovnako ako pri Nextcloude má program odmien za nahlásenie chýb.

Pokračujúci vývoj projektu

OwnCloud bol vydaný v marci 2010 a je rovnako ako Nextcloud naďalej udržiavaný a sú pridávané nové funkcionality. OwnCloud na rozdiel od Nextcloudu má takmer o polovicu menej vývojárov prispievajúcich do produktu. Aktivita ownCloudu začína jemne upadať [27].

Self-Hosting

OwnCloud je nutné prevádzkovať na vlastnej infraštruktúre s možnosťou rozšírenia a použitia externých úložísk.

Implementácia riadeného prístupu

Implementácia riadneho prístupu je na podobnej úrovni ako pri Nextcloude. Je nutné rozlišovať medzi štandardnou verziou a Enterprise verziou, nakoľko štandardná neposkytuje ani zďaleka rovnaké možnosti. Niektoré funkcie chýbajú aj v Enterprise verzii OwnCloudu oproti Nextcloudu, dve z nich sú napríklad nastavovanie právomoci prístupu k aplikácii alebo prihlasovanie užívateľov pomocou SAML [16].

Možnosti dlhodobej archivácie dát

V Enterprise verzii je poskytované rozšírenie, ktoré implementuje funkcionalitu, kde sa dá automatizovane archivovať alebo vymazávať súbory na základe nastavených pravidiel. Je tu možné nastaviť obnovenie súborov z archívu. Môžu to robiť len členovia administrátorskej skupiny [28]. O dodatočnom zabezpečení takýchto dát nie sú dostupné žiadne informácie.

Zabezpečenie dôvernosti a integrity

Dáta sú zabezpečené počas prenosu pomocou HTTPS a TLS protokolu. Po uložení je možnosť nastavenia šifrovania na strane serveru. V Enterprise verzii je možné zabezpečenie týchto dát zvýšiť použitím šifrovania na strane klienta.

Prístup k službe

OwnCloud ponúka rovnaké možnosti prístupu ako u Nextcloudu, dokonca aj grafické spracovanie je veľmi podobné.

Šifrovanie na strane klienta

Ako bolo spomenuté, táto funkcionalita je poskytnutá iba v Enterprise edícii produktu. Funkcionalita je dostupná len z webového prehliadača pomocou vbudovaného javascriptu. Implementácia šifrovania v desktopových a mobilných klientoch nie je spomenutá [29].

Šifrovanie metadát a súborových štruktúr

Túto funkcionalita produktu nie je možné dohľadať pri šifrovaní na strane klienta.

Použité šifrovacie algoritmy

OwnCloud voľne neposkytuje podrobnú dokumentáciu šifrovania na strane klienta, takže použité postupy nie sú úplne jasné. Pri šifrovaní na strane serveru je to o niečo lepšie. OwnCloud uvádza, že zákazníci majú možnosť ľahkého prístupu a možnosti k upraveniu šifrovania. Napríklad si môžu manažovať šifrovacie kľúče alebo aj správanie šifrovania. Cieľom šifrovania na strane serveru je rovnako ako pri Nextcloude zabezpečenie dát na vzdialených úložiskách proti neoprávnenému prístupu. Postup začína generovaním páru kľúčov asymetrickej kryptografie o dĺžke 4096 bitov. Následne je privátny kľúč zašifrovaný užívateľovým heslom pomocou AES-256. Ďalej si vygeneruje 256 bitový kľúč pre každý súbor. Predošlé dva kroky administrátor môže upraviť podľa svojich potrieb. Potom sa pomocou daných kľúčov a algoritmu AES v móde CFB zašifrujú tieto súbory [30].

Cena

Štandardná verzia je zadarmo, rovnako ako pri Nextcloude. Dá sa k nej priplatiť podpora, kde je cena 1500€/rok pre 25 užívateľov. Cena za jedného užívateľa postupne klesá, čím viac užívateľov je. Enterprise licencia poskytuje extra funkcionalitu oproti štandardnej verzii. Spotrebiteľ nepríde o prístup k zdrojovému kódu všetkých

súčastí produktu. V prípade vykonania zmien v kóde odstraňuje povinnosť tieto zmeny zverejniť. Ceny sa rovnako ako pri podpore štandardnej verzii odvíjajú od počtu užívateľov. Začínajú na 3600€/rok za 25 užívateľov [31].

Výkonnosť

Podobne ako pri Nextcloude je výkon závislý od správnej konfigurácie. Informácie k výkonnosti nie sú dostupné a budú vyhodnotené v prípade výberu k praktickej realizácii.

Obmedzenie používania

V ownCloude takisto nie sú žiadne umelé bariéry pri používaní ako napríklad obmedzený počet užívateľov alebo súborov. Dokonca aj limitácie súborového a operačného systému sú udávané rovnako ako pri Nextcloude [32].

Bonusové možnosti

OwnCloud rovnako ponúka aplikácie pre rozšírenie cloudovej služby, ktorých je v porovnaní s Nextcloudom väčšie množstvo, ale mnoho z nich je prístupných len v Enterprise verzii.

2.4.3 Zhodnotenie analýzy produktov

Z podrobného zhodnotenia je jasné, že Nextcloud a ownCloud sú veľmi podobné. Oba podporujú takmer rovnaké množstvo funkcionalít a rozdiely sú na prvý pohlaď minimálne.

Hlavné rozdiely sú v licenciách, kde má Nextcloud výhodu vďaka tomu, že je celý produkt voľne dostupný pod open source licenciou AGPLv3. Štandardná verzia ownCloudu pod touto licenciou nemá ani zďaleka takú funkcionalitu. Samozrejme podľa spôsobu použitia je možné, že duálne licencovanie ownCloudu môže byť výhodou, nakoľko prípadné zmeny sa nemusia zverejňovať pod Enterprise licenciou. Tu ale nastáva problém, že v takom prípade si spoločnosť za túto licenciu musí platiť vždy pokiaľ využíva takto upravený produkt. V Nextcloude si organizácia platí podporu produktu nie funkcionalitu. To je výhodou, nakoľko v prípade, že organizácia túto podporu nepotrebuje, môže za ňu prestať platiť v ktoromkoľvek bode. Organizácia aj naďalej bude mať právo prevádzkovať túto službu na ich serveri. Takisto problém pri duálnom licencovaní ownCloudu je v tom, že nie je verejný celý zdrojový kód. Vďaka tomu je väčšia šanca existencie bezpečnostných chýb alebo dlhšej doby prípadného nahlásenia tejto chyby.

Prioritou Nextcloudu je bezpečnosť, v ktorej má aj miernu prevahu vďaka šifrovaniu na strane klienta v klientskych aplikáciách oproti ownCloudu, kde šifrovanie prebieha pomocou webového prehliadača. OwnCloud nikde neuvádza možnosť šifrovania na strane klienta v klientskych aplikáciách. O bezpečnosti týchto systémov svedčí aj používanie vo vládnych sektoroch ako napríklad použitie Nextcloudu v Nemeckej federálnej vláde alebo na Francúzskom ministerstve vnútra [33].

Pokiaľ ide o rýchlosti odozvy na bezpečnostné chyby a dotazy na podporu je na tom Nextcloud lepšie. Takisto má aj väčšiu komunitu, čo vedie k rýchlejšiemu nájdeniu riešenia problémov, ktoré by sa mohli objaviť.

Cenovo pri porovnaní platenej podpory pri Nextcloude a OwnCloude Enterprise licencie to na 50 užívateľov vychádza lacnejšie pre Nextcloud a to o 2300€ za rok, v prípade prémiovej podpory od Nextcloudu. Nextcloud prichádza s výhodou, že v prípade nedostatočnej využiteľnosti podpory, môže odber ukončiť.

Z tohto porovnania vyplýva, že oba produkty sú veľmi schopné, ale Nextcloud má v niektorých ohľadoch prevahu. Preto bude vybrané na nasledovnú konfiguráciu v lokálnom testovacom prostredí.

3 SPREVÁDZKOVANIE ZABEZPEČENÉHO ÚLOŽISKA

V tejto kapitole sa nachádza inštalácia úložiska so všetkými závislosťami nutnými k prevádzke. Následné nastavenie všetkej vyžadovanej funkcionality v ňom. Ďalej bude vykonaný popis funkcií. Nakoniec bude spravená jednoduchá výkonová analýza.

3.1 Parametre inštalácie

Nextcloud má vo svojej dokumentácii uvedené minimálne požiadavky pre prevádzku Nextcloud serveru. Pri požiadavkách systému je nutné brať ohľad na počet užívateľov, množstvo aplikácií a možnú veľkosť nahrávaných súborov. V tejto práci je realizovaná len základná inštalácia na virtuálnom počítači pomocou nástroja VMware Player. Parametre inštalácie a virtuálneho stroja sú v tabuľke 3.1.

Tab. 3.1: Parametre inštalácie

Operačný systém	Ubuntu Server 20.0.4.1 LTS
Počet jadier CPU	2
RAM	4 GB
HDD	20 GB
Webový server	Apache
Databáza	MariaDB 10.3.25
PHP	7.4.3

3.2 Inštalácia cloudového úložiska Nextcloud

Nextcloud server je na systéme Ubuntu možné nainštalovať niekoľkými spôsobmi. Niektoré sú jednoduchšie, ale obmedzujú možnosti konfigurácie počas inštalácie a aj v budúcnosti. Iné sú zložitejšie, ale poskytujú viac možností.

Medzi najjednoduchšie patrí inštalácia pomocou aplikácie Snapd, ale možnosti počas inštalácie sú obmedzené. Táto inštalácia by mala nainštalovať aj väčšinu závislostí. Ďalšia jednoduchá možnosť je pomocou skriptu od Nextcloud VM Appliance, ktorá poskytuje skripty, ktoré prevedú inštaláciou a nastavením serveru [34].

Hlavná a jediná oficiálne odporúčaná možnosť je podľa dokumentácií, pomocou .tar alebo .zip archívu Nextcloudu. V dokumentácii je priložený návod na inštaláciu

s nastavením a získaním všetkých závislostí. Tento postup je najzložitejší a inštalácia prebieha podľa neho.

3.3 Nastavenie úložiska

Pred začiatkom inštalácie nutných závislostí je vhodné aktualizovať všetky balíčky v operačnom systéme. Následne sa nainštalujú závislosti pomocou príkazov:

```
$ sudo apt install apache2 mariadb-server libapache2-mod-php7.4
$ sudo apt install php7.4-gd php7.4-mysql php7.4-curl
                   php7.4-mbstring php7.4-intl
$ sudo apt install php7.4-gmp php7.4-bcmath php-imagick
                   php7.4-xml php7.4-zip
```

Databáza sa spustí pomocu príkazov:

```
$ sudo /etc/init.d/mysql start  \#spustenie
$ sudo mysql -uroot -p  \#zaciatok editovania
```

Následne sa v nej pustia SQL príkazy pre vytvorenie databázy a užívateľa, ktorému sa pridajú práva k manipulácií s databázou. Pomocou tohto užívateľa sa neskôr bude Nextcloud pripájať k databáze, takže je nutné mu dať silné heslo. Časti v zátvorkách sú všeobecné a boli nahradené údajmi v tabuľke 3.2:

Výpis 3.1: Nastavenie databázy Nextcloudu

```
CREATE USER '<užívatel>'@'<IP adresa>'                                  1
IDENTIFIED BY '<heslo>';                                                2
CREATE DATABASE IF NOT EXISTS <názov databázy>                          3
CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci;                      4
GRANT ALL PRIVILEGES ON <názov databázy>.*                              5
TO '<užívatel>'@'<IP adresa>' WITH GRANT OPTION;                        6
FLUSH PRIVILEGES;                                                       7
Quit;                                                                   8
```

Tab. 3.2: Parametre pre databázu

Užívateľ	DBuser
Heslo	S3kj::r?q[Vr13GgDD
IP adresa	127.0.0.1
Názov databázy	Nextcloud

Ako ďalší krok je nutné stiahnuť archív pre Nextcloud server zo stánok Nextcloudu. Je možné overiť si integritu a autenticitu stiahnutého súboru pomocou podpisov SHA256, MD5 a programu PGP, ku ktorým Nextcloud poskytuje overovacie hodnoty. Po obstaraní archívu je nutné ho rozbaliť a premiestniť rozbalený súbor:

```
unzip nextcloud-20.0.2.zip
mv nextcloud /var/www
```

Keď sú zaobstarané závislosti a archív Nextcloudu, prejde sa na nastavenie webového serveru, ktorým je v tomto prípade Apache. Treba vytvoriť konfiguračný súbor v zložke „/etc/apache2/sites-available/“:

```
$ cd /etc/apache2/sites-available/
$ nano nextcloud.conf
```

Do tohto súboru sa zapíše základná konfigurácia:

```
Alias /nextcloud "/var/www/nextcloud/"
<Directory /var/www/nextcloud/>
  Require all granted
  AllowOverride All
  Options FollowSymLinks MultiViews
  <IfModule mod\_dav.c>
    Dav off
  </IfModule>
</Directory>
```

Spustenie stránky:

```
a2ensite nextcloud.conf
```

Dodatočná konfigurácia serveru - odporúčaná dokumentáciou pre korektnú funkčnosť:

```
a2enmod rewrite
a2enmod headers
a2enmod env
a2enmod dir
a2enmod mime
service apache2 restart
```

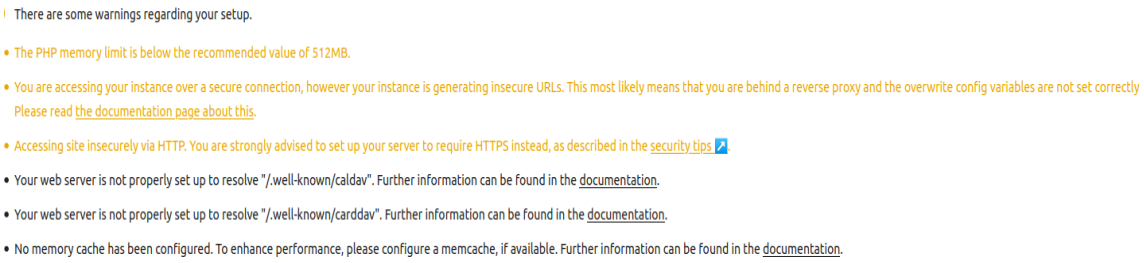
Po tomto sa spustí zabezpečenie pomocou SSL pre pripojenie na stránku. Certifikát pre túto stránku je len základný self-signed certifikát, ktorý poskytuje webový server. Na testovacie účely to postačuje.

```
a2enmod ssl
a2ensite default-ssl
service apache2 reload
```

V tomto bode je možné pre inštaláciu zvoliť 2 možnosti. Jedna je grafickým rozhraním cez prehliadač a druhá je pomocou príkazového riadku. V tejto práci je

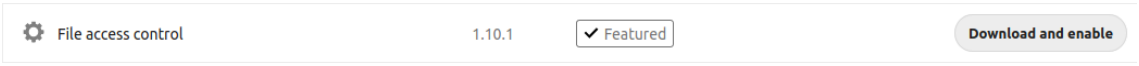
znázornená inštalácia pomocou grafického rozhrania, nakoľko príkazový riadok tu nemá žiadny benefit.

Po zadaní do prehliadača „https://localhost/nextcloud“ sa dostaneme na prvé menu, kde sa vytvorí administrátorský účet a pripojí sa na databázu. Je možné zakliknutie inštalácie odporučených aplikácií. Aplikácie nie sú potrebné pre ciele tejto práce, preto toto políčko ostane nezaškrtnuté. Po vytvorení sa dostaneme na užívateľské rozhranie Nextcloudu. Toto bola len základná inštalácia a samotný Nextcloud upozorňuje na nedostatky v konfigurácií, ako je vidieť na obrázku 3.1.



Obr. 3.1: Nedostatky v konfigurácií Nextcloud serveru

Nextcloud má po inštalácii takmer väčšinu funkcionalít, ktoré požadujeme. Ďalšie sa dajú jednoducho pridať v sekcii „apps“, kde sa dajú jednoducho stiahnuť a doinštalovať, ako je možné vidieť na obrázku 3.2.



Obr. 3.2: Pridávanie ďalších aplikácií do Nextcloud serveru

Takisto v tejto sekcii je možné vypnúť podporu nepotrebných aplikacií. Vypli sme všetky nepotrebné aplikácie ako editovanie textových dokumentov či prezeranie správ alebo pdf súborov. Primárny cieľ tohto úložiska je, aby sa naň zapisovali zašifrované súbory, kde takýto náhľad nie je možný. V tomto prípade sme spustili alebo nechali spustené nasledovné aplikácie:

- *Activity* – užívateľ môže vidieť aktivitu spojenú s ich súbormi v Nextcloude,
- *Auditing/Logging* – poskytuje možnosť logovať prístupy k súborom alebo iné citlivé akcie,
- *Brute-force settings* – ochrana serveru a užívateľov proti útokom hrubou silou na heslá,
- *Collaborative tags* – umožňuje štítkovanie súborov užívateľmi,
- *Workflow external scripts* – administrátor môže posielať súbory do externých skriptov pomocou definovaných pravidiel,

- *Contacts interaction* – zbiera dáta o užívateľoch a kontaktoch a na základe nich poskytuje adresný zoznam pre dáta,
- *Default encryption module* – umožňuje šifrovanie dát na lokálnom úložisku servera,
- *End-To-End Encryption* – povoľuje šifrovanie a dešifrovanie dát na koncových zariadeniach,
- *File access control* – zaisťuje kontrolu prístupu k súborom na základe pravidiel,
- *File sharing* – zdieľanie súborov v rámci Nextcloudu,
- *Files automated tagging* – automatické označovanie súborov na základe definovaných pravidiel,
- *Group folders* – vytváranie zdieľaných priečinkov pre skupiny,
- *Log Reader* – umožňuje jednoduché čítanie logov administrátorovi,
- *Monitoring* – poskytuje užitočné informácie o serveri,
- *Notifications* – umožňuje užívateľom dostávať notifikácie a iným aplikáciám umožňuje ich odosielanie,
- *Password policy* – nastavenie požadovanej heslovej politiky,
- *Privacy* – umožňuje vidieť, kde sú dáta uložené a kto k ním môže pristúpiť,
- *Right click* – poskytuje možnosti užívateľom a vývojárom mať menu pri kliknutím pravým tlačidlom myši na súbor či priečinok,
- *Two-Factor TOTP Provider* – umožňuje dvoj-faktorovú autentizáciu pomocou systému TOTP,
- *Update notification* – upozornenia o dostupnosti aktualizácii pre Nextcloud,
- *Versions* – táto aplikácie automaticky uchováva staršie verzie súborov pri prepísaní, aby ich mohol užívateľ znova obnoviť.

3.4 Funkcionalita

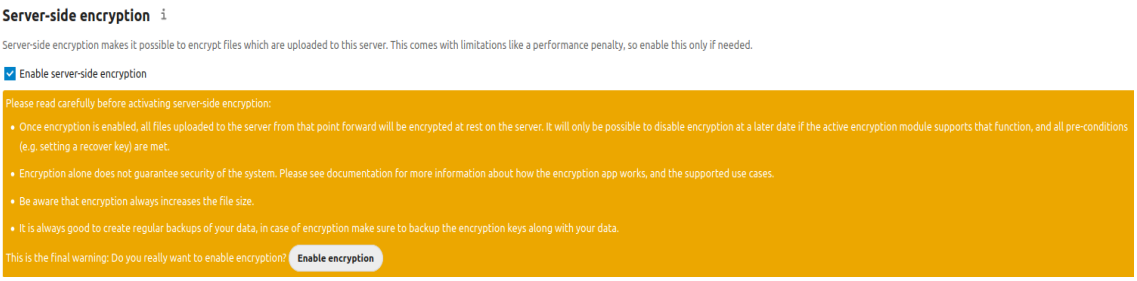
Táto podkapitola sa venuje dostupnej funkcionalite k dosiahnutiu cieľov tejto práce. Popíšu sa tu aplikácie, ktoré je nutné nastaviť skôr ako sa začnú využívať. Ďalej sa tu popíšu postupy ako je možné nahrávať zašifrované súbory, pridávať užívateľov a skupiny, nastavovanie ich právomocí, zdieľanie súborov a možnosti realizácie archivovania súborov.

3.4.1 Šifrovanie súborov

Šifrovanie na strane serveru

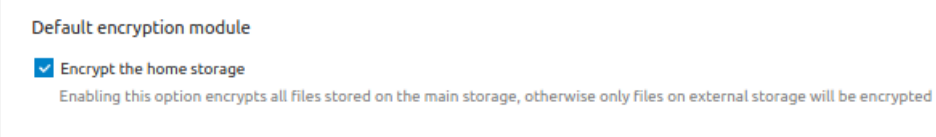
Po stiahnutí a zapnutí aplikácii k podpore šifrovania je nutné v administratívnych nastaveniach v sekcii bezpečnosť zapnúť šifrovanie na strane serveru. Toto šifrovanie

je náročnejšie na výkon a pamäť, nakoľko súbory budú vždy väčšie ako originálne varianty. Na daný fakt je užívateľ upozornený pri zapnutí, pozri obrázok 3.3.



Obr. 3.3: Zapnutie šifrovania na strane serveru

Je vhodné skontrolovať, či je zapnutý aj modul „Default encryption module“, pozri obrázok 3.4

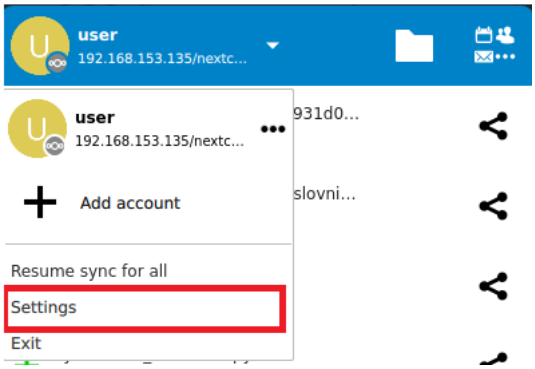


Obr. 3.4: Overenie zakliknutia šifrovania lokálnej pamäte

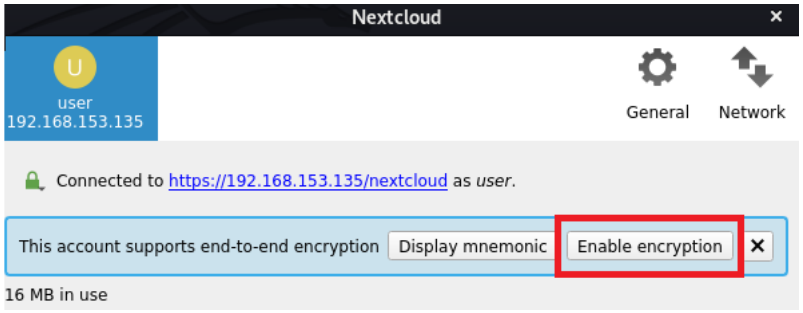
V tomto bode sú všetky nahrané súbory uložené zašifrovane na lokálnom disku serveru, ale aj na prípadných vzdialených diskoch.

Ukladanie súborov pomocou šifrovania na strane klienta

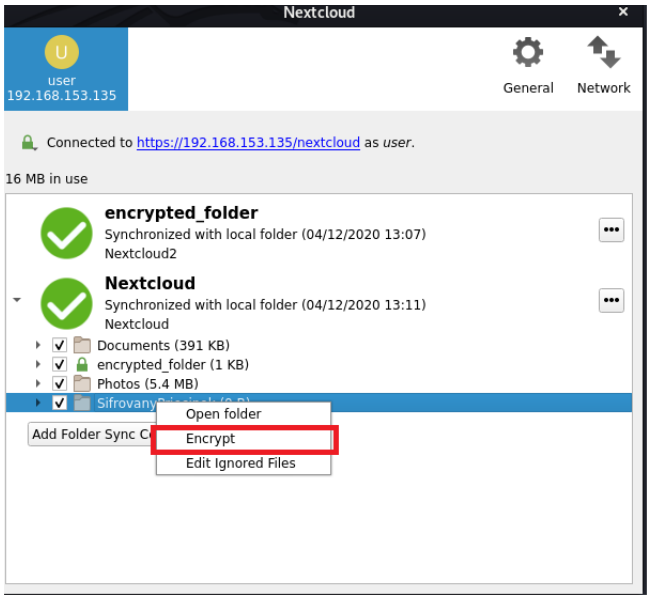
Po stiahnutí a spustení šifrovania na strane klienta nie je nutné žiadne ďalšie nastavenie od administrátora. Potom si užívateľ musí stiahnuť klientsku aplikáciu zo stránok Nextcloudu. Tu sa musí prihlásiť a prejsť do nastavení, pozri obrázok 3.5. Tu povolí šifrovanie, ako je vidieť na obrázku 3.6. Následne si vytvorí v synchronizačnej zložke pre klienta súbor, kde zapne šifrovanie, toto je možné vidieť na obrázku 3.7. Pred povolením je vhodné si pozrieť „mnemonic“, ktorý sa skladá z 12 slov a slúži k možnosti synchronizovania zašifrovaných súborov v iných klientoch. Je možné ho znova pozrieť pri reštarte aplikácie. Do tohto súboru už nebude môcť vkladať dáta cez prehliadač len pomocou klienta. Všetky dáta vložené do tohto priečinka budú zašifrované pred odoslaním na server. Takýto súbor je v klientovi označený zeleným zámkom. Na obrázku 3.8 a 3.9 je vidieť ako je zobrazený súbor v priečinku a v prehliadači po nahraní na server so zapnutým šifrovaním na strane klienta.



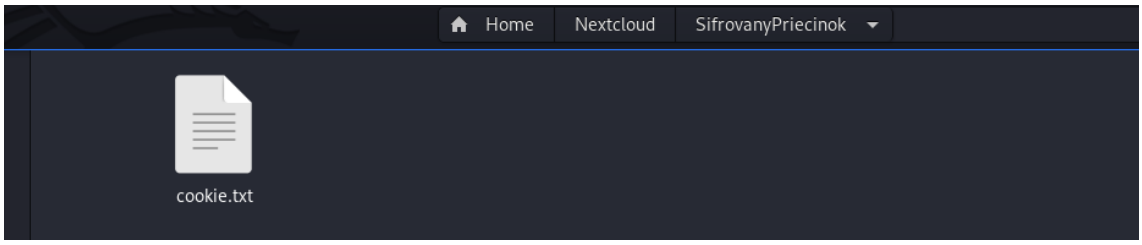
Obr. 3.5: Vstup do nastavení z klienta



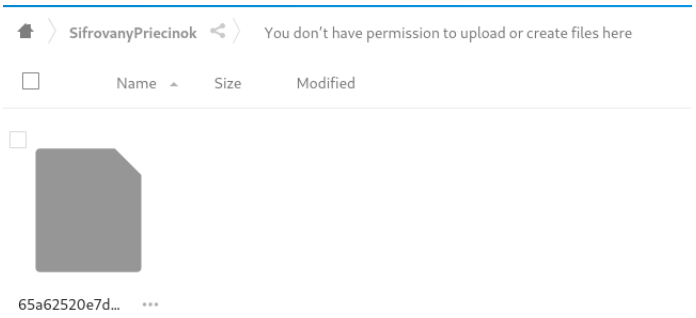
Obr. 3.6: Zapnutie šifrovania na strane klienta



Obr. 3.7: Zapnutie šifrovania pre priečinok



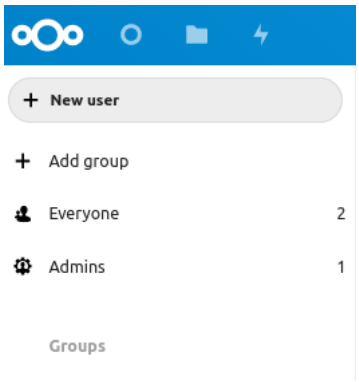
Obr. 3.8: Nahranie súboru do šifrovaného priečinku



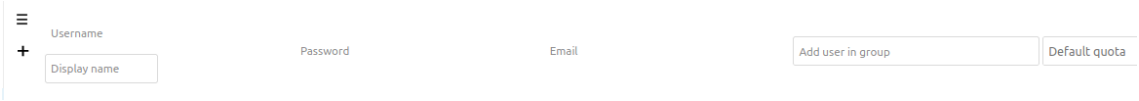
Obr. 3.9: Zobrazenie šifrovaného súboru v prehliadači

Pridávanie užívateľov a skupín

Pridávanie skupín a užívateľov má v našej inštancii možnosť iba administrátor. On ich môže pridávať v sekcií „users“. Tu je na ľavej strane lišta, pozri obrázok 3.10. Pre pridanie novej skupiny stačí kliknúť na „add group“ a napísať názov skupiny. Potom sa naskytne možnosť pridať do nej užívateľov. Pre pridanie užívateľa je tlačidlo „New user“, kde sa nastaví užívateľove meno, počiatočné heslo, ktoré si môže neskôr zmeniť a email. Podľa možností sa môže rovno priradiť do skupiny. Nakoniec je možnosť mu pridať maximálnu kvótu dát, ktoré môže nahrať. Je tu šanca aj aplikácie, ktorá umožní prístup k vybraným súborom ako hosť alebo možnosť zapnúť registráciu nových užívateľov pri prístupe na stránku. Nakoľko cieľom tejto práce je mať maximálnu kontrolu nad dátami a prístupom k nim, tieto aplikácie nie sú spustené.



Obr. 3.10: Pridávanie užívateľov a skupín

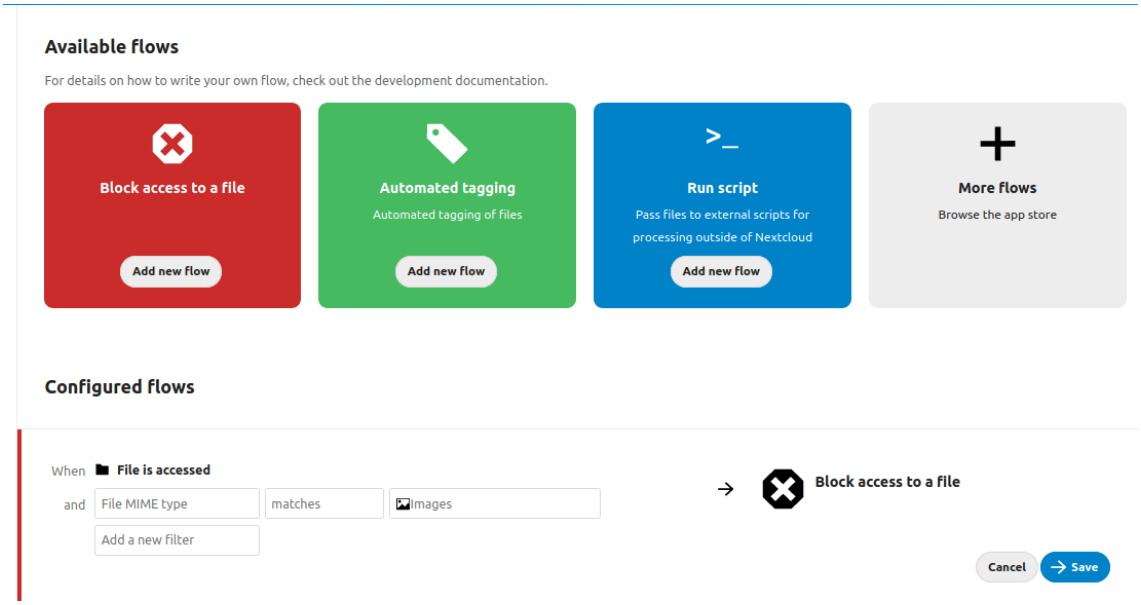


Obr. 3.11: Pridávanie užívateľa

3.4.2 Nastavovanie právomocí

Aplikácia „File access controll“

K samotnému definovaniu pravidiel je potrebné mať nainštalovanú aplikáciu „File access control“, ktorá sa ďalej nastavuje v administrátorských nastaveniach v sekcii „Flows“. Tu sa klikne na „Add new flow“ v kocke, ktorá vraví „Block access to a file“ a následne sa definujú pravidlá na základe rôznych parametrov. Tie sú: kolaboratívny štítok súboru, typ MIME súboru, názov súboru, veľkosť súboru, prístupová IPv4 alebo IPv6 adresa, čas a časová zóna odoslanej požiadavky, požadovaná URL, typ klienta odosielajúceho požiadavku a členstvo v skupine.

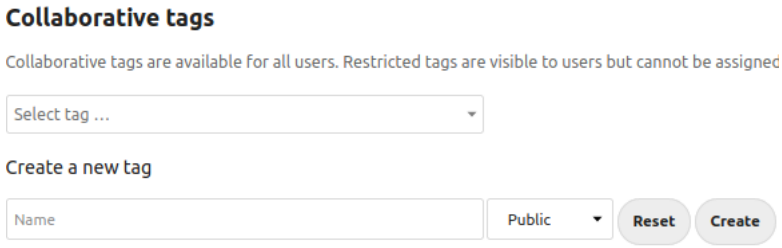


Obr. 3.12: Riadenie prístupu

Kolaboratívne štítky

K nastavovaniu právomocí môžu slúžiť aj štítky, ktoré môžeme definovať jednotlivým súborom alebo priečinkom a na základe týchto štítkov určovať právomoci. Štítky sa vytvárajú v administrátorských nastaveniach v sekcii základných nastavení. Tu sa zvolí názov štítku a jeho typ. Typy sú:

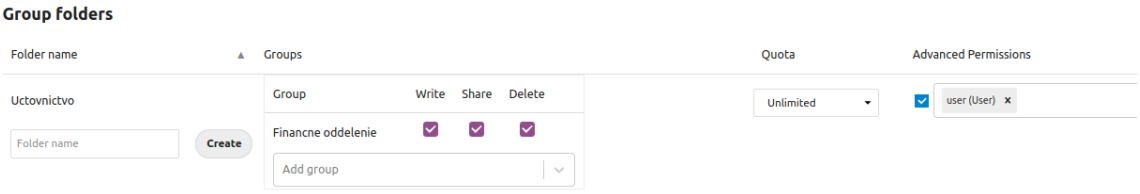
- public (verejný) - tento typ bežný užívatelia sami vidia a môžu ho priradiť,
- restricted (obmedzený) - užívatelia ho vidia, ale nemôžu ho sami priradiť, túto právomoc majú len členovia administrátorskej skupiny,
- invisible (neviditeľný) - užívatelia ho nevidia a ani ho nemôžu priradiť.



Obr. 3.13: Vytváranie štítkov

Skupinové priečinky

Pomocou aplikácie „Group folders“ sa dajú vytvárať skupinové priečinky, kde je možné pridať prístup niekoľkým skupinám a nadefinovať im právomoci k úprave, zdieľaniu alebo vymazaniu. Pokiaľ nebude zaškrtnuté ani jedno pole, skupina bude mať právo len k čítaniu.

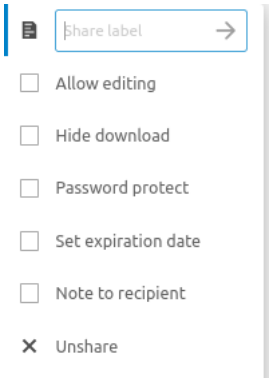


Obr. 3.14: Zdieľané priečinky

3.4.3 Zdieľanie súborov

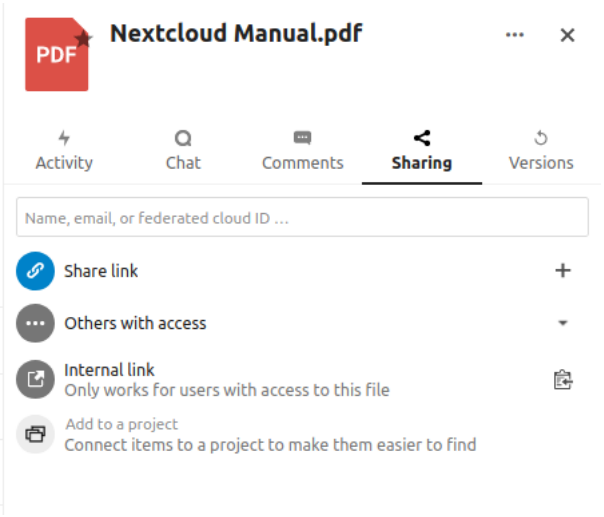
Jednou z možností zdieľania je pomocou zdieľaných priečinkov, ktorých funkcionalita bola popísaná v podkapitole *Skupinové priečinky*. Ďalšou z možností je zdieľanie

jednotlivých súborov. Tie je možné zdieľať pomocou linku, interného linku alebo mena/emailu užívateľa cloudového úložiska. Pri zdieľaní je možné definovať právomoci editovania súboru. Takisto je možné vytvoriť zdieľanie chránené heslom a expiračným dátumom, po ktorom už toto zdieľanie bude ukončené.



Obr. 3.15: Nastavovanie právomocí pri zdieľaní súboru

K zdieľaniu sa dá dostať pri kliknutí pravým tlačidlom na súbor alebo priečinok a klinútím na možnosť informácie. Následne sa otvorí lišta, kde je možné definovať pravidlá. Táto lišta je znázornená na obrázku 3.16.



Obr. 3.16: Možnosti zdieľania

Zdieľanie súborov zašifrovaných na strane klienta

Zdieľanie súborov zašifrovaných na strane klienta je komplikovanejšie. Táto funkcionalita nevyzerá byť úplne dokončená, nakoľko je pomerne nová. Aj napriek tomu, že Nextcloud túto možnosť uvádza vo svojich propagačných materiáloch, neposkytuje ku jej používaniu žiadne oficiálne návody. Pomocou pokusov bol objavený nasledovný neoficiálny návod ako túto funkcionalitu používať:

1. užívateľ 1 vytvorí priečinok a vynúti synchronizáciu,
2. užívateľ 1 zapne šifrovanie priečinku a nazdieľa priečinok užívateľovi 2,
3. užívateľ 2 synchronizuje priečinok a pridá do neho falošný súbor,
4. užívateľ 2 pridá do tohto priečinku falošný súbor, ten súbor užívateľ 1 nikdy neuvidí. Užívateľ 2 následne vynúti synchronizáciu,
5. užívateľ 1 následne môže nahrať citlivý šifrovaný súbor, ktorý chce zdieľať s užívateľom 2,
6. užívateľ 2 synchronizuje klient a bude môcť pristúpiť k citlivému súboru.

Takýto spôsob funguje len pri použití klientskych aplikácií od verzie 3.0 vyššie. Takisto užívateľ 2 nemôže odstrániť falošný súbor, lebo sa naruší šifrovanie. Po vytvorení takto zdieľaného súboru sa dajú šifrované dáta jednoducho zdieľať ďalším užívateľom. Nakoľko sa funkcionalita vyvíja, tento postup sa môže zjednodušiť.

3.4.4 Možnosti realizácie archivovania súborov

Nextcloud nemá priamo funkcionalitu, ktorá by archivovala súbory, ale pomocou spomínaného štítkovania súborov je možné označovať súbory a na základe toho manipulovať s takýmito súbormi. Označovanie súborov štítkami sa dá robiť aj pomocou aplikácie „Files automated tagging“, kde sa dajú definovať pravidlá rovnako ako pri obmedzení prístupu, ktoré bolo spomínané v predošlých kapitolách. Ďalej pre prácu s takto označenými súbormi sa dá použiť aplikácia „Workflow external scripts“, kde je možné na základe definovaných pravidiel posielať súbory do externých skriptov mimo Nextcloudu. Tieto pravidlá sa dajú definovať rovnako ako v aplikáciách „Files automated tagging“ alebo „Files access control“. Možné riešenie je oštítkovaním súborov, ktoré chceme archivovať a externými skriptami ich uložiť na ďalšom úložisku alebo s ním vykonať ďalšie úkony. Bohužiaľ v dokumentácii pre aplikáciu na prácu s externými skriptami je uvedené, že nie je kompatibilná so šifrovanými súbormi. Sú možné minimálne dve riešenia na základe požiadaviek pre archivovanie súborov. Jedným z nich je vytvorenie novej aplikácie ako rozšírenie Nextcloudu alebo vyskúšať pracovať s aplikáciou „Workflow external scripts“, kde by sa definovali len externé skripty.

3.5 Meranie výkonnosti

Meranie výkonnosti serveru zmeriame pomocou posielania požiadaviek PROPFIND, GET a PUT. K tomu nám poslúži záťažový skript „oc-stress.php“ voľne dostupný v projekte ownCloud. Pri teste sa posielal rôzny počet požiadaviek a sledovať zmenu počtu vybavenia požiadaviek za jednu sekundu. Hodnoty budú zaznamenané do grafu. Hodnoty závisia nielen od bežiacich aplikácií na danom serveri, ale aj od samotného výpočtového výkonu zdrojov poskytnutých pre virtuálny stroj, na ktorom je server hosťovaný. V tomto scenári nehrá rolu rýchlosť pripojenia, nakoľko testy budú prebiehať v rámci serveru. Z vykonaných meraní bolo zistené, že server nemá problém zvládať veľké počty požiadaviek, pokiaľ ich neprichádza viac naraz. Nemá to veľký vplyv na rýchlosť odbavených požiadaviek, ale na ich úspešnosť. Čím viac paralelných požiadaviek prichádza, tým je menšia úspešnosť vybavenia požiadaviek. Namerané hodnoty sú v prílohe. Z týchto hodnôt bola vypočítaná priemerná úspešnosť vybavenia požiadaviek na aplikáciu, ktorá je zobrazená v tabuľke 3.3. Nad 150 paralelných požiadaviek začína server so zlyhávaním úspešného odbavovania týchto požiadaviek. V tomto bode sa občasne začínal vracať chybový kód 301 alebo 302, ktorý patrí k protokolu http.

Tab. 3.3: Úspešnosť vybavenia požiadaviek

Typ požiadavky	Maximálny počet paralelných požiadaviek	Úspešnosť [%]
PUT	5	94,875
PUT	10	93,521
GET	5	95,104
GET	10	94,688
PROPFIND	5	95,583
PROPFIND	10	92,188

Tento test sme sa pokúsili overiť vykonaním ďalšieho testu, kedy sme pomocou aplikácie Apache JMeter posielali opätovné požiadavky na obnovenie stránky, ktorá zobrazuje súbory. Jednalo sa o požiadavky typu GET, ktoré boli posielané z oddeleného virtuálneho stroja. Priemerná veľkosť prenášaných požiadaviek bola 7454,5 bytov. Namerané hodnoty je možné vidieť v tabuľke 3.4. Z tohto scenáru testu sa nepodarilo replikovať problém s neúspešnosťou odbavenia pri zvýšenej paralelizácií. Jediný vplyv zvýšenia paralelných požiadaviek bol na rýchlosť odozvy, s ktorou server reagoval.

Tab. 3.4: Meranie výkonu Nextcloudu

paralelné požiadavky	odoslané požiadavky	min. čas [ms]	max. čas [ms]	priemerný čas [ms]	chyby [%]
5	2000	200	942	591	0
5	4000	200	942	599	0
10	2000	268	1645	1195	0
10	4000	219	1645	1188	0
20	2000	252	3324	2389	0
40	2000	507	7129	4781	0
80	1600	1110	14560	9440	0

Nepodarilo sa vykonať podobné testy s použitím požiadaviek POST, PUT, PROPFIND. V dôsledku komplikovaného zabezpečenia Nextcloudu je náročné nastaviť JMeter, aby úspešne opakoval správy pre vkladania dát na server.

Nakoniec sme ešte skúsili maximálnu rýchlosť nahrávania veľkých súborov a následného sťahovania, pričom maximálna rýchlosť nahrávania bola 80.9 MB/s a sťahovania 50.7MB/s. Tento test bol vykonaný manuálne pomocou jedného užívateľa.

Testy sa odohrali na neoptimalizovanej inštancii Nextcloudu. Takisto na virtuálnom servery, kde bežal Nextcloud boli spustené aj iné aplikácie. Na základe týchto poznatkov sa dá usúdiť, že výsledky nepoukazujú na maximálny potenciál aplikácie.

4 REALIZÁCIA DLHODOBEJ ARCHIVÁCIE

Nextcloud neposkytuje funkcionalitu zabezpečenia dlhodobej archivácie súborov a zabezpečenia ich integrity počas tejto doby. Nakoľko možnosť dlhodobej archivácie je jedným z hlavných požiadaviek v tejto práci, je nutné vytvoriť dodatočný systém, ktorý túto možnosť poskytne. V tejto kapitole bude popísaný návrh a implementácia tohto systému.

4.1 Teoretický návrh archivácie

Cieľom systému pre archiváciu súborov je bezpečné uchovanie kópie súboru na oddelenom médiu, oddelenom disku alebo úplne inom serveri. Tieto kópie musia byť identické s originálnymi dátami a musíme byť schopní zabezpečiť dôkaz aj po dlhej dobe. Dlhá doba znamená, že dáta musia ostať nedotknuté počas niekoľkých rokov. Táto práca sa zameriava na softvérovú časť riešenia, nakoľko chyby hardwaru sa veľmi ťažko predvídajú a je nutné mať vždy zálohy. Hlavným problémom pri dlhodobej archivácii súborov je, že dôkaz o ich integrite sa opiera o kryptografické protokoly existujúce v dobe vývoju týchto systémov. Problémom je, že v budúcnosti tieto protokoly môžu byť prelomené a znehodnotené. Môže sa tak stať kvôli existujúcej, no doteraz neobjavenej chybe alebo narastajúcemu výpočtovému výkonu. Vďaka tomu sa nemôžeme opierať len o jeden dôkaz, ale poskytnúť ich niekoľko a pravidelne dôkazy obnovovať alebo pridávať pomocou moderných kryptografických systémov. V tejto práci bude implementovaný systém podobný tomu, ktorý sa popisuje v práci [35]. Oba tieto systémy majú za cieľ zabezpečiť platnosť digitálneho podpisu po dlhšiu dobu ako platí certifikát k podpisovaciemu kľúču. Práca sa opiera o tri základné podmienky a to, že:

1. Dokument bol podpísaný medzi dvomi určitými bodmi v čase, vďaka čomu zabránime, aby sa posunul čas podpisu.
2. Obsah dokumentu sa počas doby uloženia nezmenil.
3. Podpis bol vytvorený pomocou podpisovacieho kľúču, ktorý bol platný v dobe vytvorenia podpisu.

K dosiahnutiu prvej podmienky sa využívajú časove pečiatky (timestamp, TS), ktoré sa získajú od vzdialenej časovej autority (TSA). Prvá časová pečiatka sa vypýta pre súbor samotný, respektíve kryptografický haš tohto súboru, pretože súbory môžu byť veľmi veľké a nechceme ich posielať po sieti pre získanie časovej pečiatky. Druhá časová pečiatka sa potom získa pre vytvorený digitálny podpis tohto súboru. Vďaka tomuto zabezpečíme dôkaz o tom, že vytvorenie podpisu sa stalo medzi časmi prvej a druhej pečiatky. Na základe toho vieme určiť čas, v ktorom certifikát a podpisovací

kľuč mal byť platný. Certifikáty majú limitovanú platnosť. K spätnému overeniu platnosti tohto certifikátu sa používa certificate revocation list (CRL), ktorý musí byť obstaraný od certifikačnej autority.

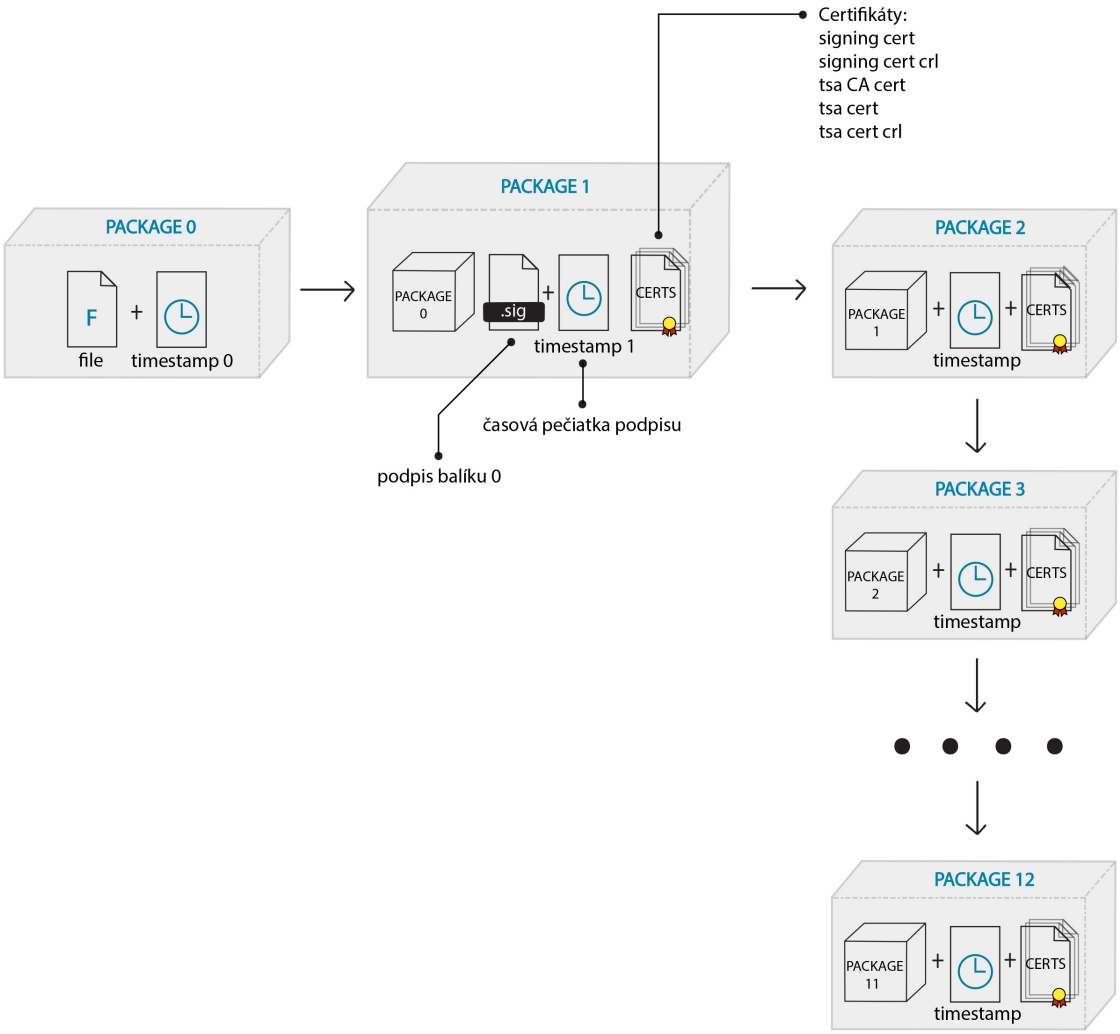
Na rozdiel od systému popisovanom v práci [35], systém v tejto práci opomína klientsku časť. Všetky dôkazy zabezpečuje server. Serverová časť má 3 hlavné úlohy: archiváciu, obnovu časových pečiatok a overenie platnosti dôkazov zaobstaraných v čase.

Archivácia má za úlohu získať súbory, ktoré sa majú archivovať. Pred ich prenesením do svojho úložiska sa pomocou kryptografického hašu vytvorí kontrolný súčet dát aby sa dalo zaručiť, že sa počas prenosu tieto dáta nepozmenili. Následne po prenesení dát sa získa časová pečiatka TS0 od TSA, ktorá nám určuje dobu, kedy sa dáta uložili na server. Tieto 2 súbory sa vložia do balíku „Package0“. Z neho vytvorí kryptografický haš tohto balíku, ktorý sa podpíše podpisovacím kľúčom serveru, ku ktorému bol vydaný certifikát. Týmto zaručíme integritu daných dát. Potom sa vypýta časová pečiatka k podpisu. Táto druha pečiatka nám zabezpečí možnosť overenia, že dáta boli podpísané v nejakom časovom okne kľúčom s platným certifikátom. Predchádzajúcim postupom máme zabezpečenú prvú podmienku. Ďalej sa k týmto dátam priloží certifikát k použitému podpisovaciemu kľúču a CRL získaný od certifikačnej autority. Vďaka tomuto je zabezpečená tretia podmienka. Ku každej novej časovej pečiatke sa uloží aj certifikát a CRL pre TSA. Následne nám vznikne „Package1“.

Nakoľko hlavným cieľom je zabezpečiť dlhodobú archiváciu a tieto dôkazy časom degradujú na kvalite, je nutné pridávať nové časové pečiatky, vďaka čomu sa predĺži spoľahlivosť tvrdení. Týmto spôsobom sa tvoria vrstvy. Vždy po nejakom časovom období sa pridá nová časová pečiatka. Pred jej pridaním sa overí posledná časová pečiatka. Vytvorí sa kryptografický haš celého balíka a vypýta si novú časovú pečiatku od TSA. Ďalej si samozrejme uloží len použité certifikáty TSA. Toto má efekt pokiaľ sa počíta s tým, že TSA používa vždy najnovšie a bezpečné kryptografické protokoly k vytvoreniu časovej pečiatky. Proces sa opakuje vždy po nejakej dobe, ktorú si určíme, napríklad po 2 rokoch alebo keď vieme, že nejaký nami použitý kryptografický protokol bol prelomený. Vďaka tomuto je zabezpečená druhá podmienka.

Server musí byť schopný overiť platnosť tvrdení. Overovanie celého balíku dôkazov prebieha postupným overovaním od poslednej časovej pečiatky. Po jej overení sa balík postupne rozbaľuje až k pôvodnému súboru. [35]

Celá vyššie popísaná štruktúra je znázornená na obrázku 4.1.



Obr. 4.1: Štruktúra obaľovania dát

4.2 Návrh systému

Systém sa skladá z Nextcloud klientov, ktorí budú bežať na strojoch užívateľov Nextcloud serveru s povinnosťou nainštalovania dodatočných aplikácií a samostatného archivačného systému. Nextcloud klienti ani server nebudú musieť obsahovať žiadne zmeny, a teda sa zachovajú výhody prístupu k aktualizáciám od vývojárov. Nextcloud server bude musieť mať nainštalované aplikácie umožňujúce otagovávanie súborov a aplikáciu „workflow external scripts“, ktorá umožňuje spúšťanie externých skriptov a pridávanie parametrov na základe definovaných pravidiel. Aplikáciu bude nutné jemne upraviť, aby poslala potrebné informácie z Nextcloud serveru do archivačného systému. Jedná sa o informácie o vlastníkovi súboru a plnej systémovej ceste k nemu. Táto aplikácia je zverejnená osobitne - spadá pod licenciu GNU AGPLv3, čo znamená, že jej úprava bude musieť byť uverejnená pod tou istou licenciou.

Nextcloud server a klienti budú sprostredkovávať všetky dôležité funkcie ako zabezpečené nahrávanie súborov, šifrovanie súborov či už pred, počas alebo po uložení na server. Takisto bude sprostredkovávať právomoci a možnosti zdieľaných priečinkov - ako bolo popísané v predošlých kapitolách. Nextcloud poskytuje aj možnosti šifrovania dát. Bude možné používať oba spôsoby, teda aj šifrovanie na strane klienta, aj na strane serveru. Vhodnejšie by bolo používať len šifrovanie na strane serveru, nakoľko sú tieto dáta uložené šifrovane a nie je možné sa k ich obsahu dostať jednoducho a názov súboru ostane rovnaký. Takisto pri šifrovaní na strane serveru je jednoduchší manažment kľúčov. Kľúče použité k šifrovaniu na strane serveru je možné zálohovať. O bezpečný prenos dát na Nextcloud serveri sa postará protokol TLS. Aj keď je možné používať šifrovanie na strane klienta, bolo by veľmi náročné zabezpečiť manažment kľúčov užívateľov. Tie sa môžu kedykoľvek zmeniť a nie je možné ich zálohovať. V takom prípade by došlo k strate všetkých uložených a archivovaných dát.

Archivačný systém napojený na Nextcloud bude mať za úlohu archivovať vybrané súbory a zabezpečiť dôkaz o zachovaní ich integrity počas dlhej doby - ako bolo spomenuté v predošlej kapitole. Posledná úloha tohto systému bude mať možnosť spätného overenia pravosti takto archivovaných súborov na základe žiadosti. Tieto súbory budú archivované už zašifrované.

Systém sa bude teda skladať s niekoľkých komponentov. Cieľom je, aby ho bolo možné dobré škálovať a aby mohol bežať či už na jednom alebo aj na rôznych strojoch, pokiaľ sa zabezpečí spojenie prvkov podľa požiadaviek. Prinieslo by to výhodu rýchlejšieho systému, nakoľko operácie so súbormi sú výpočtovo pomerne náročné.

4.2.1 Popis behu systému

Užívateľ nahrá súbor na Nextcloud server, následne ho označí definovaným tagom, vďaka čomu sa spustí externý skript s dodatočnými informáciami. Informácie budú obsahovať predovšetkým cestu k uloženému súboru. Skript vytvorí úlohu na Rabbitmq serveri.

Časť systému zodpovedná za archiváciu bude počúvať na danom kanáli Rabbitmq serveru a konzumovať správy v ňom. V tomto prípade dostane správu na spustenie archivácie súboru na danej ceste. Táto časť bude musieť byť spustená na mieste, kde bude mať prístup k súborovému systému, do ktorého ukladá Nextcloud dáta. Takisto potrebuje prístup k databáze. Na základe konfigurácie sa stanový dĺžka platnosti časovej pečiatky.

V priebehu času nami nastavená platnosť časových pečiatok vyprší. To skript vyrieši Retimestamping task scheduler, ktorý bude zodpovedný za každodenné kontro-

lovanie databázy. V nej bude zisťovať končiacu platnosť časovej pečiatky. V prípade, že by sa blížil koniec, vytvorí sa úloha pre Retimestamping worker na RabbitMQ serveri. Retimestamping worker zoberie daný balík a vytvorí sa preň nová časová pečiatka od aktuálnej autority. Spolu s certifikátom tejto časovej autority sa zabalí do nového balíku.

Overovanie archivovaných súborov sa bude vykonávať osobitne na základe spustenia od správcu, nakoľko by automatizované overovanie mohlo spôsobiť veľmi veľkú náročnosť na výpočtový výkon. Bude možné spustiť manuálne overenie všetkých alebo len vybraných súborov. Zistené výsledky sa následne pošlú na zadané emailové adresy. V budúcnosti to bude možné rozšíriť o novú aplikáciu v Nextcloude, ktorá by vytvárala úlohy pre overenie týchto súborov pri snahe užívateľa k nim pristúpiť a zároveň by mu povedala, či bolo overenie úspešné. Nakoľko rozsah ostatných častí bude časovo náročný, takúto aplikáciu nebude možné stihnúť zrealizovať v tak krátkom čase.

4.2.2 Komponenty systému

Jednotlivé komponenty a ich primárne úlohy v systéme:

- *Nextcloud Server* – jeho primárnou úlohou je komunikácia s klientmi, poskytovanie cloudového úložiska a vyvolávanie archivovania na základe označenia súboru definovaným štítkom.
- *RabbitMQ Server* – jedná sa o takzvaný message broker. Je to veľmi rozšírený open source projekt. Podporuje asynchrónne posielanie sprav, smerovanie správ do zásobníkov, konzolu pre manažment a monitorovanie a mnoho ďalšieho. V tomto systéme je jeho primárnou úlohou zabezpečenie komunikácie medzi jednotlivými komponentmi a distribúcia úloh medzi jednotlivými pracovníkmi. K tomu používa takzvané queues (poradovníky) a exchanges. Je tu možnosť nastavenia šifrovanej komunikácie pomocou TLS medzi jednotlivými komponentmi.
- *Archivation Worker* – archivačný pracovník je konzument úloh s Rabbitmq serveru, kde načúva na definovanom poradovníku a čaká na úlohy. Úlohy v tomto poradovníku slúžia pre realizáciu archivácie súborov, obstaranie dôkazov a vytvorenie kópií ako bolo popísane v predošlej kapitole.
- *Retimestamping Worker* – pracovník pre obnovovanie časových pečiatok je taktiež konzument úloh s Rabbitmq serveru pre kontrolu a následnú obnovu časových pečiatok.
- *Validation Worker* – validačný pracovník slúži pre validovanie archivovaných súborov na základe úloh s Rabbitmq serveru.

- *Databáza pre archiváciu* – obsahuje dáta spojené s archiváciou súborov ako napríklad kontrolné súčty, názvy, cesty, časy a podobne. Tieto informácie sa využívajú jednotlivými pracovníkmi. Hlavným benefitom je, že dôkazy je možné nezávisle overiť s informáciami z databázy.
- *Retimestamping task scheduler* – skript, ktorý bude každý deň kontrolovať v databáze časy platnosti časových pečiatok a ak sa blíži ich vypršanie, pridá úlohu do poradovníku na Rabbitmq serveri pre Retimestamping worker. Konzola pre spustenie validácie – manuálne spustiteľný skript, ktorý vytvorí úlohy na Rabbitmq serveri pre Validation Worker.
- *Archivation task maker* – skript pre vytvorenie úlohy na Rabbitmq serveri pre Archivation worker.
- *Timestamping Authority* – autorita poskytujúca časové pečiatky. Úložisko dát archivovaných súborov – úložisko, do ktorého budú ukladané balíčky s kópiami archivovaných súborov a dôkazy o ich integrite.
- *Úložisko dát Nextcloudu* - úložisko, do ktorého ukladá súbory Nextcloud server.

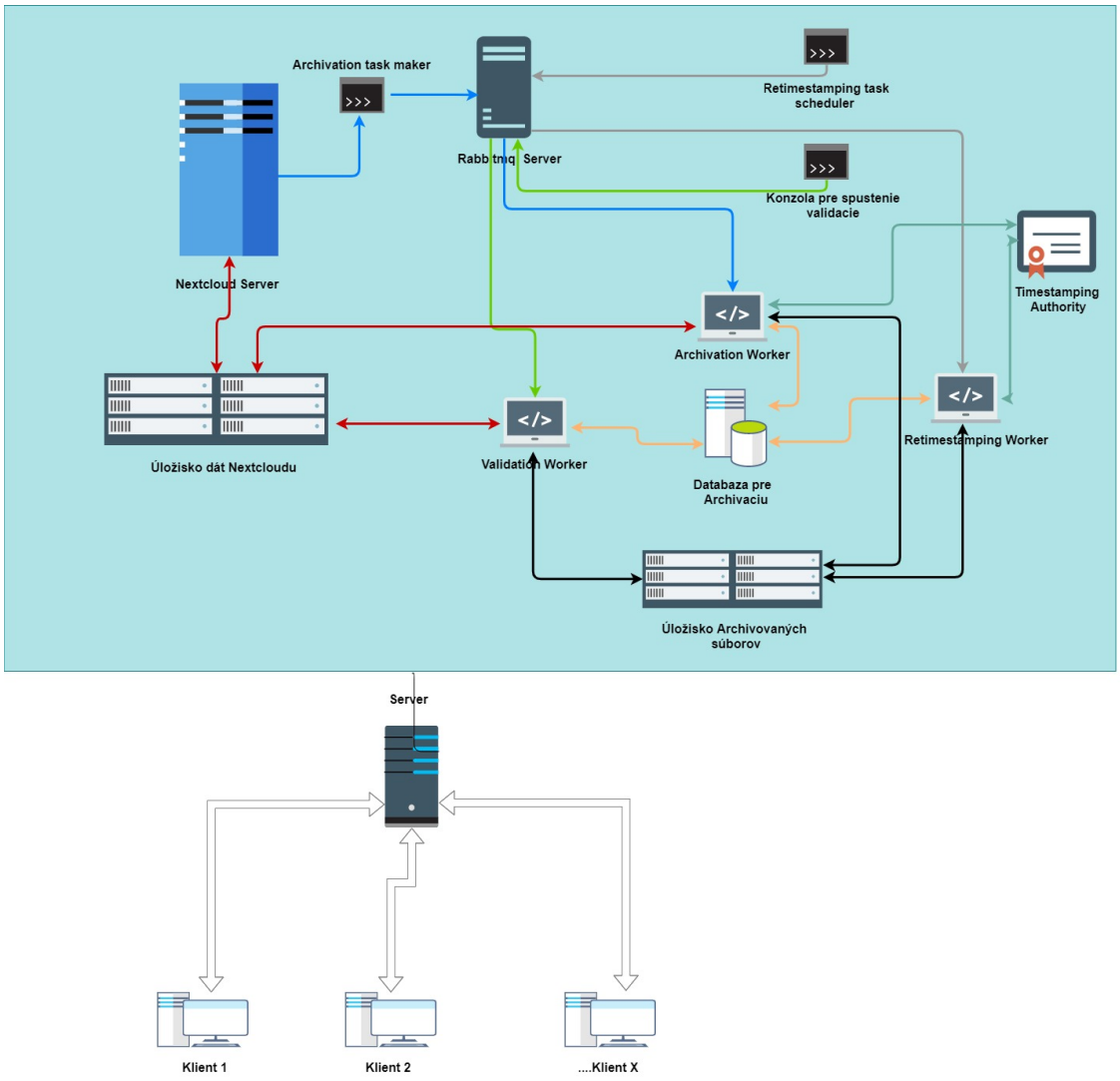
Na obrázku 4.2 je možné vidieť schému kompletného systému. V hornej časti nákresu je možné vidieť obsah serveru. Každá farba spoja medzi komponentmi označuje čo s čím komunikuje, či to, kde musí byť zabezpečené spojenie. „Archivation Wroker“ musí mať prístup k súborovému systému, kde Nextcloud ukladá svoje dáta a podobne.

4.2.3 Popis systému z užívateľského hľadiska

Pri tomto riešení je nutné, aby administrátor nainštaloval vyššie spomenuté aplikácie na inštanciu Nextcloud serveru. Následne definoval tag a pridal užívateľov, ktorí majú oprávnenie ho používať. Potom v aplikácii „workflow external scripts“ vytvorí nové pravidlo, kedy sa pri použití daného tagu na súbor spustí externý skript. Užívateľ, ktorý bude chcieť archivovať súbor a má právomoci k používaniu daného tagu ním označí súbor. O všetko ostatné sa postará Nextcloud spolu s archivačným systémom.

4.3 Implementácia riešenia

Implementácia tohto systému sa skladala z dvoch celkov. Z úpravy Nextcloudovej aplikácie Workflow external scripts a implementácie archivačného systému. Archivačný systém vo svojej podstate nie je nijako viazaný na Nextcloud a je možné ho používať aj inými spôsobmi.



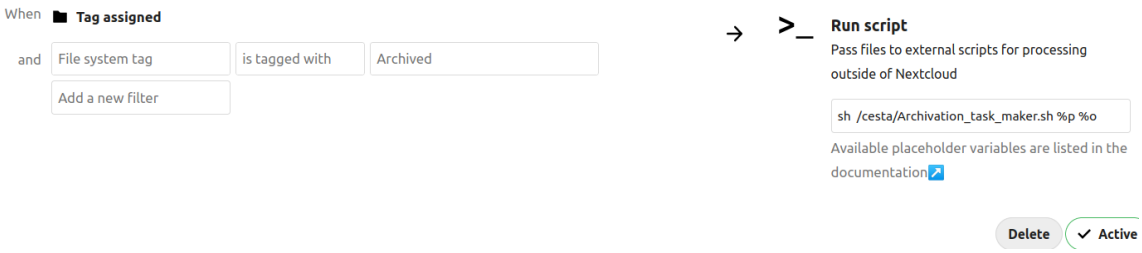
Obr. 4.2: Návrh systému

4.3.1 Úprava Workflow External Scripts

Jedná sa o dodanie funkcionality k získaniu potrebných informácií o súbore. Aplikácia je zodpovedná za spustenie externých skriptov na základe definovaných pravidiel. Pri spustení externých skriptov používa takzvaný placeholder pre pridanie hodnoty z Nextcloudu ako parameter pre daný skript. Pôvodná aplikácia nemala funkčné placeholdere pre získanie plnej cesty k súboru na Nextcloude pri zapnutom šifrovaní na strane serveru. Túto funkcionalitu bolo nutné pridať.

Bola pridaná možnosť získať plnú systémovú cestu k súboru a meno daného súboru. Plnú systémovú cestu predáme pomocou parametru „%p" a celé pôvodné meno súboru pomocou „%m". Celá táto zmena bola urobená len v súbore operations.php. Všetky pôvodné funkcie boli zachované. Ako príklad sa používa spustenie skriptu na

vytvorenie úloh na Rabbitmq serveri. To nastane, keď niekto označí nejaký súbor štítkom „Archived". Príklad použitia príkazu v aplikácií workflow external scripts je možné vidieť na obrázku 4.3:



Obr. 4.3: Príklad použitia upravenej aplikácie pre Nextcloud

Zapnutie tejto aplikácie je možné niekoľkými spôsobmi. Najjednoduchšie je aktivovať ju cez Nextcloud a následne upraviť obsah zložky, kde je táto aplikácia (/<zložka inštalácie nextcloudu>/apps/workflow_script/), prekopírovaním upravených súborov. Upravený bol len súbor operations.php. Otestovaný bol vo verzii Nextcloudu 20.0.7 a verzia Workflow external scripts bola 1.5.1. Táto aplikácia je v prílohách pod názvom „Script_trigger". Aplikácia sa spúšta pod užívateľom www-data. Je nutné aby mal tento užívateľ oprávnenia k prístupu a spúštaniu skriptu pre tvorbu archivačných úloh.

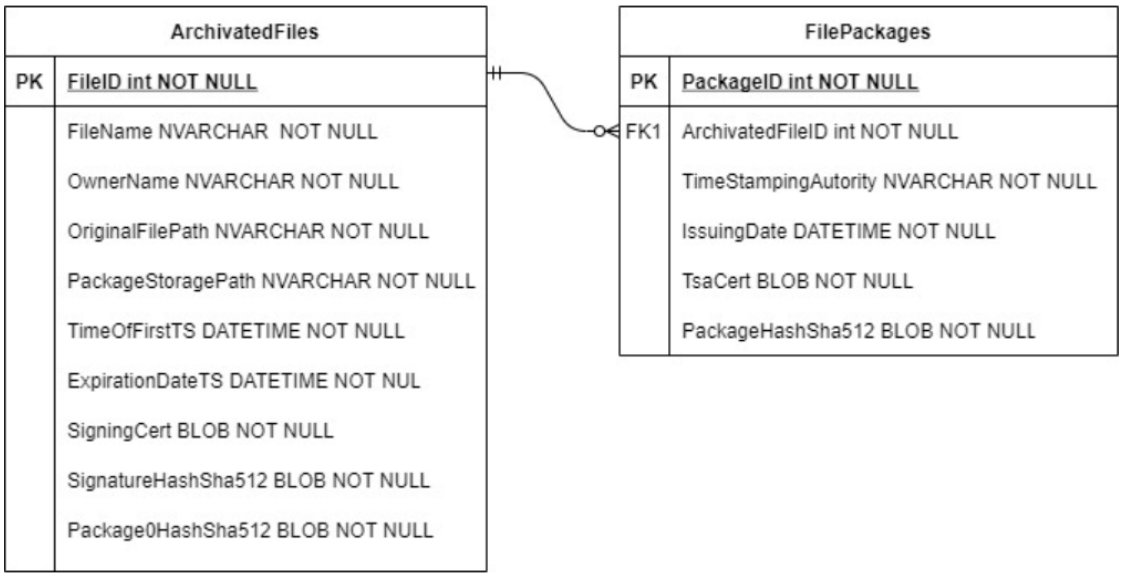
4.3.2 Implementácia Archivácie

V tejto kapitole budú popísané jednotlivé časti systému a ich implementácia.

Databáza

V systéme je nutná existencia databázy uchovávajúca informácie o archivovaných súboroch. Záznamy do databázy sa pridávajú počas behu jednotlivých pracovníkov. K realizácií sa použila databázová služba MySQL, v ktorej sa vytvorili tabuľky podľa ERD diagramu, ktorý je možný vidieť na obrázku 4.4. Databáza sa skladá z dvoch tabuliek - „ArchivatedFiles" a „FilePackages". Tabuľka „ArchivatedFiles" slúži k uchovávaniu informácií ohľadom jedného archivovaného súboru. Počas doby archivovania sa tento súbor obaľuje do balíkov a informácie o jednotlivých balíkoch sa vkladajú do tabuľky „FilePackages". Vzťah medzi tabuľkami je jeden ku mnohým. Databáza by mala byť prevádzkovaná na oddelenom stroji ako všetky ostatné komponenty systému. V kóde je spravená takzvaná knižnica s funkciami k databáze, ktoré jednotlivé moduly používajú. K pripojeniu tejto knižnice k vytvorenej databáze je nutné posunúť konfiguračný súbor podľa špecifikácie od firmy MySQL pre

vytvorenie spojenia. Odkaz k tejto konfigurácii je v prílohe s návodom aj vo vnútornej dokumentácii kódu. Databázu je nutné vytvoriť pred spustením ktoréhokoľvek z modulov. K jej vytvoreniu sú v prílohách dve sql queries, ktoré je nutné spustiť. Jeden pre vytvorenie databázy užívateľa a pridania právomocí užívateľovi a druhý k vytvoreniu jednotlivých tabuliek.



Obr. 4.4: ERD databázy

Komunikácia medzi modulmi a RabbitMQ

Každý modul je oddelený a je možné ho spustiť samostatne. To prináša výhodu, že pracovníci, ktorí spotrebovávajú časť výpočtového výkonu môžu byť spustení samostatne na oddelených strojoch a môže ich byť niekoľko. Prináša to vyššiu rýchlosť odbavovania úloh, ktoré majú robiť aj istú mieru redundancie v prípade chýb, aktualizácií a podobne. Aby existovali takéto výhody je nutné zabezpečiť to, že títo pracovníci vedia, čo majú robiť. Na to slúžia úlohy, ktoré sa tvoria v skriptoch. Tie budú popísane neskôr. K ich distribúcii sa používa RabbitMQ server. Server v tomto riešení hosťuje jednotlivé poradovníky s úlohami. Každý pracovník má definovaný svoj vlastný poradovník úloh. Tento poradovník je typu FIFO, teda prvý dnu a prvý von. Pracovník si vezme úlohu z tohto poradovníka, dokončí ju a následne potvrdí. Táto úloha odtiaľ zmizne. Pokiaľ ju nedokončí kvôli nejakej chybe, úloha ostane v poradovníku pre konzumáciu iným pracovníkom v budúcnosti. Systém ponúka možnosť centrálnej správy a dohľadu nad ním. Vytvára nám mnoho štatistík o chode

sieti, vieme pozrieť aké sú úlohy v jednotlivých poradovníkoch a podobne. Tento systém nepotrebuje pripojenie k vonkajšiemu svetu. K využitiu systému je v práci spravená spoločná knižnica funkcií. Dve najhlavnejšie sú vytvorenie pripojenia k Rabbitmq serveru, ktoré je podstatné pre všetky moduly a ďalšia je k vytvoreniu takzvaného spotrebiteľa. Ten je zodpovedný za konzumáciu úloh s poradovníkov a spúšťanie funkcií, ktoré sú mu predané podľa jeho určenia. Tohto spotrebiteľa implementujú všetci pracovníci. Je možné v ňom doimplementovať rôzne funkcie pre vzdialený manažment pracovníkov, ako napríklad stopnutie konzumácie na nejaký časový okamih a podobne.

Skript na vytvorenie úlohy pre archiváciu súboru

„Archivation task maker“ je bash skript dizajnovaný pre spustenie pomocou upraveného rozšírenia Nextcloudu spomínaného vyššie. Jeho úlohou je spustiť python skript archivation_task.py, ktorý vytvorí úlohu v poradovníku archivácie na Rabbitmq serveri. Takto je to navrhnuté z dôvodu, že aplikácia workflow external scripts nespúšta spoľahlivo python skripty. Bash skript potrebuje k spusteniu parameter cesty k súboru, ktorý sa bude archivovať a meno užívateľa, ktorý súbor vlastní. Pred spustením je nutné v ňom upraviť cestu k python skriptu, ktorý má spúštať. Tento python skript potrebuje cestu ku konfiguračnému súboru, tú treba v bash skirpte taktiež nastaviť. V základe je pre ukážku nastavená cesta používaná počas testovania. Konfiguračný súbor musí obsahovať údaje o pripojení k Rabbitmq serveru. Následne je možné tento bash skript spustiť. Ten spustí python skript definovavný vo vnútri s predanými parametrami. Po spustení automaticky vytvorí úlohu na základe týchto informácií. Pri spúšťaní pomocou workflow external scripts je nutné dobre nastaviť cron, na ktorý sa táto aplikácia spolieha pri vyvolávaní skriptov. Návod k nastaveniu cronu je v dokumentácii Nextcloud serveru. Je to jediný skript, ktorý musí byť pustený na serveri, kde beží aj Nextcloud. Jeho výkonová náročnosť je zanedbateľná, nakoľko nepracuje so žiadnymi súbormi ani nevykonáva žiadne časovo náročné operácie. V prípade, ak chce človek spustit obsiahnutý python skript a spustí ho bez parametrov alebo so zlými parametrami, vyhodí mu to nápovedu, ako ho má použiť. Takúto nápovedu implementujú všetky python skripty slúžiace pre spustenie nejakej z častí systému.

Kontrola platnosti časových pečiatok

Je nutné pravidelne kontrolovať platnosť posledných časových pečiatok pri skriptoch. Na to nám slúži skript „retimestamping task scheduler“. Podľa nastavenia, ktoré sa mu predáva pri spustení opakuje kontroly v pravidelných časových intervaloch. Je

možné mu zadať opakovanie len v hodinových intervaloch. Po uplynutí daného časového úseku sa spustí prvá kontrola. Tá získa všetky záznamy z databázy z tabuľky „ArchivatedFile“, kde porovnáva rozdiel medzi polom „ExpirationDateTS“ (obsahuje čas a dátum konca platnosti časovej pečiatky) a aktuálnym časom, v ktorom skript beží. Ak je čas platnosti menej ako 24 hodín, ID daného záznamu sa pridá do zoznamu archivovaných súborov, ktorých časovú pečiatku bude treba obnoviť. Nakoniec, keď sa prejdú všetky záznamy z databázy, vytvorí sa úloha pre každé jedno ID v poradovníku pre Retimestamping worker na Rabbitmq serveri. Tieto úlohy obsahujú iba ID záznamu archivovaného súboru. Pre spustenie tohto skriptu je nutné mu predať konfiguračný súbor s pripojením na Rabbitmq server a pripojením na databázu. Takisto potrebuje zadať počet hodín, po ktorých sa majú kontroly opakovať. Tento skript je už výpočtovo náročnejší, obzvlášť ak je veľké množstvo záznamov v databáze, aj keď ich nenačítava všetky naraz. Z toho dôvodu je vhodnejšie, aby bežal na stroji oddelenom od Nextcloud serveru.

Konzola pre validáciu archivovaných súborov

V rámci riešenia zadania musí byť spôsob pre správcov alebo užívateľov získať dôkaz o tom, že archivované súbory neboli porušené. K tomu aby sa vykonala validácia je nutné vytvoriť úlohy pre daný pracovník. Tie sa tvoria pomocou konzolovej aplikácie validation task. Tá má za úlohu interagovať s užívateľom a na základe získaných informácií vytvoriť dané úlohy. Túto aplikáciu je nutné spustiť s konfiguračným súborom obsahujúcim dáta k vytvoreniu pripojenia na Rabbitmq serveri, na ktorom sa na základe údajov od užívateľa vytvoria úlohy. Konzola umožňuje zadať validáciu rôzneho množstva súborov. Po spustení sa aplikácia opýta, či chceme dáta o súboroch načítať zo súboru alebo ručne. Pokiaľ zadá, že dáta chce načítať ručne, je nutné napísať číslo - koľko súborov chce dať overiť. Podľa zadaného množstva od neho vypýta údaje k týmto súborom. Tie je možné zadať ručne. Ako údaj sa zadáva ID archivovaného súboru alebo meno súboru a pôvodného vlastníka, nakoľko kombinácia mena súboru a mena vlastníka súboru musí byť vždy jedinečná. Pokiaľ sa vyberie zadanie zo súboru, je nutné zadať cestu k súboru typu yaml, ktorý bude obsahovať list informácií. Rovnako je možné použiť ID alebo meno súboru a vlastníka. Príklad zadania informácií zo súboru a jeho formát je uložený medzi ostatnými príkladmi v riešení pod file_info.yaml. Informácie sa ukladajú do zoznamu a musia byť unikátne. Následne po zadaní informácií o súboroch sa zadajú emailové adresy, na ktoré bude výsledok o validácií odoslaný. Takisto je tu voľba, či ich chceme zadať zo súboru alebo ručne. Ak nechceme adresy zadať zo súboru, musíme rovnako zvoliť číslo o tom, na koľko emailových adries chceme výsledok poslať. Podľa zadaného čísla sa vypýtajú emailové adresy. Pokiaľ sa budú zadávať zo súboru, tak

je nutné zadať cestu k súboru typu txt, kde bude jedna emailová adresa na jeden riadok. Pre úspešné dokončenie musí byť zadaný aspoň jeden súbor pre validáciu a aspoň jedna emailová adresa, na ktorú bude výsledok o nej odoslaný. Po zadaní všetkých dát je ešte možnosť proces opakovať a vybrať ďalšie dáta alebo ho ukončiť. Ak sa zadá ukončenie procesu, pre každý zadaný súbor v liste sa vytvorí jedna úloha pre príslušných pracovníkov na Rabbitmq serveri s informáciami o súbore a všetkými emailovými adresami, ktoré boli zadané. Týmto je proces ukončený a výsledky budú doručené po tom, čo jeden s pracovníkov pre validáciu dokončí úlohu úspešne. Na emaile sú inštrukcie, že v prípade negatívneho výsledku je nutné kontaktovať administrátora. Do budúcnosti by bolo vhodné zapracovať takúto aplikáciu do samotného Nextcloudu alebo separátne s grafickým prostredím pre jednoduchšie používanie užívateľmi. Takto budú musieť užívatelia kontaktovať administrátora, ku ktorým súborom je nutné vytvoriť dôkazy. Výhodou tohto riešenie je, že sa bude šetriť výpočtový výkon. Užívatelia to budú vyžadovať len v prípade, že to naozaj potrebujú.

Archivation Worker

Primárnu úlohu archivácie vykonáva „archivation wroker“. Jeho hlavnou úlohou je archivovať vybraný súbor. To, aký súbor a odkiaľ má archivovať dostane pomocou úlohy z Rabbitmq serveru. V tejto úlohe musí byť definované meno pracovníka, ktorý ju má konzumovať, cesta k súboru, ktorý sa má konzumovať a meno autora súboru z Nextcloudu. Pokiaľ úloha nebude mať presný formát, vyhodí sa známa výnimka a táto úloha sa vymaže, nakoľko je chybná a žiadny ďalší pracovník ju neskonzumuje. Samotný pracovník využíva všeobecného spotrebiteľa úloh, ktorému priradí funkciu, ktorá sa vykoná po obdržaní úlohy. Touto funkciou je archivovanie. Táto funkcia sa spustí v novom vlákne, preto je pracovník schopný spúšťať niekoľko úloh naraz. Funkciu potom vykonáva ďalší objekt, ktorému je nutné predať konfiguráciu archivácie pri vytvorení. Konfigurácia obsahuje informácie o tom, kam sa majú ukladať archivované súbory, aká dlhá ma byť počiatočná platnosť časovej pečiatky pred jej obnovou, informácie k schopnosti podpísať súbor, možnosti nastavenia vzdialeného prístupu k pôvodnému úložisku Nextcloudu a informácie ohľadom TSA. Následne sa spustí funkcia archivácie na základe predaných informácií ako cesta a meno vlastníka.

Ako prvé sa vytvoria prázdne záznamy tabuliek ohľadom archivácie, ktoré sa počas behu celé vyplnia informáciami a zapíšu sa do databázy, pokiaľ všetko prebehne ako má. Tabuľka archivačných súborov sa hneď začne vypĺňať menom vlastníka, ktorý bol predaný ako parameter a menom súboru, ktorý sa zistí z predanej cesty. Pred spustením samotnej manipulácie zo súbormi sa overí podpisovací certifikát a certifikát TSA pomocou CRL. Overovanie prebieha skontrolovaním, či CRL na-

ozaj patrí k danému certifikátu pomocou overenia podpisu a následnou kontrolou sériového čísla certifikátu. Kontorluje sa, či dané číslo nie je obsiahnuté medzi odvolanými certifikátmi v CRL.

Ďalej na základe konfigurácie predanej do tohto procesu sa zistí ako je nastavený vzdialený prístup k pôvodnému úložisku odkiaľ sa majú archivovať súbory. Je možnosť aby tento pracovník bežal na tom istom stroji ako Nextcloud server. V takom prípade musí byť v konfigurácií nastavená hodnota pre vzdialený prístup „False“. Vtedy sa vytvorí iba kópia súborov z jedného miesta disku na iné miesto, ktoré by podľa zadania mal byť aspoň separátny disk. Je možné tohto pracovníka spustiť na inom stroji ako je úložisko Nextcloudu. V takom prípade sa využíva SFTP spojenie na toto úložisko. To musí byť nastavené a musí byť pridelený prístup pre užívateľa, pomocou ktorého sa tam bude pracovník pripájať. Takisto prihlasovanie k tomuto SFTP spojeniu musí byť nastavené len pomocou autentizačných kľúčov. V takom prípade je nutné v konfigurácii vzdialeného prístupu zadať informácie ako IP adresu hosťa, na ktorý sa bude pripájať, ssh port, ktorý sa využíva k spojeniu a prístupové údaje ako meno užívateľa pre prihlásenie, cestu ku kľúču používaného k prihláseniu a heslo k nemu. Takéto prihlasovanie je bezpečnejšie ako využívanie prihlasovania iba pomocou hesla užívateľa. Je vhodné takéto prihlasovanie úplne zakázať.

Následný proces je rovnaký pre oba systémy. Pred prenesením a uložením dát sa vytvorí kryptograficky haš pomocou algoritmu SHA-2 o dĺžke 512 bitov. Táto hašovacia funkcia považovaná za bezpečnú organizáciou Nist, ktorá odporúča minimálnu dĺžku 256 bitov [36]. Tento haš sa uloží a po prenesení súboru na cieľové úložisko sa z preneseného súboru vytvorí znova kryptografický haš. Vďaka tomu máme kontrolný súčet a dôkaz o tom, že prenos prebehol správne a súbory sú identické.

Keďže haš archivovaného súboru už máme, môžem pokročiť na získanie prvej časovej pečiatky od TSA. K tomu slúži v konfigurácii sekcia, ktorá predáva informácie o použitej TSA. Týmito informáciami je URL adresa vzdialenej TSA, cesta k certifikátu tejto autority, URL adresa pre stiahnutie aktuálneho CRL a certifikát certifikačnej autority, ktorú TSA používa. V tomto prípade pre vývoj a testovanie poslúžila autorita na webovej stránke „freeTSA.org“, ktorá poskytuje zadarmo službu získania časových pečiatok. Táto služba sa drží štandardu protokolu pre časové pečiatky RFC3161 [37]. Nakoľko sa k implementácii získania týchto pečiatok používa knižnica tretej strany, je možné vybrať jednu zo siedmich rôznych autorít poskytujúcich rovnakú službu [38]. V prípade budúceho vývoja a záujmu o implementáciu vlastnej TSA v rámci systému by bolo pravdepodobne nutné upraviť danú knižnicu. V prípade správnej konfigurácie TSA sa získa časová pečiatka pre haš pôvodného súboru. Nebolo by vhodné posielať citlivé a pravdepodobne veľké súbory zbytočne po sieti tretej strane. Z hašu tohto súboru sa pri tvorbe požiadavky pre TSA znova vypočíta ďalší haš typu SHA 2 o dĺžke 512 bitov. Po získaní sa táto

časová pečiatka uloží k archivovanému súboru pod názvom „timestamp0“.

Je nutné tieto 2 súbory spojiť, aby boli spolu ako jeden celok. K tomu sa využíva vkladanie do archívov typu tar. Napriek tomu, že tar archívy podporujú kompresiu, tá je vypnutá, nakoľko ňou nechceme ohroziť integritu súborov. Oba súbory sa postupne vkladajú do archívu s názvom „Package0“ a po ich vložení sú odstránené, aby sa vyhlo zbytočným duplicitám.

Aby sa splnili ciele dizajnu treba podpísať pôvodný súbor. Je možné, že súbory môžu mať desiatky gigabytov a nebolo by vhodné podpisovať celý súbor o takej veľkosti. Vytvorí sa kryptografický haš pre archív „Package0“, ktorý je omnoho rýchlejší. Následne sa podpíše haš a celý podpis sa uloží pod názvom „signature.sig“ do zložky s archívom „Package0“. Tento podpis sa ukladá v kódovaní typu „base64“. Ku podpisovaniu je nutné v predanej konfigurácii nastaviť cestu k certifikátu k podpisovaciemu kľúču, cestu k súkromnému kľúču pre podpis, heslo k súkromnému kľúču a cestu k poskytnutému CRL súboru od certifikačnej autority. Nakoľko implementácia získania CRL od rôznych certifikačných autorít by bola komplikovaná a nebolo by ju možné v podmienkach tejto práce otestovať, táto práca sa spolieha na manuálne získanie a nastavenie cesty k CRL. K podpísaniu sa načíta kľuč z určenej cesty a k autentizácií sa použije heslo z konfigurácii. Po vytvorení podpisu sa z neho rovnakým algoritmom vypočíta haš a vypýta sa druhá časová pečiatka. Týmto sme previazali čas uloženia, podpisu a platnosť podpisovacieho kľúča do jedného časového okna. Táto pečiatka sa uloží pod názvom „timestamp1“ k zvyšným súborom. Následne pre budúce overenie je možné k týmto súborom pridať použité certifikáty v procese, aby sa všetko dalo overiť. Tie sa nakopírujú do zložky s názvom „certificates“. CRL TSA certifikátu sa najskôr stiahne z URL adresy v konfiguračných súbore a tak sa zapíše medzi ostatné certifikáty.

Finálne sa vytvorí tar rovnakým spôsobom ako pri archíve „Package0“ súbore, ktorý obaľuje všetky doteraz vytvorené súbory a to: archív „Package0“, zložku s certifikátmi, podpis a časovú pečiatku. Uloží sa pod názvom „Package1“ a vytvorí sa z neho haš. Tento haš je uložený do databázových záznamov, ktoré sa zapíšu do databázy posledným krokom archivácie. Informácie v týchto záznamoch majú identifikačný, lokalizačný a overovací účel pri budúcej práci so súbormi.

Ak bol celý proces úspešný, vráti sa stav „OK“, ktorý zabezpečí, že pracovník potvrdí vykonanie úlohy a tá zmizne. Následne pracovník čaká na ďalšie úlohy, ktoré môže skonzumovať z Rabbitmq servera. K spusteniu tohto pracovníka sa používa skript „run_archivation_worker.py“, ktorý potrebuje konfiguračný súbor s pripojením k Rabbitmq serveru, databázy a vlastnú konfiguračnú časť, ktorá bola popísaná vyššie. Príklad tohto súboru existuje v priloženej práci. Po spustení nastavia logovania na Rabbitmq servery sa spustí pracovník. Balík po archivovaní súboru zväčší celkovú veľkosť archivovaných dát o 45,6KB.

Retimestamping Worker

Úlohou tohto pracovníka je ochraňovať a predlžovať silu dôkazov existujúcich archivovaných súborov získavaním nových časových pečiatok pre vytvorené balíky obsahujúce archivované súbory a dôkazy o ich integrite. To vykonáva na základe úloh z príslušneho poradovníku na Rabbitmq serveri. Tie sa tvoria ako bolo popísane v kapitole o kontrole platnosti časových pečiatok. Tento pracovník musí byť spustený na rovnakom stroji ako sú uložené archivované dáta.

Všetci pracovníci fungujú na rovnakom princípe ako bolo popísané vyššie, čo sa týka konzumácie úloh. Jedinou zmenou je, že sa pri konzumácií definuje iná úloha. V tomto prípade je ňou obnova časovej pečiatky. Použité hašovacie funkcie a funkcie pre získanie časovej pečiatky sú taktiež všade rovnaké. Kľúčom k zabezpečeniu efektivity a zmysluplnosti tohto systému je používať takú TSA, ktorá používa bezpečné a aktuálne kryptografické protokoly k vytvoreniu časovej pečiatky. K úspešnému spusteniu tohto pracovníka je nutné mu predať konfiguračný súbor, ktorý obsahuje pripojenie k Rabbitmq serveru, databáze a vlastnej konfigurácii. Jeho vlastná konfigurácia obsahuje iba informáciu o tom, ako dlho má byť ďalšia časová pečiatka platná a informácie k použitiu vzdialenej časovej autority. Každý pracovník má svoj štartovací skript.

Úloha obsahuje ID archivovaného súboru z databázy, ktorému bude získaná nová časová pečiatka. Pred jej získaním je nutné overiť poslednú pečiatku. Po spustení tohto procesu sa ako prvé získa príslušný databázový záznam archivovaného súboru a posledný záznam z tabuľky „FilePackages“. Pomocou jednotlivých informácií zo záznamov sa pristúpi k súboru a začne sa overovať. Overuje sa haš posledného balíku, ktorý je obsiahnutý vo vnútri. Haš vnútorného balíku sa získava v rámci archívu bez jeho extrahovania. Ak jeden z nich nesedí, vyhodí sa známa výnimka, ktorá sa zaloguje, celý proces sa ukončí a úloha sa potvrdí. Túto výnimku by bolo vhodné monitorovať v logovacích systémoch, nakoľko vraví o tom, že pravdepodobne došlo k narušeniu integrity archivovaných súborov. V prípade, že oba haše sedia, overí sa posledná získaná časová pečiatka vo vnútri balíku. Overovanie vykonáva vzdialená TSA, ktorá je špecifikovaná v konfiguračnom súbore. Rovnako ako pri overovaní hašu, v prípade, že časovú pečiatku sa nepodarí overiť je nutné monitorovať logy, kde sa objaví táto chyba. Ak sa úspešne overí časová pečiatka, prejde sa do ďalšej úlohy a tou je získanie novej časovej pečiatky a tým predĺženie sily dôkazov.

Haš posledného baličku bol vytvorený, tak sa preň vypýta nová časová pečiatka od vzdialenej TSA, ktorá je definovaná v konfiguračnom súbore. Po jej obdržaní sa zapíše k ostatným súborom. Rovnako sa vytvorí zložka s certifikátmi. Tentokrát sa tam ukladajú len certifikáty spojené s TSA a rovnako sa sťahujú CRL z URL adresy poskytnutéj v konfiguračnom súbore. Po obdržaní všetkých dát je možné

obaliť tieto dáta do nového balíku. Tak sa vytvorí balík typu tar s názvom „PackageF{PackageID}“, kde sa namiesto „{PackageID}“ dá ID balíkú, ktorý sa práve obaľuje. Tým je zaručená unikátnosť pre každé meno balíku. Názov neobsahuje poradové číslo balíku, nakoľko ich vo vnútri môže byť neznámy počet a nebolo by efektívne ho celý rozbaľovať alebo ťahať všetky záznamy o danom balíku z databázy. Po vytvorení balíku sa takisto vytvorí jeho haš, ktorý sa zapíše do záznamu. V tomto pracovníkovi sa počas behu obnovy časovej pečiatky vytvorí nový záznam pre tabuľku „FilePackages“, ktorý sa naplní získanými dátami. Počas obnovy časovej pečiatky sa aktualizuje aj pole „ExpirationDateTS“ v príslušnom zázname z tabuľky „ArchivatedFiles“. Nakoniec, keď všetko prebehne úspešne, zmeny sa zapíšu do databázy. Pracovník ako vždy pri úspešnom dokončení potvrdí úlohu a začne konzumovať ďalšiu - ak existuje. Ak ďalšia úloha neexistuje, pracovník čaká.

Priestorová náročnosť takéhoto obaľovania nie je veľká, nakoľko sa obaľovanie bude vykonávať raz za niekoľko rokov. Jedno takéto obalenie časovej pečiatky zväčší súbor o 20,5 KB. Ak by sa súbor obaľovať každé 2 roky po dobu dvadsiatich rokov, jeho kapacitu by to zväčšilo iba o 205 KB.

Validation Worker

Poslednou potrebnou funkcionalitou bolo spätné overenie platnosti dôkazov vytvorených počas doby úschovy súborov. K tomu slúži pracovník popísaný v tejto podkapitole. Funguje na rovnakom princípe ako predošlé dva popísané vyššie, akurát vykonáva funkciu overovania. Tú vykonáva postupným rozbaľovaním vrstiev v archivovanom balíku. Po úspešnom overení kontroluje existenciu archivovaného súboru na pôvodnej ceste, kde bol uložený. Následne sa odošle email s informáciami o výsledku validácie na emailové adresy, ktoré boli zadané v úlohe. Tento pracovník je nutné spustiť na stroji, kde sú archivované dáta, nakoľko k nim potrebuje mať prístup.

Konfiguračný súbor potrebný pre validačný pracovník obsahuje informácie pripojenia k Rabbitmq serveru, pripojenie k databáze a vlastnú konfiguráciu. Vlastná konfigurácia obsahuje informácie o TSA, ktorá bude použitá pre overovanie, informácie k vzdialenému prístupu, ktorý je použitý len k overeniu existencií a integrity súboru z pôvodnej cesty a informácie k rozosielaniu emailov o výsledku validácie.

Úloha obsahuje informácie o súbore, ktorý je nutné validovať. Tie sú formou mena súboru a mena užívateľa, ktorých kombinácia je vždy unikátna, alebo formou ID databázového záznamu. Ďalej obsahuje zoznam emailových adries, kde bude rozposlaný výsledok validácie. Po spustení validácie sa získa príslušný záznam z tabuľky „ArchivatedFiles“ a k nemu všetky existujúce záznamy „FilePackages“. Tie prídu zoradené podľa dátumu od najnovšieho po najstarší.

Je potrebné postupne skontrolovať každý jeden balík vo vnútri posledného až

kým neprídeme ku koreňovému balíku s názvom „Package1“. K tomu musíme rozbaliť tar archívy. Dočasne sa vytvorí zložka, do ktorej sa začnú postupne rozbaľovať a validovať dôkazy v nich. Táto zložka sa po dokončení úlohy alebo v prípade poruchy celá vymaže a tým sa nevytvárajú žiadne zabudnuté duplikáty, ktoré by zaberali pamäť. Pri voľbe veľkosti pamäte a dizajnovaní fyzickej architektúry, na ktorých budú aplikácie spustené, je nutné brať do úvahy, že pre validáciu musí byť aspoň toľko voľného miesta na disku, koľko je veľkosť archivovaného balíku, ktorý sa bude validovať.

Po začatí programu sa opakovane rozbaľujú balíky. Balíky, ktoré obaľujú pôvodný majú všetky názov „PackageF{PackageID}“. Na základe toho sa tvorí podmienka, aká validácia sa má vykonať. Jeden typ je určený pre balíky obaľujúce pôvodný, ktorý obsahuje len časovú pečiatku a informácie k nej. Druhý typ je koreňový balík s názvom „Package1“, ktorý obsahuje podpis, časovú pečiatku a ešte aj úplne pôvodný balík. Balíky sa extrahujú a validujú prvým spôsobom až kým balík vo vnútri extrahovaného súboru nemá názov „Package1“, vtedy sa pustí validácia koreňového balíka.

Validácia obaľovacieho balíka sa vykonáva rovnako ako keď sa overuje časová pečiatka v prípade jej obnovy. Tento postup je popísaný v predchádzajúcej kapitole. V tomto prípade je tu pridaná verifikácia certifikátu, ktorý bol použitý pri vydaní spolu s priloženým CRL, ktorý je v zložke „certificates“. Overovanie všetkých certifikátov prebieha rovnako ako bolo popísané pri archivácií. Po úspešnom overení sa rozbalí balík, ktorý je obsiahnutý vo vnútri tohto a zistí sa jeho názov. Teda ak je názov „pacakge“, celý proces opakuje. Ak je názov „Package1“, spustí sa kontrola koreňového pôvodného balíka.

Kontrola koreňového balíka začne tým, že sa vypočíta haš pre „Package1“, ktorý sa porovná s tým, ktorý je uložený v databázovom zázname a ak sú zhodné, balík sa rozbalí. Nasleduje overenie hašu pôvodného balíka „Package0“. Po zhode je možné overiť digitálny podpis tohto hašu s použitím uloženého certifikátu. Ďalej je nutné overiť časovú pečiatku „timestamp1“, ktorá sa overuje rovnakým spôsobom ako pri ostatných pečiatkach. Potom sa rozbalí posledný balík „Package0“, kde sa vypočíta kryptografický haš a znova sa overí s databázou a s ním sa overí aj priložená časová pečiatka „timestamp0“. Následne sa overia certifikáty, použité TSA a certifikát k podpisovaciemu kľúču s priloženými CRL. Certifikáty TSA sú rovnaké pre „timestamp0“ a „timestamp1“, nakoľko boli vykonané pomocou rovnakej TSA a sú uložené spolu v balíku „Package1“. Posledný krok k úspešnému overeniu je validácia pomocou porovnania kryptografických hašov archivovaného a originálneho súboru, pokiaľ existuje stále v pôvodnej lokácií. K prístupu k pôvodnej lokácií slúžia informácie z konfiguračného súboru ohľadom vzdialeného pripojenia. Funguje to rovnakým spôsobom ako pri „Archivation Worker“, ktorého spôsob fungovania bol

popísaný v predošlých kapitolách. V prípade, že tento originálny súbor je zmenený, neexistuje alebo nie je prístupný, vyhodí sa špeciálna výnimka. Tá sa odchytáva osobitne, aby bolo možné užívateľovi oznámiť, že pôvodný súbor bol upravený, ale integrita archivovaného súboru bola nedotknutá a je možné získať pôvodnú verziu. Všetky výnimky a chyby spojené s validáciou okrem tejto spomenutej sa taktiež odchytávajú v rámci procesu validácie. V prípade, že sa nejaká objaví, je možné, že integrita archivovaných súborov bola narušená. Ak všetko prejde ako má, tento stav sa odrazí v emaile o výsledku. Jeho obsahom môžu byť 3 druhy správ:

- integrita overovaného súboru bola zachovaná,
- integrita overovaného súboru bola zachovaná v archíve, ale originálny súbor bol upravený,
- integrita overovaného súboru v archíve bola poškodená.

Email s informáciou sa následne sformátuje a odošle na všetky emailové adresy z úlohy. V prípade negatívneho výsledku sa pridajú do emailu informácie o kontakte na správcu s povinnosťou ho kontaktovať. K správnemu sformátovaniu a odoslaniu emailovej správy je nutné predať vhodnú konfiguráciu v súbore. Táto konfigurácia musí obsahovať informácie ako emailový server, ktorý bude použitý k odoslaniu správ, email použitý k odoslaniu, heslo k tomuto emailu a kontaktné údaje správcu - emailovú adresu a telefónne číslo. Dôvod, prečo by mali byť emaily rôzne je, že email používaný na odosielanie správ pomocou skriptu je slabšie zabezpečený, nakoľko musí mať povolené používanie menej zabezpečenými aplikáciami. Emailový server použitý pre testovanie bol napríklad „smtp.google.com“. Po odoslaní emailu o výsledku sa proces validácie ukončí a úloha potvrdí.

Chytanie výnimiek a logovanie

Celý beh pracovníkov sa zaznamenáva pomocou logov. Tie sú rozdelené podľa účelu na debug, info, warning a exception. Počas pustenia aplikácií do produkcie by mali byť pustené len na level info, nakoľko debug poskytuje citlivejšie informácie a slúži len pre vývojarov. Všetky logy počas behu pracovníkov sa posielajú do špeciálnej exchange „log“, na ktorú je naviazaný poradovník s názvom „logs“ na Rabbitmq serveri. Sú im ponechané originálne názvy. Logy neobsahujú citlivé informácie a sú posielané bez špeciálneho zabezpečenia. Logy je možné konzumovať ako bežné úlohy s Rabbitmq serveru pomocou osobitnej aplikácie. V tomto prípade je spravený len základný skript s názvom „consuming logs“, ktorý tieto logy vypisuje na konzolu. Tento prístup posielania všetkých logov do jedného kanálu umožňuje centrálne monitorovanie systému. Pri nasadení tohto riešenia by bolo nutné zabezpečiť monitorovanie logov, pretože na základe ich levelu vieme zisťovať problémy v aplikácií alebo zo vstupom od užívateľov a na základe toho upozorniť správcov ako jednať.

Pracovníci počas behu môžu vyhodiť chyby, ktoré sa následne odchytávajú. Pokiaľ sa jedná o chybu na základe vstupných dát alebo niečo, čo nie je kritické pre beh aplikácie, tieto chyby sa zalogujú pod levelom „warning“. Chyby je nutné prešetrovať, nakoľko stále hovoria, že proces neprebehol správne, ale nemajú neodkladnú prioritu. Po takejto známej chybe sa úloha potvrdí na Rabbitmq serveri a už ju nebude môcť žiadny ďalší pracovník konzumovať. Príklad takejto chyby je, že sa nepodarilo zapísať záznam do databázy pravdepodobne na základe toho, že tam už taký záznam existuje. V prípade, že sa v systéme objaví závažná chyba ako napríklad strata spojenia na databázu alebo akákoľvek neznáma chyba, tak sa zaloguje pomocou levelu „exception“. Chyby pod týmto levelom je nutné okamžite prešetrovať, nakoľko značia chybu priamo s aplikáciou alebo systémom, na ktorom beží. V takomto prípade sa úloha taktiež potvrdí, ale jej obsah sa prepošle do poradovníka pre chybné úlohy. Vďaka tomuto sa žiadna úloha nestratí. Administrátor môže pozrieť aká úloha bola chybná a na základe toho sa pokúsiť zreplikovať vadu. Po oprave chýb môže tieto úlohy preposlať do správnych poradovníkov naspäť, kde sa opäť vykonajú. Týmto spôsobom je systém pripravený k vývoju alebo nasadeniu komplexnejšieho monitorovacieho systému. Monitorovací systém bude nutné vyriešiť, nakoľko systém archivácie závisí od mnohých premenných, ktoré sa môžu pokaziť. Napríklad prístup k súborovému systému alebo k databáze, dostatok miesta na disku a podobne.

4.3.3 Testovanie a výsledky implementácie

Testovanie systému prebiehalo po častiach. To znamená, že výkon každej časti systému sa testoval osobitne. Testy Nextcloudu a ich výsledky boli popísane ešte v predošlej kapitole. Ako bolo popísané, systém sa skladá z niekoľkých častí. Hlavnou časťou sú pracovníci. K tomu aby mohli fungovať je nutné im predať úlohu. Naraz sa vytvorí niekoľko úloh. Dôvodom je, že aplikácia „Workflow External Scripts“ spustí tieto skripty podľa toho ako je nastavený cron. Ten je možné nastaviť v rôznych časových intervaloch. Pre príklad, niekoľko užívateľov označí súbor štítkom „archive“ v priebehu 2 minút 10 krát. Ak by bol cron nastavený na pravidelné spúšťanie každé 2 minúty, stalo by sa to, že náš skript by sa spustil 10 krát a vytvoril by 10 úloh. Tie by následne archivačný pracovník začal konzumovať. Preto sa vykonalo meranie, kedy sme zistili, že pomocou skriptu na tvorbu úloh sa vytvorí 100 úloh za približne 4 sekundy na vzdialenom Rabbitmq servery. Prebehlo meranie rýchlosti vykonania úloh v závislosti na ich počte a veľkosti súboru. Rovnako prebehlo aj jedno dodatočné meranie, kedy sa meralo ovplyvnenie rýchlosti vykonávania úloh pri spustení niekoľkých pracovníkov. Merania boli vykonávané s tým, že sa súbory archivovali alebo overovali aj zo vzdialeného úložiska, aby sa dal nasimulovať vplyv prenesenia a zapísania súborov na disk. Túto rýchlosť ovplyvňuje najmä rýchlosť spojenia a

rýchlosť zapisovania dát na disk. Bolo vykonané aj meranie času archivácie rôznych veľkostí súborov za použitia jedného disku. Vďaka tomuto meraniu bolo možné vytvoriť obraz o tom aký veľký vplyv na systém ma rýchlosť spojenia a zápis na nový disk. Tabuľky s nameranými hodnotami budú priložené v prílohe a ich výsledky popísané v nasledujúcich sekciách.

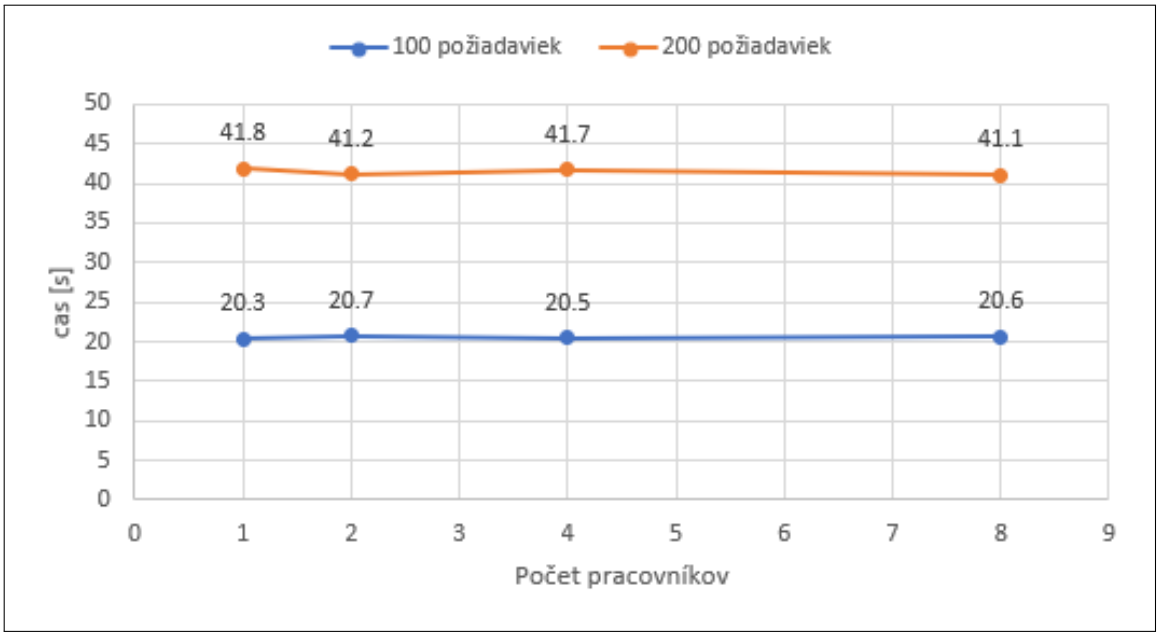
Parametre testovacieho prostredia

Archivačný systém bol testovaný pomocou 2 virituálnych strojov, oba systémy boli spustené vo virtualizačnom prostredí Oracle VM VirtualBox. Obom strojom boli priradené 4 GB operačnej pamäte, 15 GB priestoru na HDD.

Na jednom stroji bol spustený Rabbitmq server a zároveň slúžil ako vzdialené úložisko odkiaľ sa archivovali súbory. Na druhom stroji bola spustená MySQL databáza a jednotlivý pracovníci.

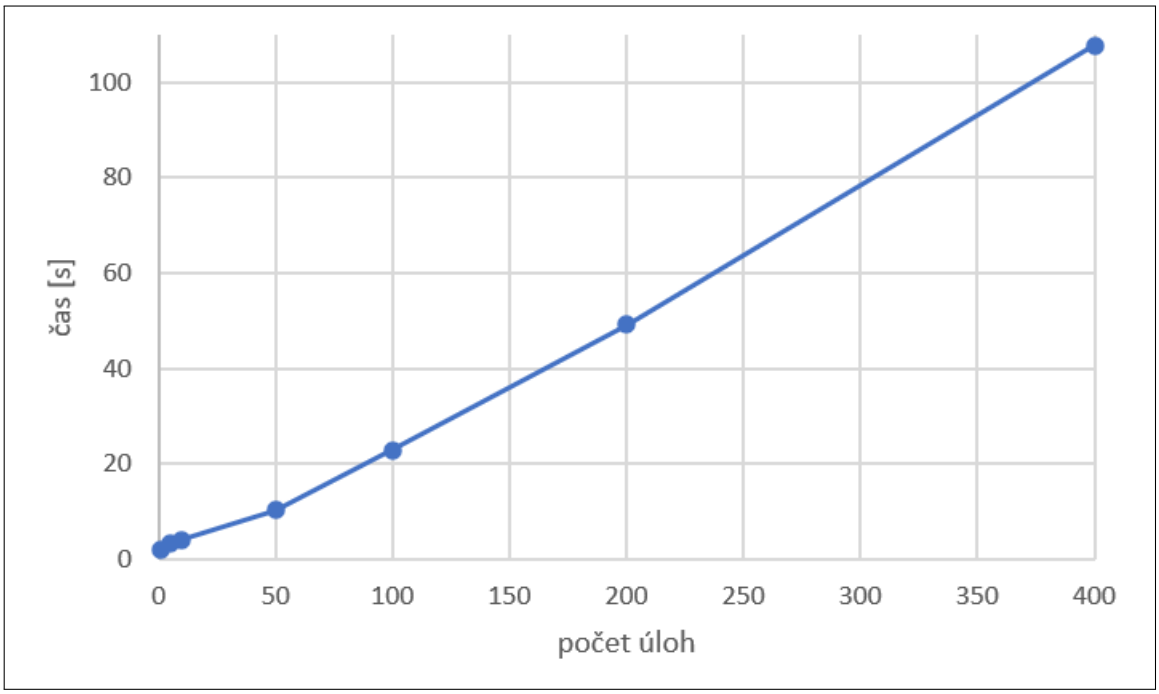
Meranie rýchlosti archivovania

Meranie vplyvu počtu pracovníkov na rýchlosť spracovania požiadaviek preukázal, že nemá žiadny vplyv. Nakoľko sú úlohy jednotlivými pracovníkmi spúšťané vo vláknach, tak v prípade, že pracovníci používajú jeden procesor, ich počet nijako neovplyvní rýchlosť. Meranie prebiehalo na súbore o veľkosti 6 bytov uloženom na lokálnom disku. Tento výsledok je možné vidieť s nasledujúceho grafu 4.5.



Obr. 4.5: Vplyv počtu pracovníkov na rýchlosť odbavovania požiadaviek

V prípade merania vplyvu počtu úloh archivácie na dobu spracovania rastie čas adekvátne k ich počtu, no nárast nie je kritický. Meranie bolo vykonávané na súbore o veľkosti presne 1 KB. Toto meranie je znázornené v grafe 4.6.

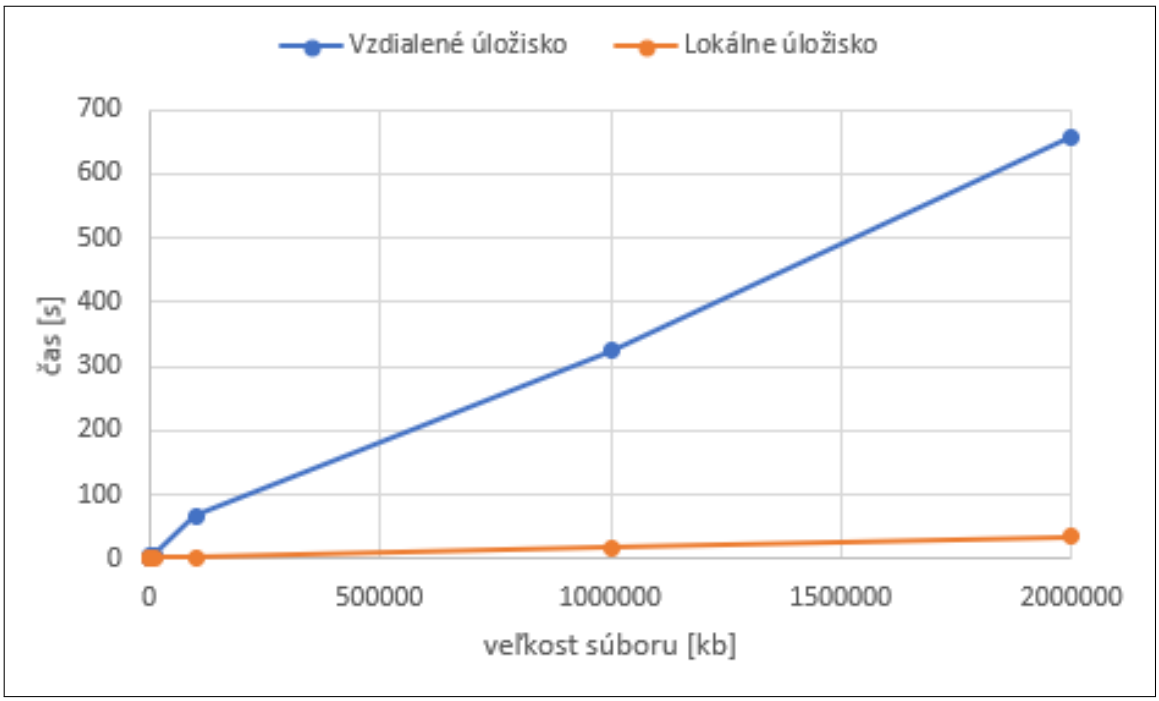


Obr. 4.6: Vplyv počtu úloh na rýchlosť odbavovania požiadaviek

Vplyv veľkosti súborov je pre rýchlosť systému omnoho kritickejší. V grafe 4.7 sú zaznačené namerané hodnoty. Nakoľko je tam veľmi veľký skok v hodnotách, nie je dobre vidieť všetky body. Čiara popisujúca rýchlosť archivácie súborov z lokálneho úložiska je na časovej osy omnoho nižšie ako čiara popisujúca archiváciu zo vzdialeného úložiska. Za to môže primárne to, že tieto dáta sa museli preniesť po sieti a zapísať na pevný disk. Pri archivácii v rámci jedného úložiska sú tieto dáta už zapísané na disku a nezapisujú sa znova. Tento prístup by v praktickom nasadení nemal zmysel. Pri testovaní je na ňom vidieť aký časový vplyv má približne na celom procese archivačný pracovník.

Meranie rýchlosti obnovy časových pečiatok

Meranie obnovy časových pečiatok prebehlo vytvorením rôzneho počtu úloh, pričom každá úloha musela obnoviť časovú pečiatku pre rôzny súbor. Veľkosť súborov bola 1 KB. Tento proces je o niečo rýchlejší ako proces archivácie, nakoľko sa tu nevykonávajú zložitejšie operácie so súbormi.



Obr. 4.7: Vplyv veľkosti súborov na rýchlosť odbavovania požiadaviek

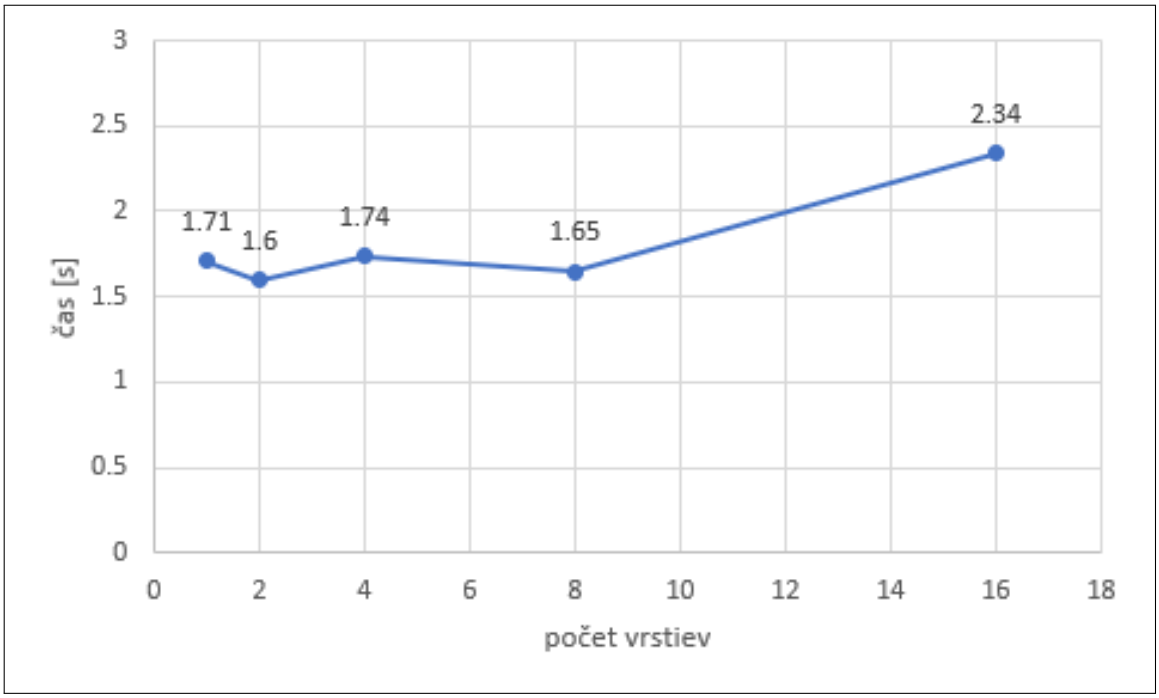
Vplyv veľkosti súborov na obnovu pečiatok taktiež nie je taký významný ako pri archivácií. Hlavnou príčinou je, že tu neprebieha zapisovanie na disk. Každopádne sa počítajú haše týchto súborov, takže nejaký vplyv tam je.

Meranie rýchlosti validácie

Úlohy sa vykonávali podobnou rýchlosťou ako u zvyšných pracovníkov. Graf vyzerá veľmi podobne ako u archivačného pracovníka. Hodnoty pre toto je možné vidieť v tabuľke v prílohe.

Pri validácii bolo nutné zmerať vplyv počtu vrstiev vyprodukovaných pri obnovovaní platnosti časových pečiatok. Maximálna hodnota bola zvolená 16, nakoľko obnovovanie by malo prebiehať v niekoľkoročných intervaloch. Vplyv na výkonnosť je zanedbateľný, nakoľko rozdiel medzi jednou vrstvou a šestnástimi je len 0,6 sekundy. Tieto výsledky je možné vidieť v grafe 4.8.

Veľkosť overovaných dát veľmi ovplyvňuje rýchlosť validácie. Graf je podobný ako u archivačného pracovníka, hodnoty sú v tabuľke v prílohe. Časová náročnosť je takmer taká náročná ako pri archivácii. To je z dôvodu, že posledný krok validácie je overenie originálneho súboru. Overenie urobíme vypočítaním hašu. K tomu aby sme ho vypočítali sa po častiach prenáša k pracovníkovi. To je možné vylepšiť tým, že odstránime tento krok overovania. Druhá a lepšia možnosť je, že sa v budúcnosti



Obr. 4.8: Vplyv obaľovania na rýchlosť odbavovania požiadaviek

implementuje spôsob, kedy by sa dáta neprenášali, ale prenášal by sa len vypočítaný haš.

Výsledky

Z testovania jednotlivých pracovníkov sme dospeli k záveru, že systém je plne funkčný a jeho primárne obmedzenia sú rýchlosť pripojenia k vzdialenému úložisku a rýchlosť zápisu na disk. Do budúcnosti by bolo vhodné optimalizovať posledný krok validácie, čím by sa masívne zrýchlila jeho činnosť. Hodnoty jednotlivých meraní ku grafom je možné nájsť v prílohách. Vyskytnuté chyby počas testovania a meraní sa opravili. Jedinou potencionálnou chybou, ktorá nebola odstránená je, že pri veľkom množstve úloh sa môžu vyskytovať chyby. To nastáva z dôvodu, že pracovníci nijako neobmedzujú množstvo vlákien, v ktorých sú úlohy paralelne spustené. Chyba sa vyskytovala len v prípade, keď bolo zadefinovaných viac ako 50 rovnakých úloh naraz a keď sa pri vykonávaní úloh pristupovalo naraz k rovnakému súboru z mnohých vlákien. Tento problém sa primárne prejavoval pri SFTP spojení. Pri správnom a definovanom používaní systému sa táto chyba vôbec neprejavuje, jedná sa o špeciálny testovací scenár. To je z dôvodu, že každý súbor by mal byť archivovaný len raz a žiadna z operácií by sa pri správnom používaní systému nemala opakovať viac ako raz v krátkej dobe pre jeden a ten istý súbor.

Záver

Táto práca obsahuje základnú teóriu problematiky cloudových úložísk, riadenia prístupu, zabezpečenia a archivácie dát. Ďalej sa tu rozobrala problematika voľne dostupného softvéru a proprietárnych riešení pre realizáciu zabezpečeného cloudového úložiska. Boli tu popísané možné výhody, ale aj nevýhody, obavy či riziká proprietárnych a voľne dostupných riešení pre cloudové úložisko, do ktorého by užívatelia chceli nahrávať vysoko citlivé dáta a zachovať ich dôvernosť a integritu. V tomto bode sa dospelo k záveru, že k riešeniu takéhoto úložiska sú výhodnejšie voľne dostupné produkty pod licenciami typu open source alebo free software. Najväčším argumentom pre toto rozhodnutie bolo to, že softvéry pod takýmito licenciami majú verejne dostupný zdrojový kód aplikácie. Následne sa spravil výber produktov, ktoré mohli umožniť nejakú s funkcionalít k dosiahnutiu zabezpečeného úložiska s možnosťami aplikovania riadeného prístupu, archivácie dát a najmä zachovanie dôvernosti a integrity dát. V prvej stručnejšej analýze sa vyselektovali dva najvhodnejšie produkty, ktoré by mohli priniesť túto funkcionalitu. Boli to ownCloud a Nextcloud. V dôkladnejšej analýze sa predstavili niektoré z výhod Nextcloudu, vďaka ktorým bol vybraný k realizácii testovacej inštancii na lokálnom virtuálnom stroji. Podarilo sa zrealizovať základnú inštaláciu a nastavenie požadovaných funkcionalít ako možnosť definovať právomoci pre riadenie prístupu, zachovať dôvernosť a integritu dát vďaka šifrovaniu na strane klienta. Takisto použitie bezpečného prenosu na server pomocou podpory HTTPS spojenia. Ďalej boli popísané možnosti nastavenia funkcionalít, ktoré boli hlavnými cieľmi tejto práce.

Podarilo sa nakonfigurovať zabezpečené úložisko poskytujúce široké možnosti nastavenia prístupov, skupinových oprávnení, nastavenia šifrovaného spojenia, šifrovanie súborov samotné, ale jedna funkcionalita chýbala - konkrétne možnosť zaistenia dlhodobej archivácií dát. Teda možnosť, kedy sa po nahraní dát na server vytvorí ich kópia a je plne zabezpečená ochrana integrity tejto kópie počas dlhej doby uloženia. K riešeniu dodania tejto funkcionality boli predložené možné riešenia. Bola vybraná kombinácia riešení. V poslednej časti práce sa vytvoril návrh riešenia archivácie súborov, ktorý bol následne implementovaný. Návrh kombinoval úpravu existujúcej aplikácii v Nextcloude spolu s dedikovaným archivačným systémom.

Úprava bola vykonaná na aplikácii s pôvodným názvom „Workflow Extrnal Scripts“, pričom táto úprava musela byť ďalej zverejnená pod licenciou GNU AGPLv3. V tejto úprave boli pridané len možnosti predania ďalších informácií do skriptov, ktoré spúšťala. Táto aplikácia bežiaca na Nextcloud serveri má slúžiť ako prepojenie medzi Nextcloud serverom a Archivačným systémom. Primárne hovorí archivačnému systému, ktoré súbory je nutné archivovať.

Ďalším prvkom tohto riešenia bol Archivačný systém. Ten sa skladal z niekoľ-

kých komponentov. Jeho hlavnou úlohou bolo vytvorenie kópie originálneho súboru a zabezpečenia dôkazov o jeho integrite. Takisto muselo byť možné manuálne spustiť proces overenia archivovaného súboru a výsledok tohto overenia musel byť odoslaný užívateľom na ich emailové adresy. V pôvodnom návrhu sa uvažovalo, že by sa archivácia diala na rovnakom stroji, na ktorom je pustený Nextcloud a presun by bol len do dedikovaného úložiska. Okrem tejto možnosti sa podarilo implementovať aj možnosť, kedy je program schopný dostať sa k súborom aj vzdialene, bezpečným spôsobom. Takisto je možné overiť integritu archivovaných súborov aj po dlhej dobe. Všetky časti archivačného systému sa podarilo implementovať do funkčného stavu. Rovnako pri návrhu tohto systému sa myslelo na možnosť škálovateľnosti riešenia a zvýšenia procesného výkonu pri archivácii jednotlivých súborov. Z prevádzkového hľadiska archivačného systému je nutné realizovať monitorovací systém udalostí počas behu systému. Takýto systém je možné vyriešiť zakomponovaním systémov tretích strán, ale tomuto riešeniu sa táto práca už nevenuje. Jedna z vecí, ktorá by potrebovala vylepšenie je načítavanie konfiguračných súborov pomocou formátu yaml do jednotlivých komponentov systému, nakoľko takýto prístup nie je bezpečný, pretože prístup k ich obsahu je jednoduchý.

Návrh pre vylepšenie do budúcnosti je realizácia aplikácii do Nextcloudu, ktorá by umožnila užívateľom získať stav o validácií bez kontaktu správcu, ktorý musí za nich túto funkciu spustiť. Taktiež konfigurácia celého systému je pomerne náročná, nakoľko sa skladá z niekoľkých častí. Prináša to ale výhodu väčších možností použitia jednotlivých systémov. Cieľom tejto práce bolo aj dotestovanie a podrobné zdokumentovanie systému záťažovými testami v praktickom nasadení. Boli vykonané základné záťažové testy na Nextcloud serveri v testovacom prostredí. Archivačný systém bol širšie testovaný a bola meraná jeho výkonnosť. Tieto testy sa taktiež odohrávali len vo virtualizovanom testovacom prostredí. Z týchto meraní sme zistili, že archivačný systém by mal zvládnuť zvýšenú záťaž. Takisto bol vykonaný test systému kombinovane, kedy Nextcloud server pomocou upravenej aplikácie spustil archiváciu na archivačnom systéme. Kombináciou týchto systémov boli splnené všetky vytýčené ciele tejto práce, okrem testu systému v reálnom nasadení.

Literatúra

[1] ROUSE, Margaret: *Cloud storage* [online]. [cit. 2020-10-09]. Dostupné z: <https://searchstorage.techtarget.com/definition/cloud-storage>

[2] CZERNIAK,M. a CLIMER,S.: *What Is Cloud Storage Part 2: Types Of Cloud Storage: Private Cloud* [online]. [cit. 2020-10-10]. Dostupné z: <https://gomindsight.com/insights/blog/types-of-cloud-storage>

[3] ROUSE, Margaret: *Data archiving* [online]. [cit. 2020-10-12]. Dostupné z: <https://searchdatabackup.techtarget.com/definition/data-archiving>

[4] MARTIN, James A.: *What is access control? A key component of data security* [online]. 2019 [cit. 2020-10-26]. Dostupné z: <https://www.csoonline.com/article/3251714/what-is-access-control-a-key-component-of-data-security.html>

[5] *What is access control? | Authorization vs authentication* [online]. [cit. 2020-10-26]. Dostupné z: <https://www.cloudflare.com/learning/access-management/what-is-access-control/>

[6] FAOUZI, Jaidi: *Advanced Access Control to Information Systems: Requirements, Compliance and Future Directives, Advances in Security in Computing and Communications* [online]. 2017 [cit. 2020-10-26]. Dostupné z: <https://www.intechopen.com/books/access-control>

[7] ROUSE, Margaret: *Access control: Types of access control* [online]. [cit. 2020-11-05]. Dostupné z: <https://searchsecurity.techtarget.com/definition/access-control>

[8] DOROTHEA KER, Anna: *United States of Surveillance* [online]. 2020 [cit. 2020-12-03]. Dostupné z: <https://theprivacyissue.com/government-surveillance/united-states-of-surveillance-us-history-spying>

[9] MACASKILL, Ewen a GREENWALD, Glenn: *NSA Prism program taps in to user data of Apple, Google and others* [online]. 2013 [cit. 2020-11-10]. Dostupné z: <https://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data>

[10] WALSH, Ray: *How secure are Dropbox, OneDrive, Google Drive and iCloud?* [online]. 2019 [cit. 2020-12-03]. Dostupné z: <https://proprivacy.com/cloud/guides/how-secure-is-cloud-storage>

[11] *Čo je Slobodný softvér?: Definícia slobodného softvéru* [online]. [cit. 2020-11-14]. Dostupné z: <https://www.gnu.org/philosophy/free-sw.html>

[12] KONČITÝ, Patrik OPEN DATA, OPEN SOURCE SOFTWARE A PRÁVO: Free a open-source software. Brno, 2020. Bakalářská. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce JUDr. Matěj Myška, Ph.D.

[13] *DLL Hijacking ëxchndl.dll"* [online]. [cit. 2020-11-24]. Dostupné z: <https://github.com/haiwen/seafile-client/issues/1309>

[14] *Nextcloud.com/LICENSE* [online]. [cit. 2020-11-24]. Dostupné z: <https://github.com/Nextcloud/Nextcloud.com/blob/master/LICENSE>

[15] *Stay in control* [online]. [cit. 2020-11-24]. Dostupné z: <https://Nextcloud.com/permissions/>

[16] KS, Ashutosh: *Self-hosted Cloud Storage: Nextcloud vs. ownCloud vs. Seafile: Table of comparison* [online]. 2020 [cit. 2020-11-24]. Dostupné z: < https://www.hongkiat.com/blog/Nextcloud-ownCloud-seafile/>

[17] *Nextcloud Files: Workflow management* [online]. [cit. 2020-11-24]. Dostupné z: <https://Nextcloud.com/files/>

[18] *End-to-end Encryption* [online]. [cit. 2020-11-24]. Dostupné z: <https://nextcloud.com/endtoend/>

[19] *Encryption details* [online]. [cit. 2020-11-24]. Dostupné z: <https://docs.nextcloud.com/server/latest/admin_manual/configuration_files/encryption_details.html>

[20] *Hardening and security guidance* [online]. [cit. 2020-11-24]. Dostupné z: <https://docs.nextcloud.com/server/latest/admin_manual/installation/harden_server.html?highlight=https>

[21] *Fresh from the conference: Nextcloud Desktop client 2.6.0RC1 with new Login Flow, second test version of Virtual Drive* [online]. 2019 [cit. 2020-12-03]. Dostupné z: < https://nextcloud.com/blog/fresh-from-the-conference-nextcloud/>

[22] *Plans and Pricing for Nextcloud Enterprise* [online]. [cit. 2020-11-24]. Dostupné z:
<https://Nextcloud.com/pricing/]>

[23] *Uploading big files > 512MB* [online]. [cit. 2020-11-24]. Dostupné z:
<https://docs.nextcloud.com/server/15/admin_manual/configuration_files/big_file_upload_configuration.html>

[24] *Code signing* [online]. [cit. 2020-12-07]. Dostupné z:
<https://docs.nextcloud.com/server/latest/admin_manual/issues/code_signing.html>

[25] FELDMAN, David: *Nextcloud vs ownCloud – The Whole Story* [online]. 20.7.2020 [cit. 2020-11-24]. Dostupné z:
<https://civihosting.com/blog/Nextcloud-vs-ownCloud/>

[26] *OwnCloud, AGPLv3 and the ownCloud Commercial License* [online]. [cit. 2020-11-24]. Dostupné z:
< https://owncloud.com/news/ownCloud-agplv3-ownCloud-license/>

[27] *Compare Nextcloud and ownCloud's popularity and activity* [online]. [cit. 2020-11-24]. Dostupné z:
<https://selfhosted.libhunt.com/compare-server-nextcloud-vs-owncloud?rel=cmp-lib>

[28] *File Lifecycle Management* [online]. [cit. 2020-11-24]. Dostupné z:
<https://marketplace.owncloud.com/apps/files_lifecycle>

[29] *Comprehensive Encryption* [online]. [cit. 2020-11-24]. Dostupné z:
<https://ownCloud.com/features/comprehensive-encryption/>

[30] *OwnCloud Security and Encryption 2.0 A Technical Overview* [online]. [cit. 2020-12-03]. Dostupné z:
<https://owncloud.com/wp-content/uploads/2015/09/Whitepaper_ownCloud_Security_and_Encryption_ENG_160404.pdf>

[31] *Pricing* [online]. [cit. 2020-11-24]. Dostupné z:
<https://ownCloud.com/pricing/>

[32] *Uploading big files > 512MB* [online]. [cit. 2020-11-24]. Dostupné z:
<https://doc.owncloud.org/server/9.0/admin_manual/configuration_files/big_file_upload_configuration.html>

[33] POORTVLIET, Jos: *EU governments choose independence from US cloud providers with Nextcloud* [online]. 2019 [cit. 2020-11-25]. Dostupné z: < https://nextcloud.com/blog/eu-governments-nextcloud/>

[34] *Server installation.: Simplified.* [online]. [cit. 2020-12-03]. Dostupné z: <https://github.com/nextcloud/vm>

[35] TRONCOSO, Carmela, PRENEEL, Bart a DE COCK Danny: *Improving secure long-term archival of digitally signed documents* [online]. 31.10.2008 [cit. 2021-5-17]. Dostupné z: <https://www.researchgate.net/publication/221103820_Improving_secure_long-term_archival_of_digitally_signed_documents>

[36] *NIST Policy on Hash Functions* [online]. 2015 [cit. 2021-5-17]. Dostupné z: <https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions>

[37] *FreeTSA.org: Time Stamp Authority* [online]. [cit. 2021-5-17]. Dostupné z: <https://www.freetsa.org/index_en.php>

[38] Rfc3161ng [online]. [cit. 2021-5-17]. Dostupné z: <https://github.com/trbs/rfc3161ng>

Zoznam symbolov a skratiek

NIST	Národný úrad pre štandardy a technológie – National Institute of Standards and Technology
OS	Operačný systém – Operation System
GPL	Všeobecná verejná licencia – General Public License
AGPL	Všeobecná verejná licencia – Affero General Public License
v3	verzia 3 – Version 3
LGPL	Menej všeobecná verejná licencia – Lesser General Public License
CVE	Bežné zranitelnosti a odhalenia – Common Vulnerabilities and Exposures
LDAP	Protokol ľahkého prístupu k adresáru – Lightweight Directory Access Protocol
SAML	Značkovací jazyk bezpečnostných tvrdení– Security Assertion Markup Language
GDPR	Všeobecná regulácia ochrany dát – General Data Protection Regulation
HTTP	Hypertextový prenosový protokol – Hypertext Transfer Protocol
HTTPS	Zabezpečený hypertextový prenosový protokol – Hypertext Transfer Protocol Secure
TLS	Bezpečnosť prenosovéj vrstvy – Transport Layer Security
AES	Pokročilý šifrovací štandard – Advanced Encryption Standard
GCM	Galois/Counter Mode
CTR	Počítadlo – Counter
RSA	Rivest–Shamir–Adleman
SHA	Bezpečný hašovací algoritmus – Secure Hash Algorithm
HMAC	Hašovaný autentifikačný kód správy – Keyed-Hash Message Authentication Code
CFB	Cipher Feedback Mode

LTS	posledný – Latest
RAM	Pamäť s náhodným prístupom – Random Access Memory
HDD	Pevný disk – Hard Drive
CPU	centralna procesorová jednotka – Central Processing Unit
PGP	Celkom dobré súkromie – Pretty Good Privacy
SSL	Vrstva bezpečných socketov – Secure Sockets Layer
TOTP	Časovo založené jednorazové heslo – Time-based One-Time Password
URL	Jednotný vyhľadávač prostriedku – Uniform Resource Locator
TS	Časová pečiatka – Timestamp
TSA	Autorita časových pečiatok – Time stamping Autorithy
CRL	Zoznam odvolaných certifikátov – Certificate Revocation List
FIFO	Prvý dnu, prvý von – First in, First out
CLA	Licenčná dohoda prispievateľa – Contributor Licence Agreement
ERD	Relačný diagram – Entity Relationship Diagram

Zoznam príloh

A Záťažové testy

Táto príloha obsahuje tabuľky k nameraným hodnotám počas vykonanej výkonovej analýzy Nextcloud serveru a archivačného systému.

Tab. A.1: Namerané hodnoty závislosti počtu pracovníkov a času spracovania úloh

počet úloh	počet pracovníkov	čas [s]
100	1	20,3
100	2	20,7
100	4	20,5
100	8	20,6
200	1	41,8
200	2	41,2
200	4	41,7
200	8	41,1

Tab. A.2: Hodnoty závislosti počtu úloh a času pre archivačného pracovníka

počet úloh	čas [s]
1	2,1
5	3,5
10	4,1
50	10,3
100	22,8
200	49,2
400	107,6

Tab. A.3: Hodnoty závislosti veľkosti súboru a času spracovania úlohy archivačným pracovníkom pri použití lokálneho úložiska

veľkosť súboru [KB]	čas [s]
1	2,1
10	2,2
100	2,3
1000	2,9
10000	2,7
100000	3,6
1000000	18,7
2000000	35,9

Tab. A.4: Hodnoty závislosti veľkosti súboru a času spracovania úlohy archivačným pracovníkom pri použití vzdialného úložiska

veľkosť súboru [KB]	čas [s]
1	2,3
10	2,6
100	2,9
1000	3,9
10000	5,7
100000	67,5
1000000	323,7
2000000	657,3

Tab. A.5: Hodnoty závislosti počtu úloh a času pre validačného pracovníka

počet úloh	čas [s]
1	1,71
5	3,36
10	6,68
50	23,95
100	38,3
200	59,2
400	115,6

Tab. A.6: Hodnoty závislosti času validácie a počtu vrstiev

počet vrstiev balíkov	čas [s]
1	1,71
2	1,6
4	1,74
8	1,65
16	2,34

Tab. A.7: Hodnoty závislosti veľkosti súboru a času spracovania úlohy pre validačného pracovníka

veľkosť súboru [KB]	čas [s]
1	2,1
10	1,5
100	2,2
1000	1,5
10000	4,5
100000	31,7
1000000	315,3
2000000	603,9

Tab. A.8: Hodnoty merania vzťahu medzi počtom úloh a časom, za ktorý sú tieto úlohy vykonané pracovníkom pre obnovu časových pečiatok

počet úloh	čas [s]
1	1,5
5	2
10	2,8
50	7,9
100	15,7
200	32,8
400	62,6

Tab. A.9: Hodnoty závislosti veľkosti súboru a času spracovania úlohy pracovníkom na obnovu časových pečiatok

veľkosť súboru [KB]	čas [s]
1	1,4
10	2
100	1,8
1000	1,85
10000	2,1
100000	2,9
1000000	9,2
2000000	18,5

Tab. A.10: Namerané hodnoty pre Nextcloud server

Typ požiadavky	Počet paralelne posielaných požiadaviek	Počet odoslaných požiadaviek	Počet vybavených požiadaviek za 1 sekundu	Počet úspešne vybavených požiadaviek
PUT	5	25	4,458	21
PUT	5	50	5,035	46
PUT	5	100	5,115	96
PUT	5	200	5,153	199
PUT	5	400	4,846	394
PUT	5	800	4,912	794
PUT	10	25	4,858	21
PUT	10	50	4,256	46
PUT	10	100	4,65	92
PUT	10	200	4,997	192
PUT	10	400	4,702	392
PUT	10	800	4,98	793
GET	5	25	4,531	21
GET	5	50	4,482	46
GET	5	100	4,961	96
GET	5	200	5,075	200
GET	5	400	4,963	397
GET	5	800	4,735	795
GET	10	25	5,083	24
GET	10	50	4,084	41
GET	10	100	4,679	94
GET	10	200	4,849	197
GET	10	400	4,292	394
GET	10	800	4,595	793
PROPFIND	5	25	3,77	21
PROPFIND	5	50	3,792	47
PROPFIND	5	100	4,362	97
PROPFIND	5	200	4,374	199
PROPFIND	5	400	4,234	398
PROPFIND	5	800	4,264	796
PROPFIND	10	25	4,952	21
PROPFIND	10	50	3,678	42
PROPFIND	10	100	4,199	91
PROPFIND	10	200	4,322	194
PROPFIND	10	400	4,495	393
PROPFIND	10	800	4,119	791

B Návod k spusteniu archivačného systému

B.1 Inštalácia databázy

V systéme musí existovať Mysql databáza, tú nainštalujeme nasledovnými príkazmi:

```
$ apt install gnupg
$ wget https://dev.mysql.com/get/mysql-apt-config_0.8.17-1_all.deb
$ sudo apt install ./mysql-apt-config_0.8.17-1_all.deb
$ sudo apt update
$ sudo apt install mysql-server
```

To či je databáza spustená overíme pomocou:

```
$ sudo systemctl status mysql
```

Následne môžeme prejsť k vytváraniu databázy a tabuliek. To spravíme tak, že spustíme tento príkaz zo zložky, kde je uložený archivačný systém:

```
$ mysql -u root -p <
            Archivation-System/database/sql_scripts/Create_db.sql
```

Keď máme vytvorenú databázu, vytvoríme v nej tabuľky pomocou nasledovného skriptu:

```
$ mysql -u root -p <
            Archivation-System/database/sql_scripts/Create_tables.sql
```

B.2 Inštalacia Rabbitmq serveru

Pre komunikáciu medzi jednotlivými časťami systému slúži Rabbitmq server. Ten je možné nainštalovať a prevádzkovať na oddelenom stroji. Pomocou nasledovných príkazov, ktoré je možné nájsť na oficiálnej stránke Rabbitmq.

```
$ sudo apt-get install curl gnupg debian-keyring
                            debian-archive-keyring
                            apt-transport-https -y
$ sudo apt-key adv - -keyserver "hkps://keys.openpgp.org"
                    - -recv-keys
$ sudo apt-key adv - -keyserver "keyserver.ubuntu.com"
                    - -recv-keys "F77F1EDA57EBB1CC"
$ curl -1sLf 'https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey'
                    | sudo apt-key add -
```

Programu apt nakonfigurujeme zdroje k Rabbitmq a jeho súčastiam.

```
$ nano /etc/apt/sources.list.d/rabbitmq.list
```

Do otvoreného súboru dopíšeme nasledovné riadky:

```
deb http://ppa.launchpad.net/rabbitmq/rabbitmq-erlang/ubuntu
bionic main
deb-src http://ppa.launchpad.net/rabbitmq/rabbitmq-erlang/ubuntu
bionic main
deb https://packagecloud.io/rabbitmq/rabbitmq-server/ubuntu/
bionic main
deb-src https://packagecloud.io/rabbitmq/rabbitmq-server/ubuntu/
bionic main
```

Následne spustíme:

```
$ sudo apt-get update -y
$ sudo apt-get install -y erlang-base \
    erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets \
    erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key \
    erlang-runtime-tools erlang-snmp erlang-ssl \
    erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl
$ sudo apt-get install rabbitmq-server -y --fix-missing
```

V tomto prípade by mal byť Rabbitmq server nainštalovaný a zapnutý, to vieme overiť pomocou:

```
$ systemctl status rabbitmq-server.service
```

Zapnutie manažment konzoly je dobrovoľné ale pre tento prípad ju zapneme a budeme v nej vytvárať poradovníky pre pracovníkov.

```
$ sudo rabbitmq-plugins enable rabbitmq_management
```

Po jej úspešnom zapnutí k nej môžeme pristúpiť cez prehliadač na lokálnej adrese http://localhost:15672, kde sa prihlásime pomocou základných prihlasovacích údajov, meno je „guest“ a heslo je taktiež „guest“. Ďalej je nutné vytvoriť poradovníky na Rabbimq serveri. To je možné viacerými spôsobmi. V tomto prípade sa vytvorí cez manažment konzolu, kde prejdeme do časti „queues“ a vytvorím nové poradovníky pomocou „add a new queue“. Je nutné vytvoriť poradovníky pre logy, archiváciu, obnovu pečiatok a validáciu. Je na užívateľovi ako ich nazve, pretože tento názov sa definuje potom v konfiguračných súboroch pre jednotlivých pracovníkov, to neplatí pre poradovník s logmi, ten sa musí volať „logs“. Je nutné vytvoriť aj poradovník pre chybne vykonané úlohy s názvom „failed_tasts“. V tomto prípade vytvoríme poradovníky s názvami: „failed_tasts“, „logs“, „archivation“, „validation“ a „retimestamping“. Ďalej je nutné ešte vytvoriť takzvanú „Exchange“, ktorú treba spojiť s poradovníkom pre logy. To sa spraví v zložke „exchange“ a rovnako sa tu pridá pomocou tlačidla „add a new exchange“, kde sa definuje typ „fanout“ a dá sa jej meno „log“. Potom v tabuľke uvidíme túto pridanú „Exchange“. Musíme ju rozklinkať a definovať políčka v sekcii „add binding from this exchange“. Tu stačí zadať len políčko „To queue“, kde sa zadá názov poradovníka, teda „logs“, a potvrdí sa

tlačítkom „bind“. V tomto prípade je spravená veľmi základná konfigurácia Rabbitmq serveru schopná spustenia na lokálnom stroji. V prípade, keby chceme spustiť Rabbitmq server na vzdialenom stroji, musíme v ňom vytvoriť nového užívateľa a pridať mu príslušné právomoci. To vieme spraviť taktiež cez manažment konzolu.

B.3 Spustenie archivačného systému

Na stroji, kde budeme spúšťať archivačný systém, musí byť nainštalovaný python3. Takisto je nutné mať nainštalovaný program PyPI, ktorý umožňuje sťahovanie python knižníc.

V zložke s archivačným súborom je textový súbor s názvom „requirments.txt“, ten je nutný s pomocou programu PyPI spustiť v termináli alebo príkazovom riadku

```
$ pip3 install -r requirments.txt
```

Teraz je možné spustenie jednotlivých skriptov a pracovníkov. Každý z nich potrebuje svoju konfiguráciu. Príklady ako má vyzerať sú v zložke ArchivationSystem/example_configs&files. K spusteniu všetkých skriptov na tvorenie úloh je tu príklad pod názvom testing_config.yaml. V ňom upravíme nastavenie Rabbitmq, teda IP adresu stroja s Rabbitmq, prihlasovacie údaje a dobrovoľne aj konfiguráciu zabezpečenú cez TLS. Príklad a popis konfigurácie je vidieť na nasledujúcom výpise.

Výpis B.1: Konfiguarácia pre pripojenie na Rabbitmq server

```
rabbitmq_connection:
    host: 'localhost' #IP adresa, kde je Rabbitmq server
    virtual_host: '/'
    port: '5672' #port, na ktorý sa pripája,
    #v prípade, že chceme použiť TLS, tak to bude port 5673
    credentials:
      name: guest #meno užívateľa na Rabbitmq serveri
      password: guest #heslo
    enable_ssl: False
    #Pokiaľ nechceme používať ssl/tls zabezpečenie,
    #tak ponecháme 'False'.
    #V prípade, že chceme zapnúť TLS/SSL, odstránime 'False,'
    #odkomentujeme všetky nasledovné riadky
    #a vyplníme ich príslušnými údajmi.
      #Server_name_id: str #optional
      #certificate_file: str-path
      #private_key_file: str-path
      #pk_password: str
      #CA_file: str-path
      #CA_path: str-path
      #CA_data: str-path
  rabbitmq_logging: #Informácie pre posielanie logov
    host: 'localhost'
    port: '5672'
    username: guest
    password: guest

  rabbitmq_info:
    consumer_ID: 'Archivator'
    #meno, pod ktorým bude príslušný pracovník konzumovať úlohy
    task_queue: 'task'
    #Meno poradovníka, kde budú zaslané alebo konzumované úlohy
    control_exchange: 'control'
```

Konfiguráciu databázy potrebuje len retimestamping_task_scheduler.py a pracovníci. Jej štandard a možnosti sú definované na stránke `https://dev.mysql.com/doc/connector-python/en/connector-python-connectargs.html`. Základná konfigurácia vyzerá nasledovne:

Výpis B.2: Konfiguarácia pre pripojenie na databázu

```
db_config:
    user : 'test_user' #meno užívateľa s prístupom k databáze
    password : 'Password1' #heslo užívateľa
    host : '127.0.0.1' #IP adresa, na ktorej je spustený SQL server
    database : 'ArchivationSystemDB' #názov používanej databázy
```

Po vytvorení konfigurácie sme schopní spustiť skripty na tvorbu úloh. Každý pracovník má vlastnú konfiguráciu, ich príklady spolu s popisom jednotlivých parametrov sú taktiež v zložke Archivation-System/example_configs&files/.

Pokiaľ je úložisko odkiaľ chceme archivovať súbory na vzdialenom stroji, musíme k nemu nastaviť prístup pomocou ssh. To platí pre archivačný a validačný pracovník. Spravíme to tým, že si najskôr vygenerujeme rsa kľúče pre ssh. Tie musíme upraviť do PEM formátu. To spravíme pomocou príkazu.

```
$ ssh-keygen -b 4096
$ ssh-keygen -p -m PEM -f ~/.ssh/id_rsa
```

Opýta sa nás to, kde chceme tieto kľúče uložiť. Túto cestu je nutné si zapamätať, pretože sa musí nadefinovať v konfiguračnom súbore pre archivačný pracovník. Treba myslieť na to, že užívateľ, za ktorého sa pracovník pripája na vzdialené uložisko musí mať právomoci k prístupu k daným súborom. To isté platí aj pre heslo, ktoré zadáme. Rovnako to platí aj pri lokálnom spustení. Následne tento kľuč musíme skopírovať na stroj s úložiskom. Skopírovanie spravíme pomocou príkazu. Namiesto mena užívateľa zadáme meno užívateľa na vzdialenom stroji, za ktorého sa budeme prihlasovať a namiesto „hostname“ zadáme IP adresu toho stroja:

```
$ ssh-copy-id menouzivatela@hostname
```

Použité údaje sa zadajú do konfiguračného súboru nasledovne:

Výpis B.3: Konfiguarácia pre pripojenie k vzdialenému úložisku

```
host: '' #IP adresa stroja s úložiskom
port: '22'
credentials:
    username: 'server'
    #meno užívateľa, za ktorého sa bude program prihlasovať
    key_filepath: ''
    #cesta k vygenerovanému autentizačnému kľúču
    password: '' #heslo k autentizačnému kľúču
```

V prípade, že úložisko je na lokálnom stroji, zadáme „remote_access: False“ a vymažeme zvyšné riadky sekcie.

V nasledovnej časti sú príklady konfiguračných súborov pre jednotlivých pracovníkov s popisom údajov v nich. Na užívateľovi je získanie príslušných častí ako je podpisovací kľuč, certifikát, CRL tohto certifikátu a podobne. V prípade záujmu o testovanie pomocou self-signed certifikátu, kedy je nemožné získať CRL, je nutné zakomentovať riadky: 86, 88 a 154 v súbore Archivation-System/archivation/archiver.py a riadok 207 v súbore Archivation-System/validation/validator.py. Pri všetkých riadkoch je komentár oznamujúci túto informáciu. V prípade záujmu o spustenie skriptu na konzumáciu logov je nutné nastaviť v consume_logs.py správne prihlasovacie údaje a IP adresu pre Rabbitmq server.

Výpis B.4: Konfigurácia pre archivačného pracovníka

```
archivation_system_info:
  storage_dir_path: '/home/server/Desktop/storage'
  #cesta, kde budú archivované súbory
  validity_length: 2
  #doba platnosti časových pečiatok
  signing_info:
  #sekcia s informáciami pre vytvorenie a overenie podpisu
  #mená všetkých súborov musia byť rovnaké
      certificate_path: '/home/server/Desktop/cert.pem'
      #cesta k certifikátu podpisovacieho kľúču,
      private_key_path: '/home/server/Desktop/key.pem'
      #cesta k podpisovaciemu kľúču
      pk_password: 'Password1' #heslo k tomuto kľúču
      crl_path: '' #CRL certifikatu
  remote_access: False
  TSA_info: #informácieho o TSA
    url: 'https://freetsa.org/tsr' #URL adresa TSA
    tsa_cert_path: '/home/server/Downloads/tsa.crt'
    #stiahnutý certifikát TSA
    tsa_crl_url: 'https://www.freetsa.org/crl/root_ca.crl'
    #URL, kde TSA poskytuje CRL k svojmu certifikátu
    tsa_ca_pem: '/home/server/Downloads/cacert.pem'
    #stiahnutý kľuč certifikačnej autority TSA.
```

Výpis B.5: Konfiguarácia pre pracovníka na obnovovanie pečiatok

```
retimestamping_info:
  validity_length: 2 #doba platnosti časových pečiatok
  TSA_info:
    url: 'https://freetsa.org/tsr' #URL adresa TSA
    tsa_cert_path: '/home/server/Downloads/tsa.crt'
    #stiahnutý certifikát TSA
    tsa_crl_url: 'https://www.freetsa.org/crl/root_ca.crl'
    #URL, kde TSA poskytuje CRL k svojmu certifikátu
    tsa_ca_pem: '/home/server/Downloads/cacert.pem'
    #stiahnutý kľuč certifikačnej autority TSA
```

Výpis B.6: Konfiguarácia pre validačný pracovník

```
validation_info:
  contact:
    email_server: smtp.gmail.com
    #emailový server pre odosielanie výsledkov
    sender_email: ''
    #email pre odosielanie správ
    sender_password: ''
    #heslo k tomuto emailu
    email: admin@gmail.com
    #Kontaktný email ak by bola validácia negatívna
    phone: 'xxxxxxxxxxxx'
    # Kontaktne číslo ak by bola validácia negatívna
  remote_access: False
  TSA_info:
    url: 'https://freetsa.org/tsr'
    #URL adresa TSA
    tsa_cert_path: '/home/server/Downloads/tsa.crt'
    #stiahnutý certifikát TSA
    tsa_crl_url: 'https://www.freetsa.org/crl/root_ca.crl'
    #URL, kde TSA poskytuje CRL k svojmu certifikátu
    tsa_ca_pem: '/home/server/Downloads/cacert.pem'
    #stiahnutý kľuč certifikačnej autority TSA.
```

Toto bolo základné nastavenie archivačného systému. Podrobnejšie informácie a možnosti nastavenia Rabbitmq a databázy je možné nájsť na ich oficiálnych stránkach.

Príklad spustenia pracovníkov je nižšie. Každý musí byť spustený v rôznych termináloch. V príkladoch sú použité cesty k ukážkovým konfiguračným súborom.

```
$ python3 run_archivation_worker.py
        -c 'example_config&files/archivation_worker_config.yaml'
$ python3 run_retimestamping_worker.py
        -c 'example_config&files/retimestamping_worker_config.yaml'
$ python3 run_validation_worker.py
        -c 'example_config&files/validation_worker_config.yaml'
```

Príklady spustenia skriptov pre tvorbu úloh s nutnými parametrami sú nižšie.

```
$ python3 archivation_task
        -c 'path/config.yaml'
        -fp '/path/file'
        -o '/owner'
$ python3 retimestamping_task_scheduler.py
      -ho 2
      -c 'path/config.yaml'

$ python3 validation_task.py -c 'path/config.yaml'
```

C Obsah elektronickej prílohy

- `/` ..... koreňový adresár priloženého archívu
  - `Archivation-System` ..... koreňový adresár archivačného systému
    - `archivation` ..... zložka s kódom pre archivaciu
      - `archivation_worker.py`
      - `archiver.py`
    - `common` ..... zložka s funkciami používanými naprieč systémom
      - `...`
    - `database` ..... zložka so súbormi pre databázu
      - `sql_scripts`
        - `Create_db.sql`
        - `Create_tables.sql`
        - `sql_queries.py`
      - `table_classes`
      - `db_library.py`
    - `example_configs&files` ..... zložka s príkladmi konfiguračných súborov
      - `archivation_worker_config.yaml`
      - `files_info.yaml`
      - `recipients_file.txt`
      - `retimestamping_worker_cfg.yaml`
      - `testing_config.yaml`
      - `validation_worker_config.yaml`
    - `rabbitmq_connection` ..... zložka s kódom pre komunikáciu s Rabbitmq
      - `...`
    - `retimestamping` ..... zložka s kódom na obnovu časových pečiatok
      - `retimestamping_worker.py`
      - `retimestamper.py`
    - `task_makers` ..... zložka s kódom pre vytváranie úloh
      - `...`
    - `validation` ..... zložka s kódom pre validáciu
      - `validation_worker.py`
      - `validator.py`
    - `Archivation_task_maker.sh`
    - `archivation_task.py`
    - `consuming_logs.py`
    - `requirements.txt`
    - `retimestamp_scheduler.py`
    - `run_archivation_worker.py`
    - `run_retimestamping_worker.py`
    - `run_validation_worker.py`
    - `validation_task.py`
  - `script_trigger` ..... zložka s upravenou aplikáciou Workflow External Scripts
    - `...`