

# Identifikace vlastníků bitcoinových adres

Identification of Bitcoin Address Owners

Bc. Adam Šárek

Diplomová práce

Vedoucí práce: Ing. Jan Plucar, Ph.D.

Ostrava, 2023

# Zadání diplomové práce

Student: **Bc. Adam Šárek**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Identifikace vlastníků bitcoinových adres  
Identification of Bitcoin Address Owners

Jazyk vypracování: čeština

## Zásady pro vypracování:

Bitcoin je v současnosti nejrozšířenější kryptoměnou na planetě. Pomocí kryptoměny Bitcoin je realizován decentralizovaný elektronický platební systém, jehož základy definoval člověk nebo skupina lidí používajících alias Satoshi Nakamoto. Pro identifikaci odesílatele a příjemce platby se využívá Bitcoinových adres. Bitcoinová adresa je jedinečný identifikátor, který slouží jako virtuální místo, kam lze kryptoměnu odeslat.

V této práci se student zaměří na tvorbu prostředí a nástrojů pro deanonymizaci vlastníků Bitcoinových adres. Proces deanonymizace je proces propojení veřejné bitcoinové adresy s digitálním identifikátorem uživatele (přezdívka, email a v omezených případech i jméno a příjmení). Navržené prostředí bude obsahovat databázi Bitcoinových adres, u kterých budou doplněny veškeré nalezené informace. Dále musí student vytvořit prostředí (např. webovou stránku), ve kterém bude schopen uživatel data procházet a ručně editovat. Posledním úkolem je návrh a implementace nástroje, který by byl schopen automatizovaně vyhledávat, analyzovat a ukládat informace o Bitcoinových adresách.

## Hlavní body zadání:

1. Rešerše aktuálního stavu.
2. Návrh prostředí, které se bude skládat z několika samostatných nástrojů a databáze Bitcoinových adres.
3. Implementace nástrojů.
4. Otestování řešení a diskuze výsledků.

## Seznam doporučené odborné literatury:

1. WU, Yan, et al. A bitcoin transaction network analytic method for future blockchain forensic investigation. IEEE Transactions on Network Science and Engineering, 2020, 8.2: 1230-1241.
2. HONG, Seongho; KIM, Heeyoul. Analysis of Bitcoin exchange using relationship of transactions and addresses. In: 2019 21st International Conference on Advanced Communication Technology (ICACT). IEEE, 2019. p. 67-70.
3. FLEDER, Michael; KESTER, Michael S.; PILLAI, Sudeep. Bitcoin transaction graph analysis. arXiv preprint arXiv:1502.01657, 2015.
4. MARCHENKO, Yuriy; KNOTTENBELT, William J.; WOLTER, Katinka. EthExplorer: A Tool for Forensic Analysis of the Ethereum Blockchain. In: European Workshop on Performance Engineering. Springer, Cham, 2019. p. 100-117.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Plucar, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního oboru: prof. RNDr. Václav Snášel, CSc.

V IS EDISON zadáno: 07.11.2022 11:59:22

## Abstrakt

Cílem práce je nejprve prozkoumat existující nástroje a databáze obsahující Bitcoinové adresy včetně dalších údajů s nimi spojených a poté implementovat vlastní řešení. Řešení se skládá ze dvou nástrojů. První prochází volně dostupné webové stránky, ty parsuje a následně z nich stahuje dostupná data. Z dostupných dat provádí selekci zejména Bitcoinových adres. Druhý nástroj poskytuje data uživateli na základě jeho role v systému. Uživatel má možnost vyhledat konkrétní kryptoměnovou adresu či získat seznam adres, ve kterém lze filtrovat dle kryptoměny či zdroje dat. Nástroj pak navíc obsahuje API, které nabízí stejnou funkcionalitu bez nutnosti použití webového rozhraní. Nástroj pro sběr dat je navržen v jazyce Python s využitím PostgreSQL databáze. Druhý zmíněný nástroj je napsán v JavaScriptu v prostředí Node.js. Oba nástroje je možné provozovat souběžně.

## Klíčová slova

Bitcoin; Bitcoinová adresa; pseudonymní; databáze; nahlášení; crawler; web; API; Python; Node.js; JavaScript; PostgreSQL

## Abstract

The goal of this work is to first examine existing tools and databases containing Bitcoin addresses and other data associated with them, and then implement our own solution. The solution consists of two tools. The first one crawls freely available websites, parses them and then downloads the available data from them. From the available data, it performs a selection of Bitcoin addresses in particular. The second tool provides data to the user based on his role in the system. The user is able to search for a specific cryptocurrency address or obtain a list of addresses, which can be filtered by cryptocurrency or data source. The tool then additionally includes an API that offers the same functionality without the need for a web interface. The data collection tool is designed in Python using a PostgreSQL database. The second tool mentioned above is written in JavaScript in Node.js. Both tools can be run in parallel.

## Keywords

Bitcoin; Bitcoin address; pseudonymous; database; report; crawler; website; API; Python; Node.js; JavaScript; PostgreSQL

## **Poděkování**

Rád bych na tomto místě poděkoval vedoucímu mé diplomové práce, Ing. Janu Plucarovi, Ph.D. za aktivní přístup, odborné vedení a zpětnou vazbu poskytnutou při vypracování práce. Dále bych chtěl poděkovat krajskému ředitelství policie Olomouckého kraje za náhled do oblasti kryptoměnové kyberkriminality a jejího vyšetřování.

# Obsah

Seznam použitých symbolů a zkratk	8
Seznam obrázků	10
Seznam tabulek	11
<b>1 Úvod</b>	<b>13</b>
<b>2 State of the art</b>	<b>15</b>
2.1 Stávající práce shromažďující BTC adresy při odhalování podvodů . . . . .	15
2.2 Databáze adres a nahlášených podvodů . . . . .	19
2.3 Shrnutí relevantních zdrojů . . . . .	21
<b>3 Open Source Intelligence</b>	<b>22</b>
3.1 Nástroje pro analýzu kryptoměn . . . . .	22
<b>4 Požadavky a návrh systému</b>	<b>25</b>
4.1 Požadavky . . . . .	25
4.2 Případy užití . . . . .	26
4.3 Návrh databáze . . . . .	28
4.4 Komponenty systému . . . . .	29
<b>5 Implementace nástroje BTCAbuseCrawler</b>	<b>30</b>
5.1 Procházení zdrojů . . . . .	30
5.2 Propojování dat s adresami . . . . .	35
5.3 Zjištění kryptoměnové příslušnosti altcoinových adres . . . . .	35
5.4 Automatizovaný běh . . . . .	36
5.5 Správa databáze . . . . .	37
<b>6 Implementace nástroje BTCAbuseSearch</b>	<b>40</b>
6.1 Vyhledávání kryptoměnové adresy . . . . .	41

6.2	Seznam adres . . . . .	48
6.3	Stránka se statistikami . . . . .	49
6.4	Uživatelské účty . . . . .	50
6.5	API . . . . .	53
<b>7</b>	<b>Použité technologie</b>	<b>55</b>
7.1	PostgreSQL . . . . .	55
7.2	Node.js . . . . .	56
7.3	Express.js . . . . .	56
7.4	Embedded JavaScript templates . . . . .	56
7.5	GitHub . . . . .	57
<b>8</b>	<b>Výsledky</b>	<b>58</b>
8.1	Poměr nahlášených BTC adres . . . . .	58
8.2	Další nahlášené adresy . . . . .	59
<b>9</b>	<b>Závěr</b>	<b>60</b>
	<b>Literatura</b>	<b>61</b>
	<b>Přílohy</b>	<b>63</b>
<b>A</b>	<b>Uživatelská příručka</b>	<b>64</b>
A.1	Příprava nástrojů . . . . .	64
A.2	Spuštění nástrojů . . . . .	65
A.3	Adresářový strom . . . . .	66
<b>B</b>	<b>Datový slovník</b>	<b>68</b>
<b>C</b>	<b>Specifikace testovacího zařízení</b>	<b>74</b>

# Seznam použitých zkratek a symbolů

altcoin	– Alternativní kryptoměna, označení pro kryptoměny mimo Bitcoin
API	– Application Programming Interface
BTC	– Bitcoin
CSS	– Cascading Style Sheets
CSV	– Comma-separated values
EJS	– Embedded JavaScript templates
FK	– Cizí klíč
GB	– Gigabajt
GNU	– GNU's Not Unix!
GZ	– GNU zip
HTML	– Hyper Text Markup Language
HTTP	– Hypertext Transfer Protocol
HTTPS	– Hypertext Transfer Protocol Secure
ID	– Identifikátor
IP	– Internet Protocol
JIT	– Just-in-Time
JS	– JavaScript
JSON	– JavaScript Object Notation
kB	– Kilobajt
LTS	– Dlouhodobá podpora
OS	– Operační systém
OSINT	– Open Source Intelligence
PHP	– PHP: Hypertext Preprocessor
PK	– Primární klíč
RAM	– Random Access Memory
SQL	– Structured Query Language
SSD	– Solid-state drive
TB	– Terabajt



TXT	– Textový soubor
UAC	– Řízení uživatelských účtů
UC	– Příklad užití
UML	– Unified Modeling Language
UQ	– Unikátní klíč
URL	– Uniform Resource Locator
WAL	– Write-Ahead Log
XML	– Extensible Markup Language

# Seznam obrázků

2.1	Příklad Bitcoinového generátoru [1] . . . . .	16
2.2	Proces shromažďování dat o politických kandidátech USA [4] . . . . .	17
3.1	Ukázka nástroje z webu <code>chainalysis.com</code> [20] . . . . .	23
3.2	Ukázka nástroje z webu <code>elliptic.co</code> [21] . . . . .	23
4.1	Diagram případů užití . . . . .	26
4.2	Diagram vztahu entit databáze . . . . .	28
4.3	Diagram komponent systému . . . . .	29
5.1	Ukázka seznamu souborů ve složce <code>crawled_data</code> . . . . .	35
5.2	Ukázka konzole crawleru . . . . .	37
6.1	Úvodní stránka webu . . . . .	41
6.2	Příklad kryptoměnového mixéru [22] . . . . .	43
6.3	Příklad nahlášení obsahujícího přezdívkou . . . . .	46
6.4	Příklad nahlášení obsahujícího jméno . . . . .	47
6.5	Ukázka části seznamu BTC adres . . . . .	49
6.6	Ukázka části statistik . . . . .	50
6.7	Správa uživatelů . . . . .	52

# Seznam tabulek

2.1	Zdroje kryptoměnových adres . . . . .	21
4.1	Specifikace požadavků na uživatelské účty . . . . .	25
4.2	Případ užití 14: Úprava role uživatele . . . . .	27
6.1	Typy informací obsažené v nahlášení kryptoměnové adresy . . . . .	44
6.2	Možnosti jednotlivých uživatelských rolí . . . . .	51
6.3	Limity počtu požadavků na API pro jednotlivé uživatelské role . . . . .	51
6.4	Seznam funkcí API . . . . .	53
6.5	Seznam parametrů v požadavcích API . . . . .	54
B.1	Databázová tabulka account . . . . .	68
B.2	Databázová tabulka address . . . . .	69
B.3	Databázová tabulka address_data . . . . .	69
B.4	Databázová tabulka currency . . . . .	70
B.5	Databázová tabulka data . . . . .	70
B.6	Databázová tabulka role . . . . .	70
B.7	Databázová tabulka session . . . . .	71
B.8	Databázová tabulka source . . . . .	71
B.9	Databázová tabulka source_label . . . . .	72
B.10	Databázová tabulka source_label_url . . . . .	72
B.11	Databázová tabulka token . . . . .	73
B.12	Databázová tabulka url . . . . .	73
C.1	Specifikace testovacího zařízení . . . . .	74

# Seznam výpisů zdrojového kódu

8.1	Získání počtu podvodných BTC adres . . . . .	58
8.2	Získání celkového počtu BTC adres . . . . .	58
8.3	Získání počtu podvodných altcoinových adres . . . . .	59

# Kapitola 1

## Úvod

Motivací práce bylo navrhnout a implementovat prostředí složené z několika nástrojů s cílem vytvořit a udržovat databázi Bitcoinových adres. Databáze kromě Bitcoinových adres obsahuje také další informace jako uživatelské nahlášení podvodů spojených s adresami a zdroje, ze kterých byly nahlášení získány. Informace jsou v databázi vzájemně propojeny, což umožňuje jejich nalezení po vyhledání určité adresy.

Pro uživatele je primárním cílem vyhledávání získat informace o možném riziku spojeném s danou adresou. Výhodou agregace nahlášení z několika zdrojů je to, že u vyhledané adresy lze všechny zobrazit přehledně na jednom místě bez nutnosti navštívit jakýkoliv externí web. Následnou analýzu pak umožňují odkazy na přesné místo, kde byly informace získány a také odkazy na blockchainové průzkumníky, kde je možné nalézt veškeré transakce dané adresy.

K databázi má uživatel přístup skrze implementovaný nástroj BTCAbuseSearch a je možné pomocí něj vyhledávat konkrétní adresy či procházet seznam nahlášených adres, ve kterém lze filtrovat adresy dle zdroje, na kterém byla adresa nahlášena či dle kryptoměny. Volba kryptoměny je zde proto, jelikož v rámci průzkumu stávajících řešení byly nalezeny zdroje, které z většiny obsahují i jiné než jen Bitcoinové adresy. Díky tomu tak byla výsledná databáze rozšířena i o nahlášení jiných kryptoměn, a to z toho důvodu, že tak poskytuje širší využití. Stále je však databáze zaměřena primárně na Bitcoinové adresy a je možné v případě potřeby toto rozšíření později odebrat.

Vedoucím práce bylo vyvíjené řešení na počátku jeho vývoje prezentováno na krajském ředitelství policie v Olomouci. Policie projevila o práci určitý zájem a díky jejím doporučením je v řešení obsaženo více úrovní uživatelských rolí, aby měla případně vlastní přístup k datům a také bylo zahrnuto zálohování procházených stránek do souborů pro ověření pravosti informací při případné změně původního zdroje.

Co se týká obsahu jednotlivých kapitol, tak první kapitola se věnuje studiu state of the art nástrojů, které shromažďují Bitcoinové adresy jejich vyhledáváním na internetu. Cílem kapitoly je nejen získat přehled o možných řešeních, ale také ujasnit si, jak by mohla finální databáze vypadat. Jsou zde také uvedeny zdroje, které byly následně použity při implementaci řešení.

Druhá kapitola pojednává o Open Source Intelligence (OSINT), což je způsob získávání informací z veřejně dostupných zdrojů. Popisuje, jaká data lze pomocí OSINT získat, k čemu mohou sloužit a k čemu jsou v práci využívány. Následně je představeno několik nástrojů pro analýzu kryptoměn, které využívají tuto metodu.

Další kapitola se pak zabývá analýzou požadavků, případy užití jednotlivých nástrojů, architekturou systému a návrhem databáze. Zmíněné informace jsou doplněny o vhodné UML diagramy.

Jelikož se řešení skládá z několika nástrojů, tak byla jejich implementace rozdělena do dvou samostatných kapitol.

První z nich se zabývá crawlerem. Crawler nejprve vytváří databázi, zavádí určité optimalizace databázového serveru a poté prochází požadované zdroje. Procházení začíná získáním seznamu všech Bitcoinových adres z blockchainu. Následně jsou procházeny zdroje obsahující seznamy nahlášení a zdroje které je možné prohledávat pouze pomocí jejich vyhledávače. K tomu jsou také mimo jiné potřeba již získané adresy z blockchainu. Data jsou po stažení propojena s adresami, aby bylo možné je použít při vyhledávání, a nakonec jsou také identifikovány kryptoměny adres, kterým by tato informace v databázi chyběla. Celkově je v rámci implementace crawleru zavedeno několik optimalizací, aby byl celý proces co možná nejrychlejší, nicméně i tak je jeden běh programu otázkou víceméně několika dnů. Například pouze stažení veškerých dat z blockchainu trvá při běžném připojení několik hodin. To je také jedním z důvodů, proč je celý proces co možná nejvíce automatizován, aby po spuštění nástroje nebylo nutné do programu jakkoliv zasahovat.

Webový nástroj se v podstatě skládá ze dvou nástrojů, jelikož kromě webové aplikace je jeho součástí také API. Implementace nástroje BTCAbuseSearch nejprve představuje řešení vyhledávání adresy a to, jaké informace jsou u dané adresy na webu obsaženy. Dále se zabývá deanonymizací vlastníka Bitcoinové adresy pomocí těchto informací a jejich důvěryhodností. Následně se věnuje stránce se seznamem nahlášených adres a způsobu jejich filtrování. Zmíněna je také stránka s celkovými statistikami aktuálního stavu databáze. Dále pak přibližuje zapojení uživatelských účtů a jejich možnosti nejen v rámci webové aplikace, ale také v rámci API. Jaké funkce API nabízí a pro jaké účely byla vytvořena, představuje poslední část implementace.

V následující kapitole se práce zabývá popisem jednotlivých technologií použitých při implementaci. Jsou zde zmíněné technologie PostgreSQL, Node.js, Express.js, EJS a GitHub.

Poslední kapitola obsahuje zhodnocení výsledků práce. Představuje jak celkový počet BTC adres, tak počet získaných nahlášených BTC adres, které uvádí do kontextu s dalšími statistikami z internetu týkajícími se kriminality spojené s Bitcoinem. Je zde také kratší zmínka o shromážděných altcoinových adresách vzhledem k možnému rozšíření práce o podporu dalších kryptoměn.

Součástí práce je také několik příloh. První se zabývá přípravou a spuštěním obou nástrojů, což se hodí minimálně při prvním nasazování na cílovém zařízení. Další příloha obsahuje datový slovník, který přibližuje vnitřní strukturu databáze a význam jednotlivých tabulek. Poslední příloha pak obsahuje specifikace zařízení, na kterém bylo celé řešení vyvíjeno a testováno a od nějž se odvíjí veškeré časové odhady běhu nástrojů.

## Kapitola 2

# State of the art

State of the art se zaměřuje na průzkum stávajících řešení v oblasti shromažďování Bitcoinových adres a jejich následnému ukládání do databáze společně s dalšími údaji. Cílem je získat přehled o aktuálním stavu, inspirovat se či využít silných stránek existujících nástrojů a identifikovat možné vylepšení a funkce, které je potřeba implementovat.

V sekci 2.1 je zmíněno několik prací zaměřených na shromažďování BTC adres. Práce mají za cíl odhalit kryptoměnové podvody, které se na internetu vyskytují. Jedná se např. o nezákonné financování politických kampaní, odhalení Ponziho schémat či podvodu Bitcoinového generátoru. V následující sekci jsou pak sepsány veškeré nalezené databáze adres spojených s nahlášenými podvody. Zmíněné databáze jsou pak shrnuty na konci kapitoly.

## 2.1 Stávající práce shromažďující BTC adresy při odhalování podvodů

### 2.1.1 Automatická detekce a analýza podvodu Bitcoinového generátoru

Autoři pracovali na nástroji sloužícímu k vyšetřování „podvodu Bitcoinového generátoru“. Ten funguje na jednoduchém principu, kdy podvodník se s pomocí jednoduchého webu snaží z oběti vylákat peníze tím, že požaduje provedení platby v bitcoinech na uvedenou Bitcoinovou adresu a na oplátku nabízí dvojnásobek odeslané sumy. Peníze však již nikdy nepošle.

Jejich nástroj umí takový druh webů detekovat a k jejich vyhledávání používá vlastní crawler a sadu vyhledávačů. Nalezené weby jsou následně monitorovány s cílem sledování plateb a použitých Bitcoinových adres. Systém se od ostatních odlišuje tím, že nepoužívá analýzu transakcí na blockchainu, nýbrž aktivně vyhledává podvodné weby. Díky tomu je schopen nalézt Bitcoinové adresy, které mají velmi málo či dokonce žádné přijaté transakce. Více než polovina adres, na které byly nakonec nějaké prostředky odeslány, byly detekovány ještě před tím, než na ně byla odeslána první platba.

V období od listopadu 2019 do února 2020 bylo s tímto druhem podvodu nalezeno více než 1 300 adres na více než 500 doménách. Celkově bylo na shromážděné adresy odesláno více než 5 milionů amerických dolarů. Průměrná transakce činila 47,3 dolaru. Data k jejich práci je možné nalézt na <https://bit.ly/bgsieeesb2020>. [1]



Obrázek 2.1: Příklad Bitcoinového generátoru [1]

### 2.1.2 Platforma pro shromažďování kryptoměnových adres

Uvedená práce se zaměřuje na vývoj platformy pro sběr a zobrazování metadat o kryptoměnových adresách nejen z povrchového, ale také z temného webu. Součástí platformy je také crawler, který na začátku projde všechny weby, které by měly být prohledány a následně ukládá URL adresy v nich obsažené včetně příslušných metadat. Adresy pak dále prochází a případně z nich získává potřebné informace pomocí scraperů. Scrapery analyzují webové stránky podle procházené adresy a extrahují zajímavá metadata. K scrapování je použita PHP knihovna Goutte. [2]

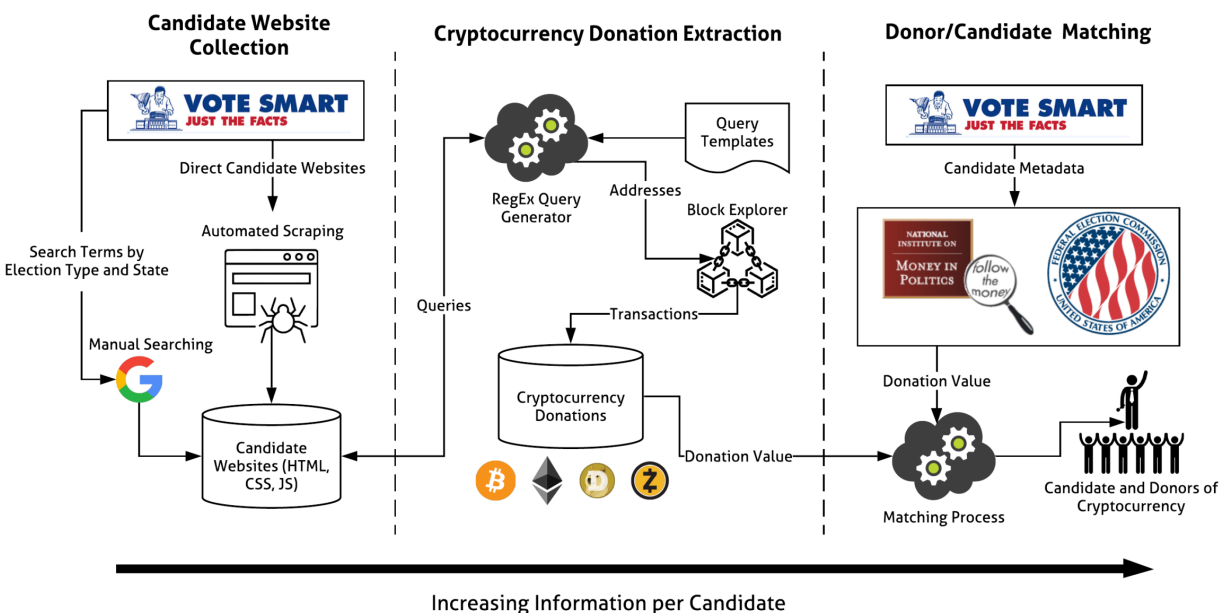


### 2.1.3 Údržba úložiště Bitcoinových adres prostřednictvím cíleného procházení webu

Autorka práce používá pravidelné crawlování uživatelem zadaných kryptoměnových stránek, jejichž výsledky ukládá do XML souboru, který je možné exportovat do libovolné databáze. Získané adresy a metadata mohou být použity při vyšetřování trestných činů spojených s Bitcoinem. Data o transakcích získaných adres jsou poté dostupné na Bitcoinovém blockchainu. [3]

### 2.1.4 Odhalování nezákonných kryptoměnových příspěvků v politických kampaních

Kryptoměny jsou v dnešní době používány také k financování amerických politických kampaní. Alespoň to tvrdí následující práce, která analyzuje aktuální stav používání kryptoměn v kampaních napříč USA a shromažďuje data o webových stránkách kandidátů, kryptoměnových adresách a zprávách o darech zaslaných federálním regulačním orgánům. Při porovnání transakcí nad danými adresami a potvrzeními od federální volební komise je možné pozorovat, že mnoho kandidátů své kryptoměnové dary nehlásí. Někteří kandidáti také výrazně podhodnocují přijaté dary. Práce také uvádí doporučení pro vytvoření systému, kde by bylo možné využít kryptoměny pro politické dary, aniž by mohlo dojít k nesprávnému zacházení s finančními prostředky či vyhýbání se federálním směrnicím. V rámci centrální platformy by dárce zaregistrovali své kryptoměnové adresy, které by poté sloužili k ověření identity dárce.



Obrázek 2.2: Proces shromažďování dat o politických kandidátech USA [4]

K analýze využití kryptoměnových darů byla shromážděna data z webových stránek (získaných ze stránek VoteSmart [5]) 4 808 kandidátů, které byly doplněny o ručně vyhledané stránky na Google pomocí parametrů získaných od VoteSmart. Všechny ručně získané stránky místo adres používaly službu BitPay. Pomocí automatizačního nástroje Selenium poté bylo úspěšně staženo 4 637 stránek. Celkem bylo nalezeno 20 kryptoměnových adres nad kterými proběhlo 157 transakcí. [4]

### 2.1.5 Detekce Bitcoinových Ponziho schémat pomocí dolování dat

Autoři práce se zaměřili na dolování dat s cílem detekovat Ponziho schémata, která využívají Bitcoin. Ponziho schéma je označení pro podvodnou investici, ve které jsou uživatelům vypláceny prostředky pomocí investic nových uživatelů. Celý podvod zdánlivě funguje až do chvíle, kdy již není možné najít nové investice. Navzdory tomu, že jsou Ponziho schémata v mnoha zemích nezákonná, se dnes po celém světě rozšiřují a využívají k tomu právě Bitcoin. Kybernetičtí zločinci zneužívají Bitcoinové pseudonymity k provádění svých podvodů, jelikož je složité vysledovat vlastníky daných adres.

K získání Bitcoinových adres je provedeno manuální vyhledávání na stránkách Reddit a [bitcointalk.org](https://bitcointalk.org), což jsou hlavní diskusní fóra o Bitcoinu. Práce se zaměřuje na podkategorii „Hazardní hry: Hry založené na investorech“ na [bitcointalk.org](https://bitcointalk.org), kde podvodníci obvykle propagují Ponziho schémata jako „investiční programy s vysokým výnosem“ nebo různé hazardní hry. Pouze v několika případech reklamy výslovně obsahují Bitcoinovou adresu. Obvykle je však k získání dané adresy nutné navštívit konkrétní webovou stránku, na které je dané Ponziho schéma provozováno. Mnoho webových stránek však již není online, a tak je snaha obnovit jejich snímek prostřednictvím [archive.org](https://archive.org). Na každé získané stránce jsou pak ručně vyhledány jednotlivé adresy. Toto vyhledávání je dále doplněno o adresy vypsane na [blockchain.info/tags](https://blockchain.info/tags), což je webová stránka, která umožňuje uživatelům označovat Bitcoinové adresy. Většina označených adres také obsahuje odkaz na stránku, na které se objevuje. V rámci dané práce byl také vyvinut crawler, který automaticky prohledává získané stránky a ohodnocuje je na základě počtu obsažených slov spojených s Ponziho schématem. K tomuto účelu je použit slovník obsahující slova jako např. „Ponzi“, „zisk“, „multiplikátor“ či „investice“. Crawler prošel více než 1 500 stránek (souvisejících s přibližně 3 500 označeními) a zjistil, že přibližně 900 z nich obsahuje nějaké slovo související s tímto podvodem. Přibližně 600 stránek však již není přístupných, a to ani přes [archive.org](https://archive.org). [6]

### 2.1.6 Analýza útoku Man-in-the-middle nad Bitcoinovou adresou

Poslední práce se zabývá analýzou útoku Man-in-the-middle nad Bitcoinovou adresou. Tento útok spočívá v tom, že útočník nahradí původní Bitcoinovou adresu za svou vlastní. Na první pohled nemusí být zřejmé, že k takovéto změně došlo, jelikož jsou kryptoměnové adresy v podstatě dlouhý řetězec různých znaků a čísel. Běžný uživatel si nemá jak ověřit, že daná adresa skutečně patří osobě, které chce své kryptoměny poslat. Vzhledem k tomu pak také tento útok může být i dlouhodoběji úspěšný, alespoň tedy za předpokladu, že příjemce neočekává relativně pravidelný příjem kryptoměn.

Analýza spočívá v procházení stránek, které obsahují Bitcoinové adresy, a tedy přijímají platby v bitcoinech. Dále pak určuje, kolik adres je vystaveno útoku Man-in-the-middle. Webové stránky, které byly při analýze procházeny jsou získávány ze stránky blockchain.info, která obsahuje u jednotlivých adres také stránky na kterých se adresy nacházejí. K následnému vyhledání Bitcoinové adresy na dané stránce je použito filtrování pomocí regulárních výrazů. Součástí analýzy je také to, kolik adres používá HTTP/HTTPS během transakcí a kolik adres je aktivních či neaktivních. Celkově bylo nalezeno přibližně deset tisíc adres (z toho okolo osm tisíc aktivních), nad kterými proběhlo okolo 750 tisíc transakcí o celkové hodnotě přibližně 850 tisíc bitcoinů. [7]

## 2.2 Databáze adres a nahlášených podvodů

V rámci přípravy databáze Bitcoinových adres a informací s nimi spojených bylo nejprve potřeba najít již existující zdroje. Ideálně by pak každý zdroj měl obsahovat také nějaké další informace a metadata, které by bylo možné dále využít.

Při průzkumu existujících prací na téma Bitcoinu a dalších kryptoměn bylo nalezeno několik prací, které se zmiňují o nějaké interní databázi adres. Pouze jedna z prací (Bitcoin Generator Scam [1]) však obsahovala odkaz k datům, které navíc obsahovaly pouze seznam adres bez dalších informací k jednotlivým adresám.

Existující databáze adres samozřejmě nebyly sbírány pouze z akademických zdrojů, ale také pomocí vyhledávačů povrchové části internetu (vyhledávač Google). Tato akce již byla o něco úspěšnější, jelikož se podařilo najít celkem devět dalších zdrojů. Kvalita zdrojů se ve velkém liší, jak v počtu adres, tak v počtu informací ke konkrétním adresám, až po metody přístupu k těmto datům.

Vzhledem k tomu, že veškerá historie Bitcoinových transakcí je zaznamenána na Bitcoinovém blockchainu, tak jediné adresy, které mohou být součástí transakcí či mohou obsahovat nějaký zůstatek jsou právě adresy, které se někdy objevily na blockchainu.

Zdroj LoyceV zpracovává blockchainové data a poskytuje seznam všech adres v pořadí dle jejich výskytu na blockchainu či seznam všech adres s nenulovým zůstatkem v pořadí dle výše zůstatku. Seznam nenulových adres je identický se seznamem poskytovaným službou BlockChair, jelikož LoyceV data čerpá právě z BlockChair. Nevýhodou tohoto zdroje je, že se o jeho udržování stará jednotlivec, který může tuto činnost kdykoliv ukončit. V takovém případě by bylo nutné vytvořit vlastní řešení pro zpracování blockchainových dat. Dále také může být problém rozsah dat, jelikož počet všech adres dosahuje více než jedné miliardy, a tedy může být obtížné adresy do databáze uložit. [8]

BlockChair kromě již zmíněných nenulových adres obsahuje také data o Bitcoinových blocích, transakcích, vstupech a výstupech. Blockchair je engine pro vyhledávání a analýzu na blockchainu Bitcoinu a dalších kryptoměn. Služba si dává za cíl se stát Googlem pro blockchainya. Data poskytuje v souborech, které jsou aktualizovány jednou denně a případně také pomocí API. Hlavním problémem je, že bezplatná API umožňuje max. 1 440 požadavků za den a stahování souborů má limitovanou rychlost na 100 kB/s, což vede k tomu, že stahování dat, oproti předchozí zmíněné službě, trvá i více než několik hodin. [9]

BitcoinAbuse je veřejná databáze Bitcoinových adres používaných podvodníky, hackery a zločinci. BitcoinAbuse umožňuje nahlášovat Bitcoinové adresy a nahlášení poté stahovat pomocí API v podobě CSV souboru. Soubory obsahují nahlášení buď za jeden den, třicet dní či za celou historii, nicméně poslední možnost není aktuálně v provozu. Data jde každopádně kromě API také ručně vyhledat přímo na webu. [10]

CheckBitcoinAddress je stránka, za kterou stojí firma Apirone, která poskytuje služby pro příjem, zpracování a posílání plateb v kryptoměnách. CheckBitcoinAddress čerpá Bitcoinové adresy ze služby BitcoinAbuse a na webu poskytuje seznam veškerých nahlášení k daným adresám ve kterém je možné ručně vyhledávat. Nevýhodou služby je absence API. [11]

BitcoinAIS je služba, která sleduje Bitcoinové adresy používané v nevyžádaných emailech, ransomware, malware a k jiným podvodům. Na webu je možné ručně vyhledávat v seznamu posledních několika nahlášení. Nevýhodou služby je, že neposkytuje API pro vyhledávání a také, že zobrazuje pouze omezený počet nahlášení. [12]

Bitcoin Generator Scam obsahuje data získané v práci „An Automatic Detection and Analysis of the Bitcoin Generator Scam“. [1] Práce vznikla s cílem nalézt podvody typu „Bitcoinového generátoru“, které se objevují na stránkách nabízejících znásobení odeslané částky v bitcoinech. Data jsou poskytovány v textových souborech. Jeden soubor obsahuje Bitcoinové adresy s transakcemi, druhý bez transakcí a třetí obsahuje adresy jiných kryptoměn. Nevýhodou je, že data obsahují pouze seznamy adres bez dalších metadat jako např. čas jejich získání či konkrétní zdroj. Jedinou informací navíc je seznam všech podvodných domén, nicméně i u tohoto seznamu chybí propojení domény s konkrétními adresami. [13]

CryptoBlacklist je webová stránka, která obsahuje vyhledávač, který pro zadanou kryptoměnovou adresu vypíše, zda je obsažena v databázi existujících nahlášení či nikoliv. Nevýhodou služby je, že obsahuje pouze limitovaný počet posledních nahlášených adres a vyhledávat je možné pouze skrze integrovaný vyhledávač. [14]

CryptoScamDB je open-source web provozovaný společností MyCrypto, která se zaměřuje na správu účtů na síti Ethereum. Služba CryptoScamDB poskytuje API, které vrací JSON, ve kterém jsou všechny adresy, které byly nahlášeny. U každé adresy jsou pak jednotlivé nahlášení, ve kterých se daná adresa objevuje. Nevýhodou tohoto zdroje je, že obsahuje nejen Bitcoinové adresy, ale i adresy dalších kryptoměn, kterých je většina. [15]

SeeKoin se prezentuje jako vyhledávač pro Bitcoin a další kryptoměny. Na webu obsahuje seznam adres spojených s nahlášenými podvody, ve kterém je možné ručně vyhledávat. Nevýhodou zdroje je, že obsahuje relativně špatný stránkový systém, takže je u něj těžké určit, kolik opravdu obsahuje nahlášení. [16]

Cryptscam částečně čerpá z BitcoinAbuse a vypisuje pouze 1 000 posledních nahlášení nejen Bitcoinových adres. Zda byla adresa součástí nahlášení je nicméně možné vyhledat pomocí integrovaného vyhledávače. Nevýhodou tohoto zdroje je omezený počet posledních nahlášení a vyhledávání dostupné pouze skrze integrovaný vyhledávač. [17]

BitcoinWhosWho je zdroj, který poskytuje pouze integrovaný vyhledávač, a tedy není možné zjistit kolik adres obsahuje bez pokusu o vyhledání všech adres vyskytujících se na blockchainu. Zdroj pak obsahuje několik informací o vyhledané adrese, ale nahlášení k dané adrese příliš informací neobsahují. [18]

## 2.3 Shrnutí relevantních zdrojů

Statistiky výše uvedených zdrojů jsou shrnuty v tabulce 2.1, která jednotlivé zdroje seřazuje dle potenciálního počtu Bitcoinových adres. Data k jednotlivým zdrojům jsou platná k 1.11.2022.

Zdroj	Počet adres/nahlášení	Pouze BTC	Přístup
LoyceV	1 043 489 822 unikátních adres	ANO	Soubor (TXT)
BitcoinAbuse	307 641 nahlášení	NE	API (CSV) / Web
CheckBitcoinAddress	88 868 unikátních adres	NE	Web
CryptoBlacklist	37 881 unikátních adres	NE	Vyhledávač
Bitcoin Generator Scam	6 395 unikátních adres	NE	Soubor (TXT)
BitcoinAIS	5 699 nahlášení	NE	Web
CryptoScamDB	4 478 unikátních adres	NE	API (JSON)
Cryptscam	1 000 nahlášení	NE	Vyhledávač
SeeKoin	neurčitý počet adres	ANO	Web
BitcoinWhosWho	neurčitý počet adres	ANO	Vyhledávač

Tabulka 2.1: Zdroje kryptoměnových adres

Jak tedy bylo zjištěno, aktuálně na internetu není mnoho kvalitních zdrojů, a dokonce ani práce zabývající se podvody spojenými s Bitcoinem nezveřejňují vlastní databáze adres. Vytvářená databáze by tedy měla obsahovat, pokud možno všechny Bitcoinové adresy z blockchainu získané z LoyceV a pak dále veškeré informace z dalších zdrojů získané ať už pomocí jejich API či vlastního crawleru. Databáze by pak měla být finálně zveřejněna, aby mohla sloužit k dalšímu využití a snížení aktuálního nedostatku relevantních zdrojů.

## Kapitola 3

# Open Source Intelligence

Open Source Intelligence (OSINT) označuje způsob získávání, analýzy a interpretace informací z veřejně dostupných zdrojů. Umožňuje získat informace o osobách, organizacích a jejich aktivitách, a lze jej využít pro různé účely, jako například k zajištění bezpečnosti, řešení kriminálních případů nebo ke sběru informací v oblasti podnikání. Jedním ze základních principů OSINT je pak ověřování informací z více zdrojů pro zajištění určité úrovně spolehlivosti získaných dat. [19]

V práci je OSINT využíván pro získávání nahlášení podvodů spojených s kryptoměnovými adresami při tvorbě databáze podvodných Bitcoinových adres, které mohou být spojeny s nelegálními aktivitami, jako jsou phishingové útoky, ransomware, obchodování s drogami a jiné podobné trestné činnosti. Součástí získaných informací pak mohou být jména, kontaktní údaje a další relevantní informace spojené s osobami, které za těmito aktivitami stojí, což může pomoci při jejich identifikaci či deanonymizaci při následné analýze s využitím několika dalších nástrojů. Adresy lze následně sdílet s orgány činnými v trestním řízení, aby se zabránilo dalšímu šíření takových aktivit.

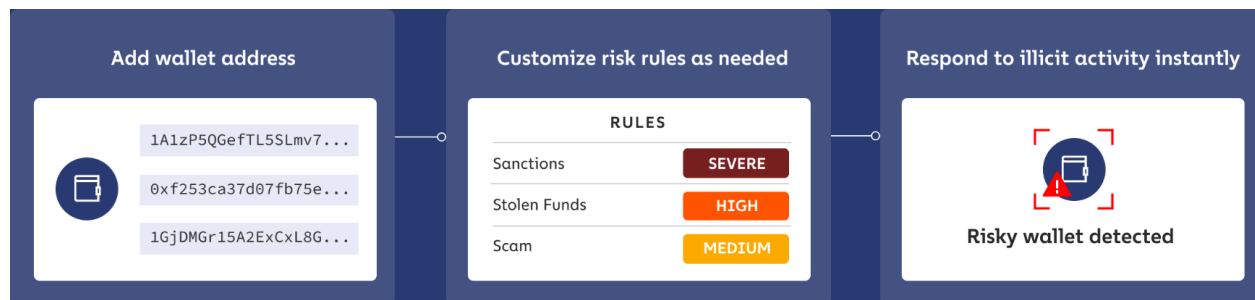
### 3.1 Nástroje pro analýzu kryptoměn

#### 3.1.1 Blockchain.com

Existuje mnoho OSINT nástrojů, které lze použít pro analýzu kryptoměn. Jedním z nich je blockchainový průzkumník `blockchain.com`, který umožňuje procházet transakce na blockchainu a sledovat tok kryptoměn mezi několika adresami. Nástroj lze využít ve spolupráci s vytvořenou databází tak, že se pro vybranou adresu, která byla označena jako podvodná, provede průzkum transakcí pro dohledání adres, kam mířily kryptoměny spojené s nahlášeným podvodem. Pokud se podaří nalézt konečnou adresu, která pak byla použita např. při transakci s nějakým obchodníkem, tak by mohlo být možné dohledat osobu, která je s daným podvodem spojena, nebo takovou osobu alespoň zná. Výhodou nástroje je možnost jeho bezplatného použití. Na druhou stranu však neposkytuje žádnou pokročilejší analýzu a všechny transakce je nutné prohledávat ručně.

### 3.1.2 Chainalysis

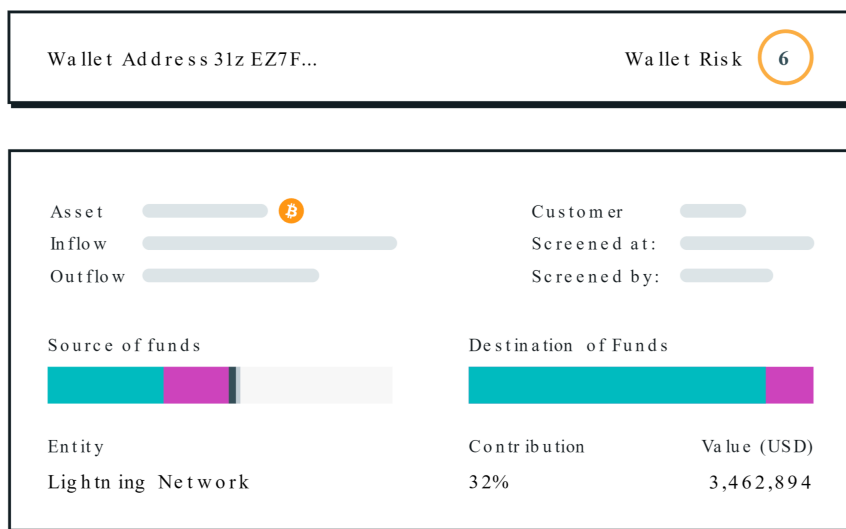
Chainalysis je placený nástroj speciálně zaměřený na analýzu transakcí. Pomocí vlastních technik a algoritmů dokáže sledovat pohyb kryptoměn v síti a získat informace odkud a kam kryptoměny vedly. Umožňuje identifikovat transakce spojené s podvodnými aktivitami, jako jsou např. ransomware útoky či praní špinavých peněz a je schopen identifikovat určité propojení mezi některými adresami. Chainalysis je určený pro profesionální investigativní týmy zabývající se kryptoměnovou kriminalitou či pro finanční instituce, které bojují proti praní špinavých peněz. [20]



Obrázek 3.1: Ukázka nástroje z webu chainalysis.com [20]

### 3.1.3 Elliptic

Dalším nástrojem, který je možné použít při boji s kryptoměnovými podvody, je Elliptic. Poskytuje automatizované řešení, které je schopné v reálném čase monitorovat několik blockchainů a identifikovat použití kryptoměn k praní špinavých peněz či financování terorismu. K identifikaci používá technologie strojového učení a umělé inteligence. Využití najde především u policie, regulátorů a jiných institucí, které se snaží potírat finanční kriminalitu spojenou s kryptoměnami. [21]



Obrázek 3.2: Ukázka nástroje z webu elliptic.co [21]

Ve srovnání s Chainalysis je Elliptic zaměřen více na identifikaci nelegálních aktivit, zatímco Chainalysis se soustředí spíše na sledování toku transakcí. Chainalysis je však o něco rozšířenější, podporuje více kryptoměn a je také levnější, takže je potřeba zvážit, který z nástrojů blíže odpovídá požadavkům uživatele. Oba nástroje by pak mohly těžit z databáze podvodných adres získaných v této práci, jelikož by mohla rozšířit jejich vlastní databáze. Implementované řešení lze použít ve spolupráci s těmito nástroji pro ověření důvěryhodnosti uživatelských nahlášení a pro odstranění případných falešných nahlášení, které se mezi nimi nachází. Tímto způsobem by mohla z již zmíněných nástrojů benefitovat nejen tato práce, ale také všechny zdroje, které implementované řešení prochází.



## Kapitola 4

# Požadavky a návrh systému

Kapitola analýzy a návrhu nejprve představuje požadavky policie na fungování uživatelských účtů. Dále uvádí možné případy užití včetně jednoho konkrétního, který je rozebrán podrobněji. Nechybí pak ani návrh databáze a celkový náhled na jednotlivé komponenty systému. Kapitola se zaměřuje pouze na klíčové části systému.

### 4.1 Požadavky

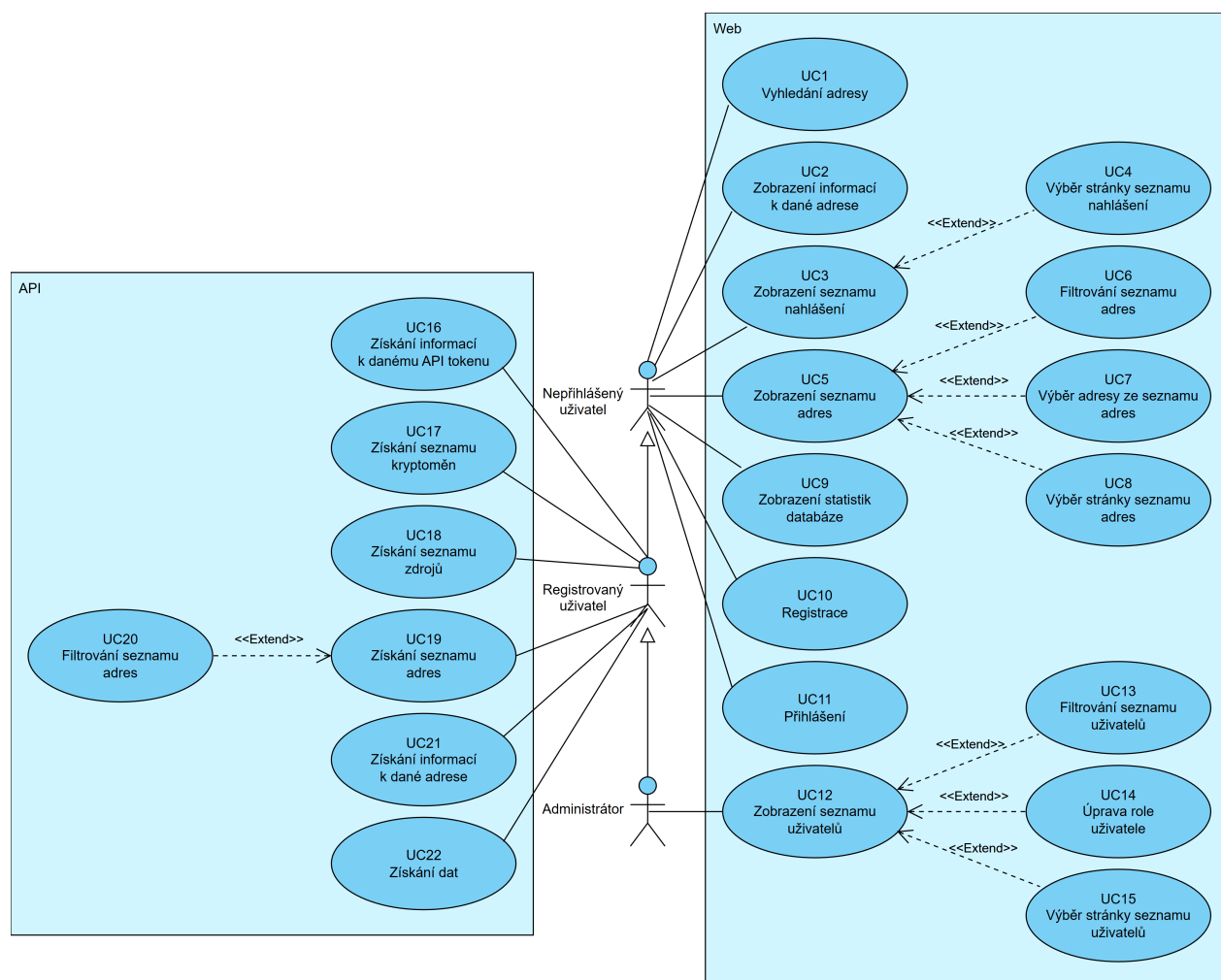
ID	Popis	Typ	Druh požadavku	Případ užití
1	Uživatelské účty z pohledu systému	H	–	–
2	Systém umožňuje zobrazení seznamu uživatelů pouze administrátorům	P	Funkční	–
3	Systém zamezuje přístup k API uživatelům, kteří mají naplněn limit počtu použití	P	Funkční	–
4	Uživatelské účty z pohledu administrátora	H	–	–
5	Systém umožňuje zobrazení seznamu uživatelů	P	Funkční	12
6	Systém umožňuje upravení role uživatele	P	Funkční	14
7	Uživatelské účty z pohledu registrovaného uživatele	H	–	–
8	Systém poskytuje informace k danému API tokenu	P	Funkční	16
9	Uživatelské účty z pohledu nepřihlášeného uživatele	H	–	–
10	Systém umožňuje vyhledání adresy	P	Funkční	1
11	Systém zamezuje přístup k API, jelikož uživatel nevlastní API token	P	Funkční	–

Tabulka 4.1: Specifikace požadavků na uživatelské účty

#### Legenda tabulky požadavků:

**H** – hlavička požadavku, **P** – požadavek

## 4.2 Případy užití



Obrázek 4.1: Diagram případů užití

### 4.2.1 Případy užití webu

Na webu je hlavní rozdíl v tom, zda uživatel vystupuje v roli administrátora či nikoliv. Administrátor má totiž navíc od běžného uživatele přístup k zobrazení seznamu uživatelů a také k úpravě rolí jednotlivých uživatelů. Může tedy změnit roli libovolného uživatele na jednu ze čtyř dostupných variant – host, uživatel, insider a administrátor. Uživatelské role pak mimo jiné ovlivňují především to, k jakým datům se ve finále uživatel na webu či pomocí API dostane, nicméně více o omezení přístupu k datům daným těmito rolemi je uvedeno v sekci 6.4.2.

## 4.2.2 Případy užití API

API poskytuje data pouze registrovaným uživatelům webu, a to jednoduše tak, že k přístupu do API je potřeba mít API token, který musí být součástí odesílaného požadavku. Token se generuje při registraci a určuje, k jakým datům má uživatel přístup, a to na stejném principu, který již byl zmíněn v sekci 4.2.1 a je dále podrobněji rozvinut v sekci 6.4.2. Kromě toho pak token společně s rolí uživatele ovlivňují to, jak často a kolik celkově za určitý čas může být s daným tokenem odesláno požadavků.

## 4.2.3 Vybraný případ užití

V tabulce 4.2 byl pro ukázkou vybrán případ užití s číslem 14, který jako jeden z mála pracuje s uživatelským vstupem. Případ popisuje situaci, ve které se administrátor snaží změnit roli některého z uživatelů.

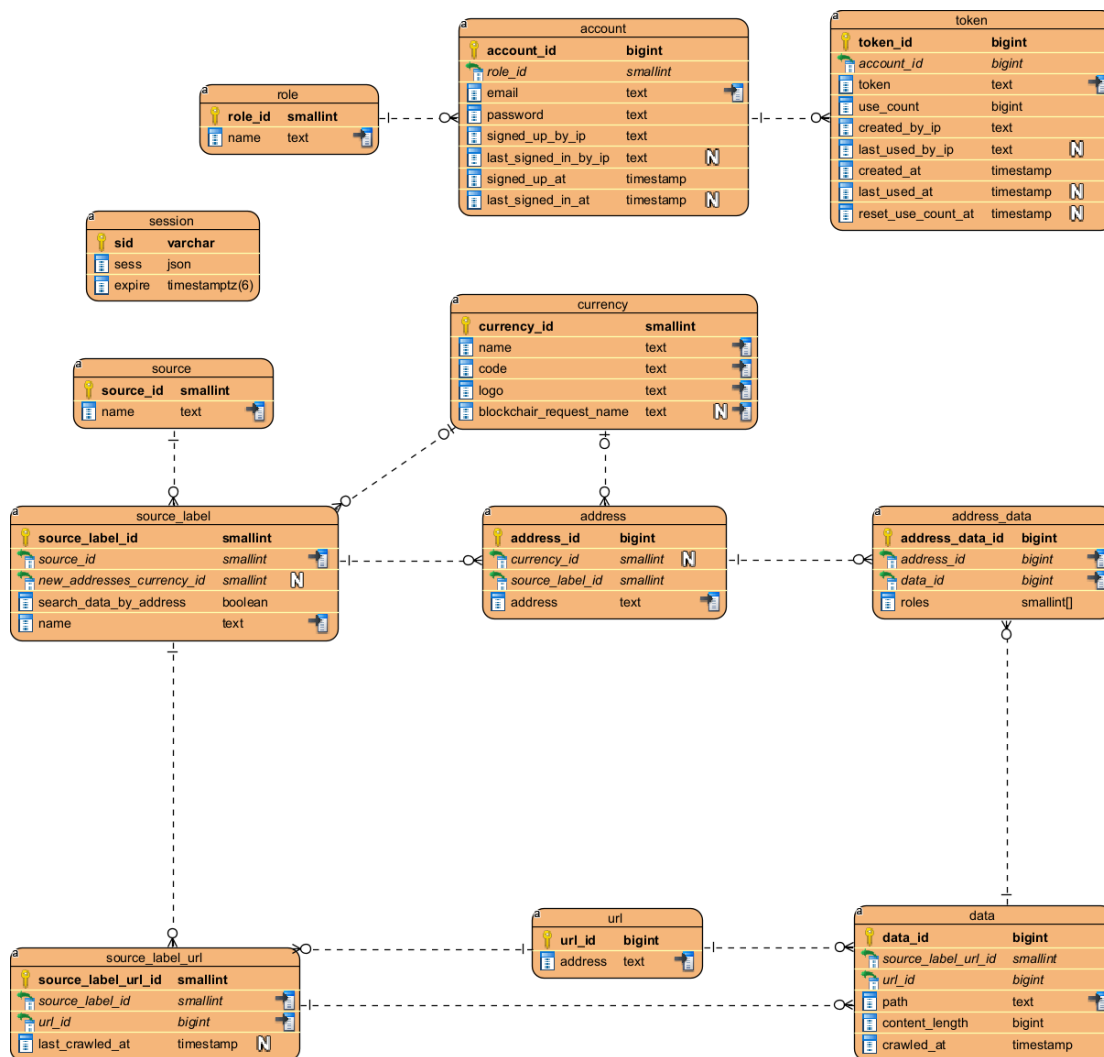
<b>Identifikace</b>	UC14	
<b>Cíl případu užití</b>	Upravit roli uživatele	
<b>Primární aktér</b>	Administrátor	
<b>Vstupní podmínky</b>	Administrátor je přihlášen a klikl na odkaz s textem „Accounts“	
<b>Výstupní podmínky</b>	Role některého z uživatelů je upravena	
<b>Základní scénář</b>	<b>Krok</b>	<b>Akce</b>
	1	Systém ověří, že v systému existuje alespoň jeden uživatel.
	2	Systém vypíše seznam uživatelů.
	3	Aktér u vybraného uživatele změni roli.
	4	Aktér potvrdí změnu stisknutím tlačítka „Edit“.
	5	Systém ověří, že zadané údaje jsou v pořádku.
	6	Systém změni roli daného uživatele.
<b>Rozšiřující scénář 1a: V systému neexistuje žádný uživatel</b>	<b>Krok</b>	<b>Akce</b>
	1a1	Systém zjistí, že v systému neexistuje žádný uživatel.
	1a2	Systém vypíše hlášku: „No account has been found.“.
	1a3	Případ užití končí.
<b>Rozšiřující scénář 5a: Upravovaný uživatel již neexistuje</b>	<b>Krok</b>	<b>Akce</b>
	5a1	Systém zjistí, že upravovaný uživatel již neexistuje.
	5a2	Systém zobrazí hlášku: „Account does not exist.“.
	5a3	Případ užití pokračuje krokem 1.
<b>Rozšiřující scénář 5b: Aktér je posledním administrátorem</b>	<b>Krok</b>	<b>Akce</b>
	5b1	Systém zjistí, že aktér je posledním administrátorem.
	5b2	Případ užití pokračuje krokem 1.

Tabulka 4.2: Případ užití 14: Úprava role uživatele

## 4.3 Návrh databáze

Databáze se skládá z dvanácti tabulek, z nichž nejdůležitějšími tabulkami jsou tabulka **address**, obsahující unikátní kryptoměnové adresy, **data**, která udržuje záznamy o stažených souborech, **url**, která obsahuje všechny procházené URL a **address\_data**, kde jsou propojeny kryptoměnové adresy se staženými daty. Propojení umožňuje při následném vyhledání na webu či v API zobrazit odpovídající data související s danou adresou. Dosud nezmíněné tabulky, které jsou propojené s již zmíněnými tabulkami v diagramu 4.2, jsou tabulky umožňující filtrování seznamu adres dle zdroje či kryptoměny, což je jedna ze základních funkcí nástroje BTCAbuseSearch. Zbylé tabulky souvisí s uživatelskými účty, které řídí přístup k databázi a k API.

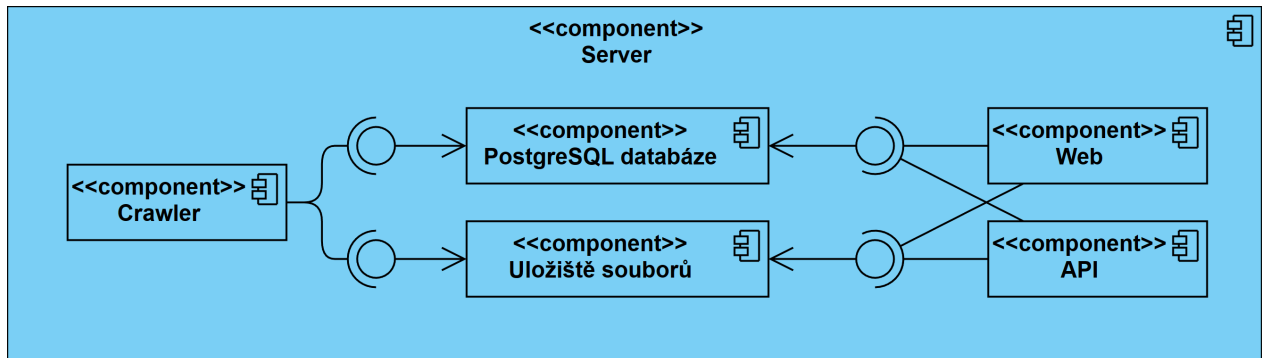
K databázi byl vytvořen také datový slovník, jenž obsahuje detailní popis jednotlivých tabulek, který je možné nalézt v příloze B.



Obrázek 4.2: Diagram vztahu entit databáze

## 4.4 Komponenty systému

Server je postaven na architektuře klient-server. Server se skládá z několika komponent, které se starají o získávání, zpracování a uchování dat, ke kterým má poté klient přístup skrze web či API. Crawler se stará primárně o získávání a zpracování dat týkajících se Bitcoinových adres a data ukládá jak do databáze, tak na disk. Uchování dat v databázi zajišťuje jejich snadné a rychlé procházení, což umožňuje např. vyhledávat nahlášené Bitcoinové adresy či jiné údaje. Ukládání na disk je pak jedním z požadavků policie, která chce mít zálohu získaných souborů z procházených webů pro případ, kdy by některý ze zdrojů změnil svůj obsah či by byl jeho provoz ukončen.



Obrázek 4.3: Diagram komponent systému

## Kapitola 5

# Implementace nástroje BTCAbuseCrawler

Prvním implementovaným nástrojem je BTCAbuseCrawler. Tento nástroj je navržen v programovacím jazyce Python a jeho hlavním cílem je procházení zdrojů dat, ze kterých lze pomocí vhodného nastavení hlavičky a frekvence požadavků získat nahlášení spojená s Bitcoinovými, ale i jinými kryptoměnovými adresami. Vzhledem k tomu, že je potřeba zajistit dlouhodobě funkční běh nástroje, tak je kromě nastavení hlavičky a frekvence požadavků vhodné splňovat pravidla crawlování, které si daný zdroj vymezuje v souboru `robots.txt`. Aby pak data byla co nejdříve uložena do databáze nejen pro zajištění jejich zachování, ale také pro jejich následné použití v BTCAbuseSearch, tak je proces vkládání optimalizován na úrovni SQL příkazů, které doplňují vlákna. Ta umožňují souběžné crawlování více zdrojů současně.

Samotné ukládání dat je pak dále rozepsáno podrobněji a netýká se pouze databáze, která obsahuje důležitá metadata, ale také disku. Data jsou na disk ukládána v podobě souborů ve formátu HTML, TXT či JSON. K rozhodnutí ukládat soubory jako případnou zálohu při změně původního zdroje došlo po konzultaci s policií, která toto uvedla jako jednu z očekávaných funkcí.

Protože jedním z cílů řešení má být vyhledání nalezených dat ke konkrétním kryptoměnovým adresám, tak je potřeba adresy s danými daty propojit.

Aby mohl být nástroj spuštěn dlouhodobě a samostatně bez větších zásahů, tak je potřeba zajistit jeho automatizovaný běh a správu databáze, čemuž se věnují poslední dvě sekce kapitoly.

### 5.1 Procházení zdrojů

Prvním krokem při procházení je načtení seznamu URL ze všech zdrojů, které byly specifikovány v souboru `setup.json` a byly do databáze nahrány při vytváření databáze. Zmíněné URL slouží jako počáteční bod pro danou crawlovací větev a obvykle nejsou jediné v dané větvi, jelikož obsahem URL jsou obvykle seznamy kryptoměnových adres či podvodů s nimi spojených a seznamy také obsahují URL, které je potřeba navštívit.

### 5.1.1 Získání seznamu Bitcoinových adres

Jelikož je práce zaměřena primárně na Bitcoin, tak je na místě u daných adres kontrolovat, zda se jedná o platnou Bitcoinovou adresu či nikoliv. Dalo by se říct, že Bitcoinovou adresu lze poznat jednoduše, jelikož začíná buď „1“, „3“ či „bc1“. Nicméně to neznamená, že libovolná sekvence znaků začínajících těmito znaky by kdy mohla na Bitcoinovém blockchainu existovat. Dokonce z toho nelze poznat ani to, zda na blockchainu již někdy byla obsažena, což je pro dané řešení důležité. Je zřejmé, že ověření Bitcoinové adresy není programově možné a je třeba využít přímo Bitcoinového blockchainu.

Na internetu existuje několik služeb, které zobrazují informace k Bitcoinové adrese, nicméně aby tyto informace mohl program použít, musel by po jednom adresy kontrolovat odesláním požadavků, což je nejen neefektivní, ale také časově náročné vzhledem k velkému počtu adres, se kterým program pracuje. Jako možné řešení byla zvolena služba LoyceV, která byla již zmíněna v sekci 2.2. Služba umožňuje stahovat seznam Bitcoinových adres z blockchainu a data poskytuje na denní a týdenní bázi. Uložení seznamu Bitcoinových adres z blockchainu do databáze pak lze platnost dané Bitcoinové adresy ověřit pouhým porovnáním toho, zda se v databázi nachází, což je řádově rychlejší operace než odesílání požadavku na externí server.

Jelikož má soubor se seznamem Bitcoinových adres minimálně několik desítek gigabajtů, tak je jeho stažení a uložení do databáze relativně zdoluhavý proces (při testování obvykle trvá i několik hodin), tak je potřeba provést určité optimalizace, které v danou chvíli neumožňují ukládání žádných dalších dat, a proto je prvním a v danou chvíli jediným crawlovaným zdrojem pouze LoyceV.

Jelikož je soubor se seznamem adres příliš velký na to, aby mohl být celý nahrán do paměti, tak je při stahování zpracováván postupně po částech. Denní seznamy adres jsou ve formátu TXT, takže u nich je třeba získat jednotlivé řádky, které obsahují samotné adresy. U týdenního seznamu to však neplatí, jelikož je v souboru, který má komprimovaný formát GZ. U stahování tohoto souboru je tedy zpracováván po částech, dekomprimovaný pomocí Python knihovny `isal_zlib` a až následně jsou z textu získávány jednotlivé řádky adres. Oba formáty souborů jsou při stahování rovněž ukládány na disk, jako případná záloha např. kdyby daný zdroj přestal existovat.

### 5.1.2 Procházení zdrojů se seznamy nahlášení

Zdroje se seznamy nahlášení jsou zdroje, které poskytují data přímo na webu bez nutnosti použití vyhledávače. Ve zdrojích není potřeba vyhledávat postupně adresu po adrese pomocí již získaného seznamu adres, takže je jejich crawling výrazně rychlejší. U těchto zdrojů lze snadněji získat přehled o celkovém počtu adres či nahlášení, jelikož mají obvykle očíslované stránky, kde na každé stránce je stejný počet adres / nahlášení a mnohdy také obsahují tlačítko, kde lze vyčíst číslo poslední stránky. Některé weby dokonce zobrazují vlastní statistiku.

### 5.1.3 Procházení zdrojů s vyhledávačem

Pro zdroje, které neobsahují seznam adres či nahlášení, či je takový seznam pouze velmi omezený, je nutné použít vyhledávač, který daný zdroj poskytuje. Aby bylo možné zmíněné zdroje využít, je potřeba mít připravený seznam adres, které se budou vyhledávat, což je také důvod, proč jsou uvedené zdroje procházeny jako poslední. Procházení pomocí vyhledávače je pomalé a na prostředky náročné řešení, jelikož se pro každou adresu vytváří samostatný požadavek. Zároveň je také nutné projít všechny adresy z databáze, jelikož není předem jasné, o kterých adresách zdroj obsahuje nějaké informace. Nicméně i přes všechny zmíněné nevýhody není vhodné se těmito zdroji vyhýbat, jelikož i tak je celkový počet zdrojů relativně nízký a každý takový zdroj může obsahovat důležité informace.

### 5.1.4 Nastavení hlavičky a frekvence požadavků

Správné nastavení hlavičky požadavků umožňuje nástroji splynout s požadavky běžných uživatelů. Čím běžnější je v danou chvíli hlavička daného požadavku, tím menší je riziko na případné selektivní zablokování konkrétní hlavičky. Proto by také měla být hlavička aktualizována minimálně vždy při vydání nové verze nejpoužívanějšího prohlížeče, což je v době psaní prohlížeč Google Chrome ve verzi 111.0.0 s hlavičkou obsahující user-agent: „Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36“. Na základě údajů, které poskytuje web [useragents.me](https://useragents.me) se aktuálně (17. 3. 2023) jedná o nejběžnější user-agent mezi uživateli osobních počítačů s podílem okolo 38 %. V prostředí internetu si servery obvykle kontrolují hlavičky, aby zabránily např. DDoS útokům. Přidání korektní hlavičky a správné chování se nesnaží obejít ochranné kontroly webu, ale spíše prokázat legitimitu požadavků.

Během odesílání požadavků se při vývoji vyskytly určité komplikace, kdy externí server odmítal odpovídat či byl dočasně zahlcen. Toto je při procházení běžný jev způsobený tím, že jsou na server odesílány požadavky ve větší míře, než jakou je server schopen zpracovat. Obecně neexistuje žádné pravidlo, jak dlouho by se mělo mezi jednotlivými požadavky čekat, jelikož to záleží vždy na konkrétním serveru a také velikosti a formátu přenášených dat. Lze však postupným testováním zjistit, při jaké frekvenci nedochází, k již zmíněným problémům a taková frekvence pak může fungovat i dlouhodoběji. Samozřejmě se můžou vyskytnout menší výkyvy, kdy bude mít server vyšší návštěvnost i od jiných uživatelů, takže je dobré sledovat, zda se v programu neobjevují chybové hlášky, které by mohly toto naznačovat. Rychlost procházení je samozřejmě možné kdykoliv změnit v souboru `config.json`.

Při vývoji byla jako adekvátní zvolena rychlost přibližně pět požadavků za sekundu a délka čekání na odpověď každého požadavku na 30 vteřin. Hodnoty nevedly k žádným problémům ani několik hodin od spuštění programu.



### 5.1.5 Respektování robots.txt

Každý server má právo si určit, ke kterým cestám umožní a ke kterým zakáže přístup robotům, a to pomocí souboru `robots.txt` v kořenovém adresáři webu. Server si také může zvolit, jakých konkrétních crawlerů se bude dané pravidlo týkat. Pravidla by pak měla být crawlery dodržována, aby server nezatěžovaly zbytečnými požadavky, které by mohly vést nejen k nestabilitě serveru, ale také k možné blokaci daného crawleru.

Pro tyto účely je v programu využíváno Python knihovny `urllib` a konkrétně její třídy `RobotFileParser`. Tato třída umožňuje zpracovat soubor `robots.txt` a nad jejím objektem lze poté zavolat metodu `can_fetch`, která je pro daný user-agent a URL schopna určit, zda k ní máme či nemáme přístup. Pokud soubor `robots.txt` na webu chybí či je jeho obsah prázdný, tak je daný web procházen bez omezení.

### 5.1.6 Optimalizace pomocí vláken

Vlákna umožňují současně vykonávat více operací na jednou. Jelikož při procházení lze k určitému zdroji přistupovat pouze omezenou rychlostí. Aby zdroj nebyl příliš zahlcen, tak je možné využít čas a prostředky na to, aby se současně komunikovalo s více zdroji na jednou. Právě z toho důvodu jsou využívána vlákna, jelikož umožňují současné odesílání požadavků na více zdrojů, bez nutnosti čekání na odpověď. Nevýhodou může být vyšší zátěž na zařízení, jelikož jsou jeho prostředky využívány mnohem intenzivněji, na druhou stranu však díky efektivnímu využití času trvá průchod všemi zdroji výrazně kratší dobu.

### 5.1.7 Optimalizace ukládání do databáze

Časově náročné je nejen přenášení dat ze zdroje, ale také jejich ukládání do databáze. Během procházení je do databáze postupně ukládáno velké množství dat a tento proces tak trvá velmi dlouho. Například uložení týdenní aktualizace seznamu Bitcoinových adres tak bez jakýchkoliv optimalizací může trvat i více než týden, jelikož průměrný počet vkládaných záznamů za sekundu se na testovaném zařízení pohybovalo přibližně okolo 100. Použití takové aktualizace pak nemá velký význam, jelikož její data jsou v době dokončení již zastaralá.

První změnou, která výrazně navyšuje počet záznamů za sekundu je odebrání primárních, cizích a unikátních klíčů, kontrol hodnot a toho, zda jsou hodnoty nenulové. Vnitřní kontroly pak nejsou potřeba, jelikož je z externího zdroje zajištěno, že jsou adresy v seznamu unikátní a další hodnoty je možné přidat i bez těchto kontrol. Navíc je veškeré přidávání BTC adres prováděno v jedné transakci, takže jsou uvedené kontroly po dokončení ukládání opět obnoveny a po dokončení transakce pak není z tohoto pohledu patrná žádná změna. Touto optimalizací se podařilo zvýšit počet vkládaných záznamů za sekundu přibližně na dvojnásobek.

Dalším způsobem, který byl již částečně zmíněn je zakázání automatických transakcí. Místo nich pak lze vytvářet transakce, které budou dokončeny vždy až po uložení většího množství dat.

Spouštění a ukončování transakce trvá určitý čas, a proto je použití jedné transakce pro jeden vložený řádek velmi neefektivní. Tato optimalizace je nejzásadnější, jelikož díky ní vzrostl počet záznamů za sekundu na více než 200 000.

Poslední změna, která zvýšila u seznamu BTC adres počet záznamů za sekundu až na konečných 250 až 300 tisíc, je změna samotného SQL příkazu. Obvykle se pro ukládání do databáze používá příkaz `INSERT`. V PostgreSQL však existuje ještě příkaz `COPY`, který slouží pro vkládání velkého množství dat. Příkaz má jednu nevýhodu, kterou je to, že při libovolné chybě, která se při vkládání objeví, není možné vkládání dokončit. Žádné chyby se však díky kvalitě zdroje LoyceV neočekávají, a tak je možné příkaz použít.

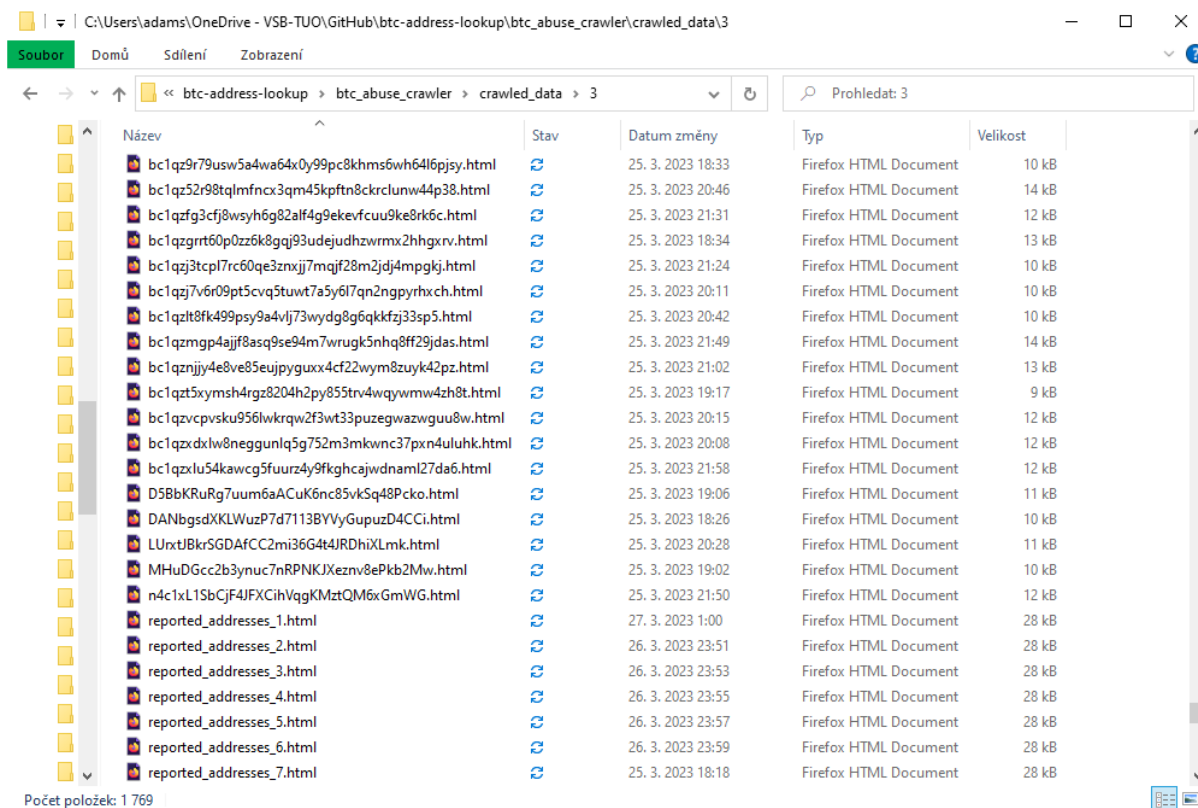
Kromě ukládání seznamu BTC adres se ukládání i jiná data, která již vyžadují určité kontroly, aby např. neobsahovaly duplikátní informace. Z výše zmíněných optimalizací se tedy používá pouze optimalizace transakcí. Není však jediná. I příkaz `INSERT` je možné zefektivnit, a to přidáním až 1 000 řádků v rámci jednoho příkazu `VALUES`. Tento limit není vybrán náhodně. Jedná se o maximální počet řádků, který je v PostgreSQL příkaz `VALUES` schopen pojmout.

### 5.1.8 Ukládání souborů

Při procházení se stahují a ukládají soubory ve formátu HTML, TXT, JSON či GZ. Soubory jsou používány jak na webu, tak v API, a to při vyhledání konkrétní kryptoměnové adresy. Slouží totiž jako zdroj dat o nahlášeních spojených s danou adresou a mohou obsahovat informace jako např. o jaký podvod se jedná, na jaké platformě se objevuje, kdo za ním stojí a také kdy byl nahlášen. Zmíněné informace pak doplňuje také delší popis daného podvodu. Vzhledem k tomu, že ne každý zdroj obsahuje veškeré informace, tak jsou data uchovávána v souborech. Soubory pak také umožňují informace zachovat v takové podobě, v jaké byly staženy, což umožňuje případnou kontrolu toho, zda nebyla data upravena. Zálohování souborů je pak také jedním z požadavků policie, se kterou byla práce konzultována.

Pro uchování souborů je využita složka `crawled_data`, ve které jsou jednotlivé soubory ukládány do podsložek, které mají názvy odvozené od identifikátorů URL podkategorií zdrojů. Dle názvu souboru pak lze jednoduše odvodit, jaký obsah se v něm nachází. Příkladem mohou být názvy – „3QkNpeRH89ZpofYbQ5gvckfBAZQ6V3yShp.html“ (soubor s nahlášeními k dané adrese), „reported\_btc\_addresses\_1.html“ (soubor s první stranou BTC adres, které byly nahlášeny).

Soubory pak dále mohou sloužit jako záloha v případě, kdy by zdroj, ze kterého byly staženy, byl vypnut, ať už dočasně či trvale. Tato možnost rozhodně není pouze hypotetická, jelikož i při vývoji několikrát došlo k tomu, že byl některý ze zdrojů nedostupný.



Obrázek 5.1: Ukázka seznamu souborů ve složce `crawled_data`

## 5.2 Propojování dat s adresami

Po dokončení procházení konkrétní URL je vždy potřeba všechny adresy, které byly obsaženy na dané stránce či v jejím souboru, propojit se souborem, který je na serveru uložen. To pak umožňuje pomocí nástroje BTCAbuseSearch adresy vyhledávat a načítat data s nimi spojená. Propojení páruje identifikátor adresy s identifikátorem dat do dvojice, u které lze také určit, jaké uživatelské role na webu či v API k těmto datům mají přístup. Omezení na straně propojení bylo místo omezení na straně dat zvoleno z toho důvodu, že lze takto vybrat konkrétní adresy, které z nějakého důvodu nechceme určité uživatelské roli zpřístupnit a stejně tak mohou být selektivně vybrána i některá data. Kdyby totiž byly omezení čistě na straně adresy či dat, pak by to velmi omezovalo právě možnosti volby.

## 5.3 Zjištění kryptoměnové příslušnosti altcoinových adres

Po získání seznamu Bitcoinových adres a dokončení procházení zdrojů je na řadě zařazení adresy ke konkrétní kryptoměně. Zmíněný problém se týká pouze ne-Bitcoinových adres, jelikož Bitcoinové adresy jsou v databázi všechny.

Mohlo by se zdát nepodstatné k jaké kryptoměně adresa patří zvláště, když má být řešení zaměřeno primárně na Bitcoin. Jelikož však zdroje obsahují kromě Bitcoinových adres také velké množství altcoinových adres, tak je vzhledem k možnému rozšíření práce vhodné minimálně implementovat určitou formu zjištění. Zmíněný způsob řešení je pak možné případně zefektivnit či přejít na úplně jiný způsob v případě, že by se systém rozšiřoval o práci i s jinými kryptoměnami než jen s Bitcoinem. Dávalo by pak smysl opět získávat seznam adres přímo z příslušných blockchainů konkrétních kryptoměn.

Zjištění kryptoměny probíhá skrze odesílání požadavků na službu Blockchair, která obsahuje data z několika blockchainů současně. Do URL požadavku stačí zadat kryptoměnu a adresu např. takto: „`blockchair.com/KRYPTOMĚNA/address/ADRESA`“ a ze zdrojového kódu stránky lze poté získat informaci o tom, zda se na daném blockchainu adresa nachází či nikoliv.

Ne vždy musí být daná adresa součástí pouze jednoho blockchainu. U Bitcoinu se např. může stát, že daná adresa bude také součástí blockchainu kryptoměny Bitcoin Cash, jelikož její historie vychází ze stejného blockchainu. Tento problém je řešen pořadím zjišťovaných kryptoměn tak, že první kryptoměna, která je u dané adresy nalezena je k ní přiřazena a další zjištění již nenásleduje. U adres nacházejících se současně na obou blockchainech Bitcoinu a Bitcoin Cash to tedy znamená, že budou vždy považovány jako adresy Bitcoinu.

Díky tomuto řešení lze v databázi uchovávat unikátní seznam adres, které jsou zároveň vždy součástí právě jednoho blockchainu. Řešení to není ideální, jelikož by správně měly být k dané adrese přiřazeny všechny její kryptoměny, nicméně toto řešení bylo zvoleno z toho důvodu, že se práce zaměřuje primárně na Bitcoin. Případná podpora dalších kryptoměn je pak implementována pouze navíc a tak, aby nebylo u Bitcoinu přistupováno k žádným ústupkům. Dalším důvodem je pak to, že u dat, která jsou získána procházením zdrojů, není možné jednoduše určit, pro který blockchain jsou určeny. Při propojování dat s adresami by tak byly propojeny pravděpodobně s adresami na více blockchainech, což by bylo velmi nepřesné, jelikož adresa na jiném blockchainu nemusí být spojena s nahlášeným podvodem. Tato situace může sice nastat i nyní, nicméně je větší pravděpodobnost, že se bude nahlášení týkat přímo Bitcoinu, jelikož je tato síť oproti ostatním výrazně větší, a to jak v počtu adres, tak transakcí mezi nimi.

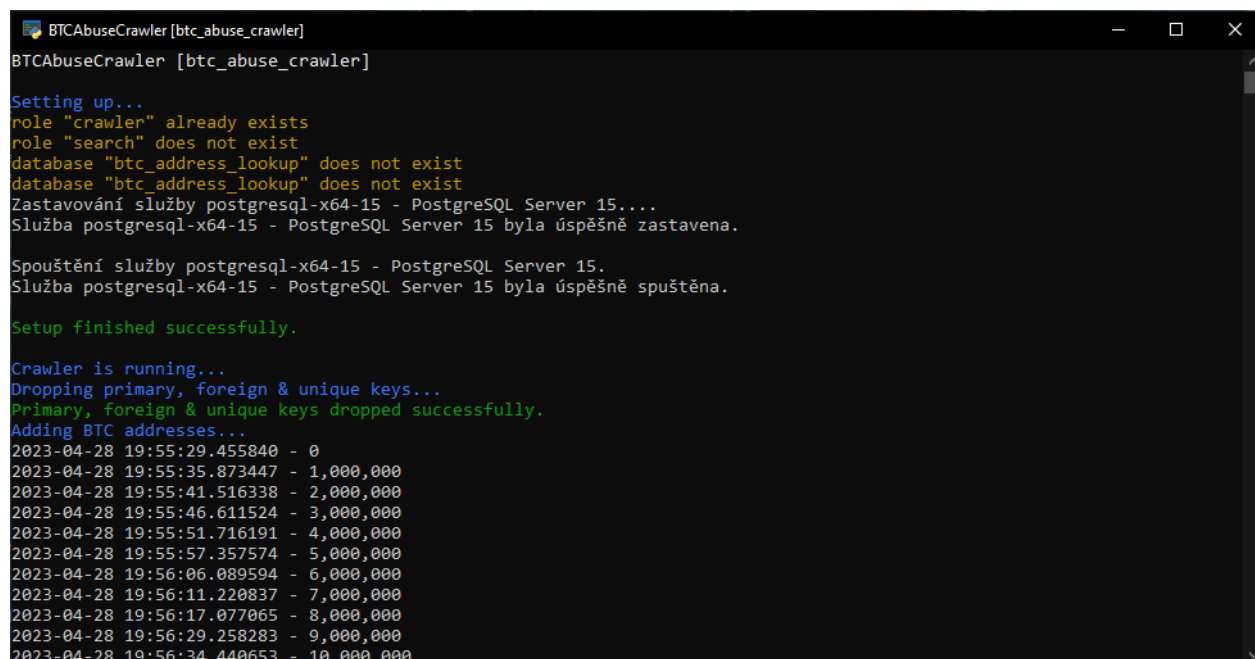
Pokud by bylo cílem práci rozšířit i o další kryptoměny, pak by samozřejmě bylo nutné buď ustoupit od unikátního seznamu adres či povolit u dané adresy přiřazení více kryptoměn současně a vyřešení problému s propojením dat a adres.

## 5.4 Automatizovaný běh

Automatizovaný běh programu zajišťuje jeho nepřetržitý chod bez nutnosti zásahu ze strany uživatele. V hlavní smyčce programu je procházení zabaleno do try-catch bloku. Ať už blok proběhne úspěšně či nikoliv, vždy na jeho konci dojde k čekání na další opakování hlavní smyčky. Doba čekání se odvíjí od nastavení v souboru `config.json`.

Samotný program pak nenabízí téměř žádnou interakci s uživatelem, jelikož jedinou změnu jeho chování je možné ovlivnit zadáním vstupních parametrů – resetování databáze, restartování databáze a smazání instalační konfigurace. Pokud dané parametry zadány nejsou, pak program neresetuje databázi, restartuje ji a nesmaže instalační konfiguraci. Po spuštění pak již uživatel nemá možnost do procesu vstupovat a může jej tak pouze ukončit.

Konzole programu vypisuje hlášky týkající se výjimek a úspěšně dokončených operací. Informuje tím uživatele o aktuálním stavu crawleru v čase, což může pomoci při odhalování potenciálních chyb.



```
BTCAbuseCrawler [btc_abuse_crawler]
BTCAbuseCrawler [btc_abuse_crawler]

Setting up...
role "crawler" already exists
role "search" does not exist
database "btc_address_lookup" does not exist
database "btc_address_lookup" does not exist
Zastavování služby postgresql-x64-15 - PostgreSQL Server 15...
Služba postgresql-x64-15 - PostgreSQL Server 15 byla úspěšně zastavena.

Spouštění služby postgresql-x64-15 - PostgreSQL Server 15.
Služba postgresql-x64-15 - PostgreSQL Server 15 byla úspěšně spuštěna.

Setup finished successfully.

Crawler is running...
Dropping primary, foreign & unique keys...
Primary, foreign & unique keys dropped successfully.
Adding BTC addresses...
2023-04-28 19:55:29.455840 - 0
2023-04-28 19:55:35.873447 - 1,000,000
2023-04-28 19:55:41.516338 - 2,000,000
2023-04-28 19:55:46.611524 - 3,000,000
2023-04-28 19:55:51.716191 - 4,000,000
2023-04-28 19:55:57.357574 - 5,000,000
2023-04-28 19:56:06.089594 - 6,000,000
2023-04-28 19:56:11.220837 - 7,000,000
2023-04-28 19:56:17.077065 - 8,000,000
2023-04-28 19:56:29.258283 - 9,000,000
2023-04-28 19:56:34.440653 - 10,000,000
```

Obrázek 5.2: Ukázka konzole crawleru

Program by mohl být dále rozšířen o uživatelský vstup, který by např. vybíral, které zdroje budou procházeny či dočasně vypnul zjišťování kryptoměn adres. Dále by pak mohl být vypisován průběžný postup procházení či dostupnost nových dat u jednotlivých zdrojů.

## 5.5 Správa databáze

Jednou ze součástí crawlovacího nástroje je správa databáze. Díky ní je možné s nástrojem pracovat okamžitě po instalaci potřebných doplňkových programů zmíněných v příloze A. Cílem implementace správy databáze je zjednodušit proces vytváření či resetování databáze, její naplňování určitými úvodními daty, nastavování optimalizačních parametrů či restartování jejího procesu.

### 5.5.1 Vytvoření a resetování databáze

Jednou z prvních věcí, kterou crawler provádí po jeho spuštění, je příprava databáze. Jelikož je v rámci životního cyklu řešení crawler prvním nástrojem, který s databází pracuje. Je tedy příhodné do něj přidat zavedení či resetování databáze, aby tento proces nemusel být zajišťován manuálně. Minimálně při vývoji je totiž obnovování a opětovné nastavování databáze velmi časté a vzhledem k tomu, že je proces pokaždé stejný a také časově náročný, tak je vhodné jej automatizovat. Zmíněná volba velmi urychlila počáteční vývoj, a i nyní lze díky ní rychle provádět změny.

Databáze se vytváří pomocí souboru `setup.py`, který načítá nastavení ze souboru `setup.json`. Ve zmíněném souboru je možné zadat přístupové údaje k databázi, seznam uživatelů s přístupem k databázi a seznam všech tabulek, které do ní mají být přidány. U tabulek je pak možné specifikovat konkrétní parametry sloupců a zadat výchozí řádky jako např. předem známý seznam podporovaných kryptoměn. Během vytváření databáze se pak aplikují určité optimalizační parametry, které jsou více rozvedeny v sekci 5.5.2.

Vzhledem k tomu, že s některými parametry souboru je počítáno např. při crawlování, tak je potřeba úpravy provádět opatrně. Změna pořadí či názvu některých nastavení by mohla ovlivnit stabilitu daného řešení a vyžadovala by změnu i na straně zdrojového kódu.

Resetování databáze uvádí databázi do původního stavu. Proces odebírá veškerá data, tabulky, a dokonce i uživatele databáze, aby bylo možné otestovat její opětovné zavedení. Součástí resetování je i odebrání dříve crawlovaných souborů a resetování optimalizačních parametrů.

### 5.5.2 Optimalizační parametry

Vzhledem k tomu s jak velkým počtem dat crawler pracuje, tak je potřeba využít veškerých možností optimalizace s cílem snížit čas potřebný pro nahrání dat, aby bylo zajištěno jejich zachování. PostgreSQL server umožňuje nastavit několik parametrů, které ovlivňují jeho výkon.

- `synchronous_commit=off` – Asynchronní commit zajišťuje zvýšení výkonu bez rizika poškození dat, které je možné např. u `fsync=off`. Jedná se tedy o kompromis mezi výkonem a bezpečností dat.
- `bgwriter_lru_maxpages=0` – Maximální počet bufferů, které proces zapíše během každé iterační. Zakázáním zapisování se sníží vstupně-výstupní zátěž.
- `log_checkpoints=off` – Zakazuje zaznamenávání logů o záchytných bodech.
- `min_wal_size=4096` – Definuje minimální velikost WAL (transakční logování) na 4 GB.
- `max_wal_size=16384` – Definuje maximální velikost WAL na 16 GB.
- `wal_level=minimal` – Nastaví minimální úroveň informací zaznamenávaných v transakčním logu, což výrazně zrychluje vykonávání databázových operací.

- `max_wal_senders=0` – Zakazuje replikační streamování.

Následující parametry obsahují hodnoty vypočítané pro RAM o velikosti 16 GB. Crawler je schopen velikost RAM získat a hodnoty procentuálně vynásobit, takže není třeba je manuálně nastavovat.

- `work_mem=2097151` – Nastavuje pracovní paměť na 4 GB.
- `maintenance_work_mem=1048576` – Nastavuje pracovní paměť pro údržbu na 1 GB.
- `shared_buffers=524288` – Nastavuje paměť sdíleného bufferu na 4 GB.
- `temp_buffers=131072` – Nastavuje paměť dočasného bufferu na 1 GB.

### 5.5.3 Restartování databázového serveru

Restartováním databázového serveru se potvrdí změny provedené aplikováním optimalizačních parametrů zmíněných výše. Ve Windows PostgreSQL server vystupuje v podobě procesu, který může restartovat pouze správce, což je také důvod, proč je nutné při spuštění nástroje potvrdit spuštění v režimu správce. To, zda se má server restartovat je možné změnit vstupním parametrem programu, nicméně bez této změny se ve výchozím stavu databázový server restartuje po vytvoření či resetování databáze. Pro restartování se v Pythonu používá knihovna `ctypes`, která obsahuje přístup k Windows API a jejímu rozhraní Shell (`shell32`).

## Kapitola 6

# Implementace nástroje BTCAbuseSearch

Webový nástroj BTCAbuseSearch poskytuje přístup k databázi s údaji o kryptoměnových adresách. Nástroj je navržen v programovacím jazyce JavaScript a využívá běhové prostředí Node.js. Jako vstupní vrátka k informacím slouží webové stránky a API. Zatímco webové stránky jsou určené pro snadné, přehledné a uživatelsky přívětivé použití, tak API cílí spíše na zapojení do aplikací třetích stran.

V rámci implementace webových stránek je nejprve popsán proces vyhledávání kryptoměnové adresy. V tomto procesu dochází k načítání a zpracování dat souvisejících s vyhledanou adresou a jejich následnému zobrazení na webu. V datech se nachází informace o nahlášených podvodech, tj. název podvodu, platforma, datum a čas nahlášení a mnohdy také název jedince či skupiny, která za daným podvodem stojí. Ne vždy se však musí jednat o pravdivé informace, jelikož zdroje, ze kterých je čerpáno umožňují podat nahlášení komukoliv, zdarma a zřejmě také bez jakékoliv moderace. Veškeré informace je tedy nutné brát s rezervou.

Dále je zmíněno zobrazení seznamu všech kryptoměnových adres v databázi, k čemuž nástroj využívá samostatnou stránku. Tato stránka umožňuje adresy filtrovat dle vybrané kryptoměny a také zdroje, ke kterému se váže alespoň jedno její nahlášení. Jednotlivé adresy v seznamu pak obsahují odkaz na stránku s jejími informacemi stejně, jako by byla daná adresa vyhledána.

Součástí webových stránek jsou dále statistiky obsahující počty adres u jednotlivých zdrojů, které jsou navíc rozděleny dle příslušných kryptoměn. Vzhledem k časové náročnosti získání statistik z databáze se dané údaje aktualizují vždy jednou za určitý časový interval a po vybranou dobu jsou načítány pouze z mezipaměti. Zmíněnou optimalizací se šetří nejen prostředky serveru, ale také čas uživatele, jelikož u tohoto typu aplikace není nutné v danou chvíli poskytovat skutečný počet adres v reálném čase.

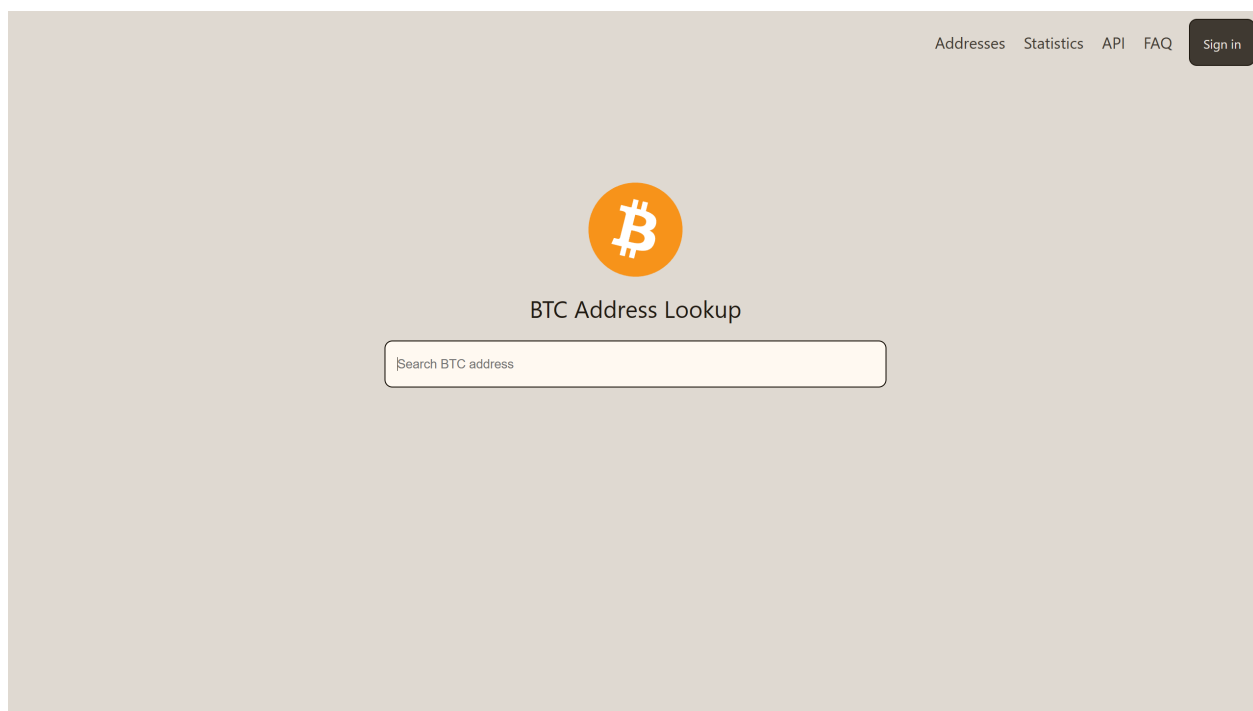
Z pohledu uživatele je poté zásadní zmínit zapojení uživatelských účtů a proces jejich registrace a přihlášení, jelikož každý uživatelský účet vystupuje v nějaké uživatelské roli, což ovlivňuje, k jakým datům má přístup. Účet pak navíc uživateli otevírá možnost k využití API. Při registraci se totiž generuje API token, který je pro využití API vyžadován.



Poslední zmíněnou částí webového nástroje je API. API poskytuje přístup k datům v databázi bez nutnosti použití již zmíněných webových stránek. Data jsou v rámci API poskytována ve formátu JSON a použití API je vhodné pro zapojení do aplikace třetí strany. Pro omezení přístupu k datům pro jednotlivé uživatele a zabránění potenciálnímu zneužití je pro použití API vyžadován token, který zaznamenává každý požadavek. Při dosažení limitu pro daný uživatelský účet je pak přístup k API pro daný token dočasně zablokován.

## 6.1 Vyhledávání kryptoměnové adresy

Nejdůležitější funkcí řešení je vyhledávání kryptoměnové adresy. Adresu lze na webu vyhledat jejím zadáním do vyhledávacího pole uprostřed (na úvodní stránce) či na vrchu obrazovky. Po potvrzení stisknutím klávesy [Enter] či odesláním formuláře se daný požadavek zpracuje a web se přesměruje na příslušnou URL. Na zmíněné URL se poté nachází údaje k samotné adrese.



Obrázek 6.1: Úvodní stránka webu

Cílem vyhledávání je nejprve načíst informace z databáze a z disku, zpracovat je a poté poskytnout odpovídající výstup. V případě, že byla adresa nahlášena na jednom z používaných zdrojů, jejichž seznam je uveden v tabulce 2.1, a již byla tato informace získána crawlerem, tak se dané nahlášení zobrazí na webu. Lze tak jednoduše zjistit, zda daná adresa již byla na některém ze zdrojů nahlášena, a to bez nutnosti jejich navštívení. Pro snadné a rychlé ověření existence nahlášení je u každého z nich uveden příslušný odkaz, ze kterého bylo dané nahlášení staženo.

Výstup může dále obsahovat odkazy na blockchainové průzkumníky, a to za určitých pravidel, o nichž pojednává sekce 6.1.3. Blockchainový průzkumník obsahuje informace jako seznam transakcí, jejich vstupy, výstupy a také čas a částku, která byla přijata či odeslána. Informace pak umožňují manuální kontrolu toho, zda by mohlo být některé z nahlášení oprávněné. Např. pokud někdo nahlásil danou adresu s tím, že na ní poslal 1 BTC a transakce o této částce je v seznamu transakcí obsažena, pak lze předpokládat, že dané nahlášení může být odůvodněné.

### 6.1.1 Načítání dat

Data o vybrané adrese jsou načítána z databáze a z disku. V databázi je uložena informace o příslušnosti adresy k některé z kryptoměn, datum získání posledních dat k dané adrese a seznam identifikátorů dat, které s danou adresou souvisí. Pomocí identifikátorů jsou poté z databáze získány URL řetězce, ze kterých byly soubory staženy a také cesty k daným souborům na disku. Právě díky těmto cestám je možné následně načítat soubory a po jejich zpracování zobrazit odpovídající informace na webu.

### 6.1.2 Zpracování dat

Jak již bylo zmíněno v předchozí sekci 6.1.1, data jsou načítána ze souborů, které obsahují informace o dané adrese a byly staženy z některého z vybraných zdrojů. Mohlo by se zdát jako nejlepší řešení soubory načítat a zpracovávat již v crawleru a na webu pouze získávat odpovídající informace. Toto řešení je sice vhodnější, pokud je očekávána vyšší návštěvnost webu, jelikož není potřeba data zpracovávat pro každý jeden webový požadavek, nicméně také vyžaduje vyšší kapacitu disku, jelikož v podstatě dochází k duplikaci již lokálně dostupných dat. Od této možnosti bylo tedy nakonec upuštěno, vzhledem k omezeným prostředkům při vývoji. Mimo jiné již uložení všech BTC adres z blockchainu zabírá několik desítek GB.

U HTML souborů je zpracování založeno na vyhledání specifických elementů pomocí předem navržených CSS selektorů. Selektory umožňují vybrat důležité elementy obsahující potřebné informace. Některé textové řetězce mohou obsahovat mezery na začátku či konci, což je ošetřeno využitím funkce `trim()`, která mezery odebírá. U nahlášení se pak pracuje datem jeho odeslání, což je při zpracování problém, jelikož téměř každý zdroj používá jiný formát. Z toho důvodu byla vytvořena funkce `parseDateTime()`, která umí pracovat s daty obsahujícími pomlčky, tečky, čárky, textovou formou měsíců a také čas. Po zpracování jsou tedy všechny formáty zpracovány do jednoho objektu třídy `Date`.

Zpracování souboru ve formátu JSON je o něco jednodušší jednak díky tomu, že jej používá pouze jeden zdroj a také díky tomu, že jsou data v něm uložena způsobileji. Z již zmíněných funkcí je tedy využívána pouze funkce `trim()`.

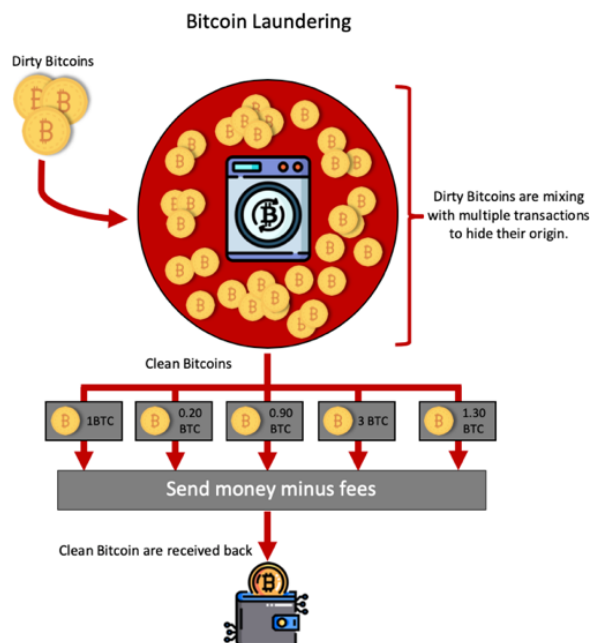
### 6.1.3 Informace k dané adrese

Po vyhledání kryptoměnové adresy nebo jejím vybrání v seznamu adres, po načtení a zpracování dat se na webu objeví informace spojené s touto adresou. Nejprve je zde pomocí obrázku znázorněna její příslušnost k určité kryptoměně. V případě, že se jedná o jednu z podporovaných kryptoměn na webu `blockchair.com` jako Bitcoin, Ethereum a další, pak jsou zde přítomny také odkazy na dva blockchainové průzkumníky, na nichž se daná adresa nachází.

Prvním průzkumníkem je vždy `blockchair.com` a dalším ten, který podporuje danou kryptoměnu. Ne každý průzkumník totiž obsahuje více než jednu či dvě kryptoměny. Zároveň druhý průzkumník může fungovat jako alternativa v případě, kdy první zmíněná možnost není z nějakého důvodu dostupná. Pomocí blockchainového průzkumníka lze prohlížet vstupy, výstupy, částky a také čas všech transakcí, a to u všech kryptoměn, které uvedené informace poskytují veřejně. Tímto případem je také Bitcoin.

Podvodníci se obvykle snaží ztížit analýzu transakcí a dohledání cílové adresy. Využívají k tomu kryptoměny, které neposkytují veřejnou historii transakcí. Příkladem takové kryptoměny je Monero. Monero poskytuje přístup k historii transakcí pouze s pomocí privátního klíče, takže bez něj není možné transakce procházet.

Častěji se ovšem k podvodům využívá Bitcoin a u něj se pro ztížení analýzy používá kryptoměnových mixérů. Mixér funguje tak, že podvodník pošle z dané adresy určitou částku, a to stejné udělají i ostatní aktéři v dané síti. Poté mixér provede spoustu transakcí na různé adresy a na cílovou adresu podvodníka dorazí o něco menší množství kryptoměny dle výše transakčních poplatků v dané síti.



Obrázek 6.2: Příklad kryptoměnového mixéru [22]

Právě pro identifikaci mixérů je možné využít průzkumníky. Hodí se však také pro ověření legitimacy jednotlivých nahlášení. Jedním z problémů s těmito nahlášeními je, že sice mohou pomoci při odhalení podvodu, nicméně stránky, na které jsou nahlášení nahrávány jejich pravost neověřují a odeslat je může prakticky kdokoli. Pokud je však v nahlášení obsažena určitá částka, kterou daná oběť odeslala, či je uveden čas, kdy k dané transakci došlo, pak je možné zmíněné údaje v průzkumníkovi ověřit. Zahrnutí jednoho či více odkazů na průzkumníky také není mezi weby zabývajícími se nahlášeními nic nového a používá je např. služba `bitcoinabuse.com`.

Všechny dosud zmíněné informace k dané adrese se týkají všech adres v databázi, tedy i těch, které nejsou součástí žádného nahlášení. Web a také API lze tedy mimo jiné využít také k rychlé kontrole toho, zda je vyhledaná adresa obsažena na Bitcoinovém blockchainu.

Nejdůležitější součástí stránky s informacemi k dané adrese je seznam nahlášení. Web poskytuje již zpracovaná data z databáze v podobě několika informačních položek ke každému nahlášení. Všechny možné informační položky jsou uvedené v tabulce 6.1. Nahlášení jsou poté pro lepší orientaci rozdělena do jednotlivých stránek po 20 nahlášeních.

Jednou z žádaných funkcí řešení je možnost identifikace a deanonymizace vlastníků Bitcoinových adres. Případné možnosti v této oblasti jsou blíže představeny v sekci 6.1.4, která navíc s pomocí několika snímků z webu představuje možné řešení s využitím tohoto nástroje. To, zda je jakékoliv nahlášení z webu k těmto účelům vůbec možné použít a jak lze ověřit důvěryhodnost nahlášení přibližuje sekce 6.1.5.

Typ	Popis
Datum	Datum a čas, kdy bylo dané nahlášení publikováno na použitém zdrojovém webu.
Typ	O jaký typ podvodu se jedná. Může být vybrán ze seznamu webem předem připravených možností či zadán ručně.
Platforma	Na jaké platformě se daný podvod objevil.
Země	Z jaké země nahlášení pochází.
URL	URL adresa na detail adresy na zdrojovém webu, kde je možné ověřit, že takové nahlášení skutečně existuje.
Podvodník	Jaký subjekt za daným podvodem stojí.
Popis	Delší popis nahlášení. Může obsahovat detailní informace o daném podvodu.
Chybová hláška	Objevuje se v případě, kdy adresa v databázi sice existuje, ale není spojena s žádným nahlášením.

Tabulka 6.1: Typy informací obsažené v nahlášení kryptoměnové adresy

#### 6.1.4 Deanonymizace vlastníka Bitcoinové adresy

Bitcoinová síť je pseudonymní, což znamená, že žádná Bitcoinová adresa není přímo spojena s jejím vlastníkem. Vlastníkem Bitcoinové adresy je subjekt, který má přístup k jejímu privátnímu klíči, který mu umožňuje podepisovat transakce, a tedy odesílat určité množství bitcoinů na libovolnou adresu. To však vede k několika možným scénářům:

- Existuje více subjektů a každý zná celý privátní klíč. Je tedy **více vlastníků adresy** a každý z nich může provádět transakce.
- Existuje pouze jeden subjekt, který zná celý privátní klíč a je tedy **jediným vlastníkem adresy**.
- Existuje více subjektů, kdy každý zná pouze část privátního klíče, ale ani jeden nezná celý privátní klíč. Je třeba spojit všechny části privátního klíče, aby bylo možné autorizovat transakce, a tedy **nikdo de facto není vlastníkem**.
- Neexistuje ani jeden subjekt, který by znal celý privátní klíč a nelze ani spojit několik částí klíče od jednotlivých subjektů. Odhaduje se, že přibližně 20 % všech bitcoinů je obsaženo na adresách k nimž již **nikdo nemá přístup**. [23]

Jak je z výše uvedených možností patrné, vlastnictví Bitcoinové adresy nelze jednoznačně určit. Dalo by se namítnout, že lze vlastníka adresy odhalit právě tím, že v určitý moment provede transakci, a tedy zaručeně v daný moment je jejím vlastníkem. Toto tvrzení je pravdivé pouze za předpokladu, kdy máme stoprocentní jistotu, že tuto transakci skutečně provedla daná osoba např. pomocí hardwarové peněženky. Problémem tohoto ověření je pak také fakt, že daný subjekt může po provedení transakce privátní klíč ztratit či se jej zbavit, čímž daná adresa přijde o jejího vlastníka.

Nelze ani očekávat, že by privátní klíč daný subjekt zveřejnil, jelikož by tím umožnil přístup k dané adrese komukoliv, kdo by pomocí něj mohl veškeré bitcoiny poslat na vlastní adresu.

Při opuštění vlastníků privátních klíčů se dá na danou problematiku podívat ještě z jednoho pohledu. Pro příklad si lze představit následující situaci.

Podvodník si založí účet např. na Twitteru či jiné sociální síti a na zmíněné síti zveřejní svou adresu s cílem vylákat od obětí kryptoměny. Aby daný podvod byl alespoň trochu uvěřitelný, tak k adrese přiloží ještě zprávu o tom, že po odeslání kryptoměny se uživateli vrátí její dvojnásobek. Prohledáme v tomto případě Twitter, nalezneme Bitcoinovou adresu v příspěvku od podvodníka a známe vlastníka. Takto jednoduché to však není. Existuje zde řada problémů:


- Jak zjistíme, že ten, kdo daný příspěvek odeslal skutečně vlastní její privátní klíč?
- Jak poznáme, že skutečný vlastník privátního klíče nepoužil bílého koně, tedy jiný subjekt, který příspěvek publikoval na sociální síti za něj?
- Patří uživatelský účet reálné osobě nebo se jedná o účet vytvořený robotem?
- Jak odlišíme propagaci podvodu podvodníkem od upozornění na podvod policií?

Z výše uvedených problémů tedy vyplývá, že je víceméně irelevantní, kdo danou adresu zveřejňuje, jelikož není možné určit, zda se skutečně jedná o vlastníka dané adresy.

Prohledávání sociálních sítí s cílem propojit určité účty s Bitcoinovými adresami tedy může pomoci identifikovat další místa na internetu, kde se daná adresa vyskytuje a v omezeném případě dokonce také může vést k identifikaci vlastníka adresy. Využití takovéto funkce je nicméně omezeno na specifické případy, kdy osoba na sociální síti vystupuje pod skutečným jménem a následně je možné vlastnictví ověřit tak, že daná osoba prokazatelně odešle transakci z dané adresy. Jak již bylo zmíněno tato kontrola je velmi individuální a bez ní k deanonymizaci nemůže dojít.

Od deanonymizace ovšem nebylo v práci upuštěno úplně. Díky tomu, že práce pracuje s uživatelskými nahlášeními adres, tak je možné občas najít nahlášení, u nichž je nějaká osobní informace obsažena. Příklady takovýchto nahlášení je možné vidět na obrázcích 6.4 a 6.3. Využitelnost takovéto informace potom záleží na důvěryhodnosti daného nahlášení a rozhodně nelze tvrdit, že by jakákoliv informace musela být pravdivá. Určitým způsobem však lze alespoň částečně ověřit pravdivost určitých informací, o čemž blíže pojednává následující sekce 6.1.5.

## Address

**Address**  
 17WqsBwFEUij1Tgu2RMiL1wFRq6kRhGUNq

**Blockchain explorers**  
[View on blockchain.com](#)  
[View on blockchair.com](#)

**Last update**  
25. 3. 2023 18:22:01

Date	Type	Country	URL
10. 1. 2023 22:53:48	Drug dealer scammer	United States	<a href="#">View on checkbitcoinaddress.com</a>

**Abuser**  
Homamdan


**Description**  
User is on telegram by the name of homamdan .

Obrázek 6.3: Příklad nahlášení obsahujícího přezdívku

Vyvíjené řešení tedy nepostrádá deanonymizační prvky úplně, jelikož lze narazit na nahlášení obsahující osobní informace, ale jejich využití je více dedikováno na samotného uživatele a jeho následnou analýzu.

# Address

## Address

 12qusMrNaac75kHFWMKPQuuSn7L44xozvV

## Blockchain explorers

[View on blockchain.com](#)

[View on blockchair.com](#)

## Last update

25. 3. 2023 18:10:35

Date	Type	Country	URL
3. 3. 2023 19:22:59	ransomware	United States	<a href="#">View on cryptscam.com</a>

## Abuser

Recover

## Description

Good work deserves recommendation... i lost over 2.3 BTC on Instagram bitcoin scam.. Right about 2 weeks after my ordeal with them I tried using the recommendation from someone on one of the comment section (www.ukmcybersecurity.com) I was able to get all my money back in less than 48 hours.. Contact (www.ukmcybersecurity.com) to recover all your stolen bitcoins free of charge

Date	Type	Country	URL
3. 3. 2023 18:57:07	other	United States	<a href="#">View on cryptscam.com</a>

## Abuser

Michael Wisard

## Description

This man was going to help me get back some money from scam, he promised to pay me back toda yesterday and the day before that, but he started to ask for more and more fees, I even had to ask my sister to help me and she paid 800 euro in total. He has used 2 walletaddresses wich I sent to. From the date 22 februari-23. After the last sending this morning and he was supposed to send the bitcoin, he called and told me that I had to pay 550 euro more fir autorisationfee whatever that is. I asked him to pay everything back but he say he can because the funds are locked in the wallet. I want him to send back everything to me. But he refuse. Can you assisist on this. I guess this upfront fees are scams.

Obrázek 6.4: Příklad nahlášení obsahujícího jméno

### 6.1.5 Důvěryhodnost nahlášení

Nedílnou součástí stránky s informacemi k vyhledané adrese jsou uživatelská nahlášení. Nahlášení jsou obsažena na zdrojových webech a stažena pomocí crawleru. Zdrojové weby umožňují téměř komukoliv nahlásit podvod spojený s kryptoměnami a obvykle umožňují kromě Bitcoinových adres nahlásit i adresy jiných kryptoměn. Výhodou těchto služeb je, že pomocí jednoduchého formuláře umožňují rychle upozornit na danou hrozbu. Na druhou stranu však nahlášení nejsou na webech žádným způsobem moderována, a tedy mohou obsahovat zavádějící informace. Motivů k odeslání zbytečného, nepravdivého či škodlivého nahlášení může být více. Podvodníci se např. mohou snažit zahltit služby falešnými zprávami, aby mezi nimi nebylo snadné nalézt nahlášení jejich vlastního podvodu. V horším případě pak mohou zneužít nahlášení jiného podvodu s cílem propagovat podvod vlastní a teoreticky tak zneužít danou platformu k opačnému účelu, než k jakému byla zamýšlena. Zahlcením lze také cílit na možnou demotivaci uživatele danou službu použít.

Nelze však říci, že by na daných platformách neexistovaly žádné smysluplné nahlášení. Z pohledu oběti může být často náročné správně identifikovat situaci a všechny okolnosti podvodu, a podvod tedy nemusí být vždy správně a dostatečně podrobně popsán. Nicméně jakékoliv nahlášení, které u dané adresy je, může vést ke zvýšené obezřetnosti uživatele.

Detailně popsané nahlášení obsahující místo, průběh a jiné další okolnosti podvodu mohou uživatelům pomoci se takovému podvodu vyhnout. Pokud pak navíc nahlášení obsahuje jakékoliv informace o podvodníkovi, ať je to jeho jméno, přezdívka, telefon, email, účet na sociální síti či jazyk použitý při komunikaci, tak tyto informace mohou být použity při pátrání po dané osobě.

## 6.2 Seznam adres

Další důležitou funkcí webu je zobrazení seznamu nahlášených adres. Seznam obsahuje všechny kryptoměnové adresy, ke kterým má aktuální uživatel přístup a které jsou v databázi spojené alespoň s jedním staženým souborem.

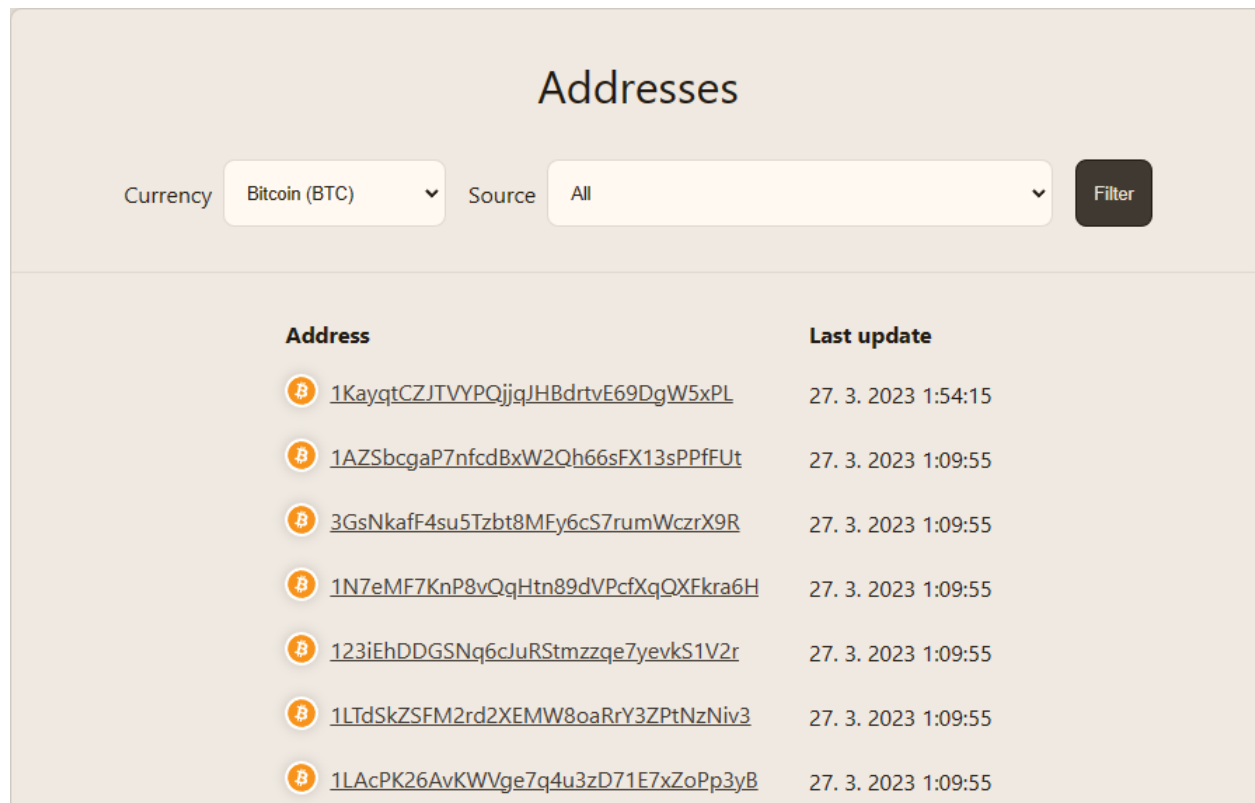
Seznam adres je seřazen sestupně dle data a času poslední aktualizace. Nejdříve se tedy v seznamu objeví naposledy aktualizované adresy. Datum poslední aktualizace je pak určeno nejnovějším staženým souborem spojeným s danou adresou. Toto seřazení bylo zvoleno záměrně a to proto, aby aktuální informace byly zobrazeny co možná neblíže počátku seznamu, jelikož běžný uživatel spíše využije aktuálnější informace než ty, které byly získány při prvním spuštění crawleru.








U jednotlivých adres je pak zobrazena ikona kryptoměny, ke které daná adresa patří. Je tak možné jednoduše identifikovat příslušnost k určité kryptoměně, bez nutnosti přecházet na stránku s danou adresou. Zmíněný přechod však není nikterak náročný, jelikož všechny adresy obsahují odkaz, který vede na stejnou stránku, která je zobrazena po vyhledání adresy. Nemí tedy nutné využívat vyhledávač, i když i ten je na stránce se seznamem adres obsažen.



### 6.2.1 Filtrování seznamu adres

Jednou z klíčových funkcí seznamu adres je možnost jeho filtrování. Bez výběru filtru je ve výchozím stavu zobrazen seznam všech adres napříč všemi zdroji. Pokud je u kryptoměny vybrán např. Bitcoin a u výběru zdroje je ponechána možnost „All“, pak jsou zobrazeny všechny Bitcoinové adresy bez ohledu na zdroj. V případě, že je u kryptoměny ponechána tato možnost, a navíc je vybrán zdroj, např. „CryptoBlacklist“, tak je zobrazen seznam BTC adres v tomto zdroji. Pokud je cílem získat adresy libovolných kryptoměn tohoto zdroje, pak stačí u kryptoměny zvolit možnost „All“.








Address	Last update
 <a href="#">1KayqtCZJTVYPQjjqJHBdrtvE69DgW5xPL</a>	27. 3. 2023 1:54:15
 <a href="#">1AZSbcgaP7nfcDBxW2Qh66sFX13sPPfFUt</a>	27. 3. 2023 1:09:55
 <a href="#">3GsNkafF4su5Tzbt8MFy6cS7rumWczrX9R</a>	27. 3. 2023 1:09:55
 <a href="#">1N7eMF7KnP8vQqHtn89dVPcfXqQXfKra6H</a>	27. 3. 2023 1:09:55
 <a href="#">123iEhDDGSNq6cJuRStmzzqe7yevkS1V2r</a>	27. 3. 2023 1:09:55
 <a href="#">1LTdSkZSFM2rd2XEMW8oaRrY3ZPtNzNiv3</a>	27. 3. 2023 1:09:55
 <a href="#">1LAcPK26AvKWVge7q4u3zD71E7xZoPp3yB</a>	27. 3. 2023 1:09:55

Obrázek 6.5: Ukázka části seznamu BTC adres

## 6.3 Stránka se statistikami

Další funkcí webu je zobrazení souhrnných statistik nahlášených adres jednotlivých zdrojů. Statistiky jsou členěny nejen dle zdroje, ale také dle jeho podkategorií. U každého celku jsou pak ještě rozděleny na jednotlivé kryptoměny. Zdroje jsou seřazené sestupně dle počtu adres a celkový počet adres je možné najít úplně na prvním řádku. Příklad toho, jak mohou statistiky vypadat, je možné vidět na obrázku 6.6. Je zde však uvedena pouze ukázka z webu, kde bylo zobrazení statistik implementováno. Celkové statistiky výsledné databáze podvodných adres jsou blíže představeny v kapitole 8.

Statistics	
Data / Currency	Addresses
<a href="#">All</a>	35 818
 <a href="#">Bitcoin (BTC)</a>	25 591
 <a href="#">Pending</a>	10 227
<a href="#">SeeKoin / Reported BTC Addresses</a>	17 929
 <a href="#">Bitcoin (BTC)</a>	17 929
<a href="#">CheckBitcoinAddress / Reported Addresses</a>	8 250
 <a href="#">Bitcoin (BTC)</a>	5 009
 <a href="#">Pending</a>	3 241

Obrázek 6.6: Ukázka části statistik

### 6.3.1 Ukládání do mezipaměti

I přes veškeré optimalizace a využití indexů je získání počtů adres z databáze stále nejpomalejší součástí webu. Stránka jejíž část je uvedena na obrázku výše se při testování načítala přibližně 7 až 8 sekund. Mohlo by se zdát, že tato doba není příliš dlouhá, nicméně její délka se odvíjí od počtu adres. Dá se očekávat, že při delším používání crawleru (uvedené statistiky odpovídají přibližně 6 hodinám spuštění crawleru) bude počet nahlášených adres stoupat, a tedy s ním i čas na získání těchto statistik. Vzhledem k tomu, že zobrazení statistik není jednou ze zásadnějších funkcí webu, na jejichž aktuálnosti by byla jakákoliv součást aplikace závislá, tak lze dlouhá doba načítání vyřešit následovně. Po prvním načtení se statistiky uloží do mezipaměti, ze které jsou následně po určité době získávány. Tímto způsobem se nejen výrazně zrychlí používání statistik, ale také sníží nároky na databázový server, jelikož počítání adres je také výpočetně náročný proces.

## 6.4 Uživatelské účty

Uživatelské účty poskytují příležitost pro vydefinování hierarchie uživatelů, což ovlivňuje, k jakým datům mají uživatelé v aplikaci přístup, jaké omezení mají v rámci API a jaké další funkce mohou používat.

### 6.4.1 Registrace a přihlášení

Uživateli je při registraci přidělena uživatelská role. V případě, že se v systému registruje jako první, tak je mu přidělena nejvyšší, administrátorská role. Očekává se tedy, že po zprovoznění nástroje BTCAbuseSearch se jako první registruje jeho provozovatel, aby se náhodou administrátorem nestal jiný uživatel webu. Při registraci je pak uživateli vygenerován API token, který mu umožňuje API používat. O tom, jak se od sebe jednotlivé role liší a kdo má jaké možnosti a limity pojednává sekce 6.4.2.

### 6.4.2 Uživatelské role

Jak již bylo v práci několikrát zmíněno, uživatelské role ovlivňují především to, jaká data bude mít uživatel dostupná a jaké limity bude mít při používání API. Nejméně možností má nepřihlášený uživatel, který nemá přístup k API a na webu má přístup pouze k datům určeným pro návštěvníka. Nejvíce možností má pak administrátor, který má kromě velmi vstřícného limitu použití API také možnost přistupovat k datům určeným specificky pro tuto roli, u kterých se neočekává žádné omezení. Navíc pak jako jediný může spravovat uživatele a upravovat jejich roli. Veškeré informace týkající se uživatelských rolí jsou dále shrnuty ve dvou následujících tabulkách 6.2 a 6.3.

Typ uživatele	Použití API	Správa uživatelů	Přístup k datům určeným pro
Nepřihlášený uživatel	NE	NE	Návštěvníka
Návštěvník	ANO	NE	Návštěvníka
Uživatel	ANO	NE	Uživatele
Insider	ANO	NE	Insidera
Administrátor	ANO	ANO	Administrátora

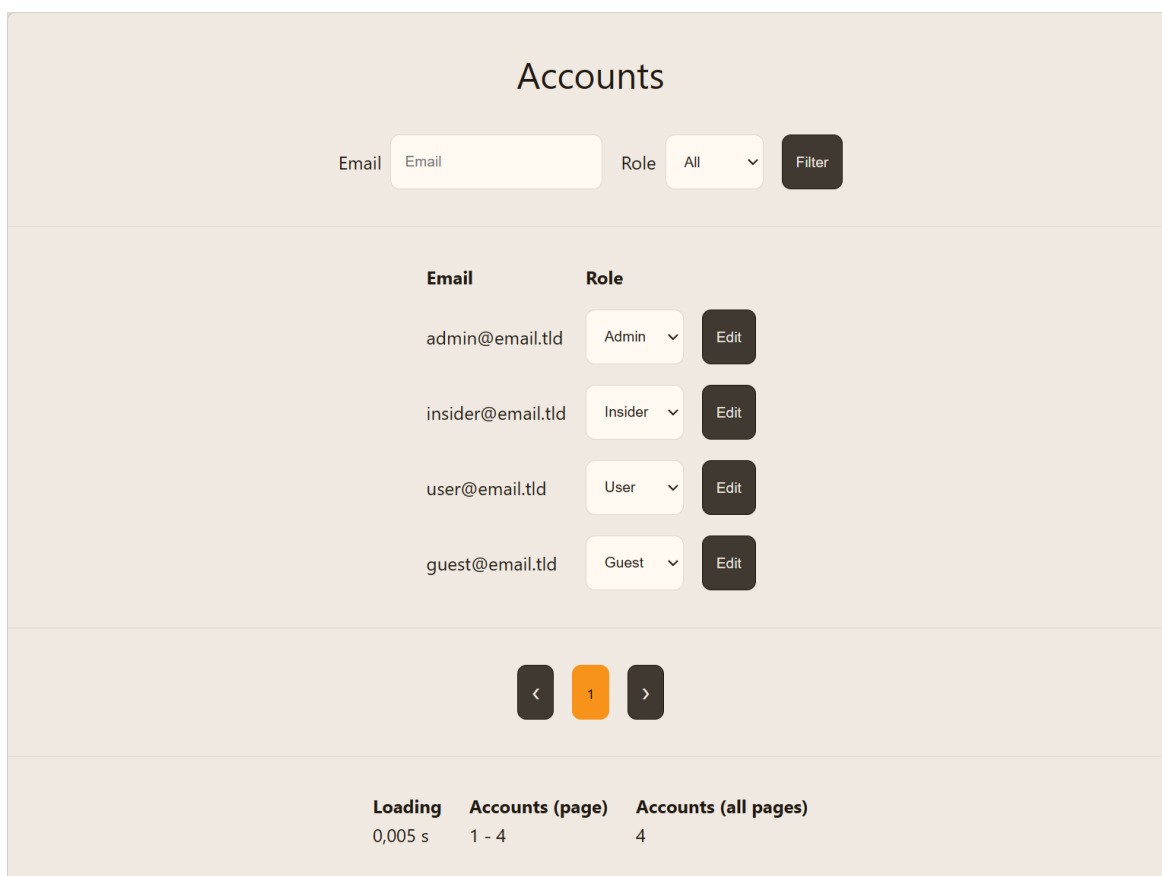
Tabulka 6.2: Možnosti jednotlivých uživatelských rolí

Role	Max. počet požadavků	Obnovení počtu požadavků po dosažení limitu za	Průměrný povolený počet požadavků za sekundu
Návštěvník	100	1 den	$1/864$
Uživatel	3 600	1 hodinu	1
Insider	1 800	10 minut	3
Administrátor	300	60 sekund	5

Tabulka 6.3: Limity počtu požadavků na API pro jednotlivé uživatelské role

### 6.4.3 Správa uživatelů

Administrátor je jediná uživatelská role, která má po přihlášení na webu přístup ke správě uživatelů. Po přihlášení uživatele se, stejně jako u jiných rolí, objeví stránka s informacemi o daném účtu. Na stránce je pak kromě možnosti odhlášení navíc ještě možnost, která odkazuje právě na již zmíněnou správu uživatelů.



Obrázek 6.7: Správa uživatelů

Hlavní funkcí na stránce se správou uživatelů je možnost upravit uživatelskou roli jednotlivých uživatelů. Administrátor může uživateli přidělit jednu ze čtyř následujících rolí – návštěvník, uživatel, insider a administrátor. Cílem této volby je upravit omezení použití aplikace pro daného uživatele. Volba ovlivňuje nejen celkový přístup k datům, které se mezi jednotlivými rolemi liší, ale také frekvenci požadavků na API. V případě, že je danému uživateli přidělena role administrátora, pak je mu taktéž umožněno spravovat uživatele.

Na stránce existuje speciální situace, kdy změna uživatelské role umožněna není, a to v případě, kdy je aktuální uživatel posledním administrátorem v systému a pokusí se změnit svou vlastní roli. V takovém případě by tedy v systému nezůstal žádný administrátor, který by měl přístup k správě uživatelů, což je nežádoucí stav.

Jak je z obrázku 6.7 patrné, tato stránka využívá některé společné prvky s jinými stránkami, jako je filtrace seznamu uživatelů a jeho rozdělení na několik stránek. Filtrování umožňuje omezit zobrazený seznam uživatelů pouze na uživatele vybrané uživatelské role. Po zadání emailové adresy je pak možné účet s danou adresou vyhledávat buď mezi všemi uživateli či opět pouze u vybrané skupiny uživatelů.

## 6.5 API

API je poslední důležitou funkcí webového nástroje. Umožňuje přístup k databázi adres a nahlášení bez nutnosti použití webového prohlížeče a je vhodné především pro implementaci do aplikací třetích stran.

Přístup k datům je stejně jako na webu řízen uživatelskými rolemi. Každý uživatel vlastní jeden API token, který ovlivňuje počet požadavků, které je schopen provést za určitý čas (viz tabulka 6.3). Tento limit nijak neovlivňuje ostatní uživatele. Token musí být umístěn v každém odeslaném požadavku a to způsobem, který je uveden v tabulce níže.

### 6.5.1 Funkce API

Následující tabulka 6.4 obsahuje všechny funkce API. U každé funkce je nejprve zmíněna relativní adresa, na kterou je potřeba požadavek odeslat. Pomocí velkých písmen jsou v adrese zapsány parametry, které je možné do adresy vložit. Tučně jsou zvýrazněny ty parametry, které jsou pro získání informací vyžadovány, zatímco kurzívou jsou uvedeny čistě volitelné parametry. Dále tabulka zmiňuje informace, které je možné při přijetí odpovědi očekávat.

Relativní adresa požadavku	Očekávané informace
/api/address/ <b>ADDRESS</b> ?token= <b>TOKEN</b>	Obsahuje informace o dané adrese – příslušná kryptoměna, seznam identifikátorů datových souborů a datum poslední aktualizace.
/api/addresses?token= <b>TOKEN</b> &page= <i>PAGE</i> &currency= <i>CURRENCY</i> &source= <i>SOURCE</i>	Obsahuje seznam nahlášených adres. U každé je uvedena příslušná kryptoměna, seznam identifikátorů datových souborů a datum poslední aktualizace.
/api/currencies?token= <b>TOKEN</b>	Obsahuje seznam kryptoměn. U každé je uveden její název a zkrácené označení (např. BTC).
/api/data/ <b>DATA</b> ?token= <b>TOKEN</b>	Obsahuje informace o datovém souboru – URL, ze které byl získán, délku obsahu, datum získání a obsah.
/api/sources?token= <b>TOKEN</b>	Obsahuje seznam zdrojů. U každého je uveden identifikátor a název.
/api/tokens/ <b>TOKEN</b>	Obsahuje informace týkající se API tokenu – počet použití, datum vytvoření, datum posledního použití a datum obnovení počtu použití.

Tabulka 6.4: Seznam funkcí API

### 6.5.2 Parametry požadavků API

Jak již bylo zmíněno v předešlé sekci 6.5.1, adresy, na které jsou odesílány požadavky obsahují určité parametry. Parametry označují místa, která mají být nahrazena určitým řetězcem znaků. Některé řetězce je možné získat použitím API, jako třeba zkrácené označení kryptoměny či identifikátor zdroje a jiné jako třeba API token je nutné získat přímo z webu. Pomocí parametrů je možné s API pracovat podobně jako s webovou aplikací a např. u seznamu adres lze také filtrovat dle kryptoměny či zdroje. Snahou je tedy zachovat většinu funkcí, ale navíc nabídnout přístup také pomocí tohoto rozhraní.

V tabulce 6.5 jsou uvedeny parametry požadavků API, které byly zmíněny v předchozí sekci 6.5.1. U každého z nich je pak uvedena její očekávaná hodnota a také místo, kde je možné ji nalézt.

Parametr	Očekávaná hodnota	Kde lze nalézt požadovanou hodnotu
ADDRESS	Kryptoměnová adresa	Je možné ji nalézt v seznamu adres na webu či v rámci API.
CURRENCY	Zkrácené označení kryptoměny (např. „BTC“)	Lze nalézt na webu na stránce k API či při získání seznamu kryptoměn v rámci API.
DATA	Číselný identifikátor datového souboru	Je obsažen v seznamu identifikátorů datových souborů u informací k dané adrese v rámci API.
PAGE	Číslo stránky	Na webu na stránce se seznamem adres je číslo stránky označeno v navigaci u příslušného tlačítka.
SOURCE	Číselný identifikátor zdroje	Nachází se v seznamu zdrojů v rámci API.
TOKEN	API token vygenerovaný při registraci uživatele na webu	Je možné jej nalézt v uživatelském účtu po přihlášení.

Tabulka 6.5: Seznam parametrů v požadavcích API

## Kapitola 7

# Použité technologie

Při implementaci prostředí bylo využíváno několik níže zmíněných technologií. Jako první je zmíněn databázový systém PostgreSQL, který je využíván k ukládání dat. Dále jsou uvedeny Node.js, Express.js a EJS, které zajišťují správný chod serveru, jeho komunikaci směrem k uživateli a také odpovídající HTML výstup. Jako poslední je pak zmíněn GitHub, který byl používán k zaznamenávání veškerých implementovaných změn.

### 7.1 PostgreSQL

PostgreSQL je open-source objektově-relační databázový systém, který používá a rozšiřuje jazyk SQL o další funkce, které zlepšují škálovatelnost a bezpečné ukládání dat. Tento systém je spolehlivý, robustní a výkonný, což je také hlavní důvod, proč byl pro tuto práci zvolen oproti jiným alternativám. [24]

Hlavní výhodou PostgreSQL je jeho rychlost ukládání dat, což umožňuje zkrátit celkový čas potřebný pro crawlování externího obsahu, jelikož při crawlování je do databáze ukládáno velké množství dat. Snaha data co nejrychleji uložit do databáze má smysl z toho důvodu, že by data z externích zdrojů mohla být odebrána či by zdroje nemusely být do budoucna dostupné. Čím dříve se tedy dostanou do databáze, tím dříve je zajištěno jejich uchování a také jejich použití v rámci vyvíjeného řešení. [25]

Rychlost čtení je pak o něco méně důležitá, jelikož toto řešení nevyžaduje použití real-time dat, které jsou navíc pro většinu uživatelů stejné. Uživatel tedy nemusí nutně pracovat s aktuální reprezentací dat v databázi a data může načítat např. z mezipaměti serveru, čímž se snižuje význam rychlého čtení z databáze. [25]

Vyvíjená databáze používá PostgreSQL ve verzi 15.2 a k databázi přistupují všechny nástroje řešení skrze různé rozhraní balíčků knihoven jednotlivých programovacích jazyků. Crawler, který je napsán v jazyce Python k tomu používá knihovny: [psycopg](#) a [psycopg\\_pool](#) a web společně s API pak Node.js balíčky: [pg](#) a [connect-pg-simple](#).

## 7.2 Node.js

Node.js je open-source, multiplatformní běhové prostředí JavaScriptu. Jedná se o asynchronní událostmi řízený běhový modul JavaScriptu navržený k vytváření škálovatelných síťových aplikací. Node.js nepoužívá vlákna OS ani blokování procesu, jelikož téměř žádná funkce v Node.js přímo neprovádí vstupně-výstupní operace, což je v kontrastu k dnešnímu běžnějšímu modelu souběžnosti. [26]

Node.js obsahuje V8 JavaScript engine, stejný jako v Google Chrome a dalších prohlížečích. Je napsán v C++ a je schopen běžet na systémech macOS, Linux, Windows a dalších. Engine kompiluje JavaScriptový kód pomocí JIT (just-in-time), čímž urychluje jeho spuštění. [27]

Ve vyvíjeném řešení je použité prostředí Node.js ve verzi 18.15 LTS (verze s dlouhodobou podporou), kde je s pomocí webového frameworku Express.js vytvořena API, a navíc ještě s pomocí šablonovacího jazyka EJS také web. Node.js tedy poskytuje prostředí pro zpracování HTTP požadavků obou nástrojů, k nimž zároveň poskytuje odpovídající obsah v podobě HTML (web) či JSON (API).

## 7.3 Express.js

Express.js je rychlý, flexibilní, robustní a minimalistický webový framework pro Node.js. Poskytuje funkce, které umožňují práci s HTTP požadavky, což zjednodušuje jejich zpracování a vývoj vlastní API. Vzhledem k tomu, že je Express.js pouze tenkou vrstvou nad Node.js, tak zásadně nezpomaluje jeho výkon, což je jeden z důvodů, proč je tento framework součástí řady dalších populárních frameworků. [28]

## 7.4 Embedded JavaScript templates

Embedded JavaScript templates zkr. EJS je jednoduchý šablonovací jazyk, který umožňuje generovat HTML elementy pomocí JavaScriptu. Nástroj BTCAbuseSearch tento jazyk používá na straně serveru, kde se pomocí EJS generuje HTML stránka, kterou následně webový prohlížeč načítá a zobrazuje. EJS umožňuje vytvářet šablony, kde lze pomocí speciálních značek vkládat dodatečný obsah, který je součástí finální HTML stránky ještě před jejím odesláním prohlížeči. Tímto způsobem je zajištěno efektivní znovupoužití stránek, jelikož se obvykle mění pouze malá část jejich celkového obsahu (např. jednotlivé stránky v rámci seznamu adres). Vzhledem k tomu, že jsou stránky z pohledu klienta statické a nevyžadují lokální použití JavaScriptu, tak jsou zároveň funkční i při jeho zakázání v prohlížeči. [29]



## 7.5 GitHub

GitHub je webová platforma pro ukládání a spravování kódu. Umožňuje sledovat změny kódu v čase a případně se vrátit k předchozím verzím software. Pro vytváření jednotlivých verzí software používá nástroj Git, což je open-source verzovací systém.

Při vývoji práce je GitHub využíván převážně pro sledování změn při inkrementálním vývoji všech nástrojů řešení. Seznam změn je možné najít v historii commitů, kde každý jeden commit obsahuje krátký popis toho, k jakým změnám v daném commitu došlo. Součástí projektového repositáře je pak také značkovací soubor README.md, který obsahuje seznam funkcí, které by měly být implementovány s tím, že u každé je také grafické označení aktuálního stavu vývoje dané funkce. [30]

## Kapitola 8

# Výsledky

Databáze vytvořená z dat získaných procházením zdrojů nahlášení podvodů spojených s kryptoměnami obsahuje bezmála 25 591 nahlášených Bitcoinových a 10 227 nahlášených altcoinových adres. Zmíněné údaje je možné ověřit v databázi pomocí příkazů 8.1 a 8.3.

---

```
SELECT COUNT(DISTINCT address_data.address_id)
FROM address_data
JOIN address ON address.address_id = address_data.address_id AND currency_id = 1
```

---

Výpis kódu 8.1: Získání počtu podvodných BTC adres

K dosažení těchto výsledků bylo potřeba nechat běžet crawler přerušovaně přibližně 6 hodin od 25. 3. 2023 do 27. 3. 2023 na zařízení s komponentami uvedenými v příloze C. Jedná se o očekávané počty vzhledem k tomu, že každý jeden požadavek na cílový server od jeho odeslání až po získání odpovědi trvá určitý čas. Lze předpokládat, že by po delším běhu programu mělo dojít až k dosažení výsledků uvedených u zdrojů BitcoinAbuse či CheckBitcoinAddress (viz tabulka 2.1) a to z důvodu, že jsou procházeny celé jejich seznamy. Výsledná databáze by pak mohla být i o něco větší, jelikož některé další zdroje mohou obsahovat adresy, které nebudou součástí jiných zdrojů.

### 8.1 Poměr nahlášených BTC adres

Pro zjištění poměru nahlášených BTC adres je nejprve potřeba znát celkový počet BTC adres. K tomu lze využít databáze, jelikož obsahuje veškeré adresy z Bitcoinového blockchainu stažené ze zdroje LoyceV. Ke dni 25. 3. 2023 se jedná celkem o 1 127 482 941 unikátních Bitcoinových adres.

---

```
SELECT COUNT(*)
FROM address
WHERE currency_id = 1
```

---

Výpis kódu 8.2: Získání celkového počtu BTC adres

Z výše uvedených údajů tedy vychází, že byly získány nahlášení přibližně 0,02 promile všech Bitcoinových adres. Potenciál 90 tisíc by pak představoval 0,08 promile. Pro kontext je potřeba uvést další data, která pomohou určit, zda je možné tento výsledek považovat za uspokojivý.

Dle reportu zveřejněného společností Chainalysis bylo v roce 2022 k nelegálním aktivitám použito 0,24 % všech kryptoměnových transakcí. Chainalysis bohužel neuvádí, jaké kryptoměny jsou součástí tohoto zjištění, takže nelze toto procento aplikovat pouze na Bitcoin. [31]

Článek na serveru [cointelegraph.com](https://cointelegraph.com) pak uvádí, že ve stejném roce došlo k 93,1 milionům Bitcoinových transakcí. V transakcích se však mohly často objevovat ty stejné adresy a také ne každá transakce musela obsahovat pouze jeden výstup. Z toho tedy nelze odhadnout, kolik adres bylo použito v transakcích. [32]

Server [glassnode.com](https://glassnode.com) poskytuje počet aktivních BTC adres za určité časové období. Aktivními adresami jsou myšleny ty, které vystupují na jakékoliv straně transakce. Z údajů uvedených na tomto serveru pak vychází, že průměrný měsíční počet aktivních adres ve sledovaném roce 2022 se pohybuje okolo 16,5 milionu. [33]

Ze všech zmíněných údajů je patrné, že i přestože je na Bitcoinovém blockchainu dnes již více než miliarda unikátních adres, tak počtu transakcí za rok či aktivně používaných adres za měsíc není ani zdaleka tolik. Dále je potřeba počítat s tím, že ne všechny adresy, které jsou používány k nelegálním aktivitám, budou nahlášený na některém z veřejně dostupných zdrojů. Z tohoto pohledu tak lze získaný počet nahlášených BTC adres považovat za adekvátní.

## 8.2 Další nahlášené adresy

Jak již bylo zmíněno, databáze obsahuje také 10 227 nahlášených adres, které jsou spojeny s jinými kryptoměnami, než je Bitcoin. Jedná se o adresy kryptoměn jako je Ethereum, Cardano a další.

---

```
SELECT COUNT(DISTINCT address_data.address_id)
FROM address_data
JOIN address ON address.address_id = address_data.address_id AND (currency_id <> 1
OR currency_id IS NULL)
```

---

Výpis kódu 8.3: Získání počtu podvodných altcoinových adres

K těmto adresám se crawler dostal primárně díky tomu, že byly v nemalém množství obsaženy na procházených webech. Cílem práce však není zaměřovat se na altcoinové adresy, a tak je tato část databáze určena spíše pro další rozšíření práce.

## Kapitola 9

# Závěr

Cílem práce bylo zanalyzovat state of the art nástrojů, které pracují s databází Bitcoinových adres a navrhnout vlastní řešení, které vyřeší některé jejich nedostatky a případně zavede nové funkce.

Stanovené cíle byly v práci splněny. Bylo nalezeno několik nástrojů zaměřených na shromažďování Bitcoinových adres, avšak pouze malá část z nich data poskytovala veřejně. Zmíněné nástroje je však možné použít společně s navrženým řešením při analýze podvodů spojených s kryptoměnami či pro odhalení vlastníků Bitcoinových adres.

Vytvořené řešení se skládá ze dvou nástrojů pojmenovaných BTCAbuseCrawler a BTCAbuseSearch. První prochází veřejně dostupné zdroje nahlášení kryptoměnových podvodů a data ukládá do databáze a na disk. Druhý nástroj pak data poskytuje na webových stránkách či prostřednictvím API. Zatímco webové stránky slouží pro jednoduché a uživatelsky přívětivé použití, tak API je určeno především pro zapojení do aplikací třetích stran. Implementované řešení je plně funkční a k jeho nasazení na cílové zařízení je možné použít přiloženou uživatelskou příručku. Během vývoje systému bylo využito několika různých programovacích jazyků (Python a JavaScript) a technologií (PostgreSQL, Node.js, Express.js, EJS a GitHub). Pomocí nástroje BTCAbuseCrawler se podařilo shromáždit celkem více než 25 tisíc nahlášených Bitcoinových adres a více než 10 tisíc nahlášených altcoinových adres. Uvedené výsledky lze považovat za uspokojivé.

U obou nástrojů je stále prostor pro další rozšíření. Crawler by mohl procházet ještě více veřejných zdrojů než doposud. Bylo by také dobré zvážit rozšíření zaměření i na další kryptoměny, jelikož bylo shromážděno relativně hodně altcoinových adres a např. na síti Ethereum bylo v roce 2022 provedeno průměrně až čtyřikrát tolik transakcí jak na Bitcoinu. [32] Webová aplikace by mohla lépe začlenit responzivní design, podporu mobilních zařízení či tmavý režim. Oba nástroje by pak mohly svá nastavení obsahovat přímo na webu.

# Literatura

1. BADAWI, Emad; JOURDAN, Guy-Vincent; BOCHMANN, Gregor; ONUT, Iosif-Viorel. An Automatic Detection and Analysis of the Bitcoin Generator Scam. In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2020, s. 407–416. Dostupné z DOI: 10.1109/EuroSPW51379.2020.00061.
2. BAMBUCH, Vladislav. *Platform for Cryptocurrency Address Collection*. Excel, 2020.
3. MACHARIA, Caroline Wanjira. *Maintaining a bitcoin address repository through focused web crawling*. 2017. Dis. pr. Strathmore University.
4. SANDERS, Dakota Blake. Oops, Did I Do That? Uncovering Apparent Mishandling, Tax Fraud, and Illegal Contributions via a Systematic Study of Cryptocurrency Donations to US Political Campaigns. 2020.
5. *VoteSmart* [online]. [cit. 2022-10-13]. Dostupné z: <https://justfacts.votesmart.org/>.
6. BARTOLETTI, Massimo; PES, Barbara; SERUSI, Sergio. Data Mining for Detecting Bitcoin Ponzi Schemes. In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2018, s. 75–84. Dostupné z DOI: 10.1109/CVCBT.2018.00014.
7. GARBA, Abba; GUAN, Zhi; LI, Anran; CHEN, Zhong. Analysis of Man-In-The-Middle of Attack on Bitcoin Address. In: *ICETE (2)*. 2018, s. 554–561.
8. *LoyceV* [online]. [cit. 2023-02-24]. Dostupné z: <http://alladdresses.loyce.club/>.
9. *BlockChair* [online]. [cit. 2023-02-24]. Dostupné z: <http://blockchair.com/dumps>.
10. *BitcoinAbuse* [online]. [cit. 2022-10-19]. Dostupné z: <https://www.bitcoinabuse.com/>.
11. *CheckBitcoinAddress* [online]. [cit. 2023-02-24]. Dostupné z: <https://checkbitcoinaddress.com/abuse-reports-to-bitcoin-address>.
12. *BitcoinAIS* [online]. [cit. 2023-02-24]. Dostupné z: <https://bitcoinais.com>.
13. *BitcoinGeneratorScam* [online]. [cit. 2023-02-24]. Dostupné z: <http://ssrg.site.uottawa.ca/bgsieeesb2020/#urls>.
14. *CryptoBlacklist* [online]. [cit. 2023-02-24]. Dostupné z: <https://www.cryptoblacklist.io>.

15. *CryptoScamDB* [online]. [cit. 2022-10-19]. Dostupné z: <https://cryptoscamdb.org/>.
16. *SeeKoin* [online]. [cit. 2023-02-24]. Dostupné z: <https://www.seekoin.com/address.php>.
17. *Cryptscam* [online]. [cit. 2023-02-24]. Dostupné z: <https://cryptscam.com>.
18. *BitcoinWhosWho* [online]. [cit. 2023-02-24]. Dostupné z: <https://www.bitcoinwhoswho.com/>.
19. *Open Source Intelligence (OSINT)* [online]. [cit. 2023-04-16]. Dostupné z: <https://www.informacnigramotnost.cz/skoleni-workshopy/open-source-intelligence-osint/>.
20. *Chainalysis Address Screening* [online]. [cit. 2023-04-16]. Dostupné z: <https://www.chainalysis.com/cryptocurrency-address-screening/>.
21. *Elliptic Lens* [online]. [cit. 2023-04-22]. Dostupné z: <https://www.elliptic.co/solutions/crypto-wallet-screening>.
22. *European police seize BestMixer, saying it helped launder \$200 million worth of cryptocurrency* [online]. [cit. 2023-04-28]. Dostupné z: <https://cyberscoop.com/bestmixer-bitcoin-laundering-mcafee-europol/>.
23. *How Much Bitcoin Is Lost Forever?* [online]. [cit. 2023-04-04]. Dostupné z: <https://www.hedgewithcrypto.com/how-much-bitcoin-is-lost/>.
24. *PostgreSQL* [online]. [cit. 2023-03-07]. Dostupné z: <https://www.postgresql.org/>.
25. *PostgreSQL vs MySQL: What is the Difference between them?* [online]. [cit. 2023-03-07]. Dostupné z: <https://blog.devart.com/postgresql-vs-mysql.html>.
26. *Node.js* [online]. [cit. 2023-03-07]. Dostupné z: <https://nodejs.org/>.
27. *What is the Node.js (Node) runtime environment? – TechTarget Definition* [online]. [cit. 2023-03-07]. Dostupné z: <https://www.techtarget.com/whatis/definition/Nodejs>.
28. *Express.js – Node.js web application framework* [online]. [cit. 2023-03-07]. Dostupné z: <https://expressjs.com/>.
29. *EJS – Embedded JavaScript templates* [online]. [cit. 2023-03-07]. Dostupné z: <https://ejs.co/>.
30. *What Is GitHub? A Beginner's Introduction to GitHub* [online]. [cit. 2023-03-07]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-github/>.
31. *2023 Crypto Crime Trends: Illicit Cryptocurrency Volumes Reach All-Time Highs Amid Surge in Sanctions Designations and Hacking* [online]. [cit. 2023-04-23]. Dostupné z: <https://blog.chainalysis.com/reports/2023-crypto-crime-report-introduction/>.
32. *Ethereum transactions 338% higher in 2022 but Bitcoin remains most popular* [online]. [cit. 2023-04-23]. Dostupné z: <https://cointelegraph.com/news/ethereum-transactions-338-higher-in-2022-but-bitcoin-remains-most-popular>.

33. *Bitcoin: Active Addresses - Glassnode Studio* [online]. [cit. 2023-04-23]. Dostupné z: <https://studio.glassnode.com/metrics?a=BTC&ema=0&m=addresses.ActiveCount&mAvg=0&mMedian=0&resolution=1month>.

# Příloha A

## Uživatelská příručka

Uživatelská příručka je určena pro osoby zodpovědné za nasazení řešení na cílovém zařízení. Nejprve je zmíněn proces přípravy obou nástrojů a programů, které je potřeba nainstalovat a následně je uvedeno, jak a v jakém pořadí je možné nástroje spustit. Dále je uveden adresářový strom, který obsahuje strukturu adresáře a popis jednotlivých složek a souborů.

### A.1 Příprava nástrojů

Pro zprovoznění nástrojů je nejprve potřeba nainstalovat vybrané programy třetích stran, které jsou vyžadovány danými nástroji a bez nichž není možné nástroje spustit. V následujících dvou sekcích (A.1.1 a A.1.2) je postup instalace a přípravy rozepsán do bodů, jejichž pořadí je třeba se držet.

Dále je také třeba zmínit, že je nutné programy instalovat ve verzi pro operační systém Windows, jelikož řešení bylo vyvíjeno a průběžně testováno v systému Windows 10 a později také ve Windows 11. Funkčnost v starších verzích Windows je možná, nicméně není zaručena. Pro další operační systémy pak dané řešení nebylo testováno.

#### A.1.1 BTCAbuseCrawler

1. Stáhnout a nainstalovat PostgreSQL 15.2 – během instalace je nutné nastavit heslo: **postgres**
2. Stáhnout a nainstalovat Python 3.11 – během instalace je nutné vybrat možnost: přidat **python.exe** do PATH
3. V „Nastavení / Aplikace / Rozšířené nastavení aplikací / Aliasy pro spouštění aplikací“ vypnout **python.exe**
4. Restartovat počítač
5. Jít do projektového adresáře **btc\_abuse\_crawler**



6. Přejmenovat soubor `example_db.json` na `db.json`
7. Změnit heslo připojení k databázi v `db.json`
8. Přejmenovat soubor `example_setup.json` na `setup.json`
9. Změnit hesla uživatelů databáze v `setup.json`
10. Otevřít příkazový řádek
11. V příkazovém řádku nastavit aktuální pracovní adresář na projektový adresář  
`btc_abuse_crawler`
12. Nainstalovat balíčky knihoven pomocí příkazu: `pip install -U -r requirements.txt`

### A.1.2 BTCAbuseSearch

1. Stáhnout a nainstalovat Node.js 18.15 LTS (verze s dlouhodobou podporou)
2. Restartovat počítač
3. Jít do projektového adresáře `btc_abuse_search`
4. Přejmenovat soubor `example_db.json` na `db.json`
5. Změnit heslo připojení k databázi v `db.json`
6. Otevřít příkazový řádek
7. V příkazovém řádku nastavit aktuální pracovní adresář na projektový adresář  
`btc_abuse_search`
8. Nainstalovat balíčky knihoven pomocí příkazu:  
`npm i -g npm-check-updates && ncu -u && npm i`

## A.2 Spuštění nástrojů

Jakmile jsou všechny potřebné programy nainstalovány a nástroje připraveny, je možné přejít k jejich spuštění. Nejprve je nutné spustit crawler, jelikož se při jeho prvním spuštění vytvoří databáze a nahrají se výchozí data. Chvilí poté je pak možné spustit webový nástroj.

Vzhledem k tomu, že crawler nejprve začíná stahováním seznamu Bitcoinových adres z blockchainu, které trvá minimálně prvních přibližně 50-70 minut (dle specifikací daného zařízení), tak během uvedené doby není možné očekávat jakákoliv data, která by bylo možné na webu zobrazit. Je to dáno tím, že je daná databázová tabulka během této doby uzamčená pro rychlejší zápis, a tedy než je tento proces dokončen, tak se jeví jako by byla prázdná.

Po dokončení tohoto procesu by pak již mělo být na webu možné zobrazit první získaná data, což je indikátor toho, že by měla být aplikace funkční. Pokud se u některých zdrojů zobrazují v příkazovém řádku crawleru chyby, či na webu data daného zdroje chybí, pak to znamená, že daný zdroj buď ještě nebyl crawlován či je aktuálně nebo dokonce trvale nedostupný. Dostupnost externích zdrojů není možné zaručit, a tedy je vhodné aplikaci spustit co nejdříve, aby bylo těchto komplikací co nejméně.

Stejně jako u přípravy, tak i u spuštění jednotlivých nástrojů je uveden odpovídající bodový postup.

### A.2.1 BTCAbuseCrawler

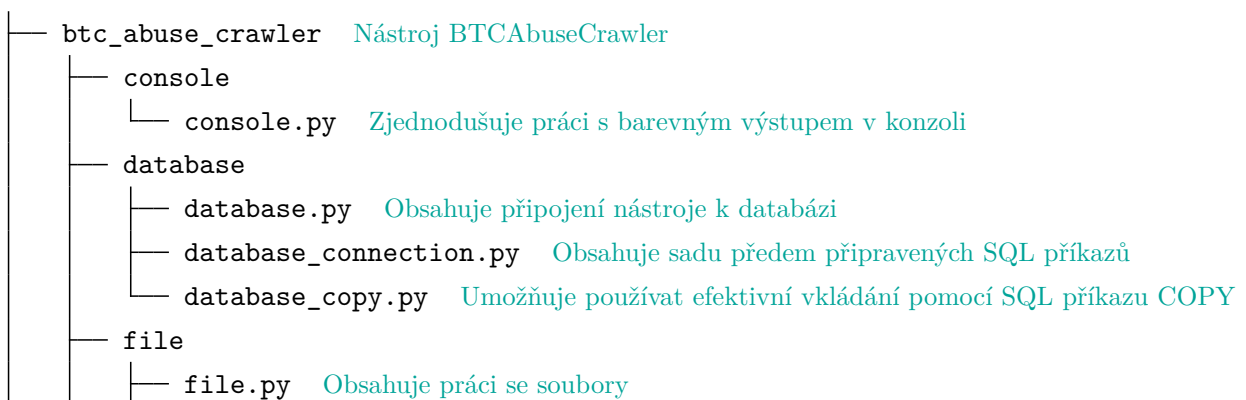
1. Otevřít příkazový řádek (jako administrátor)
2. V příkazovém řádku nastavit aktuální pracovní adresář na projektový adresář  
`btc_abuse_crawler`
3. Spustit crawler pomocí příkazu: `python main.py`
4. V případě, že se zobrazí okno Řízení uživatelských účtů, tak vybrat možnost „Ano“

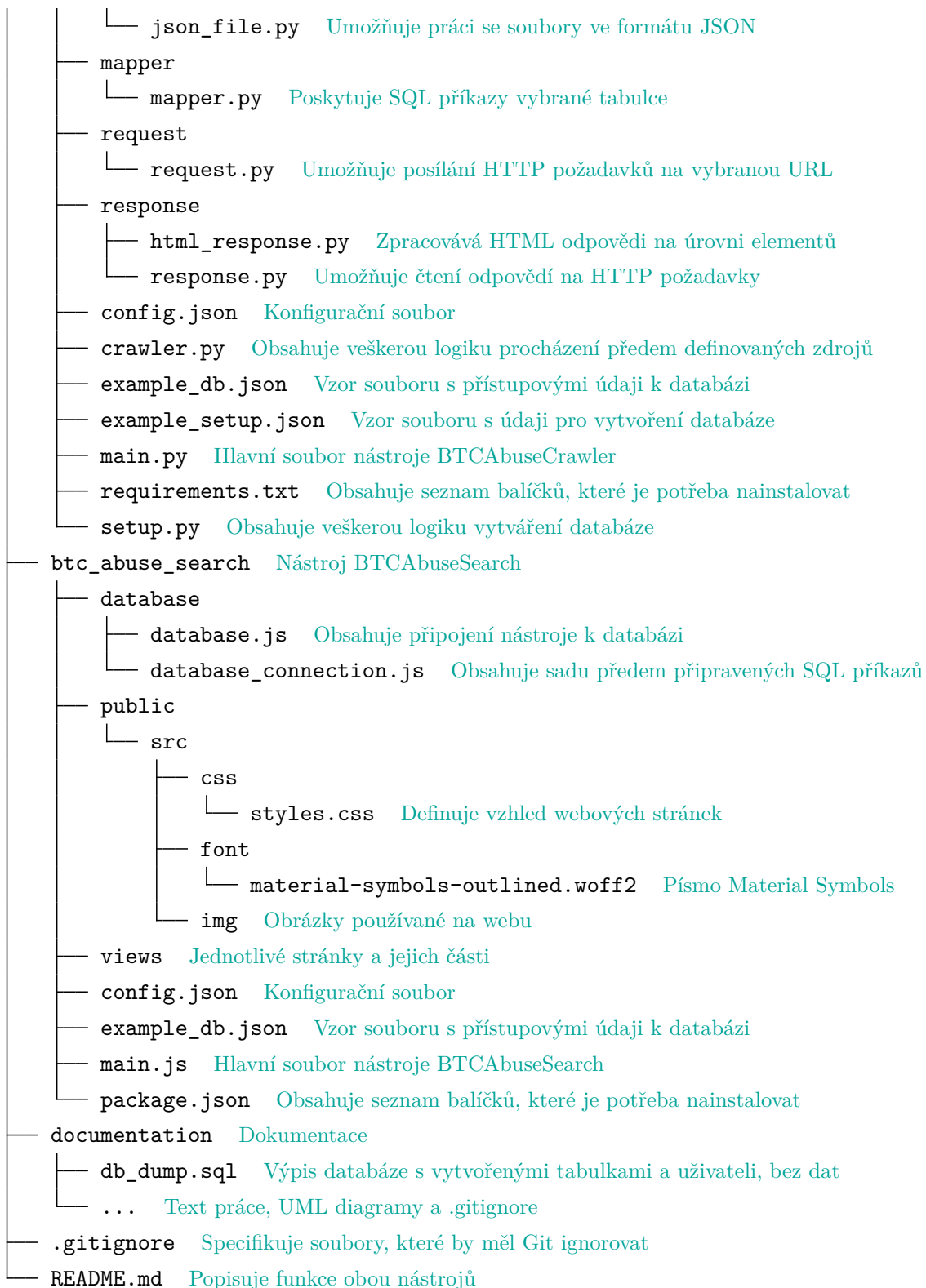
### A.2.2 BTCAbuseSearch

1. Otevřít příkazový řádek
2. V příkazovém řádku nastavit aktuální pracovní adresář na projektový adresář  
`btc_abuse_search`
3. Spustit webový nástroj pomocí příkazu: `node main.js`

Web je následně možné navštívit ve webovém prohlížeči na adrese: „`http://localhost:3000`“.

## A.3 Adresářový strom





## Příloha B

# Datový slovník

Zatímco diagram vztahu entit databáze v sekci 4.3 graficky a přehledně znázorňuje především propojení a velikost databázové struktury, tak datový slovník slouží pro podrobnější popis jednotlivých tabulek a atributů a jejich využití v systému.

### Tabulka account

Tabulka account uchovává informace o uživatelských účtech na webu. Přihlašování umožňují atributy email a password, které jsou porovnávány jak při pokusu o přihlášení, tak také při pokusu o registraci. Následně jsou zde informační atributy zaznamenávající IP uživatele a čas registrace i posledního přihlášení, což jsou údaje, které se mohou hodit k nějaké další kontrole či statistice. Velký význam pak má identifikátor role, který určuje, k jakým údajům má uživatel přístup ať už na webu či pomocí API.

Atribut	Dat. typ	Klíč	Null	Význam
account_id	BigSerial	PK	Ne	Identifikátor uživatele
role_id	SmallInt	FK	Ne	Identifikátor role
email	Text	UQ	Ne	Email uživatele
password	Text		Ne	Heslo uživatele (hash)
signed_up_by_ip	Text		Ne	IP uživatele (registrace)
last_signed_in_by_ip	Text		Ano	IP uživatele (poslední přihlášení)
signed_up_at	Timestamp		Ne	Čas registrace
last_signed_in_at	Timestamp		Ano	Čas posledního přihlášení

Tabulka B.1: Databázová tabulka account

## Tabulka address

Tabulka address slouží k uchování kryptoměnové adresy. Adresa je pak doplněna o identifikátor kryptoměny, který indikuje, zda se jedná o adresu na Bitcoinovém, Ethereumovém či jiném blockchainu. Tento údaj pak umožňuje podle těchto kryptoměn adresy na webu filtrovat. Identifikátor podkategorie zdroje říká, kde byla daná adresa při crawlování poprvé nalezena, což je údaj, který systém aktuálně nevyužívá, ale mohl by se potenciálně hodit při jeho rozšiřování.

Atribut	Dat. typ	Klíč	Null	Význam
address_id	BigSerial	PK	Ne	Identifikátor adresy
currency_id	SmallInt	FK	Ano	Identifikátor kryptoměny
source_label_id	SmallInt	FK	Ne	Identifikátor podkategorie zdroje
address	Text	UQ	Ne	Kryptoměnová adresa

Tabulka B.2: Databázová tabulka address

## Tabulka address\_data

Tabulka address\_data propojuje kryptoměnovou adresu s daty získanými crawlováním a to pomocí identifikátorů adresy a dat. Atribut roles obsahuje všechny identifikátory rolí, které mají k daným datům přístup.

Atribut	Dat. typ	Klíč	Null	Význam
address_data_id	BigSerial	PK	Ne	Identifikátor propojení adresy a dat
address_id	BigInt	FK, UQ <sub>1</sub>	Ne	Identifikátor adresy
data_id	BigInt	FK, UQ <sub>1</sub>	Ne	Identifikátor dat
roles	SmallInt[]		Ne	Pole identifikátorů rolí

Tabulka B.3: Databázová tabulka address\_data

## Tabulka currency

Tabulka currency obsahuje informace o kryptoměnách, se kterými systém pracuje. Seznam kryptoměn systém přejímá ze seznamu podporovaných blockchainů na stránce [blockchair.com](https://blockchair.com). V tabulce jsou pak zaznamenány název a kód kryptoměny, dále cesta k souboru s jejím logem a URL řetězec z již zmíněné stránky. Na zmíněné URL jsou poté identifikovány kryptoměny u adres, kde tato informace chybí. Jedná se však pouze o altcoinové adresy, jelikož všechny BTC adresy jsou v systému již obsaženy.

Atribut	Dat. typ	Klíč	Null	Význam
currency_id	SmallSerial	PK	Ne	Identifikátor kryptoměny
name	Text	UQ	Ne	Název kryptoměny
code	Text	UQ	Ne	Kód kryptoměny
logo	Text	UQ	Ne	Cesta k souboru s logem kryptoměny
blockchair_request_name	Text	UQ	Ano	URL řetězec kryptoměny na Blockchair

Tabulka B.4: Databázová tabulka currency

## Tabulka data

Tabulka data slouží k uchování metadat souboru získaného při crawlování. Obsahuje identifikátor URL podkategorie zdroje, která říká, z jakého zdroje byl daný soubor získán a také identifikátor URL, který obsahuje konkrétní URL, ze které byl soubor stažen. Cesta k souboru s daty pak umožňuje jeho načtení a zpracování na webu. Délka obsahu souboru a čas crawlování slouží pouze jako užitečná informace navíc.

Atribut	Dat. typ	Klíč	Null	Význam
data_id	BigSerial	PK	Ne	Identifikátor dat
source_label_url_id	SmallInt	FK	Ne	Identifikátor URL podkategorie zdroje
url_id	BigInt	FK	Ne	Identifikátor URL
path	Text	UQ	Ne	Cesta k souboru s daty
content_length	BigInt		Ne	Délka obsahu souboru
crawled_at	Timestamp		Ne	Čas crawlování

Tabulka B.5: Databázová tabulka data

## Tabulka role

Tabulka role zaznamenává uživatelské role, které mají význam jak z pohledu oprávnění na webu, tak z pohledu přístupu k datům.

Atribut	Dat. typ	Klíč	Null	Význam
role_id	SmallSerial	PK	Ne	Identifikátor role
name	Text	UQ	Ne	Název role

Tabulka B.6: Databázová tabulka role

## Tabulka session

Tabulka session slouží k uchování sezení uživatele webu a je používána Node.js balíčkem express-session.

Atribut	Dat. typ	Klíč	Null	Význam
sid	Varchar	PK	Ne	Identifikátor sezení
sess	JSON		Ne	Data sezení
expire	Timestamptz(6)		Ne	Čas vypršení sezení

Tabulka B.7: Databázová tabulka session

## Tabulka source

Tabulka source obsahuje názvy zdrojů používaných při crawlování a slouží také pro filtrování adres na webu.

Atribut	Dat. typ	Klíč	Null	Význam
source_id	SmallSerial	PK	Ne	Identifikátor zdroje
name	Text	UQ	Ne	Název zdroje

Tabulka B.8: Databázová tabulka source

## Tabulka source\_label

Tabulka source\_label zaznamenává podkategorie zdrojů, a to z toho důvodu, že každý zdroj může obsahovat více typů dat, které nemusí být součástí stejné skupiny i přestože jsou získány ze stejného zdroje. Jako příklad lze uvést Bitcoin Generator Scam, který obsahuje zdroj pouze Bitcoinových adres a pak také zdroj adres i jiných kryptoměn. Pokud se pro danou podkategorii na daném zdroji nachází pouze adresy jedné kryptoměny, pak je tato kryptoměna uvedena v příslušném identifikátoru, který je použit v tabulce address pro označení adres přidanych tímto zdrojem. V případě, že se má při crawlování použít webový vyhledávač, tak je tato informace zaznamenána v atributu search\_data\_by\_address.

Atribut	Dat. typ	Klíč	Null	Význam
source_label_id	SmallSerial	PK	Ne	Identifikátor podkategorie zdroje
source_id	SmallInt	FK, UQ <sub>1</sub>	Ne	Identifikátor zdroje
new_addresses_currency_id	SmallInt	FK	Ano	Identifikátor kryptoměny (pro nově přidané adresy)
search_data_by_address	Boolean		Ne	Použít vyhledávač (crawlování)
name	Text	UQ <sub>1</sub>	Ne	Název podkategorie zdroje

Tabulka B.9: Databázová tabulka source\_label

## Tabulka source\_label\_url

Tabulka source\_label\_url obsahuje URL podkategorií zdrojů, jelikož i podkategorie zdrojů mohou mít více URL. Příkladem je zdroj Bitcoin Generator Scam, který obsahuje dvě URL se seznamy Bitcoinových adres. Toto oddělení není pro tuto práci důležité, takže URL patří do stejné podkategorie. Jakmile je crawlování pro danou URL podkategorie dokončeno, tak je zaznamenán čas posledního crawlování.

Atribut	Dat. typ	Klíč	Null	Význam
source_label_url_id	SmallSerial	PK	Ne	Identifikátor URL podkategorie zdroje
source_label_id	SmallInt	FK, UQ <sub>1</sub>	Ne	Identifikátor podkategorie zdroje
url_id	BigInt	FK, UQ <sub>1</sub>	Ne	Identifikátor URL
last_crawled_at	Timestamp		Ano	Čas posledního crawlování

Tabulka B.10: Databázová tabulka source\_label\_url

## Tabulka token

Tabulka token uchovává informace o API tokenu. Kromě samotného tokenu zaznamenává uživatele, se kterým je spojena, počet použití tokenu, IP uživatele a čas při vytvoření a posledním použití a také čas, kdy se počet použití resetuje. Počet použití a čas resetování zajišťují to, že na základě uživatelské role je daná API limitovaná tak, aby nemohla být využívána přes limit na úkor stability systému.



Atribut	Dat. typ	Klíč	Null	Význam
token_id	BigSerial	PK	Ne	Identifikátor API tokenu
account_id	BigInt	FK	Ne	Identifikátor uživatele
token	Text	UQ	Ne	API token
use_count	BigInt		Ne	Počet použití
created_by_ip	Text		Ne	IP uživatele (vytvoření)
last_used_by_ip	Text		Ano	IP uživatele (poslední použití)
created_at	Timestamp		Ne	Čas vytvoření
last_used_at	Timestamp		Ano	Čas posledního použití
reset_use_count_at	Timestamp		Ano	Čas resetování počtu použití

Tabulka B.11: Databázová tabulka token

## Tabulka url

Tabulka url obsahuje unikátní URL používané při crawlování.

Atribut	Dat. typ	Klíč	Null	Význam
url_id	BigSerial	PK	Ne	Identifikátor URL
address	Text	UQ	Ne	URL adresa

Tabulka B.12: Databázová tabulka url

## Příloha C

# Specifikace testovacího zařízení

Při vývoji bylo řešení několikrát otestováno nejen kvůli případným chybám, ale také proto, aby mohly být objeveny problémová místa z hlediska výkonu. Testování pomáhá při volbě vhodné optimalizace, ošetření výjimek a určení přibližného času potřebného k získání určitého množství nahlášených adres. Výsledky testování jsou blíže popsány v kapitole 8.

Jelikož se rychlost vykonávání crawleru odvíjí od specifikací zařízení, na kterém je nástroj spuštěn, tak je vhodné uvést, jaké komponenty byly součástí zařízení, na němž bylo řešení testováno. Specifikace jsou uvedeny v tabulce C.1.

<b>Operační systém</b>	Windows 10 (64-bit)
<b>Procesor</b>	Intel Core i5-6600K
<b>RAM</b>	16 GB
<b>Grafická karta</b>	Radeon RX 480
<b>Úložiště</b>	Samsung SSD 970 EVO 1 TB

Tabulka C.1: Specifikace testovacího zařízení