EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/B/2018/36097107837917956

PRAKTICKÉ POUŽITIE TECHNOLÓGIE BLOCKCHAIN

BAKALÁRSKA PRÁCA

16.05.2018 **Andrej Kaňuch**

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/B/2018/36097107837917956

PRAKTICKÉ POUŽITIE TECHNOLÓGIE BLOCKCHAIN

Bakalárska práca

Študijný program: Hospodárska informatika
Študijný odbor: Hospodárska informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Vedúci záverečnej práce: Ing. Igor Bandurič, PhD.

Bratislava 2018 **Andrej Kaňuch**

ZADANIE ZÁVEREČNEJ PRÁCE

Technológia blockchain, na ktorej je postavený Bitcoin je univerzálna technológia, ktorej uplatnenie v ekonomickej sfére je rozsiahle, aj keď sa ešte hľadajú praktické možnosti využitia. Cieľom práce je vytvoriť prehľad o tejto technológii spolu s praktickou ukážkou jej použitia - implementačný príklad.

Čestné vyhlásenie (nepovinná časť)

Vyhlasujem, že som celú bakalársku prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Bratislava, 15. máj.2018 .....

vlastnoručný podpis

Poďakovanie

Rád by som sa poďakoval môjmu školiteľovi Ing. Igorovi Banduričovi, PhD. za odborné vedenie, cenné pripomienky, rady a smerovanie, ktoré mi poskytoval počas písania tejto práce.

ABSTRAKT

KAŇUCH, Andrej: Praktické použitie technológie blockchain .Andrej Kaňuch – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra aplikovanej informatiky. – Vedúci Ing. Igor Bandurič, PhD. – Stupeň odbornej kvalifikácie: Bakalár. Bratislava: Fakulta hospodárskej informatiky EUBA ,2017

Bakalárska práca je zameraná na bližšie pochopenie technológie Blockchain. Táto technológia sa stáva záujmom veľkých firiem ale aj fanúšikov kryptomien po celom svete. Tento špeciálny druh distribuovanej účtovnej knihy si nachádza cestu k bežne vyskytujúcim sa činnostiam človeka.

Kľúčové slová: Blockchain,internet,kryptomeny,bezpečnosť,financie,transakcie.

ABSTRACT

KAŇUCH, Andrej: Practical usage of blockchain technology. [Bachelor Thesis] / Andrej Kaňuch - University of Economics in Bratislava. Faculty of business informatics; Department of Applied Informatics. - Tutor Ing. Igor Bandurič, PhD. - Degree: Bachelor. Bratislava: Faculty of Economics, EUBA, 2017

Bachelor work is focused on understanding the Blockchain technology. This technology is becoming an interest for big companies and for fans of cryptocurrencies all over the world. This special kind of distributed ledger book finds a way to the commonplace human activities. The work contains principles of operation and areas in which Blockchain can be used.

Keywords: Blockchain, internet, cryptocurrencies, security, finance, transactions.

OBSAH

ZOZNAM TABULIEK A ILUSTRÁCIÍ 9
ZOZNAM TABULIEK A ILUSTRÁCIÍ 10
ZOZNAM SKRATIEK....11
1. Súčasný stav riešenia problematiky 13
1.1. Blockchain....14
1.1.1. Blok....14
1.1.2. Genezis blok....15
1.1.3. Štruktúra bloku....15
1.1.4. Štruktúra hlavičky bloku....16
1.2. Hash funkcia....16
1.2.1. Cieľ použitia hash funkcie 17
1.2.2. Podmienky hashovacej funkcie....17
1.2.3. Hash funkcia v Bitcoine....19
1.3. Časová pečiatka 19
1.4. Proof of Work a Proof of Stake Koncept 20
1.4.1. Proof of Work 20
1.4.2. Proof of Stake....21
1.5. Transakcie....21
1.5.1. Anonymita transakcií 22
1.5.2. Sieť transakcií 23
1.5.3. Overenie transakcie....23
1.6. Adresy....26
1.7. Peňaženka 26
1.7.1. Funkcie peňaženky....27
1.7.2. Typy peňaženiek 28
1.8. Príklady praktického využitia Blockchainu....29
1.8.1. Potreba smart kontraktov 29
1.8.2. Tokeny 30
1.8.3. Hudba....31
1.8.4. Verejná správa identít 31
1.8.5. E-governance hlasovací systém 31
1.8.6. Zdravotníctvo....32
1.9. Ethereum blockchain 32

1.9.1. Blockchain a Ethereum architektúra 33
1.9.2. Ethereum Virtual Machine 34
1.9.3. Ether 35
1.9.4. Gas 35
1.9.5. Smart kontrakty 36
2. Cieľ a metodika práce 38
3. Výsledky práce 39
3.1. Základné informácie pre tvorbu smart kontraktov 39
3.2. Remix IDE 40
3.3. Typy dát 41
3.4. Viditeľnosť funkcií a premenných. 42
3.5. Konštruktor smart kontraktov 42
3.6. Inštalácia a spustenie Ethereum TestRPC 42
3.7. Inštalácia Web3.js 44
3.8. Vytvorenie UI 45
3.9. Použitie Web3.js na prepojenie a interakciu so smart kontraktmi 47
3.10. Vytvorenie smart kontraktu 48
3.11. Zasielanie hlasov 50
4. Diskusia a záver 54
5. Zoznam použitej literatúry 56
Príloha 58

ZOZNAM TABULIEK A ILUSTRÁCIÍ

Obrázok 1. Premena vstupov na výstupy pomocou hash funkcie 17
Obrázok 2. Zobrazenie nadväznosti blokov.....19
Obrázok 3. Merkle tree root.....25
Obrázok 4. Bitcoin adresa v Blockchaine.....26
Obrázok 5 Zobrazenie REMIX IDE 40
Obrázok 6 Zobrazenie názvu a adresy kontraktu.....41
Obrázok 7. Detailné zobrazenie kontraktu.....41
Obrázok 8. Overenie správneho nainštalovania Node,js a npm. 43
Obrázok 9. Inštalácia TestRPC pomocou npm. 43
Obrázok 10. Zobrazenie účtov a privátnych kľúčov.....44
Obrázok 11. Vytvorenie a presunutie sa do projektového priečinka. 44
Obrázok 12. Vytvorenie package.json pomocou npm. 45
Obrázok 13. Inštalácia web3.js pomocou npm. 45
Obrázok 14. Zobrazenie hlavičky v index.html.....45
Obrázok 15. DOM štruktúra stránky 46
Obrázok 16. Zobrazenie potrebných skriptov.....46
Obrázok 17. Interakcia so smart kontraktmi.....47
Obrázok 18. Definovanie Ethereum účtu.....47
Obrázok 19. Zistenie ABI pomocou REMIX IDE.....48
Obrázok 20. Pomenovanie triedy v Solidity. 48
Obrázok 21. Vytvorenie Mappingu vo vnútri kontraktu. 48
Obrázok 22. Vytvorenie poľa kandidátov.....48
Obrázok 23. Vytvorenie konštruktora.....49
Obrázok 24. Inkrementácia hlasov o jeden hlas 49
Obrázok 25. Funkcia ukladajúca dosiahnutý počet hlasov. 49
Obrázok 26. Overenie počtu kandidátov 50
Obrázok 27. Vytvorenie inputu pre odosielanie hlasov. 50
Obrázok 28. Zobrazenie tlačidla pre odoslanie hlasov. 50
Obrázok 29. Zobrazenie objektu funkcií. 51
Obrázok 30. Zoznam kandidátov a im priradených ID.....51
Obrázok 31. Odoslanie hlasov do Blockchainu. 52
Obrázok 32. Cyklus pre zobrazenie počtu hlasov hneď po načítaní stránky. 52

ZOZNAM TABULIEK A ILUSTRÁCIÍ

Tabuľka 1. Ukážka dátového rozdelenia hlavičky. 15
Tabuľka 2. Štruktúra bloku 16

ZOZNAM SKRATIEK

API	Application Programming Interface
ABI	Application Binary Interface
DOM	Document Object Model
FPV	Full Payment Verification
IBM	International Business Machines
ICO	Initial Coin Offering
IDE	Integrated Development Environment
IOT	Internet of Things
IPO	Innitial Public Offering
MD5	Message-Digest Algorithm 5
NPM	Node Package Manager
PoS	Proof of Stake
SHA-1	Secure Hash Algorithm 1. generation
SHA-256	256 bitový Secure Hash Algorithm 2. generation
SPV	Simplified Payment Verification
TxIn	Transaction input
TxOut	Transaction output
VM	Virtual Machine

ÚVOD

Bitcoin má v poslednej dobe dominantné postavenie v oblasti krypto investícií. A aj keď je pravda, že Bitcoin bol prvý, najväčší a najvýznamnejší míľnik , nie je to jediná technológia ktorá by nás v tomto sektore mala zaujímať. Inovácia ktorá by nás mala zaujímať a na ktorej je postavený Bitcoin je Blockchain. Bitcoin nie je Blockchain, jedná sa o dve samostatné veci, ale Blockchain je tým, čo poháňa Bitcoin. Spolu s kryptografickou ťažbou Blockchain umožňuje decentralizáciu kryptomeny. Využitie Blockchainu je v dnešnej dobe vítaná technológia. Dokonca aj medzi tými, ktorí si myslia, že Bitcoin ako kryptomena predstavuje bublinu, Blockchain sa stále považuje za prelom vo svete internetu. Všetka evidencia v súčasnosti má v sebe znaky korupcie a spolieha sa na ústredný orgán ktorý ma povolenie potvrdiť pravdivosť danej informácie, ale Blockchain umožňuje aby sa kompletná evidencia a jej história distribuovali, kopírovali a uchovávali v neporušiteľnom stave. Zamyslime sa nad hoci akým záznamom o čomkoľvek, čo by teoreticky mohlo byť potvrdené so 100% istotou. Mohli by sme vidieť, kedy bolo niečo vyhlásené, napísané alebo potvrdené, kedy boli peniaze poslané, prijaté atď. A to všetko bez spoliehania sa na inštitúciu. To je dôvodom prečo je Blockchain taký populárny a čo prinieslo toľko pozornosti a investícií do tohto sektora, ale tento nečakaný objem priniesol obrovské množstvo neistoty a volatility. Myslím si , že existuje obrovské množstvo prípadov použitia Blockchainu, ktorý je stále podceňovaný a pravdepodobne zatienený Bitcoinom a menovým aspektom Blockchainu v tejto dobe. Jednoduchý príklad použitia je v lodnej doprave a distribúcii tovaru. Medzinárodné námorné spoločnosti, výrobné fabriky a sociálne siete majú obrovské množstvo údajov, ktoré ale nie sú nedotknuteľné. Pravdepodobne údaje a história sú uložené v databázach SQL alebo iných rozsiahlych databázach, ktoré by mohli byť ľahko napadnuté, ich obsah zmenený alebo odstránený. Tieto spoločnosti utrácajú obrovské množstvo financií, aby sa daný skutok nestal, ale stále je tu možnosť hrozby. Blockchain by mohol umožniť aby sa spotrebovalo menej výdavkov na bezpečnosť dátových sietí, ako aj riešenie, ktoré vedie záznamy informácií rôzneho typu.

1. SÚČASNÝ STAV RIEŠENIA PROBLEMATIKY

Napriek tomu, že Blockchain technológia spôsobila vznik Bitcoinu v roku 2009, stále po deviatich rokoch existencie, sme na začiatku využitia tejto technológie. Inovatívne riešenia, ktoré majú potenciálny spoločenský dosah, sa skúmajú každý deň pričom vždy sa objavuje pozitívny dopad pre spoločnosť . Vzhľadom na pokles spotrebiteľskej dôvery posledných rokov v bankovom sektore, Blockchain by mohol byť riešením aby sa to dalo napraviť. Zatiaľ bol Blockchain väčšinou používaný na tvorbu kryptomien. Ethereum nadácia začala nový vývoj v tejto oblasti implementáciou generického programovateľného Blockchainu , ktorý je použiteľný v širokej škále aplikácií. Týmto sa otvára cesta využitia nie len v profitabilných oblastiach. IBM a Samsung dokonca experimentujú s použitím Blockchainu na napájanie internetu vecí (IoT). Technológia Blockchain je sľubnou technológiou ktorá môže byť v budúcnosti použitá , pretože je v súčasnosti pod radarom mnohých subjektov vo finančnom sektore, preto je zaujímavé preskúmať oblasti nových možností, ktoré daná technológia ponúka.

1.1. Blockchain

Blockchain sa stal novým poľom pôsobností kapitálových spoločností, ktoré následne priťahujú pozornosť bánk, vlád a iných obchodných spoločností. Nedávne pokusy súvisiace s Blockchainom zahŕňali oficiálne Blockchainy vytvorené spoločnosťami ako Fadada.com a Microsoft ,taktiež sme sa mohli stretnúť s Blockchainom vytvoreným pre sledovanie potravín od spoločnosti Walmart a IBM. Blockchain sa stáva najvýraznejším vynálezom od čias vzniku internetu. Zatiaľ čo internet spája svet, aby umožnil nové obchodné modely založené na online obchodných procesoch, Blockchain pomôže efektívnejšie vyriešiť otázku dôvery prostredníctvom počítačov pripojených k sieti.

Vzhľadom na to, že Blockchain bol pôvodne vytvorený Satoshim Nakamotom v roku 2008 ako hlavná zložka na podporu transakcií digitálnej meny Bitcoin, Blockchain je známy ako verejná účtovná kniha pre všetky transakcie a vyriešila problém dvojitého výdavku kombináciou peer-to-peer technológie s kryptografiou a verejným kľúčom. Doslova Blockchain môžeme chápať ako reťazec blokov ktorý obsahuje informácie, ktoré napríklad zaznamenávajú Bitcoinové transakcie. Samozrejme, existuje prísny súbor pravidiel, ktoré určujú, ako overiť platnosť bloku a uistiť sa, že blok nebude zmenený alebo vymazaný. **Algoritmy a výpočtová infraštruktúra na vytváranie, vkladanie a používanie blokov sa považujú za blockchain technológiu**. (Zhao, 2016)

Každých 10 minút sa blockchain neustále zväčšuje pridaním nových blokov do reťazca. Baníci (Mineri) to robia z dôvodu zaznamenania najnovších transakcií. Všetky bloky v Blockchaine sú v chronologickom poradí. Každý uzol (miner) má kópiu Blockchainu, ktorú si automaticky stiahne miner pri vstupe do Bitcoin siete. Všetky informácie o všetkých transakciách, ktoré boli kedy vykonané, sú zaznamenané v Blockchaine. (Swan, 2015)

1.1.1. Blok

Údaje o transakciách sú natrvalo zaznamenané v súboroch nazývaných bloky. Môžu sa považovať za jednotlivé dokumenty kde sú zaznamenané zmeny o stave vlastníctva nehnuteľnosti alebo ako účtovná kniha ktorá obsahuje akciové transakcie.

Bloky sú časovo usporiadané do lineárnej sekvencie (známej aj ako blockchain). Nové transakcie sú neustále tvorené minermi do nových blokov, ktoré sa pridajú ku koncu reťazca a nemôžu byť nikdy zmenené alebo odstránené po prijatí sieťou.

1.1.2. Genezis blok

Genezis blok nazývame ten blok ktorý bol vytvorený ako prvý v celom Blockchaine, daní blok funguje ako bezpečnostný koreň pre každý uzol v Blockchaine. Prvý Genezis blok na svete obsahuje skrytú správu v transakcii - hovorí *" The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."* Je to poznámka od Satoshi-ho Nakomota zakladateľa Bitcoinu, táto správa sa vyznačuje dvomi zaujímavosťami- označuje čas prvého bloku a po druhé ekonomicky poukazuje na problémy so súčasnými vládnymi bankami, ktoré sú príliš veľké na to, aby zlyhali a o probléme morálneho hazardu. (Morabito, 2017)

1.1.3. Štruktúra bloku

Blok predstavuje konštrukciu dátového kontajneru, ktorý agreguje transakcie do Blockchainu. Blok sa skladá z hlavičky obsahujúcej metadáta, po ktorej nasleduje zoznam transakcií ktoré tvoria väčšinu veľkosti bloku. Hlavička bloku predstavuje 80 bajtov, pričom priemerná transakcia má veľkosť najmenej 250 bajtov. Priemerný blok obsahuje viac ako 500 transakcií. (Antonopoulos, 2015)

Veľkosť	**Názov**	**Popis**
80 bajtov	Hlavička bloku	Obsahuje ďalšie dátové kontajnery
1-9 bajtov	Počítanie transakcií	Koľko transakcií nasleduje
-	Transakcie	Počet transakcií zaznamenaných v tomto bloku

Tabuľka 1. Ukážka dátového rozdelenia hlavičky.
(Zdroj: Autor)

1.1.4. Štruktúra hlavičky bloku

Veľkosť	Názov	Popis
4 bajty	Verzia	Číslo potrebné pre sledovanie aktualizácie protokolu
32 bajtov	Hash predošlého bloku	Odkaz na hash predchádzajúceho(rodičovského) bloku v reťazci
32 bajtov	Merkle Root	Hash koreňového Merkle tree nachádzajúci sa v transakciách konkrétneho bloku
4 bajty	Časová pečiatka	Približný čas vytvorenia bloku
4 bajty	Obtiažnosť	Obtiažnosť Proof-of-work algoritmu konkrétneho bloku
4 bajty	Náhodné číslo(nonce)	Číslo potrebné pre algoritmus proof-of-work

Tabuľka 2. Štruktúra bloku
(Zdroj: Autor)

Hlavička bloku pozostáva z troch dátových kontajnerov obsahujúcich metadáta. Prvý obsahuje hash odkaz na predchádzajúci blok, tento hash spája aktuálny blok s predchádzajúcim blokom v reťazci blokov. Druhý súbor metaúdajov, obsahuje konkrétnu obtiažnosť, časovú pečiatku a náhodné číslo (nonce) ktoré sa používa pri proof-of-work algoritme. Tretí súbor metadát obsahuje Merkle tree root, je to dátová štruktúra používaná na efektívne zhrnutie všetkých transakcií v bloku.

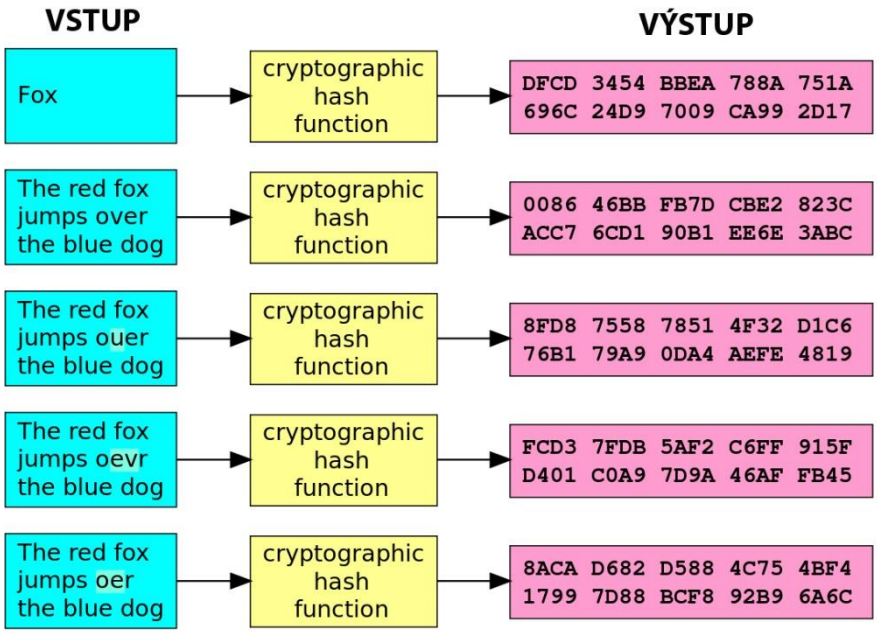
1.2. Hash funkcia

Je to funkcia používaná na transformovanie vstupu dát rôznej veľkosti na výstupný reťazec s určitou fixnou dĺžkou. Tento reťazec nazývame heš(angl. Hash),odtlačok vstupných dát. Hash dĺžka závisí od typu hash funkcie, MD5 je dlhý 128 bitov –32 znakov, SHA-1 je dlhý 160 bitov -40 znakov. Z dôvodu vyššej bezpečnosti sa používa tá hashovacia funkcia ktorá má výstup 160 a viac bitov. V konkrétnejšom príklade sa táto metóda dá použiť v prípade ak sekvencia písmen

ľubovoľnej dĺžky predstavuje vstup - čo nazývame reťazec - a vráti postupnosť písmen s pevnou dĺžkou. Či je vstupný reťazec jedno písmeno, slovo, veta, výstup bude mať vždy rovnakú dĺžku. Bežným používaním tohto typu hashovej funkcie je uloženie hesiel. (MuniCZ, 2017)

1.2.1. Cieľ použitia hash funkcie

V distribuovanom systéme peer-to-peer je potrebné pracovať s obrovským množstvom údajov o transakciách. V dôsledku toho je potrebné ich jednoznačne identifikovať a porovnávať ich čo najrýchlejšie a čo najjednoduchšie.



Obrázok 1. Premena vstupov na výstupy pomocou hash funkcie (Zdroj:https://en.wikipedia.org/wiki/File:Cryptographic_Hash_Function.svg)

Preto je cieľom identifikovať údaje o transakciách a prípadne aj akýkoľvek druh dát jedinečným spôsobom prostredníctvom digitálnych odtlačkov. (Drescher, 2017)

1.2.2. Podmienky hashovacej funkcie

Rýchle poskytovanie hash hodnôt pre rôzne dáta- Táto vlastnosť je kombináciou dvoch vlastností. Po prvé, hash funkcia je schopná vypočítať hodnoty hash pre všetky druhy údajov. Po druhé, hash funkcia je schopná vykonať výpočet

rýchlo. Tieto vlastnosti sú dôležité, pretože je nežiadúce aby funkcia hash trvala dlhší čas.

Deterministickosť-Deterministický znamená, že hash funkcia poskytuje rovnaké výsledné hodnoty pre identické vstupné údaje. To znamená, že akékoľvek pozorované rozdiely v hash hodnote musia byť výlučne spôsobené nezrovnalosťami vstupných údajov. Nie interných procesov funkcií hash.

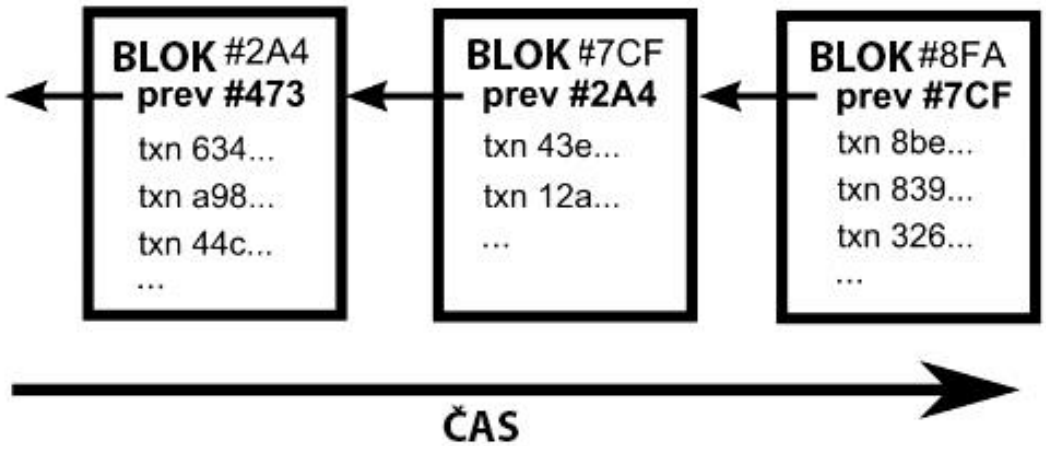
Pseudonáhodnosť- Pseudonáhodnosť znamená, že hodnota vrátená funkciou hash sa mení nepredvídateľne pri zmene vstupných údajov. Aj keď vstupné dáta boli zmenené iba trochu, výsledná hodnota hash sa bude líšiť. Inak povedané, hodnota hash zmenených údajov musí byť vždy nepredvídateľná. Výslednú hodnotu by nemalo byť možné odhadnúť na základe vstupných údajov.

Jednosmerná funkcia- Jednosmerná funkcia neposkytuje žiadny spôsob, ako získať vstupné hodnoty pomocou výstupov. Jednosmerná funkcia znamená taktiež, že nemôže byť použitá iná cesta k vstupu . Nie je možné obnoviť pôvodné vstupné údaje založené na hash hodnote. To znamená, že hodnoty hash nepovedia nič o obsahu vstupných údajov rovnakým spôsobom ako odtlačok prsta vám nepovie nič o osobe, ktorá vytvorila daní otlačok prsta .Jednosmerné funkcie nie sú invertovateľné.

Odolnosť voči kolízii-Funkcia hash sa nazýva odolná voči kolízii, ak je veľmi ťažké alebo nemožné nájsť dve alebo viac odlišných údajov, pre rovnakú hodnotu hash. V tomto prípade môžeme považovať hodnoty hash vytvorené funkciou za jedinečné a preto použiteľné na identifikáciu údajov. V situácií ak by sme získali rovnakú hodnotu hash pre rôzne údaje , čelili by sme hashovej kolízii. Znamená to, ako keby máme dvoch ľudí s rovnakými odtlačkami prstov. (Phillip Rogaway, 2004)

1.2.3. Hash funkcia v Bitcoine

Hash funkcie sú súčasťou blok algoritmu, ktorý sa používa na zápis nových transakcií do Blockchainu prostredníctvom ťažby kryptomien. Pri ťažbe Bitcoinov sú vstupy pre funkciu všetky najnovšie transakcie, ktoré ešte neboli potvrdené (spolu s ďalšími vstupmi týkajúcimi sa časovej značky a odkazom na predchádzajúci blok).



Obrázok 2. Zobrazenie nadväznosti blokov

(Zdroj: https://blog-archive.bitgo.com/the-challenges-of-block-chain-indexing)

Pri ťažbe Bitcoinov sú vstupy pre funkciu všetky najnovšie transakcie, ktoré ešte neboli potvrdené (spolu s ďalšími vstupmi týkajúcimi sa časovej značky a odkazom na predchádzajúci blok).

1.3. Časová pečiatka

Časová pečiatka je sekvencia znakov alebo kódovaných informácií určujúcich, kedy sa vyskytla určitá udalosť v našom prípade kedy vznikol určitý blok, zvyčajne sa v časovej pečiatke udáva dátum a čas dňa, niekedy je to dokonca presne na zlomok sekundy. Dôveryhodné časové označenie je proces bezpečného sledovania času vytvorenia a úpravy dokumentov. To znamená, že nikto , ani vlastník dokumentu by nemal mať možnosť ho zmeniť, akonáhle bude zaznamenaný. Je dôležité, aby integrita časovej pečiatky nebola nikdy ohrozená. Blockchain zmenil všetko, tým že umožnil bezpečne poskytovať informácie o časovej pečiatke decentralizovaným spôsobom s použitím digitálneho podpisu a hashu. Akonáhle sú digitálne dáta zašifrované a výsledný hash je zabudovaný do Blockchainu ,časová pečiatka slúži ako bezpečný dôkaz o presnom čase, kedy údaje vznikli. (David LEE Kuo Chuen, 2017)

1.4. Proof of Work a Proof of Stake Koncept

Motivácia za overovacím procesom je dosiahnutie dohody o obsahu v distribuovanej knihe (Blockchaine). Kontroly založené na dohode sú : **decentralizovanosť** (t.j. Informácie sú vložené do Blockchainu automaticky, bez potreby tretej strany) a **automatizovaný proces**. Tieto dva nástroje sa najčastejšie používajú na vybudovanie dohody, ktoré sa potom vložia do Blockchainu.

1.4.1. Proof of Work

Decentralizovaná hlavná kniha (Blockchain) je základnou štruktúrou databázy používanej pre transakcie s digitálnymi menami vrátane Bitcoin transakcií, dôvodom je že slúži ako úložný priestor pre celú históriu transakcií. Je nanajvýš dôležité poznamenať, že fungovanie systémov digitálnej meny by malo zahŕňať bezpečnostné prostriedky proti útokom na Blockchain. Ak sa útočník rozhodne minúť určitú sumu peňazí a potom sa pokúsi zvrátiť túto konkrétnu transakciu, útočník by mohol zverejniť svoju vlastnú verziu Blockchainu, v ktorej táto transakcia nie je zahrnutá, potom by účastníci nemali žiadnu formu prístupu k správnej verzii Blockchainu ktorá existovala pred útokom.

Zabezpečenie siete Bitcoin závisí od protokolu zabezpečenia siete nazvaného "proof of work" (PoW). Tento protokol zabezpečenia siete je časť údajov, ktoré sú nákladné na vytvorenie. Z toho vyplýva, že na vykonanie špecifickej úlohy tento protokol vkladá dodatočnú obtiažnosť. Pri zohľadnení Bitcoinu by sa malo poznamenať, že v rámci určitého časového obdobia sa každá uskutočnená transakcia zaznamená a uloží do Bitcoin Blockchainu. Blok sa potom vysiela do všetkých uzlov v rámci siete Bitcoin. V tomto prípade sa používa schéma Hashcash Proof of Work. Táto Hashcash Proof of Work schéma bola zavedená v roku 1997 Adamom Backom .

V rámci tejto Hashcash schémy každý účastník pridá do bloku časť údajov nazvanú "nonce", čím vytvorí blok + nonce. Tento blok + nonce sa potom odoberie a umiestni do algoritmu nazývaného "algoritmus hash". Tento algoritmus hash pozostáva z hashu, ktorý zodpovedá niektorým konkrétnym predpokladom. Algoritmus hash potom prináša komplexnú matematickú kompozíciu, v ktorej sa každý účastník pokúša poskytnúť riešenie použitím funkcie SHA-256 (Secure Hash Algoritmus) hash funkcie.

Akonáhle je riešenie poskytnuté matematickému výpočtu účastníkom, potom sa predpokladá, že sú splnené konkrétne predpoklady dokladu Proof of Work schémy, následne vzniká "blok + nonce + hash". Hneď, ak k tomu dôjde, potom blok + nonce + hash je zahrnutý do Bitcoin Blockchainu a je vysielaný do každého z účastníckych uzlov v sieti. Okrem toho protokol (proof of work) funguje tak, že fyzicky obmedzené zdroje pomáhajú sieti. Tieto fyzicky obmedzené zdroje sú:

- hardvér potrebný na spustenie matematických výpočtov.
- elektrický výkon potrebný na prevádzku hardvéru.

Z toho vyplýva, že používanie protokolu (Proof of Work) je veľmi náročný na zdroje. V dôsledku toho boli vytvorené mnohé podobné systémy, ktoré nie sú založené na nákladných výpočtoch, pričom "Proof of Stake" je jeden z nich.

1.4.2. Proof of Stake

Systém Proof of Stake (PoS) slúži ako alternatíva k Proof of Work. Proof of Stake je schéma založená na menej nákladných výpočtoch. Z toho vyplýva, že Proof of Stake schéma nie je založená na nákladných výpočtoch v porovnaní s Proof of Work. Skôr ako od závislosti na obmedzených zdrojov (nákladné výpočty), Proof of Stake závisí od subjektov, ktoré majú podiel/vlastníctvo v sieti. Inými slovami, môžeme povedať, že zdrojom, na ktorom závisí bezpečnosť siete, je vlastníctvo samotnej kryptomeny. Na overenie pravosti a prijatí transakcie (či už poplatky za transakciu alebo presun kryptomien) musí byť určité množstvo mincí vo vlastníctve minera. Pravdepodobnosť, že miner je úspešný pri vytváraní nového bloku, závisí od množstva mincí vlastneného minerom ,z toho vyplýva že je nezávisle od výpočtového výkonu a kedykoľvek sa použije schéma Proof of Stake. Preto je možné schvaľovať transakcie každú minútu. Proof of Stake schéma má viac výhod oproti Proof of Work. Jedna výhoda PoS nad PoW je dosť výrazná a je to nízka latencia PoS. Jedným z problémov, s ktorým sa stretáva Proof of Work, je otázka centralizácie, pretože zainteresované strany s veľkými podielmi by sa mohli pokúsiť o nadvládu nad sieťou. (Morabito, 2017)

1.5. Transakcie

Transakcia predstavuje presun hodnoty kryptomeny, v tomto prípade Bitcoinov, ktorá sa následne odošle do siete a chronologicky sa akumuluje do blokov. Transakcia zvyčajne odkazuje na predošlý výstup transakcie. Transakcie nie sú anonymné ,takže

je možné prehľadávať a nahliadnuť do každej transakcie uloženej v Blockchaine. Akonáhle sa transakcie dostanú do stavu keď po nich nasleduje väčšie množstvo ďalších transakcií môžeme danú transakciu považovať za nezvrátiteľnú.

Transakcie ktoré boli doposiaľ vytvorené sú viditeľné v Blockchaine a môžu byť prezerané pomocou hexadecimálneho editora. Blockchain prehliadač je web-stránka kde si môžeme prezerať konkrétne transakcie ktoré sú zaznamenané v čitateľných vzťahoch. Od tohto sa odvodzuje užitočnosť pre možnosť overovania platieb. (Bit181)

Transakcie sa skladajú zo zoznamu transakčných vstupov (TxIn) a zo zoznamu transakčných výstupov (TxOut). Každý výstup transakcie (TxOut) obsahuje dva typy údajov: čiastku a adresu príjemcu. Adresa je odvodená z verejného kľúča. Takže len majiteľ súkromného kľúča môže odomknúť finančné prostriedky uložené v TxOut. Ak chceme odomknúť finančné prostriedky, vlastník privátneho kľúča musí podpísať transakciu, ktorá bude odosielať finančné prostriedky na novú adresu. Vstup transakcie (TxIn) obsahuje odkaz na predchádzajúci výstup transakcie a podpis, ktorý dokazuje, že finančné prostriedky z predchádzajúcich referencií TxOut môžu byť vynaložené. Tento podpis sa musí vykonať pomocou privátneho kľúča priradeného k verejnému kľúču v adrese. Ak sa podpis nezhoduje, transakcia sa považuje za neplatnú a sieť ju zruší. Transakcia zoskupuje niekoľko TxIn a TxOut (aspoň jedno z nich). Účel transakcie je distribúcia finančných prostriedkov zo vstupov na výstupy . Tieto výstupy nesmú byť použité, inak je transakcia neplatná. Ak má byť transakcia platná, suma súm vstupov musí byť vyššia ako suma súm výstupov. Rozdiel medzi vstupmi a výstupmi, ak existujú, je poplatok za transakciu. Poplatok za transakciu získavajú mineri, ktorí zahŕňajú transakciu do bloku. (Franco, 2015)

1.5.1. Anonymita transakcií

S cieľom analyzovať, ako môžu kombinácie služieb poskytovať zvýšenie anonymity pre používateľov Bitcoinu, začneme definovaním slova anonymita. Anonymita znamená, že subjekt v rámci súboru iných entít nie je možné identifikovať. (Talking, 2010)

V komunikačnej sfére môže byť súbor anonymity rozdelený medzi anonymitu

odosielateľa a anonymitu príjemcu nastavený. Anonymita v tomto kontexte znamená, že útočník nemôže rozhodnúť, či odosielateľ komunikuje s určitým príjemcom. Hoci systém môže dosiahnuť vysokú anonymitu v globálnom meradle úroveň anonymity určitého subjektu v systéme môže byť nízka, ak má útočník k dispozícii detailné informácie umožňujúce prelomiť anonymitu.

1.5.2. Sieť transakcií

V Bitcoin sieti nie je používateľ fyzickým vlastníkom Bitcoinov .Vlastníctvo Bitcoinov je uložené v Blockchaine ako výstupy transakcie, ktoré sa vzťahujú na adresu príjemcu. Transakcia predstavuje platbu, ktorá je podpísaná súkromným kľúčom predchádzajúceho vlastníka určitej sumy, ktorý teraz chce zmeniť vlastníctvo mincí verejnému kľúču uvedenému v transakcii.(t.j. inému používateľovi). Množstvo Bitcoinov, ktoré vlastní používateľ, môže byť vypočítané ako súčet všetkých nevyužitých transakcií patriacim k jeho adresám. Bitcoin transakcia obsahuje záznam jednej alebo viacerých predchádzajúcich transakcií, tie predstavujú vstup. Je potrebné vyčerpať celú kumulatívnu hodnotu vstupných transakcií, v opačnom prípade by to znamenalo stratu Bitcoinov. Preto má štandardná transakcia zvyčajne dve výstupné adresy, z ktorých jedna patrí odosielateľovi ktorý dostane zmenu transakcie, druhá patrí príjemcovi platby. Z týchto informácií je možné vytvoriť graf transakcií. (Bit18)

1.5.3. Overenie transakcie

Hlavnou výhodou kryptomeny ako takej je, že nie je potrebné dôverovať jednotlivcovi lepšie povedané tretej strane, alebo ústrednej inštitúcii. Programy ktoré sa nachádzajú v peňaženke určitej kryptomeny však musia byť schopné overiť či prijaté transakcie sú platné. V tejto súvislosti je dôležité rozlišovať medzi Blockchainom (nemenným verejným dokumentom, ktorý uvádza všetky platné transakcie určitej kryptomeny) a niekoho kópiou Blockchainu, ku ktorému máme prístup. Prvá je abstraktná koncepcia, zatiaľ čo druhá je praktická realita. Pri pripojení peňaženky do siete kryptomeny sa peňaženka pripája k viacerým uzlom, ktoré odosielajú našej peňaženke transakčné dáta ale nemôžeme predpokladať, že údaje sú platné. Platná transakcia musí mať: správny digitálny podpis a použiť kryptomeny(napr. Bitcoiny), ktoré vznikli ako odmena z vyriešeného bloku a ešte neboli použité niekde inde. Všetky peňaženky môžu overiť prvú podmienku s úplnou istotou, ale druhá podmienka sa rieši

rôznymi stupňami takzvanej istoty v závislosti od návrhu peňaženky.

Peňaženky môžu overiť transakcie buď tým, že si ponechajú svoju vlastnú kompletnú kópiu Blockchainu, ktorá sa označuje ako **full payment verification** (FPV) alebo pomocou skrátenej verzie, ktorá sa nazýva **simplified payment verification** (SPV). (Wilmer, 2014)

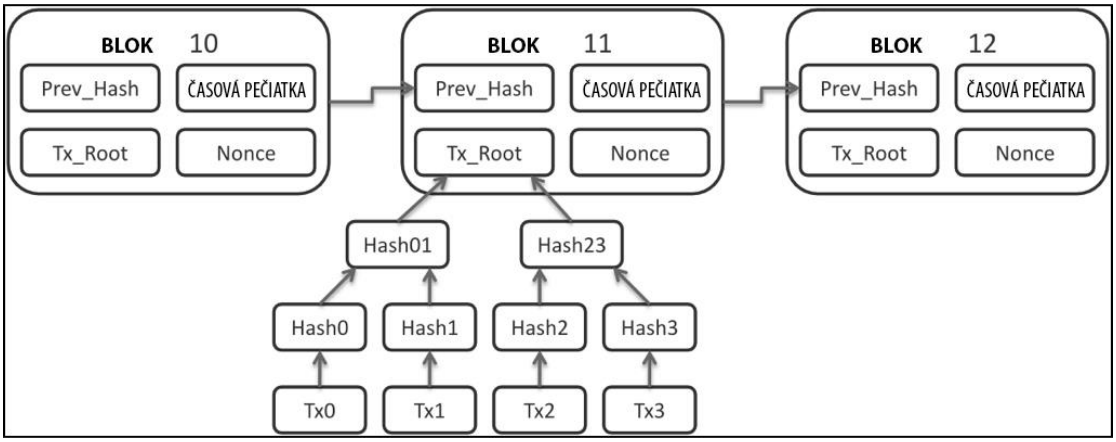
Full payment verification (FPV): Full payment verification, nazývané aj hrubé alebo ťažké peňaženky, vyžadujú kompletnú kópiu Blockchainu. Tieto typy môžu overiť, či kryptomena použitá v transakcii pochádza z vyťaženého bloku skenovaním odzadu, transakciu po transakcii v Blockchaine až po ich prvotný blok (peňaženky sú schopné skontrolovať, či neboli náhodou dané Bitcoiny použité dvakrát). Tieto peňaženky sú často aktívnym účastníkom v sieti Bitcoin v tom, že nielen spracovávajú transakcie používateľa, ale tiež overujú a odovzdávajú transakcie iných ľudí (v v týchto prípadoch sa počítače s takýmito programami nazývajú plné uzly). Všetci Bitcoin mineri predstavujú taktiež plné uzly (to znamená, že potrebujú úplnú kópiu Blockchainu). (Ghassan Karame, 2016)

Jedným z problémov pri full payment verification je, že sú veľmi náročné na zdroje a trvá dlho, kým sa inicializujú. Bitcoin blockchain vo svojom 5. roku existencie , bol nebol väčší než 15 GB a zahŕňal 35 miliónov transakcií
(po 10. rokoch existencie, pravdepodobne bude 100-krát väčší).Inštalácia full payment Bitcoin peňaženky môže trvať niekoľko dní (závisí to na rýchlosti sťahovania) na stiahnutie celého Blockchainu. Stiahnutie Blockchainu vyžaduje pripojenie k iným plným uzlom a podmienku zistiť, ktorý blockchain má najväčší proof-of-work. Pre notebooky a iné domáce zariadenia, na ktorých beží FPV peňaženka môže overenie platby pôsobiť nepríjemne z dôvodu veľkého objemu dát. Našťastie existuje spôsob, ale je potrebné urobiť mierny kompromis v dôvere, ale na oplátku dosiahneme viac výpočtovej efektívnosti na overenie transakcie.

Simplified Payment Verification (SPV): SPV peňaženky, tiež nazývané ako ľahké peňaženky, nie sú schopné skontrolovať platnosť transakcie, skôr môžu skontrolovať, či plné uzly konkrétne mineri potvrdili dané transakcie. Cieľom SPV peňaženky je skontrolovať, či transakcia bola overená minermi a či bola zahrnutá do nejakého bloku

v Blockchaine. Táto metóda funguje spoľahlivo, pokiaľ mineri pridávajú bloky do Blockchainu, konajú pravdivo a povoľujú iba platné transakcie (čo je bezpečný predpoklad, pokiaľ jednotlivec nemá pod kontrolou viac ako 51 percent hashingovej sily siete).

Dômyselnosť SPV závisí od schopnosti overenia prostredníctvom hash funkcie, a tým že transakcia bola zahrnutá do bloku bez toho, aby bolo potrebné sa pozrieť na niektorú z transakcií nachádzajúcich sa v bloku. Aby to bolo možné urobiť, SPV peňaženky si musia stiahnuť hlavičky každého bloku v Blockchaine. Dôležité je, že hlavička obsahuje hash všetkých transakcií v rámci tohto bloku, štruktúrované tak, aby každá peňaženka mohla ľahko skontrolovať, či transakcia patrí konkrétnemu bloku vzhľadom na jej hodnotu hash.

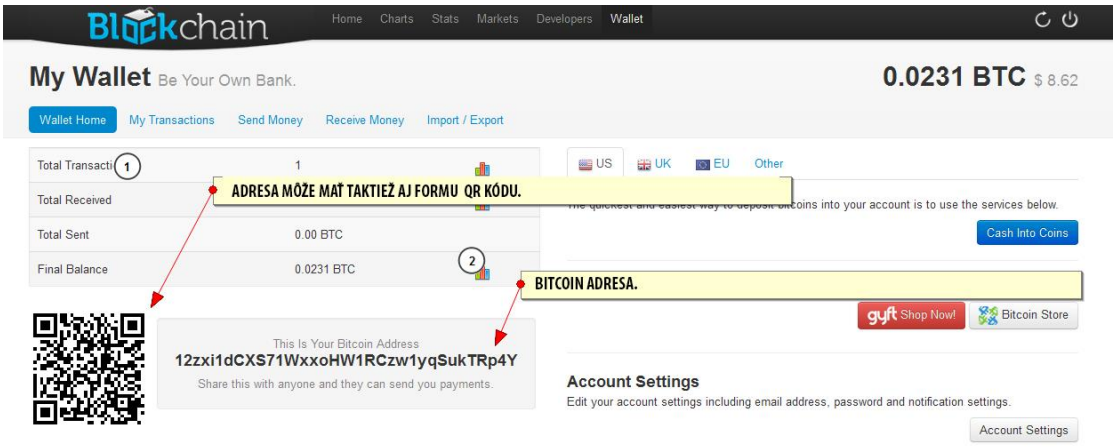


Obrázok 3. Merkle tree root.
(Zdroj: https://blockgeeks.com/guides/what-is-hashing/)

Táto štruktúra hash sa nazýva Merkle tree. Pomocou tohto Merkle tree návrhu môžu SPV peňaženky bezpečne potvrdiť, či transakcie ktoré obdŕžali, boli zahrnuté do Blockchainu bez toho, aby si stiahli celý blockchain. Sťahovanie hlavičiek blokov vyžaduje len zlomok pamäte, ktorá je potrebná pre celý blockchain. Preto môžu byť peňaženky SPV ľahko spustiteľné na mobilných zariadeniach. Bitcoin peňaženka, ktorá používa SPV, môže ponúknuť väčšinu, ale nie všetky bezpečnostné prvky ako FPV peňaženka.

1.6. **Adresy**

V prípade ak vlastníme určité množstva Bitcoinov, tak v skutočnosti ich priamo nevlastníme .To, čo máme, je Bitcoin adresa = (podobné číslu bankového účtu) a kľúč k tejto adrese (ďalšie číslo, ktoré funguje ako kód PIN k vyššie uvedenému číslu). Bitcoinová adresa je uvedená v transakciách v Blockchaine, kľúč je jediná vec, ktorú vlastníme, to je to čo robí Bitcoiny našim majetkom. V prípade ak chceme odoslať Bitcoiny z našej adresy na inú adresu (podobné ako posielanie peňazí cez PayPal), vygenerujeme transakciu, ktorá sa odosiela do siete a pridá sa do ďalšieho bloku transakcií. Akonáhle je daná transakcia v bloku, táto transakcia je verejne viditeľná v Blockchaine navždy. Peňaženka je miesto, kde máme uložené všetky kľúče. Obvykle peňaženka predstavuje program, ktorý vytvára , spravuje adresy a ukazuje množstvo Bitcoinov. Taktiež je možné vygenerovať novú adresu a jej zodpovedajúci kľúč, kedykoľvek uznáme za vhodné. Taktiež je možné si ponechať kľúče na peňaženke ,na počítači pripojenom na internet, na USB kľúči alebo dokonca na papieri. (Gerard, 2017)



Obrázok 4. Bitcoin adresa v Blockchaine
(Zdroj: http://bitkidz.com/tutorial-how-to-open-a-bitcoin-wallet-on-blockchain-info-audio/)

V prípade straty kľúču, všetky naše Bitcoiny sú navždy stratené. Ak sa niekto dostane ku kľúču, môže odcudziť všetky Bitcoiny. V prípade posielania Bitcoinov na neexistujúcu adresu, dochádza k nezvrátiteľnej strate. V prípade odoslania Bitcoinov na nesprávnu adresu, dochádza taktiež k nezvrátiteľnej transakcii.

1.7. **Peňaženka**

Rovnako ako je potrebné vlastniť bankový účet na odosielanie alebo prijímanie finančných prostriedkov, je taktiež potrebné niečo, čo sa nazýva krypto peňaženka pre odosielanie a prijímanie kryptomien napríklad Bitcoinov. Zaujímavé je, že na rozdiel

od tradičných peňaženiek, digitálne peňaženky neuchovávajú žiadnu hodnotu. Pre lepšie pochopenie ako fungujú digitálne peňaženky uvedieme príklad Bitcoin peňaženky.

Bitcoin peňaženka je v podstate softvérový program. Rovnako ako je potrebný e-mailový program ako Gmail na správu e-mailov, Bitcoin peňaženka je potrebná pre správu Bitcoinov. Bitcoin peňaženka ukladá kolekciu súkromných kľúčov a verejných kľúčov. Peňaženka je prepojená s Blockchainom, takže je možné odosielať a prijímať peniaze, ako aj sledovať stav účtu. Celkovo peňaženka funguje ako účtovná kniha kde sú zobrazené všetky naše transakcie. Peňaženka je chránená pred neoprávneným prístupom a je zvyčajne zašifrovaná heslom. Akonáhle si nainštalujeme softvérovú Bitcoin peňaženku, zobrazí sa nám Bitcoin adresa peňaženky. To slúži ako identifikačné číslo, ktoré nás priradí ako člena siete Bitcoin. Súkromný kľúč umožňuje minúť Bitcoiny z tejto našej adresy. Súkromný kľúč je reprezentovaný buď ako reťazec písmen a čísel, alebo ako QR kód .

1.7.1. Funkcie peňaženky

Bitcoin peňaženka jednoducho povedané zhromažďuje všetky naše adresy na jedno miesto. Niektoré ďalšie veľmi dôležité funkcie ktoré peňaženka obsahuje sú:

- Komunikácia s Blockchainom a možnosť používateľa vidieť celkové dostupné finančné prostriedky.
- Vytváranie nových adries na prijímanie nových finančných prostriedkov.
- Generovanie alebo načítanie QR kódov, ktoré reprezentujú adresy, transakcie atď.
- Zasielanie finančných prostriedkov na adresu zvolenú používateľom.
- Sledovanie stavu potvrdenia transakcií a publikovanie transakcií do Blockchainu.
- Vytvorenie zálohy peňaženky a obnovenie zálohy podľa požiadaviek používateľa. (Bennett, 2017)

1.7.2. Typy peňaženiek

V Bitcoinovom svete existujú rôzne typy peňaženiek, ktoré možno použiť na ukladanie súkromných kľúčov. Ako softvérový program poskytujú používateľom aj niekoľko rôznych funkcií na správu a uskutočňovanie transakcií v sieti Bitcoin.

Nedeterministické peňaženky: Tieto peňaženky obsahujú iba náhodne vytvorené privátne kľúče. Kľúčovou úlohou klienta tohto typu je generovanie niekoľko kľúčov vytvorených pri prvom spustení a generovanie kľúčov podľa potreby. Správa veľkého počtu kľúčov je veľmi náročná na proces, ktorý je náchylný na chyby, čo môže viesť ku krádežiam a strate kryptomeny. Okrem toho je potrebné vytvoriť pravidelné zálohovanie kľúčov a vhodne ich chrániť, aby sa zabránilo krádeži alebo strate.

Deterministické peňaženky: V tomto type peňaženky sú kľúče odvodené od hodnoty seedu prostredníctvom hash funkcií. Toto sede číslo je generované náhodne a je obyčajne reprezentované čitateľnými mnemotechnickými kódovými slovami. Mnemonické kódové slová sú definované v BIP39. Táto fráza môže byť použitá pomerne jednoduchšie na obnovenie všetkých kľúčov a spravovanie súkromných kľúčov.

Hierarchické deterministické peňaženky: Sú definované v BIP32 a BIP44, HD peňaženky ukladajú kľúče v stromovej štruktúre odvodenej od seedu. Seed generuje nadradený kľúč (hlavný kľúč), ktorý sa používa na generovanie dcérskych kľúčov. Generovanie kľúčov v peňaženkách HD sa nevytvára priamo, namiesto toho produkuje niektoré informácie (informácie o generovaní súkromných kľúčov), ktoré možno použiť na generovanie sekvencie súkromných kľúčov. Kompletná hierarchia súkromných kľúčov v HD peňaženke je ľahko obnoviteľná, ak je známy hlavný súkromný kľúč. Je to preto, že HD peňaženky sú veľmi ľahko udržiavateľné a sú prenosné.

Papierové peňaženky: Ako naznačuje názov, ide o papierovú peňaženku s požadovaným kľúčom vytlačeným na papieri. To si vyžaduje fyzické uloženie na bezpečné miesto. Papierové peňaženky je možné generovať online od rôznych poskytovateľov služieb, ako napríklad https://bitcoinpaperwallet.com .

Hardvérové peňaženky: Ďalšou metódou je použitie zariadenia odolného proti odcudzeniu kľúčov. Toto zariadenie môže byť vytvorené na mieru alebo v telefónoch s NFC funkciou. Trezor peňaženky sú najčastejšie používané hardvérové Bitcoin peňaženky.

Online peňaženky: Online peňaženky, ako naznačuje názov, sú uložené online a poskytujú sa ako služba zvyčajne cez cloud. Poskytujú webové rozhranie pre používateľov na správu peňaženky a na vykonávanie rôznych funkcií, ako je napríklad vytváranie a prijímanie platieb. Sú ľahko použiteľné, ale to naznačuje, že používateľ dôveruje poskytovateľovi peňaženiek. (Bashir, 2017)

1.8. Príklady praktického využitia Blockchainu

Je zrejmé, že Blockchain technológia je oveľa viac než len Bitcoin. Pre všetky oblasti ako financie, zdravotná starostlivosť, média, vláda a iné odvetvia sa inovatívne využitie objavuje každý deň. Potenciál Blockchainu môže narušiť takmer každý priemysel nejakým spôsobom preto ho nemožno odmietať. V nasledujúcej kapitole si predstavíme oblasti, kde sa začína používať technológia Blockchain.

1.8.1. Potreba smart kontraktov

Vývoj autonómnych skriptov na zvládnutie akéhokoľvek druhu podnikania alebo procesu založeného na digitálnych aktívach uložených v Blockchaine, ktoré môžu spúšťať seba samých, seba samých overiť a samo-obmedziť dohodu medzi dvoma stranami, viedlo k vzniku myšlienky o inteligentných zmluvách. Úlohou blockchain siete je odstrániť potrebu dôveryhodnej tretej strany na vyriešenie ľubovoľných právnych sporov, ktoré môžu vzniknúť medzi účastníkmi zmluvy. Modulárne a opakovateľné inteligentné zmluvy(smart kontrakty) umožňujú vytvárať aplikácie pre špecifické prípady použitia a sú zakódované do Blockchainu na konkrétnej adrese známej ako zmluvná adresa, ktorá je určená v čase nasadenia. Vždy, keď uzly iniciujú činnosť, ktorú kontroluje inteligentná zmluva, transakcia sa odošle na zmluvnú adresu a virtuálny stroj vykoná zmluvný kód pomocou zadania údajov z transakcie. Inými slovami, inteligentné kontrakty sú samo nastavovacie aplikácie, ktoré prebiehajú presne tak, ako sú naprogramované s nulovými prestojmi, cenzúrou alebo interferenciou tretej strany. Inteligentné kontrakty zabraňujú oneskoreniam a výdavkom, ktoré vznikli z fyzických kontraktov. Môžu sa vykonávať nezávisle bez vplyvu akýchkoľvek

externých subjektov a majú potenciál prijať potrebné opatrenia počas udalostí porušenia zmluvy. Vo všeobecnosti sa sľuby zavádzajú do kódu, aby sa vytvorila "nerozbitná" zmluva. Takéto zmluvy budú mať nulovú flexibilitu a budú sa držať svojho motta. Nový programovací jazyk, ako je Solidity, Serpent atď., Je určený na písanie inteligentných zmlúv, ktoré môžu presne špecifikovať výsledky s nulovou nejednoznačnosťou v zmluvných výkladoch a môžu sa zdržať akéhokoľvek porušenia. Tým, že súhlasia s používaním inteligentnej zmluvy, účinne odovzdávajú kontrolu nad aspektom plnenia zmluvných záväzkov voči digitalizovanému procesu, ktoré nemožno odôvodniť alebo ovplyvniť. (S. Asharaf, 2017)

1.8.2. Tokeny

Tokeny si v jednoduchosti môžeme predstaviť ako minoritné projekty fungujúce na Blockchaine kryptomeny (najčastejšie Ethereum). Reprezentujú nejakú hodnotu súkromného subjektu (firmy) a na trhu s kryptomenami sa obchodujú rovnakým štýlom ako akcie na klasickej burze. S tým rozdielom, že spoločnosť nepredáva akcie, ale tokeny. Ich cena nie je vyjadrená finančne, ale vo forme kryptomeny, na ktorej daný token funguje, teda najčastejšie v mene Ethereum. Spoločnosti často využívajú tokeny na svoje počiatočné financovanie, čo označujeme ako ICO – Initial Coin Offering. Ide o ekvivalent IPO – Innitial Public Offering, čo je bežný spôsob financovania firiem vstupujúcich na burzu. ICO sa ale nevzťahuje len na Ethereum. Pokojne sa dá aplikovať aj na blockchain inej kryptomeny.

IPO a tokeny zažili najmä začiatkom tohto roka obrovský boom a mnohé startupy boli financované najmä vďaka tokenom. Aj vďaka tomuto ošiaľu narástla cena kryptomeny Ethereum z niekoľkých dolárov na približne 300 – 400$, ktoré dosahuje dnes. Finanční analytici sa ale obávajú, že ide o bublinu, ktorá by mohla spôsobiť problémy.

Blockchain technológia teda nie je navrhnutá len na rozmach kryptomien a na vytvorenie dôvernej, verejnej a rovnoprávnej siete pre obchodovanie, ale má naozaj rozsiahle využitie, napríklad aj v investovaní, zdravotníctve, reklame, či v ďalších odvetviach. Dnes už dokonca môžete využívať aj online zoznamku založenú na Blockchaine. (18Fe, 2016)

1.8.3. Hudba

Hudobný priemysel už vyvíja technológiu založenú na Blockchaine, ktorá sa používa rôznymi spôsobmi. Existuje niekoľko spoločností, ktoré vyvíjajú aplikácie založené na blokoch, ktoré menia spôsob distribúcie, zdieľania, zakúpenia hudby a spôsobu platenia autorských honorárov za predaj. Peertracks, Uio Music a Mycelia sú len niekoľkými začínajúcimi spoločnosťami, ktoré pracujú na platformách založených na Blockchaine, aby umelci mohli predávať svoju hudbu priamo svojim fanúšikom bez potreby sprostredkovateľa. Spoločnosť Spotify nedávno zakúpila spoločnosť Mediachain, ktorá vyvinula systém založený na Blockchaine, ktorý umelcom umožňuje vytvárať digitálne záznamy pre skladby na Bitcoin Blockchaine. Spoločnosť Spotify má za cieľ využívať blockchain platformu spoločnosti Mediachain na vytvorenie spravodlivejších a transparentnejších platieb umelcom pre ich hudbu.

1.8.4. Verejná správa identít

Riadenie identity využívajúce blockchain technológiu je kľúčovou inováciou, ktorá by mohla vytvoriť cestu pre bezpečnosť a založenie iných priemyselných odvetví. Ak môžeme dôverovať niekomu, kto tvrdí, že je danou osobou , môžeme sa s ním spojiť pre celý rad ďalších aplikácií. Technológia Blockchain rieši mnohé existujúce problémy s digitálnou identitou. V súčasnosti je relatívne jednoduchšie vytvoriť falošné identity alebo ukradnúť online identitu niekoho iného. Heslá nie sú zabezpečené a centralizované databázy sú citlivé na útoky. Po napadnutí centralizovanej databázy môže poskytnúť prístup ku všetkým údajom zákazníka uloženým v systéme. Identifikačné systémy založené na Blockchaine poskytujú digitálne podpisy pomocou kryptografie. Sú jedinečné, nezvratné, bezpečné a je takmer nemožné duplikovať alebo pristupovať k údajom bez povolenia. Identifikácia založená na Blockchaine je v budúcnosti reálnou možnosťou s Estónskou vládou ako príkladom a spoločnosťami ako ShoCard, ktoré už budujú systémy správi identít v Blockchaine. V budúcnosti by sa blockchain mohol používať pri digitálnych identitách, pasoch, vodičských preukazoch, povoleniach na pobyt, osvedčeniach o narodení, sobášnych osvedčeniach a iných formálnych identifikáciách.

1.8.5. E-governance hlasovací systém

Po vybudovaní technológie, ktorá umožňuje digitálnu identitu a digitálne

podpisy, je ľahké overiť totožnosť niekoho iného z množstva ďalších transakcií a akcií vykonaných online. Digitálne hlasovanie je technológia, ktorú sa nepodarilo úspešne implementovať v rôznych krajinách kvôli bezpečnostným rizikám a problémom s ochranou súkromia. Estónsko, Dánsko a Nórsko experimentovali s digitálnym hlasovaním. Jedine Estónsko však úspešne spustilo rozsiahly digitálny hlasovací systém. Dánsko využilo blockchain technológiu na hlasovanie v malom rozsahu s liberálnou alianciou, ktorá je politickou stranou v Dánsku a použila blockchain technológiu v roku 2014 ako hlasovací systém. Vďaka použitiu hlasovacieho systému založeného na Blockchaine by volič mohol skontrolovať, či bol jeho hlas úspešne poslaný, zatiaľ čo sa zachováva súkromie a ukrytie identity. Vďaka tomu by bolo hlasovanie prístupnejšie aj pre mnohých ľudí, čím by sa potenciálne zvýšila účasť na voľbách.

1.8.6. Zdravotníctvo

Blockchain poskytuje distribuovanú verejnú knihu, v ktorej sa pri zmene v jednej knihe aktualizujú všetky ostatné kópie súčasne. Tým sa zabezpečí, že každý má najaktuálnejšie platné údaje zodpovedajúce všetkým kópiám v sieti. To má veľký potenciál byť aplikované v odvetví zdravotníctva. V prípade navštívenia viac ako jedného lekára alebo nemocnice, je zrejmé , že zakaždým, pri navštívite nového lekára alebo nemocnice, ide o obrovské množstvo dokumentov o našej anamnéze, alergii a ďalších lekárskych informáciách, ktoré sme mohli uviesť už niekoľkokrát predtým na iných miestach. Ukladanie týchto informácií do zdieľanej databázy zdravotných záznamov by znamenalo, že lekári, nemocnice, chirurgovia, zdravotné sestry a zdravotní pracovníci by mali prístup k zdieľaným údajom o pacientovi. Mali by mať úplné podrobnosti o lekárskych záznamoch, šetrili čas a pomáhali im robiť komplexnejšie rozhodnutia pri liečbe pacienta. To môže potenciálne zachrániť život v prípade, že sa pacient dostane do života ohrozujúcej situácie. Záznamy všetkých základných zdravotných informácií ako informácie krvného typu, alergií na určité lieky, kontaktov v núdzových situáciách, aktuálnych liekov ktoré pacient môže užívať, alebo iné podrobnosti by boli okamžite dostupné pri použití Blockchainu. (Fleming, 2017)

1.9. Ethereum blockchain

V dnešnej dobe sa Bitcoin považuje za najväčší verejný Blockchain a slúži ako

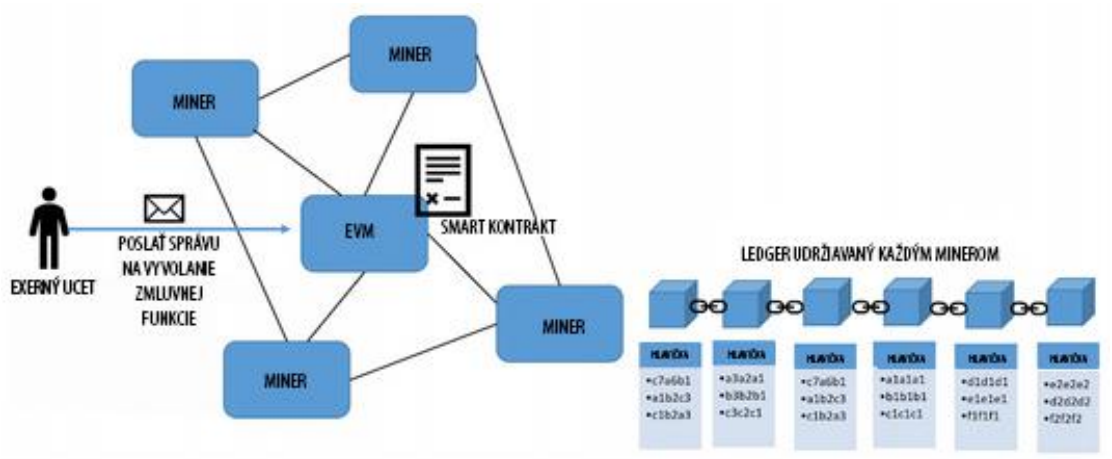
model, na ktorom sú založené mnohé ďalšie aplikácie založené na Blockchaine. Avšak Bitcoinový Blockchain predstavuje iba jednu implementáciu tejto technológie. Vzhľadom na to, že čoraz viac priemyselných odvetví začína skúmať potenciál Blockchainu, objavujú sa pravidelne nové modely, ktoré často slúžia špeciálnemu účelu. Pri pohľade na Bitcoin vidíme, že v podstate jeho cieľom je decentralizovaná digitálna mena typu peer-to-peer. Bitcoin rieši jediný konkrétny problém a to : ako zabezpečiť bezpečné finančné transakcie na báze peer-to-peer odkiaľkoľvek na svete a zároveň eliminovať potrebu dôvery, sprostredkovateľa alebo centralizovaného orgánu ako banky. Zatiaľ čo Bitcoin nie je dokonalý, bol pomerne úspešný z hľadiska účelu, na ktorý bol určený. Keďže vývojári a podnikatelia začali vnímať potenciál Blockchainu, ktorý ďaleko presahoval finančné transakcie, mnohí si začali predstavovať alternatívne štruktúry, ktoré by mohli byť vhodnejšie na splnenie rôznych funkcií. Vitalik Buterin, vývojár, ktorý vynašiel Ethereum, predstavil otvorenú platformu, kde by Blockchain mohol vykonávať rôzne funkcie. Namiesto Blockchainu, ktorý iba jednoducho uchováva dáta o finančných transakciách, rovnako ako v prípade Bitcoinu, je Ethereum Blockchain určený na vykonanie kódu založeného na overených transakciách. Namiesto jednoduchého presunu finančných prostriedkov z účtu A na účet B, podobne ako v prípade Bitcoinu, by spoločnosť Ethereum mohla vytvoriť prostredie, v ktorom by transakcia z účtu A na účet B mohla spustiť veľkú škálu udalostí. Napríklad transakcie v Ethereum sa dajú použiť na registráciu nového názvu domény, na prevod vlastníckych práv, na správu registrácie voličov alebo na vykonanie bezpečných zmlúv medzi dvoma alebo viacerými stranami. V skutočnosti sa "transakcie" v rámci spoločnosti Ethereum často označujú ako "inteligentné zmluvy". (Mathis, 2017)

1.9.1. Blockchain a Ethereum architektúra

Blockchain je architektúra pozostávajúca z viacerých komponentov a to, čo robí Blockchain jedinečným, je spôsob, akým tieto komponenty fungujú a vzájomne spolupracujú. Niektoré z dôležitých komponentov Ethereum sú Ethereum Virtual Machine (EVM), mineri, blok, transakcia, algoritmus, účet, inteligentná zmluva, ťažba, Ether a Gas. V tejto kapitole budeme diskutovať o každej z týchto zložiek.

Blockchain sieť pozostáva z viacerých uzlov patriacich minerom a z niektorých uzlov, ktoré nepatria minerom, ale pomáhajú pri realizácii inteligentných zmlúv a transakcií.

Tieto uzly sú známe ako EVM. Každý uzol je pripojený k inému uzlu v sieti. Tieto uzly používajú protokol peer-to-peer, aby medzi sebou komunikovali. Štandardne používajú port 30303 na komunikáciu medzi sebou. Každý miner udržiava určitú verziu knihy. Kniha obsahuje všetky bloky reťazca. Pri väčšom množstve minerov je celkom možné, že každá minerská inštancia môže mať rôzne verzie Blockchainu. Mineri preto synchronizujú svoje bloky na priebežnej báze, aby zabezpečili, že každá verzia Blockchainu u minera je rovnaká ako u iných. EVM tiež pokrýva inteligentné zmluvy. Inteligentné zmluvy pomáhajú rozšíriť Ethereum tým, že do neho zapíšu vlastné obchodné funkcie. Tieto inteligentné zmluvy môžu byť vykonané ako súčasť transakcie a sledujú proces ťažby, ako bolo spomenuté vyššie. Osoba, ktorá má účet v sieti, môže poslať správu o prenose Ethera z účtu do iného alebo môže poslať správu, ktorá vyvolá funkciu v rámci zmluvy. Ethereum ich nerozlišuje, pokiaľ ide o transakcie. Transakcia musí byť digitálne podpísaná s privátnym kľúčom držiteľa účtu. To má zaistiť, aby sa pri overovaní transakcie a zmene zostatkov viacerých účtov mohla určiť totožnosť odosielateľa. (Modi, 2018)



Obrázok 5. Komunikácia v Blockchaine
(Zdroj: Solidity Programming Essentials)

1.9.2. Ethereum Virtual Machine

EVM predstavuje jeden globálny 256-bitový "počítač", v ktorom sú všetky transakcie lokálne na každom uzly v sieti a sú vykonané v relatívnej synchronizácii. EVM je globálne prístupný virtuálny stroj pozostávajúci z menších počítačov. Tento obrovský počítač, ktorý má ktokoľvek, kto predstavuje uzol alebo má peňaženku, umožňuje jednoduchý pohyb s ľubovoľne veľkým množstvom hodnôt (peniazmi) takmer v momente rozhodnutia účastníka. Hoci môže tento globálny virtuálny počítač

používať každý, nikto nemôže vytvoriť falšované peniaze vo vnútri systému, alebo taktiež nemôže presunúť prostriedky bez povolenia.

Vo všeobecnosti je virtuálny počítač emuláciou počítačového systému iným počítačovým systémom. Tieto emulácie sú založené na rovnakých počítačových architektúrach ako cieľ ich emulácie, ale väčšinou reprodukujú túto architektúru na inom hardvéri, než aký bol určený. Virtuálne počítače môžu byť vytvorené pomocou hardvéru, softvéru alebo oboch. V prípade Ethereum je to oboje. Namiesto bezpečného pripojenia tisícov samostatných strojov. Ethereum bezpečne prevádzkuje jeden veľmi veľký stroj, ktorý môže zahŕňať celú planétu. Z pohľadu vývojára softvéru EVM predstavuje aj prostredie pre malé programy ako smart kontrakty, ktoré môže sieť vykonať. (Dannen, 2017)

1.9.3. Ether

Ether je názov kryptomeny pre Ethereum Blockchain. Ether dostal pomenovanie podľa substancie, o ktorej sa predpokladalo, že preniká celým priestorom a umožňuje vznik vesmíru. V tomto zmysle je éter látka, ktorá tvorí Ethereum. Ether stimuluje sieť, aby bola zabezpečená prostredníctvom Proof of Work, presne tak ako napríklad Bitcoin stimuluje Bitcoin Blockchain. Na vykonanie akéhokoľvek kódu v sieti Ethereum je potrebný Ether. V momente použitia na vykonanie akéhokoľvek smart kontraktu v Ethereum Blockchaine sa Ether sa označuje ako Gas. Vykonanie kódu v inteligentnej zmluve tiež stojí určité množstvo Etheru. Táto funkcia prináša utilitu pridaného tokenu. Pokiaľ jednotlivci chcú používať Ethereum na aplikácie a zmluvy, Ether bude mať hodnotu mimo špekulácií. Divoký nárast hodnoty Etheru z neho urobil populárny prvok, o ktorý sa zaujímajú finančné burzy ,preto je Ether široko obchodovaný na burzách po celom svete. Avšak volatilná povaha a nízka hĺbka trhu robia z Ethera rizikovou investíciou. (Laurence, 2017)

1.9.4. Gas

V predchádzajúcej časti bolo spomenuté, že poplatky sa platia pomocou jednotky Ether a to pre akúkoľvek akciu vykonanú v Ethereum. Ether sa obchoduje na verejných burzách a jeho cena sa pohybuje každý deň. Ak sa na zaplatenie poplatkov používa Ether, náklady na používanie tej istej služby môžu byť v určitých dňoch veľmi vysoké a nízke v iných dňoch. To môže viesť k tomu že ľudia budú čakať, kým ceny

Etheru klesnú, aby uskutočnili transakcie. To nie je to ideálne pre platformu, ako je Ethereum. Gas pomáha pri zmierňovaní tohto problému. Gas je vnútornou menou spoločnosti Ethereum. Výkonnosť a náklady na využívanie zdrojov sú predurčené v spoločnosti Ethereum z hľadiska Gas jednotiek. To je tiež známe ako Gas cost. Existuje aj Gas price, ktorú možno upraviť na nižšiu cenu, keď cena Etheru rastie a taktiež aj pre úpravu vyššej ceny, keď cena Etheru klesá. Napríklad na vyvolanie funkcie v zmluve, ktorá mení slovný reťazec bude stáť určitú hodnotu Gas, ktorá je vopred určená preto užívatelia platia v Gas jednotkách, aby zabezpečili plynulé vykonanie tejto transakcie.

1.9.5. Smart kontrakty

Táto aplikácia Blockchainu sa považuje za najambicióznejšiu. Kľúčová myšlienka inteligentných zmlúv je, že termíny a informácie môžu byť uzavreté v zmluve v prípade splnenia určitých podmienok automaticky. Ak však podmienky nie sú splnené, zmluvu nebude možné uzatvoriť. Možnosť vložiť informácie do blokov umožňuje vytvoriť bezpečné kontrakty medzi jednotlivcami, ktorí nepotrebujú potvrdenie treťou stranou. Navyše ani jedna strana nemôže porušiť zmluvné podmienky v inteligentnej zmluve .

Inteligentné zmluvy je možné vytvoriť zo všetkých druhoch zmlúv, peniaze môžu byť odoslané za určitých okolností a organizácie môžu do svojich zmlúv naprogramovať, aby automaticky vyplácali dividendy v prípade ak sa dosiahne určitá úroveň zisku. Tieto inteligentné zmluvy zmenia spôsob, akým sa dnes vytvárajú zmluvy ,dôvodom je že sú lacnejšie, nie je potrebný sprostredkovateľ na overenie zmlúv. (Andrea Pinna, 2016)

Smart kontrakty sú samočinné programy, ktoré sa nachádzajú v Blockchaine a sú schopné uchovávať pravidlá, dôsledky a výpočty pre každú transakciu, ktorá sa uskutočnila v Blockchaine. Koncept smart kontraktov prvýkrát formálne vytvoril Nick Szabo v roku 1994. Smart kontrakty môžu mať ako vstupy akúkoľvek formu údajov. Tento revolučný koncept automatizuje presadzovanie zmluvných dohôd bez akýchkoľvek sprostredkovateľov alebo dôveryhodných tretích strán a zlepšuje transparentnosť, pretože každý jednotlivý uzol v Blockchaine dodržiava protokoly špecifikované v smart kontrakte.

Všetky zmluvné transakcie sú uložené v Blockchaine v chronologickom poradí pre budúci prístup spolu s kompletnou možnosťou auditu udalostí uložených v Blockchaine. Taktiež zúčastnené uzly nemôžu manipulovať alebo meniť zmluvné podmienky uložené v Blockchaine, týmto sa odstraňuje šanca pre podvodné zmluvy. Celá sieť smart kontraktov sa chová ako veľký centrálny počítač, ale vyhýba sa riziku zlyhania, nákladom a dôvery v centralizovaného výpočtového mechanizmu. Podobne ako každá právna zmluva alebo dohoda, smart kontrakt môže zahŕňať všetky možné dôsledky transakčného správania a môže vydať opatrenia, ktoré sa majú prijať pre každý scenár. Inteligentné zmluvy umožňujú riešiť všetky platné a neplatné transakcie, ktoré sa môžu vyskytnúť v rámci Blockchainu a dokonca aj sledovať abnormálne správanie z účastníckych uzlov. Inteligentné zmluvy neuchovávajú žiadne údaje, namiesto toho poskytujú usmernenie o tom, ako sa dáta ukladajú do blokov. Ich potenciál je v znížení a dokonca eliminovaní šance na podvody a zníženia režijných nákladov mnohých obchodných transakcií.

Zvážme jednoduchý príklad dodávateľského reťazca, ktorý v súčasnosti kontrolujú sprostredkovatelia väčšinou prostredníctvom centralizovanej platformy. Vybrali sme tento vzorový scenár, pretože dodávateľský reťazec je hlavným obchodným scenárom kde sprostredkovatelia zohrávajú dôležitú úlohu pri kontrole činností od výroby až po distribúciu. Mnoho nezákonných aktivít, ako sú rôzne podvody čierny trh atď., sa dejú v dodávateľských reťazcoch, aby sa vynásobil zisk sprostredkovateľa pričom výrobcovia a zákazníci na to doplácajú. Pred vynájdením blockchain platformy bolo prakticky nemožné spravovať a riadiť jednotlivé aktivity cez internet bez pomoci centralizovanej organizácie, ktorá zabezpečuje, že údaje nebudú manipulované ani zneužívané žiadnou osobou. V dôsledku nedostatku stabilnej decentralizovanej platformy nezávislí jednotlivci nemohli potvrdiť, či transakcia bola úspešná bez toho, aby sa spoliehali na dôveryhodnú centrálnu inštitúciu, aby overila, či táto konkrétna transakcia bola uskutočnená. Smart kontrakt okrem toho poskytuje slobodu ľuďom vykonávať transakcie s digitálnymi údajmi alebo údajmi v celej blockchain sieti bezpečným, bezúhonným a nemeniteľným spôsobom. Tento spôsob by mohol pomôcť v dodávateľskom reťazci tým, že umožní právny súbor protokolov pre všetky transakcie, ktoré sa dejú v dodávateľskom reťazci. Systém verejných účtovných kníh môže navyše zvýšiť prehľadnosť a tým aj spravodlivé rozdelenie a tvorbu cien na výrobky. (Andrea Pinna, 2016)

2. CIEĽ A METODIKA PRÁCE

V tejto kapitole si bližšie priblížime technológiu blockchain na jednoduchom príklade. Aplikácia je veľmi jednoduchá, jej podstatou je simulovať jednoduché voľby čo pozostáva z inicializovania zoznamu súťažiacich, možnosti aby niekto hlasoval pre kandidátov a zobrazenie celkového počtu hlasov prijatých pre každého kandidáta.
Cieľom nie je len naprogramovanie aplikácie, ale aj pochopenie procesu kompilácie, tvorby a interakcie s aplikáciou.

3. VÝSLEDKY PRÁCE

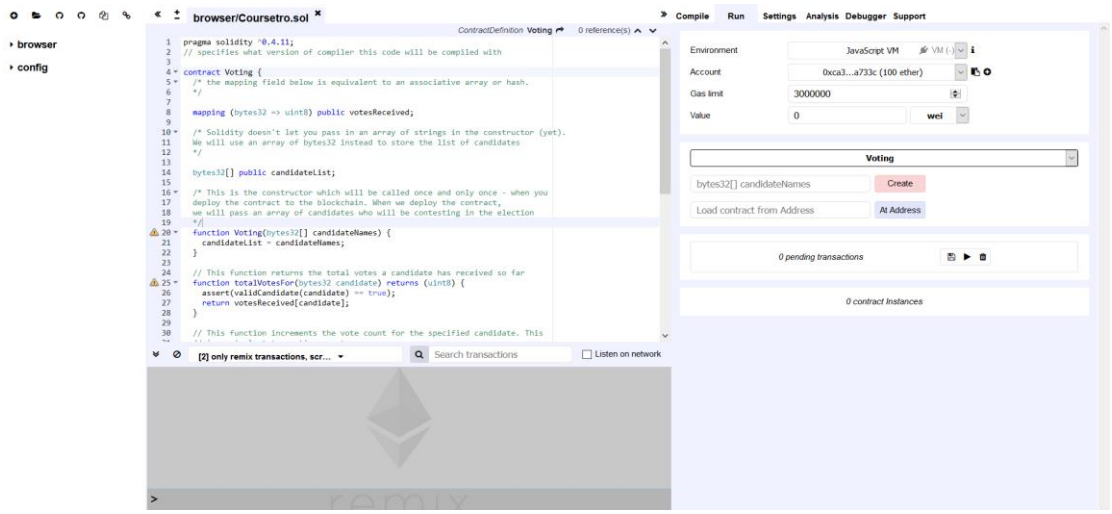
Blockchain predstavuje nový prístup riadenia a štruktúrovania dát, špecializovaných na anonymitu vlastníctva transakčných aktivít. Po prvotnom predstavení teoretickej koncepcie, každá z etáp programovania je prezentovaná a prípadne zhrnutá v tejto kapitole.

3.1. Základné informácie pre tvorbu smart kontraktov

Remix IDE poskytuje integrované vývojové prostredie (IDE) pre vývoj smart kontraktov. Zameriava sa na vývoj a nasadenie inteligentných zmlúv vytvorených v Solidity jazyku. Remix poskytuje rôzne výhody ako:

- Vytvorenie smart kontraktov (Remix integruje editor pre jazyk Solidity).
- Debugovanie smart kontraktov.
- Prístup k stavu a vlastnostiam už existujúceho smart kontraktu.
- Analyzovanie Solidity kódu ,čo ma následný vplyv na menšiu chybovosť a nižšiu nadbytočnosť kódu. (Ethereum Remix)

Remix IDE je prehliadačovo založené IDE vytvorené tvorcami kryptomeny Ethereum. Remix IDE je možné nainštalovať lokálne pomocou Github Repo ,alebo je možné použiť online verziu na stránke http://remix.ethereum.org. V našom prípade sme zvolili online verziu Remix IDE z dôvodu ľahšieho písania kódu, následne sa daný kód skopíruje do projektového priečnika. Podstatou Remix IDE je možnosť písania a následného vytvorenia Solidity smart kontraktov. Taktiež nám Remix IDE poskytuje užitočné nástroje.



Obrázok 6 Zobrazenie REMIX IDE
(Zdroj: Autor)

Pri navštívení stránky na ľavej strane vidíme stĺpec zobrazujúci naše súbory, ktoré obsahujú naše smart kontrakty v .sol formáte. V strede prehliadača sa nachádza priestor do ktorého sa píše Solidity kód. V spodnej časti sa nachádza debugger. Pravý stĺpec obsahuje viacero záložiek obsahujúcich nástroje ako compile ,run ,setting atď.

V tomto momente nás zaujíma iba Run záložka. V záložke Enviroment si v tomto momente vyberieme možnosť Javascript VM ,neskôr si danú možnosť zmeníme. Javascript EVM(Ethereum Virtual Machine) znamená že náš smart kontrakt pobeží iba lokálne, to znamená že daný smart kontrakt nie je uložený v reálnom Blockchaine. Túto metódu používame z dôvodu rýchleho vytvorenia kontraktov.

3.2. **Remix IDE**

Pri navštívení online verzie Remix IDE nám stránka poskytuje na ukážku jednoduchý smart kontrakt. Pre vytvorenie nového smart kontraktu klikneme na ikonu (+) v ľavom hornom horu. Súbor si v našom prípade pomenujeme Voting.sol.

Prvý riadok kódu obsahuje informáciu o verzii Solidity ktorú budeme používať v našom prípade sa jedná o verziu 0.4.11.

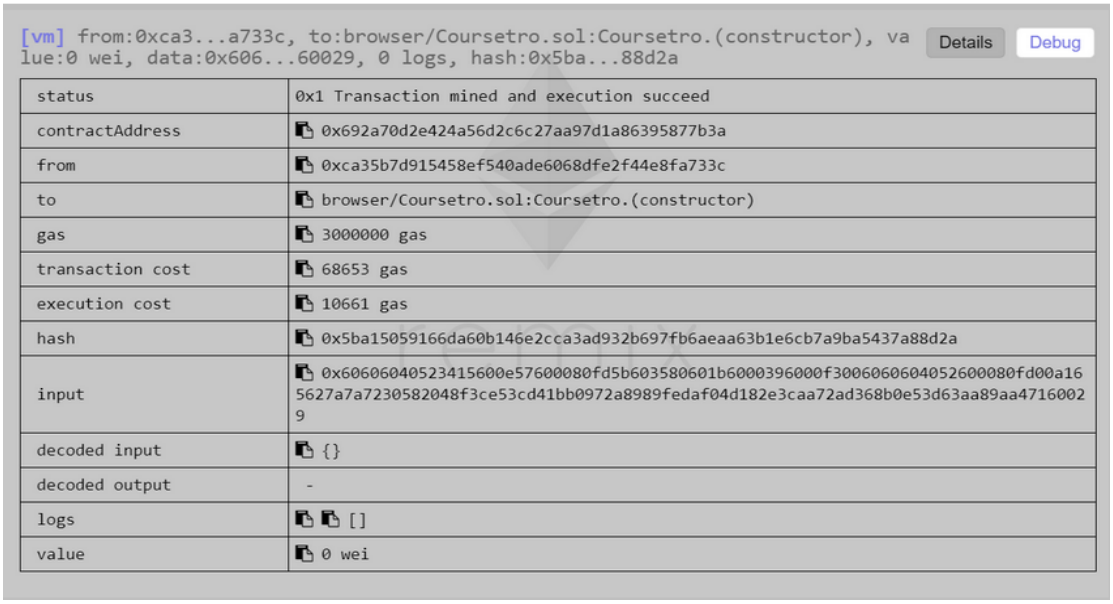
```
pragma solidity ^0.4.11;
contract Voting {
```

Obrázok 7. Verzia Solidity a názov kontraktu.
(Zdroj: Autor)

V ďalšom kroku je potrebné si zadefinovať názov triedy napríklad v našom prípade Voting. Na pravej strane prehliadača pod záložkou Run sa nachádza tlačidlo Create. Po kliknutí sa nám zobrazí nová sekcia v ktorej sa nachádza názov nášho kontraktu a adresa. Zobrazenie tohto okna znamená že smart kontrakt existuje na danej zobrazenej adrese.

Obrázok 8 Zobrazenie názvu a adresy kontraktu
(Zdroj: Autor)

Pre možnosť úplného zobrazenia informácií klikneme v spodnej časti prehliadača v záložke debugger na tlačidlo Details.



Obrázok 9. Detailné zobrazenie kontraktu.
(Zdroj: Autor)

3.3. **Typy dát**

Solidity je statický jazyk to znamená že typ premennej musí byť definovaný už

pred kompiláciou. V Solidity budeme používať typy ako:

- Boolean-tento typ vracia hodnoty true alebo false.
- INT/UNIT-oboje INT a UNIT reprezentujú číselné hodnoty. Hlavným rozdielom medzi nimi je že INT je schopné ukladať aj záporné čísla.
- Address-Tento typ reprezentuje 20 bytovú hodnotu, ktorá ukladá Ethereum adresu.
- Bytes32-reprezentuje fixnú veľkosť poľa.
- Bytes- reprezentuje dynamickú veľkosť poľa.
- String-dynamický uložený text.
- Mapping-Hash tabuľky s typmi kľúčov a typmi hodnôt.
- Struct- Umožňuje definovať nové typy.

3.4. Viditeľnosť funkcií a premenných.

Solidity momentálne pozná 4 typy viditeľnosti pre funkcie a premenné, dané typy sú:

- Public-Tento typ umožňuje definovať funkcie a premenné ktoré môžu byť volané.
- Private-Funkcie a premenné sú dostupné iba konkrétnemu kontraktu.
- Internal-Funkcie a premenné je možné volať iba interne.
- External- Funkcie a premenné je možné volať z iných transakcií a kontraktov. Nie je možné ich volať interne jedine pomocou this. Napríklad „this.nazovFunkcie()“.

3.5. Konštruktor smart kontraktov

Každý smart kontrakt obsahuje funkciu konštruktor. Tento konštruktor je volaný pri vytváraní kontraktu. Vo vnútri je možné definovať hodnoty premenných.

3.6. Inštalácia a spustenie Ethereum TestRPC

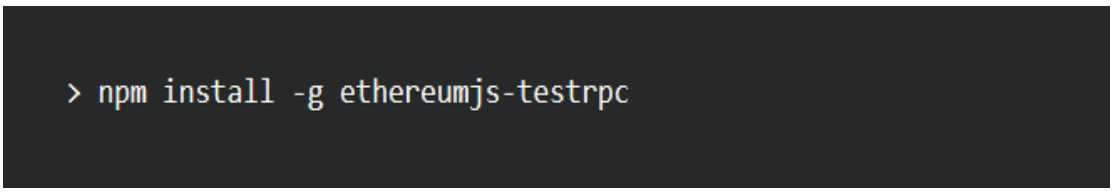
Ethereum TestRPC je Node.js klient slúžiaci na testovanie a tvorbu smart kontraktov. Pretože TestRPC je založený na Node.js ,je potrebné si ho nainštalovať taktiež s NPM(Node Package Manager). Node.js je open source platforma, ktorá nám umožňuje vytvárať rýchle a škálovateľné sieťové aplikácie pomocou jazyka JavaScript. Node.js je postavený na V8,čo je moderný virtuálny stroj JavaScript, ktorý ovláda

webový prehliadač Google Chrome. Aplikácia Node.js využíva model I / O s neblokujúcim prístupom k udalostiam, ktorý je jednoduchý a efektívny. Node.js dokáže

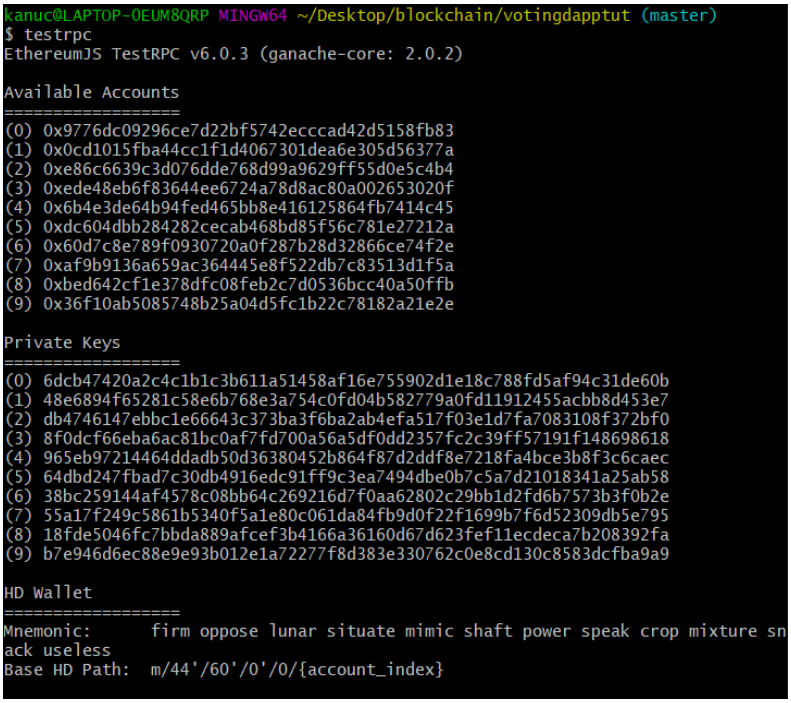
Obrázok 10. Overenie správneho nainštalovania Node,js a npm.
(Zdroj: Autor)

spracovať viacero súbežných sieťových pripojení s malou záťažou, čo je ideálne pre aplikácie náročné na dátovú aplikáciu v reálnom čase.Node.js je možné stiahnuť zo stránky https://nodejs.org .Po inštalácií je potrebné overenie či inštalácia prebehla úspešne. Overiť si to môžeme pomocou konzoly alebo otvorením GitBashu, kde do konzoly napíšeme príkaz:

V prípade správneho nainštalovania nám konzola vypíše verzie node.js a npm.
Následne pomocou npm si nainštalujeme Ethereum TestRPC:



Obrázok 11. Inštalácia TestRPC pomocou npm.
(Zdroj: Autor)

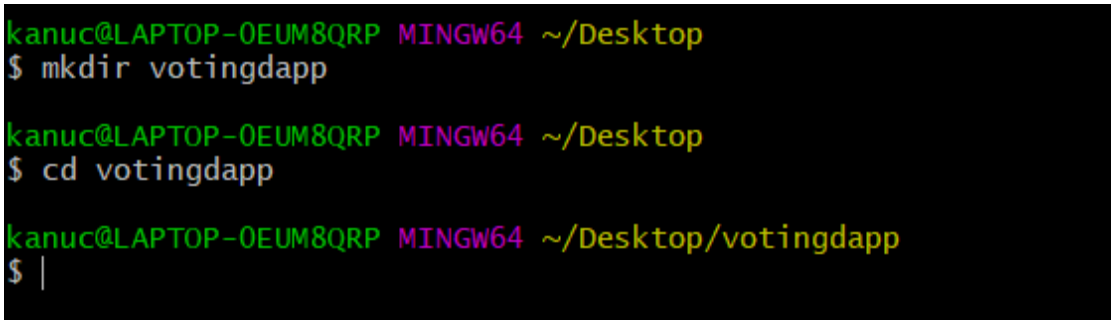


Obrázok 12. Zobrazenie účtov a privátnych kľúčov.
(Zdroj: Autor)

V prípade úspešnej inštalácie spustíme príkaz testrpc ,testrpc nám poskytne 10 rôznych účtov a privátnych kľúčov a taktiež nám poskytne lokálny server bežiaci na localhost:8545

3.7. **Inštalácia Web3.js**

Web3.js je oficiálne Ethereum Javascript API.Web3.js sa používa pre interakciu so Ethereum smart kontraktmi. Pred inštaláciou je potrebné si vytvoriť priečinok. Priečinok si môžeme vytvoriť pomocou príkazu napríklad mkdir votingdapp ,potom sa presunieme do daného priečinka pomocou príkazu cd votingdapp.



Obrázok 13. Vytvorenie a presunutie sa do projektového priečinka.
(Zdroj: Autor)

Následne v danom priečinku je potrebné vytvoriť package.json, ktorý má v sebe uložené dependencies (rôzne knižnice potrebné pre správne fungovanie programu).

```
> npm init
```

Obrázok 14. Vytvorenie package.json pomocou npm.
(Zdroj: Autor)

Potom môžeme zadať príkaz ktorý nám nainštaluje Web3.js.

```
> npm install ethereum/web3.js --save
```

Obrázok 15. Inštalácia web3.js pomocou npm.
(Zdroj: Autor)

3.8. **Vytvorenie UI**

Vo vývojovom prostredí ktoré budeme používať (v tomto prípade PHP Storm) si otvoríme priečinok ktorý sme si už vytvorili v predchádzajúcich krokoch. Môžeme si všimnúť že v danom priečinku sa už nachádza node_modules priečinok, ktorý obsahuje nami nainštalovaný web3.js,ktorý sme nainštalovali pomocou NPM. Následne si vytvoríme súbor index.html. V tomto súbore si vytvoríme jednoduchú html stránku skladajúcu sa z rôznych sekcií, základom je vytvorenie html tagov do ktorých následne vložíme kód. Ako prvé sa do html tagov vloží hlavička ktorá obsahuje informácie ako kódovanie znakov pre dokument HTML, prvky zobrazenia poskytujúce prehliadaču

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Hlasovacia DApp</title>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700' rel='stylesheet' type='text/css'>
    <link href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css' rel='stylesheet' type='text/css'>
</head>
```

Obrázok 16. Zobrazenie hlavičky v index.html.
(Zdroj: Autor)

pokyny pre kontrolu rozmerov a mierky stránky, názov stránky, link pre css súbor ktorý obsahuje rôzne štýly pre lepšie vyobrazenie informácií na stránke.

V body tagoch sa nachádzajú sekcie v ktorých je napríklad zobrazené meno kandidáta a jemu priradený počet hlasov, textové pole do ktorého sa zadáva názov kandidáta ktorému chceme dať hlas.

```
<body class="container">
    <h1>A Simple Voting Application</h1>
    <div class="table-responsive">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>Candidate</th>
                    <th># of Votes</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Rama</td>
                    <td id="candidate-1"></td>
                </tr>
                <tr>
                    <td>Nick</td>
                    <td id="candidate-2"></td>
                </tr>
                <tr>
                    <td>Claudius</td>
                    <td id="candidate-3"></td>
                </tr>
            </tbody>
        </table>
    </div>
    <input type="text" id="candidate" />
    <a href="#" onclick="voteForCandidate()" class="btn btn-primary">Vote</a>
</body>
```

Obrázok 17. DOM štruktúra stránky.
(Zdroj : Autor)

V spodnej časti kódu sa nachádzajú linky pre potrebné Javascript skripty ako napríklad link pre Javascript súbor web3.js,J-Query a link pre nami vytvorený Javascript.

```
<script src="https://cdn.rawgit.com/ethereum/web3.js/develop/dist/web3.js"></script>
<script src="https://code.jquery.com/jquery-3.2.1.min.js" integrity="sha256-hwg4gsxgFZhOsEEamdOYGBf13FyQuiTwlAQgxVSNgt4="
        crossorigin="anonymous"></script>
<script src="./index.js"></script>
```

Obrázok 18. Zobrazenie potrebných skriptov.
(Zdroj: Autor)

3.9. Použitie Web3.js na prepojenie a interakciu so smart kontraktmi

Pre jednoduchšie písanie Javascript kódu budeme používať knižnicu Jquerry ktorá v tomto prípade uľahčí písanie kódu. Tento kód hovorí že ak web3 nie je definovaný, tak sa použije iný provider. Ak nie je definovaný (čo predstavuje časť kódu nachádzajúcich sa za else),môžeme ručne špecifikovať provider.

```
if (typeof web3 !== 'undefined') {
    web3 = new Web3(web3.currentProvider);
} else {
    // set the provider you want from Web3.providers
    web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
}
```

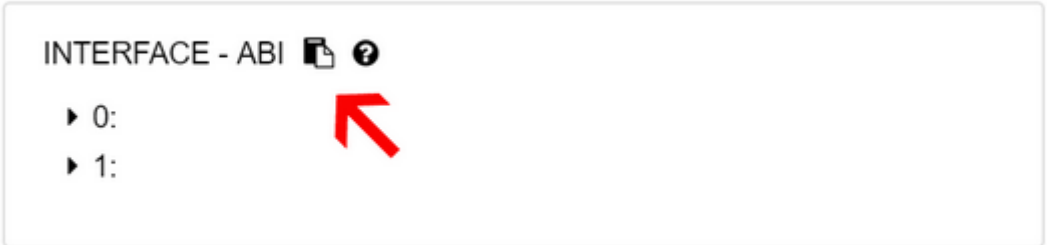
Obrázok 19. Interakcia so smart kontraktmi.
(Zdroj: Autor)

Následne je potrebné špecifikovať základný Ethereum účet ,ktorý získame pomocou metódy web3.eth.defaultAccount.Tento účet predstavuje osobu ktorá chce odovzdať svoj hlas.

```
web3.eth.defaultAccount = web3.eth.accounts[0];
```

Obrázok 20. Definovanie Ethereum účtu.
(Zdroj: Autor)

V ďalšom kroku je potrebné použiť metódu web3.eth.contract() pre inicializáciu alebo vytvorenie kontraktu na danú adresu. Metóda akceptuje jeden parameter, ktorý referuje na ABI(Application Binary Interface).ABI nám dovoľuje zavolať funkciu a prijať dáta zo smart kontraktu. ABI získame z REMIX IDE, po kliknutí na Compile potom na Details. Následne musíme nájsť sekciu s názvom INTERFACE-ABI z ktorej si skopírujeme dáta do našej metódy web3.eth.contract.



Obrázok 21. Zistenie ABI pomocou REMIX IDE.
(Zdroj: Autor)

Keďže v posledných krokoch sme vytvorili interakciu s našim kontraktom pomocou premennej contractInstance. Posledným krokom je definovanie aktuálnej kontrakt adresy. Adresu opätovne získame z Remix IDE ktorá sa nachádza v tabe Run.

3.10. **Vytvorenie smart kontraktu**

Prvý riadok ako sme už uvádzali vyššie uvádza verziu Solidity ktorú používame. Následne uvádzame názov nášho kontraktu v tomto prípade sa volá Voting.

```
pragma solidity ^0.4.11;
contract Voting {
```

Obrázok 22. Pomenovanie triedy v Solidity.
(Zdroj: Autor)

Celý náš nasledujúci kód sa bude nachádzať vo vnútri nášho kontraktu čo znamená že sa nachádza v zložených zátvorkách. V jazyku Solidity sa mapovanie označuje ako hashová tabuľka, ktorá pozostáva z kľúčových typov a dvojíc typov hodnôt. Mapovanie definujeme ako ktorýkoľvek iný typ premennej, v našom prípade vytvárame Mapping, ktorý akceptuje najprv key typ (v našom prípade to bude typ poľa) a typ hodnoty bude číslo.

```
mapping (bytes32 => uint8) public votesReceived;
```

Obrázok 23. Vytvorenie Mappingu vo vnútri kontraktu.
(Zdroj: Autor)

Solidity nám v momentálnych verziách nedovoľuje vložiť pole stringov v konštruktore. Z tohto dôvodu použijeme pole bytes32 namiesto použitia poľa kde by sme ukladali mená kandidátov.

```
bytes32[] public candidateList;
```

Obrázok 24. Vytvorenie poľa kandidátov
(Zdroj: Autor)

Nasledujúci kód predstavuje konštruktor ktorý bude zavolaný iba jeden krát a to v momente keď sa vloží kontrakt do Blockchainu. Po vložení kontraktu do Blockchainu prejdeme poľom kandidátov ,ktorý budú vo voľbách.

```
function Voting(bytes32[] candidateNames) {
  candidateList = candidateNames;
}
```

Obrázok 25. Vytvorenie konštruktora.
(Zdroj: Autor)

Pre volebné hlasovanie je dôležité sčítavanie bodov. Táto funkcia inkrementuje hlasy pre daného kandidáta vždy o 1.

```
function voteForCandidate(bytes32 candidate) {
  assert(validCandidate(candidate) == true);
  votesReceived[candidate] += 1;
}
```

Obrázok 26. Inkrementácia hlasov o jeden hlas.
(Zdroj: Autor)

Po tomto kroku je potrebné zobraziť všetky hlasy ktoré boli doposiaľ nazbierané pre určitého kandidáta.

```
function totalVotesFor(bytes32 candidate) returns (uint8) {
  assert(validCandidate(candidate) == true);
  return votesReceived[candidate];
}
```

Obrázok 27. Funkcia ukladajúca dosiahnutý počet hlasov.
(Zdroj: Autor)

```
function validCandidate(bytes32 candidate) returns (bool) {
  for(uint i = 0; i < candidateList.length; i++) {
    if (candidateList[i] == candidate) {
      return true;
    }
  }
  return false;
}
```

Obrázok 28. Overenie počtu kandidátov.
(Zdroj: Autor)

Taktiež je potrebné overiť zoznam kandidátov. Overenie prebieha jednoduchou formou kde sa overí dĺžka poľa kandidátov.

3.11. **Zasielanie hlasov**

Zasielanie hlasov daným kandidátom prebieha pomocou j-Query,čo je Javascript knižnica uľahčujúca písanie Javascript kódu. V súbore index.html je potrebné si vytvoriť textové pole do ktorého budeme zadávať meno nášho kandidáta. Taktiež je potrebné vytvoriť tlačidlo ktorým potvrdíme odoslanie hlasu kandidátovi a následne sa daná informácia vloží do Blockchainu.

```
<input type="text" id="candidate" />
<a href="#" onclick="voteForCandidate()" class="btn btn-primary">Vote</a>
```

Obrázok 29. Vytvorenie inputu pre odosielanie hlasov.
(Zdroj: Autor)

Odoslanie hlasu nám zabezpečuje funkcia voteForCandidate(),ktorá bude zavolaná iba v momente kliknutia na tlačidlo Vote.

Jednoducha Hlasovacia Aplikacia

Kandidat	Pocet hlasov
Samuel	6
Richard	5
Andrej	4

Richard Hlasovat

Obrázok 30. Zobrazenie tlačidla pre odoslanie hlasov.
(Zdroj: Autor)

Funkcia voteForCandidate() je naviazaná na window objekt, window objekt predstavuje celé otvorené okno prehliadača obsahujúce DOM. Je potrebné aby hodnota ktorá bola zadaná bola uložená do premennej v našom prípade to je candidateName. Následne si zavoláme z objektu contractInstance funkciu voteForCandidate, ktorú sme si vytvorili v kontrakte a jej úlohou je navýšenie počtu hlasov určitého kandidáta o 1 hlas.

Obrázok 31. Zobrazenie objektu funkcií.
(Zdroj: Autor)

Následne je potrebné zistiť ID bunky v ktorej sa zvýši počet hlasov. Túto funkcionalitu zabezpečujeme pomocou zadaného mena kandidáta do textového poľa a následného odoslania, pomocou mena kandidáta si vieme zistiť do ktorej bunky treba priradiť hlas s pomocou objektu kde k menu je priradené ID.

```
var candidates = {
  Rama: "candidate-1",
  Nick: "candidate-2",
  Claudius: "candidate-3"
};
```

Obrázok 32. Zoznam kandidátov a im priradených ID.
(Zdroj: Autor)

Potom túto informáciu zašleme do Blockchainu pomocou funkcie totalVotesFor ktorá sa taktiež nachádza v kontrakte a je volaná vo funkcii voteForCandidate, voteForCandidate sa spúšťa v momente odoslania hlasu.

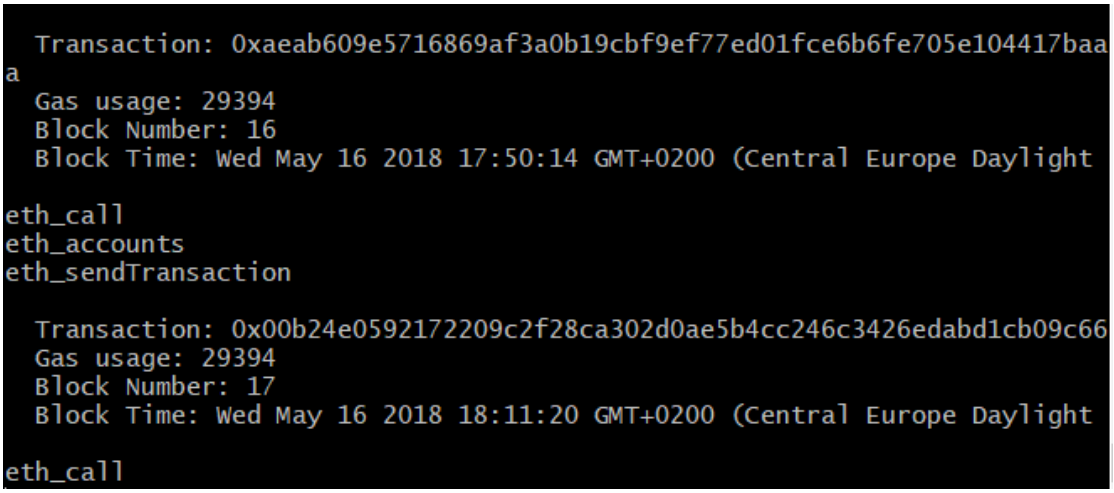
```
window.voteForCandidate = function() {
  candidateName = $("#candidate").val();
  contractInstance.voteForCandidate(candidateName,{ from: web3.eth.accounts[0] },function() {
      let div_id = candidates[candidateName];
      $("#" + div_id).html(contractInstance.totalVotesFor.call(candidateName).toString());
      console.log( $("#" + div_id).html(contractInstance.totalVotesFor.call(candidateName).toString()));
    }
  );
};
  var candidates = {
      Rama: "candidate-1",
      Nick: "candidate-2",
      Claudius: "candidate-3"
  };
```

Obrázok 33. Odoslanie hlasov do Blockchainu.
(Zdroj: Autor)

Taktiež je potrebné aby boli zobrazené počty hlasov hneď po navštívení stránky, tento problém rieši cyklus ktorý si zistí dĺžku poľa kandidátov ,mená kandidátov a počet hlasov im priradených. To zabezpečí že aj po obnovení stránky sú hodnoty zobrazené.

```
for (var i = 0; i < candidateNames.length; i++) {
  let name = candidateNames[i];
  let val = contractInstance.totalVotesFor.call(name).toString();
  $("#" + candidates[name]).html(val);
  console.log(val);
}
```

Obrázok 34. Cyklus pre zobrazenie počtu hlasov hneď po načítaní stránky.
(Zdroj: Autor)



Obrázok 35. Zobrazenie blokov a informácií v bloku.
(Zdroj: Autor)

Každý jeden hlas ktorý bol zaslaný predstavuje jeden blok ktorý sa ukladá chronologicky do Blockchainu. Ako je možné vidieť na obrázku vyššie jeden blok v sebe uchováva informácie ako číslo transakcie, číslo bloku v poradí, čas vytvorenia bloku a Gas-u čo predstavuje dôležitú zložku bloku. Gas je názov špeciálnej jednotky používanej v Ethereum Blockchaine. Určuje, koľko "práce" je potrebných na vykonanie akcie alebo súboru opatrení. Dôvod prečo je Gas dôležitý je že pomáha zaistiť aby transakcie odosielané do siete mali primeraný poplatok.

4. DISKUSIA A ZÁVER

Účelom tejto práce bolo preskúmať prínosy pomocou technológie Blockchain v aplikačných oblastiach, kde transparentnosť a dôvera je potrebná. Prvotný cieľ vytvorenia uzlov ktoré by medzi sebou komunikovali, synchrónne aktualizovali Blockchain pomocou Oracle VM VirtualBox, Vagrantu alebo Docker-a mi nevyšiel, preto sme zvolil inú metódu a to manuálne meniť adresu používateľa ktorý by odosielal svoj hlas kandidátom ,keďže TestRPC ponúka takúto možnosť , zmena sa realizuje priamo v kóde programu . Fungujúci dôkaz o koncepte digitalizovaného hlasovacieho systému bol navrhnutý a úspešne implementovaný pomocou Ethereum Blockchain a to z dôvodu unikátnosti tejto technológie. To sa uskutočnilo prostredníctvom rozsiahleho výskumu. Analýza implementácie dokázala, ako vlastnosti Ethera vo všeobecnosti možno výhodne využiť napríklad pri hlasovaní . Riešenie samoobslužného hlasovacieho systému založené na dôvere bolo automatizované pomocou inteligentných zmlúv . Používatelia preto už nemôžu zneužiť transakčný systém, pretože pravidlá transakcie sú diktované Blockchainom. Okrem toho transparentnosť Blockchainu umožňuje používateľom overiť či transakcie sa uskutočňujú tak ako by mali. Ďalej kvôli neustálej kontrole celej siete Blockchainu, systém nemôže byť napadnutý alebo inak poškodený, takže môžeme systém považovať za veľmi bezpečný. To sú všetky vlastnosti, ktoré sú žiaduce v systémoch založených na dôvere. Momentálne nie je potrebné riešenie problému veľkosti Blockchainu pre bežných používateľov, ktorí predstavujú jednotlivé uzly v sieti. Existujú návrhy a teórie o tejto téme, ale zatiaľ zostávajú neotestované. Teda v tomto okamihu, čím väčší je Blockchain, tým viac centralizovaným sa stáva. Podobne je žiaduci aj nižší čas tvorenia blokov, avšak to by spôsobilo nestabilnosť siete. V dôsledku toho môžu niektoré aplikácie trpieť.

Problémom je aj že každá činnosť v sieti, stojí nejaké peniaze, to je tiež prekážkou na prekonanie toho, aby užívatelia mohli túto technológiu prijať. To sú všetky problémy, ktoré je potrebné riešiť, aby bol Blockchain rozšírený medzi ľuďmi. Preto sú inovácie a výskum stále potrebné v tejto oblasti. Hoci tieto prekážky existujú, možno tvrdiť, že potenciálne technológia Blockchain môže mať značný vplyv na spoločnosť vo všeobecnosti. Tvrdím, že technológia Blockchain je jednou z najsľubnejších technológií doteraz, v oblasti vykonávania digitálnych volieb. Existujú však aspekty bezpečnosti, ktoré nie sú v tomto prípade použitia riešiteľné prostredníctvom Blockchainu, ktoré je potrebné zohľadniť. Pokiaľ ide o riešenie problémov

zabezpečenia digitálnych práv, tvrdím že vlastnosti Blockchainu sú veľmi prospešné v tejto oblasti. Problém má korene v dôsledku nedostatku dôveryhodných orgánov. V tomto konkrétnom prípade použitia je ťažké nájsť zjavné nevýhody použitia Blockchainu, keďže faktory ako čas tvorby blokov a náklady majú malý význam v tomto zameraní, preto by bolo vhodné používať technológiu Blockchain.

Na záver tejto práce ,technológia Blockchain má určite obrovský potenciál vďaka svojim revolučným funkciám. Technológia je už uplatniteľná v obrovskom množstve aplikácií. Ak jej ďalší vývoj dokáže prekonať výzvy, ktorým čelí Blockchain, predpokladám, že to bude mať zásadný vplyv na finančný sektor, ako aj na spoločnosť.

5. ZOZNAM POUŽITEJ LITERATÚRY

Andrea Pinna, Wiebe Ruttenberg. 2016. Distributed ledger technologies in securities post trading. [Online] April 2016. [Dátum: 18. February 2018.] https://www.ecb.europa.eu/pub/pdf/scpops/ecbop172.en.pdf.

Antonopoulos, Andreas M. 2015. Mastering Bitcoin. *Mastering Bitcoin.* [Online] 2015. [Dátum: 20. December 2017.] http://chimera.labs.oreilly.com/books/1234000001802/ch07.html.

Bashir, Imran. 2017. Mastering Blockchain. s.l. : Packt Publishing Ltd, 2017, s. 146-147.

Bennett, Sean. 2017. Bitcoin: Understanding Bitcoin, Mining, Investing & Trading for Beginners. 2017, s. 27-28.

BitcoinWiki. [Online] [Dátum: 16. February 2018.] https://en.bitcoin.it/wiki/Transaction.

BitcoinWiki. [Online] [Dátum: 15. February 2018.] https://en.bitcoin.it/wiki/Transaction#general_format_of_a_Bitcoin_transaction_.28inside_a_block.29.

CryptoCompare. [Online] 28. September 2017. https://www.cryptocompare.com/coins/guides/what-is-the-bitcoin-genesis-block/.

Dannen, Chris. 2017. Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain. Brooklyn, New York, USA : Apress, 2017, s. 47-48.

David LEE Kuo Chuen, Robert H. Deng. 2017. Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2. *ChinaTech, Mobile Security, and Distributed Ledger*. s.l. : Academic Press, 2017, s. 168.

Drescher, Daniel. 2017. Blockchain Basics: A Non-Technical Introduction in 25 Steps. *1st ed. Edition.* Frankfur am Main,Germany : Apress, 2017, pp. 80-81.

Fleming, Stephen. 2017. Blockchain Technology. *Introduction to Blockchain Technology and its impact on Business Ecosystem* . s.l. : Pronoun, 2017.

Franco, Pedro. 2015. Understanding Bitcoin. *ryptography, Engineering and Economics (The Wiley Finance Series) 1st Edition.* West Sussex : Wiley, 2015, s. 77.

Gerard, David. 2017. Attack of the 50 Foot Blockchain: Bitcoin, Blockchain, Ethereum & Smart . 2017, s. 12-13.

Ghassan Karame, Elli Androulaki. 2016. Bitcoin and blockchain security. Norwood,MA : Artech House, 2016, s. 126.

EthereumREMIX.https://remix.readthedocs.io/en/latest/. [Online]

Hacktrophy. [Online] [Dátum: 18. February 2018.] https://hacktrophy.com/preco-je-

blockchain-zasadna-inovacia/.

Laurence, Tiana. 2017. Blockchain For Dummies. New Jersey : Wiley Brand, 2017, s. 58-59.

Mathis, Tim. 2017. Ethereum. *Your Guide To Understanding Ethereum, Blockchain,and Cryptocurrency.* 2017, s. 19-20.

Modi, Ritesh. 2018. Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain. Birmingham : Packt Publishing Ltd, 2018, s. 15-16.

Morabito, Vincenzo. 2017. *Business Innovation Through Blockchain: The B3 Perspective.* s.l. : Springer, 2017.
—. Business Innovation Through Blockchain: The B3 Perspective. Milan : s.n., s. 10-11.

MuniCZ.https://is.muni.cz/el/1431/jaro2012/M0170/33770420/referat_Batorova.pdf?lang=cs.

Phillip Rogaway, Thomas Shrimpton. 2004. Fast Software Encryption. *Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance.* 388 : Springer, 2004, s. 371.

S. Asharaf, S. Adarsh. 2017. Decentralized Computing Using Blockchain Technologies and Smart Contracts: Emerging Research and Opportunities. *Advances in Information Security, Privacy, and Ethics.* Hershey : IGI Global, 2017, s. 44-47.

Swan, Melanie. 2015. Blockchain. *Blueprint for a New Economy.* Sebastopol : O'Reilly Media, Inc., 2015, s. 20.
Talking, A. Pfitzmann and M. Hansen. A terminology for. 2010. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Managment. 2010.

Wilmer, Conrad Barski and Chris. 2014. Bitcoin for the Befuddled. San Francisco : No starch press, 2014, s. 212-214.
2017**.** www.muni.cz. [Online] 12. Decembeer 2017. [Dátum: 10. February 2018.]

Zhao, J.L., Fan, S. & Yan, J. 2016. Springer Berlin Heidelberg. [Online] 2016. [Dátum: 15. February 2018.] https://doi.org/10.1186/s40854-016-0049-2.

PRÍLOHA

Príloha 1

CD

CD obsahuje:

- Zdrojový kód programu
- Súbory potrebné spustenie programu
- Internetovú aplikáciu