Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Technologie blockchain v oblasti e-business

Bakalářská práce

Filip Šafanda

Školitel: doc. Ing. Ladislav Beránek CSc. MBA.

České Budějovice 2020

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Filip Šafanda
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika

Katedra/ústav PřF JU, kde bude práce vypracována a obhájena: Ústav aplikované informatiky

Školitel: doc. Ing. Ladislav Beránek, CSc.
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PřF JU: .....
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant: .....
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce:
Technologie blockchain v oblasti e-business

Cíle práce:

Cílem práce je analyzovat technologii blockchain a chytrých kontraktů a možné použití těchto technologií zejména v oblasti e-business. V praktické části bude vytvořen vlastní blockchain na bázi platformy Ethereum, a bude vytvořena ukázka nasazení vlastního chytrého kontraktu ve zvolené oblasti e-business. Dále bude následovat celkové vyhodnocení výsledků a formulace závěrů a doporučení

Základní doporučená literatura:

ANTONOPOULOS, A.M., WOOD, G. Mastering Ethereum: Building Smart Contracts and DApps. Kalifornie: O'Reilly Media, Inc., 2018. ISBN 978-1491971949.
NARAYANAN, Arvind. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton: Princeton University Press, 2016. ISBN 978-0691171692.
Chytrý kontrakt Etherium. *Crypto Brain: Váš průvodce kryptoměna ve světě těžbě a obchodování* [online]. 2018, 22.8.2018 [cit. 2019-04-28]. Dostupné z: https://cryptobrain.info/cs/smart-kontrakty-na-etherium/
Blockgeeks. *How to Write an Ethereum Election Smart Contract* [online]. [cit. 2019-04-28]. Dostupné z: YouTube: https://www.youtube.com/watch?v=AItPWZ63_mw

Financování práce .....

Školitel práce doc. Ing. Ladislav Beránek, CSc .....podpis: .....

U externích vedoucích fakultní garant práce .....podpis: .....

Garant oboru bak. studia (nepožaduje se u oboru biologie) .....podpis: .....

Vedoucí katedry/ústavu PřF JU, kde proběhne obhajoba.....podpis: .....

Případný souhlas vedoucího ústavu AV .....podpis: .....

V Českých Budějovicích dne 2.5.2019 .....Podpis studenta .....

Bibliografické údaje

Šafanda F., 2020: Technologie blockchain v oblasti e-business. [Blockchain technology in e-business. Bc.Thesis, in Czech] – 55 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato bakalářská práce se zabývá technologií blockchain, chytrými kontrakty a možnostmi použití těchto technologií zejména v oblasti e-business. V teoretické části bude představena technologie blockchain, její funkcionalita a vlastnosti, dále bude představena technologie chytrých kontraktů, její význam pro kryptoměnu Ethereum a využití těchto technologií v oblasti e-business. Dále následuje praktická část, ve které již dojde k vytvoření vlastního blockchainu na bázi platformy Ethereum, a dále k nasazení vlastního chytrého kontraktu. V závěrečné fázi dojde k vyhodnocení výsledků a interpretaci závěru.

Anotation

This bachelor thesis pursues blockchain technology, smart contracts and use possibilities of these technologies particularly in e-business field. In theoretical part, there will be introduced blockchain technology, its functionality and features, furthermore, there will be introduced smart contracts technology, its importance for Ethereum cryptocurrency, and use of all these technologies in e-business field. The practical part of this thesis will be concerned about creating our own blockchain, based on Ethereum platform, and afterwards implementation of our very own smart contract will happen. In the final phase of this thesis, there will be interpretation of overall results.

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 20. 5. 2020

Filip Šafanda

Poděkování

Na úvod bych rád poděkoval mému vedoucímu práce panu doc. Ing. Ladislavu Beránkovi CSc. MBA. za čas věnovaný při vedení této práce, cenné rady a věcné připomínky.

Obsah

1. Úvod

V úvodní částí si rozebereme téma jako celek, cíle, kterých chceme dosáhnout a prostředky, které použijeme pro dosažení stanovených cílů. Dále dojde k představení osnovy celé práce, jak teoretické, tak i praktické části.

1.1 Cíl práce

Cílem práce je provést analýzu technologie blockchain a chytrých kontraktů. Nejprve dojde k detailnímu seznámení se s technologií blockchain, její historií, funkcionalitou, výhodami a nevýhodami. Seznámíme se s dosavadním využitím technologie blockchain a také její budoucností.

Posléze si představíme technologii chytrých kontraktů, definici, historii a chytré kontrakty ve spojitosti s kryptoměnou Ethereum. Dojde k představení pojmu e-business a k analýze možných způsobů využití technologie chytrých kontraktů v oblasti e-business.

V praktické části dojde k vytvoření vlastního blockchainu na bázi platformy Ethereum a poté k nasazení vlastního chytrého kontraktu ve zvolené oblasti e-business.

1.2 Použité nástroje a metody

Při zhotovení teoretické části jsem nejprve použil metody zpracování dat textů odborné literatury. V úvahu jsem bral také informace čerpané z odborných diskuzí, internetových fór a z dostupných technických dokumentací.

V praktické části této práce byly získané teoretické informace posléze využity při konfigurování vlastního blockchainu a při návrhu a implementaci chytrého kontraktu.

2. Blockchain

Ačkoliv prozatím není formálně stanovena žádná exaktní definice blockchainu, tak je blockchain v informatice považován za jedinečný druh distribuované decentralizované databáze, ve které je možné uchovávat neustále se rozšiřující počet záznamů (bloků). (2) Každý blok obsahuje kryptografický hash předešlého bloku, timestamp, transakční data a nonce. (4)

Technologie blockchainu u kryptoměn funguje na principu peer-to-peer sítě, která je složena z tisíců nodů, tzn. počítačů z celého světa. Nody se mohou libovolně připojovat do sítě, a odpojovat, aniž by to narušilo funkcionalitu celistvé sítě. Každému nodu, který se do sítě připojí, se vytvoří kopie záznamů z celé sítě. V blockchainu je vše schvalováno pomocí konsensu všech nodů (uživatelů). Záznamy jsou chráněny vůči neoprávněným vlivům z vnější strany, tak i ze strany samotných uzlů peer-to-peer sítě. (1)

Nejběžnější využití technologie blockchain je použití jej jako transparentní účetní knihu (tzv. ledger) kryptoměn, kde budou trvale uschovávány transakce provedené uživateli.

2.1 Historie

První počin, který předchází vzniku technologie blockchain, se objevil již roku 1991, kdy výzkumní pracovníci Stuart Haber a W. Scott Stornetta představili výpočetně praktické řešení pro časové známkování digitálních dokumentů, takové, aby dokumentům nebylo možné zaměnit časové údaje a byly naprosto nezmanipulovatelné. (1)

Tento systém využíval kryptograficky zabezpečené řetězce bloků k uložení dokumentů označených časovými známkami, přičemž v roce 1992 byla do tohoto systému implementována navíce funkcionalita Merkle trees, která způsobila výrazné zlepšení efektivity, protože umožňovala najednou ukládat více dokumentů do jednoho bloku. (11) Tato technologie zůstala nevyužitou až do roku 2004, kdy byla patentována, a to 4 roky před příchodem dnes již nejznámější kryptoměny Bitcoin.

V roce 2008 dosud neznámý člověk (možná i skupina lidí) vystupující pod pseudonymem Satoshi Nakamoto vydal tzv. whitepaper, elektronický dokument, kde vysvětlil funkcionalitu Bitcoinu, který byl nejprve představen jako elektronický platební systém, který fungoval právě na technologii blockchainu. Na začátku roku 2009 Satoshi Nakamoto uvolnil zdrojový kód Bitcoinu, ve kterém byl samotný blockchain již zakomponován. (6)

Zprvopočátku byl blockchain užit jen v Bitcoinu, a byl od něj neoddělitelný. Později však začali vznikat nové kryptoměny, s vlastním převzatým blockchainem, který byl použit původně v Bitcoinu. Po určité době si lidé uvědomili, že blockchain lze využít i odděleně bez samotného Bitcoinu, tudíž vzrostl zájem o technologii blockchain jako takovou, a postupně vznikala i nová uplatnění této jedinečné technologie. (1)

2.2 Funkce blockchainu

Jak již bylo řečeno, blockchain je databáze, ve které je možné uchovávat záznamy, které nebude možno později měnit. Tyto data jsou ve své podstatě chráněná seskupení bloků informací řetězených za sebe. Dohromady utváří neměnnou „účetní knihu“, distribuovanou mezi zúčastněné nody pomocí peer-to-peer sítě. Tyto nody jsou výpočetní platformy, které komunikují s uživateli.

Ve struktuře blockchainu rozeznáváme dva druhy nodů.

Full node - každý full node je součástí peer-to-peer sítě, uchovává celistvou kopii blockchainu, je schopen spouštět transakce a rozšiřuje síť bloků. (2) Full nody jsou sobě ekvivalentní v mezích funkcionality.

Partial node - jsou součástí peer-to-peer sítě, ale neuchovávají v sobě kopii blockchainu, tudíž využívají služeb full nodů ke spouštění transakcí. Dále nerozšiřují síť bloků. (2)

Účel blockchainu je sdílení dat napříč všemi uživateli, kteří mají k blockchainu přístup skrze aplikaci. Tito uživatelé také verifikují všechny transakce určitými vnitřními algoritmy. Tyto algoritmy svojí podstatou představují decentralizovanost blockchainu, kdy můžeme říci, že

blockchain není regulován žádnou konkrétní entitou, ale samotnými uživateli a jejich počítači rozmístěnými po celém světě. (2)

Přístup k těmto datům týkající se zápisu a čtení může být buď naprosto neomezený, nebo i s určitými omezeními. Sdílená data jsou chráněna proti modifikaci, rozumějíce jakýkoliv sebemenší zásah by byl jednoduchým způsobem a okamžitě detekován. (15) Z těchto důvodů, jakmile se informace zapíše do blockchainu, je považována za neměnnou.

Celkově rozeznáváme tři druhy blockchainů.

Public

Tento typ blockchainu je konstruován tak, aby byl naprosto decentralizovaný. Žádná entita tudíž nezasahuje do toho, jak jsou transakce v blockchainu uchovávány, a ani to, v jakém pořadí se transakce postupně zpracovávají. Public blockchain jde také velmi těžko globálně zcenzurovat, z toho důvodu, že možnost připojit se do sítě je umožněna komukoliv, bez omezení lokality uživatele, národnosti, a podobně. (3) Výše zmíněné vlastnosti by způsobily autoritám extrémní obtíže při pokusu o vypnutí nodů, a posléze celého blockchainu.

Private

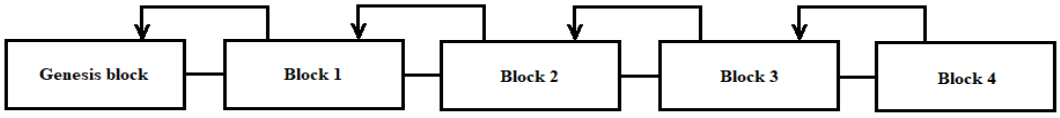
Zde je blockchain v několika bodech odlišný od předešlého public blockchainu. Uživatelé, kteří chtějí být součástí blockchain sítě, musí být odsouhlaseni k přístupu. Transakce jsou neveřejné a jsou k nahlédnutí pouze uživatelům, kterým byl přidělen přístup do sítě. Blockchain tohoto typu se vyznačuje větší mírou centralizace, než je tomu u public blockchainu. (3) Private blockchain je využíván společnostmi, které mají zájem o spolupráce v rámci vícero subjektů, a zároveň nechtějí, aby jejich citlivá data byla veřejně přístupná v rámci public blockchainu. Entity provozující private blockchain poté mají přirozeně větší kontrolu nad uživateli sítě a nad její strukturou.

Hybrid

Hybridní blockchain využívá výhody neveřejnosti a náležitosti udělování oprávnění při přístupu do sítě, a zároveň využívá prvky bezpečnosti a transparentnosti převzaté z public blockchainu. Umožňuje společnostem jistou možnost výběru toho, která data mohou být veřejně přístupná, a která mají být naopak ponechána jakou soukromá. Čerpá z toho, že

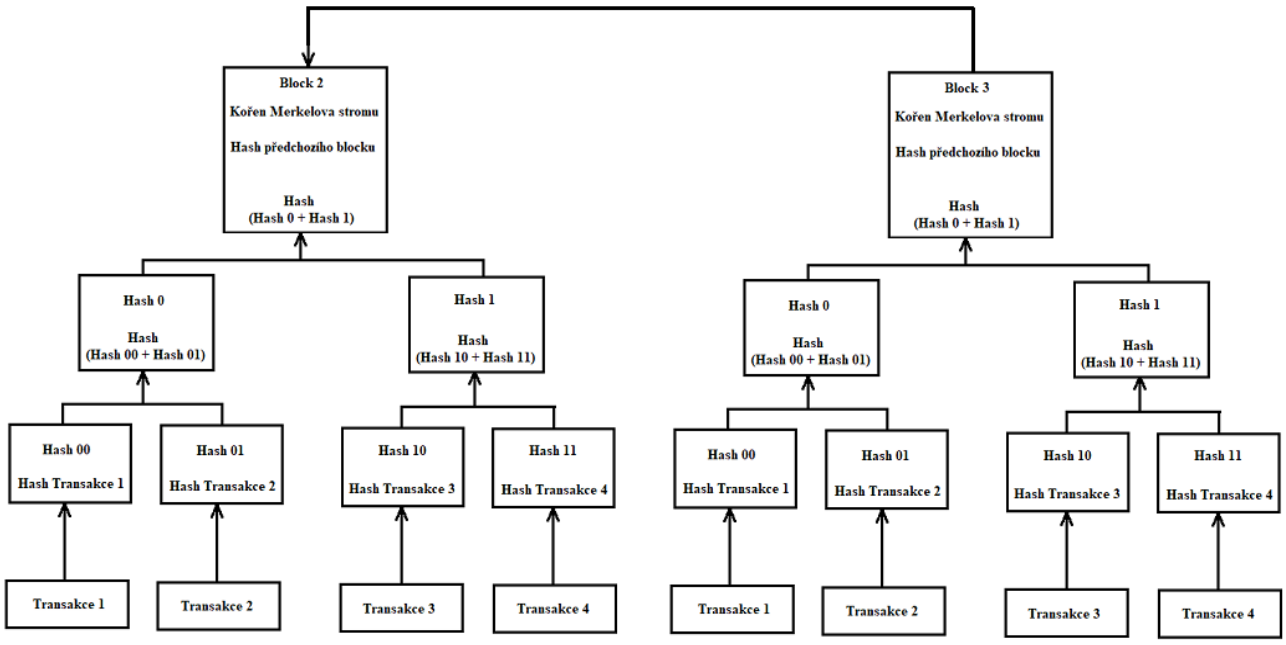
umožňuje pomocí patentované technologie Interchain jednoduché připojování se k jiným protokolům blockchainu. (3)

Jak již bylo řečeno, data se v blockchainu uchovávají formou transakcí, seskupených do bloků. První kdy vytvořený blok v blockchainu, nese název Genesis Blok, jinak udáváno jako „blok 0“. Tento blok je zakódovaný v software, není tedy referencován žádným blokem před ním. (50) První blok se vytvoří hned po úvodní inicializaci Genesis Bloku, a svým pořadím se utvoří hned za ním, posléze přibývají další bloky, a tím se tvoří řetězce.



Obrázek 1 - schéma logiky řetězení bloků

Každý blok obsahuje zahashované informace předešlého bloku, aby bylo zajištěno kryptografické ochrany. Blockchain používá hashovací algoritmus SHA256. (4) Tato hashovaná informace se skládá z dat a digitálního podpisu předešlého bloku a z hashe všech předešlých bloků jdoucích až k samotnému Genesis bloku. Tento systém se jinak nazývá Merkle tree. Jedná se o stromovou strukturu fungující na principu opakovaného hashování transakcí do párů, dokud nevznikne konečný hash, tzv. kořen. Ten bývá zakódován v hlavičce bloku. (11)



Obrázek 2 - zobrazuje bloky v procesu hashování pomocí Merkle tree

Kvůli bezpečnosti bloků každý blok obsahuje informace o hlavičce předešlého bloku, tímto obsahuje také informace o hashi ze všech předešlých bloků. Při útoku a neoprávněném průniku k datům v bloku uvnitř blockchainu, a pokusu o změnu dat, by musel útočník provést změny ve všech blocích následujících po prvotním zasaženém bloku. (17)

2.3 Blockchain bitcoinu

První úspěšná implementace technologie blockchain proběhla při vzniku kryptoměny Bitcoin. Kdy někdo pod přezdívkou „Satoshi Nakamoto“ se rozhodl užít právě tuto technologii jako základní stavební kámen pro Bitcoin. Blockchain byl využit jako veřejná databáze pro ukládání dat, konkrétně pro ukládání všech transakcí, které se poté ukládají v podobě bloků a řetězí se za sebe.

Pro šifrování bloků blockchain využívá SHA256, z rodiny SHA – 2 hashovacích funkcí. Která vytváří ze vstupních dat výstup (otisk) fixní délky. Její hlavní vlastností je, že ze znalosti otisku je prakticky nemožné rekonstruovat vstupní data. Malá změna na vstupu vede k velké změně na výstupu, tj. k vytvoření zásadně odlišného otisku. Ke každému takovému bloku se přiřadí tzv. timestamp, což je soubor znaků, který znázorňuje právě samotný vznik každého bloku a také hash předešlého bloku, který obsahuje tzv. nonce. (4)

Tento blockchain nedokáže fungovat sám o sobě, k jeho funkcionalitě je zapotřebí přítomnosti koncových uživatelů a také tzv. miners - těžařů.

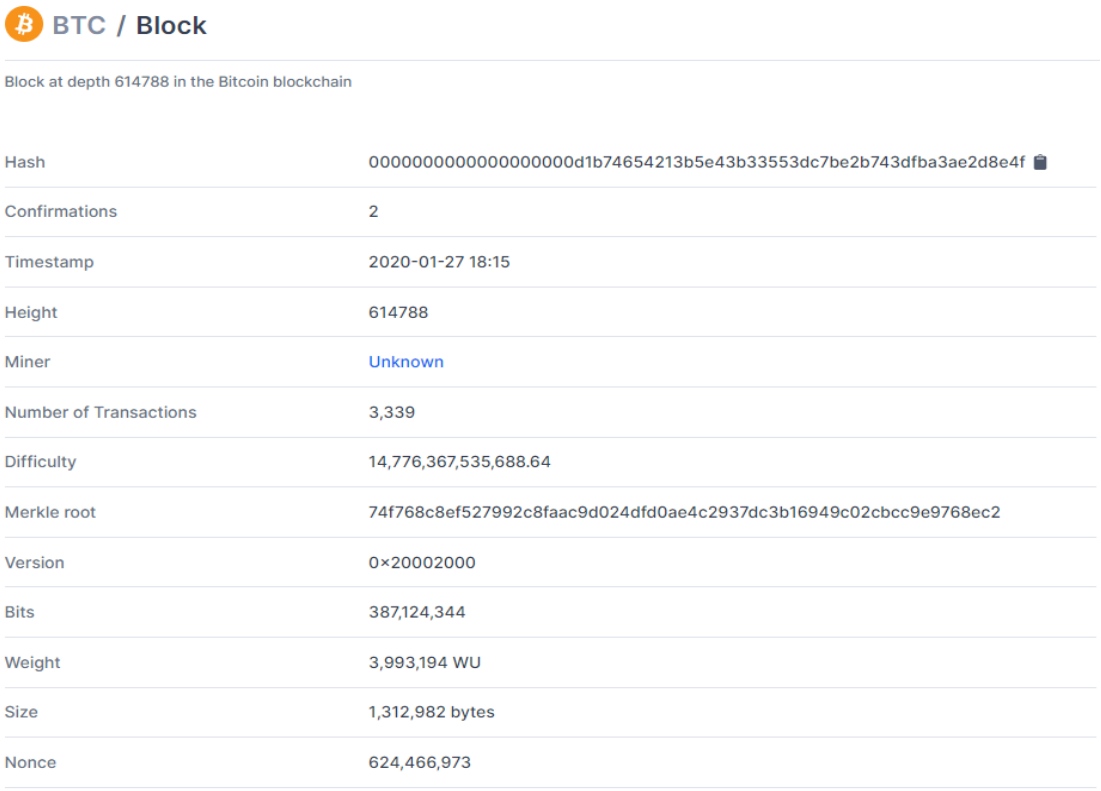
2.3.1 Koncoví uživatelé

Jsou uživatelé, kteří mají své osobní peněženky, které slouží jako adresy pro platby. Uživatelé si také udržují distribuovanou databázi všech proběhlých transakcí v síti – blockchain. Každý uzel tedy ví, kolik měny náleží dané peněžence, jelikož tato databáze je plně transparentní. (2) Každá peněženka je tvořena soukromým a veřejným klíčem. Posílání měny mezi peněženkami probíhá tak, že uživatel si vytyčí určitý obnos virtuální měny (musí také počítat s poplatkem za provedení transakce), vytvoří transakci, a tu podepíše svým privátním klíčem. Tuto transakci pošle prostřednictvím sítě všem uzlům, ke kterým je připojen, a tyto uzly to posléze pošlou dalším uzlům, až se informace rozšíří tímto způsobem po celé síti. Informace o

nadcházející transakci se k adresátovi dostane prakticky ihned, ale prozatím ještě není potvrzena.

2.3.2 Těžaři

To může být kdokoliv, kdo se chce procesu těžby zúčastnit. Tímto procesem potvrzují transakce právě uživatelům předtím, než budou odsouhlaseny, a posléze odeslány adresátovi na cílovou peněženku. (12) Tito lidé mají za úkol hledat takový blok, který splňuje podmínky stanovené v již zmiňované nonce. Nonce je zkratka pro výraz number only used once. Je to 4 bajtová informace obsažená v hlavičce každého bloku, ve které jsou nastaveny parametry pro těžaře. Těžaři poté hledají takovou nonce, aby se hash SHA-2 nového bloku vešel pod sítí stanovený limit, definovaný právě v hlavičce každého těženého bloku. Obtížnost nalezení takové nonce se zvyšuje s tím, jak přibývá výpočetní síla celé sítě. (5)



BTC / Block

Block at depth 614788 in the Bitcoin blockchain

Hash	00000000000000000d1b74654213b5e43b33553dc7be2b743dfba3ae2d8e4f
Confirmations	2
Timestamp	2020-01-27 18:15
Height	614788
Miner	Unknown
Number of Transactions	3,339
Difficulty	14,776,367,535,688.64
Merkle root	74f768c8ef527992c8faac9d024dfd0ae4c2937dc3b16949c02cbcc9e9768ec2
Version	0×20002000
Bits	387,124,344
Weight	3,993,194 WU
Size	1,312,982 bytes
Nonce	624,466,973

Obrázek 3 - transakce č. 3 339 bloku č. 614 788 v Bitcoin blockchainu a transakční detaily

Tento princip byl pojmenován jako Proof-Of-Work, jedná se o logiku zavedenou Satoshi Nakamotem, ve které definuje způsob, jak odměňovat uživatele, kteří provedli činnost ve prospěch sítě. Těžaři dostávají odměny v podobě transakčních poplatků a odměny za

potvrzení bloků, ve formě samotné měny bitcoin, která jim je po sléze připsána na jejich peněženku. (12)

2.3.3 Hardware a proces těžby

V procesu těžby se využívá dosavadních osobních počítačů, kde je možno těžbu vykonávat pomocí CPU nebo GPU. Povětšinou se k vykonávání těžby používají počítačové sestavy obsahující několik GPU.

Dnes se již ale začala využívat technologie ASIC minerů. (14) Jedná se zákaznický integrovaný obvod, který byl vyvinut, a posléze se začal vyrábět přímo a pouze pro účely těžby kryptoměny, dokáže tedy řešit matematické úlohy stanovené Proof-Of-Work principem mnohem efektivněji, než těžební sestavy založené na dosavadní architektuře osobních počítačů. Z těchto důvodů umožňuje těžařům využít větší množství výpočetního výkonu za využití menšího množství spotřebované elektrické energie, a to vede k získání většího obnosu kryptoměny ve formě odměn. (12)

Obrázek 4 - sestava určená pro těžbu Bitcoinu založená na dosavadní architektuře osobních počítačů. (zdroj 53)

ASIC minery bývají většinou digitální obvody ve standardní křemíkové CMOS technologii. Někdy se však při návrhu používají smíšené obvody, tzn. obvody, které obsahují jak digitální, tak i analogové obvody. (14)

Obrázek 5 - ASIC miner používaný k těžbě kryptoměny (zdroj 52)

2.3.4 Potvrzování transakcí

Přibližně každých deset minut se v Bitcoin blockchainu vytvoří nový blok, který se posléze zařazuje za již existující bloky, které svým řetězením tvoří právě samotný blockchain. Tyto bloky jsou utvářeny transakcemi jednotlivých uživatelů, a transakce musí být schváleny procesem těžby těžařů. Za bezpečně schválenou transakci v rámci Bitcoinu můžeme oficiálně považovat transakci, která prošla úspěšně procesem schvalování šest krát, tudíž byla šest krát schválena. To v praxi znamená, že za oficiálně schválenou transakci se v blockchainu zařadí úspěšně ještě dalších 5 bloků. (7)

V rámci blockchainu dochází k událostem spojených se změnami v software protokolu sítě. Tento protokol ovlivňuje parametry celé sítě a pravidla kryptoměn. Pokud dojde k implementaci aktualizace v rámci bitcoinového klienta, poté je již otázka samotných těžařů a provozovatelů uzlů, aby si stáhli a nainstalovali nejaktuálnějšího klienta. Tyto změny se nazývají tzv. forky. (13)

Rozlišujeme dva typy forků:

Soft fork – soft fork je zpětně kompatibilní změna pravidel, kdy aktualizované uzly zvládají přijímat transakce vytvořených podle starších pravidel zavedených neaktualizovanou verzí bitcoinového klienta. (13) Neaktualizované uzly, ale nemají přístup k nové funkcionalitě sítě, která byla zavedena nejnovější aktualizací.

Hard fork – jedná se o zpětně nekompatibilní úpravu pravidel. V tomto případě aktualizované uzly nepřijímají transakce vytvořených podle pravidel neaktualizované verze bitcoinového klienta. Všechny uzly musí provést náležité aktualizace na nejaktuálnější verzi bitcoinového klienta, aby mohli zůstat součástí bitcoinové sítě. Za celou historii bitcoinového blockchainu došlo pouze k jedinému hard forku (rok 2013). (13)

2.4 Výhody a nevýhody blockchainu

Značné množství nadšenců do této technologie je přesvědčeno, že se jedná o další digitální revoluci. Vlastně jim nemůžeme upřít pravdu. S příchodem blockchainu dochází ke změnám v dosavadních systémech, odpadá potřeba tzv. middlemana, vznikají decentralizované digitální měny, a mnoho dalšího. Přes veškeré události spojené s příchodem této jedinečné technologie, a i přes také nekončící vyzdvihování všech charakteristických atribut blockchainu ve formě superlativů, mnoho z těchto výše zmiňovaných nadšenců opomíná patřičně vyhodnotit jak kladné prvky této technologie, tak samozřejmě také i ty záporné.
V první řadě si nejprve představíme několik klíčových vlastností této technologie, které bychom mohli považovat za výhody, posléze si rozebereme i některé ze zásadních záporných vlastností.

2.4.1 Výhody

Decentralizovanost

Technologie blockchain je decentralizovaná technologie, tento charakteristický rys je jednoznačně hlavním benefitem této technologie. Proč je pro nás právě decentralizovanost tak důležitá? Odpověď je velice jednoduchá. Na základě této vlastnosti zaniká nutnost spolupráce

s prvky třetích stran, anebo odpadá nutnost zákroků administrátorů. To znamená, že systém pracuje bez zásahu jakékoliv regulační či řídící entity. (9)

Neměnnost

Neměnnost je dosahována pomocí transakcí, které se schvalují pomocí rozesílání informací po celé síti a jejich postupného ověřování. V okamžiku, kdy se transakce uloží do blockchainu ve formě bloku, se stávají tato data neměnnými, rozumějíc, že s tímto způsobem uloženými daty, již v budoucnu nebude možné provádět jakékoliv změny a samozřejmě tyto data nebude možné ani odstranit. (2) Toto však záleží, na konkrétním druhu blockchainu, pokud totiž zvolíme privátní blockchain s jistou mírou centralizace, soubory mohou být i měněny i odstraňovány, z důvodu dostupné možnosti přizpůsobit si parametry sítě vlastním potřebám, a v tomto případě jsou veškerá rozhodnutí uskutečňována pouze jedinou osobou.

Pokud ale hovoříme o blockchainu jako takovém, tudíž o jeho původní myšlence se zachovalou decentralizací (např. blockchain kryptoměny Bitcoin), tak hovoříme o systému, kde jsou veškeré transakce, které se kdy zapíší do blockchainu zkopírovány na všechny počítače připojené do této blockchain sítě. Uživatelé sítě blockchain mají kontrolu nad všemi transakcemi a informacemi. Ke změně nebo odstranění jakéhokoliv záznamu by bylo zapotřebí vysoké množství výpočetního výkonu. (15) Čím více počítačů je připojeno do sítě, tím je síť blockchainu bezpečnější v otázce útoků, čím je méně počítačů připojených do sítě, tím je blockchain samozřejmě vystaven vyššímu riziku. Potenciálnímu útočníkovi se na základě tohoto principu zvyšuje šance, že bude schopen akumulovat dostatečné množství výpočetního výkonu, aby byl v rámci tohoto útoku schopen zastoupit více než 51% výpočetní síly celistvé sítě, které by bylo dostačující pro schválení transakce obsahující instrukce k požadované změně datových informací obsažených v síti. Tento kybernetický útok bývá v literatuře označován jako tzv. 51% útok. (17)

Transparentnost

Transparentnost je dosažena pomocí procesu duplikace transakcí. Jak již bylo zmíněno výše, každá jednotlivá transakce je duplikována a přenesena na všechny jednotlivé počítače v síti daného blockchainu. Každý účastník sítě má možnost kdykoliv nahlédnout do všech transakcí, které kdy byly evidovány v rámci sítě. To také znamená, že veškeré akce provedené

v rámci sítě mohou být zobrazeny kterémukoliv uživateli, tudíž žádná provedená transakce nezůstane bez povšimnutí. (16)

Bezpečnost

Vysoká míra zabezpečení blockchainu závisí na jednotlivých unikátních vstupů prvků, chápaných jako uživatelů vstupujících do sítě. Každému uživateli, který vstoupí do sítě blockchainu je přidělen unikátní identifikátor spojený s jeho osobním uživatelským účtem. Další velice podstatný způsob zabezpečení je právě samotný řetěz kryptograficky hashovaných transakcí ve formě bloků. Vždy v okamžiku vzniku nového bloku, je nutné vypočítat hash hodnoty pro následující blok, který bude následně vytvořen po zařazení posledního bloku. Nový hash musí v každém případě obsahovat hodnotu hashe předešlého bloku. Obecně řečeno, každý nový hash obsahuje informaci o typu hashe, identifikátor bloku, hash předešlého bloku, timestamp, unikátní identifikátor uživatele, úroveň těžaře a v poslední řadě merkle tree, kde jsou obsaženy informace o všech předešlých transakcí a jejich hashů. (17) Hashe jsou generovány automaticky na základě klíče nodu. V tomto případě je nemožné pozměnit jakékoliv informace týkající se kterýchkoliv hodnot hashů.

2.4.2 Nevýhody

Spotřeba elektrické energie

Jako jedním ze zásadních problémů blockchainu je považována nadměrná spotřeba elektrické energie, která je nezbytně nutná k uchovávání záznamů do ledgeru v reálném čase. Pokaždé, kdy dojde k vytvoření nového node, tak okamžitě dochází ke komunikaci mezi novým nodem, a všemi ostatními již v síti působícími nody. Tato funkcionalita přispívá k transparentnosti celé sítě v rámci verifikace transakcí, a také k procesu jejich duplikace.
Těžaři se pokouší maximalizovat výpočetní výkon svých zařízení, která využívají k procesu těžby. (18) Každý node dovoluje značnou toleranci chybovosti systému, to zaručuje 0% nedostupnosti sítě, aby mohla být data navždy ukládána do sítě, aby data byla naprosto neměnná a nezmanipulovatelná, a to vše v reálném čase. Všechny výše zmíněná kritéria ovšem způsobují nadměrnou spotřebu elektrické energie. Jeden z dalších markantních důvodů zvýšené spotřeby elektrické energie je také právě samotné ověřování a potvrzování všech transakcí.

Kriminalita

Anonymita v rámci decentralizovaného blockchainu a v prostředí kryptoměn vzbudila zájem subjektů, kteří toto anonymní prostředí začali využívat jako prostředek pro obtížně dohledatelný způsob plateb za zboží a služby spjaté s ilegální činností. Jako názorný příklad slouží např. internetový černý trh Silk Road. Silk Road byl součástí dark webu a byl spuštěn v roce 2011. Veškeré obchody provedené na této webové stránce byly placeny pomocí kryptoměn. V říjnu roku 2013 byla tato původní stránka uzavřena FBI. Ross William Ulbrich byl posléze obviněn ze založení a administrace stránky, a posléze odsouzen na doživotí bez možnosti propuštění. (19)

Míra volatility

Virtuální kryptoměny postavené nad technologií blockchainu mívají velmi volatilní pozici na obchodních burzách. Jeden z důvodů vysoké volatility je fakt, že decentralizovaná technologie blockchainu, společně s kryptoměnami, jsou dvě naprosto nové technologie na trhu. (20) Ve skutečnosti to má takový dopad, že společnosti, investoři, vlády a jiné další zájmové skupiny svým zájmem či nezájmem výrazně ovlivňují volatilitu kryptoměn.

2.5 Aktuální využití technologie blockchain

Blockchain je stále veřejností považován za velice novou technologii, jejíž úplný potenciál nám pravděpodobně prozatím stále nebyl odhalen. V této části si představíme možné využití této technologie nasazením namísto dosavadních zavedených systémů v určitých oblastech, a také jejíž aktuální využití.

2.5.1 Kryptoměny

První úspěšné využití technologie blockchain bylo právě v rámci kryptoměny Bitcoin. Kryptoměny mohou řešit nepřeberné množství problémů, záleží právě na jejich samotné implementaci. Pokud hovoříme o Bitcoinu, pak tuto kryptoměnu vnímáme jako měnu pro platební transakce. Kryptoměna Ethereum jde velice podobným směrem, navíc s možností využití této sítě jako prostředí pro aplikace chytrých kontraktů a decentralizovaných aplikací. V dnešní době je na trhu takových kryptoměn již nepřeberné množství, pojďme si několik z nich představit.

VeChain – tvůrci této platformy se snaží o vybudování distribuovaného obchodního ekosystému, který umožňuje transparentní posun informací od výrobce až ke spotřebiteli. Pomocí technologie VeChain lze tedy sledovat dodávané položky řetězce, který vás ujistí o pravosti a kvalitě zboží. (21) Uplatnění pro tuto kryptoměnu je převážně v oblastí luxusního zboží. Tento projekt funguje již 3 roky, a někteří zákazníci ho v dnešní době skutečně aktivně využívají. Konkrétně se jedná o firmy původem v zahraničí, jako jsou PWC, Kuehne & Nagel, BitOcean, Fanghuwang. (22)
Jak tato technologie funguje? VeChain kombinuje technologii blockchainu, internetu věcí a vlastního chytrého čipu, který využívá pro sledování položek během jejich životního cyklu. Zákazník má možnost si přímo při koupi zboží ověřit pravost či původ produktu. Tento čip bývá implementován ve formě RFID, QR nebo NFC čipů. Tyto čipy jsou spárovány s jejich blockchainem, na základě neměnných údajů zde vzniká jistota pravosti produktu. (21)

Monero – jedná se o plně decentralizovanou kryptoměnu, která vznikla v dubnu 2014. Na vývoji se podíleli téměř tři stovky vývojářů z celého světa, kteří při vývoji usilovali především o maximální anonymitu a bezpečí pro každého uživatele. Monero také nepatří žádné korporaci, a to znamená, že nemůže být zrušeno žádnou třetí stranou, ani zásahem určitého státu, či úpravou v legislativě. (23) Využívá unikátní technologie, jako jsou například kruhové podpisy, díky kterým je odesílatel k nerozeznání od skupin jiných uživatelů, aj., pomocí jichž zpracovává každou transakci v rámci naprosté anonymity. (24)

Siacoin – tato kryptoměna představuje decentralizovanou platformu pro ukládání dat. Je v provozu již od roku 2015. Uživatelé této platformy si pronajímají úložiště, nebo naopak nabízejí volné místo na datových nosičích ve svých osobních počítačích jako úschovnu pro data ostatních uživatelů. Této funkcionality tvůrci dosáhli za pomocí chytrých kontraktů, které jsou uloženy v Siacoin blockchainu. Chytré kontrakty kontrolují platby za propůjčení volného datového prostoru. (25) Hostitel obdrží platbu až poté, co u sebe přechová data po předem stanovenou dobu. Data se také nesmějí ztratit či nijak poškodit, v takové případě by hostitel odměnu nedostal.

2.5.2 Zdravotnictví

Jeden z více nejkritičtějších případů využití technologie blockchain je právě v oblasti zdravotnictví. Momentálně tato oblast pociťuje úskalí dosavadního centralizovaného přístupu, který vede k neefektivnímu zacházení s pacienty, konkrétně s jejich zdravotnickými záznamy.

Pacienti pociťují tento efekt především v důsledku úschovy a zpracování dat informačními systémy, kde samotní pacienti jsou nuceni přenášet jejich zdravotnické záznamy mezi lékaři na vyžádáni. Za zmínku stojí časté rozdílnosti v rámci datové bezpečnosti, uchovávání a formátování dat za použití rozdílných standardů, což může vést až k problémům integrity samotných dat. (26) Blockchain by mohl poskytnout řešení za pomocí decentralizované databáze, ke které by měli přístup všichni uživatelé dané sítě s potřebnými pravomocemi. Všechna data by také zůstala dostatečně zabezpečena a byla by zpřístupněna pouze uživatelům s potřebnou mírou autorizace.

Veškeré transakce by zůstaly evidovány v rámci blockchain sítě, a bylo by možné do nich kdykoliv by bylo nutné nahlédnout. Pacient by měl všechna potřebná práva a přístupy k jeho zdravotnickým záznamům. V neposlední řadě, zdravotnická data zachována v rámci blockchainu na jednom místě by mohly vést ke značné výhodě např. při provozování vědeckých výzkumných činností. (27) Také by se touto implementací mohlo zamezit potenciální kontraindikaci při podání látky pacientovi, která by mohla reagovat s jinými již užívanými léky, protože by byla data o užívaných látek pacientem nepřetržitě dostupná.

2.5.3 Vzdělávací systém

Další z mnoha možných případů využití technologie blockchain je v oblasti školství, a celkového vzdělávacího systému. Existuje reálná šance, že dojde v průběhu několika následujících let k integraci školních systémů v rámci softwarových řešení postavených právě na této technologii. Rozebereme si potenciální dopad této technologie na již, zavedené systémy, a na možná zdokonalení v procesu vzdělávání, která tato technologie může přinést.

Některé instituce vidí blockchain jako efektivní technologii, která by mohla být využita k uchovávání dat studentů, sledování a využívání jejich informací o jejich akademické působnosti. Blockchain by umožnil studentům rychlý a zabezpečený přístup k jejich záznamům, a také by umožnil sdílení relevantních informací s budoucími potenciálními

zaměstnavateli. (28) Studentům by tedy zanikla povinnost kontaktovat univerzitu, nebo patřičné úřady v případě potřeby získání požadovaných informací týkajících se studentových záznamů.

V případě implementace blockchainu v rámci vzdělávacích institucí, by blockchain nesloužil pouze jako uchovatel záznamů, ale na základě blockchainu by mohla být postavena celistvá platforma určena ke sdílení dat, a v neposlední řadě také jako komunikační prostředek. Komunikace mezi profesory a studenty je považována za klíčovou. Stejně tak důležitá by měla být informovanost studentů v rámci jejich studentských povinností, termíny lekcí či jiné důležité události. V rámci této technologie založené na blockchainu by bylo umožněno kompetentním oddělením sdílet právě tyto důležité informace.

Jako jedna z nejvýznamnějších aplikací v rámci blockchainu ve vzdělávání je považován vývoj vzdělávacích platforem. Jednalo by se o systém, který by využíval blockchain k propojení akademických pracovníků, studentů a producentů obsahu. Škola by mohla tento systém využít k vytvoření takového ekosystému, kde by umožnila studentům získat přístup ke vzdělávacím materiálům, umožnila by jim také sdílet jejich nápady, nebo třeba vlastní projekty, na kterých aktivně spolupracují. (28)

V případě reálné implementace blockchainu právě v rámci této zájmové oblasti by mohlo dojít ke zjednodušení celého procesu v rámci průběhu pracovních pohovorů pro obě strany, jak pro studenta uchazeče, tak i pro zaměstnavatele. (28) Jak je již zmíněno, tento systém by umožnil studentům uchovávat a sdílet jejich akademické informace, a zároveň by tyto data bylo nemožné falzifikovat, jak již vyplývá ze samotné podstaty blockchainu.

2.6 Blockchain v e-business

Od vzniku e-business, se kvůli častým příchodům nových chytrých technologií, dočkala i tato relativně nová oblast obchodu za dobu své existence několika významných revolucí.
S příchodem Bitcoinu společně s technologií blockchain tomu pravděpodobně nebude jinak. Od příchodu těchto technologií uplynulo již několik let. Za poslední dobu se také dostala do povědomí širší veřejnosti, tudíž můžeme očekávat dopad možné integrace v rámci již

aktuálních zavedených systémů, nebo dokonce jejich nahrazení systémy vyvíjených právě na základech blockchainu. Faktem je, že ke dnešnímu datu došlo již k reálné implementaci této technologie v několika odvětvích, jednou z nich je právě e-business. V následujících několika odstavcích bych chtěl rozebrat možný dopad potenciální integrace blockchainu do oblasti e-business, nebo poukázat na změny, které přivedla již uskutečněná implementace této technologie.

2.6.1 Platby

S příchodem blockchainu se obchodníkům naskytlo již nemalé množství příležitostí integrovat blockchain v rámci svého obchodu stejně tak, jako se naskytlo možností pro spotřebitele se přizpůsobit aktivnímu využívání těchto technologií např. při provádění plateb za své objednané zboží. Např. některé mobilní aplikace určené pro zprostředkování plateb dokonce zahrnuli určité kryptoměny jako funkční možnost platby u obchodníků. (29)

Dnes jsou tyto metody plateb uskutečňovány převážně jako forma alternativních plateb nadšenců do nejnovějších technologických trendů. Ačkoliv se tedy jedná, ne o příliš používanou formu transakcí, existují již některé společnosti, které aktivně přijímají platby ve formě Bitcoinu, Etherea a v několika případech i ve formě jiných kryptoměn.

Za zmínku stojí aktuální statistiky a data týkající se první vzniklé kryptoměny Bitcoin, právě ten zaznamenává čím dále tím více potvrzených transakcí, nejaktuálnější data hovoří dokonce o provedení téměř 350 000 bitcoinových transakcí denně.(30) Bitcoin je pouze jednou z mnoha aktuálně existujících kryptoměn. Podle webu CoinMarketCap je ke dni 17. 02. 2020 evidováno 5 127 unikátních kryptoměn na trhu. (32) Jiné webové stránky zabývající se touto problematikou evidují sice rozdílná čísla, stále ale hovoříme o několika tisících unikátních druhů kryptoměn, např. web CoinLore eviduje k totožnému dni 3 653 existujících kryptoměn. (31)

Placení kryptoměnami vnáší do procesu plateb jiné charakteristiky, než jaké známe při platbách pomocí světovými měnami. Při provádění operací se totiž přenáší charakteristické vlastnosti blockchainu právě do té oblasti, kde nastává implementace této technologie.

Například fakt, že blockchain není regulován státem, či jakoukoliv centrální autoritou, způsobuje, že v případě platby pomocí kryptoměny nemá nikdo jiný mimo nakupujícího a prodávajícího možnost manipulovat s provedenou transakcí. Kryptoměny na platformě blockchainu nepodléhají inflaci, a ani nemohou býti znehodnoceny žádnou bankou, tudíž např. v případě kolapsu státní ekonomiky, by byla příslušná státní měna touto situací ovlivněna, ale např. Bitcoin, s ostatními decentralizovanými kryptoměnami, by touto situací ovlivněn nebyl. (33)

2.6.2 Dodavatelský řetězec

Dodavatelský řetězec je dost možná jedna z nejdůležitějších součástí předmětu zájmu každého obchodníka působícího v oblasti elektronického obchodu. Implementací blockchainu právě v rámci dodavatelských procesů by pravděpodobně vyřešilo několik známých problémů.

Blockchain může být využit jako rozšíření řešení problematiky konkrétně v oblasti sledování zásilek, a ukládání záznamů spojených s historií manipulace zásilek. (34) Uchovávání těchto dat v rámci blockchainu by zajistilo zpětnou nezmanipulovatelnost zadaných údajů, informace o zásilkách by také byli zpřístupněny všem autorizovaným uživatelům v rámci sítě nezávisle na jejich geografické lokaci.

V informačním systému pro sledování zásilek vyvíjeném nad technologií blockchainu by uchovávání záznamů a sledování položek bylo velmi jednoduché. K informacím o produktech by mohlo být přistupováno např. pomocí RFID tagů, nebo pomocí obdobných vestavěných sensorů podobných technologií. V těchto čipech by byly uloženy relevantní informace týkající se konkrétní zásilky či produktu. Pohyb jednotlivých zásilek by byl sledován již od jejich samotného vzniku (první záznam v blockchainu), až po jejich aktuální lokaci. Každý záznam by byl zaznamenán v rámci unikátního chytrého kontraktu, kde by se pomocí transakcí cílených na adresu konkrétního chytrého kontraktu vázaného na konkrétní produkt či zásilku přenášela potřebná data. (35) Výše popisované řešení by mohlo být analogicky převzaté a rozšířené až v otázce vývoje skladového informačního sytému, kde by blockchain mohl být použit jako back-end technologie ve využití jako databáze pro uchovávání skladových dat zásilek.

2.7 Budoucnost blockchainu

Blockchain je stále velice nová technologie, která při své cestě bude muset ještě překonat mnoho překážek předtím, než bude moci společnost využít jejího plného potenciálu. Podobně tomu bylo v říjnu, roku 1990, kdy Sir Tim Berners-Lee představil světu tři základní principy technologií, které vedli ke vzniku World Wide Webu, takového, jak ho v dnešní době známe. Koncem roku 1990 byla první webová stránka nahrána na internet. Tuto stránku si lidé mohli zobrazit a čerpat z ní obsažené informace, které byly přenášeny primárně pomocí modemů a telefonních linek. Přenášeny byly, avšak pouze textové informace, jelikož přenos zvukových dat a videosouborů byl nad rámec technologických možností zdejší doby. Webové stránky jako jsou například Google, Youtube a Facebook byly v této době něco, nad čím nikdo ani neuvažoval. Dnešní stádium vývoje blockchainu bychom mohli přirovnat právě ke stádiu internetu v letech 1990. (37)

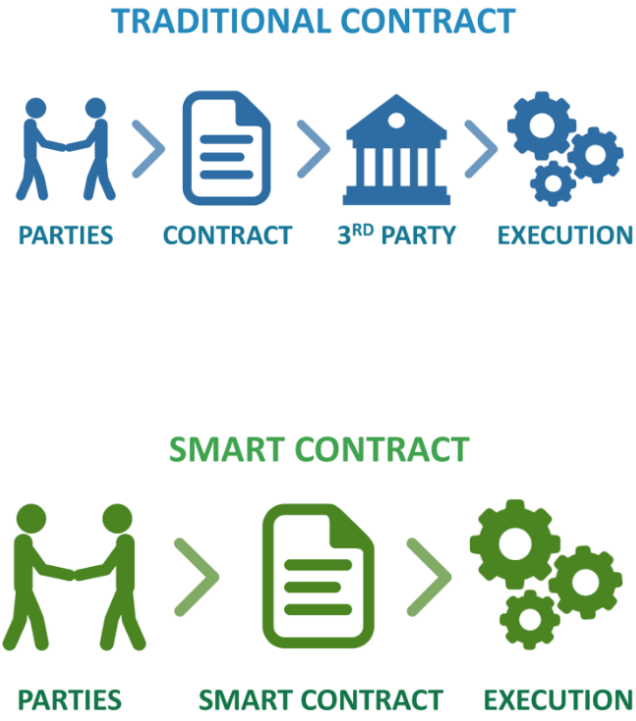
Počátkem příchodu blockchainu, byl blockchain středem zájmu prakticky pouze několika skupin nadšenců, kteří s blockchainem experimentovali. Velkou změnu týkající se vnímání blockchainu veřejnou společností přinesl rok 2019, kdy došlo ke změně přístupu velkých obchodních firem k blockchainu, jako k technologii takové. Benefity, které by mohl blockchain přinést implementací do specifických oblastí, jako jsou např. neměnnost dat, transparentnost, bezpečnost, nezůstaly těmito subjekty bez povšimnutí. (36)
Některé z těchto subjektů začali vnímat tuto technologii jako reálnou, a začali zvažovat možnosti její implementace při řešení obchodních problémů. K dnešnímu dni existuje již nepřeberné množství reálných aplikací této technologie.

Jakým směrem se bude vyvíjet budoucnost blockchainu lze s těží odhadnout. Můžeme však z povahy blockchainu předpokládat, že směr vývoje se bude dělit mezi dva hlavní pilíře. Jedním z nich by mohli být aplikace, které vyžadují decentralizovanost společně s vysokou mírou zabezpečení, a tím druhým pilířem veškerá odvětví, ve kterých bude zastoupena značná část z oblasti umělé inteligence. (36)

3. Chytré kontrakty

Termín chytrých kontraktů byl poprvé zaznamenán již roku 1994 počítačovým inženýrem a kryptografem Nickem Szabo, který chytré kontrakty definoval jako soubor závazků zaznamenaných v digitální formě, specifikovaných pomocí algoritmů, které obsahují protokoly požadující splnění digitálních závazků zúčastněných stran. (38) Praktického využití Szabovo návrhu se svět dočkal až s příchodem technologie blockchain.

Chytrý kontrakt je spustitelný kód, který se spouští v rámci blockchainu. Tyto kontrakty umožňují uzavřít dohodu mezi dvěmi nedůvěryhodnými stranami bez nutnosti využití přítomnosti třetí, důvěryhodné strany. (38)



Obrázek 6 - tradiční kontrakt v porovnání s chytrým kontraktem (zdroj 54)

Hlavním cílem chytrého kontraktu je automatizovat plnění podmínek dohody vždy, když dojde ke splnění kódem specifikovaných požadavků.

3.1 Technologie chytrých kontraktů

Chytrý kontrakt se skládá ze stavu peněženky, soukromého datového pole a spustitelného zdrojového kódu. Každému chytrému kontraktu je přiřazena 20 bajtová adresa sloužící k jeho jednoznačné identifikaci. Po nasazení kontraktu do sítě blockchainu, se zdrojový kód kontraktu stává neměnným. (38) S chytrým kontraktem uživatel interaguje zasíláním transakcí na adresu daného kontraktu. Funkcionalita kontraktu závisí právě na zdrojovém kódu daného kontraktu, pomocí něhož můžeme např. zapisovat nebo číst ze soukromého datového pole, uchovávat finanční prostředky v rámci stavu peněženky chytrého kontraktu, posílat či obdržovat zprávy ostatních uživatelů blockchainu, nebo dokonce i pomocí již nasazeného kontraktu nasazovat další kontrakty, a podobně. (38) Kontrakty můžeme rozlišovat na dva tipy. Prvním z nich je kontrakt nedeterministický, to znamená, že kontrakt po nasazení do sítě vyžaduje čerpání dat ze zdrojů třetích stran, rozumějme tomu jako zdroje alokované mimo danou síť blockchainu. Druhým tipem je deterministický kontrakt, který po dobu svého běhu nevyžaduje přístup k datům umístěných mimo síť blockchainu. (40)

3.2 Chytré kontrakty a Ethereum

Převážná část dokonce i interesovaných jedinců by pravděpodobně označila Ethereum za kryptoměnu. Avšak není tomu tak. Ethereum je totiž veřejná open-source distribuovaná výpočetní platforma postavená na blockchainu, která také mimo jiné umožňuje funkcionalitu chytrých kontraktů. (39) Naopak mluvíme-li o Etheru, tak se jedná právě o kryptoměnu, která je generována platformou Ethereum jako odměna těžařům.

V rámci sítě Ethereum byla integrována tzv. EVM – Ethereum Virtual Machine. EVM můžeme vnímat jako integrované run-time prostředí na každém Ethereum nodu po celé blockchain síti Ethereum, díky němuž máme možnost na této platformě spouštět tzv. EVM bytecode. Bytecode vzniká kompilací vysokoúrovňového jazyka Solidity na formát bytecodu kompatibilního s EVM. (39)

3.3 Chytré kontrakty v e-business

V případě využití chytrých kontraktů v e-business, např. v otázce platebních systémů, mohou být chytré kontrakty naprogramovány tak, aby se jejich stěžejní zdrojový kód spustil až

v případě oboustranného splnění požadavků kontraktu mezi kupujícím a prodávajícím. Představme si modelovou situaci takovou, že zákazník zadá objednávku prostřednictvím webových stránek prodávajícího. Pomocí platební brány však odešle požadovanou částku za objednané zboží na adresu chytrého kontraktu, tyto peníze zůstanou na peněžence chytrého kontraktu až do té doby, než se naplní požadavky obou zúčastněných stran, které budou stanoveny prostřednictvím samotného zdrojového kódu obsaženého v kontraktu. Po naplnění těchto požadavků, se provede automatické spuštění stěžejní části zdrojového kódu uvnitř chytrého kontraktu, na jehož základě budou posléze doposud uložené peníze v rámci peněženky samotného chytrého kontraktu automaticky převedeny na požadovanou peněženku patřící prodávajícímu. Výše popsaný algoritmus je samozřejmě pouhou demonstrací možné funkcionality.

Naše modelová situace samozřejmě pokrývá jen zanedbatelnou část potenciálu využití chytrých kontraktů na bázi blockchainu Ethereum. Podobný systém může být aplikován na velice široké spektrum dílčích procesů v této oblasti, ať už se jedná o organizační procesy spojené s dodavatelským řetězcem, skladováním zásob a jejich evidenci, nebo záležitosti spojené s platbami, a jiné. (41)

```
pragma solidity ^0.4.11;

contract MyContract {
    uint i = (10 + 2) * 2;
}
```

Obrázek 7 – jednoduchá funkce napsaná v jazyce Solidity

```
60606040525b600080fd00a165627a7a7230582012c9bd00152fa1c480f6827f81515
bb19c3e63bf7ed9ffbb5fda0265983ac7980029
```

Obrázek 8 – byte code funkce výše

4. Sledování zásilek

Předmětem praktické části této práce bude vytvoření chytrého kontraktu v oblasti dodavatelského řetězce v e-business. Konkrétně se budeme zaobírat otázkou logistiky, a sice přímo možností implementace technologie blockchainu při procesu sledování zásilek. Před samotným návrhem a implementací chytrého kontraktu bych považoval a vhodné si nejprve stručně analyzovat současné technologie používané při sledování zásilek.

V dnešní době globalizace trhu, se procesy týkající se dodavatelského řetězce rozrůstají do nejrůznějších oblastí, a také se stávají čím dále komplexnějšími. Výrobky se často designují v několika týmech, na různých lokacích, a k výrobě dochází na dalším odlišném místě, přičemž skladové prostory se často nachází rozprostřené stovky, i tisíce kilometrů od výrobních hal. Posléze jsou výrobky jako konečné produkty distribuovány k odběratelům do celého světa. V rámci celého procesu dodavatelského řetězce dochází k transportu zásilek.

Zásilky prochází různými státy, sklady, a jsou také vystavovány vnějším vlivům, jako například variabilitě teplotních hodnot, vlhkosti, ale také tím, jak je se zásilkami manipulováno zaměstnanci dopravních společností. Některé ze společností zabývající se poskytováním infomačních řešení v oblasti sledování zásilek, používají při tvorbě svých produktů technologie, jako jsou např. datové interfacy, elektronická výměna dat (EDI), B2B reporting, čárové kódy, QR kódy, RFID, NFC, GPS. Taková řešení jsou většinou složena ze skladového systému, kde hlavní roli hraje databáze pro evidenci jednotlivých zásilek a RFID čipů, které umožní sledovatelnost polohy zásilky. (43)

Výše zmíněné často využívané technologie, avšak disponují určitými limitujícími faktory. Bývají velmi omezené v oblasti logistických aktivit, např. nepřesné při identifikaci lokace. (43) Některá data se zaznamenávají formou nedigitálních informací, přičemž neexistuje žádný prvek, který by byl schopný ověřit přesnost a integritu tohoto typu dat, např. při záznamu dat v papírově podobě. Často dochází k významným problémům při nutnosti přístupu více rozdílných subjektů ke kompletním sdíleným datům zásilkové společnosti, většinou totiž, dochází-li ke kooperaci více než jednoho dopravců při procesu doručování jedné zásilky, tak každý z těchto subjektů využívá vlastních komplexních systémů řešení otázky dodavatelského řetězce. Nahlédnutí k původním datům dopravce, který se zásilkou operoval dříve, je často

nemožné, stejně tak není zaručena kompatibilita mezi různými systémy dopravci. Nejpodstatnějším nedostatkem je ale nemožnost dosavadních systémů poskytnout všem účastníkům řetězce přístup k transparentním datům napříč systémy. Dostupnost transparentních dat v reálném čase může být zajištěna pomocí kombinace technologií blockchainu s integrací IoT. (42)

4.1 Implementace

V této části se budeme zabývat již samotnou implementací chytrého kontraktu, než však přistoupíme k samotným krokům vedoucím k implementaci, bude nejprve nutné zprovoznit vlastní síť blockchainu na bázi platformy Ethereum. Pro potřeby implementace kontraktu a v rámci demonstrace celého procesu jsem se po zvážení možností rozhodl pro využití vlastní privátní sítě platformy. V několika následujících odstavcích si představíme technologie, které byly použity při praktické části této práce.

4.1.1 Použité technologie

Geth

Jedná se o konzolového klienta psaného v programovacím jazyce Go, který umožňuje uživateli připojení se k protokolu Ethereum a k jeho blockchainu. Uživateli bude poté umožněno provádět těžbu, vést Ethereum účty, odesílat transakce, interagovat s chytrými kontrakty, a podobně. Při spuštění programu Geth, se klient ve výchozím nastavení připojí na hlavní blockchain Ethereum (tzv. Mainnet). Uživatel má také možnost vytvoření vlastního privátního blockchainu pomocí dalších konfiguračních metod, kterých budeme využívat v praktické části této práce. Geth je volně ke stažení na www.github.com. (44)

MetaMask

MetaMask se nazývá plug-in webových prohlížečů, umožňující připojení se na blockchain Ethereum, lze s jehož pomocí efektivněji spravovat Ethereum účty, odesílat, přijímat a potvrzovat transakce, nebo dokonce i spouštět decentralizované aplikace přímo v rozhraní MetaMask. (45)

Remix IDE

Jde o prostředí určené k vývoji chytrých kontraktů za pomocí programovacího jazyka Solidity, je napsáno v programovacím jazyce JavaScript, a autory vydáno jako open source software. Umožňuje také interakci s chytrými kontrakty bez nutnosti použití příkazové řádky. (46)

Solidity

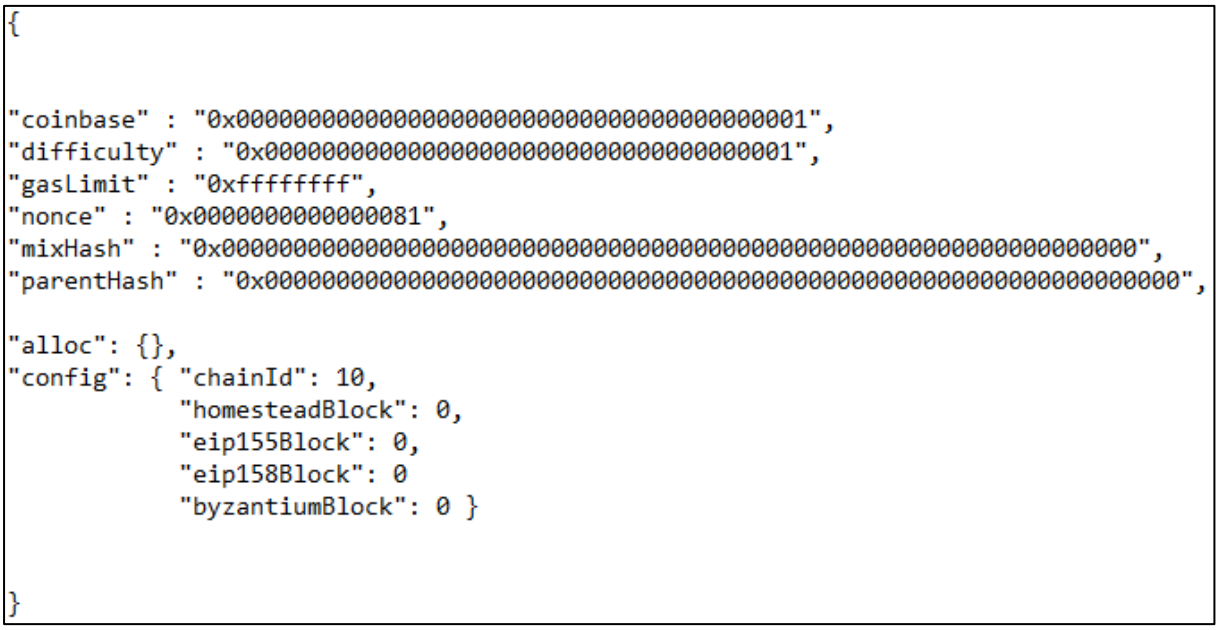
Solidity je objektově orientovaný jazyk určený pro psaní chytrých kontraktů. Používá se k implementaci chytrých kontraktů na několika různých platformách, nejznámější z nich je Ethereum. (47)

4.2 Založení vlastní privátní sítě

Před nasazením chytrého kontraktu dojde k založení vlastní sítě Ethereum. Pokud bychom se rozhodli k využití mainnetu Ethereum platformy, nenacházeli bychom se na síti jako jediní uživatelé. Bylo by také zapotřebí synchronizovat již existující blockchain záznamy v síti se záznamy na našem zařízení, což by mohlo být velmi časově nákladné. Celková velikost všech provedených transakcí zapsaných v blockchainu od doby vytvoření genesis bloku je evidována ke dni 24. 2. 2020 přibližně o velikosti 124,5 GB dat na disku každého node. (48)

Pro zavedení chytrého kontraktu do sítě Ethereum je nutná úhrada transakčního poplatku, museli bychom se tedy aktivně účastnit těžby ethera, nebo bychom byli nuceni směnit požadované množství ethera za naše reálné peníze. (49) Pro výše zmíněné důvody jsem se rozhodl pro použití vlastní privátní sítě platformy.

Prvním potřebným krokem v procesu vytváření naší vlastní sítě bude vytvoření souboru, který slouží jako konfigurační soubor genesis bloku, tudíž prvního bloku naší privátní sítě. Jedná se o soubor s koncovkou „json“, ve kterém se budou nacházet požadující konfigurační data, upravená naším potřebám.



```
{


"coinbase" : "0x0000000000000000000000000000000000000001",
"difficulty" : "0x0000000000000000000000000000000000000001",
"gasLimit" : "0xffffffff",
"nonce" : "0x0000000000000081",
"mixHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",
"parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",

"alloc": {},
"config": { "chainId": 10,
            "homesteadBlock": 0,
            "eip155Block": 0,
            "eip158Block": 0
            "byzantiumBlock": 0 }


}
```

Obrázek 9 – obsah konfiguračního souboru

Konfigurační soubor by měl být uložen ve složce, dedikované pouze k uchovávání dat týkající se naší privátní blockchain sítě. Později budou v totožné složce vytvořeny další soubory obsahující data našeho blockchainu.

Popis jednotlivých atribut konfiguračního souboru: (50)

coinbase: 160- ti bitová adresa, na kterou byly přeposlány veškeré odměny za vytěžení tohoto konkrétního bloku

difficulty: upravuje složitost těžitelnosti bloku, respektive upravuje čas nutný k vytěžení jednoho bloku, čím vyšší hodnota, tím více výpočtů musí těžař provést

gasLimit: maximální stanovená hodnota poplatku za transakci, která nelze překročit

nonce: 64- bitový hash, který společně s mixHash potvrzuje využití dostatečného výpočetního výkonu nutného pro vytvoření tohoto konkrétního bloku

mixHash: 256- bitový hash, který společně s nonce potvrzuje využití dostatečného výpočetního výkonu nutného pro vytvoření tohoto konkrétního bloku
parentHash: 256-bitový hash odkazující na celou hlavičku rodičovského bloku (obsahuje také jeho nonce a mixHash), svými odkazy tedy tvoří řetězec bloků, v případě genesis bloku, který nemá žádný rodičovský, tedy předcházející blok, je tedy tato hodnota nastavena na hodnotu 0

alloc: umožňuje možnost definování peněženek před samotným spuštěním sítě

config: tato sekce v konfiguračním souboru není povinná, lze odstranit z konfiguračního souboru, uvedl jsem jej zde pro příklad dalších parametrů konfiguračního souboru

chainID: identifikátor blockchainu sloužící k rozlišení blockchainů

homesteadBlock: verze používané sítě, Homestead je označení druhé veřejné produkční verze Ethereum (první byla verze Frontier)

eip155Block, eip158Block, byzantiumBlock: úpravy verzí protokolu (forky), není nutné využívat těchto protokolů pro účely této práce, nastavené tedy na hodnotu 0

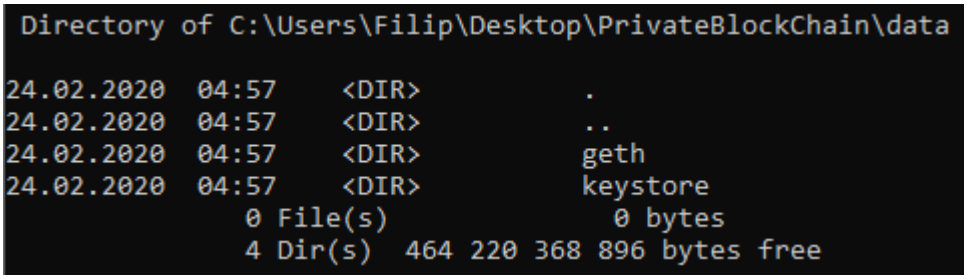
Náš vytvořený genesis json soubor inicializujeme pomocí patřičného příkazu, který spustíme v příkazovém řádku systému Windows. Následující příkaz tedy, inicializuje nový genesis block a definuje vlastnosti sítě, pomocí konfiguračního souboru. První cesta ukazuje na složku, do které chceme ukládat data blockchainu (přepínač --datadir), druhá cesta ukazuje na konfigurační soubor.

```
geth --datadir C:/Users/Filip/Desktop/PrivateBlockChain/data init C:/Users/Filip/Desktop/PrivateBlockChain/genesis.json
```

```
C:\Users\Filip>geth --datadir C:/Users/Filip/Desktop/PrivateBlockChain/data init C:/Users/Filip/Desktop/PrivateBlockChain/genesis.json
INFO [02-24|04:57:02] Maximum peer count                       ETH=25 LES=0 total=25
INFO [02-24|04:57:02] Allocated cache and file handles         database=C:\\Users\\Filip\\Desktop\\PrivateBlockChain\\data\\geth\\chaindata cache=16 handles=16
INFO [02-24|04:57:02] Writing custom genesis block
INFO [02-24|04:57:02] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [02-24|04:57:02] Successfully wrote genesis state         database=chaindata                                                    hash=cb1840…f9e140
INFO [02-24|04:57:02] Allocated cache and file handles         database=C:\\Users\\Filip\\Desktop\\PrivateBlockChain\\data\\geth\\lightchaindata cache=16 handles=16
INFO [02-24|04:57:03] Writing custom genesis block
```

Obrázek 10 – inicializace genesis bloku

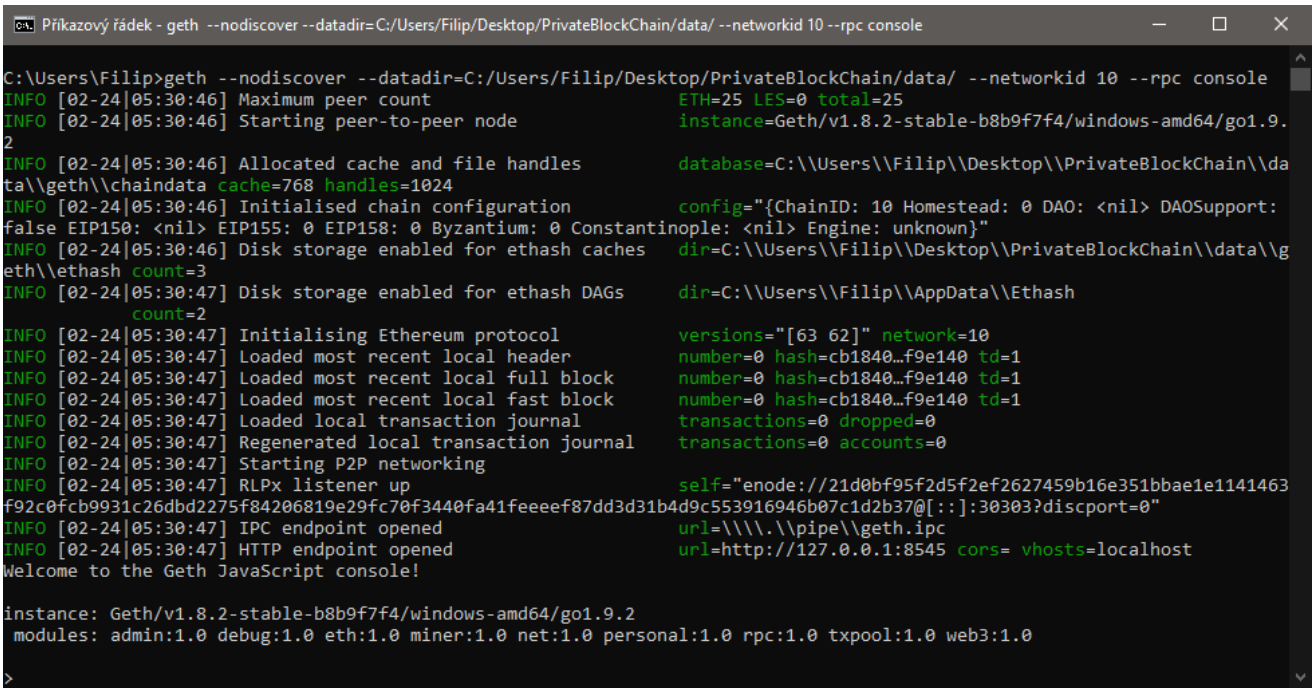
V této fázi bychom již měli mít úspěšně inicializovaný náš genesis block, v případě neúspěšného provedení příkazu, bychom byli upozorněni chybovou hláškou, a odkázáni na patřičnou oblast, kde se v našem konfiguračním souboru vyskytuje chyba. Pokud bychom přeci jen chtěli ověřit, zda příkaz opravdu proběhl správně, můžeme nahlédnout do složky zvolené pro ukládání dat. V této složce by se měli nacházet dvě další složky. Složka geth a keystore.



Obrázek 11 – výpis složky data

Po úspěšné inicializaci genesis bloku, a ověření, zda příkaz opravdu proběhl bez chyb, nastává posun směrem k dalším potřebným krokům. Pomocí následujícího příkazu uvedeme naší privátní síť do běhu.

```
geth --nodiscover --datadir=C:/Users/Filip/Desktop/PrivateBlockChain/data/ --networkid 10 --rpc console
```



Obrázek 12 – uvedení privátní sítě do běhu

Popis jednotlivých přepínačů příkazu: (51)

--nodiscover – znemožňuje možnost automatického objevování a následné připojování peerů v síti

--networkid – nastavíme na hodnotu, kterou jsme uvedli v našem konfiguračním souboru genesis bloku uvedenou jako chainid, v našem případě tedy na hodnotu 10, jedná se o síťový identifikátor

--rpc – spustí rpc interface, který umožňuje přístup klientům k našemu blockchainu (nutné např. k propojení s MetaMask)

console – umožní spouštět příkazy přímo v cmd.exe z příkazové řádky geth

4.3 Založení uživatelského účtu

Po úspěšném založení a spuštění naší privátní sítě však bude zapotřebí založit účet našemu testovacímu uživateli. Tento účet následně využijeme jako těžaře, tudíž jako prostředek pro nasazení našeho chytrého kontraktu. Příkazem „personal.newAccount()“ založíme nový účet uživateli. Účet bude zašifrován heslem, a uložen ve formě souboru do adresáře našeho blockchainu. Po spuštění příkazu, budeme vyzváni k vytvoření hesla. V případě úspěšného provedení příkazu, se posléze na konzoli vypíše adresa uživatele, kterého jsme právě založili. V případě nutnosti lze příkazem „eth.accounts“ vypsat všechny uživatele.

```
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xf9253fc6dec67b3a80ed043326372dcb2914a766"
>
```

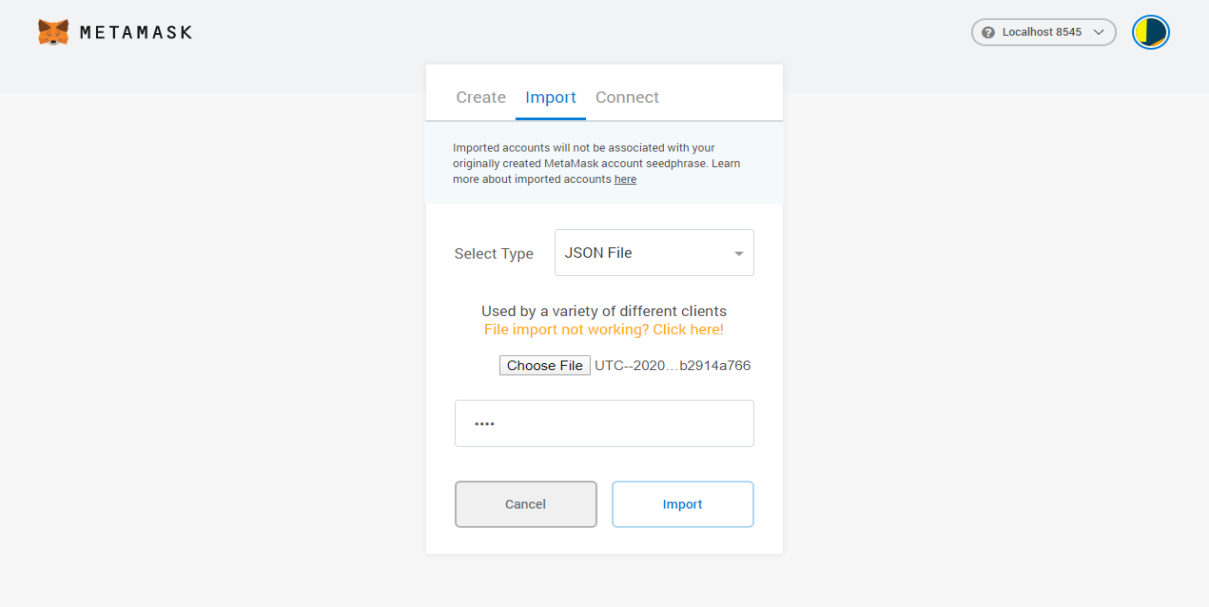
Obrázek 13 – založení účtu uživateli

```
> eth.accounts
["0xf9253fc6dec67b3a80ed043326372dcb2914a766"]
>
```

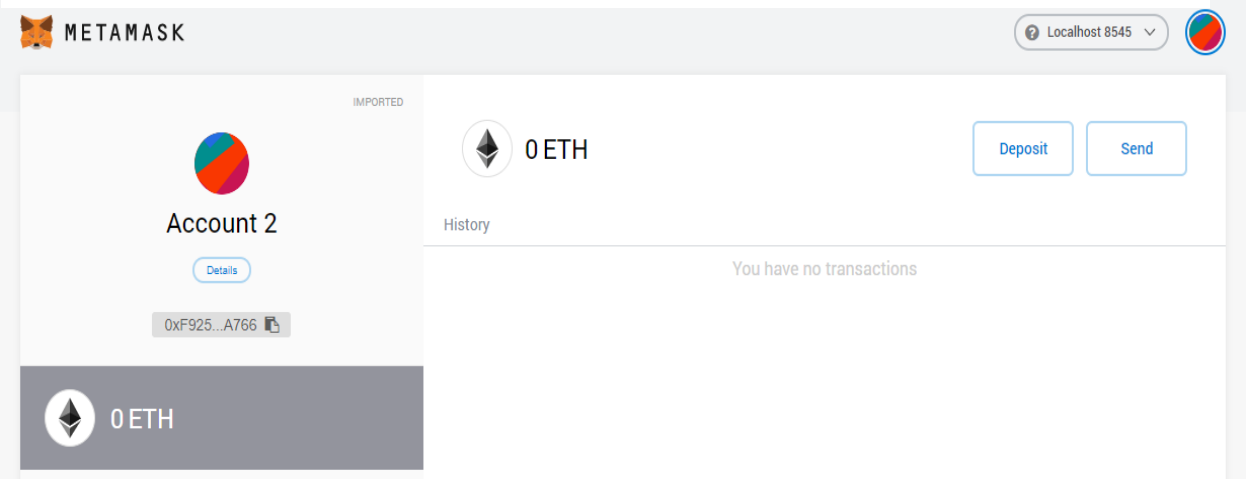
Obrázek 14 – výpis všech uživatelů

4.4 Propojení Geth s MetaMask

Z důvodu zdokonalení interakce s naším vytvořeným testovacím uživatelem, zjišťování zůstatku na jeho peněžence, taktéž s chytrými kontrakty, jejich nasazování, a přehledu potvrzování transakcí, v následujících odstavcích propojíme webový plug-in MetaMask s naší privátní sítí. Po vytvoření účtu, a následného přihlášení se do prostředí MetaMask je nutné, abychom provedli správnou konfiguraci. V nabídce voleb sítí zvolíme tedy položku „Localhost 8545“ a následně importujeme náš šifrovaný soubor uživatelského účtu.



Obrázek 15 – import účtu uživatele do prostředí MetaMask



Obrázek 16 – uživatelské rozhraní MetaMask

4.5 Těžba

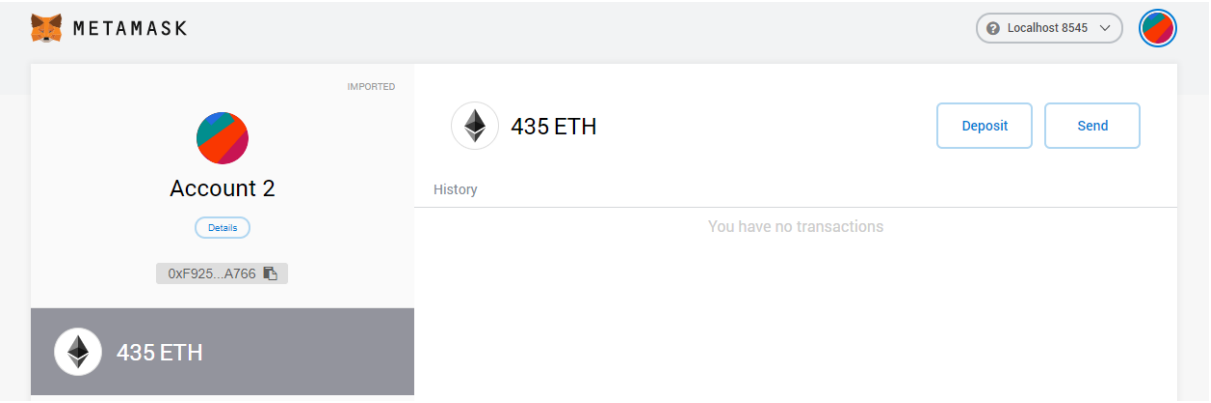
Při spouštění naší sítě, máme nyní spárovaného klienta Geth s uživatelským účtem. Abychom mohli zavádět chytré kontrakty, a posléze s nimi interagovat, je nutné vlastnit dostatečné množství Etherea pro zaplacení transakčních poplatků. Proto tedy pomocí příkazu miner.start() spustíme těžbu kryptoměny. Vytěžené digitální mince se budou připisovat na peněženku patřící uživateli, který je spárován s Geth klientem.
Naše vytěžené digitální mince, avšak nemají žádnou hodnotu, nejedná se totiž o Ether vytěžený na mainnetu Ethereum platformy, ale o Ether vytěžený v rámci naší testovací privátní sítě platformy. V případě, že bychom chtěli těžbu ukončit, použijeme příkaz miner.stop().

```
instance: Geth/v1.8.2-stable-b8b9f7f4/windows-amd64/go1.9.2
INFO [03-01|11:45:21] Etherbase automatically configured       address=0xF9253fc6Dec67B3A80ed043326372Dcb2914A766
coinbase: 0xf9253fc6dec67b3a80ed043326372dcb2914a766
at block: 0 (Thu, 01 Jan 1970 01:00:00 CET)
 datadir: C:\Users\Filip\Desktop\PrivateBlockChain\data
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0
```

Obrázek 17 – v pravé části obrázku lze vidět adresa spárovaného uživatele

```
INFO [03-01|11:59:47] Successfully sealed new block            number=8 hash=19bbb2…4b9a15
INFO [03-01|11:59:47] ⛓ □□ block reached canonical chain        number=3 hash=0c0e48…70735d
INFO [03-01|11:59:47] ⛓ □□ mined potential block                number=8 hash=19bbb2…4b9a15
INFO [03-01|11:59:47] Commit new mining work                   number=9 txs=0 uncles=0 elapsed=0s
INFO [03-01|11:59:48] Successfully sealed new block            number=9 hash=7898b3…ed8950
INFO [03-01|11:59:48] ⛓ □□ block reached canonical chain        number=4 hash=293dff…b68412
INFO [03-01|11:59:48] ⛓ □□ mined potential block                number=9 hash=7898b3…ed8950
INFO [03-01|11:59:48] Generating DAG in progress               epoch=1 percentage=4  elapsed=5.456s
INFO [03-01|11:59:48] Commit new mining work                   number=10 txs=0 uncles=0 elapsed=0s
INFO [03-01|11:59:49] Generating DAG in progress               epoch=1 percentage=5  elapsed=6.932s
INFO [03-01|11:59:49] Successfully sealed new block            number=10 hash=b55bfb…e4d38b
INFO [03-01|11:59:49] ⛓ □□ block reached canonical chain        number=5  hash=f354de…2c13ef
INFO [03-01|11:59:49] ⛓ □□ mined potential block                number=10 hash=b55bfb…e4d38b
INFO [03-01|11:59:49] Commit new mining work                   number=11 txs=0 uncles=0 elapsed=0s
INFO [03-01|11:59:50] Successfully sealed new block            number=11 hash=42a521…a43f5a
INFO [03-01|11:59:51] ⛓ □□ block reached canonical chain        number=6  hash=1d7cf1…8559f3
INFO [03-01|11:59:51] ⛓ □□ mined potential block                number=11 hash=42a521…a43f5a
INFO [03-01|11:59:51] Commit new mining work                   number=12 txs=0 uncles=0 elapsed=343.7µs
INFO [03-01|11:59:51] Generating DAG in progress               epoch=1 percentage=6  elapsed=8.496s
INFO [03-01|11:59:52] Generating DAG in progress               epoch=1 percentage=7  elapsed=10.392s
INFO [03-01|11:59:54] Successfully sealed new block            number=12 hash=4f03b4…1c596d
INFO [03-01|11:59:54] ⛓ □□ block reached canonical chain        number=7  hash=db9d6e…fd7e87
INFO [03-01|11:59:54] ⛓ □□ mined potential block                number=12 hash=4f03b4…1c596d
INFO [03-01|11:59:54] Commit new mining work                   number=13 txs=0 uncles=0 elapsed=53.4µs
INFO [03-01|11:59:54] Successfully sealed new block            number=13 hash=c89209…e6df39
INFO [03-01|11:59:54] ⛓ □□ block reached canonical chain        number=8  hash=19bbb2…4b9a15
INFO [03-01|11:59:54] ⛓ □□ mined potential block                number=13 hash=c89209…e6df39
INFO [03-01|11:59:54] Commit new mining work                   number=14 txs=0 uncles=0 elapsed=287.7µs
INFO [03-01|11:59:54] Generating DAG in progress               epoch=1 percentage=8  elapsed=12.335s
```

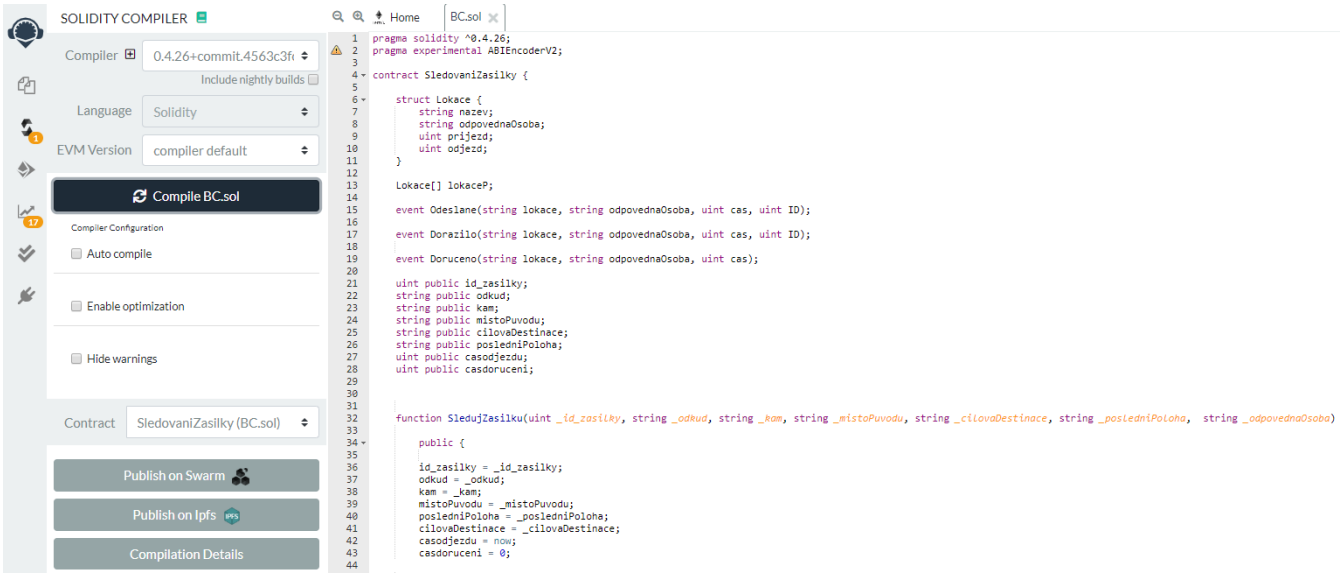
Obrázek 18 – probíhající těžba Etheru zobrazená prostřednictvím Geth konzole



Obrázek 19 – vytěžené Ethery připsané na uživatelském účtu

4.6 Zavedení chytrého kontraktu

Po úspěšném nastavení a spuštění sítě, založení uživatele a natěžení Ethera, které je stěžejní pro úhrady poplatků spojené s transakcemi, se dostáváme ke kroku, ve kterém již dojde k nasazení samotného chytrého kontraktu do sítě. K nasazení chytrého kontraktu, a k interakci s ním, použijeme internetové prostředí Remix, a browserový plug-in MetaMask. Kombinace těchto dvou technologií nám zprostředkuje námi žádanou funkcionalitu. V několika následujících odstavcích si představíme stěžejní části kódu:



Obrázek 20 – prostředí Remix, které bude použito pro nasazení chytrého kontraktu

```
function SledujZasilku(uint _id_zasilky, string _odkud, string _kam, string _mistoPuvodu, string _cilovaDestinace, string _posledniPoloha,  string _odpovednaOsoba)

    public {

    id_zasilky = _id_zasilky;
    odkud = _odkud;
    kam = _kam;
    mistoPuvodu = _mistoPuvodu;
    posledniPoloha = _posledniPoloha;
    cilovaDestinace = _cilovaDestinace;
    casodjezdu = now;
    casdoruceni = 0;

    lokaceP.push(Lokace({
        nazev: _mistoPuvodu,
        odpovednaOsoba: _odpovednaOsoba,
        prijezd: 0,
        odjezd: now
    }));

 emit Odeslane(_mistoPuvodu, _odpovednaOsoba, now, lokaceP.length - 1);

}
```

Obrázek 21 – funkce SledujZasilku určená k prvotnímu vytvoření a zaslání zásilky

Funkce SledujZásilku je úvodní funkce, která slouží k prvotnímu zavedení zásilky, a souběžně také k prvotnímu odeslání zásilky. Uloží do pole lokaceP název lokace, ze které byla zásilka odeslána, jméno odpovědné osoby, která přebrala zásilku k odeslání, čas příchodu zásilky je stanoven na hodnotu 0, protože v tomto případě se jedná o výchozí lokaci, a čas odeslání zásilky bude stanoven na aktuální čas v době průběhu funkce pomocí timestamp.

Tato funkce využívá proměnné:

uint id_zasilky – unikátní identifikátor konkrétní zásilky

string odkud – lokace, ze které byla zásilka odeslána

string kam – lokace, do které byla zásilka odeslána

string mistoPuvodu – prvotní lokace, kde došlo k první manipulaci se zásilkou

string posledniPoloha – poslední destinace, kde se zásilka nacházela

string cilovaDestinace – lokace, která je považována za konečný cíl zásilky

uint public casodjezdu – timestamp v UNIX formátu, v době, kdy došlo k odeslání zásilky

uint public casdoruceni – timestamp v UNIX formátu, v době, kdy došlo k doručení zásilky do cílové destinace

```
function dorazilo(string  _nazev, string _odpovednaOsoba)

    public
    returns (bool success)

{
    posledniPoloha = _nazev;
    uint _prijezd;

    if (_prijezd == 0) {
        _prijezd = now;
    }

    lokaceP.push(Lokace({
        nazev: _nazev,
        odpovednaOsoba: _odpovednaOsoba,
        prijezd: _prijezd,
        odjezd: 0
    }));

emit Dorazilo(_nazev, _odpovednaOsoba, _prijezd, lokaceP.length - 1);
    return true;
}
```

Obrázek 22 - funkce dorazilo, sloužící k zaznamenání doručení zásilky do nové lokace

Funkce sloužící k vytvoření záznamu o doručení zásilky do nové lokace. Využívá parametry string _nazev a string _odpovednaOsoba. Pomocí podmínky if, nastaví čas příjezdu formou timestamp na aktuální čas, kdy došlo k průběhu funkce. Uloží do pole lokaceP také název lokace příchozí zásilky, jméno odpovědné osoby, která přebírá zásilku, čas příjezdu, a čas odjezdu, který nastavuje na hodnotu 0, protože k odeslání zásilky v této situaci protazím nedošlo.

```
function odeslane(uint _ID, string _kam)

    public
    returns (bool success)

{
    kam = _kam;
    odkud = lokaceP[_ID].nazev;
    uint _odjezd;

    if (_odjezd == 0) {
        _odjezd = now;
    }

    lokaceP[_ID].odjezd = _odjezd;

emit Odeslane(lokaceP[_ID].nazev, lokaceP[_ID].odpovednaOsoba, _odjezd, _ID);
    return true;
```

Obrázek 23 - funkce odeslane, sloužící k zaznamenání odeslání zásilky do další lokace

Funkce sloužící k zaznamenání odeslání zásilky do další lokace. Funkce využívá parametry uint _ID a string _kam. Pomocí podmínky if, nastaví čas odjezdu formou timestamp na aktuální čas. Pomocí parametru _ID, funkce v závislosti na požadovaném ID získané na vstupu od uživatele, změní čas odjezdu a cílovou lokaci v proměnné lokaceP.

```
function doruceno(string _nazev, string _odpovednaOsoba)

    public
    returns (bool success)

{
    uint _prijezd;

    if (_prijezd == 0) {
        _prijezd = now;
    }
     casdoruceni = _prijezd;
     posledniPoloha = _nazev;

    lokaceP.push(Lokace({
        nazev: _nazev,
        odpovednaOsoba: _odpovednaOsoba,
        prijezd: _prijezd,
        odjezd: 0
    }));

emit Dorazilo(_nazev, _odpovednaOsoba, _prijezd, lokaceP.length - 1);
emit Doruceno(_nazev, _odpovednaOsoba, _prijezd);
    return true;
}
```

Obrázek 24 – funkce doruceno, sloužící k zaznamenání doručení zásilky do cílové lokace

Jedná se o funkci s parametry string _nazev a sring _odpovednaOsoba, která slouží k zaznamenání doručení zásilky do stanovené cílové lokace. Funkce pomocí podmínky if stanoví čas příjezdu. Do pole lokaceP uloží název cílové lokace, kam byla zásilka doručena, jméno odpovědné osoby, čas, kdy byla zásilka doručena. Čas odjezdu bude v tomto případě stanoven na hodnotu 0, protože se již nepředpokládá další nakládání se zásilkou.

```
function detaily()
    public
    view
    returns (uint, string, string, string, string, string, uint, uint)
{

    return (id_zasilky, odkud, kam, mistoPuvodu, cilovaDestinace, posledniPoloha, casodjezdu, casdoruceni);

}
```

Obrázek 25 - funkce detaily, sloužící k zobrazení informací o zásilce

V tomto případě jde o funkci, která poslouží pouze k návratu zvolených proměnných, které považuji za stěžejní při sledování zásilky. Funkce nedovolí uživateli vkládat žádné hodnoty.

```
function lokace()
    public
    view
    returns (Lokace[])
{

    return lokaceP;

}
```

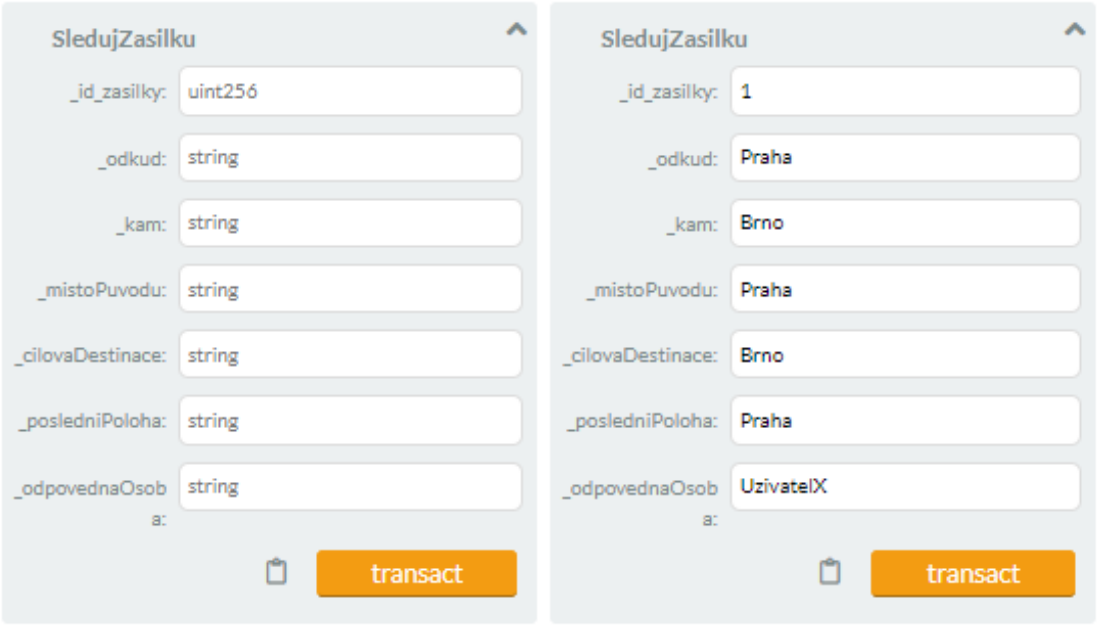
Obrázek 26 – funkce lokace, která vrací obsah pole lokaceP

4.7 Interakce s chytrým kontraktem

Po správném zavedení našeho chytrého kontraktu můžeme v prostředí Remix nalézt naše funkce, které jsme nadefinovali pomocí zdrojového kódu psaného v programovacím jazyce Solidity. Tyto funkce je možné využívat ke komunikaci s chytrým kontraktem. Pomocí funkcí můžeme data do chytrého kontraktu vkládat, ale také i námi vložená data posléze získávat.

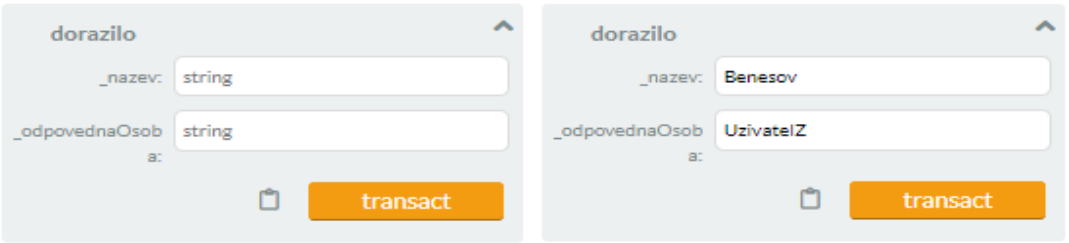
Obrázek 27 – kontrakt nasazený do naší privátní sítě

Výše máme možnost vidět funkce našeho chytrého kontraktu a také ukazatele, které slouží pouze k navrácení hodnot, nemají žádný prostor pro uživatelský vstup, tyto ukazatele mají v prostředí Remix modrá tlačítka. Pro získávání dat od uživatele slouží námi nadefinované funkce, které mají v prostředí Remix oranžová tlačítka, a pole, do kterého máme možnost zadávat hodnoty na vstupu.

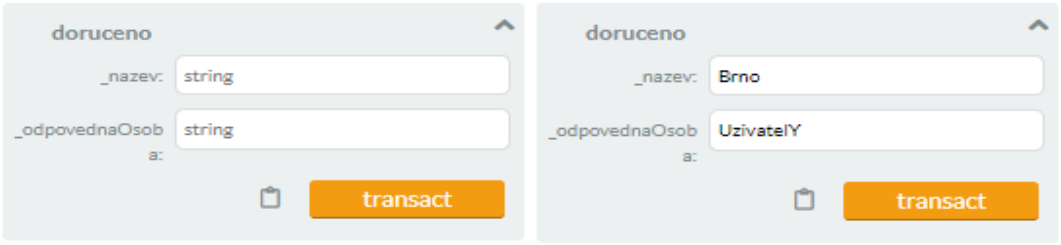


Obrázek 28 - funkce SledujZasilku

Pro inicializaci zásilky, zvolíme funkci SledujZasilku, kde po vyplnění hodnot, a provedení transakce, se uvedené informace zapíší do datového prostoru v rámci adresy daného chytrého kontraktu. Tyto skutečnosti mohou býti viděny v konzoli prostředí Remix, stejně tak jako veškeré informace týkající se všech dalších transakcí, ať už se jedná o transakce funkcí informace ukládající, nebo funkcí, které mají za úkol, tyto informace pouze z našeho chytrého kontraktu získávat.

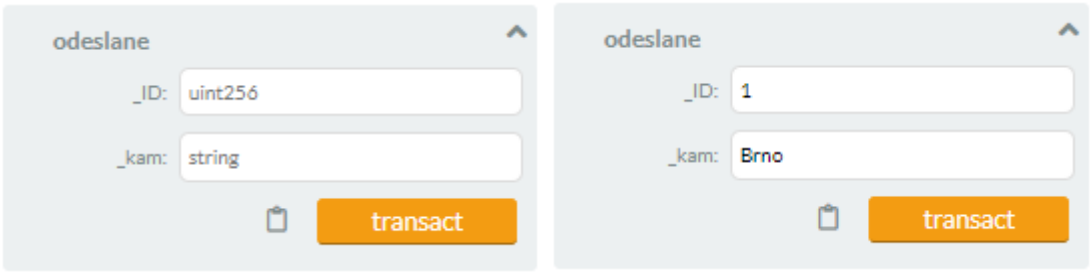


Obrázek 29 – funkce dorazilo



Obrázek 30 – funkce doruceno

Dále vznikají nové možnosti pro uživatele, v závislosti na aktuálním dění životního cyklu sledované zásilky. Máme tedy možnost, v případě že zásilka byla již doručena do cílové destinace, a již se nepočítá s budoucím pohybem zásilky, tedy zvolit funkci doruceno, která signalizuje ukončení procesu doručování konkrétní zásilky. Pokud však se jedná v životním cyklu zásilky o stav, který není považován za konečný, respektive se nejedná o cílovou destinaci zásilky, ale o destinaci přechodnou, tak zvolí uživatel funkci dorazilo, která pouze signalizuje změnu stavu poslední lokace zásilky, kde došlo k manipulaci s námi sledovanou zásilkou.



Obrázek 31 – funkce odeslane

Jako jakýsi mezikrok slouží funkce odeslane, kterou uživatel volí v případě odeslání zásilky z jedné lokace do druhé, za předpokladu užití jedné z budoucích funkcí dorazilo, nebo případně funkci doruceno, v případě že se zásilka vyskytne v cílové destinaci.

Obrázek 32 - funkce určené k získávání dat z našeho chytrého kontraktu

V případě, že uživatel užije všechny funkce, nebo alespoň první inicializační funkci chytrého kontraktu, dojde k uložení těchto informací do datového prostoru určeného přímo pro tuto adresu, na které se nachází náš chytrý kontrakt. Tyto informace se aktualizují vždy s použitím každé další funkce. Veškerá přenesená data pomocí transakce do našeho blockchainu si však máme možnost pomocí funkcí určených k získávání dat zobrazit v grafickém rozhraní prostředí Remix, jak již bylo řečeno, tyto funkce mají modré transakční tlačítko, které pokud uživatel aktivuje, dojde k provedení, přesně tak, jak můžeme vidět na obrázku výše.


```
"from": "0x68047B0C9EC40Fd8f1ff0a5617B897D86dA3BD59",
"topic": "0x5b6565ee09ca9f32f75fcc334a73777e647582d26579841c6d442774cd8f1856",
"event": "Doruceno",
"args": {
        "0": "Brno",
        "1": "UzivatelY",
        "2": "1583779963",
        "lokace": "Brno",
        "odpovednaOsoba": "UzivatelY",
        "cas": "1583779963",
        "length": 3
}
```


Obrázek 33 – výpis konzole prostředí Remix

Vždy při užití některé z funkcí, dostaneme odpověď prostřednictvím Remix konzole, kde máme možnost detailněji nahlédnout do pozadí transakce. Vidíme zde z jaké adresy transakce přišla, vidíme zde id transakce, název právě provedené funkce, které se tyto hodnoty týkají. Dostaneme také výčet všech funkčních argumentů, jak v podobě indexu, tak ve formě proměnné spárované s odpovídající hodnotou, na posledním řádku nám konzole vypisuje délku argumentů, reprezentovanou jejich počtem.

5. Závěr

Cílem praktické části práce bylo především vytvoření vlastního blockchainu na bázi platformy Ethereum, a vytvoření vlastního chytrého kontraktu ve zvolené části e-business, a posléze vytvořený chytrý kontrakt implementovat do sítě našeho blockchainu. Předtím však došlo k důkladné analýze technologií, potřebných k těmto krokům, která spadá pod část teoretickou.

Nejprve byla rozebrána podstata blockchainu, jeho původ, co je to za technologii, a na jakém principu funguje. Dále byla provedena analýza funkční struktury blockchainu, kde byl použit jako příklad blockchain první vzniklé kryptoměny, a to Bitcoinu. Posléze byla probrána problematika potenciálního využití blockchainu v několika různých odvětví, kde v některých zmiňovaných případech již došlo v aktuálním světe k úspěšné implementaci. Poté byl uskutečněn přesun na téma chytrých kontraktů, kde byla tato technologie představena.

Z počátku praktické části analyzujeme problematiku současných způsobů sledování zásilek, užívaných technologií v rámci těchto procesů a jejich úskalí. Než jsme započali se samotnou implementací chytrého kontraktu, bylo nutné vytvořit vlastní síť platformy Ethereum, kterou bylo nutné využít jako testovací prostředí. Autor zvolil způsob vlastní privátní sítě, z důvodů uvedených výše v textu práce. Po vytvoření našeho testovacího uživatele a spuštění procesu těžby Etherea mohlo již dojít k samotnému nasazení vlastního chytrého kontraktu, který umožňuje uživateli sledovat manipulaci s konkrétní zásilkou.

Chytrý kontrakt byl napsán v programovacím jazyce Solidity, autorizace transakce proběhla v rámci naší privátní sítě pouze jedním námi vytvořeným uživatelem prostřednictvím těžby. Samotná implementace chytrého kontraktu byla vícekrát testována. Došlo k zjištění pravděpodobné chyby v některém z použitého software, jelikož občasně docházelo v rámci testů k neschválení transakce, tento problém byl vždy úspěšně vyřešen přeinstalací webového rozšíření MetaMask, vyčištěním dočasných souborů a souborů cookies.

Autor práce je přesvědčen, že implementace chytrého kontraktu by byla použitelná v praxi, ale pouze za předpokladu optimalizace zdrojové kódu pro potřeby konkrétního logistického subjektu. Velmi vhodné by bylo taktéž spojit tuto technologii s dalšími podpůrnými systémy, popřípadě i jinými hardwarovými články. Pomocí kombinací s prvky IoT, by mohla být

zajištěna například dostupnost informací v reálném čase. K tomuto řešení, je však za potřebí využití dalších systémů.

6. Seznam použité literatury

(1) *History of Blockchain* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://www.binance.vision/blockchain/history-of-blockchain

(2) *Blockchain functional introduction* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.pwc.be/en/news-publications/insights/2017/blockchain-functional-introduction.html

(3) *Differences between public and private blockchains* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://dragonchain.com/blog/differences-between-public-private-blockchains/

(4) *What is blockchain hashing* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://hedgetrade.com/what-is-blockchain-hashing/

(5) *Bitcoin nonce explained* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://www.mycryptopedia.com/bitcoin-nonce-explained/

(6) *Bitcoin* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://cs.wikipedia.org/wiki/Bitcoin

(7) *Potvrzení kryptoměnových transakcí* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://finex.cz/confirmations-potvrzeni-kryptomenovych-transakci/

(8) *The Advantages and Disadvantages of the Blockchain Technology* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://www.researchgate.net/publication/330028734_The_Advantages_and_Disadvantages_of_the_Blockchain_Technology

(9) *Blockchain usage* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://101blockchains.com/blockchain-usage/#prettyPhoto

(10) *Co je to kryptoměna Vechain* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://kryptoportal.cz/co-je-kryptomena-vechain-ven/

(11) *Merkle tree* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://blockonomi.com/merkle-tree/

(12) *How does bitcoin mining work* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.investopedia.com/tech/how-does-bitcoin-mining-work/

(13) *Bitcoin forks guide* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://blockgeeks.com/guides/bitcoin-forks-guide/

(14) *Zákaznický integrovaný obvod* [online]. 2017 [cit. 2020-05-01]. Dostupné z: https://cs.wikipedia.org/wiki/Z%C3%A1kaznick%C3%BD_integrovan%C3%BD_obvod

(15) *Bitcoin immutability* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.binance.vision/glossary/immutability

(16) *Bitcoin transparency* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://bitcoin.org/en/protect-your-privacy

(17) *Blockchain security* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.binance.vision/blockchain/what-makes-a-blockchain-secure

(18) *A study on the issue of blockchain energy consmuption* [online]. 2020 [cit. 2020-05-01]. Dostupné z:

https://www.researchgate.net/publication/337653483_A_Study_on_the_Issue_of_Blockchain's_Energy_Consumption

(19) *Silk Road* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://cs.wikipedia.org/wiki/Silk_Road_(online_market)

(20) *Bitcoin volatility* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.investopedia.com/articles/investing/052014/why-bitcoins-value-so-volatile.asp

(21) *VeChain* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.vechain.org/

(22) *List of VeChain partnerships* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://usethebitcoin.com/list-of-vechain-partnerships/

(23) *What is Monero?* [online]. 2017 [cit. 2020-05-01]. Dostupné z: https://blockgeeks.com/guides/monero/

(24) *Kruhový podpis* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://getmonero.cz/zdroje/moneropedie/ringsignatures.html

(25) *What is Siacoin?* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://blockgeeks.com/guides/what-is-siacoin-complete-expert-guide-blockgeeks/

(26) *Blockchain in healthcare* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://blockgeeks.com/guides/blockchain-in-healthcare/

(27) *Blockchain technology in healthcare* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://www.researchgate.net/publication/332685807_Blockchain_Technology_in_Healthcare_A_Comprehensive_Review_and_Directions_for_Future_Research

(28) *Blockchain in education* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://medium.com/universablockchain/blockchain-in-education-49ad413b9e12

(29) *Zeux* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://www.zeux.tech/

(30) *Transaction charts* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.blockchain.com/charts

(31) *CoinLore* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://www.coinlore.com/

(32) *Coinmarketcap* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://coinmarketcap.com/

(33) *Bitcoin regulation* [online]. 2017 [cit. 2020-05-01]. Dostupné z: https://www.digitaltrends.com/computing/dont-worry-about-bitcoin-regulation-it-cant-be-stopped/

(34) *Blockchain in supply chain management* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://medium.com/@infopulseglobal_9037/blockchain-in-supply-chain-management-key-use-cases-and-benefits-6c6b7fd43094

(35) *Blockchain and supply chain* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://blockgeeks.com/guides/blockchain-and-supply-chain/

(36) *Blockchain future* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://101blockchains.com/blockchain-the-future/

(37) *World Wide Web* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://en.wikipedia.org/wiki/World_Wide_Web

(38) *Smart contracts* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://www.investopedia.com/terms/s/smart-contracts.asp

(39) *Ethereum EVM* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://hackernoon.com/getting-deep-into-evm-how-ethereum-works-backstage-ac7efa1f0015

(40) *Functional smart contracts compliance* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://hackernoon.com/from-ethereum-to-infinity-functional-smart-contracts-compliance-with-community-requirements-c5bc05eccc48

(41) *Future of commerce* [online]. 2017 [cit. 2020-05-01]. Dostupné z: https://www.huffpost.com/entry/how-blockchain-is-redefining-the-future-of-commerce_b_59a44849e4b0cb7715bfd78d?ncid=engmodushpmg00000004

(42) *Shipment tracking on the blockchain an easier life for merchants* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://medium.com/@gambproject/shipment-tracking-on-the-blockchain-an-easier-life-for-merchants-a7156670ad73

(43) *Product tracking* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://www.infosys.com/Oracle/insights/Documents/product-tracking-tracing.pdf

(44) *Geth Github* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://github.com/ethereum/go-ethereum/wiki/geth

(45) *MetaMask* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://metamask.io/

(46) *Remix IDE* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://remix.ethereum.org/

(47) *Solidity documentation* [online]. 2018 [cit. 2020-05-01]. Dostupné z: https://solidity.readthedocs.io/en/v0.4.24/

(48) *Blockchain size* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://blockchair.com/ethereum/charts/blockchain-size

(49) *Blockchain transaction fee* [online]. 2020 [cit. 2020-05-01]. Dostupné z: https://wirexapp.com/help/article/what-is-the-blockchain-fee-0078

(50) *Genesis block explained* [online]. [cit. 2020-05-01]. Dostupné z: https://www.asynclabs.co/blog/params-in-ethereum-genesis-block-explained/

(51) *Geth parameters* [online]. 2019 [cit. 2020-05-01]. Dostupné z: https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options

(52) Obrázek a – Obrázek sestava určená pro těžbu [online]. [cit. 2020-05-01]. Dostupné z: https://itbukva.com/gadgets/review/15649-best-mining-rig.html

(53) Obrázek b - [online]. [cit. 2020-05-01]. Dostupné z: https://mrminer.org/gb/miners/11913-innosilicon-a10-ethmaster-500mh.html

(54) Obrázek c - *Obrázek tradiční kontrakt v porovnání s chytrým kontraktem* [online]. [cit. 2020-05-01]. Dostupné z: https://dzone.com/articles/what-is-smart-contracts-blockchain-and-its-use-cas-1

7. Seznam použitých obrázků

8. Příloha A – zdrojový kód chytrého kontraktu

```
pragma solidity ^0.4.26;
pragma experimental ABIEncoderV2;

contract SledovaniZasilky {

    struct Lokace {
        string nazev;
        string odpovednaOsoba;
        uint prijezd;
        uint odjezd;
    }

    Lokace[] lokaceP;

    event Odeslane(string lokace, string odpovednaOsoba, uint cas, uint
ID);

    event Dorazilo(string lokace, string odpovednaOsoba, uint cas, uint
ID);

    event Doruceno(string lokace, string odpovednaOsoba, uint cas);

    uint public id_zasilky;
    string public odkud;
    string public kam;
    string public mistoPuvodu;
    string public cilovaDestinace;
    string public posledniPoloha;
    uint public casodjezdu;
    uint public casdoruceni;



    function SledujZasilku(uint _id_zasilky, string _odkud, string _kam,
string _mistoPuvodu, string _cilovaDestinace, string _posledniPoloha,
string _odpovednaOsoba)

        public {
```

```
    id_zasilky = _id_zasilky;
    odkud = _odkud;
    kam = _kam;
    mistoPuvodu = _mistoPuvodu;
    posledniPoloha = _posledniPoloha;
    cilovaDestinace = _cilovaDestinace;
    casodjezdu = now;
    casdoruceni = 0;

    lokaceP.push(Lokace({
        nazev: _mistoPuvodu,
        odpovednaOsoba: _odpovednaOsoba,
        prijezd: 0,
        odjezd: now
    }));

 emit Odeslane(_mistoPuvodu, _odpovednaOsoba, now, lokaceP.length - 1);

}



function dorazilo(string  _nazev, string _odpovednaOsoba)

    public
    returns (bool success)

{
    posledniPoloha = _nazev;
    uint _prijezd;

    if (_prijezd == 0) {
        _prijezd = now;
    }

    lokaceP.push(Lokace({
        nazev: _nazev,
        odpovednaOsoba: _odpovednaOsoba,
        prijezd: _prijezd,
        odjezd: 0
```

```
        }));

    emit Dorazilo(_nazev, _odpovednaOsoba, _prijezd, lokaceP.length - 1);
        return true;
    }


    function odeslane(uint _ID, string _kam)

        public
        returns (bool success)

    {
        kam = _kam;
        odkud = lokaceP[_ID].nazev;
        uint _odjezd;

        if (_odjezd == 0) {
            _odjezd = now;
        }

        lokaceP[_ID].odjezd = _odjezd;

    emit Odeslane(lokaceP[_ID].nazev, lokaceP[_ID].odpovednaOsoba, _odjezd,
_ID);
        return true;
    }

    function doruceno(string _nazev, string _odpovednaOsoba)

        public
        returns (bool success)

    {
        uint _prijezd;

        if (_prijezd == 0) {
            _prijezd = now;
        }
         casdoruceni = _prijezd;
         posledniPoloha = _nazev;
```

```
        lokaceP.push(Lokace({
            nazev: _nazev,
            odpovednaOsoba: _odpovednaOsoba,
            prijezd: _prijezd,
            odjezd: 0
        }));

    emit Dorazilo(_nazev, _odpovednaOsoba, _prijezd, lokaceP.length - 1);
    emit Doruceno(_nazev, _odpovednaOsoba, _prijezd);
        return true;
    }


    function detaily()
        public
        view
        returns (uint, string, string, string, string, string, uint, uint)
    {

        return (id_zasilky, odkud, kam, mistoPuvodu, cilovaDestinace,
posledniPoloha, casodjezdu, casdoruceni);

    }


    function lokace()
        public
        view
        returns (Lokace[])
    {

        return lokaceP;

    }


        function() public payable {

        revert();
```

```
        }
    }
```