Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Integrace vlastního krypto tokenu do webové aplikace

Diplomová práce

Autor: Bc. Filip Roškot
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.

Hradec Králové srpen 2023

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 21.8.2023 Filip Roškot

Poděkování:

Děkuji vedoucí diplomové práce Mgr. Daniele Ponce, Ph.D., za metodické a svědomité vedení práce a nekonečnou dávku tolerance a vstřícnosti. Dále také děkuji své manželce, která se dost zasloužila o to, abych vysokou školu už konečně dostudoval.

Anotace

Diplomová práce se věnuje oblasti krypto tokenů. Zabývá se navržením a tvorbou tokenu splňujícím standard ERC-20. V teoretické části práce je pozornost věnována zejména blockchainu Ethereum a jeho standardům pro chytré smlouvy, dále také decentralizovaným aplikacím. Okrajově se práce také zaobírá ostatními blockchainovými sítěmi. V praktické části je vytvořen krypto token v programovacím jazyce Solidity, který je nasazen na testovací síti Etherea. Token je následně integrován do webové aplikace vyvinuté pomocí frameworku React.js a knihovny Ethers.js. Aplikace spolupracuje s webovým rozšířením kryptoměnové peněženky MetaMask. V rámci aplikace je umožněno token přesouvat mezi ostatními uživateli a také ho získávat při splnění určitých podmínek.

Klíčová slova: Blockchain, Kryptoměny, Krypto tokeny, Ethereum, Chytré smlouvy, Webová aplikace

Annotation

Title: Integration of a custom crypto token into a web application

The diploma thesis focuses on the area of crypto tokens. It deals with the design and creation of a token complying with the ERC-20 standard. In the theoretical part of the thesis, attention is primarily given to the Ethereum blockchain and its standards for smart contracts, as well as decentralized applications. The thesis also briefly addresses other blockchain networks. In the practical part, a crypto token is created using the Solidity programming language, which is deployed on the Ethereum test network. The token is then integrated into a web application developed using the React.js framework and the Ethers.js library. The application interacts with the web extension of the MetaMask cryptocurrency wallet. Within the application, users are able to transfer the token among themselves and also acquire it by fulfilling specific conditions.

Keywords: Blockchain, Cryptocurrencies, Crypto tokens, Ethereum, Smart Contracts, Web application

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

V dnešní době digitální revoluce a rozmachu technologie se objevuje mnoho inovativních konceptů a nástrojů, které mění způsob, jakým vnímáme tradiční finanční systém a hospodářské procesy. Jedním z nejzajímavějších a nejkontroverznějších témat, která tato revoluce přinesla, jsou kryptoměny a krypto tokeny. Tato digitální aktiva se stala středem pozornosti finančních expertů, technologických nadšenců a veřejnosti obecně. O signifikantnosti této tematiky svědčí i nemalý objem mediálního prostoru, který je kryptoměnám věnován.

Kryptoměny, jako například Bitcoin, a krypto tokeny, jako jsou např. tokeny postavené na chytrých smlouvách na platformě Ethereum, se odlišují od tradičních měn a finančních nástrojů tím, že jsou založeny na technologii blockchainu. Blockchain je distribuovaný a nezměnitelný účetní systém, který umožňuje zaznamenávání transakcí a aktivit v bezpečném a transparentním prostředí. Tato technologie nabízí řadu potenciálních výhod, včetně nižších transakčních nákladů, rychlejších mezinárodních plateb a snížení potřeby prostředníků.

Integrace krypto tokenů do webových aplikací, tak jak je známe, by mohla otevřít nesčetné možnosti ukládání hodnot a vytváření nových trendů. Lze si i představit, že už by se nikdo nemusel obávat toho, že jednou provozovatel odpojí servery a jeho digitální vlastnictví padne vniveč. Mohla by nastat nová doba, kdy digitální vlastnictví zůstane nadále „žít“ na blockchainu a neexistuje žádná centrální autorita, která by to mohla změnit.

2 Cíl práce

Tato diplomová práce si klade za cíl vytvořit krypto token s předem stanovenými vlastnostmi a na konkrétně vybraném blockchainu splňujícím jeho předepsané standardy. Následně bude tento token nasazen a integrován do webové aplikace, která umožní jeho základní obsluhu.

Hlavním záměrem této práce je představení možnosti vytvoření a následné integrace krypto tokenů, a tím přispět k postupnému přechodu na Web 3.0, který spočívá ve využití decentralizovaných technologií, jako je blockchain. Práce může být přínosem těm, kteří již mají určité znalosti o kryptoměnách, krypto tokenech nebo jsou přímo vývojáři webových aplikací.

V první části této práce budou shrnuty teoretické poznatky ohledně technologie blockchain a samotných kryptoměn, které tuto technologii využívají. Dále bude detailně popsán blockchain Ethereum, který položil základy pro první chytré smlouvy. Následně bude zkoumán koncept Webu 3.0 z perspektivy decentralizace a budou představeny i decentralizované aplikace. Vzhledem k povaze těchto komunitních sítí budou informace získány především z open-source a dalších veřejně dostupných zdrojů.

Ve druhé části práce budou definovány požadavky na výběr blockchain sítě a jejího standardu pro implementaci krypto tokenů. Získané teoretické znalosti spolu s těmito požadavky budou aplikovány při návrhu samotného tokenu. S využitím znalostí o kryptoměnách, blockchain technologiích a standardech pro tokeny bude tento nový token navržen a vytvořen s důrazem na specifikace stanovené v počátečním plánu.

Tato práce nezkoumá právní aspekty týkající se kryptoměn a krypto tokenů. Nemá také vzbuzovat zájem o investování nebo spekulace v tomto kontextu. Nezabývá se poskytováním finančního či právního poradenství a neřeší otázky ekologických dopadů.

3 Blockchain

Tato kapitola pojednává o technologii blockchainu (z angličiny „blokový řetězec“) a jejím využití. V českém jazyce máme od roku 2017 ekvivalent anglického termínu „blockchain“ ve slově „bločenka“, avšak tento termín se zatím příliš nerozšířil. [3]

Pro mnoho lidí může být blockchain stále novým pojmem, a proto je vhodné začít s jednoduchým vysvětlením. Představme si účetní knihu, která obsahuje různé záznamy. Každý záznam musí být správný a neporušený. Blockchain je také jako účetní kniha, s tím rozdílem, že je to digitální kniha umístěná na internetu. To znamená, že se stává decentralizovanou knihou, do které přispívá mnoho lidí. Můžeme si blockchain také představit jako řetězec souborů, kde je uchováváno vše, co má být trvale zaznamenáno. Základní blockchain spojuje tyto soubory do jednoduchého řetězce. Pokročilejší blockchainy spojují soubory a tvoří síť podobnou struktuře internetu. [1] Jinými slovy – blockchain je speciální druh databáze, známý též jako decentralizovaná digitální účetní kniha, kterou spravuje celá řada počítačů rozprostřených po celém světě. Data jsou uspořádána do bloků, které jsou chronologicky uspořádány a zabezpečeny kryptografickými metodami. [4]

3.1 Historie

Blockchain představuje formu distribuované účetní knihy (Distributed Ledger Technology) [5]. Tato technologie získala popularitu s nástupem kryptoměny Bitcoin (viz Kapitola 4.3.4). Díky silné mediální expozici a faktu, že Bitcoin se stal synonymem pro kryptoměny, by se mohlo zdát, že Bitcoin byl první kryptoměnou, avšak myšlenka kryptoměn se objevila již v 80. letech 20. století. Představil ji doktor David Chaum. [6]

Chaum byl původně motivován touhou vytvořit zabezpečený anonymní digitální hlasovací systém. Vytvořil kryptografickou techniku zvanou "slepé podpisy" (blind signatures), která umožňovala pseudonymitu při výměně dat. Tato metoda využívala veřejnou kryptografii, což znamená, že každý uživatel měl svůj veřejný (public key) a tajný soukromý klíč (secret private key). Slepé podpisy umožňovaly

podepsat data digitálně se soukromým klíčem, a takto podepsaná data byla použitelná s jistotou, že byla podepsána konkrétní osobou, ačkoliv byl znám pouze veřejný klíč. To umožňovalo ověřování původce zprávy bez nutnosti znát jeho skutečnou identitu. Na základě této techniky Chaum představil anonymní elektronický platební systém a v roce 1989 vzaložil společnost DigiCash, která vyvinula kryptoměnu eCash (viz Kapitola 3.3.1.1) s důrazem na platební anonymitu a nevysledovatelnost pomocí slepých podpisů. [6]

Samotná technologie blockchainu má své počátky na konci 70. let, kdy počítačový vědec jménem Ralph Merkle patentoval Hashový strom, známý též jako Merkleův strom. Tyto stromy představují datovou strukturu v oblasti počítačové vědy, která propojuje bloky dat pomocí kryptografie. Na konci 90. let informatik Stuart Haber a fyzik W. Scott Stornetta využili Merkleův strom pro implementaci systému, ve kterém nebylo možné zpětně datovat nebo manipulovat s časovými razítky dokumentů. Tím vytvořili první příklad technologie blockchainu v historii. [2]

3.1.1 Merkleův strom

Hashový strom je datová struktura, kterou je možné efektivně ověřit integritu dané sady dat. To je obzvláště důležité v kontextu P2P (Peer-to-peer)[1] sítí, kde účastníci potřebují sdílet informace a neustále je ověřovat. [72]

V případě Merkleova kořenového hashe lze ověřit prakticky cokoli. Data jsou rozdělena do bloků (částí), kde každá část je následně zahashována. Poté dojde k vytvoření párů z těchto hashů, na které je následně znovu aplikována hashovací funkce. Tyto kroky se opakují až do té doby, než zbývá pouze jediný hash – hlavní hash neboli kořenový hash (nebo také Merkleový hash), který reprezentuje hash všech dat. [72]

Hashové stromy mají několik různých možností použití. Pro kryptoměny jsou naprosto nezbytné. Jsou nedílnou součástí každého bloku a najdeme je vždy v hlavičce. Části hashového stromu najdeme z hashů (známé také jako TXID, transaction id) z každé transakce v bloku. [72]

[1] „Peer-to-peer (P2P) systém je distribuovaný systém sestávající ze vzájemně propojených uzlů schopných se vzájemně a samostatně organizovat do definovaných síťových topologií za účelem sdílení zdrojů.“ [85]

3.2 Technologie

Blockchain technologie je běžně spojována s kryptoměnami, jako je např. Bitcoin. Je to decentralizovaná databáze záznamů transakcí, které jsou distribuovány a které jsou ověřeny sítí počítačů po celém světě. Místo jediného ústředního orgánu, jako je např. banka, je na záznamy dohlíženo skrze komunitu, a ne pouze jednotlivcem, tudíž nemůže dojít ke zpětné změně nebo úplnému smazání historie transakcí. Ve srovnání s běžnou centralizovanou databází tak jakékoli informace nemohou být zmanipulovány díky vlastnostem této technologie, kde je nutné potvrzení ze strany dalších účastníků komunity, mezi které je databáze distribuována. Jinými slovy, když je běžná centralizovaná databáze umístěna na konkrétním serveru, která se stává autoritou, může dojít prakticky k ovlivnění jakýkoliv záznamů. Oproti tomu u blockchain technologie, která umožňuje komukoli v síti přistoupit k záznamům všech ostatních, a tím zamezit jedné centrální entitě získat kontrolu nad sítí, je možnost ovlivnění záznamů téměř nemožná. Vždy, když někdo provádí transakci, je její pravost ověřována sítí. Pokud je transakce ověřena, je propojena s předchozí a tvoří tak řetězec transakcí. Tento řetězec je poté nazýván blockchain. [87]

3.3 Kryptoměny

Tato kapitola představuje historii a vznik prvních kryptoměn. Dále představuje aplikované důležité mechanismy pro správné fungování s blockchain technologií a případné bezpečnostní hrozby. Uvedeny budou také dvě ikonické kryptoměny, které mají výrazný vliv na svět kryptoměn a blockchainových technologií. Kapitola také obsahuje informace o způsobu uchování blockchainových digitálních aktiv a v neposlední řadě vysvětluje správnou terminologii potřebnou pro navazující části této práce.

3.3.1 Historie

Jak bylo uvedeno v kapitole historie blockchain technologie, první známou kryptoměnou byl eCash vyvinutý v roce 1989 společností DigiCash, založenou doktorem Davidem Chaumem.

3.3.1.1 eCash

Kryptoměna eCash, která byla zaměřena na anonymitu a nevysledovatelnost plateb pomocí slepých podpisů, byla navržena s cílem umožnit mikroplatby skrze kryptograficky zabezpečené soubory, které nesly určitou finanční hodnotu a byly generovány výměnou za běžnou měnu (např. americké dolary). Uživatelé nebo zákazníci by nejprve převedli prostředky ze svého bankovního účtu na svůj eCash účet. Poté byl vytvořen zašifrovaný soubor nesoucí hodnotu, který byl uložen na počítači zákazníka. K tomuto účelu dostal zákazník speciální program pro správu těchto zašifrovaných souborů a provádění plateb pomocí nich. Pro vyřešení problému dvojího utrácení zavedl Chaum koncepci, kde byly banky zodpovědné za ověření, zda daný eCash již nebyl použit v jiných transakcích. Například, pokud Ábel poslal hodnotu eCash ve formě zašifrovaného souboru Kainovi, Kain by poté tento eCash mohl poslat vydávající bance. Tato banka by poté ověřila, že eCash již nebyl použit v jiné transakci. Tato koncepce však stále zahrnovala zprostředkovatele – konkrétně banky – jako autoritu. [6]

eCash byl testován jako mikroplatební systém v USA v letech 1995 až 1998. Zákazníci platili transakční poplatky, zatímco obchodníci měli eCash zdarma. Tento model připomínající fungování kreditních karet a platebních terminálů je podobný tomu, co vidíme v současnosti. Avšak eCash nezískal širokou přízeň u amerických uživatelů, kteří nakonec dávali přednost kreditním kartám. Naopak v Evropě byl eCash více akceptován finančními institucemi, neboť platební karty nebyly tak rozšířené a byly preferované hotovostní transakce. Několik evropských institucí proto přijalo eCash v roce 1998. Přestože měl eCash určitý úspěch, firma DigiCash vyhlásila bankrot ještě ve stejném roce [6].

3.3.1.1.1 E-gold

V roce 1996 představili Douglas Jackson a Barry Downey kryptoměnu s názvem e-gold, provozovanou společností Gold & Silver Reserve Inc., která se později přejmenovala na e-gold Ltd. Tato kryptoměna byla podpořena zlatem, což naznačoval i název. V roce 2004 zaznamenala e-gold kritický nárůst uživatelů a obchodníků, když dosáhla více než 1,3 milionů účtů. Uživatelé si museli vytvořit

účet na webové stránce, což jim umožňovalo provádět rychlé převody v měně e-gold na jiné účty. Tato platforma také nabízela rozhraní pro programování aplikací (API), což umožňovalo integrovat platby do různých služeb a e-commerce platforem. [6]

Přestože se e-gold stala prvním úspěšným online platebním systémem a přinesla mnoho inovací v oblasti zabezpečené komunikace pro platby pomocí šifrování SSL a flexibilní integrace platebních služeb do externích systémů skrze API, stala se také cílem raných podvodů s phishingem a škodlivým softwarem. Inspirací pro tyto útoky byl první známý phishing útok proti finanční instituci v červnu 2001 [39]. Další útoky směřovaly proti e-gold v roce 2003. V roce 2007 byl provoz e-gold pozastaven kvůli právním problémům, zejména ohledně praní špinavých peněz. Přestože se e-gold potýkal s těmito problémy, zůstává prvním úspěšným online platebním systémem, který položil základy pro mnoho technologií, jež jsou stále využívány v e-commerce, včetně zabezpečené komunikace pomocí šifrování SSL a integrace platebních služeb do externích systémů skrze API. [6]

3.3.1.2 Hashcash

V roce 1997 vytvořil Adam Back jednu z prvních implementací systému Proof of Work (PoW, viz Kapitola 4.3.2), který nazval Hashcash. Koncept Proof of Work původně vytvořili Cynthia Dwork a Moni Naor, později byl formálně rozpracován Markusem Jakobssonem a Ariem Juerlsem. Tento systém slouží jako ekonomické opatření ke snížení zneužívání služeb, jako je například spamování nebo útoky typu Denial of Service (DoS). Útoky DoS se zaměřují na zahlcení služby velkým množstvím žádostí, což způsobí, že služba nemůže reagovat a nakonec selže. [6]

V systémech Proof of Work musí žadatel o službu nejprve provést určitý úkol, například vyřešit matematický problém, aby byla jeho žádost akceptována. Tento úkol by měl být dostatečně obtížný, aby se předešlo nadměrnému zasílání žádostí, ale zároveň by měl být snadno ověřitelný poskytovatelem služby. Obvykle uživatelé sami neřeší tyto úkoly, ale koncová zařízení provádějí tyto výpočty automaticky. Hashcash předvedl, že PoW může být prakticky využitelný k omezení spamu a rizika DoS. [6]

Dnes hraje Proof of Work klíčovou roli v mechanismech konsenzu využívaných v blockchainových sítích, jako je například Bitcoin. [6]

3.3.1.3 B-money

V roce 1998 přišel informatik Wei Dai se dvěma návrhy protokolů pro digitální měnu nazvanou B-money. Jeho motivací byla myšlenka krypto-anarchie, kterou představil Timothy C. May. Krypto-anarchie se týká trvalé absence vlády a zdůrazňuje potřebu anonymity a odolnosti vůči cenzuře, zejména v oblasti plateb. V tomto kontextu je systém považován za odolný vůči cenzuře, pokud není možné třetí stranou modifikovat nebo blokovat data. [6]

B-money představovala systém, ve kterém by platby probíhaly pomocí kryptograficky zabezpečených kryptoměnových mincí. Tyto mince by sloužily jako směnný prostředek, který by lidem umožnil efektivně spolupracovat. Tímto způsobem by bylo možné dosáhnout cílů krypto-anarchie v oblasti finančních transakcí. [6]

V téže době, v roce 1998, měl také Nick Szabo počáteční myšlenky pro vývoj konceptu BitGoldu, který se také zaměřoval na vytvoření digitálního zlatého standardu a možnosti uskutečňovat bezpečné a anonymní transakce pomocí kryptografie. [6]

3.3.1.4 BitGold

BitGold, i když nebyl nikdy skutečně realizován, je považován za přímého předchůdce architektury Bitcoinu. Návrh BitGoldu od Nicka Szaba zahrnoval použití Proof of Work (PoW), což znamenalo, že uživatelé by využívali výpočetní sílu k řešení kryptografických rovnic, které by systém přiděloval. Szabo se rovněž snažil o eliminaci zprostředkovatelů a dvojího utrácení díky implementaci algoritmických a strukturálních vylepšení. I přesto, že byly tyto problémy stále přítomné, jelikož předchozí řešení stále závisela na důvěryhodných autoritách, Szabo v základu konceptu BitGoldu předpokládal plně decentralizovanou kryptoměnu nahrazením zprostředkovatelských procesů automatizovanými. Avšak stejně jako u B-money, i BitGold selhal v dosažení širokého rozšíření a adopce. [6]

3.3.1.5 Bitcoin

V roce 2008 byla zveřejněna bílá kniha (Whitepaper), jejímž autorem nebo autory byla osoba či skupina používající pseudonym Satoshi Nakamoto. Tato kniha popisovala kryptoměnu Bitcoin a technologii blockchain. Zmíněné koncepty se staly základem pro vytvoření prvního funkčního kryptoměnového systému.

O rok později, v roce 2009, byl spuštěn Bitcoin jako první kryptoměna na světě. Tento revoluční koncept přinesl myšlenku decentralizovaného digitálního platidla, které umožňovalo přímé transakce mezi účastníky bez nutnosti zprostředkování finančních institucí. [6]

3.3.1.6 Období po příchodu Bitcoinu

V roce 2011 vznikla kryptoměna Litecoin. Tato měna byla založena na podobné technologii jako Bitcoin, avšak nabízela rychlejší dobu zpracování transakcí a používala odlišný algoritmus pro těžbu. [65]

V roce 2013 byly spuštěny první kryptoměnové burzy, včetně známé burzy Mt. Gox. Tento rok také přinesl nárůst různých alternativních kryptoměn, nazývaných "altcoiny[2]", vedle Bitcoinu. [6]

3.3.1.7 Ethereum

V roce 2013 začal Vitalik Buterin pracovat na projektu Ethereum, který měl sloužit jako platforma pro decentralizované aplikace a chytré smlouvy. Ethereum bylo poprvé představeno na konferenci v roce 2014. [67, 68]

V roce 2015 byla spuštěna open-source platforma Ethereum, což umožnilo vývojářům vytvářet decentralizované aplikace a chytré smlouvy. Tato událost byla revoluční, protože Ethereum šlo dále než Bitcoin a umožnilo programovatelné funkce na blockchainu. [67, 68]

V roce 2016 se uskutečnil významný moment v historii Etherea v podobě hromadného prodeje tokenů (ICO[3]) projektu Decentralizované autonomní organizace (viz Kapitola 3.6.3). DAO byla organizací řízenou chytrými smlouvami

[2] Altcoin je jakákoli krypto mince nebo kryptoměna, která není přímo Bitcoin [66].
[3] Initial coin offering (ICO)

a tokeny, avšak také se stala terčem útoku, což mělo za následek ztrátu velké části finančních prostředků. [67, 68]

V roce 2017 Ethereum prožilo, podobně jako Bitcoin, období rychlého růstu a zvýšeného zájmu. Ethereum umožnilo vytváření vlastních tokenů a hromadné prodeje tokenů (ICO) se staly populárním způsobem financování nových projektů. V roce 2020 se Ethereum stalo základem pro mnoho decentralizovaných aplikací (dApps viz Kapitola 3.6) a chytrých smluv. Projekt Ethereum 2.0 je současným vývojovým směrem, který má za cíl řešit škálovatelnost a efektivitu sítě. [67, 68]

Od roku 2020 pokračuje vývoj různých kryptoměn a projektů z oblasti blockchainu. Technologie stále pokračují ve zdokonalování a nacházejí uplatnění v různých odvětvích, včetně financí, logistiky, zdravotnictví a dalších. [67, 68]

3.3.2 Mechanismy konsenzu

„Stejně jako u většiny distribuovaných výpočetních systémů se účastníci kryptoměnové sítě musí pravidelně shodovat na aktuálním stavu blockchainu.“ [36]

Technologie blockchain umožňuje vytvářet přesný účetní záznam tím, že nespoléhá na centrální autoritu, ale na algoritmus, který zahrnuje mnoho nezávislých lidí nebo počítačů, nazývaných síťové uzly. Tento algoritmus se jmenuje (decentralizovaný) konsenzuální algoritmus. [6, 34]

Konsenzus v distribuovaných systémech zajišťuje, že stav, hodnota nebo informace jsou správné a shodují se u většiny uzlů. Konsenzuální mechanismus zaručuje, že tento krok je prováděn spravedlivě a nezávisle na jakékoli zúčastněné straně, nebo v případě soukromých blockchainových sítí, aby se dosáhlo dalších cílů požadovaných sítí např. decentralizované kontroly. Jinými slovy – v decentralizovaných systémech musí distribuované uzly dospět ke konsenzu, protože neexistuje žádná centrální autorita, která by mohla převzít zodpovědnost. [6, 34]

Mechanismy konsenzu ověřují datové vstupy a výstupy, což se promítá do automatického auditu digitálních transakcí, které jsou dnes běžně bez lidského dohledu nebo zásluhy. Vytvářejí prostředí, kde nemusíte věřit, že druhá strana v transakci je čestná, protože zajišťují, že informace jsou neměnné a bezpečné.

Cílem mechanismu konsenzu ve světě kryptoměn je zabránit špatným aktérům v úmyslném podvádění, čemuž zabraňuje následující řešení problému, tzv. koncept byzantské odolnosti proti chybám. [6, 35, 36]

Problém byzantských generálů vznikl v roce 1982 jako logické dilema, jež představuje způsob vysvětlení problému s důvěrou, nesprávnou komunikací a nesouhlasících podnětů mezi uživateli decentralizovaných systémů, což platí i pro mnoho blockchainů. Toto riziko nesprávné komunikace nebo úmyslné zlomyslné akce lze nalézt v blockchainech, kde distribuované uzly potřebují souhlasit s validací informací. Decentralizovaná povaha systému znamená, že stimuly, jako např. dvojité utrácení, mohou vést k podvodným transakcím nebo zpětnému otočení legitimních transakcí. To může vést k soutěžení nebo nepřesným účetním zámkům. [34]

Některé konsenzuální mechanismy se více zaměřují na zabezpečení a decentralizaci, což je dobré pro vedení záznamů, zatímco jiné podporují větší rychlost a efektivitu, např. podporují větší počet platebních transakcí za sekundu. [34] Mezi takové mechanismy patří například Proof of authority, Proof of reputation, Proof of importance, Proof of elapsed time a Delegated proof of stake. Dále také Proof of Work a Proof of Stake, které budou blíže představeny v následující kapitole.

3.3.2.1 Problém byzantských generálů

Koordinační problém účastníků decentralizovaného systému, z nichž někteří navíc můžou být nečestní, se nazývá problém byzantských generálů.

Problém byzantských generálů představuje konceptuální scénář, který zobrazuje složitosti, jež mohou vzniknout při snažení dosáhnout dohody a koordinace v rámci distribuované sítě, kde účastníci (generálové) musejí dosáhnout shody ohledně společného postupu. Tento problémový scénář zahrnuje tři byzantské generály, který každý vede svou vlastní armádu a nachází se kolem nepřátelského města. Pro úspěšný útok na město je zapotřebí souhlas všech generálů s koordinovaným časem útoku. [34, 35, 36]

Avšak komunikace mezi generály je nespolehlivá a může probíhat pouze prostřednictvím poslů, kteří musí překonat nepřátelské území. Tímto vzniká možnost pro různé problémy, které mohou ovlivnit správné doručení zpráv:

1. **Selhání komunikace**. Poslové mohou být zajati, eliminováni nebo zdrženi během své cesty mezi jednotlivými stanovišti. To vytváří nejistotu ohledně toho, zda budou zprávy úmyslně doručeny do svého cíle včas a v nepoškozeném stavu.
2. **Zásah do zpráv**. V případě zajetí poslů má nepřítel možnost získat přístup k zprávě a buď ji upravit, nebo naopak poslat klamnou zprávu k ostatním stanovištím. Toto by mohlo vést k šíření nepravdivých informací mezi generály, což by mohlo negativně ovlivnit proces rozhodování.
3. **Zrádní generálové**. Je také možné, že jeden či více generálů by mohlo být zrádci, kteří záměrně šíří falešné zprávy s cílem manipulovat a zavést ostatní generály k nesprávným rozhodnutím. Tato situace by mohla vážně narušit schopnost generálů dosáhnout konsenzu a společně se dohodnout na akci. [34, 35, 36]

Problém byzantských generálů upozorňuje na komplexitu dosahování dohody a koordinovaných rozhodnutí v prostředí, které je distribuované a potenciálně nepřátelské. Tento problém má široké uplatnění napříč různými oblastmi, včetně informatiky a blockchain technologie, kde má dosažení shody mezi distribuovanými uzly klíčový význam. Mechanismy konsenzu, jako je Proof of Work a Proof of Stake, které nalézáme v blockchainových sítích, jsou koncipovány tak, aby řešily podobné výzvy. Zajišťují, že účastníci mohou souhlasit s aktuálním stavem sítě nebo platností transakcí, přestože se mohou vyskytnout nepřátelské útoky nebo selhání komunikace. [34, 35, 36]

„Když toto dilema přeneseme do kontextu blockchainu, tak každý generál představuje síťový uzel a uzly musí dosáhnout konsenzu ohledně aktuálního stavu systému. Jinými slovy, aby nedošlo k úplnému selhání, musí se většina účastníků distribuované sítě shodnout a provést stejnou akci.“ [36] Existuje několik různých přístupů k řešení problému byzantských generálů, které si kladou za úkol vytvořit systém schopný odolat byzantským chybám – taková řešení se označují jako

algoritmy konsenzu a usilují o dosažení odolnosti proti chybám na blockchainu. [36, 38]

3.3.2.2 Proof of Work

Algoritmus konsenzu „důkaz práce“ (Proof of Work, zkráceně PoW) se stal nepostradatelným prvkem v oblasti kryptoměn, který nejvíce proslavil přímo Bitcoin. Tento sofistikovaný mechanismus konsenzu vyžaduje značné množství výpočetního úsilí jako důkaz provedené práce, proto vytváří pevný základ pro ověřování a potvrzení transakcí a nových bloků v rámci blockchainu. [34, 40, 41, 42]

Jeho podstata spočívá v tom, že účastníci, známí také jako uzly, jsou nuceni řešit náročný matematický problém pomocí výpočetního výkonu svých počítačů. Tento problém je konstruován tak, aby jeho řešení vyžadovalo značné množství času a výpočetního úsilí. To má za následek, že nalezení řešení je náročné a náhodné a nelze ho předem odhadnout, což zajišťuje náhodnost celého procesu. Přesto lze ostatními účastníky snadno ověřit, že práce byla provedena v souladu s pravidly. Účastník, který první najde správné řešení tohoto matematického problému, získává nejen prestižní právo vytvořit nový blok transakcí, ale také právo na odměnu za svoji účast v procesu. Tato odměna je udělována často ve formě kryptoměnových jednotek. [34, 40, 41, 42]

3.3.2.3 Proof of Stake

Dalším algoritmem konsensu je tzv. „důkaz hodnoty“ (Proof of Stake, zkráceně PoS). Mechanismus je v podstatě alternativou k dříve široce používanému algoritmu PoW, jehož znakem je náročnost na výpočetní výkon. PoS přináší změnu, jakým způsobem jsou bloky ověřovány v blockchainu. Vlastníci kryptoměny zde vkládají své krypto mince jako „zástavu“ za šanci ověřit bloky a získat tak odměnu. [34, 40, 41, 42]

Tento algoritmus náhodně vybírá tzv. validátory, což jsou majitelé kryptoměn, které jsou zastaveny jako záruka. Validátoři jsou pak odpovědni za vytváření nových bloků. Proces tohoto algoritmu je postaven na třech klíčových krocích. Nejprve je vybrán navrhovatel, který má na starost vytvoření nového bloku. Poté je

tento navržený blok podroben schválení a nakonec validaci, která potvrzuje jeho pravost. Navrhovatel, který navrhne blok, který je následně schválen, je odměněn podobně jako v případě PoW mechanismu, tedy přímo pomocí nativní kryptoměny. Držitelé s větším množstvím vlastněné kryptoměny mají větší šanci na výběr. Lze to tak přirovnat k loterii, kde každý, kdo si koupí lístek, má šanci na výhru. Nicméně lidé s větším počtem lístků mají vyšší pravděpodobnost výhry, protože mají více možností být vybráni. Tímto způsobem PoS dosahuje konsensu a zabezpečení sítě prostřednictvím náhodného výběru, přičemž držitelé většího množství kryptoměny mají větší podíl na této náhodě. [34, 40, 41, 42]

„PoS mechanismus nevyžaduje energeticky náročné těžební operace, což zlepšuje některé slabiny PoW konsenzu, jako je velká spotřeba energie, zatímco zachovává bezpečnost sítě.“ [34]

3.3.3 Hashování

Hashování se zabývá procesem generování výstupu pevné velikosti z proměnně velkého vstupu. Tento proces je realizován pomocí matematických vzorců nazývaných hashovací funkce. [69]

I když ne všechny hashovací funkce zahrnují kryptografii, kryptografické hashovací funkce jsou klíčovým prvkem kryptoměn. Díky nim jsou blockchainy a další distribuované systémy schopny dosáhnout vysoké úrovně integrity a bezpečnosti dat. [69]

Kryptografické hashovací funkce jsou deterministické. To znamená, že pokud se vstup nezmění, výstup z hashovacího algoritmu bude vždy stejný. Tato vlastnost je důležitá pro konzistenci a kontrolu integrity dat v různých aplikacích.

Hashovací metody používané v kryptoměnách obvykle fungují jako jednosměrné funkce, což znamená, že není snadné rekonstruovat původní vstup pouze z výstupu, a to bez značného úsilí a časových a výpočetních nároků. Jinými slovy, je relativně snadné vygenerovat hash z určitého vstupu, avšak poměrně složité provést opačnou operaci a získat původní vstup pouze z daného výstupu. [69]

3.3.3.1 Bezpečností hrozby

Blockchainy jsou navrženy s ohledem na bezpečnost a odolnost vůči různým druhům útoků. Nicméně žádný systém není zcela imunní vůči všem rizikům. V následující části bude představen výčet několika známých bezpečnostních hrozeb pro kryptoměny, potažmo blockchain.

- Dvojité utrácení (Double Spending)

Jak už bylo zmíněno v textu výše, při tomto útoku jde hlavně o snahu utratit stejné kryptoměnové prostředky několikrát. Tento útok je obzvláště problematický u kryptoměn, které nepoužívají rychlý mechanismus konsenzu. [44]

- 51% útok (51% Attack)

V obecném pojetí se jedná o útok, kdy se útočník snaží ovládnout více než 50 % sítě z důvodu získání kontroly, a tím nabýt možnost rozhodovat např. o podpisu transakcí, kdy útočník může na příklad zabránit zpracování jakýchkoliv transakcí nebo je částečně cenzurovat. Útočník poté může beztrestně provádět jakékoli další typy útoků, jako je např. dvojité utrácení. V extrémních případech může dojít až k manipulaci s konsensuálními pravidly blockchainu a nastavení nových pravidel, která budou v síti přijata. [44]

V mechanismu konsenzu PoW jde útočníkovi o získání minimálně 50 % celkového výpočetního výkonu. Vzhledem k tomu, že výpočetní výkon se odvíjí od investice ve značné množství elektřiny a výpočetních zdrojů, je získání této nadpoloviční části velice náročné, a to alespoň v Bitcoin síti, a případná finanční odměna je daleko menší než cena k provedení tohoto útoku. [44, 45, 46]

U mechanismu PoS jde útočníkovi o získání minimálně 50 % veškerých existujících nativních kryptoměn daného blockchainu, což se jeví u obrovských sítí na první pohled jako nemožné. [44, 45, 46]

U obou mechanismů záleží na velikosti sítě. Obecně platí, že čím je síť větší (obsahuje více uzlů), její tržní hodnota je větší, tím se šance na provedení takového útoku zmenšuje, protože k jeho provedení by bylo potřeba obrovské

množství peněz. [44, 45, 46] Např. u kryptoměny Ethereum, kde je tržní hodnota všech mincí v oběhu aktuálně (srpen 2023) 220 miliard dolarů, by případný útočník musel odkoupit, respektive by musel vlastnit přes 50 %, což odpovídá cca 110 miliardám amerických dolarů. [47]

- Sybil útok (Sybil Attack)

Sybil útok je druh útoku, který se vyskytuje ve sdílených sítích, včetně blockchainů [49]. Tento útok je pojmenován po knize „Sybil" od spisovatelky Flory Rhety Schreiber, kde hlavní postava trpí duševní poruchou rozštěpené identity [48], V kontextu blockchainu a sítí obecně se termín „Sybil útok" používá k popisu situace, kdy jedna osoba nebo entita vytvoří velký počet falešných identit nebo uzlů s cílem získat větší kontrolu nad sítí nebo ovlivnit konsenzus skrze hlasování. [44, 49]
Pomocí tohoto útoku lze také zaplavit síť falešnými transakcemi, a tím narušit propustnost, což vede k těžšímu zpracování legitimních transakcí. [44, 49]
„V průběhu let věnovali počítačoví vědci mnoho času výzkumu, aby přišli na to, jak Sybil útoky odhalovat a zabránit jim. Výsledky dosáhly různé míry účinnosti, ale žádná zaručená obrana prozatím neexistuje." [49]

- Eclipse útok (Eclipse Attack)

Omezujícím faktorem pro mnoho uzlů je šířka pásma. Přestože existuje obrovské množství zařízení s tímto softwarem, průměrné zařízení (uzel) nemůže přímo navázat spojení s mnoha z nich kvůli stanoveným omezením. Např. v Bitcoinovém softwaru je povoleno maximálně 125 spojení.
Při tomto útoku se útočník snaží izolovat specifický uzel od zbytku sítě tím, že uzavře jeho připojení k většině ostatních uzlů a snaží se veškerá spojení navázat na uzly kontrolované útočníkem. To může vést k tomu, že daný uzel bude přijímat a posílat pouze údaje od útočníka. [50]

- Zneužití chyb v chytrých smlouvách (Smart Contract Exploits)

Blockchain, jako je Ethereum, podporují chytré smlouvy, které mohou obsahovat chyby. Útočník může využít těchto chyb k odcizení kryptoměn nebo jiných digitálních aktiv, které jsou uloženy ve smlouvě. [22]

- Útok bez zastavených kryptoměn (Nothing at Stake Attack)

Tento druh útoku se váže hlavně k mechanismu konsenzu PoS, ve kterém může útočník provádět snahy o dvojité utrácení nebo jiné útoky bez rizika finanční ztráty, protože nemá ve skutečnosti zastavěné kryptoměny v síti. Je to způsobeno tím, že validátorům nic nebrání současně potvrzovat konfliktní verze historie. Validátor může poslat různé verze historie různým částem sítě nebo poslat jinou historii klientovi, který se právě připojil.
Rozšířený mechanismus „delegated Proof of Stake" tento problém do jisté míry řeší tím, že validátora, který vytvořil dvě historie, potrestá tím, že o svoje zastavené mince přijde. [51]

3.3.4 Bitcoin

Bitcoin je revoluční kryptoměnou a decentralizovaným platebním systémem, který byl poprvé představen v roce 2008 pod pseudonymem Satoshi Nakamoto v článku „Bitcoin: A Peer-to-Peer Electronic Cash Systém" [52]. Tento digitální fenomén změnil způsob, jakým lidé chápou peníze, platby a finanční transakce. Jeho základními prvky jsou blockchainové technologie, mechanismus konsenzu Proof of Work a předem známé finální množství kryptoměnových mincí.
Bitcoin je komunikační protokol a peer-based systém podporující přenos virtuálních měnových jednotek. Používá hashovací funkce a digitální podpisy k implementaci peněz, ale na rozdíl od řady předchozích návrhů se nespoléhá na centrální důvěryhodné autority. Peníze jsou přesouvány mezi stranami pomocí transakcí, jejichž vlastnictví je udáváno záznamy transakcí, veřejnými klíči a kontrolou odpovídajících soukromých klíčů. Systémová opatření brání podvodnému zdvojení měny (Double spending). [53]

3.3.4.1 Síť

Bitcoin je decentralizována síť typu peer-to-peer (P2P) a využívá speciální kryptografické protokoly a sotwarové aplikace k propojení uzlů přes internet. Jednotky měny v síti se nazývají Bitcoiny. Uživatelé mohou posílat a přijímat Bitcoiny přes síť tím, že odesílají své digitálně podepsané zprávy na ostatní uzly sítě pomocí peněženek (viz Kapitola 3.3.6) pro kryptoměnu Bitcoin. Tyto transakce měny jsou pak zaznamenány v distribuované, replikované veřejné databázi nebo účetní knize známé také jako blockchain. [54]

Síť Bitcoinu vyžaduje minimální infrastrukturu pro sdílení transakcí. Decentralizovaná síť dobrovolných účastníků nebo uzlů je dostatečně rozsáhlá. Zprávy jsou vysílány na nejlepší úrovni a uzly mohou opustit a znovu se připojit k síti podle svého uvážení. Při opětovném připojení k síti se každý uzel stahuje a ověřuje nové bloky od ostatních uzlů, aby udržel svou lokální kopii blockchainu. [54]

3.3.4.2 Těžba

V procesu těžby Bitcoinu je klíčovým prvkem mechanismus konsenzu Proof of Work (PoW). Nové transakce na blockchainu jsou posílány do "mempoolu"[4] nevyřízených transakcí. Těžařův úkol spočívá v ověření platnosti těchto čekajících transakcí a uspořádání jich do bloku, které poté tvoří kandidátský blok. Těžař se snaží převést tento kandidátský blok na platný, tzv. potvrzený blok. K dosažení toho musí těžař řešit složitý matematický problém, který vyžaduje významné množství výpočetního výkonu. Za každý úspěšně vytěžený blok získá těžař blokovou odměnu, což zahrnuje nově vytvořené kryptoměny a také transakční poplatky. [79, 87, 90]

Prvním krokem v procesu těžby je získání nevyřízených transakcí z "mempoolu" a jejich postupné "hashování" pomocí hashovací funkce. Během tohoto procesu se generují pevně velké výstupy známé jako hash. [79, 87]

[4] Mempool (zkratka slov memory a pool) je mechanismus kryptoměnového uzlu (node) pro ukládání informací o nepotvrzených transakcích [91].

Druhým krokem je vytvoření Merkleova stromu z těchto hashů z předchozího kroku. Tímto procesem vznikne kořenový hash. [79, 87]

Ve třetím kroku je třeba najít platnou "hlavičku" bloku. Hlavička bloku funguje jako identifikátor pro každý blok a zajišťuje, že každý blok má jedinečný hash. Při tvorbě nového bloku těžaři kombinují hash předchozího bloku s kořenovým hashem svého kandidátského bloku, aby vytvořili nový hash pro tento blok. Součástí tohoto procesu je také přidání libovolného čísla, které je známé jako nonce[5]. [79, 87]

Když tedy těžař usiluje o ověření svého kandidátského bloku, musí spojit kořenový hash, hash předchozího bloku a nonce a všechny tyto hodnoty podrobit hashovací funkci. Jeho cílem je opakovat tento proces, dokud není schopen vytvořit platný hash. [79, 87]

Kořenový hash a hash předchozího bloku nelze změnit, což znamená, že těžaři musí opakovaně měnit hodnotu nonce, dokud není nalezen platný hash. Aby byl výstup (blokový hash) považován za platný, musí být menší než specifická cílová hodnota určená protokolem. Při těžbě Bitcoinů je požadováno, aby blokový hash začínal určitým počtem nul, což se nazývá obtížnost těžby. [79, 90]

Těžaři musí opakovaně hashovat hlavičku bloku s různými nonce a tento proces opakovat, dokud nenajdou platný blokový hash. Když takový hash najdou, těžař, který jej objevil, předá tento blok do sítě. Ostatní uzly pak prověří, zda je tento blok a jeho hash platný, a v případě schválení přidají nový blok do své kopie blockchainu. [79, 90]

V tomto okamžiku se kandidátský blok stane potvrzeným blokem a všichni těžaři pokračují v hledání dalšího bloku. [79]

3.3.5 Klíče a seed

Ve schématu kryptografie s veřejným klíčem používá odesílatel k zašifrování informace veřejný klíč, zatímco příjemce používá k jejímu dešifrování soukromý klíč. Vzhledem k tomu, že jsou tyto dva klíče odlišné, je možné veřejný klíč bezpečně sdílet, aniž by to jakkoli ohrozilo bezpečnost soukromého klíče. Každý

[5] Nonce odkazuje na číslo nebo hodnotu, kterou lze použít pouze jednou [87].

pár asymetrických klíčů je jedinečný, takže zprávu zašifrovanou pomocí veřejného klíče si může přečíst pouze osoba, která vlastní odpovídající soukromý klíč. [73]
Jedním z nejrozšířenějších algoritmů pro asymetrické šifrování, který se dnes používá, je algoritmus RSA. Tento modul vygeneruje dva klíče – jeden veřejný, který je možný sdílet, a druhý soukromý, který by měl zůstat utajen. Algoritmus RSA poprvé popsali v roce 1977 informatici Rivest, Šamir a Adleman a dodnes je hlavní součástí kryptografických systémů s veřejným klíčem. [73]
Kryptografie s veřejným klíčem se také významně podílí na technologii blockchain a u kryptoměn. Při zakládání nové peněženky se generuje dvojice klíčů (veřejný a soukromý klíč). Adresa peněženky se generuje pomocí veřejného klíče, takže ji lze bezpečně sdílet. Soukromý klíč se naopak používá k vytváření digitálních podpisů a na ověřování transakcí, a proto musí být utajen. [73]
Jakmile je transakce ověřena potvrzením hashe obsaženého v digitálním podpisu, může být přidána do blockchainové účetní knihy. Tento systém ověření digitálního podpisu zajišťuje, že finanční prostředky může přesouvat pouze osoba, která má soukromý klíč spojený s příslušnou kryptoměnovou peněženkou. [73]
Pokaždé, když někdo vytvoří krypto peněženku, vytvoří se řada čísel, kterým se říká seed. Pomocí těchto čísel program peněženky vygeneruje frázi 12 nebo 24 náhodných slov, z nichž každé je spojeno s určitým číslem v seedu. [77]
Seed neboli obnovovací fráze (recovery phrase) může vypadat např. takto:

office	fatigue	decrease	volume
shuffle	mention	proof	public
frown	used	biology	upgrade

Tabulka 1 Příklad obnovovací fráze, Zdroj: Autor

Část toho, co dělá počáteční frázi jedinečnou, je pořadí slov. Pořadí musí být zadáno ve správné návaznosti, aby fráze fungovala. Po zadání počáteční fráze do peněženky je přístup povolen, i když uživatel ztratil svůj soukromý klíč, nebo zařízení, na kterém měl peněženku, bylo odcizeno nebo rozbito. [77]

3.3.6 Způsob uchování a manipulace

Jediným způsobem uchovávání kryptoměn je použití kryptoměnových peněženek. Kryptoměnová peněženka představuje jednoduchý nástroj umožňující interakci s blockchainovou sítí. Rozlišujeme několik druhů těchto peněženek, které lze seskupit do tří kategorií: softwarové, hardwarové a papírové peněženky. V závislosti na způsobu, jakým fungují, mohou být také označovány jako online nebo offline peněženky. [71]

Navzdory obvyklému chápání, kryptoměnové peněženky neslouží k fyzickému ukládání digitálních aktiv. Místo toho slouží jako nástroje pro interakci s blockchainem. Tyto peněženky umožňují generování informací potřebných k provádění transakcí s kryptoměnami na blockchainu, a to jak přijímání, tak odesílání. Tato data zahrnují páry veřejných a soukromých klíčů. [71]

Součástí kryptoměnové peněženky je také adresa, což je alfanumerický identifikátor vytvořený z veřejného a soukromého klíče. Tato adresa je konkrétním místem na blockchainu, kam lze zasílat mince. Adresu lze bezpečně sdílet s ostatními, aby mohli posílat platby. Naopak soukromý klíč slouží k přístupu k vlastním kryptoměnám bez ohledu na typ použité peněženky. To znamená, že i když by došlo k ohrožení vašeho zařízení, můžete své kryptoměny stále ovládat z jiného zařízení, pokud máte přístup ke správnému soukromému klíči nebo obnovovací frázi (seed). Je důležité si uvědomit, že kryptoměny nikdy fyzicky neopouštějí blockchain, pouze se přesunou z jedné adresy na druhou. [71]

Jak bylo zmíněno dříve, kryptoměnové peněženky se obvykle dělí do dvou hlavních typů: online a offline. Offline peněženky, také nazývané „cold wallets“, jsou navrženy tak, aby zůstávaly odpojeny od internetu. Namísto toho využívají fyzická média, jako jsou hardware peněženky nebo papíry, pro uchování klíčů v izolovaném a offline prostředí. Díky tomu jsou tyto peněženky odolné vůči rizikům spojeným s online hackerskými útoky. Offline peněženky jsou považovány za mnohem bezpečnější způsob uchovávání kryptoměn. [70]

Na druhé straně existují online peněženky, také označované jako „hot wallets“, které jsou nějakým způsobem připojeny k internetu. Tyto peněženky usnadňují rychlý a pohodlný přístup ke kryptoměnám, ale zároveň jsou náchylné k vyšším

rizikům spojeným s kybernetickými útoky a zabezpečením. Přestože online peněženky nabízejí pohodlí, měly by být používány opatrně a zabezpečeny vhodnými metodami, aby se minimalizovalo riziko krádeže nebo úniku kryptoměn. [70]

Další kategorií peněženek jsou peněženky softwarové. Ty existují v několika typech a jsou většinou nějakým způsobem připojeny k internetu. Mezi takové patří např. webové, desktopové nebo mobilní. Webové peněženky jsou peněženky přístupné prostřednictvím prohlížeče, které nepotřebují žádnou instalaci ani stahování. Tyto peněženky často nalezneme na různých kryptoměnových burzách, jako je Coinbase, Kraken nebo Binance. Kategorie webových peněženek zahrnuje také rozšíření pro prohlížeče, například MetaMask nebo Trust Wallet. [70]

Desktopové peněženky jsou navrženy pro instalaci a použití na lokálním počítači. Poskytují obvykle větší kontrolu nad uživatelskými daty než webové peněženky. Některé příklady desktopových peněženek jsou Electrum a Exodus. Mobilní peněženky jsou určeny pro chytré telefony a jsou optimalizovány pro mobilní zařízení. Mezi ně patří Trust Wallet, MetaMask a také Coinbase. [70]

Následující kategorií jsou hardwarové peněženky, což jsou fyzická elektronická zařízení, která využívají generátor náhodných čísel (RNG) k vytváření veřejných a soukromých klíčů. Tyto klíče jsou poté uchovávány přímo v zařízení, které není připojeno k internetu. Hardwarové peněženky jsou takovým způsobem offline úložištěm a jsou považovány za jednu z nejbezpečnějších možností pro uchování kryptoměn. V této oblasti existují dva hlavní zástupci, a to společnosti Ledger a Trezor (česká firma). [70]

Poslední kategorií jsou papírové peněženky. Papírová peněženka je fyzický kousek papíru, na kterém jsou vytištěny kryptoměnová adresa a soukromý klíč ve formě QR kódů. Tyto kódy lze následně naskenovat pro provádění kryptoměnových transakcí. Některé webové stránky umožňují stáhnout si šablony papírových peněženek a generovat nové adresy a klíče i v režimu offline. Tímto způsobem jsou papírové peněženky relativně odolné proti online hackerům a lze je vnímat jako alternativu k offline úložištím. [70]

Přestože může být lákavé používat papírové peněženky, stojí za zmínku, že mají některé zásadní nevýhody a používání se v současnosti obecně považuje za

rizikové. Hlavní nevýhodou papírových peněženek je, že jsou vhodné pouze pro celkové odeslání celého zůstatku a neumožňují snadné částečné odesílání prostředků. [70]

3.3.7 Kryptoměna a krypto token

Tato kapitola se zaměřuje na podstatný rozbor rozdílů mezi třemi významnými koncepty v oblasti kryptoměn a digitálních aktiv. Zabývá se kryptoměnami (crypto currency), což jsou digitální formy měny, jež operují na základě kryptografických principů a decentralizovaných technologií.

Kryptoměny, jakožto zástupci moderního finančního ekosystému, se však neliší pouze v názvu od dalších významných entit, a to krypto mincí (crypto coin) a krypto tokenů (crypto token). Krypto mince jsou konkrétní inkarnací kryptoměn, jež často slouží k transakčním účelům a jsou vnímány jako platidlo pro digitální svět. Na druhé straně krypto tokeny představují mnohem širší spektrum digitálních aktiv, které mohou mít různorodé využití, včetně reprezentace podílu na určitém projektu, práv na digitálních platformách či umožnění specifických funkcionalit v rámci důmyslných chytrých smluv. [37, 43]

Výše uvedená klasifikace může být ovšem matoucí, a to především kvůli častým záměnám a nepřesnostem. Pro to, abychom byli schopni plně pochopit následující fáze této práce, se jeví jako klíčové jednotlivé pojmy rozebrat a zasadit je do kontextu.

Ačkoli mince a tokeny využívají technologii blockchainu, existuje mezi krypto mincemi a krypto tokeny několik významných rozdílů. Krypto mince představují formu digitální měny, známé také jako kryptoměna. Tyto mince často slouží jako původní „měna“ v rámci blockchainu a mají hlavní účel ukládat hodnotu a fungovat jako prostředek směny. Naopak krypto tokeny jsou digitálními aktivy, která vznikají na existujícím blockchainu, často díky použití chytrých smluv. Tyto tokeny mohou plnit rozličné funkce, od reprezentace fyzických objektů až po poskytování přístupu k různým službám a specifickým funkcím spojeným s danou platformou. [37, 43]

Krypto mince jsou skutečně původní pro své vlastní blockchainy. Například mince na Bitcoinovém blockchainu se nazývá Bitcoin (BTC) a na Ethereum blockchainu

máme Ether (ETH). Tyto krypto mince mají hlavní účel jako nástroj pro ukládání hodnoty a slouží jako prostředek směny, podobně jako tradiční fiat měny[6]. Tato schopnost mincí fungovat jako platidlo je také důvodem, proč se jim říká kryptoměny. [37, 43]

Zajímavý je rovněž způsob, jakým jsou tyto mince získávány. Většinou jsou kryptoměny získávány těžbou, což je proces, kdy se nové mince vytvářejí a existující transakce jsou ověřovány pomocí mechanismů konsenzu, jak je popsáno v předchozím textu. Těžba je způsob, jakým se do oběhu dostávají nové mince a jak je zajištěna bezpečnost a integrita blockchainové sítě. [37, 43]

Analogicky k tomu, jak jsou krypto mince konstruovány pomocí technologie blockchain, jsou i krypto tokeny navrženy prostřednictvím této technologie. Nicméně zásadní rozdíl tkví v tom, že krypto tokeny nejsou základním platidlem v rámci daného blockchainu. Spíše jsou vytvářeny nad stávajícím blockchainem a často využívají chytré smlouvy k plnění různých funkcí. Zatímco krypto mince simuluje tradiční měnu, krypto tokeny představují spíše aktiva nebo dokonce „práva“. [37, 43]

Krypto token může symbolizovat podíl v decentralizované autonomní organizaci (DAO viz. 3.6.3), digitální produkt, nezaměnitelný token (NFT viz. 3.6.4) nebo dokonce fyzický objekt. Krypto tokeny lze nakupovat, prodávat a obchodovat podobně jako krypto mince, ale neslouží jako prostředek směny mezi účastníky. Místo toho zastávají specifické účely v rámci ekosystému, kde jsou vytvořeny. [37, 43] Dále také platí, že na rozdíl o nativních kryptoměn, nejsou tokeny drženy účty. Tokeny existují pouze uvnitř chytré smlouvy, která je jako samostatná databáze. [75]

Pro srovnání s reálnými předměty lze krypto tokeny připodobnit k cennému předmětu, kupónu nebo poukazu, zatímco krypto mince mohou být přirovnány k běžným měnám jako euro nebo koruna.

Většina krypto tokenů je koncipována pro specifické použití v rámci projektu blockchainu nebo decentralizované aplikace. Oproti krypto mincím se tokeny

[6] Fiat měna je tzv. nucené oběživo neboli zákonné platidlo [71].

netěží, ale jsou vytvářeny a rozdistribuovány vývojářem či týmem za projektem. [37]

3.4 Ethereum

Ethereum je blockchain, který integruje počítačový systém. Slouží jako základní infrastruktura pro tvorbu aplikací a organizací, a to pomocí decentralizovaného přístupu, který nevyžaduje žádná povolení a zároveň je odolný vůči cenzuře. Díky tomu vzniká větší kontrola nad projekty a aktivitami, aniž by byla nutná závislost na autoritách třetích stran. [86, 88]

V rámci ekosystému Ethereum existuje jediný základní počítačový systém známý jako Ethereum virtuální stroj (Ethereum Virtual Machine, dále jen EVM). Stav tohoto virtuálního stroje je sdílen a uznaný všemi účastníky sítě. Každý jednotlivý uzel v síti Ethereum uchovává vlastní kopii tohoto stavu EVM. Kromě toho má každý účastník možnost zaslat požadavek na vykonání libovolného výpočtu na tomto počítači. [94]

V případě, že je takový požadavek odeslán, ostatní účastníci v síti jej ověří a provedou výpočet. To má za následek změnu stavu uvnitř EVM. Tato změna stavu je následně potvrzena a distribuována po celé síti, čímž se zajišťuje konsensus o novém stavu virtuálního stroje. Tímto způsobem Ethereum umožňuje provádět decentralizované výpočty a uchovává konzistentní stav EVM mezi všemi účastníky.

Výpočetní požadavky spojené s transakcemi jsou označovány jako „transakční požadavky“. Detailní záznamy všech provedených transakcí spolu s aktuálním stavem virtuálního stroje Ethereum (EVM) jsou uloženy v blockchainu. Tento blockchain následně projde procesem uložení a schválení všemi uzly v síti. [86, 88, 94]

Kryptografické mechanismy hrají klíčovou roli při zajišťování integrity transakcí. Jakmile jsou transakce ověřeny jako platné a přidány do blockchainu, zajišťuje kryptografie, že s nimi nelze později manipulovat. Tyto mechanismy také zabezpečují, že všechny transakce jsou podepsány a prováděny s odpovídajícími oprávněními vlastníků účtů neboli adres peněženek. [86, 88, 94]

Díky kryptografii jsou transakce v blockchainu chráněny před nedovolenými změnami a zároveň je zajištěno, že každá transakce je legitimní a autorizovaná. To je klíčové pro důvěru a bezpečnost celého systému. [94]

3.4.1 Ether

Ether (ETH) je nativní kryptoměna Etherea a také jediná přijatelná forma platby za transakční poplatky. Jeho hlavním účelem je vytvoření ekonomického prostředí, které podporuje vznik trhu pro výpočty. To znamená, že účastníci sítě mají ekonomickou motivaci provádět ověřování transakcí, plnit požadavky na výpočty a poskytovat výpočetní zdroje, čímž je zajištěna stabilita a funkčnost sítě, protože účastníci jsou odměňováni Etherem za svou účast a přínos k celkovému chodu blockchainového systému. [86, 95]

Každý účastník, který předkládá požadavek na provedení transakce v síti Ethereum, je povinen zaplatit určitou částku ETH jako odměnu za zpracování transakce. Tato odměna slouží jako poplatek za provádění výpočtů a ověřování transakce na síti. [86, 95]

Odměna je poté udělena tomu, kdo je zodpovědný za vykonání výpočtů a ověření transakce, což může být například těžař (v případě Proof of Work) nebo validátor (v případě Proof of Stake). Tímto způsobem je zajištěno, že účastníci, kteří přinášejí hodnotu a zpracovávají transakce, jsou odměněni za svou práci a zároveň jsou motivováni k udržení a podpoře funkčnosti sítě. [86, 95]

Výše zaplaceného ETH odpovídá zdrojům, které jsou potřebné k provedení daného výpočtu či transakce. Tyto poplatky mají také klíčovou úlohu v prevenci možných útoků, kdy by útočníci mohli zahlcovat síť nepřiměřeným množstvím transakčních požadavků (viz část o hrozbách). Stejně tak brání i útokům, při kterých by byly použity zdrojově náročné skripty (kódy), protože každý požadavek na síť musí být placený, a to za využité výpočetní zdroje. Tímto způsobem se zabezpečuje, že účastníci, kteří využívají síť, budou motivováni k používání sítě zodpovědným způsobem a nebudou zneužívat zdrojů. Taktéž se tak snižuje riziko nadměrného zahlcování sítě a zajišťuje se její efektivní fungování. [86, 95]

Termín „**minting**“ (česky „ražení“) se v kontextu Etherea používá k označení procesu, při kterém je vytvořen nový Ether (ETH) na blockchainu Ethereum. Je

důležité zdůraznit, že nový Ether může být vytvořen pouze platformou Ethereum samotnou, nikoliv běžnými uživateli. Tento proces „ražení“ je spojen s odměnou za ověření nových bloků, kdy je blok úspěšně navržen a přidán do blockchainu. Tímto způsobem je zajištěno, že nový Ether vstupuje do oběhu a může být použit pro různé transakce a aktivity na platformě Ethereum. [86, 95]

Proces „**burning**“ (česky „spalování“) Etheru je další důležitý aspekt v ekonomice Etherea. Když se Ether spálí, znamená to jeho trvalé odstranění z oběhu, což má významný dopad na celkový zásobovací limit této kryptoměny. Při každé transakci v blockchainu Ethereum, kdy uživatel platí za svou transakci, je základní poplatek za plyn (viz následující kapitola) spálen, tedy odstraněn. Tímto způsobem je zajištěno, že Ether je nejenom vytvářen jako odměna za nově vytvořené bloky, ale také je trvale odstraňován z oběhu v procesu běžných transakcí. Tento mechanismus spalování je navržen tak, aby bylo Ethereum ekonomicky udržitelné a aby se kompenzovalo vydávání nových Etherů odměnou za potvrzování bloků. Když je poptávka po transakčních poplatcích vysoká, Ethereum spálí více Etheru, než je vytvořeno, což může pomoci udržovat zásobu této kryptoměny v rozumných mezích. [86, 95]

Ethereum využívá několik menších zúčtovacích jednotek, které slouží k vyjádření hodnoty mnoha transakcí v jeho síti. Tyto jednotky umožňují přesnější udání transakčních poplatků a hodnoty Etheru v různých situacích.

Nejmenší jednotkou je „Wei“, což je základní jednotka pro hodnotu v Ethereu. Další významnou jednotkou je „Gwei“ (gigawei), což odpovídá jedné miliardě Wei. Gwei se často používá k udání poplatků za plyn a transakce v rámci sítě Ethereum, neboť samotné Wei by bylo kvůli své nízké hodnotě nepříliš praktické pro běžné použití. [86, 95]

3.4.2 Účty

Ethereum má dva typy účtů:

- Účet vlastněný externě (Externally-owned account)
- Účet chytré smlouvy (contract account)

Oba typy účtů umožňují přijímat, držet a posílat ETH a krypto tokeny v rámci ekosystému a vzájemně interagovat s nasazenými chytrými smlouvami.

Pro účet vlastněný externě platí, že jeho zřízení je zdarma. „Může iniciovat transakce, avšak předmětem transakce může být pouze ETH nebo krypto token. Skládá se z páru kryptografických klíčů – veřejného a soukromého klíče, které řídí aktivitu účtu.“ [96]

U smlouvy je zřizovací cenou poplatek za nasazení smlouvy na blockchain, protože používá síťové uložiště. „Transakce lze odesílat pouze jako odpověď na přijetí transakce. Transakce z externího účtu na účet smlouvy může spustit kód, který může provádět mnoho různých akcí, jako je převod tokenů nebo dokonce vytvoření nové smlouvy.“ [96]

Externí účet v síti Ethereum je vytvořen pomocí kryptografických klíčů, které zahrnují veřejný a privátní klíč. Tato párová klíčová struktura je klíčovým prvkem v kryptografii blockchainu, který zajišťuje autenticitu a bezpečnost transakcí. Veřejný klíč slouží k identifikaci uživatele nebo účtu a může být sdílen s ostatními, aby bylo možné přijímat platby a provádět transakce. Privátní klíč je tajný a slouží k podepisování transakcí. Skrze tento klíč uživatel dokazuje, že je skutečným vlastníkem účtu a má právo provádět transakce spojené s tímto účtem.

Je důležité zdůraznit, že když mluvíme o držení kryptoměn, ve skutečnosti držíme kontrolu nad příslušnými privátními klíči. Samotné kryptoměny jsou zaznamenány na blockchainu a jsou asociovány s veřejnými adresami. Uživatelé mohou odesílat platby a transakce tím, že podepíšou tyto transakce svým privátním klíčem.

„Při vytváření nového externího účtu nám většina knihoven vygeneruje náhodný soukromý klíč. Soukromý klíč se skládá z 64 hexadecimálních znaků a lze jej zašifrovat heslem.“ [96]

Příklad: ffffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd036415f

„Veřejný klíč je generován ze soukromého klíče pomocí Elliptic Curve Digital Signature Algorithm[7]. Veřejnou adresu pro externí účet získáme tak, že vezmeme

[7] Elliptic Curve Digital Signature (ECDSA) je kryptografický algoritmus pro vytváření a ověřování digitálních podpisů pomocí eliptických křivek [92].

posledních 20 bajtů hash Keccak-256[8] veřejného klíče a přidáme **0x** na začátek.“ [96] Adresa potom může vypadat například jako „0x8f433895656734cb4c00dd3d5fd53dd45f9a80cd“.

Nové veřejné klíče je možné odvodit z daného soukromého klíče, což umožňuje generovat nové adresy pro transakce a komunikaci. Na druhou stranu, je matematicky prakticky nemožné odvodit soukromý klíč z veřejných klíčů. Z toho důvodu je bezpečné uchovávání soukromého klíče nezbytné. [96]

Adresa chytré smlouvy je také reprezentována 42 znaky dlouhým hexadecimálním řetězcem, který začíná prefixem „**0x**“. Např. „0x9bfe44b38d32e6d333bbc5082fafd454ab6584b3“. Adresa smlouvy je odvozena z adresy tvůrce a počtu transakcí odeslaných z této adresy, což se nazývá „nonce“ (viz následující kapitola). [96]

3.4.3 Transakce

Transakce v síti Ethereum jsou kryptograficky podepsané instrukce, které mají za úkol aktualizovat stav sítě Ethereum. Každá transakce je iniciována určitým účtem a obsahuje instrukce, co má být provedeno – například převod ETH z jednoho účtu na druhý, provedení chytré smlouvy nebo jiné operace. [86, 88]

Transakce se týká akce, která je zahájena externě vlastněným účtem, jejž spravuje fyzická osoba (člověk) a není to chytrá smlouva. Když osoba provádí transakci, např. převod ETH z jednoho účtu na druhý, je tato akce vyjádřena v rámci transakce. Pokud např. Ábel pošle Kainovi 5 ETH, Ábelův účet musí být ponížen o 5 ETH a účet Kainův navýšen o 5 ETH. [81]

Kdykoli chce uživatel provést transakci, která změní stav EVM, musí ji vyslat do celé sítě. Každý uzel v síti může vysílat požadavky na provedení transakce v EVM. Když uzel odešle požadavek na transakci, validátoři se podílí na jejím ověření a následném provedení změny stavu EVM. Poté, co je transakce úspěšně ověřena a provedena, nový stav EVM, který zahrnuje tuto změnu, je šířen do zbytku sítě. To

[8] Keccak-256 je hashovací algoritmus, který slouží k vytváření krátkých, pevně dlouhých reprezentací (hashů) ze vstupních dat libovolné délky [93].

zajišťuje, že všechny uzly v síti budou mít konzistentní pohled na aktuální stav blockchainu. [81]
Odeslaná transakce obsahuje následující informace:

- from
  - Adresa externího účtu odesílatele, který bude transakci podepisovat (chytré smlouvy nemohou odesílat transakce).
- recipient
  - Adresa příjemce (v případě externího účtu dojde k převodu hodnoty, v případě chytré smlouvy transakce provede kód).
- signature
  - Vygeneruje se, když soukromý klíč odesílatele podepíše transakci a potvrdí, že odesílatel tuto transakci autorizoval.
- nonce
  - Postupně se zvyšující počítadlo, které ukazuje číslo transakce z účtu.
- value
  - Množství ETH, které se má převést od odesílatele k příjemci.
- input data
  - Volitelné pole pro zahrnutí libovolných údajů.
- gasLimit
  - Maximální množství jednotek plynu, které může transakce spotřebovat.
- maxPriorityFeePerGas
  - Maximální cena spotřebovaného plynu, která bude zahrnuta jako spropitné pro validátor.
- maxFeePerGas
  - Maximální poplatek za jednotku plynu, který je odesílatel ochotný zaplatit za transakci. [81]

Objekt transakce bude vypadat takto:

```
{
  from: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
  to: "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
  gasLimit: "21000",
  maxFeePerGas: "300",
  maxPriorityFeePerGas: "10",
  nonce: "0",
  value: "10000000000"
}
```

Obrázek 1 Objekt transakce, Zdroj: [81]

```
{
  "id": 2,
  "jsonrpc": "2.0",
  "method": "account_signTransaction",
  "params": [
    {
      "from": "0x1923f626bb8dc025849e00f99c25fe2b2f7fb0db",
      "gas": "0x55555",
      "maxFeePerGas": "0x1234",
      "maxPriorityFeePerGas": "0x1234",
      "input": "0xabcd",
      "nonce": "0x0",
      "to": "0x07a565b7ed7d7a678680a4c162885bedbb695fe0",
      "value": "0x1234"
    }
  ]
}
```

Obrázek 2 Příklad volání o podpis transakce, Zdroj: [81]

Objekt transakce však musí být podepsán pomocí soukromého klíče odesílatele. To dokazuje, že transakce mohla pocházet pouze od odesílatele a nebyla odeslána podvodně. Příklad volání o podpis viz Obr. 2. [81]

Příklad odpovědi:

Obrázek 3 Příklad odpovědi volání o podpis transakce, Zdroj: [81]

Pomocí hashe podpisu lze kryptograficky prokázat, že transakce přišla od odesílatele a odeslala se do sítě.

Na Ethereu existuje několik různých typů transakcí:

- Pravidelné transakce z jednoho účtu na druhý
- Transakce typu nasazení chytré smlouvy (bez cílové adresy)
- Provedení volání chytré smlouvy (cílová adresa je adresa smlouvy)

Životní cyklus transakce probíhá ve čtyřech krocích.

1. Transakční hash se generuje kryptograficky a vypadá např. jako „0x97d99bc7729211111a21b12c933c949d4f31684f1d6954ff477d0477538ff017“.
2. Transakce je poté odeslána do sítě a přidána do „bazénu“ transakcí, které čekají na ověření v síti.

3. Transakce musí být vybrána validátorem, který ji musí začlenit do bloku, aby ji ověřil a považoval za „úspěšnou“.
4. S průběhem času bude blok obsahující konkrétní transakci nejprve označen jako „justified“ a poté „finalized“. Tyto úrovně zabezpečení činí mnohem jistější, že daná transakce byla úspěšně zapsána a nebude nikdy změněna. [81]

3.4.4 Bloky

Bloky jsou dávkami transakcí, které jsou spojeny dohromady pomocí kryptograficky odvozených hashů. Každý blok obsahuje referenci na hash předchozího bloku v blockchainu, a tím vzniká řetězec bloků. [86, 88]

Kryptografická povaha těchto hashů a způsob, jakým jsou odvozovány z dat v bloku, zajišťuje, že bloky jsou provázány a nezměněný blok má stále stejný hash. Tím je zaručena integrita celého řetězce bloků. Pokud by někdo pokusil změnit jakékoli údaje v minulých blocích, změnil by se i hash těchto bloků a tato změna by se projevila všude ve všech následujících blocích. To by bylo snadno detekovatelné ostatními uzly v síti, což brání podvodům a zajišťuje důvěryhodnost blockchainu.

Dávkování transakcí do bloků je klíčovým prvkem, který umožňuje udržet synchronizovaný stav a dosáhnout dohody mezi všemi účastníky v síti Ethereum. Během tohoto procesu jsou transakce seskupovány do bloků, které jsou potom potvrzeny, odsouhlaseny a synchronizovány ve všech uzlech sítě. [86, 88]

Tímto způsobem se zajišťuje, že všechny transakce v síti jsou prověřeny, provedeny podle pravidel a v souladu s konsenzem. Poté, co je blok validován, je přidán k existujícímu blockchainu a stává se součástí celkové historie transakcí. Tento proces dávání transakcí do bloků umožňuje efektivní a spolehlivé fungování sítě Ethereum, což zajišťuje důvěru v její integritu a korektnost. [86, 88]

Bloky jsou přesně řazeny v řetězci, a to díky odkazům na nadřazené bloky, čímž vzniká nezvratný záznam transakční historie. Transakce samotné jsou také v rámci bloků přesně řazeny. [86, 88]

Současně všichni účastníci v síti s konkrétním blokem souhlasí a pracují na začlenění aktuálních živých transakcí do následujícího bloku. Tím je zajištěna

kontinuální aktualizace blockchainu a udržení shody mezi všemi uzly ohledně stavu sítě a historie transakcí. [86, 88]

Proces sestavení bloku je zrealizován náhodně vybraným validátorem, který provádí potřebné výpočty a validace pro vytvoření nového bloku. Jakmile je blok vytvořen, je šířen do celé sítě, kde je přidán ke konečnému blockchainu v každém uzlu. Následně je vybrán další validátor pro vytvoření dalšího bloku a proces se opakuje. [86, 88]

„I když se požadavky na transakce vyskytují desetinásobně za sekundu, bloky jsou na Ethereum vytvářeny a potvrzeny pouze jednou za 12 vteřin.“ [97]

Tyto časové prodlevy se nazývají „sloty“. Každý slot představuje časový interval, během kterého je vždy vybrán jeden validátor, který má za úkol navrhnout blok. Za předpokladu, že všichni validátoři jsou online a plně funkční, bude každý slot obsahovat blok, což zajišťuje pravidelné vytváření bloků každých 12 sekund. Nicméně se mohou vyskytnout situace, kdy někteří validátoři nejsou online nebo jsou nedostupní, když jsou voláni k vytvoření bloku v daném slotu. V takovém případě může být slot prázdný, což znamená, že blok v tomto časovém slotu nebude vytvořen. To může dočasně prodloužit dobu mezi vytvářením bloků. [86, 88]

Poslední důležitou poznámkou je, že samotné bloky jsou omezeny svou velikostí. Každý blok má cílovou velikost stanovenou na 15 milionů jednotek plynu, která je jednotkou pro měření výpočetních nároků transakcí a operací na síti Ethereum. Tato velikost bloku však může být dynamicky upravována v rámci určitých mezí až do maximálního limitu 30 milionů jednotek plynu. [86, 88]

Omezení velikosti bloků je důležité z několika důvodů. Zaprvé to zajišťuje, že bloky nebudou příliš velké a nezpůsobí nadměrnou zátěž na plné uzly sítě. Kdyby bloky mohly být libovolně velké, mohlo by to vést ke zpomalení provozu sítě a problémům s decentralizací, protože menší a méně výkonné uzly by mohly ztratit schopnost držet krok s nároky na velké bloky. [86, 88]

Zároveň větší bloky vyžadují více výpočetního výkonu pro zpracování a ověření transakcí. To může omezit schopnost menších uzlů rychle a efektivně zpracovávat bloky, což by mohlo vést k centralizaci sítě. Omezení velikosti bloků je tedy

důležitým opatřením, které pomáhá udržovat síť Ethereum decentralizovanou a odolnou proti nadměrné centralizaci. [86, 88]

3.4.5 Virtuální stroj Ethereum

Virtuální stroj Ethereum (Ethereum Virtual Machine, neboli EVM) „existuje jako jedna jediná entita udržovaná tisíci propojenými počítači na kterých běží Ethereum klient.“ [98] EVM je jádro sítě, které umožňuje provozovat chytré smlouvy a transakce na distribuovaném systému. Je to prostředí, ve kterém jsou prováděny veškeré operace v rámci sítě Ethereum. EVM udržuje stav všech Ethereum účtů a chytrých smluv a provádí výpočty na základě instrukcí obsažených v transakcích a smlouvách. Jeho hlavním cílem je zajistit konzistenci stavu mezi všemi uzly v síti a provádět operace v souladu s pravidly Ethereum protokolu. Každý uzel v síti udržuje svou kopii EVM a současný stav blockchainu. Změny stavu EVM jsou zaznamenány v blocích na blockchainu, a to díky procesu vytváření nových bloků a jejich ověřování. Každý nový blok, jenž je vytvořen, obsahuje odkaz na předchozí blok a transakce, které byly provedeny v tomto bloku. EVM se stará o výpočet nového stavu na základě těchto transakcí a předchozího stavu. [86, 88]

Díky EVM jsou účty a chytré smlouvy v síti Ethereum schopny provádět různé operace, jako je převod Etheru, spouštění programovatelných smluv a mnoho dalšího. Tímto způsobem je Ethereum schopné poskytovat rozsáhlé možnosti pro vytváření decentralizovaných aplikací a služeb. [86, 88]

Ve skutečnosti je Ethereum konečným automatem[9], kde stav zahrnuje všechny účty, zůstatky, chytré smlouvy a další informace. EVM definuje pravidla, jakým způsobem se tento stav mění v každém bloku, a umožňuje spouštění strojového kódu v podobě chytrých smluv. [86, 88]

Ethereum je mnohem více než jen jednoduchá distribuovaná účetní kniha. Jedná se o programovatelnou platformu, na které mohou vývojáři vytvářet chytré smlouvy

[9] Konečný automat je výpočetní model primitivního počítače, který se skládá z několika stavů a z několika přechodů a který dokáže přijmout nebo zamítnout předané slovo. [84]

a aplikace, které reagují na vstupy a provádějí složité akce podle definovaných pravidel. [86, 88]

EVM se chová podobně jako matematická funkce: na základě vstupu generuje deterministický výstup. Proto je poměrně užitečné popsat Ethereum formálněji jako systém s funkcí přechodu stavu:

$$Y(S, T) = S'$$

Zadáním starého platného stavu (S) a nové sady platných transakcí (T) funkce přechodu stavu Y(S, T) vytváří nový platný výstupní stav (S'). [98]

3.4.6 Plyn

Plyn (gas) je jednotka, která měří náročnost na výpočetní zdroje potřebné k provedení určitých operací, jako jsou převody, volání smlouvy a další akce. Platba za plyn je způsob, jak zajistit, že uživatelé, kteří chtějí provádět transakce nebo spouštět kódy v chytrých smlouvách, musí zaplatit za spotřebované zdroje. Poplatek za plyn je množství plynu použité k provedení určité operace, vynásobené cenou za jednotku plynu. Poplatek se platí bez ohledu na to, zda je transakce úspěšná nebo neúspěšná. [86, 88]

Poplatky za plyn musí být placeny v nativní měně blockchainu, tedy Etheru (ETH). Ceny plynu jsou obvykle uváděny v gwei, což je označení části ETH. Každý gwei se rovná jedné miliardě ETH (0,000000001 ETH nebo 10^{-9} ETH). Například místo toho, že plyn stojí 0,000000001 Etheru, lze říci, že plyn stojí 1 gwei. Slovo „gwei" je zkratkou pro „giga-wei", což se rovná 1 miliardě wei. Samotný wei je potom nejmenší jednotkou ETH. [99]

Při odesílání transakce může být nastaveno množství plynu, které je odesílatel ochotný zaplatit za provedení transakce. Tímto způsobem se bude transakce ucházet o zařazení do dalšího bloku. Je důležité najít správnou rovnováhu mezi množstvím plynu a cenou plynu, kterou je odesílatel ochotný zaplatit. Pokud nabídne příliš málo plynu, validátoři budou pravděpodobně tuto transakci ignorovat nebo ji zařadí na nižší prioritu zpracování, což by mohlo způsobit, že daná transakce bude provedena později nebo dokonce vůbec. Na druhou stranu, při příliš velké nabídce plynu dochází k nadměrnému poplatku za transakci.

Množství plynu a cena plynu jsou důležitými faktory při zasílání transakcí na Ethereu, protože ovlivňují rychlost a pravděpodobnost provedení transakce v síti. Celková cena plynu se dělí na dvě části: „base fee“ a „priority fee“. Základní poplatek (base fee) je stanoven protokolem a je nutné ho vždy zaplatit, aby byla transakce považována za platnou. Prioritní poplatek (priority fee) je prakticky spropitné pro validátory. Jeho výše je nepřímo motivuje k vybrání transakce pro další blok. [86, 88]

Transakce, u kterých je zaplacen pouze základní poplatek, je technicky platná, ale je nepravděpodobné, že bude zahrnuta do následujících bloků, protože nenabízí žádnou odměnu validátorům. Správná velikost priority poplatku je stanovena aktuálním vytížením sítě. Pokud je síť vytížena, bude nutné zaplatit více, pokud není, poplatek bude nižší. [86, 88]

Řekněme, že máme příklad, kdy Ábel chce poslat Kainovi 1 ETH. Převod vyžaduje 21 000 jednotek plynu a základní poplatek je 10 gwei. Ábel použije 2 gwei jako spropitné. Vzoreček pro výpočet poplatku za transakci bude (počet jednotek plynu * (základní poplatek + prioritní poplatek)). Základní poplatek je stanoven sítí a je v tomto případě 10. Po dosazení dostaneme následující výčet: 21 000*(10+2) = 252 000 gwei, což se rovná 0,000252 ETH. Z Ábelova účtu odejde 1,000252 ETH. Kainovi bude připsán 1 ETH. Validátor dostane spropitné ve výši 0,000042 ETH (2*21 000 gwei) a spálí se 0,00021 ETH (10*21 000 gwei). [99]

Pro provedení transakce v síti mohou uživatelé určit parametr maxFeePerGas, který určuje maximální poplatek, který jsou ochotni zaplatit za každou jednotku plynu v transakci. Pokud chceme, aby byla transakce provedena, maximální poplatek za jednotku plynu musí být vyšší, než je aktuální cena plynu na síti. Součet základního poplatku a spropitného tvoří celkový poplatek za transakci. Odesílateli transakce je poté vrácen rozdíl mezi maximálním poplatkem (maxFeePerGas) a celkovým poplatkem za transakci. Tímto způsobem mohou uživatelé lépe kontrolovat výši poplatků, které jsou ochotni platit za provedení svých transakcí. [86, 88, 99]

3.4.7 Sítě

Existuje hlavní síť Ethereum nazývaná Mainnet, která je určena pro běžné transakce a interakce. Je tedy primárním veřejným produkčním blockchainem. Kromě toho existují také testovací sítě, které slouží pro vývoj, testování a experimentování s chytrými smlouvami a aplikacemi, aniž by bylo nutné používat skutečný Ether. Tyto testovací sítě se nazývají testnets a jsou navrženy tak, aby byly izolované od hlavní sítě a neměly na ni vliv. [90, 100]

Existuje několik různých testnets, jako například Sepolia, Goerli, Rinkeby atd., každá s odlišnými vlastnostmi a použitím. Tyto testnets poskytují testovací Ethery, které se nijak nepropojují s reálným Etherem v hlavní síti. [90, 100]

Je důležité mít na paměti, že prostředky (ETH) a transakční historie nejsou sdíleny mezi různými sítěmi (Mainnet a testnets). To znamená, že pro každou síť (včetně testovacích) existuje samostatný účet a samostatné prostředky. [90, 100]

Používání testovacích sítí je užitečné, protože umožňuje vývojářům a uživatelům prověřit své aplikace a smlouvy v bezpečném prostředí, aniž by museli riskovat reálné finance a operace v hlavní síti. [90, 100]

3.4.8 Mechanismus konsenzu

Ethereum odhlasovalo historickou změnu (tzv. přechod) po vzájemné dohodě většinové části sítě a na podzim roku 2022 změnilo svůj mechanismus konsenzu z PoW na PoS. V angličtině je tato změna nazývána „The Merge“. V reakci na změnu mechanismu se spotřeba energie pro provoz blockchainu snížila odhadem až o 99,95 %. V létě 2022 byla spotřeba sítě téměř stejná jako spotřeba celé České republiky. [101, 102]

Jedná se o jednu z klíčových změn, které se na Ethereum síti chystají. Dalším z cílů bude navýšení počtu transakcí za sekundu až do výše 100 tisíc (momentálně je síť schopna zvládnout okolo 10–15 transakcí za vteřinu) a celkové zlevnění transakcí. V Ethereum síti stojí transakce obvykle v desetinách až jednotek dolarů (poplatek převeden z ETH na fiat), ale existují výkyvy, kdy může poplatek vzrůst až na několik desítek, v extrémních případech, až stovek dolarů. [89, 101, 102]

Proof of Stake byl blíže již představen (viz Kapitola 3.3.2), přesto lze doplnit několik faktů. Validátoři v PoS nezajišťují složitý výpočetní proces, jak tomu bylo

u PoW, ale spíše vkládají určitý kapitál ve formě ETH jako záruku za své dobré chování a správné fungování sítě. [89, 101, 102]

V rámci PoS systému jsou validátoři zodpovědní za kontrolu, zda jsou nové bloky šířené po síti platné, a příležitostně sami vytváří a šíří nové bloky. Tímto způsobem se snaží zajistit bezpečnost a důvěryhodnost sítě. Validátoři jsou motivováni, aby dodržovali pravidla sítě a nezasahovali do svých vlastních zájmů, protože pokud by se pokusili podvést síť nebo jednat nečestně, mohou přijít o část nebo celý svůj vkládaný kapitál. Za podvod lze považovat dvě primární chování – navrhování více bloků v jednom slotu (prostor pro vznik nového bloku, tzv. jednou za 12 vteřin) a předkládání protichůdných atestací (hlasů). [89, 101, 102]

Tento mechanismus je navržen tak, aby odměňoval důvěryhodné a spravedlivé účastníky sítě a trestal ty, kteří by chtěli zneužívat systém nebo narušovat jeho integritu. [89, 101, 102]

Pokud se uživatel chce stát validátorem, je zapotřebí vložit 32 ETH (což odpovídá aktuálně cca 1,3 milionům korun) do tzv. „stake“, což je záruka za správné a poctivé chování v síti. Tato záruka zajišťuje, že validátoři budou motivováni provozovat síť spolehlivě a nezneužívat systém. Dalším krokem validátora je, že musí spustit tři samostatné části softwaru: realizačního klienta (provádí transakce), konsenzuálního klienta (komunikuje s ostatními validátory) a samotného validátora (provádí ověření a tvorbu nových bloků). Po vložení do smlouvy a spuštění softwaru se uživatel připojí k aktivační frontě, která omezuje počet nových validátorů, kteří se mohou k síti připojit. To zabraňuje náhlému přetížení sítě novými validátory. Po aktivaci obdrží validátor nové bloky od ostatních validátorů v síti. Každý nový blok obsahuje transakce, které musí být znovu provedeny a ověřeny. Validátor pak odešle hlas (atestaci) ve prospěch tohoto bloku, což znamená, že souhlasí s jeho platností a správností. [89, 101, 102]

3.4.9 Chytré smlouvy

V praxi účastníci sítě Ethereum obvykle nepíší nový kód pokaždé, když potřebují provést výpočet na EVM. Namísto toho vývojáři aplikací nahrávají předem vytvořené programy, které obsahují opakovaně použitelné kousky kódu, přímo do stavu EVM. Uživatelé pak mohou zadat požadavek na provedení těchto

programů s různými parametry. Tyto programy, které jsou nahrány do sítě a spouštěny přímo na ní, se nazývají chytré smlouvy (smart contracts). Tyto smlouvy jsou v podstatě samo vykonatelné (self-executing) kódy, které automaticky provádějí předem definované akce na základě splnění určitých podmínek. Chytré smlouvy umožňují provádět složité interakce a transakce mezi účastníky, aniž by bylo nutné stále psát nový kód nebo něco manuálně spouštět nebo potvrzovat. Tím se zjednodušuje a urychluje vývoj aplikací na blockchainu. [86, 103]

Chytrou smlouvu na velmi základní úrovni můžeme přirovnat k prodejnímu automatu. Je to prakticky skript, který je aktivován voláním s určitými parametry a následně provádí určité akce nebo výpočty, pokud jsou splněny předem dané podmínky. Můžete si to představit jako automat, který reaguje na vstupy a provede předem definované kroky. [86, 103]

Například prostřednictvím jednoduché chytré smlouvy dodavatele může být digitální aktivum vytvořeno a přiřazeno vlastnictví, pokud volající osoba pošle určité množství ETH konkrétnímu příjemci. Tímto způsobem jsou vytvářeny složitější interakce a operace, které probíhají automaticky a spolehlivě na základě definovaných pravidel, bez potřeby stále ručně zasahovat do procesů. [86, 103]

Každý má schopnost vytvořit chytrou smlouvu a následně ji zveřejnit v síti pomocí blockchainu, který funguje jako datová vrstva. Tento proces většinou zahrnuje poplatek, který je zaplacen síti za nasazení smlouvy do blockchainu. Poté mohou uživatelé v síti volat tuto chytrou smlouvu a tím „aktivovat“ její kód. I zde se opět často platí poplatek za provedení akce. [86, 103]

Díky těmto chytrým smlouvám mají vývojáři schopnost vytvářet a implementovat různě složité aplikace a služby pro uživatele. Mezi takové aplikace patří např. tržiště, finanční nástroje, hry a mnoho dalšího. Tímto způsobem je umožněna decentralizovaná tvorba, nasazování a interakce s aplikacemi, což otevírá obrovský potenciál pro různorodé inovace a využití blockchainové technologie. [86, 103]

3.4.10 Standardy chytrých smluv

Na stránce https://eips.ethereum.org/erc jsou k dispozici desítky, ne-li stovky jiných standardů vytvořených lidmi z komunity. Samotná síť Ethereum ale nejvíce propaguje výčet, který je znázorněn v této kapitole níže. [75, 104]

ERC je zkratka pro žádost o připomínky k síti Ethereum (Ethereum Request for Comments). Jedná se o technické dokumenty, které nastiňují programovací standardy na síti Ethereum. Standard ERC-20 vytvořil v roce 2015 Vitalik Buterin a Fabian Vogelsteller. Tento standard navrhuje relativně jednoduchý formát pro tokeny založené na Ethereu. Jakmile jsou nové tokeny vytvořeny, je možné je automaticky importovat se službami a softwarem podporující ERC-20, jako jsou např. softwarové peněženky, hardwarové peněženky, burzy apod. [75]

- ERC-20 – Standardní rozhraní pro zastupitelné (zaměnitelné) tokeny, jako jsou hlasovací tokeny, zastavitelné tokeny nebo virtuální měny.
    - ERC-1363 – Definuje rozhraní tokenu pro ERC-20, které podporuje spuštění kódu příjemce po **transfer** nebo **transferFrom** nebo kódu útraty po **approve**.
- ERC-721 – Standardní rozhraní pro NFT (nezaměnitelné tokeny).
    - ERC-2309 – Standardizovaná událost emitovaná při vytváření nebo přenosu jednoho či více nezaměnitelných tokenů pomocí po sobě jdoucích identifikátorů tokenů.
    - ERC-4400 – Rozšíření rozhraní pro roli zákazníka EIP-721.
    - ERC-4907 – Přidává časově omezenou roli s omezenými oprávněními k tokenům ERC-721.
- ERC-777 – Standard tokenu, který obsahuje určitá vylepšení oproti ERC-20, ale v současné době není doporučen používat.
- ERC-1155 – Standard tokenu, který může obsahovat jak zaměnitelné, tak nezaměnitelné tokeny.
- ERC-4626 – Tokenizovaný standard trezoru určený k optimalizaci a sjednocení technických parametrů trezorů s výnosem. [104]

3.4.11 Oracle

Oracles jsou zdroje dat, které získávají informace z vnějších datových zdrojů a následně je přenesou na blockchain, aby je mohly využít chytré smlouvy. Tato praxe je nezbytná, neboť chytré smlouvy běžící na Ethereu nemají možnost získat přímý přístup k informacím uloženým mimo blockchain. Vedle toho, že sbírají a přenášejí data mimo blockchain, mohou oracles také umožňovat přenášení informací z blockchainu do vnějších systémů. Například existuje oracle, které umožňuje odemknout chytrý zámek, pokud uživatel odešle poplatek přes transakci na Ethereum. [86, 105]

Oracles slouží jako spojovací článek mezi chytrými smlouvami na blockchainu a datovými poskytovateli mimo řetězec. Bez nich by chytré smlouvy měly omezený přístup pouze k datům uloženým na blockchainu. Oracle poskytuje mechanismus, který umožňuje spouštět funkce chytrých smluv na základě dat získaných mimo blockchain. [86, 105]

Oracles se liší podle několika faktorů, jako je zdroj dat (jednoduchý nebo více zdrojů), model důvěryhodnosti (centralizovaný nebo decentralizovaný) a architektura systému (okamžité čtení, publikování–odběr a žádost–odpověď). Dalším způsobem, jak lze oracles rozlišovat je podle toho, zda získávají externí data pro použití v on-chain chytrých smlouvách na blockchainu (vstupní oracles), zda přenášejí informace z blockchainu do off-chain aplikací mimo řetězec (výstupní oracles) nebo zda provádějí výpočty mimo blockchain (výpočetní oracles). [86, 105]

Většina vývojářů vnímá chytré smlouvy jako jednoduché kusy kódu, které běží na určitých adresách v blockchainu. Nicméně širší náhled na chytré smlouvy ukazuje, že jsou to samostatné spustitelné softwarové programy, schopné automaticky prosazovat dohody mezi stranami, jakmile jsou splněny specifické podmínky – odtud i termín „chytré smlouvy“. [86, 105]

Použití chytrých smluv k prosazování dohod mezi lidmi v Ethereu není jednoduché, protože tato blockchainová platforma je deterministická. Deterministický systém vždy generuje stejné výsledky při konkrétním počátečním stavu a vstupu. Za účelem dosažení tohoto deterministického provádění

blockchainy omezují uzly na dosažení konsenzu na jednoduchých binárních (pravda/nepravda) otázkách a pracují pouze s daty uloženými na samotném blockchainu. Například otázka „Má tento účet dostatek prostředků k provedení transakce?“. Pokud by blockchainy přijímaly informace z vnějších zdrojů, determinismus by nemohl být dosažen, což by znemožnilo uzlům dosáhnout shody o platnosti změn stavu blockchainu. Situace, kdy uzel A provede kód chytré smlouvy a získá výsledek „10“, zatímco uzel B po provedení stejné transakce získá výsledek „19“, by mohla narušit konsenzus a ohrozit důvěryhodnost Etherea jako decentralizované počítačové platformy. [86, 105]

Výše popsaný scénář také zdůrazňuje problém s navrhováním blockchainů pro získávání informací z externích zdrojů. Oracle však tento problém řeší tak, že přebírá informace z off-chain zdrojů a ukládá je na blockchain, aby je mohly spotřebovat chytré smlouvy. Vzhledem k tomu, že informace uložené on-chain jsou neměnné a veřejně dostupné, mohou uzly Etherea bezpečně používat off-chain data importovaná z oracles k výpočtu změn stavu, aniž by došlo k porušení konsenzu. [86, 105]

Za tímto účelem se oracle obvykle skládá z chytré smlouvy běžící on-chain a několika komponent mimo řetězec. On-chain smlouva přijímá požadavky na data z jiných chytrých smluv, které následně předává off-chain části (označované jako oracle node). Tento uzel může komunikovat se zdroji dat například pomocí aplikačního programovacího rozhraní (API) a odesílat transakce k uložení požadovaných dat do úložiště chytré smlouvy. [86, 105]

V podstatě oracle blockchainu překlenuje informační propast mezi blockchainem a vnějším světem a vytváří „hybridní chytré smlouvy“. Tyto hybridní chytré smlouvy fungují na základě kombinace kódu z on-chain smlouvy a off-chain infrastruktury.

Skvělým příkladem může být systém, který poskytuje náhrady za zrušené lety. Chytrá smlouva dostává prostřednictvím oracle informaci přes API, že určitý let byl zrušen. Na základě této události chytrá smlouva iniciovaná oraclem spouští proces vrácení peněz za letenky všem postiženým zákazníkům. [86, 105]

3.5 Web 3.0

Tato kapitola se věnuje novému etapovému vývoji internetového prostředí, který představuje Web 3.0. Po úspěších Webu 1.0, kdy internet propojoval informace, a Webu 2.0, kde se začaly utvářet sociální sítě a interaktivita, nás Web 3.0 zavádí do éry decentralizace, interoperability a umělé inteligence.

3.5.1 Historie

Web 1.0 byl prvním krokem v evoluci internetu. V této první fázi byli většinou konzumenti obsahu a tvůrci byli hlavně vývojáři, kteří vytvářeli statické webové stránky s informacemi ve formě textu a obrázků. Období Webu 1.0 trvalo přibližně od roku 1991 do roku 2004. Charakteristickým rysem Webu 1.0 byly statické weby, které neposkytovaly dynamický obsah. Data a informace byly spíše načítány ze statických souborů než z databází a webové stránky byly zcela neinteraktivní. Web 1.0 lze přirovnat k internetu určenému hlavně pro čtení a konzumaci obsahu. [56, 57]

Éra Webu 2.0 odstartovala v roce 2004 s nástupem sociálních médií a nových interaktivních platforem. Web se transformoval z pouhého nástroje pro čtení na místo, kde bylo možné nejen číst, ale také vytvářet obsah. Společnosti začaly nabízet uživatelům platformy nejen pro konzumaci obsahu, ale i pro tvorbu vlastního obsahu a zapojení do interakcí s ostatními uživateli. S nárůstem online aktivit začala omezená skupina firem ovládat velkou část online provozu a vytvářené hodnoty. Model financování prostřednictvím reklamy také nabyl na důležitosti. Uživatelé mohli aktivně tvořit obsah, ale jeho vlastnictví a zisk z monetizace byl často v rukou těchto společností. [56]

Pro mnoho společností je nashromáždění většího množství dat klíčové pro vytváření personalizovaných reklam. Tímto způsobem se zvyšuje pravděpodobnost vyššího počtu kliknutí, což v konečném důsledku vede ke zvýšení příjmů z reklam. Soustředění a centralizace uživatelských dat tvoří základ pro fungování současné podoby internetu, jak ho známe a používáme dnes. [58]

Aplikace Webu 2.0 opakovaně zažívají úniky dat. Ve Webu 2.0 nemáme prakticky žádnou kontrolu nad vlastními daty, ani nad tím, jak jsou uložena. Všechna tato data pak vlastní a kontrolují společnosti, které mají tyto platformy na starosti.

Uživatelé, kteří žijí v zemích s omezenou svobodou slova, jsou také ohroženi. Vlády často mají schopnost vypnout servery nebo zabavit bankovní účty jednotlivcům, pokud mají podezření, že se vyjadřují názory nesouhlasící s oficiální propagandou. S centralizovanými servery je pro vlády relativně snadné zasahovat, řídit nebo dokonce zcela vypínat aplikace podle toho, jak to považují za vhodné. Díky tomu, že bankovní služby jsou také digitální a podléhají centralizované kontrole, vlády mají schopnost zasahovat i do této oblasti. Mohou zakázat přístup k bankovním účtům nebo omezit přístup k finančním prostředkům v době ekonomických krizí, extrémní inflace nebo jiných politických nepokojů. [56, 58]

Krátce po spuštění Etherea v roce 2014 spoluzakladatel Gavin Wood vyjádřil slovy řešení problému, který mnoho prvních osvojitelů kryptoměn pociťovalo: „Web vyžadoval příliš mnoho důvěry. To znamená, že většina webu, který dnes lidé znají a používají, spoléhá na to, že důvěřuje hrstce soukromých společností, aby jednaly v nejlepším zájmu veřejnosti.“ [56]

3.5.2 Web 2.0 vs. Web 3.0

Web 3.0 se stal univerzálním pojmem, který popisuje vizi nového a vylepšeného internetu. V jádru Webu 3.0 se nachází využití blockchainů, kryptoměn a NFT (non-fungible tokens), které mají za cíl vrátit uživatelům sílu a kontrolu ve formě vlastnictví. [56]

Mezi Webem 2.0 a Webem 3.0 existuje několik zásadních rozdílů, ale klíčovým prvkem je decentralizace. Web 3.0 přináší vylepšení stávajícího internetu o řadu dalších funkcí a vlastností. [58] O Webu 3.0 platí následující:

- Je decentralizovaný
  - Na místo, aby byl kontrolovaný a vlastněný centralizovanými subjekty, se jeho „vlastnictví“ rozděluje mezi jeho tvůrce a uživatele.
- Je bez oprávnění
  - Každý má stejný přístup, nikdo není omezený nebo nikomu není zakázán přístup.
- Má nativní platby

        - Používá kryptoměnu k placení peněz online, místo aby se spoléhal na „zastaralou“ infrastrukturu bank a zpracovatelů plateb.
    - Funguje bez důvěry ve 3. stranu
        - „Funguje pomocí pobídek a ekonomických mechanismů, místo aby se spoléhal na důvěryhodné třetí strany“. [56, 58]

V současné verzi webu jsou aplikace nasazovány na jeden server, s jednou databází, často spravované jedním poskytovatelem.
V rámci Webu 3.0 aplikace běží buď na blockchainech, decentralizovaných sítích mnoha P2P uzlů, nebo kombinaci obou, které tvoří konkrétní kryptoměnový protokol, např. Ethereum. Tyto aplikace se často označují jako dApps (decentralizované aplikace). Dalším pojmem často spojovaným s novou generací webu jsou kryptoměny. Je to tím, že jsou úzce spojeny s fungováním blockchainu.
Nativní platby pomocí kryptoměn, zprostředkovány peněženkami, jako je např. MetaMask, které umožňují integraci do webové aplikace, umožňují zprostředkovat snadné, anonymní a bezpečné mezinárodní platby a transakce bez nutnosti zadávání citlivých údajů. [58]
Správa identity v prostředí Webu 3.0 je také rozdílná, než jsme zvyklí. Současný stav umožňuje různé druhy přihlášení přes e-mail a heslo, OAuth, sociální sítě apod., které téměř vždy vyžadují, aby uživatelé předali citlivé a osobní údaje. Ve Webu 3.0 jsou identity vázány na adresu peněženky uživatele, který s aplikací interaguje. Tím je zajištěna bezpečnost, anonymita a možnost snadného „přenesení“ identity mezi aplikacemi v případě, kdy se uživatel rozhodne používat stejnou peněženku ve více aplikacích. [56, 58]
Web 3.0 umožňuje přímé vlastnictví prostřednictvím krypto tokenů nebo NFT. Nikdo, dokonce ani vývojáři projektu, do kterého se uživatel zapojí, nemá moc vzít vlastněný obsah zapsaný na blockchainu (NFT, krypto tokeny). Proto není nutné se obávat o ztrátu digitálního obsahu. Tento obsah je také umožňován svobodně obchodovat na dalších platformách nebo tržištích. [56]
Navzdory četným výhodám současného Webu 3.0 existuje mnoho omezení, které je potřeba ještě vyřešit. Důležité funkce, jako je přihlášení pomocí peněženky prostřednictvím blockchainu, jsou již dostupné pro všechny a bez poplatků.

Nicméně náklady spojené s transakcemi jsou stále mnoha lidmi považovány za relativně vysoké (viz Kapitola 3.4.6). Od toho se odvíjí i škálovatelnost blockchainů a jejich schopnost zpracování transakcí v daném čase. Také technická překážka při začátku používání technologií Webu 3.0 je příliš vysoká. Uživatelé musí chápat bezpečnostní otázky a hlavně fungování peněženek. [106]

„Web 3.0 je mladý a vyvíjející se ekosystém. Gavin Wood tento termín zavedl v roce 2014, ale mnohé z těchto nápadů se staly skutečností teprve nedávno. Jsme teprve na začátku vytváření lepšího webu s Web 3.0 technologiemi, ale jak pokračujeme ve zlepšování infrastruktury, která to bude podporovat, budoucnost webu vypadá jasně.“ [106]

3.5.3 Tokenomika

„Tokenovou ekonomiku lze chápat jako podmnožinu ekonomie, která studuje ekonomické instituce, politiky a etiku výroby, distribuce a spotřeby zboží a služeb, které byly tokenizovány.“ [77]

V rámci této kapitoly zastupuje slovo token jak krypto token, tak i kryptoměny.

Blokchainové projekty navrhují tokenomická pravidla kolem svých tokenů, aby podpořily nebo odradily uživatele od různých akcí. Mezi důležité faktory, které ovlivňují hodnotu tokenu, lze zařadit následující. [78]

Nabídka a poptávka jsou hlavními faktory, které ovlivňují cenu jakéhokoli zboží nebo služby. Totéž platí pro prodej kryptoměn. Existuje několik kritických ukazatelů, které měří nabídku tokenu. První se nazývá maximální nabídka, která reflektuje maximální množství tokenů, které bude existovat. Druhým je množství v oběhu, které se týká počtu tokenů v oběhu. Tokeny mohou být vyraženy (minted) nebo spáleny (burned), nebo mohou být uzamčeny jiným způsobem. To má vliv i na cenu tokenu. [78]

Dalším faktorem je užitečnost tokenu, která se vztahuje k případům použití určeným pro daný token, např. token, který reprezentuje speciální měnu v rámci ekosystému aplikace a umožňuje s ní odemykat nebo nakupovat různý další kontent. [78]

Kromě nabídky a poptávky je důležité podívat se na způsob distribuce tokenů. Existují dva způsoby uvedení a distribuce tokenů: první férové vydání, kdy před

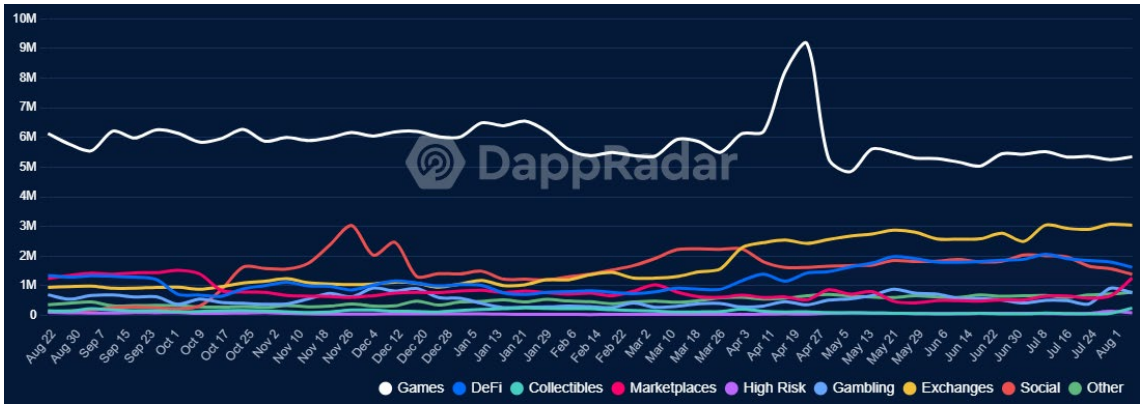
vyražením tokenů a jejich distribucí veřejnosti nedochází k předčasnému přístupu ani k soukromým alokacím; a druhým, vydáním před těžbou, které umožňuje vytěžit část krypto tokenů a distribuovat je vybrané skupině předtím, než jsou nabídnuty veřejnosti. [78]

Mnoho kryptoměnových projektů pravidelně pálí tokeny, což znamená jejich trvalé stažení z oběhu – například Ethereum, které trvale spálí určité množství tokenů u každé transakce. Když se nabídka tokenu sníží, je to považováno za deflaci. Opak, tedy když se nabídka tokenu neustále rozšiřuje, je považován za inflaci. [78]

3.6 Decentralizované aplikace

Decentralizované aplikace (dApps) představují aplikace, které fungují na blockchainu. Existuje jich velké množství a nabízejí různorodé využití, například v oblasti financí, hlasovacích systémů, her a dalších. I když se tyto aplikace mohou zdát podobné běžným aplikacím, odlišují se svým základním systémem. Namísto centralizovaného backendu využívají decentralizované aplikace chytré smlouvy na distribuované síti. Díky tomuto přístupu jsou tyto aplikace transparentnější, decentralizovanější a mohou lépe odolávat různým útokům. [59]

V této kapitole dojde k představení několika kategorií decentralizovaných aplikací.



Obrázek 4 Graf počtu aktivních a unikátních peněženek, Zdroj: [82]

Graf na obrázku č. 4 ukazuje počet aktivních a unikátních peněženek v průběhu jednoho roku. Výrazně nejvíce aktivních peněženek patří do kategorie GameFi. Následně se umístily různé platformy pro obchodování, směnárny a DeFi (Decentralizované finance) na druhé až čtvrté příčce.

3.6.1 Vlastnosti

Decentralizovaná aplikace (dApp) je aplikace postavená na decentralizované síti, která kombinuje chytré smlouvy a frontendové uživatelské rozhraní. Tyto aplikace mohou mít frontendový kód a uživatelská rozhraní napsaná v jakémkoli jazyce. Důležitá je jejich schopnost komunikovat s backend částí. [74]

Mezi jejich vlastnosti patří například:

- Decentralizace
  - DApps fungují např. na otevřené decentralizované platformě Ethereum, kde nemá kontrolu žádná osoba ani skupina.
- Determiničnost
  - Vykonávají stejnou funkci bez ohledu na prostředí, ve kterém jsou prováděny.
- Turningovsky kompletní
  - Mohou provést jakoukoli akci s ohledem na požadované zdroje.
- Izolované
  - DApps jsou spuštěné ve virtuálním prostředí, jako je např. EVM, takže pokud chytrá smlouva obsahuje chybu, nebude bránit fungování blockchainové síti. [74]

Výhody spočívají např. v soukromí, kde pro nasazení nebo interakci není nutné poskytovat skutečnou identitu. Navíc, jakmile je chytrá smlouva nasazena na blockchain, bude dostupná komukoli a kdykoli, tudíž není možné zahájit útoky typu denial of service (DoS) zaměřené na jednotlivé aplikace. DApps jsou také odolné vůči cenzuře – žádná jiná entita v síti nemůže uživatelům zablokovat odesílání transakcí, nasazovat nové smlouvy na blockchain nebo číst z něj data. Všechna data uložená na blockchainu jsou navíc neměnná. V neposlední řadě lze říci, že na chytré smlouvy lze veřejně nahlížet, tudíž je zaručeno, že budou provedeny předvídatelným způsobem, aniž by bylo nutné důvěřovat centrální autoritě. [74]

Mezi nevýhody patří např. údržba dApps, protože kód a data publikovaná na blochchainu se hůře upravují. Pro vývojáře je těžké provádět aktualizace svých dApps, jakmile jsou nasazeny, i když obsahují chyby nebo bezpečností rizika. Dále může nastat přetížení sítě. V současné době může síť, např. Ethereum, zpracovat pouze 10–15 transakcí za sekundu, právě proto platí, že pokud jsou transakce zasílány rychleji, může se skupina nepotvrzených transakcí rychle zvětšit a může nastat problém se zpracováním. Navíc může být těžší vytvořit uživatelsky přívětivé prostředí, protože pro průměrného koncového uživatele může být příliš obtížné nastavení nástrojů nezbytných pro interakci s blockchainem skutečně bezpečným způsobem. [74]

3.6.2 DeFi

„Decentralizované finance (DeFi) označují ekosystém finančních aplikací založených na blokchainových sítích.“ [62]

S DeFi poskytuje možnost provádět většinu činností, které jsou podporovány bankami, např. získávat úroky, půjčovat si peníze, poskytovat půjčky, kupovat pojištění, obchodovat s deriváty, obchodovat s aktivy a další, avšak vše probíhá rychleji a bez potřeby papírování nebo zprostředkování třetí stranou. [63]

DeFi aplikace nevyžadují žádné prostředníky nebo rozhodčí orgány. Kód specifikuje způsoby řešení všech možných sporů a uživatelé zůstávají neustále ve své kontrole nad svými finančními prostředky. Tato automatizace redukuje náklady spojené s poskytováním a používáním těchto finančních produktů a zároveň umožňuje plynulejší fungování finančního systému. [62]

DeFi jsou nejčastěji představovány formou decentralizované aplikace. Mezi výhody využívání DeFi patří například možnost půjčování vlastních kryptoměnových mincí dalším uživatelům a získávání úroků, které nejsou vypláceny jednou za měsíc, ale mohou být připisovány třeba každou minutu. Naopak získávání půjček je rychlé a probíhá bez nutnosti vyplňování papírových formulářů, jako je tomu u tradičních centralizovaných bankovních institucí. Obchodování s kryptoměnami může probíhat přímo mezi lidmi (P2P) bez jakékoli zprostředkovatelské role. Kromě toho je možné také spořit prostřednictvím kryptoměnových aktiv. [63]

Většina současných i budoucích aplikací v rámci decentralizovaného financování zahrnuje vytváření a provádění chytrých smluv. Kde běžné smlouvy využívají právní terminologii k definování podmínek mezi účastníky, chytré smlouvy pracují s počítačovým kódem. Díky tomu, že podmínky jsou zapsány v počítačovém kódu, mají chytré smlouvy unikátní schopnost automaticky vynucovat tyto podmínky. Tímto způsobem umožňují spolehlivou automatizaci mnoha obchodních procesů, které by jinak vyžadovaly manuální dozor. Využívání chytrých smluv přináší rychlejší a jednodušší způsob provádění smluv a zároveň snižuje rizika pro obě strany. [62]

3.6.3 DAO

„Decentralizovaná autonomní organizace, DAO, je komunitou vedená entita bez centrální autority, která se řídí počítačovým kódem. Protože pravidla, která určují chování organizace, jsou zabudována do jejího návrhu, má schopnost fungovat autonomně bez potřeby centrálního vedení.“ [61]

Oproti tradičním organizacím, kde existuje jednotlivá osoba nebo skupina, která má pravomoc rozhodovat a provádět jednostranná opatření, v případě DAO chybí tato centralizovaná autorita. Model řízení DAO, což je nový způsob právní struktury bez centrální moci, se opírá o návrhy, které členové komunity předkládají ke hlasování. Pokud je návrh podpořen většinou zúčastněných stran nebo splňuje určený soubor předem stanovených pravidel, je automaticky proveden. Pravidla DAO jsou definována základním týmem komunitních vývojářů v podobě chytrých smluv, které vymezují základní rámec fungování DAO. Tím je zajištěno, že pravidla a záznamy transakcí jsou transparentně uloženy na blockchainu. [61]

Členové DAO mohou mít odlišnou hlasovací sílu závislou na počtu řídících tokenů, které vlastní. Toto znamená, že člen se 100 tokeny bude mít dvojnásobný vliv ve hlasování než člen s 50 tokeny. Principem této praxe je, že jednotlivci s větší finanční investicí do DAO budou mít větší motivaci jednat v zájmu organizace.

Členové DAO nejsou vázáni žádnou formální smlouvou, nýbrž sdílí společný cíl a motivují se skrze síťové stimuly, které jsou založené na pravidlech konsenzu.

Tyto pravidla jsou zcela transparentní a zaznamenány v open-source softwaru, který řídí celou organizaci. [61]

3.6.4 NFT

NFT neboli Non-Fungible Token (Nezaměnitelný token) je kryptografický token uložený na blockchainu, sloužící k reprezentaci digitálního aktiva. Nezastupitelnost NFT spočívá v tom, že tyto digitální aktiva zastupují vlastnictví jedinečných položek, jako jsou umělecká díla, předměty ve videohrách, sběratelské karty, virtuální nemovitosti a další formy digitálního zboží. [64]

Pro NFT jsou důležité také chytré smlouvy, které umožňují vytváření, správu a převod NFT bez zprostředkovatelů automatizací a vynucování příslušných podmínek smlouvy. [64]

Správná implementace tokenových standardů je zásadním aspektem pro NFT, neboť tyto standardy zajišťují kompatibilitu a konzistenci mezi různými platformami. Tímto způsobem definují jednotná pravidla a funkce pro vytváření, řízení a přenos NFT. Například mezi nejznámější tokenové standardy pro NFT patří ERC-721 na Ethereu a BEP-721 na Binance Smart Chain, které přispívají k jednotnému a spolehlivému způsobu práce s těmito digitálními aktivy napříč různými blockchainovými sítěmi. [64]

Vytváření NFT je často označováno jako proces „ražby“ (minting). Tento proces využívá chytré smlouvy k transformaci digitálních souborů na digitální aktiva uložená na blockchainu. Když si člověk zakoupí NFT, získá vlastnická práva ke specifickému identifikátoru (ID tokenu), který je spojen s daným digitálním aktivem. Tímto způsobem se stává vlastníkem tohoto unikátního digitálního aktiva a získává exkluzivní práva k jeho použití, zobrazení a interakci. [64]

3.6.5 GameFi

Koncept GameFi představuje spojení blockchain technologie, decentralizovaného financování (DeFi) a světa videoher. Tento pojem se vztahuje k blockchainovým hrám s možností výdělku, které hráčům nabízejí ekonomické stimuly za jejich účast. V rámci ekosystému GameFi se vytváří virtuální herní prostředí, které využívá kryptoměny, krypto tokeny a NFT (Non-Fungible Tokeny).

Jádro projektů GameFi spočívá v konceptu "play-to-earn" (P2E), což je inovativní herní model. Tradiční videohry obvykle nepřinášejí hráčům finanční odměny a herní aktiva zůstávají vlastnictvím a pod kontrolou herní společnosti. Naopak P2E hry dávají hráčům plnou kontrolu nad jejich herními aktivy a zároveň jim poskytují příležitost k vydělávání skutečných peněz. [60]

Určité projekty zahrnují prvky a funkce DeFi, jako je staking, liquidity a yield farming[10]. Tyto modely umožňují hráčům uložit své herní tokeny a následně získat různé odměny, odemknout exkluzivní předměty nebo získat přístup k novým herním úrovním. [60]

Zahrnutí prvků DeFi může zároveň přispět k větší decentralizaci kryptoher. Některé GameFi projekty umožňují komunitě aktivně se zapojit do procesu rozhodování. Umožňují komunitním členům navrhovat a hlasovat o budoucích aktualizacích prostřednictvím decentralizovaných autonomních organizací (DAO) skrze jejich uzamčené tokeny. Čím více tokenů uzamknou, tím vyšší je jejich hlasovací síla. [60]

4 Vlastní implementace

V této kapitole bude popsáno, jakým způsobem bude vybrán vhodný blockchain a jeho standard, programovací jazyk a vývojové prostředí. Taktéž bude představen vývoj krypto tokenu pod zvoleným standardem a jeho finální nasazení na blockchain. V neposlední řadě bude vytvořena jednoduchá webová aplikace, která umožní autorizovat uživatele skrze webovou peněženku a jeho následnou práci s tokenem.

4.1 Analýza požadavků

Cílem této diplomové práce je vytvořit krypto token a jeho následná implementace ve webovém prostředí, která by otevřela možnosti postupného přechodu ze současných standardních aplikací Webu 2.0 na jeho nastupující verzi

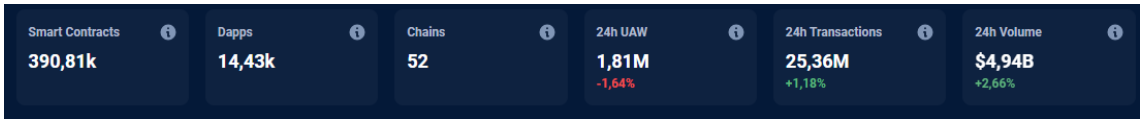
[10] "Staking", "liquidity" a "yield farming" jsou termíny spojené s kryptoměnami a decentralizovanými financemi (DeFi), které se týkají různých způsobů, jak vydělávat nebo získávat pasivní příjem prostřednictvím kryptoměn. [87]

3.0, kde je právě základním stavebním kamenem blockchain, potažmo kryptoměny a krypto tokeny. Zásadní výhodou takového krypto tokenu by mělo být jeho různorodé využití v možnostech adaptivních k aktuálnímu zaměření dané webové aplikace a případné uvedení tzv. tokenomiky (viz Kapitola 3.5.3), která podporuje zamýšlenou roli tokenu.

4.1.1 Srovnání blockchainových sítí

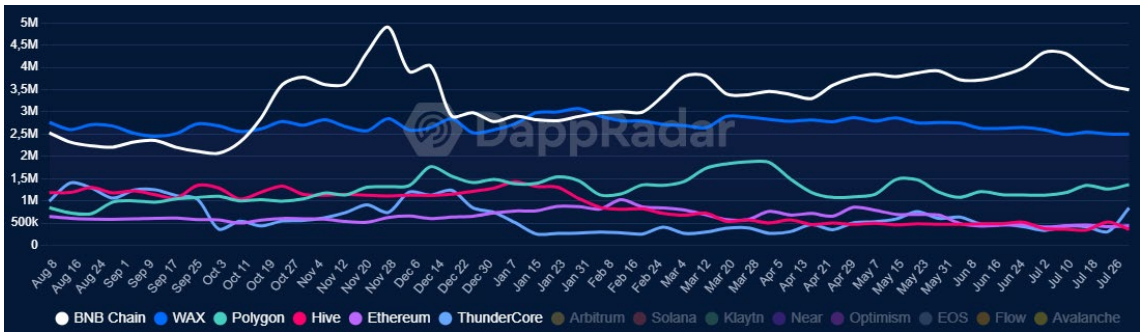
Pro vývoj krypto tokenu je klíčové správné zvolení blockchainové sítě, pro kterou bude token primárně vyvíjen, nasazen a využíván. Každý z níže představených blockchainů má své výhody i nevýhody. Obecně platí, že síť, která je nejvíce používána masou, nemusí být nejlepší volbou. Je poměrně důležité si položit otázku, jak často a jakým způsobem bude nutné s blockchainem pracovat, obzvláště zapisovat. Každý zápis totiž není zadarmo a stojí jisté zdroje, ať už v podobě poplatků nebo nutnosti uzamknutí kryptoměn (viz Kapitola 3.3.2). Dále je potřeba zvážit i dopad samotného blockchainu na časové zpracování dané transakce. Každý blockchain má nějaký limit na množství zpracovaných transakcí v daný čas. Zpravidla platí, že jak se zvyšuje objem transakcí nutných ke zpracování, tak se zvyšuje cena za jednotlivou transakci, což může vést ke zvyšování nákladů provozu celé webové aplikace, pokud je z větší části tvořena právě komunikací s blockchainem.

Na trhu neexistuje pravděpodobně lepší webová aplikace na vedení statistik a přehledu o chování decentralizovaných aplikací běžících na různých blockchainových sítích, než je DappRadar dostupná na webové stránce https://dappradar.com/. DappRadar byla spuštěna v únoru 2018 a nabízí veřejný přístup k informacím ohledně decentralizovaných aplikací, u kterých filtruje data, odstraňuje falešné a irelevantní aktivity a poskytuje užitečné informace o trhu. Aplikace jsou sledovány z hlediska jejich aktivních uživatelů, objemu tokenů a transakční aktivity, aby bylo možné nahlédnout do trendů v ekosystému decentralizovaných aplikací, potažmo aplikací napojených na blockchain. [7]



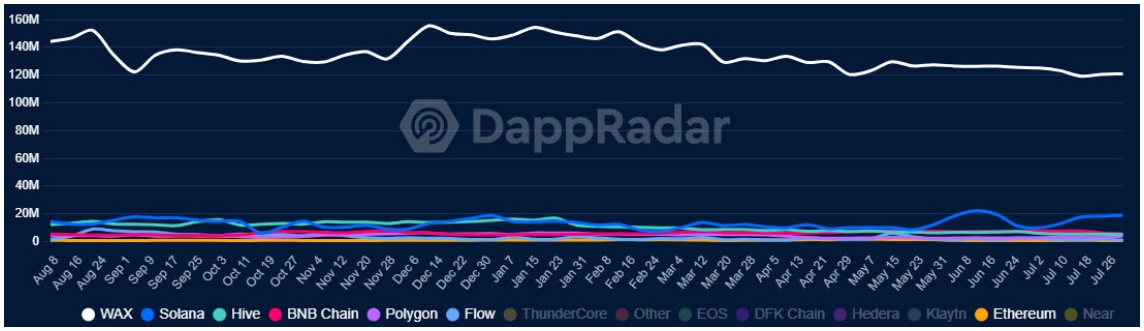
Obrázek 5 Přehled statistických dat z chytrých smluv, Zdroj: [82]

V současné době platforma DappRadar monitoruje a sleduje informace z 52 blockchainových sítí, na kterých je umístěno přes 390 tisíc chytrých smluv, které využívá téměř 14,5 tisíce decentralizovaných aplikací. Statistiky jsou navíc obohaceny o množství unikátních aktivních peněženek, které v posledních 24 hodinách komunikovaly s blockchainovou sítí, a to např. skrze chytré smlouvy, množství zapsaných transakcí a jejich hodnoty převedené na americké dolary.



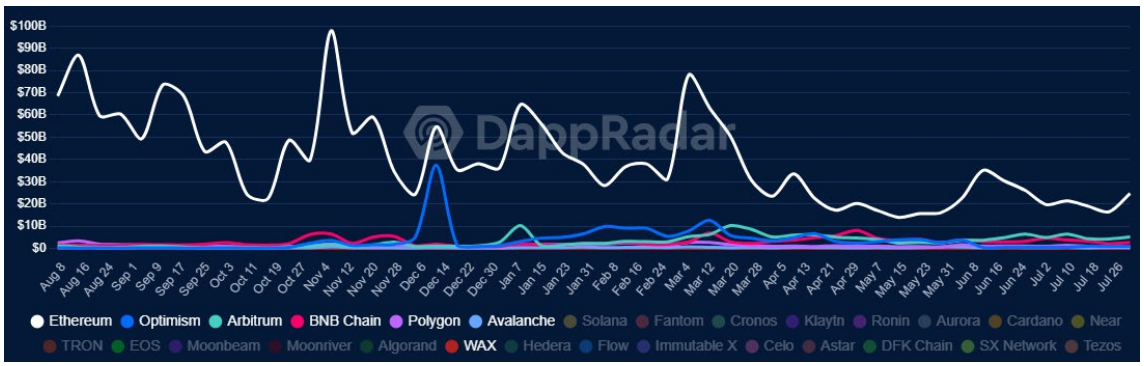
Obrázek 6 Graf aktivních unikátních peněženek, Zdroj: [82]

Jeden z rozhodujících faktorů při výběru vhodné blockchainové sítě, který může být stěžejní při rozhodování, je počet aktivních unikátních peněženek, které nějakým způsobem pracovaly s daným blockchainem. Na obrázku č. 6 s daty za uplynulý rok je zřejmé, že síť BNB Smart Chain jasně dominuje, v těsném závěsu je potom WAX. Další sítě, jako je Polygon, Hive nebo Ethereum, se obdobně drží ve spodní části grafu.



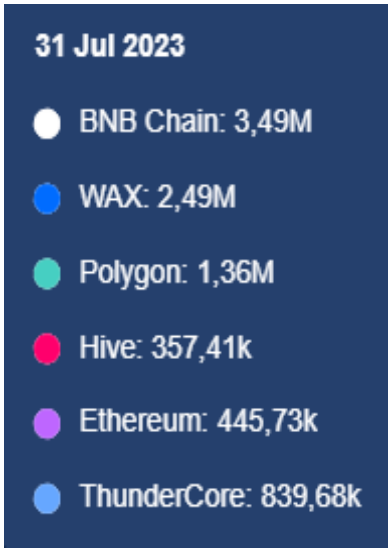
Obrázek 7 Graf počtu transakcí, Zdroj: [82]

Druhým faktorem, který by mohl být důležitý, je počet transakcí, které daná síť zpracuje během určitého časového rozpětí. Obrázek č. 7 zobrazuje, že rozhodně bezkonkurenčním výhercem je síť WAX, která několikanásobně převyšuje všechny ostatní. Je potřeba zdůraznit, že velkou roli hraje také kapacita jednotlivých blockchainů. Každá síť má jiné schopnosti a možnosti zpracovat různý objem transakcí. Síť WAX se pyšní kapacitou cca 2 500 transakcí za vteřinu [8], blockchain Solana dokonce počtem cca 4 200 transakcí za vteřinu [9], BNB Smart Chain stanovuje rychlost na cca 27 transakcí za vteřinu [10], Polygon téměř 23 [11] a v poslední řadě Ethereum síť s pouze maximálně 15 transakcemi za sekundu [12].



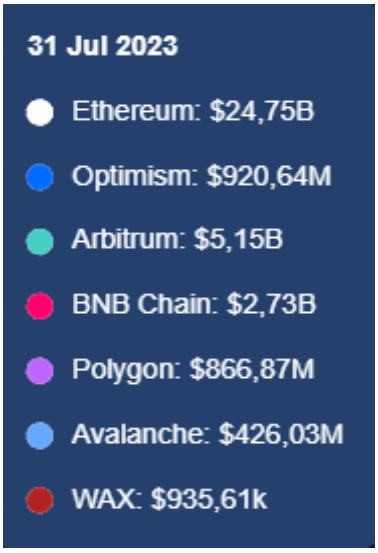
Obrázek 8 Graf celkového objemu transakcí, Zdroj: [82]

Třetím faktorem, který sleduje DappRadar, je celkový objem transakcí v amerických dolarech, viz obrázek č. 8. V této kategorii bezkonkurenčně vítězí asi nejznámější síť Ethereum, která se v jednom z období dostala v celkovém objemu téměř na hranici 100 miliard dolarů.



Obrázek 9 Legenda ke Grafu na obrázku č. 6, Zdroj: [82]

Obrázek 10 Legenda ke Grafu na obrázku č. 7, Zdroj: [82]



Obrázek 11 Legenda ke Grafu na obrázku č. 8, Zdroj: [82]

Na obrázcích č. 9, č. 10 a č. 11 můžeme vidět statistická data za konkrétní kategorie z časového úseku cca týdne v období kolem 31. července 2023.

V první kategorii unikátních aktivních peněženek vyhrál BNB Smart Chain, přesto se v dalších dvou kategoriích drží spíše v průměru. Tato síť, která vznikla hlavně díky světoznámé kryptoměnové burze Binance, je domovem převážně DeFi (viz Kapitola 3.6.2) projektů a dalších služeb, jako je PancakeSwap, ChainLink a Swipe z blockchainového světa. Samotná síť se pyšní vysokou propustností, nízkými transakčními poplatky a rychlou dobou dokončení jednotlivých transakcí. [13]

Ve druhé kategorii jasně dominuje již zmíněná WAX síť, která tvoří rekordní počet transakcí denně. Napomáhá jí k tomu její kapacita, která patří k jedněm z lepších. Přesto je její finanční objem ve třetí kategorii na posledním místě. WAX blockchain má vysoké zastoupení v rámci decentralizovaných aplikací, konkrétně v sekci GameFi (viz kapitola 3.6.5), kde je hojně využívána právě z důvodů rychlosti a relativně levných jednotlivých operací. Tento blockchain je hlavně domovem Alien Worlds projektu, kde je právě většina denních transakcí inicializována. Alien Worlds je momentálně nejpoužívanější decentralizovaná aplikace napříč kategoriemi z hlediska počtu použití univerzálních peněženek během 24 hodin. [8, 14]

Třetí kategorii jasně dominuje síť Ethereum, která přebíjí zbylé sítě. Přesto, že je síť relativně pomalá vzhledem k ostatním z výběru, její počet transakcí je poměrně

nízký. To platí také o počtu unikátních peněženek, které se sítí interagují. Tak je její finanční objem provedených transakcí zdaleka nejvyšší. Právě tento blockchain je domovem původních chytrých smluv, DeFi a MakerDAO protokolů. Ethereum stanovilo první standardy (známé jako ERC-20) pro tvorbu krypto tokenů (viz Kapitola 3.4.10), nespočet tokenů na ní také vyrostlo. Proto se nelze divit, že je tato síť nazývána zaslouženě jako průkopnická.
Pro splnění účelu této práce bude vybrána právě síť Ethereum na základě jejího prvenství v oblasti chytrých smluv. Taktéž se jedná o síť, která vzbuzuje důvěru velkého množství lidí, jelikož je zároveň sítí s největším finančím objemem transakcí.

4.1.2 Srovnání chytrých smluv a standardů

Ze standardů, které nabízí Ethereum síť, byl vybrán ERC-20, a to z důvodu, že je primárním standardem pro vytváření krypto tokenů na této platformě. O standardu ERC-20 lze říci, že obsahuje tyto vlastnosti:

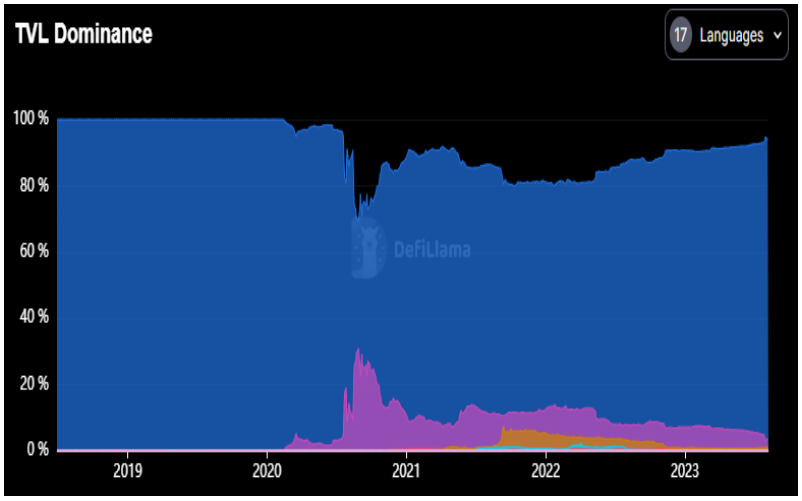
- Kompatibilita
    - Tokeny vytvořené v souladu s protokolem ERC-20 jsou navzájem kompatibilní a interoperabilní. Toto znamená, že mohou být jednoduše obchodovány a využívány různými aplikacemi a službami, které podporují tento standard.
- Transakční funkce
    - ERC-20 tokeny mohou být přenášeny mezi uživateli stejným způsobem jako ether (ETH), hlavní kryptoměna Etherea. Tím se umožňuje pohodlná transakce a výměna těchto tokenů.
- Kontrola
    - Díky standardu ERC-20 má každý uživatel možnost získat aktuální informace o svém zůstatku tokenů, což je klíčové pro sledování a správu vlastnictví.

- Transakční historie
    - ERC-20 tokeny uchovávají transakční historii, což uživatelům umožňuje sledovat, kdo a kdy provedl konkrétní transakci s těmito tokeny.
- Tokeny a Kontrakty
    - Vytvoření ERC-20 tokenů je možné pomocí chytrých smluv (smart contracts) na blockchainu Ethereum. Tyto smlouvy obsahují kód, který definuje pravidla pro token, včetně celkového množství tokenů, názvu, symbolu, počtu desetinných míst a dalších vlastností.
- Decentralizace
    - Tokeny vytvořené podle standardu ERC-20 zůstávají decentralizované, neboť jsou uloženy na blockchainu Ethereum, což je síť distribuovaná po celém světě. [107]

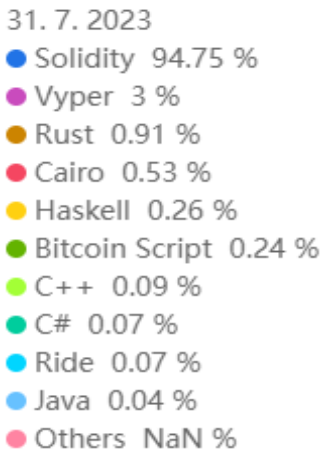
Mezi dalšími blockchainy mimo Ethereum při srovnávání a výběru byly ještě další, jako je např. Polygon síť, ThunderCore anebo BNB Chain. Všechny tyto tři sítě a jejich standardy vychází původně z ERC-20, tudíž jejich standard pro krypto tokeny se moc neliší. [107, 108, 109]
Blockchain, který se naopak liší, je WAX, který ve srovnání s ostatními výše zmíněnými nevychází z ERC-20 a není ho možné programovat ani v Solidity. Pro vývoj je nutná znalost jazyka C++ a vývojové prostředí funguje převážně pouze jako CLI (command-line interface). [110]

4.1.3 Programovací jazyky chytrých smluv

Jazyk chytrých smluv je programovací jazyk, který se používá k vytváření samoobslužných smluv, které automaticky vynucují smluvní podmínky.
Stručně řečeno, chytré smlouvy jsou počítačové programy, které jsou nasazeny a spuštěny v blockchainové síti a nabízí deterministické záruky, které umožňují více stranám dospět k dohodnutému výsledku odolnému proti neoprávněné manipulaci.



Obrázek 12 Graf dominance jazyků v chytrých smlouvách, Zdroj: [15]



Obrázek 13 Legenda ke grafu na obrázku č. 12, Zdroj: [15]

Na výše uvedeném obrázku můžeme vidět tabulku programovacích jazyků a jejich TVL Dominance (Total value locked [16]), což je celková hodnota kryptoměn uzamčených v chytrých smlouvách, ke dni 31. 7. 2023, kde jasně dominuje jazyk Solidity s téměř 95 % z celku. Na grafu je také vidět, že Solidity stále drží celkem jednoznačné prvenství během posledních let.

4.1.3.1 Solidity

Solidity je nejoblíbenější blockchainový programovací jazyk virtuálního stroje Ethereum (EVM viz Kapitola 3.4.5), který se také používá v celé řadě blockchainů kompatibilních s EVM. Solidity je vyšší objektově orientovaný jazyk, turingovsky úplný, což znamená, že vývojáři mohou psát kód podstatně rychleji,

protože mnoho problémů na nízké úrovni je již odstraněno. Tento jazyk spadá do kategorie se složenými závorkami a je ovlivněn především jazykem C++, Pythonem a JavaScriptem. [18, 86] Vliv z C++ je vidět v syntaxi deklarací proměnných, cyklů, konceptu přetěžování funkční, implicitních a explicitních převodů typů a mnoha dalších detailů. V počátcích tohoto jazyka býval Solidity částečně ovlivněn jazykem JavaScript. To bylo způsobeno rozsahem proměnných na úrovni funkcí a použitím klíčového slova **var**. Vliv JavaScriptu byl snížen počínaje verzí 0.4.0. Hlavní zbývající podobnost s JavaScriptem je v tom, že funkce jsou definovány pomocí klíčového slova **function**. Solidity také podporuje syntaxi a sémantiku importu, které jsou podobné těm, které jsou k dispozici v JavaScriptu. Další vliv na Solidity měl Python. Byly přidány modifikátory Solidity, které se snažily modelovat dekolátory Pythonu s mnohem omezenější funkčností. Kromě toho je z Pythonu převzata vícenásobná dědičnost, lineralizace C3 a **super** klíčové slovo, stejně jako obecné přiřazení a kopírování sémantiky hodnotových a referenčních typů. [19, 86]

Solidity je staticky typována, podporuje mimo jiné dědičnost, knihovny a složité typy definované uživatelem. Se Solidity můžeme vytvářet chytré smlouvy pro využití jako je hlasování, crowdfunding, slepé aukce a peněženky s více podpisy. [17, 86]

Pro nové vývojáře chytrých smluv je další výhodou to, že Solidity přichází s vestavěnými ochrannými opatřeními, která mohou zabránit nákladným chybám. Jedná se o nejrozšířenější jazyk ve vývoji Webu 3.0, z tohoto důvodu mají vývojáři přístup k většímu počtu knihoven a nástrojů, lepší dokumentaci a zvýšené podpoře na fórech. Na druhou stranu pro vývojáře, kteří nemají mnoho zkušeností s objektově orientovaným programováním, může být syntaxe Solidity někdy matoucí a její přetěžující funkce jsou často škodlivé v kontextu čitelnosti kódu. Existuje také několik zvláštností vývoje v Solidity, na které nemusí být vývojář zvyklý, jako je neschopnost nativně podporovat desetinná místa. [18, 86]

4.1.3.2 Vyper

Na druhé místo hned po Solidity lze zařadit jazyk Vyper. Je také druhým nejpoužívanějším jazykem pro vývoj na blockchainech kompatibilních s EVM. Vyper je smluvně orientován, což znamená, že byl speciálně vytvořen pro psaní chytrých smluv. Vychází z Pythonu a má podobnou syntaxi včetně silného typování. Mezi jeho další vlastnosti řadíme mimo jiné malý kompilovaný kód, efektivní generování byte kódu a umožnění zápisu záporných celých čísel s pevnou desetinnou čárkou (tj. např. funkce, kterou Solidity neumožňuje) [20], v neposlední řadě potom tzv. rozhodnutelnost, kdy je možné vypočítat přesnou horní hranici poplatku za transakci při volání jakékoli funkce. Jedním z hlavních principů jazyka je prakticky znemožnit vývojářům psát zavádějící programy (smlouvy). [21]

Může se nabízet otázka, proč používat Vyper, když existuje Solidity jako primární jazyk pro psaní chytrých smluv. Ve studii provedené v roce 2018 [22], kde byl zanalyzován téměř jeden milion nasazených chytrých smluv na platformě Ethereum, bylo zjištěno, že mnoho z těchto kontraktů mělo nějaké závažné zranitelnosti. Tyto zranitelnosti jsou produkovány v chytrých smlouvách prostřednictvím kódu, což nemusí být úmyslné, ale bez ohledu na záměry může nežádoucí kód vést k neočekávané ztrátě finančních prostředků pro uživatele. Vyper se tyto případné bezpečností problémy snaží eliminovat tím, že uživatelům umožňuje psát zabezpečený kód a vývojářům ztěžuje náhodný zápis zavádějícího, zranitelného nebo přímo nebezpečného kódu. [21]

Vyper má také jednoduchou implementaci jazyka a kompilátoru, která pomáhá s čitelností kódu a auditovatelností, což vývojářům usnadňuje vytváření chytrých smluv. A jako druhý nejrozšířenější programovací jazyk pro chytré smlouvy má Vyper mnoho stejných nástrojů a zdrojů jako Solidity. Přesto zde stále chybí široká komunitní podpora, kterou má Solidity, a postrádá nativní nástroje, které jsou v Solidity běžné. Vyper také postrádá modifikátory, dědičnost tříd, rekurzivní volání a není turingovsky kompletní. Nedostatek mnoha těchto funkcí je způsoben záměrem autorů s cílem maximalizovat auditovatelnost a zabezpečení smluv, kde vzhledem k eliminaci bezpečnostních hrozeb jsou tyto běžné funkce, které najdeme v Solidity, vynechány. Stojí za to také zmínit fakt, že je tento jazyk stále ve vývoji. [18, 21]

4.1.3.3 Ostatní

Solidity a Vyper jsou dnes dvěma velmocemi ve vývoji chytrých smluv, přesto existuje také řada nově vznikajících kódových jazyků blockchainu, jako jsou např.:

- Yul
- Cairo
- Rust

Yul je středně pokročilý jazyk pro Ethereum, který podporuje EVM. Tento jazyk byl vytvořen přímo pro převod do bytekódu, vyniká v optimalizaci chytrých smluv a snižuje náklady na poplatek za transakci. I když je Yul fantastický výukový zdroj, je nejlepší ho použít pro psaní konkrétního a výkonného kódu. Jako samostatný jazyk, v současné době, postrádá Yul nástroje a podporu ekosystému, je proto doporučen pro pokročilé vývojáře. [18]

Cairo je turingovsky úplný programovací jazyk pro chytré smlouvy, vytvořený pro programy postavené na technologii STARK (Scalable Transparent ARgument of Knowledge), která se používá pro zlepšení škálovatelnosti blockchainu. Cairo se převážně používá v rámci StarkNet blockchainu 2. vrstvy postavené na Ethereu a jeho základní funkcí je programová logika převedená na STARK důkazy, které poskytují ověřitelné výpočty na Ethereum blockchainu. I když je to výkonný jazyk pro vytváření rychlých a škálovatelných chytrých smluv, Cairo je mimo ekosystém StarkNet/StarkEx nepodporován. [18]

Rust je populární programovací jazyk pro chytré smlouvy pro mnoho blockchainů, které nejsou kompatibilní s EVM, jako jsou Polkadot a Solana, a na rozdíl od ostatních jazyků není určen pouze pro vývoj Webu 3.0. Samotný programovací jazyk Rust je efektivní, bezpečný a snižuje zbytečné nafukování kódu. Datové struktury v jazyce Rust jsou kompaktní, takže se hodí pro prostorová omezení blockchainu. Naopak je nedostatečné, že mnoho blockchainů ještě nemá plnohodnotné nástroje nebo robustní podporu pro tento jazyk. [18]

4.1.4 Vývojové prostředí

Na základě předchozí kapitoly bude zvolen programovací jazyk Solidity. Pro vývoj chytrých smluv a decentralizovaných aplikací na platformě Ethereum, které využívají jazyk Solidity, existuje několik vývojových prostředí a nástrojů, mezi které patří například:

- Remix IDE
- Visual Studio Code (s rozšířením pro Solitidy)
- Truffle
- Hardhat
- Ganache
- Solc [29]

Remix IDE je open-source vývojové prostředí, které nabízí webovou i desktopovou verzi. Je populární zejména z důvodu jeho možnosti okamžitého používání bez předchozího nastavení. Tento nástroj podporuje celý životní cyklus vývoje chytré smlouvy, tedy umožňuje je psát, testovat a nasazovat. Remix IDE se skládá ze čtyř částí, které rozdělují hlavní panel pro psaní kódu, terminál, pluginy/rozšíření a soubory. Remix má integrovaný editor kódu, debugger, testovací sady a nástroje pro nasazování, navíc poskytuje pomoc pro unit testy s nástroji, jako je CLI, assert knihovna a různé pluginy. [29, 30]

Visual Studio Code neboli také VS Code je oblíbený textový editor, který lze rozšířit o pluginy. Vývojáři Solidity mohou využívat různá rozšíření a zásuvné moduly, které se nabízí k vývoji decentralizovaných aplikací. Více než 50 rozšíření přímo podporuje vytváření chytrých smluv na Ethereum blockchainu. Rozšíření Solidity umožňuje vytvářet a upravovat kódy Solidity přímo ve VS Code. Toto rozšíření také nabízí funkce jako syntaxní zvýrazňování, auto kompletní funkce, kompilaci a integraci s testovacími rámci. VS Code také poskytuje integraci s technologiemi Webu 3.0, jako je Hardhat. [29,31]

Truffle je vývojový rámec určený pro vývoj, testování a nasazování chytrých smluv pro blockchainy Ethereum, Hyperledger, Quorum a dalších, které jsou kompatibilní s virtuálním strojem Ethereum. Vývojářům poskytuje řadu nástrojů pro vytváření

decentralizovaných aplikací, jako je automatizace různých vývojových procesů, včetně kompilace, migrace smluv, testování a nasazování. Při použití s Ganache, osobním blockchainem, a Drizzle, přední vývojovou sadou pro decentralizované aplikace, poskytuje Truffle komplexní řešení pro vývoj decentralizovaných aplikací. [29, 32]

Hardhat je další vývojový rámec pro vývoj chytrých smluv. Nabízí flexibilitu a rozšiřitelnost a je oblíbený pro své funkce jako rychlá kompilace, integrované testovaní a podpora pro sítě jako Ethereum a Biance Smart Chain. Testování je snadné kvůli lokální Ethereum sítí pro vývoj, která umožňuje nasazovat chytré smlouvy, tvořit testy a ladit kód. [29, 33]

Ganache je osobní blockchain, který lze použít pro lokální vývoj a testování chytrých smluv. Poskytuje falešný Ether a možnost simulace různých blockchainových scénářů. [32, 111]

Solc je překladač Solidity jazyka, který umožňuje kompilaci kódu do bytekódu pro Ethereum virtuální stroj. I když se jedná spíše o nástroj pro kompilaci, může být integrován do vývoje toků. [112]

Pro účely této práce bude jako vývojové prostředí vybrána webová verze Remix IDE především z důvodu velice rozsáhlých integrovaných funkcionalit, které zjednodušují vývoj chytrých smluv.

4.2 Vývoj krypto tokenu

Na základě předešlých kapitol byl vybrán blockchain Ethereum a jeho standard ERC-20 pro vývoj krypto tokenů. Chytrá smlouva bude napsána v programovacím jazyce Solidity ve vývojovém prostředí RemixIDE. Po úspěšné kompilaci bude smlouva nasazena na Ethereum testovací síti Goerli.

4.2.1 Standard ERC-20

Jak už bylo představeno v kapitole výše, pojem ERC-20 znamená Ethereum Request for Comment a číslo 20 je identifikačním číslem návrhu. ERC-20 byl navržen pro zlepšení sítě ETH a stal se jedním z nejvýznamnějších ERC standardů této doby. Standard obsahuje sadu pravidel, která musí dodržovat všechny tokeny

založené na Ethereum blockchainu; tedy i chystaný token v této práci. Chytré smlouvy pro tokeny nejsou odpovědné pouze za vytváření tokenů, ale také za zpracování transakcí a sledování zůstatku každého držitele tokenu. [23]
Tento standard definuje šest povinných funkcí, které by měla chytrá smlouva implementovat, a tři volitelné. Standard si zde lze představit jako rozhraní, které je nutné implementovat a s ním i šest zmíněných povinných metod. [23]
Mezi šest povinných metod náleží:

- totalSupply
  - Metoda, která definuje celkovou zásobu tokenů. Po dosažení tohoto limitu chytrá smlouva odmítne vytvořit nové tokeny.
- balanceOf
  - Metoda, která vrací počet tokenů, které má adresa peněženky.
- transfer
  - Metoda, která vezme určité množství tokenů z celkové zásoby a předá je uživateli.
- transferFrom
  - Další typ způsobu přenosu, který se používá k přenosu tokenů mezi uživateli.
- approve
  - Tato metoda ověřuje, zda je chytrá smlouva oprávněna přidělit určité množství tokenů uživateli s ohledem na celkovou zásobu.
- allowance
  - Tato metoda je přesně stejná jako metoda approve, kromě toho, že kontroluje, zda má jeden uživatel dostatečný zůstatek k odeslání určitého množství tokenů.

Nepovinné metody jsou:

- name
  - Navrací jméno tokenu.
- symbol
  - Navrací symbol tokenu.

- decimals
    - Navrací počet desetinných míst, která token používá.

4.2.2 Návrh, realizace a nasazení

V této části práce bude vytvořen krypto token. Ponese název CreditToken se symbolickým označením CRE. Jeho maximální množství bude stanoveno na 1 milion jednotek s přesností na 2 desetinná místa. V předešlé kapitole bylo stanoveno, že standard ERC-20 využívá rozhraní (interface) s povinnými metodami.

4.2.2.1 Realizace

K vývoji bylo zvoleno prostředí RemixIDE dostupné ve webové verzi na stránce https://remix.ethereum.org/. Pro samotný vývoj není potřeba nic dalšího instalovat, importovat nebo nastavovat. Stačí pouze vytvořit nový soubor, který bude automaticky založen s formátem sol[11].

Důvěru v chytré smlouvy lze lépe vybudovat, pokud je k dispozici zdrojový kód. Vzhledem k tomu, že zpřístupnění zdrojového kódu se vždy dotýká právních problémů s ohledem na autorská práva, kompilátor Solidity proto doporučuje použití strojově čitelných identifikátorů licence SPDX. Každý zdrojový soubor by proto měl začínat komentářem označujícím jeho licenci. [24] Chytrá smlouva v této práci tomu nebude výjimkou, a proto bude doplněna o MIT licenci jednoduchým komentářem níže.

// SPDX-License-Identifier: MIT

Druhým doplňkem každé chytré smlouvy je její pragma verze, která slouží k povolení určitých funkcí nebo kontrol kompilátoru, které by mohly zavést nekompatibilní změny. Tato anotace taktéž není povinná, ale je silně doporučena. Pro potřeby této práce a vzhledem k aktuálně dostupné verzi kompilátoru bude smlouva označena následnou anotací pro verzi 0.8.18.

pragma solidity ^0.8.18;

[11] Formát souboru v jazyce Solidity [24].

Pro standard ERC-20, jak bylo již zmíněno v předchozí kapitole, je nutné implementovat rozhraní s předepsanými povinnými metodami. Pro následující operace s tokenem byly přidány ještě dva eventy. První z nich event **transfer**, který musí být spuštěn kdykoli, kdy dojde k přesunu tokenů v rámci chytré smlouvy, včetně transakcí s nulovým počtem tokenů. Druhý event **approval** musí být spuštěn po každém úspěšném zavolání metody **approve**. Obrázek č. 14 níže zobrazuje dosavadní zápis chytré smlouvy.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.18;

// interface
interface IERC20 {
    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(
        address indexed owner,
        address indexed spender,
        uint256 value
    );

    function totalSupply() external view returns (uint256);  - gas

    function balanceOf(address account) external view returns (uint256);  - gas

    function transfer(address to, uint256 value) external returns (bool);  - gas

    function allowance(address owner, address spender)  - gas
        external
        view
        returns (uint256);

    function approve(address spender, uint256 value) external returns (bool);  - gas

    function transferFrom(  - gas
        address from,
        address to,
        uint256 value
    ) external returns (bool);
}
```

Obrázek 14 Chytrá smlouva – interface, Zdroj: Autor

V dalším kroku došlo k vytvoření třídy CreditToken, která implementuje již vytvořené rozhraní IERC20. Prvním důležitým krokem je tzv. namapování proměnné k ukládání párů hodnot key:value. Druhým krokem bylo implementovat nepovinné metody ze standardu ERC-20, přesto velice důležité pro další část práce. Jedná se o metodu **name**, pro kterou nastavíme hodnotu „CreditToken“, dále

symbol, který zvolíme „CRE“, a v neposlední řadě metodu **decimals**, která stanovuje počet desetinných míst. Jak bylo vysvětleno, jazyk Solidity neumí pracovat s desetinným místem, namísto toho připojí počet desetinných míst na konec základního celého čísla. V tomto případě chceme dosáhnout maximální rezervy podle stanoveného plánu na 1 milion jednotek, což lze zapsat explicitně jako 100 milionů, nebo může dojít ke zvolení počtu **finální počet + (znak 0 * počet desetinných míst)**. Celý vzoreček lze aplikovat pomocí zápisu níže, kdy dojde prakticky k umocnění čísla 10 počtem desetinných míst, a to celé vynásobené finálním číslem, viz obrázek č. 15 níže. V další části je implementován konstruktor, který je volán pouze jednou, a to při nasazení chytré smlouvy na blockchain. Konstruktor obsahuje metodu, která přiřadí prakticky veškerou kapacitu nově iniciovaných tokenů na adresu peněženky tvůrce, respektive na adresu, ze které bude smlouva nasazena do sítě. Následuje implementace jednotlivých povinných metod standardu z rozhraní, viz obrázek č. 15.

```
contract CreditToken is IERC20 {
    mapping(address => uint256) public balanceOf;
    mapping(address => mapping(address => uint256)) public allowance;
    string public name = "CreditToken";
    string public symbol = "CRE";
    uint8 public decimals = 2;
    uint256 public totalSupply = 1000000 * (10**uint256(decimals));

    constructor () {    infinite gas 716000 gas
        balanceOf[msg.sender] = totalSupply;
        emit Transfer(address(0), msg.sender, totalSupply);
    }

    function transfer(address recipient, uint256 amount)    infinite gas
        external
        returns (bool)
    {
        require(balanceOf[msg.sender] >= amount);
        balanceOf[msg.sender] -= amount;
        balanceOf[recipient] += amount;
        emit Transfer(msg.sender, recipient, amount);
        return true;
    }

    function approve(address spender, uint256 amount) external returns (bool) {    infinite gas
        allowance[msg.sender][spender] = amount;
        emit Approval(msg.sender, spender, amount);
        return true;
    }

    function transferFrom(    infinite gas
        address sender,
        address recipient,
        uint256 amount
    ) external returns (bool) {
        require(balanceOf[msg.sender] >= amount);
        allowance[sender][msg.sender] -= amount;
        balanceOf[sender] -= amount;
        balanceOf[recipient] += amount;
        emit Transfer(sender, recipient, amount);
        return true;
    }

    function mint(uint256 amount) external {    infinite gas
        balanceOf[msg.sender] += amount;
        totalSupply += amount;
        emit Transfer(address(0), msg.sender, amount);
    }

    function burn(uint256 amount) external {    infinite gas
        balanceOf[msg.sender] -= amount;
        totalSupply -= amount;
        emit Transfer(msg.sender, address(0), amount);
    }
}
```

Obrázek 15 Chytrá smlouva, Zdroj: Autor

Počet tokenů je pevně stanoven a dle pravidel smlouvy zůstává dále neměnný. Jedná se o finální počet 1 milion jednotek, který byl iniciován při nasazení smlouvy a jednorázově odeslán na peněženku tvůrce. Pro potřeby této práce byla chytrá

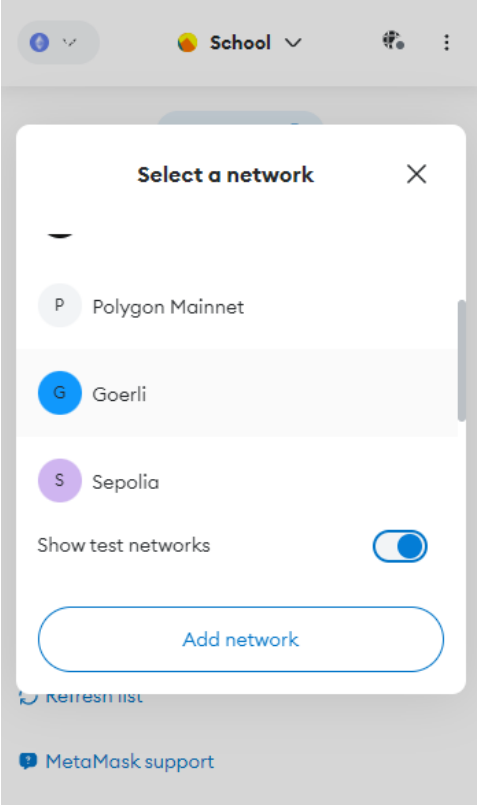
smlouva ještě obohacena o dvě metody, a to **mint** a **burn**. Jak již název napovídá, jedná se o metody, které jsou schopné upravit celkovou kapacitu tokenu. V případě **mint** lze tedy zavolat tuto metodu a provést tzv. vyražení dalších tokenů, a tak navýšit maximální kapacity, nebo naopak v případě metody **burn** může dojít ke spálení tokenů, tedy snížení maximální kapacity. Metoda mint po vytvoření nových tokenů zaručí jejich odeslání na cílovou adresu peněženky. Metoda burn předem stanovený počet spálí, respektive odebere z peněženky, z které je metoda volána.

Remix IDE nabízí přímou kompilaci chytré smlouvy v okně prohlížeče. Po dokončení kódu a jeho úspěšného zkompilování je smlouva připravena k nasazení na blockchain.

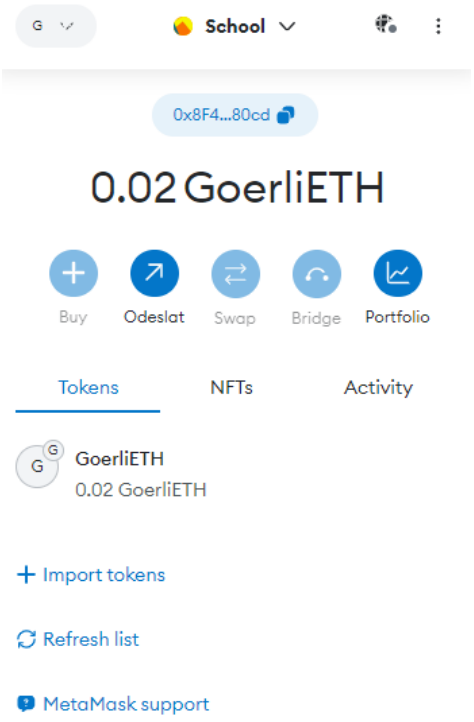
4.2.2.2 Nasazení

Pro možnost nasazení byla vybrána kryptoměnová peněženka MetaMask, která funguje jako rozšíření pro prohlížeče. MetaMask se totiž řadí k momentálně nejpoužívanější webové peněžence a zároveň je kompatibilní s řadou prohlížečů. Po jednoduché instalaci a vytvoření nové peněženky (viz Kapitola 3.3.6) je i peněženka připravena k nasazení nově vytvořené chytré smlouvy.

Blockchainová síť Ethereum disponuje v současné době třemi testovacími sítěmi. Jedná se o síť Goerli, Sepolia a Rinkeby. Tyto sítě se od sebe nijak zásadně neliší a pro účely této práce bude tedy použita např. síť Goerli. Stejně tak, jako každá jiná interakce s blockchainem, a to i nasazování smluv je prakticky transakcí, která vyžaduje nějaký poplatek. Poplatek je placen v kryptoměně daného blockchainu. V případě testovací sítě Goerli se jedná o kryptoměnu Goerli ETH se zkratkou GETH. Tuto kryptoměnu si lze bezplatně vyžádat na některé z distribučních stránek pro Goerli faucet, která uvolňuje jednou za 24 hodin malou část kryptoměny pro účely testování. Je potřeba zdůraznit, že za účelem omezení případného zneužití je v současnosti nutné splnit podmínku o vlastnictví alespoň 0.001 ETH v peněžence, na kterou se žádá o GETH. V případě splnění podmínky je možné na adrese https://goerlifaucet.com/ požádat o kryptoměnu a obvykle do několika vteřin ji obdržet (záleží na aktuálním vytížení sítě).

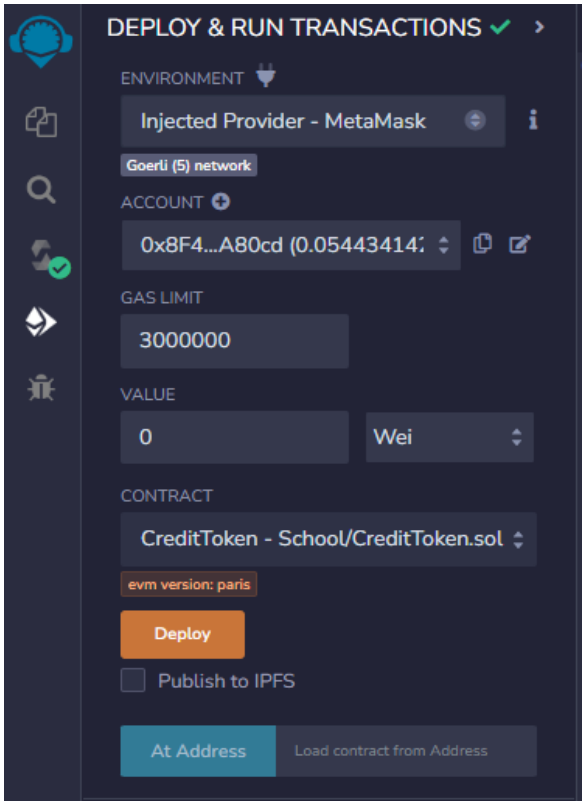


Obrázek 16 MetaMask – výběr sítě, Zdroj: Autor

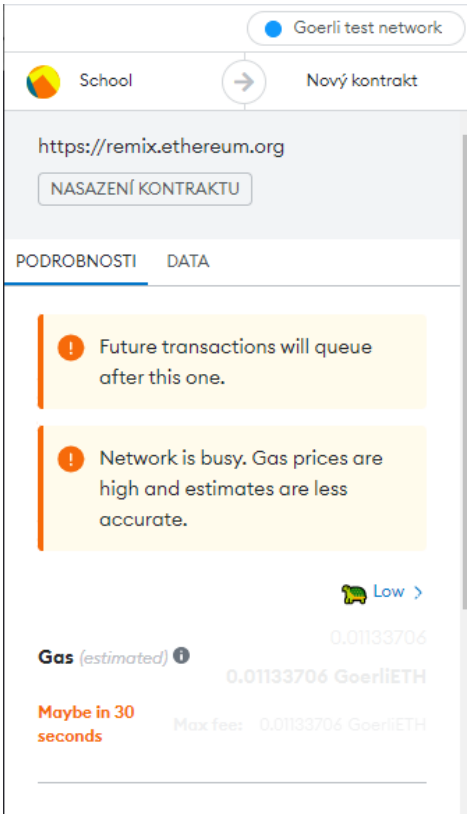


Obrázek 17 MetaMask – testovací síť, Zdroj: Autor

Dalším krokem byla nutnost přepnutí peněženky do správné sítě. V levém horním rohu po rozkliknutí nabídky bylo nutné povolit zobrazení testovacích sítí. Následovalo vyhledání námi zvolené Ethereum testovací sítě Goerli a proběhl její výběr (viz obrázek č. 16). Po přepnutí sítě se automaticky zobrazí její nativní kryptoměna a její zůstatek (viz obrázek č. 17).

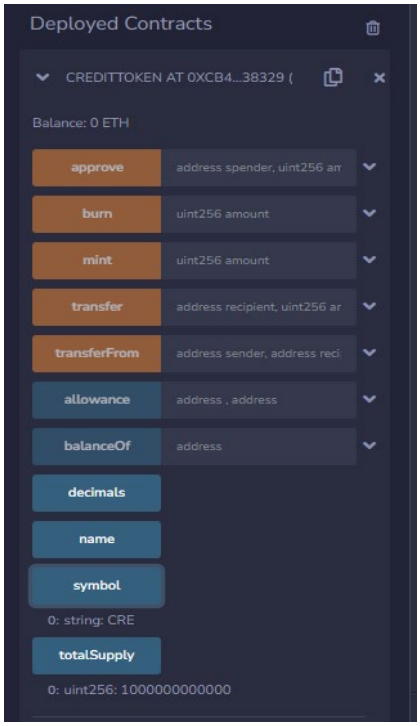


Obrázek 18 Remix IDE – nasazení smlouvy, Zdroj: Autor



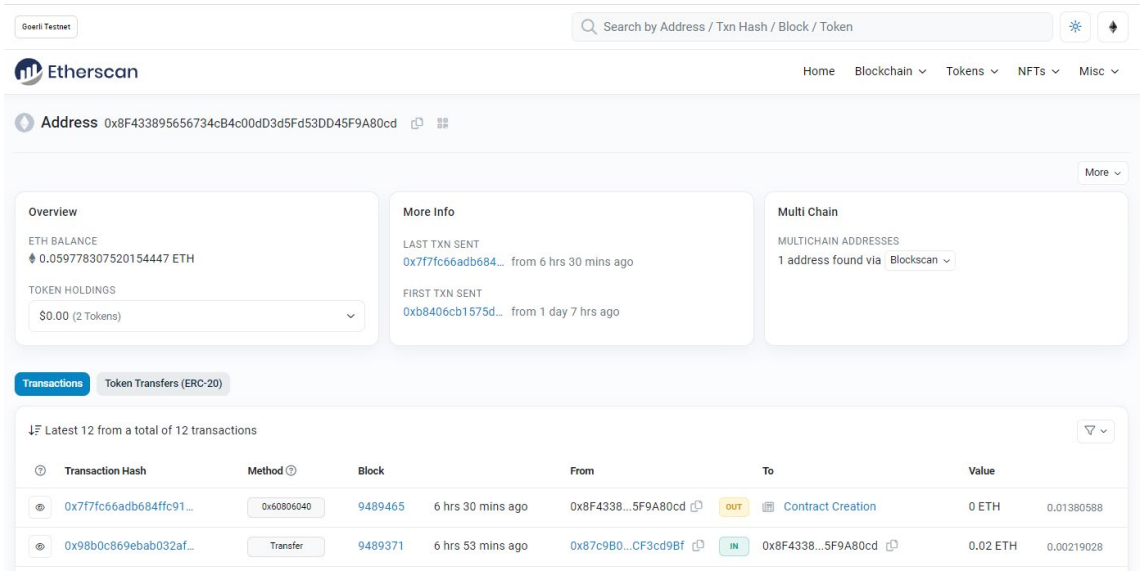
Obrázek 19 MetaMask – potvrzení nasazování smlouvy, Zdroj: Autor

Remix IDE umožňuje přímé injektování poskytovatele peněženek ve webovém prohlížeči. Proto proběhlo ověření skrze MetaMask a připojení peněženky přímo do vývojového prostředí, odkud došlo k automatické detekci vybrané sítě – Goerli – a načtení všech autorizovaných účtů s jejich aktuálním stavem GETH (viz obrázek č. 18). Odtud bylo možné chytrou smlouvu přímo nasadit na blockchain. Nasazení vyžadovalo potvrzení přes MetaMask peněženku a zaplacení poplatku za transakci (viz obrázek č. 19).



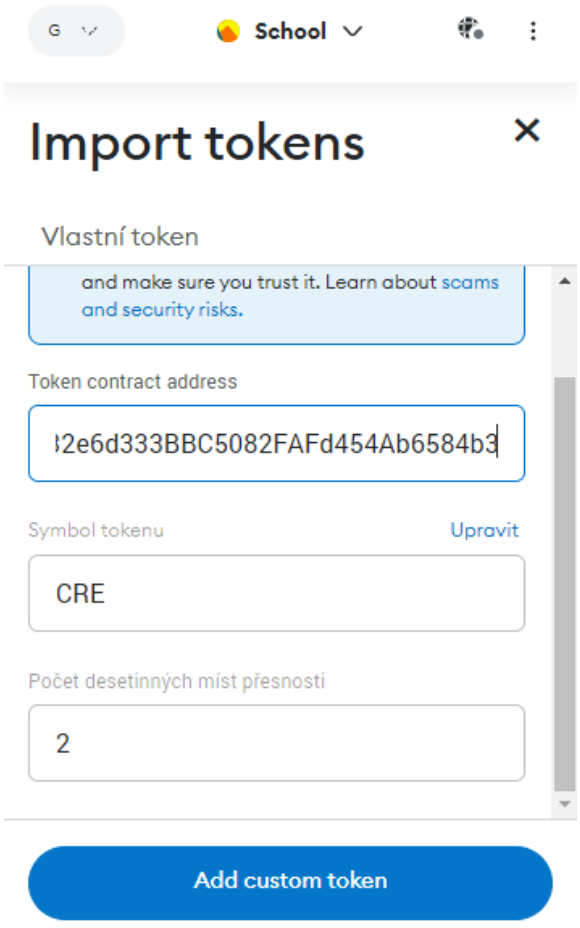
Obrázek 20 Remix IDE – ukázka volání metod nasazené smlouvy v rámci IDE, Zdroj: Autor

Jednou z výhod vývojového prostředí Remix IDE je, že dovoluje přímé ovládání, respektive dotazování na nově vytvořenou a nasazenou smlouvu a její metody, přímo ve webovém rozhraní. Na obrázku č. 20 můžeme vidět zavolání metody **symbol** a **totalSupply** a jejich návratové hodnoty.

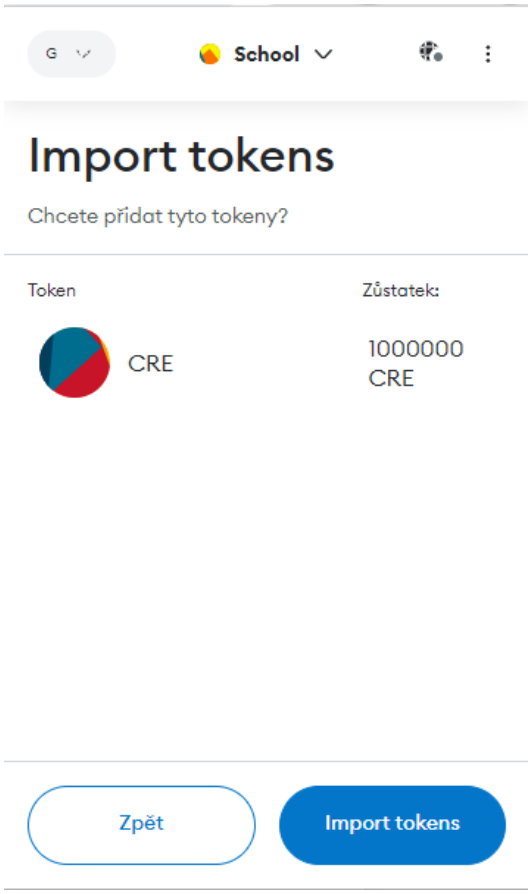


Obrázek 21 Etherscan, Zdroj: [83]

Goerli testovací síť poskytuje výborný nástroj pro sledování aktivity blockchainu. Tento nástroj je volně dostupný na adrese https://goerli.etherscan.io/. Pro vyhledání právě nasazené chytré smlouvy lze zvolit dvě cesty. První, při které můžeme zkopírovat adresu smlouvy přímo v RemixIDE po úspěšném nasazení smlouvy do sítě nebo pak přes vyhledání skrze adresu peněženky, ze které byla smlouva na blockchain nasazena, jak můžeme vidět na obrázku č. 21. Zde můžeme vidět označenou transakci s atributem „Contract Creation“, tedy vytváření smlouvy. Po rozkliknutí tohoto pole jsme přesměrováni do přehledu s adresou námi vytvořené chytré smlouvy. Zde budeme potřebovat zkopírovat danou adresu.



Obrázek 22 MetaMask – importování tokenu, Zdroj: Autor



Obrázek 23 MetaMask – kontrola a potvrzení přidání tokenu, Zdroj: Autor

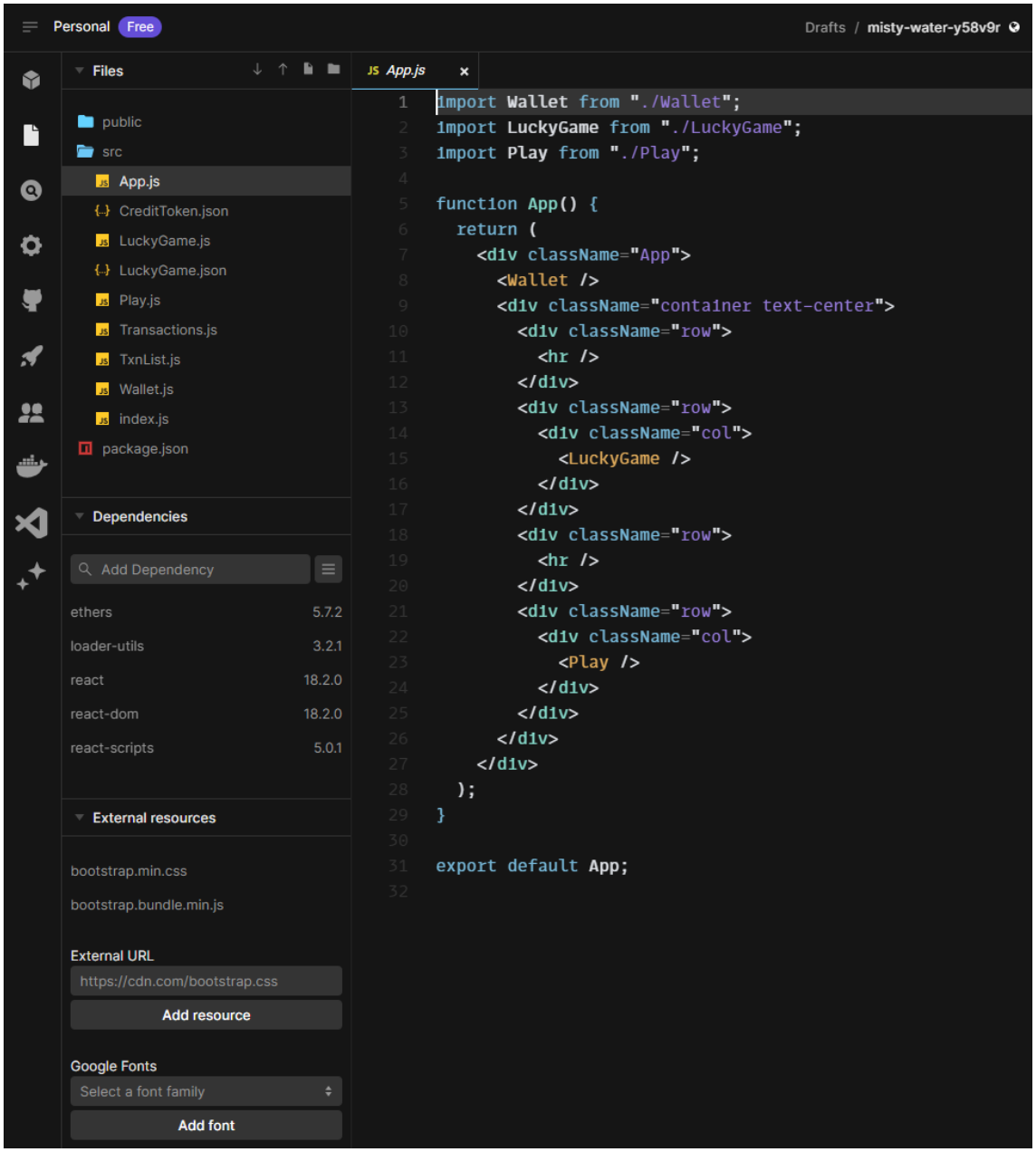
Přepneme se zpátky do MetaMask rozšíření a zvolíme možnost přidání tokenů (import tokens). Následně budeme vyzváni peněženkou k vložení adresy chytré smlouvy s tokenem (Token contract address), viz obrázek č. 22. Peněženka zbylé informace o symbolu a počtu desetinných míst doplní sama, proto stačí pokračovat na další krok, kde budeme vyzvaní k potvrzení a importaci tokenu přímo do vlastní peněženky. Na obrázku č. 23 si můžeme všimnout, že nám peněženka u daného tokenu již zobrazuje zůstatek – to je z důvodu provedení volání konstruktoru při nasazování smlouvy do sítě, kde byl implementován příkaz pro vytvoření maximálního množství tokenu a převedení do peněženky iniciátora.

4.3 Webová aplikace

Vývoj a nasazení vlastního krypto tokenu je takto finalizován. Nově vytvořené tokeny byly úspěšně převedeny do peněženky vývojáře, takže je možné

s nimi nyní dále pracovat. Druhá část této diplomové práce je zaměřena na integraci tokenu ve webové aplikaci a jeho následná obsluha.

Cílem bylo vytvořit webovou aplikaci umožňující uživateli přihlášení do dashboardu skrze jeho peněženku s načtením jeho krypto tokenů a jejich následné operace nad nimi. Pro vývoj aplikace byl zvolen jazyk React.js a jako peněženka MetaMask. Aplikace byla vyvinuta na platformě https://codesandbox.io/ (viz obrázek č. 24) umožňující vývoj v online prostředí napříč různými jazyky a frameworky bez nutnosti jakékoliv další instalace.



Obrázek 24 CodeSandBox přehled, Zdroj: Autor

Kromě React.js byla na projektu použita JavaScriptová knihovna Ethers.js, která si klade za cíl být kompletní a kompaktní knihovnou pro interakci s Ethereum blockchainem a jeho ekosystémem. Často se používá k vytváření decentralizovaných aplikací, peněženek, dalších nástrojů a jednoduchých skriptů, které vyžadují čtení a zápis do blockchainu. [25]
Pro frontend byl poté použit Bootstrap framework.

4.3.1 Integrace kryptoměnové peněženky

Uživatelé budou potřebovat způsob, jak spravovat a interagovat s tokenem. Řešení spočívá v implementaci vlastní kryptoměnové peněženky, nebo použití peněženky externí. Vytvoření vlastní peněženky není součástí této práce, tudíž byla zvolena druhá možnost s použitím externí peněženky. Jelikož se jedná o webovou aplikaci, bylo nutné zvolit peněženku spolupracující s webovým prohlížečem. Z možných vyhovujících (viz Kapitola 3.3.6) byla vybrána peněženka MetaMask.
Pro správné používání MetaMask peněženky v rámci této aplikace bylo nutné přepnout blockchain síť na Ethereum Goerli testovací síť a přidat námi vytvořený token s adresou smlouvy: „0x9BFe44B38d32e6d333BBC5082FAFd454Ab6584b3“, obdobně jak bylo provedeno v předešlé kapitole o nasazování chytré smlouvy na blockchain.
Veškerá komunikace mezi peněženkou, webovou aplikací a chytrými smlouvami (blockchainem) musí probíhat za pomoci speciální knihoven umožňujících obsluhu blockchainu. Mezi takové knihovny náleží např. Web3.js, Ipfs-core, Bitcoinjs-lib, Coingecko-api, Ethers.js nebo Truffle. [26] V rámci této práce byla vybrána knihovna Ethers.js, a to z důvodu její schopnosti zpracovávat transakce a výpočty jednodušeji a rychleji. Knihovna je zaměřena přímo na Ethereum blockchain a jeho ekosystémy, což z ní dělá vhodnou volbu. [27]

4.3.2 Implementace chytrých smluv

Pro implementaci chytrých smluv přímo v aplikaci bylo nutné získat jejich smluvní aplikační binární rozhraní (Application Binary Interface), což je soubor pravidel a specifikací, které definují, jaké typy dat mohou být přijímány a odesílány skrze funkce chytré smlouvy, jak jsou tyto data zakódována a jak jsou vrácené

výsledky dekódovány zpět pro externí volající strany. Tato pravidla je nutné respektovat. V podstatě jde o rozhraní mezi smlouvou a vnějším světem. ABI zajistí, že volání funkcí smlouvy a komunikace s ní jsou prováděny správným způsobem a že data jsou korektně interpretována. [28] Pro práci s ABI lze využít např. výše zmíněnou knihovnu Ethers.js.

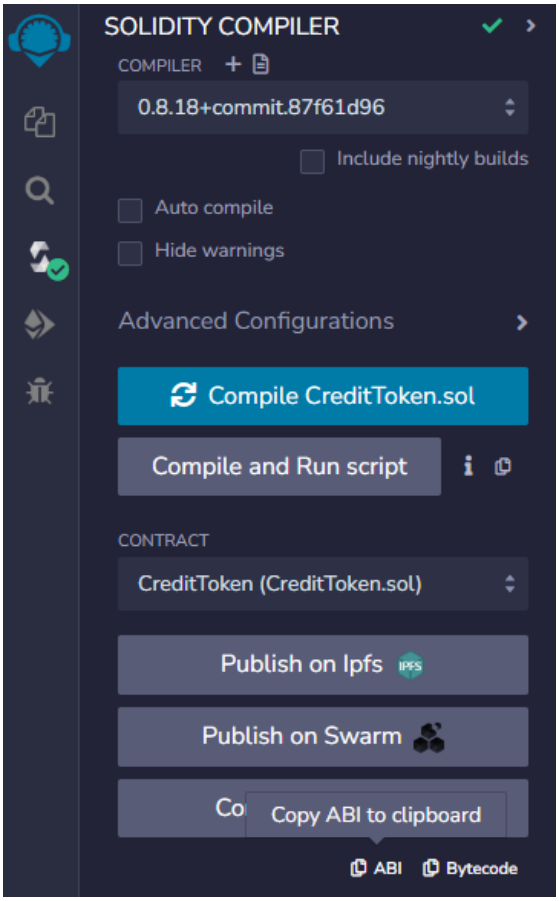
```
1  // SPDX-License-Identifier: UNLICENSED
2  pragma solidity ^0.8.18;
3
4  contract LuckyNumber {
5      uint256 secretNumber;
6
7      constructor() payable {    112431 gas 90200 gas
8          secretNumber = 9;
9      }
10
11     event LotteryEvent(bool isWinner, address indexed player);
12
13     function guessNumber(uint256 _number) public payable {    4074 gas
14         if (_number != secretNumber) {
15             emit LotteryEvent(false, msg.sender);
16         } else {
17             emit LotteryEvent(true, msg.sender);
18         }
19     }
20 }
21
```

Obrázek 25 Chytrá smlouva minihry, Zdroj: Autor

Na obrázku č. 25 můžeme vidět chytrou smlouvu. Na obrázku č. 26 poté její aplikační binární rozhraní, které přesně reflektuje její metody s parametry a návratové hodnoty.

```
[
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "_number",
        "type": "uint256"
      }
    ],
    "name": "guessNumber",
    "outputs": [],
    "stateMutability": "payable",
    "type": "function"
  },
  {
    "inputs": [],
    "stateMutability": "payable",
    "type": "constructor"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "internalType": "bool",
        "name": "isWinner",
        "type": "bool"
      },
      {
        "indexed": true,
        "internalType": "address",
        "name": "player",
        "type": "address"
      }
    ],
    "name": "LotteryEvent",
    "type": "event"
  }
]
```

Obrázek 26 ABI soubor, Zdroj: Autor



Obrázek 27 Remix IDE – Solidity compiler, Zdroj: Autor

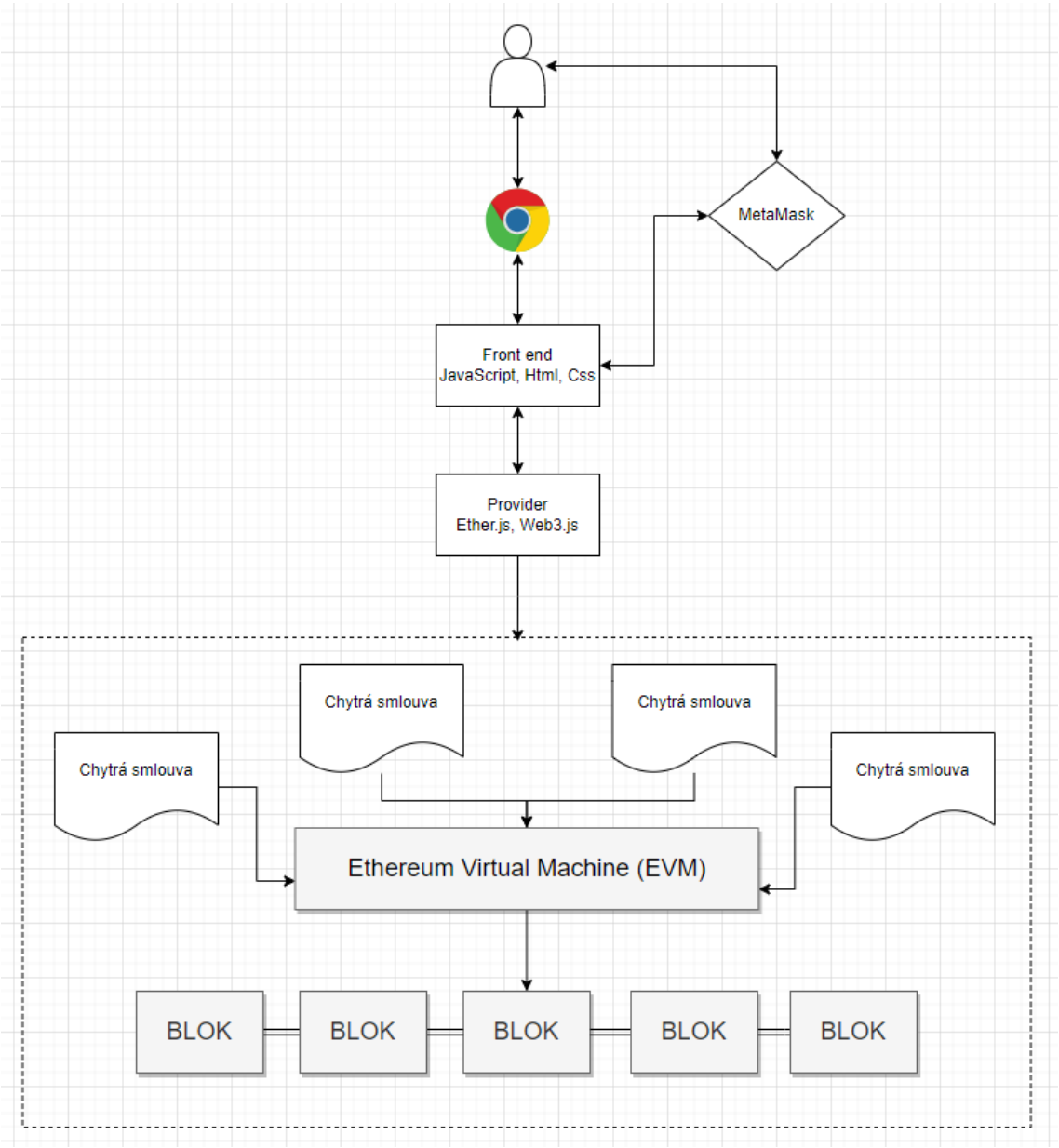
Pro získání smluvního aplikačního rozhraní v případě CreditTokenu bylo nutné otevřít Solidity kompilátor ve vývojovém prostředí Remix IDE a po úspěšné kompilaci, kdy se „objevila“ možnost zkopírovat ABI výstup, zde kód vyzvednout, viz obrázek č. 27. Takto získaný soubor byl poté uložen ve formátu JSON ve stejném adresáři jako soubory z webové aplikace.

4.3.3 Webová aplikace

V tomto projektu byl pro frontendovou uživatelskou část aplikace použit React.js.

React.js je open-source JavaScriptová knihovna, která se používá k vytváření uživatelských rozhraní (UI) pro webové aplikace. Byla vyvinuta společností Facebook a je široce používána vývojáři k vytváření dynamických a interaktivních webových stránek. [113]

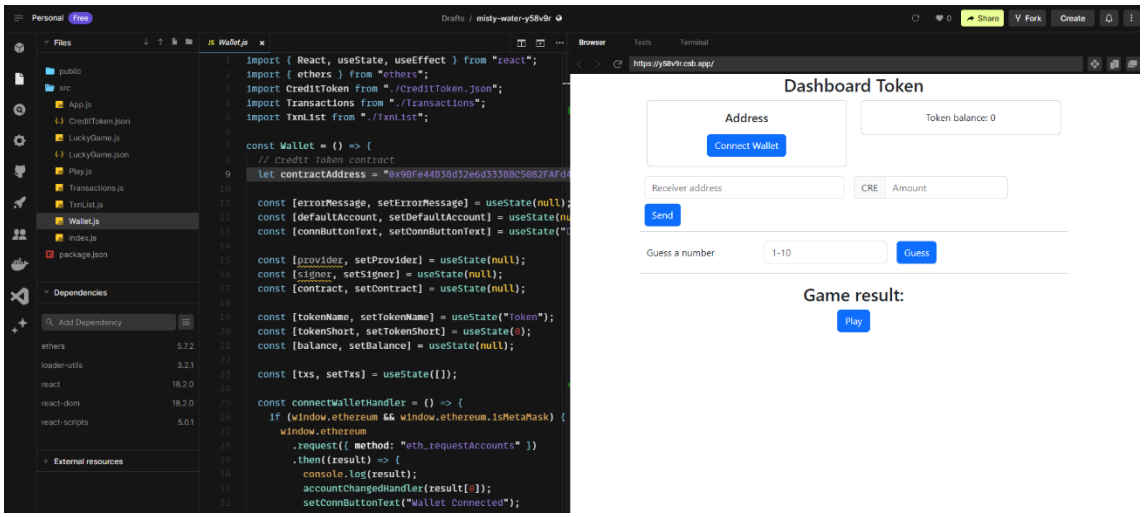
Také byl použit CSS open-source Framework BootStrap.



Obrázek 28 Vizuální návrh webové aplikace, Zdroj: Autor

Na obrázku č. 28, vidíme vizuální návrh celé aplikace. Uživatel obsluhuje internetový prohlížeč s rozšířením MetaMask, kde oba komunikují s frontendem. Samotný frontend poté pomocí knihovny Ethers.js komunikuje přímo s blockchainem Ethereum, na kterém jsou umístěny chytré smlouvy.

Po implementaci veškerého kódu aplikace i samotných chytrých smluv, je nutné představit blíže fungování výsledné aplikace. Uživatel je nejdříve nucen provést přihlášení přes MetaMask peněženku, a to skrze tlačítko pro připojení v aplikaci, viz obrázek č. 29.



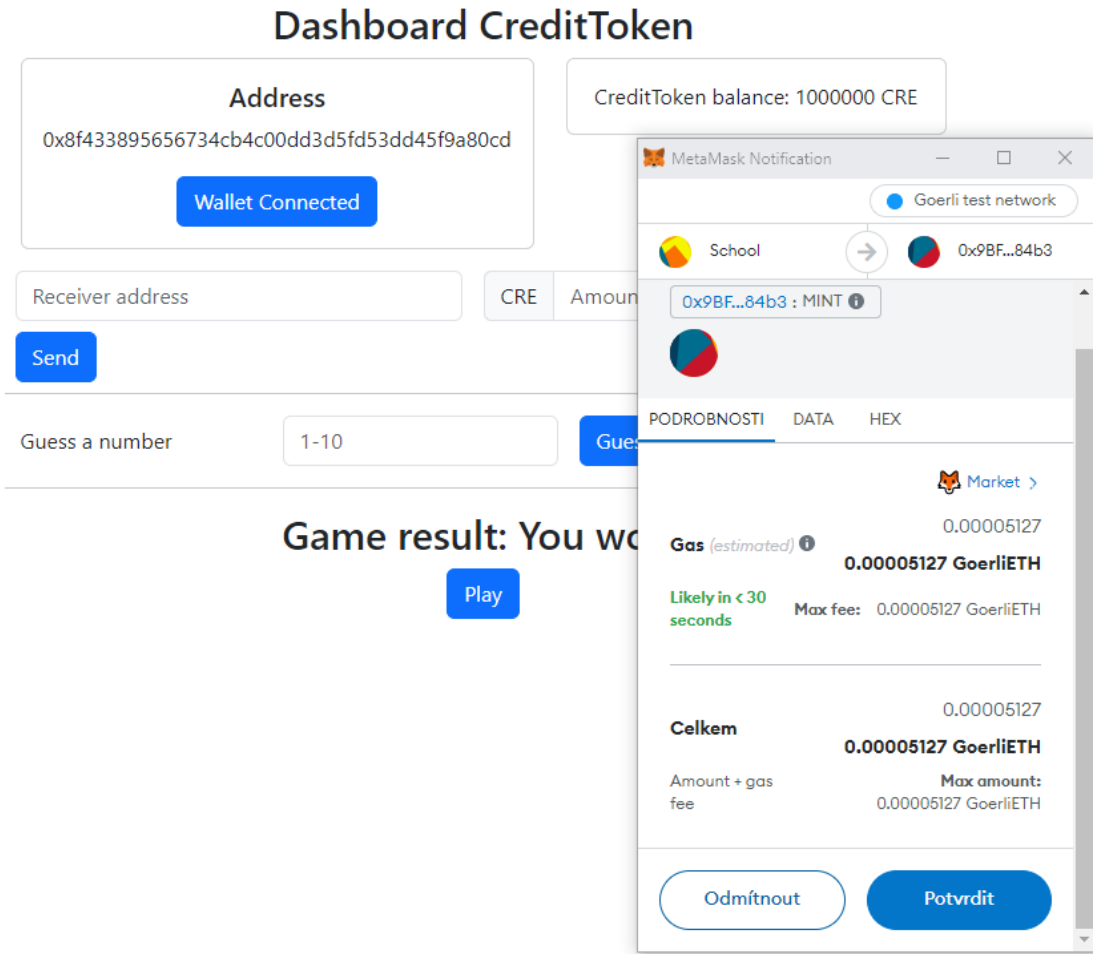
Obrázek 29 CodeSandBox – přehled, Zdroj: Autor

Po přihlášení je uživateli zobrazena jeho adresa a je načten obsah jeho peněženky, a to pouze pro CreditToken, u kterého je zobrazeno jeho množství, jež uživatel vlastní.
V aplikaci jsou implementovány tři různé způsoby operace s tokenem.

- Hra
- Převod tokenů mezi uživateli
- Hádání čísla

Prvním příkladem je hra, kterou si lze představit jako házení mincí. Uživateli jsou vráceny hodnoty pravda a nepravda. V případě nepravdy, respektive prohry, se nic neděje. V případě výhry je naopak uživatel odměněn tokenem, a to formou zavolání metody mint na chytré smlouvě pro Credit Token (viz předchozí kapitola). Rozhodovací mechanismus je řešen na straně klienta, až získání odměny je řešeno komunikací s blockchainem, kdy je nutné potvrdit transakci do sítě pro úplné získání odměny. Po zpracování je uživateli odměna přidána obvykle během

několika vteřin. Výsledek výhry a nové okno sloužící k potvrzení získání nových tokenů můžeme vidět na obrázku č. 30.

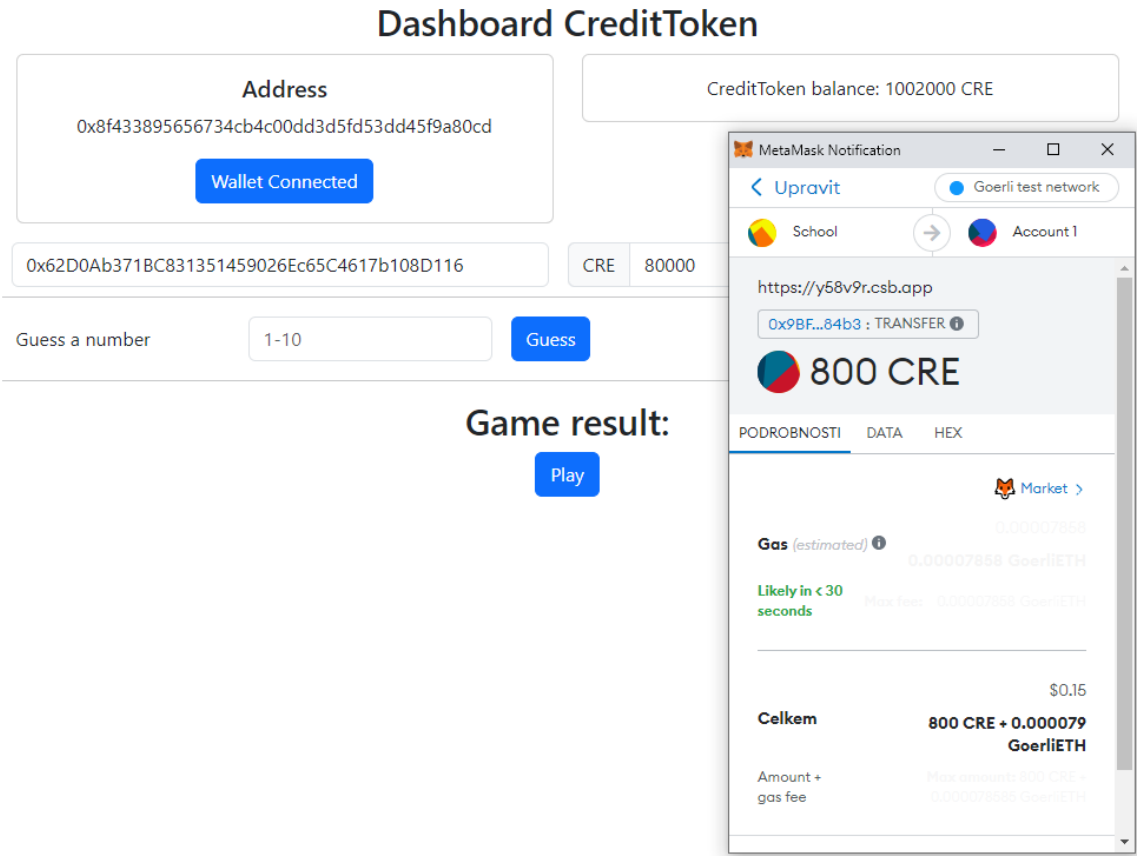


Obrázek 30 Webová aplikace – potvrzení připsání odměny, Zdroj: Autor

Uživatel je po výhře odměněn fixní částkou 1 000 tokenů. Tyto tokeny jsou vytvořeny nad rámec maximální kapacity a přesně o tuto částku je kapacita i navýšena.

Druhým příkladem je převod tokenů mezi uživateli. Tato funkce je běžně součástí snad všech peněženek, MetaMask patří mezi ně, nicméně pro potřeby aplikace je tato funkce řešena přímo v dashboardu. Uživatel nejprve vloží adresu peněženky rozdílnou od té své, poté zvolí množství tokenů, které chce odeslat. Je důležité odesílat menší množství, než vlastní, jinak nastane chyba a transakce se neprovede. Při zápisu částky je nutné přidat dvě desetinná čísla na konec reálného

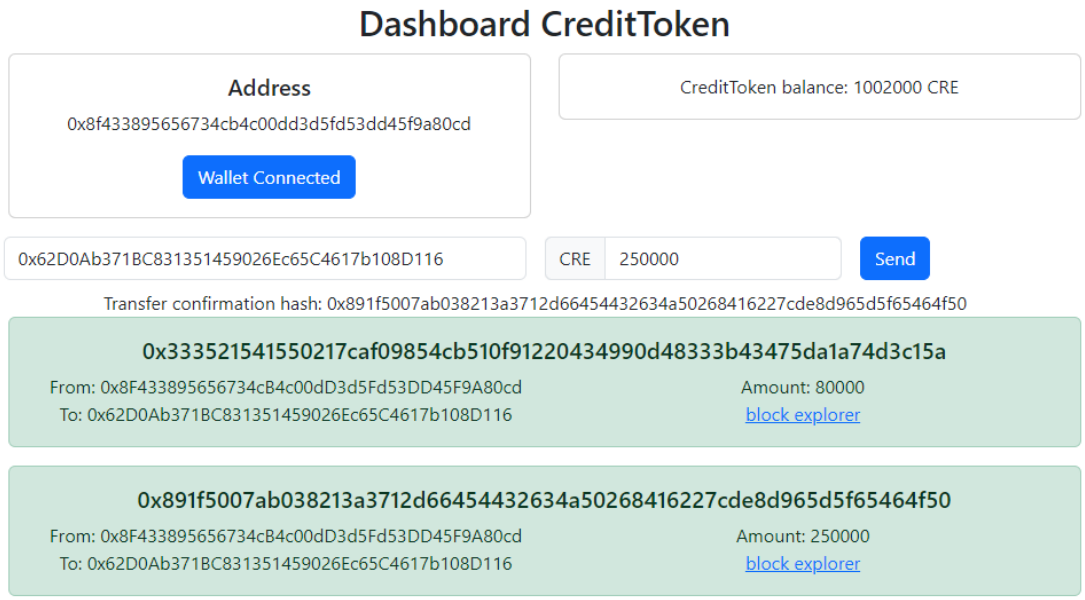
čísla. Tudíž při odeslání 80 000 tokenů se jedná reálně o počet 800, což i zobrazí požadavek o podpis skrze MetaMask rozšíření, viz obrázek č. 31.



Obrázek 31 Webová aplikace – potvrzení přesunu tokenů, Zdroj: Autor

Potvrzená a odeslaná transakce vrací tzv. hash, což je unikátní identifikátor transakce, který je zobrazen v aplikaci. Zároveň chytrá smlouva vyvolá předem definovanou událost **Transfer**, která se váže právě k transakci mezi dvěma peněženkami. Webová aplikace obsahuje funkci, která neustále naslouchá konkrétně těmto událostem pomocí knihovny Ethers.js. Tato funkce okamžitě zachytává právě provedenou událost, kdy dojde k volání metody **Transfer,** a tento výpis zobrazuje přímo v aplikaci, viz vícero zpracovaných transakcí, které můžeme vidět na obrázku č. 32. Je důležité zmínit, že události probíhají na straně chytré smlouvy nehledě na využívání webové aplikace, tudíž kdyby došlo k provedení transakce mimo webovou aplikaci, např. převodu tokenů mezi adresami peněženek pomocí peněženek externích, aplikace zachytí tyto události a zobrazí je. Aplikace tedy zachytává veškeré proběhlé události typu Transfer a zobrazuje je

všem uživatelům. Tento fakt je uveden v práci z důvodu demonstrace funkčnosti fungování událostí chytré smlouvy. Pro produkční nasazení by bylo vhodné filtrovat pouze transakce, které se týkají pouze uživatelské adresy.



Obrázek 32 Webová aplikace – přehled transakcí, Zdroj: Autor

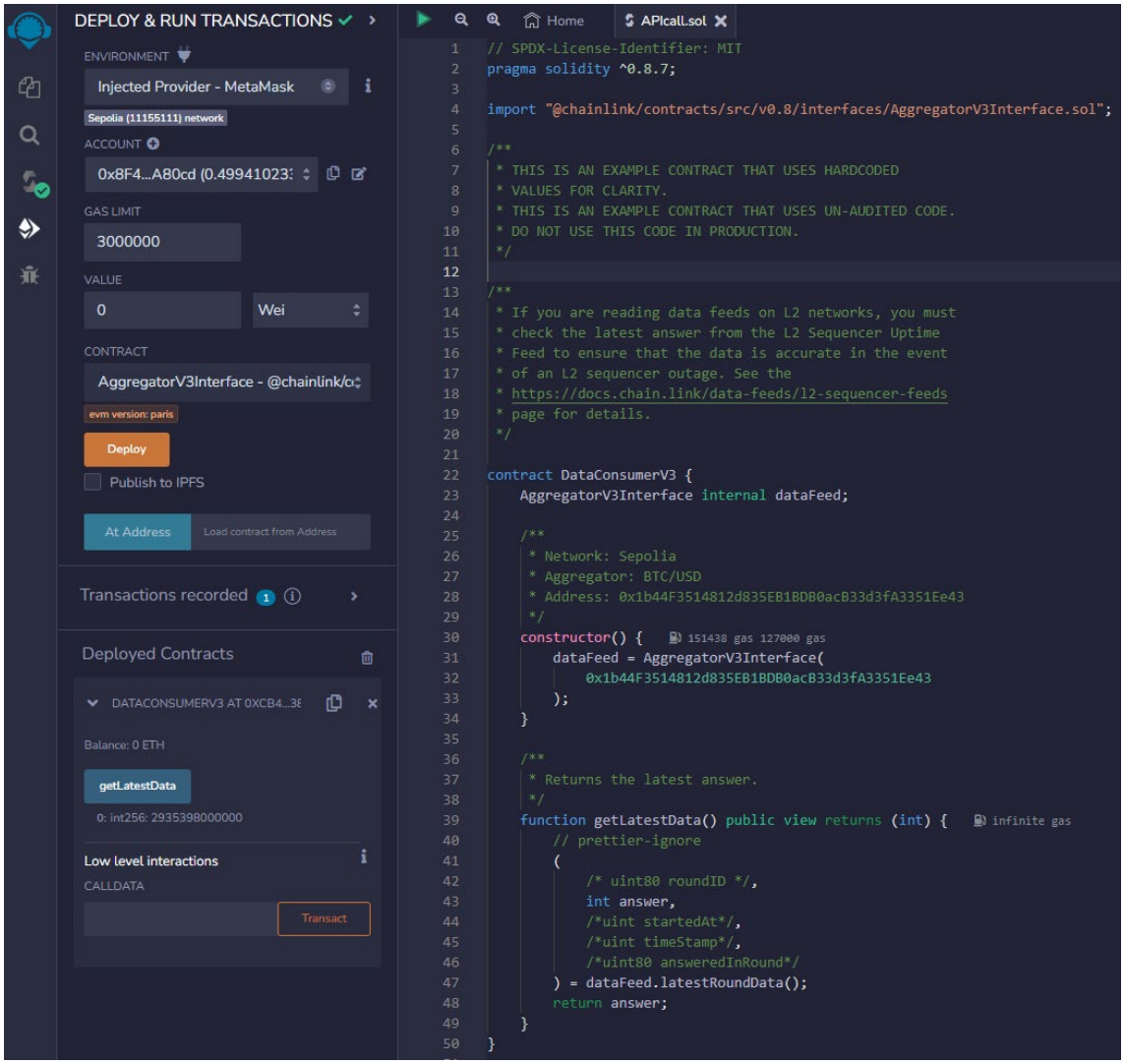
Třetím příkladem je hádání čísla. Implementace byla provedena skrze jinou chytrou smlouvu, než je CreditToken, která má předem stanovené číslo a jednu funkci, která je volána ze strany aplikace, kdy dochází k tzv. hádání výsledného čísla. Chytrá smlouva každý požadavek zpracuje a navrátí výsledek, který zavolá jako událost, zda bylo hádání správné, či nikoli. Implementace kódu můžeme vidět na obrázku č. 25. Aplikační část poté funguje obdobně jako v předchozím příkladu s událostí Transfer. Hádané číslo je porovnáno přímo v rámci chytré smlouvy, která navrátí výsledek pravda/nepravda jako událost. Webová aplikace neustále naslouchá této události a zobrazuje výsledek v rámci dashboardu.

4.4 Ostatní chytré smlouvy

Tato kapitola zodpovídá případné otázky, které by mohly vyvstat nad rámec zpracování této práce.

4.4.1 Oracle

V této podkapitole je předvedena práce s oracle, což lze charakterizovat jako zabezpečené externí API pro chytré smlouvy.



Obrázek 33 Remix IDE – chytrá smlouva s oracle, Zdroj: Autor

Na obrázku č. 33 můžeme vidět funkční aplikaci oracle (viz Kapitola 3.4.11) technologie v rámci chytré smlouvy. Oracle v prostředí chytrých smluv znamená

vnější zdroj informací nebo dat, který je použit chytrou smlouvou k získání aktuálních informací ze světa mimo blockchain. [80]

Na tomto příkladu je chytrá smlouva v prostředí Remix IDE nasazena na testovací síť Sepolia Ethereum a následně je zavolána její jediná metoda **getLatestData**, která navrací hodnotu párové ceny Bitcoinu vůči Americkému dolaru (jak můžeme vidět v levé dolní části obrázku). Jedná se o jeden z možných datových feedů společnosti Chainlink, která se specializuje na oracle služby pro blockchain. [114]

4.4.2 Bezpečnost

V rámci bezpečnosti vyvstávají dvě otázky. První z nich zní – proč lze integrovat pouze pomocí uživatelských peněženek?

Odpověď je jednoduchá. Decentralizované aplikace a aplikace spolupracující s blockchainem se pyšní hlavně decentralizací a tím, že na ně „nedohlíží" žádný kontrolní orgán, žádná autorita. To otevírá možnost plné svobody používání takovýchto aplikací. V případě dApp by měl být celý backend umístěný na blockchainu, veškerá data by se tudíž měla zapisovat (umisťovat) tam. V otázce dalších způsobů integrace, kdy uživatel nebude nucen používat peněženku, jako je např. MetaMask, by bylo nutné tuto funkcionalitu nahradit tím, že by se uživatelův soukromý (a i ten veřejný) klíč k jeho peněžence uložil někde v rámci aplikace a veškeré další akce by se automaticky podepisovaly za něj. Kromě toho, že by se musela aplikace „snížit" k používání některých autorizačních metod z Webu 2.0, by se jednalo o velkou bezpečností hrozbu pro samotného uživatele. Kdokoli by měl nebo získal přístup k jeho klíči, by získal okamžitou kontrolu nad jeho peněženkou a všemi digitálními měnami v ní umístěnými. Tím pádem by uživatel riskoval odcizení jeho obsahu peněženky, ztratil by anonymitu v rámci vystupování pouze pod adresou peněženky a ztratil by možnost rozhodování nad každou transakcí. Navíc by nebyl schopen kontrolovat poplatek za každou transakci, což by vedlo teoreticky ke zbytečným extra výdajům za ně.

Druhou otázkou je, jaké jsou další bezpečností hrozby?

Hlavní bezpečností hrozbou v otázce decentralizovaných aplikací a chytrých smluv jsou jejich chyby v implementaci. Podle studie na téma bezpečnosti chytrých kontraktů v rámci Ethereum síťě „Finding The Greedy, Prodigal, and Suicidal

Contracts at Scale“, kde byl testován téměř jeden milion chytrých smluv, bylo zjištěno, že cca 34 tisíc z nich stále obsahovalo bezpečností chyby, které by vedly k situacím, jako je nekontrolované vyražení tokenů nebo jejich spálení, případně úplné zničení smlouvy. [22]

5 Shrnutí výsledků

V předchozích kapitolách praktické části této práce došlo k analýze požadavků, a to zejména při výběru vhodného blockchainu a standardu pro psaní chytrých smluv, dále byl vybrán jazyk a prostředí pro vývoj.

Pro cíl splnění práce byl nakonec vybrán Ethereum blockchain, a to hlavně z hlediska jeho historického i současného podílu na existenci a fungování chytrých smluv a z důvodu, že je hlavní technologií pro vytváření krypto tokenů. V rámci blockchainu Ethereum došlo k výběru standardu ERC-20, který se jeví jako nejrozšířenější, a navíc z něj nespočet dalších standardů u jiných sítí vychází. Pro psaní samotné chytré smlouvy byl zvolen jazyk Solidity, který je s jasným předstihem nejrozšířenější a možná i nejvyspělejší v této oblasti. Za vývojové prostředí se stal jasným favoritem Remix IDE, který umožnil přímé nasazení smluv na blockchain.

Samotný vývoj tokenu splnil veškeré požadavky standardu, navíc byl rozšířen o dvě další metody. Chytrá smlouva s tokenem byla poté nasazena a v rámci Remix IDE také otestována na funkčnost jednotlivých metod. Poté došlo k přiřazení takto nově vzniklého tokenu do webové peněženky.

V další části byla poté vyvinuta webová aplikace, která umožňuje přihlášení skrze webovou peněženku a následné zobrazení tokenů. Aplikace je také schopna převádět tokeny mezi uživateli. Dále obsahuje dvě minihry – hádání čísla a házení mincí. Ve hře házení mincí je v případě úspěchu uživatel odměněn nově vyraženými tokeny přímo do jeho peněženky.

6 Závěry a doporučení

V rámci práce, která si dávala za úkol vytvořit vlastní krypto token a poté ho integrovat do webové aplikace, byly úvodem představeny teoretické informace světa blockchainové technologie, potažmo kryptoměn. Následně byl detailně

rozebrán Ethereum blockchain a jeho fungování pro účely pochopení implementace chytrých smluv. V další části byl popsán Web 3.0 a decentralizované aplikace.

V rámci praktické části práce došlo k analýze požadavků na vytvoření krypto tokenu, která v první řadě spočívala ve vybrání blockchainu. Bylo zjištěno, že není zcela možné určit něco jako „správný“ nebo „vhodný“ výběr sítě. Je potřeba zvážit několik otázek, u většiny z nichž může hrát podstatnou roli čistě subjektivní preference navrhovatele, mezi které patří např. subjektivní důvěra vůči danému blockchainu, jež může dosti záviset na tom, kde kdo co četl nebo zda nějaký blockchain v poslední době řešil nějaké bezpečností problémy. V tom se také nepřímo odráží důvěra uživatelské základny, která by měla případný zájem využívat nově vzniklý token v rámci ekosystému webové aplikace. Dalším klíčovým předpokladem je také otázka, kolik transakcí bude nutné provádět ze strany uživatele v rámci časového horizontu. V tom se také přímo odráží kapacita transakcí za vteřinu u jednotlivých blockchainů, která přináší různé vytížení sítě, což se může odrazit v ceně jednotlivé transakce. Ta se může v extrémních případech i několikasetnásobně zvýšit, a to je rozhodně případ, kterému by se chtěl každý vyhnout.

Přesto došlo k výběru blockchainové sítě Ethereum, kde byl aplikován standard ERC-20 pro tvorbu krypto tokenů v rámci chytrých smluv. Smlouva byla nasazena na testovací síť a tokeny byly přiřazeny do peněženky. V další části práce došlo k vytvoření webové aplikace, která umožňuje autorizaci uživatele skrze jeho kryptoměnovou peněženku, kde je uživatel nadále reprezentován přímo adresou peněženky. V rámci aplikace je uživateli umožněno tokeny přesouvat mezi dalšími peněženkami (uživateli) a také je získávat v rámci odměn za aktivitu (výhru) v rámci minihry, která je řešena vlastní chytrou smlouvou. Na závěr je ještě představena možnost používání oracle technologie, která určitě najde své uplatnění v rámci vývoje samostatné decentralizované aplikace.

Samotný stanovený cíl práce můžeme proto považovat za splněný.

V budoucnu by bylo prospěšné rozšířit aplikaci integrací chytrých smluv pro standard nezaměnitelných tokenů (NFT), které jsou v současnosti klíčovým prvkem v tomto oboru. Tím by aplikace získala další rozměr. Dále by bylo vhodné

zvážit přidání dalších funkcionalit, které by umožnily využití tokenů jako platidla například při nákupu NFT nebo pro odemykání různých částí aplikace.

Při tvorbě vlastních tokenů se může stát největším problémem technologie samotná, která vzhledem ke svým kapacitám může být pro některé projekty stále nedostatečná. Právě z tohoto důvodu například Ethereum provedlo přechod z důkazu práce (Proof of Work) na důkaz hodnoty (Proof of Stake) jako součást jednoho z několika kroků snahy o zlepšení celé technologie. Ethereum má za cíl dosáhnout kapacity až 100 tisíc transakcí za sekundu, což by znamenalo značné zvýšení v porovnání se současným průměrem sítě, který momentálně činí v průměru 10–15 transakcí za sekundu. Pokud se to podaří, tak by cena jednotlivých transakcí mohla být opravdu minimální, nehledě na to, že by tato kapacita měla stačit pro všechny současné i budoucí projekty v následujících letech. Co přinese budoucnost, zůstává otevřenou otázkou, na kterou však tato práce nemůže poskytnout odpověď.

7 Seznam použité literatury

[1] YANO, Makoto, Chris DAI, Kenichi MASUDA a Yoshio KISHIMOTO, ed. Blockchain and Crypto Currency [online]. Singapore: Springer Singapore, 2020 [cit. 2023-08-20]. Economics, Law, and Institutions in Asia Pacific. ISBN 978-981-15-3375-4. Dostupné z: doi:10.1007/978-981-15-3376-1

[2] What Is Blockchain Technology? [online]. [cit. 2023-08-20]. Dostupné z: https://aws.amazon.com/what-is/blockchain/

[3] Blockchain – o co se jedná a jak funguje (2. díl) [online]. [cit. 2023-08-20]. Dostupné z: https://www.kurzy.cz/zpravy/610340-blockchain--o-co-se-jedna-a-jak-funguje-2-dil/

[4] What Is Blockchain and How Does It Work? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-blockchain-and-how-does-it-work

[5] VESTERGAARD, Cindy, Haimanot Anbesaw BOBOSHA a Karolin LANGFELDT. Distributed Ledger Technology: Beyond the Hype. In: VESTERGAARD, Cindy, ed. Blockchain for International Security [online]. Cham: Springer International Publishing, 2021, s. 7-22 [cit. 2023-08-20]. Advanced Sciences and Technologies for Security Applications. ISBN 978-3-030-86239-8. Dostupné z: doi:10.1007/978-3-030-86240-4_2

[6] SUNYAEV, Ali a Ali SUNYAEV. Distributed Ledger Technology. In: Internet Computing [online]. Cham: Springer International Publishing, 2020, s. 265-299 [cit. 2023-08-20]. ISBN 978-3-030-34956-1. Dostupné z: doi:10.1007/978-3-030-34957-8_9

[7] DappRadar [online]. [cit. 2023-08-20]. Dostupné z: https://golden.com/wiki/DappRadar%EF%BB%BF-4NB9ZGE

[8] WaxBlock [online]. [cit. 2023-08-20]. Dostupné z: https://waxblock.io/

[9] Solana [online]. [cit. 2023-08-20]. Dostupné z: https://explorer.solana.com/

[10] Bscscan [online]. [cit. 2023-08-20]. Dostupné z: https://bscscan.com/

[11] Polygonscan [online]. [cit. 2023-08-20]. Dostupné z: https://polygonscan.com/

[12] Ethereum transactions per second chart [online]. [cit. 2023-08-20]. Dostupné z: https://blockchair.com/ethereum/charts/transactions-per-second

[13] Top 10 BNB Chain Projects in 2022 [online]. [cit. 2023-08-20]. Dostupné z: https://coinmarketcap.com/alexandria/article/top-10-binance-smart-chain-projects

[14] Top Blockchain Dapps [online]. [cit. 2023-08-20]. Dostupné z: https://dappradar.com/rankings?sort=uawCount&order=desc&range=24h

[15] Breakdown by Smart Contract Languages [online]. [cit. 2023-08-20]. Dostupné z: https://defillama.com/languages

[16] What is total value locked (TVL) in crypto and why does it matter? [online]. [cit. 2023-08-20]. Dostupné z: https://cointelegraph.com/explained/what-is-total-value-locked-tvl-in-crypto-and-why-does-it-matter

[17] Solidity Documentation [online]. [cit. 2023-08-20]. Dostupné z: https://docs.soliditylang.org/en/v0.8.21/

[18] Top 6 Smart Contract Languages in 2023 [online]. [cit. 2023-08-20]. Dostupné z: https://chain.link/education-hub/smart-contract-programming-languages

[19] Language Influences [online]. [cit. 2023-08-20]. Dostupné z: https://docs.soliditylang.org/en/v0.8.21/language-influences.html

[20] Vyper Documentation [online]. [cit. 2023-08-20]. Dostupné z: https://docs.vyperlang.org/en/stable/

[21] How to write an Ethereum smart contract using Vyper [online]. [cit. 2023-08-20]. Dostupné z: https://www.quicknode.com/guides/ethereum-development/smart-contracts/how-to-write-an-ethereum-smart-contract-using-vyper

[22] NIKOLIC, Ivica, Aashish KOLLURI, Ilya SERGEY, Prateek SAXENA a Aquinas HOBOR. Finding The Greedy, Prodigal, and Suicidal Contracts at Scale [online]. [cit. 2023-08-20]. Dostupné z: https://arxiv.org/pdf/1802.06038.pdf

[23] ERC-20: Token Standard [online]. [cit. 2023-08-20]. Dostupné z: https://eips.ethereum.org/EIPS/eip-20

[24] Layout of a Solidity Source File [online]. [cit. 2023-08-20]. Dostupné z: https://docs.soliditylang.org/en/develop/layout-of-source-files.html

[25] Ethers Documentation [online]. [cit. 2023-08-20]. Dostupné z: https://docs.ethers.org/v6/

[26] 7 Useful NPM libraries for the blockchain developers [online]. [cit. 2023-08-20]. Dostupné z: https://medium.com/geekculture/7-useful-npm-libraries-for-the-blockchain-developers-7662729d77b6

[27] Web3.js Vs Ethers.js : Know the Key Differences [online]. [cit. 2023-08-20]. Dostupné z: https://www.blockchain-council.org/web-3/web3-js-vs-ethers-js/

[28] Contract ABI Specification [online]. [cit. 2023-08-20]. Dostupné z: https://docs.soliditylang.org/en/latest/abi-spec.html

[29] The 7 Best Solidity IDEs for Developers (2023) [online]. [cit. 2023-08-20]. Dostupné z: https://www.alchemy.com/overviews/solidity-ide

[30] Remix's documentation [online]. [cit. 2023-08-20]. Dostupné z: https://remix-ide.readthedocs.io/

[31] Solidity support for Visual Studio code [online]. [cit. 2023-08-20]. Dostupné z: https://marketplace.visualstudio.com/items?itemName=JuanBlanco.solidity

[32] Truffle Documentation [online]. [cit. 2023-08-20]. Dostupné z: https://trufflesuite.com/docs/

[33] Hardhat Documentation [online]. [cit. 2023-08-20]. Dostupné z: https://hardhat.org/docs

[34] Blockchain Consensus Mechanisms: A Primer for Supervisors [online]. [cit. 2023-08-20]. Dostupné z: https://www.imf.org/en/Publications/fintech-notes/Issues/2022/01/25/Blockchain-Consensus-Mechanisms-511769

[35] What Are Consensus Mechanisms in Blockchain and Cryptocurrency? [online]. [cit. 2023-08-20]. Dostupné z: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp

[36] Vysvětlení byzantské odolnosti proti chybám [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/cs/articles/byzantine-fault-tolerance-explained

[37] Crypto Tokens vs Coins — What's the Difference? [online]. [cit. 2023-08-20]. Dostupné z: https://crypto.com/university/crypto-tokens-vs-coins-difference

[38] What Is Blockchain? [online]. [cit. 2023-08-20]. Dostupné z: https://crypto.com/university/what-is-blockchain-consensus

[39] The History of Phishing [online]. [cit. 2023-08-20]. Dostupné z: https://www.graphus.ai/blog/history-of-phishing/

[40] CONSENSUS MECHANISMS [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/consensus-mechanisms/

[41] What is "proof of work" or "proof of stake"? [online]. [cit. 2023-08-20]. Dostupné z: https://www.coinbase.com/learn/crypto-basics/what-is-proof-of-work-or-proof-of-stake

[42] ZHENG, Zibin, Hong-Ning DAI, Mingdong TANG a Xiangping CHEN, ed. Blockchain and Trustworthy Systems [online]. Singapore: Springer Singapore, 2020 [cit. 2023-08-20]. Communications in Computer and Information Science. ISBN 978-981-15-2776-0. Dostupné z: doi:10.1007/978-981-15-2777-7

[43] Coin vs Token: What Is the Difference? [online]. [cit. 2023-08-20]. Dostupné z: https://coinmarketcap.com/alexandria/article/coin-vs-token:-what-is-the-difference

[44] Sybil Attack [online]. [cit. 2023-08-20]. Dostupné z: https://coinmarketcap.com/alexandria/glossary/sybil-attack

[45] What Is a 51% Attack? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-a-51-percent-attack

[46] 51% Attack: Definition, Who Is At Risk, Example, and Cost [online]. [cit. 2023-08-20]. Dostupné z: https://www.investopedia.com/terms/1/51-attack.asp

[47] Coinmarketcap [online]. [cit. 2023-08-20]. Dostupné z: https://coinmarketcap.com/

[48] Sybil [online]. [cit. 2023-08-20]. Dostupné z: https://en.wikipedia.org/wiki/Sybil_(Schreiber_book)

[49] Vysvětlení Sybil útoků [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/cs/articles/sybil-attacks-explained

[50] What is an Eclipse Attack? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-an-eclipse-attack

[51] GAYVORONSKAYA, Tatiana a Christoph MEINEL. Blockchain [online]. Cham: Springer International Publishing, 2021 [cit. 2023-08-20]. ISBN 978-3-030-61558-1. Dostupné z: doi:10.1007/978-3-030-61559-8

[52] Bitcoin: A Peer-to-Peer Electronic Cash System [online]. [cit. 2023-08-20]. Dostupné z: https://bitcoin.org/bitcoin.pdf

[53] VAN OORSCHOT, Paul C. Computer Security and the Internet [online]. Cham: Springer International Publishing, 2021 [cit. 2023-08-20]. Information Security and Cryptography. ISBN 978-3-030-83410-4. Dostupné z: doi:10.1007/978-3-030-83411-1

[54] YI, Xun, Xuechao YANG, Andrei KELAREV, Kwok Yan LAM a Zahir TARI. Blockchain Foundations and Applications [online]. Cham: Springer International Publishing, 2022 [cit. 2023-08-20]. SpringerBriefs in Applied Sciences and Technology. ISBN 978-3-031-09669-3. Dostupné z: doi:10.1007/978-3-031-09670-9

[55] PROOF-OF-WORK (POW) [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/

[56] Introduction to Web3 [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/web3/

[57] The Architecture of a Web 3.0 application [online]. [cit. 2023-08-20]. Dostupné z: https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application

[58] What is Web3? The Decentralized Internet of the Future Explained [online]. [cit. 2023-08-20]. Dostupné z: https://www.freecodecamp.org/news/what-is-web3/

[59] Co jsou decentralizované aplikace (DApps)? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/cs/articles/what-are-decentralized-applications-dapps

[60] What Is GameFi and How Does It Work? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-gamefi-and-how-does-it-work

[61] Decentralized Autonomous Organizations (DAOs) [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/decentralized-autonomous-organizations-daos-explained

[62] A Beginner's Guide to Decentralized Finance (DeFi) [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/the-complete-beginners-guide-to-decentralized-finance-defi

[63] What is DeFi? [online]. [cit. 2023-08-20]. Dostupné z: https://www.coinbase.com/learn/crypto-basics/what-is-defi

[64] What Is an NFT? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-an-nft

[65] AGGARWAL, Shubhani a Neeraj KUMAR. Cryptocurrencies. In: The Blockchain Technology for Secure and Smart Applications across Industry Verticals [online]. Elsevier, 2021, s. 227-266 [cit. 2023-08-20]. Advances in Computers. ISBN 9780128219911. Dostupné z: doi:10.1016/bs.adcom.2020.08.012

[66] Altcoin Meaning [online]. [cit. 2023-08-20]. Dostupné z: https://www.ledger.com/academy/glossary/altcoin

[67] The history of Ethereum [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/history/

[68] History of ETH: The rise of the Ethereum blockchain [online]. [cit. 2023-08-20]. Dostupné z: https://cointelegraph.com/learn/history-of-ethereum-blockchain

[69] What Is Hashing? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-hashing

[70] Co je kryptoměnová peněženka? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/cs/articles/crypto-wallet-types-explained

[71] Co je to fiat měna a fiat peníze [online]. [cit. 2023-08-20]. Dostupné z: https://www.numismatikasova.cz/radce/co-je-to-fiat-mena-fiat-penize/

[72] Vysvětlení hashových (Merkleových) stromů a kořenového hashe [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/cs/articles/merkle-trees-and-merkle-roots-explained

[73] Co je kryptografie s veřejným klíčem? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/cs/articles/what-is-public-key-cryptography

[74] INTRODUCTION TO DAPPS [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/dapps/

[75] An Introduction to ERC-20 Tokens [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/an-introduction-to-erc-20-tokens

[76] Seed Phrase, Explained [online]. [cit. 2023-08-20]. Dostupné z: https://medium.com/blockchain/seed-phrase-explained-9468a5fd8dc3

[77] What Is the Token Economy? [online]. [cit. 2023-08-20]. Dostupné z: https://www.oreilly.com/library/view/what-is-the/9781492072973/ch01.html

[78] What Is Tokenomics and Why Does It Matter? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-tokenomics-and-why-does-it-matter

[79] What Is Cryptocurrency Mining and How Does It Work? [online]. [cit. 2023-08-20]. Dostupné z: https://academy.binance.com/en/articles/what-is-crypto-mining-and-how-does-it-work

[80] Chainlink Code Examples [online]. [cit. 2023-08-20]. Dostupné z: https://docs.chain.link/data-feeds/examples

[81] Ethereum Documentation Transactions [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/transactions/

[82] DappRadar Industry Overview [online]. [cit. 2023-08-20]. Dostupné z: https://dappradar.com/industry-overview

[83] Goerli Etherscan [online]. [cit. 2023-08-20]. Dostupné z: https://goerli.etherscan.io/

[84] KYBIC, Jan. Konečný automat [online]. [cit. 2023-08-20]. Dostupné z: https://cw.fel.cvut.cz/old/_media/courses/b3b33alp/prednasky/07a_konecny_automat.pdf

[85] HLADKÁ, Eva. 7. Peer-to-peer (P2P) networks [online]. [cit. 2023-08-20]. Dostupné z: https://is.muni.cz/el/fi/podzim2016/PB002/um/lecture7-1-1.pdf

[86] ZHENG, Gavin, Longxiang GAO, Liqun HUANG a Jian GUAN. Ethereum Smart Contract Development in Solidity [online]. Singapore: Springer Singapore, 2021 [cit. 2023-08-20]. ISBN 978-981-15-6217-4. Dostupné z: doi:10.1007/978-981-15-6218-1

[87] CAI, Liang, Qilei LI a Xiubo LIANG. Advanced Blockchain Technology [online]. Singapore: Springer Nature Singapore, 2022 [cit. 2023-08-20]. ISBN 978-981-19-3595-4. Dostupné z: doi:10.1007/978-981-19-3596-1

[88] CAI, Liang, Qilei LI, Xiubo LIANG, Liang CAI, Qilei LI a Xiubo LIANG. In-Depth Interpretation of Ethereum. In: Advanced Blockchain Technology [online]. Singapore: Springer Nature Singapore, 2022, s. 47-99 [cit. 2023-08-20]. ISBN 978-981-19-3595-4. Dostupné z: doi:10.1007/978-981-19-3596-1_2

[89] OLIVA, Gustavo A., Ahmed E. HASSAN a Zhen Ming JIANG. An exploratory study of smart contracts in the Ethereum blockchain platform. Empirical Software Engineering [online]. 2020, 25(3), 1864-1904 [cit. 2023-08-20]. ISSN 1382-3256. Dostupné z: doi:10.1007/s10664-019-09796-5

[90] GENCER, Adem Efe, Soumya BASU, Ittay EYAL, Robbert VAN RENESSE a Emin Gün SIRER. Decentralization in Bitcoin and Ethereum Networks. In: MEIKLEJOHN, Sarah a Kazue SAKO, ed. Financial Cryptography and Data Security [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, s. 439-457 [cit. 2023-08-20]. Lecture Notes in Computer Science. ISBN 978-3-662-58386-9. Dostupné z: doi:10.1007/978-3-662-58387-6_24

[91] Vysvětlení Bitcoin Mempool: Co je Mempool? [online]. [cit. 2023-08-20]. Dostupné z: https://www.litebit.eu/cz/vzdelavani/vysvetleni-bitcoin-mempool-co-je-mempool

[92] Blockchain – Elliptic Curve Digital Signature Algorithm (ECDSA) [online]. [cit. 2023-08-20]. Dostupné z: https://www.geeksforgeeks.org/blockchain-elliptic-curve-digital-signature-algorithm-ecdsa/

[93] How to Use Keccak256 Hash Function with Solidity [online]. [cit. 2023-08-20]. Dostupné z: https://www.quicknode.com/guides/ethereum-development/smart-contracts/how-to-use-keccak256-with-solidity

[94] Intro to Ethereum [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/intro-to-ethereum/

[95] Intro to Ether [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/intro-to-ether/

[96] Ethereum Accounts [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/accounts/

[97] Blocks [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/blocks/

[98] Ethereum Virtual Machine [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/evm/

[99] Gas [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/gas/

[100] Networks [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/networks/

[101] Consensus Mechanisms [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/consensus-mechanisms/

[102] PROOF-OF-STAKE [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

[103] Introduction to Smart Contracts [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/smart-contracts/

[104] Ethereum Development Standards [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/standards/

[105] Oracles [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/oracles/

[106] Web2 vs Web3 [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/web2-vs-web3/

[107] ERC-20 Token Standard [online]. [cit. 2023-08-20]. Dostupné z: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[108] ERC-20 [online]. [cit. 2023-08-20]. Dostupné z: https://wiki.polygon.technology/docs/supernets/design/bridge/assets/erc/erc20/

[109] Deploy TT20 [online]. [cit. 2023-08-20]. Dostupné z: https://docs.developers.thundercore.com/deploying-on-thundercore/config-custom-tokens-tt-20

[110] EOSIO Token Tutorial [online]. [cit. 2023-08-20]. Dostupné z: https://developers.eos.io/welcome/v2.2/tutorials/eosio_token

[111] Ganache [online]. [cit. 2023-08-20]. Dostupné z: https://trufflesuite.com/ganache/

[112] Installing the Solidity Compiler [online]. [cit. 2023-08-20]. Dostupné z: https://docs.soliditylang.org/en/develop/installing-solidity.html

[113] React [online]. [cit. 2023-08-20]. Dostupné z: https://legacy.reactjs.org/

[114] Chainlink [online]. [cit. 2023-08-20]. Dostupné z: https://chain.link/

8 Přílohy

1) Přiložený adresář

Součástí práce je přiložený adresář obsahující zdrojové kódy práce (aplikace a chytré smlouvy) umístěný na GitHub platformě. Adresář obsahuje návod, jak spustit testovací prostředí a vyzkoušet fungování aplikace včetně interakce s chytrou smlouvou.

Adresář je dostupný na následujícím webovém odkazu:
https://github.com/filiproskot/DP

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2022/2023

Studijní program: Aplikovaná informatika
Forma studia: Prezenční
Obor/kombinace: Aplikovaná informatika (ai2-p)

Podklad pro zadání DIPLOMOVÉ práce studenta

Jméno a příjmení: **Bc. Filip Roškot**
Osobní číslo: **I1900804**
Adresa: Antonína Petrofa 2113/10, Hradec Králové – Nový Hradec Králové, 50009 Hradec Králové 9, Česká republika

Téma práce: Integrace vlastního krypto tokenu do webové aplikace
Téma práce anglicky: Integration of a custom crypto token into a web application
Jazyk práce: Čeština

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.
Katedra informačních technologií

Zásady pro vypracování:

Cíl práce: Návrh a vytvoření vlastního krypto tokenu a jeho následná integrace do webové aplikace, která umožní jeho základní ovládání.
1. Úvod 2. Cíl práce 3. Blockchain 4. Vlastní implementace 5. Shrnutí výsledků 6. Závěry a doporučení 7. Seznam použité literatury 8. Přílohy

Seznam doporučené literatury:

YANO, Makoto, Chris DAI, Kenichi MASUDA a Yoshio KISHIMOTO, ed. Blockchain and Crypto Currency. Singapore: Springer Singapore, 2020. Economics, Law, and Institutions in Asia Pacific. ISBN 978-981-15-3375-4.
VESTERGAARD, Cindy, Haimanot Anbesaw BOBOSHA a Karolin LANGFELDT. Distributed Ledger Technology: Beyond the Hype. In: VESTERGAARD, Cindy, ed. Blockchain for International Security. Cham: Springer International Publishing, 2021, s. 7-22. Advanced Sciences and Technologies for Security Applications. ISBN 978-3-030--86239-8.
SUNYAEV, Ali a Ali SUNYAEV. Distributed Ledger Technology. In: Internet Computing. Cham: Springer International Publishing, 2020, s. 265-299. ISBN 978-3-030--34956-1.

Podpis studenta: Datum:

Podpis vedoucího práce: Datum:

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor: **Bc. Filip Roškot**

Studium: I1900804

Studijní program: N1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název diplomové práce: **Integrace vlastního krypto tokenu do webové aplikace**

Název diplomové práce AJ: Integration of a custom crypto token into a web application

Cíl, metody, literatura, předpoklady:

Cíl práce: Návrh a vytvoření vlastního krypto tokenu a jeho následná integrace do webové aplikace, která umožní jeho základní ovládání.

1. Úvod 2. Cíl práce 3. Blockchain 4. Vlastní implementace 5. Shrnutí výsledků 6. Závěry a doporučení 7. Seznam použité literatury 8. Přílohy

YANO, Makoto, Chris DAI, Kenichi MASUDA a Yoshio KISHIMOTO, ed. Blockchain and Crypto Currency. Singapore: Springer Singapore, 2020. Economics, Law, and Institutions in Asia Pacific. ISBN 978-981-15-3375-4.

VESTERGAARD, Cindy, Haimanot Anbesaw BOBOSHA a Karolin LANGFELDT. Distributed Ledger Technology: Beyond the Hype. In: VESTERGAARD, Cindy, ed. Blockchain for International Security. Cham: Springer International Publishing, 2021, s. 7-22. Advanced Sciences and Technologies for Security Applications. ISBN 978-3-030-86239-8.

SUNYAEV, Ali a Ali SUNYAEV. Distributed Ledger Technology. In: Internet Computing. Cham: Springer International Publishing, 2020, s. 265-299. ISBN 978-3-030-34956-1.

Zadávající pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.

Oponent: Ing. Martina Husáková, Ph.D.

Datum zadání závěrečné práce: 24.1.2022