Ekonomická fakulta
Faculty of Economics

Jihočeská univerzita v Českých Budějovicích
University of South Bohemia in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Ekonomická fakulta
Katedra aplikované matematiky a informatiky

Diplomová práce

Aplikace technologie blockchain a chytrých kontraktů v oblasti IoT

Vypracoval: Bc. Petr Knotek
Vedoucí práce: doc. Ing. Ladislav Beránek, CSc., MBA

České Budějovice 2023

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Ekonomická fakulta

Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Petr KNOTEK**
Osobní číslo:	**E21036**
Studijní program:	**N0613A140025 Aplikovaná informatika**
Specializace:	**Podniková informatika**
Téma práce:	**Aplikace technologie blockchain a chytrých kontaktů v oblasti IoT**
Zadávající katedra:	*****Katedra aplikované matematiky a informatiky**

Zásady pro vypracování

Cílem práce je návrh a implementace aplikace pro využití technologie Smart Contracts (chytrých kontraktů) v oblasti Internetu věcí (IoT). Aplikace by měla např. umožnit předávání dat v prostředí IoT s využitím Smart Contrats. V teoretické části práce poskytne přehled technologie blockchainu a různých platforem Smart Contracts. Budou analyzovány různé platformy této technologie a bude vybrána jedna z nich pro implementaci aplikace na základě zvolených parametrů. Dále bude stručně popsána oblast IoT. Praktická část bude obsahovat popis vývoje, implementace a testování navržené aplikace.

Metodický postup:

1. Rešerše problematiky blockchainu a chytrých kontraktů v oblasti IoT.
2. Výběr metod pro vytvoření aplikace, návrh architektury aplikace, vytvoření schémat.
3. Vytvoření a implementace aplikace využívající technologii Smart Contacts v oblasti IoT.
4. Testování a vyhodnocení.
5. Závěr.

Rozsah pracovní zprávy:	**50 – 60 stran**
Rozsah grafických prací:	**dle potřeby**
Forma zpracování diplomové práce:	**tištěná**

Seznam doporučené literatury:

1. BAMBARA, J. et al. (2019). *AI, IoT, and the Blockchain: Using the Power of Three to create Business, Legal and Technical Solutions.* Philadelphia: BookBaby.
2. FURNEAUX, N. (2018). *Investigating cryptocurrencies: understanding, extracting, and analyzing blockchain evidence.* Indianapolis: Wiley.
3. LYER K., & DANNEN, CH. (2018). *Building games with Ethereum smart contracts: intermediate projects for solidity developers.* New York, NY: Springer Science+Business Media.
4. MOHANTY, D. (2018). *Ethereum for architects and developers: with case studies and code samples in solidity.* Chennai, NY: Springer Science+Business Media.

Vedoucí diplomové práce: **doc. Ing. Ladislav Beránek, CSc., MBA**
***Katedra aplikované matematiky a informatiky

Datum zadání diplomové práce: **22. listopadu 2022**
Termín odevzdání diplomové práce: **14. dubna 2023**

doc. Dr. Ing. Dagmar Škodová Parmová
děkanka

doc. RNDr. Jana Klicnarová, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 23. listopadu 2022

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

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

.....	.....
Datum	Podpis studenta

Poděkování:

Děkuji vedoucímu práce doc. Ing. Ladislav Beránek, CSc., MBA za věnovaný čas, obětavou pomoc a věcné připomínky ke zpracování diplomové práce.

1 Úvod

Vždy jsem se zajímal o nové inovace, a v této práci bych se chtěl zaměřit na jednu, která by nám mohla změnit každodenní životy. Vzhledem k rychlé a obrovské revoluci, s níž se moderní svět v těchto dnech setkává, se očekává, že Blockchain, Internet of Thing, big data a další technologická vylepšení budou hrát v lidských životech významnou roli. Velký efekt nastane, když se všechny tyto technologie integrují dohromady, a vytvoří se tak funkční celek, ale to je zatím otázka budoucnosti.

Blockchainy v poslední době přitahují zájem zainteresovaných stran v širokém spektru průmyslových odvětví: od financí a zdravotnictví, až po veřejné služby, nemovitosti, a vládní sektor. Důvodem tohoto prudkého nárůstu zájmu o tuto technologii je díky blockchain aplikacím, které dříve mohly fungovat pouze prostřednictvím důvěryhodného zprostředkovatele, nyní mohou fungovat decentralizovaně, bez potřeby centrální autority, a dosáhnout stejné funkčnosti se stejnou mírou jistoty. To dříve jednoduše nebylo možné. Říkáme, že blockchain umožňuje sítě bez důvěry, protože strany mohou provádět transakce, i když si navzájem nedůvěřují. Absence důvěryhodného zprostředkovatele znamená rychlejší usmíření mezi transakčními stranami. Hojné využívání kryptografie, která je klíčovou vlastností blockchainových sítí, přináší autoritativnost za všechny interakce v síti. Chytrých smlouvách – automatizované skripty, které se nacházejí v blockchainu – integrují tyto koncepty a umožňují řádné, distribuované a automatizované pracovní postupy. Díky tomu by měly být blockchainy lákavé pro oblast IoT. Přechod na decentralizovanou síť samozřejmě nemusí mít vždy smysl. Navíc, i když je takový přechod žádoucí, požadavky aplikace mohou být takové, že blockchainová síť je nemůže splnit. Blockchainy a chytré smlouvy přinášejí řadu výhod, ale je s nimi spojeno i mnoho nevýhod.

Když se podíváme zpátky do doby, kdy se poprvé začaly objevovat nynější technologie tak se ocitneme v roce 1997, kdy Nick Szabo vytvořil termín "chytré smlouvy", a také dále vysvětlil jeho významové charakteristiky a aplikace (Szabo, 1997). V roce 2008 anonym "Satoshi Nakamoto" publikoval článek pod názvem "Bitcoin: A Peer-to-Peer Electronic Cash System", ve kterém podrobně popsal digitální měnu známou jako bitcoin, která byla vytvořena na základě blockchainu (Nakamoto, 2008). Blockchain zatím vstoupil do mnoha odvětví, jako jsou finance, řízení dodavatelského řetězce atd. Většina implementací blockchainu tvrdí, že zvyšuje efektivitu, snižuje náklady a podporuje transparentnější interakce mezi sociálními subjekty. Studie šla nad rámec současných

běžných aplikací a integrovala technologii blockchain do strategického plánování podniku, aby byla považována za konkurenční výhodu pro různé firmy. Došli k závěru, že" technologie blockchain je propojena s dalšími zdroji, a že konkurenceschopnost zdrojů se odráží prostřednictvím procesu výběru technologie (Bjørnstad, Harkestad, & Krogh, 2017). Vzhledem k tomu, že Blockchain se poprvé objevil v digitální měně Bitcoin, dochází k mírné záměně obou pojmů. Na základní úrovni můžeme jednoduše definovat "Blockchain" jako nástroj a Bitcoin nebo Ethereum jako jednu z aplikací Blockchainu.

Cílem této práce je poskytnout podrobný popis fungování blockchainů a chytrých smluv, identifikovat výhody a nevýhody, které jejich zavedení do systému přináší, a upozornit na způsoby, kterými lze využít blockchain a IoT společně. V druhé části se práce zaměří na tvorbu samotné aplikace s využitím blockchain technologie v kombinaci s chytrými kontrakty v prostředí IoT.

2 Blockchain

V této kapitole se budeme věnovat různým typům technologií blockchain a souvisejícími technologiemi. Hlavním tématem diskuse bude rozdíl mezi blockchain platformami a mechanismem konsensu.

2.1 Co je to blockchain

Technologie blockchain je distribuovaná databáze, která udržuje neměnnou veřejnou účetní knihu všech transakcí. Blockchain umožňuje zaznamenávat všechny transakce s časovým razítkem. Každý uzel v síti je zodpovědný za vedení a průběžné ověřování transakcí. Technologie blockchain zahrnuje vytváření digitálních tokenů pro digitální soubory, jako jsou dokumenty nebo transakce. Tyto digitální tokeny lze považovat za digitální otisky souborů. Tyto otisky jsou ukládány do skupin, které nazýváme "bloky". Tyto bloky jsou poté spojeny v řetězec a každý následující blok obsahuje digitální token z předchozího bloku, což zajišťuje nezměnitelnost informací. Hlavní myšlenkou technologie blockchain je registrace, potvrzování a převod všech druhů smluv a vlastností bez potřeby jakéhokoli prostředníka. Blockchain je tak považován za "stroj na důvěru" a jeho schopnost zabezpečit data a historii transakcí vedla k odhadu, že do roku 2025 bude 10 % světového HDP uloženo v blockchainu (Economist, 2015).

2.2 Historie blockchainu

Blockchain byl původně představen jako kryptograficky zabezpečený řetězec bloků v roce 1991 Stuartem Haber a W. Scott Stornetta Později, v roce 1992, Bayer, Haber a Stornetta začlenili do blockchainu Merklovy stromy, to mělo zlepšit efektivitu tak, aby bylo možné shromáždit několik dokumentů do jednoho bloku (HAYES, 2022). Koncept distribuovaného blockchainu poprvé představila anonymní osoba nebo skupina známá jako Satoshi Nakamoto v roce 2008 zveřejněním bílé knihy s názvem "Bitcoin: A Peer-to-Peer Electronic Cash System". Blockchain (bitcoin) se zrodil, když Satoshi Nakamoto vyřešil hlavolam teorie her zvaný problém byzantských generálů, který zajišťoval, že v určitém okamžiku může být blok aktiv převeden pouze na jednu další osobu, aniž by byla nutná kontrola třetí stranou. Koncept distribuovaného blockchainu byl poprvé uvolněn jako open-source software a implementován jako základ pro měnu Bitcoin v roce 2009, což umožnilo řešit problém s double spending. Tím se stal Bitcoin s využitím blockchainu první digitální měnou bez potřeby důvěryhodného správce. V původním dokumentu, který v říjnu 2008 představil Satoshi Nakamoto, byla slova blok a řetězec

použita odděleně. Když se termín dostal do širšího užívání, byl původně block chain, než se v roce 2016 stal jediným slovem blockchain (Nakamoto, 2008).

2.3 Problém byzantských generálů

Technologie blockchain odpovídá na "problém byzantských generálů" neboli jak jednotliví uživatelé zabezpečují svá data před nedůvěryhodnými subjekty. Problém byzantských generálů je praktickou verzí myšlenkového experimentu, který se nazývá Problém dvou armád. Problém je znázorněn dvěma nebo více generály, kteří obléhají město z opačných stran a snaží se koordinovat útok. Pokud generál A odešle zprávu "zaútočte zítra v poledne", netuší, zda generál B zprávu skutečně obdrží, a mohl by potenciálně pochodovat vstříc smrti, pokud zaútočí bez druhého generála. Generál B po obdržení zprávy netuší, zda je zpráva pravá, nebo byla vyslána nepřítelem, aby ho vlákal do pasti. Přesto bude předpokládat pravost a pošle odpověď potvrzující útok, ale aniž by věděl, zda generál A jeho odpověď obdržel, může se obávat, že druhý generál útok odloží, což znamená, že generál B bude zítra v poledne útočit sám a bude čelit jisté smrti. Generál A by samozřejmě mohl odeslat zprávu potvrzující přijetí potvrzení generála B, ale nikdy ve skutečnosti nebude vědět, zda zpráva dorazila na místo určení, nebo dokonce zda byla vůbec autentická. Tím se dostává do stejné situace, v jaké byl právě generál B. Tento problém se odráží tam a zpět do nekonečna, přičemž ani jeden z generálů si nikdy nemůže být jistý, zda jeho zpráva prošla, natož aby byla autentická. Abychom to vztáhli k blockchainu, můžeme to popsat následovně: Každý uživatel si může do "digitální schránky" uložit téměř cokoli, co má nějakou hodnotu. Obsah schránky může otevřít a měnit pouze pomocí jedinečného soukromého klíče. Informace uvnitř této schránky pak lze na požádání sdílet bez možnosti jejich úpravy nebo změny, nebo replikovat z původní podoby (Nakamoto, 2008).

2.4 Druhy blockchainu

Blockchainy můžeme rozdělit do tří kategorií: Veřejný blockchain, soukromý Blockchain a hybridní blockchain (nebo taky známý jako blockchain s povolením).

2.4.1 Veřejný blockchain

Plně otevřená veřejná účetní kniha nemá žádná omezení, pokud jde o oprávnění ke čtení a zápisu. Kdokoliv se může připojit k síti a získat informace, a také přidat informace. Každý, kdo je připojen k síti, má právo se účastnit konsensuálního protokolu, ověřovat nově přidané bloky a zajistit, aby nebyly v konfliktu s předchozími bloky v řetězci. Konsensuální protokol je nucen být založen na krypto-ekonomickém

mechanismu, a to kvůli otevřené povaze systému a kvůli nedostatku důvěry mezi uzly. Systém blockchain s veřejnou účetní knihou funguje bez požadavku důvěry mezi uživateli, proto je považován za plně decentralizovaný.

Mezi nejmodernější open source veřejné blockchain protokoly založené na konsensuálním algoritmu Proof of Work patří Bitcoin a Ethereum.

Vlastnosti veřejného blockchainu:

- přístupnost bez omezení – každý má povolení se připojit a stát se součástí sítě,
- participace uzlů – každý si může stáhnout kód a spustit veřejný uzel, který ověřuje transakce v síti a zapojit se tak do procesu konsensu,
- transparentnost transakcí – každý může posílat transakce prostřednictvím sítě a čekat na zařazení do blockchainu, pokud jsou platné,
- sledovatelnost – všechny transakce jsou dostupné prostřednictvím veřejného průzkumníku bloků a jsou transparentní, ale zároveň zachovávají anonymitu uživatelů.

2.4.2 Soukromý blockchain

Soukromý blockchain má určitá omezení oprávnění ke čtení a zápisu a je přísněji kontrolován než veřejná účetní kniha. Právo měnit, přidávat nebo číst informací je omezeno a je udržováno centralizovaně pro skupinu účastníků, např. organizace. Tyto systémy nemusí nutně potřebovat konsensuální protokol kvůli důvěryhodným uzlům. Soukromé účetní knihy nabízejí rychlý přístup k informacím, levné transakce a kontrolu úrovně soukromí. Příkladem aplikací je správa databází, audit atd., které jsou interní v rámci jedné společnosti, a proto veřejná čitelnost nemusí být v mnoha případech vůbec nutná. V některých případech je však žádoucí možnost veřejného auditu. Soukromé blockchainy (jako např. Multichain) představují způsob, jak využít výhod technologie blockchain vytvořením skupin a účastníků, kteří mohou interně ověřovat transakce. To s sebou nese riziko narušení bezpečnosti stejně jako u centralizovaného systému, ale má to výhody, pokud jde o škálovatelnost a soulad s pravidly ochrany osobních údajů a dalšími regulačními otázkami.

2.4.3 Federativní nebo konsorciální blockchainy

Existuje hybridní blockchain systém sestávající z některých prvků veřejné a soukromé účetní knihy, tzv. konsorciální účetní kniha. V konsorciální účetní knize je konsensuální protokol obvykle předem určen a spravován předem definovanou skupinou institucí. V konsorciálním systému blockchain může mít např. 20 institucí, které ovládají jeden uzel, a každý nově přidaný blok musí být podepsán alespoň 13 institucemi, aby byl

považován za platný (Drescher, 2017). Hybridní blockchainy systém je považován za částečně decentralizovaný. V konsorciální účetní knize má oprávnění ke čtení otevřená veřejnost nebo jen omezená skupina účastníků. Existuje i hybridní řešení, a to takové, že části informací jsou veřejné a jiné části nikoli. Federativní blockchainy (EWF – energetika, B3i – pojišťovnictví), fungují pod vedením skupiny, aniž by umožnily jakékoli osobě s přístupem k internetu účastnit se procesu ověřování transakcí. Federované blockchainy jsou rychlejší a poskytují větší soukromí transakcí. Konsorciální blockchainy se používají především v bankovním sektoru. Proces konsensu je řízen předem vybranou sadou uzlů.

2.4.4 Hlavní technologie spojené s blockchainem

Bitcoin je považován za nástroj, který vytvořil nové funkce kombinací stávajících technologií. Aby bylo možné provozovat systém, jako je ten pro elektronické peníze, bez centrální autority, je nezbytné zavést opatření, která zabrání vzniku falšování údajů a Double spendingu, jakož i mechanismus, který by udržoval systém proti případným útokům ze strany uživatelů. Aby bitcoin mohl fungovat správně, jsou s ním spojené technologie, které nám tento provoz zajistí (Peer to peer, hashovací funkce, kryptografie s veřejným klíčem a digitální podpis, Proof of Work, Proof of Stake). Jednotlivé položky jsou popsané níže.

2.4.5 Síť Peer-to-Peer (P2P)

V síti typu klient-server přebírá server odpovědnost za uchovávání a poskytování dat, zatímco klient žádá server o data a získává k nim přístup, a jejich role jsou tedy pevně stanoveny. Naproti tomu P2P sítě jsou typem sítí, kde všechny uzly drží data a vytvářejí autonomní síť. Uzly si navzájem poskytují data bez nutnosti centrálního serveru. Není určena žádná pevná role uzlů jako serveru nebo klienta. Pro fungování P2P sítí je nutné zajistit vyhledávací metody pro správu umístění uzlů a dat a metody pro přenos dat mezi uzly. Tyto sítě umožňují decentralizaci a vyšší flexibilitu oproti tradičním klient server sítím.

2.4.6 Hashovací funkce neboli hash

Technologie blockchainu se ve velké míře opírá o hashování a hashovací funkce. Hashovací funkce je matematický algoritmus, který převádí vstup na výstup, zatímco hash je výsledkem této přeměny vstupních dat. Hlavní vlastností hashovací funkce je odolnost proti kolizím, což znamená, že je velmi obtížné znovu vytvořit vstupní data pouze z jejího výstupu (hodnoty hash). Tento mechanismus se používá k detekci

falešných dat, protože ze stejných vstupních dat vždy vznikne stejný hash, zatímco i nepatrný rozdíl ve vstupních datech způsobí úplně odlišnou hodnotu hash.

2.4.7 Kryptografie s veřejným klíčem a digitální podpis

Kryptografie s veřejným klíčem je kryptografická metoda, která využívá dva klíče – soukromý klíč a veřejný klíč – pro šifrování a dešifrování dat. Problém předávání klíčů byl vyřešen rozdělením klíče na klíč pro soukromé použití (soukromý klíč) a klíč dostupný komukoli (veřejný klíč). V případě kryptografie se symetrickým klíčem vyžaduje použití stejného klíče pro šifrování a dešifrování různá bezpečnostní opatření pro předání klíče pouze příslušné protistraně. Naproti tomu kryptografie s veřejným klíčem umožňuje bezpečné doručování a přijímání souborů pouze tehdy, pokud si příjemce připraví dvojici soukromého a veřejného klíče a veřejný klíč předem doručí odesílateli. Odesílatel poté použije veřejný klíč k šifrování dat, která chce předat příjemci. Tyto data mohou být dešifrovány pouze použitím soukromého klíče příjemce. Bezpečnost může být zachována, i když veřejný klíč používají jiné osoby, pokud příjemce řádně spravuje svůj soukromý klíč. Blockchain používá k ověření pravosti transakcí mechanismus asymetrické kryptografie. Digitálním podpisem se rozumí mechanismus prokazování pravosti dat zasílaných prostřednictvím sítě pomocí dvojice klíčů jako v kryptografii s veřejným klíčem. Digitální podpis, zasílaný příjemci spolu s přidaným souborem, je vytvořen zašifrováním hodnoty hash odesílaného souboru soukromým klíčem odesílatele. Příjemce použije stejnou hashovací funkci jako odesílatel, aby sám vytvořil hashovací hodnotu souboru, a zkontroluje vytvořenou hashovací hodnotu s hashovací hodnotou získanou dešifrováním digitálního podpisu odesílatele veřejným klíčem odesílatele. Výsledkem je potvrzení, že digitální podpis odesílatele je pravý. V blockchainu se typický digitální podpis skládá ze dvou fází: fáze podepisování a fáze ověřování, jak je popsáno výše. Pro tuto úlohu se v blockchainu často používá algoritmus digitálního podpisu eliptickou křivkou (ECDSA).

2.4.8 Mechanismus konsensu

Hlavním problémem technologie blockchain s veřejnými účetními knihami je zajištění konsensu všech účastníků peer-to-peer sítě (Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, 2017). Konsensuální protokol slouží k zajištění toho, aby účastníci sítě dodržovali pravidla sítě, a aby transakce byly ověřovány ve správném pořadí. Používá se také k zajištění toho, aby informace v bloku byly správné, aby uzly (těžaři) dostávaly spravedlivou odměnu, a aby se předešlo problémům, jako je problém dvojího utrácení. Algoritmy pro dosažení konsensu s libovolnými chybami vyžadují

hlasování mezi určitou skupinou rovnocenných partnerů. Existují dva hlavní přístupy: "Nakamotův konsensus" a Byzantská tolerance chyb (BFT). Nakamotův přístup volí vůdce prostřednictvím určité formy "loterie", který poté navrhne blok, jenž může být přidán do řetězce. BFT k dosažení potřebného konsensu využívá více kol explicitního hlasování. V této části oddílu představíme některé z nejpopulárnějších mechanismů konsensu, Proof of Work a Proof of Stake jsou využívané a otestované především prostřednictvím kryptoměn.

2.4.8.1 Důkaz práce (PoW)

Důkazem práce (PoW) se obecně rozumí mechanismus, který potvrzuje nevinu osoby tím, že jí dá provést určitou práci, která je jednoduchá, ale problematická, a lze snadno ověřit, že ji provedla. Bashir tvrdí (Bashir, 2020), že jedním ze způsobů zajištění pravosti je nechat každého uživatele v síti získat jeden hlas, a nechat všechny uživatele hlasovat o tom, která transakce má být zařazena do dalšího bloku. Počet hlasů rozhoduje o tom, která sada transakcí by měla být zařazena. Tento druh konsensuálního procesu je zranitelný vůči útokům typu Sybil, kdyby si jeden uživatel mohl vytvořit více účtů, a získat v síti větší vliv. Nakamoto, tvůrce Bitcoinu, tento problém vlivu vyřešil přidáním nákladů na hlasování. Výše vlivu každého uživatele je založena na jeho výpočetním výkonu. Čím větší výpočetní výkon, tím vyšší potřebná energie, a tím vyšší náklady na hardware. To je koncept konsensuálního protokolu proof-of-work. V případě bitcoinů (které používají konsensuální protokol proof-of-work) síť shromažďuje všechny transakce provedené během stanoveného období do bloku. Úkolem uzlu je tyto transakce potvrdit a zapsat je do blockchainu hashováním informací, aby byly chráněny také před narušiteli. Uzly získávají ekonomickou motivaci k další těžbě a hashování, čím více bloků vytvoří, tím více bitcoinů obdrží. Antonyho Lewise tvrdí (Lewis, 2018), že když uzel vytvoří blok, je distribuován do sousedních uzlů. Sousední uzly nezávisle ověřují, zda jsou informace v bloku správné, a zda byla dodržena pravidla. V bitcoinové síti se doporučuje počkat alespoň na šest bloků, aby bylo jisté, že transakce je konečná. Uzly mezi sebou soutěží o to, kdo první vytvoří blok, a několik uzlů může pracovat na stejné transakci současně, tím vzniká tak tzv. blockchain fork. Blok, který je vytvořen jako první a má za sebou nejdelší blockchain, vyhrává a tento uzel získává odměnu. Přestože je tento postup náročný na hardware a energii, a zvyšuje tak náklady na těžbu, bylo empiricky prokázáno, že proof-of-work je bezpečný a robustní. Můžeme říct, že u konsensuálního protokolu proof-of-work existují určité pády, např. riziko 51% útoku, a existují vysoké

energetické náklady na výrobu jednoho bloku. Několik studii potvrdilo, že protokol proof-of-work směřuje k sebedestrukci. Těžařská komunita se zmenšuje a specializuje, přičemž velké společnosti s velkými zdroji by mohly přetáhnout jednotlivé těžaře. Díky této specializaci těžby se systém centralizuje na několik velkých společností a riziko 51% útoku se zvyšuje.

2.4.8.2 Důkaz sázky (PoS)

Aby se snížilo riziko 51% útoku a snížila se spotřeba energie, byl v rámci blockchainové komunity zaveden nový konsensuální protokol, který se nazývá proof-of-stake. Namísto dokazování, že uzel vyřešil výpočetně náročnou úlohu, jak se to dělá v protokolu proof-of-work, může uzel místo toho prokázat, že má určité množství mincí (Bashir, 2020). V případě proof-of-stake jsou k vytvoření nového bloku potřeba mince, nikoli výpočetní výkon, a uzel s největším počtem mincí získává největší vliv (Buterin, 2014). Bitcoinová komunita tvrdí, že protokol proof-of-stake sníží riziko 51% útoku. Problém se řeší tím, že pravděpodobnost 51% útoku se snižuje díky mincím, které těžař v rámci sítě investuje. Pokud má někdo 51% výpočetního výkonu v rámci proof-of-stake protokolu, musí vlastnit také 51% všech bitcoinů. Podle teorie her je tedy v zájmu většinového vlastníka mít stabilní a bezpečnou síť, a proto na ni nebude útočit. Pokud k útoku dojde, digitální měnu to pouze destabilizuje a sníží její hodnotu. Jedním z problémů protokolu proof-of-stake je otázka takzvaného fork stavu, když jeden uzel začne těžit transakci, jiný uzel by ji mohl začít těžit současně, aniž by ho to stálo výpočetní výkon. Bashir tvrdí (Bashir, 2020), že ve srovnání s protokolem proof-of-work se zvyšuje riziko zlotřilých uzlů, které vytvoří fork. Tím se zvyšuje riziko útoků s dvojím utrácením a chamtivého chování, kdy uzly začnou těžit na všech fork rozvětveních aby nepřišly o odměny za bloky. Tento problém lze vyřešit použitím bloků s kontrolním bodem, kdy bloky před kontrolním bodem nelze revidovat a problém double-spending útoků je vyřešen. V protokolu proof-of-stake zůstává riziko 51% útoku.

2.4.8.3 Delegovaný důkaz sázky (DPoS)

DPOS (Delegated proof-of-stake), tedy delegovaný důkaz o sázce. Podobně jako v případě POS dostávají těžaři přednostní právo na generování bloků podle výše svého podílu. Hlavní rozdíl mezi POS a DPOS je ten, že POS je přímá demokracie, zatímco DPOS je reprezentativní demokracie. Podílníci volí své zástupce pro generování a ověřování bloků. S výrazně menším početem uzlů, které mají blok potvrdit, může být blok potvrzen rychleji, čímž i potvrzení samotné transakce je rychlejší. Mezitím se parametry

sítě, jako např. velikost bloku a intervaly mezi bloky, mohou být upraveny. Uživatelé se navíc nemusí obávat nepoctivých delegátů, protože delegáti by mohli být snadno vyloučeni. Systém DPOS byl již implementován a je páteří systému Bitshares. (Bitshares, 2023)

2.4.8.4 Praktická byzantská tolerance chyb (PBFT)

PBFT (Practical byzantine fault tolerance) je replikační algoritmus pro tolerování byzantních poruch. Hyperledger Fabric (Seeley & IO, 2019) využívá PBFT jako svůj konsensuální algoritmus, protože PBFT by mohl zvládnout až 1/3 škodlivých byzantních replik. V jednom kole je určen nový blok. V každém kole by se podle určitých pravidel vybral primární blok, a ten je zodpovědný za uspořádání transakce. Celý proces by se dal rozdělit do tří fází: předpříprava, příprava a odevzdání. V každé fázi by uzel vstoupil do další fáze, pokud by získal hlasy od více než 2/3 všech uzlů. PBFT tedy vyžaduje, aby byl každý uzel v síti znám. Stejně jako PBFT je i Stellar Consensus Protocol byzantským dohodovacím protokolem. V PBFT neexistuje žádný hashovací postup. V PBFT se každý uzel musí dotazovat ostatních uzlů, zatímco SCP dává účastníkům právo vybrat si, které množině ostatních účastníků budou věřit. Na základě PBFT implementoval Neo (Neo, 2017) svůj dBFT. V dBFT jsou některé profesionální uzly voleny k zaznamenávání transakcí namísto všech uzlů.

2.4.8.5 Proof of Elapsed Time

Konsensus PoET (Proof of Elapsed Time) byl poprvé zpřístupněn v systému Hyperledger prostřednictvím abstraktního prostředí TEE (trusted execution environment). Pokud jde o funkčnost, PoET stochasticky vybírá konkrétní peery, kteří budou provádět požadavky se stanovenou cílovou rychlostí. Každý peer čeká po dobu určenou vzorkem, než se odebere vzorek exponenciálně rozdělený dle náhodné veličiny. Voliči vyberou peer s nejmenší velikostí vzorku. K zastavení podvádění se používá důvěryhodné prostředí pro provádění, ověřování identity a černá listina založená na kryptografii asymetrických klíčů a další zásady volby. (Hyperledger, 2023)

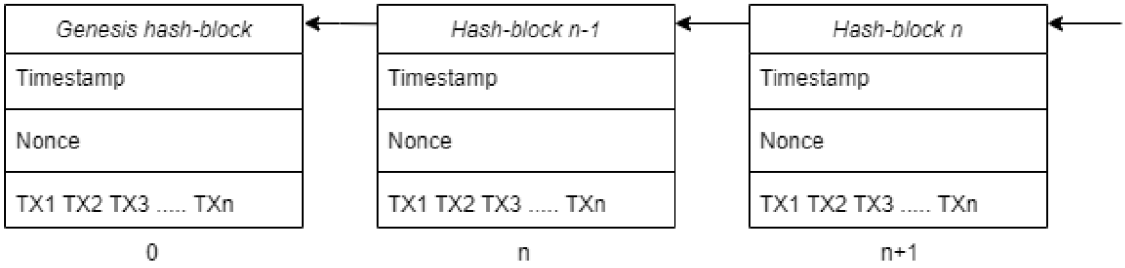
	PoW	PoS	DPoS	PBFT	PoET
Identity managment	Otevřený	Otevřený	Otevřený	S povolením	
Bezpečnosti hrozby	Více než 25 % výpočetního výkonu	Více než 51% sázka	Více než 51% volených validátorů	Více než 33 % škodlivých replik	
Energeticky usporný	Ne	Částečně	Částečně	Ano	Ano
Platformy	Bitcoin	Ethereum	Bitshares	Hyperledger Sawtooth	Hyperledger Sawtooth

Tabulka 1:Tabulka porovnání konsensu

Zdroj: Autor

2.5 Princip funkčnosti blockchainu

Na přiloženém obrázku 1 vidíme přehled fungování blockchainu. Aby blockchain fungoval se svými jedinečnými vlastnostmi, musí bloky různých dat (transakcí) obsahovat i další klíčové informace. Každý blok má odkaz, který odkazuje na bezprostředně předcházející blok prostřednictvím nadřazeného bloku, tedy primární hodnoty hash předchozího bloku. Je důležité poznamenat, že blockchain Ethereum obsahuje také uncle hash bloků (neboli hash dětí předků bloku). (Buterin, 2014) Genesis blok blockchainu je první blok bez rodičovského bloku.



Obrázek 1: Struktura bloků v blockchainu

Zdroj: Autor

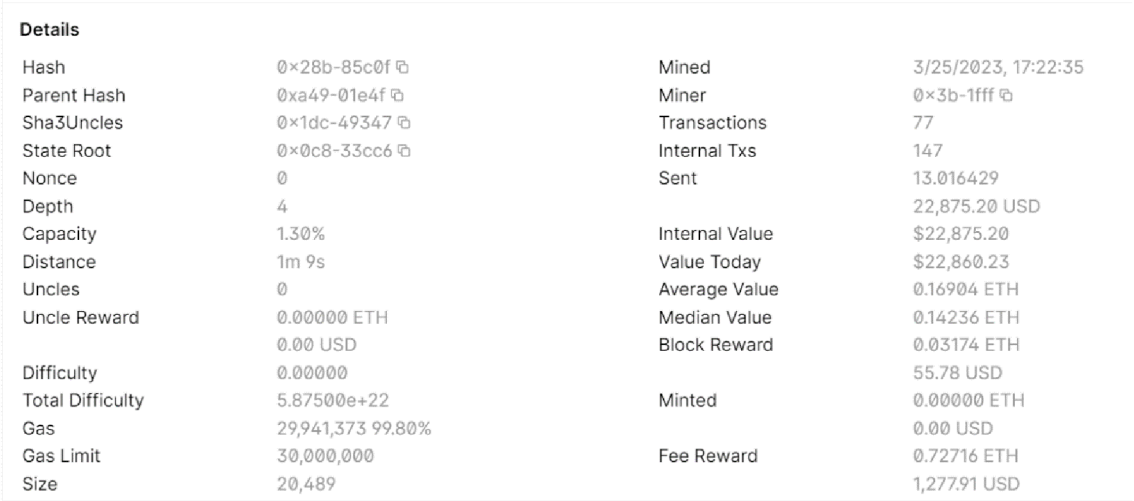
Blok se skládá z těla bloku a záhlaví bloku, přičemž záhlaví bloku obsahuje následující údaje viz tabulka 2.

	Velikost	Popis
Verze	4 bytes	Číslo verze bloku, které určuje pravidla validace bloku.
Hash záhlaví předchozího bloku	32 bytes	Jedná se o dvojitý hash SHA-256 hlavičky z předchozího bloku.
Merkle tree root Hash	32 bytes	Jedná se o Merkle tree každé transakce v bloku dvakrát zahašovaný pomocí algoritmu SHA-256.
Časové razítko	4 bytes	Toto pole obsahuje přibližný čas vytvoření bloku ve formátu Unix epoch. Jedná se o přesný okamžik, kdy těžař začal hashovat hlavičku.
Cílová obtížnost (Nbits)	4 bytes	Jedná se o aktuální cíl obtížnosti sítě/bloku.
Nonce	4 bytes	Slouží k vytvoření Hash, který je nižší než cílová obtížnost, vytvářejí ho těžaři neustálou změnou tohoto libovolného celého čísla.

Tabulka 2: Obsah záhlaví bloku v blockchainu

Zdroj: Autor

Součástí těla bloku je čítač transakcí a transakce. Maximální počet transakcí v bloku je určen velikostí každé jednotlivé transakce a také velikostí bloku. Na obrázku 2 bude znázorněno, jak vypadá blok etherea v blockchain Exploreru.



Obrázek 2: Výpis dat o bloku na blockchain exploreru

Zdroj: Autor z Aplikace Blockchain Explorer

3 Smart contracts neboli chytré smlouvy

3.1 Úvod do chytrých smluv

Chytré smlouvy jsou autonomní počítačové programy, které automaticky vykonávají smluvní podmínky dohody. Tyto smluvní podmínky jsou vloženy do chytrých smluv, které jsou implementovány nad blockchainy. Schválené smluvní doložky jsou převedeny na spustitelné počítačové programy a provedení každého smluvního prohlášení je zaznamenáno jako neměnná transakce uložená v blockchainu. Hlavní výhodou chytrých smluv nasazených nad blockchainem spočívá v tom, že blockchain zaručuje, že podmínky smlouvy nelze změnit, což znemožňuje manipulaci se smluvními podmínkami nebo jejich hackování. Očekává se tedy, že chytré smlouvy přinesou snížení nákladů na ověřování, provádění, arbitráž a prevenci podvodů. Již zmíněný americký kryptograf Nick Szabo je považován za člověka, který v roce 1994 vytvořil koncept chytrých smluv a často se zmiňoval o příkladu pronajatého automobilu s chytrou smlouvou. Chytré smlouvy mohou být užitečné při překonání problému morálního hazardu a umožňují vývojářům přidělovat přístupová oprávnění pro každou funkci v kontraktu. Jakmile je splněna jakákoli podmínka v chytré smlouvě, spouštěný příkaz automaticky provede odpovídající funkci předvídatelným způsobem.

3.2 Návrh a funkčnost chytrých smluv

Při vytváření chytrých smluv se několik stran domlouvá na povinnostech, právech a zákazech ve smlouvách. Po několika kolech diskusí a vyjednávání může být dosaženo dohody. Právníci nebo poradci pomáhají stranám navrhnout počáteční smluvní dohodu. Softwaroví inženýři pak tuto dohodu napsanou v přirozených jazycích převedou na chytrou smlouvu napsanou v počítačových jazycích, včetně deklarativních jazyků a jazyků založených na logických pravidlech. Tento proces se podobně jako vývoj počítačového softwaru skládá z návrhu, implementace a testování. Je třeba poznamenat, že tvorba chytrých smluv je iterativní proces zahrnující několik kol jednání a iterací, na nichž se podílejí zúčastněné strany, právníci a softwaroví inženýři. Výsledná chytrá smlouva je digitálně podepsaná vypočitatelná dohoda mezi dvěma nebo více stranami. Virtuální třetí strana, softwarový agent, může provádět a vymáhat některé podmínky těchto smluv. V kontextu blockchainu, kde skutečně nabývá svého smyslu, je chytrá smlouva programem řízeným událostmi, se stavem, který běží na replikované sdílené účetní knize, a který může převzít péči o aktiva v této účetní knize. Chytré smlouvy v blockchainu jsou zcela digitální a jsou napsány pomocí programovacích jazyků. Tento

kód definuje pravidla a důsledky stejným způsobem jako tradiční právní dokument a uvádí povinnosti, výhody a sankce, které mohou za různých okolností náležet jedné ze stran. Velký rozdíl je v tom, že tento kód je automaticky prováděn systémem distribuované účetní knihy způsobem, který nelze odvolat a porušit. (Florian Idelberger, 2016)

Mezi charakteristické vlastnosti kódu chytrých smluv patří:

- deterministické: Kód chytré smlouvy musí být deterministický, aby všechny distribuované uzly vygenerovaly stejný výsledek při zadání vstupu, protože jej provádí mnoho distribuovaných uzlů současně. Z toho vyplývá, že kód chytré smlouvy by neměl být náhodný, neměl by být nezávislý na čase (v malém časovém rozmezí, protože kód může být v každém z uzlů proveden v trochu jiném čase) a měl by být schopen být proveden mnohokrát,
- neměnnost: Kód chytrých smluv je neměnný. To znamená, že jej po nasazení nelze měnit. To je nepochybně výhodné z hlediska důvěryhodnosti, ale zároveň to přináší určité obtíže (například jak opravit chybu v kódu) a naznačuje to, že kód chytrých smluv vyžaduje větší kontrolu a dohled,
- ověřitelné: Při spuštění chytrého kontraktu obdrží speciální adresu. Zájemci mohou a měli by si kód před použitím chytré smlouvy prohlédnout nebo zkontrolovat.

3.2.1 Nasazení chytrých smluv

Při implementaci chytrých smluv je prvním krokem ověření kódu smlouvy před jejím nasazením na blockchain platformu. Kontrakty jsou udržovány na blockchainech a díky jejich neměnnosti je nelze měnit. V případě potřeby změn je proto nutné vytvořit nový kontrakt. Jakmile je chytrá smlouva nasazena, mají k ní všechny strany přístup prostřednictvím blockchainu a jejich digitální aktiva jsou uzamčena zmrazením příslušných digitálních peněženek. To zahrnuje zabránění jakýmkoli příchozím nebo odchozím převodům kryptoměn do peněženek spojených s kontraktem, zatímco strany jsou ověřovány prostřednictvím svých digitálních peněženek.

K nasazení chytré smlouvy lze použít různé nástroje, jako jsou platformy pro vývoj blockchainu, integrovaná vývojová prostředí (IDE) založená na blockchainu nebo nástroje příkazového řádku.

Jakmile je chytrý kontrakt nasazen, může s ním kdokoli interagovat pomocí různých metod, jako je odesílání transakcí na adresu kontraktu, volání jeho funkcí nebo interakce s ním prostřednictvím uživatelského rozhraní. (Bashir, 2020)

3.2.2 Provádění chytrých smluv

Po implementaci chytrých smluv jsou jejich ustanovení sledována a vyhodnocována. Smluvní procesy nebo závazky se automaticky provádějí, jakmile jsou splněna předem stanovená kritéria, například přijetí produktu. Deklarativní příkazy s logickými vazbami tvoří chytré smlouvy, což znamená, že při splnění podmínky se provede související příkaz, což vede k provedení transakce, která je následně ověřena těžaři blockchainu. Následné transakce a změněné stavy jsou pak trvale uloženy v blockchainu. (Bashir, 2020)

3.2.3 Dokončení chytrých smluv

Digitální záznamy všech zúčastněných stran jsou po úspěšném provedení chytré smlouvy aktualizovány tak, aby odrážely nový stav dohody. Tyto dokumenty jsou uloženy v blockchainu pro budoucí použití spolu s konkrétními transakcemi, které proběhly během provádění smlouvy. V rámci této transakce dochází k převodu digitálních aktiv z jedné strany na druhou, podobně jako když zákazník platí poskytovateli za obdržené produkty nebo služby. Po převodu jsou aktiva odblokována a chytrá smlouva dokončí svůj životní cyklus.

3.3 Problémy a výzvy při vývoji chytrých smluv

Chytré smlouvy jsou slibnou technologií, ale stále existuje mnoho překážek. Podle čtyř fází životního cyklu chytrých smluv rozdělujeme tyto významné problémy do čtyř typů.

3.3.1 Překážky při vytváření

Proces implementace chytrých smluv zahrnuje vytvoření smluv. Vzhledem k tomu, že blockchainy jsou prakticky neměnné, nelze chytré smlouvy vytvořené na blockchainech měnit ani po jejich implementaci. Vývojáři proto musí řádně řešit následující otázky.

3.3.1.1 Čitelnost

K vytváření většiny chytrých kontraktů se používají programovací jazyky jako Solidity, Go, Kotlin a Java. Po dokončení psaní kódu se zdrojový kód zkompiluje a spustí. Chytré smlouvy proto mají různé druhy kódování v závislosti na době jelikož chytré smlouvy trpí tím, že jsou napsané v aktuálním stylu dané verze programovacího

jazyka a jako například u Solidity se můžeme setkat s tím, že v každé nové verzi se upravuje syntaxi či sémantiku kódu. Stále je obtížné vytvořit programy čitelné v každé podobě.

3.3.1.2 Problémy s fungováním

U stávajících platforem chytrých smluv existuje řada funkčních problémů:

- re-entrancy znamená, že přerušenou funkci lze znovu zavolat, tento typ útoku umožňuje útočníkovi opakovaně volat funkci smlouvy způsobem, který vyčerpává její prostředky, což vede k útoku typu DoS (denial of service neboli odepření služby),
- bloková náhodnost pro některé aplikace chytrých smluv, jako jsou loterie a sázkové chytré kontrakty, mohou vyžadovat náhodnost generovaných bloků. Toho lze dosáhnout generováním pseudonáhodných čísel v časovém razítku bloku nebo nonce. Bezpečnostní problém vzniká v tu chvíli, kdy těžaři vytvoří bloky, které se odchylují od výsledků generátoru, což má za následek, že dokážeme ovládat rozložení pravděpodobnosti výsledků (Bonneau, Clark, & Goldfeder, 2015),
- limit exceed je, když chytrá smlouva provede aritmetický výpočet, ale výsledek překročí limit úložiště. To může mít za následek nesprávný výpočet částek.

Jako bezpečností řešení můžeme navrhnout používání nástrojů na kontrolu bezpečnosti chytrých kontraktů jako jsou Gasper, Oyente nebo Solgrapha a další.

3.3.2 Limity nasazení

Po vytvoření budou chytré smlouvy nasazeny na blockchainové platformy. Chytré smlouvy je však třeba pečlivě kontrolovat, aby se předešlo případným chybám. Kromě toho vývojáři chytrých smluv si musí být vědomi interakčních vzorců smlouvy, aby zmírnili dopady potenciální ztráty způsobené škodlivým chováním jako jsou podvody a útoky.

3.3.2.1 Přesnost smlouvy

Jakmile jsou chytré smlouvy nasazeny na blockchainech, je téměř nemožné provádět jakékoli revize. Proto je velmi zásadní vyhodnotit správnost chytrých smluv ještě předtím, než jsou nasazeny. Je však náročné ověřit správnost chytrých smluv kvůli složitosti modelování samotných chytrých smluv.

Řešení, která se nabízí jsou:

- **analýza byte kódu** je možnost, která nás přivádí k již zmíněnému nástroji OYENTE, kde tvůrci slibují, že díky analýze byte kódu je program schopný identifikovat potenciální bezpečnostní chyby, včetně chybně zpracovaných výjimek a problémů závislých na časovém razítku,
- **analýza zdrojového kódu** ve srovnání s analýzou byte kódu vyžaduje analýza zdrojového kódu dostupnost zdrojových kódů chytrých smluv. I když analýza zdrojového kódu nám dá více informací než předcházející analýza, je zde podmínkou přesně nastavená analýza, jinak informace, které dostaneme, mohou být bezcenné,
- **analýza založená na strojovém učení** nám umožnuje dosáhnout lepších výsledků v odhalování zranitelnosti chytrých smluv. Ve své práci H. Liu s kolegy navrhl novou techniku bezpečnostního auditu s ohledem na sémantiku nazvanou S-gram pro Ethereum. Schéma S-gram, které je kombinací N-gram modelů a statického sémantického značení, lze použít k předvídání potenciálních zranitelností pomocí identifikace nepravidelných sekvencí tokenů a optimalizace stávajících hloubkových analyzátorů (Liu, Liu, Zhao, Jiang, & Sun, 2018).

3.3.2.2 Tok dynamického řízení

Není jisté, že tok řízení chytrých smluv je neměnný, i když nasazené chytré smlouvy jsou neměnné. Zejména chytrá smlouva může komunikovat s jinými smlouvami (např. převodem finančních prostředků do smlouvy nebo vytvořením nové smlouvy). Při vytváření chytré smlouvy je třeba dbát na její tok řízení. Interakce chytrých smluv může časem vést k nárůstu počtu propojených smluv. V důsledku toho je obtížné předvídat chování smluv.

3.3.3 Problémy se spuštěním smluv

Fáze provádění chytrých smluv je zásadní, protože určuje jejich konečný stav. Během provádění chytrých smluv je třeba zvládnout řadu problémů.

3.3.3.1 Oracle

Oracles jsou entity, které podepisují tvrzení o stavu světa a jsou důvěryhodnými zdroji informací pro chytré smlouvy. Umožňují chytrým smlouvám reagovat na vnější nedeterministické události. Věštci jsou nezbytní pro připojení chytrých smluv ke

kritickým datovým zdrojům, webovým rozhraním API a platebním metodám. Chytré smlouvy lze snadno spustit, pokud jsou podmínky spuštění spojeny se záznamy v blockchainu nebo jsou jednoduchými časovými značkami. Pokud jsou však podmínky provedení mimo blockchain, je zapotřebí oracle. Oracles mohou být určeny jako důvěryhodné třetí strany, odkazovat na důvěryhodné databáze nebo využívat decentralizované oracle služby, kde účastníci hlasují o přesném výsledku. Oracles fungují jako prostředníci mezi daty z vnějšího světa nebo rozhraními API a chytrými smlouvami (Bashir, 2020).

3.3.3.2 Závislost na pořadí transakcí

Chytré smlouvy mohou být ovlivněny závislostí na pořadí transakcí (TOD). TOD může u chytrých smluv nastat, když pořadí, v jakém jsou transakce zpracovávány, ovlivňuje výsledek smlouvy. Jedním z příkladů TOD v chytré smlouvě je, když dvě transakce soutěží o změnu stejné stavové proměnné. V závislosti na pořadí, v jakém jsou transakce zpracovány, může být konečný stav proměnné odlišný. To může vést k nekonzistentním výsledkům a vytvořit bezpečnostní zranitelnosti kontraktu.

Existují však způsoby, jak TOD v chytrých kontraktech řešit. Jedním z přístupů je použití technik, jako je časová značka nebo použití specifického pořadí operací, které zajistí, že transakce budou zpracovány ve správném pořadí. Dalším přístupem je použití konsensuálních algoritmů, jako je proof of work nebo proof of stake, které mohou pomoci zabránit útokům na dvojité utrácení a zajistit, že transakce budou zpracovány v deterministickém pořadí. (Narayanan, Bonneau, Felten, Miller, & Goldfeder, 2016)

3.3.3.3 Efektivita provádění

Efektivita provádění chytrých smluv je zásadním faktorem, jelikož po nasazení smlouvy se provádějí bez lidského zásahu. Níže je několik klíčových faktorů, které mohou ovlivnit efektivitu provádění chytrých smluv

Poplatky za Gas: Jedním z hlavních faktorů ovlivňujících efektivitu provádění chytrých smluv jsou poplatky za Gas. Gas je měrná jednotka, která se používá k určení nákladů na provedení chytré smlouvy v blockchainu Ethereum. Čím vyšší jsou poplatky za plyn, tím dražší je provedení smlouvy.

Návrh kontraktu: Důležitým faktorem určujícím efektivitu je také návrh chytrého kontraktu. Dobře navržené kontrakty mohou snížit spotřebu plynu a zlepšit celkový výkon. Například smlouvy, které používají efektivnější datové struktury, nebo které dávkují transakce dohromady, mohou snížit náklady na plyn a zlepšit výkon.

Přetížení sítě: Přetížení sítě může také ovlivnit efektivitu provádění chytrých smluv. Když je síť vytížená, může zpracování transakcí a provádění smluv trvat déle. Vývojáři to mohou zmírnit výběrem správného času pro provádění kontraktů nebo použitím škálovacích řešení, jako jsou protokoly druhé vrstvy.

Hardwarová omezení: Efektivitu provádění chytrých smluv mohou ovlivnit také hardwarová omezení. Chytré smlouvy vyžadují k provádění výpočetní výkon, takže omezení hardwaru nebo infrastruktury podporující blockchain síť mohou mít vliv na výkon. Vývojáři mohou optimalizovat využití hardwaru výběrem správné blockchain infrastruktury nebo použitím specializovaného hardwaru.

Celkově je efektivita provádění chytrých smluv složité téma s mnoha faktory, které je třeba zvážit. Optimalizací využití plynu, návrhem efektivních kontraktů, řízením přetížení sítě a využitím hardwarových optimalizací mohou vývojáři zvýšit výkonnost provádění chytrých kontraktů.

3.4 Platformy pro chytré smlouvy

V poslední době se k tvorbě chytrých smluv používají platformy založené na blockchainu. Tyto platformy poskytují programátorům jednoduchá rozhraní pro vytváření aplikací pro chytré smlouvy. Mnohé ze stávajících blockchain systémů mohou podporovat chytré smlouvy. Existuje jich celá řada. Nejrozšířenější platformy pro chytré smlouvy: Ethereum, Hyperledger Fabric, Stellar a Rootstock. Klíčovým důvodem, je jejich popularita v rostoucí komunitě a s tím související technologická vyspělost.

3.4.1 Ethereum

Ethereum je decentralizovaná blockchainová platforma, která umožňuje vývojářům vytvářet a nasazovat chytré smlouvy. Byla spuštěna v roce 2015 a je nejrozšířenější platformou pro vytváření decentralizovaných aplikací (dapps). Programovacím jazykem Etherea je Solidity, který je podobný JavaScriptu, a používá mechanismus konsensu zvaný Proof of Work (PoW), ačkoli s nadcházející aktualizací Etherea 2.0 přechází na Proof of Stake (PoS). (Ethereum-community, 2023)

3.4.2 Hyperledger Fabric

Hyperledger Fabric je blockchainová platforma s povolením, která je určena pro podnikové použití. Je vyvíjena projektem Hyperledger nadace Linux Foundation a poskytuje modulární architekturu, která organizacím umožňuje přizpůsobit si blockchainové sítě na základě jejich specifických potřeb. Mezi programovací jazyky Hyperledger

Fabric patří Go, Java a Node.js a používá mechanismus konsensu založený na hlasovacím systému. (Hyperledger, 2023)

3.4.3 Stellar

Stellar je decentralizovaná blockchainová platforma, která se zaměřuje na přeshraniční platby a převody aktiv. Používá mechanismus konsensu nazvaný Stellar Consensus Protocol (SCP), který umožňuje rychlé a levné transakce. Programovacím jazykem Stellaru je Stellar Transaction Language (STL), který je podobný skriptovacímu jazyku Bitcoinu. (Stellar, 2023)

3.4.4 Rootstock (RSK)

Rootstock je platforma pro chytré smlouvy, která je postavena nad blockchainem Bitcoinu. Používá obousměrný mechanismus peggingu, který umožňuje převádět Bitcoin na blockchain Rootstock a zpět. Programovacím jazykem Rootstocku je Solidity, stejný jazyk, který používá Ethereum, a používá mechanismus konsensu zvaný Merge-Mining. (Rootstock, 2023)

3.4.5 Binance Smart Chain (BSC)

Binance Smart Chain je platforma chytrých kontraktů vyvinutá společností Binance, jednou z největších kryptoměnových burz. Je kompatibilní s virtuálním strojem Ethereum (EVM) a podporuje chytré kontrakty založené na Ethereu. BSC používá mechanismus konsensu nazvaný Proof of Staked Authority (PoSA), který je hybridem Proof of Stake (PoS) a Proof of Authority (PoA). (Binance, 2023)

3.4.6 TRON

TRON je decentralizovaná blockchainová platforma, která se zaměřuje na distribuci obsahu a zábavu. Používá mechanismus konsensu zvaný Delegated Proof of Stake (DPoS), který umožňuje rychlé a škálovatelné transakce. Programovacím jazykem společnosti TRON je Solidity a podporuje chytré smlouvy založené na platformě Ethereum. (DAO, 2018)

Každá z těchto platforem má své silné a slabé stránky a výběr platformy bude záviset na konkrétních požadavcích projektu.

3.5 Potencionální využití chytrých smluv

3.5.1 Internet věcí neboli IOT

Chytré smlouvy lze v internetu věcí použít k umožnění autonomních transakcí mezi zařízeními bez nutnosti lidského prostředníka. To by mohlo zásadně změnit způsob,

jakým spolu zařízení internetu věcí komunikují, a zvýšit jejich efektivitu a bezpečnost. Chytré smlouvy lze například použít k automatizaci procesu vyúčtování spotřeby energie mezi zařízeními inteligentní domácnosti. Domácnosti by tak již nemusely potřebovat centralizovaného dodavatele energie, což by jim ušetřilo peníze a snížilo uhlíkovou stopu. Použití chytrých smluv v logistice a řízení dodavatelského řetězce umožňuje automatické platby a sledování v reálném čase mezi zařízeními internetu věcí.

3.5.2 Zabezpečení distribuovaných systémů

Chytré kontrakty mohou zvýšit bezpečnost distribuovaných systémů tím, že umožňují bezpečné a nezměnitelné uchovávání záznamů. To je možné díky neměnnosti technologie blockchain, která zajišťuje, že jakmile je transakce jednou přidána do blockchainu, nelze ji změnit ani odstranit. Automatizací ověřování a autentizace transakcí mohou chytré smlouvy pomoci předcházet podvodným činnostem. To má zásadní význam zejména pro odvětví, jako je finančnictví a zdravotnictví, kde jsou bezpečnost a ochrana soukromí nejdůležitější.

3.5.3 Finance a právo

Pomocí chytrých smluv lze omezit využívání prostředníků, jako jsou banky nebo advokáti, a urychlit tak finanční transakce. To může snížit možnost chyb nebo podvodů a zároveň ušetřit čas a peníze. Chytré smlouvy lze použít například k automatickému provádění obchodů nebo rozhodování sporů mezi stranami. To může snížit náklady a složitost transakce tím, že odpadne nutnost clearingového střediska třetí strany. Pomocí chytrých smluv lze také provádět služby úschovy, při nichž jsou peníze uchovávány v chytré smlouvě, dokud nejsou splněna určitá kritéria. (Peters & Panayi, 2015)

3.5.4 Ověřování a potvrzení původu dat

Chytré smlouvy lze použít ke sledování vlastnictví a původu dat, čímž se zajistí jejich původ a pravost. To má zásadní význam zejména v odvětvích, jako je zdravotnictví a řízení dodavatelského řetězce, kde je bezpečnost dat a ochrana soukromí nesmírně důležitá. Organizace mohou pomocí chytrých smluv zaručit, že k datům mohou přistupovat a využívat je pouze oprávněné strany. Pro jednodušší sledování toku dat mezi různými systémy a organizacemi lze chytré smlouvy využít také k vytvoření auditovatelné stopy vlastnictví dat. (Deloitte & Alliance, 2016)

3.5.5 Sdílená ekonomika

Sdílená ekonomika díky půjčování a recyklaci zboží snižuje spotřebitelské náklady a dopady na životní prostředí, zároveň však zvyšuje účinnost zdrojů a kvalitu služeb. Většina současných platforem sdílené ekonomiky však vede k vysokým transakčním nákladům, ohrožení soukromí a nespolehlivosti důvěryhodných třetích stran pro samotného spotřebitele. Decentralizací platforem sdílené ekonomiky mohou chytré smlouvy potenciálně transformovat ekonomiku sdílení. Inovativní platformu sdílené ekonomiky postavenou na chytrých smlouvách Ethereum navrhli Bogner a kol (Bogner, Chanson, & Meeuw, 2016). Uživatelé se mohou registrovat a sdílet své věci právě prostřednictvím tohoto systému bez využití důvěryhodné třetí strany. Osobní údaje jsou chráněny i z hlediska ochrany soukromí. Účinnost systému potvrzuje i jeho skutečné použití.

3.5.6 Veřejný sektor

Vládní postupy, jako je hlasování a registrace majetku, lze pomocí chytrých smluv zefektivnit a automatizovat. Tímto způsobem lze snížit počet podvodů a zprůhlednit fungování státní správy. Chytré smlouvy lze například použít k automatizaci procesu hlasování a zaručit, že hlasy budou bezpečně a přesně zaznamenány. K zajištění přesné a transparentní evidence vlastnictví nemovitostí lze chytré smlouvy využít také při registraci nemovitostí. (Deloitte & Alliance, 2016) (OECD, 2022)

4 Internet of Things

4.1 Definice internetu věcí (IoT)

V posledních letech roste zájem o internet věcí (IoT), což je pojem, který označuje síť fyzických objektů vybavených senzory, softwarem a dalšími technologiemi, které jim umožňují shromažďovat a vyměňovat si data mezi sebou a s dalšími zařízeními prostřednictvím internetu. Termín "internet věcí" poprvé použil Kevin Ashton, spoluzakladatel Auto-ID Center na MIT, v prezentaci pro společnost Procter & Gamble v roce 1999 (Ashton, 2009). Od té doby se internet věcí vyvinul a zahrnuje širokou škálu objektů a zařízení, od jednoduchých senzorů a domácích spotřebičů, až po složitá průmyslová zařízení a autonomní vozidla.

Ve své podstatě je internet věcí ekosystém, který propojuje lidi, zařízení a internet a vytváří síť inteligentních objektů, které spolu mohou komunikovat a autonomně se rozhodovat na základě shromážděných dat (Perera, Zaslavsky, Christen, & Georakopoulos, 2014). Internet věcí je popisován jako klíčová součást Průmyslu 4.0, což je současný trend automatizace a výměny dat ve výrobních technologiích. Potenciál využití internetu věcí je obrovský, od inteligentních domů a měst až po zdravotnictví, zemědělství, dopravu a další (A. Al-Fuqaha, 2015).

V roce 2020 se na celém světě používalo více než 50 miliard zařízení internetu věcí, která generovala obrovské množství dat, jejichž objem by měl v roce 2021 dosáhnout 4,4 zettabytu. Významný je také finanční dopad internetu věcí, podle prognóz by se měl trh do roku 2025 pohybovat v rozmezí 1,6 až 14,4 bilionu dolarů, což ovlivní téměř všechna odvětví ekonomiky (A. Al-Fuqaha, 2015). Internet věcí změnil způsob, jakým komunikujeme s naším okolím, a umožnil novou úroveň propojení a sdílení dat, která má potenciál způsobit revoluci v průmyslových odvětvích a celé společnosti.

4.2 Technologie, které stojí za internetem věcí

Termín "internet věcí" (IoT) označuje obrovskou síť propojených objektů, které spolu komunikují online. Tato zařízení se běžně označují jako "věci" a mohou zahrnovat širokou škálu objektů, jako jsou senzory, chytré spotřebiče, vozidla a další. Hlavním cílem zařízení IoT je shromažďovat data z fyzického prostředí a využívat je k poskytování cenných poznatků a služeb.

4.2.1 Rozpoznávání

Je první charakteristickou vlastností zařízení internetu věcí. V síti musí být každé zařízení IoT individuálně rozpoznáno. K rozpoznaní zařízení v síti se používá technika jedinečných IP adres, kterou bychom mohly rozdělit na IPV4 a IPV6. Zpočátku se pro adresování využívala IPV4, ale s přibývajícími položkami se nyní používá 128bitová technika adresování IPV6.

4.2.2 Komunikace a vnímání

Dalším důležitým atributem je snímání, které zahrnuje sběr dat z fyzického prostředí pomocí různých snímacích zařízení jako jsou akční členy (akutátory), inteligentní senzory a tag RFID. Dále je klíčová také komunikace, která umožňuje zařízením odesílat a přijímat data ve formě dat, zpráv či souborů pomocí technologií, jako je Bluetooth, bezdrátové sítě, RFID a další. (Burhan, Rehman, Kim, & Khan, 2018)

4.2.3 Výpočet

Výpočet hraje zásadní roli při zpracování a analýze dat získaných ze zařízení internetu věcí. Jedná se o využití senzorů k provádění výpočtů a eliminaci nepotřebných nebo nadbytečných dat. K provádění výpočtů nad shromážděnými daty se používá mnoho hardwarových a softwarových platforem, včetně Audrino, Raspberry Pi, Intel Galileo, Tiny OS, Lite OS a Android. Jak uvádí Burhan, výpočty jsou nezbytné k odstranění nadbytečných nebo nepotřebných dat a zajištění efektivního fungování aplikací internetu věcí. (Burhan, Rehman, Kim, & Khan, 2018)

4.2.4 Služby

Jednou ze základních složek internetu věcí jsou služby, které označují funkce zařízení, jež jsou poskytovány uživatelům na základě informací, které dostávají. Můžeme služby rozdělit na čtyři typy služeb poskytovaných aplikacemi internetu věcí. První služba se týká identity a pomáhá získat identitu objektů, které odeslaly požadavek. Druhou službou je agregace informací, která shromažďuje všechny informace od objektů, zatímco třetí je služba spolupráce, která na základě shromážděných informací činí rozhodnutí a odesílá zařízením příslušné odpovědi. Poslední služba je všudypřítomná, slouží k odpovídání zařízením bez rigidity, pokud jde o čas a místo. (Grammatikis, Sarigiannidis, & Moscholios, 2019)

4.2.5 Sémantika

Další a poslední složkou internetu věcí je sémantika, která se týká schopnosti zařízení internetu věcí získávat správné informace ze svého fyzického prostředí a poskytovat tyto informace jako službu ve vhodnou dobu nebo v případě potřeby. Sémantika je posledním atributem zařízení IoT a je zodpovědná za usnadnění uživatelům při plnění jejich úkolů. Jinými slovy, sémantika funguje jako mozek IoT, který shromažďuje všechny informace a přijímá vhodná rozhodnutí pro zasílání odpovědí zařízením.

4.3 Zařízení a softwarové komponenty

4.3.1 Senzory

Senzory jsou zařízení, která detekují fyzikální změny v prostředí a převádějí je na elektrické signály. Senzory mohou být aktivní nebo pasivní a analogové nebo digitální. V zařízeních IoT se používají různé typy snímačů: snímače teploty, snímače vlhkosti, snímače tlaku, snímače přiblížení, snímače hladiny, akcelerometry, gyroskopy, snímače plynů, infračervené snímače a optické snímače. Využívání senzorů a IoT systému snižuje provozní náklady a zvyšuje efektivitu a bezpečnost pracovníků.

4.3.2 Akční členy (aktuátor)

Akční členy jsou zařízení, která reagují na data přijatá ze senzorů a provádějí akci. Příklady aktuátorů používaných v zařízeních internetu věcí zahrnují motory, servopohony, relé a solenoidy.

4.3.3 Mikrokontrolery

Mikrokontrolery jsou malé počítače, které jsou zabudovány v zařízeních internetu věcí a jsou zodpovědné za řízení provozu zařízení. Obvykle mají procesor, paměť, vstupní/výstupní rozhraní a komunikační rozhraní.

4.3.4 Komunikační moduly

Komunikační moduly jsou zařízení, která umožňují zařízení IoT připojit se k síti a komunikovat s ostatními zařízeními. Mezi komunikační moduly patří například moduly Wi-Fi, moduly Bluetooth a mobilní moduly

4.3.5 Zdroje napájení

Zdroje napájení jsou zařízení, která dodávají energii zařízení internetu věcí. Mohou to být baterie, solární panely a napájecí adaptéry.

4.3.6 Software

Software je kód, který běží na mikrokontroleru a umožňuje zařízení provádět různé úlohy. Může zahrnovat firmware, operační systémy a aplikační software.

4.3.7 Připojitelnost zařízení

Konektivita je důležitou součástí zařízení internetu věcí, protože jim umožňuje komunikovat s ostatními zařízeními a sítěmi. Zařízení internetu věcí mohou využívat různé typy připojení, včetně Wi-Fi, Bluetooth, mobilních a satelitních sítí. Typ připojení používaný zařízením IoT závisí na různých faktorech, jako je dosah, šířka pásma a požadavky na napájení zařízení. Hlavním způsobem připojení zařízení IoT je připojení přes internet. Tyto připojovací sítě jsou škálovatelné v závislosti na velikosti a rozsahu systému IoT. Sítě sahají od LAN (Local Area Network), což je skupina zařízení, která jsou propojena na jednom fyzickém místě, až po MAN (Metropolitan Area Network). Dále WAN (Wide Area Network) je v podstatě síť sítí, přičemž internet slouží jako největší WAN na světě. V současné době existuje několik typů sítí WAN, z nichž každá je určena pro specifický případ použití, který se dotýká téměř všech aspektů moderního života. Zařízení IoT mohou využívat různé typy připojení, včetně Bluetooth, LoraWan, Z-wave, NFC, WiFi, Cellular a Zigbee. Účelem těchto sítí je obvykle umožnit přenos dat nebo informací mezi zařízeními nebo servery. Většinu vývoje v oblasti osobních sítí má na starosti pracovní skupina IEEE 802.15.

4.3.8 Umělá inteligence (AI)

Přestože internet věcí (IoT) i umělá inteligence (AI) jsou samy o sobě silnými a produktivními technologiemi, jejich kombinace přináší ještě vyšší efektivitu. Umělá inteligence je termín, který se používá k popisu systému, jenž dokáže inteligentním způsobem provádět soubor úkolů nebo se učit z dat. Základní myšlenkou konceptu IoT je spojení technologií AI a IoT a vytvoření stroje, který dokáže vyhodnocovat data a rozhodovat se bez zásahu člověka.

Kromě toho se AI široce používá v zařízeních IoT k provádění složitých úkolů, jako je rozhodování, detekce anomálií a prediktivní údržba. Data ze snímačů IoT mohou být analyzována algoritmy AI, aby se získaly poznatky, které lze využít ke zlepšení provozu zařízení nebo systémů. (Ganne, 2023)

4.3.9 Fungování IoT

Internet věcí (IoT) způsobil revoluci ve způsobu komunikace a práce tím, že umožňuje propojení mnoha zařízení prostřednictvím internetu, což usnadňuje interakci mezi lidmi i stroji. To otevřelo nové možnosti pro osobní i podnikové aplikace a přijetí nových technologií, jako je 5G a Li-Fi, jeho možnosti dále rozšíří. Fungování systémů internetu věcí je poměrně jednoduché, kdy jsou zařízení s vestavěnými senzory připojena k platformám internetu věcí, které shromažďují a integrují data z různých zařízení a aplikují analytiku k identifikaci užitečných informací. Tyto informace pak mohou být využity k různým účelům, včetně detekce vzorů, doporučení, odhalování problémů a inteligentního rozhodování (A. Al-Fuqaha, 2015).

Příkladem IoT řešení je firma se zaměřením na výrobu sportovním vybavení, kde senzory fyzické či algoritmy v síti mohou zjistit, které oblasti jsou nejoblíbenější, což umožní majiteli firmy upravit podle toho svou obchodní strategii. Ekosystémy internetu věcí se neomezují jen na určitá odvětví ekonomiky, mají všestranné využití v automatizaci domácností a vozidel, ve výrobě, zdravotnictví, maloobchodě, obraně, finančnictví a dalších oblastech. Systémy IoT mohou navíc využívat umělou inteligenci a strojové učení ke zlepšení sběru a analýzy dat.

4.4 Architektura internetu věcí

Architektura internetu věcí se skládá ze čtyř klíčových vrstev, kterými jsou vrstva vnímání, síťová vrstva, vrstva middlewaru a aplikační vrstva. Každá vrstva má svůj vlastní soubor funkcí a hraje zásadní roli v celkové architektuře internetu věcí. (Simone Cirani, 2018)

4.4.1 Vnímací vrstva

Vrstva vnímání je první vrstvou architektury IoT a je zodpovědná za sběr dat z fyzického prostředí. Skládá se ze snímačů, akčních členů a řídicích jednotek, které zachycují, zpracovávají a přenášejí data do síťové vrstvy. Senzory jsou zodpovědné za detekci fyzikálních změn, jako je teplota, vlhkost, světlo a pohyb, a za přenos dat do aktuátorů. Akční členy zase reagují na data přijatá ze snímačů prováděním akcí, jako je rozsvícení světla, otevření dveří nebo nastavení teploty. Řídicí jednotky fungují jako prostředníci mezi snímači a akčními členy a poskytují prostředky pro programování a řízení jejich chování.

4.4.2 Síťová vrstva

Síťová vrstva je zodpovědná za přenos dat přijatých z vrstvy vnímání do vrstvy middlewaru. Skládá se z různých komunikačních technologií, jako jsou Wi-Fi, Bluetooth a Zigbee, které umožňují zařízením se propojit a komunikovat mezi sebou. Síťová vrstva je také zodpovědná za řízení toku dat a zajištění bezpečného a efektivního přenosu dat.

4.4.3 Vrstva middlewaru

Vrstva middlewaru je třetí vrstvou architektury IoT a funguje jako most mezi síťovou a aplikační vrstvou. Je zodpovědná za zpracování a správu dat přijatých ze síťové vrstvy a jejich zpřístupnění aplikační vrstvě. Vrstva middlewaru také poskytuje služby, jako je zabezpečení, ukládání dat a analýza dat.

4.4.4 Aplikační vrstva

Aplikační vrstva je nejvyšší vrstvou architektury IoT a je zodpovědná za poskytování prostředků koncovému uživateli pro interakci se zařízeními a jimi shromážděnými daty. Aplikační vrstva se skládá z různých aplikací, jako jsou mobilní aplikace, webové aplikace a desktopové aplikace, které umožňují uživatelům ovládat a monitorovat zařízení, analyzovat data a provádět různé úkoly. (Simone Cirani, 2018)

4.4.5 Transportní vrstva

Transportní vrstva je zodpovědná za řízení přenosu dat mezi síťovou vrstvou a vrstvou zpracování. Tato vrstva se často používá k zajištění bezpečného a efektivního přenosu dat. Mezi běžné protokoly používané v transportní vrstvě patří TCP (Transmission Control Protocol) a UDP (User Datagram Protocol). (Simone Cirani, 2018)

4.4.6 Vrstva zpracování

Vrstva zpracování je zodpovědná za zpracování dat přijatých ze síťové vrstvy a jejich transformaci do formátu použitelného pro aplikační vrstvu. Tato vrstva může také provádět filtrování, agregaci a analýzu dat. Vrstva zpracování může využívat různé technologie, jako je analýza velkých objemů dat, strojové učení a umělá inteligence, aby z dat získala poznatky.

4.4.7 Business vrstva

Obchodní vrstva je vrstvou architektury IoT, která je neblíže uživatelovy a je zodpovědná za správu obchodní logiky a pravidel, kterými se řídí provoz systému IoT. Tato vrstva může zahrnovat různé komponenty, jako jsou řídicí panely, pracovní postupy

a nástroje business intelligence, které koncovým uživatelům umožňují interakci se systémem a přijímání rozhodnutí. (Burhan, Rehman, Kim, & Khan, 2018)

Tímto bychom zmínily čtyři klíčové vrstvy IoT architektury, se kterýma se setkáme asi nejčastěji, což jsou vrstva vnímací, síťová, middleware a aplikační. Avšak stojí za zmínku, že počet a složení vrstev v architektuře IoT se může lišit v závislosti na konkrétním případu použití a požadavcích systému. Některé systémy IoT mohou mít jednodušší architekturu s menším počtem vrstev, zatímco jiné mohou mít složitější architekturu s více vrstvami.

4.5 Sítě a protokoly internetu věcí

4.5.1 Internet Protocol version 6 (IPv6)

Tento protokol je nejpokročilejší a nejmodernější protokol pro síťovou vrstvu internetu. Poskytuje velký adresní prostor, aby vyhovoval rostoucímu počtu zařízení internetu věcí, a je navržen tak, aby podporoval různé typy zařízení a komunikačních technologií. Vývojáři navrhli protokol IPv6 tak, aby řešil různé problémy současné verze sady internetových protokolů (IPv4), jako je vyčerpání, bezpečnost, automatická konfigurace, rozšiřitelnost a škálovatelnost. Díky svým rozšířeným schopnostem umožňuje protokol IPv6 používat nové druhy technologií, například internet věcí.

4.5.2 Radio Frequency Identification (RFID)

Radiofrekvenční identifikace (RFID) je jednou z hlavních technologií internetu věcí a jedná se o bezdrátovou komunikační technologii, která využívá rádiové vlny k identifikaci a sledování objektů.

Technologie RFID se skládá ze dvou složek: RFID štítků a RFID čteček. Štítek RFID je zařízení připevněné k objektu, který chceme sledovat nebo o němž chceme shromažďovat údaje, a může být tří typů: pasivní, poloaktivní a aktivní. Pasivní tagy RFID získávají energii úpravou elektromagnetické rádiové vlny, kterou vysílá čtečka RFID, když se jí dotazuje na data, zatímco poloaktivní tag má malý zdroj energie a získává energii z jiných zdrojů, které doplňují jeho omezený zdroj energie. Aktivní tagy RFID mají naproti tomu vestavěný zdroj energie, který napájí jejich mikročip a senzory.

Pasivní RFID štítky jsou pro internet věcí vhodnější, protože své požadavky na spotřebu energie plní z jiných zdrojů.

4.5.3 Wireless Sensor Networks (WSN)

Bezdrátové senzorové sítě (WSN) jsou sítě vzájemně propojených senzorů, které se používají ke sledování fyzikálních podmínek. Komunikují bezdrátově a používají se v průmyslových aplikacích, aplikacích pro ochranu životního prostředí a ve zdravotnictví. Shromážděná data jsou organizována v centrálních bodech neboli sinks, které zpracovávají všechna příchozí data, zpracovávají je a odesílají zpět k analýze. (Pinar, et al., 2016)

4.5.4 Constrained Application Protocol (CoAP)

Protokol CoAP (Constrained Application Protocol) je lehký internetový protokol určený pro zařízení internetu věcí s omezenými zdroji. Umožňuje snadné připojení přes omezené sítě s omezenou šířkou pásma pomocí protokolu UDP a klidové architektury podobné protokolu HTTP. CoAP se používá pro komunikaci mezi stroji (Machine to machine dále už M2M) a poskytuje jednoduchý a efektivní způsob výměny dat mezi zařízeními IoT. (Choi & Koh, 2016)

4.5.5 Message Queue Telemetry Transport Protocol (MQTT)

MQTT je protokol pro zasílání zpráv pro systémy IoT, který usnadňuje komunikaci mezi zařízeními s omezenými zdroji. Byl vyvinut pro komunikaci M2M a vzdálené sledování a shromažďuje data z různých zařízení. MQTT se skládá ze tří hlavních komponent: účastníka, vydavatele a prodejce, které zajišťují bezpečný a spolehlivý přenos informací.

4.5.6 Data Distribution Service (DDS)

DDS je flexibilní a škálovatelný protokol, který poskytuje model publish-subscribe pro distribuované systémy reálného času. Nabízí funkce, jako je správa QoS, dynamické zjišťování a automatické mapování datových typů, což usnadňuje vývoj a nasazení distribuovaných aplikací. DDS se běžně používá v kritických odvětvích, jako je zdravotnictví, doprava, energetika a obrana, a poskytuje výhody, jako je vyšší výkonnost systému, kratší doba vývoje a vyšší škálovatelnost a spolehlivost systému.

5 Aplikace blockchainu a chytrých smluv v prostředí IoT

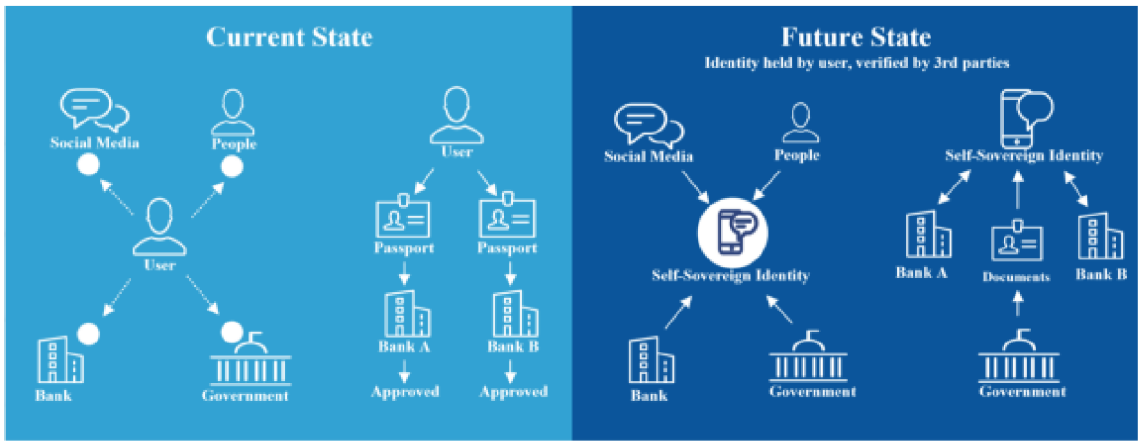
Jelikož jsme probrali tři hlavní komponenty tohoto celku v předchozích kapitolách, tak se nyní zaměříme na možné fungování v určitých případech.

5.1 Předpokládané uplatnění

Pro vypracování této kapitoly jsem využil dokument Smart Contracts: 12 Use Cases for Business & Beyond A Technology, Legal & Regulatory Introduction od Chamber of digital commerce, ze kterého jsem využil jejich grafické zpracování jednotlivých nasazení blockchainu do určitých odvětví a problematiky. Dokument posloužil i jako zdroj informací pro některé z uvedených příkladů.

5.1.1 Digitální identity

Jedním z nejslibnějších případů využití blockchainu a chytrých smluv v internetu věcí je správa digitální identity. V tomto scénáři může blockchain poskytnout neměnný záznam identity jednotlivce, který lze bezpečně sdílet mezi různými zařízeními a platformami. Chytré smlouvy lze použít k vynucení zásad řízení přístupu a usnadnění výměny ověřených informací mezi stranami. Tento přístup může pomoci zabránit krádeži identity, omezit podvody a zvýšit soukromí a bezpečnost (Deloitte & Alliance, 2016).



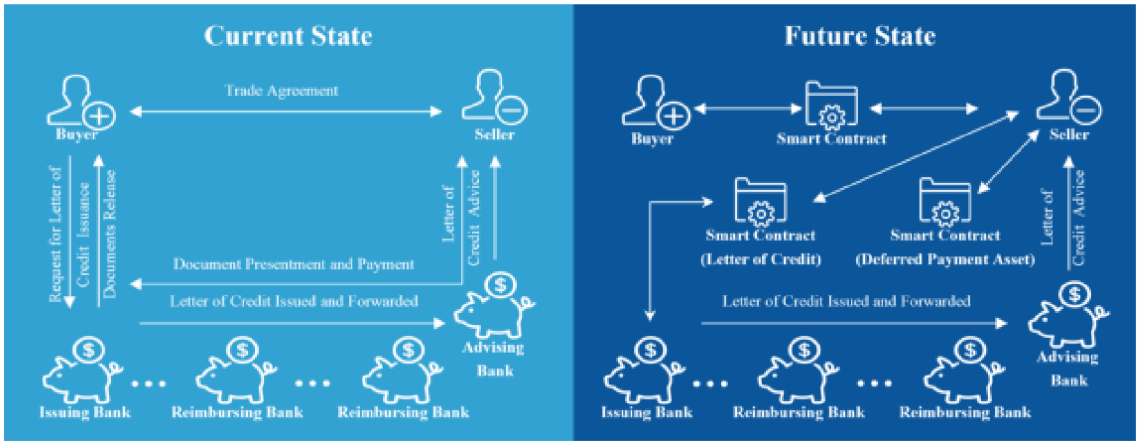
Obrázek 3: Schéma řešení digitální identity

Zdroj: (Deloitte & Alliance, 2016)

5.1.2 Uplatnění ve finančním sektoru

Ve finančním odvětví se tato implementace může stát významným přínosem. Chytré smlouvy lze využít k automatizaci různých procesů souvisejících s obchodováním s deriváty, jako jsou výzvy k úhradě marže, správa kolaterálu a vypořádání. Naproti tomu blockchain lze využít k obchodování a správě různých typů derivátů, jako jsou opce, futures a swapy, decentralizovaným, transparentním a bezpečným způsobem bez potřeby

zprostředkovatelů. Kromě toho mohou chytré smlouvy zlepšit mezinárodní převody zboží zkrácením času díky rychlým akreditivům a iniciaci obchodních plateb, umožnit větší likviditu finančních aktiv a zvýšit efektivitu financování pro kupující, dodavatele a instituce. Aby bylo možné z této reformy procesu plně těžit, je nutné do inteligentních smluv integrovat parametry a postupy složitého systému mezinárodního obchodu. (Peters & Panayi, 2015)

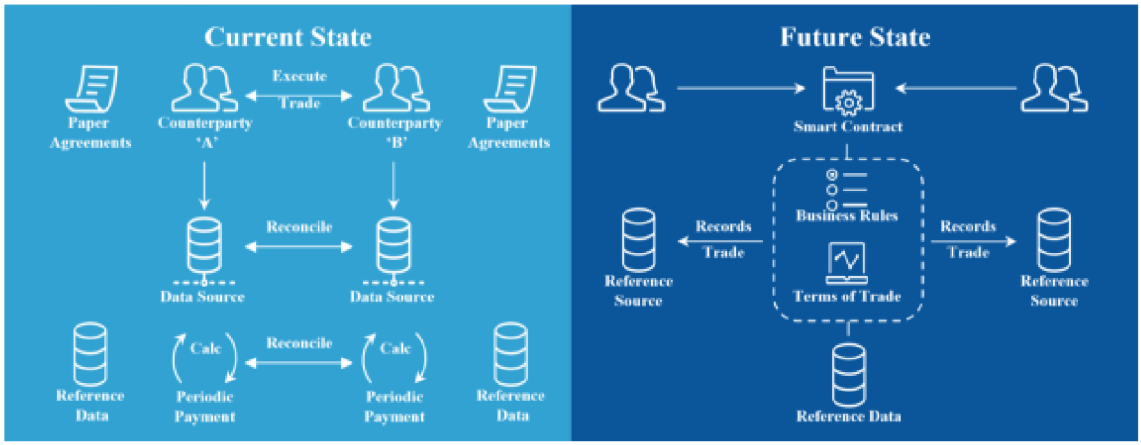


Obrázek 4: Schéma řešení pro finanční sektor

Zdroj: (Deloitte & Alliance, 2016)

5.1.3 Finanční deriváty

Obchodování s deriváty lze revolučně změnit tím, že se umožní decentralizovaný, transparentní a bezpečný způsob obchodování bez zprostředkovatelů. Blockchain může také poskytnout standardní soubor smluvních podmínek a ocenění pozic v reálném čase pro monitorování a prevenci chyb. Aby však bylo možné tuto technologii plně využít, je nutné řešit regulační reformy související s chytrými smlouvami o derivátech a integrovat parametry a postupy mezinárodního obchodního systému do chytrých smluv. To může zlepšit efektivitu financování, zkrátit dobu transakce a umožnit větší likviditu finančních aktiv, a zároveň odstranit duplicitní procesy prováděné zúčastněnými protistranami. (Peters & Panayi, 2015)



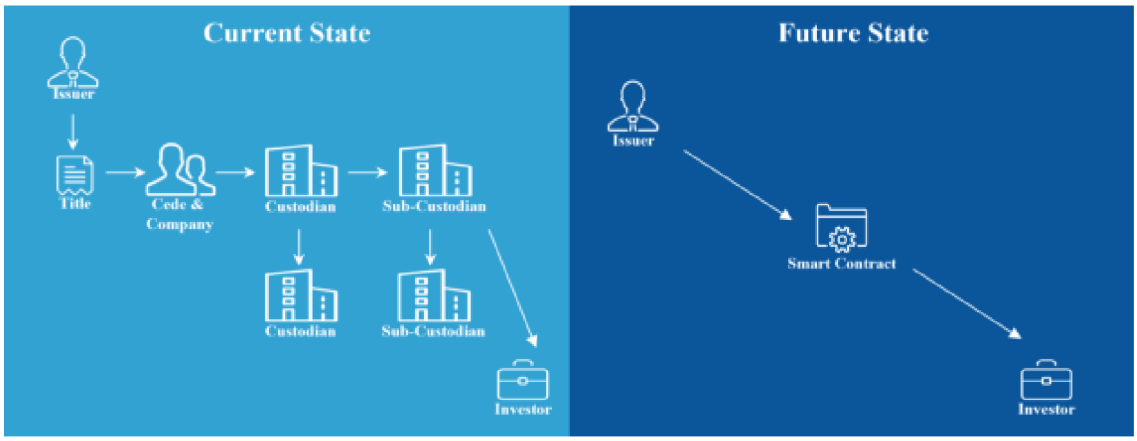
Obrázek 5: Schéma řešení pro finanční deriváty

Zdroj: (Deloitte & Alliance, 2016)

5.1.4 Cenné papíry a akcie

Díky blockchainu lze obchodovat s různými druhy cenných papírů, jako jsou akcie, dluhopisy a deriváty, obchodovat decentralizovaně, transparentně a bezpečně, bez potřeby zprostředkovatelů. Kromě toho mohou chytré smlouvy automatizovat různé procesy související s obchodováním s cennými papíry, jako je vypořádání, zúčtování a dodržování předpisů.

Mimo jiné mohou chytré smlouvy také usnadnit automatickou výplatu dividend, rozdělení akcií a správu závazků, a zároveň snížit provozní rizika a digitalizovat pracovní postupy. Pro některé emitenty však zůstává problémem viditelnost toho, kdo vlastní cenné papíry, protože se snaží tyto informace chránit. Na druhou stranu chytré smlouvy založené na blockchainu mohou umožnit standardní soubor smluvních podmínek a oceňování pozic v reálném čase pro monitorování, prevenci a omezení chyb. (Peters & Panayi, 2015)

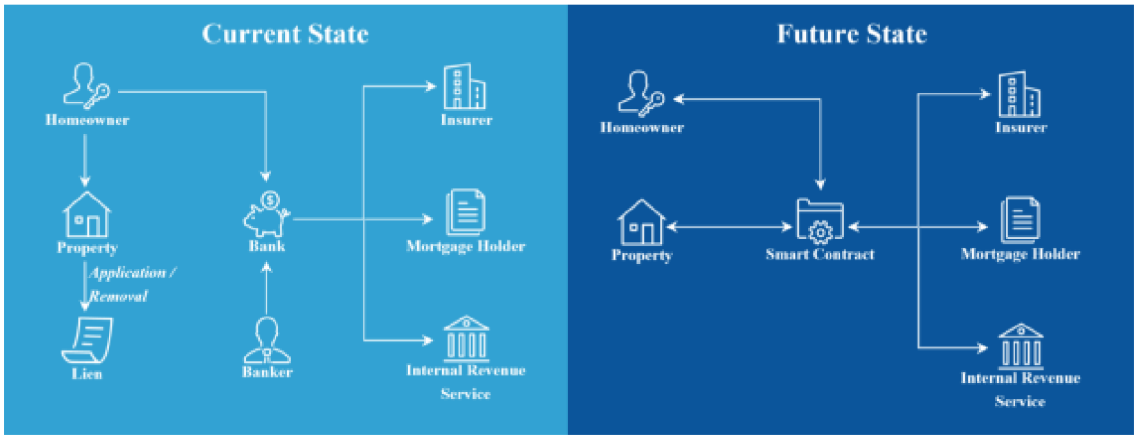


Obrázek 6: Schéma řešení pro cenné papíry a akcie

Zdroj: (Deloitte & Alliance, 2016)

5.1.5 Hypotéky

Hypoteční průmysl už dlouho hledá zjednodušený a efektivní proces. Díky blockchainu lze celý proces hypotéky, od jejího vzniku až po vyřízení, zpracovat decentralizovaně, transparentně a bezpečně, což sníží potřebu zprostředkovatelů a urychlí celý proces. Chytré kontrakty navíc mohou automatizovat různé úkoly související se zpracováním hypoték, jako je kontrola úvěruschopnosti, odhad a zpracování plateb. Díky využití digitální identity jako předpokladu mohou chytré smlouvy automaticky propojit strany zapojené do hypotečního procesu a zajistit tak proces bez tření a s menším rizikem chyb. Automatizovat lze také proces platby a po splacení úvěru lze uvolnit zástavní práva z katastru nemovitostí. Chytré smlouvy navíc mohou zlepšit přehlednost záznamů pro všechny zúčastněné strany a snížit chyby a náklady spojené s manuálními procesy. (Peters & Panayi, 2015)

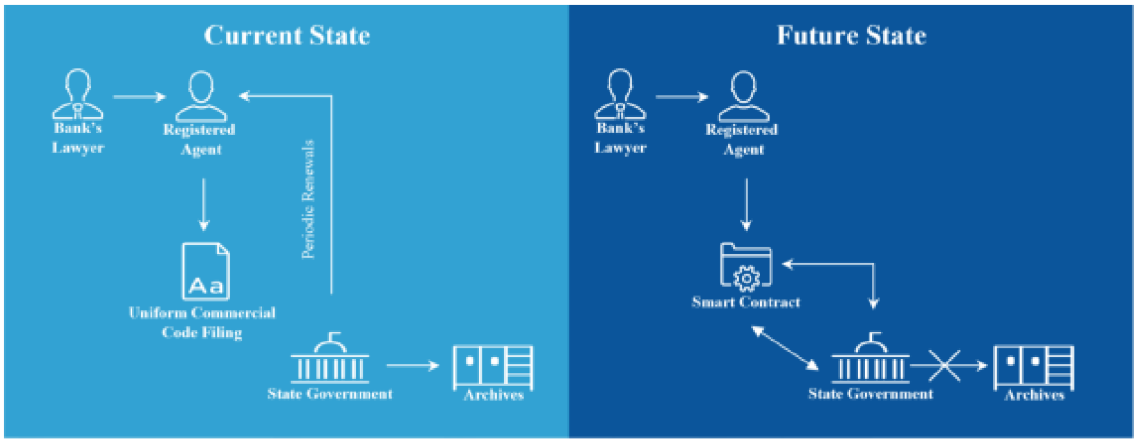


Obrázek 7: Schéma řešení hypoték

Zdroj: (Deloitte & Alliance, 2016)

5.1.6 Záznamy

Držení záznamů je důležité v různých odvětvích, včetně financí, práva a zdravotnictví. Blockchain nám nabízí možnost uchovávat a spravovat širokou škálu digitálních záznamů v decentralizovaném systému odolném proti manipulaci, ke kterému mají přístup pouze oprávněné strany. Chytré smlouvy mohou dále automatizovat proces ověřování a aktualizace těchto záznamů, čímž se sníží potřeba zprostředkovatelů a zefektivní pracovní postupy. Podobně lze chytré smlouvy využít k automatizaci dodržování právních předpisů a nařízení, jako je například Jednotný obchodní zákoník ve Spojených státech, což snižuje právní náklady a zvyšuje provozní efektivitu. (Deloitte & Alliance, 2016)

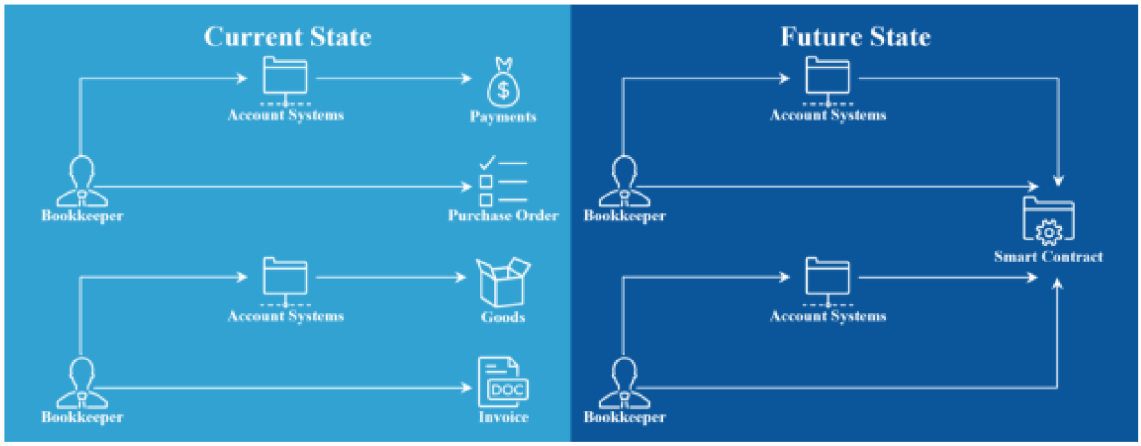


Obrázek 8: Schéma řešení záznamů

Zdroj: (Deloitte & Alliance, 2016)

5.1.7 Evidence finančních dat

Finanční odvětví lze změnit pomocí blockchainu, který zvládne finanční data ukládat do decentralizovaného systému odolného proti neoprávněné manipulaci, ke kterému mohou bezpečně přistupovat oprávněné strany. To zahrnuje zaznamenávání a správu různých typů finančních dat, jako jsou tržní údaje, údaje o transakcích, údaje o rizicích a jednotné finanční údaje napříč organizacemi. Chytré kontrakty mohou automatizovat úlohy související se záznamem, ověřováním, agregací, analýzou a dodržováním předpisů v oblasti finančních dat. (OECD, 2022)

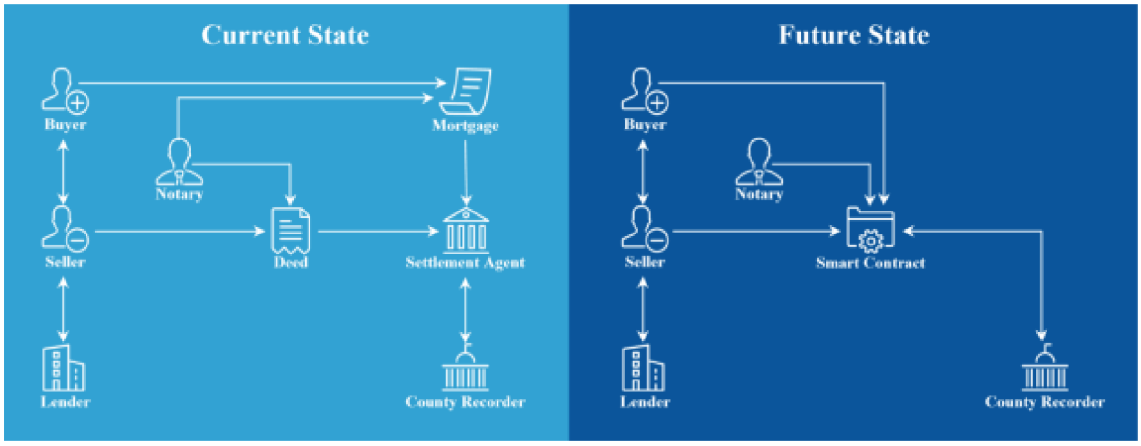


Obrázek 9: Schéma řešení evidence finančních dat

Zdroj: (Deloitte & Alliance, 2016)

5.1.8 Vlastnické právo k pozemkům a nemovitostem

Technologie blockchain má v realitním průmyslu obrovský potenciál. Pomocí blockchainu lze zaznamenávat a spravovat pozemkové tituly v decentralizovaném systému odolném proti manipulaci, což snižuje riziko podvodů a zvyšuje přesnost. Chytré smlouvy mohou automatizovat úkoly související se zápisy vlastnických práv k pozemkům, jako je ověřování vlastnictví, převod vlastnictví a zpracování plateb, což vede ke zvýšení efektivity a transparentnosti. Kromě toho může používání chytrých smluv posílit důvěru v identitu a snížit náklady na audit. K plnému využití potenciálu této technologie je třeba vyvinout společné protokoly pro elektronickou evidenci záznamů. (Deloitte & Alliance, 2016) (OECD, 2022)

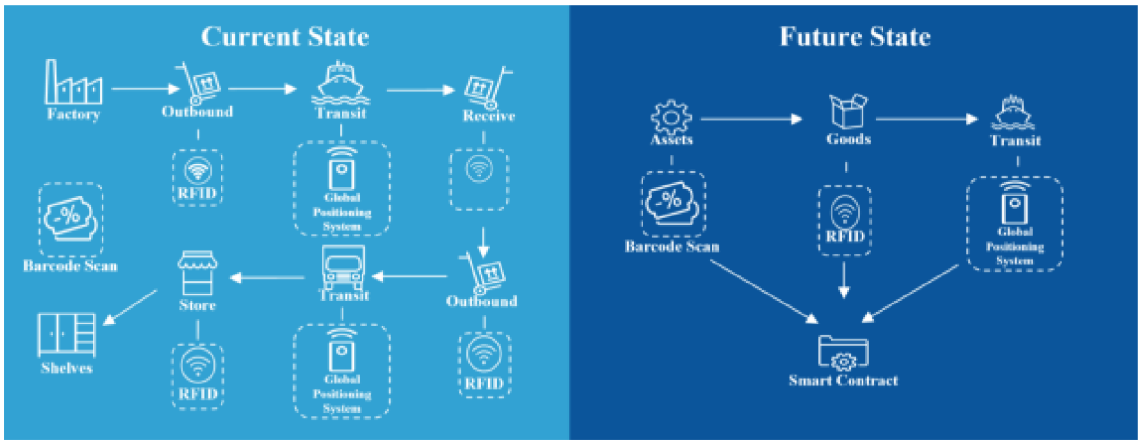


Obrázek 10: Schéma řešení vlastnického práva

Zdroj: (Deloitte & Alliance, 2016)

5.1.9 Dodavatelský řetězec

Chytré smlouvy lze také využít k automatizaci různých úkolů souvisejících s řízením dodavatelského řetězce, jako je sledování pohybu zboží, ověřování pravosti výrobků a usnadnění plateb. Kombinací blockchainu se zařízeními internetu věcí lze sledovat celý dodavatelský řetězec v reálném čase, což zlepšuje řízení zásob a snižuje plýtvání. To může vést ke snížení nákladů pro všechny strany zapojené do dodavatelského řetězce a může to být přínosné i pro spotřebitele, protože jim to poskytne větší transparentnost ohledně výrobků, které kupují. (Deloitte & Alliance, 2016)



Obrázek 11: Schéma řešení v dodavatelské řetězci

Zdroj: (Deloitte & Alliance, 2016)

5.1.10 Energetický managment

Správa energie je slibnou oblastí pro uplatnění blockchainu a chytrých smluv v internetu věcí, protože může pomoci řešit některé klíčové problémy v energetice, jako je energetická bezpečnost, spolehlivost a udržitelnost. Blockchain lze například využít k vytvoření decentralizovaného trhu s energií, kde mohou jednotlivci a podniky obchodovat s energií peer-to-peer způsobem bez potřeby zprostředkovatelů.

Kromě toho lze blockchain a chytré smlouvy využít ke správě obnovitelných zdrojů energie, jako jsou solární panely a větrné turbíny. Pomocí blockchainu lze sledovat výrobu a distribuci obnovitelné energie transparentním a bezpečným způsobem, což snižuje riziko podvodů a zvyšuje odpovědnost. (Marco Iansiti, 2017)

5.1.11 Volební systém

Volební systém s využitím blockchainu a chytrých smluv má potenciál vytvořit bezpečné a transparentní hlasovací prostředí, které ztíží manipulaci s výsledky voleb ze strany padělatelů. Záznamy o hlasování mohou být uloženy v decentralizovaném systému

odolném proti manipulaci, k němuž mají oprávněné strany bezpečný přístup, což zajistí, že každý hlas bude přesně započítán a nebude možné jej po odevzdání změnit.

Chytré smlouvy mohou také automatizovat různé úkoly související s volebním procesem, jako je registrace voličů, distribuce hlasovacích lístků a sčítání hlasů, což může pomoci omezit chyby a zpoždění, a zajistit, aby volby proběhly spravedlivě a efektivně. Kromě toho mohou blockchain a chytré smlouvy pomoci řešit klíčové problémy současného volebního systému, jako jsou podvody s voliči, potlačování volebních práv a nízká volební účast. (Chia-Hao Lee, 2022)

Technologie blockchain může například lidem usnadnit hlasování na dálku nebo hlasování pomocí mobilních zařízení, což by mohlo zvýšit volební účast. Celkově má využití blockchainu a chytrých smluv ve volebním systému potenciál způsobit revoluci ve způsobu, jakým jsou volby prováděny, a učinit je bezpečnějšími, transparentnějšími a přístupnějšími pro všechny. A proto jsem si zvolil toto téma jako cíl pro svou praktickou část, která se tímto problémem bude zabývat.

6 Vývoj aplikace s využitím technologie blockchain a chytrých smluv v prostředí IoT

V této části diplomové práce se budeme věnovat popisu a samotnému vývoji aplikace s využitím technologie blockchain a chytrých smluv v prostředí IoT. Se zvyšující se popularitou IoT v průmyslu a dalších oblastech se do popředí dostávají i blockchain technologie a s nimi chytré smlouvy a jejich význam. V rámci této kapitoly popíšeme kroky potřebné k vytvoření aplikace využívající blockchain a chytré smlouvy, včetně volby vhodného programovacího jazyka a nástrojů pro vývoj. Dále se zaměříme na propojení zařízení v námi vytvořené IoT síti zařízení. Nakonec zhodnotíme výsledky vytvořené aplikace a její přínos pro využití blockchainu v prostředí IoT.

6.1 Nastínění aplikace

Pro naší decentralizovanou aplikaci budeme muset vyřešit jakou zvolit technologii blockchain, která podporuje chytré kontrakty, zdali se vydáme cestou Etherea, nebo zdali zkusíme hyperledger možnost. Potom se musíme rozhodnout pro programovací jazyk pro chytré smlouvy, jestli se vydáme cestou Solidity či Vyper. Další na řadě bude vývoj webové aplikace, která bude sloužit jako operační panel pro náš blockchain a chytrou smlouvu a s největší pravděpodobností se budeme rozhodovat bud mezi Vue.js nebo react.js. A v poslední řadě vytvoření IoT sítě, což vytvoříme skrze můj osobní router, kde vytvoříme čistě sít pro IoT zařízení, která by z této sítě měla zvládnout komunikovat s naší JavaScriptovou aplikací.

6.2 Výběr technologie

Pro tento projekt jsme zvolili jako blockchain technologie Ethereum, která nám poslouží jako ideální nástroj pro naše účely. Jako programovací jazyk pro náší chytrou smlouvu jsme zvolili solidity, jelikož má velkou kompatibilitu s etherem, ale je také nástroj pro vytváření kódu na strojové úrovni a jeho kompilaci v rámci Ethereum Virtual Machine (EVM). Pro zpracování naší webové aplikace využijeme JavaScript a kombinací HTML a CSS pro vizualizaci webové aplikace. A pro IoT řešení se vydáme vytvořením samostatné sítě, kterou připojíme k technologii blockchain a vyzkoušíme interakce skrze zařízení Google Nest druhé generace, což by mělo zastoupit IoT zařízení v našem případě.

6.3 Nástroje a závislosti

První nástroj, který potřebujeme, je Node Package Manager neboli NPM. NPM je správce balíčků v jazyce JavaScript. Primární využití JS se samozřejmě zaměřuje na

vývoj mobilních aplikací a přidávání interakcí na webové stránky. Možnost průběžného testování a nasazování je nesmírná výhoda. Místo toho, aby vývojáři vyžadovali kompletní aktualizaci aplikace, mohou provádět malé změny, které zlepšují uživatelský komfort, přidávají funkce a zvyšují bezpečnost dat.

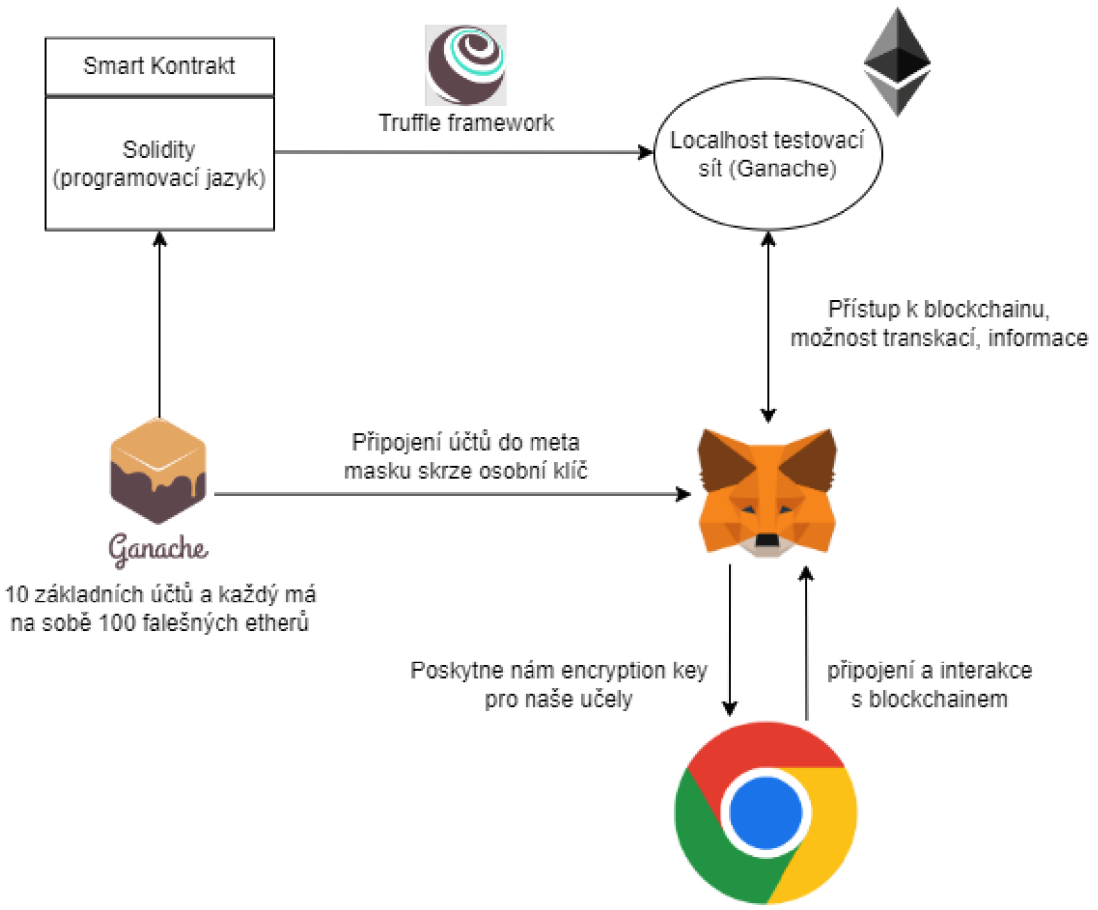
Další nástroj je Truffle Framework, který nám umožňuje vytvářet decentralizované aplikace na blockchainu Ethereum. Poskytuje sadu nástrojů, které nám umožňují psát chytré kontakty pomocí programovacího jazyka Solidity. Umožňuje nám také testovat naše chytré kontrakty a nasazovat je na blockchain. Poskytuje nám také místo pro vývoj naší aplikace na straně klienta.

Další nástroj je Ganache, lokální in-memory blockchain. Ganache je osobní blockchain pro rychlý vývoj distribuovaných aplikací Ethereum a Filecoin. Ganache můžeme používat v celém vývojovém cyklu; umožňuje nám vyvíjet, nasazovat a testovat naše dApps v bezpečném a deterministickém prostředí. Poskytne nám 10 externích účtů s adresami v našem lokálním blockchainu Ethereum. Na každém účtu je přednastaveno 100 falešných etherů.

Další a poslední nástroj je rozšíření Metamask pro Google Chrome. Abychom mohli využívat blockchain, musíme se k němu připojit. Abychom mohli používat blockchain Ethereum, musíme si nainstalovat speciální rozšíření prohlížeče. K tomu slouží metamask. Budeme se moci připojit k našemu lokálnímu blockchain Ethereu pomocí našeho osobního účtu a komunikovat s naší chytrou smlouvou.

6.4 Architektura aplikace

Obr. 12 ukazuje schéma architektury našeho systému elektronického hlasování na základě technologii blockchain. Existují tři hlavní komponenty, které zahrnují chytrý kontrakt, který je nasazen na blockchain (Ganache), front-end rozhraní pro uživatele a spojka, která nám oboje propojuje (Metamask).

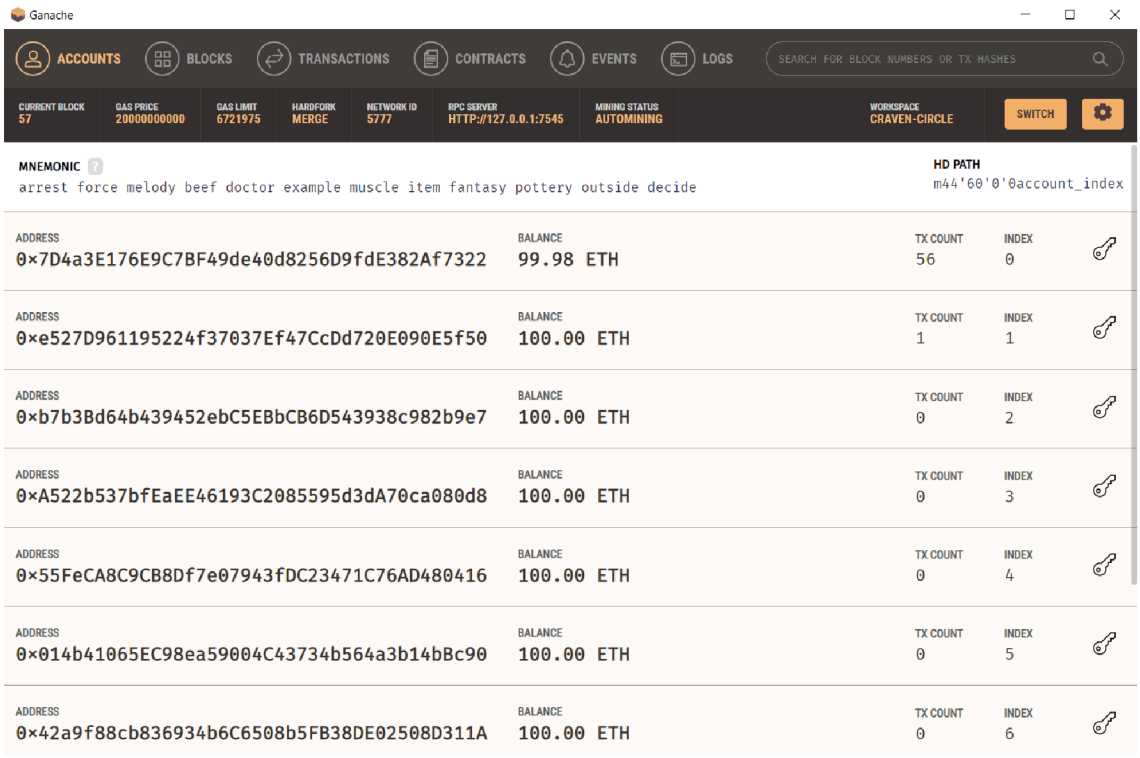


Obrázek 12: Popis architektury aplikace

Zdroj:Autor

6.5 Proces programování aplikace

Po stažení a připravení našich závislostí a nástrojů se přesuneme k jednotlivým krokům tvorby. Našim prvním krokem bude vytvoření lokálního blockchainu skrze aplikaci Ganache, která nám vytvoří 10 testovacích účtů, které můžeme volně používat, jelikož jsou nabitý falešným etherem viz obrázek 13. Při nastavení by nám normálně měl stačit předpřipravený quickstart, který ganache připravil ale v průběhu programování bylo nutné změnit nastavení, jelikož bylo zapotřebí řešit problém s tím, že ganache nelze připojit k sítí. Problém v našem řípadě byl, že došlo ke změně IP adresy adaptéru a ganache přestal fungovat. Řešení tohoto problému ve výsledku bylo, že jsme museli změnit IP adresu adaptéru zpět na dříve nastavenou, aby Ganache opět fungoval.



Obrázek 13: Zobrazení nastavené aplikace Ganache

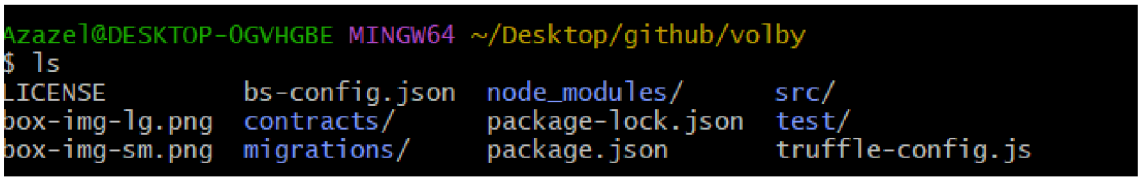
Zdroj:Autor

Další krok se může lišit, podle toho, zda programujeme na Linuxu nebo na Windows, pokud na Windows, tak budeme potřebovat ještě jeden program, což je GitBash a ten nám umožní Bash emulaci. Po zapnutí GitBash přijdou následující příkazy viz zdrojový kód 1.

```
mkdir volby
cd volby
truffle unbox pet-shop
```

Zdrojový kód 1: Kód pro počáteční kroky

Prvním příkazem si vytvoříme složku, do které následně stáhneme předpřipravené soubory od truffle suit jménem pet-shop. Pet-shop slouží jako tutoriál pro nové programátory, ale v našem případě ho využijeme proto, abychom nemuseli všechno vytvářet ručně. Obsah pet-shop souborů je znázorněn na obrázku 14.



Obrázek 14: Obsah Pet-shopu

Zdroj:Autor

6.5.1 Tvorba samotné chytré smlouvy

Vytvoříme novou jednoduchou chytrou smlouvu ve složce contracts, který bude sloužit jako kontrola, zdali dokážeme smlouvu na migrovat na blockchain viz zdrojový kód 2.

```
pragma solidity 0.4.25
contract Volby {
    string public kandidat;
    constructor () public {
        kandidat = "kandidat 1";
    }
}
```

Zdrojový kód 2: Testovací chytrý kontrakt

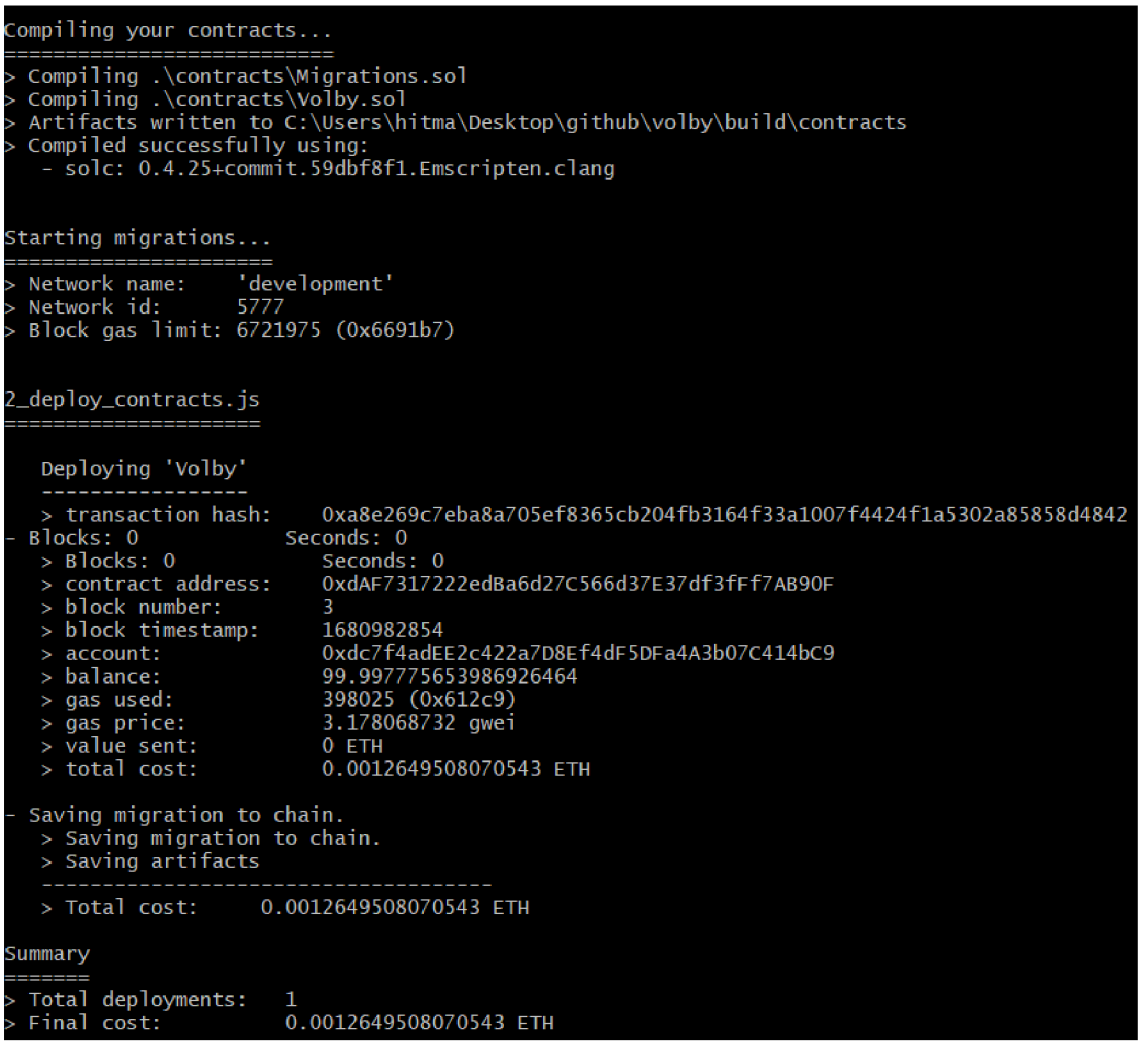
Zde jsme splnily základní požadavky na chytrou smlouvu jako je deklarace verze solidity a splnění syntaxe solidity. V tomto krátkém chytrém kontraktu máme nadefinovanou string proměnnou, která nám umožní zapisovat data do blockchainu. Jelikož je proměnná nadefinovaná jako řetězec s public viditelností, můžeme využít funkce getter, se kterou pracuje solidity a umožní nám přistupovat k této hodnotě mimo náš kontrakt. Dále funkce konstruktor, která se bude volat vždy, když nasadíme chytrý kontrakt do blockchainu. Zde nastavíme hodnotu proměnné stavu kandidáta, která se při migraci uloží do blockchainu.

Abychom mohli migrovat náší chytrou smlouvu na blockchain, budeme potřebovat vytvořit soubor pro migraci viz zdrojový kód 3.

```
var Volby = artifacts.require("./Volby.sol");
module.exports = function(deployer) {
    deployer.deploy(Volby)
}
```

Zdrojový kód 3: Soubor migrace kontraktu

Zavoláme si vytvořenou smlouvu a přiřadíme ji k proměnné "Volby". Poté ji přidáme do manifestu nasazených smluv, abychom zajistili, že bude nasazena při spuštění migrace.



Obrázek 15: Výpis z Gitbash o dokončené migraci

Zdroj:Autor

Nyní, když jsme úspěšně migrovali naší testovací chytrou smlouvu na blockchain, tak se můžeme pustit do psaní naší hlavní chytré smlouvy.

Data pro Kandidáta budeme modelovat pomocí struktury Solidity Struct, jelikož Solidity umožňuje vytvářet vlastní typy struktur. Zadali jsme, že tato struktura má ID typu unsigned integer, name typu string a voteCount typu unsigned integer. Musíme ji vytvořit instanci a přiřadit proměnné, než ji budeme moci zapsat do paměti.

Jelikož musíme struktury ukládat, tak potřebujeme solidity mapping. Mapping v Solidity je něco jako asociativní pole nebo hash, které sdružuje dvojice klíč-hodnota. Funkce mapování je celé číslo bez znaménka a hodnotou je typ struktury Kandidat, který jsme právě definovali. Tím v podstatě získáme vyhledávání na základě id pro každého kandidáta. Protože je toto mapování přiřazeno stavové proměnné, budeme do blockchainu zapisovat data, kdykoli mu přiřadíme nové dvojice klíč-hodnota.

Jelikož solidity neposkytuje žádný způsob, jak určit velikost mapování ani způsob iterace nad ním, tak jsem v našem případě odkázáni na counter cache.

Deklarovali jsme funkci addKandidat, která příjme jeden argument typu string, což pro nás bude jméno. Uvnitř funkce využijeme inkrementaci čítače kandidátů, abychom věděli, že přibyl nový kandidát. Poté jsme přidali mapping struktury Kandidat, přičemž klíč je počet kandidátů a samotná struktura Kandidat je inicializace idkandidata z aktuálního počtu kandidátů, jména z argumentu a počet hlasů 0. Funkce má viditelnost private, aby šla volat pouze ze smlouvy.

Další krokem byla schopnost volit, kterou jsme museli zaimplementovat. Základní funkce je zvýšit počet hlasů kandidáta načtením struktury Kandidat z mapování "kandidati" a zvýšením countru "voteCount". Přidá účet, který hlasoval, do mapování voličů. To nám umožní sledovat, že volič ve volbách hlasoval. K účtu, který zavolá, přistupujeme pomocí globální proměnné "msg.sender", kterou poskytuje Solidity. Implementace příkazu require, které zastaví provádění, pokud nejsou splněny podmínky. Nejprve vyžadují, aby volič ještě nehlasoval, provedeme kontrolu načtení adresy účtu s "msg.sender" z mapování a pokud tam je, účet již hlasoval. Dále vyžaduje, aby bylo platné id kandidáta. Id kandidáta musí být větší než nula a menší nebo rovno celkovému počtu kandidátů.

Poslední kus kódu, který potřebujeme, je spuštění události při každém hlasování. To nám umožní aktualizovat naši aplikaci na straně klienta, když účet hlasoval. A využijeme k tomu Event, což je řešení, jak komunikovat s klientskou aplikací nebo frontendovou webovou stránkou a upozorní nás, že se v blockchainu něco stalo. Ukázka viz zdrojový kód 4

```
    struct Kandidat {
        uint id;
        string name;
        uint voteCount;
    }
    mapping(address => bool) public volici;
    mapping(uint => Kandidat) public kandidati;
    uint public kandidatiCount;

    event votedEvent (
        uint indexed _kandidatId
    );
    constructor () public {
        addKandidat("Kandidát 1");
        addKandidat("Kandidát 2");
    }
    function addKandidat (string _name) private {
        kandidatiCount ++;
        kandidati[kandidatiCount] = Kandidat(kandidatiCount, _name, 0);
    }
    function vote (uint _kandidatId) public {
        require(!volici[msg.sender]);
        require(_kandidatId > 0 && _kandidatId <= kandidatiCount);
        volici[msg.sender] = true;
        kandidati[_kandidatId].voteCount ++;
        emit votedEvent(_kandidatId);
    }
}
```

Zdrojový kód 4: Ukazkový kód pro chytrý kontrakt

6.5.2 Tvorba testu pro chytrou smlouvu

K tvorbě našich testů využijeme testovacího frameworku Mocha a knihovny Chai assertion které jsou v balíčku trufflu. Kód testů bude v javascriptu abychom vytvořili simulaci interakce na straně klienta s naší chytrou smlouvou.

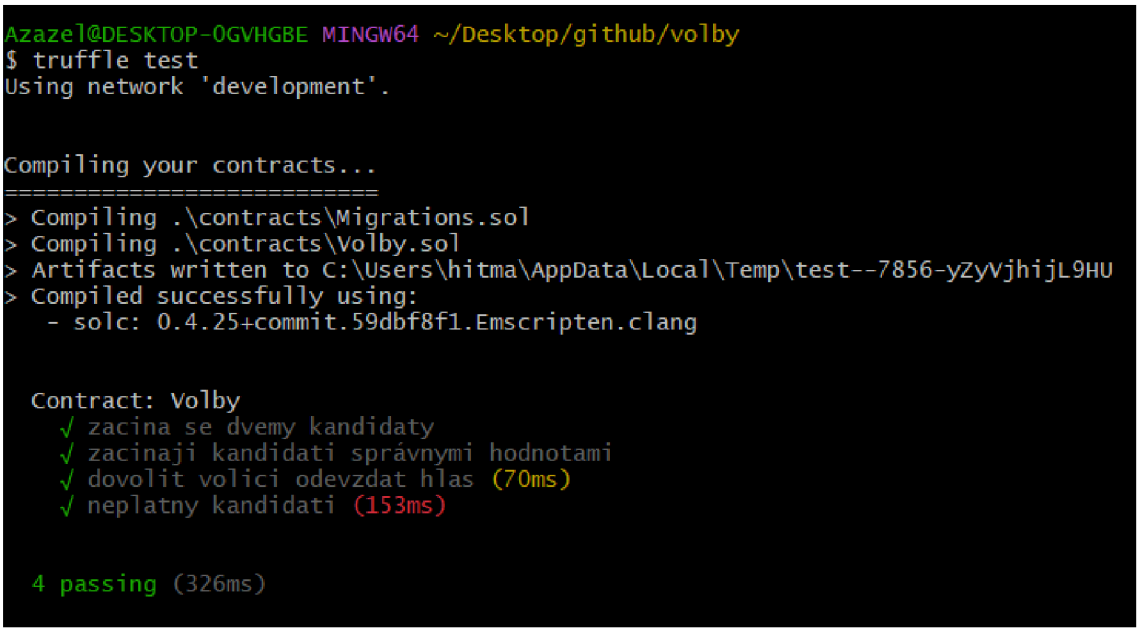
Začneme tím, že vyžadujeme skrze funkci require kontrakt a přiřadíme jej proměnné, stejně jako jsme to udělali v migračním souboru. Dále zavoláme funkci "contract" a všechny testy zapíšeme v rámci funkce zpětného volání. Funkce zpětného volání poskytuje proměnnou "accounts", která reprezentuje všechny účty v našem blockchainu.

První test kontroluje, zda byl kontrakt inicializován se správným počtem kandidátů. Druhý test kontroluje hodnoty jednotlivých kandidátů ve volbách a zajišťuje, že každý kandidát má správné id, jméno a počet hlasů. Ve třetím testu se zaměřujeme, zda funkce zvýší počet hlasů pro kandidáta a že volič je přidán do mapování vždy, když hlasuje. Poslední test viz zdrojový kód 5 má zjistit, že naše funkce vote vyhodí výjimku pro dvojí hlasování. Budeme přepokládat, že transakce selhala a že je vrácena chybová zpráva. Do této chybové zprávy nahlédneme, abychom zjistili, že chybová zpráva obsahuje podřetězec "revert". Poté můžeme zkontrolovat, že stav naší smlouvy nebyl změněn tím, že se ujistíme, že kandidáti neobdrželi žádné hlasy.

```
it("neplatny kandidati", function() {
  return Volby.deployed().then(function(instance) {
    volbyInstance = instance;
    return volbyInstance.vote(99, { from: accounts[1] })
  }).then(assert.fail).catch(function(error) {
    assert(error.message.indexOf('revert') >= 0, "error msg obsahuje revert");
    return volbyInstance.kandidati(1);
  }).then(function(kandidat1) {
    var voteCount = kandidat1[2];
    assert.equal(voteCount, 1, "kandidat 1 nedostal zadny hlas");
    return volbyInstance.kandidati(2);
  }).then(function(kandidat2) {
    var voteCount = kandidat2[2];
    assert.equal(voteCount, 0, "kandidat 2 nedostal zadny hlas");
  });
```

Zdrojový kód 5: Test pro neplatné kandidáty

Dále spustíme náš soubor s připravenými test skrze příkaz "truffle test", který spustí truffle framework a aplikuje náš test na chytrý kontrakt viz obrázek 16.



Obrázek 16: Úspěšný průchod testů

Zdroj:Autor

6.5.3 Webová aplikace

Tady se zaměříme na tvorbu aplikace na straně klienta, která bude komunikovat s naší chytrou smlouvou. Abychom si usnadnili práci stáhly jsme si pet-shop box což je předpřipravený template, který jsme si musely upravit pro svoje účely. Front-endová část je zpracovaná v HTML a do stylizován skrze CSS framework Bootstrap. Oproti tomu Back-endová část je čistě vytvořena skrze javascript. Jako testovací prostředí nám posloužilo lokální testovací server který jsme spouštěly přes příkaz "npm run dev" který nám spustil development script uložený v package.json.

Back-endová část se zabývá nastavením javascriptové knihovny web3.js která naší aplikaci umožnuje ze strany klienta komunikovat s našim blockchainem. Funkce, ve které nastavujeme celé dění je "initWeb3" nastavujeme, zdali zachytíme web3 instanci od MetaMask nebo využijeme defaultní instanci. Inicializace kontraktu je funkce načte nasazenou instanci chytré smlouvy a přiřadí jí hodnoty, které nám umožní s ní komunikovat. Pro náš cíl využijeme fukci listenforevent který se normálně používá když potřebujeme zachytit kliknutí na tlačítko a v našem případě čeká až nikdo odvolí a stránka se i hned znovu načte abychom udrželi nejaktuálnější informace.

```
initWeb3: function() {
  if (typeof web3 !== 'undefined') {
    App.web3Provider = web3.currentProvider;
    web3 = new Web3(web3.currentProvider);
        }
  else {
        App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
        web3 = new Web3(App.web3Provider);
        }
  return App.initContract();
},
```

Zdrojový kód 6: Nastavení funkce int Web3

Funkce render rozloží veškerý obsah na stránce pomocí dat z chytré smlouvy. V našem případě kandidáty, které jsme vytvořili uvnitř chytrého kontraktu. Průběh funkce je založen na smyčce která projde každého kandidáta v mapování a vykreslí ho do tabulky. Uvnitř této funkce také načteme aktuální účet, který je připojen k blockchainu, a zobrazíme ho na stránce viz zdrojový kód 7. Poslední funkce castvote nejprve se ve formuláři zeptá na kandidatId. Když voláme funkci hlasování z našeho chytrého kontraktu, předáme toto id a poskytneme aktuální účet s metadaty přes funkci from což vytvoří asynchronní volání. Po dokončení zobrazí pouze tabulku s kandidaty a jejich hlasy a hlasovací panel schová. Celý kód viz příloha 3.

```
web3.eth.getCoinbase(function(err, account) {
  if (err === null) {
    App.account = account;
    $("#accountAddress").html("Váše pěneženka: " + account);
        }
});
App.contracts.Volby.deployed().then(function(instance) {
  volbyInstance = instance;
  return volbyInstance.kandidatiCount();
}).then(function(kandidatiCount) {
  var kandidatiResults = $("#kandidatiResults");
  kandidatiResults.empty();
  var kandidatiSelect = $('#kandidatiSelect');
  kandidatiSelect.empty();
   for (var i = 1; i <= kandidatiCount; i++) {volbyInstance.kandidati(i).then(function(kandidat) {
      var id = kandidat[0];
      var name = kandidat[1];
      var voteCount = kandidat[2];
      var kandidatTemplate = "<tr><th>" + id + "</th><td>" + name + "</td><td>" + voteCount + "</td></tr>"
      kandidatiResults.append(kandidatTemplate);
      var kandidatOption = "<option value='" + id + "' >" + name + "</ option>"
      kandidatiSelect.append(kandidatOption);
    });
  }
```

Zdrojový kód 7: Částečný kód funkce render

Front-endová část je zaměřena spíše na jednoduché funkční prostředí které implementuje funkce z back-endu. Jeho vizuální řešení si ukážeme v další pod kapitole a jeho kód bude přiložen v příloze s celou aplikací.

6.6 Testování aplikace

Jak název podkapitoly napovídá, přesuneme se ke spuštění a testování naší aplikace, poté, co vyzkoušíme funkčnost na localhostu. Tak se přepneme na naší vytvořenou síť, ve které bude jeden telefon, počítač a IoT zařízení. Za IoT zařízení, které bychom mohli použít, jsme zvolili google nest hub, jelikož jím disponujeme. Domácí asistenti jsou bráni jako IoT zařízení, což nám umožnuje vyzkoušet interakce naší aplikace s ním.

Když máme vše hotové, tak si v rámci prvního kroku otevřeme Gitbash a začneme kontrolou našeho obsahu. Takže začneme příkazem "truffle test", který nám zkontroluje, jestli chytrý kontrakt funguje. Poté zapneme ganache a vytvoříme quickstart možnost, která nám vygeneruje nové účty. Dalším krokem je příkaz v Gitbash "npm run dev", to je příkaz, kterým spustíme lite-server node z balíčku npm viz obrázek 17.

```
Azazel@DESKTOP-OGVHGBE MINGW64 ~/Desktop/github/diplomka
$ npm run dev

> election@1.0.0 dev
> lite-server

** browser-sync config **
{
  injectChanges: false,
  files: [ './**/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server: {
    baseDir: [ './src', './build/contracts' ],
    middleware: [ [Function (anonymous)], [Function (anonymous)] ]
  }
}
[Browsersync] Access URLs:
 --------------------------------------
       Local: http://localhost:3000
    External: http://192.168.0.164:3000
 --------------------------------------
          UI: http://localhost:3001
 UI External: http://localhost:3001
 --------------------------------------
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
23.04.10 10:14:57 200 GET /index.html
23.04.10 10:14:57 200 GET /css/bootstrap.min.css
23.04.10 10:14:57 200 GET /js/bootstrap.min.js
23.04.10 10:14:57 200 GET /js/web3.min.js
23.04.10 10:14:57 200 GET /js/truffle-contract.js
23.04.10 10:14:57 200 GET /js/app.js
23.04.10 10:14:57 200 GET /img/load.png
23.04.10 10:14:57 200 GET /Volby.json
```

Obrázek 17: Ukázka spuštění lite-serveru

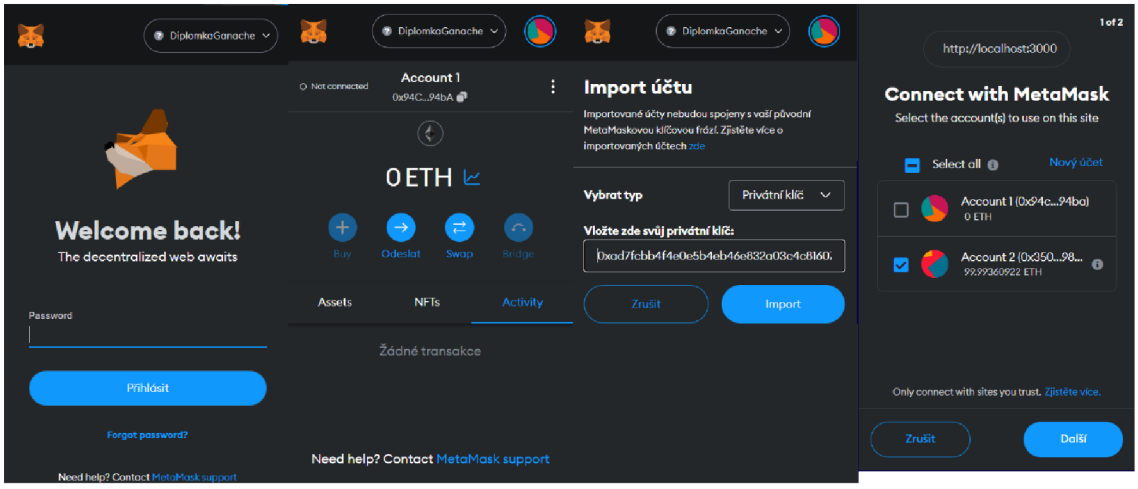
Zdroj:Autor

Potom, co se dokončí toto okno tímto způsobem, by se nám měl otevřít sám prohlížeč s naší stránkou, která by pro nás zatím měla být uzavřena, dokud nepřihlásíme nějakou krypto peněženku viz obrázek 18.

Obrázek 18: Ukázka vzhledu zamčené webové aplikace

Zdroj:Autor

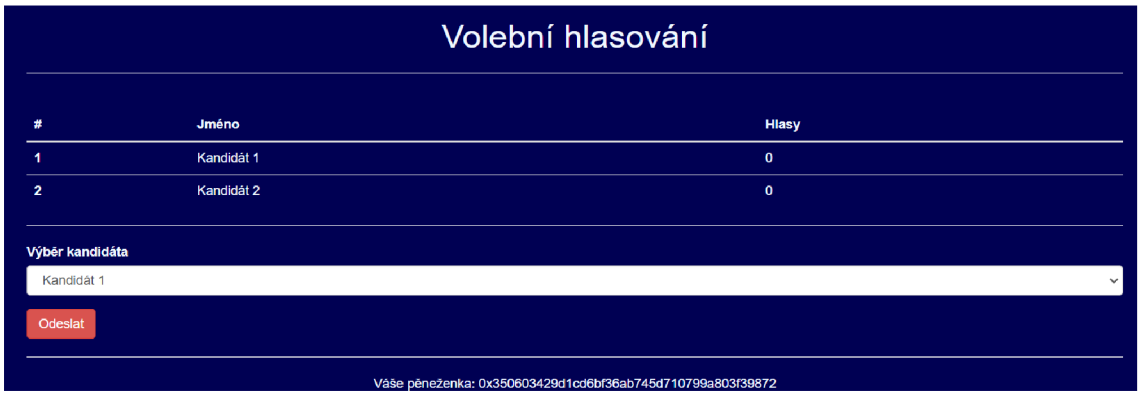
Na obrázku 18 můžeme vidět vzhled naší stránky, která čeká na web3 instanci od Metamasku nebo jiné krypto peněženky. Uzamčení stránky je řešeno přes html atribut a podmínku v back-endu. Přihlášení do naší krypto peněženky je řešeno skrze browser addon, v našem případě metamask (znázorněno na obrázku 19). Pokud by klient používal například Coinbase, funguje to také.



Obrázek 19: Přihlášení do našeho rozšíření a import učtu z ganache

Zdroj:Autor

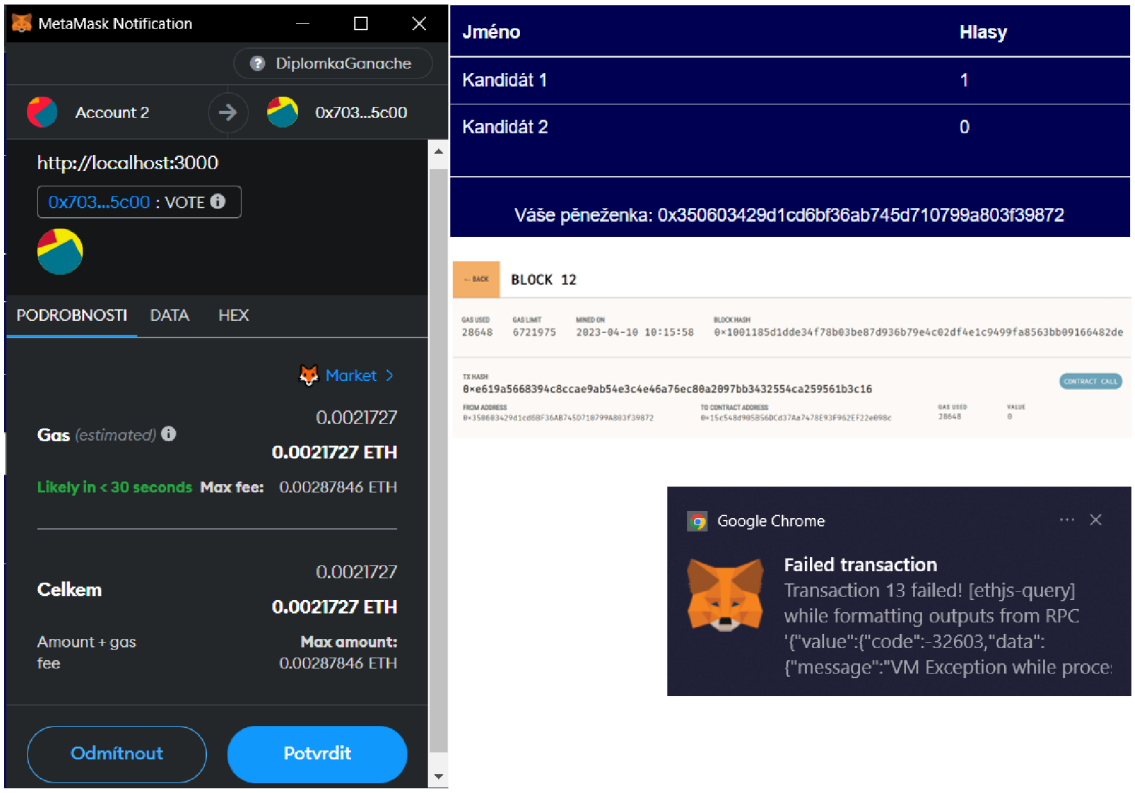
Obrázek 19 nám ukazuje, jak vypadá samotné rozšíření Metamask. V první části vidíme přihlášení do našeho uživatelského prostředí, v druhé části vidíme import našeho blockchain účtu z Ganache skrze privátní klíč. Za normální okolností není privátní klíč vidět, ale změnili jsme mu atribut, aby se vypisoval pro naše účely. Poté už bychom se měli dostat do naší volební aplikace viz obrázek 20.



Obrázek 20: Otevřená webová aplikace připravená k volení

Zdroj:Autor

Zde můžeme vidět finální styl naší webové stránky připravené k volbě. Zajímavé na této aplikaci je, že HTML v našem případě slouží jenom jako kostra a daty jí plní samotný chytrý kontrakt. Ale funkčnost samotné stránky je jednoduchá, vybereme si kandidáta ve výběru kandidáta a dáme odeslat, to nám vyhodí vyskakující okno našeho addonu, kde máme napsané veškeré informace o převodu etheru. Pokud potvrdíme, systém zapíše hlas našemu kandidátovi a aplikace se znovu načte, aby nedošlo k tomu, že nebudou vidět aktuální data viz obrázek 21.

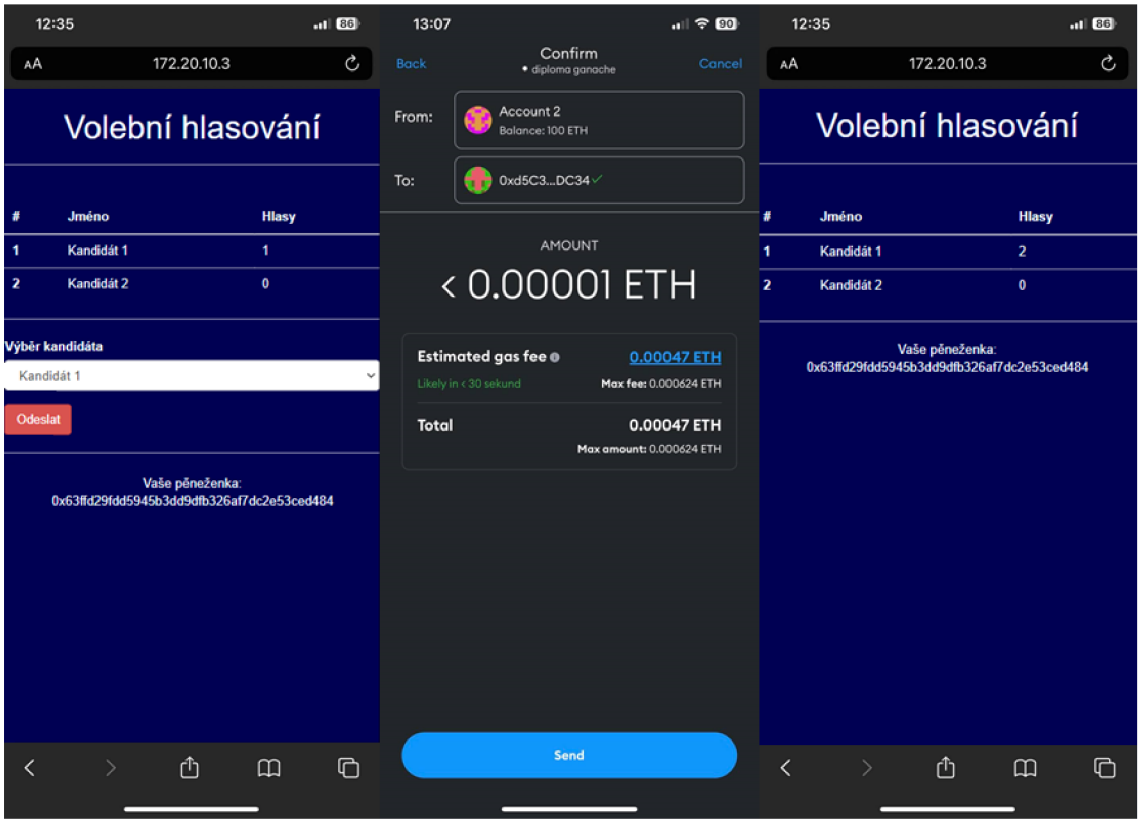


Obrázek 21: Potrvzení převodu, výsledky po volbě a double voting

Zdroj:Autor

Na Obrázku 21 máme výpis přímo v aplikaci Ganache, která nás odkazuje na blok s naší transakcí, kde můžeme vidět jednotlivé informace o naší operaci Contract call, což pro nás je poslání hlasu. Dále v prvé spodní části můžeme vidět pokus o double voting, který systém zastavil a oznámil jako transaction failed.

Nyní se přesuneme k testu na mobilním zařízení, pro naše účely využijeme iPhone XS Max, na kterém provedeme stejný úkon jako na počítači.



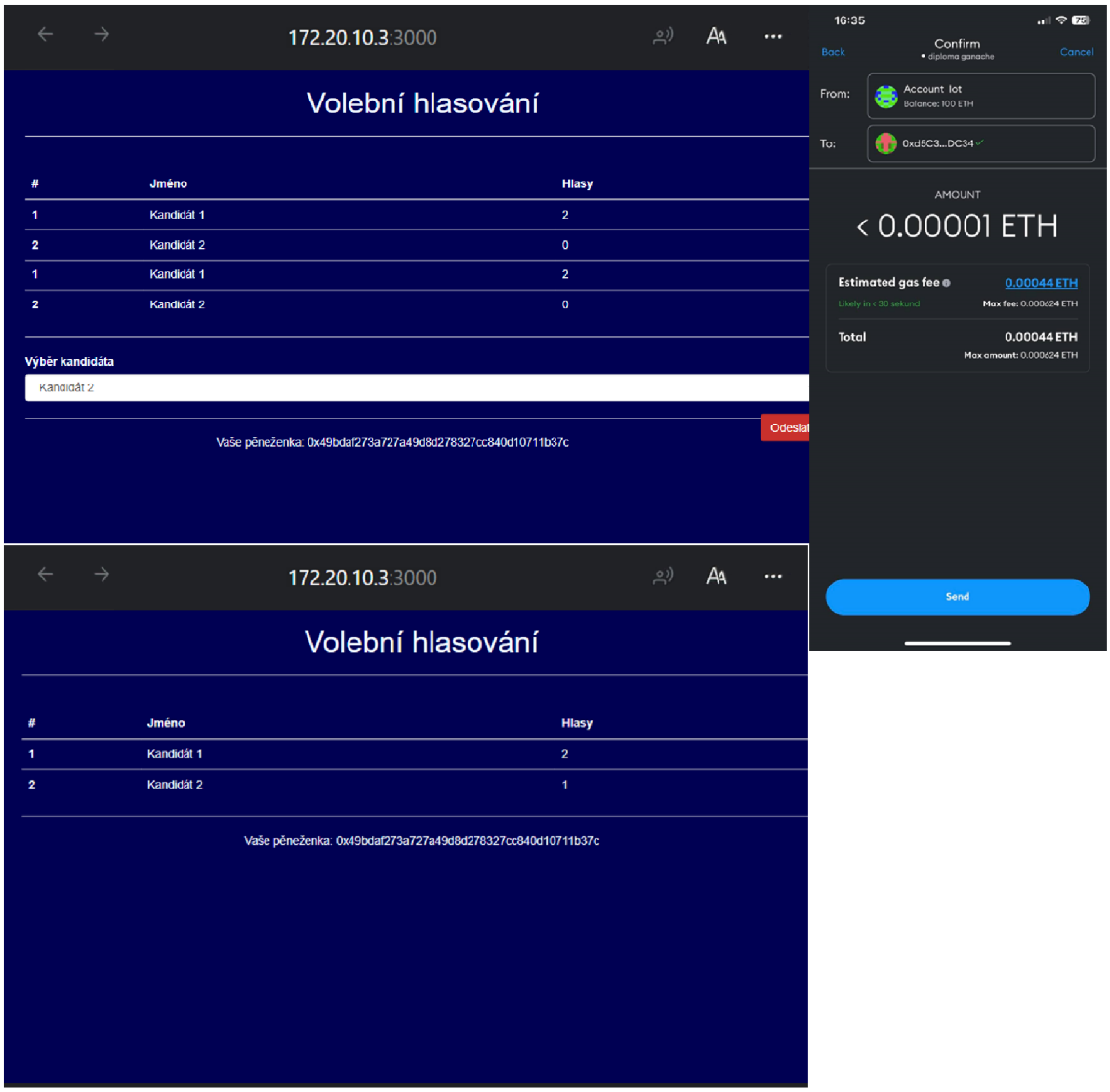
Obrázek 22: Mobilní test naší aplikace

Zdroj:Autor

Jak můžeme vidět, proces je skoro stejný, pouze v mobilní verzi musíme mít staženou aplikaci Metamask v telefonu, aby naše webová aplikace vyzvala aplikaci Metamask k proplacení volby z naší peněženky. Po úspěšném proplacení můžeme vidět, že se naše aplikace znovu načetla a už započítala další odeslaný hlas pro našeho kandidáta.

Jako poslední testování máme test s IoT zařízením. Prvotním plánem bylo sestrojit nfc čtečku jako senzor a přes nfc tag nahrát script s naší peněženkou, který by fungoval jako naše identita, a kdybychom chtěli volit v naší aplikaci, tak místo přihlášení se do Metamasku, bychom využili náš nfc tag. Bohužel se zde vyskytly problémy z hlediska možností a také našich znalostí a zkušeností ohledně tohoto typu operace. Takže jsme se vydali cestou domácích asistentů, kteří spadají do skupiny IoT zařízení také. Naše

vybrané IoT zařízení je Google nest hub, a jelikož toto zařízení nedisponuje samotnou krypto peněženkou, tak nedokáže samo dát potvrzení k platbě. V tomto hledisku nám pomůže telefonní zařízení, se kterým je spojené naše IoT zařízení a dá nám potvrzení. Operace proběhne skrze IoT zařízení a pouze potvrzení bude skrze telefon.



Obrázek 23: Iot interakce s webovou aplikací

Na obrázku 23 můžeme vidět, že naše rozložení je deformované a data z kontraktu se načetli dvakrát, ale funkčnost to nemění, tlačítko i navzdory tomu že ho část chybí odešle příkaz, který pošle žádost o schválení na telefon a otevře se nám náš účet pro IoT zařízení a požaduje po nás potvrzení. Po dání potvrzení můžeme vidět, že aplikace se znovu načetla a vidíme aktuální výsledky našich voleb.

6.7 Vyhodnocení

Po našem testování můžeme říct, že aplikace funguje a splňuje to, co jsme od ní požadovali. Z hlediska grafického designu jsou tu velké ústupky, které by v plném nasazení chtěli vyřešit, jako je responzivita samotné stránky a kompatibilita se všemi dostupnými zařízeními. Do budoucna by se určitě dali přidat i další věci, které by nám zpříjemnily používání, jako grafy s vyhodnocením či bohatší grafický styl stránky.

Z hlediska chytrého kontraktu nedošlo k žádné chybě, kterou bychom zaznamenali, jelikož jsme hlídali jednotlivé vytěžené bloky kdykoliv došlo k volbě a zavolání samotného kontraktu. Jediná věc, která nás zaujala, byla změna gas fee pro různá zařízení.

Z hlediska back-endu stránky jsme narazili na určité nuance, které bychom nyní už vytvořili jinak, například dříve zmiňovaná funkce initWeb3, v našem případě je funkce zaměřena čistě na Metamask a třeba s Coinbase nefungovala. V budoucnu by bylo vhodné se zaměřit na univerzálnost, jak z hlediska grafiky a responzivního designu pro všechna zařízení, tak z hlediska back-endu na čistotu a stabilitu kódu, ale i využití s více programy a zařízeními což pomůže vytvořit přívětivější uživatelské prostředí.

7 Závěr

Tato práce se zaměřuje na použití technologie blockchain a chytrých kontraktů v kontextu internetu věcí (IoT). Cílem práce bylo zmapovat různé technologie blockchain a chytrých kontraktů, vybrat ty nejvhodnější a vytvořit z nich aplikaci, která funguje s IoT zařízeními.

Teoretická část práce poskytuje ucelený přehled technologie blockchain, chytrých kontraktů a Internet of Things. Práce se zaobírá zkoumáním jednotlivých technologii jako jsou blockchainové platformy podporující chytré kontrakty nebo aplikace chytrých kontraktů pro předávání dat skrze IoT sít. Analýza různých platforem blockchainu umožnila na základě konkrétních parametrů vybrat tu nejvhodnější pro vývoj naší aplikace, což bylo Ethereum. Našim programovacím jazykem pro chytré kontrakty se stalo solidity kvůli své kompatibilitě s etherem. Zkoumali jsem i to, co definuje vlastně samotný IoT a jeho potencionální využití v kombinaci s blockchainem a chytrými kontrakty.

Praktická část práce začíná stručným popisem rozhodnutí o záměru, na co bude aplikace zaměřena a jaká technologie bude vybrána pro tvorbu naší aplikace. V rámci tohoto popisu se zaměříme na to, jaké nástroje budou potřeba pro náš vývoj a fungování samotné aplikace a v neposlední řadě řešení architektury samotné aplikace.

V druhé polovině praktické části se věnujeme samotnému programování jednotlivých částí naší aplikace jako je chytrý kontrakt či jeho testovací soubor skrze frameworky. A v poslední řadě samotná webová aplikace napsaná v HTML, CSS, JS, která všechny tyto části spojuje dohromady.

Navrhnutá aplikace by měla zastoupit centralizovaný volební systém, který se v minulosti setkal s mnoho podvody a neshledává se s nejlepší kritikou. Naše aplikace umožnuje decentralizované volby, které jsou zaštítěné právě technologiemi blockchain a chytrých kontraktů, které nám dávají důvěru ve svoji neměnnost a IoT sít jí dává možnost zvětšit svůj maximální dosah co se týče využití.

Testování aplikace prokázalo její funkčnost a bezpečnost a propojitelnost se zařízením internetu věcí. Aplikace splnila všechna svá očekávání, i když je zde určitě místo pro zlepšení.

Závěrem lze říct, že tato diplomová práce přinesla poznatky a vyzdvihla potenciál technologie blockchain a chytrých kontraktů v IoT. Práce ukazuje, že tyto technologie lze využít k vytvoření bezpečných a efektivních aplikací či systémů. Práce by mohla

posloužit jako zdroj informací pro další výzkum, který by se zaměřil na další vývoj aplikací a systémů s využitím těchto technologiích.

8 Summary and keywords

This thesis focuses on the use of blockchain technology and smart contracts in the context of the Internet of Things (IoT). The aim of this thesis was to map different blockchain and smart contract technologies, select the most relevant ones and create an application that works with IoT devices.

The theoretical part of this thesis provides an overview of blockchain technology, smart contracts and the Internet of Things. The thesis focuses on the study of individual technologies such as blockchain platforms which support smart contracts or the application of smart contracts for data transfer through IoT networks.

The second part of the thesis starts with a description of the decision on what the application will be focused on, what tools will be needed for our development, function of the application itself and the architecture of the application. In the second half we focus on the actual programming of each part of our application

The designed application could replace the centralized election system, which has seen much fraud in the past and has not met with the best criticism. Testing of the app has proven its functionality and security and connectivity with IoT devices. The app has met all its expectations and it is safe to say , that this thesis has provided knowledge and highlighted the potential of blockchain technology and smart con-tract in IoT.

Keywords: Internet of Things, smart contract, blockchain, programing, technology, application, election system

9 Seznam literatury

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *17*(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

Bashir, I. (2017). *Mastering Blockchain* (1st ed.). Packt.

Bjørnstad, M. V., Harkestad, J. G., & Krogh, S. (2017). *A study on blockchain technology as a resource for competitive advantage*. [Diplomová práce, Norwegian University of Science and Technology]. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2472245/17527_FULLTEXT.pdf?sequence=1&isAllowed=y

Bogner, A., Chanson, M., & Meeuw, A. (2016). A Decentralised Sharing App running a Smart Contract on the Ethereum Blockchain. *Proceedings of the 6th International Conference on the Internet of Things*, 177-178. https://doi.org/10.1145/2991561.2998465

Bonneau, J., Clark, J., & Goldfeder, S. (2015). *On Bitcoin as a public randomness source* [diplomová práce, Stanford University, Concordia University, Princeton University]. https://ia.cr/2015/1015

Burhan, M., Rehman, R., Khan, B., & Kim, B. -S. (2018). IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey. *Sensors*, *18*(9). https://doi.org/10.3390/s18092796

Buterin, V. (2023). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. *Ethereum white papers*, *2014*(1), 1-36. https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf

Cirani, S., Ferrari, G., Picone, M., & Veltri, L. (2018). *Internet of Things: architectures, protocols and standards* (1st ed.). John Wiley.

Drescher, D. ([2017]). *Blockchain basics: a non-technical introduction in 25 steps* (1st ed.). Apress.

Hayes, A. (2022). Learn how these digital public ledgers enable crypto and NFTs. *Investopedia*, *1*(1), 1. https://www.investopedia.com/terms/b/blockchain.asp

Choi, S. -I., & Koh, S. -J. (2016). Use of Proxy Mobile IPv6 for Mobility Management in CoAP-Based Internet-of-Things Networks. *IEEE Communications Letters*, *20*(11), 2284-2287. https://doi.org/10.1109/LCOMM.2016.2601318

Iansiti, M., & R. Lakhani, K. (2017). The Truth About Blockchain: It will take years to transform business, but the journey begins now. *Harvard business review*, *1*(1), 1. https://hbr.org/2017/01/the-truth-about-blockchain

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016). Evaluation of Logic-Based Smart Contracts for Blockchain Systems. *Rule Technologies. Research, Tools, and Applications*, 167-183. https://doi.org/10.1007/978-3-319-42019-6_11

Kevin, A. (2009). *That 'Internet of Things' thing*. Retrieved April 12, 2023, from https://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf

Lee, C. -H., Neo, H. -F., & Teo, C. -C. (2022). Secure E-Voting System based on Blockchain Technology. *Journal of System and Management Sciences*, *2022*(1), 1-18. https://doi.org/10.33168/JSMS.2022.0508

Liu, H., Liu, C., Zhao, W., Jiang, Y., & Sun, J. (2018). S-gram: towards semantic-aware security auditing for Ethereum smart contracts. *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 814-819. https://doi.org/10.1145/3238147.3240728

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Bitcoin.org*, *1*(1), 1-9. https://assets.pubpub.org/d8wct41f/31611263538139.pdf

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Sensing as a Service Model for Smart Cities Supported by Internet of Things. *TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES*, *1*(1), 1-12. https://doi.org/10.1002/ett

Pinar, Y., Zuhair, A., Hamad, A., Resit, A., Shiva, K., & Omar, A. (2016). Wireless Sensor Networks (WSNs). *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 1-8. https://doi.org/10.1109/LISAT.2016.7494144

Radoglou Grammatikis, P. I., Sarigiannidis, P. G., & Moscholios, I. D. (2019). Securing the Internet of Things: Challenges, threats and solutions. *Internet of Things*, *5*(1), 41-70. https://doi.org/10.1016/j.iot.2018.11.003

Smart Contracts Alliance, & Deloitte. (2016). Smart Contracts: 12 Use Cases for Business & Beyond: A Technology, Legal & Regulatory Introduction — Foreword by Nick Szabo. Website, 1(1), 1-56. https://www.the-blockchain.com/docs/Smart%20Contracts%20-%2012%20Use%20Cases%20for%20Business%20and%20Beyond%20-%20Chamber%20of%20Digital%20Commerce.pdf

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday*, *2*(9). https://doi.org/10.5210/fm.v2i9.548

W. Peters, G., & Panayi, E. Understanding Modern Banking Ledgers through Blockchain Technologies:: Future of Transaction Processing and Smart Contracts on the Internet of Money. *Oxford Mann Institute*, *1*(1), 1-33. https://arxiv.org/pdf/1511.05740.pdf

Ethereum: ETHEREUM DEVELOPMENT DOCUMENTATION. (2022). Retrieved April 12, 2023, from https://ethereum.org/en/developers/docs/

The trust machine: The technology behind bitcoin could transform how the economy works. (2015). *The Economist*, *1*(1), 1-5. https://www.economist.com/leaders/2015/10/31/the-trust-machine

Hyperledger Fabric. (2023). Retrieved April 12, 2023, from https://hyperledger-fabric.readthedocs.io/en/release-2.5/

The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them: Cryptography, Derivatives Investments, Futures Trading, Digital Assets, NFT). (2018) (1st ed.). Mango.

Smart Contracts: 12 Use Cases for Business & Beyond: A Technology, Legal & Regulatory Introduction. (2016). *Chamber of digital commerce*, *1*(1), 1-56. https://www.the-blockchain.com/docs/Smart%20Contracts%20-%2012%20Use%20Cases%20for%20Business%20and%20Beyond%20-%20Chamber%20of%20Digital%20Commerce.pdf

TRON: Advanced Decentralized Blockchain Platform. (2018). *White papers*, *1*(1), 1-40. https://tron.network/static/doc/white_paper_v_2_0.pdf

Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Retrieved April 12, 2023, from

Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. (2016) (1st ed.). Princeton University Press.

Consensus Mechanism: Neo. (2023). Retrieved April 12, 2023, from https://docs.neo.org/docs/en-us/basic/consensus/dbft.html

Why Decentralised Finance (DeFi) Matters and the Policy Implications. (2022). Retrieved April 12, 2023, from https://www.oecd.org/daf/fin/financial-markets/Why-Decentralised-Finance-DeFi-Matters-and-the-Policy-Implications.pdf

On Bitcoin as a public randomness source: Joseph Bonneau and Jeremy Clark and Steven Goldfeder. (2015). Retrieved April 12, 2023, from https://eprint.iacr.org/2015/1015.pdf

Bitshares.: Delegated Proof of Stake (DPOS). (2023). Retrieved April 12, 2023, from https://how.bitshares.works/

Rootstock documentation. (2023). Retrieved April 12, 2023, from https://dev.rootstock.io/

Introduction to Sawtooth PBFT. (2019). *Hyperledger foundation*, *1*(1), 1. https://www.hyperledger.org/blog/2019/02/13/introduction-to-sawtooth-pbft

Binance.: BNB chain documentation. (2023). Retrieved April 12, 2023, from https://docs.bnbchain.org/docs/overview

Stellar: documentation. (2023). Retrieved April 12, 2023, from https://developers.stellar.org/docs

IoT Threats & Implementation of AI/ML to Address Emerging Cyber Security Issues in IoT with Cloud Computing. (2023). *International Research Journal of Modernization in Engineering Technology and Science*, *1*(1), 1-6. https://doi.org/10.56726/IRJMETS32866

10 Seznam obrázků

11 Seznam ukázek kódů

12 Seznam příloh

Příloha č. 1: Zip soubor obsahující celkový kód pro naši aplikaci

Dále jmenované přílohy jsou součástí přílohy č. 1

Příloha č. 2: Solidity soubor obsahující celý kód chytré smlouvy (Volby.sol)

Příloha č. 3: JavaScript soubor obsahující kód na testování (volby.js)

Příloha č. 4: JavaScript soubor obsahující kód webové aplikace (app.js back-end)