Wykorzystanie Ethereum do Budowy Zdecentralizowanej Aplikacji

Wojciech Korzeniowski

Instytut Informatyki
Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska
http://www.ii.pw.edu.pl/

Streszczenie Opis smart kontraktów zdefiniowanych na platformie Ethereum. Przykład wykorzystania Ethereum do zrealizowania zdecentralizowanego systemu do głosowania. Koniec artykułu zostanie poświęcony poszczególnym zagrożeniom wynikającym z wykorzystania smart kontraktów wraz z konsekwencjami ich przeoczenia

Słowa Kluczowe: Ethereum · Smart kontrakt · Blockchain · Decentralized Application · DApp

1 Ethereum

Ethereum jest zdecentralizowaną platformą dla aplikacji, które działają dokładnie tak jak zostały zdefiniowane bez możliwości oszustwa, cenzury czy interwencji stron trzecich [1]. Platforma została uruchomiona 30 lipca 2015 przez rosyjskiego programistę nazywającego się Vitalik Buterin i jej główną funkcjonalnością jest możliwość definiowana smart kontraktów.

Ether jest kryptowalutą wykorzystywaną na platformie Ethereum. Pod względem wartości rynkowej jest drugą co do wielkości kryptowalutą na świecie, zaraz po Bitcoinie.[1] Bitcoin, uruchomiony 9 stycznia 2009 roku, został stworzony aby umożliwić transfer waluty bez ograniczeń w postaci instytucji bankowych czy lokalizacyjnych. Pomimo tego że również umożliwia definiowanie smart kontraktów, częściej do tego celu wykorzystywane jest Ethereum.

2 Smart kontrakt

Smart kontrakt jest kolejnym etapem rozwoju zastosowań technologii blockchain. Można znaleźć tłumaczenia, które opisują smart kontrakt jako cyfrowy zapis umowy, która w odpowiednich warunkach realizuje ustaloną akcję. Z technicznego punktu widzenia jest to zbiór danych, które zostaną zachowane dopóki działa conajmniej jeden z węzłów sieci Ethereum, oraz funkcji, które mogą operować na danych i są jedynym sposobem na ich zmianę.

[1] Według serwisu https://coinmarketcap.com/, stan na 2.06.2018

Ta funkcjonalność spowodowała powstanie nowego rodzaju aplikacji oznaczanych DApp, od angielskiego terminu Decentralized Application, co oznacza w języku angielskim "zdecentralizowana aplikacja". Są to aplikacje, które wykorzystują smart kontrakty, lub bardziej ogólnie blockchain, jako miejsce do przechowywania kodu i danych aplikacji. W ethereum smart kontrakty definiuje się wykorzystując język Solidity [4].

Wywołania niektórych funkcji wymagają przekazania Etheru, który może być przechowywany w smart kontrakcie jak w zwykłym portfelu lub być przekazanym dalej wraz z wywołaniem innej funkcji. Ma to miejsce na przykład w grach hazardowych gdzie aby wywołać funkcję losu należy przekazać ustaloną kwotę Etheru jako opłatę za los.

Istotną cechą, która wynika z architektury blockchainu jest fakt, iż smart kontrakt, który został utworzony na blockchainie nie może zostać zmieniony. Jest to bardzo istotne ze względu na bezpieczeństwo tworzonej aplikacji. Jeżeli popełnimy błąd w definicji kontraktu nie ma możliwości jego naprawy. Jedyne co można zrobić to utworzyć nowy kontakt i zaprzestać korzystania z jego starej wersji. Takie rozwiązanie nie zawsze jest satysfakcjonujące, szczególnie jeżeli do starej wersji kontraktu przypisany jest Ether a nie utworzyliśmy funkcji, która pozwala na jego przekazanie do innego portfela. Z drugiej strony istnienie funkcji, która pozwala na wybranie całego Etheru z kontraktu może wzbudzić podejrzenia co do intencji jego twórcy.

Funkcje zdefiniowane w smart kontrakcie można podzielić na 2 kategorie. Te, które czytają dane i te, które je zmieniają. Jako że dostęp do danych z blockchainu jest publiczny i można stworzyć własny węzeł, który jest pełną kopią blockchainu, dane można odczytać w każdej chwili. Inaczej jest w przypadku funkcji, która modyfikuje stan kontraktu lub samego tworzenia kontraktu. Wynika to z faktu, iż każdy z węzłów w sieci musi wywołać tę funkcję aby potwierdzić czy inne węzły również ją wykonały i stan blockchainu po wywołaniu funkcji się zgadza między węzłami. Fakt, iż musimy wykonać tę funkcję na wszystkich węzłach powoduje że za jej wywołanie, wołający musi zapłacić. Koszt operacji wyrażany jest w jednostce Gas. Ilość Gasu potrzebnego do wywołania funkcji jest proporcjonalna do złożoności obliczeniowej wołanej funkcji. Podczas tworzenia transakcji można określić ile maksymalnie gasu może być wykorzystane na wywołanie funkcji. Jeżeli podczas wywoływania funkcji ustalony limit zostanie osiągnięty, wywołanie zostaje przerwane aby nie przekroczyć limitu.

Wywołania funkcji, utworzenie smart kontraktu oraz transfer Etheru na inne konto nazywamy transakcją. Wszystkie transakcje trafiają do puli transakcji. Zazwyczaj kopacze podczas wykopywania nowego bloku wybierają z puli te transakcje, w których autor ustalił największą cenę za Gas. Dzieje się tak ponieważ to kopacz dostaję opłata za Gas ze wszystkich transakcji które wchodzą w skład nowo wykopanego bloku. Stąd cena Gasu zależy od aktualnego obciążenia sieci oraz od tego czy chcemy aby nasza funkcja została wywołana jak najszybciej. Po ustaleniu zbyt niskiej ceny za transakcję może się zdażyć tak, że kopacze będą

wybierać bardziej korzystne dla nich transakcje powodując że transakcja może znajdować się w puli transakcji przez bardzo długi czas.[2].

2.1 System do głosowania

Smart kontrakty mogą zostać wykorzystane do zbudowania systemu do tajnego głosowania. Organizator głosowania tworzy umowę z osobami uprawnionymi do głosowania, która pozwala uprawnionym na oddanie dokładnie jednego głosu. Następnie, po zamknięciu głosowania, opcja, która zebrała najwięcej głosów zostaje zwycięzcą głosowania.

Zastanówmy się najpierw jakie są problemy w organizowaniu głosowania bez smart kontraktów. Przede wszystkim należy zadbać o to aby głos mogła oddać tylko osoba do tego upoważniona oraz aby każda z tych osób mogła oddać tylko jeden głos. Kolejnym problemem jest sposób w jaki głosy są liczone. Jak powiedział Józef Stalin: "Nieważne, kto głosuje, ważne, kto liczy głosy."

Powyższe problemy w przypadku głosowań w Polsce rozwiązywane są przez Państwową Komisję Wyborczą, która organizuje i pilnuje porządku głosowania. Tworzone są okręgowe komisje wyborcze, których odpowiedzialnością jest kontrolowanie czy osoba oddająca głos jest do tego uprawiona. Następnie komisja skrutacyjna liczy głosy po czym ogłaszany jest wynik głosowania.

Takie rozwiązanie wymaga istnienia tak zwanej "zaufanej trzeciej strony". W przypadku wyborów w Polsce jest to PKW. Obywatele muszą zaufać że komisja w poprawny sposób zorganizuje i przeprowadzi głosowanie a następnie bezbłędnie policzy głosy i ogłosi zwycięzcę. Nie raz pojawiały się różnego rodzaju kontrowersje co do sposobu przeprowadzania wyborów. Na przykład podczas wyborów samorządowych 2014 opóźnione było ogłoszenie wyników. PKW tłumaczyła to awarią systemów informacyjnych jednak co bardziej podejrzliwi obywatele wyczuwali w tym spisek. Dodatkowo obywatele muszą ufać że każda z okręgowych komisji wyborczych będzie przestrzegać prawa i nie nadużyje swoich kompetencji aby sfałszować organizowane wybory.

W przypadku głosowania zrealizowanego na smart kontraktach nie ma potrzeby istnienia zaufanej trzeciej strony. Można zdefiniować kontrakt, którego definicja jest publicznie dostępna i każdy może sprawdzić w jaki sposób zbierane są głosy oraz jak są one zliczane. Wymaganie aby każdy z uprawionych mógł zagłosować tylko jeden raz można zrealizować poprzez przekazanie każdemu uprawnionemu jednego tokenu do głosowania, który nie może być przekazany dalej. Osoba, która wykorzystuje swój token do zagłosowania wywołuje odpowiednią funkcję na kontrakcie do głosowania w efekcie czego liczba głosów na wybraną opcję zwiększa się.

Jedną z głównych zalet realizacji systemu głosowania opartego na blockchainie jest jego transparentność. Ponieważ dane zapisane na blockchainie są publicznie dostępne do odczytu każdy z zainteresowany może sprawdzić jak przebiegało głosowanie oraz jaki jest jego aktualny stan. Ponadto technologia blockchain

[2] `https://ethgasstation.info/` - serwis do obliczania opłaty za Gas

gwarantuje że dane zapisane w blockchainie nie zostaną zmienione więc nie ma możliwości fałszowania wyników.

3 Token

Tokenem w świecie Ethereum nazywamy nową "kryptowalutę", która istnieje na blockchainie Ethereum. Istnieje przyjęty interfejs tokenu o nazwie 'ERC20', którego implementacja musi definiować między innymi całkowitą liczbę tokenów, nazwę oraz zasady przekazywania go innym użytkownikom.

Tokeny mogą być wykorzystane do zbiórki pieniędzy co jest nazywane ICO (ang. Initial Coin Offering) i jest odpowiednikiem określenia IOP (ang. Initial Public Offering) znanego giełd papierów wartościowych. Przypuśćmy że właściciel startupu potrzebuje dofinansowania do swojego biznesu. Może on utworzyć token, który następnie będzie sprzedawał za Ether. W ten sposób twórca pozyskuje Ether, którym może płacić lub wymienić na inną walutę. Natomiast kupujący w zależności od przyjętej polityki ICO, może otrzymać udziały w startupie lub możliwość wykorzystania tokenu w zamian za usługę realizowaną przez biznes twórcy tokenu. Podczas ICO cena tokenu jest ustalana przez jego twórce, w późniejszym czasie jego wartość jest weryfikowana przez rynek.

4 Komunikacja z siecią Ethereum

Sieć Ethereum [2] składa się z węzłów. Węzeł to serwer, który posiada lokalną kopię blockchainu wraz ze wszystkimi transakcjami i smart kontraktami, które zostały w nim zapisane. Węzły komunikują się ze sobą w celu ustalienia jednej, wspólnej wersji blockchainu, jest to nazywane ustaleniem konsensusu. Najbardziej rozpowszechnionym sposobem ustalenia konsensusu, który jest wykorzystywany zarówno w Ethereum jak i w Bitcoinie, jest dowód pracy. W związku z ogromnym zużyciem energii elektrycznej które jest wymagane przed dowód pracy jedną z propozycji rozwoju Ethereum jest zmiana sposobu ustalania konsensudu i przejście na dowód stawki. [3]

Aby utworzyć własny węzeł Ethereum można wykorzystać oficjalną implementację węzła, dostępną do pobrania z serwisu Github[3]. Nie jest to jednak koniecznie jeżeli chcemy zacząć przygodę z tworzeniem smart kontraktów. Serwis Infura[4] udostępnia za darmo publiczny węzeł, który można wykorzystać do komunikacji z siecią Ethereum. Komunikacja z węzłem następuje przy pomocy protokołu JSON-RPC. Jednym z bardziej znanych klientów jest klient napisany w języku JavaScript o nazwie Web3.js[5]. Wykorzystując go można stworzyć aplikację działającą w przeglądarce, która komunikuje się bezpośrednio z węzłem Ethereum.

[3] https://github.com/ethereum/go-ethereum
[4] https://infura.io/
[5] https://github.com/ethereum/web3.js/

Aby utworzyć własny kontrakt w sieci można skorzystać z narzędzia Remix[6]. Jest to edytor w, którym można napisać własny kontrakt a następnie z poziomu przeglądarki utworzyć jego kopię na blockchainie. Aby możliwe było utworzenie kontraktu na blockchainie twórca musi musi ponieść koszt utworzenia kontraktu. W przypadku narzędzia Remix oraz innych aplikacji przeglądarkowych wykorzystujących klienta Web3.js można skorzystać z wtyczki do przeglądarki o nazwie Metamask[7].

Technicznie Metamask wstrzykuje do strony globalny obiekt JavaScript o nazwie web3, który jest wykorzystywany do dalszej interakcji z blockchainem. Podczas tworzenia nowego smart kontraktu lub wywoływania funkcji, która wymaga zapłaty pojawia się okienko, które wymaga potwierdzenia czy chcemy wykonać daną akcję za określoną opłatą.

5 Architektura aplikacji

Najprostszą architekturą DApp jest klient działający w przeglądarce, który komunikuje się bezpośrednio z wybranym węzłem. Inną możliwością jest stworzenie własnego serwera, który komunikuje się z węzłem oraz aplikację przeglądarkową komunikującą się tylko z naszym serwerem bez bezpośredniej komunikacji z węzłem. Takie rozwiązanie może być wykorzystane w celu przyspieszenia aplikacji aby nie odpytywać węzeł o dane za każdym razem tylko przechowywać je w pamięci podręcznej na serwerze.

Innym sposobem na wykorzystanie blockchainu jest wykorzystanie go tylko do przechowywania wartości funkcji skrótu danych, które trzymamy w innym miejscu. Takie podejście stosowane jest ze względu na fakt, że zapis do blockchainu kosztuje proporcjonalnie do złożoności obliczeniowej wywoływanej funkcji a w przypadku zapisu wartości funkcji skrótu koszt jest stały oraz nie zależy od ilości danych, z których powstał skrót. Blockchain w takim przypadku służy tylko do weryfikacji czy dane, które odczytujemy zgadzają się z danymi, które zostały "podpisane" poprzez zapisanie ich skrótu na blockchainie.

6 Bezpieczeństwo

Dane aplikacji są najczęściej centralnym punktem stworzonej aplikacji, co powoduje że stają się celem ataków grup hakerskich. Utrata danych może spowodować że aplikacja staje się mniej wygodna w użytkowaniu [6] a także sprawić że użytkownicy tracą zaufanie do twórców aplikacji [5]. Dane mogą także wyciec [7], w takim przypadku aplikacja działa jak przed atakiem jednak dane aplikacji zostały upublicznione lub przekazane osobom które nie powinny mieć do nich dostępu.

Ciężko ocenić, który ze scenariuszy jest gorszy. Usunięcie danych może sparaliżować pracę użytkownika aplikacji lub nawet nieść konsekwencje prawne w

[6] https://remix.ethereum.org/
[7] https://metamask.io/

przypadku utraty wrażliwych danych, które są nie do odzyskania np. dokumentacji podatkowych [8]. W przypadku utraty danych możemy przewidzieć jakie konsekwencje się z tym wiążą. Inaczej jest gdy nastąpi wyciek danych. Zdarzają się sytuacje, w których serwis świadomie lub nie, udostępnia czyjeś dane. W takim przypadku ciężko jest przewidzieć w jaki sposób dane użytkowników mogą zostaną użyte. Z problemem wycieku danych musiało zmierzyć się Poznańskie Centrum Superkomputerowo-Sieciowe, które pozwalało na pobranie imienia, nazwiska oraz adresu zamieszkana po podaniu numeru PESEL [9].

Wykorzystanie technologii blockchain pozwala ograniczyć wymienione powyżej zagrożenia ponieważ usunięcie danych z blockchainu jest niemożliwe. Ciężko też mówić o wycieku danych ponieważ dane zapisane na blockchainie są publicznie dostępne. Oczywiście jeżeli chcemy przechowywać dane wrażliwe na blockchainie muszą one być zaszyfrowane i możliwe do odczytu tylko dla odpowiednich osób. Jeżeli szyfrowanie zostanie zrealizowane błędnie lub sam klucz szyfrujący zostanie przejęty przez atakującego możemy mówić o wycieku danych.

Dodatkowo jak wspomniałem w rozdziale 5, DApp może wykorzystywać tradycyjne bazy danych takie jak relacyjne czy NoSQL. W takim przypadku twórcy aplikacji muszą brać pod uwagę zarówno niebezpieczeństwa związane z wykorzystaniem wybranej bazy danych jak i te wynikające z użycia blockchainu.

W celu wykorzystania sprawdzonych rozwiązań na problemy, które często się powtarzają podczas projektowania smart kontraktów warto skorzystać ze zbioru OpenZeppelin[8]. Można znaleźć tam gotowe smart kontrakty lub funkcje, które rozwiązują dane problemy bezpieczeństwa w sprawdzony przez społeczność Ethereum sposób. Przykładem może być chęć ograniczenia możliwości wywołania niektórych funkcji tylko do autora danego smart kontraktu. Jedną z takich funkcji może być wspomniana wcześniej funkcja służąca do wybrania Etheru, który jest przechowywany na smart kontrakcie.

7 Podsumowanie

W mojej opinii technologia blockchain spowoduje rewolucję w sposobie pracy instytucj zajmującymi się obrotem walutami na miarę rewolucji, którą spowodował internet w obszarze wymiany informacji. Wiele z instytucji finansowych można zastąpić poprzed zdefiniowane smart kontraktu, który zawsze działa tak samo i nie popełnia błędów. Instytucją, która jako pierwsza została pozbawiona sensu istnienia jest giełda. Większośc giełd kryptowalut działa w oparciu o smart kontrakty. Dzięki temu mamy pewność że giełda działa za każdym razem tak samo i nie popełni błędu, który może wynikać ze zwykłej pomyłki jak i celowego oszustwa.

Wykorzystanie smart kontraktów pozwala na wyeliminowanie wymagania istnienia zaufanej trzeciej strony co jest główną ideą przyświecającą ich twórcy. Dzięki spisaniu zasad transferu, waluty w sposób jednoznaczny w interpretacji sposób oraz gwarancji wykonania danych akcji po spełnieniu wcześniej przyję-

[8] https://github.com/OpenZeppelin/zeppelin-solidity

tych warunków możemy zawrzeć umowę z kimś komu nie ufamy bez potrzeby pośrednika w postaci notariusza.

Literatura

1. Ethereum Homepage, https://www.ethereum.org/
2. Ethereum Homestead http://www.ethdocs.org/en/latest/
3. Proof of Stake FAQ https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ
4. Solidity - dokumentacj https://solidity.readthedocs.io/en/v0.4.24/
5. Awaria w Nazwa.pl – klienci stracili dane, także z backupów, https://niebezpiecznik.pl/post/awaria-w-nazwa-pl-klienci-stracili-dane-takze-z-backupow/
6. Teatr Współczesny zhackowany? Niestety to nie happening https://niebezpiecznik.pl/post/awaria-w-nazwa-pl-klienci-stracili-dane-takze-z-backupow/
7. Wyszukiwarki studentów, publiczna lista usterek i niebezpieczne punkty ksero, czyli uczelnianych wpadek cz. IV https://niebezpiecznik.pl/post/wyszukiwarki-studentow-publiczna-lista-usterek-i-niebezpieczne-punkty-ksero-czyli-uczelniany
8. Skutki utraty dokumentacji podatkowej http://www.ordynacjapodatkowa.pl/artykul,1679,5273,skutki-utraty-dokumentacji-podatkowej.html
9. Jak pozyskać dane osobowe i adresy dzieci z twojej okolicy? https://niebezpiecznik.pl/post/jak-pozyskac-dane-osobowe-i-adresy-dzieci-z-twojej-okolicy/
10. Predicting Random Numbers in Ethereum Smart Contracts https://blog.positive.com/predicting-random-numbers-in-ethereum-smart-contracts-e5358c6b8620
11. OpenZeppelin https://github.com/OpenZeppelin/zeppelin-solidity