Transkrypcja gawędy o Ethereum - bez ilustracji

To jest transkrypcja drugiego odcinka gawędy o kryptowalutach.
Prelekcję prowadzi Tomasz Zieliński, autor bloga www.informatykzakladowy.pl

Uwaga! Tekst jest zapisem godzinnego monologu. Aby lepiej się go czytało, dokonałem pewnych zmian i modyfikacji - dotyczy to głównie powtórzeń, drobnych omyłek i doboru słów, ale także zwięzłości wywodu. Pominąłem też fragmenty mające sens jedynie dla widzów słuchających gawędy na żywo.

Linki w tytułach rozdziałów prowadzą do odpowiednich fragmentów nagrania wideo, dostępnego pod adresem https://youtu.be/iRqEdzh-frw

[00:00:00] Wprowadzenie

Dzień dobry, dzień dobry! Witam wszystkich radiosłuchaczy zgromadzonych przed radioodbiornikami i youtubami. Nazywam się Tomek Zieliński, prowadzę bloga informatykzakładowy.pl a dzisiaj będzie druga część gawędy o kryptowalutach. Gawędy, bo nie lubię słowa webinar - więc te swoje wystąpienia nazywam gawędami. Na pewno będzie też i trzecia część, bo gdy spisałem wszystko, o czym jeszcze chciałbym powiedzieć, no to tego materiału wyszło strasznie dużo. Planuję dziś zrobić inaczej niż poprzednio, czyli nie cisnąć wszystkich slajdów, lecz mówić około jednej godziny. A co się nie zmieści, przeskoczy do następnego odcinka.

Wyniki ankiety a w zasadzie konkursu. Jeśli ktoś interesuje się kryptowalutami, to widział, że przedwczoraj miało miejsce gwałtowne załamanie rynku. Kursy Bitcoina poleciały jakieś 40% w stosunku do zeszłego tygodnia, więc ogólnie wszędzie krew i flaki. Wiele jest osób, które

zastanawiały się, czy już sprzedawać, czy jednak trzymać. Wczoraj rano zrobiłem taką ankietkę, czy jak będziemy dzisiaj zaczynać o 20:00 to będzie drożej czy taniej niż wczoraj rano.

Tak więc przełączmy się na Google i zobaczmy. Refresh… 37400 dolarów za 1 BTC. Rano wydawało mi się, że jednak będzie więcej, bo około południa było ponad 41 tys. USD, ale jak teraz mamy 37 tysięcy to jest to istotnie mniej niż 39 tysięcy, czyli ponad połowa ludzi, którzy odpowiedzieli na twitterową ankietę, miała rację. Lepiej niż rzut kostką.

Ponownie ostrzeżenie: bardzo, bardzo proszę, żeby nie kupować kryptowalut, dopóki nie nauczycie się więcej na ich temat. To co będę dzisiaj opowiadał, nie da wystarczającej wiedzy a można się było bardzo boleśnie przekonać, że kursy nie tylko rosną, ale i spadają. Wiele osób, które kupiło bitcoiny w zeszłym tygodniu, straciło prawie połowę na swojej inwestycji. W stosunku do pierwszej gawędy, która była w styczniu, też mamy spadek. Wtedy było około 40000 USD, teraz mamy 37000 USD, więc przez cztery miesiące się nie poprawiło. Jeśli ktoś mimo to zdecyduje się na zakup kryptowalut, bardzo proszę, żeby nie wydawał więcej pieniędzy, mógłby stracić na zawsze. No i żeby najpierw się uczyć. Jeśli chodzi o Ethereum, istnieje możliwość nauki w tak zwanym testnecie. Do tego jeszcze dojdziemy.

[00:05:12] Przypomnienie części pierwszej

Krótkie przypomnienie terminów. Na przykładzie grupy anarchistów opowiedzieliśmy, czym jest rozproszony rejestr. Był to mechanizm, w którym mieliśmy wiele kopii tych samych zapisków u wielu różnych ludzi. Dodatkowo mieliśmy jeszcze coś takiego, jak cyfrowe skróty (kryptograficzne hasze), którymi podpisywaliśmy każdą „stronę”, gdy się kończyła, i taki podpis przenosiliśmy na nagłówek nowej strony. Dzięki temu konstruowaliśmy tak zwany blockchain, czyli łańcuch bloków, w którym potrafiliśmy zidentyfikować próby fałszerstwa.

Na przykładzie tej grupy anarchistów pokazaliśmy, że jeśli oni się nie lubią, ale łączy ich jakiś wspólny interes (i co najmniej połowa członków sieci jest uczciwa), to rejestr taki będzie się trzymał kupy i będzie mógł służyć jako podstawa do ich wzajemnych rozliczeń.

Powiedzieliśmy też, że kryptowaluta Bitcoin, która opiera się na rozproszonym rejestrze połączonym blockchainem, działa w modelu komunikacji sieci peer-to-peer (P2P). To nie oznacza, że lokalizacja albo numery IP łączących się ze sobą serwerów to tajemnica. Chodzi o to, że nie ma jednego centralnego serwera, którego wyłączenie skutkowałoby załamaniem tego systemu płatności. Węzły sieci łączą się ze sobą w dość losowy, ale mocno spleciony sposób, to znaczy gdy dowolny z nich nagle zniknie, to takie zdarzenie nie spowoduje żadnej awarii. Inne maszyny, które z nim rozmawiały, nadal komunikują się ze sobą bezpośrednio.

Mówiliśmy o kryptografii klucza publicznego. W jej ramach mamy klucze prywatne, przechowywane w sekrecie, oraz klucze publiczne. W kryptowalutach publiczny jest adresem, czyli jakby numerem konta (pamiętajcie - właśnie w tym fragmencie pierwszej

części gawędy było najwięcej uproszczeń). Matematyka pozwala nam wygenerować dowolną liczbę takich par kluczy. Dyspozycja przelania jakichś środków z danego adresu może być dokonana tylko poprzez złożenie cyfrowego podpisu kluczem prywatnym (powiązanym z tym adresem). Może tego dokonać tylko dysponent klucza prywatnego. Jeśli klucz zostanie utracony, to już nikt nigdy nie będzie mógł z takich środków skorzystać.

Proof of work to mechanizm, który sprawia, że nowy blok w sieci bitcoin pojawia się średnio co około 10 minut. To były te zera na początku hasza. Im więcej tych zer jest w danej chwili wymaganych przez protokół Bitcoina, tym trudniej jest wygenerować taki blok, którego skrót będzie spełniał wymagania. To ma się udać jednemu z próbujących na całym świecie raz na 10 minut. Mechanizm się sam reguluje, więc stosownie do tego czy mocy obliczeniowej w sieci przybywa czy ubywa, no to ta regulacja trudności będzie szła w jedną albo drugą stronę.

Pamiętaliśmy, że bloki (czyli „kartki" anarchistów, czyli pomarańczowe segmenciki na obrazku powyżej) przechowywały tylko transakcje, czyli informacje o przeniesieniu środków z jednego adresu na drugi. Nie ma innego sposobu by sprawdzić, na którym adresie leżą jakieś środki, niż prześledzenie wszystkich bloków od samego początku do samego końca. Oczywiście jednocześnie będziemy weryfikować prawidłowość hashy, by upewnić się, że posiadana kopia blockchaina nie została w żaden sposób sfałszowana. Przetwarzając kolejne bloki i patrząc na kolejne transakcje, będziemy w stanie pamiętać i aktualizować salda środków trzymanych pod poszczególnymi adresami.

Oczywiście całość sieci Bitcoin jest napędzana chciwością. Górnicy, czyli ci, którzy używają swojego (lub nieswojego) prądu i swojej mocy obliczeniowej, by generować nowe bloki, oni chcą dostać za to nagrodę. No i nagrodę dostają. Po pierwsze od samej sieci, która pozwala im pozyskać darmowe bitcoiny przy wytworzeniu każdego nowego bloku, ale również jako prowizję od tych transakcji, które będą umieszczone w danym bloku. Gdyby nie prowizje, górnicy tworzyliby by same puste bloki no, ale skoro umieszczenie tam transakcji nie zwiększa kosztów, a zwiększa zyski, to transakcje się tam znajdują i mamy nowe bloki z transakcjami i cały Bitcoin się kręci.

[00:10:58] Spis treści części drugiej

O czym dzisiaj będzie? Jak widać, lista rzeczy do opowiedzenia jest dość długa. Mam nadzieję, że te pierwsze cztery punkty do kreski uda się zrobić dzisiaj, no a cała reszta no to już w części trzeciej [nie udało się, w tym odcinku skończyliśmy po trzecim punkcie]. Mam nadzieję, że nie będzie potrzebna część czwarta, bo inne tematy na gawędy też mam. Kryptowaluty są fajne, ale nie aż tak fajne [część trzecia była ostatnia]

[00:11:34] Forki kryptowalut

Forki - czyli rozgałęzienia, secesje. Skąd się biorą? Dlaczego mamy więcej niż jedną kryptowalutę mającą w nazwie „Bitcoin”?

Pomyślmy sobie, co by było, gdyby był w protokole Bitcoina albo w bieżącym oprogramowaniu, którego wszyscy używają, był jakiś błąd. Na przykład: mimo że w jednym bloku da się zmieścić 1000 transakcji, to oprogramowanie wsadza co najwyżej 500. Gdy taki błąd zostanie poprawiony, gdy wydana zostanie nowa wersja oprogramowania, nic się tak naprawdę nie zmieni. Ci, którzy mają nową wersję, będą kopać bloki w sposób trochę bardziej opłacalny, bo dostaną prowizje od większej liczby transakcji, ale dla pozostałych nic się nie zmieni. Pomyślmy jednak, co dzieje się, gdy modyfikacji ulega sam protokół. Nie jest bowiem tak, że odkąd 10 lat temu Satoshi Nakamoto zniknął, Bitcoin pozostał niezmieniony. On ewoluuje cały czas - istnieje grupa ludzi, którzy prowadzą prace rozwojowe.

Ci ludzie niekiedy wymyślają zmiany, które pozostają zgodne wstecz, czyli stare oprogramowanie będzie działać z nową wersję protokołu. Każdy, kto projektował jakiś protokół komunikacyjny albo format danych, odkrył zapewne, że:
1) fajnie w takim formacie zawrzeć, najlepiej na początku, informację o użytej wersji;
2) dobrze w definicji pierwszej wersji pozostawić w nagłówkach trochę pustych bajtów, pustego miejsca oznaczonego jako „do wykorzystania w przyszłości” no i umówić się na przykład, że w tej pierwszej wersji to będą zawsze bajty wypełnione zerami.

Wyobraźmy sobie, że mamy wersję numer jeden protokołu bitcoinowego i przychodzi nam do głowy pomysł na dołączanie do transakcji emotikonek. Uwaga - to przykład całkowicie z głowy. Bezsensowny, ale łatwy do zrozumienia. Chcemy więc, aby taka zmiana, taka aktualizacja protokołu, weszła w życie jak najłagodniej. Umawiamy się zatem, że z zapasowych bajtów, które nie były do niczego używane, weźmiemy sobie na przykład dwa i tam będziemy zapisywać numer emotikonki, którą dołączamy do danej transakcji. Ustalamy też, że same zera w tych dwóch bajtach to jest po prostu brak emotikonki (albo jakaś emotka neutralna).

Gdy taka nowa wersja oprogramowania wejdzie w życie, jej użytkownicy będą w stanie ustawiać sobie te emotikonki w swoich transakcjach. Natomiast ci wszyscy, którzy nie zdążą albo nie zechcą instalować aktualizacji - będą nadawać transakcje, które będą miały wyłącznie zera, czyli sygnał „brak emotki”. Ale - to będzie działać! Oczywiście tylko użytkownicy nowej wersji będą mogli widzieć i ustawiać emotki przy transakcjach. Pozostali nie będą niczego widzieć, mogą nawet nie wiedzieć o takich funkcji, ale nadal pozostaną członkami sieci, kopanie nadal będzie działać. To jest „soft fork” czyli zmiana zgodna wstecz.

Niestety - czasem zdarzają się zmiany, które stanowią tak zwany „hard fork”. Jest to taka zmiana protokołu, która wymusza aktualizację oprogramowania, bo stara wersja nie będzie rozumiała nowej. Mówiliśmy o pomyśle Satoshiego, że bloki Bitcoina nie mogą być większe niż jeden megabajt. To już nie jest prawda. Po drodze mieliśmy aktualizację, która zmieniła trochę sposób liczenia rozmiaru i sprawiła, że największe bloki mogą mieć obecnie nawet ponad 2 megabajty.

Oprogramowanie sprzed 10 lat nie byłoby w stanie poprawnie działać, bo gdyby dostało taki blok do weryfikacji, to blok ten byłby odrzucony jako niezgodny z (przestarzałymi dziś)

regułami. Dlatego właśnie hard forki to zmiany, na które wszyscy muszą zgodzić się zawczasu, żeby zmiana łamiąca kompatybilność mogła wejść w życie od jakiegoś konkretnego numeru bloku i by od tej chwili wszyscy stosowali nową nową wersję oprogramowania.

No i to jest trochę problem. Jak głosować, skoro to jest kryptowaluta, która ma być anonimowa a każdy anonimowy chętny może być niezależnym operatorem niezależnego węzła? Jak w ogóle możliwe jest przeprowadzenie jakiegokolwiek głosowania? Okazuje się, że jest to łatwe. W oprogramowaniu do kopania możemy bowiem zrealizować następującą funkcję: gdy pojawia się nowa propozycja zmian o nazwie BIP-123, to oprogramowanie będzie mogło rozgłaszać „tak ja jestem gotów, aby zmianę BIP-123 wprowadzić i zgadzam się na jej wejście w życie". Ci, którzy nie chcą, ustawią sobie tę flagę na wartość negatywną, czyli będą rozgłaszać, że oni głosują przeciwko.

Zaznaczmy, że tu nie ma takiej demokracji w formie „50%+1 głos". Aby uniknąć secesji, czegokolwiek co mogłoby doprowadzić do kłótni czy rozłamu, głosowania dotyczą funkcji naprawdę potrzebnych oraz rozwiązujących istotne ograniczenia - zaś progi akceptacji zmian ustala się na poziomach rzędu 95%. Jeśli tyle osób zgodzi się na jakąś propozycję w jakimś terminie, no to wtedy ona automatycznie wchodzi w życie i od oznaczonego bloku protokół się zmienia. Ci, którzy nie zaktualizują w porę oprogramowania, od tego właśnie bloku przestaną brać czynny udział w sieci, bo ani nie będą w stanie weryfikować nowych bloków, ani nie będą w stanie generować własnych, które większość sieci uznałaby za poprawne.

Czasem może się zdarzyć, że ktoś się nie podporządkuje. Stwierdzi, że ma być po ichniemu, sąd sądem, demokracja demokracją, ale rozdzielamy się. No i tu możliwe są dwa warianty - pierwsza możliwość to przyjęcie zmiany odrzuconej przez ogół. Przykładem może być na przykład Bitcoin Cash. Swojego czasu powstała propozycja, by zwiększyć rozmiar bloku z 1 do 10 MB bez większej zmiany w samych bebechach protokołu. Ta zmiana nie została przyjęta przez ogół, bo nie rozwiązywała innych istotnych problemów z przepustowością Bitcoina. Była jednak grupa ludzi, która stwierdziła, że robimy po swojemu - i zrobili forka, rozgałęzienie. Ustalili, że od danego bloku idą na swoje czyli będą rozmawiać tylko między sobą, będą sobie kopać te bloki o rozmiarze 10 MB i oczywiście rywalizować w ich kopaniu.

Druga okazja do secesji to obstawanie przy starej wersji protokołu. Przykład - społeczność zgadza się na nowy sposób liczenia rozmiaru, jednak są ludzie dla których jeden megabajt wymyślony przez Satoshiego to świętość i przykazanie po wieki, więc gdy protokół się zmienia, oni pozostają przy swojej wersji sprzed modyfikacji.

Ładnych parę lat temu, w 2014 czy 2015, pojawiły się generatory oprogramowania do kryptowalut. Podawało się nazwę, symbol, liczbę monet, warunki ich generowania, klikało się Enter i dostawało gotowe źródła koparki tej nowej kryptowaluty. W przypadku forków nie mówimy jednak o zakładaniu nowej kryptowaluty, ale wypączkowaniu jej z kryptowaluty istniejącej wcześniej.

Efekt uboczny - gdy wydzieli się nowa kryptowaluta (na przykład Bitcoin Cash albo Bitcoin Classic) to znienacka ci wszyscy, którzy uprzednio mieli bitcoiny w „głównym łańcuchu",

mają nagle kryptowaluty w obu łańcuchach. To nie znaczy, że od razu mają dwa razy więcej dolarów. Wcale nie jest powiedziane, że taka secesja się przyjmie. Po pierwsze, nowa kryptowaluta musi pojawić się na jakiejś giełdzie, żeby dało się ją wymieniać na inne krypto albo na dolary. Po drugie - musi być dostatecznie dużo kopiących, by taka kryptowaluta była odporna na atak double spend. To realne ryzyko dla tych z setek lub tysięcy małych kryptowalutek (altcoinów, shitcoinów), których wartość nagle rośnie. O tym powiemy w części trzeciej. Z tego samego powodu giełdy starają się operować tylko na kryptowalutach, które mają jakąś tam płynność i moc obliczeniową.

Jeśli ktoś miał bitcoiny przed 2017 rokiem i nie interesował się nimi za bardzo, no to może warto wrócić do tematu, bo jest posiadaczem jeszcze kilku innych coinów. W sieci można znaleźć ilustracje pokazujące, jak wyglądały forki Bitcoina, które były „miękkie" a które „twarde" itp. Nie wszystkie te ilustracje są czytelne, niektóre mylą nazwy kryptowalut z funkcjami czy cechami protokołu, które weszły lub nie weszły w życie.

Gdy spojrzałem z tydzień temu na bardzo dużą giełdę binance.com, no to tam było ileś różnych walut mających w nazwie „Bitcoin". Widzimy, że ten prawdziwy bitcoin (wtedy kosztował 57000 $ za sztukę) miał kapitalizację rzędu miliona milionów dolarów. Ale inne pokrewne mu kryptowaluty kosztowały 1000 albo 100 albo wręcz kilka dolarów przy kapitalizacji o rzędy wielkości mniejszej. Zdawałoby się, że to jest trochę bez sensu, żeby takie malutkie kryptowaluty istniały. No bo po co?

Ale tu ciekawa sprawa - istnieje oprogramowanie dla górników, które sprawdza bieżącą wartość różnych kryptowalut i automatycznie przerzuca się na kopanie tego wariantu, który jest najbardziej opłacalny. Można więc puścić koparkę wolno, by to ona decydowała o wydobywanej kryptowalucie. Minus jest taki, że nasz urobek też będzie w różnych kryptowalutach, więc zanim wyciągniemy dolary, będziemy je sobie musieli wymienić na inne krypto albo gdzieś spieniężyć. Nie zawsze się to uda - zwłaszcza gdy wykopany shoitcoin wypadnie z jakiejś giełdy albo po prostu nie będzie na nim dostatecznej płynności (aby sprzedać - ktoś musi chcieć kupić).

Pytanie od słuchacza: *kto odpowiada za takie aktualizacje i wdrożenie ich w Bitcoinie?*

Odpowiedź: grupa ludzi, którzy reprezentują największych kopaczy, tych od wielkich hal z koparkami. Ci możni ludzie dbają o to, żeby usuwać przeszkody, które mogłyby stanąć im na drodze. Więc są to osoby związane głównie z posiadaczami dużej mocy obliczeniowej albo ci, którzy byli w tym biznesie od dawna i są szanowani i słuchani.

[00:27:43] Kryptowaluta Ethereum

Ethereum - albo eter. Ja będę mówił raczej „Ethereum" (fonetycznie: itirium) o kryptowalucie, a o jednostkach - „eter". Bo to chyba bardziej po polsku.

Najpierw będzie dygresja. Jeśli chcieliście kupić jakąś kartę graficzną, w ciągu ostatniego - powiedzmy - roku, i ze zdziwieniem się przekonaliście, że nie ma, to w połowie odpowiada

za to pandemia, przerwanie łańcucha dostaw, zmniejszona z tych przyczyn produkcja. No ale druga połowa odpowiedzialności spada na górników i to, że ceny kryptowalut wzrosły - bo eter kopie się na kartach graficznych i górnicy są w stanie zapłacić za nie każdą cenę. A jak fabryki są w Chinach i nielegalne kopalnie też były w Chinach, to górnicy mają producentów bliżej. Nie trzeba wozić towaru do Europy, sprzedaż na miejscu opłaca się bardziej. Producenci będą mówić, że im się popyt ze strony górników nie podoba, ale oni tak naprawdę sprzęt produkują, sprzedają i zarabiają niezależnie od tego, kim jest nabywca. Koniec dygresji.

Czy w sieci bitcoin da się zapisywać jakieś informacje? Oczywiście oprócz informacji o nadawcy, odbiorcy i kwocie? Teoretycznie nie; w praktyce jesteśmy w stanie coś wymyślić. Na przykład, jeśli zrobimy sobie z jednego adresu taką serię przelewów:

… to może się okazać, że mieliśmy w tym jakiś pomysł i na przykład niezależnie od tego, w jakiej kolejności te transakcje się przetworzą, to będziemy w stanie odczytać kolejność zaznaczoną poniżej na żółto a kody ASCII przekazu z cyferek zaznaczonych na zielono. No i wtedy, kosztem kilku tysięcy dolarów włożonych w takie transakcje, bylibyśmy w stanie zakodować pięć bajtów składających się na moje imię. I to niestety nie ma za dużo sensu. Natomiast spostrzeżenie jest takie, że da się.

W roku 2015 pojawił się pomysł utworzenia kryptowaluty, w której rejestrze dałoby się zapisywać nie tylko dane, ale także algorytmy. I wtedy taka kryptowaluta mogłaby coś robić automatycznie. Można by opisać jakieś zachowania, które mają się wytworzyć czy zadziać w jakiś określonych okolicznościach. I faktycznie taka kryptowaluta powstała - to właśnie Ethereum.

Tym razem twórca jest znany, jest nim Witalij Buterin. Miał on naprawdę świetny pomysł, bo to nie było łatwo wymyślić. Wówczas było już wiadomo, że Bitcoin „udał się”, to znaczy spełniał założenia - jest względnie anonimowy, nie da się go łatwo wyłączyć a jego wartość może rosnąć. I w takiej sytuacji wymyślono właśnie troszkę lepszą kryptowalutę, która stanowi nie rozproszony rejestr, tylko rozproszony komputer. Co to znaczy, opowiem za chwilę.

Tak czy owak nadal mamy bloki, nadal mamy kopalnie i górników, którzy rywalizują o wygenerowanie kolejnego bloku dołączanego do blockchaina. Różnica jest taka, że w Ethereum transakcja to będzie troszkę więcej niż sam tylko przelew środków. Gdy patrzyliśmy na Bitcoina, to czytaliśmy wszystkie bloki od początku do końca, czytaliśmy transakcje, sprawdzaliśmy adresy źródłowe i docelowe, weryfikowaliśmy kwoty, prowizje, legalność i poprawność na poziomie transakcji i bloków. Mając ustalony stan blockchaina, dostawaliśmy nowy blok, sprawdzaliśmy jego poprawność, dołączaliśmy na koniec łańcucha i mieliśmy nowy ustalony stan bieżący.

Praktycznie identycznie jest w Ethereum, tylko że tam pojedyncza transakcja ma w sobie trochę więcej rzeczy. Nadal są adresy źródłowe i docelowe, kwota, ale już prowizja będzie wyznaczona trochę inaczej - jako limit, nie kwota określona na sztywno. Mamy też opcjonalne dane dodatkowe, które będą służyły do uruchamiania obliczeń, których wynik jest składowany w blockchainie. Między tym lewym dolnym blokiem niebieskim a prawym

dolnym będziemy mieli nie tylko przeksięgowania, ale także uruchamianie programów z blockchaina, które będą wpływać na wynikowy stan blockchaina.

Eter kopie się na kartach graficznych. Autorzy aktywnie utrudniają zrobienie jakichkolwiek ASIC-ów, bo nie chcą, żeby istniały dedykowane układy, których cena jest wysoka, zużycie prądu makabrycznie wysokie - za to wydajność tysiące i miliony razy większa od komputerów. Wkładane są więc wysiłki, by do kopania nadal wymagany był zwykły komputer z kartami graficznymi.

Gdy przypomnimy sobie Bitcoina, to zauważymy, że mamy tam do czynienia z jednym rodzajem adresów, dla których śledzimy stan salda. Blockchain Ethereum ma troszkę bogatszy tutaj repertuar, bo tam są dwa rodzaje adresów: 1) taki a'la bitcoinowy, powiązany z parą kluczy pozwalających na zarządzanie środkami; 2) adresy, które są generowane dla programów osadzanych w blockchainie i których saldem rządzi wyłącznie taki program. Środki można przelewać na oba rodzaje adresów.

O tym, jaki sposób te programy działają, w jaki sposób się przedostają do blockchaina, będę mówił za chwilę. W blockchainie Etherum można przechowywać dane - ale już nie w taki partyzancki sposób, jaki pokazałem wcześniej. Tu mamy prawdziwe struktury danych, prawdziwe typy danych i jesteśmy w stanie sobie na przykład zapisać tablicę liczb o jakimś rozmiarze i składować taką informację w blockchainie.

Oczywiście, nawet jeśli ten zapis będziemy chcieli skasować albo zmienić, czyli zapiszemy nową wersję tej samej informacji w kolejnym bloku, to stara wartość nie zniknie z oczu obserwatorów, będzie na zawsze obecna w jednym z wcześniejszych bloków. Wszystkie dane raz wstawione do blockchaina, każde wywołanie osadzonego programu oraz efekt jego działania - pozostaną widoczne dla wszystkich zainteresowanych.

Przykładowy program działający na sieci Ethereum. Jeśli kiedyś słyszeliście o smart kontraktach, to właśnie są programy, które uruchamia się w sieci kryptowalutowej Etherum. Kontrakt to jest po prostu program, tutaj po prawej stronie widzimy przykładowy kod źródłowy. Kontrakt (program) trafia do blockchaina skompilowany czyli w troszkę mniej czytelnej postaci. Aby ten program mógł trafić do blockchaina, to ktoś musi wygenerować transakcję, w której opłaci koszt składowania tego programu w blockchainie. Uwaga - identyczny program może trafić do blockchaina w dowolnie wielu identycznych kopiach.

Gdy transakcja zawierająca dyrektywę osadzenia programu zostanie przetworzona, to jednocześnie fakt umieszczenia transakcji w bloku wygeneruje nam unikalny adres tego osadzanego programu. Dopiero wówczas będziemy znać ten adres i będziemy mogli z niego korzystać. Jeśli do blockchaina trafi wiele kopii identycznego programu, każda będzie miała unikalny, odrębny adres. Jedyna możliwość, żeby użyć jakiegoś programu, czy też wywołać jego funkcję, to jest wysłać na jego adres jakąś transakcję. Ona nie musi przenosić środków, których właścicielem stałby się program, choć oczywiście jest to możliwe. Natomiast na pewno trzeba będzie opłacić koszty prowizji.

Ten program, który widzimy zielonej ramce, jest bardzo prosty. Symbol storedData to pojedyncza zmienna liczbowa, jaką ten program będzie sobie przechowywał w blockchainie. Mamy funkcję „set", która ustawia nową wartość tej zmiennej (funkcję „get" na razie

pominiemy). Gdy taki program trafi do blockchaina, to każdy będzie w stanie wywoływać tę funkcję. Nie ma żadnych zabezpieczeń sprawdzających, czy użytkownik jest uprawniony do takiej akcji. Każdy będzie mógł wywołać tę funkcję w ramach swojej transakcji i sprawić, by zmienna przyjęła zadaną wartość. Jeśli zrobi to wiele osób - wygrywa ostatnia transakcja w bloku i to właśnie ona decyduje, jaka wartość zostanie ustalona po jego przetworzeniu. Taki kontrakt (program) nie ma sensownego zastosowania, ale jest bardzo prosty.

Powiemy sobie teraz co trzeba zrobić, żeby uruchomić funkcję. Jeśli jesteśmy kimś, kto ma trochę eteru i chce ten eter zużyć na wywołanie takiej funkcji, to robimy tak. Tworzymy nową transakcję, w której deklarujemy jakąś prowizję (o tym zaraz), wskazujemy kontrakt i funkcję do wywołania (czyli na przykład funkcję „set") i określamy, jakie argumenty (parametry) otrzyma ta funkcja. Przykład - jako argument „x" ustalimy liczbę 1000.

I teraz przetworzenie tej transakcji przez górników będzie polegało na tym, że oni zdekodują, o który kontrakt i funkcję chodzi, zdekodują argumenty i zaczną wykonywać tę funkcję kontraktu. Ten kod się wykonuje i wykonuje, a jak skończy (tutaj byłaby tylko jedna linijka do wykonania), to wtedy jego rezultat (tutaj - wartość zmodyfikowanej zmiennej) jest zapisywany do blockchaina. Podobnie, jak przetwarzanie transakcji w Bitcoinie polegało na przeksięgowaniu środków z jednego adresu na drugi, tak jest i tu, ale efekt będzie troszkę bardziej skomplikowany. Mogą się zmienić na przykład zapisane zmienne, może się wykonać jakaś inna operacja - ale wszystkie argumenty muszą wejść transakcją, a wszystkie dane wyjściowe i trwałe efekty wykonania muszą być zapisane w blockchainie.

Koszt i pracochłonność. Różne operacje potrzebują różnej mocy obliczeniowej, mają też różny czas wykonania. Czas i koszt wykonania funkcji będą zależeć do argumentów, np. funkcja obliczająca pierwiastek zużyje więcej zasobów przy dużych liczbach - musimy więc różnicować prowizję jaką dostanie górnik. Pierwiastkowanie małej liczby da małą prowizję, dużej liczby - dużą prowizję.

Maksymalna prowizja jest deklarowana przez wywołującego funkcję. Wyraża się ją w dwóch etapach - po pierwsze przez określenie liczby jednostek „paliwa" (po angielsku: gas) które przekładają się wprost na dozwolony czas / złożoność obliczeń. Wywołujący deklaruje też, ile eteru zapłaci za każdą jednostkę paliwa - limit paliwa pomnożony przez koszt jednego paliwa to pełna kwota maksymalnej prowizji.

Jeśli na przykład w programie będzie błąd skutkujący nieskończoną pętlą, to limit paliwa będzie bezpiecznikiem. Gdy zużycie paliwa dojdzie do limitu, przerywamy liczenie, stan blockchaina się nie zmienia, no ale prowizja wpada do kieszeni górnika. Czemu? Choć program przerwał wykonanie, to praca górnika związana z tworzeniem bloku została wykonana. Wysiłek (prąd, czas) włożono, więc nawet przy przerwanym wykonaniu programu prowizja nadal jest pobierana, natomiast wszystko inne jest wycofywane.

Ethereum dzieli się na znacznie więcej części, bo jeden bitcoin to było 10^9, czyli dziesięć miliardów satoshi. Tutaj mamy 10^{18} więc o wiele, wiele więcej, to znaczy granularność tej kryptowaluty jest znacznie większa. W przeciwieństwie do prowizji za przelew, mamy prowizję za zużyte paliwo, czyli za pracę jaką górnik musiał włożyć. Blok pojawia się co 12 sekund, czyli znacznie szybciej, natomiast jest w nim znacznie mniej transakcji, w efekcie

przepustowość Ethereum jest raptem tam dwa-trzy razy wyższa od Bitcoina, czyli nadal straszliwie mała.

Co ciekawe, nie ma tutaj czegoś takiego jak górne ograniczenie liczby etherów które się pojawią. Ludzie, którzy ten protokół utrzymują i rozwijają, decydują np. o zmniejszeniu prowizji od nowego bloku z 3 eterów na 2 etery ale nie ma arbitralnego limitu. Co ciekawe ograniczanie podaży nie jest konieczne żeby wartość eteru rosła.

Przeliczenie kosztu paliwa na rzeczywiste etery - to jest też ciekawy pomysł, to znaczy nie możemy powiedzieć, że jeden gas (jedno paliwo) to jest tam umowny jeden cent, bo jeśli wartość etheru wzrośnie bardzo dużo, to wtedy ludzi nie byłoby stać na jakiekolwiek transakcje, po prostu za dużo by kosztowały. Mamy troszkę inny model, w którym każdy, kto chce wykonać jakąś operację i robi transakcję, która ma jakiś tam limit wykonania, on jednocześnie deklaruje, ile jest w stanie zapłacić za jedno paliwo, czyli za tą jednostkę czasu.

To nie jest wtedy do końca takie jasne dla górników, czy bardziej opłacalna będzie transakcja z tym większym jednostkowym kosztem paliwa (bo ona może się wykonać szybciutko i prowizja będzie mała) czy też postawić na jakąś transakcję z programu, o którym wiemy, że on się wykonuje zazwyczaj długo - więc zarobi więcej mimo nieco mniejszej prowizji ze sztukę paliwa.

Dzięki temu, że paliwo, czyli nakład pracy, jest oderwany od ceny za jedno paliwo, to jesteśmy w stanie na przykład wydawać tyle samo za transakcje, nawet jeśli wartość etheru rośnie. Na obrazku widzimy, że jeśli jeden ether wskoczyłby z tysiąca dolarów do dziesięciu tysięcy dolarów, ale nasza hojność, czyli nasza deklaracja ile płacimy za jedno paliwo, zmniejszy się 10 razy, no to per saldo wartość prowizji, jaką płacimy (wyrażona w dolarach) będzie identyczna. Te liczby są oczywiście wymyślone, obecne minimum na transakcję to bodajże 20000 paliwa, ale wiadomo o co chodzi.

Kod aktywowanej transakcji wykona się na wszystkich węzłach Ethereum - bo każdy górnik weryfikuje każdy blok z transakcją, czyli musi ją wykonać i sprawdzić, czy rzeczywiście wszystko się zgadza, czy wynik jest taki sam. To sprawia, że te obliczenia muszą być całkowicie deterministyczne, u każdego na świecie takie same, czyli tam nie może być żadnej funkcji losowej - bo wtedy każdy miałby inny wynik i konsensus byłby niemożliwy.

To sprawia, że ciężko jest zaprogramować jakąkolwiek loterię w sieci Ethereum. Nie dlatego, że nie ma źródeł losowości, bo na przykład hash poprzedniego bloku będzie losowy, nie możemy go zawczasu przewidzieć. Problem leży gdzie indziej - jeśli mamy taką loterię i każdy górnik chciałby ją wygrać, to ów górnik zrobi tak: spojrzy na hash poprzedniego bloku, sprawdzi wynikający z niego wynik loterii i dowie się, czy wygrywa. Jeśli nie wygrywa to on tej transakcji w ogóle nie wstawia do kopanego bloku. Transakcja kończąca loterię mogłaby nie zostać nigdy wykonana, bo nikłe są szanse, by akurat bieżącemu potencjalnemu zwycięzcy (jedynemu, który wstawi do bloku transakcję kończącą) udało się wykopać nowy blok.

Analogicznie - ciężko sensownie oprzeć przetwarzanie o datę lub godzinę (numer bloku zawsze rośnie o jeden). Jeśli chodzi o datę wykonania, węzły umieszczają ją w bloku i tutaj

jedyny jedyny wymóg jest taki, że data musi być większa od poprzedniej. Ktoś teoretycznie mógłby sobie przelecieć po przyszłych wartościach i jeśli ma zapaść jakiś wynik loterii, która się kończy za miesiąc i on wygrywa akurat to spróbować wystawić blok z datą za miesiąc. Na szczęście inne węzły odrzucą coś takiego jako oszustwo, nie dołączą takiego bloku do blockchaina, nie zdoła się on rozpropagować.

Kontrakty mogą wywoływać inne kontrakty. Jest to dość kosztowne, ale możliwe. Bieżąca wersja protokołu mówi, że transakcje w jednym bloku mogą w sumie przepalić do 15 milionów paliwa (ten parametr sieci Ethereum zmienia się w czasie).

Wiele osób chce korzystać z Ethereum, więc koszt czasem mocno rośnie, tzn. rośnie deklarowana hojność za jedno paliwo. To jest slajd sprzed 11 maja - gdy sobie popatrzymy na koszt zapisu danych w blockchainie hojność transakcji tego dnia, to okazuje się, że zapisanie jednego bajtu to jest ponad pół dolara, czyli jakiejś 600 dolarów za kilobajt. Drogo. Można pomyśleć, że będzie skrajnie drogo, jeśli jakieś zawirowanie podbije cenę dziesięciokrotnie, prawda?

Tak więc wczoraj mieliśmy gwałtowne załamanie kursu Ethereum i te największe wieloryby chciały przepchnąć swoje transakcje po naprawdę wysokich cenach. Oni robili przebicie nie 10x zwyczajne wartości, tylko 1000x zwyczajne wartości. Widziałem np. taką transakcję, w której ktoś za przywalenie przez giełdę eteru wartego 2 miliony dolarów zadeklarował 80000 USD prowizji (36-38 eterów). No to był dobry dzień również dla górnika, który wykopał taki blok. Normalne prowizje dla górnika to 2-3 etery za blok. No a tutaj nagle było 40 więc suto.

Oczywiście giełdy kryptowalut od razu się pozawieszały, wszelkie stronki z kursami i statystykami również. Nikt się nie potrafił wystarczająco wyskalować. Jak jest fajnie to jest fajnie, ale gdy zaczyna się jakaś jazda, to wtedy tylko najbogatsi są w stanie zrealizować to, co rzeczywiście sobie zamierzyli, a cała reszta może się tylko przyglądać.

Podsumowanie: w eterze mamy 2 rodzaje adresów - posiadany przez ludzi oraz przypisany do kontraktów. Kontrakt to jest program, który został zapisany w blockchainie. Nowy blok co 13 sekund, około 200 transakcji na blok no i prowizje zależą od hojności zlecającego oraz rzeczywistego nakładu pracy wymaganego do wykonania obliczeń przez dany kontrakt (daną funkcję z danymi parametrami).

Pytania od uczestników

Jestem ciekaw szczegółów, dlaczego nie da się stworzyć ASICów do Ethereum.

Da się - ale delikatna modyfikacja protokołu sprawi, że one przestaną działać. Jeden twardy fork i ASIC jest do wyrzucenia, a jego zaprojektowanie i wyprodukowanie to jest bardzo droga sprawa. A skoro wiemy, że twórcy protokołu będą z ASICami walczyć, to nikt ich nie produkuje, bo to nie ma sensu.

Czy treść smart kontraktu jest przechowywana w postaci zrozumiałej czy skompilowanej?

Jest skompilowana. Można ją oczywiście zdekompilować. Nie będzie to wyglądało tak ładnie jak przykład w Solidity, będzie bardziej przypominać jakiś asembler, ale da się to robić. Więc

tak - są narzędzia do pracy z tym kodem. Można oczywiście podglądać cudze smart kontrakty i szukać w nich błędów, bo wszystkie są publiczne i dokładnie widać co i jak robią.

Ile obecnie waży cały blockchain Etherum

On jest trochę mniejszy od blockchaina Bitcoina, chyba jakieś obecnie 280 GB. Rośnie nieco szybciej, ale to nie są dramaty. Pojemność czy cena jednostkowa twardych dysków wydają się rosnąć szybciej niż blockchain Ethereum.

[00:58:17] Smart kontrakty (z przykładami)

Smart kontrakty, które widzieliśmy, nie są programami gotowymi do wykonywania na istniejącym komputerze. Służy do tego tak zwana maszyna wirtualna, to jest nieistniejący komputer, którego działanie trzeba zasymulować. To może być podobne do np. GameBoya, konsolki Nintendo - w dzisiejszych czasach możemy sobie zagrać w te gry z GameBoya na pececie, w Windowsie, emulując działanie sprzętu.

W środku konsolki był fizyczny Zilog Z80, procesor sprzed 40 lat. Tego procesora nie ma w pececie, procesor klasy x86 musi symulować działanie tamtego starego CPU. To jest bardzo bliskie temu, co normalnie określamy jako maszynę wirtualną. Symulujemy zmiany stanu całego sprzętu - procesora, pamięci, syntezatora dźwięków - by udawać wykonywanie kodu gry na nieistniejącym Z80.

No i tutaj jest podobnie, to znaczy mamy opisane jak działa maszyna wirtualna EVM (Ethereum Virtual Machine), czyli jak działałby nieistniejący procesor rozumiejący kod smart kontraktów. Każdy może napisać własny symulator, więc istnieją różne implementacje maszyny wirtualnej Ethereum. Język programowania, który widzieliśmy wcześniej, to jest Solidity i on jest najpopularniejszym narzędziem do pisania smart kontraktów. Nie znaczy, że jedynym.

Ekosystem narzędzi jest bogaty. Mamy różne kompilatory. Mamy dekompilatory, które ze skompilowanego bajtkodu tworzą coś względnie czytelnego dla programisty. Są debuggery, które pozwalają wykonać kod krok po kroku i obserwować zmiany stanu, aby upewnić się, że wszystko działa jak należy. Jest coś takiego jak testnet, czyli taka osobna sieć Ethereum w której monety nie mają żadnej wartości. Automaty rozdają tam za darmo testowe ethery, które można zużyć na potrzeby testów. Od strony technicznej testnet działa jednak dokładnie tak samo.

Oczywiście moc obliczeniowa „publicznego” testnetu jest znikoma, bo nikt tam nie wkłada żadnych rzeczywiście dużych mocy, natomiast izolowany testnet możemy stworzyć w całości na własnym pececie. On też pozwoli nam na eksperymenty, które nic nie kosztują. I całe szczęście - przy dzisiejszych cenach nauka i nieudane eksperymenty na głównym blockchainie ETH kosztowałyby krocie.

Kontrakt to taki automat. Do działania wzbudza go przychodząca transakcja. Kontrakty mogą wywoływać się nawzajem, ale pierwotną transakcją będzie zawsze ta pochodząca z adresu należącego do człowieka (kontrolowanego przez klucz prywatny). Kontrakty mogą korzystać z wartości zwracanych przez funkcje z innych wywołanych kontraktów. Nic nie stoi na przeszkodzie, żeby kontrakty zakładały nowe kontrakty. Pierwotny wywołujący musi jednak pokryć pełny koszt całego paliwa całego łańcucha wywołań. Nie może być tak, że smart kontrakt z własnych pieniędzy opłaci dalsze wywołania.

Smart kontrakt ma znikome możliwości interakcji z czymkolwiek. Nie ma dostępu do plików, nie może się odwoływać do żadnych urządzeń, internetu ani czegokolwiek podobnego. Tak naprawdę wie tylko to, co jest w historii blockchaina, ma dane o bieżącej transakcji i… tyle.

Ważne - gdy dany kontrakt trafi do blockchaina, zostanie tam już na zawsze. Nie da się poprawić błędnego kontraktu. Wcześniej pokazywaliśmy przykład z ustawianiem wartości zmiennej. Możliwe jest przygotowanie - zawczasu oczywiście - kontraktu, który będzie miał taką dynamiczną przekierowywaczkę, która dzisiaj może wywołać jeden kontrakt, jutro inny, pojutrze jeszcze inny. No oczywiście taka przekierowywaczka będzie się cieszyła dużo mniejszym zaufaniem niż kontrakt, którego postać jest ustalona na zawsze. Dynamiczne przekierowanie sprawia, że nie możemy przewidzieć rzeczywistych efektów wykonania.

Co może robić kontrakt? No przede wszystkim tak jak mówiłem, ma dostęp do parametrów wywołania, czyli do tej transakcji, która go wywołała. Zna transferowaną ilość eteru, adres pierwotnego źródła transakcji, stan blockchaina, własne zmienne, może też odczytywać inne dane, może wysłać środki czyli zainicjować przelew z tego co sam ma. Kontrakt może być dysponentem eterów, może wywołać inne kontrakty, może zapisywać dane do blockchaina i w sumie niewiele więcej. Oczywiście typowe struktury programów, czyli pętle, rozgałęzienia, skoki - to wszystko jest dostępne. Ale nie więcej. Nie ma oczywiście mowy o żadnym ekranie, sieci, dysku, czymkolwiek.

Przykładowe smart kontrakty

Teraz będzie kilka przykładów, jak mogą działać i co mogą robić smart kontrakty.

Przykład pierwszy to jest skarbonka. I to jest taki kontrakt, który przyjmuje kasę. A jak się wywoła jego metodę, to on sprawdza, czy ma więcej niż 10 eterów (albo jakąś inną liczbę, jaką tam sobie zaprogramujemy). Jeśli ma mniej, to nic nie robi, kończy działanie. Jeśli ma więcej, to wtedy wszystko, co ma - przelewa gdzie indziej. Gdy zbieramy na wakacje, no to możemy sobie ciułać, przelewać po kawałeczku, wiedząc, że dopóki nie uciułamy całości, to nic nie wróci. Ale gdy uciułamy, to w jednej chwili jesteśmy w stanie zainkasować całość środków.

Drugi przykład - maszynka do głosowania, czyli coś, co bardzo chcieliby zrobić politycy, ale nie wiedzą, że sensownie się nie da. Tutaj będziemy mieli do czynienia ze smart kontraktem, który ma dwie funkcje. Pierwsza funkcja tworzy nowe głosowanie i tam będziemy mieli numer głosowania, opis głosowania, listę adresów uprawnionych do oddania głosu oraz

jakąś tam listę opcji, na które można głosować. A druga funkcja to jest tak naprawdę zagłosowanie (możemy sprawić, że przyjęty będzie tylko głos osoby uprawnionej). Ta funkcja rejestrująca głos sprawdzi też od razu, czy ponad połowa już zagłosowała, albo czy któraś z opcji ma już większość, albo czy wynik może się jeszcze zmienić. No i jeśli dojdzie do wniosku, że głosowanie się rozstrzygnęło, to wynik będzie zapisany w blockchainie i tym samym głosowanie ulegnie zakończeniu. Przeprowadzone oczywiście w sposób totalnie jawny.

Czy jesteśmy w stanie zrobić coś, żeby to było jakkolwiek tajne, czyli na przykład takie Allegro? Czyli ślepą aukcję, gdzie nie wiemy ile kto deklaruje zapłaty, ale wiemy, że zwycięzca zapłaci tyle, ile zadeklarował drugi w kolejności?. Normalnie się nie da, bo każdy mógłby spojrzeć na jawne zmiany w blockchainie, ale w ślepej aukcji to to ma działać inaczej. Każdy ma rzucić swoją własną tajną cenę. Potrzebujemy jakiegoś mechanizmu, który rozdzieli proces na dwa etapy.

Pierwszy etap to przyjmowanie zaszyfrowanych deklaracji czy głosów, no i zapewne pieniędzy, żeby od razu była gwarancja zapłaty. Drugi etap to przyjmowanie kluczy do odszyfrowania ofert, by smart contract mógł wszystkie odszyfrować, porównać no i żeby od zwycięzcy zainkasował oczekiwaną wartość, a wszystkim pozostałym odesłał ich niewykorzystane środki (przegrali więc nie muszą nic płacić).

To trzy przykładowe proste programy. W dokumentacji Ethereum oraz Solidity można znaleźć więcej różnych takich pomysłów na to, co dałoby się zrobić na blockchainie.

Niebezpieczeństwa smart kontraktów

Niebezpieczeństwa? Są bardzo liczne. Maszyna wirtualna Ethereum i język Solidity są niepodobne do czegokolwiek innego i ograniczenia, które tam występują, są również specyficzne, inne od tych, z którymi mamy do czynienia w „normalnym” programowaniu. Trudno wykryć przypadki brzegowe, czyli na przykład wyczerpanie się paliwa. Albo precyzyjnie zaplanować, co się stanie, jeśli w trakcie wykonania kodu przekroczymy maksymalną głębokość stosu (maksymalnie 1024 zagnieżdżone wywołania funkcji).

Prosty przykład: popatrzmy sobie na skarbonkę, ale tym razem zrobimy taką skarbonkę dla paczki znajomych. Każdy przelewa pieniądze (ile może) a potem raz na miesiąc wywoływana jest funkcja, która sprawdza, ile się uzbierało, kto wpłacił w zeszłym miesiącu - i rozsyła te środki dzielone po równo każdemu kto wpłacił. Nie od razu widać niebezpieczeństwo z tym związane. Tymczasem mamy tutaj pętlę, w której atakujący kontroluje liczbę wykonań - poprzez liczbę wpłat. Atakujący może wpłacić do skarbonki milion malutkich przelewików, aby wartość pieniężna wzrosła niewiele, ale by na koniec miesiąca ten kontrakt musiał rozesłać milion przelewów. Nie ma szans, żeby starczyło mu paliwa na opłacenie miliona przelewów.

Zrobienie dużej liczby zasileń takiej skarbonki sprawi, że skarbonka nigdy już nie będzie w stanie się opróżnić i wszystkie środki przepadną na zawsze. I jeśli nie zauważymy tego

niebezpieczeństwa przed wrzuceniem programu do blockchaina, przed pierwszymi wpłatami - to przygoda z blockchainową skarbonką może zakończyć się dla kogoś katastrofą finansową.

Istnieją różne automatyczne mechanizmy, które sprawdzają najpopularniejsze podatności i błędy w kodzie smart kontraktów. Ich twórcy powinni szczególnie unikać ryzykownych konstrukcji i zapobiegać potencjalnie szkodliwym efektom ubocznym.

Pytanie od uczestnika

Czy jest jakiś sposób skomunikowania smart kontraktu ze światem zewnętrznym, żeby na przykład potrafił rozstrzygnąć zakład na podstawie publicznych danych?

Właśnie jest problem, bo nie bardzo. Jeśli chcielibyśmy zrobić jakiś smart kontrakt, który rozstrzygnie nam zakład o pogodę albo o wyścigi konne, albo o cokolwiek, czego nie ma w blockchainie, to jedyną możliwością jest to, żebyśmy mieli jakiegoś zaufanego dostawcę tych danych, który będzie je wstawiał do blockchaina. Nie ma innego źródła informacji. Jedyne, co możemy domniemywać, to czy minęła już jakaś data (tylko przy założeniu, że w sieci Ethereum nie ma w danej chwili jakichś grubych wałków związanych z datą). Znamy numer bloku - wiemy, że on rośnie z grubsza stałym tempie, więc i na nim się można też oprzeć gdy chcemy wiedzieć, co kiedy jest, ale to no problem jest taki, że nie ma po prostu takiej informacji w blockchainie, więc smart kontrakt nie będzie w stanie takiej informacji sam siebie pozyskać znikąd.

[01:14:53] Zakończenie

Więc tak to. Minęła godzina piętnaście, temat tokenów spada do następnego odcinka. Oglądajcie następny odcinek, bo będzie. Bardzo wszystkim dziękuję za udział, nadal mamy 64 osoby na żywo, więc bardzo fajnie.

Aha! Oczywiście dzwoneczki, kciuki w górę, i tak dalej - tradycyjnie mam no to wszystko wyrąbane. Nie musicie klikać niczego. I tak będę pisał na social mediach, że jest nowy odcinek, i będę wysyłał informację w newsletterze. A jeśli na czymś mi zależy, to na subskrypcjach newslettera. Jeśli jeszcze ktoś nie jest zapisany to ja bardzo proszę żeby się zapisać. Tam jest bardzo mało spamu to znaczy inaczej - spamu nie ma nigdy, jest jeden mail na cztery tygodnie albo rzadziej. Chyba, że gawęda - wtedy są dwa w jednym tygodniu. To jest kanał komunikacji, który mnie uniezależnia od jakichkolwiek Googli czy Facebooków, więc ja wolę subskrybentów od followersów czy obserwujących.

Bardzo dziękuję i do usłyszenia w następnym odcinku. Cześć.