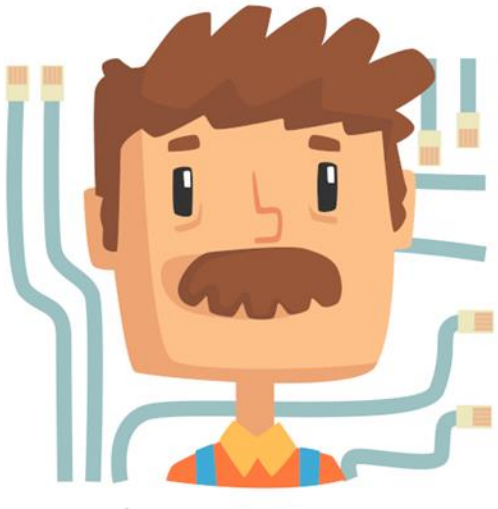
Gawęda o kryptowalutach część druga



Wyniki ankiety

Informatyk Zakładowy
@InfZakladowy

Dzisiaj rano BTC jest po 39800 dolarów. Jutro (piątek) o 20:00 zaczynamy gawędę o kryptowalutach. W momencie rozpoczęcia BTC będzie po:

mniej niż $39000 💰	**52,7%**
między $39000 a $41000 💸	24,6%
powyżej $41000 🤑	22,7%

370 głosów · Wyniki końcowe

8:47 AM · 20 maj 2021 · Twitter Web App

Gawędę sponsoruje słowo „uproszczenie”

Nie kupuj kryptowalut mając tylko wiedzę z tej prelekcji. Ucz się i ćwicz na małych kwotach! Używaj testnetu!

A → D, 10

D → E, 15

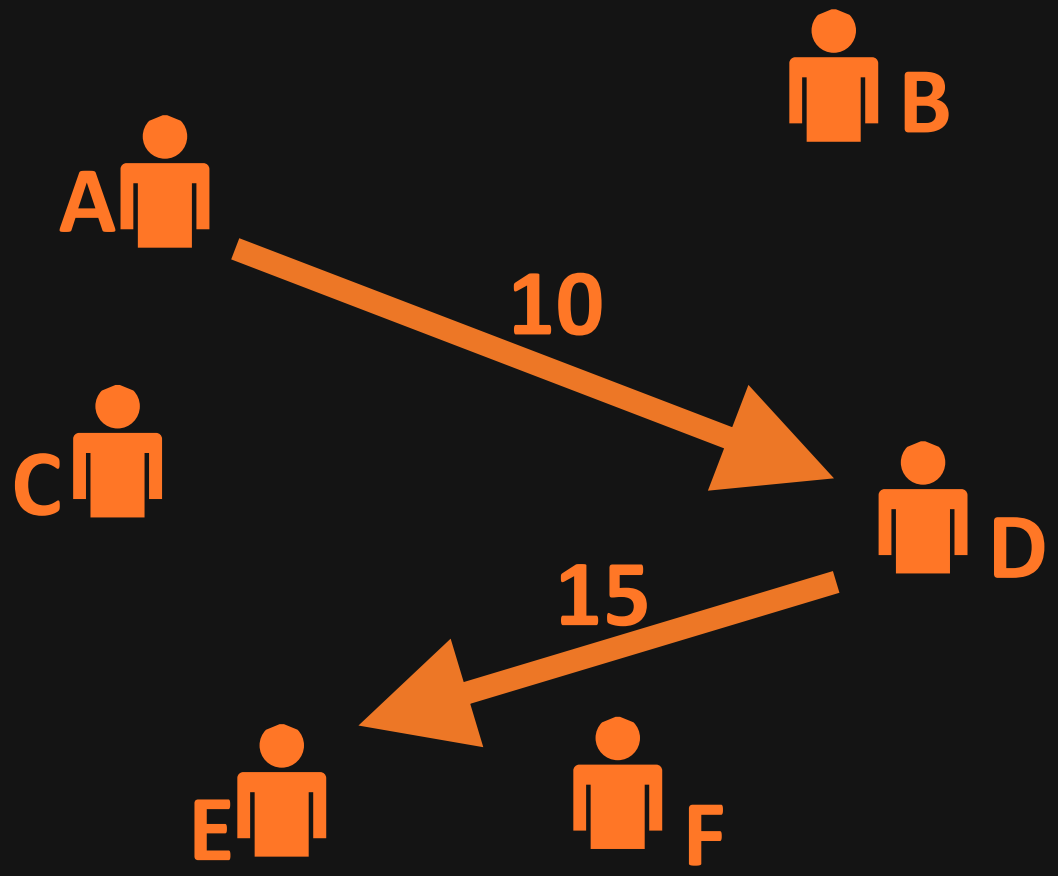
… → …, …

… → …, …

… → …, …

5d825022bd617f7a

Rejestr pożyczek



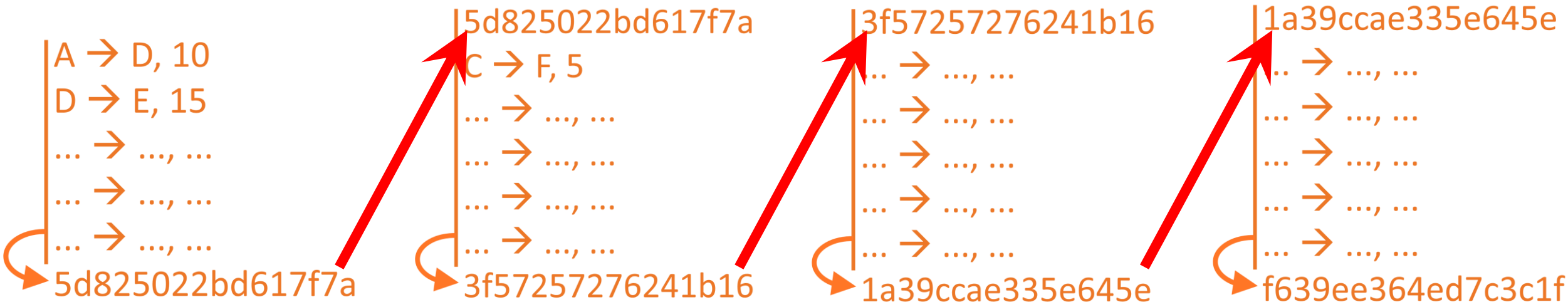
Grupa znajomych

W części pierwszej…

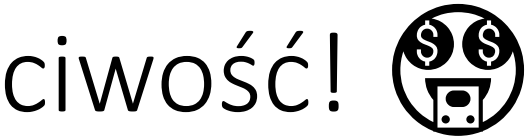
- Rozproszony rejestr
- Blockchain
- Sieci P2P

- Kryptografia klucza publicznego
- Proof of work i regulacja trudności

Co wiemy do tej pory



- każda transakcja w każdym bloku to przekazanie środków z jednego adresu na inny adres; nie rejestrujemy nigdzie „bieżącego stanu konta”
- jeśli prześledzimy całego blockchaina od samego początku to dowiemy się, które portfele (adresy) mają saldo większe od zera

Chciwość! 

💵 Aby cała sieć Bitcoina działała, górnicy muszą formować nowe bloki.

💸 Napędza ich chciwość, twórca nowego bloku dostaje prowizję za wykopanie i prowizję od przetworzonych transakcji.

💰 Efekt – mamy nowe bloki z transakcjami.

O czym dzisiaj będzie

- Forki kryptowalut
  (czyli np. co to jest Bitcoin Classic albo Bitcoin Cash)
- Co to jest Ethereum
- Co to są smart kontrakty
- Co to są tokeny
- Co to jest ICO
- Co to jest NFT
- Co to jest Proof of stake
- Chia i Proof of space
- Co to jest mixer (tumbler)
- Czy kryptowaluty dają anonimowość?
- Shitcoiny

slido.com
#76766

Forkowanie kryptowalut

slido.com
#76766

Upgrade (aktualizacja) protokołu

Hipotetyczny problem: bieżąca wersja oprogramowania do kopania kryptowalut zawiera a) błędy, b) ograniczenia. Co robić? (przykład ograniczeń – limit wielkości bloku BTC wynoszący 1 megabajt)

- Soft fork – starsza wersja oprogramowania działa (koegzystencja)
- Hard fork – zmiany wymuszające aktualizację oprogramowania z powodu modyfikacji protokołu

Kolejny problem – jak głosować za zmianami?

Głosowanie za/przeciw zmianom protokołu

Rozwiązanie: bardzo łatwe

1. Osoby odpowiedzialne za rozwój protokołu opisują proponowaną zmianę i podają termin głosowania
2. Właściciele kopalni rozgłaszają (lub nie) gotowość wprowadzenia tej zmiany poprzez włączenie (lub nie) odpowiedniej flagi (przełącznika) w oprogramowaniu do kopania
3. Jeśli na zakończenie głosowania operatorzy znacznej większości mocy obliczeniowej zgłaszają gotowość, zmiana wchodzi w życie

A co, jeśli ktoś się nie podporządkuje?

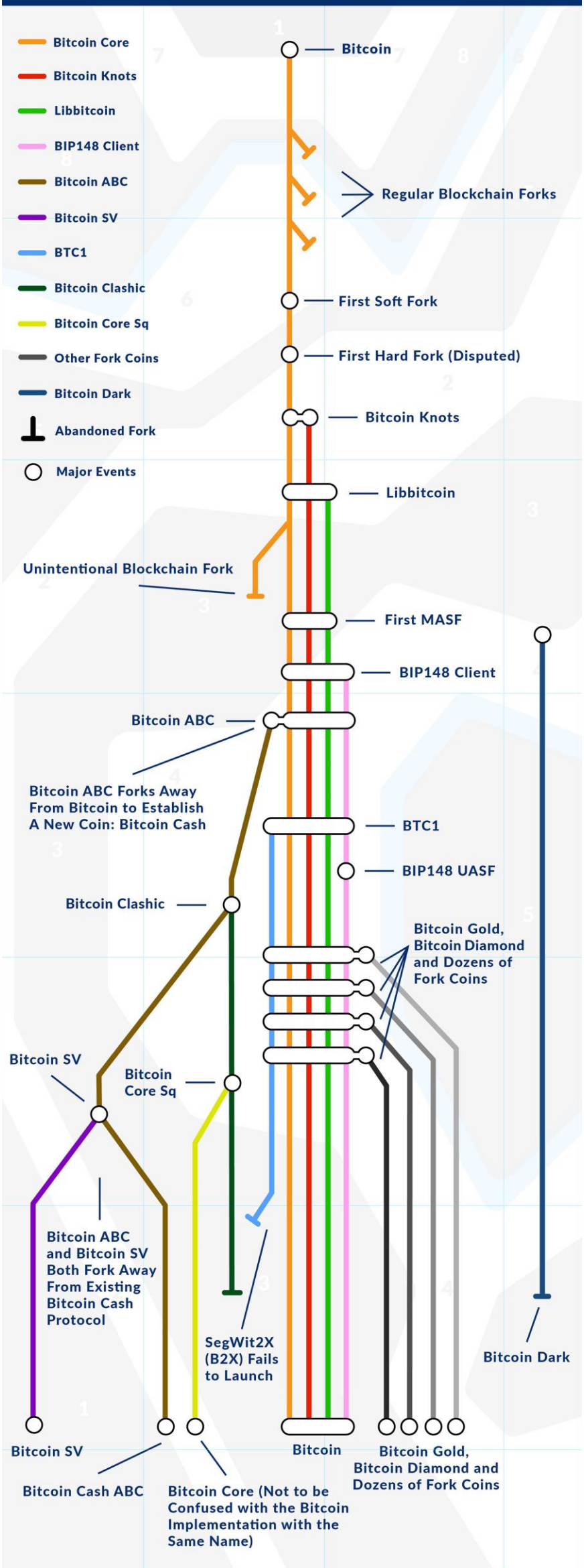
Secesja! Schizma! Rozwidlenie! Fork!

Możliwe są oba warianty – przyjęcie zmiany odrzuconej przez ogół lub obstawanie przy starej, porzuconej przez większość wersji protokołu.

W obu przypadkach bieżący blockchain pozostaje *ważny*, tzn. zachowany zostaje stan posiadania z ostatniego wspólnego bloku

(zakładanie nowej kryptowaluty na istniejącym protokole to co innego)

A MAP OF BITCOIN FORKS



https://bitcoinmagazine.com

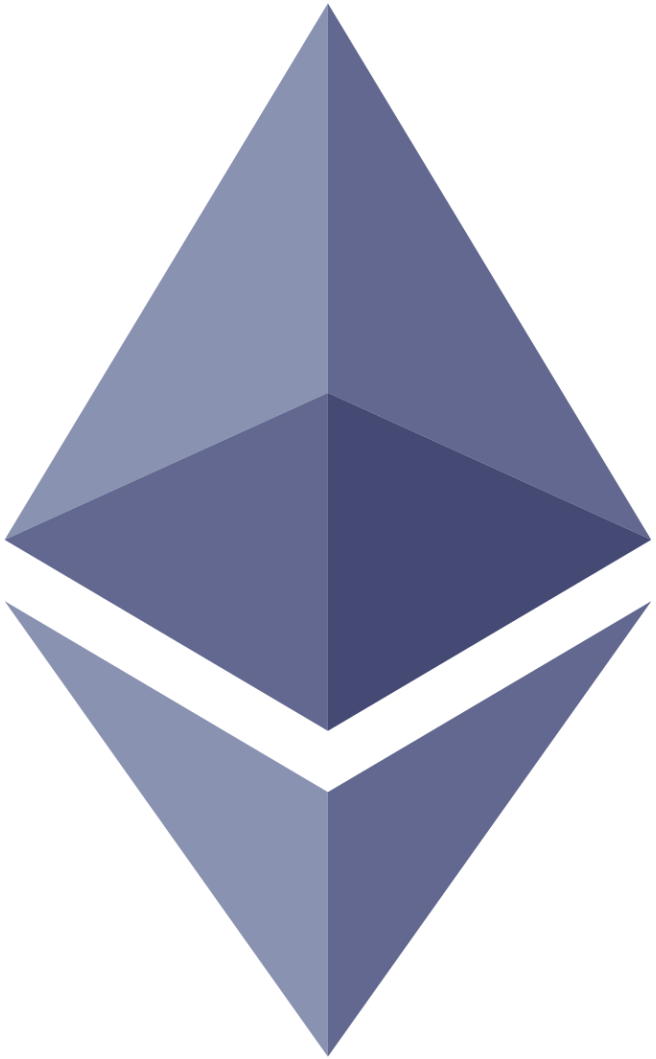
Źródło: https://www.visualcapitalist.com/major-bitcoin-forks-subway-map/

Name	Last Price	24h Change	Market Cap
BTC Bitcoin	$57,650.01	+0.75%	$1,078,390.59M
BCH Bitcoin Cash	$1,436.09	+9.19%	$26,902.41M
WBTC Wrapped Bitcoin	$57,730.76	+0.00%	$9,916.12M
BTG Bitcoin Gold	$120.16	+0.38%	$2,104.47M
BCD Bitcoin Diamond	$7.34	-9.92%	$1,368.86M
BCHA Bitcoin Cash ABC	$35.46	-1.35%	$658.60M

Źródło: binance.com, losowy moment w maju 2021

Ethereum

(i trochę o kontraktach)



slido.com
#76766

Najpierw będzie dygresja

Druga dygresja

- Czy w sieci Bitcoin da się zapisać informacje?
- Załóżmy, że robimy z jednego adresu serię transakcji na następujące kwoty (jednostka to satoshi, nie bitcoin):
  - 00100116
  - 00200111
  - 00300109
  - 00400101
  - 00500107

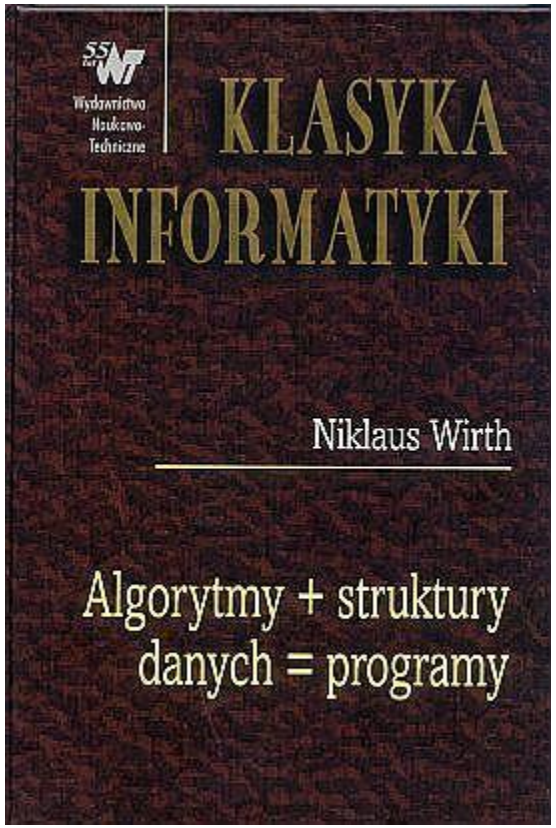
Druga dygresja

- Czy w sieci Bitcoin da się zapisać informacje?
- Załóżmy, że robimy z jednego adresu serię transakcji na następujące kwoty (jednostka to satoshi, nie bitcoin):
  - 00100116 **t**
  - 00200111 **o**
  - 00300109 **m**
  - 00400101 **e**
  - 00500107 **k**

numer bajtu

kod ASCII

Trzecia dygresja



- Wiemy, że umiemy zapisać w rozproszonym rejestrze dowolne dane, ale sieć Bitcoin nie umie z nimi niczego robić
- A gdyby tak istniała kryptowaluta potrafiąca robić transakcje automatyczne? Dowolne? Opisane algorytmem?

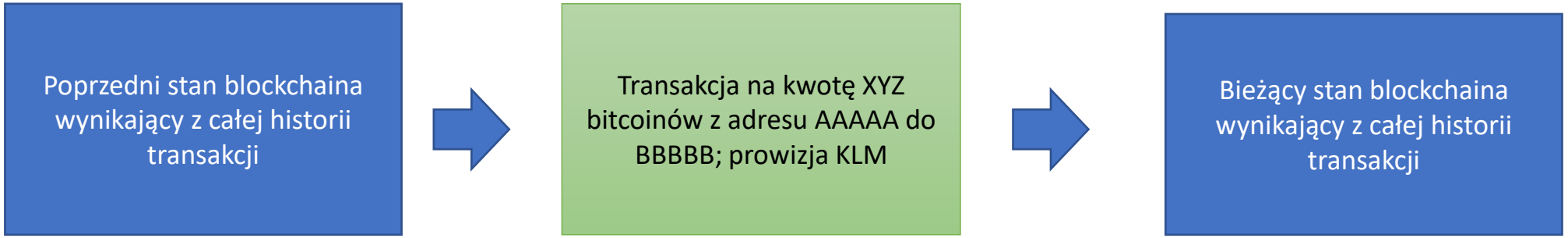
Ethereum

- Druga pod względem kapitalizacji TODO SPRAWDZIĆ i popularności WIADOMIX kryptowaluta
- Twórca miał ciekawszy pomysł – tym razem w użyciu jest nie tylko rozproszony rejestr, a rozproszony komputer (maszyna wirtualna)

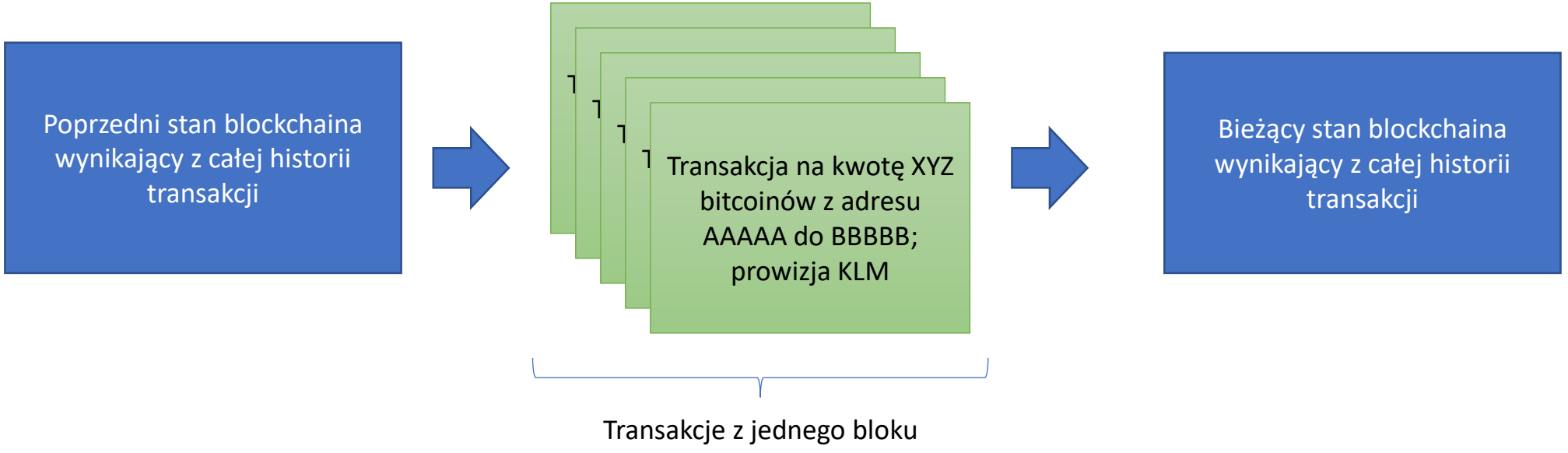
Vitaly Dmitriyevich Buterin

- Nadal mamy ten sam mechanizm konsensusu – prawdziwe jest to, co zapisano w rozproszonym rejestrze, w blokach podpisywanych przez kopiących (ale „transakcja” to teraz więcej, niż tylko przelew środków)

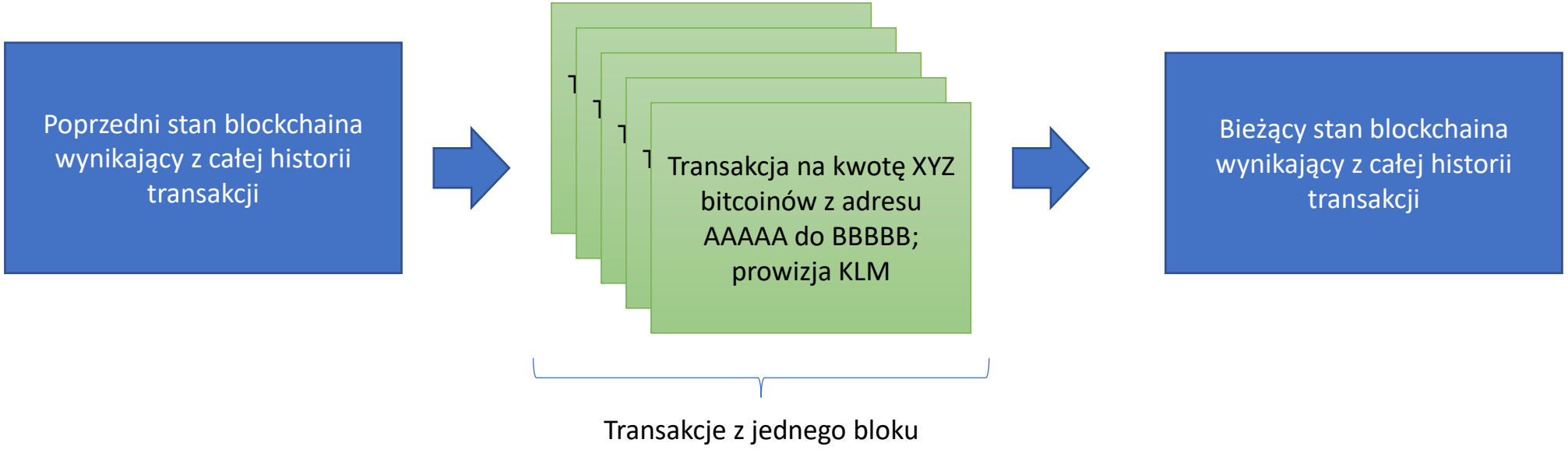
BTC


Poprzedni stan blockchaina wynikający z całej historii transakcji
Transakcja na kwotę XYZ bitcoinów z adresu AAAAA do BBBBB; prowizja KLM
Bieżący stan blockchaina wynikający z całej historii transakcji

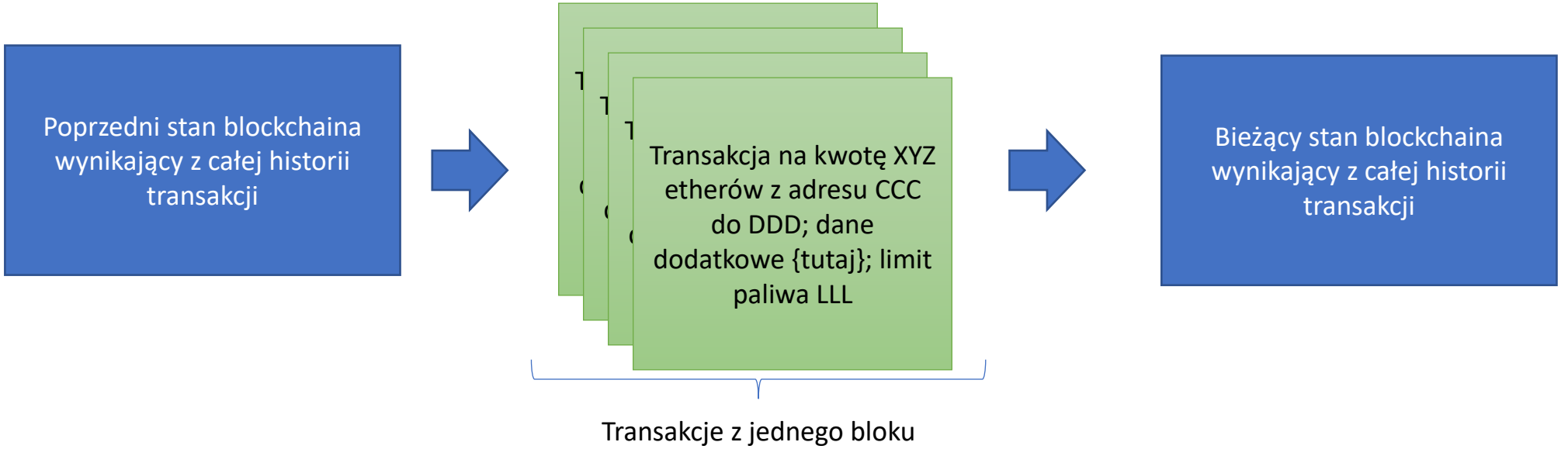
BTC


Poprzedni stan blockchaina wynikający z całej historii transakcji
Transakcja na kwotę XYZ bitcoinów z adresu AAAAA do BBBBB; prowizja KLM
Bieżący stan blockchaina wynikający z całej historii transakcji
Transakcje z jednego bloku

BTC


Poprzedni stan blockchaina wynikający z całej historii transakcji
Transakcja na kwotę XYZ bitcoinów z adresu AAAAA do BBBBB; prowizja KLM
Bieżący stan blockchaina wynikający z całej historii transakcji
Transakcje z jednego bloku

ETH


Poprzedni stan blockchaina wynikający z całej historii transakcji
Transakcja na kwotę XYZ etherów z adresu CCC do DDD; dane dodatkowe {tutaj}; limit paliwa LLL
Bieżący stan blockchaina wynikający z całej historii transakcji
Transakcje z jednego bloku

Blockchain ETH

- Ether kopie się na kartach graficznych a nie układach ASIC, ale nie wchodzimy w szczegóły (czy kiedykolwiek wchodzimy?…)

- Blockchain BTC ma jeden rodzaj adresów i umie pamiętać tylko saldo
- Blockchain ETH:
  - ma dwa rodzaje adresów („dla ludzi” i „dla programów”)
  - dla każdego adresu blockchain trzyma saldo
  - blockchain potrafi rejestrować także dane
  - dane mogą mieć typy (np. bool, int, string) oraz struktury (np. tablice, mapy)

Smart contract

- Żeby uruchomić program w rozproszonym komputerze, musimy go w osobnej transakcji rozesłać do całej sieci ETH
- Programy to **kontrakty**
- **Kontrakt** jest tworzony przez transakcję pokrywającą koszt zapisania go w blockchainie
- Każdy kontrakt dostaje unikalny **adres publiczny**

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

Uruchamianie funkcji

- Żeby wywołać funkcję, tworzymy transakcję adresowaną do smart kontraktu
- Transakcja wywołująca funkcję zawiera prowizję za uruchomienie programu, dostanie ją górnik
- Dane wejściowe i wyjściowe muszą przejść przez blockchain

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

Nakład pracy CPU = zużyte paliwo (gas)

	A	B	C	D	E	F	G	H	I	J
1		Approximations						Gas price		
2	Param	Compute (µs)	History (bytes)	State (bytes)	Bandwidth	Bloom topic	Mem quad	Computed	**Actual**	
3	DUP	3						3	**3**	FASTESTSTEP
4	SWAP	3						3	**3**	FASTESTSTEP
5	PUSH	3						3	**3**	FASTESTSTEP
6										
7	ADD	3						3	**3**	FASTESTSTEP
8	MUL	5						5	**5**	FASTSTEP
9	SUB	3						3	**3**	FASTESTSTEP
10	DIV	5						5	**5**	FASTSTEP
11	SDIV	5						5	**5**	FASTSTEP
12	MOD	5						5	**5**	FASTSTEP
13	SMOD	5						5	**5**	FASTSTEP
14	ADDMOD	8						8	**8**	MIDSTEP
15	MULMOD	8						8	**8**	MIDSTEP
16	EXPBASE	10						10	**10**	SLOWSTEP
17	EXPBYTE	10						10	**10**	
18	SIGNEXTEND	5						5	**5**	FASTSTEP
19	LT	3						3	**3**	FASTESTSTEP
20	GT	3						3	**3**	FASTESTSTEP
21	SLT	3						3	**3**	FASTESTSTEP

Nakład pracy CPU = zużyte paliwo (gas)

- koszt wyrażony jest w jednostkach umownych, proporcjonalnych do kosztu CPU oraz składowania danych
- taka sama ścieżka wykonania funkcji oznacza (prawie) taki sam koszt wyrażony w paliwie

Operation	Gas	Description
ADD/SUB	3	Arithmetic operation
MUL/DIV	5	
ADDMOD/MULMOD	8	
AND/OR/XOR	3	Bitwise logic operation
LT/GT/SLT/SGT/EQ	3	Comparison operation
POP	2	Stack operation
PUSH/DUP/SWAP	3	
MLOAD/MSTORE	3	Memory operation
JUMP	8	Unconditional jump
JUMPI	10	Conditional jump
SLOAD	200	Storage operation
SSTORE	5,000/ 20,000	
BALANCE	400	Get balance of an account
CREATE	32,000	Create a new account using CREATE
CALL	25,000	Create a new account using CALL

Dygresja

Bitcoin

- 1 bitcoin = 10^9 satoshi
- prowizje za przelew
- blok co 10 minut
- max 21 milionów bitcoinów
- 7-8 transakcji na sekundę
- kopanie na ASIC-ach

Ethereum

- 1 ether = 10^{18} wei (10^9 gwei)
- prowizje za zużyte paliwo
- blok co 12 sekund
- brak limitu (18 mln co roku)
- 15-20 transakcji na sekundę
- kopanie na GPU

Przeliczenie kosztu paliwa na ETH (oraz $$$)

- w sieci BTC transakcja składa się z głównej kwoty przelewu oraz zadeklarowanej prowizji dla górnika
- w sieci ETH jest podobnie, ale prowizja zależy od kosztu wykonania (nie znamy go z góry), zlecający deklaruje maksimum

- rozwiązanie: zlecający transakcję deklaruje, ile *gwei* zapłaci za jeden *gas* czyli (w uproszczeniu) ile ETH płaci za sekundę CPU; dołącza do transakcji swoją maksymalną kwotę prowizji
- górnicy przetwarzając transakcję pobierają prowizję za rzeczywisty koszt wykonania, reszta kwoty wraca do zlecającego

Przykład (niekoniecznie realistyczny):

Rok 20xx

- 1 ETH = $1000
- Koszt wykonania = 10000 paliwa
- Hojność = 10 gwei / 1 paliwo
- Wartość prowizji =
  100000 gwei * 1000 USD/ETH =
  0.0001 ETH * 1000 USD/ETH =
  0.10 USD

Rok 20xy

- 1 ETH = $10000
- Koszt wykonania = 10000 paliwa
- Hojność = 1 gwei / 1 paliwo
- Wartość prowizji =
  10000 gwei * 10000 USD/ETH =
  0.00001 ETH * 10000 USD/ETH =
  0.10 USD

Obliczenia muszą być deterministyczne

- Consensus – wszystkie węzły sieci ETH weryfikujące poprawność bloku muszą być w stanie wykonać dokładnie identyczną sekwencję obliczeń
- Efekt – brak wartości losowych w języku Solidity: *The problem with randomness in Ethereum is that Ethereum is a deterministic Turing machine, with no inherent randomness involved.*
- Ale uwaga - *t*ransakcja zna datę wykonania oraz numer bloku
- Oczywiście nadal każdy górnik dowolnie dobiera transakcje z puli oraz układa je w dowolnej kolejności

Dodatkowe informacje

- Kontrakty mogą wywoływać inne kontrakty (ale jest to kosztowne)
- Sieć ETH określa limity paliwa zużywanego w jednym bloku (15 mln)

- Koszt składowania danych bywa gigantyczny – do $600 za **KILOBAJT!**
- Czyli ponad DWA ZŁOTE ZA BAJT!

Dodatkowe informacje [ŁAMIĄCA WIADOMOŚĆ]

- Kontrakty mogą wywoływać inne kontrakty (ale jest to kosztowne)
- Sieć ETH określa limity paliwa zużywanego w jednym bloku (15 mln)

- Koszt składowania danych bywa gigantyczny – do $600 za ~~**KILOBAJT!**~~ **10 bajtów**
- ~~Czyli ponad DWA ZŁOTE ZA BAJT!~~

- Poprzedni slajd pochodził z 11 maja, wtedy ceny skoczyły x10. Gdy przedwczoraj kursy kryptowalut poleciały na twarz, najtwardsi zawodnicy robili transakcje po cenach x1000

Małe podsumowanie

- W Ethereum mamy dwa rodzaje adresów – posiadane przez ludzi (z kluczem publicznym i prywatnym) oraz przypisane do kontraktów
- Kontrakt to program zapisany w blockchainie
- Blockchain Ethereum rośnie o nowy blok co ~13 sekund, w środku ~200 transakcji, oprócz przenoszenia środków pieniężnych można w nim zapisywać dane i wywoływać funkcje zdefiniowane w kontraktach
- Prowizje w sieci Ethereum zależą od pracochłonności realizowanych operacji i decyzji zlecającego, ich wartość może się zmieniać niezależnie od wyceny ETH

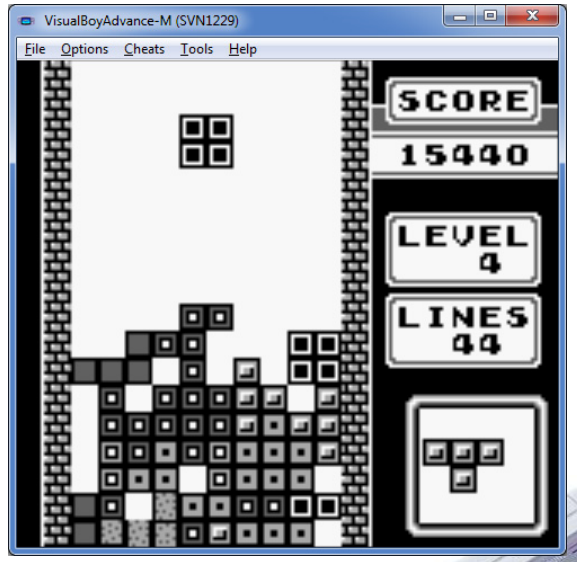
Smart kontrakty

dApp, decentralized app

slido.com
#76766

Czym są kontrakty w sieci Ethereum?

- Kontrakty to programy napisane dla maszyny wirtualnej EVM
- EVM jest maszyną stosową, operuje na strukturach danych charakterystycznych dla blockchaina ETH
- Język programowania dla Ethereum to (głównie) Solidity
- Kompilatory, dekompilatory, debuggery, testnet



Czym są kontrakty w sieci Ethereum?

Przykłady za chwilę

- Kontrakt to taki automat, który będzie coś robił tylko, gdy będzie adresatem transakcji (nadanej przez człowieka) lub wiadomości (praktycznie to samo, ale nadanej przez inny kontrakt)
- Kontrakt ma dostęp do historii blockchaina i prawie niczego innego (żadnych plików, sieci, urządzeń – tylko blockchain i dane z transakcji)

- Kontrakt jest umieszczany w blockchainie raz na zawsze, nie da się zmienić jego działania (możliwe jest samozniszczenie, z bonusem!)
- Ale można zrobić kontrakt z dynamicznym przekierowaniem

Co może robić smart kontrakt?

czyli: jakie operacje można umieścić w smart kontrakcie

- Obsługa parametrów wywołania (treść danych, wartość ETH)
- Odczyt stanu blockchaina (m.in. własne zmienne)
- Wysyłanie środków
- Wywoływanie innych kontraktów
- Zapisywanie danych do blockchaina
- pętle, rozgałęzienia, skoki, itd.

Przykład nr 1 - skarbonka

- Smart kontrakt ma jedną funkcję, która – po odpaleniu – sprawdza, czy zasoby „pieniężne” (liczba etherów w portfelu) przekracza 10 ETH. Jeśli tak, przekazuje wszystkie środki pod adres XYZ

Przykład nr 2 – maszynka do głosowania

- Smart kontrakt ma dwie funkcje
- Pierwsza funkcja tworzy nowe głosowanie, parametry:
  - numer głosowania
  - opis głosowania
  - lista adresów uprawnionych do oddania głosu
  - lista opcji na które można głosować
- Druga funkcja zbiera głosy, przyjmując numer oraz decyzję; przy każdym wywołaniu sprawdza czy głos jest legalny i czy głosowanie się rozstrzygnęło, jeśli tak to emituje zdarzenie

Przykład nr 3 – ślepa aukcja

- Maszynka taka, jak Allegro – przyjmuje oferty od licytujących i wyznacza zwycięzcę pobierając od niego cenę wskazaną przez drugiego oferenta
- Nie do zrobienia wprost, z powodu jawności blockchaina i możliwości manipulacji przez kopiącego
- Wymaga dwóch etapów – przyjmowania zaszyfrowanych ofert w okresie licytacji i przyjmowania kluczy do odszyfrowania oferty

Smart kontrakty - niebezpieczeństwa

- Solidity i EVM są nieintuicyjne i niepodobne do zwykłego programowania
- Trudno wykryć przypadki brzegowe
- Jeszcze trudniej przewidzieć ich konsekwencje (o DAO innym razem)

- Przykład: skarbonka dla paczki znajomych – każdy wpłaca ile może a skarbonka co miesiąc dzieli między wszystkich swą zawartość

Smart kontrakty - niebezpieczeństwa

- Solidity i EVM są nieintuicyjne i niepodobne do zwykłego programowania
- Trudno wykryć przypadki brzegowe
- Jeszcze trudniej przewidzieć ich konsekwencje (o DAO innym razem)

- Przykład: skarbonka dla paczki znajomych – każdy wpłaca ile może a skarbonka co miesiąc dzieli między wszystkich swą zawartość
- pętla kontrolowana przez atakującego, Out Of Gas Exception

Smart kontrakty - bezpieczeństwo

https://github.com/crytic/slither

Detectors

Num	Detector	What it Detects	Impact	Confidence
1	`abiencoderv2-array`	Storage abiencoderv2 array	High	High
2	`array-by-reference`	Modifying storage array by value	High	High
3	`incorrect-shift`	The order of parameters in a shift instruction is incorrect.	High	High
4	`multiple-constructors`	Multiple constructor schemes	High	High
5	`name-reused`	Contract's name reused	High	High
6	`public-mappings-nested`	Public mappings with nested variables	High	High
7	`rtlo`	Right-To-Left-Override control character is used	High	High
8	`shadowing-state`	State variables shadowing	High	High
9	`suicidal`	Functions allowing anyone to destruct the contract	High	High
10	`uninitialized-state`	Uninitialized state variables	High	High
11	`uninitialized-storage`	Uninitialized storage variables	High	High
12	`unprotected-upgrade`	Unprotected upgradeable contract	High	High

https://medium.com/@knownsec404team/ethereum-smart-contract-audit-checklist-ba9d1159b901

[1] Arithmetic overflow

When calling addition, subtraction, multiplication and division, you should use the safeMath library instead, otherwise it will easily lead to calculation overflow, resulting in inevitable loss.

```
pragma solidity ^0.4.18;

contract Token {

  mapping(address => uint) balances;
  uint public totalSupply;

  function Token(uint _initialSupply) {
```

Czy blockchain gwarantuje niezmienność danych do końca świata?

Z czego wynikały forki Ethereum?

Co się stało, gdy górnicy weszli w zmowę?

Oglądajcie następny odcinek gawędy!

INFORMATYK ZAKŁADOWY.PL