

BLOCKCHAIN

Dla programistów

Michał Rudnicki • 2017
michal.rudnicki@epsi.pl

Bitcoin - pierwszy blockchain

Jest rok 1992.

Zadanie: wytłumaczyć
internet swojej matce.



Do czego służy blockchain?

Internet

Sieć komputerowa do
kopiowania danych



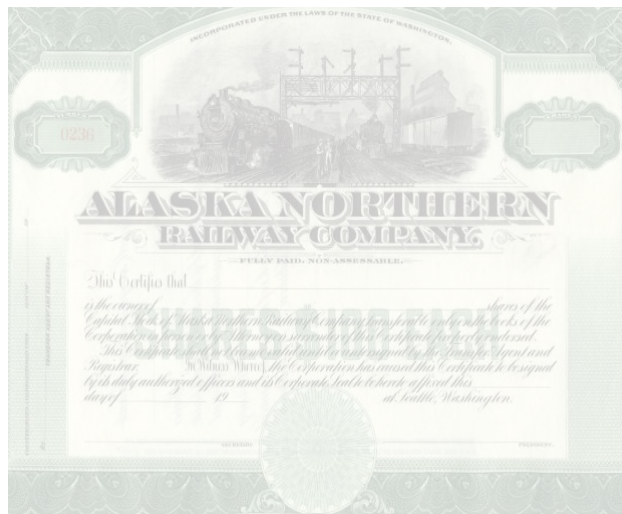
Internet

Sieć komputerowa do
kopiowania danych

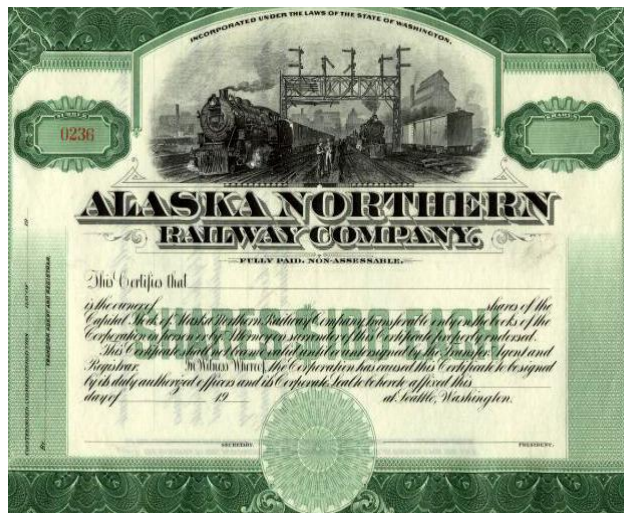
Blockchain

Sieć komputerowa do
przenoszenia danych

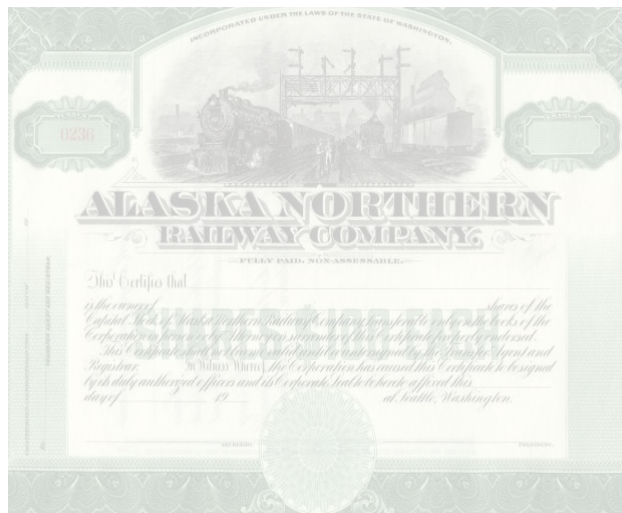




Pieniądze



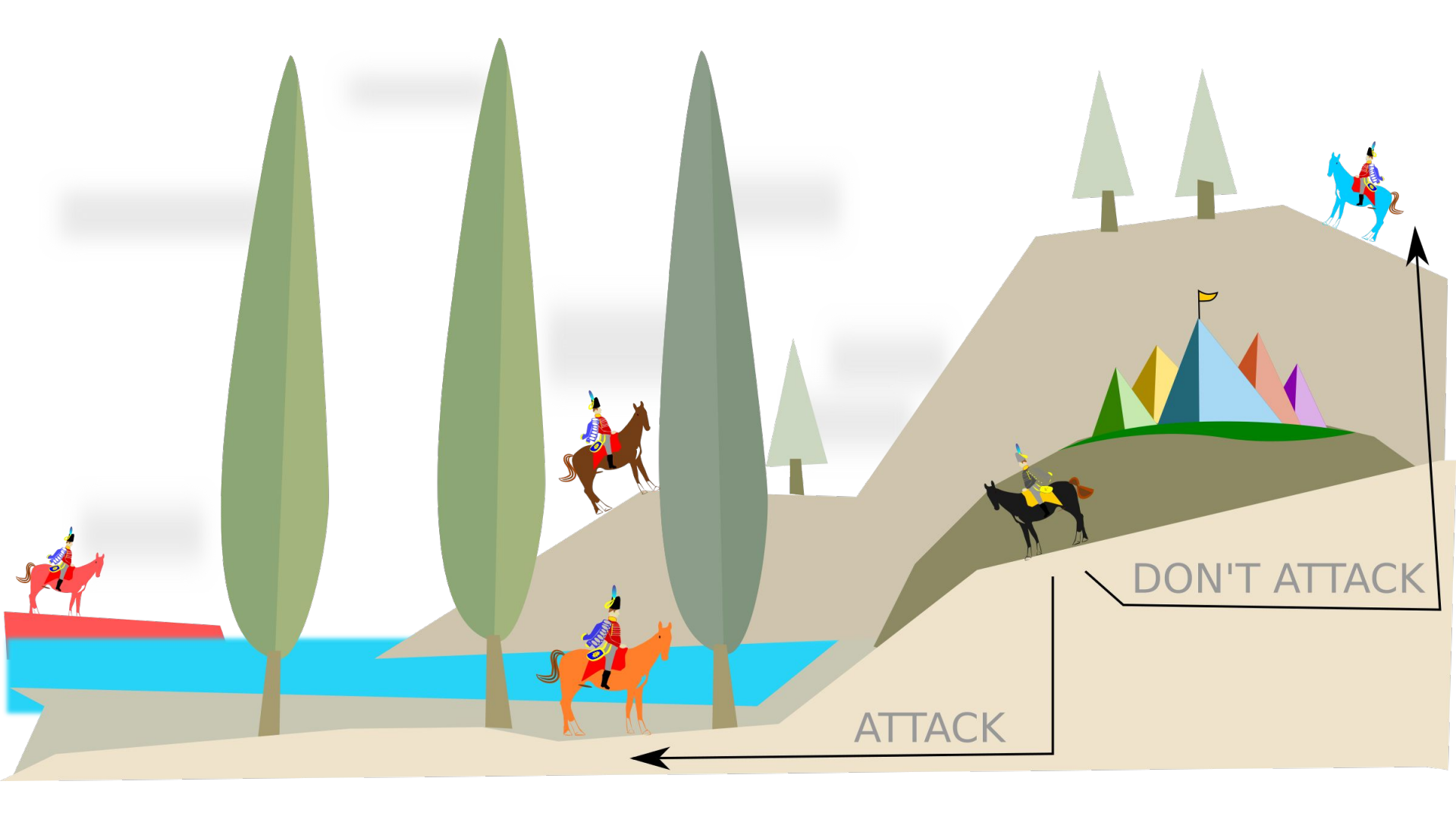
Akcje, udziały



Tytuły własności

Co to jest blockchain?

**Rozproszona baza danych
bez centralnego arbitra (mastera).**



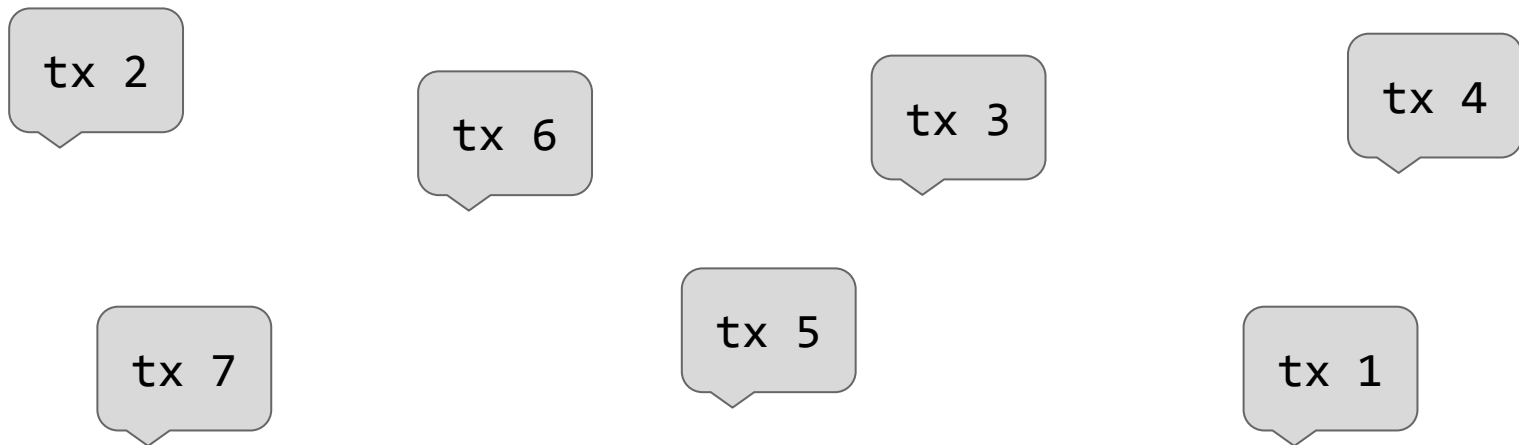
ATTACK

DON'T ATTACK

**Konsensus poprzez "losowanie"
tymczasowego lidera**

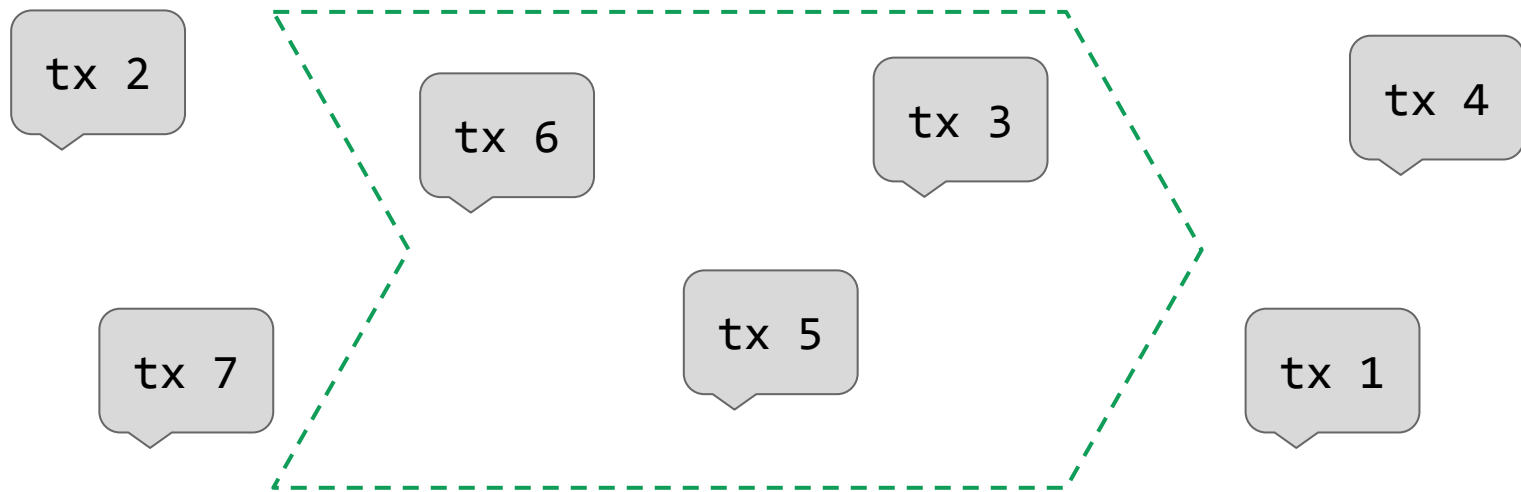
Konsensus poprzez "losowanie" lidera

1. każdy uczestnik sieci może ogłosić dowolną transakcję



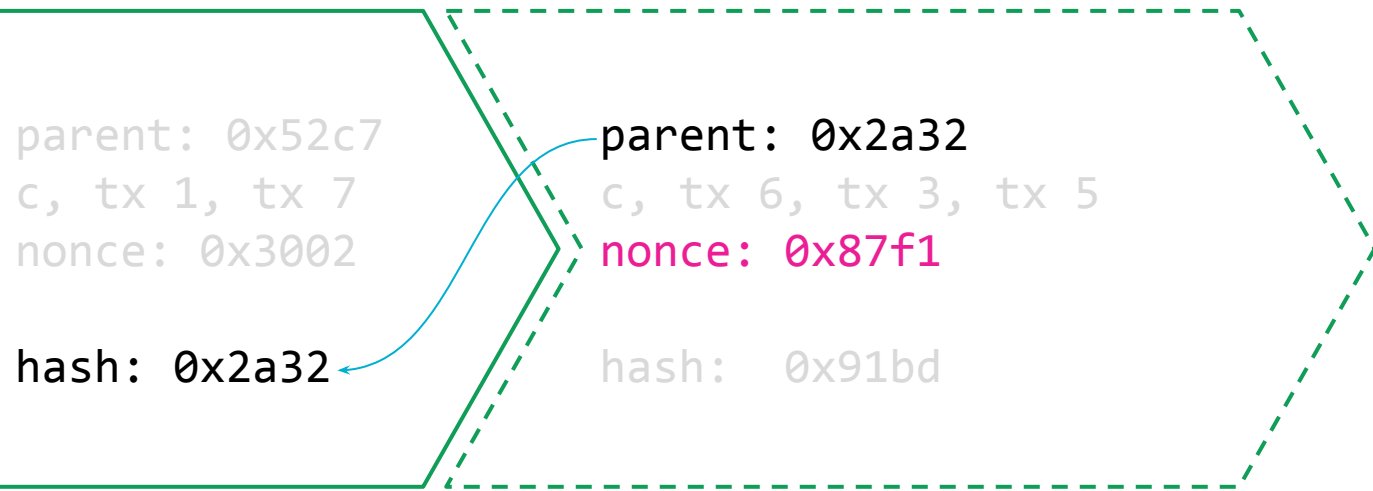
Konsensus poprzez "losowanie" lidera

2. co 10 minut "losowany" jest arbiter, który ma prawo ogłosić które transakcje uznaje się za ważne - tworzy **blok**



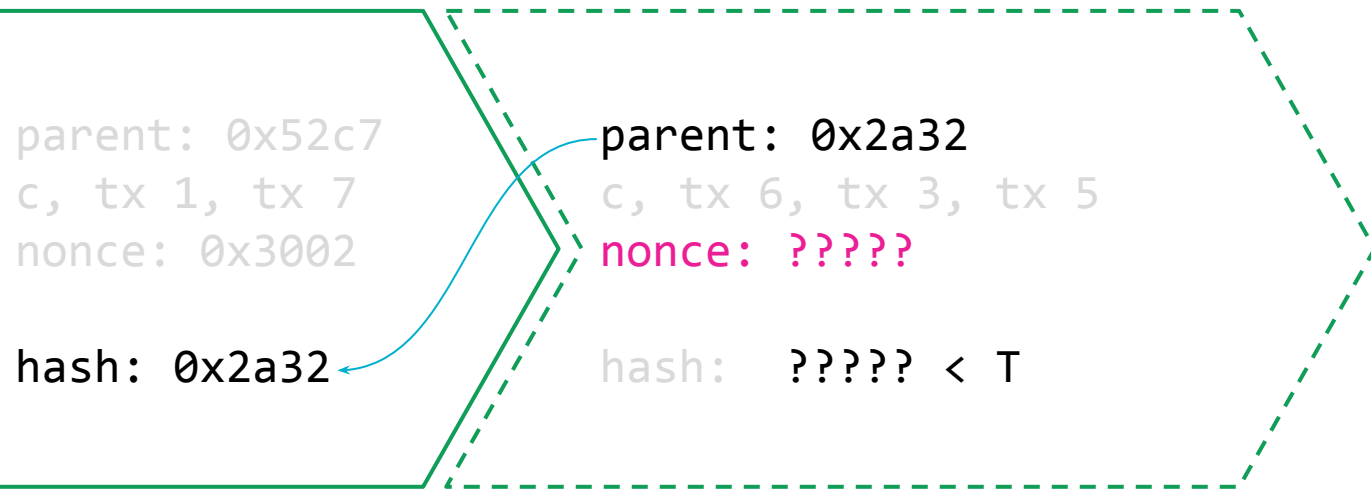
Konsensus poprzez "losowanie" lidera

3. blok ma referencję do poprzedniego bloku, listę transakcji, **pole losu** oraz hasz wszystkiego powyżej



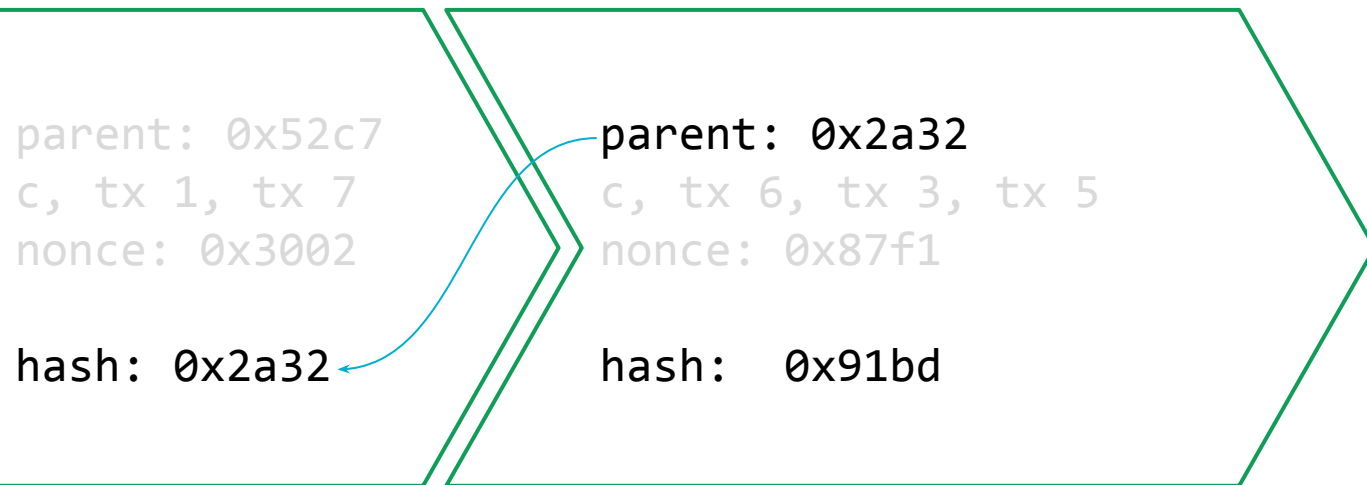
Konsensus poprzez "losowanie" lidera

4. wartość **poła losu** musi być taka, by hasz bloku $< T$
(trudne obliczeniowo)



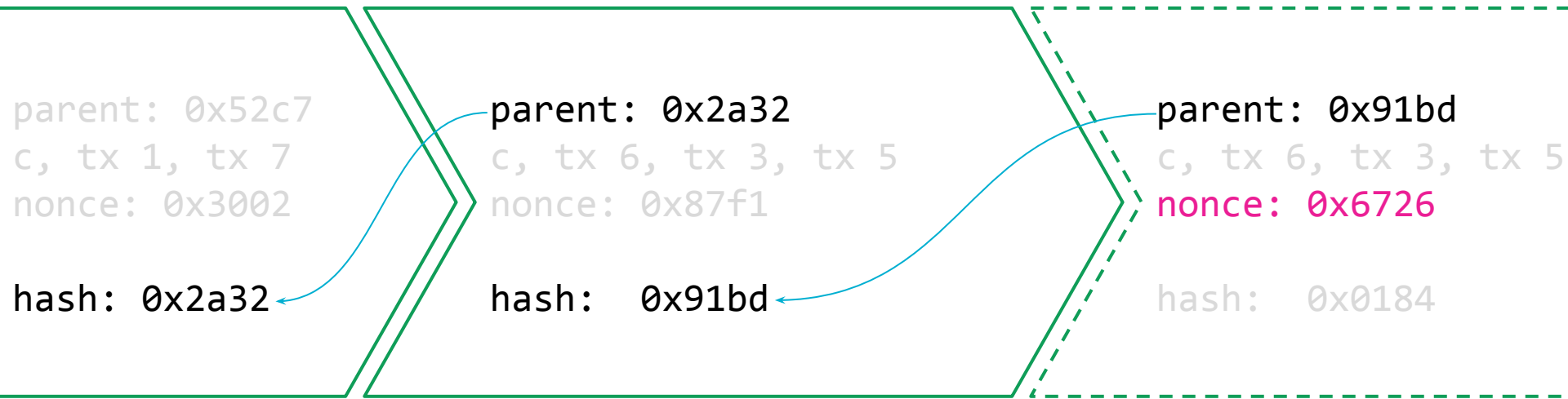
Konsensus poprzez "losowanie" lidera

5. uczestnicy sieci weryfikują taki blok
(łatwe obliczeniowo)



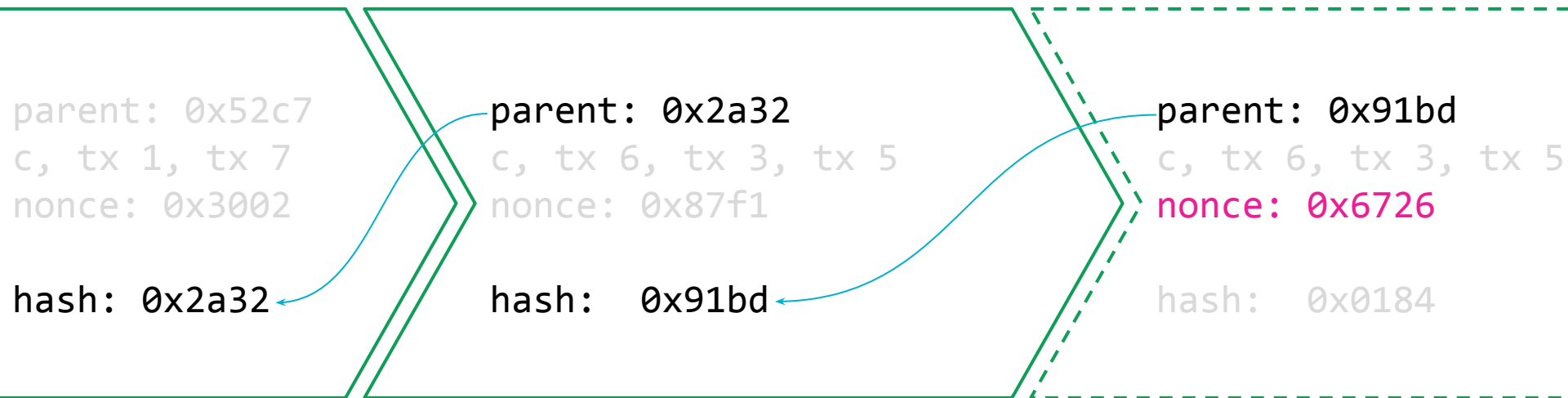
Konsensus poprzez "losowanie" lidera

6. rozpoczyna się poszukiwanie kolejnego bloku



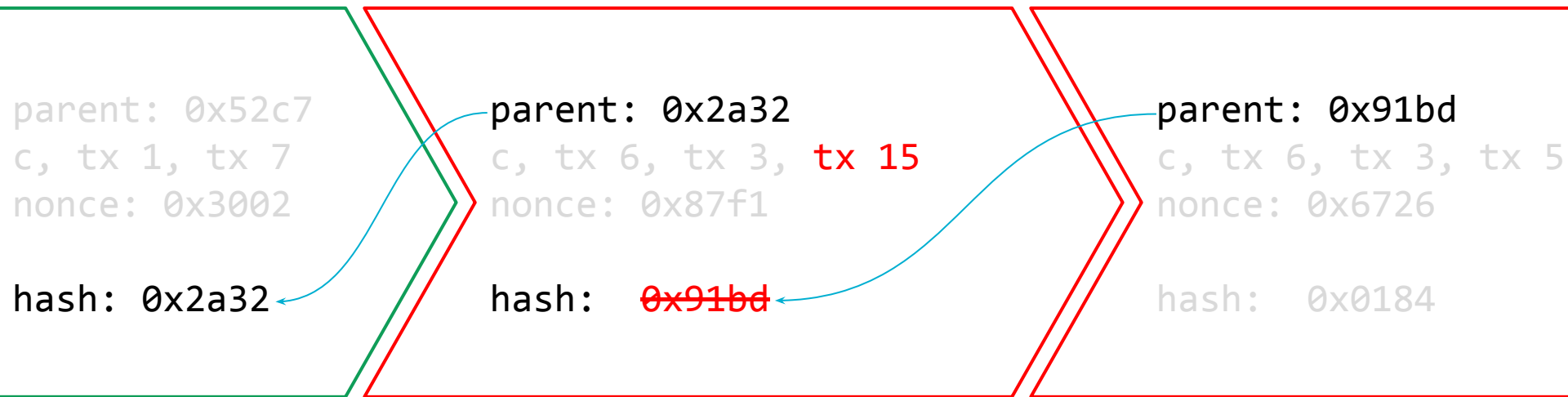
Konsensus poprzez "losowanie" lidera

sieć (wszyscy uczestnicy) rekalkuluje wielkość T co X bloków
by uśredniony czas znalezienia bloku wynosił 10 minut (Bitcoin)



Konsensus poprzez "losowanie" lidera

próba zafałszowania historii spowoduje niezgodność haszu
oraz unieważnienie następujących bloków



parent: 0x2a32

c, tx 6, tx 3, tx 15

nonce: 0x87f1

hash: 0x460a

parent: 0x460a

c, tx 6, tx 3, tx 5

nonce: 0x6726

hash: 0x63b3

parent: 0x52c7

c, tx 1, tx 7

nonce: 0x3002

hash: 0x2a32

parent: 0x2a32

c, tx 6, tx 3, tx 5

nonce: 0x87f1

hash: 0x91bd

parent: 0x91bd

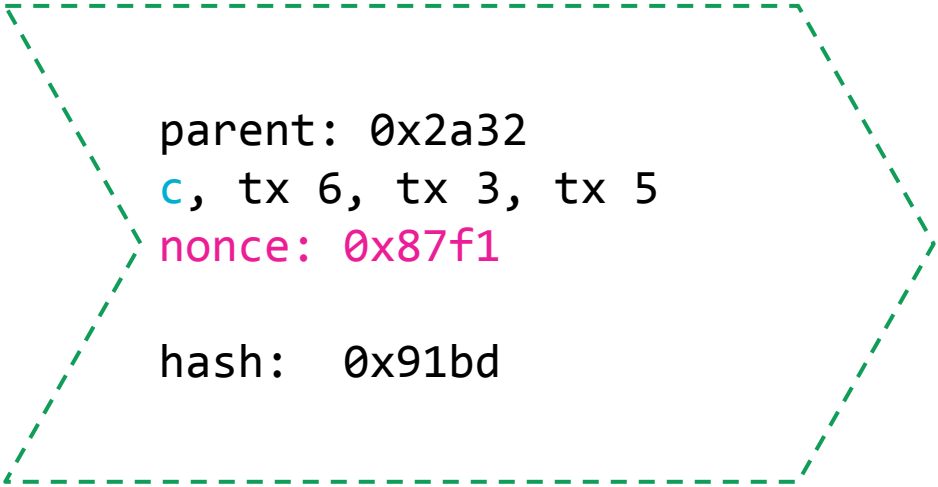
c, tx 6, tx 3, tx 5

nonce: 0x6726

hash: 0x0184

Konsensus poprzez "losowanie" lidera

7. znalazca odpowiedniej wartosci
pola losu otrzymuje 50 BTC* "z powietrza" (transakcja c)



```
parent: 0x2a32
c, tx 6, tx 3, tx 5
nonce: 0x87f1

hash: 0x91bd
```



Co siedzi w transakcji?

klucz => wartość

hasz klucza publicznego => skrypt

hasz klucza publicznego => 2 skrypty
(odblokowujący i blokujący)

Kryptografia Klucza Publicznego



Klucz prywatny

1. Tajny
2. Służy do cyfrowego podpisania wiadomości
“ja to ja”
3. Służy odszyfrowywaniu wiadomości zaszyfrowanej kluczem publicznym

Klucz publiczny

1. Jawny
2. Służy do weryfikowania podpisanej przeze mnie wiadomości
“czy on to on?”
3. Służy do szyfrowania wiadomości wysyłanej
do posiadacza klucza prywatnego

Jak wysłać bezpiecznego maila?

1. Podpisuję maila swoim kluczem prywatnym
2. Szyfruję maila twoim kluczem publicznym
3. Wysyłam
4. Odbiorca odszyfrowuje maila swoim kluczem prywatnym
(gwarancja poufności)
5. Odbiorca weryfikuje mój podpis cyfrowy moim kluczem publicznym
(gwarancja tożsamości)

Skrypt odblokowujący

“Z adresu XYZ, którego jestem właścicielem - oto podpis cyfrowy - prześlij ... BTC na adres ABC.”

Skrypt blokujący

“By umożliwić wysłanie z ABC, wymagaj przedstawienia ważnego podpisu cyfrowego dla tego adresu.”

Zagadka do rozwiązania przez właściciela
Kłucza ABC.

Odblokowywanie + blokowanie

Baza danych z kontrolą dostępu
na poziomie wierszy.

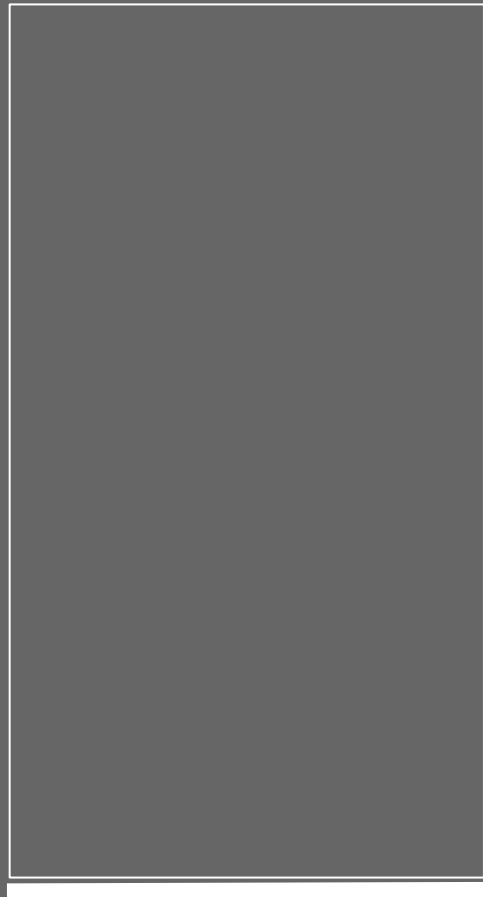


Chuck Moore
Forth



Forth

2 3 ADD 5 EQUALS



Forth

2 3 ADD 5 EQUALS



Forth

2 3 ADD 5 EQUALS



2

Forth

2 3 ADD 5 EQUALS



2

Forth

2 3 ADD 5 EQUALS



3
2

Forth

2 3 ADD 5 EQUALS



3
2

Forth

2 3 ADD 5 EQUALS



(pop)
(pop)

Forth

2 3 ADD 5 EQUALS



5

Forth

2 3 ADD 5 EQUALS



5

Forth

2 3 ADD 5 EQUALS



5
5

Forth

2 3 ADD 5 EQUALS



5
5

Forth

2 3 ADD 5 EQUALS



(pop)
(pop)

Forth

2 3 ADD 5 EQUALS



true

Forth

2 3 ADD 5 EQUALS

true

true

Bitcoin script

Blokujący (z poprzedniej transakcji):

```
DUP HASH160 <pub_key_hash> EQUAL CHECKSIG
```

Odblokowujący (aktualna transakcja):

```
<sig> <pub_key>
```

Bitcoin script

<sig> <pub_key>

DUP HASH160 <pub_key_hash>

EQUAL CHECKSIG



Bitcoin script

<sig> <pub_key>



DUP HASH160 <pub_key_hash>

EQUAL CHECKSIG

<sig>

Bitcoin script

<sig> <pub_key>



DUP HASH160 <pub_key_hash>

EQUAL CHECKSIG

<pub_key>
<sig>

Bitcoin script

<sig> <pub_key>

DUP HASH160 <pub_key_hash>



EQUAL CHECKSIG

<pub_key>
<pub_key>
<sig>

Bitcoin script

<sig> <pub_key>

DUP HASH160 <pub_key_hash>



EQUAL CHECKSIG

<pubkeyhash>
<pub_key>
<sig>

Bitcoin script

<sig> <pub_key>

DUP HASH160 <pub_key_hash>



EQUAL CHECKSIG

<pubkeyhash>
<pubkeyhash>
<pub_key>
<sig>

Bitcoin script

<sig> <pub_key>

DUP HASH160 <pub_key_hash>

EQUAL CHECKSIG



<pub_key>
<sig>

Bitcoin script

<sig> <pub_key>

DUP HASH160 <pub_key_hash>

EQUAL CHECKSIG



true



Profit!

Jeszcze tylko skrypt blokujący
dla następnej transakcji:

DUP HASH160 <pub_key_hash> EQUAL CHECKSIG

<N> <sig1> <sig2> ... <sigM> CHECKMULTISIG

Sprawdza N-z-M podpisów.

`<t> CHECKLOCKTIMEVERIFY`

Blokuje wykonanie transakcji do czasu `<t>`.

RETURN <meta>

Pozwala na przesłanie danych <meta> w transakcji.

Home Layout Tables Charts SmartArt Formulas Data Review

Edit Font Alignment Number Format

Fill Paste Clear Calibri (Body) 12 A A Wrap Text General Conditional Formatting

Normal Bad Good Neutral

	A	B	C	D	E	F	G	H	I	J
1	Date	# Sold	Revenue	Company Name	State	Name	Email	Referral Source	Temporary Text to Colum	Company
2	6/2/11	65	\$ 195.00	Company 1	MA	John	John@Company 1.com	Google	John	Company 1.com
3	6/3/11	67	\$ 201.00	Company 2	CT	Susan	Paul@Company 2.com	Display Ads	Paul	Company 2.com
4	6/4/11	59	\$ 177.00	Company 3	MA	Anna	Romy@Company 3.com	Google	Romy	Company 3.com
5	6/5/11	58	\$ 173.00	Company 4	MA	Tim	Tim@Company 4.com	Google	Tim	Company 4.com
6	6/6/11	55	\$ 164.00	Company 5	CT	Hal	Hal@Company 5.com	Display Ads	Hal	Company 5.com
7	6/7/11	52	\$ 155.00	Company 6	CT	Paul	Susan@Company 6.com	Press	Susan	Company 6.com
8	6/8/11	49	\$ 146.00	Company 7	MA	Alex	Alex@Company 7.com	Display Ads	Alex	Company 7.com
9	6/9/11	46	\$ 137.00	Company 8	MA	Jake	Jake@Company 8.com	Display Ads	Jake	Company 8.com
10	6/10/11	43	\$ 128.00	Company 9	CT	Hugh	Hugh@Company 9.com	Google	Hugh	Company 9.com
11	6/11/11	52	\$ 156.00	Company 10	CT	Alicia	Alicia@Company 10.com	Google	Alicia	Company 10.com
12	6/12/11	45	\$ 135.00	Company 11	CT	Kimberly	Kimberly@Company 11.com	Google	Kimberly	Company 11.com
13	6/13/11	60	\$ 180.00	Company 12	NY	Mary Anne	Mary Anne@Company 12.com	Direct	Mary Anne	Company 12.com
14	6/14/11	62	\$ 186.00	Company 13	NY	Sue	Sue@Company 13.com	Press	Sue	Company 13.com
15	6/15/11	35	\$ 105.00	Company 14	NH	Betty	Betty@Company 14.com	Press	Betty	Company 14.com
16	6/16/11	90	\$ 270.00	Company 15	VT	Jessie	Jessie@Company 15.com	Press	Jessie	Company 15.com
17	6/17/11	92	\$ 276.00	Company 16	NY	Chris	Chris@Company 16.com	Press	Chris	Company 16.com
18	6/18/11	70	\$ 210.00	Company 17	NY	Christopher	Christopher@Company 17.com	Press	Christopher	Company 17.com
19	6/19/11	65	\$ 195.00	Company 18	NY	Alec	Alec@Company 18.com	Bing	Alec	Company 18.com
20	6/20/11	50	\$ 150.00	Company 19	NY	Stinson	Stinson@Company 19.com	Bing	Stinson	Company 19.com
21	6/21/11	45	\$ 135.00	Company 20	CT	Matthew	Matthew@Company 20.com	Display Ads	Matthew	Company 20.com
22	6/22/11	50	\$ 150.00	Company 21	CT	Joe	Joe@Company 21.com	Direct	Joe	Company 21.com
23	6/23/11	50	\$ 150.00	Company 22	MA	Romy	Anna@Company 22.com	Direct	Anna	Company 22.com

Stan (globalnie spójny)

+

Skrypt (globalnie deterministyczny)

Globalny komputer

Ethereum, Lisk



EXPLORE

WORKING FILES

math_test.sol math
perm_db.sol kern
fallback_test.sol lang
sig_helper.sol lang
dataflow.sol data
authority.sol control
proxy_actor.sol control
auth.sol control

standard_authority.sol control

CONTRACTS

control

auth.sol
auth_test.sol
authority.sol
proxy_actor.sol
proxy_actor_test.sol
standard_authority.sol

standard_authority_test.sol
update.sol

data

dataflow.sol
median.sol
median_test.sol

kern

kern.se
klog.sol
perm_db.sol

standard_authority.sol control

```
1 import 'dappsys/control/authority.sol';
2 import 'dappsys/control/auth.sol';
3
4 contract DSStandardAuthority is DSAuthority, DSAuth
5 {
6     function DSStandardAuthority() {
7         _is_root[msg.sender] = true;
8     }
9
10    mapping(address=>bool) public _is_root;
11    mapping(address=>mapping(address=>mapping(bytes4=>bool))) _can_call;
12
13    function can_call( address caller
14                      , address callee
15                      , bytes4 sig )
16        constant
17        returns (bool)
18    {
19        return _can_call[caller][callee][0x0000] == true
20            || _can_call[caller][callee][sig];
21    }
22
23    event set_can_call_event( address caller, address callee, bytes4 sig, bool can );
24
25    function set_can_call( address caller
26                          , address callee
27                          , bytes4 sig
28                          , bool can )
29        auth()
30        returns (bool success)
31    {
32        _can_call[caller][callee][sig] = can;
33        set_can_call_event( caller, callee, sig, can );
34        return true;
35    }
36    event set_root_event( address who, bool is_root );
```

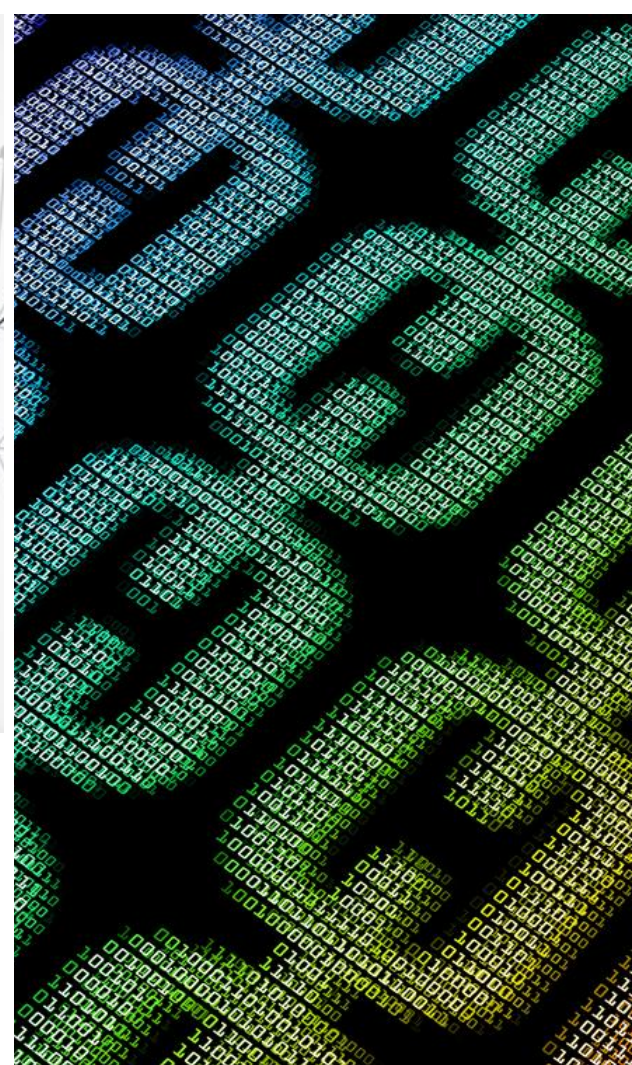
Kiedy stosować blockchain?

- Kwestia zaufania
- Kwestia kontroli
- Kwestia powszechności
- Kwestia wydajności

W pozostałych przypadkach lepiej postawić Oracula.



Wasze pytania



Dziękuję!

Michał Rudnicki
michal.rudnicki@epsi.pl