

20598 – Finance with Big Data

Week 11 (Last) Lecture: Cryptocurrency and Smart Contracts

Clément Mazet-Sonilhac

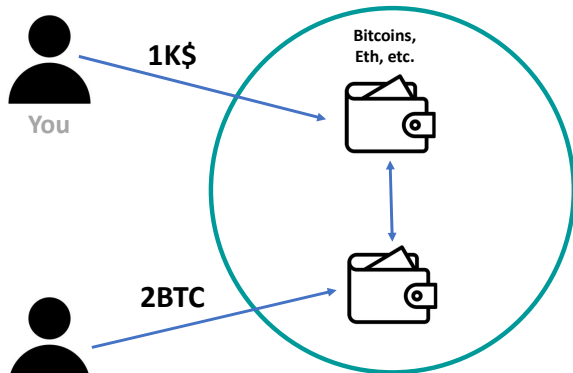
`clement.mazetsonilhac@unibocconi.it`

Department of Finance, Bocconi University

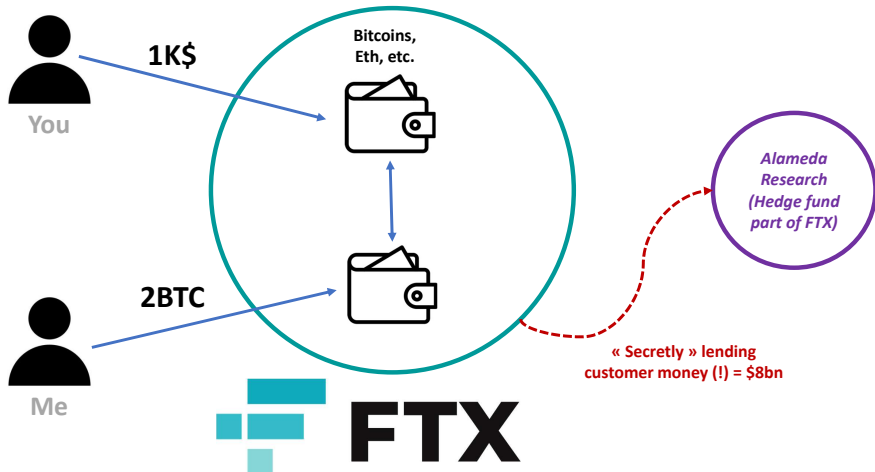
A close-up portrait of Sam Bankman-Fried, the former CEO of FTX. He has dark, curly hair and is looking slightly off-camera with a serious expression. The background is blurred, showing what appears to be a large, ornate chandelier.

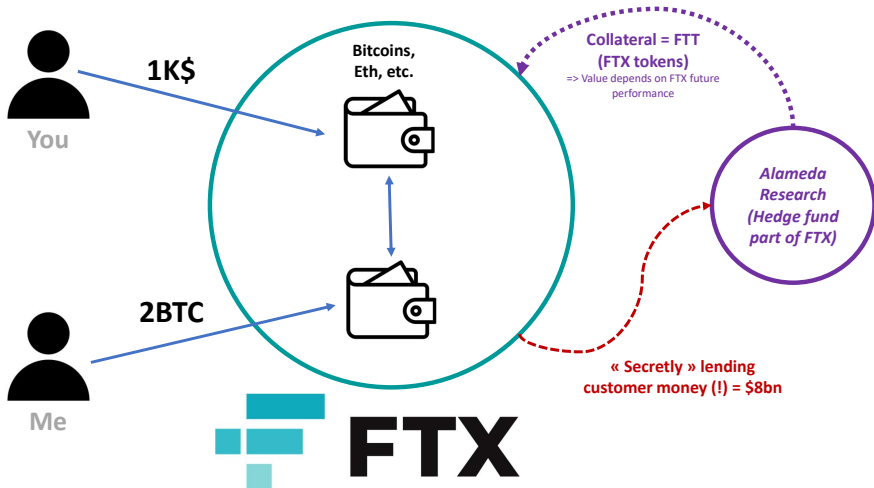
**BREAKING
NEWS**

**NEW FTX CEO: "COMPLETE
FAILURE OF CORPORATE CONTROLS"**



FTX

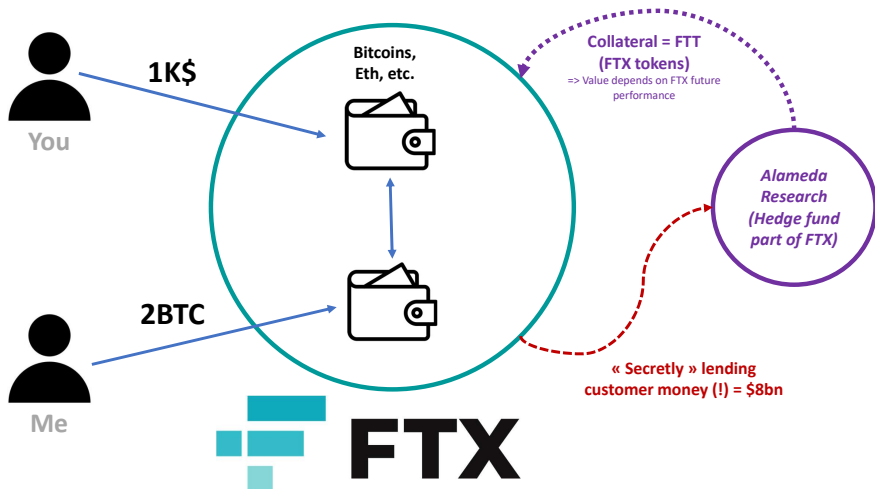




The value of FTT as collateral



Source : CoinMarketCap – [Market Price of FTT](#)



- According to SBF, \$8bn of extra liabilities = old cash deposits from before FTX had bank accounts (debts had been forgotten about for years)

Don't forget about a \$8bn liability

Source : Vox – [article](#)

but not about whether you were lending out customer funds? seems like such an obvious thing for them to worry about!

Yesterday, 9:37 PM

ya

but it's complicated:

- a) it wasn't quite lending them out--it was messier and more organic than that; each step was in isolation rational and reasonable, and then when I finally added it all up last week it wasn't
- b) most exchanges did some variant on what we did--just not as big and without the run on the bank (at least recently) and more intentionally
- c) everyone wants to be clever, and the clever thing to do is some complicated 3d chess involving customer orders or data or something like that which makes no actual sense

like, "oh FTX doesn't have a bank account, I guess people can wire to Alameda's to get money on FTX"

....3 years later...

'oh fuck it looks like people wired \$8b to Alameda and oh god we basically forgot about the stub account that corresponded to that and so it was never delivered to FTX'

Yesterday, 9:41 PM

Don't rely on a Metaverse-based auditing firm

C. Financial Reporting

55. The FTX Group received audit opinions on consolidated financial statements for two of the Silos – the WRS Silo and the Dotcom Silo – for the period ended December 31, 2021. The audit firm for the WRS Silo, Armanino LLP, was a firm with which I am professionally familiar. The audit firm for the Dotcom Silo was Prager Metis, a firm with which I am not familiar and whose website indicates that they are the “first-ever CPA firm to officially open its Metaverse headquarters in the metaverse platform Decentraland.”⁴

56. I have substantial concerns as to the information presented in these audited financial statements, especially with respect to the Dotcom Silo. As a practical matter, I do not believe it appropriate for stakeholders or the Court to rely on the audited financial statements as a reliable indication of the financial circumstances of these Silos.

Source : FTX Chapter 11 First Declaration by John Ray III

Conclusions?

- Nothing specific to crypto



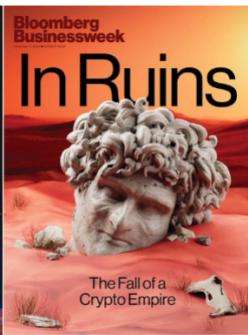
Emin Gün Sirer @el33th4xor · Nov 19

That, plus the thing that failed was a centralized exchange (CEX).



Peter Brandt @PeterLBrandt · Nov 19

This is the type of news that comes at or near a major market bottom
\$BTC \$ETH



43

52

331



Conclusions?

- Nothing specific to crypto but...
- Lender of last resort + regulation is important (Diamond and Dybvig, 1983)
- Transparency (public disclosure)
- Corporate control enforcement + audit: good accounting system for keeping track of where the money is



Recalling definitions

- Blockchain:
- Cryptocurrency:

Recalling definitions

- **Blockchain:** digital database or ledger distributed among the nodes of a peer-to-peer network ([Haber and Stornetta, 1991](#), see also [this](#))
- **Cryptocurrency:**

Recalling definitions

- **Blockchain:** digital database or ledger distributed among the nodes of a peer-to-peer network ([Haber and Stornetta, 1991](#), see also [this](#))
 - IBM supply chain Blockchain solution [\[video here\]](#) and [\[case studies\]](#)
- **Cryptocurrency:**

Recalling definitions

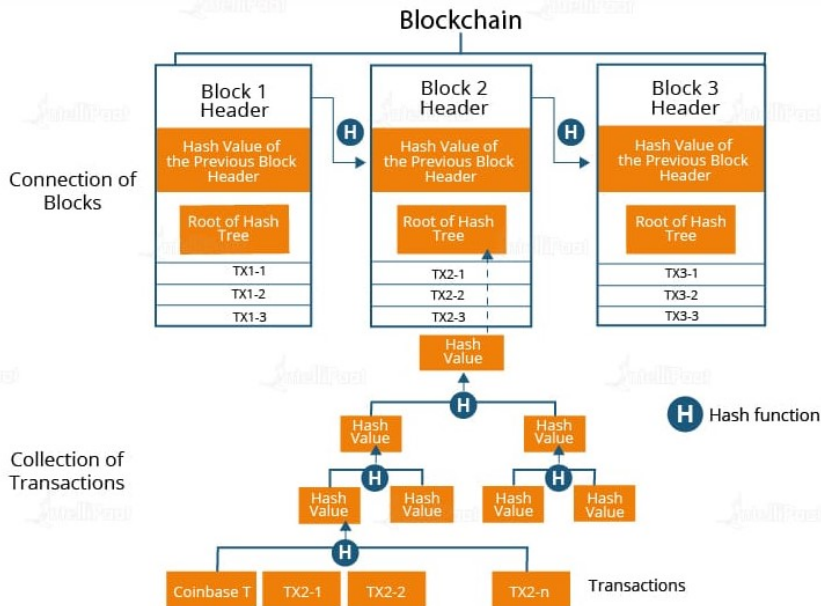
- **Blockchain:** digital database or ledger distributed among the nodes of a peer-to-peer network ([Haber and Stornetta, 1991](#), see also [this](#))
 - IBM supply chain Blockchain solution [\[video here\]](#) and [\[case studies\]](#)
- **Cryptocurrency:** a digital or virtual currency secured by cryptography and based on a decentralized network (a blockchain)

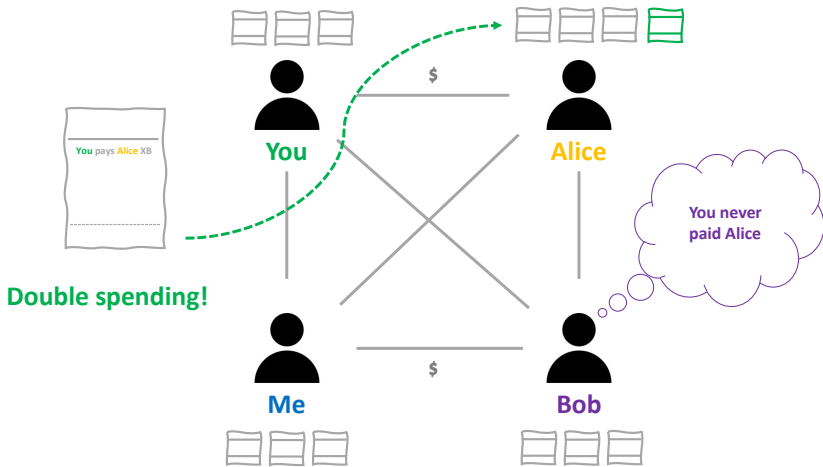
The promise of Bitcoin

A digital currency that is both:

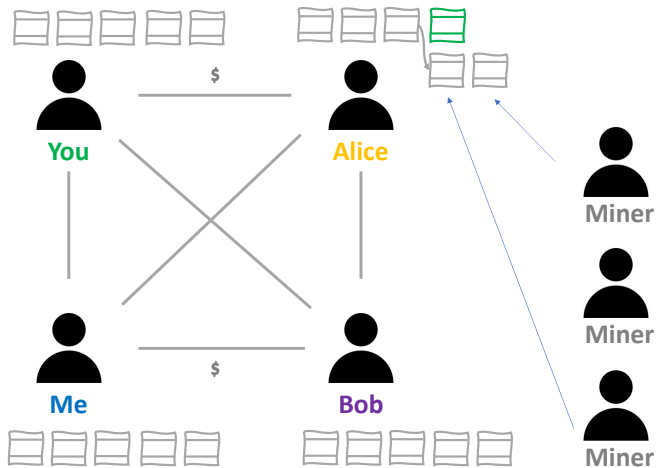
Decentralized
Secured and Immutable
Scalable

Bitcoin and the **blockchain**

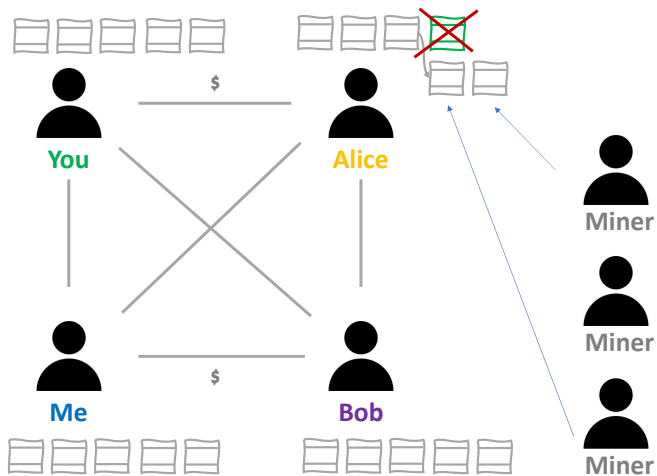




- No double spending: Proof of Work

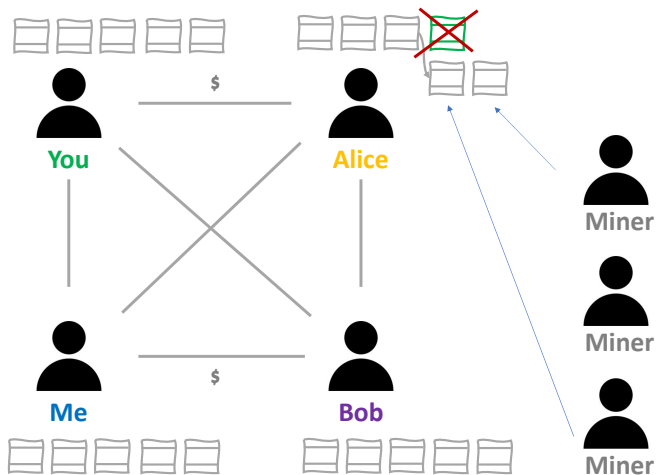


- No double spending: Proof of Work
- Nodes only keep the longest chain



- The fraudulent chain is rejected

What about the **limitations**?

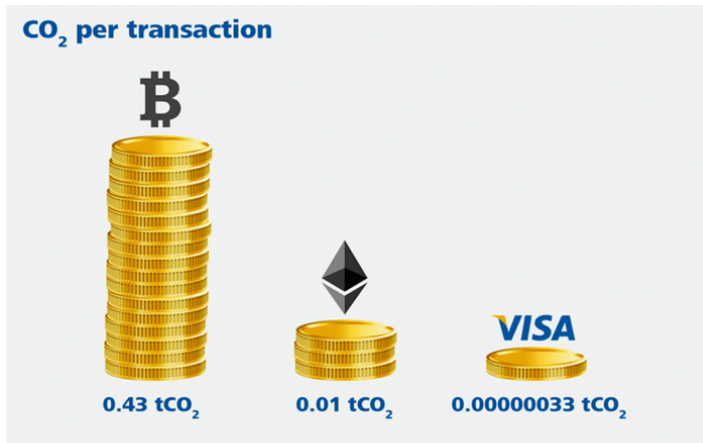


Limitation 1: Costly

- Mining is costly

Limitation 1: Costly

- Mining is **costly** (huge negative externality like CO₂ emissions)
- 200 million tons of CO₂ since its launch

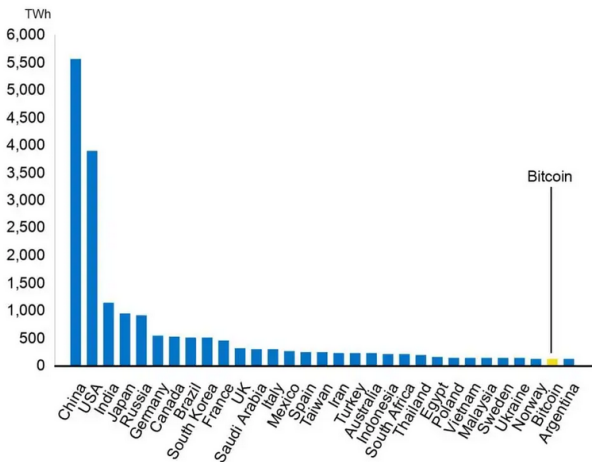


Future of Payments- Carbon Footprint of Bitcoin vs Ethereum vs Visa. Source: cleancoins.io

Limitation 1: Costly

Bitcoin uses more energy than Argentina

If Bitcoin was a country, it would be in the top 30 energy users worldwide



National energy use in TWh

Source: University of Cambridge Bitcoin Electricity Consumption Index

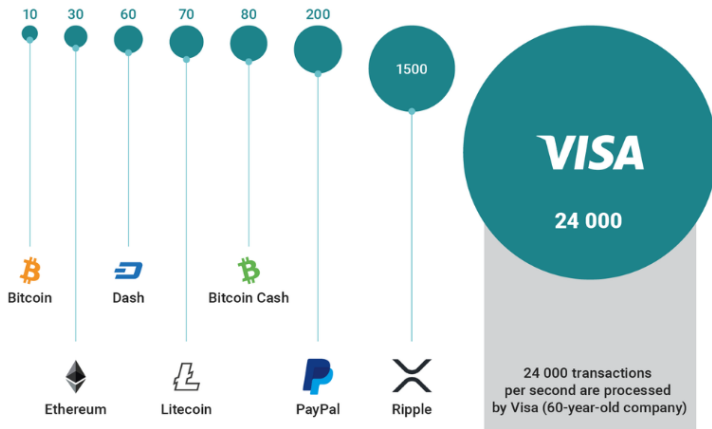
BBC

Limitation 2: Slow

- Mining is slow (limit in number of transactions)

Limitation 2: Slow

- Mining is slow (limit in number of transactions)
- Max $\approx$ 2400 transactions per block



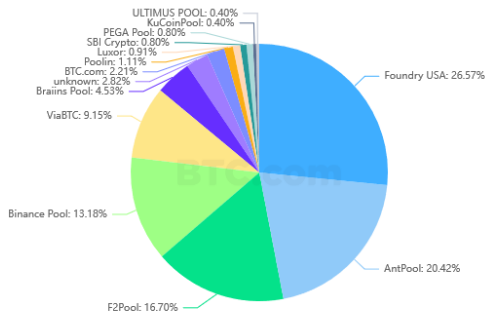
Limitation 3: Limited supply and centralization

- Monetary policy (93.1% of BTC already mined)
- What if a miner has more than 51% of CPU? See [here](#) in real-time.

Limitation 3: Limited supply and centralization

- Monetary policy (93.1% of BTC already mined)
- What if a miner has more than 51% of CPU? See [here](#) in real-time.

Pool Distribution



More Bitcoin weaknesses: Bitcoin Script

- Part of Bitcoin that we haven't talked about yet (because almost no-one sees it): [Bitcoin Script](#)
- It's a [minimalist programming language](#) that handles the details of a transaction such as signature verification
- Why was an entire mini-language created for Bitcoin?
 - to support transactions which require m out of n people to sign to become valid
 - to support slightly more advanced payment logic, such as i inputs, j outputs
- The details of the Bitcoin script specification can be found [here](#)

More Bitcoin weaknesses: Bitcoin Script

- Bitcoin Script has a **huge** problem: it's Turing-incomplete

More Bitcoin weaknesses: Bitcoin Script

- Bitcoin Script has a **huge** problem: it's Turing-incomplete

Compare the following lines of code. This code will run and halt:

```
sleep(1)
```

This code will run forever:

```
while True:  
    continue
```

But what will this code do:

```
while j < i:  
    j = exp(i)-sin(i)*i**2  
    i = tanh(j)
```

General question: Is there a general algorithm which can determine whether this (or any other) code will come to a final stop, looking just at the code and the initial values of its variables (e.g. i and j in the last example)?

More Bitcoin weaknesses: Bitcoin Script

- Alan Turing wrote in 1936: *there exists no general algorithm that can decide whether a program will or will not come to a stop*

More Bitcoin weaknesses: Bitcoin Script

- Alan Turing wrote in 1936: *there exists no general algorithm that can decide whether a program will or will not come to a stop*
- Direct consequences for the design of Blockchain scripting languages like Bitcoin script:
 - either they are by purpose made Turing-incomplete (e.g. by taking away the possibility to have loops)
 - or they can potentially run forever (on the miners' machines!)

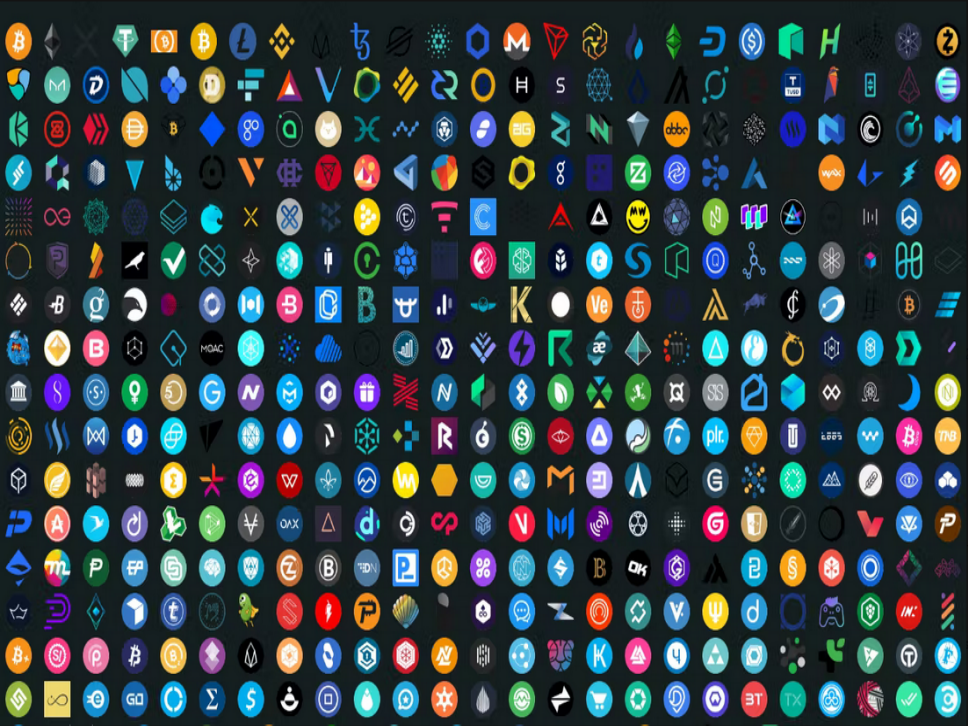


More Bitcoin weaknesses: Bitcoin Script

- Since the scripts must be executed by the network's validator nodes (not on the computer which submitted them), this is a **big security problem**
- Satoshi Nakamoto implemented Bitcoin script as a **Turing-incomplete language**
- It lacks the instructions for **loops** such that it will always be executed sequentially and come to a halt
- The price for this is that **only the most simple algorithms** can be implemented in **Bitcoin script**



5' break



- A Blockchain invented in 2015 by **Vitalik Buterin** (aged 19)



- A Blockchain invented in 2015 by [Vitalik Buterin](#) (aged 19)
- Ethereum has its own currency ETH
- But it's much more than a cryptocurrency: the Blockchain's [protocol](#) allows users to write smart contracts and build decentralized applications
 - it's Turing-complete!
- Read [his view](#) on the FTX blow up. Also the [white](#) and [yellow](#) paper (not easy!)





$$(23) \quad L_R(R) \equiv (R_z, R_a, R_b, R_l)$$

We assert that the status code R_z is a non-negative integer:

$$(24) \quad R_z \in \mathbb{N}$$

We assert that R_a , the cumulative gas used, is a non-negative integer and that the logs Bloom, R_b , is a hash of size 2048 bits (256 bytes):

$$(25) \quad R_a \in \mathbb{N} \quad \wedge \quad R_b \in \mathbb{B}_{256}$$

The sequence R_l is a series of log entries, $(O_0, O_1, \dots)$. A log entry, O , is a tuple of the logger's address, O_a , a possibly empty series of 32-byte log topics, O_t and some number of bytes of data, O_d :

$$(26) \quad O \equiv (O_a, (O_{t0}, O_{t1}, \dots), O_d)$$

$$(27) \quad O_a \in \mathbb{B}_{20} \quad \wedge \quad \forall x \in O_t : x \in \mathbb{B}_{32} \quad \wedge \quad O_d \in \mathbb{B}$$

We define the Bloom filter function, M , to reduce a log entry into a single 256-byte hash:

$$(28) \quad M(O) \equiv \bigvee_{x \in \{O_a\} \cup O_t} (M_{3:2048}(x))$$

where $M_{3:2048}$ is a specialised Bloom filter that sets three bits out of 2048, given an arbitrary byte sequence. It does this through taking the low-order 11 bits of each of the first three pairs of bytes in a Keccak-256 hash of the byte sequence.³ Formally:

$$(29) \quad M_{3:2048}(\mathbf{x} : \mathbf{x} \in \mathbb{B}) \equiv \mathbf{y} : \mathbf{y} \in \mathbb{B}_{256} \quad \text{where:}$$

$$(30) \quad \mathbf{y} = (0, 0, \dots, 0) \quad \text{except:}$$

$$(31) \quad \forall i \in \{0, 2, 4\} : \mathbf{B}_{2047-m(\mathbf{x}, i)}(\mathbf{y}) = 1$$

$$(32) \quad m(\mathbf{x}, i) \equiv \text{KEC}(\mathbf{x})[i, i+1] \bmod 2048$$

$$p_T(k, T) \equiv \left(\text{RLP}(k), \begin{cases} \text{RLP}(L_T(T)) & \text{if } T_x = 0 \\ (T_x) \cdot \text{RLP}(L_T(T)) & \text{otherwise} \end{cases} \right)$$

and

$$(35)$$

$$p_R(k, R) \equiv \left(\text{RLP}(k), \begin{cases} \text{RLP}(L_R(R)) & \text{if } R_x = 0 \\ (R_x) \cdot \text{RLP}(L_R(R)) & \text{otherwise} \end{cases} \right)$$

($\cdot$ is the concatenation of byte arrays).

Furthermore:

$$(36) \quad \text{TRIE}(L_S(\sigma)) = P(B_H)_{H_r}$$

Thus $\text{TRIE}(L_S(\sigma))$ is the root node hash of the Merkle Patricia tree structure containing the key-value pairs of the state σ with values encoded using RLP, and $P(B_H)$ is the parent block of B , defined directly.

The values stemming from the computation of transactions, specifically the transaction receipts, B_R , and that defined through the transaction's state-accumulation function, Π , are formalised later in section 11.4.

4.3.3. Serialisation. The function L_B and L_H are the preparation functions for a block and block header respectively. We assert the types and order of the structure for when the RLP transformation is required:

$$(37) \quad L_H(H) \equiv (H_p, H_o, H_c, H_r, H_l, H_e, H_b, H_d, H_i, H_1, H_g, H_s, H_x, H_m, H_n)$$

$$(38) \quad L_B(B) \equiv (L_H(B_H), \tilde{L}_T^*(B_T), L_{\Pi}^*(B_U))$$

where $\tilde{L}_T$ takes a special care of EIP-2718 transactions:

$$(39) \quad \tilde{L}_T(T) = \begin{cases} L_T(T) & \text{if } T_x = 0 \\ (T_x) \cdot \text{RLP}(L_T(T)) & \text{otherwise} \end{cases}$$

Ethereum : Stopping the unstoppable

- Vitalik Buterin found a great solution to manage the Turing-complete risk

Ethereum : Stopping the unstoppable

- Vitalik Buterin found a great solution to manage the **Turing-complete risk**
- With a car, you can't go any further if you've run out of gas. Why not apply this principle to scripts for the Blockchain?

Ethereum : (Out) of Gas

- Every instruction costs a certain amount of gas
- You're required to endow every contract with a certain amount of gas (`gasLimit` parameter) that gets depleted when processed by miners, and miners get paid
- When your contract is out of gas, it stops running, fees remain paid. No matter what you did or didn't get to do
- The gas price is indirectly determined by miners:
 - miners pick transactions according to the `gasPrice` that the transaction is bidding for and decline to process contracts that bid too low a gas price
 - High `gasPrice` buys you priority processing. Check the current market prices of your fuel grade at the [gas station](#)

Ethereum 's Architecture

- Vitalik Buterin blockchain features a **Turing-complete** virtual machine coded in **Solidity**
- Two kind of accounts:
 - **external accounts** which are controlled by public keys (as in Bitcoin)
 - **contract accounts** which are governed by a **smart contract**, i.e. a collection of code being executed on the Ethereum Virtual Machine
- Every time a contract account receives a *message* its code activates, allowing it to read and write to internal storage
- Homework: read the full **white paper** of Ethereum

Ethereum 's Smart Contracts

Examples of functions that can be codified as **Solidity** smart contracts :

- Financial derivatives: code can make payments at a given point in the future which will be contingent on the value of other inputs like weather

Ethereum 's Smart Contracts

Examples of functions that can be codified as **Solidity** smart contracts :

- Financial derivatives: code can make payments at a given point in the future which will be contingent on the value of other inputs like weather
- Cryptotokens : code that issues and manages **a set of unique tokens** which may give their owners rights to receive cash flows from an investment, or derive utility from a service

Ethereum 's Smart Contracts

Examples of functions that can be codified as [Solidity](#) smart contracts :

- Financial derivatives: code can make payments at a given point in the future which will be contingent on the value of other inputs like weather
- Cryptotokens : code that issues and manages [a set of unique tokens](#) which may give their owners rights to receive cash flows from an investment, or derive utility from a service
- Distributed organizations : One view of the firm is that it comprises of a web of contracts that regulate state-contingent payments. In principle one could think to implement distributed organizations, firms, venture capitalists, financial intermediaries via smart contracts (see [DAO](#))

Ethereum 's Smart Contracts: Code is Law

- The **main advantage of smart contracts** is that they execute **automatically**:
 - In a normal contractual agreement, the counterparty can default on its obligations
 - Then, a lengthy legal process will try to force that the contract is enforced
 - In smart contracts, instead the enforcement is automatic

Ethereum 's Smart Contracts: Code is Law

- The **main advantage of smart contracts** is that they execute **automatically**:
 - In a normal contractual agreement, the counterparty can default on its obligations
 - Then, a lengthy legal process will try to force that the contract is enforced
 - In smart contracts, instead the enforcement is automatic
- But code is just code. It will do what you told it to do, which is sometimes very different from what you wanted it to do
 - Bugs are going to be executed
 - Bad real-world contracts get renegotiated. Bad smart contracts just execute nevertheless.

Ethereum 's Token

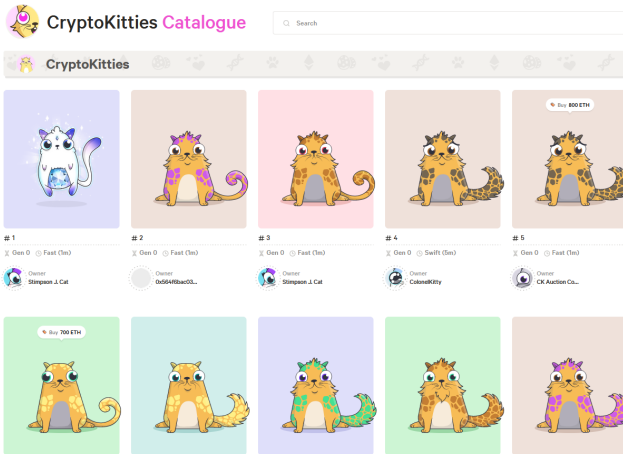
- One of the **most frequently used contract** on Ethereum is an ERC-20 compliant token contract
- It's a contract allowing you to issue a **finite number of tokens whose ownership** can be traced by the blockchain
- What can tokens be good for?
 - Utility tokens: like bus tickets or train tickets, they are proofs that you paid the relevant fare for a service that you're receiving. Utility tokens allows token holders to perform a specific function on the network.
 - Security token: ownership rights
 - Non-fungible token: ownership rights + unicity

Ethereum 's Token: Example

- Can you name one very famous (non-fungible) Token based on an Ethereum smart contract?

Ethereum's Token: Example

- Can you name one very famous (non-fungible) Token based on an Ethereum smart contract?



Ethereum 's Token: Example

- Soon after its launch in November 2017, [CryptoKitties](#) became one of the most successful distributed token contract ever on the Ethereum blockchain
 - [Cryptokittie](#) was single most frequently traded contract on the Ethereum network in 2018
 - 25% of network traffic on Ethereum
- Causing substantial scalability problems and slowdown on the Ethereum network, resulting also in massive Ether devaluation!

Let's code a Cryptozombies in Solidity

- The cryptozombies.io website was designed to make fun of those silly applications
- Instead of crowding the Blockchain with zombies, it teaches how to code in [Solidity](#)
- It's really fun and simple



Let's code a Cryptozombies in Solidity

```
// First step: declare the environment:
```

```
pragma solidity >=0.5.0 <0.6.0;
```

```
contract ZombieFactory {
```

```
}
```

Let's code a Cryptozombies in Solidity

```
// Second step: Create the size of the DNA and a Zombie structure (and list)

pragma solidity >=0.5.0 <0.6.0;

contract ZombieFactory {

    uint dnaDigits = 16;
    uint dnaModulus = 10 ** dnaDigits;

    struct Zombie {
        string name;
        uint dna;
    }

    Zombie[] public zombies;

}
```

Let's code a Cryptozombies in Solidity

```
// Third step: Create functions to generate zombies randomly
pragma solidity >=0.5.0 <0.6.0;

contract ZombieFactory {

    uint dnaDigits = 16;
    uint dnaModulus = 10 ** dnaDigits;

    struct Zombie {
        string name;
        uint dna;}

    Zombie[] public zombies;

    function _createZombie(string memory _name, uint _dna) private {
        zombies.push(Zombie(_name, _dna));}

    function _generateRandomDna(string memory _str) private view returns (uint) {
        uint rand = uint(keccak256(abi.encodePacked(_str)));
        return rand % dnaModulus;}

    function createRandomZombie(string memory _name) public {
        uint randDna = _generateRandomDna(_name);
        _createZombie(_name, randDna);}
}
```

Let's try our Cryptozombies smart contract

Click [here](#)

Ethereum 's Token: ICOs

- Like stock, which is sold in an initial public offering (IPO), tokens are sold in **initial coin offerings** (ICO)
- Unlike IPOs, many successful ICO's **have been made without any help from investment banks** and an army of lawyers

Ethereum 's Token: ICOs

- Like stock, which is sold in an initial public offering (IPO), tokens are sold in **initial coin offerings** (ICO)
- Unlike IPOs, many successful ICO's **have been made without any help from investment banks** and an army of lawyers
- Legality, however, has been a big problem. Some crypto tokens fall under the same regulations as stocks
- Failure to follow the regulations for stock issuance (registration with SEC, prospectus, etc) is against the law
 - EOS sold over \$4bn in cryptotokens in 2018 without a live product to show
 - got into trouble with the SEC, settled for a fine of \$24M

Ethereum 's Token: ICOs

- Like stock, which is sold in an initial public offering (IPO), tokens are sold in **initial coin offerings** (ICO)
- Unlike IPOs, many successful ICO's **have been made without any help from investment banks** and an army of lawyers
- Legality, however, has been a big problem. Some crypto tokens fall under the same regulations as stocks
- Failure to follow the regulations for stock issuance (registration with SEC, prospectus, etc) is against the law
 - EOS sold over \$4bn in cryptotokens in 2018 without a live product to show
 - got into trouble with the SEC, settled for a fine of \$24M
- To evade regulation and oversight, many ICOs were held by firms located on the Cayman Islands or elsewhere outside U.S. jurisdiction. Naive investor demand for crypto-assets was stable irrespective of the legality of the transaction.

Ethereum 's Token: ICOs

- The technical ease of holding an ICO highlights that cryptoassets, when regulated and overseen properly, hold the promise of greatly improving access to (crowd-)funding for small and medium size firms that are too small to ever even consider an IPO
- Global scope and tradability of tokens makes it easier to attract investors in highly specific assets
- Cost of issuance are (were) much lower
- Boundary between financial investors and clients of the company becomes more blurry when utility tokens are used to crowdfund the production of goods/services
- Economists' view on ICOs : Potentially useful to overcome financing frictions
 - [Li and Mann, 2022](#), [Lee and Parlour, 2020](#) and [Li and Mann, 2021](#)

Economics of ICOs: Boreiko and Sahdev, 2018

Table 1. Blockchain funding vs. earlier alternatives

Characteristics	IPOs	Venture capital	Reward- and equity-crowdfunding	ICOs / ITOs
Marketing channels	underwriters	private negotiations	online platforms	social media
Investor base	exclusive, local	Partners	exclusive, local	inclusive, global
Intermediation	syndicates	VCs	online platforms	no
Asymmetric information	average	Low	above average	highest
Funding currency	fiat	Fiat	fiat	crypto/fiat
Investor Protection	courts	Automatic	courts	no
Regulation	"standardized"	"standardized"	country-specific	no
Acquisition of	ownership stake	ownership stake	product/ownership	future service
Monitoring ex-post	governance mechanisms	VCs	absent	absent
Lock-ups	owners	No	unclear	developers
Market-making	lead underwriter	N/A	No	decentralized
Liquidity	low	N/A	No	high

The table lists the main characteristics of ICO fundraising mechanism as compared to VC, Crowdfunding and IPO alternatives.

- First empirical analysis of ICOs

Economics of ICOs: Howell et al., 2020

- 1500 ICOs that collectively raise \$12.9 billion
- Which issuer and ICO characteristics predict successful real outcomes (increasing issuer employment and avoiding enterprise failure)?

Economics of ICOs: Howell et al., 2020

Relationship between issuer characteristics and employment

Dependent variable:	Employment			Old employment (4)	Employment growth	
	(1)	(2)	(3)		(5)	(6)
Utility value	0.308*** (0.102)	0.341*** (0.106)	0.343*** (0.106)	0.184** (0.078)	0.125* (0.068)	0.146** (0.060)
White paper	0.185 (0.190)	0.200 (0.170)	0.210 (0.167)	0.341** (0.154)	-0.156 (0.113)	-0.123 (0.117)
Incentive pool	0.177* (0.088)	0.148 (0.094)	0.192* (0.093)	0.251*** (0.077)	-0.074 (0.094)	-0.050 (0.095)
Insider vesting	0.230** (0.105)	0.228** (0.104)	0.230** (0.096)	0.265*** (0.081)	-0.034 (0.056)	-0.062 (0.060)
Budget	0.212* (0.102)	0.219** (0.092)	0.217*** (0.072)	0.198 (0.117)	0.014 (0.058)	0.024 (0.084)
VC equity	0.854*** (0.125)	0.838*** (0.134)	0.809*** (0.130)	0.603*** (0.129)	0.251*** (0.080)	0.289*** (0.085)
Male	-0.507** (0.205)	-0.388 (0.249)	-0.440* (0.221)	-0.051 (0.299)	-0.456 (0.305)	-0.448* (0.228)
Crypto experience	0.083 (0.111)	0.079 (0.122)	0.024 (0.106)	-0.109 (0.118)	0.192* (0.092)	0.180** (0.082)
Finance experience	0.195** (0.077)	0.183** (0.091)	0.115 (0.078)	0.109 (0.124)	0.086 (0.132)	0.061 (0.117)
Comp. sci. experience	0.252*** (0.031)	0.239*** (0.082)	0.238*** (0.040)	0.206*** (0.047)	0.046 (0.055)	0.071 (0.068)
Entrep. experience	0.397*** (0.107)	0.387*** (0.098)	0.412*** (0.085)	0.281*** (0.078)	0.116 (0.100)	0.145 (0.106)
Observations	961	961	940	961	961	940
R ²	.168	.244	.204	.185	.058	.089

Economics of ICOs: Howell et al., 2020

Table 5
Relationship between issuer characteristics and issuer failure

Dependent variable:	Failure			
	(1)	(2)	(3)	(4)
Utility value	-0.070*** (0.019)	-0.755*** (0.258)	-0.049** (0.018)	-0.097*** (0.028)
White paper	-0.073** (0.029)	-0.438 (0.277)	-0.091*** (0.027)	-0.081 (0.053)
Incentive pool	-0.080*** (0.017)	-1.017*** (0.350)	-0.076*** (0.018)	-0.078*** (0.023)
Insider vesting	-0.032** (0.015)	-0.978** (0.423)	-0.029** (0.012)	-0.032** (0.013)
Budget	-0.058*** (0.019)	-0.770*** (0.289)	-0.050** (0.020)	-0.062*** (0.018)
VC equity	-0.079*** (0.013)		-0.063*** (0.013)	-0.067*** (0.013)
Male	0.059* (0.030)	0.661 (0.617)	0.036 (0.036)	0.051 (0.041)
Crypto experience	-0.025* (0.014)	-0.360 (0.245)	-0.029** (0.013)	-0.024 (0.017)
Finance experience	-0.019 (0.014)	-0.294 (0.233)	-0.002 (0.014)	-0.032* (0.018)
Comp. sci. experience	-0.046*** (0.013)	-0.770*** (0.192)	-0.049*** (0.013)	-0.050*** (0.018)
Entrep. experience	-0.029 (0.026)	-0.397 (0.357)	-0.028 (0.025)	-0.042** (0.020)
Observations	961	954	940	961
R ²	.152		.179	.242

- Success associated with disclosure, credible commitment to the project, and quality signals (also [this paper](#))

Decentralized Finance (Schar, 2021)

Ethereum vs. Bitcoin: Other differences

- Initial amount of coins: ETH were sold while BTC were mined (less control and wealth concentration in hands of original developers)
- Permanent linear issuance of new currency, so no upper ceiling (unlike BTC), not planning for built-in deflation and allowing for coin loss over time.
- Strong governance model
- Recently: Proof of Stake (more in a few slides)



Stablecoins

Stablecoins

- A cryptocurrency **designed to hold a steady value**, in sharp contrast to the extreme price volatility seen for Bitcoin and Ethereum
- Stablecoins are meant to be **useful**, not to make their owners rich by soaring in value
- Useful in two ways:

Stablecoins

- A cryptocurrency **designed to hold a steady value**, in sharp contrast to the extreme price volatility seen for Bitcoin and Ethereum
- Stablecoins are meant to be **useful**, not to make their owners rich by soaring in value
- Useful in two ways:
 - by allowing crypto owners to conduct transactions without having to take volatility into account
 - by offering them a safe haven for their holdings, protected from wild swings in the crypto market (without having to convert their holdings into traditional money)
- The most popular stablecoin, **Tether**, can be exchanged for more than 4,000 other cryptocurrencies on centralized exchanges

Stablecoins: Tether (+0.10%) vs. Bitcoin (-65.5%)



Stablecoins: How does it work?

- Stablecoins are **pegged to another currency**, e.g. linked to the US dollar.
- They do that in two ways.

Stablecoins: How does it work?

- Stablecoins are **pegged to another currency**, e.g. linked to the US dollar.
- They do that in two ways. The most popular (and mechanical) is the following:
 - Tether and many other of the biggest cryptocurrencies maintain a reserve of cash or cash-equivalent assets whose value in theory matches the total value of the stablecoin in circulation
 - That is, when a user pays Tether \$1 for a token, that money is supposed to be held in Tether's bank accounts

Stablecoins: How does it work?

- Stablecoins are **pegged to another currency**, e.g. linked to the US dollar.
- They do that in two ways. The most popular (and mechanical) is the following:
 - Tether and many other of the biggest cryptocurrencies maintain a reserve of cash or cash-equivalent assets whose value in theory matches the total value of the stablecoin in circulation
 - That is, when a user pays Tether \$1 for a token, that money is supposed to be held in Tether's bank accounts
- Some stablecoins, however, took the second route: operating as an **algorithmic stablecoin** (e.g., TerraUSD).

Algorithmic stablecoin

- It's a currency backed by automated operations meant to maintain a stablecoin's value by **increasing or decreasing its supply**.
- The idea was to create a fully decentralized currency not ultimately backed by a centralized issuer of assets like the US government
 - Smart contracts are programmed to automatically create more units of a stablecoin or destroy existing units in response to swings in its supply and demand
- When the stablecoin trades above its pegged value, more tokens are created and the price comes down
- When the stablecoin trades below the peg, more tokens are taken out of the circulation and the price comes up.
- A sister token with a volatile price is usually involved.

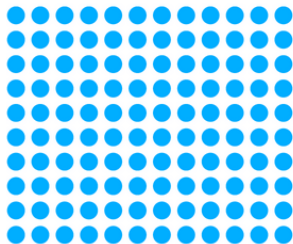
What was TerraUSD's model?

- TerraUSD was launch in 2020 by Do Kwon (31 yo)
- TerraUSD was linked to a sister token, Luna, whose price was set by the market
- Because 1 Terra was defined as being equal to \$1 worth of Luna, that meant that while the amount of Luna handed over in a swap for Terra would vary, a holder of \$1 in Terra would always get \$1 in value back
- That created arbitrage incentives for traders that were designed to keep the value of Terra at or close to \$1.

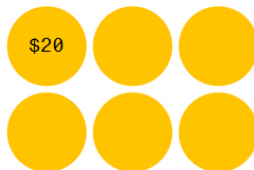
What was TerraUSD's model?

1 UST is always considered to be **equal to \$1**

UST
(Stablecoin)



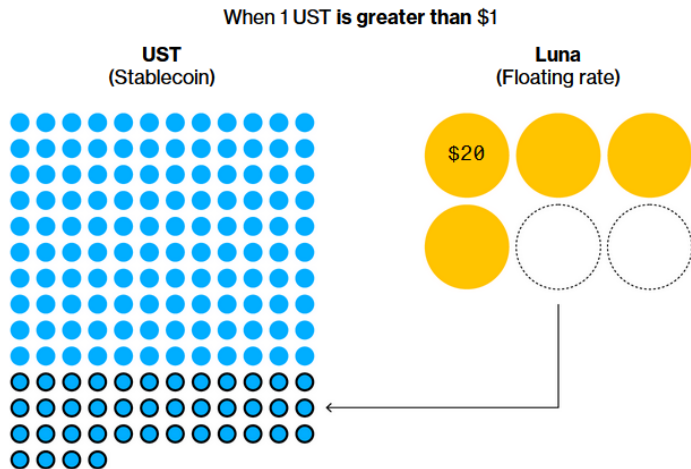
Luna
(Floating rate)



- One Terra (UST) can always be exchanged for a floating quantity of Luna with Market value of \$1
- Whenever either of them is exchanged, the currency is taken out of the circulation by the smart contract programmed to maintain the system

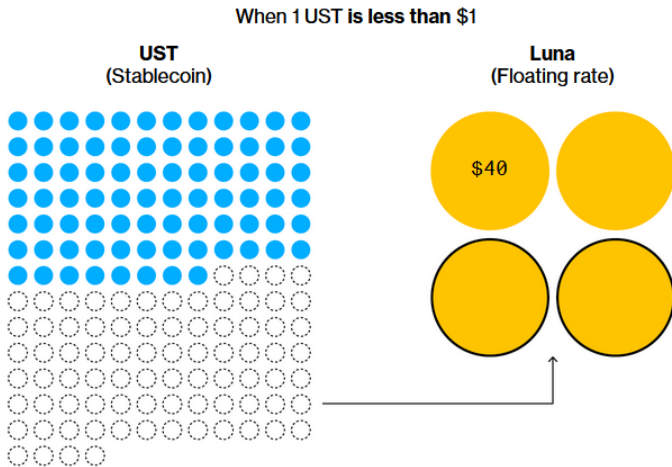
Source : Bloomberg's [article](#)

What was TerraUSD's model?



- When 1 Terra (UST) is $> \$1 \rightarrow$ incentive to trade \$1 worth of Luna against one Terra ($> \$1$)
- This puts more Terra into circulation ; driving down the price of Terra

What was TerraUSD's model?



- Vice versa, when 1 Terra (UST) is $< \$1 \rightarrow$ incentive to trade 1 Terra for \$1 worth of Luna
- This takes some Terra out of circulation ; driving up the price of Terra

Why do people would like to buy Terra/Luna in the first place?

- The primary challenge for all new cryptocurrencies is to create reasons for people to buy them
- Demand can be driven by the prospect of huge gains if the token catches on, or can be primed by providing a reward to currency holders

Why do people would like to buy Terra/Luna in the first place?

- The primary challenge for all new cryptocurrencies is to create reasons for people to buy them
- Demand can be driven by the prospect of huge gains if the token catches on, or can be primed by providing a reward to currency holders
- In the case of TerraUSD, the main attraction involved Anchor:
 - A lending project based on Terra blockchain that promised interest rates as high as 20% (??) for Terra deposits
 - The high returns for depositing Terra on Anchor will in turn be subsidized by new investors in Terra

A Ponzi scheme?

- Some called Terra a new form of **Ponzi scheme** (a form of fraud that lures investors and pays profits to earlier investors with funds from more recent investors)
- Others more politely said that Terra's business model had a vulnerability at its heart: Terra's peg to Luna would only work if people believed that Luna would keep its value...
... and that they would only keep their value if more and more people bought them
- Since the main attraction was the high returns for depositing Terra on Anchor – which in turn was being subsidized by new investors – that didn't seem like a sustainable enterprise
- If demand for either UST or Luna fell, they said, the value of both could evaporate in what's known as a “death spiral.”

A Ponzi scheme? No, according to very clever people

Terra Money: Stability and Adoption

Evan Kereiakes, Do Kwon, Marco Di Maggio, Nicholas Platias

April 2019

Abstract

While many see the benefits of a price-stable cryptocurrency that combines the best of both fiat and Bitcoin, not many have a clear plan for the adoption of such a currency. Since the value of a currency as a medium of exchange is mainly driven by its network effects, a successful new digital currency needs to maximize adoption in order to become useful. We propose a cryptocurrency, Terra, which is both price-stable and growth-driven. It achieves price-stability via an elastic money supply, enabled by stable mining incentives. It also uses seigniorage created by its minting operations as transaction stimulus, thereby facilitating adoption. There is demand for a decentralized, price-stable money protocol in both fiat and blockchain economies. If such a protocol succeeds, then it will have a significant impact as the best use case for cryptocurrencies.

How \$60 Billion in Terra Coins Went Up in Algorithmic Smoke?

- Again, some rumors and a *bank run*

The collapse of Terra (Luna)

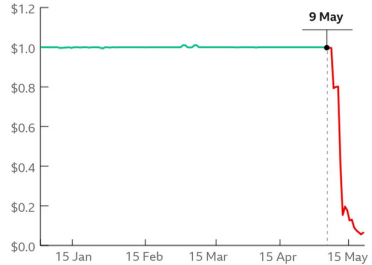


Daily closing price in USD, last three months (22 February - 22 May 2022)

Source: CoinMarketCap

B B C

The collapse of TerraUSD (UST)



Daily closing price in USD, year to date (01 January - 22 May 2022)

Source: CoinMarketCap

B B C

How \$60 Billion in Terra Coins Went Up in Algorithmic Smoke?

- Approximately 200,000 South Korean households and Luna investors harmed by the crash
- Filled a class-action lawsuit against Terra
- The South Korean government also announced that it would pursue Kwon on criminal charges
- Kwon Do created Luna 2.0 last June

Interpol seeks arrest of failed crypto-firm boss Do Kwon

© 26 September



| Terra founder Do Kwon

Interpol has asked law enforcement agencies worldwide to find and arrest Do Kwon, founder of the failed cryptocurrency Terra.



5' break



Proof of Stake

Proof of Stake Idea

 Author

Topic: Proof of stake instead of proof of work (Read 32365 times)

QuantumMechanic (OP)
Member


Activity: 110
Merit: 19


 **Proof of stake instead of proof of work**
July 11, 2011, 04:12:45 AM
Merited by ETBitcoin (3), Vod (2), webtricks (2), d5000 (1), drays (1)

#1

I've got an idea, and I'm wondering if it's been discussed/ripped apart here yet:

I'm wondering if as bitcoins become more widely distributed, whether a transition from a proof of work based system to a proof of stake one might happen. What I mean by proof of stake is that instead of your "vote" on the accepted transaction history being weighted by the share of computing resources you bring to the network, it's weighted by the number of bitcoins you can prove you own, using your private keys.

For those that don't want to be actively verifying transactions, and so that not all private keys need to be facing the network, votes could be delegated to other addresses via some kind of nonstandard Bitcoin transaction. In this way, voting power would accumulate with trusted delegates instead of miners. New bitcoins and transaction fees could be randomly and periodically distributed to delegates, weighted by the number of votes they've accumulated, thereby incentivising diversity of the delegates and direct voters.

If the implementation could be done, it proved to maintain at least a similar level of privacy and trustworthiness, and it only minimally complicated the UX, I'm thinking that a proof of stake based fork could out-compete a proof of work one due to much lower transaction fees, since its network wouldn't need to support the cost of the miners' computing resources. (Note that the vote delegation scheme has bandwidth/storage overhead that would offset these savings by some amount which would hopefully be relatively small.)

Some other potential improvements this system could offer:

- Possibly quicker, more definite confirmation of transactions, depending on how it can be implemented.
- The "voting power" may be more trustworthy, since it would accumulate in a bottom-up fashion via a network of trust, instead of in the somewhat arbitrary way it accumulates now. (Note the potential problem of vote-buying here.)
- It would remove the physical point of failure of bitcoin mining equipment, which can be confiscated or made illegal to run.
- It could be used to provide stakeholders a means of making their voices heard (via the delegated voting system it establishes) when it comes to proposals for software updates and protocol changes.

Anyway, I just wanted to throw the idea out here to see if there are any obvious reasons why it couldn't be implemented, and to hopefully spark a discussion amongst those better qualified than me.

Cheers.

Ethereum and Proof of Stake

- Validator (pseudo-) randomly chosen depending on its stake (i.e., the amount of money owned)
- Verify and validate a block
- If she/he validates fraudulent transactions, loose its stake
- Receive transaction fees ([video explainer](#))

More consensus algorithms

- Physical proofs
 - proof of work (BTC, ETH 1.0)
 - proof of elapsed time (use Intel's trusted execution environment)
 - proof of (disk) space (or retrievability, data possession ; Chia)
 - proof of space-time (similar, but over some amount of time)
 - proof of capacity (Signum)
- Economic proofs
 - proof of stake (ETH 2.0, . . .)
 - pure proof of stake (Algorand)
 - proof of authority (Lugano's triple-A blockchain)
 - proof of burn

More consensus algorithms

- Physical proofs
 - proof of work (BTC, ETH 1.0)
 - proof of elapsed time (use Intel's trusted execution environment)
 - proof of (disk) space (or retrievability, data possession ; Chia)
 - proof of space-time (similar, but over some amount of time)
 - proof of capacity (Signum)
- Economic proofs
 - proof of stake (ETH 2.0, . . .)
 - **pure proof of stake (Algorand)**
 - proof of authority (Lugano's triple-A blockchain)
 - proof of burn

Algorand

Brief History of Algorand

- 2017: Work started, goal to improve over Bitcoin's inefficiencies
- 2018: Company was funded by Silvio Micali (MIT, Turing prize)
- Feb. 2018: \$4M seed funding from Pillar and Union Square Ventures
- Oct. 2018: \$62M additional venture capital funding
- 2019: Auction of ALGO token (ICO)
- 2020: First carbon negative blockchain

First carbon negative blockchain

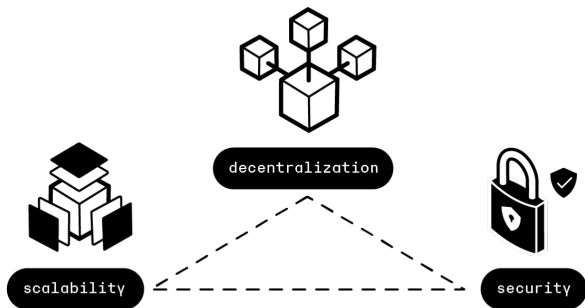
[About Us](#)[Resources](#)[Marketplace](#)[API](#)[Blog](#)[Contact](#)[Log in](#)[Sign up](#)

About us

At CLIMATE TRADE we connect companies willing to offset their carbon emissions to a large number of verified environmental projects.

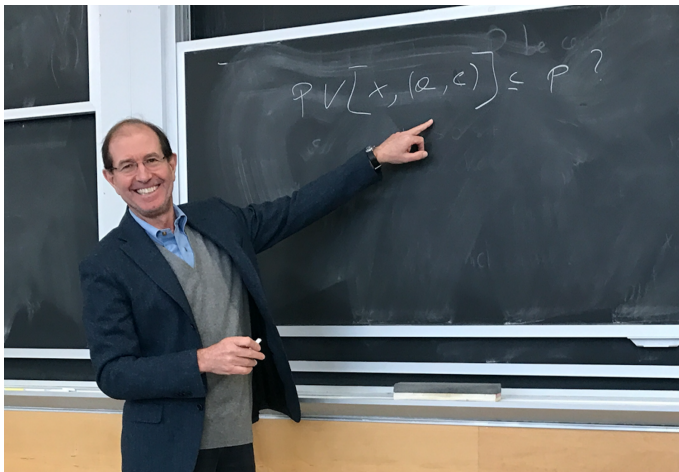
We aim to help companies achieve carbon neutrality by providing our innovative emissions offsetting services.

Solving Blockchain Trilemma



- Vitalik Buterin: you can have scalability, decentralization or security, **but you cannot have all three**

Silvio Micali key idea



- Scaling consensus? [\[Read article here\]](#)

Pure Proof of Stake

- PoW works well if more than 51% of miners are trustworthy
- PoS works well if more than 51% of the nodes are trustworthy
- What if not? If 51% of nodes are compromised, a majority vote would confirm fake transactions

Pure Proof of Stake

- PoW works well if more than 51% of miners are trustworthy
 - PoS works well if more than 51% of the nodes are trustworthy
 - What if not? If 51% of nodes are compromised, a majority vote would confirm fake transactions
- + In Proof of Stake, validators self-select (only the largest stake holders participate)
- Issue: rich get richer, more centralization

Pure Proof of Stake

- PoW works well if more than 51% of miners are trustworthy
 - PoS works well if more than 51% of the nodes are trustworthy
 - What if not? If 51% of nodes are compromised, a majority vote would confirm fake transactions
- + In Proof of Stake, validators self-select (only the largest stake holders participate)
- Issue: rich get richer, more centralization
- Solution: **Pure Proof of Stake** (all the users can be selected, but then they vote depending on their stake. Not one validator, but a pool of validators)

Algorand White Paper

ALGORAND*

Jing Chen
Computer Science Department
Stony Brook University
Stony Brook, NY 11794, USA
jingchen@cs.stonybrook.edu

Silvio Micali
CSAIL
MIT
Cambridge, MA 02139, USA
silvio@csail.mit.edu

Abstract

A public ledger is a tamperproof sequence of data that can be read and augmented by everyone. Public ledgers have innumerable and compelling uses. They can secure, in plain sight, all kinds of transactions —such as titles, sales, and payments— in the exact order in which they occur. Public ledgers not only curb corruption, but also enable very sophisticated applications —such as cryptocurrencies and smart contracts. They stand to revolutionize the way a democratic society operates. As currently implemented, however, they scale poorly and cannot achieve their potential.

Algorand is a truly democratic and efficient way to implement a public ledger. Unlike prior implementations based on proof of work, it requires a negligible amount of computation, and generates a transaction history that will not “fork” with overwhelmingly high probability.

Algorand is based on (a novel and super fast) message-passing Byzantine agreement.

- Read the introduction of the [Algorand white paper](#)
- Probability to fork: one in a trillion

1341v9 [cs.CR] 26 May 2017

Algorand Empirical White Paper

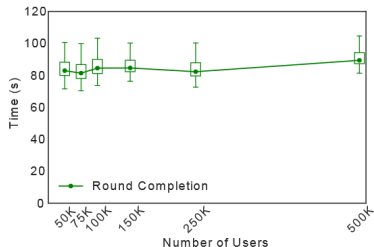


Figure 6: Latency for one round of Algorand in a configuration with 500 users per VM, using 100 to 1,000 VMs.

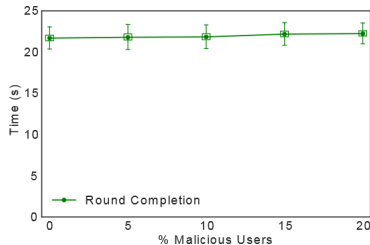


Figure 8: Latency for one round of Algorand with a varying fraction of malicious users, out of a total of 50,000 users.

- Read the [Algorand empirical white paper](#)
- Main results: achieves 125x Bitcoin's throughput, and incurs almost no penalty for scaling to more users

Crypto Hands-On with Algorand

Step 1: Download the Algorand Pera app

iOS



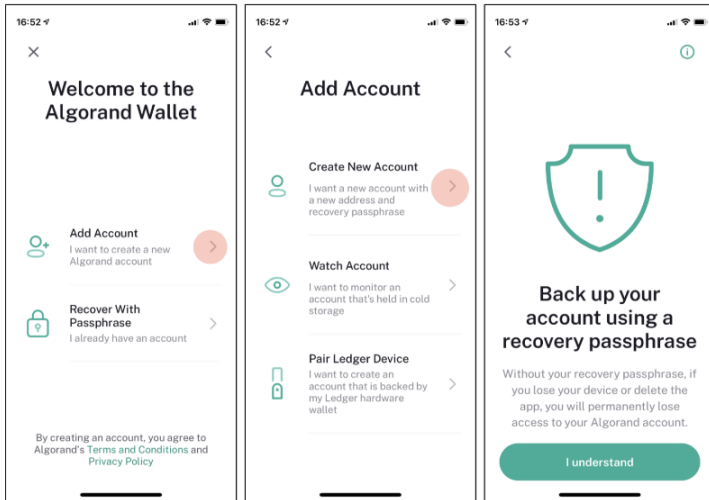
Android



- Scan the QR code with your phone camera

Crypto Hands-On with Algorand

Step 2: Create an account



- Store the passphrase = a secret sequence of words that can be hashed to your private and public keys

Crypto Hands-On with Algorand

Disclaimer

- Algorand has a TESTNET mode created for students/academics/developers
- We will use this mode → no real money
- DO NOT BUY REAL \$ALGO FOR THIS COURSE !

Crypto Hands-On with Algorand

Step 3: Set the TESTNET mode and get 5 Algo

1. Go to settings / Developer Settings / Node Settings / TestNet
2. Go to settings / Developer Settings / Algorand Dispenser = Get your 5 \$ALGO

This is your first transaction!

Crypto Hands-On with Algorand

Step 4: Go to algoexplorer.io to check your first transaction

AlgoExplorer

Algorand Blockchain Explorer

7FMCIAMOGUT6T3RQN

🔍

TESTNET

🌙

👤

New

NFTs

Assets

Apps

Statistics

Blockchain

Tools

API

Governance

Transaction Overview

Transaction ID	7FMCIAMOGUT6T3RQNIQXRQ2SPHRR4XUIHUIMRDPGJ2U3HZFUFBZA	Timestamp	Fri, 25 Nov 2022 16:51:56 GMT
Block	25805766	Type	Transfer

Transaction Details

Sender:	HZ57J3K46JJXILONBBZOHX68KPXEM2VWXNRFUED6DKFD5ZD24PMJ3MVA	Copy
Amount:	⬆ 5	
Receiver:	N4K2ZSB74MCCUEBNI4YSMU2YDJ75QYNM2EWZMO327XZODP5BXLPPYPZB2BA	Copy

Note

Message Pack

Text

Base 64

Hex

"dispenser v2"

Crypto Hands-On with Algorand

Step 5: Create an account on [Purestake.com](https://purestake.com)

Allows you to:

1. Read Blockchain data (via Purestake API)
2. Submit transactions (via Purestake API)

Retrieve and store "Your API key", a secret password that allows you to connect to the Blockchain

Crypto Hands-On with Algorand

Step 1 to 5 summary

- You've created a test account with the Pura app
 - You've a public key (the name of your wallet) and a mnemonic (or passphrase)
 - You've a private key that you'll be able to retrieve with Python functions
 - You've credited your test account with 5 \$ALGO (*fake* money)
- You've created a developer access on Purestake.com
- You're ready to start playing with the Blockchain writing Python script!

Crypto Hands-On with Algorand

Step 6: Follow the steps in the Notebooks

- Notebooks are available on BBoard, please keep the architecture of the folders
 - You'll start to create other wallets (called Alice, Bob, etc.) in Python to play with them
 - You'll also associate your Pera wallet to an account called [MyAlgo](#)
- Then, you'll be able to make transactions and create tokens (ASA)

Crypto Hands-On with Algorand

Step 7: I want your money

- **Homework:** add my wallet to your `credentials` file (call this account `Clement`)
- Make me a 1\$ALGO payment

My public key is :

N4K2ZSB74MCCUEBNI4YSMU2YDJ75QYNM2EWZMO327XZODP5BXLPPYPZB2BA

- You should add this to your `credentials` file

```
},
"Clement": {
  "public": "N4K2ZSB74MCCUEBNI4YSMU2YDJ75QYNM2EWZMO327XZODP5BXLPPYPZB2BA",
  "private": "",
  "mnemonic": ""
}
```



5' break

Three months of Finance with Big Data

- Asset Pricing, Banking and VC/PE, and foundations of Blockchain
- Learnt about 18 Nobel Prize winners in Economics (+ 1 Turing Prize tomorrow)
- 8 PC Labs (including a Hackathon): >> 20.000 lines of code

Finance with Big Data with Nobel Prize Winners



Finance with Big Data with Nobel Prize Winners



Richard Feynman (1965), Milton Friedman (1976), Harry Markowitz and William Sharpe (1990), Robert Lucas (1995), George Akerlof, Michael Spence and Joseph Stiglitz (2001), Diamond, Mortensen, and Pissarides (2010), Eugene Fama (2013), Ben Bernanke, Douglas Diamond and Philip Dybvig (2022), Acemoglu, Johnson and Robinson (2024)

Women in Economics and Finance



A. Schoar (MIT), G. Gopinath (Harvard and IMF), E. Duflo (MIT, 2021), T. Whited (U. Michigan), Anne Case (Princeton), S. Stancheva (Harvard), E. Nakamura (Berkeley), K. Jin (LSE), S. Athey (Stanford), Elinor Ostrom (UCLA, 2009), M. Bertrand (Chicago), M. Farboodi (MIT), P. Sapienza (Northwestern), E. Carletti (Bocconi), Goldin (Harvard, 2023)

Thank you!

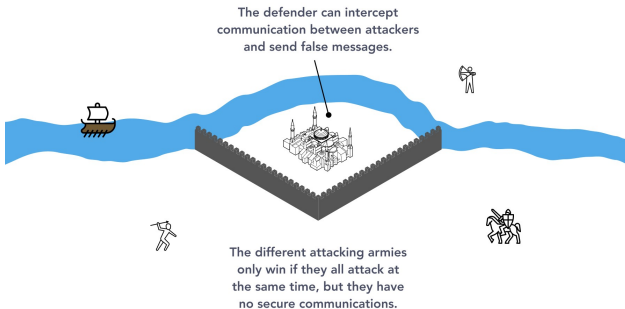
Appendix

Beyond Bitcoin

- Real innovation of Bitcoin: the idea of **Proof of Work** $\Rightarrow$ implementation of a *consensus algorithm*
- Famous game theory idea known as the **Byzantine General's problem** (1982):

The Byzantine Generals Problem

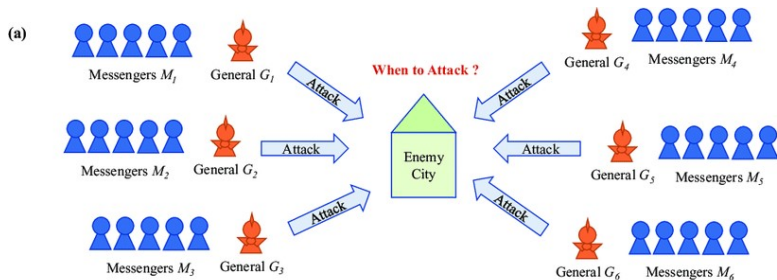
A game theory problem: How do decentralized parties arrive at consensus without a trusted central party?



Beyond Bitcoin

- Real innovation of Bitcoin: the idea of **Proof of Work** $\Rightarrow$ implementation of a *consensus algorithm*
- Famous game theory idea known as the **Byzantine General's problem (1982)**:
 - A set of Generals, and their armies surrounding a palace they want to capture
 - They must choose the exact time of the attack to maximize chances of success
 - Success only achieved if at least 50% of them attack at the same time
 - Certain generals may be traitors $\rightarrow$ misleading signals about the timing
 - They can only communicate with the other generals via their messenger
 - They have no way to check the authenticity of the message that they receive from the messenger

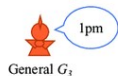
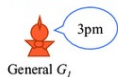
Bitcoin's solution



Source : [Kao et al., 2019](#)

Bitcoin's solution

(b) When to Attack ?



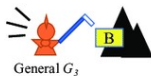
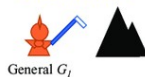
Time T_0

Start Working



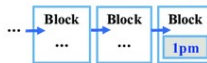
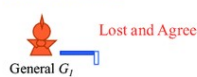
Time T_1

Work Complete



Time T_2

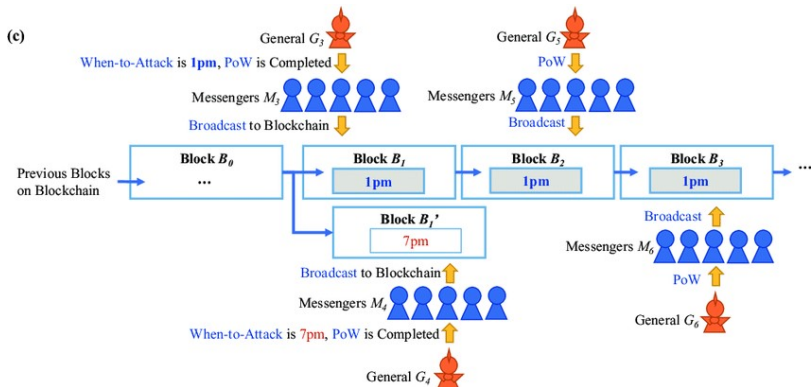
Add to Blockchain



Time T_3

Source : Kao et al., 2019

Bitcoin's solution



Source : Kao et al., 2019