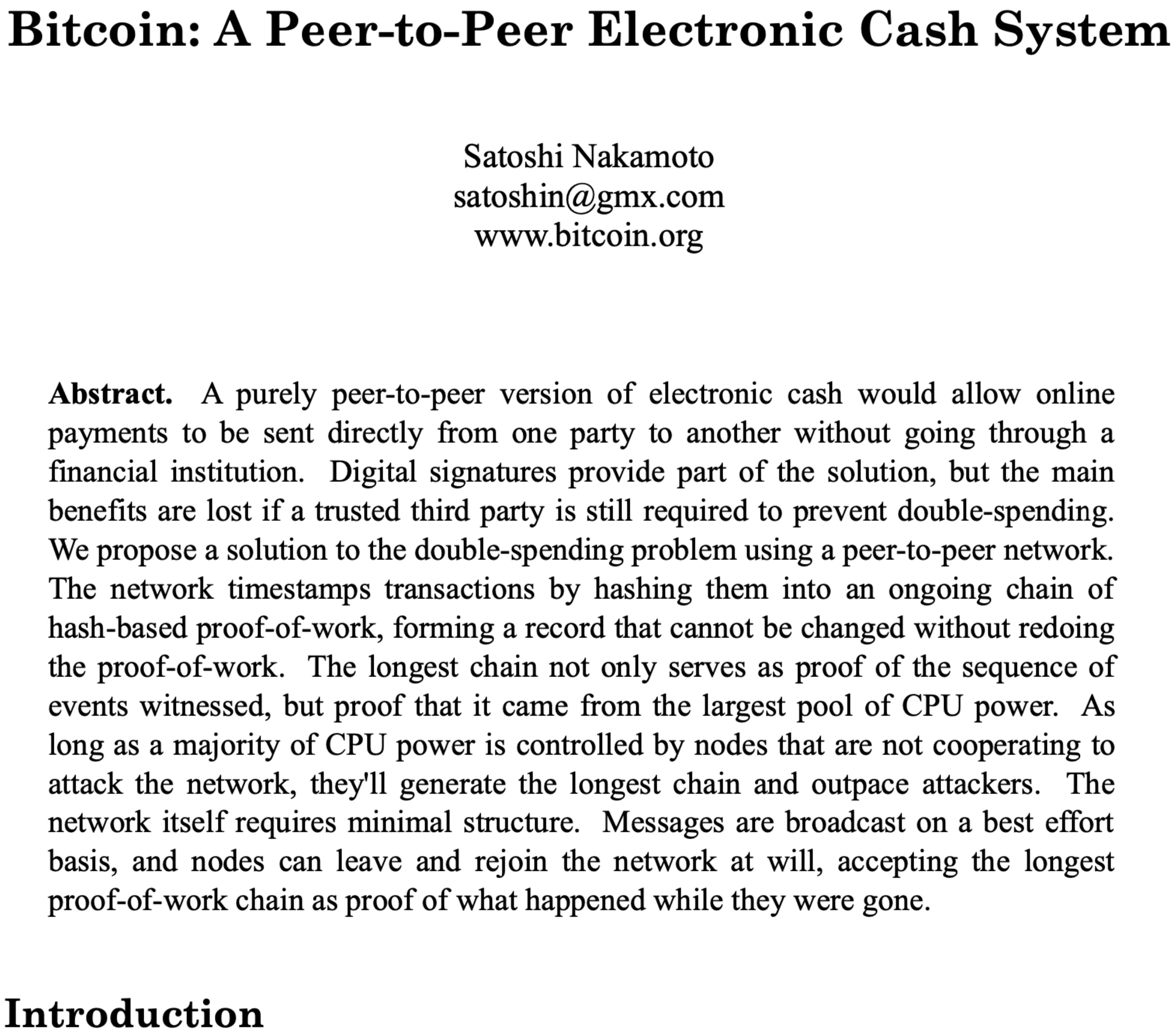
Bitcoin

Part 2 – Consensus Mechanism

Prof. Marco Comuzzi
Department of Industrial Engineering
Ulsan National Institute of Science and Technology (UNIST)
mcomuzzi@unist.ac.kr

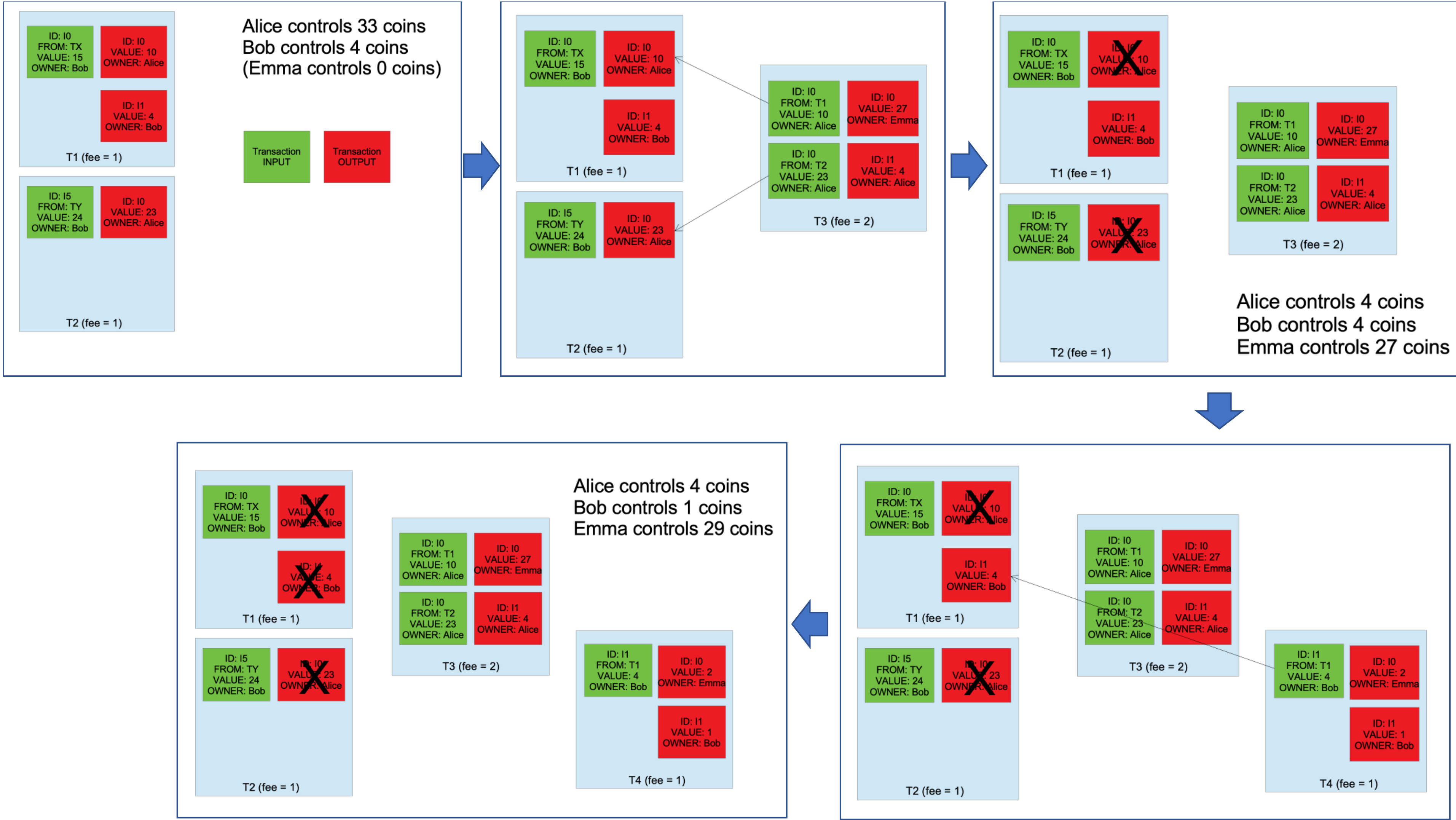


Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Introduction

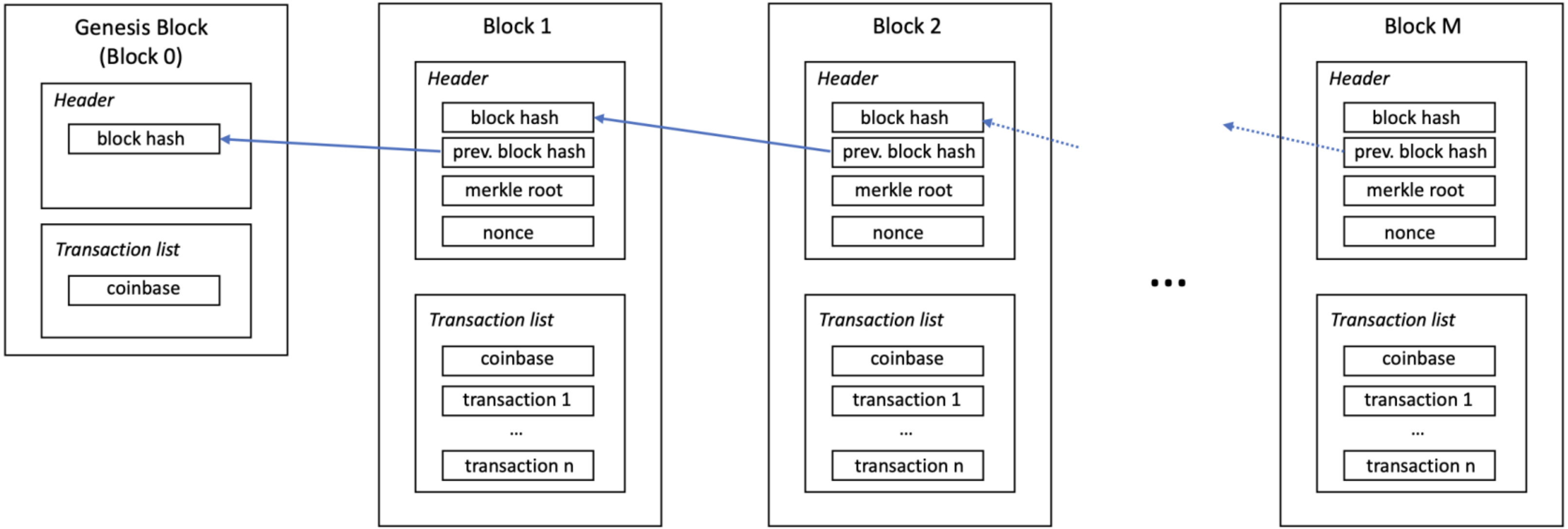


Alice controls 33 coins
Bob controls 4 coins
(Emma controls 0 coins)

Alice controls 4 coins
Bob controls 4 coins
Emma controls 27 coins

Alice controls 4 coins
Bob controls 1 coins
Emma controls 29 coins

The Bitcoin Ledger



Header: metadata about a block, including hashes

Transaction list, including the “coinbase” transaction

Consensus in Bitcoin

The nodes of the Bitcoin network must agree on the content of the ledger

Nodes must agree on the order of the blocks
(the order of transactions within a block is fixed)

Without consensus, double spending will be possible

Consensus in Bitcoin

1. Lifecycle of transactions in Bitcoin
2. Assembly of new blocks
3. Mining of new blocks

1.
Lifecycle of Bitcoin transactions

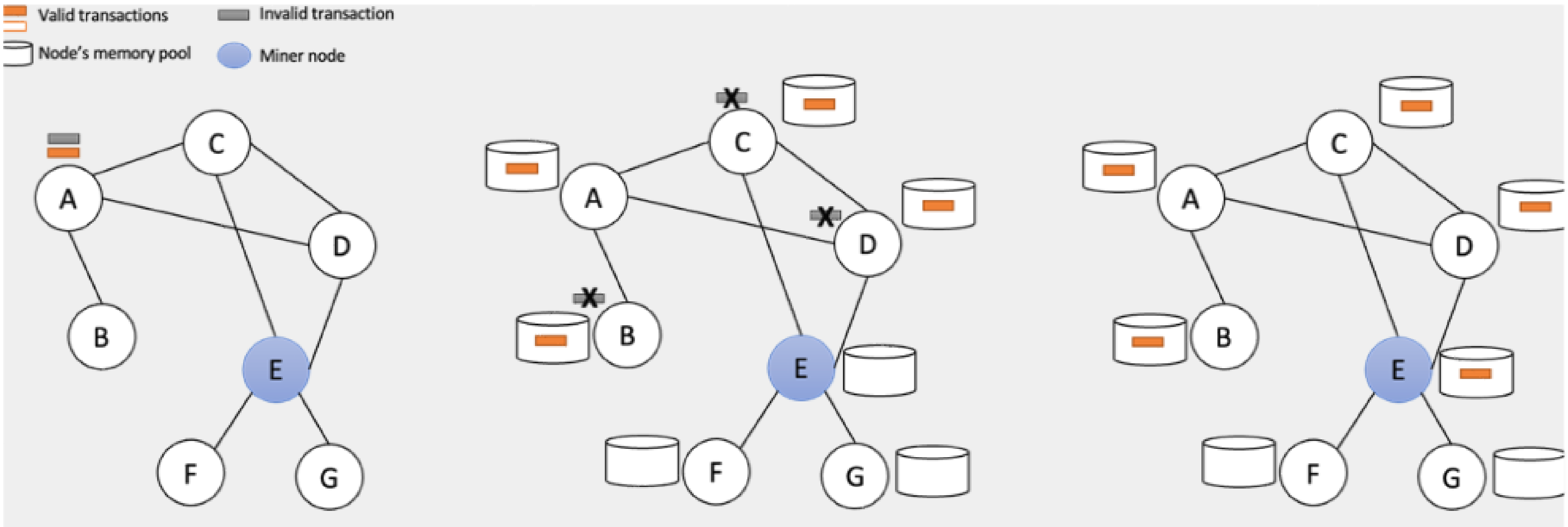
Lifecycle of a transaction, P1

A transaction T is first created by a node and sent to its peers.

Every peer **validates** the transaction T and, if it passed the validation:
(i) **forwards** T to its peers and
(ii) **stores** the validated transaction T in a local transaction repository, called the '**memory pool**'.

This mechanism is **recursive** across the nodes and defines the **gossiping** protocol in Bitcoin. Through this mechanism, a valid transaction T reaches all the nodes in the Bitcoin network.

Lifecycle of a transaction, P1



1) Node A issues two transactions and forwards them to peers C, D, B.

2) A's peers validate the transactions, store the valid one in their memory pool and forward it to peers; they discard the invalid one.

3) C, D forward the valid transaction to its peer E. E validates it, stores in its memory pool (and will forward it to its peer F and G).

Transaction validation

To validate a transaction T, a node verifies that:

1) T is well-formed
E.g., list of inputs is non-empty, every output has an unlocking script

2) The sum (in BTC) of the inputs is greater than the sum of the outputs

3) Every transaction input matches a UTXOs of a previous transaction

Lifecycle of a transaction, P2 (in a block)

At some point, a node running the **“mining”** functionality of the Bitcoin protocol removes T from its memory pool and includes it in a ‘candidate’ block.

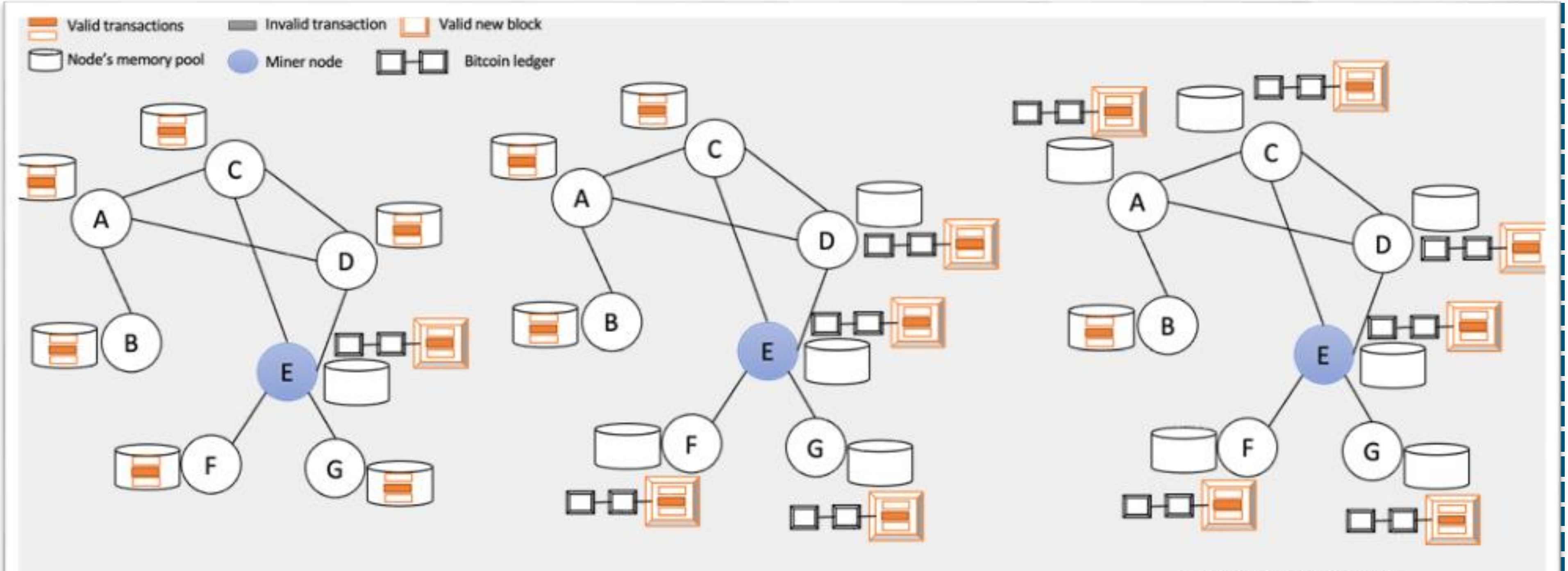
The mining node runs the **Proof-of-Work** (PoW) mechanism on the candidate block. If this is successful, then the node has succeeded in creating and new candidate block. The node can now forward the newly mined candidate block to its peers

Every peer **validates** the candidate block. If the candidate block is valid, then a node:

(i) **removes** all the transactions in it (including T) from the memory pool and

(ii) **forwards** the valid **block** to its peers. In this way, T reaches all the nodes of the Bitcoin network. Every node is now able to calculate the balance of all the nodes in the network by considering also the transfer of Bitcoins specified by T.

Lifecycle of a transaction, P2 (in a block)



4) E creates a new candidate block using the transactions in its memory pool. E runs the PoW and, if successful, the new block is forwarded to E's peers.

5) F, G, and D validate the new block, remove the transactions in from the memory pool, and store the new block in their ledger

6) D forwards the new block to its peers A and C. They validate the block, remove the transaction in it from their memory pool, store it in their ledger (and forward it to their peers)

2.
Assembly of new blocks

Miner nodes

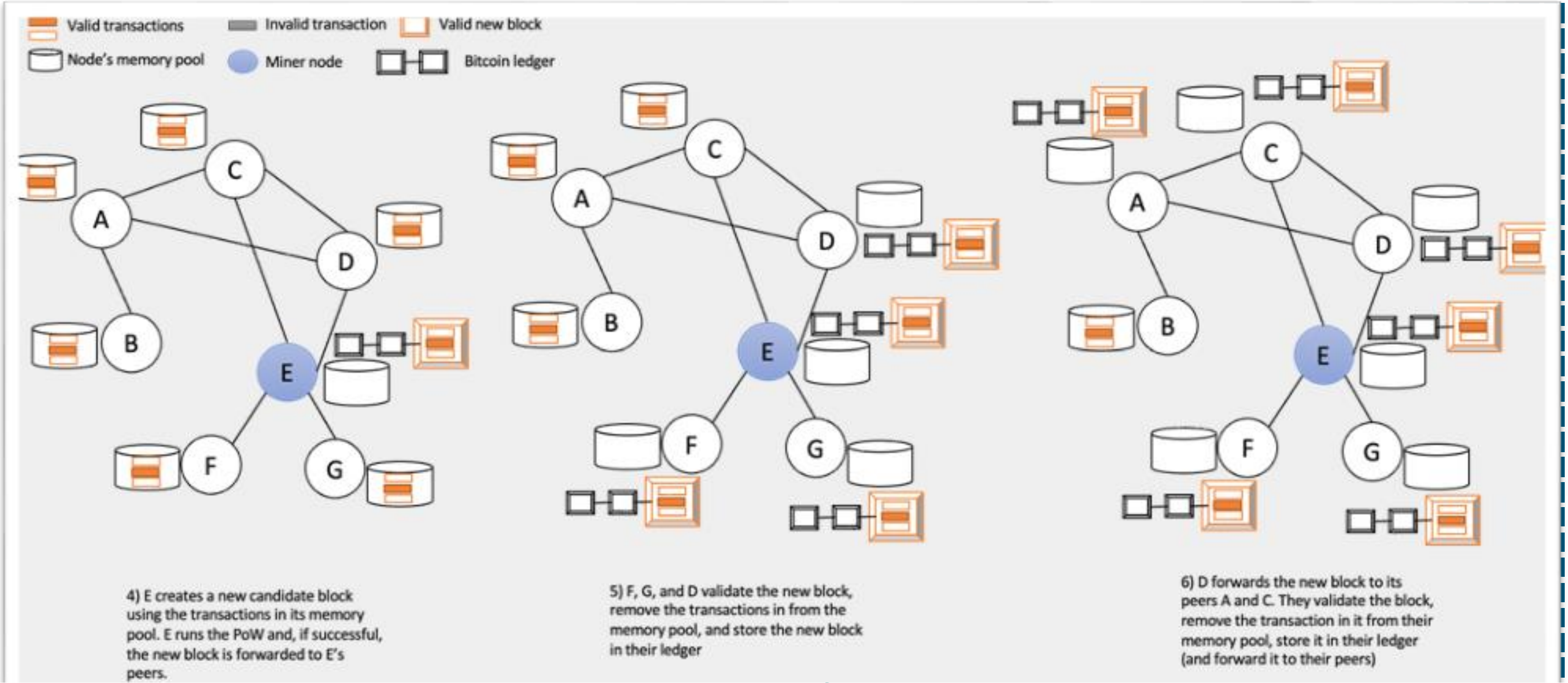
Miner nodes are the ones that assemble new blocks

The mining functionality is part of the Bitcoin protocol, so every node **could** mine a new block

In practice (see later), it takes a lot of computational power to be successful at mining, so only a limited number of nodes are "competitive" at mining

(there are no special nodes in Bitcoin, still every node can be a miner)

Lifecycle of a transaction, P2 (in a block)



4) E creates a new candidate block using the transactions in its memory pool. E runs the PoW and, if successful, the new block is forwarded to E's peers.

5) F, G, and D validate the new block, remove the transactions in from the memory pool, and store the new block in their ledger

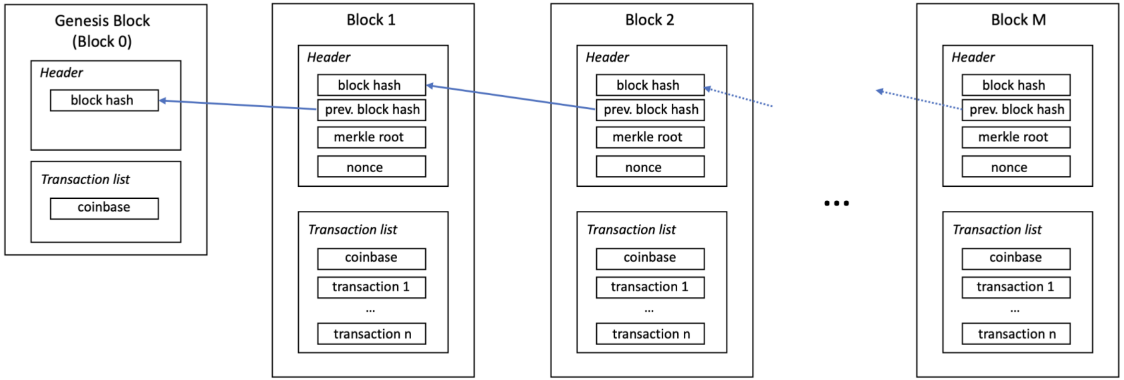
6) D forwards the new block to its peers A and C. They validate the block, remove the transaction in it from their memory pool, store it in their ledger (and forward it to their peers)

Assembling a new block

1. Which transactions are included in the block?

2. Which is the 'nonce' of the block?

3. Where is the block included in the ledger, i.e., which should be its previous block?



Choosing transactions for a block

Only transactions in the memory pool can go in a block

Miners prefer transactions that pay higher fees

Higher fee = higher reward for a miner

Miners prefer transactions with earlier timestamps

Increases the "social welfare" (the chance that new transactions can be validated)

Calculate the nonce for the new block

The nonce is calculated executing the PoW algorithm (see later)

PoW is computationally (very) expensive and is the only possible to calculate a correct nonce

So, a correct nonce of a block proves that the miner has done the computational work required for assembling the block correctly

Incentives for block assembly

PoW costs money (electricity for running the computation)

Miners have no incentive in assembling an invalid block (they will simply waste a lot of money)

Coinbase transaction (fixed reward + transaction fees) creates the incentive for the miner to assemble a new block

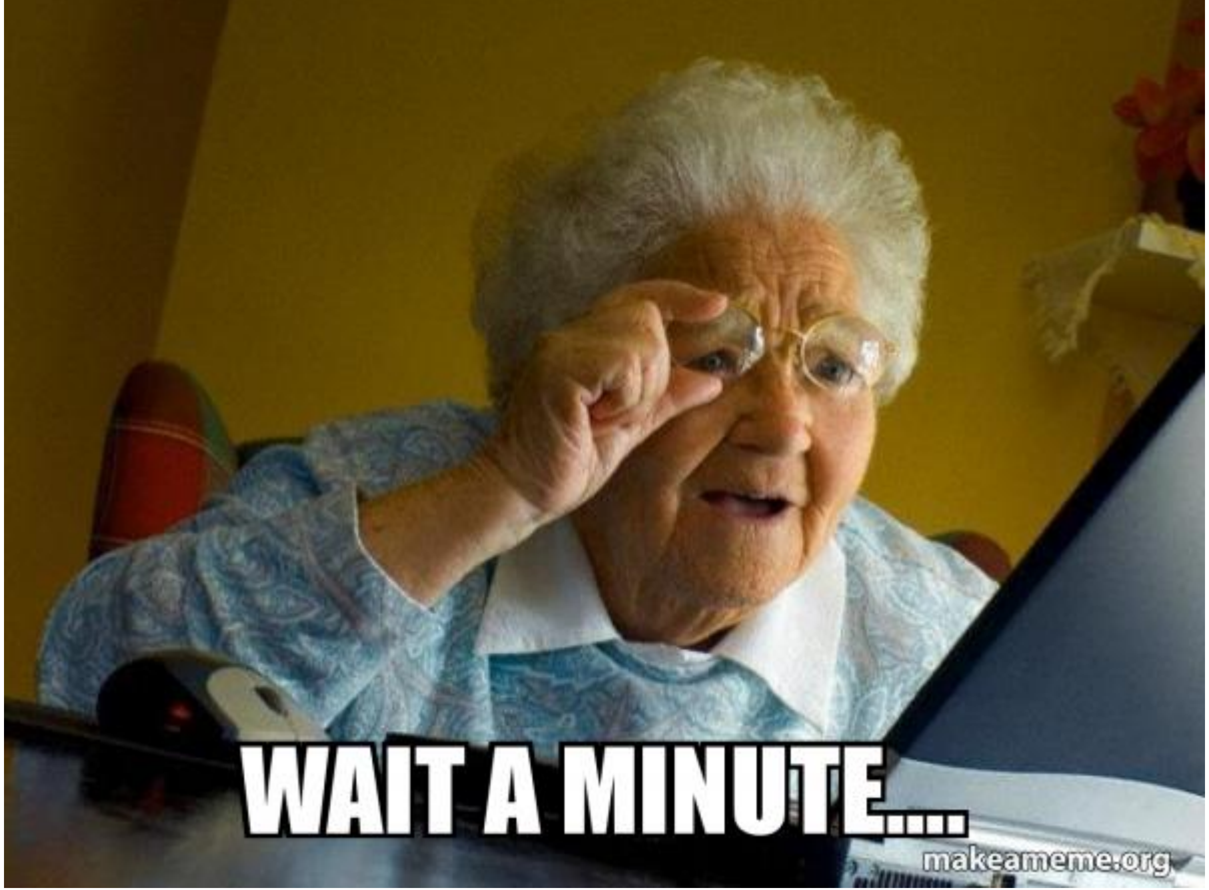
Where to put the new block?

The longest chain always wins!

General rule:

Miners must assembly the new block on top of the highest block in their local copy of the ledger

Highest block = extending the chain characterized by the most computational power to be created (more blocks = more PoW)



Why should there be more than one possible chain to extend?

Validation of new blocks

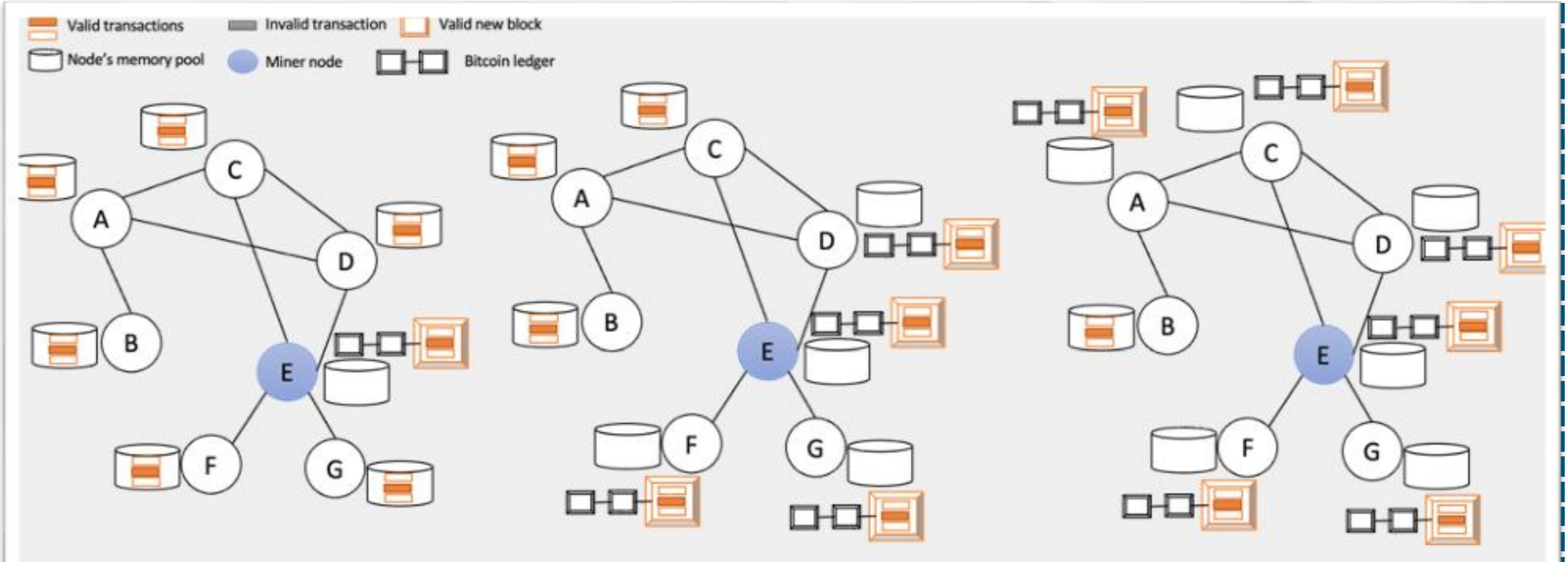
When validating a new block a node checks that:

The block is well-formed. For instance, it verifies that the list of transactions is not empty and that it includes a coinbase transaction.

The information in the block header is correct: For instance, the hashes (of the block and of the previous block) must be correct and, most importantly, the nonce of the block must be correct (see later)

The content of the coinbase transaction is valid, i.e., the output generated by it matches the sum of the fixed block reward and the fees paid by all the transactions in the block.

Lifecycle of a transaction, P2 (in a block)



4) E creates a new candidate block using the transactions in its memory pool. E runs the PoW and, if successful, the new block is forwarded to E's peers.

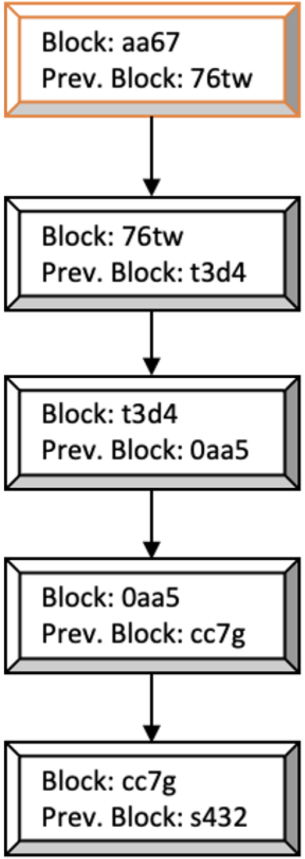
5) F, G, and D validate the new block, remove the transactions in from the memory pool, and store the new block in their ledger

6) D forwards the new block to its peers A and C. They validate the block, remove the transaction in it from their memory pool, store it in their ledger (and forward it to their peers)

Where does a new block fit?

A miner node has assembled a new node from the top of the chain in its ledger

A new block may extend the chain in the ledger from the top… or it may not!



Case 1

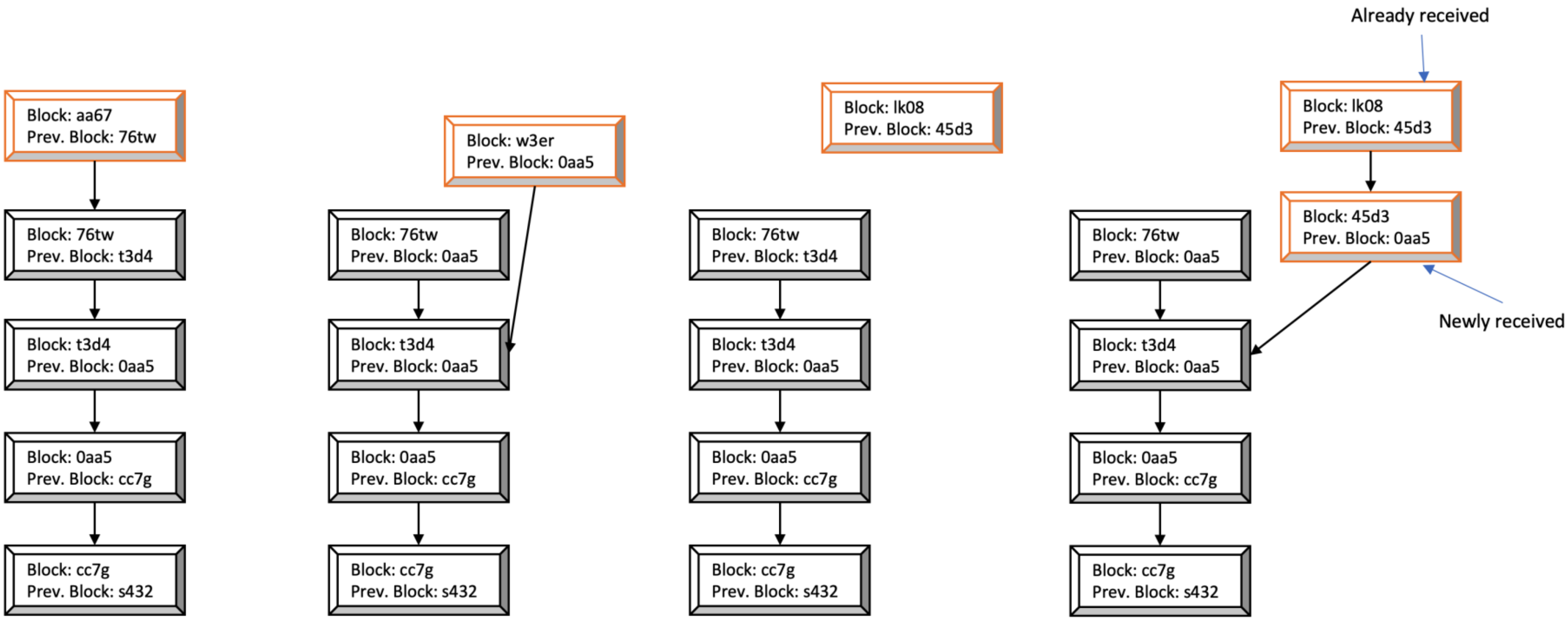
Other cases

“Normal”

“Fork”

“Orphan”

Fork, including already received orphans



Block: aa67
Prev. Block: 76tw

Block: 76tw
Prev. Block: t3d4

Block: t3d4
Prev. Block: 0aa5

Block: 0aa5
Prev. Block: cc7g

Block: cc7g
Prev. Block: s432

Case 1

Block: w3er
Prev. Block: 0aa5

Block: 76tw
Prev. Block: 0aa5

Block: t3d4
Prev. Block: 0aa5

Block: 0aa5
Prev. Block: cc7g

Block: cc7g
Prev. Block: s432

Case 2

Block: lk08
Prev. Block: 45d3

Block: 76tw
Prev. Block: t3d4

Block: t3d4
Prev. Block: 0aa5

Block: 0aa5
Prev. Block: cc7g

Block: cc7g
Prev. Block: s432

Case 3

Already received

Block: lk08
Prev. Block: 45d3

Block: 45d3
Prev. Block: 0aa5

Newly received

Block: 76tw
Prev. Block: 0aa5

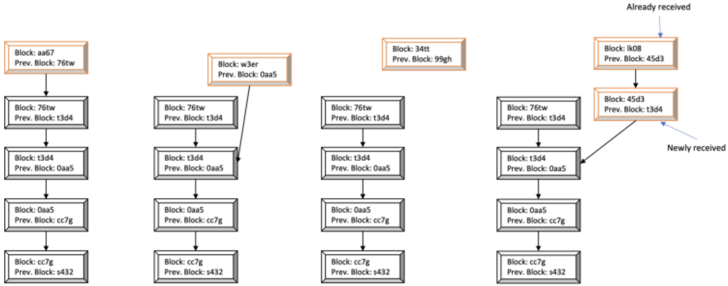
Block: t3d4
Prev. Block: 0aa5

Block: 0aa5
Prev. Block: cc7g

Block: cc7g
Prev. Block: s432

Case 4

When to “trust” a block?



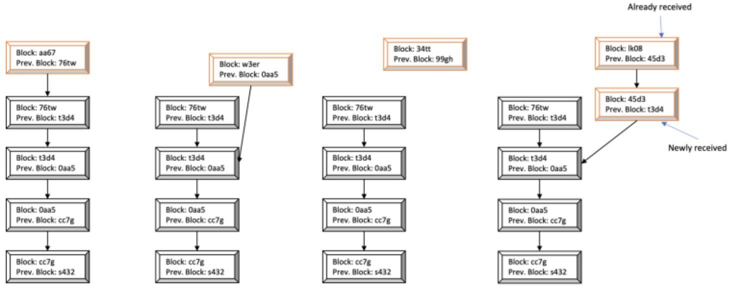
The ledger stored by a node may have orphans and forks

Forks are limited around the last blocks received

Practically, the chance that a new block forks more than 5-block deep in the longest chain in the ledger is zero

Nodes should ”trust” only the transactions in blocks that are 6-block deep in their longest chain

When to “trust” a block?

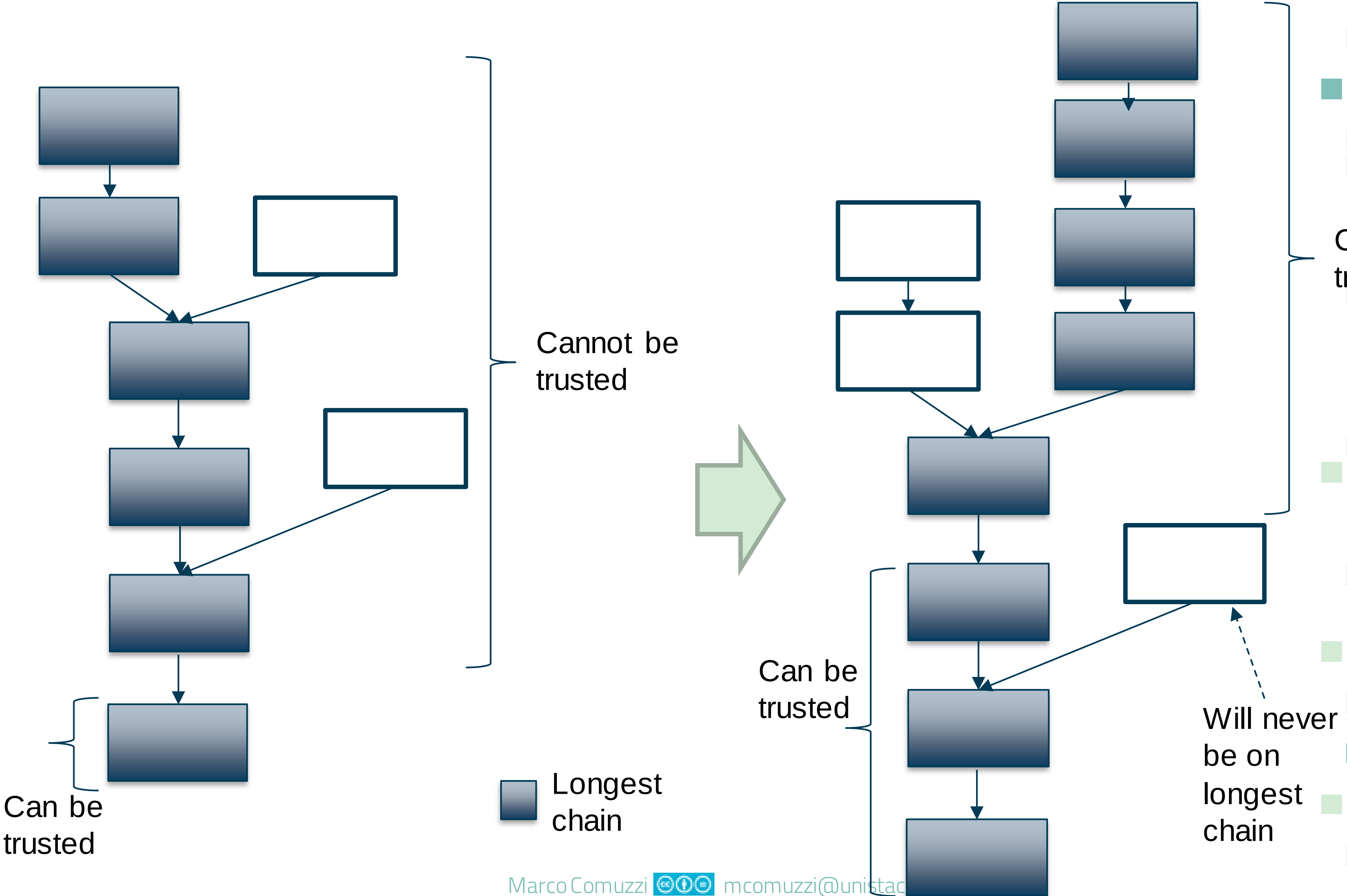


The Bitcoin protocol is designed to have one new block assembled by a miner every 10 minutes (see later how)

A 6-block deep block was mined on average 1 hour before

Nodes should not trust transactions in blocks received in the last hour

(Bitcoins to buy a coffee?)



Cannot be trusted

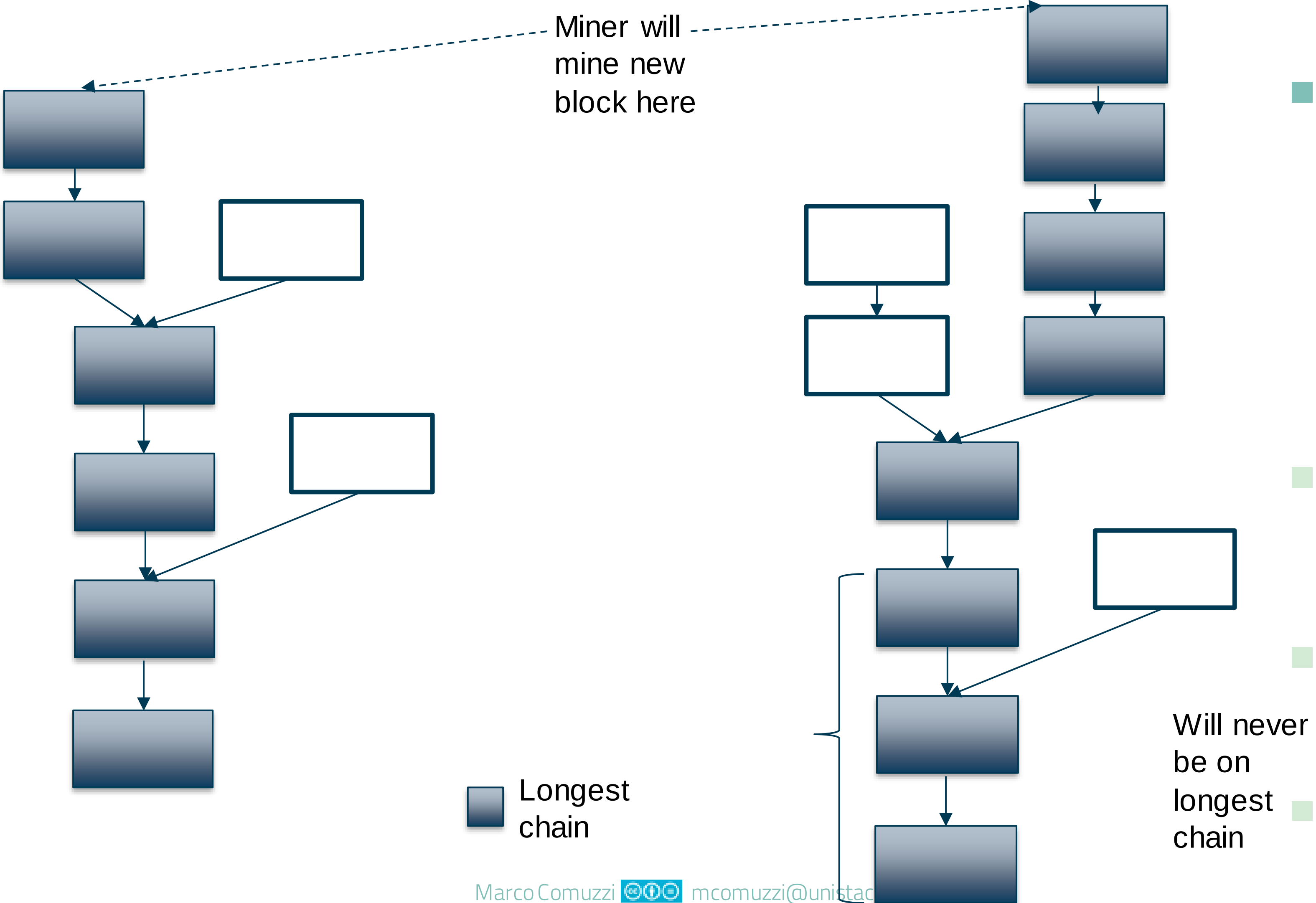
Can be trusted

▨ Longest chain

Cannot be trusted

Can be trusted

Will never be on longest chain



Miner will mine new block here

▨ Longest chain

Will never be on longest chain

“Emergent” consensus in Bitcoin

There is not a precise moment time in which all nodes agree to add a new block to the ledger

The ledgers of the nodes do not even contain exactly the same blocks at any point in time (gossiping)

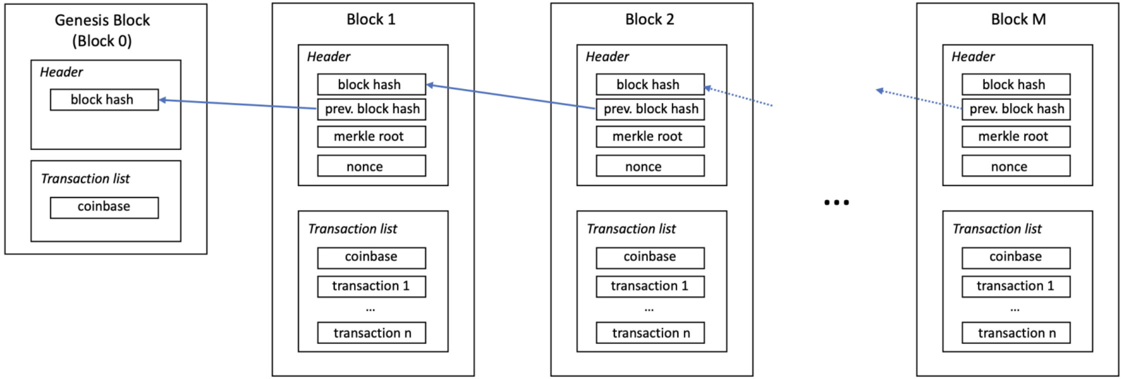
The consensus **emerges** from the interaction of the nodes and the rules of the protocol (= all nodes can be fairly sure that a transaction in a block 6 block-deep in the longest chain will be stay there forever)

3.
Mining of new blocks

Proof-of-Work

What is the computational problem that a miner must solve to calculate a correct nonce for a new block?

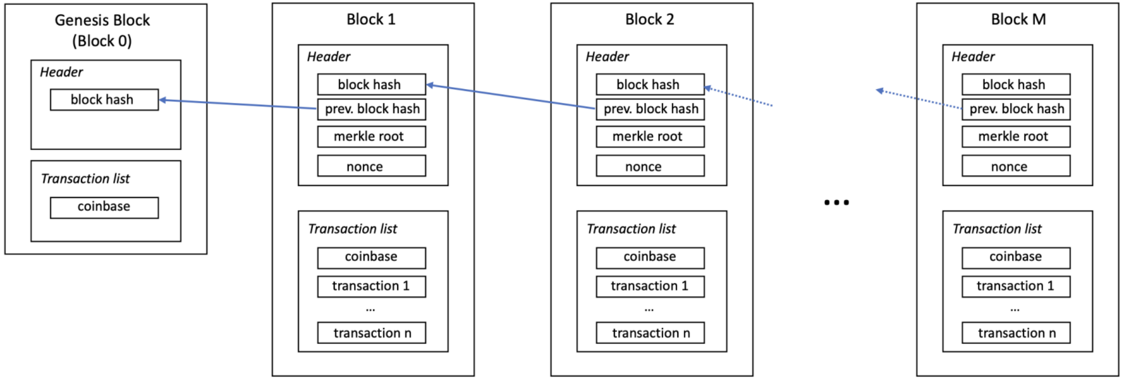
How can any node prove that the miner has done the computational “work” required?



Proof-of-Work

What is the computational problem that a miner must solve to calculate a correct nonce for a new block?

How can any node prove that the miner has done the computational "work" required?



Let's play a game with dice

A player throws two dice simultaneously

A player wins if the sum of the scores obtained on each dice is less than a target S

If S=13, the player has 100% winning chance

If S = 12, the player has 35/36 chances to win (~97%)

If S = 3, the player has 1/36 chances to win (~2%)

Winning chances and player’s “work”

How many times should a player throw the dice to be fairly confident of throwing a winning combination?

If S=12, probably only a few times?

If S=3, probably around 40 times?

The target S determines the amount of “work” (=dice throwing) that a player should do to be “fairly confident” of winning

One last note: winning may require multiple throws, but verifying that a player won takes almost no work at all (just look at the dice!)

Finding a correct nonce

The problem that a miner node must solve to find a correct nonce is the following:

"Find a value of a nonce such that the value of the SHA256 of the hash of the candidate block is less than a given target S"

SHA256 hash is a number represented by 256 bits, so the problem is well-defined

Practically…

If S=5,000,000 in Base-10, then S=4c4b40 in Base-16

So, a miner must a find a nonce such that the hash of the block is less than:

0000000000000000000000000000000000000000000000000000000000 4c4b40

This means that, in order to be lower than S, the hash of a block should start with at least 56 zeros.

Finding a correct nonce: how?

Is there a "smarter" way to solve the problem than "brute force"
(= trying all the possible values of nonces until we found a good one)?

NO

Why?

If there was a way, then it means that we have found a "smart" way to "predict" the value of a hash given its input. This clearly contradicts the properties of a good hashing function.

Higher target S ➔ more "allowed" hashes ➔ easier to calculate nonce

Lower S, higher "mining "difficulty"

The Proof-of-Work, summing up

Miners can only solve the problem using "brute force":
nonce = 0, 1, 2, …. Until they find a correct one

The fact that a nonce is correct is a proof that the miner has done the computational work required (there is no other possible way!)

Finding the nonce is hard (thousands of hashing operations), verifying it is simple (1 hashing operation)

The target S can be adjusted to make the problem more or less hard

(In Bitcoin, S is set to have a new block mined on average every 10 minutes)

Block hashes in history

0000000000000000000d36a105e1f07e697fdabf36bde99b275f2e0427d26d6e
(mined on 2019-03-14 06:05:08, height = 566,975)

00000000000011906b491883ab0f16f0e690b133ca860b199b775c3cf6581c21
(mined on 2011-06-11 09:11:11, height = 130,000)

00000000f093cf2c5865f2bb851d5fa65ff1794b02fea4b0d90ff451e6e5d1c0
(mined on 2009-01-10 20:20:48, height=56)

Proof-of-Work: Consequences

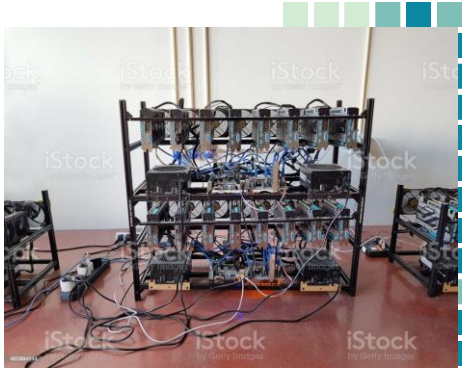
Miners have an incentive to increase their computational power, to be able to mine faster than other nodes

To keep the 10-minute average, the mining difficulty tends to increase over time, while miners become on average increasingly computationally powerful

A miner with 51% of the mining power in the network may be able to take control of it
(= mining blocks on average faster than other miners)

History of Bitcoin mining

2009: Bitcoin mining was a hobby for crypto-geeks, doable with a good desktop pc



2010: first implementation of PoW on GPU released, mining requiring “GPU farms”

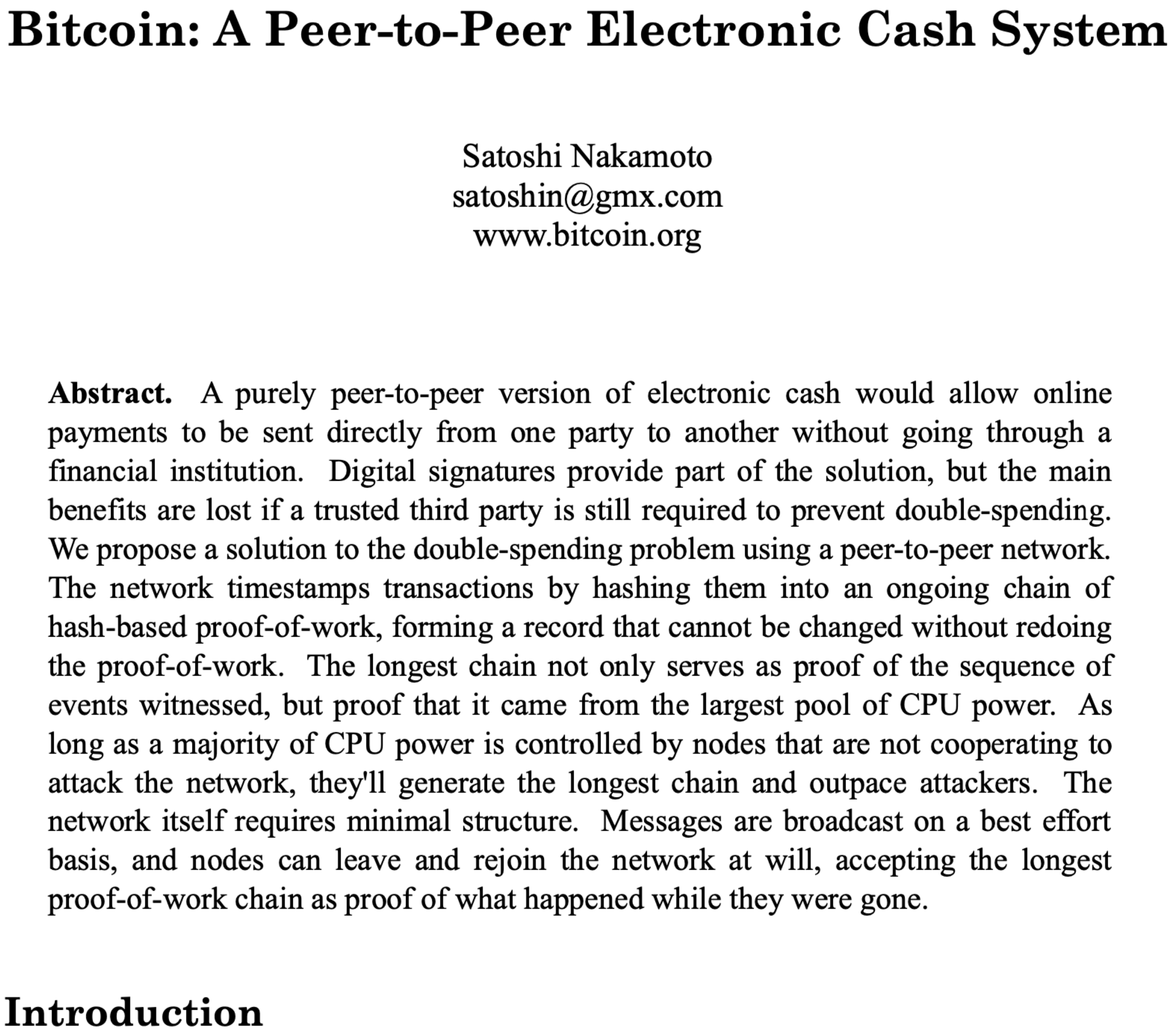
2011: First ASIC (Application-Specific Integrate Circuit) for PoW appeared

Mining consortium

A set of nodes, possibly with low-end mining power, join forces

The range of possible nonces is split across the nodes

The rewards of new blocks (coinbase) are split across nodes owners according to the power contributed to the consortium

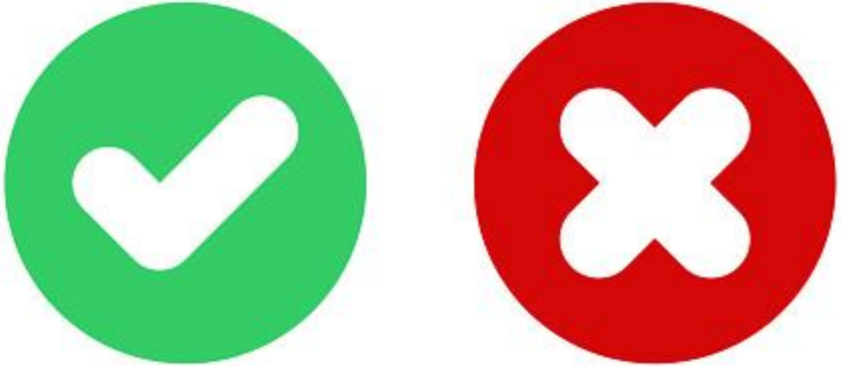


Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Introduction



Bitcoin: the good stuff

- A fully digital cash system without intermediaries
  - Only users, no banks, no governments
- Resilient
  - As long as one node in still alive, Bitcoin will be alive
- Could be the future of cash?

Bitcoin: the bad stuff

- Slow
  - It takes about 1 hour for one transaction to be confirmed
  - OK to buy a house, but not a coffee
- Anonymous and unregulated
  - Frequently used in the "dark Web" for illegal dealings
- Value in USD extremely volatile
  - Be careful when investing!

The Bitcoin experiment in El Salvador

Bitcoin made “legal tender” in 2021

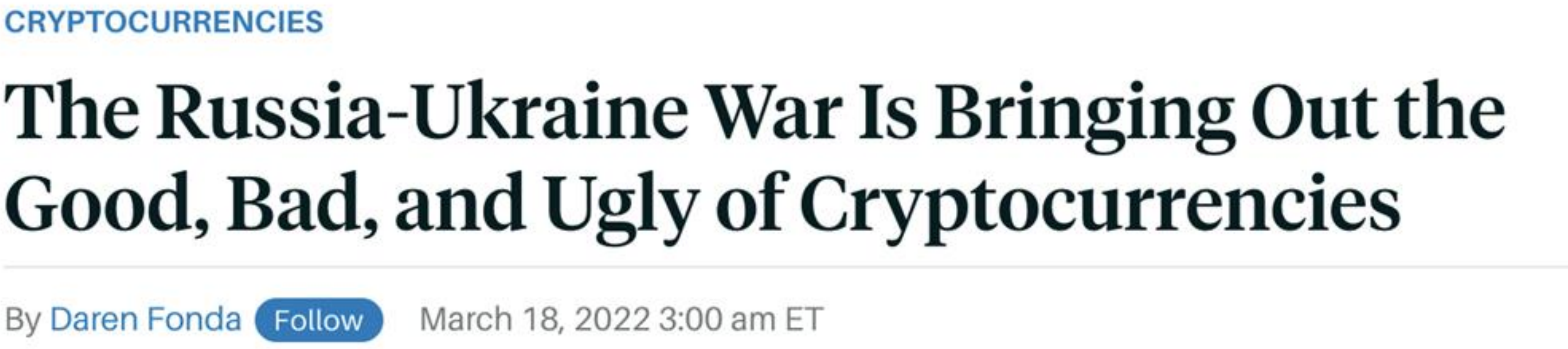
Improve transition to digital cash

Reduce dependence from US dollars

Reduce fees on overseas transfers

A lot of technical problems

Volatility of Bitcoin price is an issue



CRYPTOCURRENCIES

The Russia-Ukraine War Is Bringing Out the Good, Bad, and Ugly of Cryptocurrencies

By Daren Fonda March 18, 2022 3:00 am ET

Donations made in Bitcoins can reach Ukraine more easily

Russian state businesses may use Bitcoin to bypass Western sanctions

THANKS!

https://sites.google.com/site/marcocomuzziphd

http://iel.unist.ac.kr/

You can find me at:

@dr_bsad

mcomuzzi@unist.ac.kr