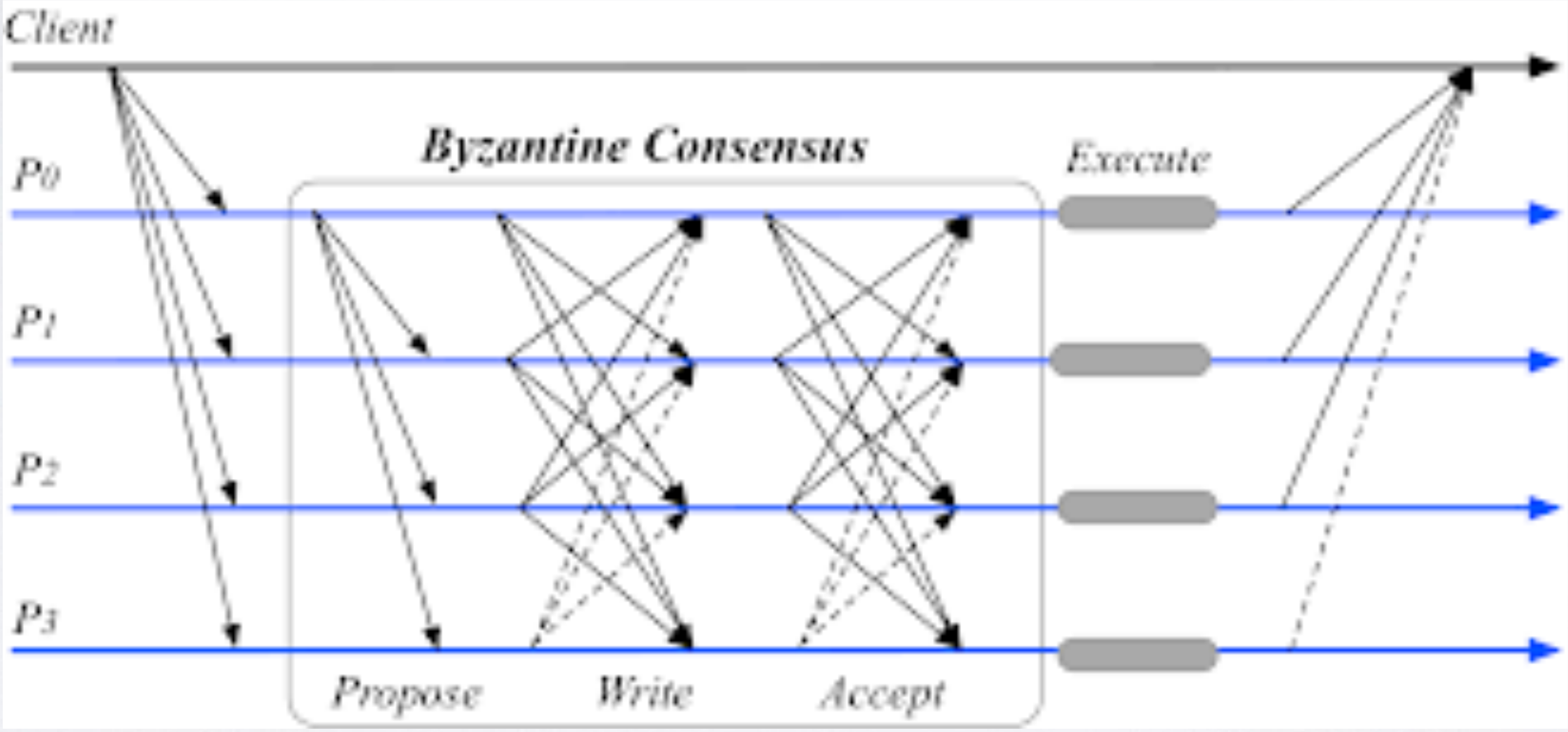
Blockchains & Cryptocurrencies

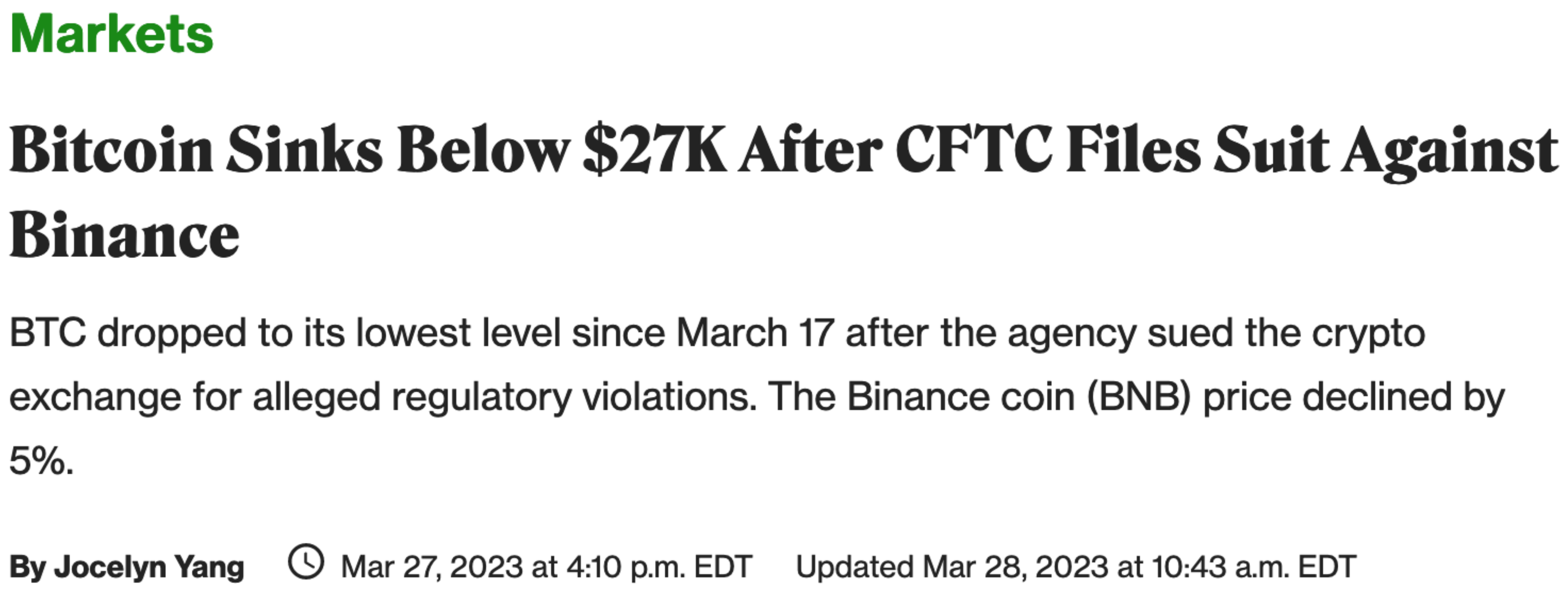
Alternative Consensus Techniques



Instructor: Matthew Green & Abhishek Jain
Spring 2023

Image source: https://eprint.iacr.org/2021/911.pdf

News?



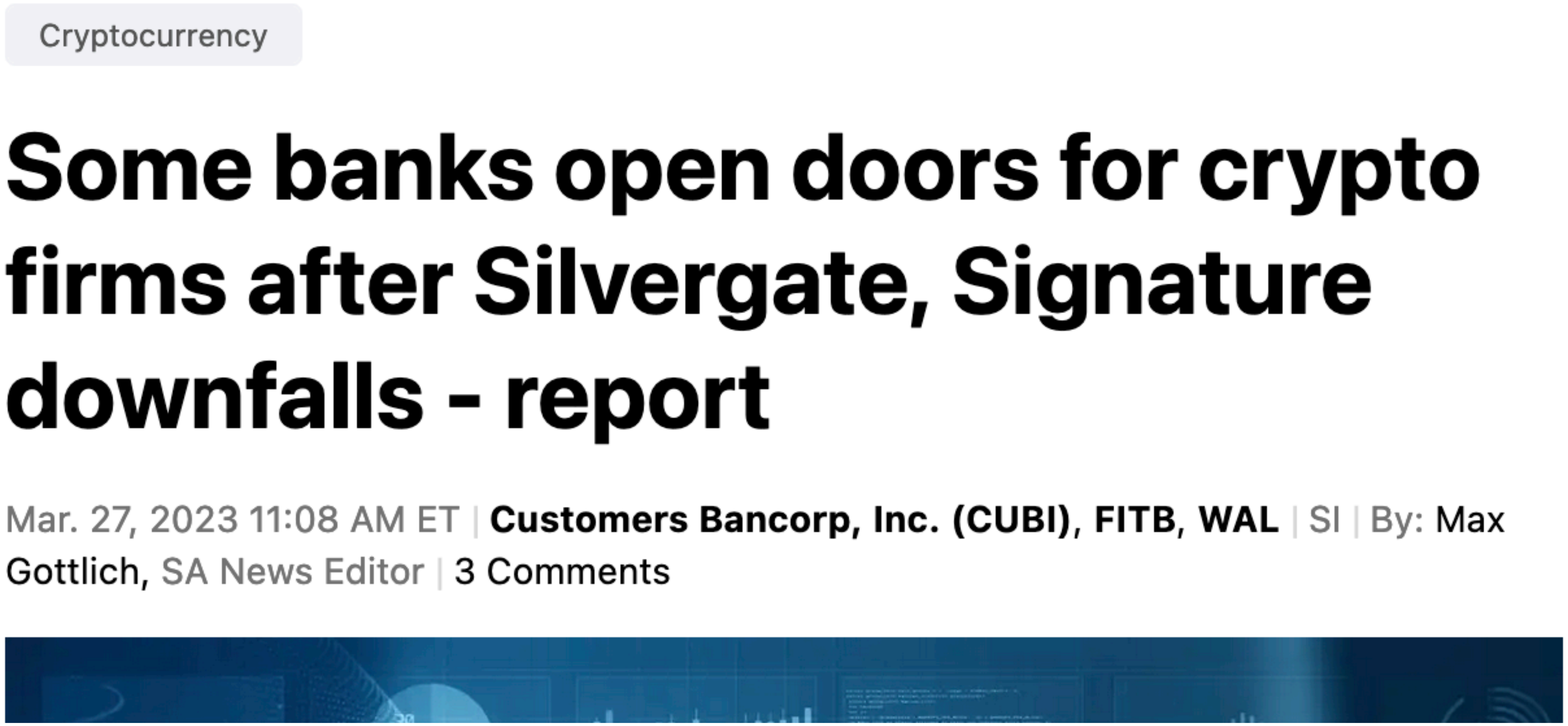
Markets

Bitcoin Sinks Below $27K After CFTC Files Suit Against Binance

BTC dropped to its lowest level since March 17 after the agency sued the crypto exchange for alleged regulatory violations. The Binance coin (BNB) price declined by 5%.

By Jocelyn Yang Mar 27, 2023 at 4:10 p.m. EDT Updated Mar 28, 2023 at 10:43 a.m. EDT

News?

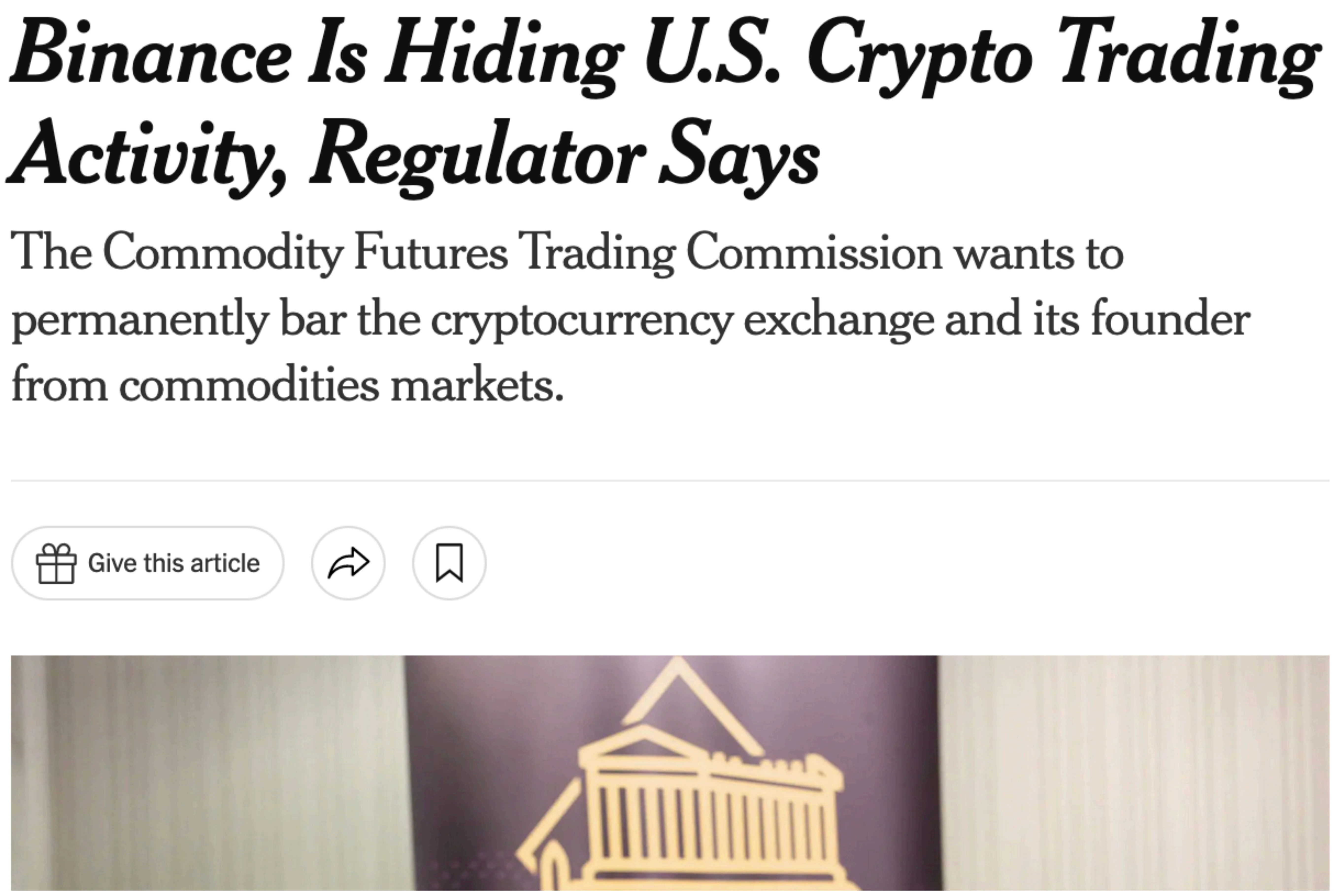


Cryptocurrency

Some banks open doors for crypto firms after Silvergate, Signature downfalls - report

Mar. 27, 2023 11:08 AM ET | **Customers Bancorp, Inc. (CUBI)**, **FITB**, **WAL** | SI | By: Max Gottlich, SA News Editor | 3 Comments

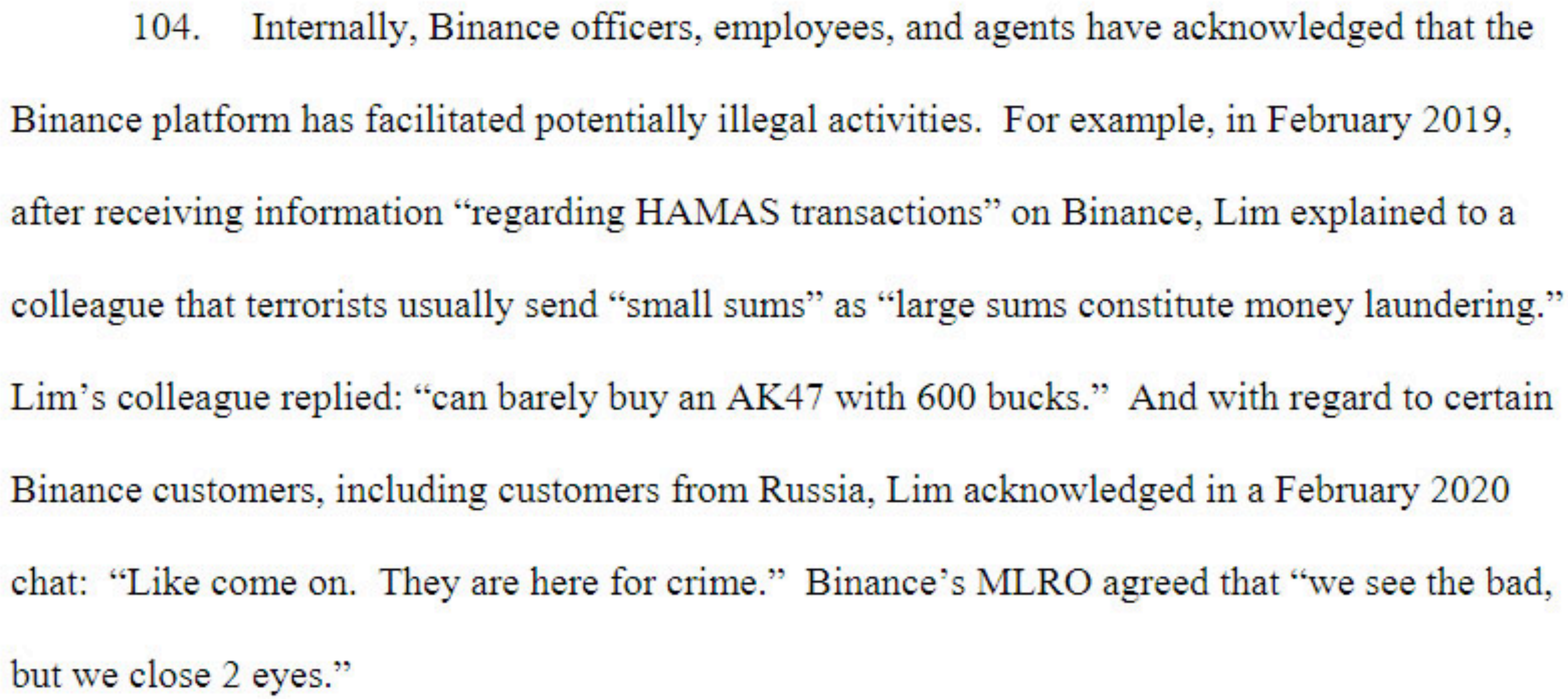
News?



Binance Is Hiding U.S. Crypto Trading Activity, Regulator Says

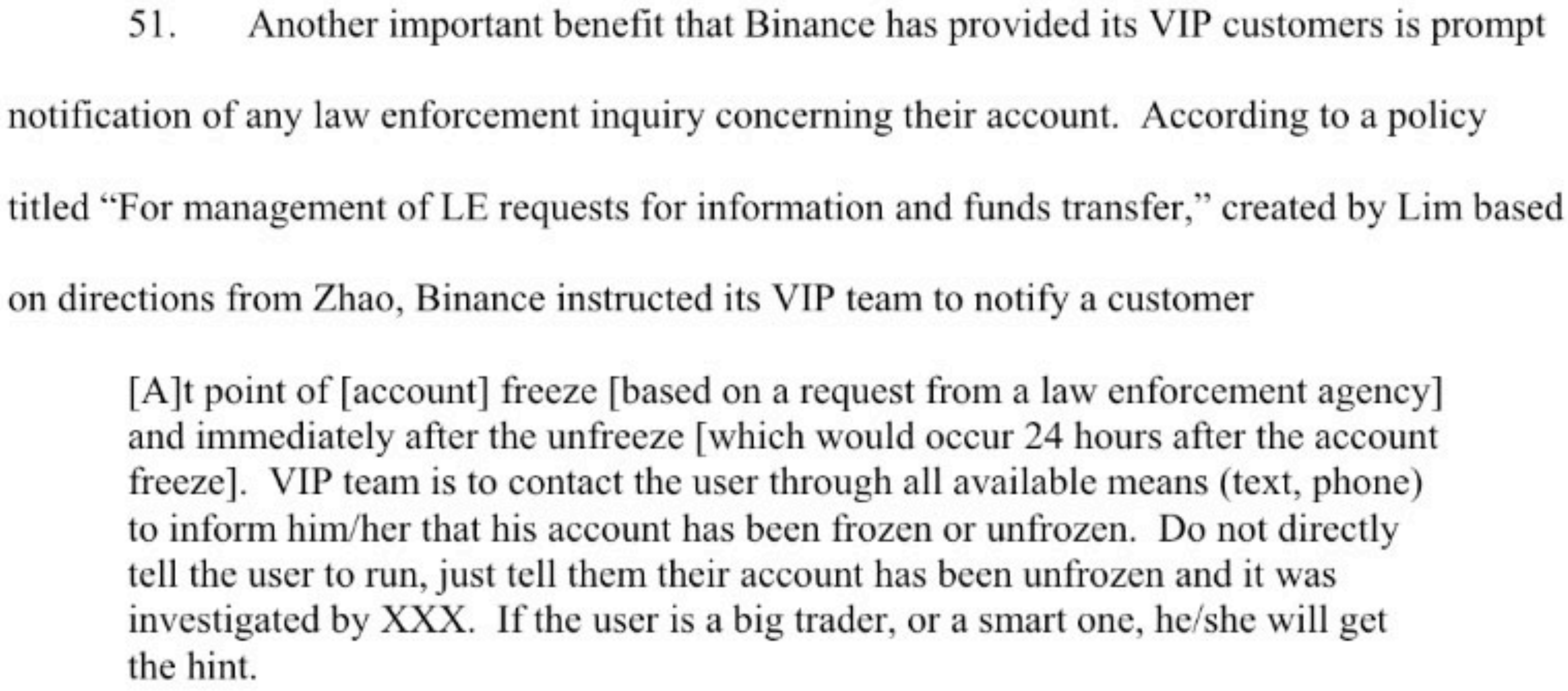
The Commodity Futures Trading Commission wants to permanently bar the cryptocurrency exchange and its founder from commodities markets.

News?



104. Internally, Binance officers, employees, and agents have acknowledged that the Binance platform has facilitated potentially illegal activities. For example, in February 2019, after receiving information "regarding HAMAS transactions" on Binance, Lim explained to a colleague that terrorists usually send "small sums" as "large sums constitute money laundering." Lim's colleague replied: "can barely buy an AK47 with 600 bucks." And with regard to certain Binance customers, including customers from Russia, Lim acknowledged in a February 2020 chat: "Like come on. They are here for crime." Binance's MLRO agreed that "we see the bad, but we close 2 eyes."

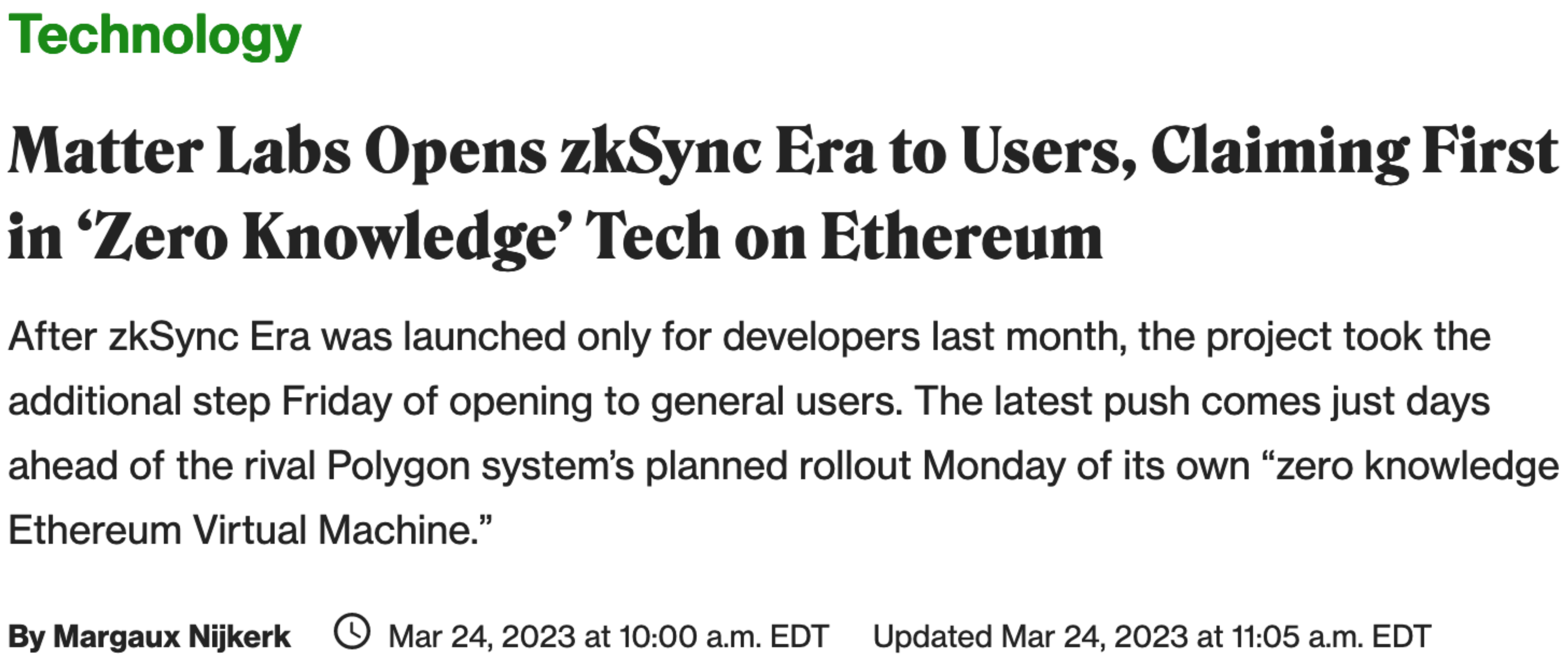
News?



51. Another important benefit that Binance has provided its VIP customers is prompt notification of any law enforcement inquiry concerning their account. According to a policy titled "For management of LE requests for information and funds transfer," created by Lim based on directions from Zhao, Binance instructed its VIP team to notify a customer

> [A]t point of [account] freeze [based on a request from a law enforcement agency] and immediately after the unfreeze [which would occur 24 hours after the account freeze]. VIP team is to contact the user through all available means (text, phone) to inform him/her that his account has been frozen or unfrozen. Do not directly tell the user to run, just tell them their account has been unfrozen and it was investigated by XXX. If the user is a big trader, or a smart one, he/she will get the hint.

News?



Technology

Matter Labs Opens zkSync Era to Users, Claiming First in ‘Zero Knowledge’ Tech on Ethereum

After zkSync Era was launched only for developers last month, the project took the additional step Friday of opening to general users. The latest push comes just days ahead of the rival Polygon system’s planned rollout Monday of its own “zero knowledge Ethereum Virtual Machine.”

By Margaux Nijkerk Mar 24, 2023 at 10:00 a.m. EDT Updated Mar 24, 2023 at 11:05 a.m. EDT

Today

- So far we have mainly talked about Nakamoto consensus
  - Reminder: Nakamoto uses proof-of-work and probabilistic consensus to determine consensus power
  - Today we will talk about alternative consensus techniques:
    - Byzantine Fault Tolerance (BFT) approaches
    - Snowflake-to-avalanche (based on gossip protocols and consensus collapse)
    - Proof-of-stake vs. proof-of-authority vs. proof-of work

Today (and tomorrow)

- Specific examples:
  - BFT protocols (Tendermint, Hotstuff)
  - BFT + sortition (Algorand)
  - Gossip consensus (Avalanche)
  - Weird (Ethereum PoS, we'll get there next lecture)

Let’s define consensus (again)

"Then we are agreed nine to one that we will say our previous vote was unanimous!"

Consensus (definition)

- *English:* Finding an acceptable proposal that all members can support

Consensus (definition)

- *English:* Finding an acceptable proposal that all members can support

- *Computer science fault-tolerant consensus:*

  - Agreement among processes (or agents) on a single data value, where some of the processes may fail or be unreliable in other way. Requirements include:

    - Termination

    - Integrity/validity

    - Agreement

Properties of a consensus protocol

- **Agreement:** All correct processes must agree on the same value.

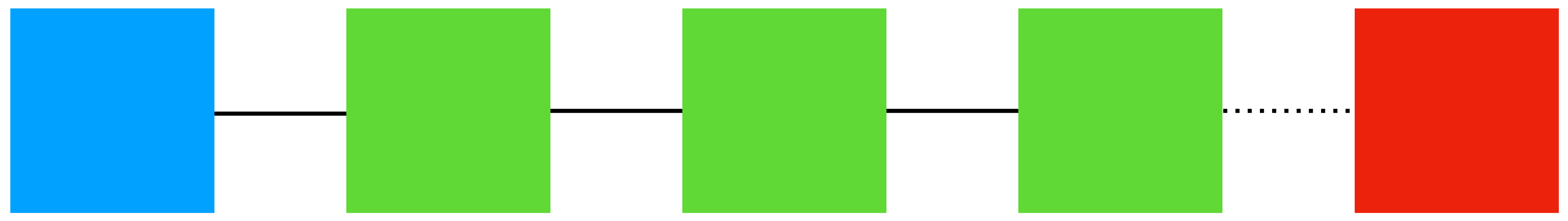
- **Weak validity:** For each correct process, its output must be the input of some correct process.

- **Strong validity:** If all correct processes receive the same input value, then they must all output that value.

- **Termination:** All processes must eventually decide on an output value

In cryptocurrency

- All nodes ("validators") have agreed on a blockchain:

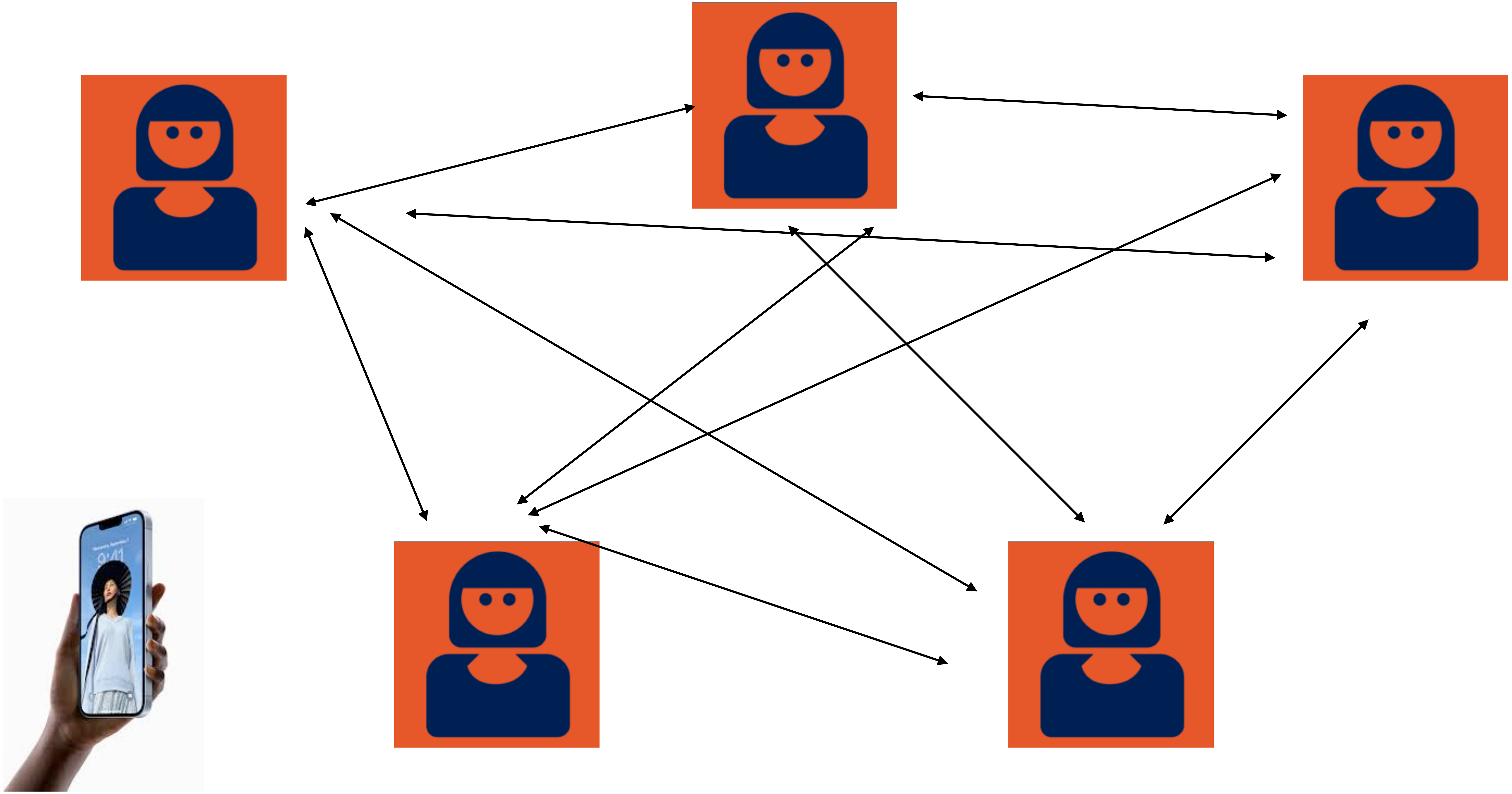


Typical goal is for all validators to <u>agree</u> on the next block in the chain (and by implication, the final hash of the new chain)

Models (and assumptions)

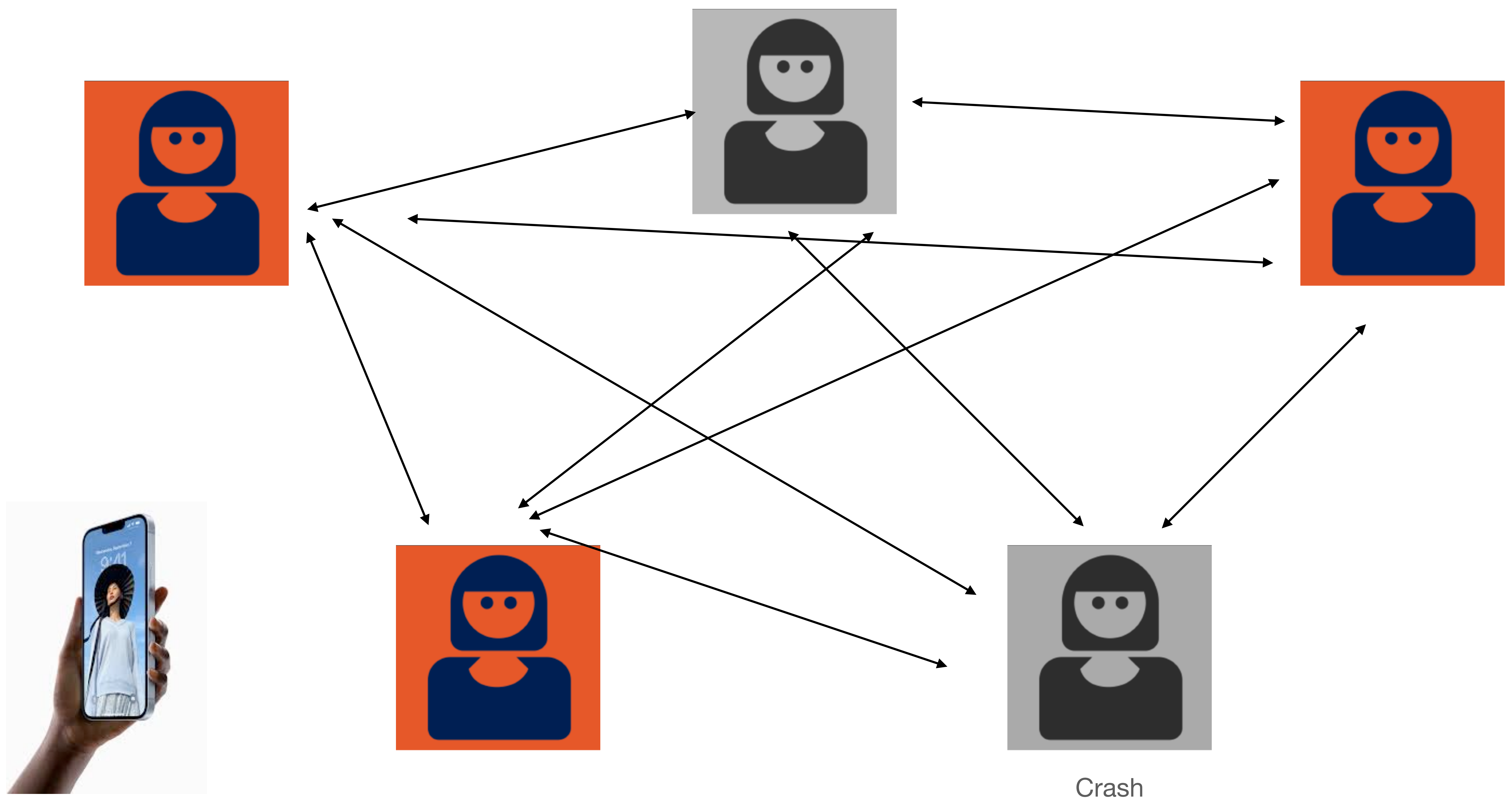
- Parties (aka "validators"):
  - There exists a set of parties who can communicate through (e.g.,) point-to-point communication links
  - Classical (non-cryptocurrency) assumption: all parties are known to each other, and can communicate securely
    - In practice this is requires (at least) cryptography!
  - Some parties are honest: they will run the protocol as written
  - Other parties are faulty: they will depart from the protocol, become non-live, accidentally faulty, or *malicious*

Models (and assumptions)



Client

Models (and assumptions)

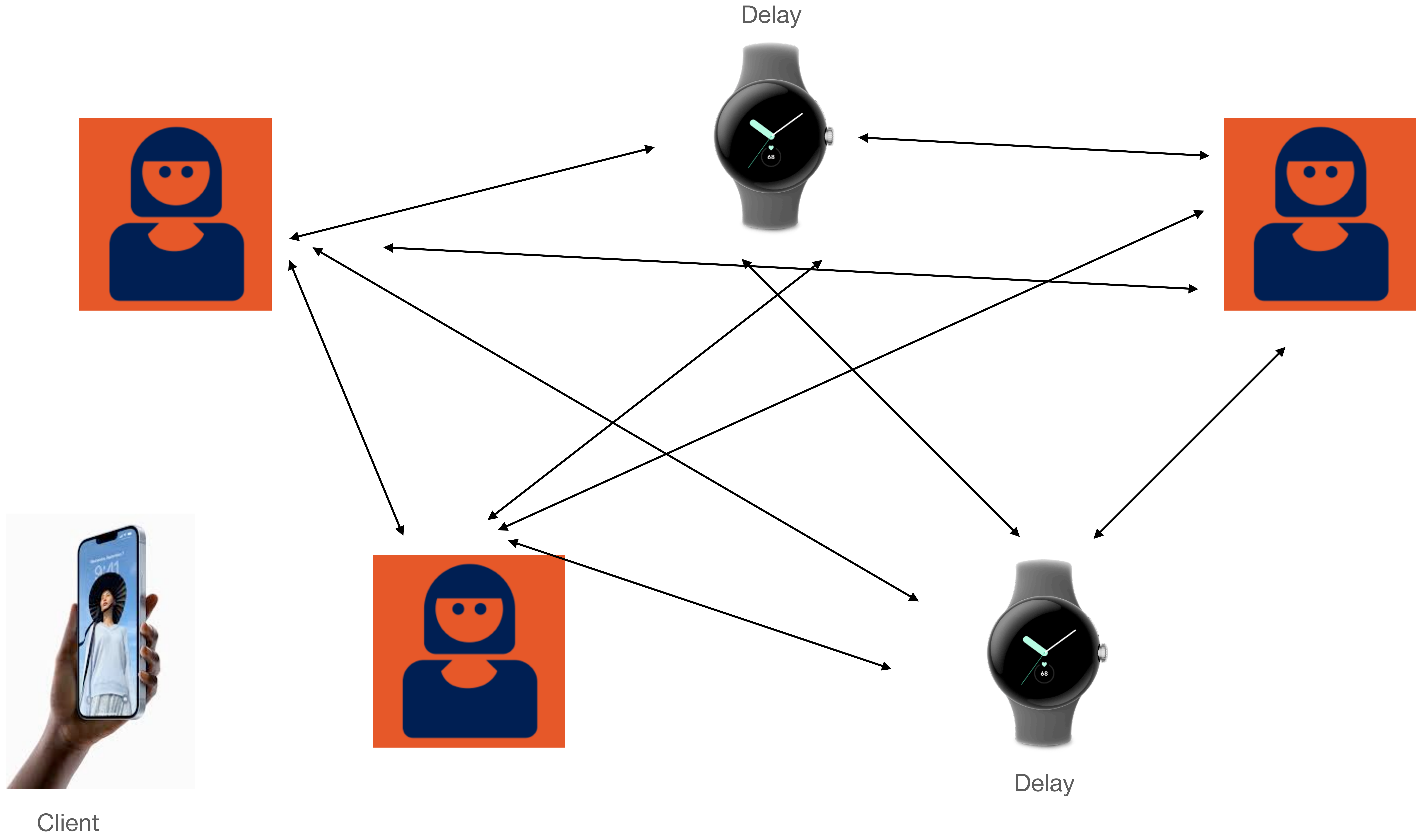


Crash

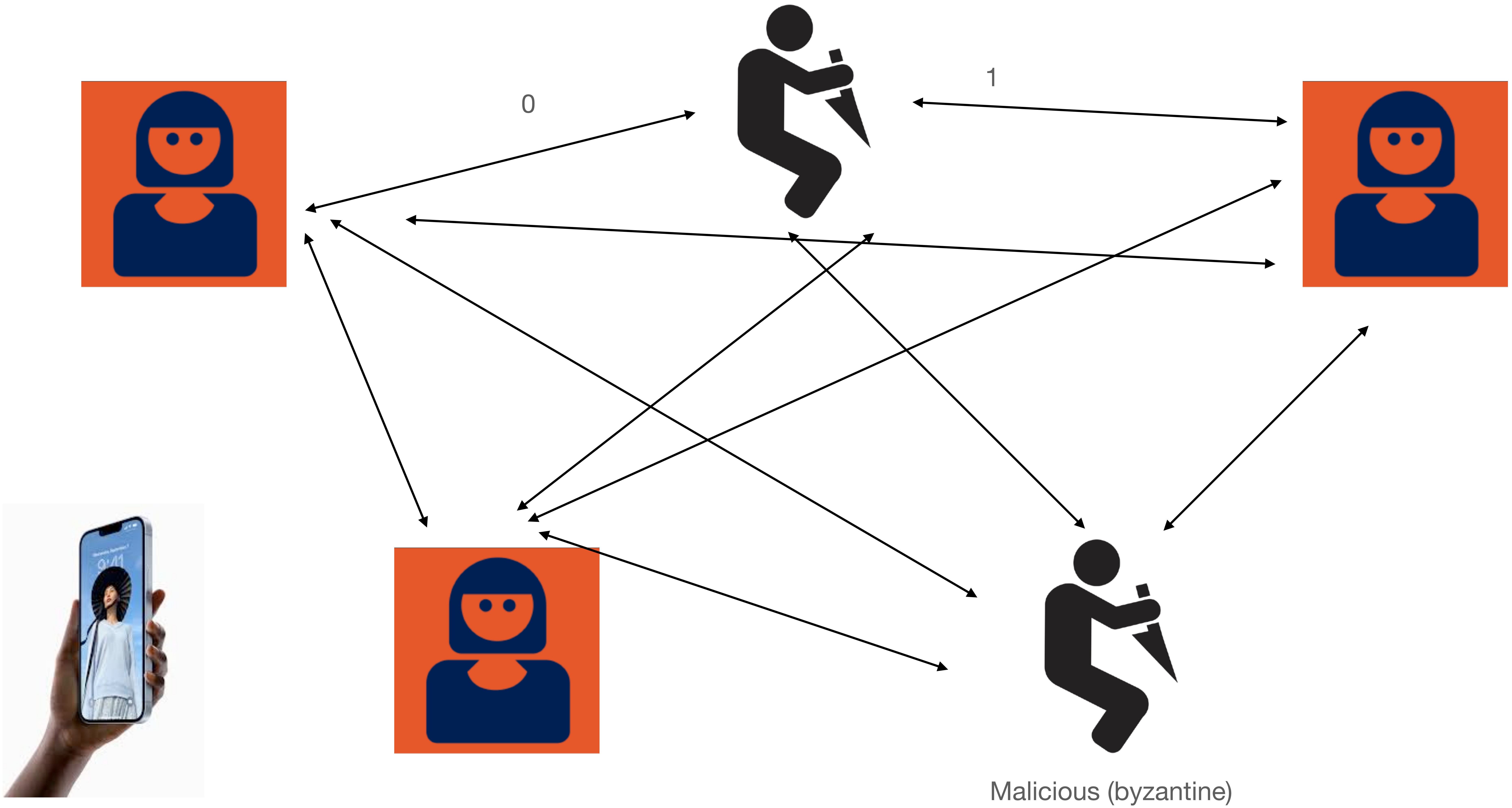
Crash

Client

Models (and assumptions)



Models (and assumptions)



Malicious (byzantine)

0

1

Malicious (byzantine)

Client

Synchronous vs. asynchronous

- Synchronous protocols:
  - Messages take place in "rounds": all messages are sent and received within a single round
- Asynchronous protocols (AKA the real world):
  - Messages can have various network delays and won't always arrive within rounds
  - There are impossibility results here!
  - In practice we can use timeouts to deal with these (timeout makes delayed message equivalent to a "crash")

Cryptocurrency-specific issues

- Traditional BFT protocols were built for <u>distributed systems</u>:
    - E.g., a distributed database running at Google
    - E.g., computers inside a Seawolf submarine…
    - Implication: *all parties are known, each party gets an equal vote*
        - <u>We can put a bound on the "number of faulty parties"</u>

Cryptocurrency-specific issues

- Traditional BFT protocols were built for <u>distributed systems</u>:
  - Think, a distributed database running at Google
  - Or the computers inside a Seawolf submarine…
  - Implication: *all parties are known, each party gets an equal vote*
    - <u>We can put a bound on the "number of faulty parties"</u>

<u>In cryptocurrency systems the participants are volunteers!</u>

<u>They could all be "sybil" attackers and faulty!</u>

Selecting participants (validators)

- **"Proof of authority" (PoA)**
  - Someone (e.g., Binance) selects the participants and stands them up, assigns public keys. *Example*: Binance Smart Chain validators
- **Proof of stake (PoS)**
  - Validators must possess (and lock up) a large amount of on-chain currency. Keys come from associated wallets. *Examples*: Ethereum, Avalanche, Cardano, etc. Participants change periodically.
- **Proof of work/space/resources etc. (PoW)**
  - Validators are selected by expending resources:
    through some Nakamoto consensus, etc.

Byzantine Fault Tolerance (BFT)

- Goal is to agree on some consensus statement in the presence of malicious (faulty) nodes

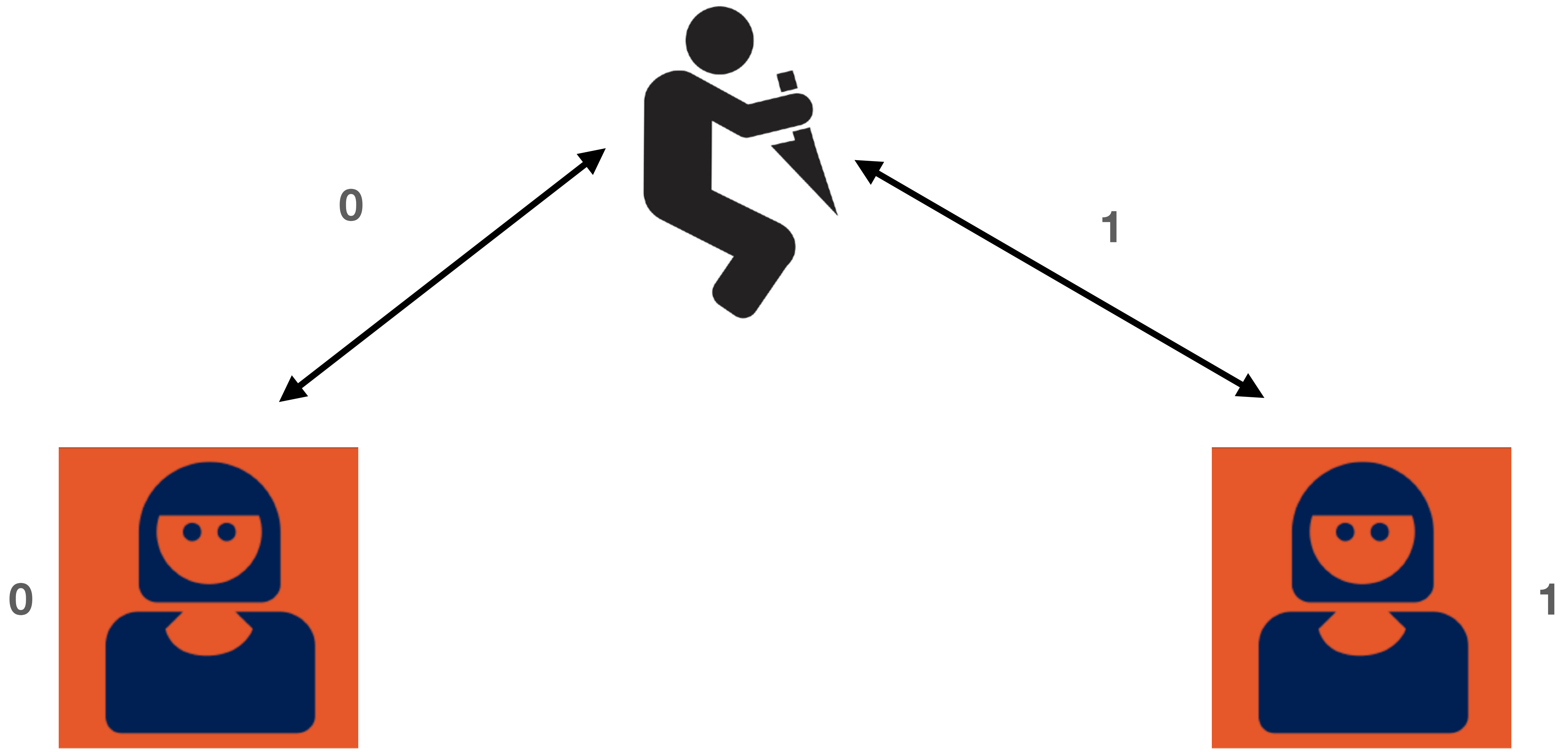
- Faulty nodes may send different messages to different nodes

- Some basic results:

  - Let f be the number of byzantine faulty nodes:
    then a network must possess $N = 3f+1$ nodes in order to achieve consensus (in synchronous consensus)

  - FLP theorem: in an asynchronous network where messages may be delayed but not lost, there is no consensus algorithm that is guaranteed to terminate in every execution (under assumptions)

(Bad) example: f=1, N=3



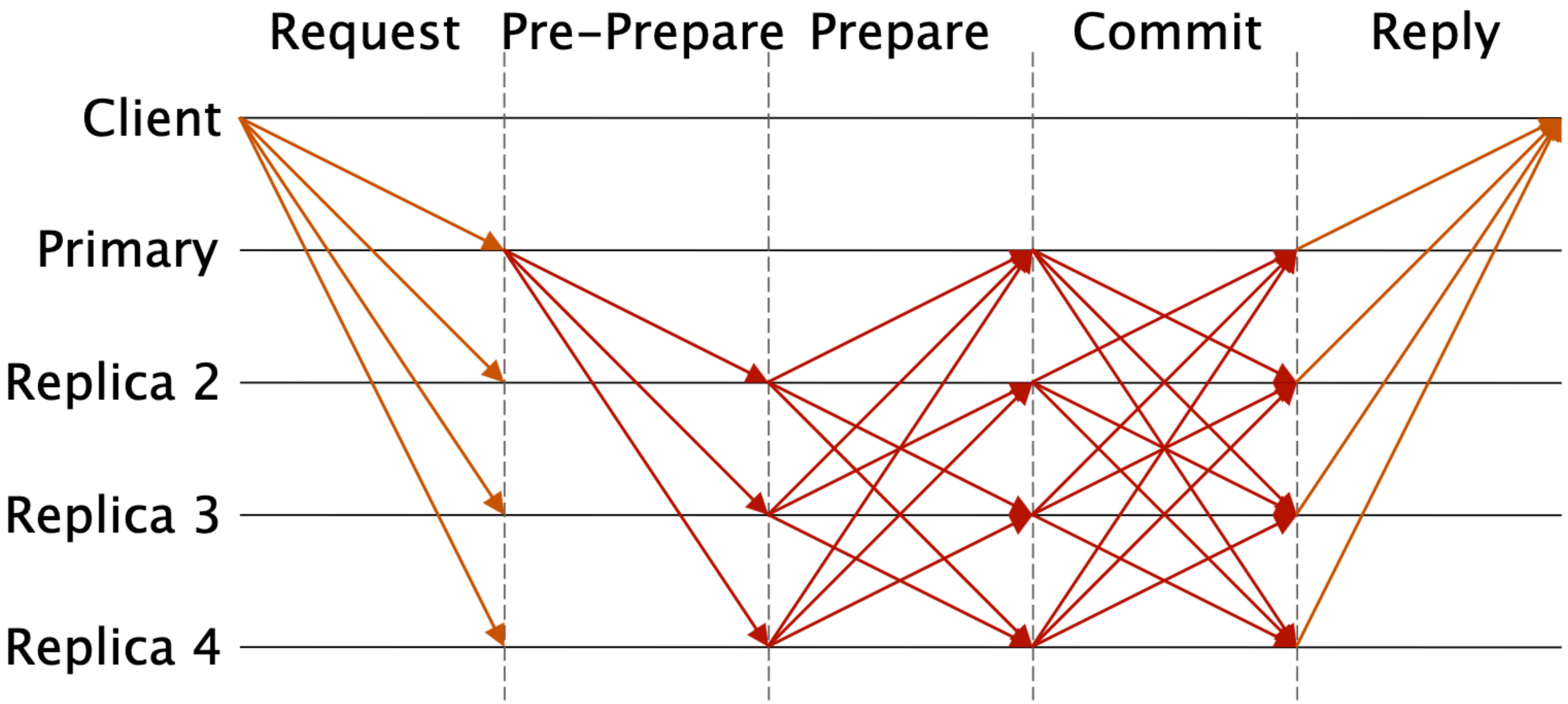
“I think 2/3 of the nodes agree on 0”

“I think 2/3 of the nodes agree on 1”

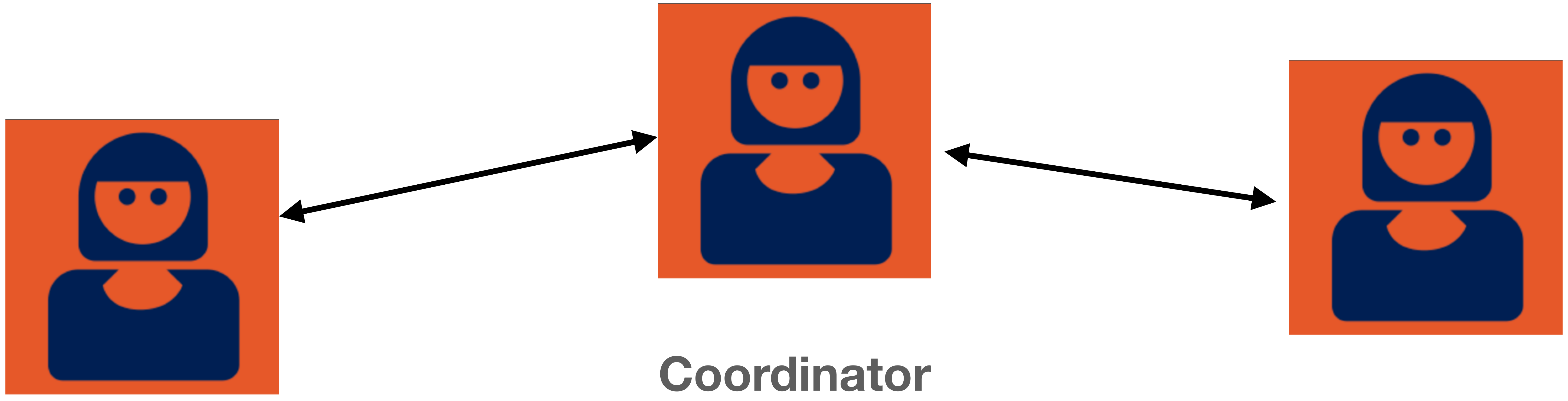
Byzantine Fault Tolerance (BFT)

- Some example algorithms (classical):
  - PBFT, Paxos
  - These were mostly used for research/teaching and had limited deployment
- In cryptocurrency systems:
  - Tendermint (used in Cosmos etc.), variants like IBFT (Polygon)
  - HotStuff (from Facebook/Libra)
  - Algorand BFT

Byzantine Fault Tolerance (BFT)

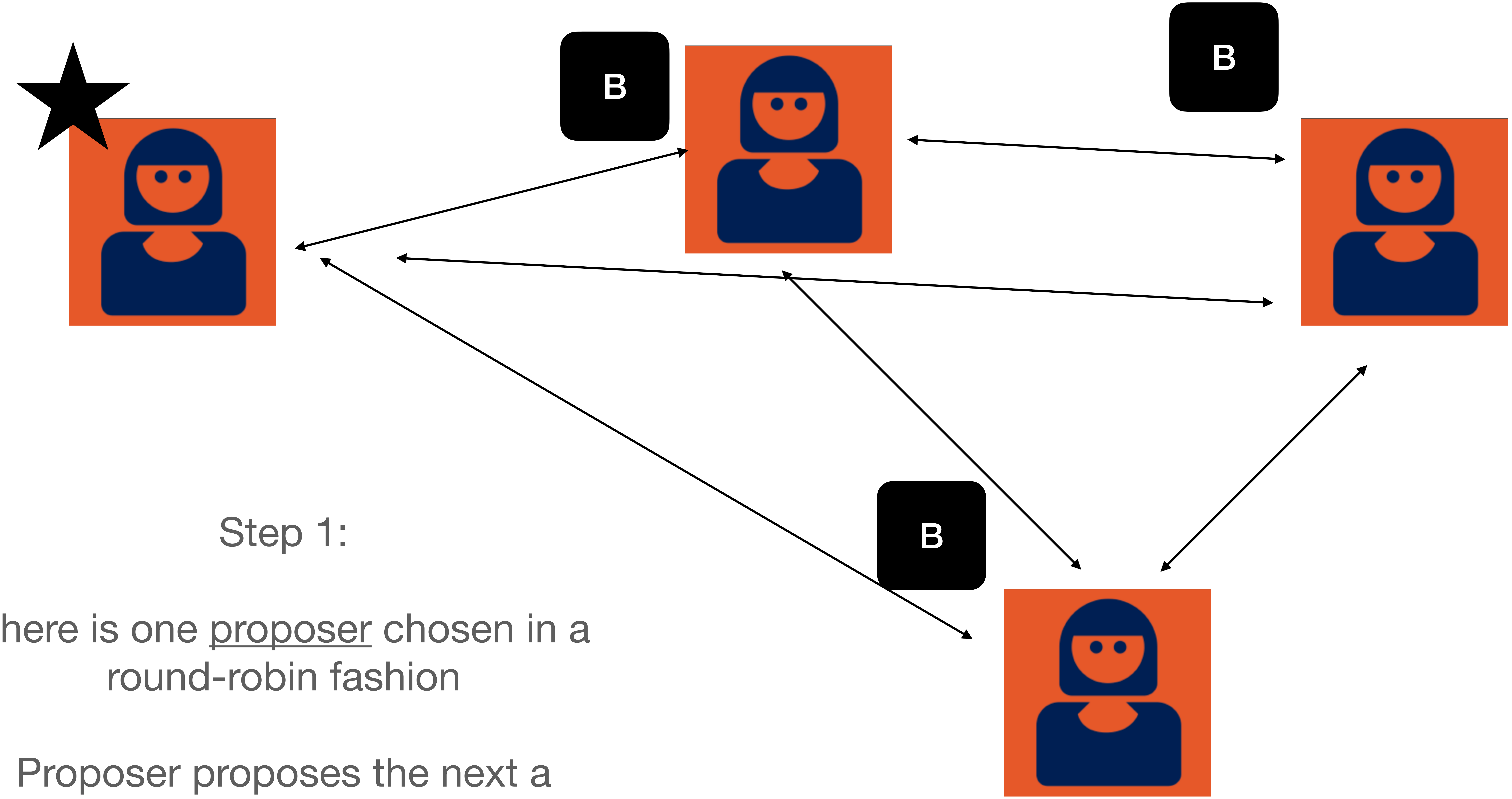


Leaders vs. leaderless



- Benefits of leaders:
  - Clients can send messages to leader, which assigns ordering
  - Simulates synchrony from asynchrony
- Downside of leaders:
  - Leaders can fail/be malicious

Tendermint: proposal

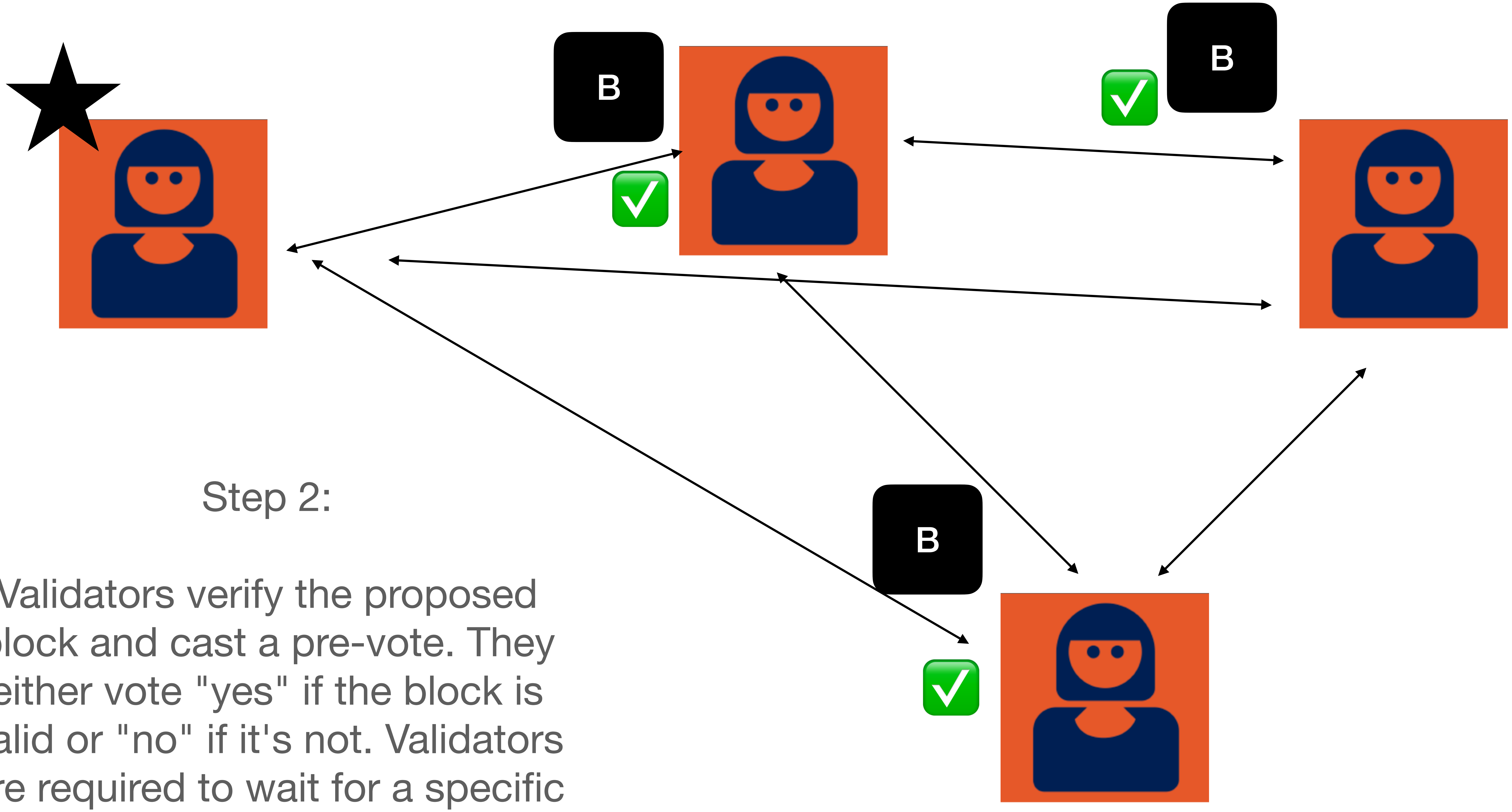


Step 1:

There is one proposer chosen in a round-robin fashion

Proposer proposes the next a block and sends it to all other participants

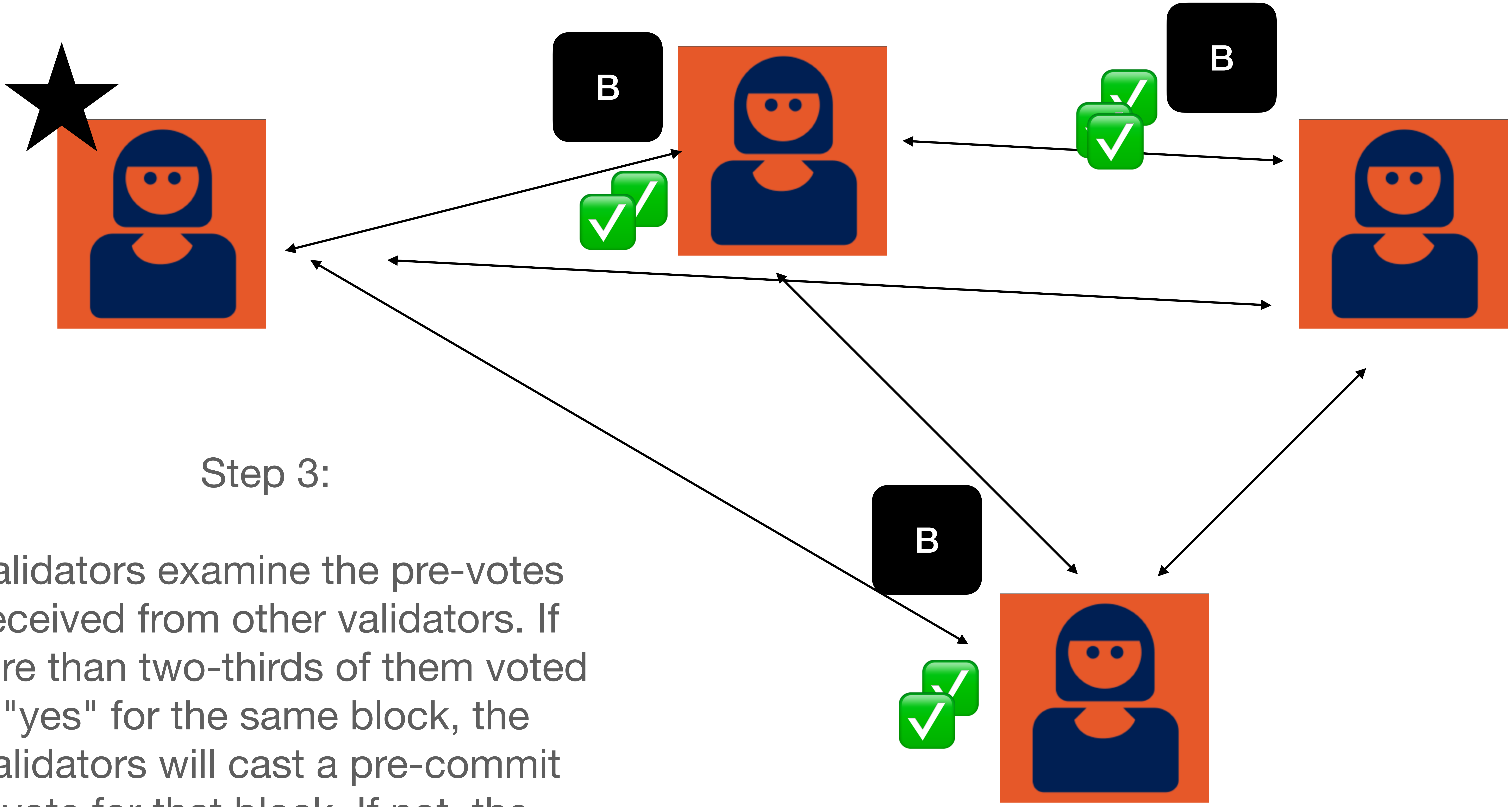
Tendermint: pre-vote



Step 2:

Validators verify the proposed block and cast a pre-vote. They either vote "yes" if the block is valid or "no" if it's not. Validators are required to wait for a specific time before voting to prevent premature decisions.

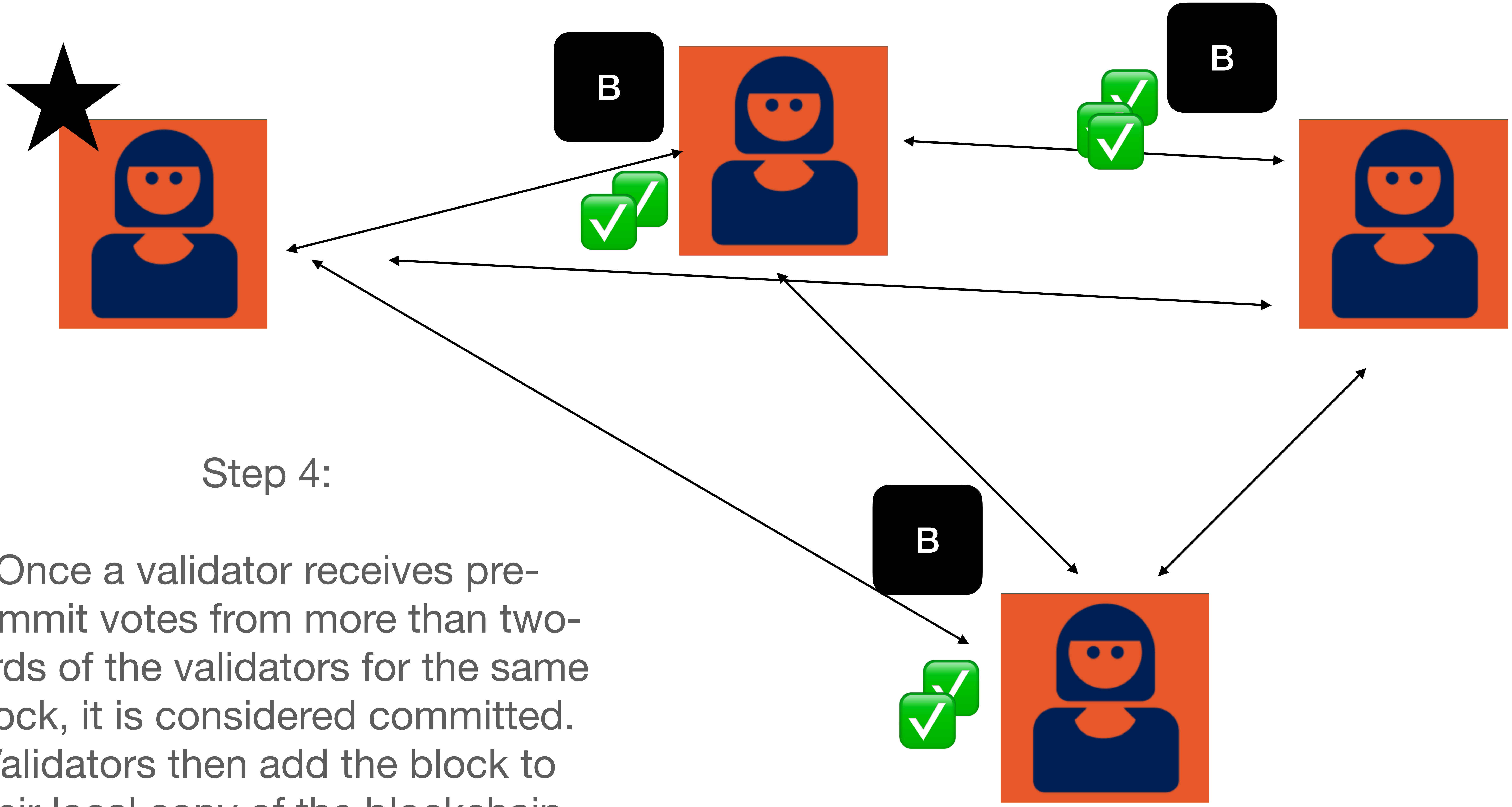
Tendermint: pre-commit



Step 3:

Validators examine the pre-votes received from other validators. If more than two-thirds of them voted "yes" for the same block, the validators will cast a pre-commit vote for that block. If not, the process moves to the next round with a new proposer.

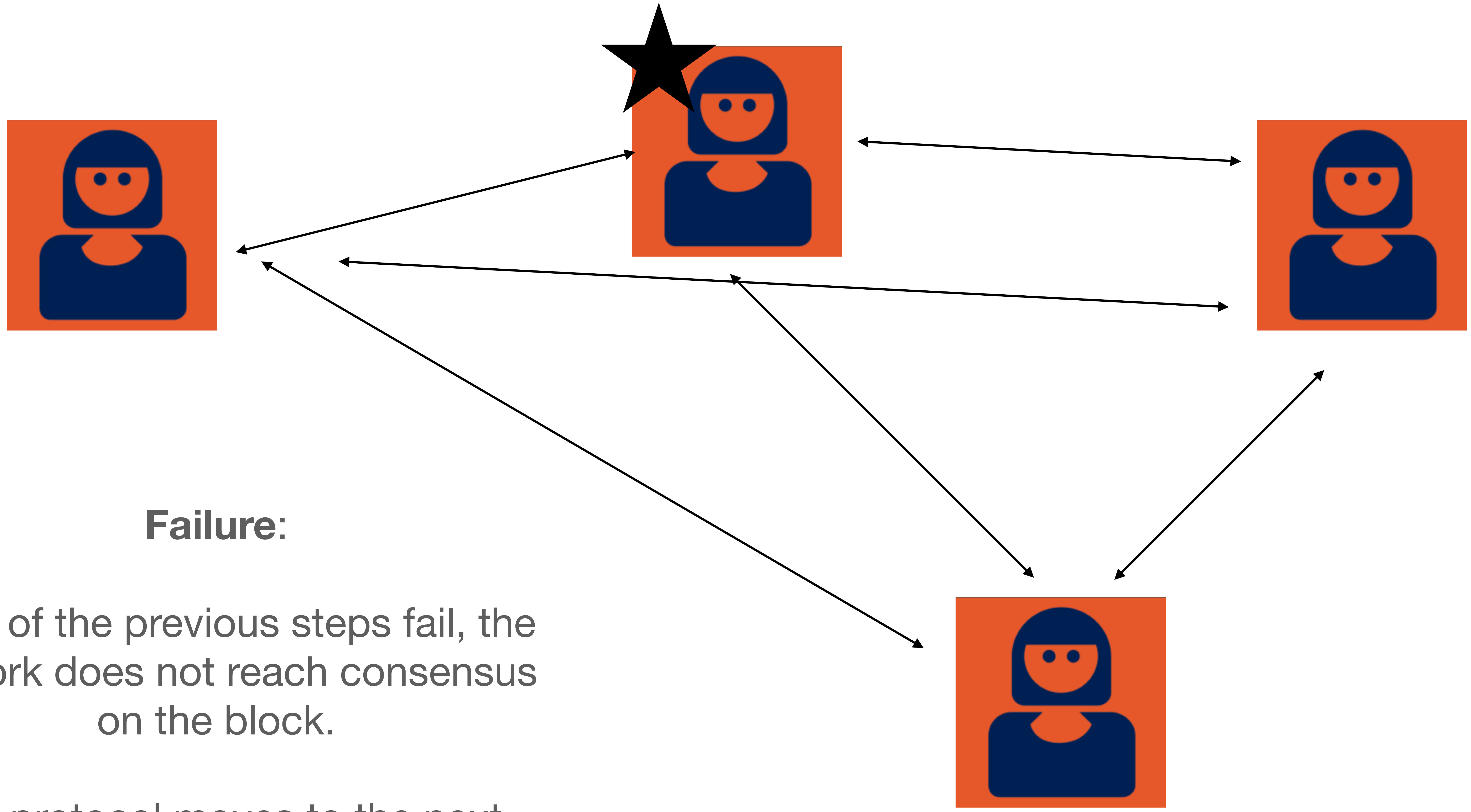
Tendermint: commit



Step 4:

Once a validator receives pre-commit votes from more than two-thirds of the validators for the same block, it is considered committed. Validators then add the block to their local copy of the blockchain, and the process starts again with a new proposal.

Tendermint: failure cases



Failure:

If any of the previous steps fail, the network does not reach consensus on the block.

The protocol moves to the next round with a new proposer.

Risks of BFT

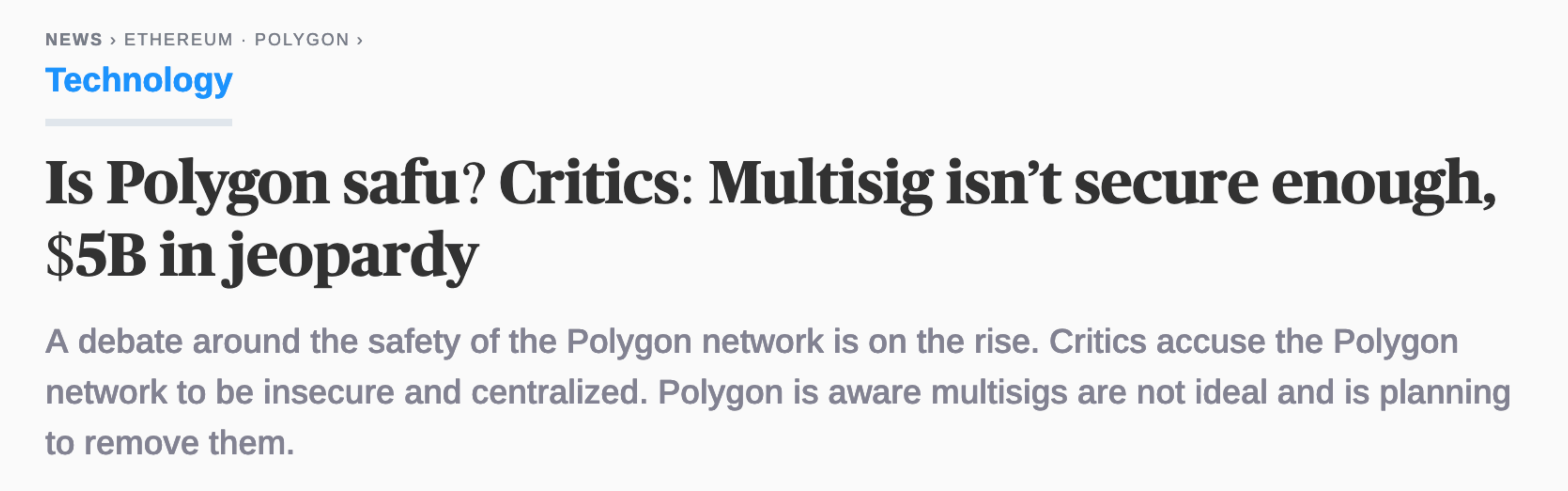
- Downsides of BFT (e.g., Tendermint etc.)
  - All nodes must communicate with each other ("multicast") and network partitions can result in stalled networks
  - Too many faulty nodes can produce stalled networks
  - DoS of the network can easily knock nodes offline
  - Failure of validators can be fatal, since we need the chain working to select new validators!

Tendermint: choosing validators?

- In proof-of-authority variants, validators are simply chosen
  - Tell all nodes that "this is your validator set"
- In some networks (Polygon), there is an auction process
  - Periodically (on Ethereum!!) people bid to a smart contract
  - The winners of this bidding get to be validators
    (they submit their public keys)

BFT: choosing validators?

- In proof-of-authority variants, validators are simply chosen
  - Tell all nodes that "this is your validator set"
- In some networks (Polygon), there is an auction process
  - Periodically (on Ethereum!!) people bid to a smart contract
  - The winners of this bidding get to be validators



NEWS › ETHEREUM · POLYGON ›

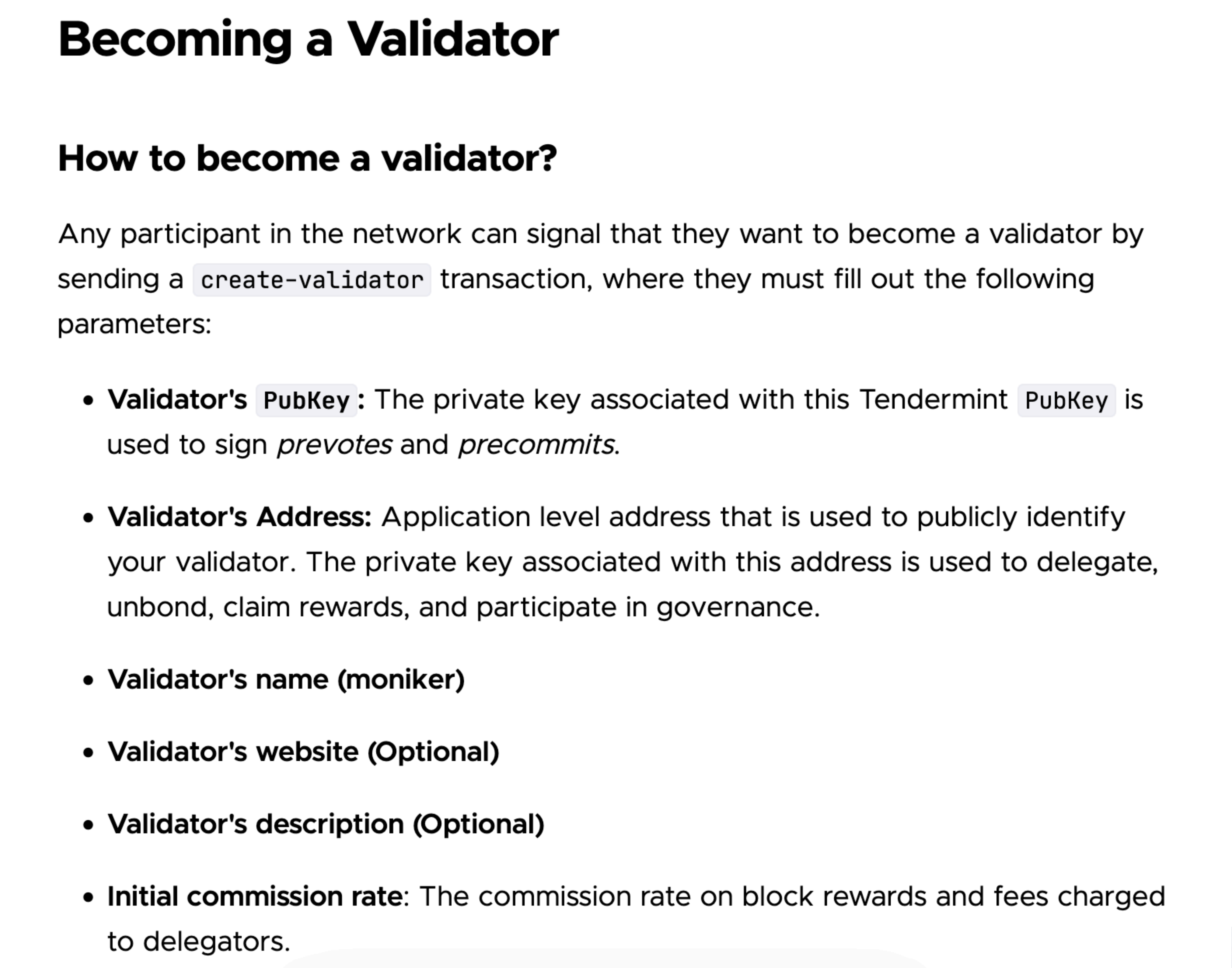
Technology

Is Polygon safu? Critics: Multisig isn't secure enough, $5B in jeopardy

A debate around the safety of the Polygon network is on the rise. Critics accuse the Polygon network to be insecure and centralized. Polygon is aware multisigs are not ideal and is planning to remove them.

BFT: choosing validators?

- In e.g., Cosmos, this is done on-chain:



Becoming a Validator

How to become a validator?

Any participant in the network can signal that they want to become a validator by sending a `create-validator` transaction, where they must fill out the following parameters:

- **Validator's `PubKey`:** The private key associated with this Tendermint `PubKey` is used to sign *prevotes* and *precommits*.
- **Validator's Address:** Application level address that is used to publicly identify your validator. The private key associated with this address is used to delegate, unbond, claim rewards, and participate in governance.
- **Validator's name (moniker)**
- **Validator's website (Optional)**
- **Validator's description (Optional)**
- **Initial commission rate**: The commission rate on block rewards and fees charged to delegators.

Algorand

- Proposed/launched by Silvio Micali ~2017
  - Based on a single-round BFT protocol
    - The idea here is that once a node broadcasts, someone might hack into it! So you want to broadcast quickly and then go away.
    - Because the protocol is single-round, you need a way to decide who is the "leader" quickly
    - This is done through cryptographic sortition

Algorand

- Proposed/launched by Silvio Micali ~2017
  - Based on a single-round BFT protocol
    - Idea: have thousands of validators, elect a small sub-committee each round, have them do a “quick” BFT
    - This BFT should be extremely rapid, and not interactive
    - The main challenge here is selecting the committee
    - This is done through a cryptographic sortition lottery, based on stake

VRFs and “sortition”

- Remember in Bitcoin, we used a “lottery”
  - Any node could solve a PoW puzzle and broadcast that solution to the network (bound to a block)
  - If the block is valid, each honest node will accept the first solution they see
  - (The gnarly cases are when two valid solutions arrive at different parts of the network)

Idea: replace the PoW lottery

- Assumptions:
  - We have already agreed on a blockchain of length T-1
  - Within that blockchain, N validators have "staked" some funds associated with their public key
  - The amount of staked funds may be different across each node!

Algorand™

Protocol: initial condition

sk1, pk1

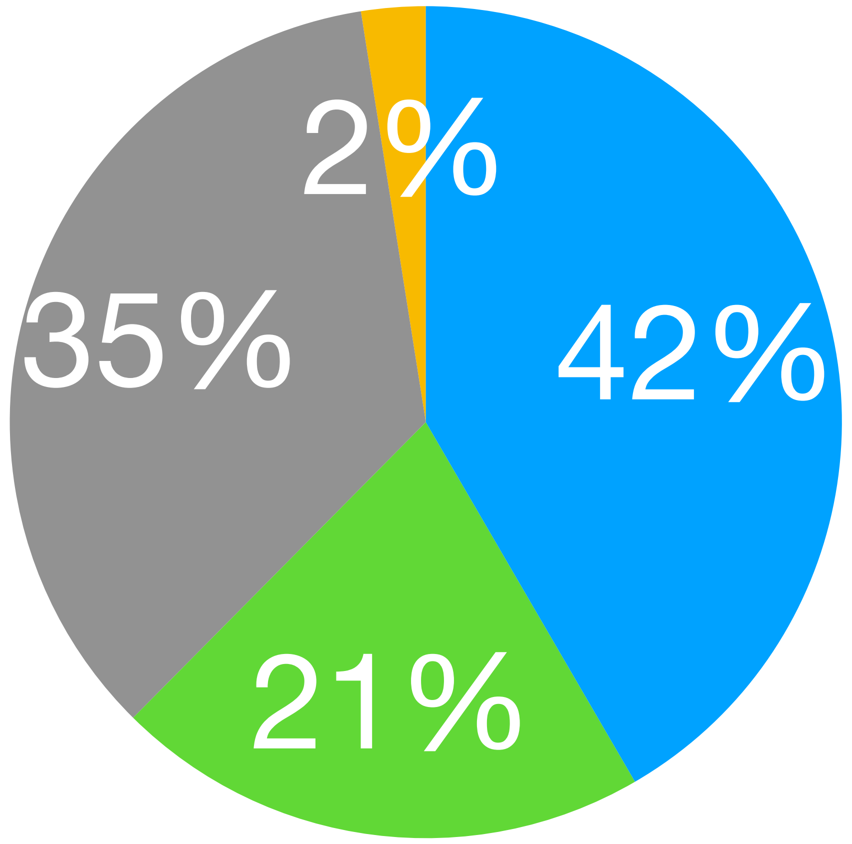
stake: 1000

sk2, pk2

stake: 500

sk3, pk3

stake: 843



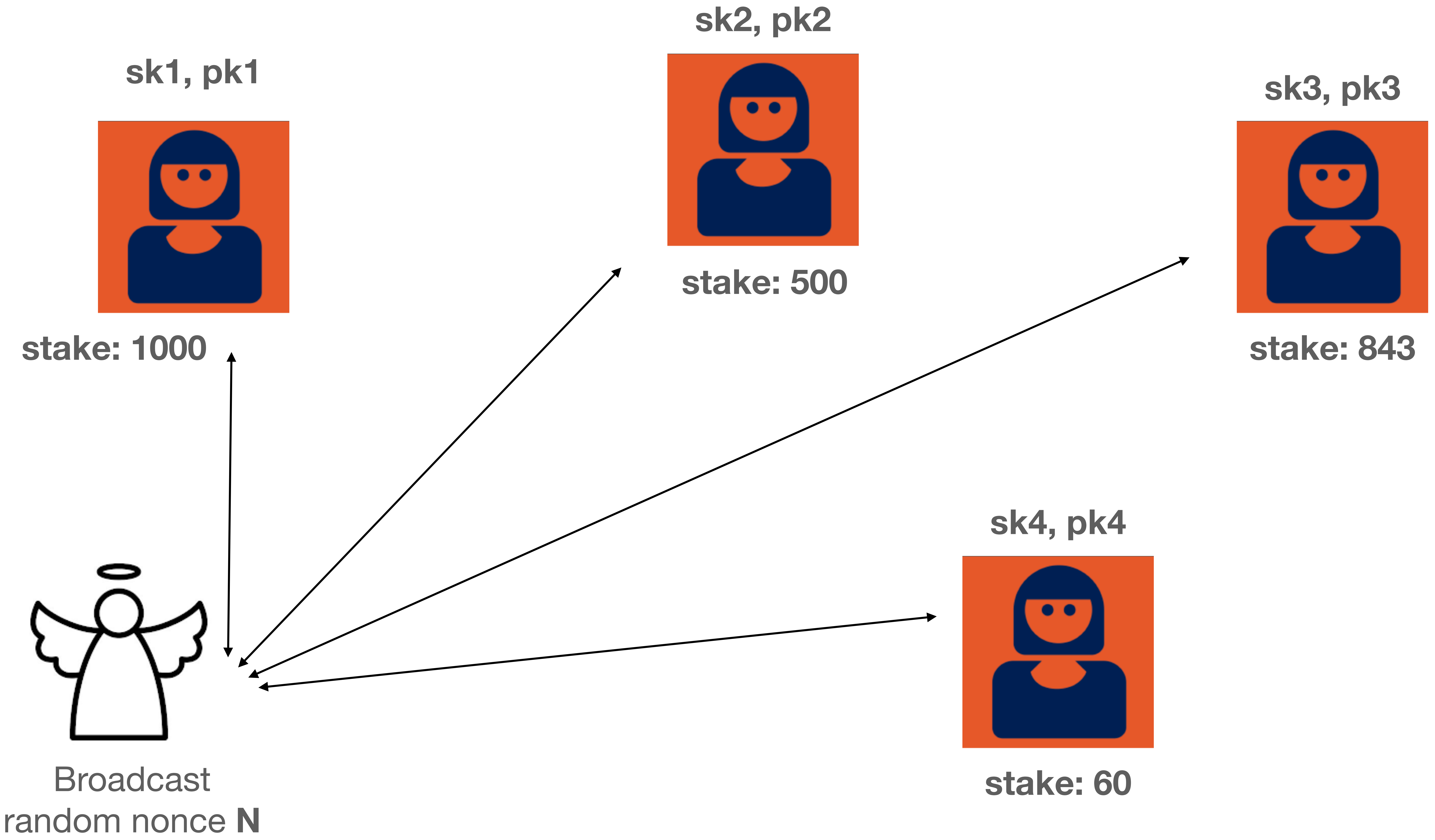
sk4, pk4

stake: 60

Idea: replace the PoW lottery

- We need a cryptographic tool: <u>Verifiable Random Functions</u>
  - A VRF has three algorithms:
    Setup, **Keygen**, **Evaluate**, **Verify**
  - *Optional:* Setup() generates global parameters (*params*)
  - Keygen(): generates a user's public/secret keypair (*pk*, *sk*)
  - Evaluate(sk, message): Produces a <u>pseudorandom</u> output **R** and a proof **π**
  - Verify(pk, R, **π**): determines if this value is correct

Protocol: step 1



sk1, pk1

stake: 1000

sk2, pk2

stake: 500

sk3, pk3

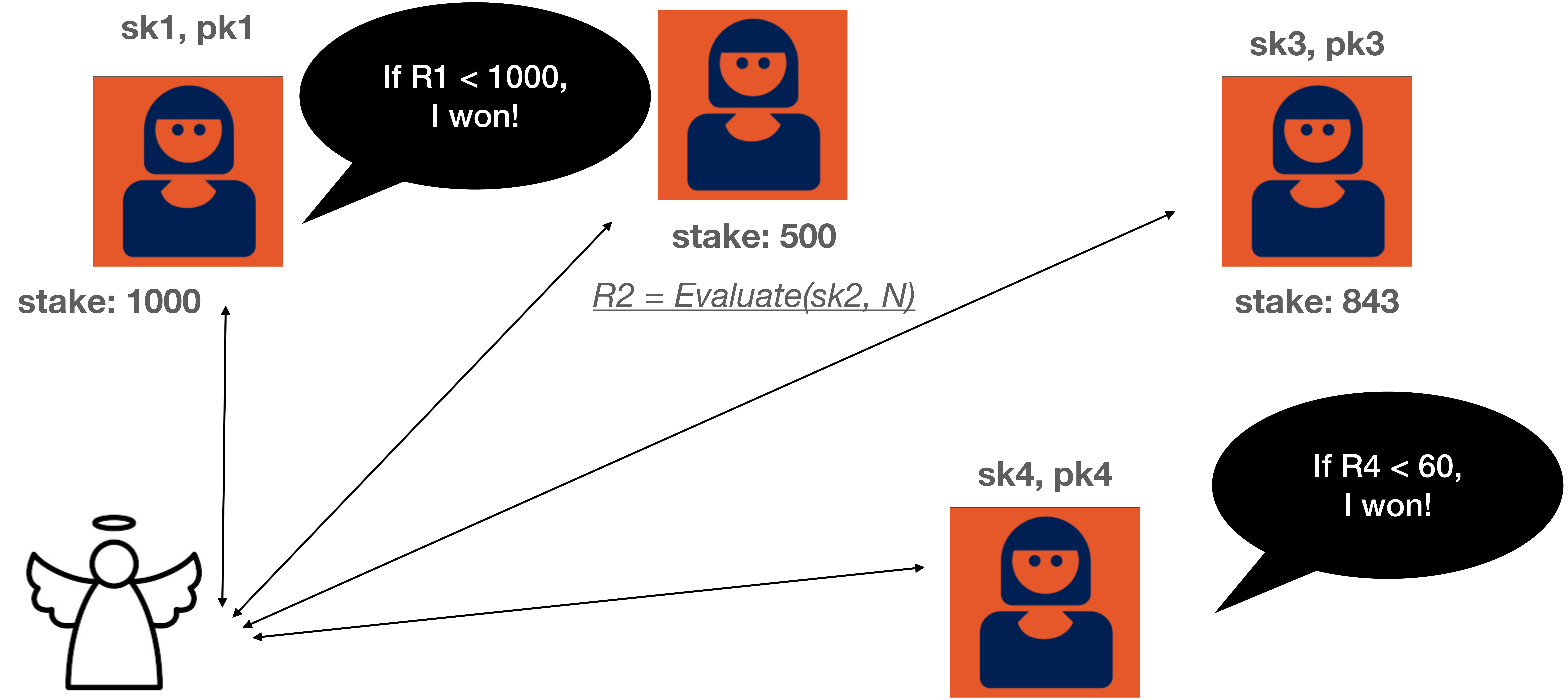
stake: 843

sk4, pk4

stake: 60

Broadcast
random nonce **N**

Protocol: step 2



R1 = Evaluate(sk1, N)

sk1, pk1

stake: 1000

If R1 < 1000,
I won!

sk2, pk2

stake: 500

R2 = Evaluate(sk2, N)

R3 = Evaluate(sk3, N)

sk3, pk3

stake: 843

sk4, pk4

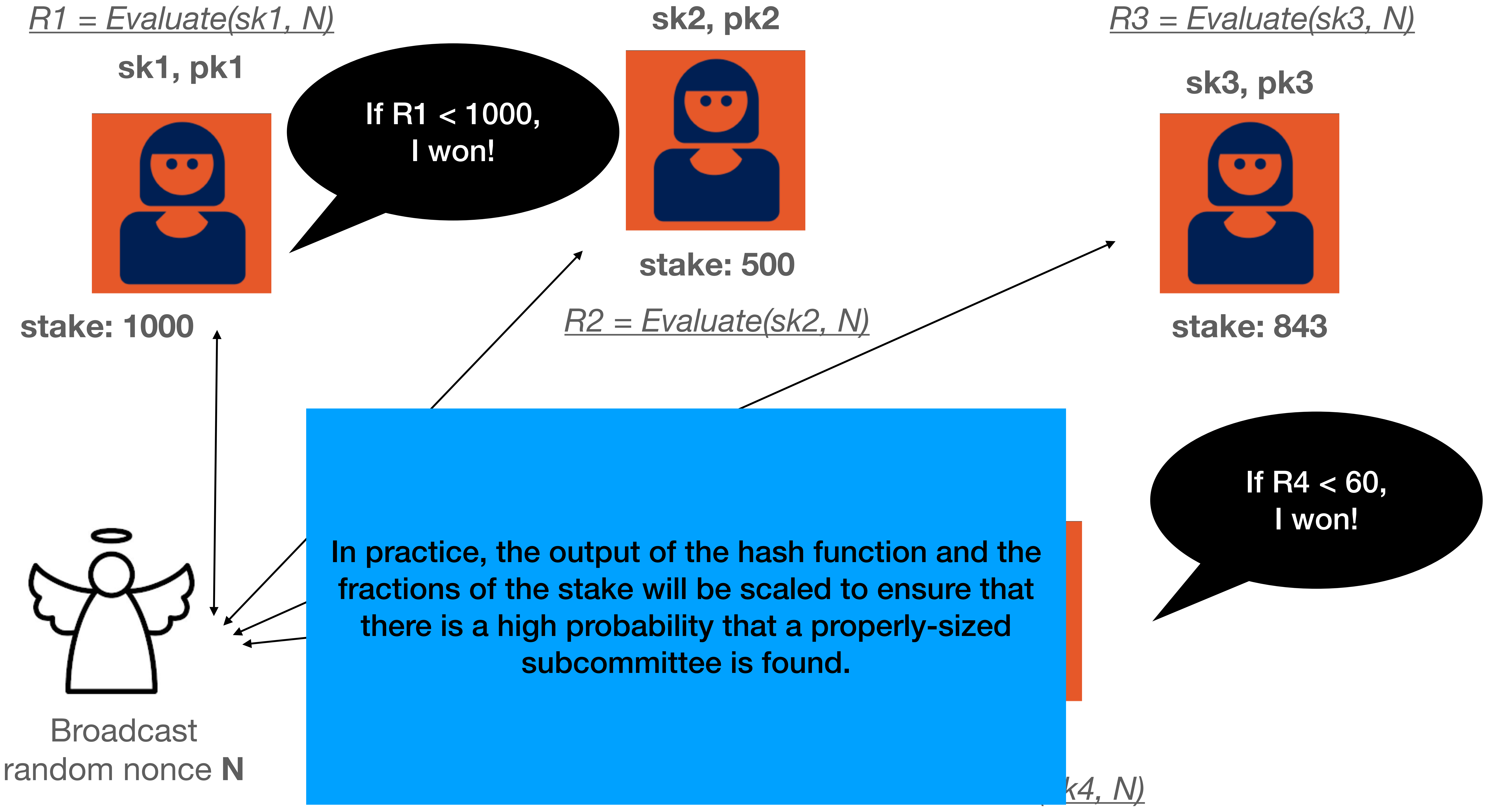
If R4 < 60,
I won!

stake: 60

R4 = Evaluate(sk4, N)

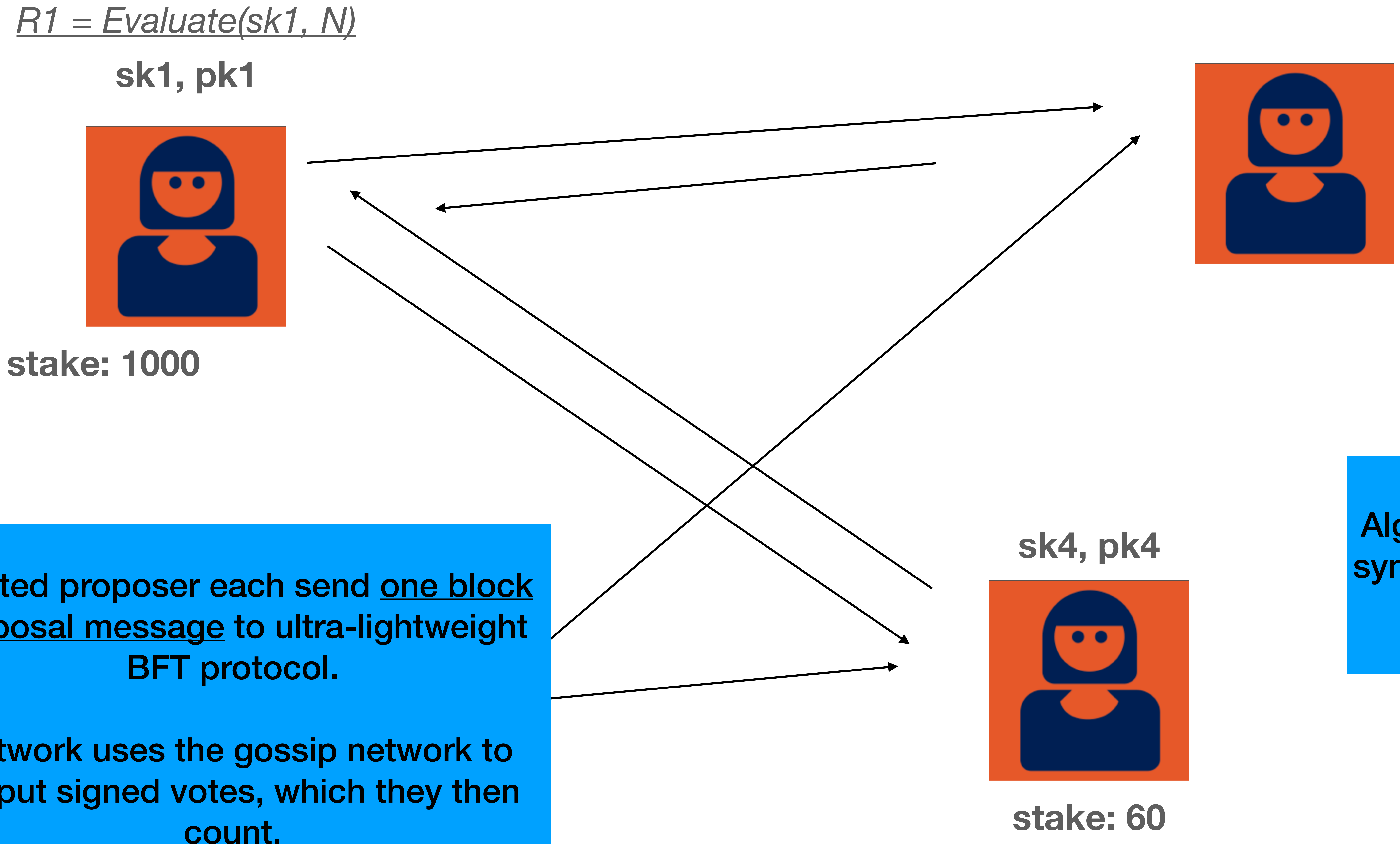
Broadcast
random nonce **N**

Protocol: step 2



In practice, the output of the hash function and the fractions of the stake will be scaled to ensure that there is a high probability that a properly-sized subcommittee is found.

Protocol: step 3



Elected proposer each send <u>one block proposal message</u> to ultra-lightweight BFT protocol.

Network uses the gossip network to output signed votes, which they then count.

Algorand uses loosely synchronized clocks to detect timeouts.

Avalanche

- Uses a different and probabilistic approach
  - Unlike Nakamoto consensus, does not rely on lotteries (I.e., randomness at the proposer side)
  - Instead each validator randomly samples a subset of the nodes it knows about, and queries them on their opinions