Lykke

# Artificial Intelligence
# vs
# Blockchain Consensus

Mikhail Nikulin, Co-founder Lykke

# A New Decentralized World

- Digital currency is truly digital
- Set of bits on your flash drive might be a value
- Digital currency is not someone’s IOU
- One can transfer money without centralized 3rd party (as well as cash)
- No payments censorship
- Smart contracts
- Decentralized Autonomous Organizations

# Benz Patent Motor Car: The first automobile (1885-1886)

Why The Blockchain Is Slow

# Consensus?

# Consensus algorithms

- BFT
- Hashgraph
- Avalanche
- Algorand / Ethereum Capser FFG
- Nakamoto (Bitcoin)
- Bitcoin NG

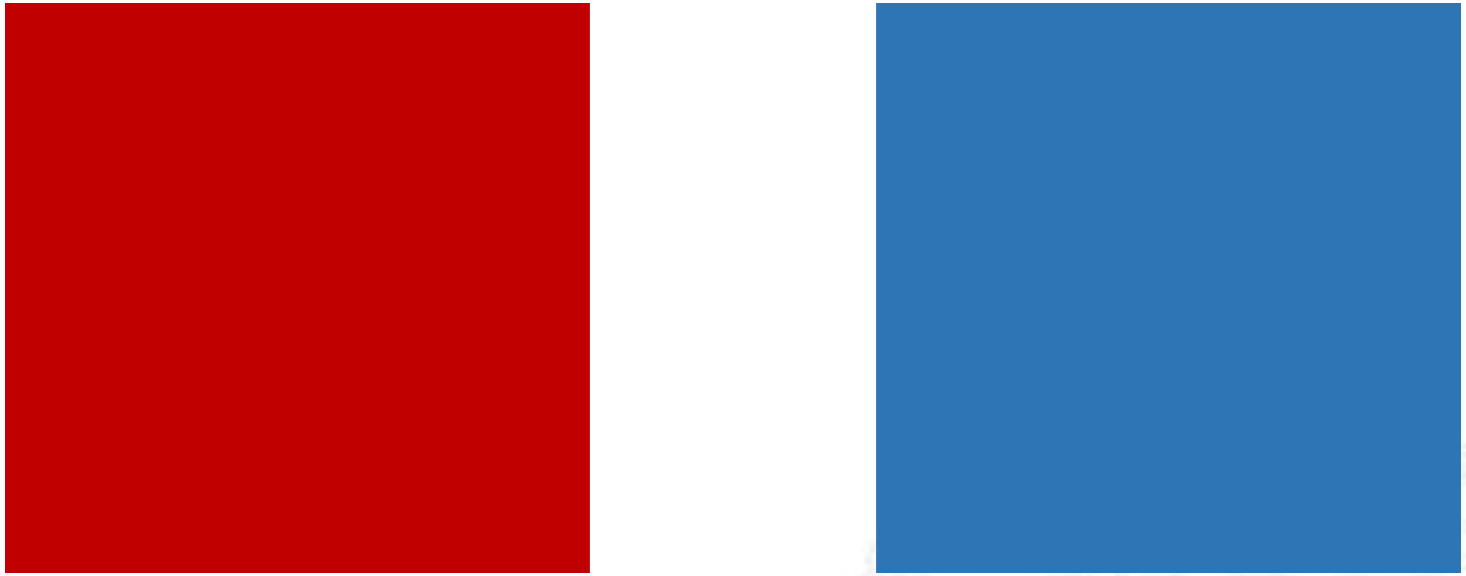
# BFT Consensus

Traditional consensus is referred to Byzantine Fault Tolerance (BFT)

*Assumption: Attacker controls <1/3*

- You will reach consensus
- You know when you reach consensus
- You never are wrong

# BFT state-machine



# Nakamoto Consensus (Bitcoin)

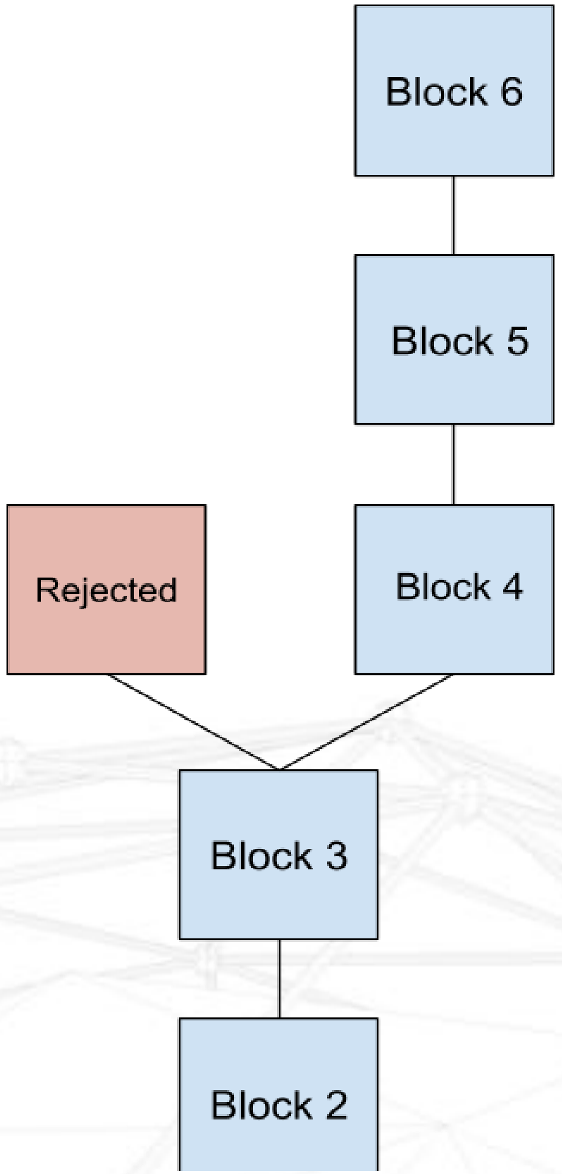
Nakamoto Consensus is based on PoW

Bitcoin is not BFT

Bitcoin is not deterministic but rather probabilistic consensus

In Bitcoin, there is never a moment in time where you know that you have consensus and you'll never be wrong. All that happens is that you become more confident over time.

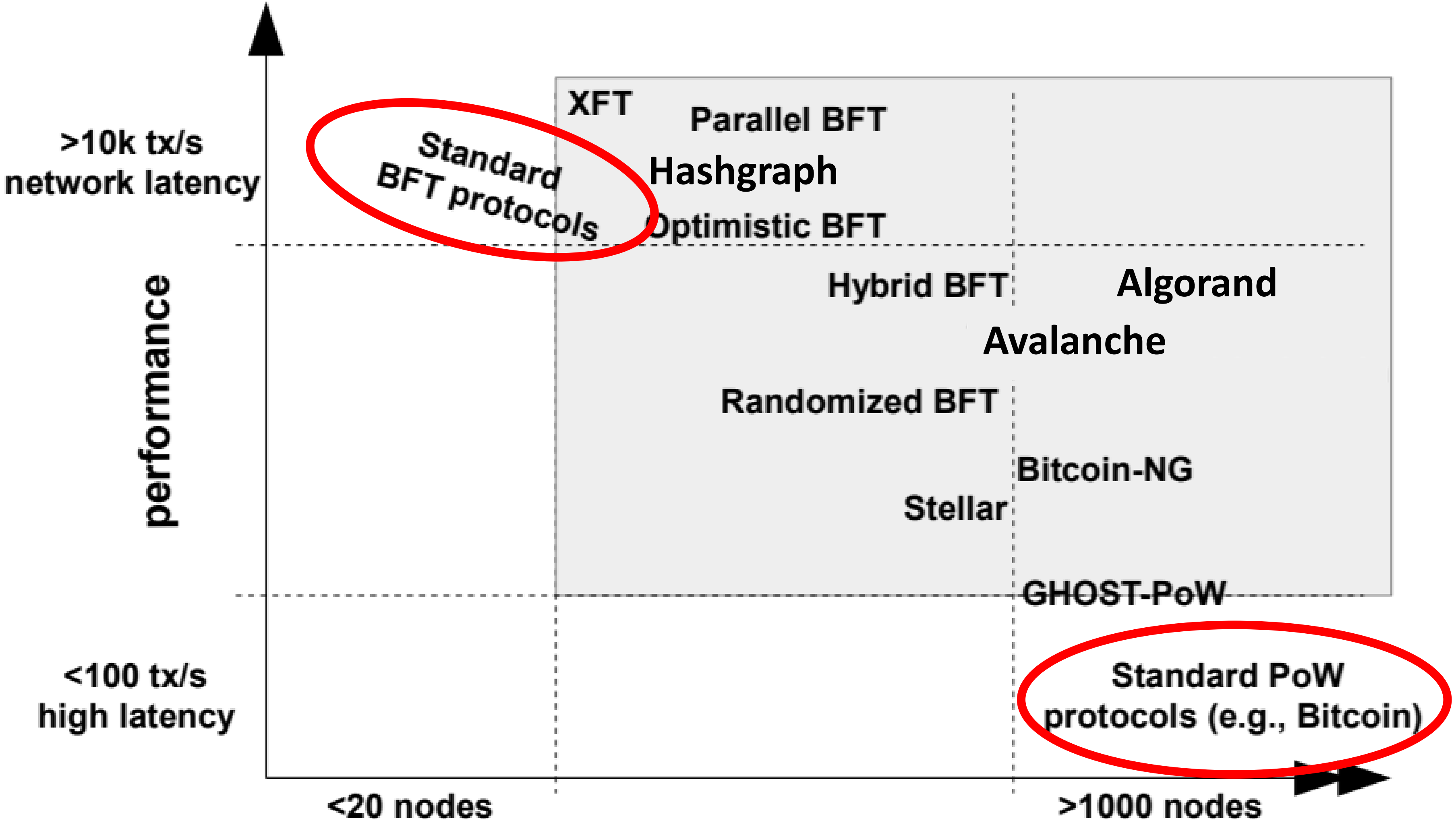
Bitcoin is synchronous protocol



# PoW vs BFT

| | PoW consensus | BFT consensus |
|---|---|---|
| Node identity management | **open, entirely decentralized** | permissioned, nodes need to know IDs of all other nodes |
| Consensus finality | no | **yes** |
| Scalability (no. of nodes) | **excellent (thousands of nodes)** | limited, not well explored (tested only up to $n \leq 20$ nodes) |
| Scalability (no. of clients) | **excellent (thousands of clients)** | **excellent (thousands of clients)** |
| Performance (throughput) | limited (due to possible of chain forks) | **excellent (tens of thousands tx/sec)** |
| Performance (latency) | high latency (due to multi-block confirmations) | **excellent (matches network latency)** |
| Power consumption | very poor (PoW wastes energy) | **good** |
| Tolerated power of an adversary | $\leq 25\%$ computing power | $\leq 33\%$ voting power |
| Network synchrony assumptions | physical clock timestamps (e.g., for block validity) | **none for consensus safety** (synchrony needed for liveness) |
| Correctness proofs | no | **yes** |

# PoW vs BFT

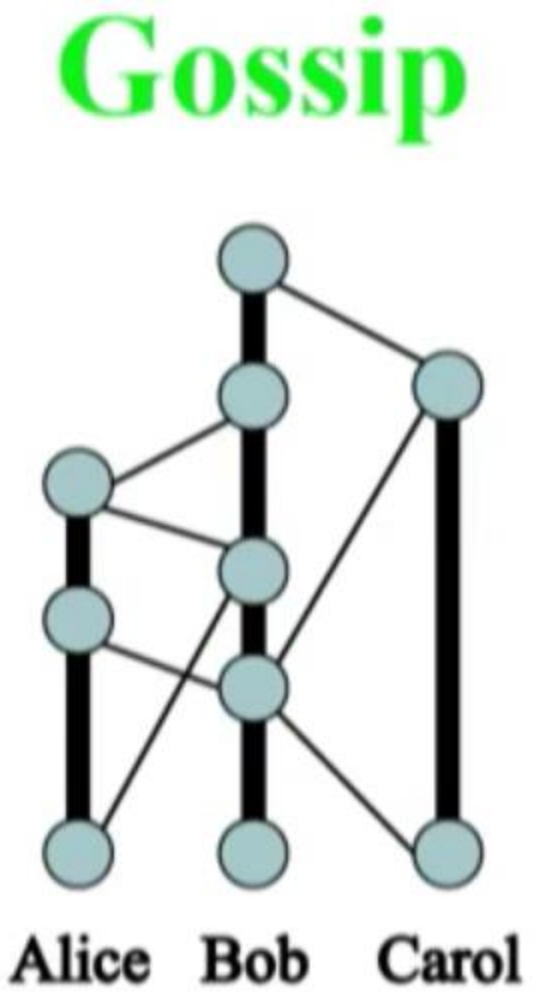

>10k tx/s
network latency
performance
<100 tx/s
high latency
<20 nodes
>1000 nodes
Standard
BFT protocols
XFT
Parallel BFT
Hashgraph
Optimistic BFT
Hybrid BFT
Algorand
Avalanche
Randomized BFT
Bitcoin-NG
Stellar
GHOST-PoW
Standard PoW
protocols (e.g., Bitcoin)

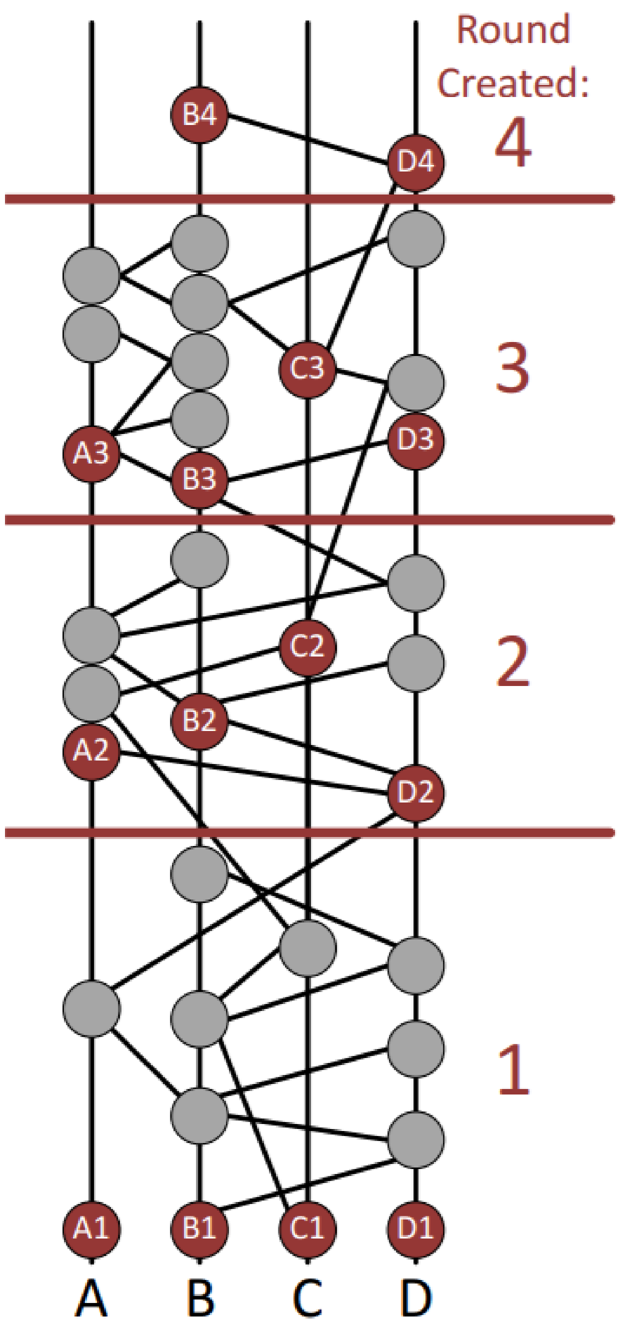
# Bitcoin-NG

- Bitcoin-NG uses standard PoW for leader election, declaring a node which mines a block with standard difficulty (called a key block) to become a leader until a new key block is mined.
- The leader can append microblocks to the chain, which are not subject to PoW mining but are merely hashchained together.
- Bitcoin-NG mixes leader election, often seen in BFT protocols, with a leader-centric protocol

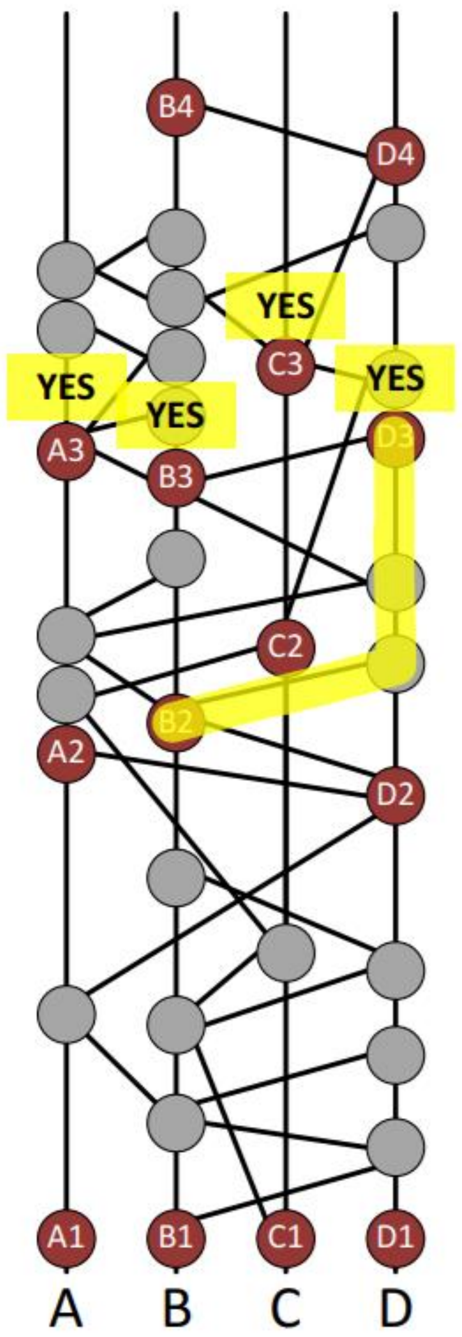
# Hashgraph

- Asynchronous BFT
- Every node in Hashgraph can spread signed information (called events) on newly-created transactions and transactions received from others, to its randomly chosen neighbors.
- These neighbors will aggregate received events with information received from other nodes into a new event, and then send it on to other randomly chosen neighbors.
- This process continues until all the nodes are aware of the information created or received at the beginning.
- History of the gossip protocol can be illustrated by a directed graph, i.e., each node maintains a graph representing sequences of forwarders/witnesses for each transaction

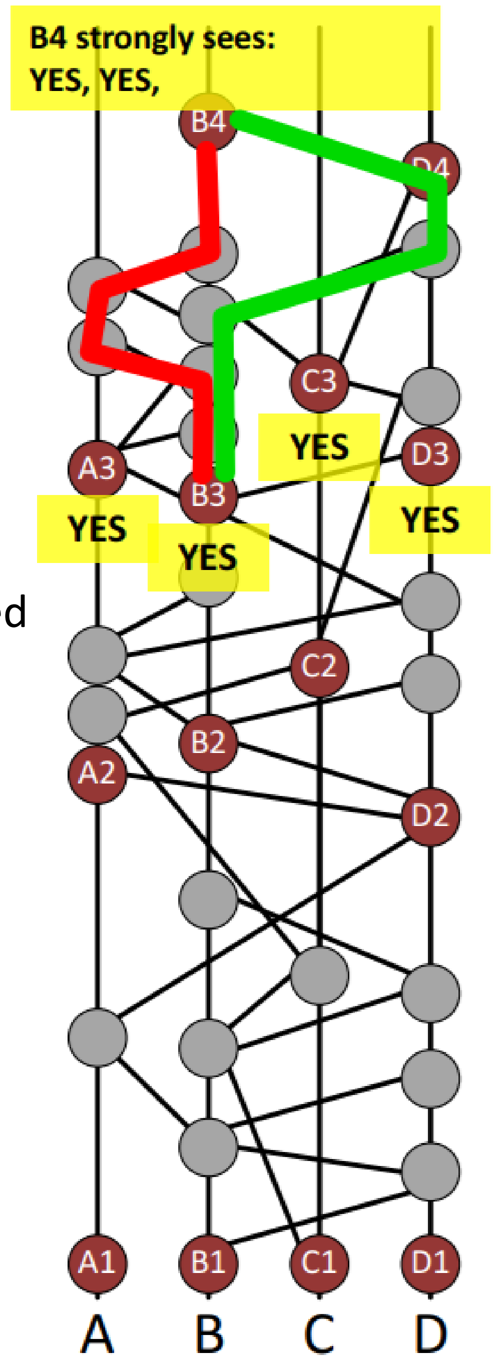





B2 is famous

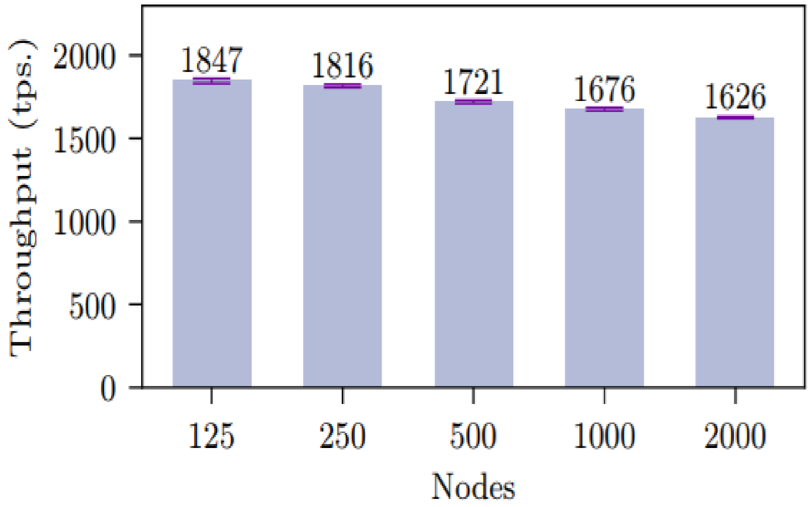



B2 confirmed



# Avalanche

- Partially synchronous BFT
- Non deterministic - similar to Bitcoin, Avalanche leaves determining the acceptance point of a transaction to the application
- A single node of the network starts with picking a small number of random peer nodes, say five, and asks them to choose a color.
- Each peer node then responds with a vote which the requesting node uses to form a weighted result of all votes. In the figure above (in the first frame), from the node's perspective the entire network is tilting towards red based on one round of polling.




- This process repeats itself for everybody else.

# Avalanche

```
procedure SNOWFLAKELOOP(u, col₀ ∈ {R, B, ⊥})
    col := col₀, cnt := 0
    while undecided do
        if col = ⊥ then continue
        𝒦 := SAMPLE(𝒩\u, k)
        P := [QUERY(v, col)   for v ∈ 𝒦]
        for col′ ∈ {R, B} do
            if P.COUNT(col′) ≥ α · k then
                if col′ ≠ col then
                    col := col′, cnt := 0
                else
                    if ++cnt > β then ACCEPT(col)
```

```
procedure SNOWBALLLOOP(u, col₀ ∈ {R, B, ⊥})
    col := col₀, lastcol := col₀, cnt := 0
    d[R] := 0, d[B] := 0
    while undecided do
        if col = ⊥ then continue
        𝒦 := SAMPLE(𝒩\u, k)
        P := [QUERY(v, col)   for v ∈ 𝒦]
        for col′ ∈ {R, B} do
            if P.COUNT(col′) ≥ α · k then
                d[col′]++
                if d[col′] > d[col] then
                    col := col′
                if col′ ≠ lastcol then
                    lastcol := col′, cnt := 0
                else
                    if ++cnt > β then ACCEPT(col)
```

# Avalanche

**procedure** AVALANCHELOOP
  **while** true **do**
    find $T$ that satisfies $T \in \mathcal{T} \wedge T \notin \mathcal{Q}$
    $\mathcal{K} := \text{SAMPLE}(\mathcal{N} \backslash u, k)$
    $P := \sum_{v \in \mathcal{K}} \text{QUERY}(v, T)$
    **if** $P \geq \alpha \cdot k$ **then**
      $c_T := 1$
      // update the preference for ancestors
      **for** $T' \in \mathcal{T} : T' \xleftarrow{*} T$ **do**
        **if** $d(T') > d(\mathcal{P}_{T'}.\text{pref})$ **then**
          $\mathcal{P}_{T'}.\text{pref} := T'$
        **if** $T' \neq \mathcal{P}_{T'}.\text{last}$ **then**
          $\mathcal{P}_{T'}.\text{last} := T'$, $\mathcal{P}_{T'}.\text{cnt} := 0$
        **else**
          $\text{++}\mathcal{P}_{T'}.\text{cnt}$
    // otherwise, $c_T$ remains 0 forever
    $\mathcal{Q} := \mathcal{Q} \cup \{T\}$   // mark T as queried

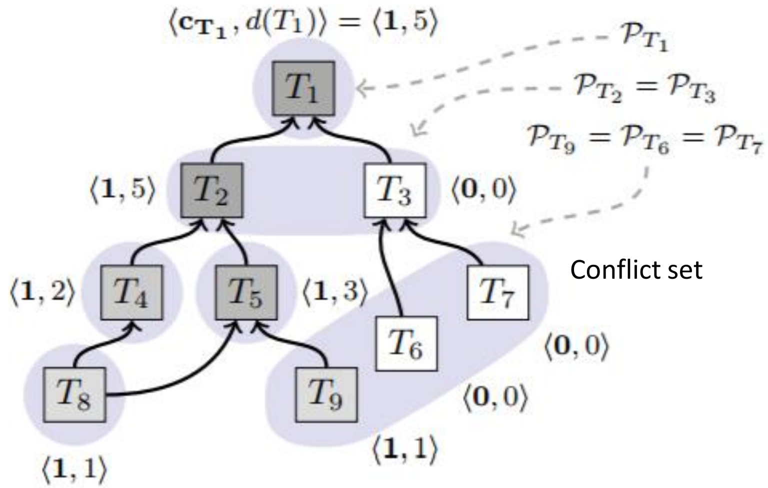



Figure 7: Example of chit and confidence values, labeled as tuples in that order. Darker boxes indicate transactions with higher confidence values.

# Algorand / Ethereum Casper FFG

**procedure** $BA\star(ctx, round, block)$:

$hblock \leftarrow \text{Reduction}(ctx, round, H(block))$
$hblock_\star \leftarrow \text{Binary}BA\star(ctx, round, hblock)$
// Check if we reached "final" or "tentative" consensus
$r \leftarrow \text{CountVotes}(ctx, round, \text{FINAL}, T_{\text{FINAL}}, \tau_{\text{FINAL}}, \lambda_{\text{STEP}})$
**if** $hblock_\star = r$ **then**
  **return** $\langle \text{FINAL}, \text{BlockOfHash}(hblock_\star)\rangle$
**else**
  **return** $\langle \text{TENTATIVE}, \text{BlockOfHash}(hblock_\star)\rangle$

**Algorithm 3:** Running $BA\star$ for the next *round*, with a proposed *block*. H is a cryptographic hash function.

**procedure** $\text{CommitteeVote}(ctx, round, step, \tau, value)$:

// check if user is in committee using Sortition (Alg. 1)
$role \leftarrow \langle \text{"committee"}, round, step\rangle$
$\langle sorthash, \pi, j\rangle \leftarrow \text{Sortition}(user.sk, ctx.seed, \tau, role,$
    $ctx.weight[user.pk], ctx.W)$
// only committee members originate a message
**if** $j > 0$ **then**
  $\text{Gossip}(\langle user.pk, \text{Signed}_{user.sk}(round, step,$
    $sorthash, \pi, H(ctx.last\_block), value)\rangle)$

**procedure** $\text{Binary}BA\star(ctx, round, block\_hash)$:

$step \leftarrow 1$
$r \leftarrow block\_hash$
$empty\_hash \leftarrow H(\text{Empty}(round, H(ctx.last\_block)))$
**while** $step < \text{MAXSTEPS}$ **do**
  $\text{CommitteeVote}(ctx, round, step, \tau_{\text{STEP}}, r)$
  $r \leftarrow \text{CountVotes}(ctx, round, step, T_{\text{STEP}}, \tau_{\text{STEP}}, \lambda_{\text{STEP}})$
  **if** $r = \text{TIMEOUT}$ **then**
    $r \leftarrow block\_hash$
  **else if** $r \neq empty\_hash$ **then**
    **for** $step < s' \leq step + 3$ **do**
      $\text{CommitteeVote}(ctx, round, s', \tau_{\text{STEP}}, r)$
    **if** $step = 1$ **then**
      $\text{CommitteeVote}(ctx, round, \text{FINAL}, \tau_{\text{FINAL}}, r)$
    **return** r
  $step{+}{+}$

  $\text{CommitteeVote}(ctx, round, step, \tau_{\text{STEP}}, r)$
  $r \leftarrow \text{CountVotes}(ctx, round, step, T_{\text{STEP}}, \tau_{\text{STEP}}, \lambda_{\text{STEP}})$
  **if** $r = \text{TIMEOUT}$ **then**
    $r \leftarrow empty\_hash$
  **else if** $r = empty\_hash$ **then**
    **for** $step < s' \leq step + 3$ **do**
      $\text{CommitteeVote}(ctx, round, s', \tau_{\text{STEP}}, r)$
    **return** r
  $step{+}{+}$

# How to Make Consensus Work Safe and Fast

# Sharding is the way to scale BFT

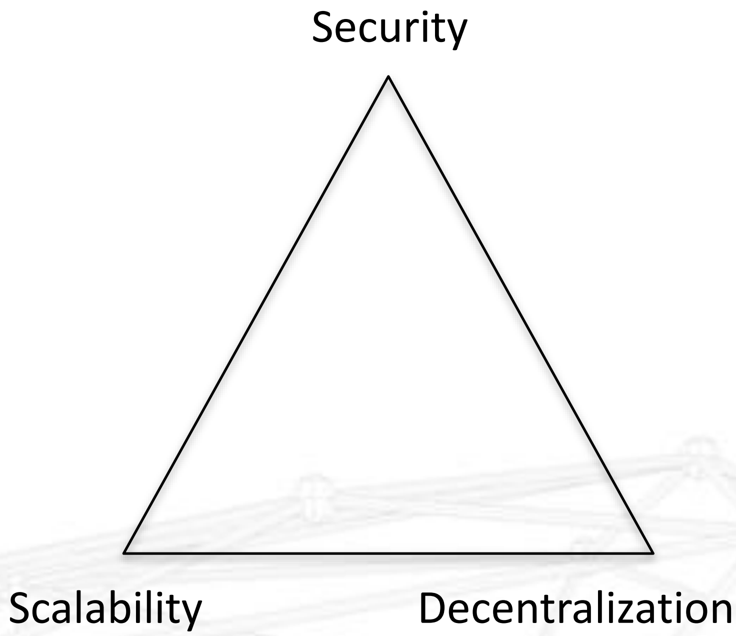
- Sharding improves the scalability of the blockchain by splitting the network into smaller “groups/pieces”
- Sharding is good for nodes but then you need to have identity management
- Sharding does not work for pure PoW

*In fact, if you split the network into 100 pieces, you only need 1% of the total hash-power to takeover a shard*

# Blockchain Trilemma

Blockchain systems can only have 2 of those 3:

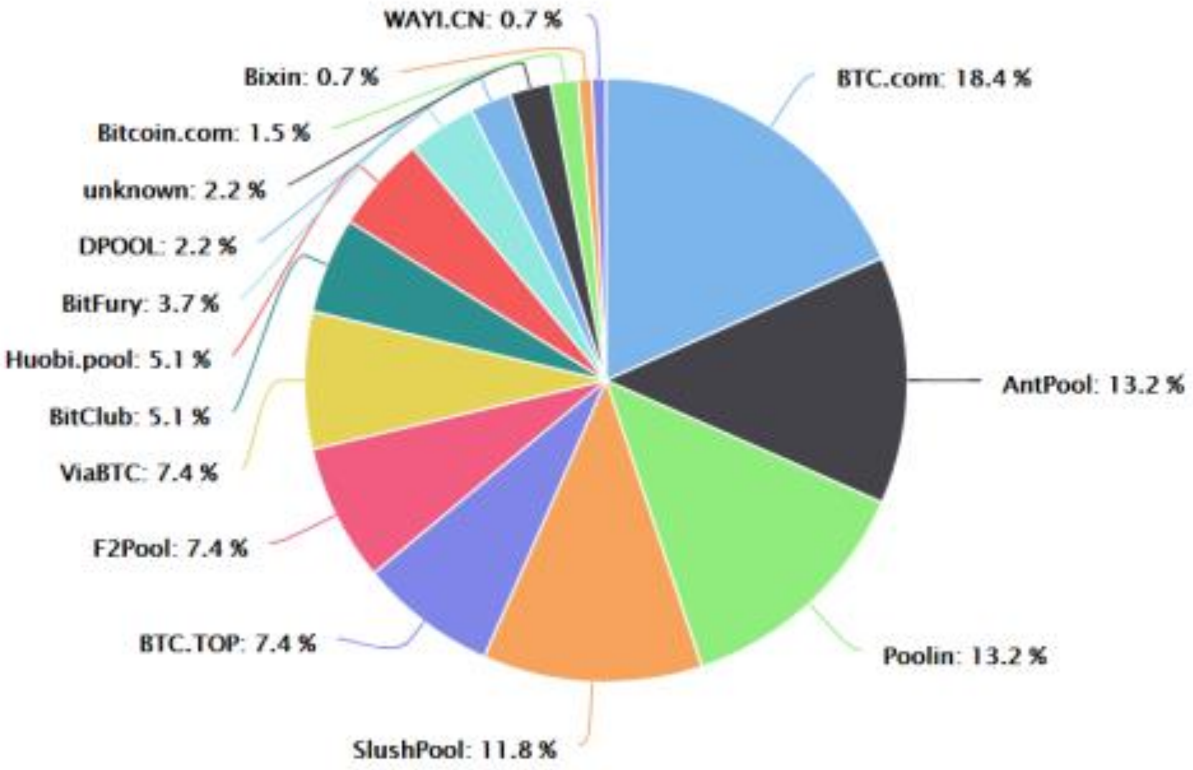
- Scalability
- Decentralization
- Security



# Decentralization Myth – Bitcoin

13 mining pools controlling more than 95% of the network

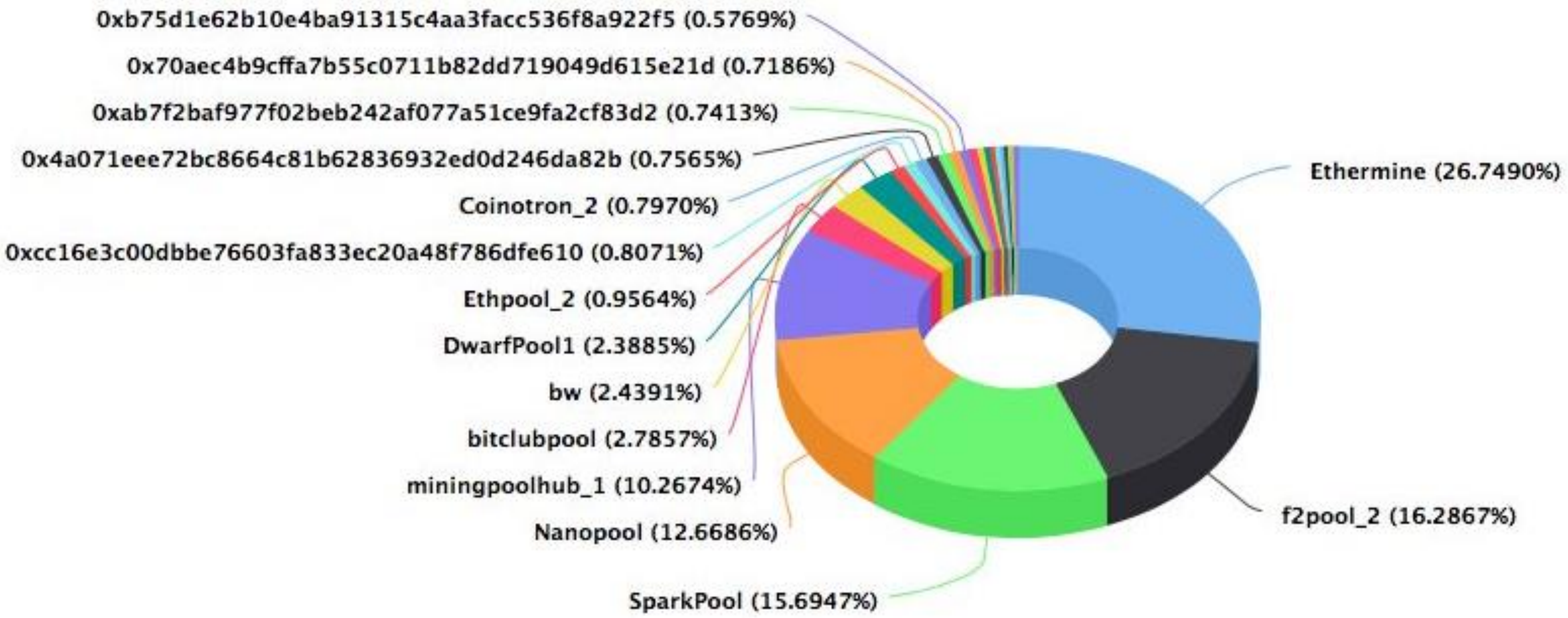
4 mining pools controlling more than 50%



# Decentralization Myth – Ethereum Top 25 Miners

25 distinct nodes

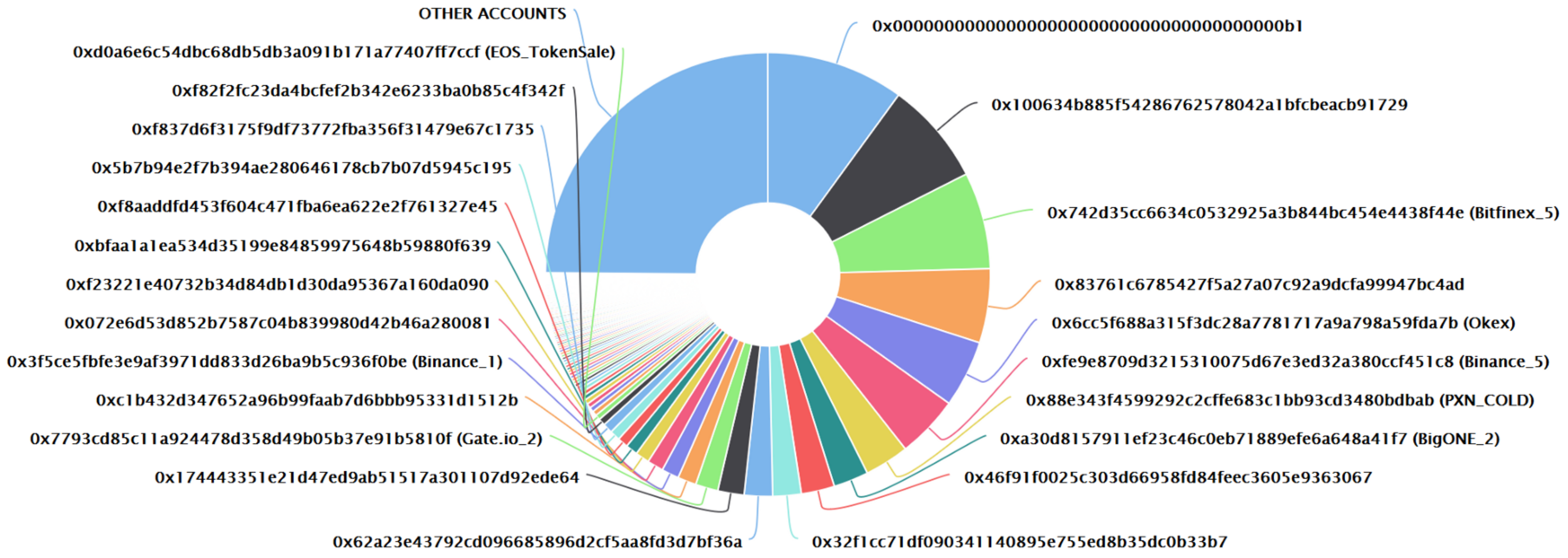
Top 100 accounts controlling 50% of ETH



# Decentralization Myth – EOS top 100 Token Holders

Top 100 accounts controlling 75% of tokens

25 mining nodes that produces blocks



# POS shading major challenge

> *It's really important to mention that validators are super-frequently reshuffled between shards (possibly even once per block), so it's actually quite hard to "target" one specific shard for an attack. This is a large part of where sharding's at least theoretical success in breaking the trilemma comes from.*
>
> *Vitalik Buterin*

…but what about stake centralization?

# Before Scaling Consensus…

We should first decide on the subject of consensus:

1. Government Blockchain (…nonsense)
2. Proof of Authority (1 vote = 1 authority)
3. Proof of Work (1 vote = 1 CPU)
4. Proof of Stake (1 vote = 1 coin)
5. Proof of Storage (1 vote = 1 Kb)
6. Proof of …
7. ???

# Which Resource Is the Most Equally Distributed On the Planet

Hint: definitely it’s not money…

?

# Universal Declaration of Human Rights

### Article 1

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

# Constructing Proof of Identity

?

# Constructing Proof of Identity

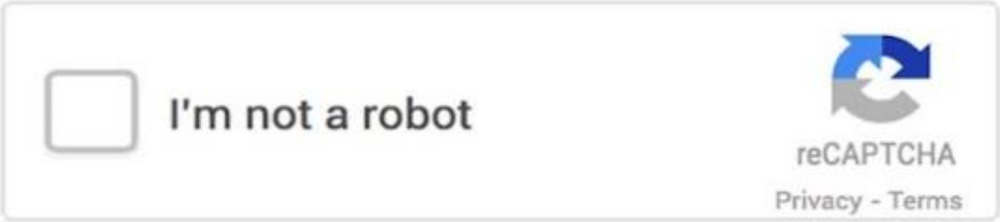
We should NOT do:

1. Global humans registry?
2. DNA based registry?
3. Self-governed identity register?

What we should know about identity:

1. Identity is the human being
2. The human being is unique

# Reverse Turing Test


- Centralized
- Hard to produce for different languages
- Image recognition might be handled with AI

| Hate solving CAPTCHAs | Love solving CAPTCHAs |
| --- | --- |

# Before-After captcha

Before

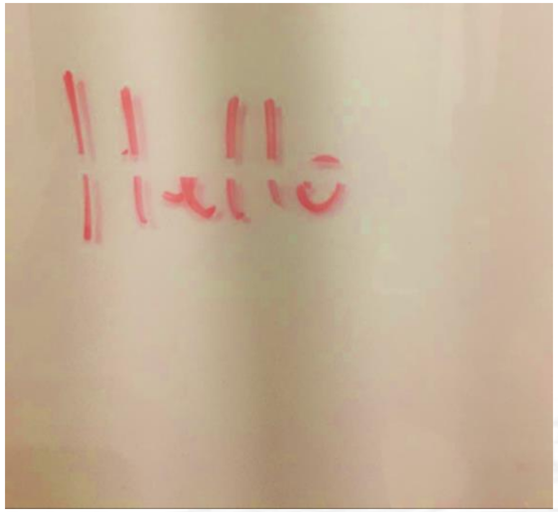
Before

# Before-After captcha

Before

Before

# Before-After captcha


Before

Before

# Before-After captcha

Before

Before

# Before-After captchas

1. Hard for AI. Common sense reasoning is required to solve captcha.
2. Might be created by any person easily
3. The bigger the network the bigger the diversity. Diversity is the key.
4. No translation needed
5. Right answers for captchas are discovered by consensus

# Proof of Identity – Uniqness

Proof of uniqueness - can be achieved by having multiple people during globally synchronized online the session for solving series of captchas

# Proof of Identity – Uniqness

Proof of uniqueness - can be achieved by having multiple people during globally synchronized online the session for solving series of captchas

# Summary

1. Whatever consensus algorithm is built it should care about decentralized foundation
2. Deep decentralization is the basis for the safe scalability (sharding)
3. There is no trilema, but dilemma only – to be decentralized or not to be
4. Anonymous digital identity might be the possible solution

# Universal Declaration of Human Rights – 10.12.1948

## Article 22

Everyone, as a member of society, has the right to social security and is entitled to realization, through national effort and international co-operation and in accordance with the organization and resources of each State, of the economic, social and cultural rights indispensable for his dignity and the free development of his personality.

# Universal Declaration of Human Rights – 10.12.1948

## ARTICLE 12

No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

# Universal Declaration of Human Rights – 10.12.1948

### Article 7

All are equal before the law and are entitled without any discrimination to equal protection of the law. All are entitled to equal protection against any discrimination in violation of this Declaration and against any incitement to such discrimination.

Thank you