Clubs, Coins, and Crowds: Fairness and Decentralization in Blockchains and Cryptocurrencies

Prof. Bryan Ford
Decentralized/Distributed Systems (DEDIS)

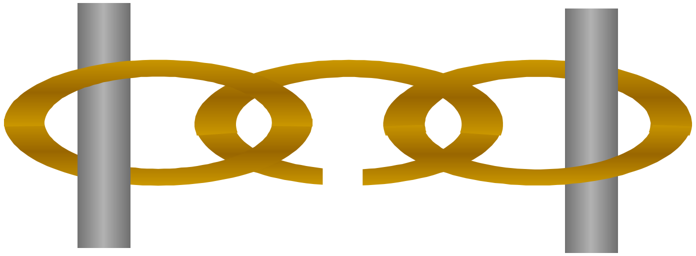
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CDMDSW – June 20, 2018

A Fundamental Problem

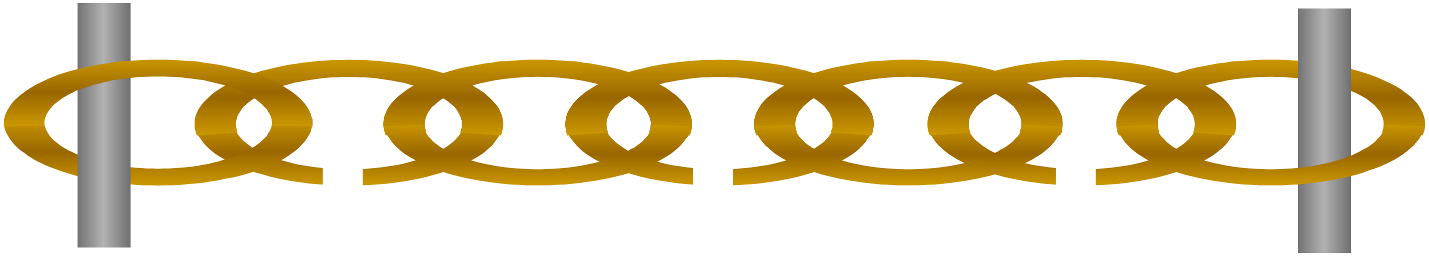
In today’s IT systems, security is an afterthought

- Designs embody “weakest-link” security



Scaling to bigger systems → weaker security

- Greater chance of any “weak link” breaking

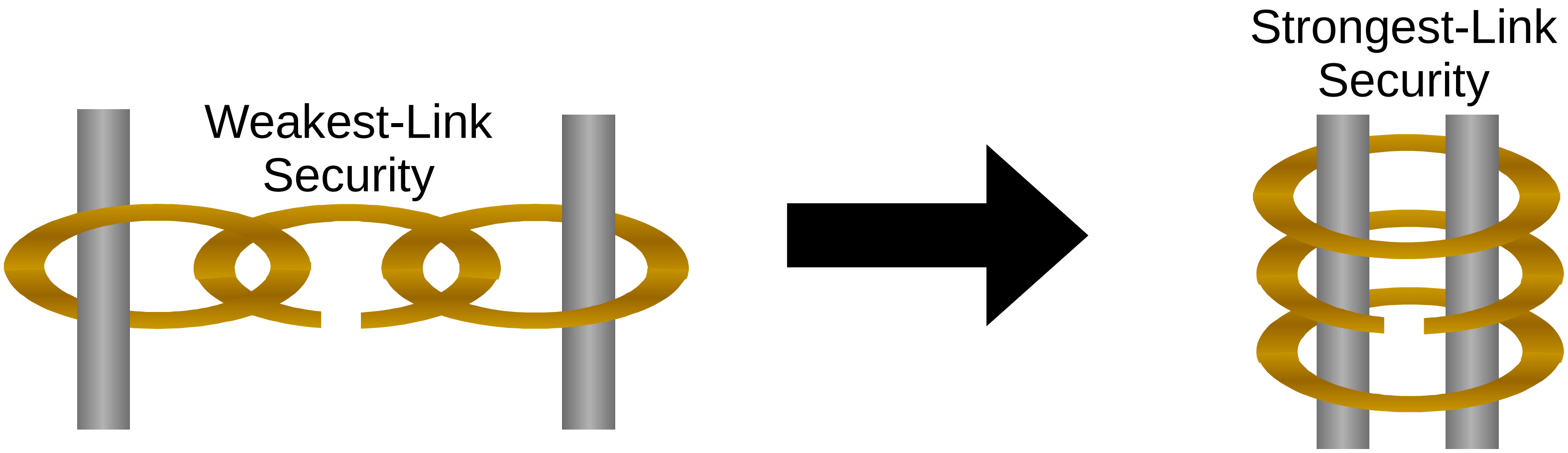


The DEDIS lab at EPFL: Mission

Design, build, and deploy secure privacy-preserving
Decentralized and Distributed Systems (DEDIS)

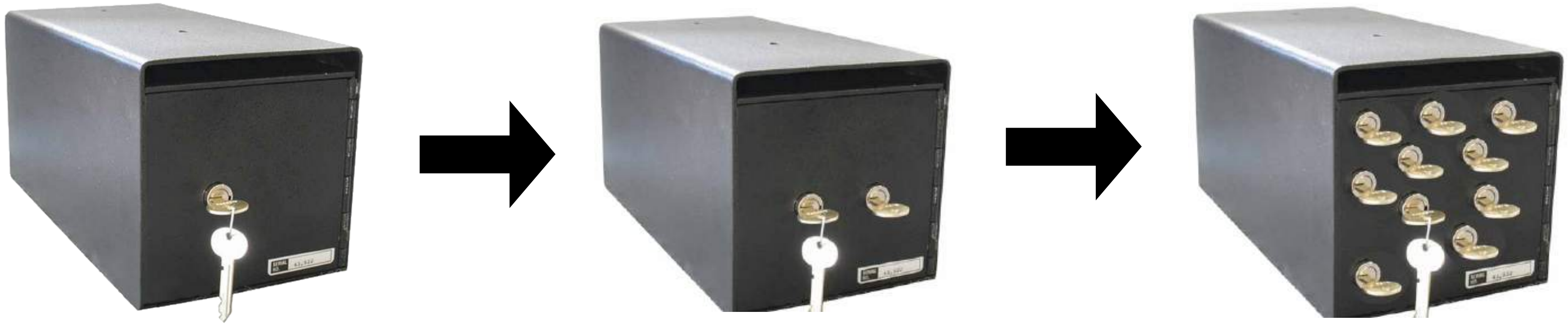
- **Distributed:** spread widely across the Internet & world
- **Decentralized:** independent participants, no central authority, *no single points of failure or compromise*

Overarching theme: building decentralized systems
that **distribute trust** widely with **strongest-link security**



Turning Around the Security Game

Design IT systems so that making them bigger makes their security *increase* instead of *decrease*



Weakest-link security

Strongest-link security

Scalable Strongest-link security

DEDIS Laboratory Members

Bryan Ford
Associate Professor

Philipp Jovanovic
Postdoctoral Scholar

Stevens Le Blond
Research Scientist

Linus Gasser
Software Engineer

Jeff R. Allen
Software Engineer

Kelong Cong
Software Engineer

Lefteris Kokoris-Kogias
Ph.D. Student

Kirill Nikitin
Ph.D. Student

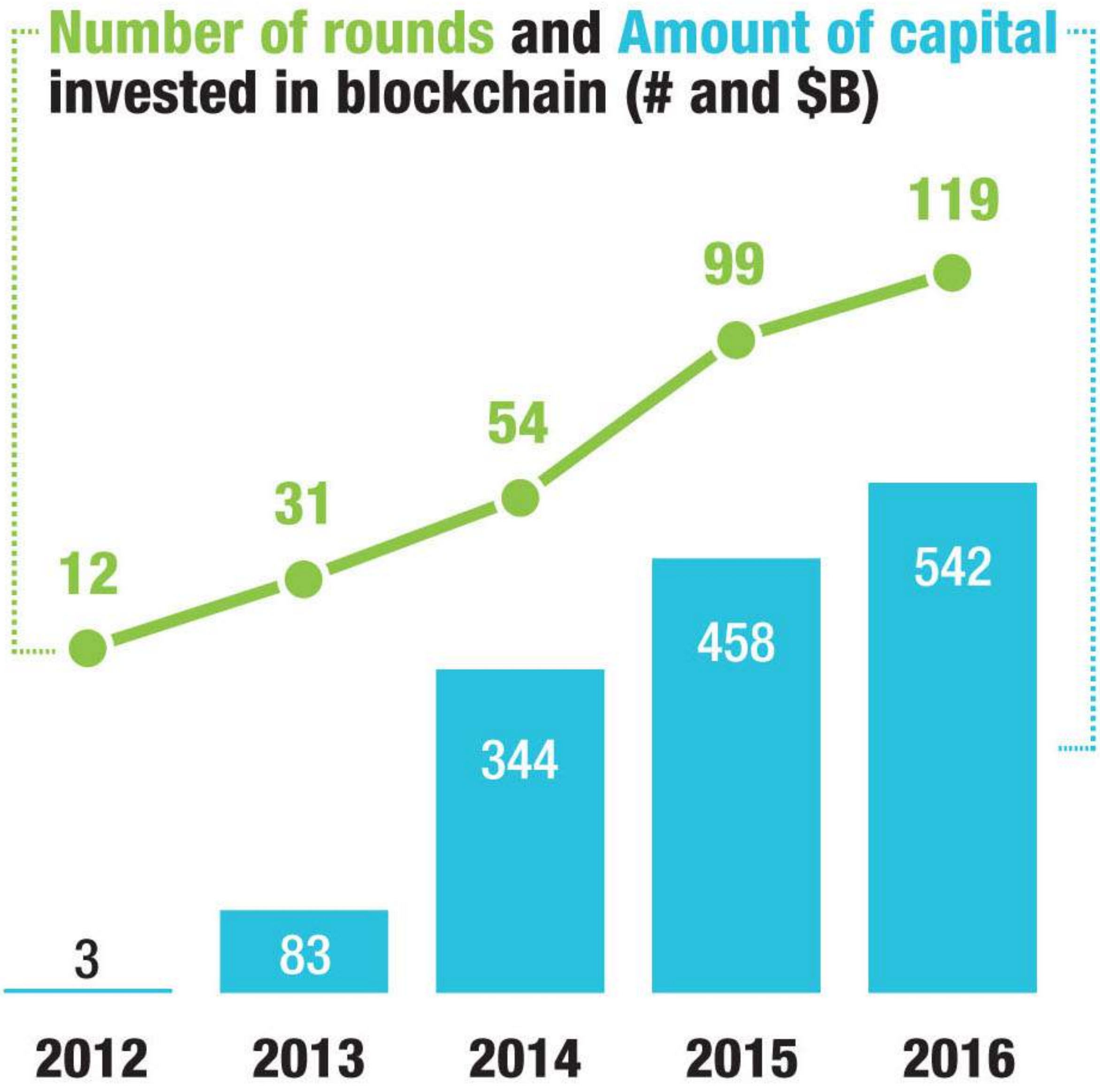
Cristina Basescu
Ph.D. Student

Nicolas Gailly
Ph.D. Student

The Call of the Blockchain

(credit: Tony Arcieri)

Broad Promise & Global Interest



There is a **decreasing tendency towards launching new blockchain companies:**

2016	**169**
2015	**221**
2014	**233**
	new companies launched

There is an **increase in investment rounds:**

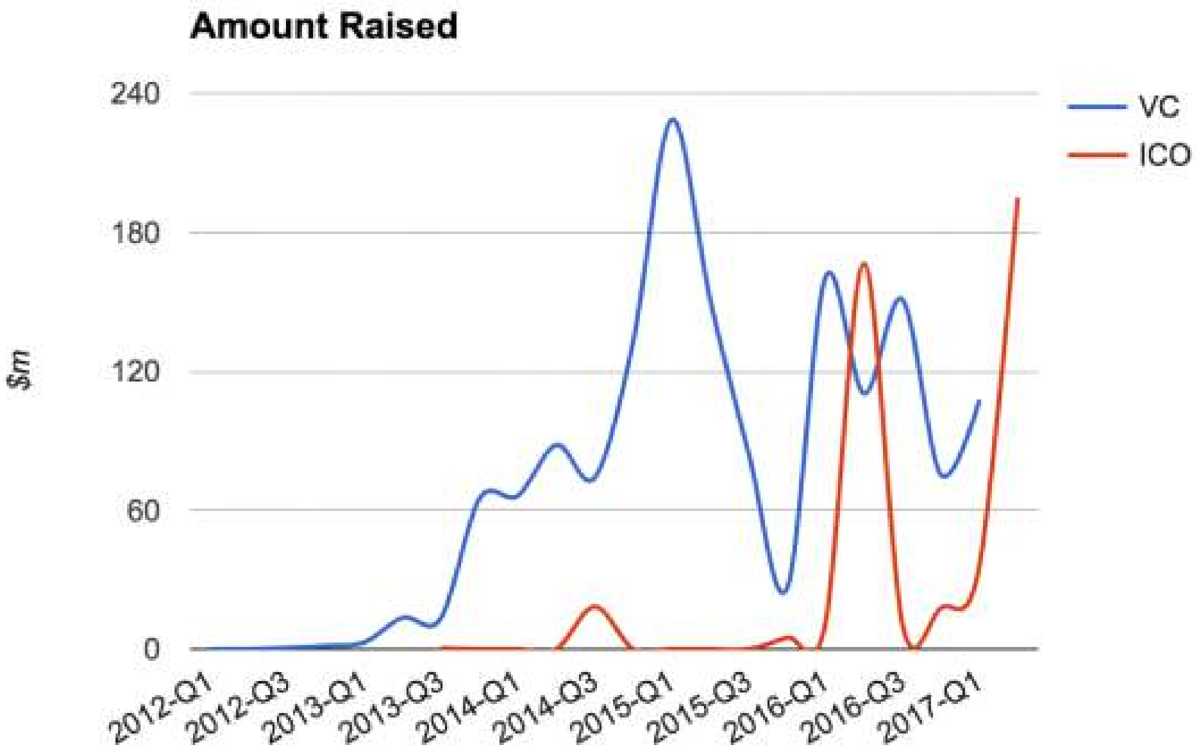
2016	**119**
2015	**99**
2014	**54**
	rounds

Source: Money of the Future, Life.SREDA

And a “new” form of investment...

ICOs: “Initial Coin Offerings”

- Digital tokens representing digital goods and services *yet to be created…*



Data sourced from CoinDesk, Smith & Crown, VentureSource, VentureScanner, and CrunchBase. Amounts raised valued in USD at time of sale close. Data may be incomplete.

Bitcoin (2008)

First successful decentralized cryptocurrency…

Bitcoin (2008)

First successful decentralized cryptocurrency…

...and a fascinating study in **seductively wrong** answers to key issues in decentralized systems

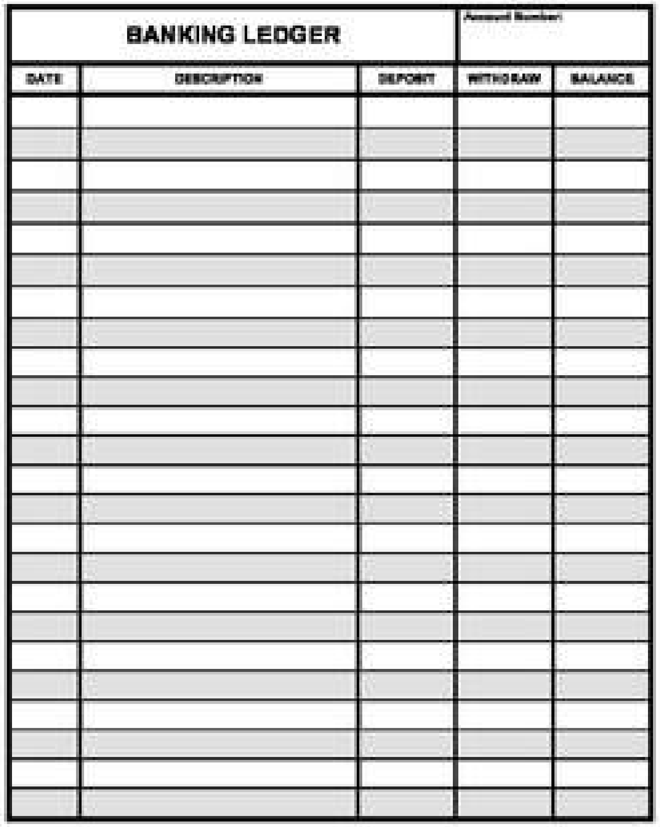
How to track wealth (or anything)?

Things

- Gold, beads, cash...

Ledgers

- Who owns what?



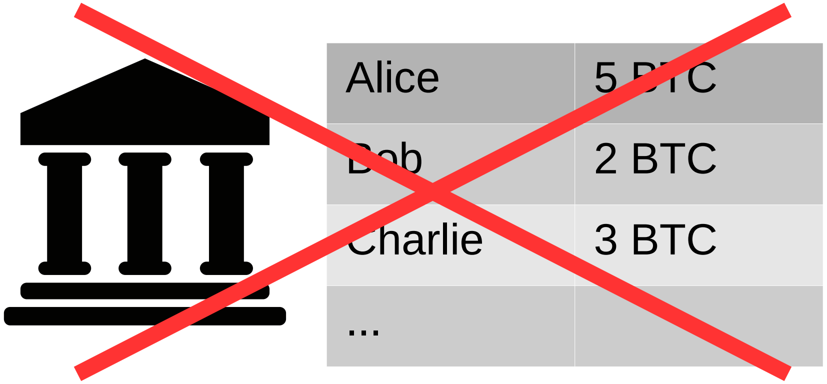
BANKING LEDGER

Account Number:

DATE	DESCRIPTION	DEPOSIT	WITHDRAW	BALANCE

Distributed Ledgers

Problem: we don't want to trust any designated, centralized authority to maintain the ledger



Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Solution: "everyone" keeps a copy of the ledger!

- Everyone checks everyone else's changes to it

Alice's copy

Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

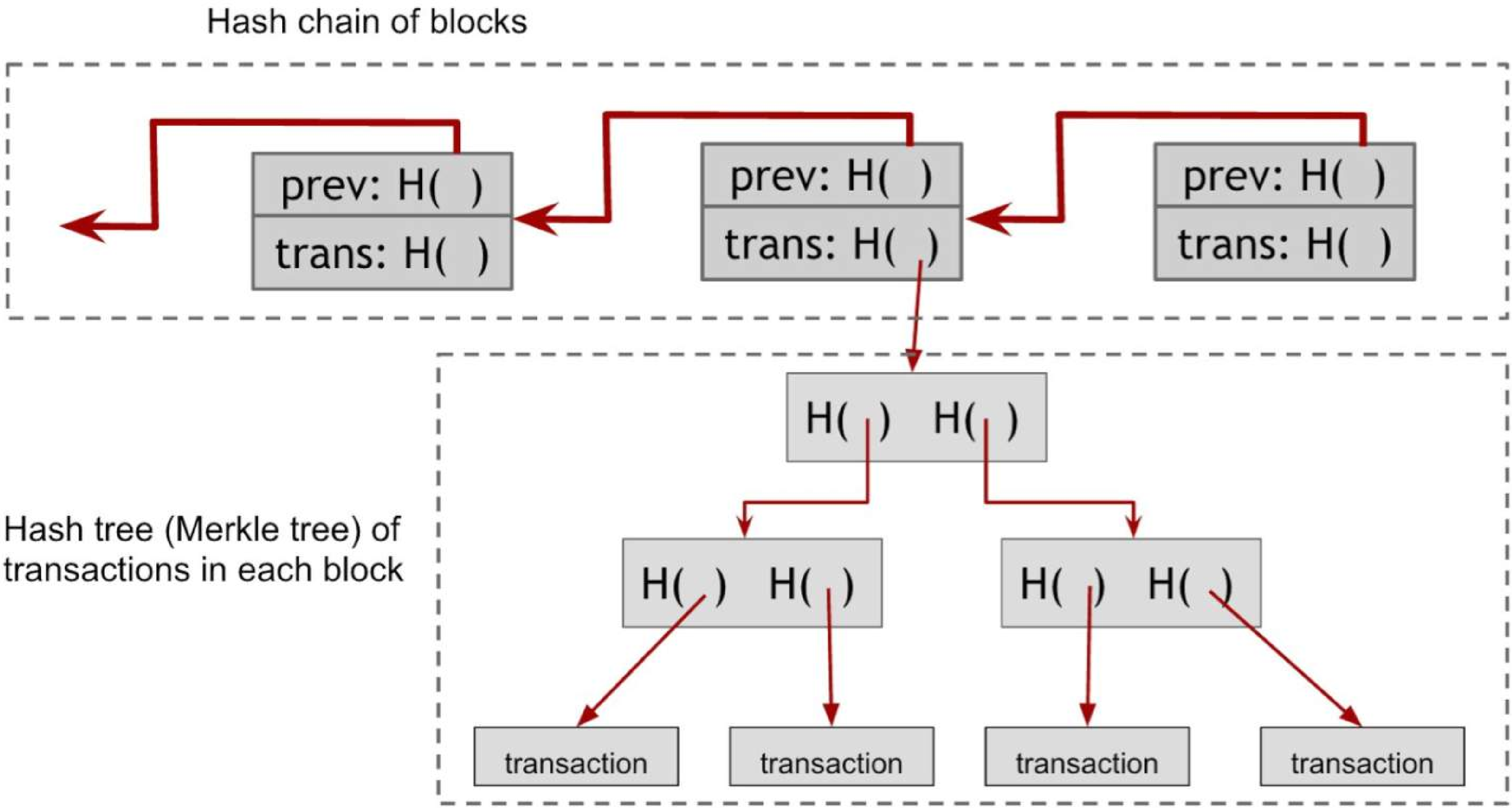
Bob's copy

Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Charlie's copy

Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Example: The Bitcoin Blockchain



Applications of Distributed Ledgers

Can represent a distributed electronic record of:

- Who owns how much **currency**? (Bitcoin)
- Who owns a name or a digital work of art?
- What are the terms of a **contract**? (Ethereum)
- When was a **document** written? (notaries)

- What is the **provenance** of a part? (supply chain)
- Who **are** you? (self-sovereign identity)
- Who used **data** for what purpose? (access logs)
- …

But… Today’s Blockchains Suck

Public/permissionless (e.g., Bitcoin, Ethereum)

- Weak probabilistic consistency
- Long transaction delays, low throughput
- Clients must be online, well-connected to follow
- Mining is inefficient, insecure, re-centralizing

Private/permissioned (e.g., HyperLedger, R3, …)

- Weak security – single points of compromise

Talk Outline

- Key challenges in decentralized systems, and partial solutions to some of them
  - Scalable secure coordination
  - Membership and fairness
  - Governance and incentives
- Conclusion: democratic decentralization?

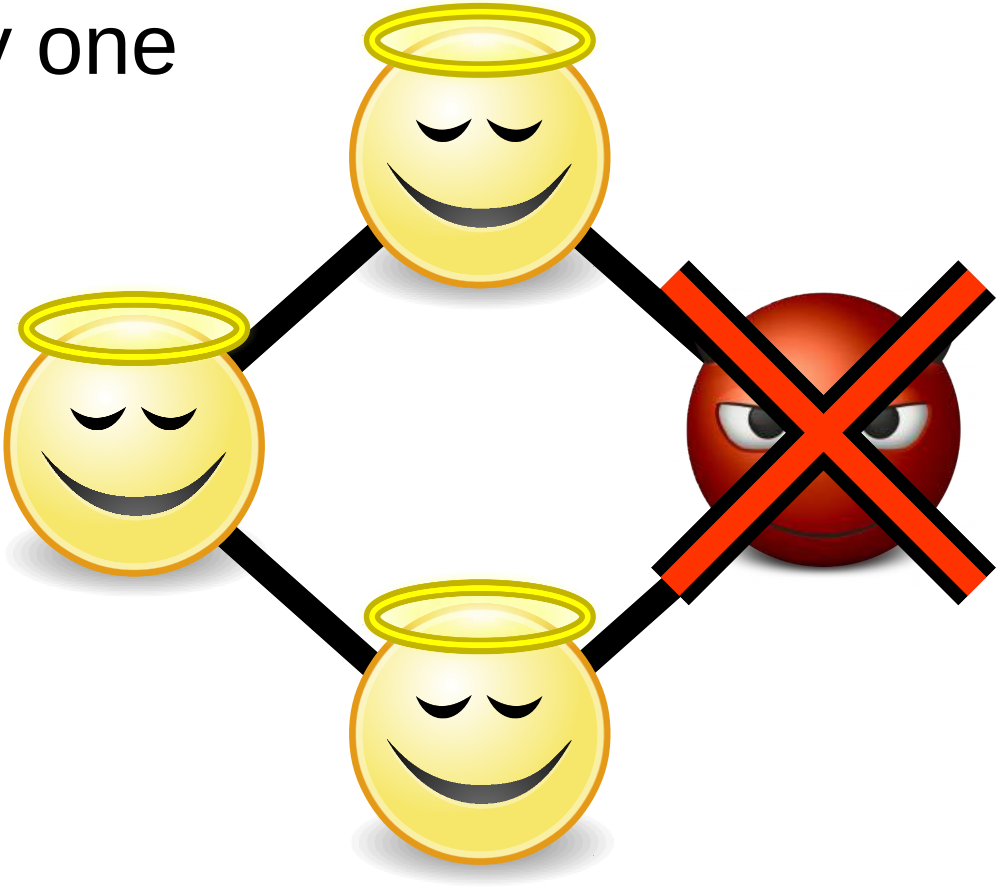
Talk Outline

- Key challenges in decentralized systems, and partial solutions to some of them
  - **Scalable secure coordination**
  - Membership and fairness
  - Governance and incentives
- Conclusion: democratic decentralization?

The Distributed Trust Principle

Many algorithms allow us to **distribute trust** among multiple (preferably independent) parties

Work correctly despite any one (or several) participants being compromised, maliciously colluding



Example algorithms:

- Byzantine consensus
- Threshold cryptography (signing, encryption, …)

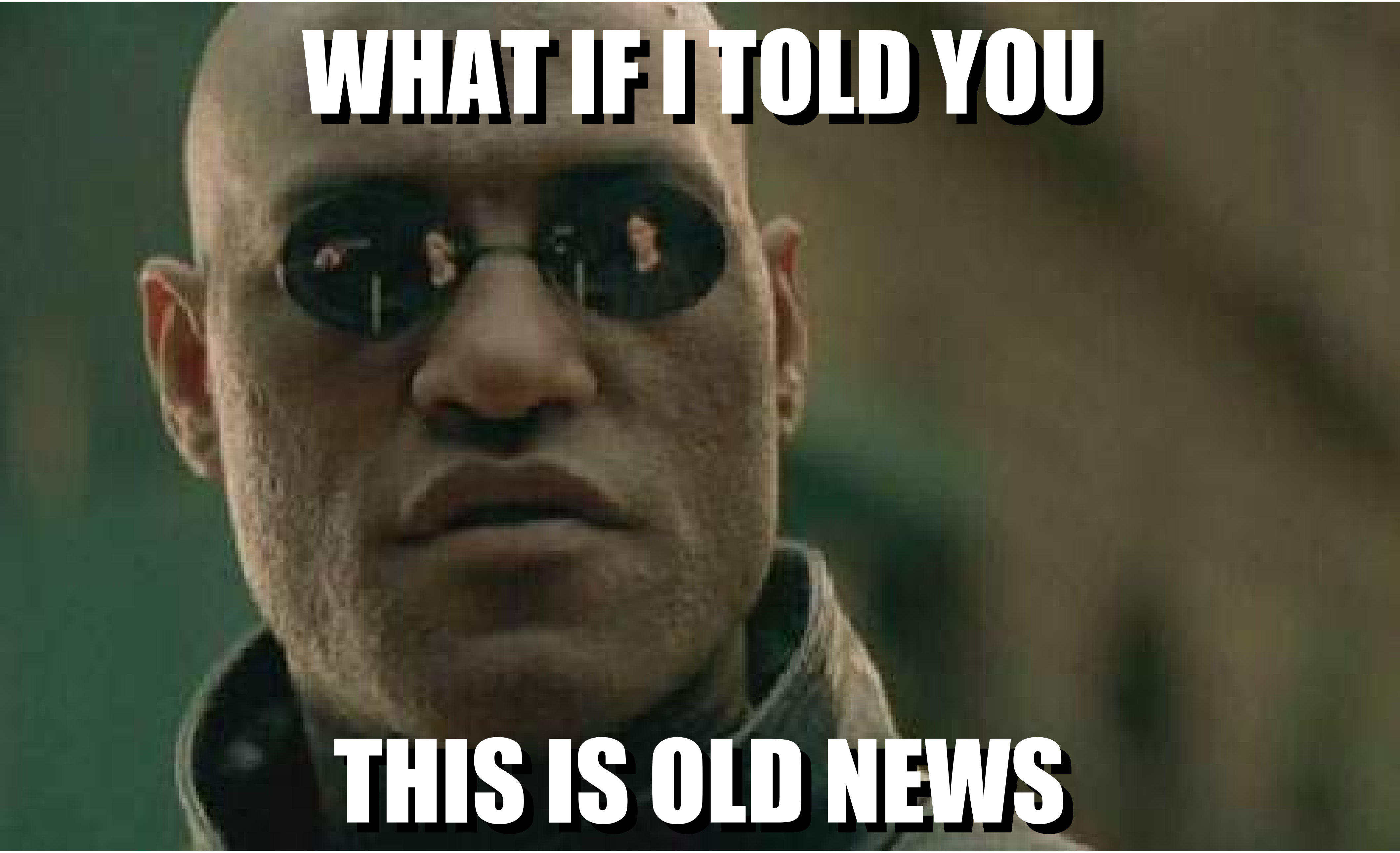
The Distributed Trust Principle

Many algorithms allow us to **distribute trust** among multiple (preferably independent) parties

Work correctly despite any one (or several) participants being compromised, maliciously colluding

Example algorithms:

- Byzantine consensus
- Threshold cryptography (signing, encryption, …)


WHAT IF I TOLD YOU
THIS IS OLD NEWS

Bitcoin’s Key Technical Innovation

Build a Byzantine consensus protocol:

- **Open** to anyone wishing to participate
- **Scalable** to thousands of participants or more

In the process, Bitcoin’s architecture conflates the problems of **membership** and **consensus**

- Resulting in many technical limitations and massive confusion among blockchain fans
  - e.g., PoW is about *membership*, not *consensus*

Nakamoto Consensus

Public blockchains such as Bitcoin, Ethereum use consensus by **crypto-lottery**

1) **Miners** print their own “lottery tickets” by solving crypto-puzzle (**proof-of-work**)

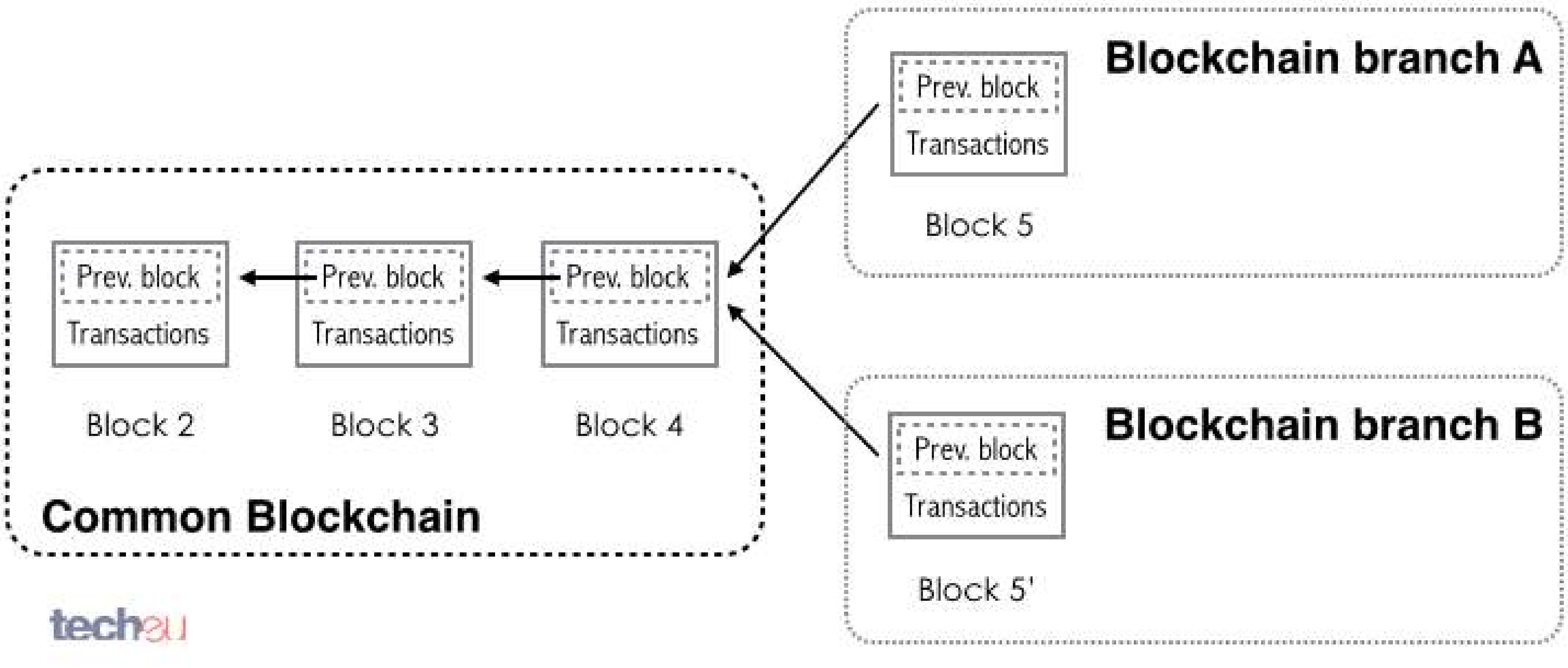
2) Winner gets to add one **block** to blockchain; typically gets **reward**: e.g., print new money

3) All miners gravitate to **longest chain.** Repeat.

Consensus is only probabilistic

If two miners win at **about the same time,**

the blockchain **forks**:

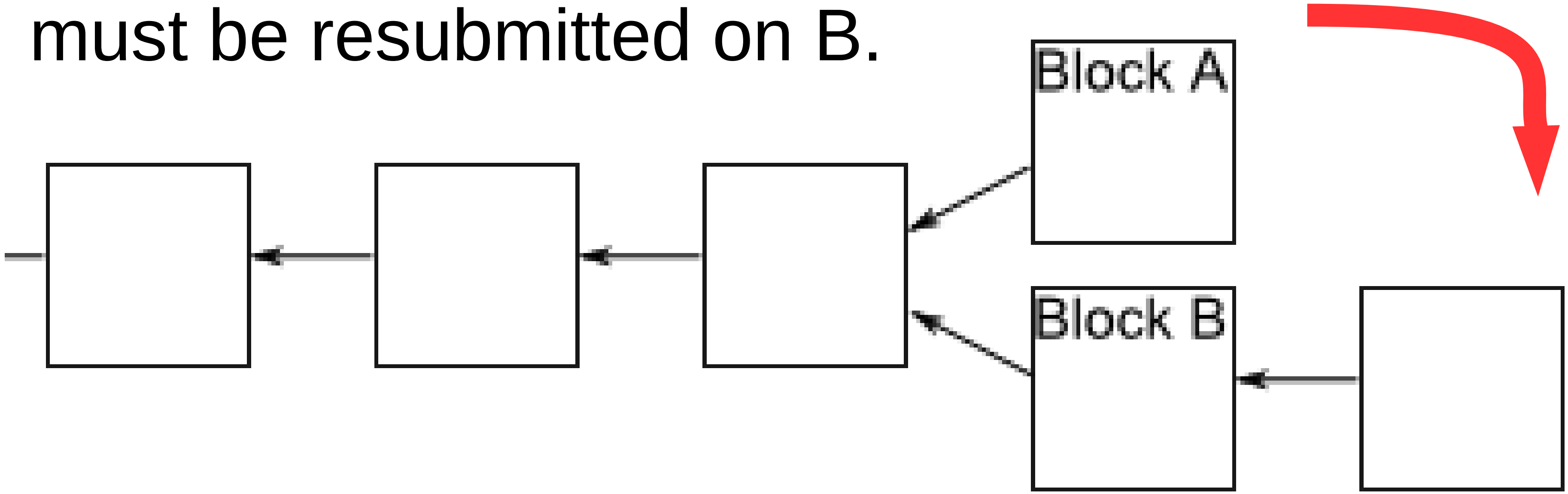


Resolving Temporary Forks

Example:

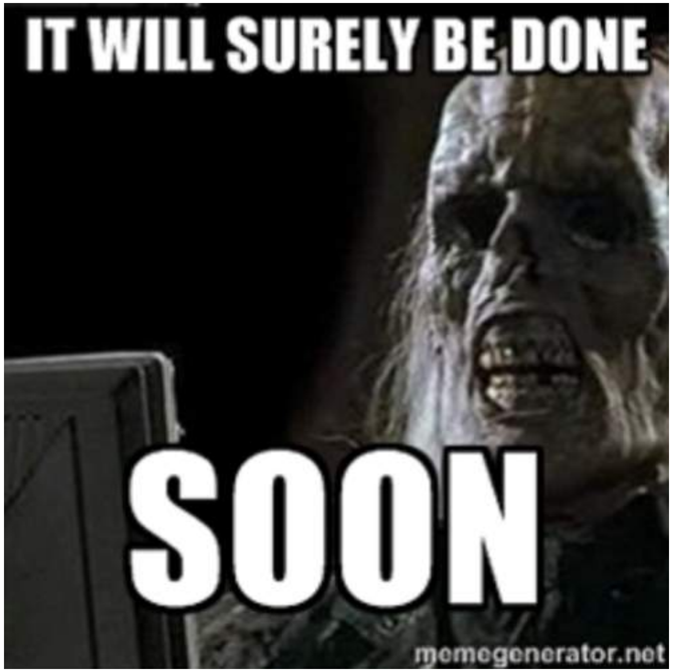
As soon as miner “wins” a ticket to extend B, miners on block A “jump ship” to B's history.

- Any transactions only appearing in block A will “disappear from history,” must be resubmitted on B.



Drawbacks of Nakamoto Consensus

- **Transaction delay**
  - Any transaction takes ~10 mins *minimum* in Bitcoin
- **Weak consistency:**
  - You’re not *really* certain your transaction is committed until you wait ~1 hour or more
- **Low throughput:**
  - Bitcoin: ~7 transactions/second
- **Proof-of-work mining:**
  - Wastes huge amount of energy

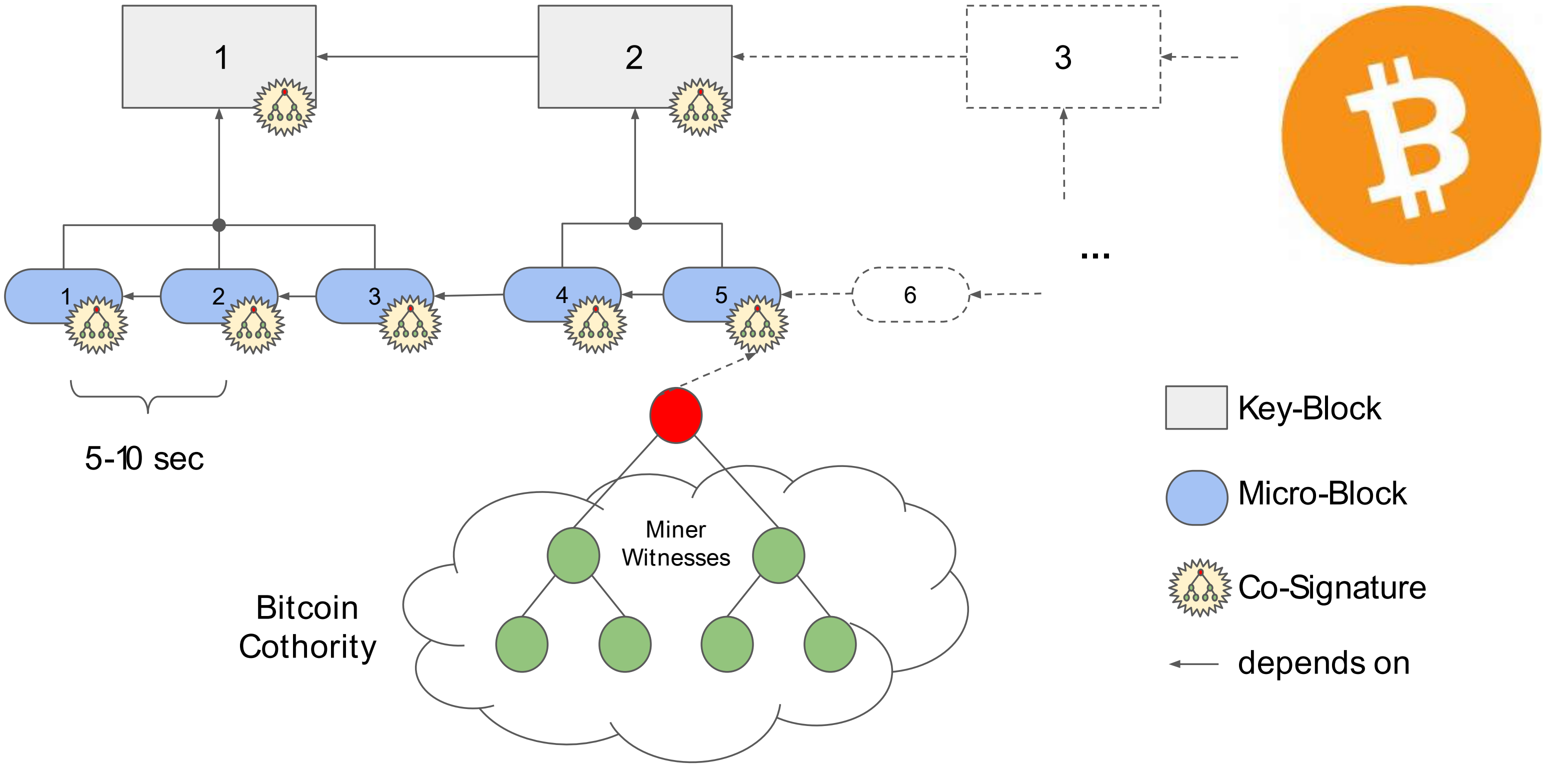


Scaling Blockchains is Not Easy

ByzCoin: Fast, Scalable Blockchains

DEDIS lab project presented in [USENIX Security ‘16]

- Permanent transaction commitment in seconds
- 700+ TPS demonstrated (100x Bitcoin, ~PayPal)



Byzantine Consensus for Blockchains

PBFT: “Practical Byzantine Fault Tolerance”

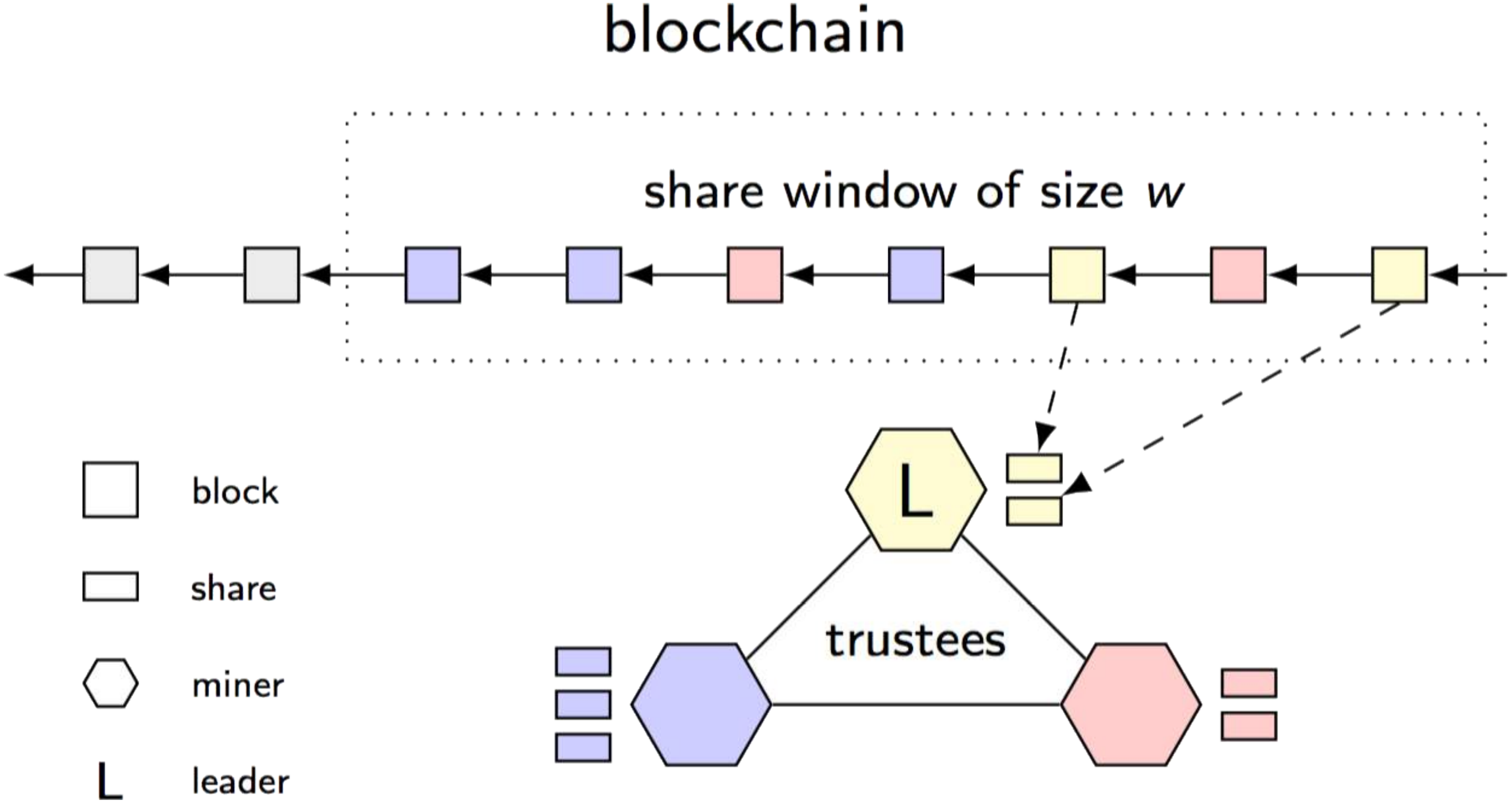
- Castro/Liskov ‘99 – mature, many refinements

Not directly suitable to permissionless blockchains

1) PBFT assumes closed consensus group, Bitcoin mining is in principle “open to all”
2) PBFT implementations assume small groups: typically tested with n=4, never more than ~15; Bitcoin has 1000s of miners, maybe 100k

ByzCoin Consensus Windows

Keeps Bitcoin’s proof-of-work (PoW), but mining yields **temporary membership share** in a gradually-rotating consensus group

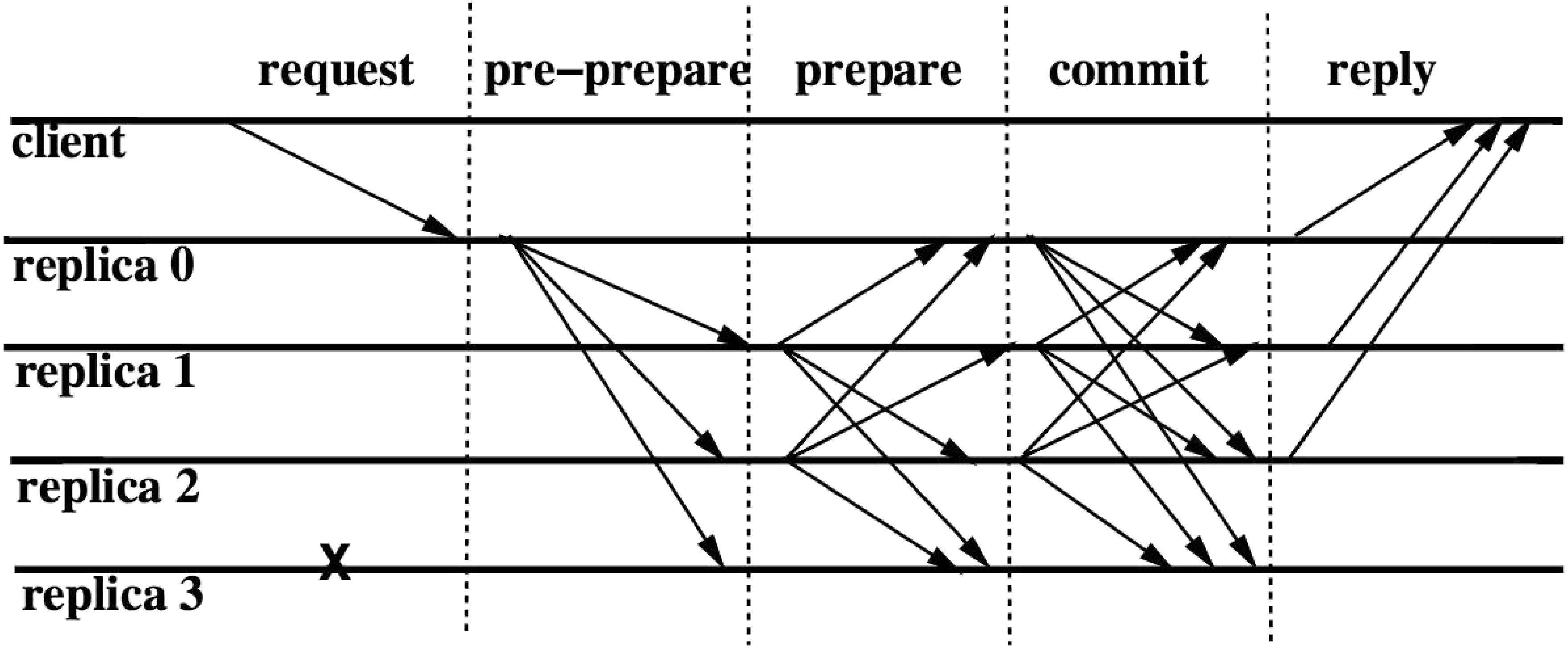


Why PBFT Doesn't Readily Scale

Three phase: pre-prepare, prepare, commit

In prepare & commit, leader must get at least two-thirds of all participants to "sign-off"

- Nodes sign-off via broadcast: O(N^2)



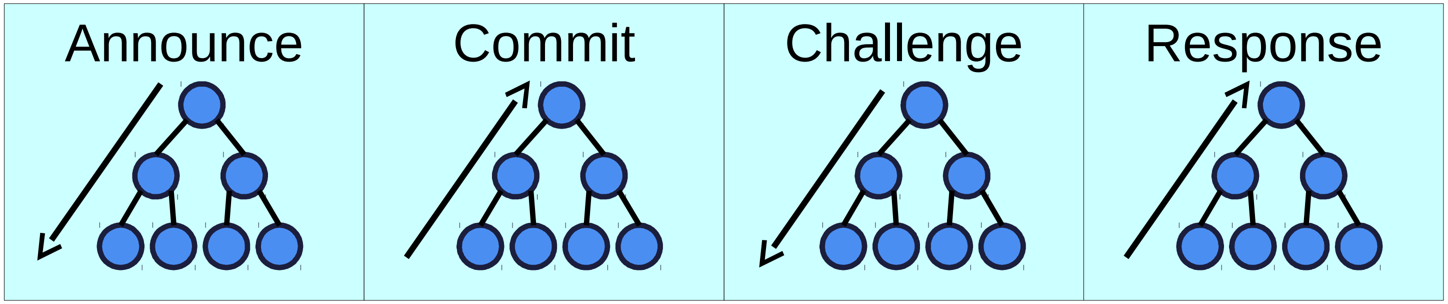
PBFT with Collective Signing (CoSi)

Builds on CoSi, presented in [IEEE S&P '16]

ByzCoin runs **collective signing** (CoSi) rounds to implement PBFT prepare, commit phases

- Efficient tree-structured communication
- Sign-offs compressed into 1 signature

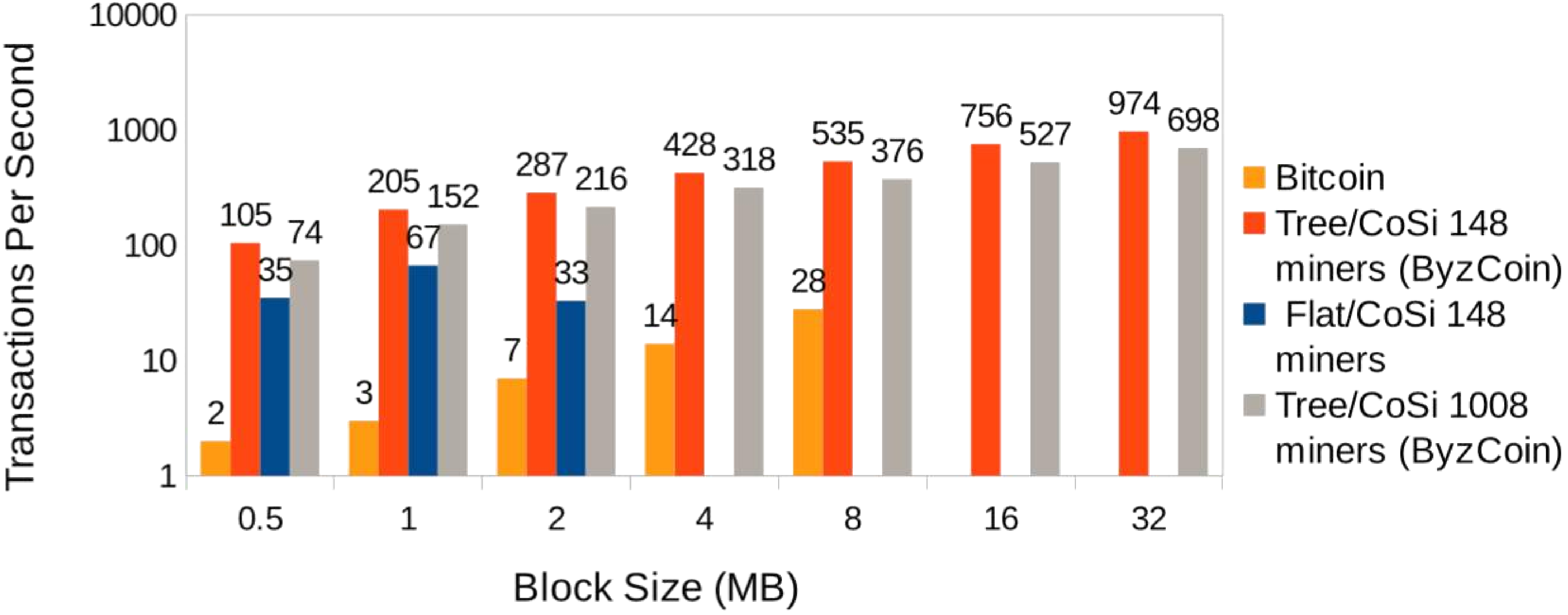
Reduce round cost from O(N^2) to ~O(N)



ByzCoin transaction throughput

~100x improvement for similar block size

- higher throughput than PayPal
- scales to >1000 consensus peers



Next Problem: Horizontal Scaling

Most blockchains require *each* miner or validator to **replicate all state** and **verify all transactions**

Therefore:

- Each stores all of a constantly-growing history
- Adding participants does not increase capacity

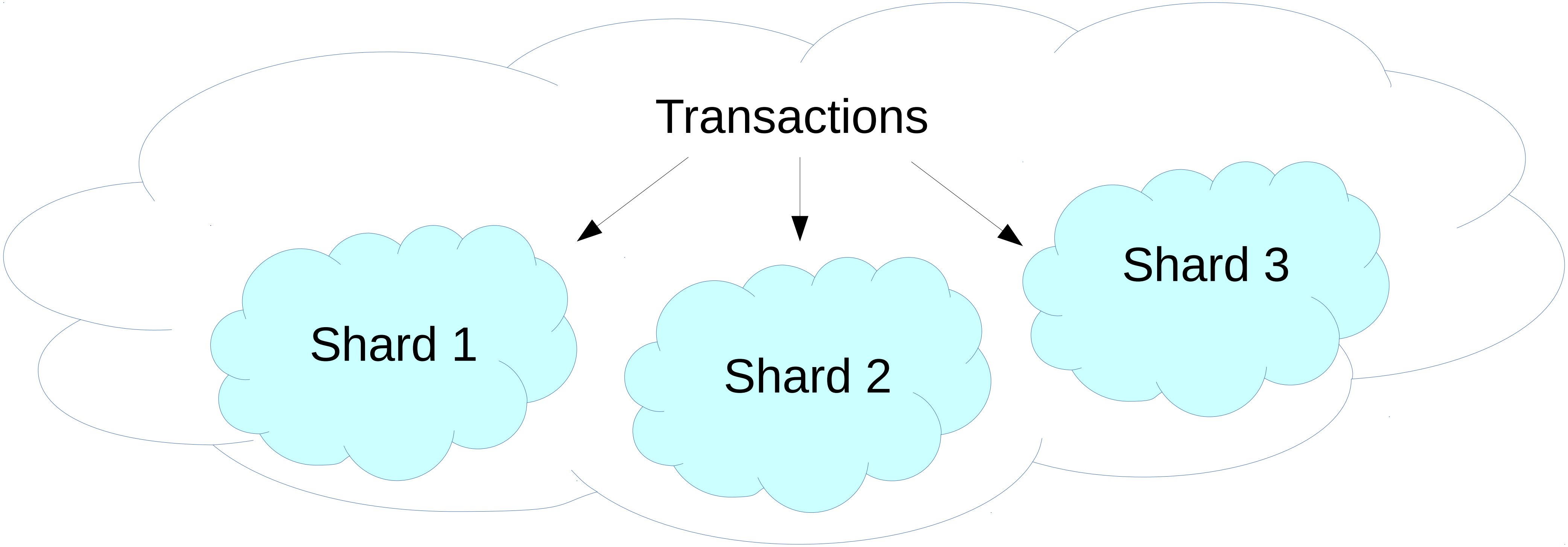
Not really scalable in either storage or throughput

Horizontal scaling: more nodes → more capacity

Horizontal Scaling via Sharding

OmniLedger: A Secure Scale-Out Ledger [S&P 18]

- Break large collective into smaller subgroups
- Builds on scalable bias-resistant randomness protocol (IEEE S&P 2017)
- 6000 transactions/second: competitive with VISA



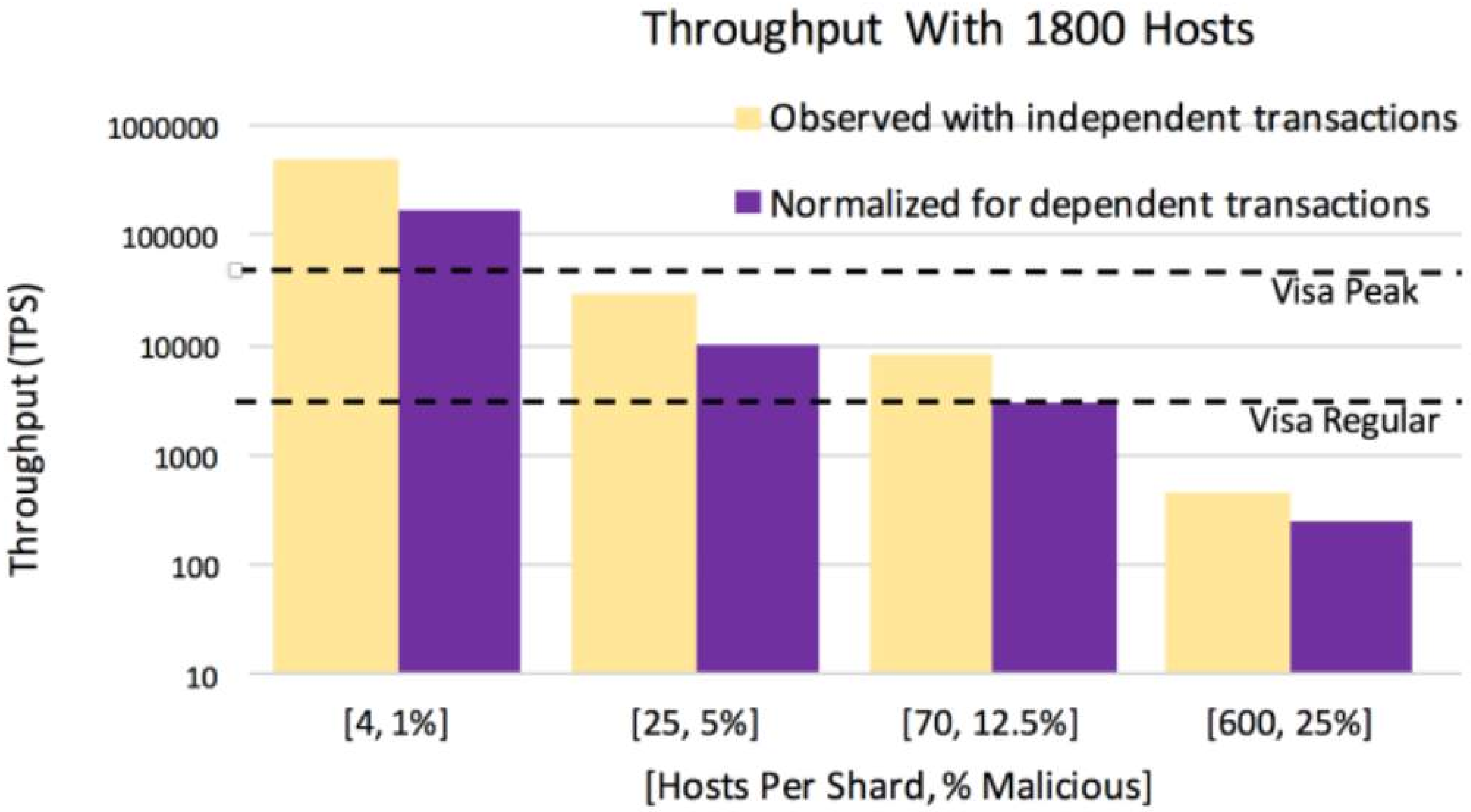
OmniLedger: Key Intuition

At any time a (possibly slow) consensus process maintains *large* (~1000s) list of miners/validators

- Uses RandHound/RandHerd to form smaller (10s, 100s) representative subgroups or *shards*
  - Subgroup size is security/performance tradeoff
  - Periodically re-form shards as network evolves
- Each shard manages subset of state (accounts)
- Transactions processed by one or a few shards
  - Typically one shard per account transaction affects
  - Inter-shard commit protocol ensures consistency

OmniLedger Throughput

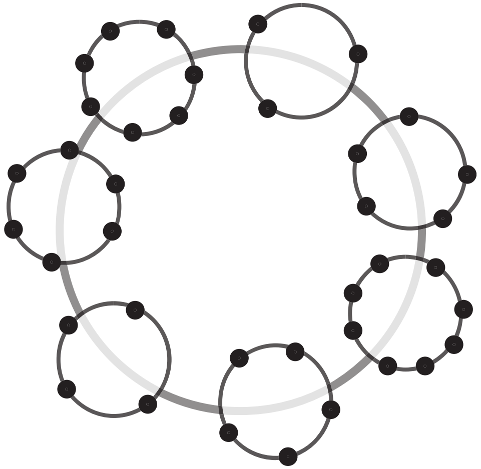
Wide range of performance/security settings



Problem: Unbiased Public Randomness

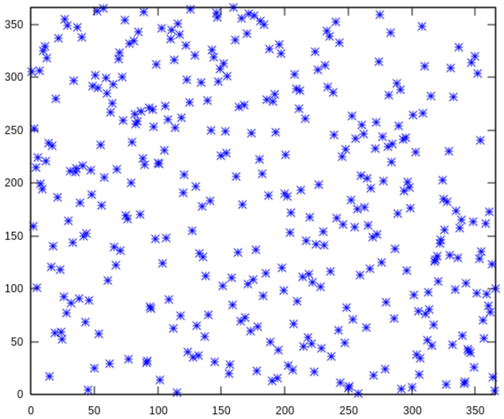
For many purposes we need to "flip coins" in public, convince everyone result is **fair** and **unbiased**.

- Choose a lottery winner fairly and transparently
- Fair sampling: e.g., risk-limiting audits of elections
- Pick representative quorums from large pools
  - e.g., for secure blockchain sharding
- Divide large user network into smaller random anonymity sets
  - e.g., Herbivore [Goel/Sirir '04]



Secure Public Randomness is Hard

Vietnam War Lotteries (1969)



'European draws have been rigged': Ex-FIFA president Sepp Blatter claims to have seen hot and cold balls used to aid cheats

Former FIFA president Sepp Blatter said he had witnessed rigged draws for European foo
competitions

Man hacked random-number generator to rig lotteries, investigators say

New evidence shows lottery machines were rigged to produce predictable jackpot numbers on specific days of the year netting millions in winnings

'Computer whiz' rigged lottery number generator to produce predictable numbers a couple of times a year

Strawman 1: Commit-and-Reveal

1. Each of *n* nodes pick a random secret s_i, broadcast a commit to secret, e.g., $C_i = \mathrm{H}(s_i)$
2. "Everyone" reveals their secrets s_i, combines to form final output, e.g., $s = \Sigma_\mathrm{i}(s_i)$

Problem: vulnerable to either DoS or bias attacks

- Require *everyone* to reveal → DoS attacks
- Tolerate up to *f* missing secrets → attacker can choose favorite of 2^f outcomes

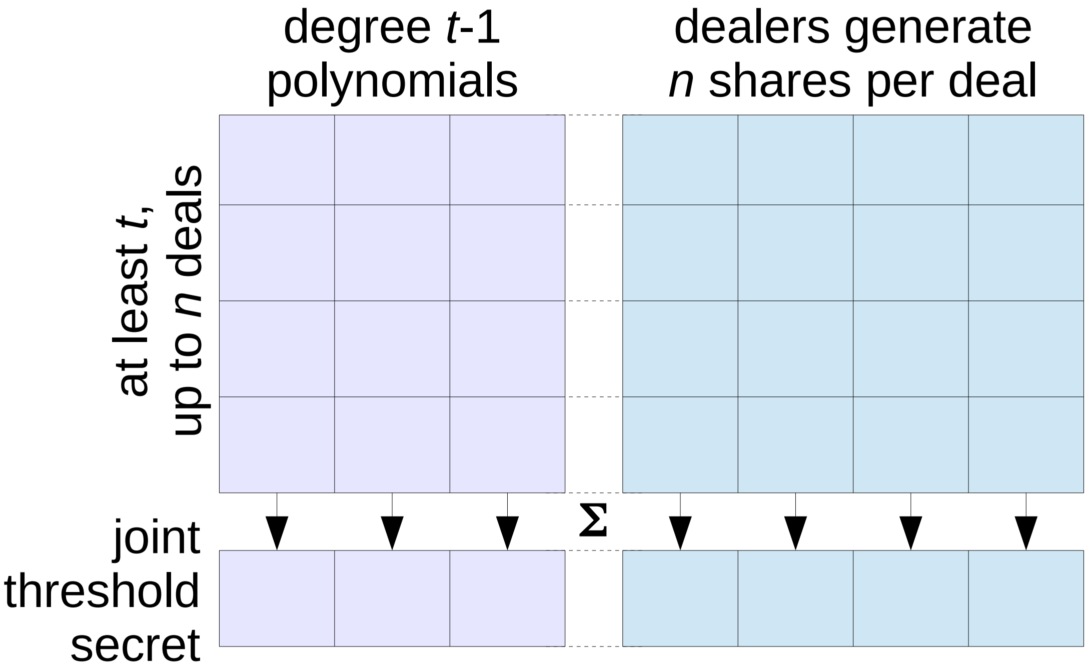
Strawman 2: Shamir Secret Sharing

- Each of *n* nodes “deals” secret s_i all *n* nodes via *t*-of-*n* publicly verifiable secret sharing (PVSS)
- Agree (BFT) on at least *t* of these secret deals
- Homomorphically sum polynomials and reveal

Works, secure! ☺

- [Cachin et al, …]

O(n^2) comm.,
O(n^3) compute ☹



degree *t*-1
polynomials

dealers generate
n shares per deal

at least *t*,
up to *n* deals

Σ

joint
threshold
secret

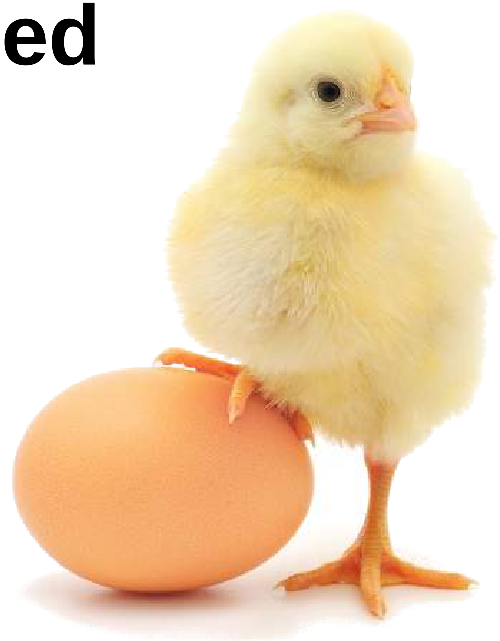
The Chicken-and-Egg Problem

More scalable if we could use *smaller groups*…
but need randomness to *sample* them securely!

- Sharding needs randomness needs sharding

Addressed by **RandHound**, **RandHerd** protocols

- **Scalable Bias-Resistant Distributed Randomness** [IEEE S&P ‘17]
- **RandHound**: bootstrap protocol, O(*n* log *n*) efficiency
- **RandHerd**: repeating beacon, O(log *n*) cost/node/round



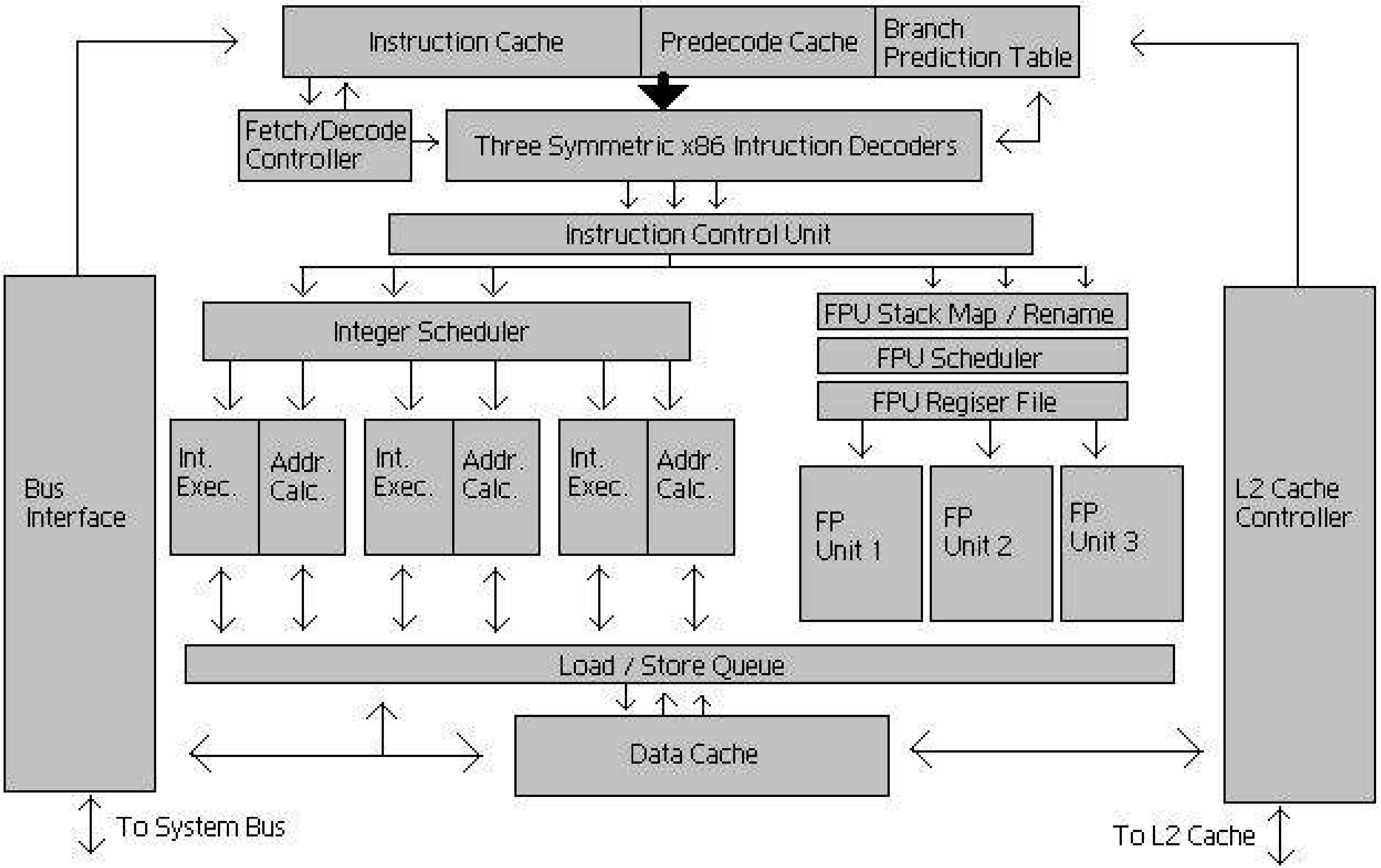
What’s Next in Blockchain Scaling?

Many interesting future directions, such as:

- Special-purpose shards for greater functionality
  - Example: public randomness shard (RandHound)
  - Example: on-chain secret caretaking (SCARAB)
- Locality-preserving shards to reduce latency
- Blockchains for edge networks?
  - Sensor data management, sharing, privacy, …

Towards General-Purpose Scalable Decentralized Computing

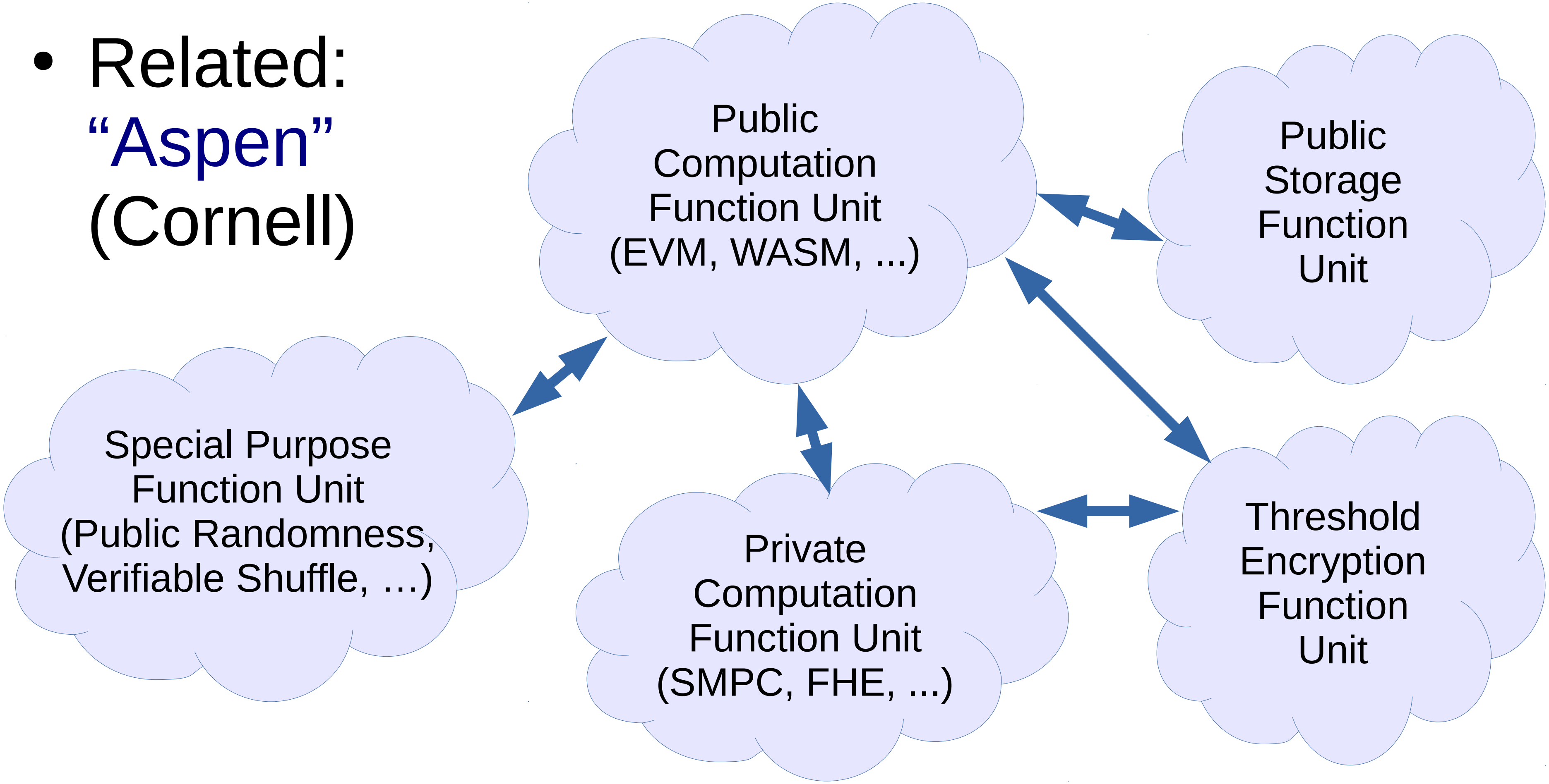
Analogy: CPUs now composed of many special-purpose functional units...



Towards General-Purpose Scalable Decentralized Computing

Goal: build scalable **decentralized** architecture

- Ecosystem of anytrust/threshold “function units”
- Related: “Aspen” (Cornell)

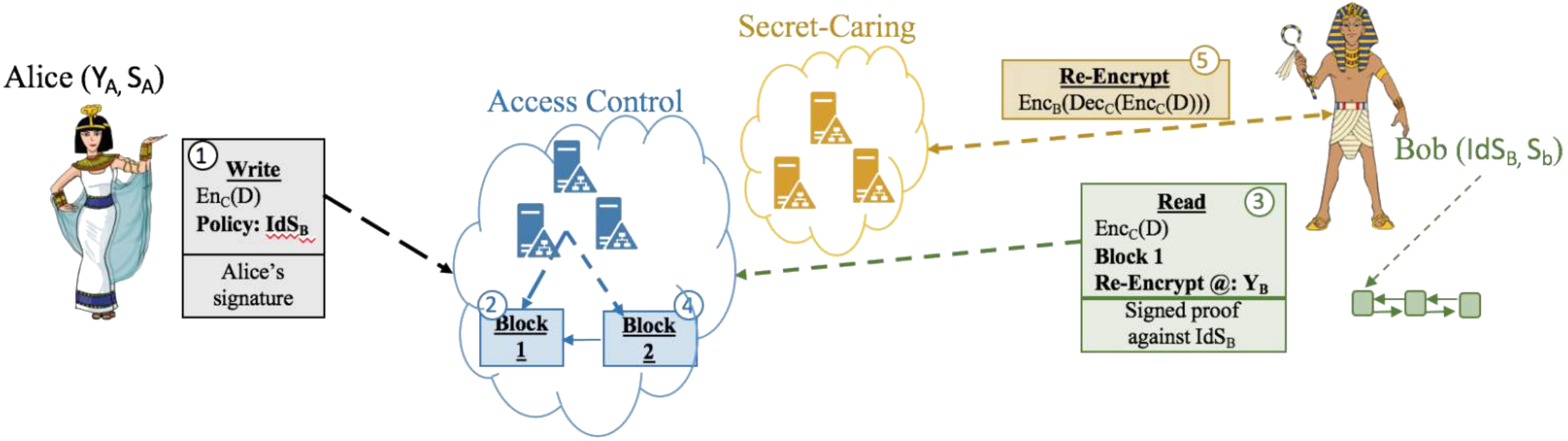

Public Computation Function Unit (EVM, WASM, ...)
Public Storage Function Unit
Special Purpose Function Unit (Public Randomness, Verifiable Shuffle, …)
Private Computation Function Unit (SMPC, FHE, ...)
Threshold Encryption Function Unit

On-Chain Secret-Holding Shards

“SCARAB: Hidden in Plain Sight” [preprint]

Allow blockchain to hold and *manage secrets* via verifiable, transparent, dynamic access policies

- Example: decryption keys, access lists for documents
- Example: login credentials for access to services

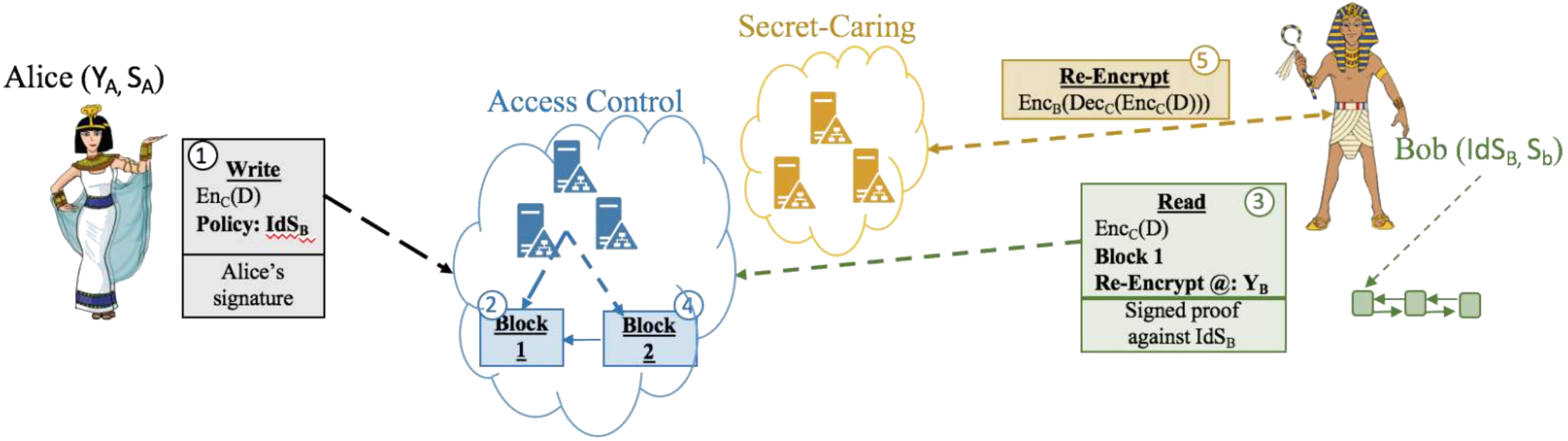


On-Chain Secret-Holding Shards

On-chain policies can determine how and when secrets used, who should have access when

- Any access change immediately, atomically applied
- Tamper-proof log of all uses or attempted uses

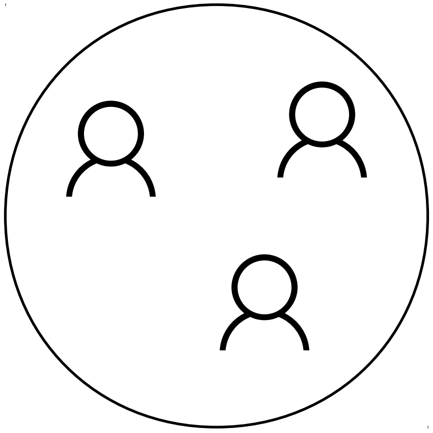
Enforce workflow, data retention/deletion policies



Locality Sharding

Problem: Strong global consensus requires us to pay global speed-of-light latencies

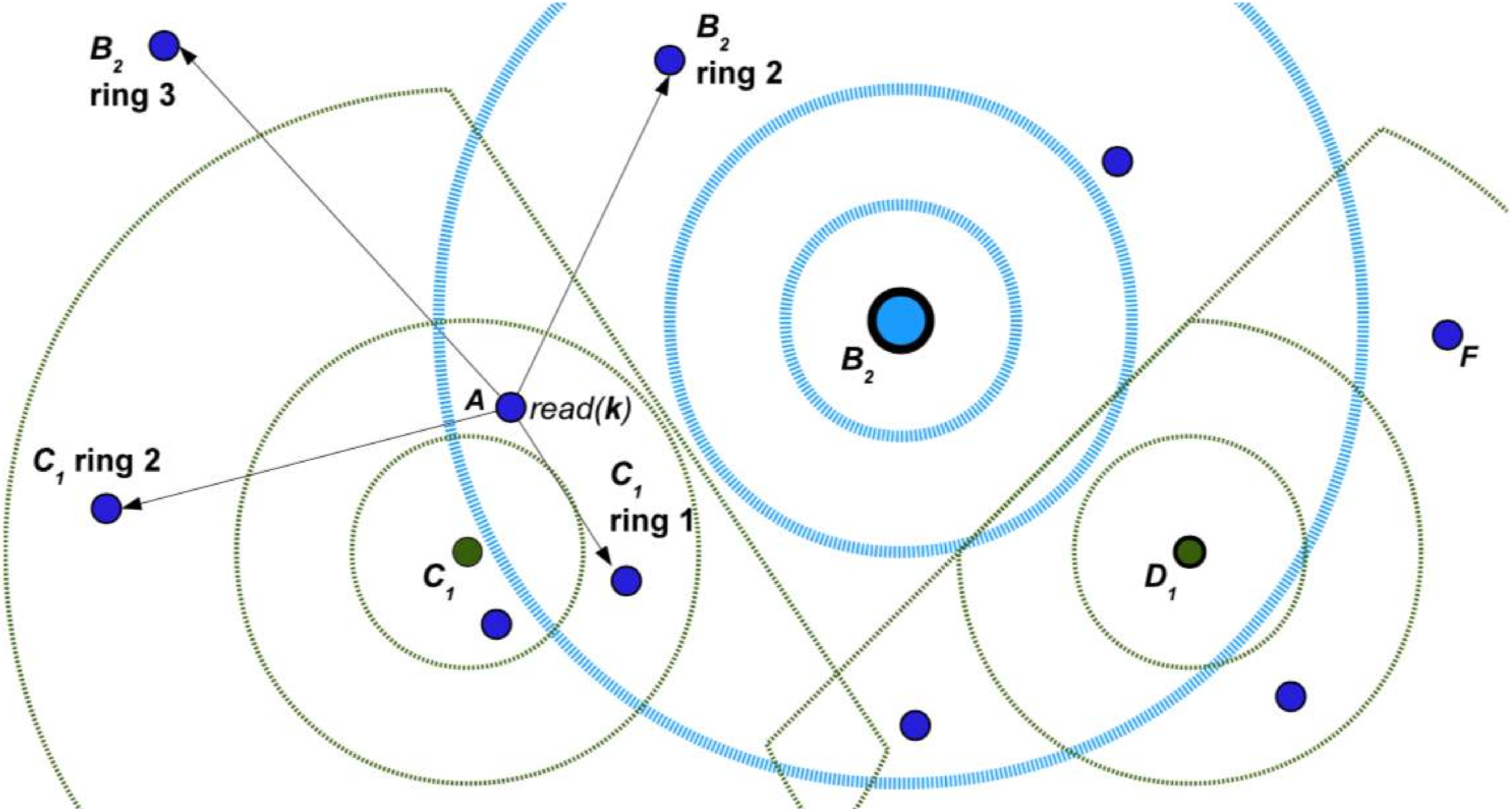
- But many interacting users are likely to be near each other in geography, network topology, network latency



Can we create many *local* blockchain shards, such that for any group of interacting users, they use a “nearby” shard offering low latency?

Locality from Graph Algorithms

Crux: Locality-Preserving Distributed Systems [preprint]



Scalable Coordination: Summary

Bitcoin’s architecture was a brilliantly wrong conflation of membership & consensus protocols

- De-conflating them is not trivial but massively improves performance, scalability, consistency
  - Bitcoin-NG, ByzCoin, OmniLedger
- Critical scalability tool: public randomness
  - RandHound/RandHerd, used in OmniLedger
- In the future we’ll see many different types of shards with different compositions, purposes

Talk Outline

- Key challenges in decentralized systems, and partial solutions to some of them
  - Scalable secure coordination
  - **Membership and fairness**
  - Governance and incentives
- Conclusion: democratic decentralization?

[credit: me.me]

Membership in Decentralized Systems

Any organization must have a way to define:

- Who are the **members** involved in decisions?
- How much **power** does each member hold?

Example: how does Bitcoin define membership?

- Permissionless: open to anyone, *in principle…*
- But only those willing to undergo (repeatedly) a particular, otherwise useless “hazing ritual”

In this sense, Bitcoin is similar to a fraternity.

Membership via Hazing Ritual

Can be anything that not everyone is willing or able to do on a whim → create a *barrier to entry*

Often uncomfortable and/or embarrassing…

Membership via Hazing Ritual

Other times, just plain weird

- MIT ‘58: using Oliver Smoot to measure bridge

Membership via Hazing Ritual

Or especially difficult, requiring cooperation

- Yap: chisel a giant circular “coin” out of stone available only on another, distant island

Bitcoin’s Hazing Ritual

Digitally flip coins.

Many coins.

Billions of them.

By forming new “blocks” and feeding them into a *cryptographic hash*

- Converts any information to pseudorandom number

Repeat endlessly.

Power Distribution in Bitcoin

How much **power** does each member wield?

- Proportional to member’s rate of coin-flipping: number of “hashes per second”, or **hashpower**
- More energy, faster chips → more hashpower

Value in Bitcoin

How does Bitcoin create **value** for its members?

Each time a miner wins coin-flipping lottery:

- Gets to *create a limited amount of new Bitcoin*
- Collects *transaction fees* from all transactions committed in the new block the miner added

Competition-based mining difficulty creates scarcity, supports the “value” of Bitcoin currency

JUST...ONE...
...MORE...BITCOIN

Bitcoin: a Planetary Hazing Ritual

Bitcoin is a **currency backed by energy waste**:

- Bitcoin makes BTC scarce by making miners prove they wasted energy

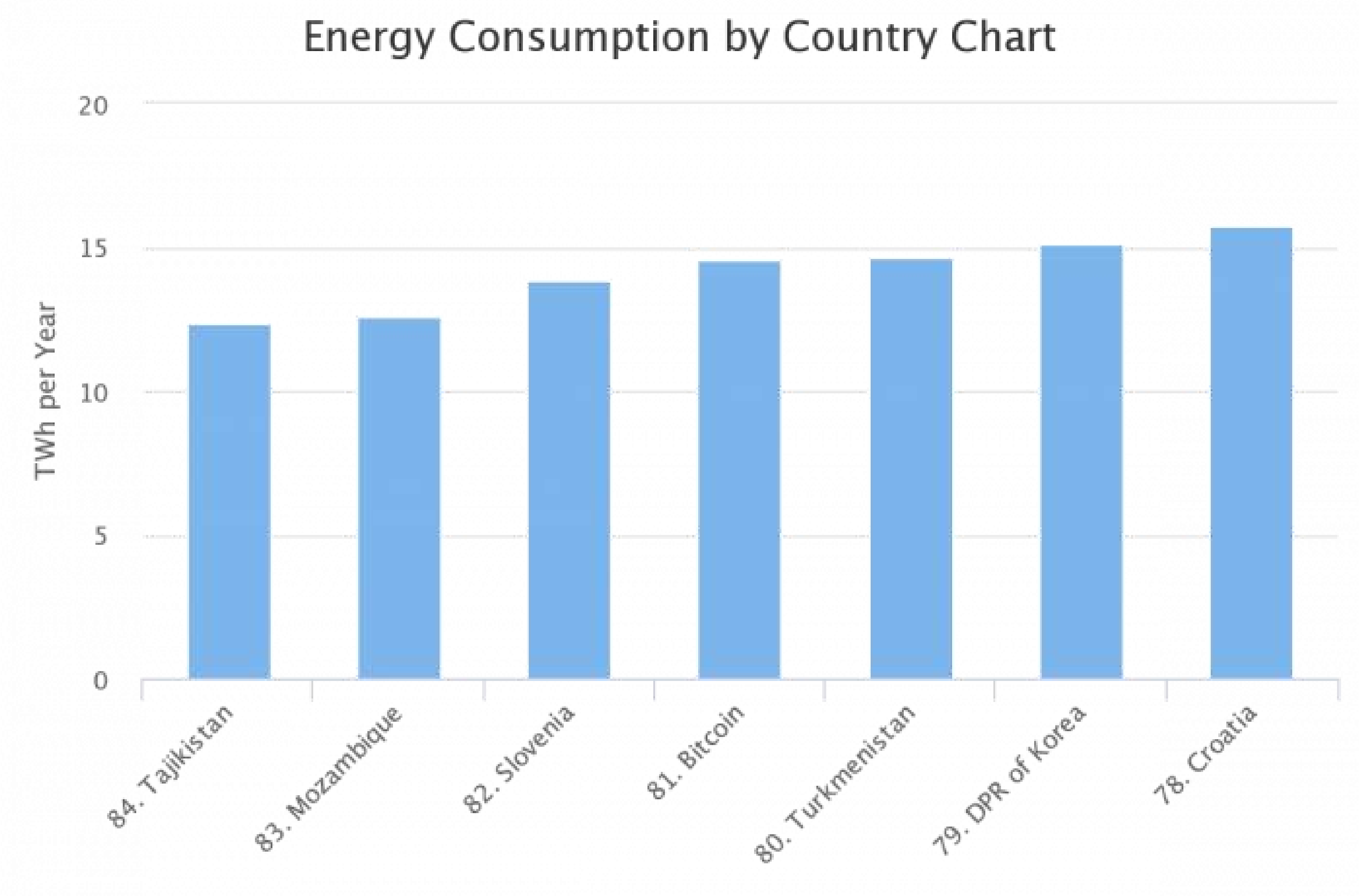
Proof of [useless] work: solve crypto-puzzle

- Takes lots of CPU cycles (energy) to **create**
- But trivial, cheap for anyone to **verify**

Like hazing, serves **no purpose** but prove you did it

Bitcoin Energy Consumption

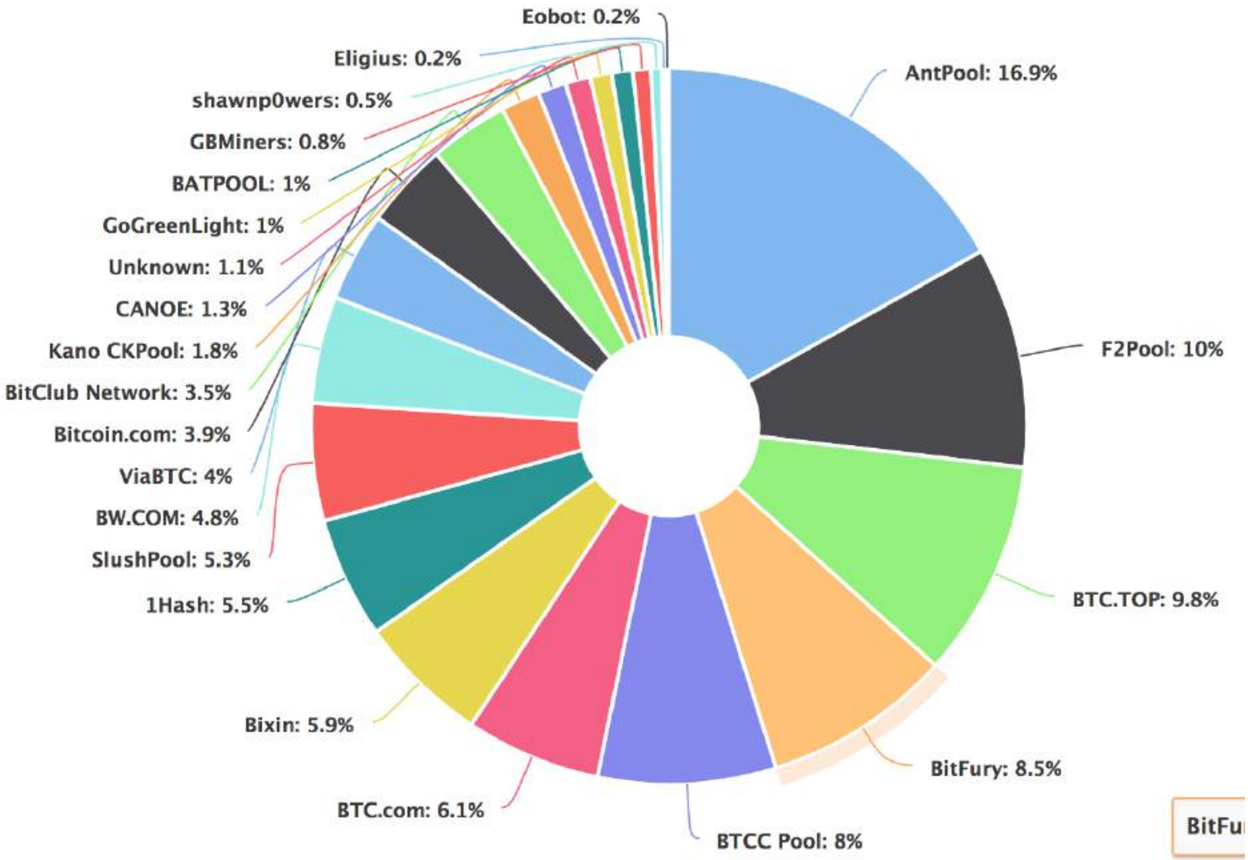
Bitcoin **wastes more energy** than the entire (useful) energy consumption of many countries



Not Even Decentralized Anymore

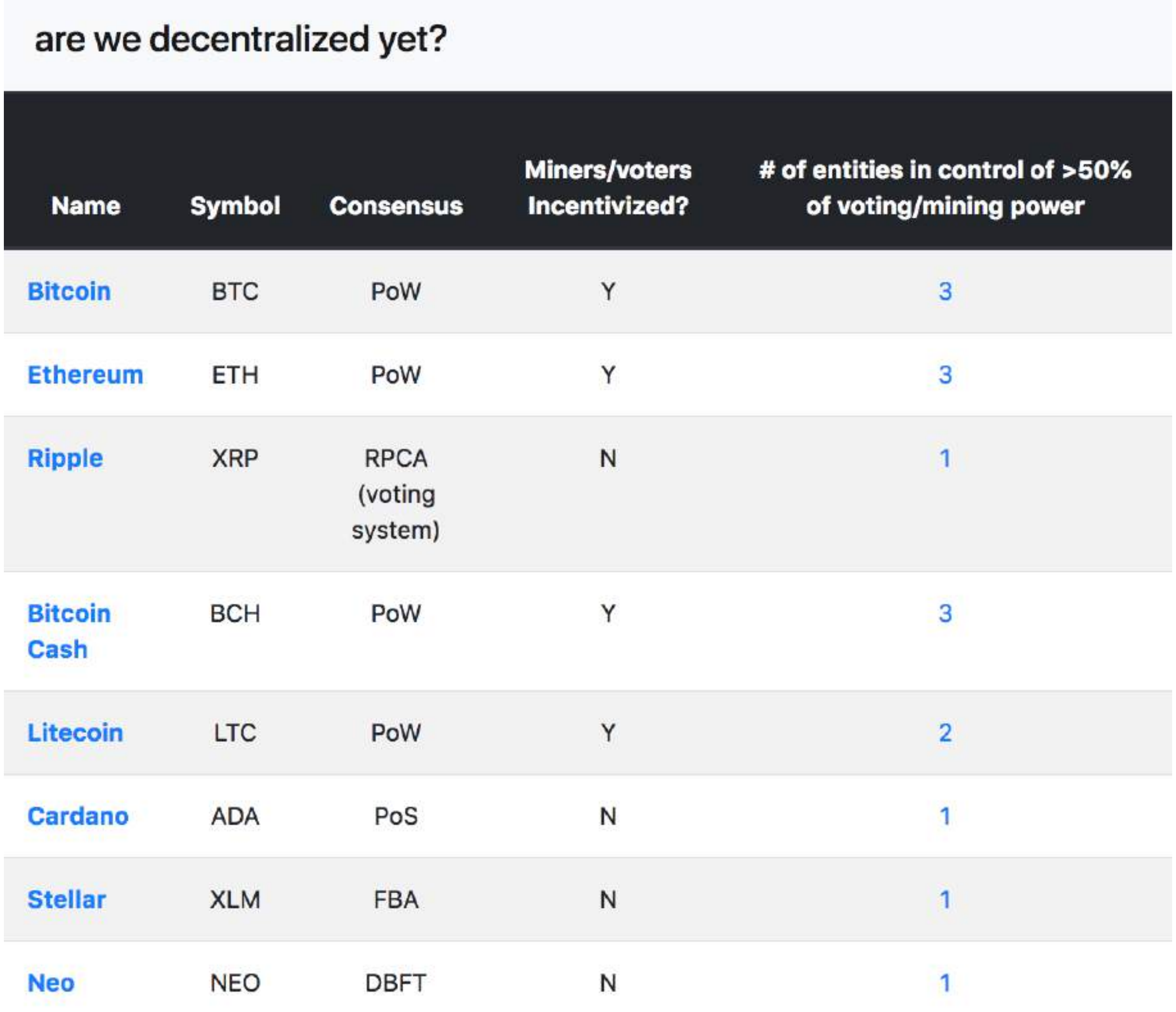
Market incentives drive consolidation of hashrate or “voting power” to a few powerful mining pools

- Over 60% currently in one country (China)
- Any faction >51% can control or veto decisions, censor, etc.



A Problem Not Unique to Bitcoin

Most cryptocurrencies aren’t that decentralized



are we decentralized yet?

Name	Symbol	Consensus	Miners/voters Incentivized?	# of entities in control of >50% of voting/mining power
Bitcoin	BTC	PoW	Y	3
Ethereum	ETH	PoW	Y	3
Ripple	XRP	RPCA (voting system)	N	1
Bitcoin Cash	BCH	PoW	Y	3
Litecoin	LTC	PoW	Y	2
Cardano	ADA	PoS	N	1
Stellar	XLM	FBA	N	1
Neo	NEO	DBFT	N	1

Alternative: Permissioned Ledgers

Just decide **administratively** who participates; Fixed or manually-changed group of “miners”

- ☺ No proof-of-work needed → low energy cost
- ☺ More mature consensus protocols applicable
- ☹ Higher human organizational costs
- ☹ No longer open for “anyone” to participate

Alternative: Proof-of-Stake (PoS)

- **Proof-of-Stake:** assigns consensus shares in proportion to prior capital investment
  - ☺ Could address energy waste problem
  - ☹ Major unsolved security & incentive problems
- But implementing PoS securely isn’t trivial…

Key Challenges with Proof-of-Stake

Implementing proof-of-stake securely requires:

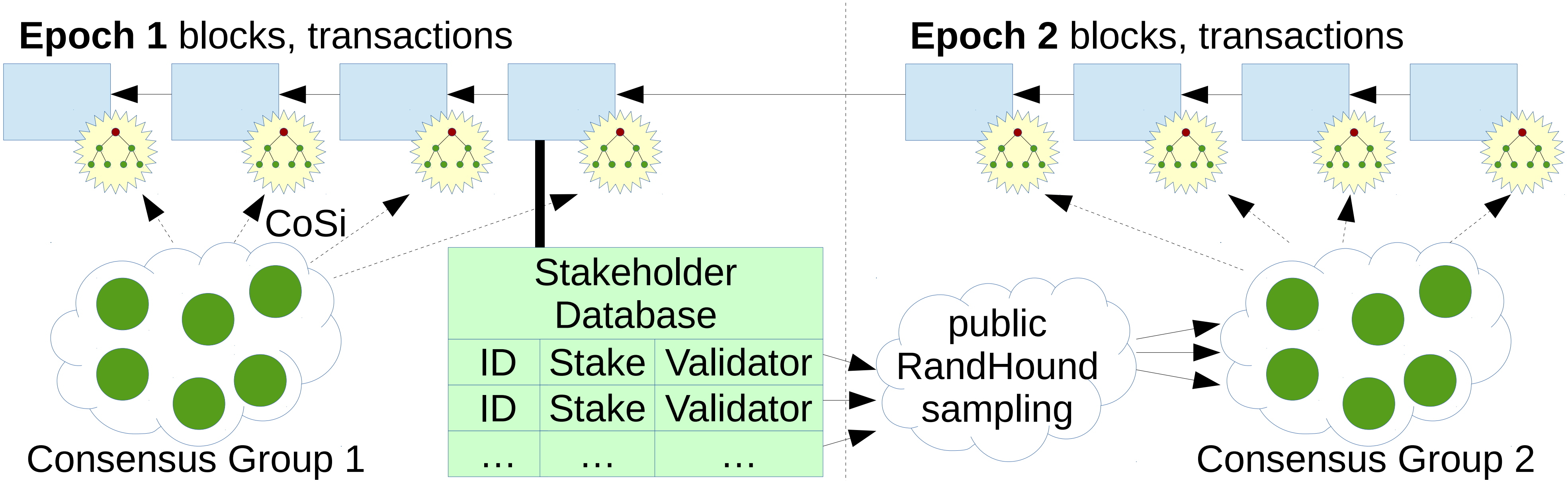
- **Agreement** on current set of stake-holders
  - e.g., list of public keys with number of “shares” each
- **Randomness** to sample future “minters” or consensus group members securely & fairly
- **Verifiability** of current state of the system
  - allow parties to distinguish the “one true blockchain” & avoid “nothing-at-stake” problem (chain mining)

Need tools from ByzCoin, RandHerd, Chainiac.

Modular Proof-of-Stake

Assume we have a ByzCoin-like consensus group

- Use PBFT to agree on transactions and stake
  - List of stakeholders, # shares each, their validators
- After epoch, RandHound-sample next group
  - Old group collectively signs new, forms **SkipChain**



Epoch 1 blocks, transactions

Epoch 2 blocks, transactions

CoSi

Consensus Group 1

Stakeholder Database

ID	Stake	Validator
ID	Stake	Validator
…	…	…

public RandHound sampling

Consensus Group 2

Problem: Efficient Verification

How does anyone who might be long out-of-date, securely confirm the latest blockchain state?

- Especially after being offline for months, years?
- Without “just trusting” central party (exchange)?

Weak SPV approach: just verify block headers

- Still must gossip with many parties
- Still costs bandwidth, especially to “catch up”
- Vulnerable to (costly but feasible) fake views

Chainiac: Traversable Blockchains

DEDIS work appearing in [USENIX Security ‘17]

- SkipChains: light-weight cryptographic verification forward and backward in time
- Applied to secure key & software updates

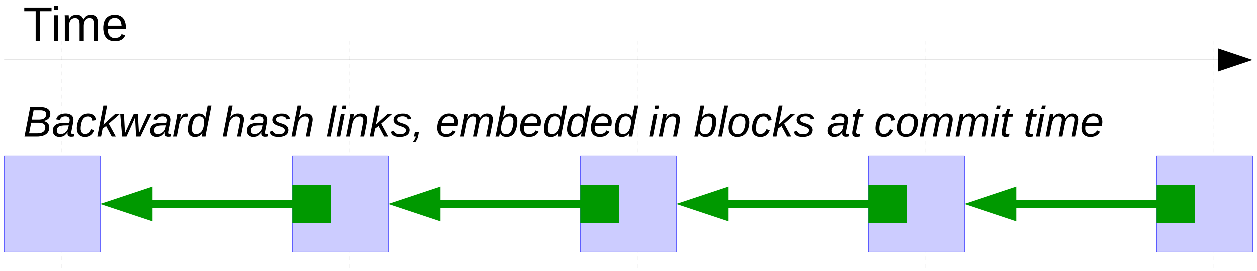
ByzCoin already **collectively signs** each block

- With 1 signature check, anyone can confirm that hundreds/thousands of parties validated
- Problem: the *set* of validators keeps changing!
  - Slightly different set of public keys every ~10 mins

Backward and Forward Verifiability

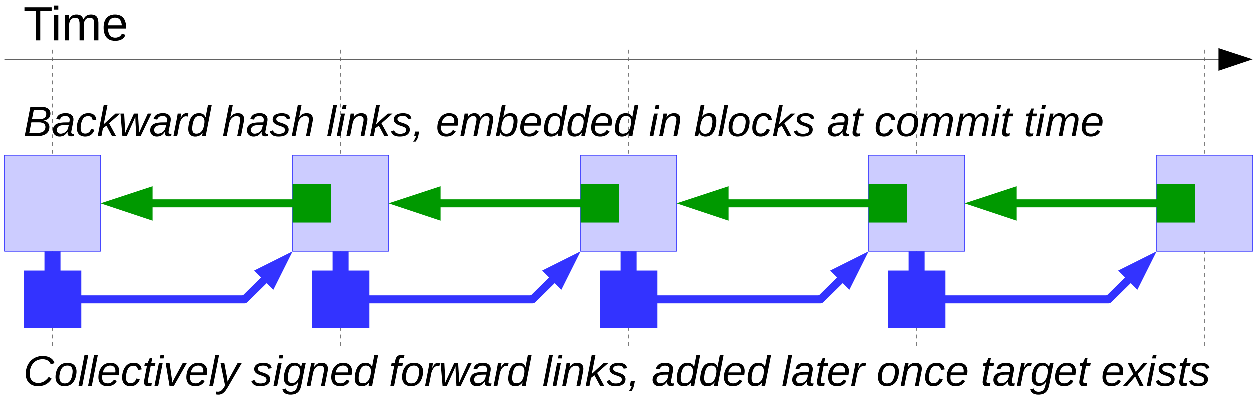
Standard blockchains traversable only **backward**

- Via hash back-links from current head



Chainiac adds traversability **forward in time**

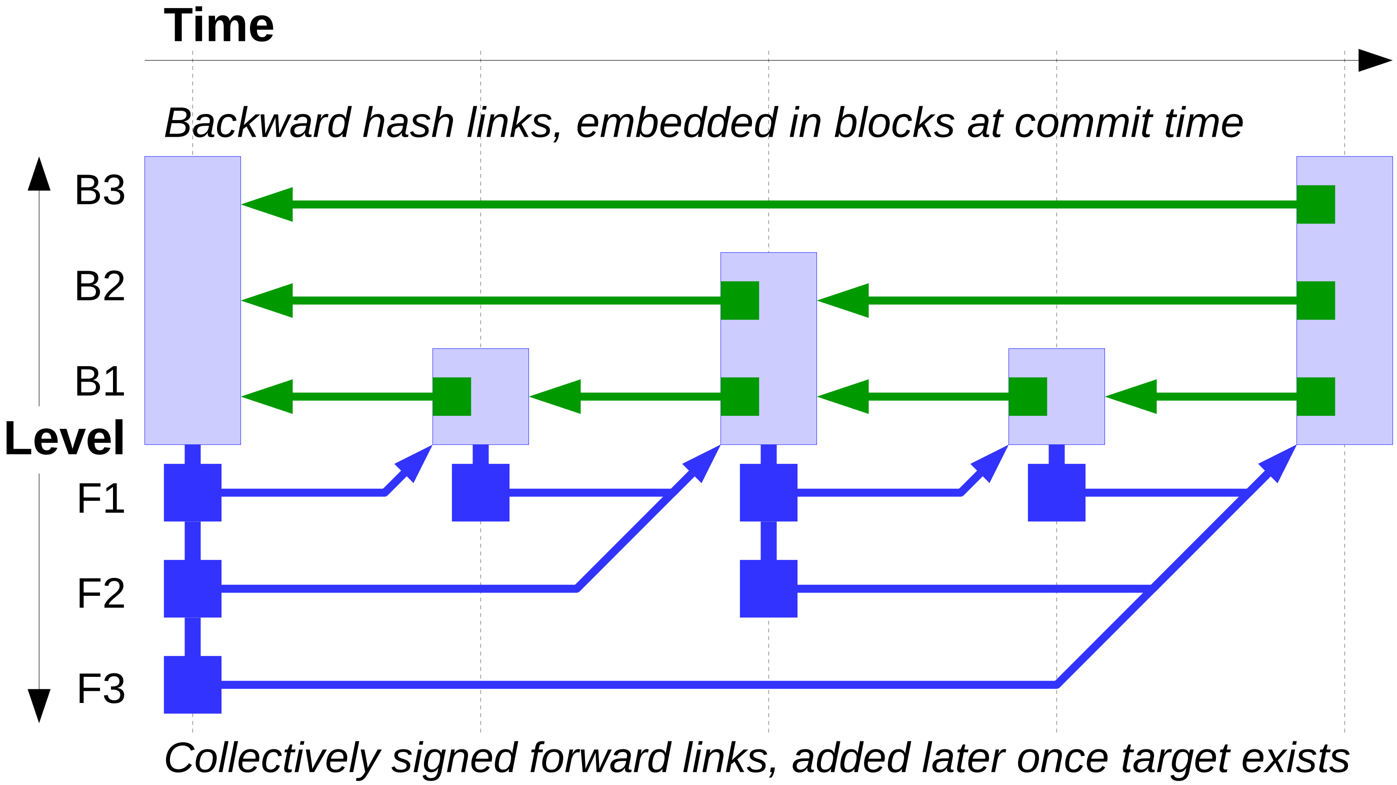
- Collective signature by prior consensus group


Time
Backward hash links, embedded in blocks at commit time
Collectively signed forward links, added later once target exists

Leaping Through Time: SkipChains

Each block validates *prev* w/hash, *next* w/sig

- Higher level hashes, sigs → longer hops
- O(log N) traversal arbitrarily forward, back



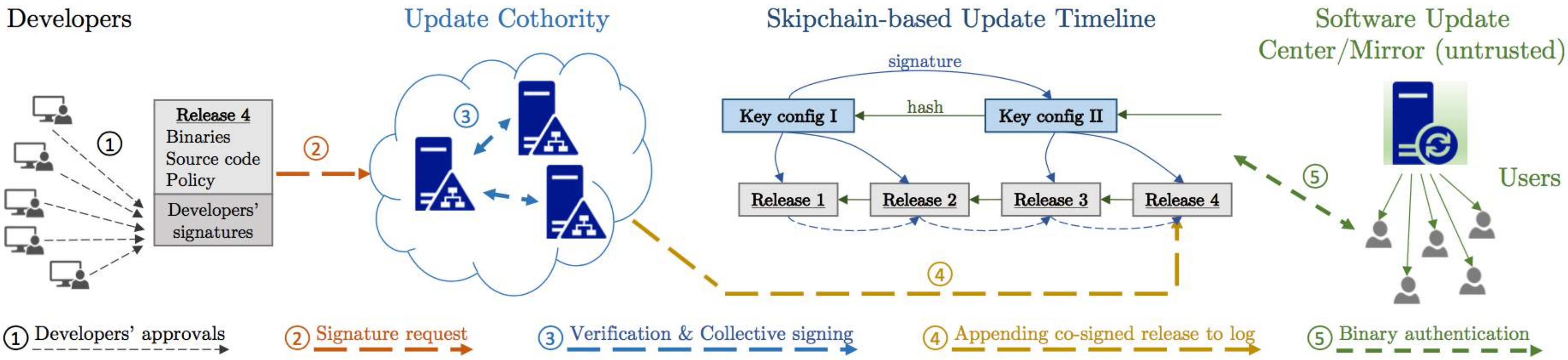
Time
Backward hash links, embedded in blocks at commit time
B3
B2
B1
Level
F1
F2
F3
Collectively signed forward links, added later once target exists

Chainiac: Secure, Transparent Software Development & Updates

Create end-to-end secure development pipeline

- Development: peer review, signoff workflow
- Build: independent verification of exact binaries
- Distribution: offline/P2P updates via SkipChains

Applicable to open source & proprietary software



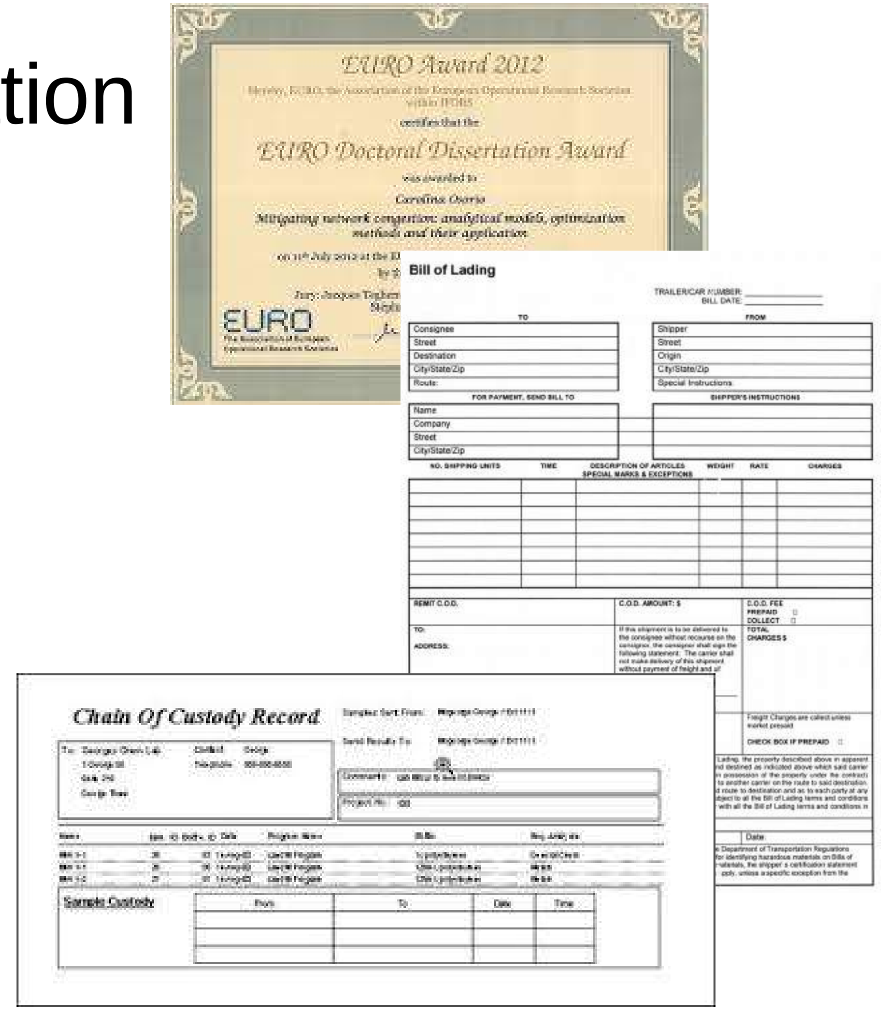
Other Applications of SkipChains

Enable Offline/P2P verification

- Works even if Internet is unavailable, slow, costly

Broad applications

- Software/key updates
- Blockchain-Attested Degrees, Awards, …
- Chain-of-Custody, Bills of Lading, …



EURO Award 2012

Hereby, EURO, the Association of the European Operational Research Societies within IFORS

certifies that the

EURO Doctoral Dissertation Award

was awarded to

Carolina Osorio

Mitigating network congestion: analytical models, optimization methods and their application

on 11th July 2012 at the E

by t

Jury: Jacques Teghem

Steph

EURO

The Association of European Operational Research Societies

Bill of Lading

TRAILER/CAR NUMBER: ______
BILL DATE: ______

TO

Consignee
Street
Destination
City/State/Zip
Route:

FROM

Shipper
Street
Origin
City/State/Zip
Special Instructions

FOR PAYMENT, SEND BILL TO

Name
Company
Street
City/State/Zip

SHIPPER'S INSTRUCTIONS

NO. SHIPPING UNITS	TIME	DESCRIPTION OF ARTICLES SPECIAL MARKS & EXCEPTIONS	WEIGHT	RATE	CHARGES

REMIT C.O.D.
TO:
ADDRESS:

C.O.D. AMOUNT: $

If this shipment is to be delivered to the consignee without recourse on the consignor, the consignor shall sign the following statement: The carrier shall not make delivery of this shipment without payment of freight and all

C.O.D. FEE
PREPAID □
COLLECT □
TOTAL CHARGES $

Freight Charges are collect unless marked prepaid

CHECK BOX IF PREPAID □

Date

Chain Of Custody Record

Sample Custody

From	To	Date	Time

See: “How Do You Know It’s On the Blockchain?”

Recap: Modular Proof-of-Stake

- **Agreement:** inductively assume a consensus group exists at any given point in time
  - ByzCoin's PBFT decides current stakeholder state
- **Randomness:** sample next consensus group
  - Use RandHerd in current consensus group to secure, representative sample to form *next* group
- **Verifiability:** distinguishing the true blockchain
  - Chainiac's SkipChains provide collective signatures
  - Attackers can't create valid fake blockchains without compromising many existing validators

How important is Proof-of-Stake?

A Proof-of-Stake cryptocurrency is essentially an automated analog of a **shareholder corporation.**

- May help hasten the robot takeover,
  but won’t fix the world.

It’s all just “Proof-of-Investment”

Proof-of-Work, Proof-of-Stake, Proof-of-Storage, and most Proof-of-* proposals are variants of **Proof-of-Investment**, aka investment capitalism.

- The more of *whatever* you can afford to commit, the more voting power and rewards you get.

All organizations based on “Proof-of-Investment” inherit basic problems from investment capitalism.

- Larger stakeholders can exploit advantages to further increase their percentage of the pie.

All prone to re-centralization, aka, **rich get richer**

Towards Democratic Blockchains

Can we build decentralized systems that will securely *remain* decentralized?

My bet is on the principles of **democracy**.

One Person One Vote

Proof-of-Personhood [IEEE S&B ‘17]

- Like Proof-of-Stake, but “**one person one vote**”
- Enforce via Pseudonym Parties [SocialNets ‘08]

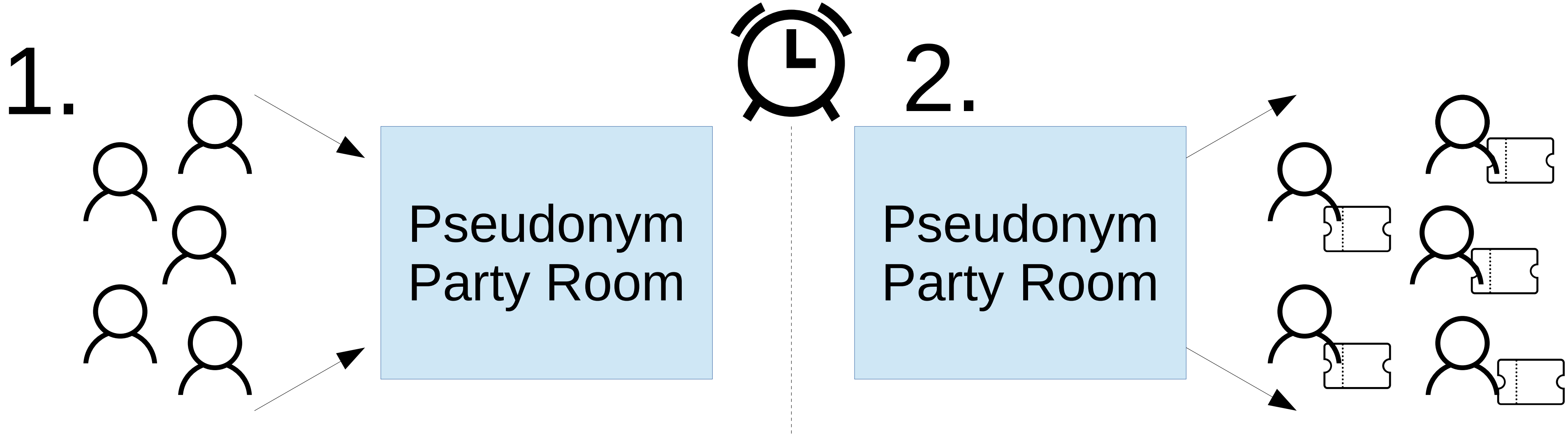
Pseudonym Parties: Summary

Locally-organized regular **physical meetings**

- Anyone can *enter* room until a set deadline
- Then can only *exit*, each getting one credential

No need for IDs, biometrics, PGP key-signing, etc

- Just bodies: can be in only one place at a time



Scaling Pseudonym Parties

Many local communities host pseudonym parties independently but with **synchronized deadlines**

- One person, one credential, *across all parties*

Local communities federate, monitor each other to build large-scale trust network of communities

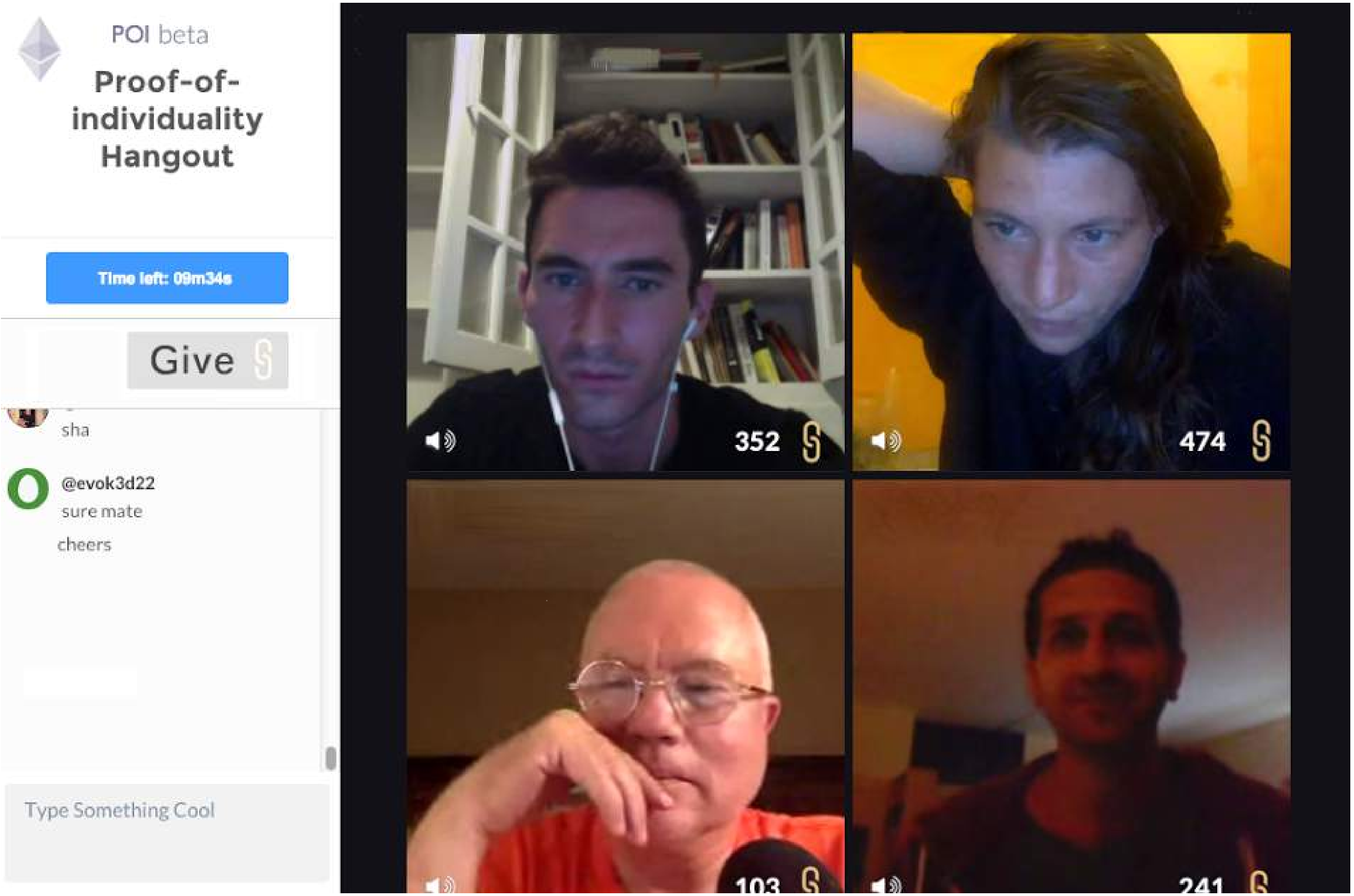
- e.g., each party must host RandHound-chosen group of observers from other communities

Easier than securing trust networks of individuals

- Organizers can be expected to have geek skills; ordinary participants just need to show up

Other potential approaches

Proof-of-Individuality, an online video equivalent



sha

@evok3d22
sure mate
cheers

Summary: Membership & Fairness

Any decentralized system needs to define who its members are and how much power each has

- **Proof-of-Work:** a disaster that can & must die
- **Permissioned:** a reasonable, efficient approach for federations that are closed anyway
- **Proof-of-Stake:** a useful step with interesting technical challenges, but not the final answer
  - Same with all “Proof-of-Investment” foundations
- **Proof-of-Personhood:** a democratic foundation for decentralization based on real people

Talk Outline

- Key challenges in decentralized systems, and partial solutions to some of them
  - Scalable secure coordination
  - Membership and fairness
  - **Governance and incentives**
- Conclusion: democratic decentralization?

Organizations and Governance

Humans have been banding together to form organizations throughout recorded history…

But if governance breaks, organizations collapse.

Can Blockchains Self-Govern?

[credit: “A View to a Kill”]

Bitcoin “Governance”

More like, “We Don’t Need No Governance”

But how to decide how to evolve & upgrade Bitcoin?

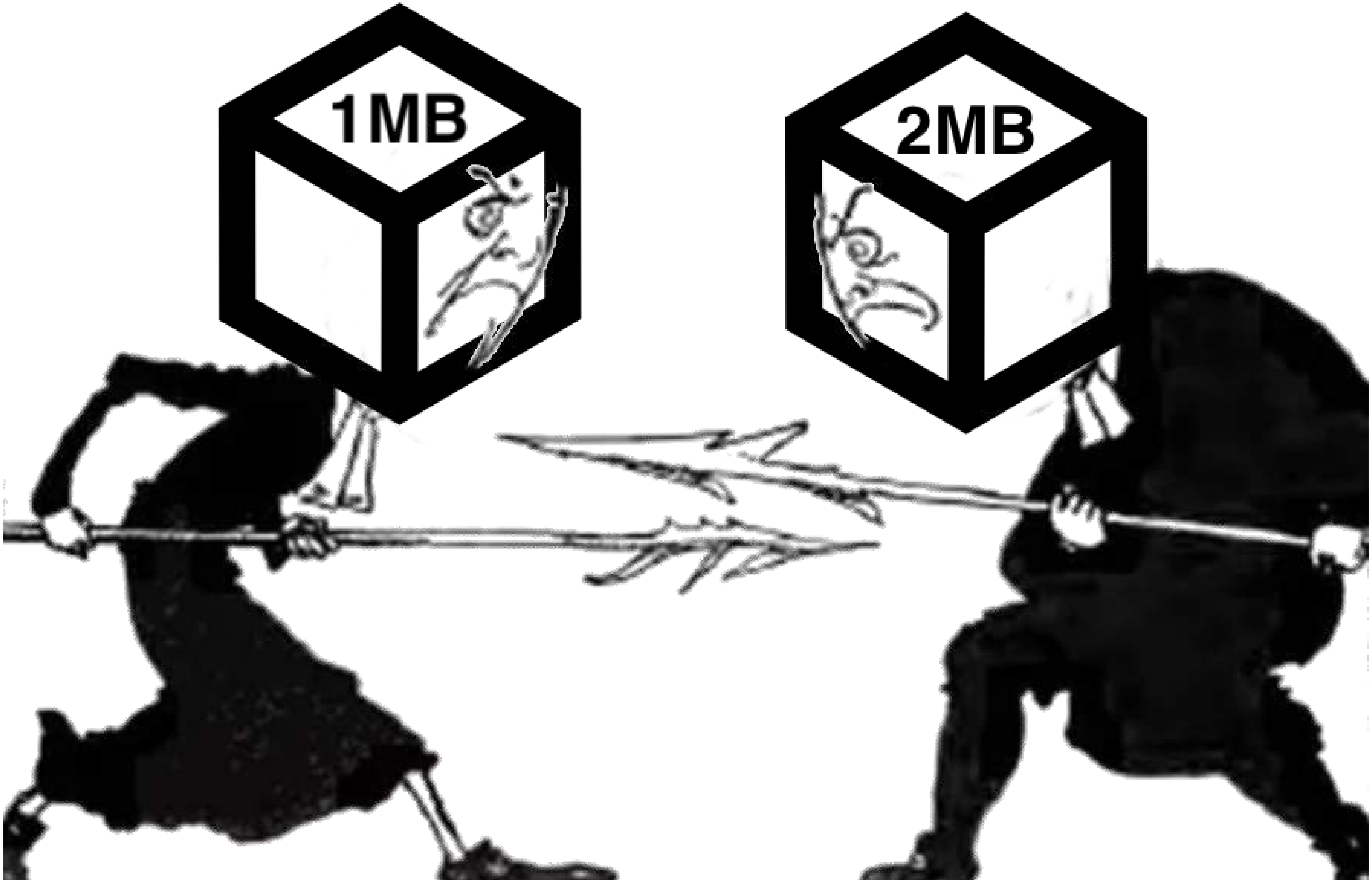
- Uncontroversial decisions: “consensus” by influence among developers, miners
- Controversial decisions: hmmm…

“Gulliver’s Travels”

War between Big-Endians and Little-Endians

“Bitcoiner’s Travels”

War between Big-Blockians and Little-Blockians



Blockchain Governance Challenges

Many governed by conventional organizations

- Ethereum, Zcash, …

But how to create a stable *self-governing* system?

- Decision processes mediated by the system
- Used to develop, evolve, upgrade the system

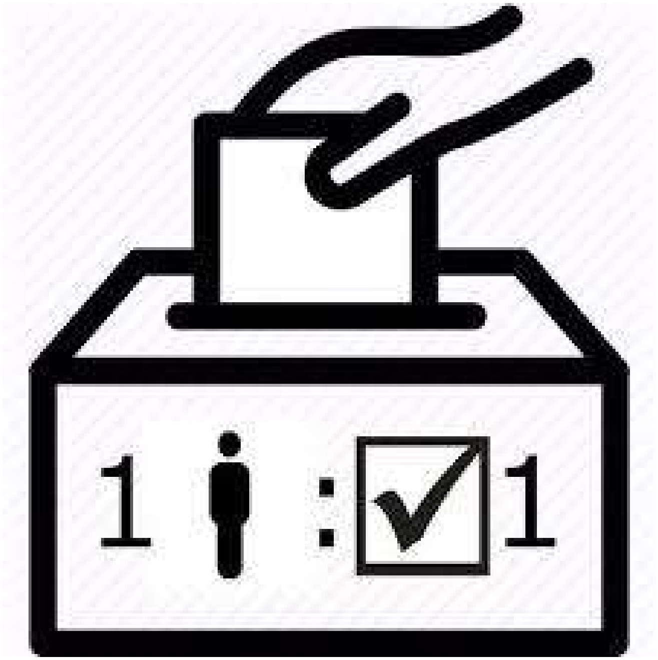
Huge open design space of governance models

- Any bug, vulnerability could be fatal
- Need ways to experiment, evaluate safely

Democratic Decentralized Systems?

Can we build secure *democratically* self-governing online decentralized systems?

- Pervasive “one person, one vote” principle



Key Elements to Governance

A blockchain self-governance system must have:

- Secure **foundation** for membership and power, invulnerable to Sybil attacks & gradual takeover
- Secure **decision-making** processes enabling members to make decisions collectively
- Secure **information-gathering** processes to keep power-wielding members well-informed
- Secure **incentives** to participate and invest time, effort, and other external resources

Membership and Decision-Making

For democratic governance,
proof-of-personhood is a natural foundation

- Literally enforce “one person one vote” in governance decisions

DEDIS blockchain infrastructure already includes components for decision-making via **voting**

- On-chain ElGamal secrets, verifiable shuffles: common tools in verifiable voting systems
    - Part of e-voting system for use within EPFL

Complete, scalable system still future work…

Decentralized Information Feeds

No democratic governance system is secure if its voters are susceptible to bot-driven propaganda

- Anyone can lie, but Sybil attacks amplify them

Secure democratic self-governance online needs discussion forums, newsfeeds, reputation systems that only count “likes” or “upvotes” of *real people*

- Creates “anonymity vs accountability” tension
  - Anonymity for freedom of expression (Twitter, Tor)
  - Accountability for abuse-resistance (Facebook)

Towards Privacy *with* Accountability

Anonymous messaging and credential systems can enforce “one pseudonym per real ID” rule

- With pseudonym parties: “one nym per person”

But pseudonymity is a weak form of anonymity

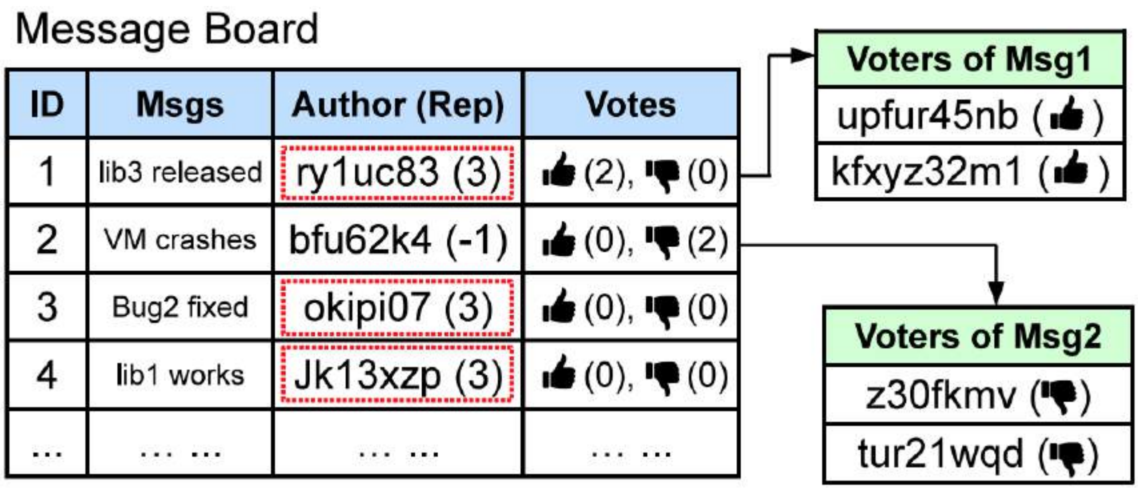
- Privacy degrades rapidly over time with use
- Intersection & statistical disclosure, differential privacy budget problem, …

Towards Privacy *with* Accountability

A more powerful tool: *anonymous reputation*

Early prototype: AnonRep [NSDI ‘16]

- Users post information *fully* anonymously, perform peer review (e.g., upvotes/downvotes)
- System encrypts reputation balances
- Posters reveal only reputation buckets (e.g., “>1000”)



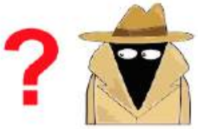
Message Board

ID	Msgs	Author (Rep)	Votes
1	lib3 released	ry1uc83 (3)	👍(2), 👎(0)
2	VM crashes	bfu62k4 (-1)	👍(0), 👎(2)
3	Bug2 fixed	okipi07 (3)	👍(0), 👎(0)
4	lib1 works	Jk13xzp (3)	👍(0), 👎(0)
...	... ...	... ...	... ...

Voters of Msg1
upfur45nb (👍)
kfxyz32m1 (👍)

Voters of Msg2
z30fkmv (👎)
tur21wqd (👎)

Zcash, zkLedger tools may help



Incentives to Participate

One of Bitcoin’s most brilliant ideas was incentivizing participation via new built-in currency

- Bitcoins were initially worth nothing, but low barrier to entry, interest, FOMO changed that

But two key problems with Bitcoin financial model

- Proof-of-work basis leads to re-centralization
- Deflationary 21M-total-coins model incentivizes speculation and HODLing over productive uses
  - “Bitcoin has no value, so it can have any price”-Lipton

A Democratic Crypto-Economy?

Can we build a stable, sustainable, democratic cryptocurrency to power decentralized systems?

- Democratic “equal-opportunity” foundation
  - Each *human* participant gets equal base resources (then free to become unequal by using them wisely)
  - Protect new economy from legacy rich & powerful
  - Protect next generation’s starting opportunity from domination by past generations’ winners & losers
- Incentivize productive use rather than HODLing
  - Keep price more stable & bound to real-use value

A Democratic Crypto-Economy?

One possible design sketch:

- Distribute new coins via Proof-of-Personhood
  - e.g., each participant gets 1 new coin per day
- Coins are “use-it-or-lose-it” via stable inflation
  - e.g., new year’s coins get 1/50th of value space
    - Like a 50-year coin lifetime but via gradual devaluation
    - Enough for investment over a modern human lifetime
    - But ensure each generation makes room in currency’s value space for next generation’s equal opportunity

Relation to Universal Basic Income

Intriguing idea in many respects…

- Simplify social “safety net”, tax structure, etc.

Many challenges,
open questions

- Such as:
  how to decide
  “how much”
  per person?

A Permissionless Basic Income?

A democratic cryptocurrency wouldn’t need to decide “how much” to give each participant

- Everyone gets to “mint” same amount per day
- Democratic cryptocurrency acquires value from scarcity, collective utility, participant buy-in
- No one decides “how much” a coin is worth: value floats to reflect coin’s collective utility

Due to security foundation in *human* participants, might still work after robots/AI take all our jobs?

Summary: Governance & Incentives

Decentralized systems need governance, with:

- Secure **foundation** for stable decentralization
- Secure **decision-making** methods, e.g., voting
- Secure **information-gathering** methods resistant to Sybil-attack propaganda campaigns
- Secure **incentives** for people to participate & invest their time, attention, other resources

I claim governance can & should be **democratic**

Towards Democratic Decentralization

We have many of the technical tools we need for scalable, *democratic* decentralized systems

- Scalable Byzantine consensus, public randomness, verifiable blockchains, sharding

Can we fill in the remaining missing pieces?

- “One person one vote” security foundation
- Democratic information feeds, voting, currency

Conclusion

Learning from Bitcoin’s genius and its mistakes illuminates key decentralized systems challenges:

- **Scalable secure coordination** via scalable BFT, public randomness, sharding, SkipChains
- **Membership and fairness** via Proof-of-Stake, or better yet, Proof-of-Personhood
- **Governance and incentives** yet to be built for equitable, stable, democratic self-governance

Thank you!

dedis.epfl.ch github.com/dedis

Code available on GitHub…

All are welcome to use it and build on it...

Kyber: Advanced Crypto Library for Go

- https://github.com/dedis/kyber
- Public-key Encryption, Signatures, Shamir Secret Sharing, Zero-Knowledge Proofs, Verifiable Shuffles, Optimized Ed25519, …

Cothority: Collective Authority Software Suite

- https://github.com/dedis/cothority
- CoSi, ByzCoin, RandHound, OmniLedger, …