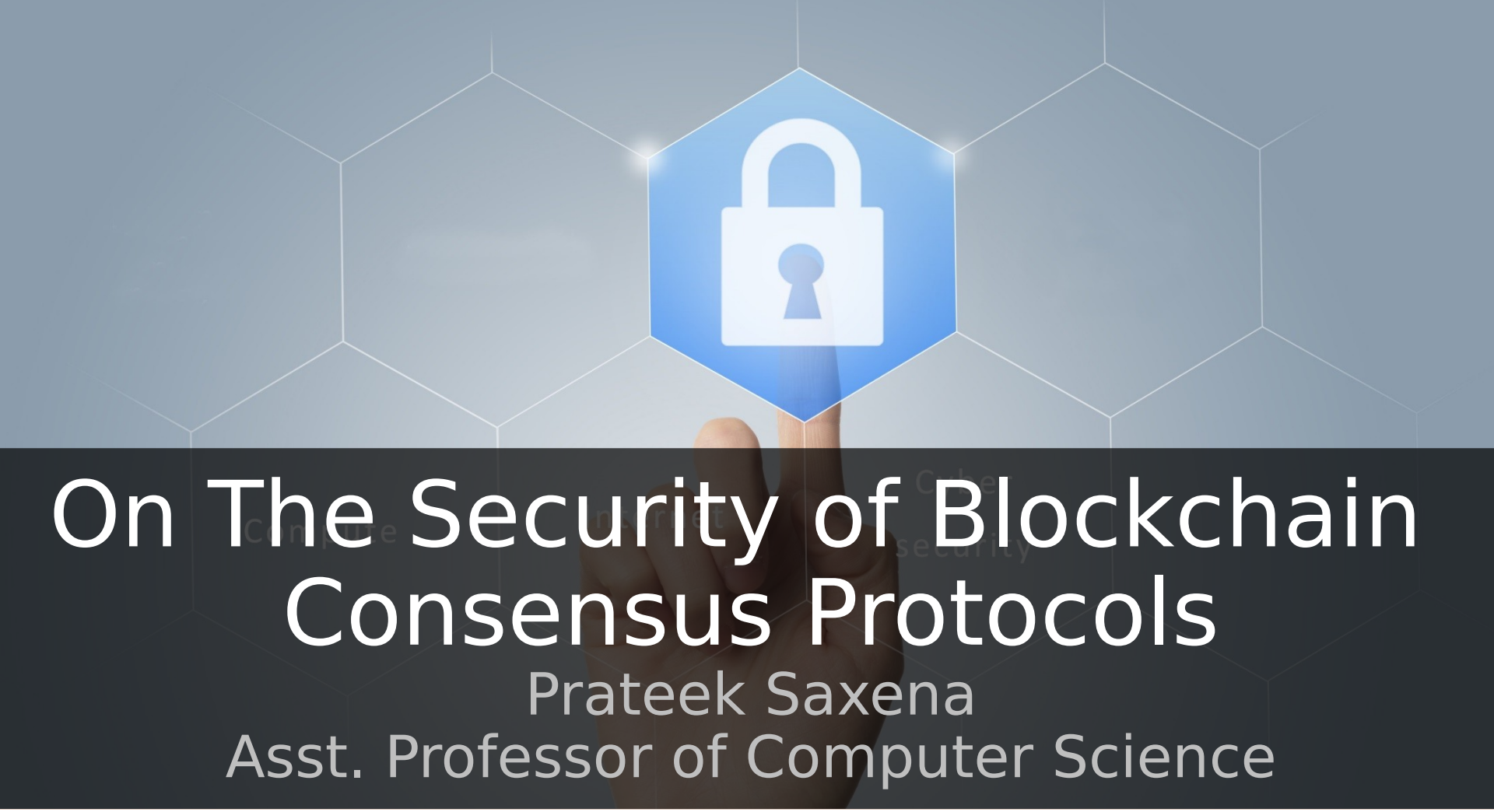


On The Security of Blockchain Consensus Protocols

Prateek Saxena
Asst. Professor of Computer Science

NUS National University of Singapore | School of Computing

The Origin of Blockchains

Blockchains: Origin & Today

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.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

2008

bitcoin
ANQUAN
STELLAR
Blockstream
ethereum
ripple
Digital Asset Holdings
intel
t0.com
Ledg
Tendermint
IBM Chain

2018

Application: Self-regulating Currency

Alice

Bob

Mary

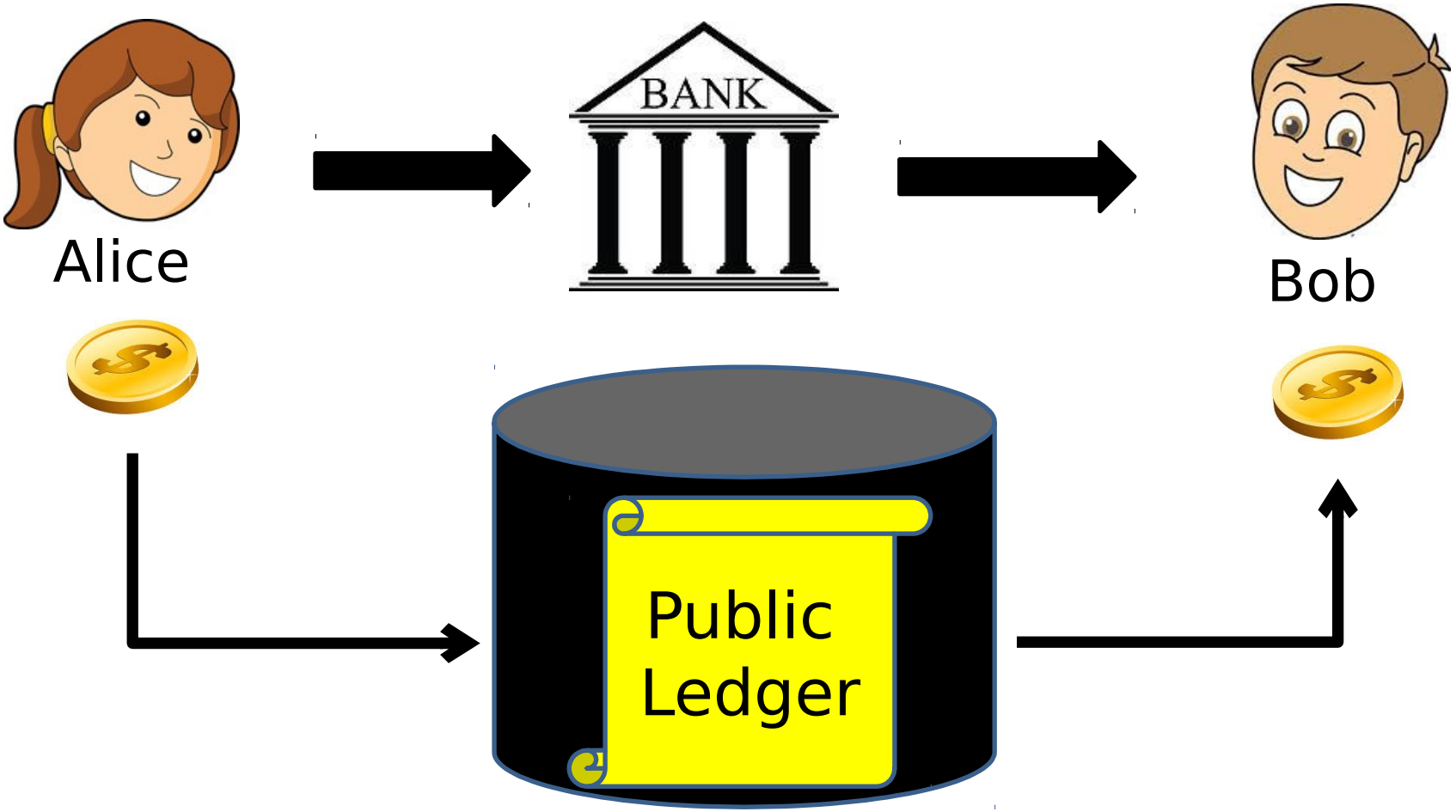
TX-1: Bob -> Mary
TX-2: Alice -> Mary

TX-1: Alice -> Bob
TX-2: Bob -> Mary

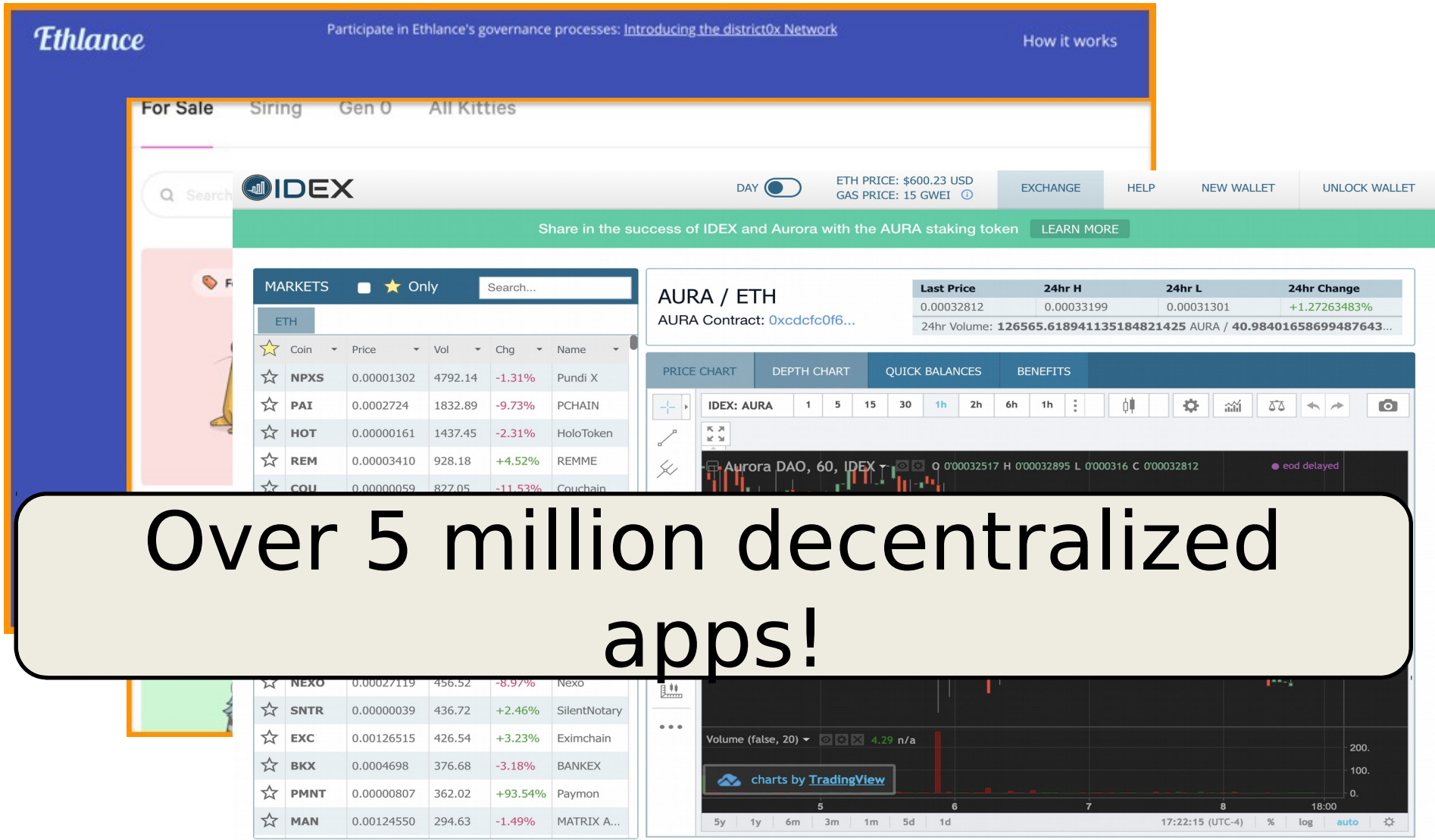
TX-1: Alice -> Bob
TX-2: Alice -> Mary

Double-spending

Application: Self-regulating Currency



From Payments To General-Purpose Computing



Over 5 million decentralized apps!

Outline

- Motivation & The Consensus Problem
- The Power of Simplicity
- Challenges & Recent Advantages
- Future Directions

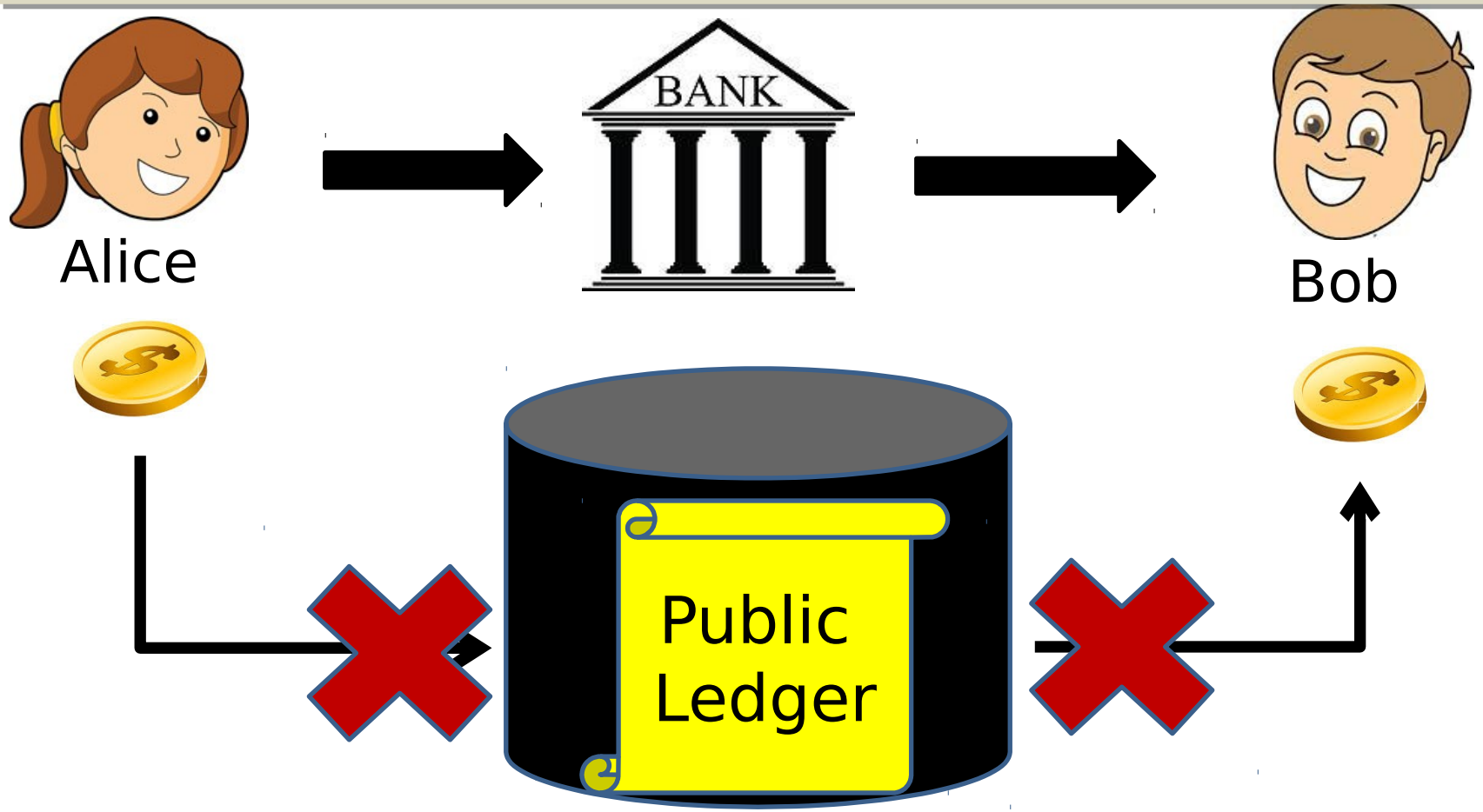
Why Might We Care?

A New Model of Trust

- Basis For Trust In Prior Systems:
  - Blind Faith / Assumption
  - Reputation
  - Incentives
  - Regulation

- A New Model: Self-regulation
  - Anyone can connect and audit the operations
  - (Extremely) High Availability
  - No permission needed, no centralized coordinator

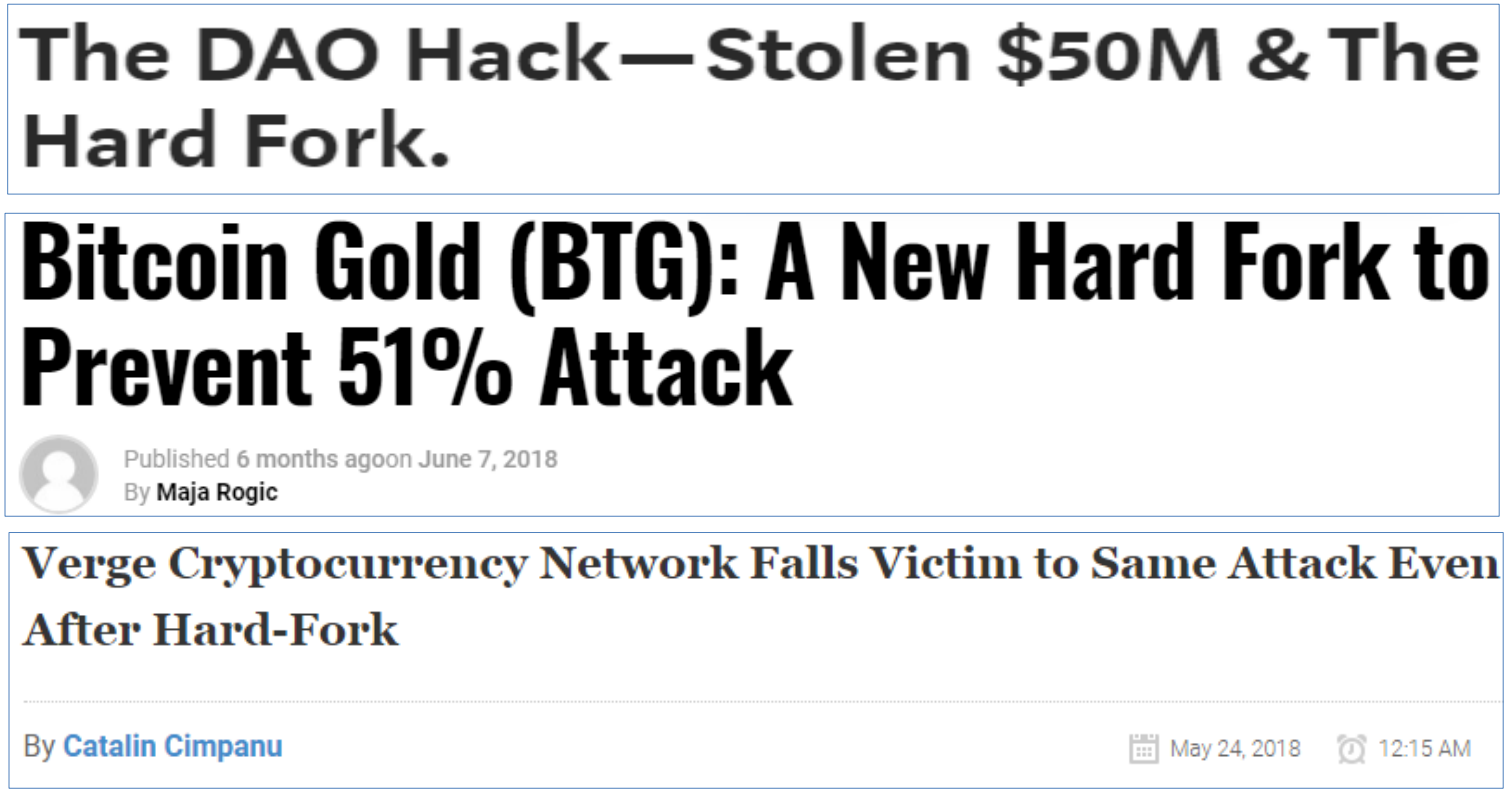
A New Model of Trust



- Prevent censorship of transactions (Fairness)
- Provide Availability of infrastructure (Resilience)

A New Model of Trust

- A Shift in the Design Philosophy:
  - Security First, Performance Later!
  - Once Deployed, no upgrades



The DAO Hack—Stolen $50M & The Hard Fork.

Bitcoin Gold (BTG): A New Hard Fork to Prevent 51% Attack

Published 6 months agoon June 7, 2018
By Maja Rogic

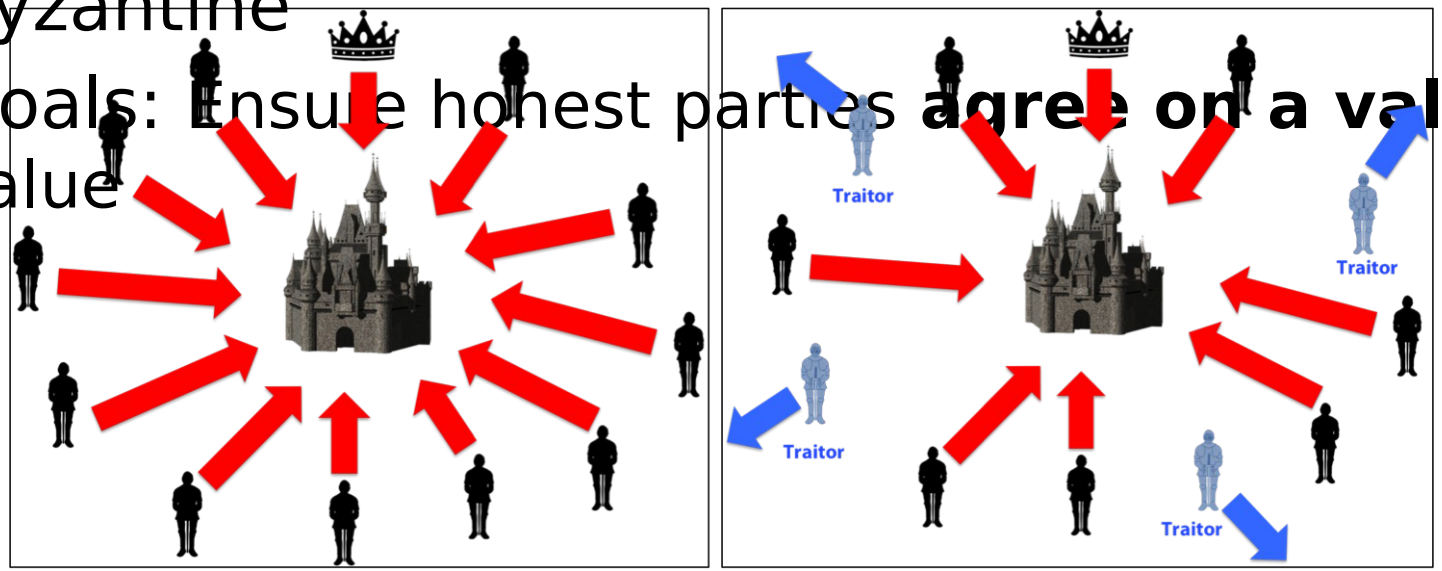
Verge Cryptocurrency Network Falls Victim to Same Attack Even After Hard-Fork

By Catalin Cimpanu

May 24, 2018 12:15 AM

A New Perspective On Classical Problem

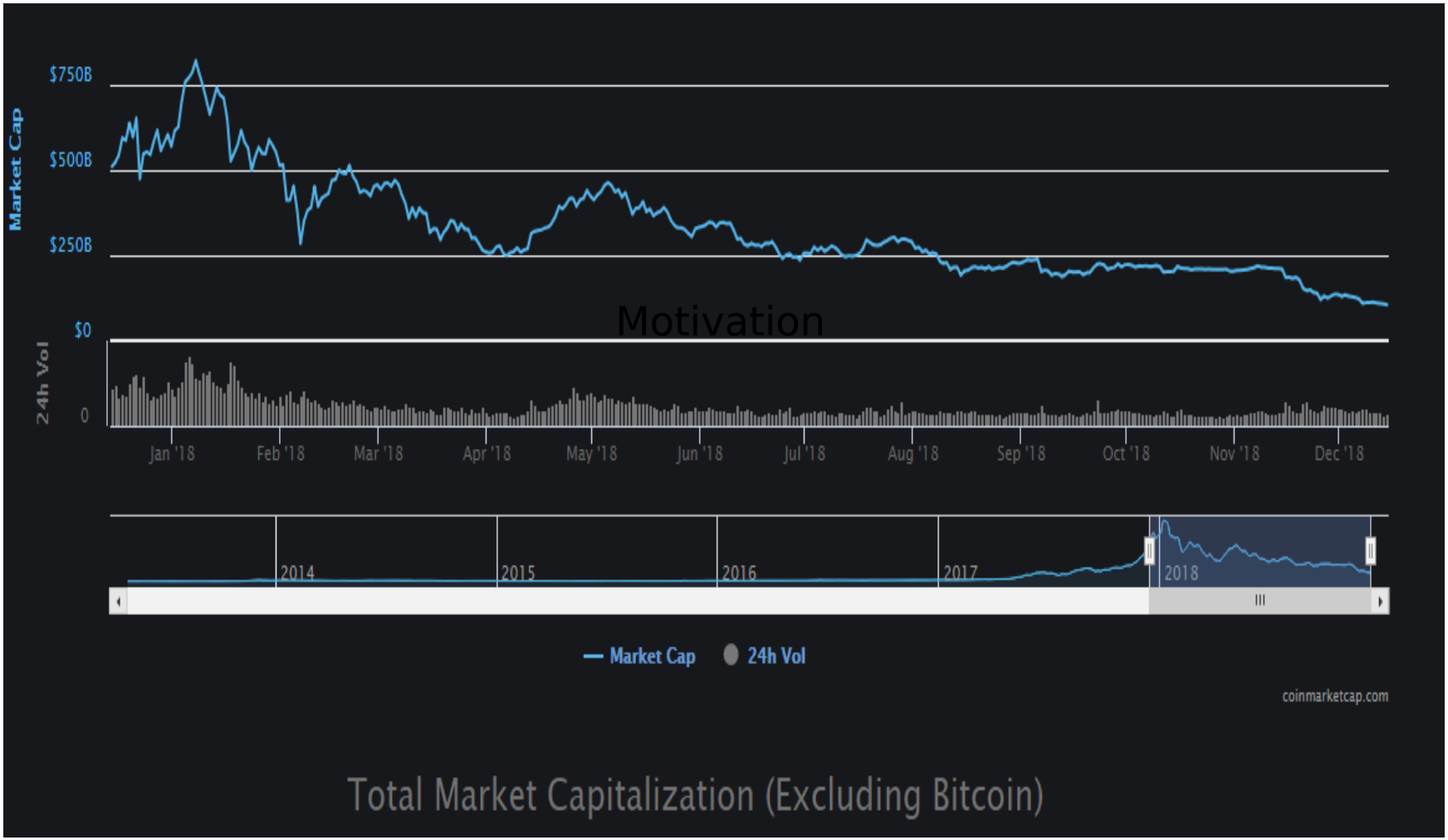
- Byzantine Agreement Problem (Lamport et al. 82):
  - A fraction **f** out of **n** of parties malicious, i.e., Byzantine
  - Goals: Ensure honest parties **agree on a valid** value



Coordinated Attack Leading to Victory Uncoordinated Attack Leading to Defeat

Blockchain Consensus ⊆ BA

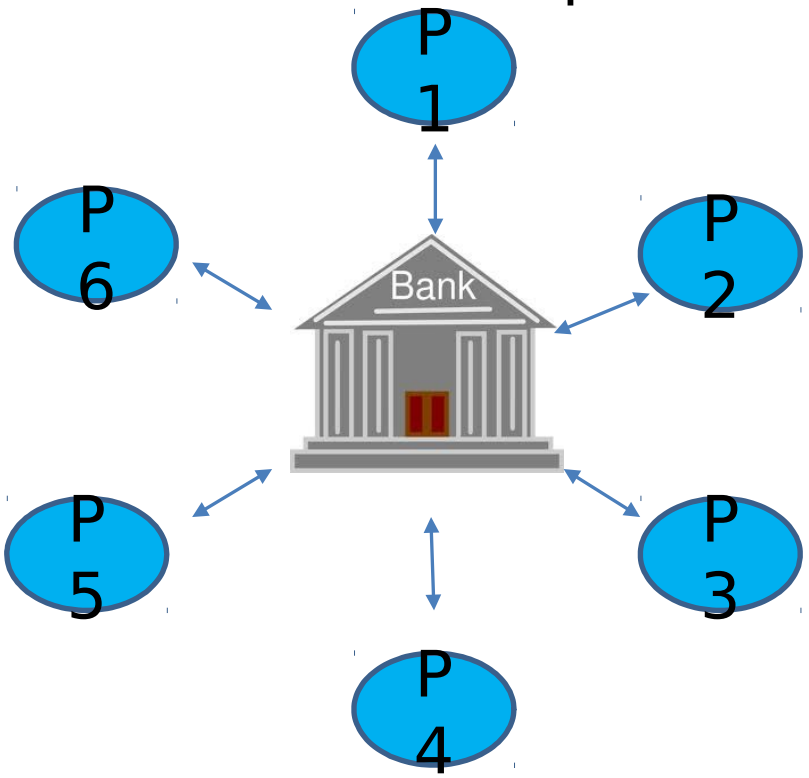
The Commercial Relevance



The Blockchain Consensus Problem

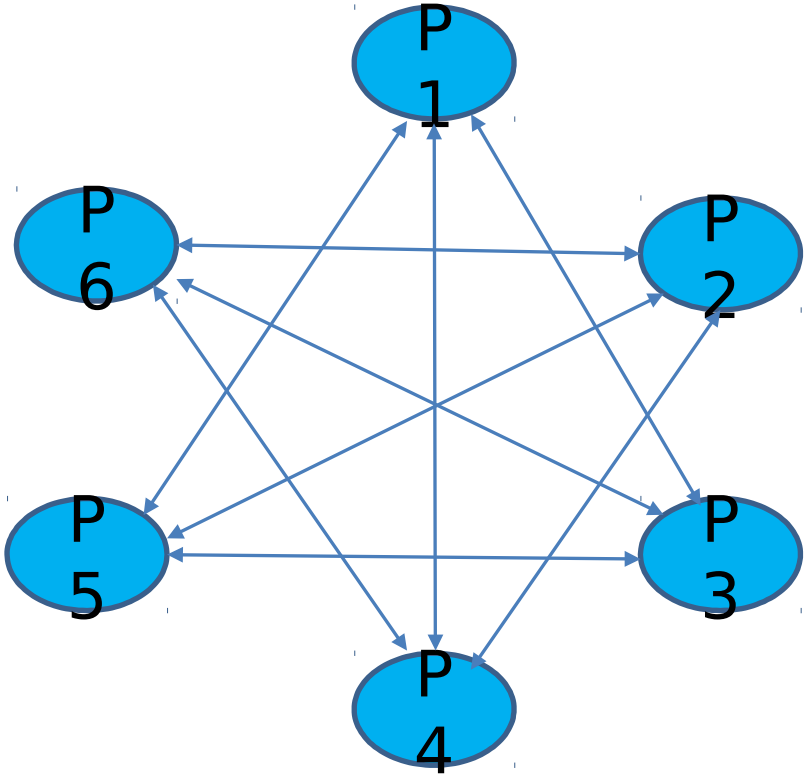
No Centralized Trust

Centralized Operator



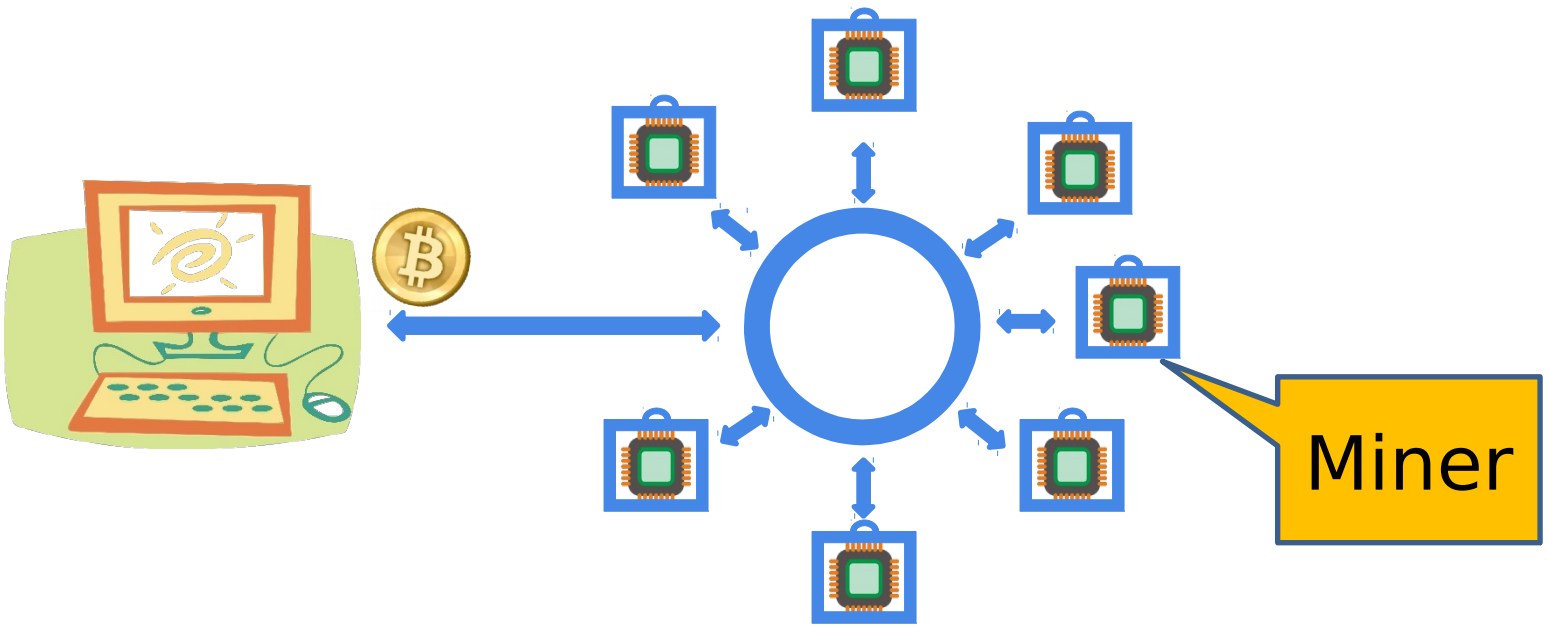
Traditional

De-Centralized Operators



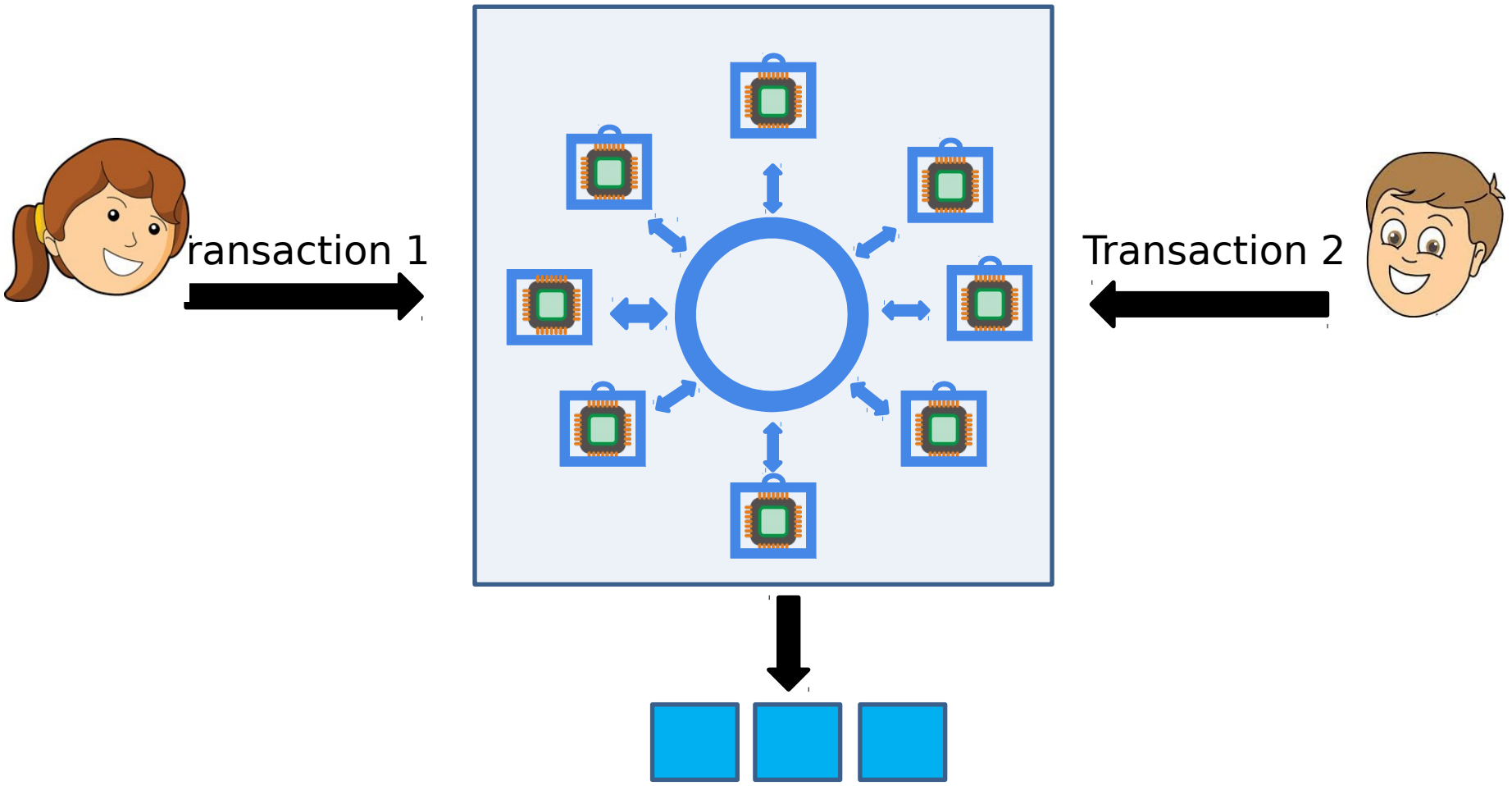
Blockchains

Blockchains: A network of “miners”



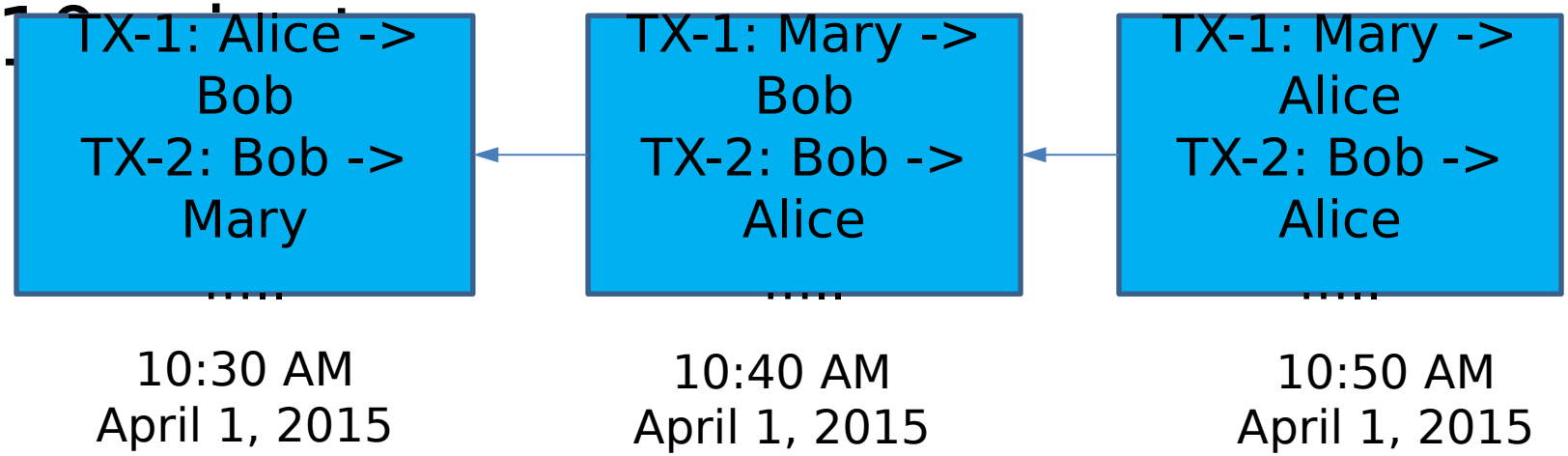
- Permissionless
  - Anyone can join / leave without centralized co-ordination

Goals of A Blockchain



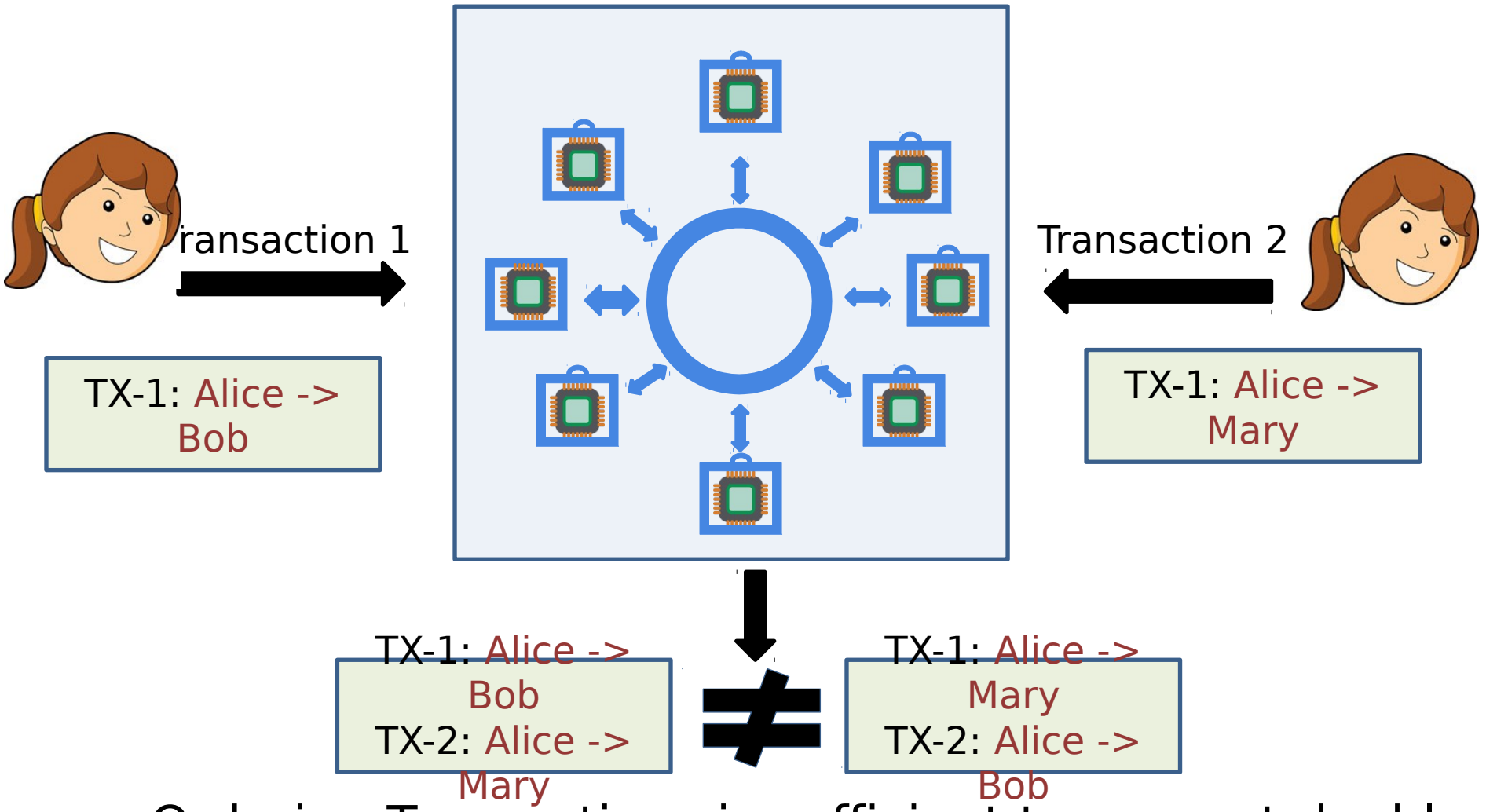
Goals of A Blockchain

- A continuous process… 1 block every 10 minutes



- Transactions are totally ordered in “blocks”
- Blocks are totally ordered in time
  - Anyone can verify their order

Key Challenge: Agreement over Transaction Ordering



Ordering Transactions is sufficient to prevent double-

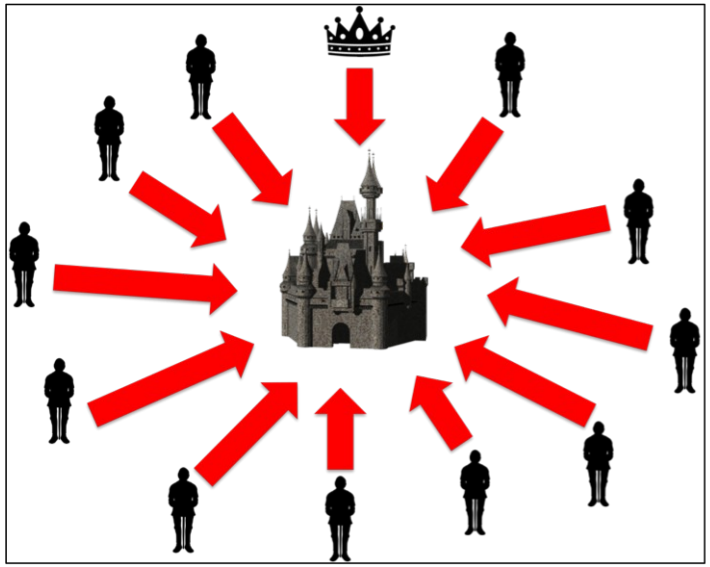
Blockchain Consensus Problem

- Assumptions:
  - Users have **no pre-established identities, anyone joins anytime**
  - A majority of miners are honest!
  - Network is synchronous (Blocks transmitted within some delay)
- Security Properties:
  - **Stability:** A block once confirmed can’t be changed
  - **Agreement:** Miners order the blocks same way
  - **Fairness:** Your confirmed blocks are proportional to the computational power you have connected
- Performance Goals:
  - **Throughput**: Lots of transactions per unit time
  - **Latency:** Short timeframe to confirm a transaction
  - **Decentralization:** Large # of miners proposing transaction blocks

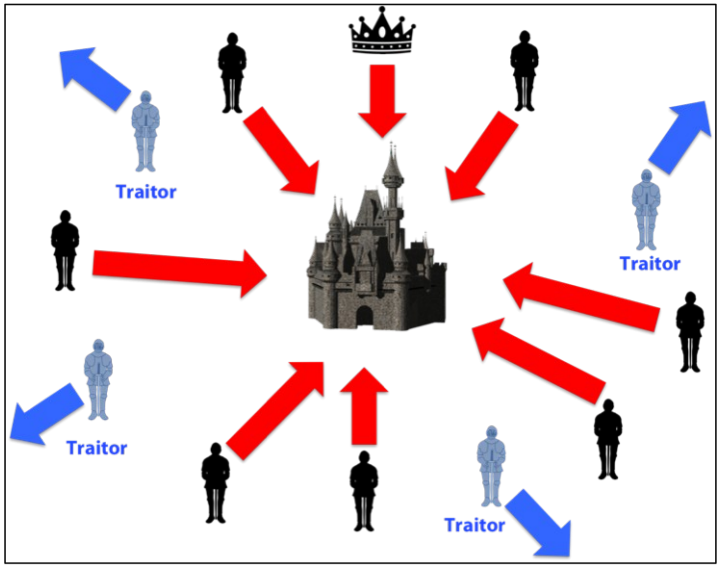
The Power Of Simplicity

Classical Byzantine Agreement (BA)

- Byzantine Agreement Problem (Lamport et al. 82):
  - A set of parties {P1, P2, …. Pn} have inputs
  - A fraction **f** out of **n** are malicious, i.e., Byzantine
  - Goals:
    - Ensure that all honest parties **agree on the same** value
    - The agreed value is **valid**, i.e. input of some honest node



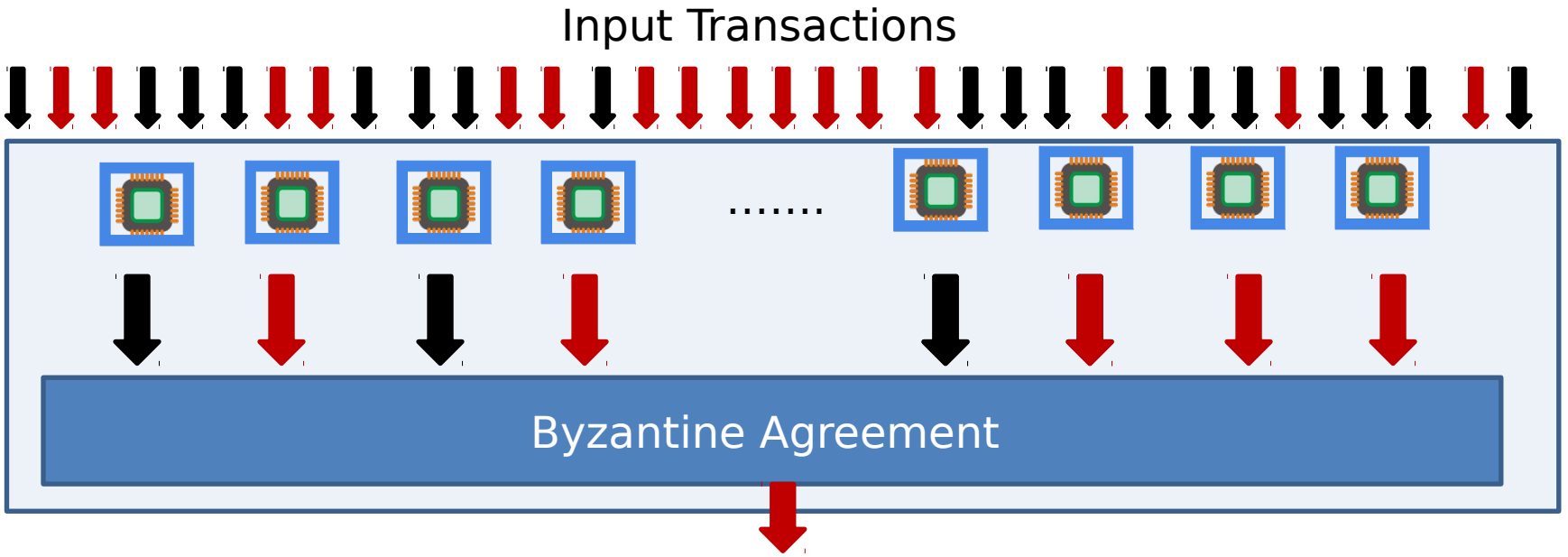
Coordinated Attack Leading to Victory



Uncoordinated Attack Leading to Defeat

Repurposing BA Protocols?

- Yes, repeated rounds of BA
- Agree on 1 block per round
- Honest miners sign that block with round id.



- Challenge: Participants must be known a-priori
  - Chicken-n-egg: Agreeing on participants is itself…

Caveat: BA Protocols Are Complex

- A philosophical viewpoint
  - Simplicity mattes in practice
- Recent Design Flaws:
  - Zyzzyva [SOSP'07] is a landmark fast BFT protocol
  - A flaw found 10 years later [Abraham et al. - arxiv2017]
- Blockchain Consensus is a *simpler* BA solution
  - Mild assumption: parties have equal computation power

Bitcoin's Solution: Nakamoto Consensus Protocol

- Miners keep a local copy of the blockchain
- Miners solve a computational Proof-of-Work puzzle:

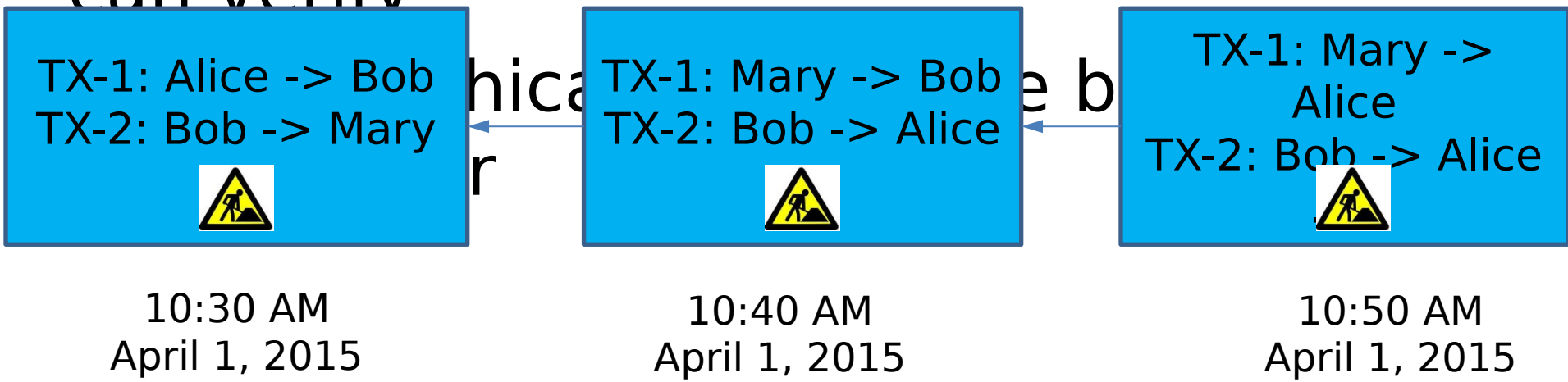
- Successful miners (usually one) broadcast solution
- Miners check the received solutions, and if valid:
  - Extend their chain with that block
- Confirm block on the longest chain after it is k-deep
    - Bitcoin proposes k = 6

Computational Puzzles as a Sybil Defense

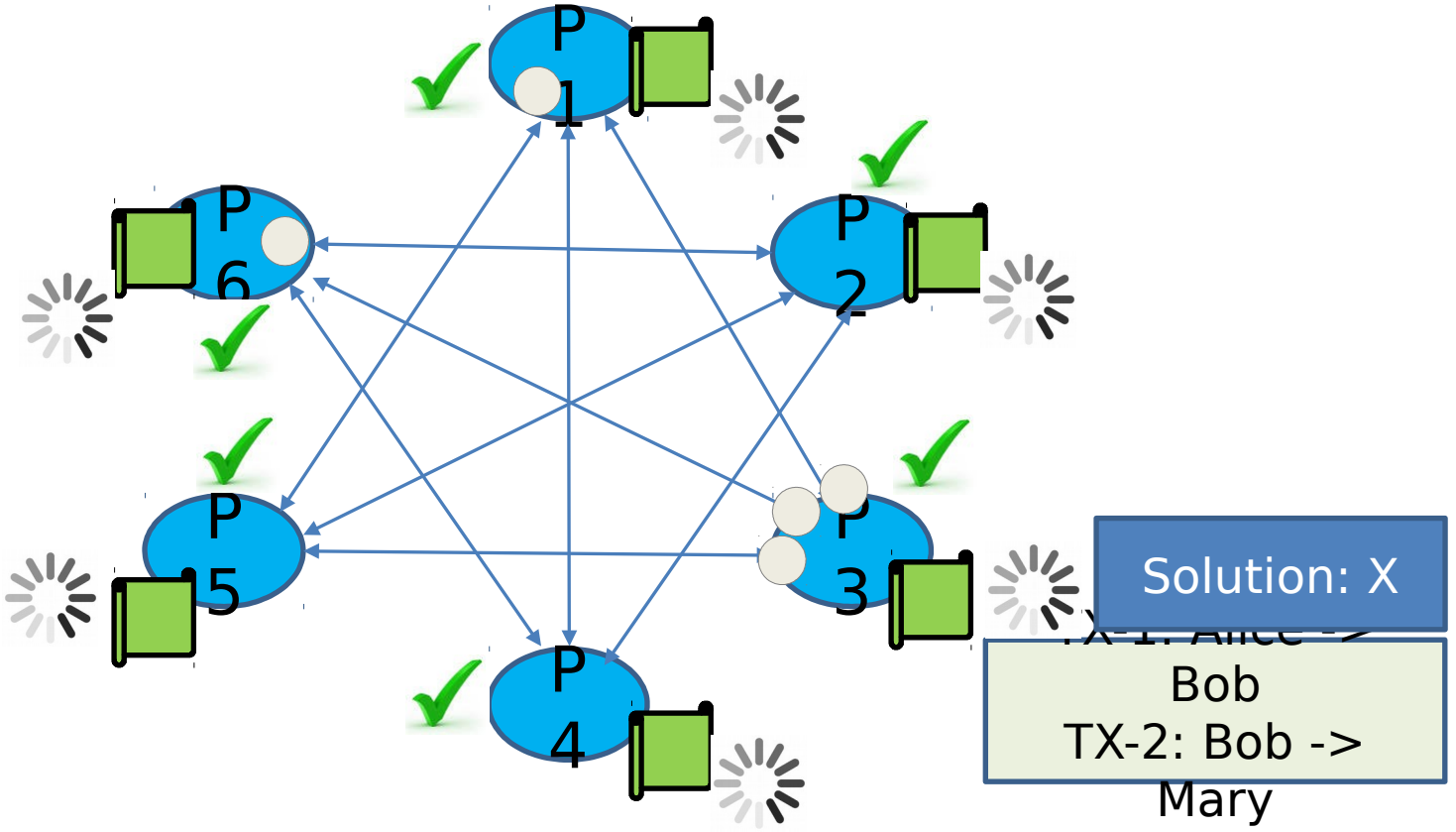
- Puzzle X: Compute “s” such that

  H(**s** || last_block_hash || new_block) < **d**

  - “**d**” is the number of leading zeros desired
  - “**d**” adjustable, based on the mining power (last block interval)
- Consumes power to solve, but anyone can verify



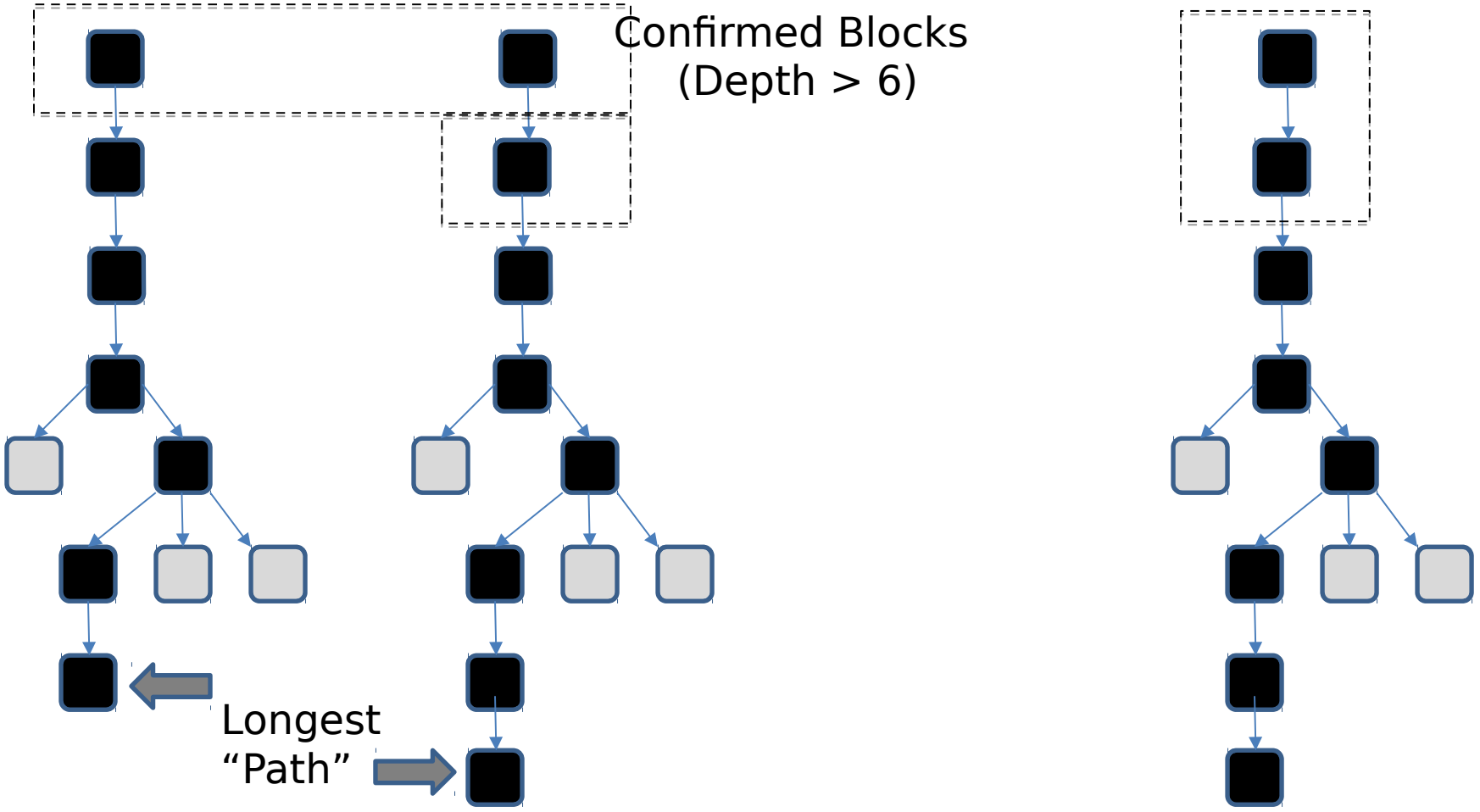
Nakamoto Consensus: Overview



PoW solver (block founder) is a **leader**. Everyone accepts his solution, if valid.

- We didn't know how many computers connected, yet we elected one block!

Nakamoto Consensus: Overview



Miner A Local View

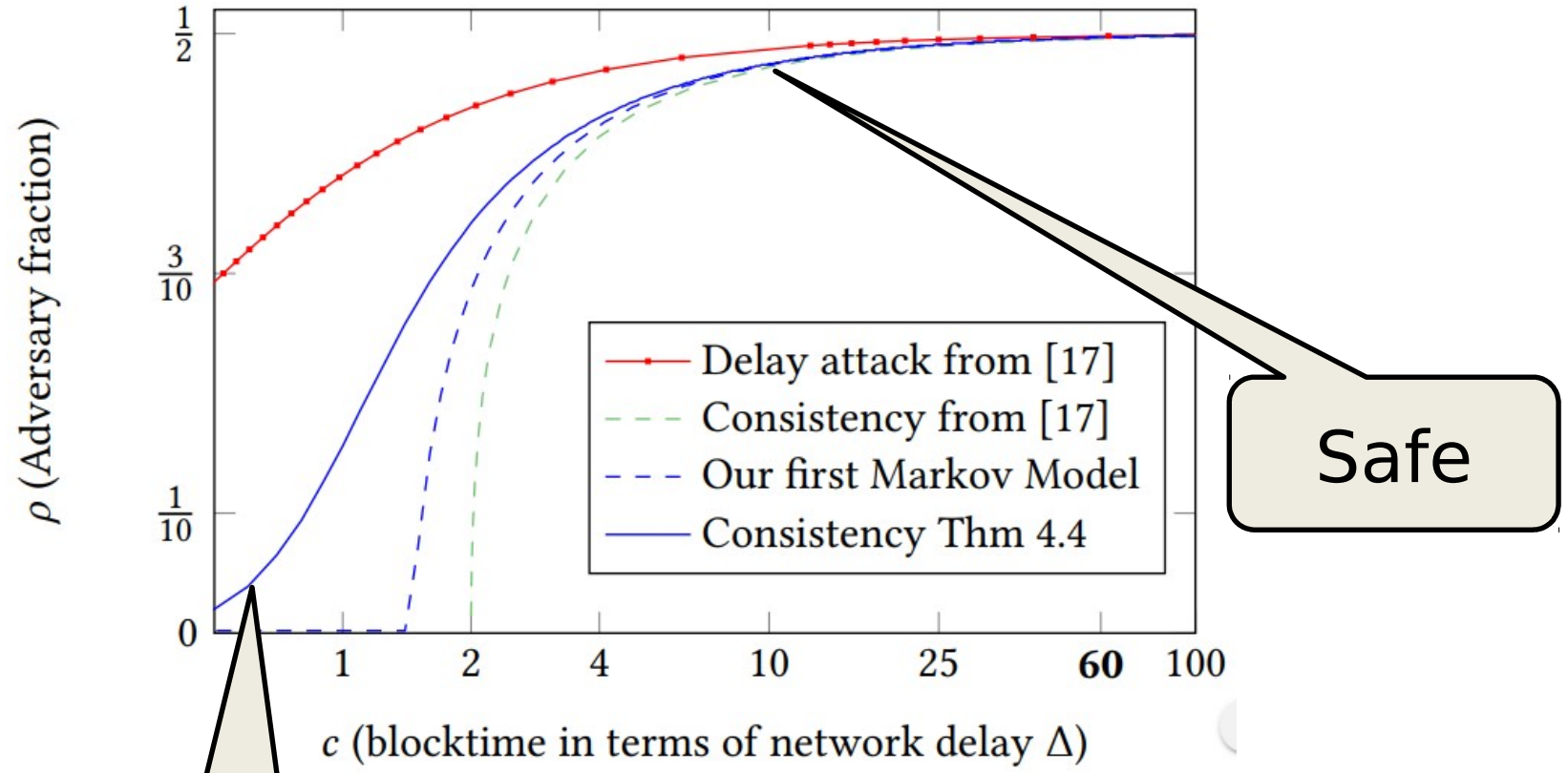
Miner B Local View

Combined System View
(Taking comp

Why Simplicity Matters...

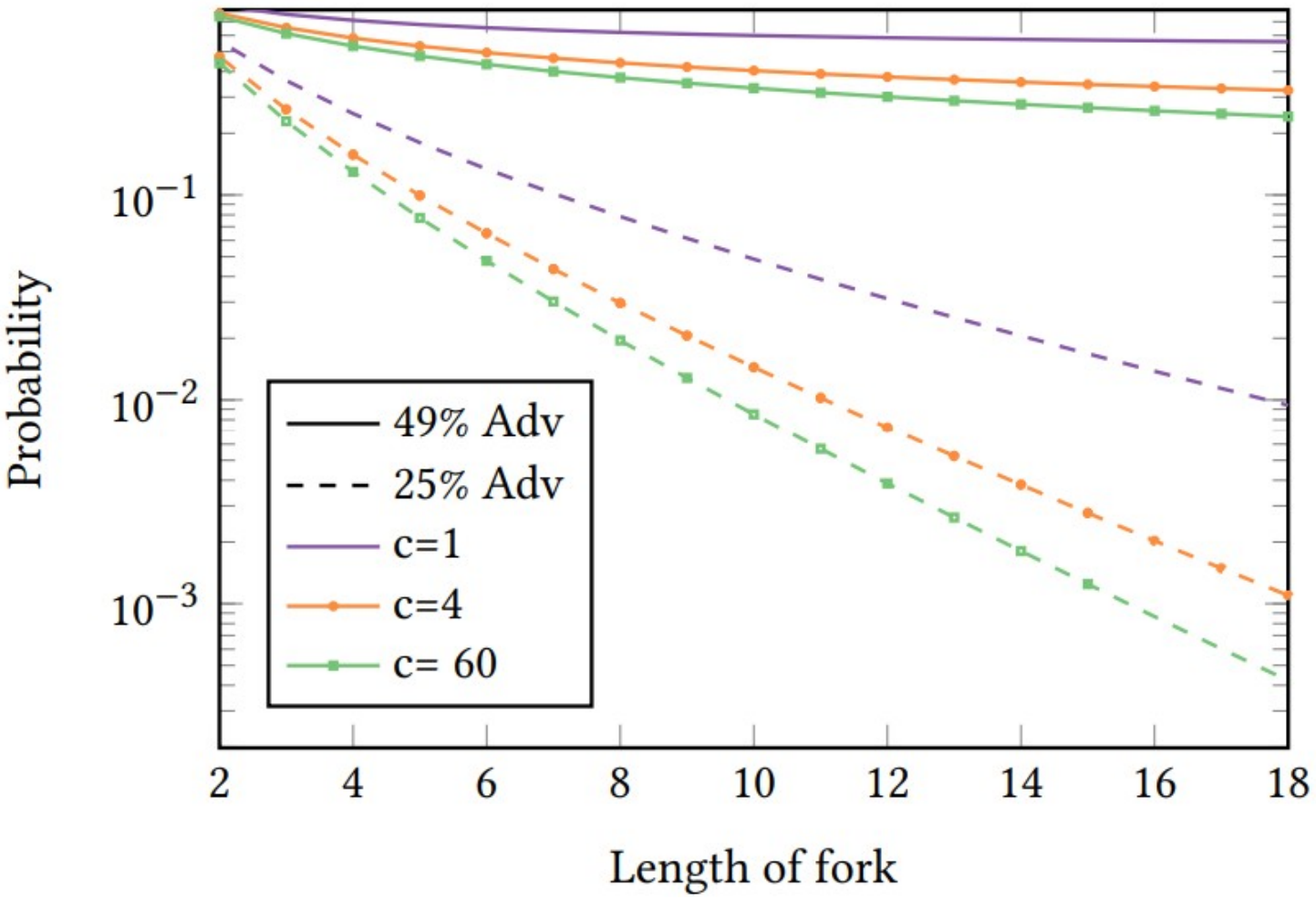
- Admits analysis and proofs
- **Safety & Liveness** holds for Nakamoto
  - Certain large parameter values must be chosen
- Rough outline of proof:
  - Define Epoch as one “block propagation delay” (BPD)
  - Count “Good” vs. “Bad” events
    - Good: A single block is mined in a epoch by honest miners
    - Bad: More than one block mined in an epoch
    - Bad: Malicious miner mines one block more than honest
  - Show that union of all “bad” events happen with negligible probability in “k”

Carefully Established Results



hs from Kiffer, Rajaraman, Shelat – CCS'18. (Also see EuroCrypt'15, TACT'17)

At high block rate, forks are likely...



hs from Kiffer, Rajaraman, Shelat – CCS'18. (Also see EuroCrypt'15, TACT'17)

Research Challenges (I)

Security vs. Performance

bitcoin

- 2-4 Kilobytes / second
- 6-12 TXs per second
- 3-60 minutes latency

ethereum

- Support limited computations
- Outages and Unavailability
- A cryptoKitties app clogged the entire network

Demand from Practice: 1,200 - 50,000 TXs/s

PayPal VISA MasterCard

Security vs. Performance

- Goal: Show all properties simultaneously:
  - Near-optimal **Throughput**
    - Scale up to a constant fraction of available bandwidth
  - Near-optimal **Resilience**
    - Byzantine adversary with power fraction f < 1/2
  - **Decentralization**
    - Many block proposers per second, difficult to attack/bribe
  - Low **Confirmation Latency**
    - “The Buy Coffee” Problem: Latency below 15 epochs

Security vs. Performance: State-of-the-art

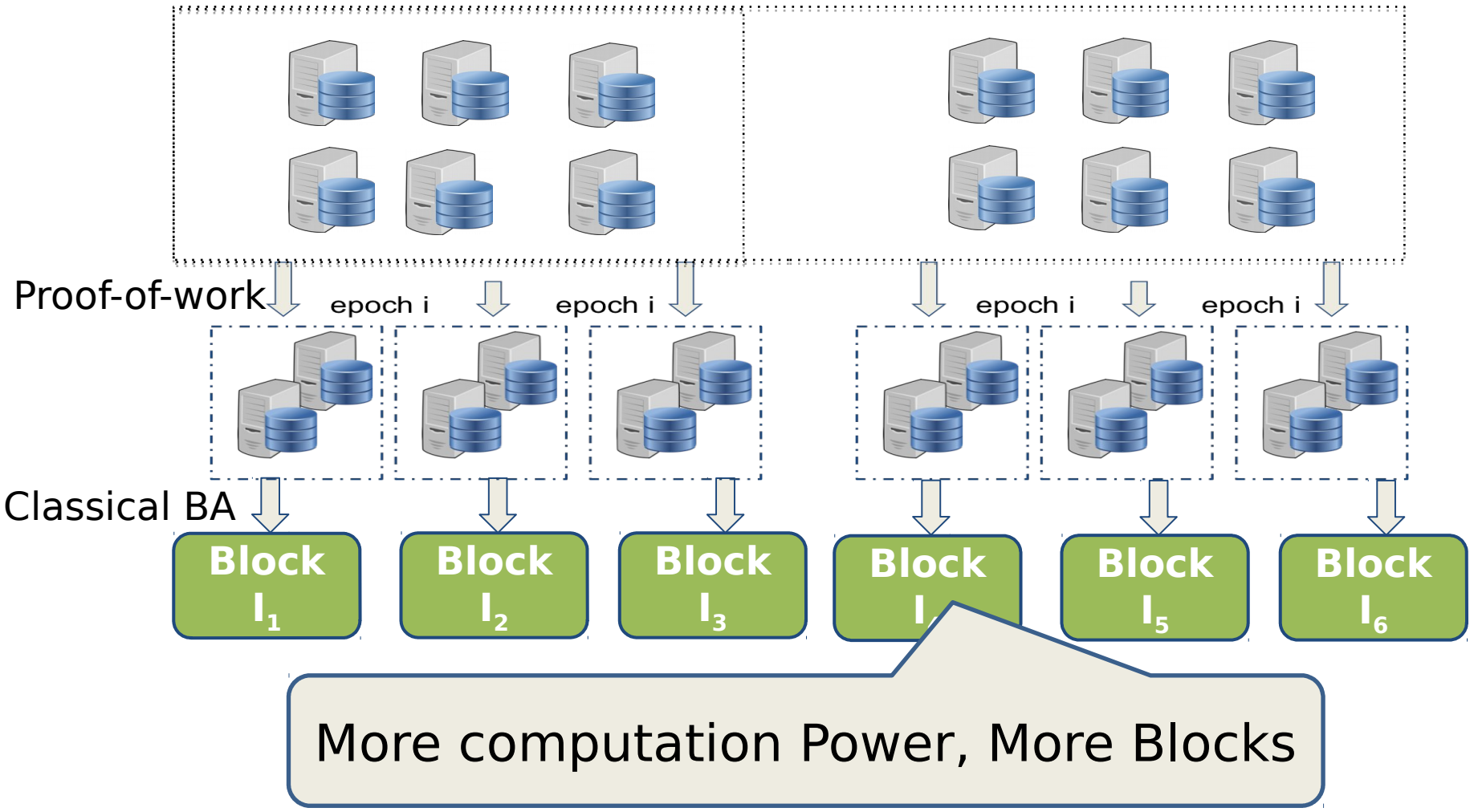
Approach	Resilience	Throughput	Decentralization	Latency
Nakamoto with reduced block intervals	$f < \frac{1}{3}$	Low	Medium	Good
Nakamoto with large blocks	$f < \frac{1}{2}$	High	Low	Medium
AlgoRand (with BA) [SOSP'17]	$f < \frac{1}{3}$	High	Low	Good
Sharding (with BA) [CCS'16, S&P'18,CCS'18]	$f < \frac{1}{3}$	**High**	**Medium**	**Good**
Parallel Chains [arxiv'18]	$f < \frac{1}{2}$	**High**	**Good**	**Medium**

50 proposers per sec

30 secs.

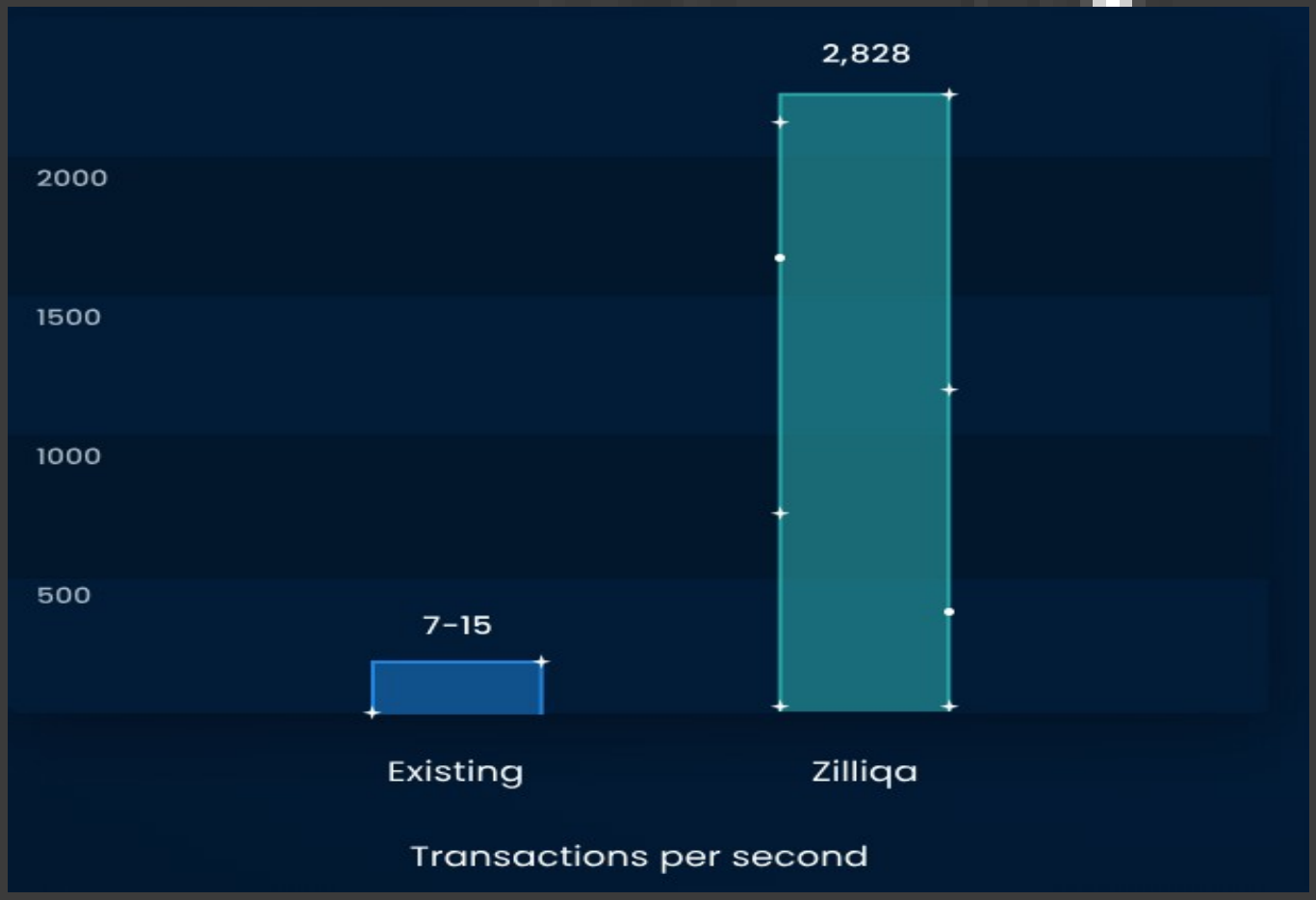
10 mins

Our Solution: Blockchain Sharding



Elastico – CCS'16 (Also see Omniledger – Oakland'18,

Commercialized as the Zilliqa blockchain



OHIE: Composing Parallel Chains

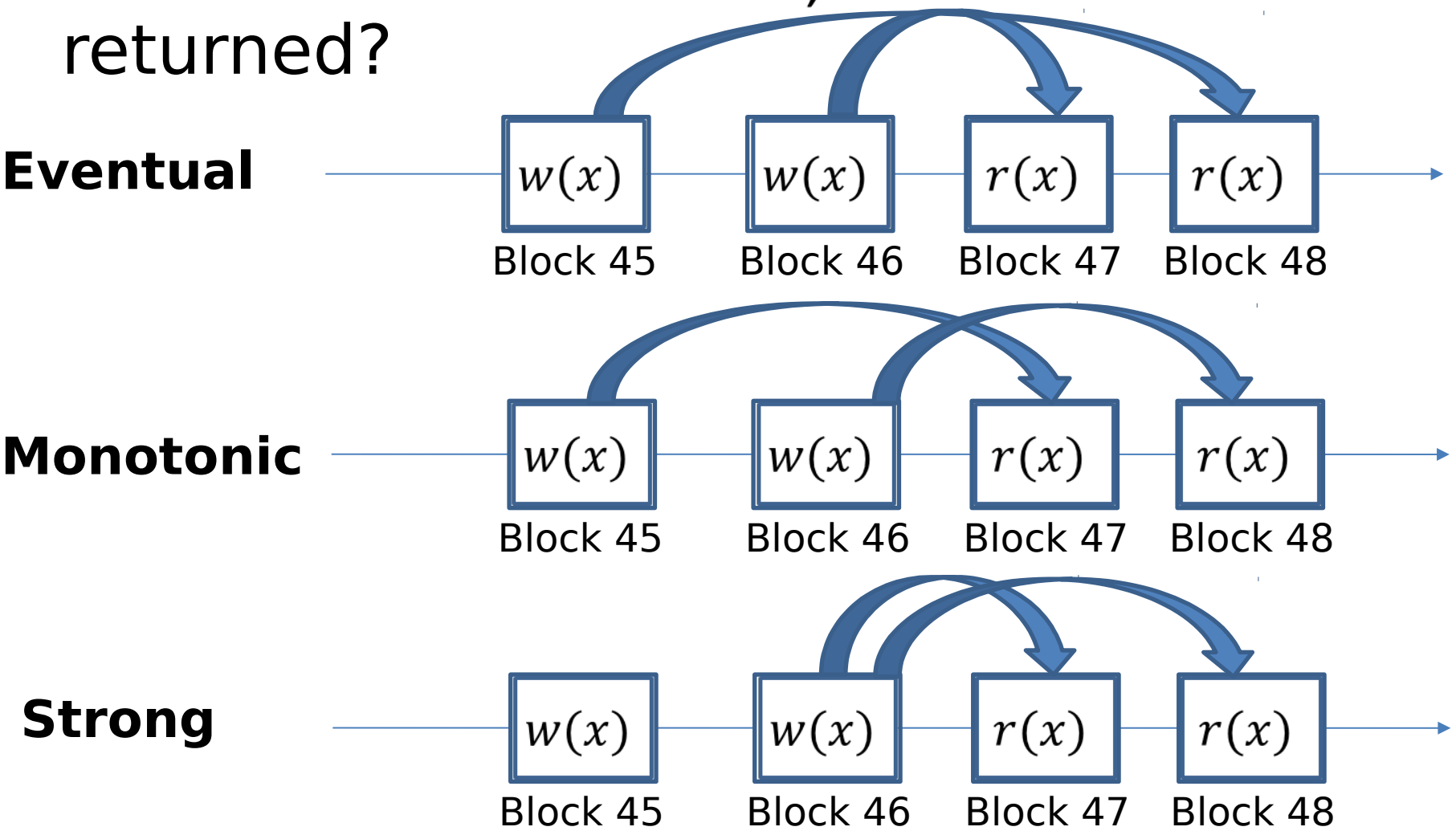
- Near-optimal throughput
- High Resilience: $f < \frac{1}{2}$
- High decentralization
  - 20x over prior constructions
- Confirmation Latency:
  - 2x of Nakamoto
- Modular and Simple
  - Full proofs of safety and liveness

OHIE: Blockchain Scaling Made Simple – **Yu e**

Research Challenges (II)

Defining the Consistency Model

- If a smart contract reads x, which write is returned?



Do developers understand consistency?

'$300m in cryptocurrency' accidentally lost forever due to bug

33

Etherdice is down for maintenance. We are having troubles with our smart contract and will probably need to invoke

Over 34,000 Ethereum Smart Contracts Found To Be Vulnerable

Blockchainist. Former poker pro.
Jul 21, 2017 · 16 min read

A hacker stole $31M of Ether—how it happened, and what it means for Ethereum

Transaction Ordering Inconsistencies

```
1 contract MarketPlace{
2   uint public price;
3   uint public stock;
4   /.../
5   function updatePrice(uint _price){
6     if (msg.sender == owner)
7       price = _price;
8   }
9   function buy (uint quant) returns (uint){
10    if (msg.value < quant * price || quant > stock)
11      throw;
12    stock -= quant;
13    /.../
14  }}
```

Two transactions, one to updatePrice () and one to buy(), will have different results based on the order in which they're present in the final order.

- Oyente: Detected Bugs In Existing Smart Contracts
  - Run with **19366** contracts, **3056** due to re-ordering TXs
  - 30 mins timeout per contract

Towards Efficient Detection Techniques

- Multi-Transaction Vulnerabilities
  - Run with **970,898** contracts
  - **10 seconds** timeout per contract

Category	**#Candidates flagged** (*distinct*)	Candidates without source	#Validated	% of true positives
Prodigal	**1504** (*438*)	1487	1253	*97*
Suicidal	**1495** (*403*)	1487	1423	*99*
Greedy	**31,201** (*1524*)	31,045	1083	*69*
Total	34,200 (2,365)	34,019	3,759	89

Over 34,000 Ethereum Smart Contracts Found To Be Vulnerable

MAIAN – Finding The Greedy, Prodigal and Suicidal Contracts – ACSA

More Challenges & Future Directions

- Bitcoin consumes more electricity than Ireland!
  - Switch to non-computational Sybil defenses (PoS)
  - Fundamental tradeoffs between PoW vs PoS?
- Moving Computationally Intensive Tasks Off-chain
  - Trusting off-chain computation?

Takeaways

Takeaways

- Open Decentralized Systems are a new area…
  - No centralized trust assumptions, permissionless
- The Power of Simplicity
  - Helps the practitioner and in establishing confidence via proofs
- Many advances trading off between ideal properties
  - Yet to see an optimal solution! (Low latency, high decentralization)
- Need for new models and drawing new connections:
  - Consistency properties
  - Sybil resistance mechanisms
  - Incentive mechanism design

Thank you!