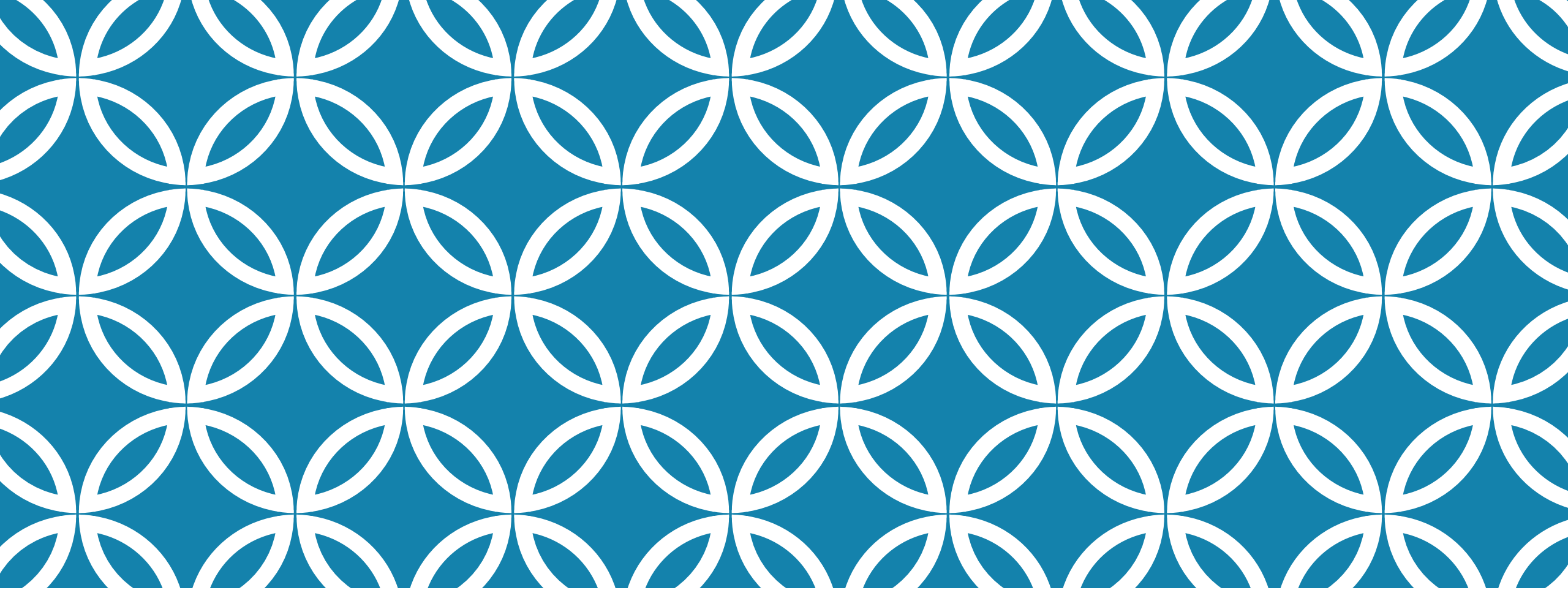


BLOCKCHAINS FOR IOT

Ken Birman

BLOCKCHAINS FOR IOT

Lucas Mearian. Not afraid of hyperbole!

What is blockchain? <u>The most disruptive tech in decades</u>!

"The distributed ledger technology, better known as blockchain, has the potential to eliminate huge amounts of record-keeping, save money and disrupt IT in ways not seen since the internet arrived."

Lucas Mearian, ComputerWorld Staff Writer

A MORE TECHNICAL ANSWER?

"A blockchain, originally block chain, is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle Tree root hash).

By design, a blockchain is resistant to modification of the data."

Wikipedia

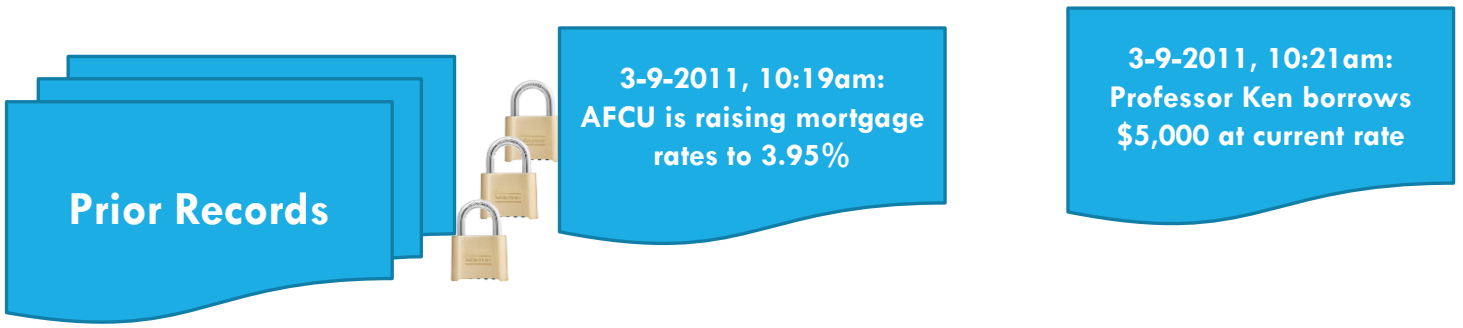
BEST USES FOR BLOCKCHAIN?

Secure, trustworthy shared log (append-only file)

Records describe some form of announcement or transaction of broad value to the users. There are standard (web-like) languages for recording things in these records.

Example: Bank announcements of variable mortgage loan rates. Public announcements of weddings, divorces, births, deaths.

BEST USES FOR BLOCKCHAIN?



BEST USES FOR BLOCKCHAIN?

If the blockchain is public, shared and tamperproof there can never be any basis for disagreement about the information.

The blockchain is publicly replicated to ensure that even the bank can't cheat.

Cryptography prevents tampering

… OR AIRPLANE MAINTENANCE LOGS

The New York Times

Hundreds of Planes Are Stranded in Russia. They May Never Be Recovered.

Western companies that own the planes face little prospect of getting them back, meaning billions of dollars in losses.

While a few planes may have been recovered abroad before international flights were halted, they are of little use to their owners without the meticulous maintenance records that accompany every aircraft and are often stored by airlines themselves, experts said. And the longer a plane is stuck in Russia, the greater the concern that work on the jet's body, engines and flight systems may not be logged, causing its value to plummet.

"Unless you have those records, the aircraft is virtually worthless," said Quentin Brasie, the founder and chief executive of ACI Aviation Consulting. "They're literally more important than the asset itself."

TERMINOLOGY

In BlockChain settings, a *transaction* is a digital record describing some event.

Some transactions are complete and self-contained.

But BlockChain also supports transaction languages in which one transaction might refer to events defined by a past *or future* transaction.

CRYPTOGRAPHIC HASH

A cryptographic hash is a bit string computed from some block of data in a manner that yields a constant-length result irrespective of the data size, and yet such that it would be infeasible to find other data that would hash to the same result.

There are a number of hashing schemes. A highly robust one is SHA-256. SHA-512 is even stronger. MD5 and SHA-1 have been compromised and are unsafe.

EVEN FASTER HASH METHODS EXIST

(single core performance, all Golang implementations, see benchmark).

BenchmarkHighwayHash	11,986 MB/s
BenchmarkSHA256_AVX512	3,552 MB/s
BenchmarkBlake2b	972 MB/s
BenchmarkSHA1	950 MB/s (insecure)
BenchmarkMD5	684 MB/s (insecure)
BenchmarkSHA512	562 MB/s
BenchmarkSHA256	383 MB/s

Note: the AVX512 version of SHA256 uses the multi-buffer crypto library technique as developed by Intel, more details can be found in sha256-simd.

https://blog.minio.io/highwayhash-fast-hashing-at-over-10-gb-s-per-core-in-golang-fee938b5218a

HARDWARE CAN GET EVEN FURTHER

FPGA and ASIC solutions can be purchased that will run SHA-256 or SHA-512 at speeds of 25,000 to 30,000 MB/s

In some parts of the world there are entire datacenters equipped with huge numbers of these accelerator solutions. China dominates the business.

Very hard to be a BlockChain miner with a single desktop computer today!

CRYPTOGRAPHIC SIGNATURE: ENCRYPTED HASH

Given a message, anyone can compute the cryptographic hash for it

Some applications need a way to sign a document. It is important that this kind of signature be "irrefutable".

For this, we first compute a hash, then encrypt it in a special way so that only the signatory could encrypt it, yet anyone can decrypt and check it.

PUBLIC/PRIVATE KEY PAIR (NORMALLY, RSA)

This is a classic cryptographic method.

RSA creates two "keys", both just long numbers together with a modulus *n* that itself is a product of two very long prime numbers. Call them K, $\overline{K}$

One is designated as the public key and shared. You keep the other private.

$$RSA_K(RSA_{\overline{K}}(X)) = RSA_{\overline{K}}(RSA_K(X)) = X$$

WHY RSA WORKS

In 1796, Gauss came up with the theory that ultimately gave us the (very simple) RSA technology. Gauss himself didn't suggest this application.

In RSA encryption and decryption are just mathematical steps that involve a form of "bignum" arithmetic (modular exponentiation), performed block by block.

RSA is secret because there is no known method for factoring a giant composite number that might have 1000's of binary digits. If we could factor the modulus, it would be trivial to recover the secret key from the public one.

Quantum computers *might* offer a path to doing so, but it would require devices with millions of qbits, way beyond anything feasible anytime soon.

RSA STRENGTHS, WEAKNESSES

Very widely supported, basis of most "certificates" used in the Internet.

Many tricks exist, based on commutativity of RSA computation. Basically, for *any* two RSA keys, A and B, $RSA_A(RSA_B(M)) = RSA_B(RSA_A(M))$.

But RSA is fairly slow. The speed is a function of the data size. We don't casually encrypt entire messages with RSA: it would be feasible but slow.

HOW WOULD PROCESS P SIGN MESSAGE M?

1. Compute the SHA-256 hash of M.
2. Now use P's private key to encrypt the hash: $SHA(M)_{private\text{-}key\text{-}of\text{-}P}$

Process Q can easily verify that M has not been tampered with:

1. Q recomputes the SHA-256 hash for M
2. Now Q uses RSA with P's public key to crypt P's signature.
3. If they match, then Q has confirmed that M hasn't changed.

NOTARIZING BLINDED DATA

There is even a method, by David Chaum, for signing an object that the signatory cannot see. It would be useful for secure voting:

- Prepare your ballot, then blind it (encrypt) and obtain a signature.
- The signature is proof that your vote was valid and only cast once. Submit it for counting now, unblinded, via a secure anonymous "onion route"
- The ballot itself has no identifying information, and neither does the signature. So a third party can see that your vote is valid, and can count it, and yet can't learn how any particular individual voted.
- Chaum also showed how to get a receipt which can be used to be sure your vote was properly tabulated.

PARALLELISM?

A further win is to maximize parallelism and reduce record sizes.

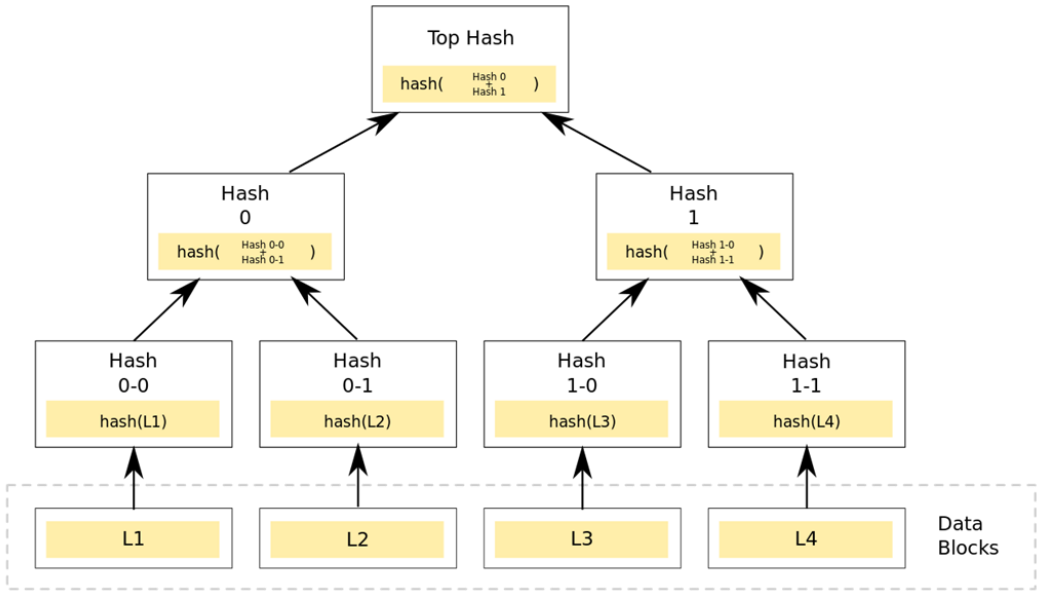
In the case of BlockChain, each block contains a set of transactions represented as binary records. These records might be huge, hence slow to hash (and very slow to encrypt, were you to try that).

By having the creator store the record someplace reasonable and then just storing signatures in the BlockChain, we use it as efficiently as possible.

MERKLE TREE: A TREE OF SIGNED RECORDS.

Rather than making one list of N records and then hashing them, we often create a binary tree of hashes.

Very common in BlockChains, permits us to run SHA-256 or SHA-512 in its fastest "mode" of operation.



Often we think of the entire Block chain as a sequence of Merkle trees that change (only) by appending new subtrees (which also changes the root node)

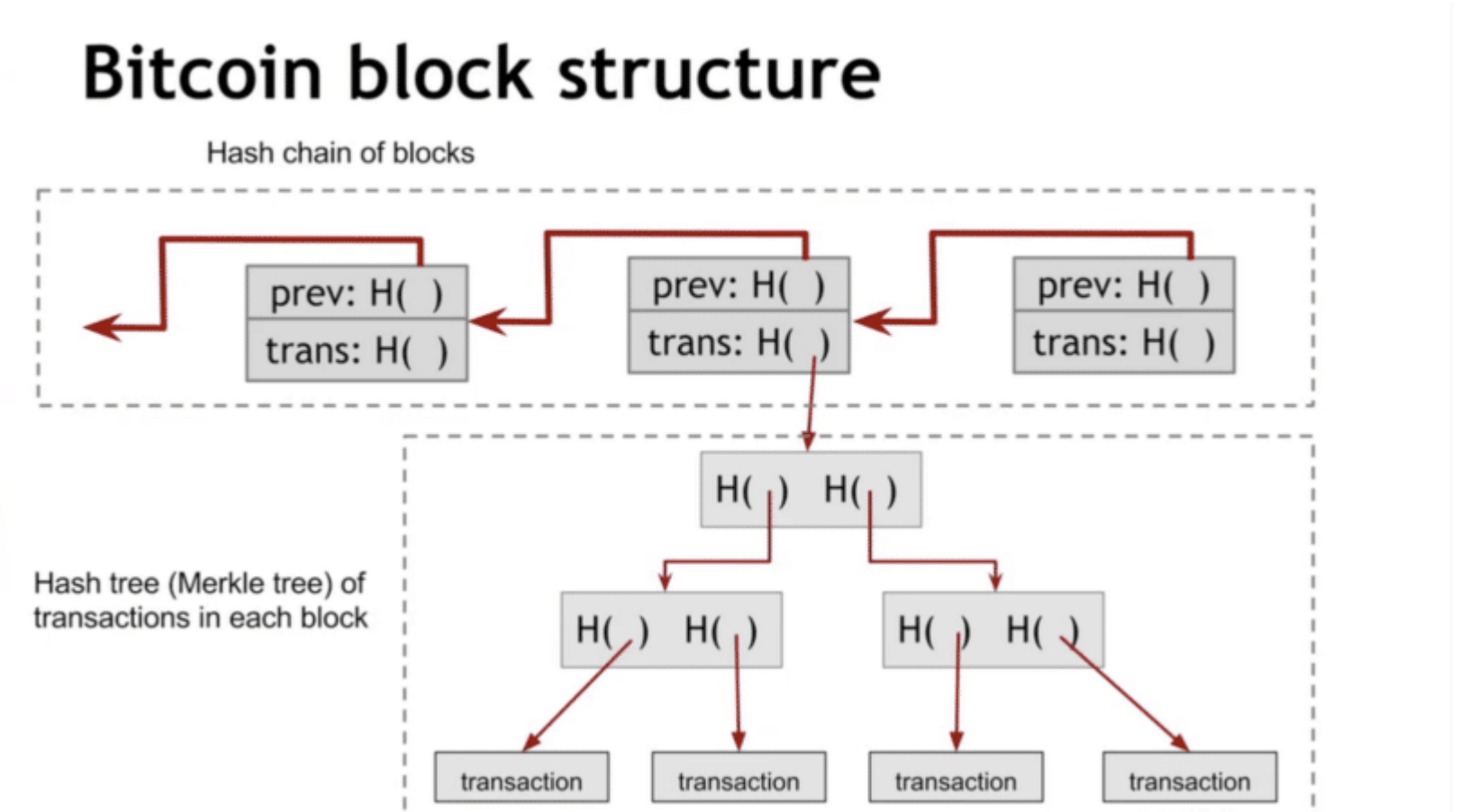
THIS ALREADY GIVES US A BASIC SOLUTION!

Compute a series of records, each containing transactions signed by the initiator. The record needs to include the "name" of the initiator so that anyone needing to do so can look up the matching public key.

Associate each record with a key for lookup, and insert the (key,record) objects into the Merkle tree.

Then create some form of cryptographic proof that the new tree extends the prior tree. Logs are just one possible representation.

OUR BASIC SOLUTION



https://coincentral.com/merkle-tree-hashing-blockchain/

WHY IS THIS SECURE?

If anyone tampers with any record in the chain, we can sense this by recomputing the Merkle tree. The signature won't match.

To verify the entire chain, block by block recompute the Merkle tree, then recompute the sequence of pairwise hash values.

Verification is required when an untrusted BlockChain is initially loaded.

PERMISSIONED/PERMISSIONLESS

BlockChain solutions split into two categories.

A **permissioned** BlockChain is managed by an authorized group of servers, which could be geodistributed or inside some datacenter. We generally assume that they don't have true Byzantine faults.

A **permissionless** BlockChain is managed by an anonymous group of servers that volunteer to play the role, and might come and go at will.

… AND TWO MORE SUB-CASES FOR THE PERMISSIONED CASE

Permissioned blockchains can arise in two settings.

- In a data center, just use Paxos plus logic to compute the needed cryptographic signature chain. WAN mirrors protect against attempts to compromise the whole data center and rewrite the whole log.
- At geoscale, a permissioned blockchain just means the miners get an authorization to mine – it can be withdrawn if they misbehave.

When people say "permissioned blockchain" they normally mean "in a wide-area environment, with Byzantine behavior controlled by the permissioning key-management technique."

ATTACKS ARE AN ISSUE!

Wide-area replicated BlockChains can come under many forms of attack.

- Compromised server might try to hand out "fake" versions of the chain.
- It might try to generate huge rates of transactions on its own, and not include your transactions. This is a form of DDoS attack.
- It could try to corrupt individual transactions or records.

PROOF OF WORK

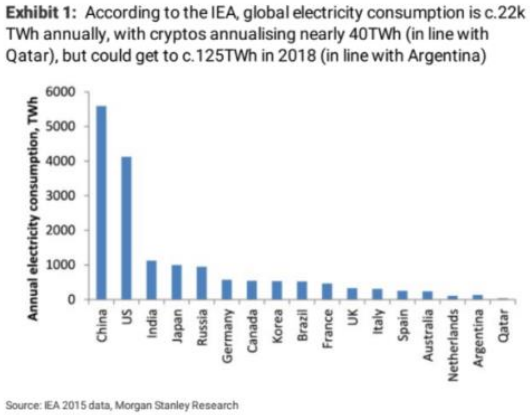
A Proof of Work mechanism adds one more field to the blocks: a "nonce".

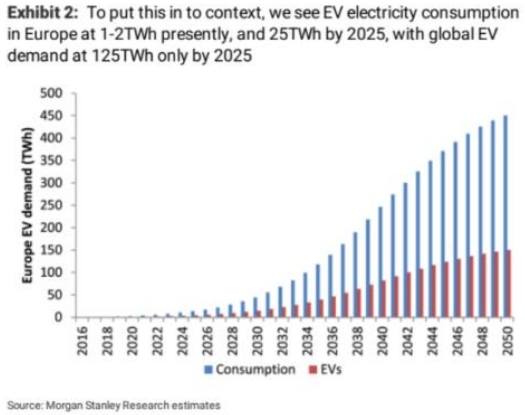
The nonce is just a bit string of some size.

The rule is that to append B_{k+1} to the chain, in addition to hashing it with the hash of the prior block, P must also find a nonce such that when the nonce is included and a new hash is computed, the hash value ends with some desired number of 0 bits.

PROOF OF WORK

Finding such a nonce is hard work!





So while P could be keen to append its block, it may need to search for this nonce for many seconds or minutes.

The difficulty of finding a nonce value that will work prevents DDoS attacks.

… OTHER IDEAS HAVE BEEN TRIED, BUT ARE NOT VERY SUCCESSFUL

For example, proof of "stake" centers on the idea of doing mining in a way that favors wealthy miners who can prove they are big players.

The "stake" is like a printout of a bank account. "I own fifty buildings in New York City, seven golf courses, a hotel chain… I mined more blocks that Bill Gates and Jeff Bezos combined. I am "Mr. Big." So. you can trust me to mine blockchain blocks (and to keep all my profits from doing that").

Not very democratic. And so, not very popular!

HOW MOST BLOCKCHAINS WORK TODAY

You can download the software as a VM or even compile it yourself.

You launch the code on your own servers – this makes you a "miner" as soon as the system has initialized itself.

The system downloads the entire current BlockChain, from other machines already running the BlockChain software (there is a web site listing some you can contact for copies).

ASIDE: WHY THE WHOLE TREE?

There is a lot of research (meaning, a huge amount) on ways to download and verify just a portion of the blockchain. None is really secure, yet.

In some settings this is going to be absolutely obligatory, but right now it isn't how the big public blockchains work.

The most promising approaches provide "countersignatures" for the portion you want, provided by authorities you happen to trust.

NEXT, YOU VERIFY THE BLOCKCHAIN

You'll need to recompute all the Merkle trees and the chain of hashes.

In fact this may not take enormously long… today. Few BlockChains have huge amounts of content.

But someday, we might have BlockChains with hundreds of billions of records and total sizes in the petabytes. Then download speed and verification time and storage will become an issue!

MEANWHILE, NEW BLOCKS ARRIVE

Each block extends some specific sequence of prior blocks.

If you turn out to have downloaded the wrong sequence, you may have to truncate your chain and download the longer sequence.

This is a “rollback”. During startup, substantial rollbacks can occur. Later they shouldn’t (assumes a fully connected network of mostly “correct” miners).

AT THIS POINT YOU CAN CREATE TRANSACTIONS

So, you open for business.

Someone shows up to buy a glass of your fresh lemonade.

You'll generate the transaction (think "credit card payment slip") and submit it to the system. It enters a pool of pending transactions.

WHEN WILL YOUR TRANSACTION GO THROUGH?

Within an hour or so, you should see that your transaction got included into some block, and also that everyone seems to have adopted that block.

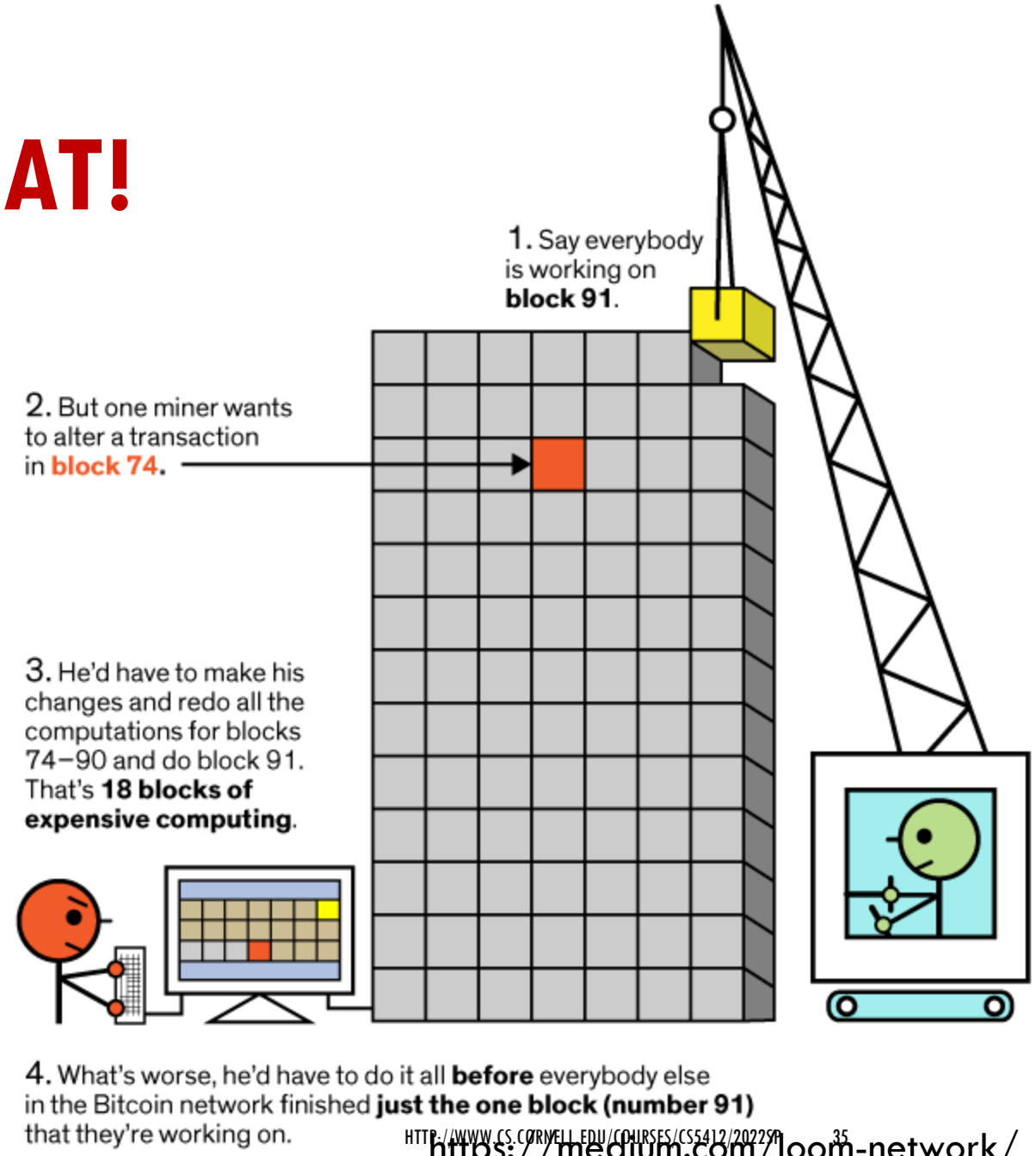
The chain has moved six or more blocks into the future.

So now you can hand that glass of frosty bliss to your happy customer!

BUT NOBODY CAN CHEAT!

To modify a past record you need to also modify every signature subsequent to that record.

The step where you have to find these nonce values will be very slow and you'll lose the race.



https://medium.com/loom-network/

WHAT IF THE ATTACKER IS A COUNTRY?

A country could build whole datacenters, equipped with hardware to compute SHA-256 at ultra-high speeds.

In this case P (using the datacenter) could generate a lot of blocks quickly, for which they would be paid. Or could have an entire second BlockChain starting from months ago, and *longer than the official main one*.

To prevent most such attacks, BlockChain solutions make the proof-of-work task harder as a function of the rate at which blocks are being found.

WHAT IF THE ATTACKER IS A COUNTRY?

In effect, if P controls enough computing power, he can "gain control" of the BlockChain. The proof-of-work can become so hard that only P has the compute power to solve the puzzle!

P could then refuse to post some transactions, or cause trouble in other ways.

But this form of attack has not (yet) been seen.

WHAT ABOUT RACES?

Permissionless BlockChains are at risk of a "race" situation in which one group of miners is working to append record R, and some other group, record S. A tie can easily occur.

BlockChain systems "adopt the longest chain" (may the best miners win). This can cause a rollback if a few blocks were appended by group A, but then group B suddenly publishes a longer extension.

In practice, rollbacks longer than 6 blocks are never observed.

IN CONTRAST, PERMISSIONED BLOCKCHAIN DOESN'T NEED PROOF OF WORK

A permissioned system is operated by known, trusted, authorized servers.

They won't attack the chain by trying to overload it with transactions in an unfair way, and they would charge for any transactions they append on behalf of external clients.

So we can avoid this costly step with datacenter BlockChain solutions.

WHAT IF YOU DON'T REALLY TRUST THE PERMISSIONED PROVIDER?

We can mix methods: a global "proof" with a local "data store"

Our permissioned provider can commit to some form of cryptographic root of each new version of the log (or tree), and to a proof that the new version extends the old version.

The "commit" is broadly shared and pins the provide down. Then for an append or a query, the provider can be asked to also provide a proof that they did the append, or that the query response is correct & complete

WHAT'S IN A TRANSACTION?

Some BlockChain systems are very rigid. For example, a BitCoin BlockChain record can only support a few operations on BitCoins.

These represent transactions: Ken sells Ittay a packet of gum for 10₿

In a permissionless scheme, Ken would probably wait for a while before handing the gum to Ittay. With permissions, rollback risk can be reduced or even completely eliminated.

FANCIER TRANSACTIONS

There are several standards for encoding fancier “digital contracts” into records suitable for BlockChain.

One, called HyperLedger, uses HTML as its underlying “language”.

A second, Ethereum, has a sophisticated language of its own, and can even encode computational tasks into the transaction record.

ONE ISSUE: VALIDATION

In existing blockchain systems, every participant maintains a replica of the entire blockchain.

But as blockchains grow, this could become unworkable. A blockchain might be hundreds of gigabytes long… users will want to download portions

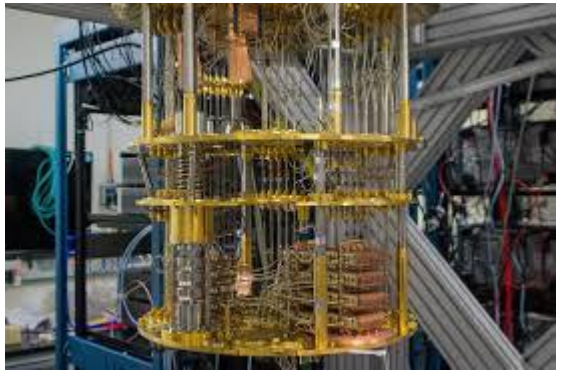
If a user only downloads the record they are seeking, how can the system validate that the *entire chain* is intact and properly signed?

MORE ISSUES

With permissionless BlockChain, is it really "safe" to trust that after six blocks have been appended, the chain won't roll back and invalidate my transaction? ("When should Ken give the lemonade to Sally?")

If a smart contract references future events, what would be the "semantics" of that contract, in a PL sense? Does the meaning depend on waiting for the future to occur? Can chains of dependencies arise, or contracts that are undecidable, or infeasibly complex to "evaluate"?

A REALLY BIG ISSUE



Quantum computers are advancing rapidly.

But many blockchains use cryptographic techniques known to be at risk if practical quantum computers emerge and can deal with really large quantum computations (on "bignum" numbers with thousands of digits).

People are asking: Is this becoming a real risk?

WILL QUANTUM COMPUTING BREAK CRYPTOGRAPHY?

Which is closer to the truth?

A quantum computer can make non-deterministic guesses, check to see if any are right (like guessing the factors of an RSA key), and then output the correct one.

- A quantum computer can compute a near-infinite number of discrete fourier transforms "concurrently", but you can only read out one data-point of the result at a time.

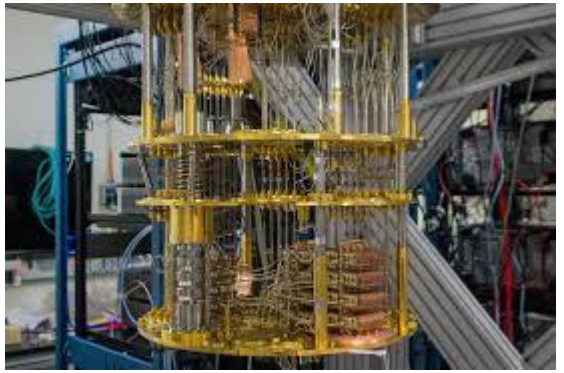
PUBLIC MISUNDERSTANDING

Popularity of the "many worlds" interpretation of physics has clouded the public conception of what a quantum computer can do!

In fact many worlds could be a valid model, for the most elementary level of Planck-scale physics (the layer where people talk about mbranes and string theory, and loop-quantum gravity).

But our macroscopic ("causally emergent") world is very remote from that most basic layer of physical reality.

SHOR'S ALGORITHM



To factor RSA, Shor's algorithm requires a special circuit specific to the size of the keys.

Then we input "all possible" n-bit integers, where n is the key length, like 1024. This involves a "coherent entanglement" of n qbits. But due to errors, qbits rapidly decohere. Error correction will require vastly more qbits, and nobody is sure how many. Perhaps millions or billions.]

The entangled data is then transformed by the circuit, which computes a DFFT

READING THE OUTPUT

You read the output of a quantum computer by setting up the experiment again and again and then repeatedly extracting a single sample.

Over time, the values you read build up to a kind of probability density image, like a photo created pixel by pixel.

In the case of Shor's algorithm this photo shows peaks that hint at the values of the factors. Now you can search for the factors close to those peaks. Quality of the search will depend on the sharpness of the peaks.

A LOT OF ASSUMPTIONS!

Nobody knows how quantum error correction "scales". Today it works for 3 to 5 q-bit entanglements, at best.

Nobody knows how complex a computation we can perform without destroying coherence. In fact these quantum DFFT operations must be reversible in order to remain coherent, and hence perfectly precise.

Nobody knows how quickly we can set up such a run and sample it.

Nobody knows how sharp the peaks will need to be as a function of key length.

AND WORST OF ALL…

Unfortunately, neither Euler nor Ramanujan really looked closely at this question!

Nobody knows if factoring large numbers is even a "hard" problem!

True, we lack a fast solution today. But the complexity of factoring is unknown.

But perhaps some numerical savant will find a solution… with classical computers! The same goes for finding a nonce with the desired hashing properties to mine blocks…

THE ENTIRE EDIFICE COULD COLLAPSE!

If you bet heavily on BlockChain, you are betting that people will figure out a way to ensure that it won't yield to some kind of attack.

But in fact this is just a bet, today.

PROVABLY SECURE SYSTEMS

There is a theory of **semantic cryptography** safe against quantum attacks. It was developed by Goldwasser and Micali, who won the Turing Award for the insight.

They proved that secure encryption schemes must be probabilistic, rather than deterministic, with many possible encrypted texts corresponding to each message.

The **Goldwasser–Micali** (GM) lattice cryptosystem demonstrates the idea.

COULD A BLOCKCHAIN USE LATTICE CRYPTOGRAPHIC TECHNIQUES?

At present, lattice cryptography is too computationally slow for practical use, and also causes too much "inflation" in the size of data.

Each bit in the data becomes a point in a very high dimensional space, leading to a billions-to-one increase in message sizes.

But continued research may yield much more compact solutions with the same properties. A new research initiative just started on this topic.

SUMMARY OF BLOCKCHAIN CONCERNS

Permissioned or Permissionless? Energy cost of permissionless block mining.

If the BlockChain gets really large, costs of downloading a copy.

Cost of verifying that the BlockChain hasn't been tampered with.

National-scale "disruption" scenarios that cause massive rollbacks, chaos.

Accidental loss of some chunk of the chain, making verification impossible.

Smart contracts might be too smart for their own good.

MORE CONCERNS

https://www.joe.ie/news/pics-this-pile-of-cash-worth-22bn-was-found-inside-the-insane-home-of-a-mexican-drug-lord-409313

Today's most enthusiastic Blockchain use cases seem to center on a mix of illegal transactions, money laundering, and a gigantic technology boom but without much of a "market" for the associated products.

The model also depends on some hardness assumptions: finding a nonce, factoring RSA key. Quantum computers could shake up these assumptions.

Unresolved privacy concerns: "everything is on the table."

WALKING THROUGH THE ISSUES

Which issues would arise on a “smart farm”?

Would a BlockChain solve those issues? What new risks would it introduce?

What limits the speed of new technology adoption?

BLOCKCHAIN ON A FARM

Main uses seem to be for audit trails of various kinds:

- Capture data about something we are supposed to trace or record.
- Write it digitally into the ledger, securely. Tamperproof and automatic
- Auditors given access to the record.

But they will want to know:

- Why should I trust this BlockChain record? Is the underlying data valid?

BLOCKCHAIN FOR SENSORS

A smart farm would store BlockChain records that include sensor data.

Very likely the sensors themselves would be securely connected to their host machines, for example using Azure's IoT Hub or the AWS equivalent.

Reminder: With an IoT Hub model, the sensor itself uses security keys, and will only connect and talk to the hub. This enables a *digital twin* concept.

TRUST WITH SENSORS

… so we can assume the connection to the sensor is secure. But how would a digital twin for a farm work?

Azure IoT Hub will only allow authorized sensors to be part of the system, and it patches the software and configuration automatically. Feeds events to the Azure IoT function server, where functions consume them.

So presumably these functions log the event records to the Blockchain.

QUESTIONS THIS MIGHT NOT ANSWER

Is this sensor the correct one for the information the application claims to have captured?

Is the sensor working correctly?

Has the data the sensor generated been modified before it was logged?

CAN WE ANSWER THE QUESTIONS AN AUDITOR MIGHT ASK?

A sensor records shows that cow 2143 was milked on Tuesday at 10am. Later the milk turned out to have a dangerous bacteria in it, like Listeria. It got through and a consumer became quite sick.

Was she properly clean when she was milked? Had she been evaluated for mastitis as required by the health department? Was she periodically checked for overall health? Did she receive any "off records" meds?

CAN WE ANSWER THE QUESTIONS AN AUDITOR MIGHT ASK?

Realistically, an audit using sensor data would provide "evidence" but can't answer these questions. Non-technical people might find this surprising, but in cloud computing we have seen why many either *cannot* be answered, or *we can't have confidence our answers will be correct.*

BlockChain does protect against tampering, and does record what the sensors reported, and when. This is already valuable, as long as we are honest about the capabilities and limitations.

TO HAVE REAL CONFIDENCE…

Azure IoT Hub would need to log **management** events too.

The audit-trail examination tool would need to visualize this information and be able to convince the auditor that yes, this is the proper sensor, it seems to be properly calibrated, the data wasn't tampered with…

The mention of time suggests that we might also need to log events related to the way the system tracks time, or at least have a "story" there.

BLOCKCHAIN-SPECIFIC FORM THIS TAKES?

We noted that early adopters are people with transactions to carry out anonymously, or maybe with money to launder. Strongly motivated mostly because for now, Blockchain feels like a way to evade oversight and taxes.

Business community has many people keen to adopt the next new thing. Startup frenzy and huge fortunes made on ICOs adds fuel to the flames.

But farming is a case for mainstream use and these questions need to be answered. This is why the mainstream technology community is more cautious.

LET'S LOOK AT A BLOCKCHAIN CREATED SPECIFICALLY FOR IOT

Cornell "smart farms" research effort (CIDA) is highly visible.

Led by Susan McCouch, Hakim Weatherspoon, Steve Wolf and Abe Strouck.

One early accomplishment: Vegvisir, a BlockChain specifically for agriculture.

SOME ISSUES THEY THOUGHT ABOUT

A lot of the "events" that matter in an agriculture or farming setting are in remote places, disconnected from the main system.

The BlockChain would probably be used primarily as an audit trace, to track events in the food chain from farm to table.

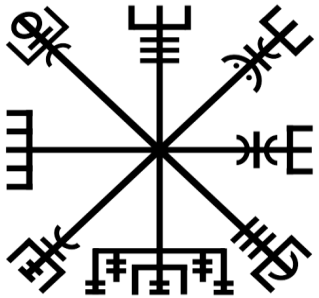
So this raises issues like intermittent connectivity, how we know that the sensor that generated a record is the "correct one" for that role, etc.

CONNECTIVITY: JUST ONE ISSUE OF MANY!

Vegvisir is a research project and a proof of concept, but not deeply integrated with Azure IoT Edge.

Any real product will need more ties to the Azure infrastructure.

But an Azure Blockchain would also benefit: as a part of the official Azure ecosystem, we might gain better answers to some of the trust issues!



VEGVISIR

Slides from Robbert van Renesse

Talk presented at ICDCS 2018

A BLOCKCHAIN FOR THE FOOD SUPPLY CHAIN

Robbert van Renesse

joint work with Hakim Weatherspoon, Danny Adams,
Kolbeinn Karlsson, and Stephen B. Wicker

Initiative for Crypto-Currencies and Contracts (IC3)
Cornell Digital Agriculture Initiative

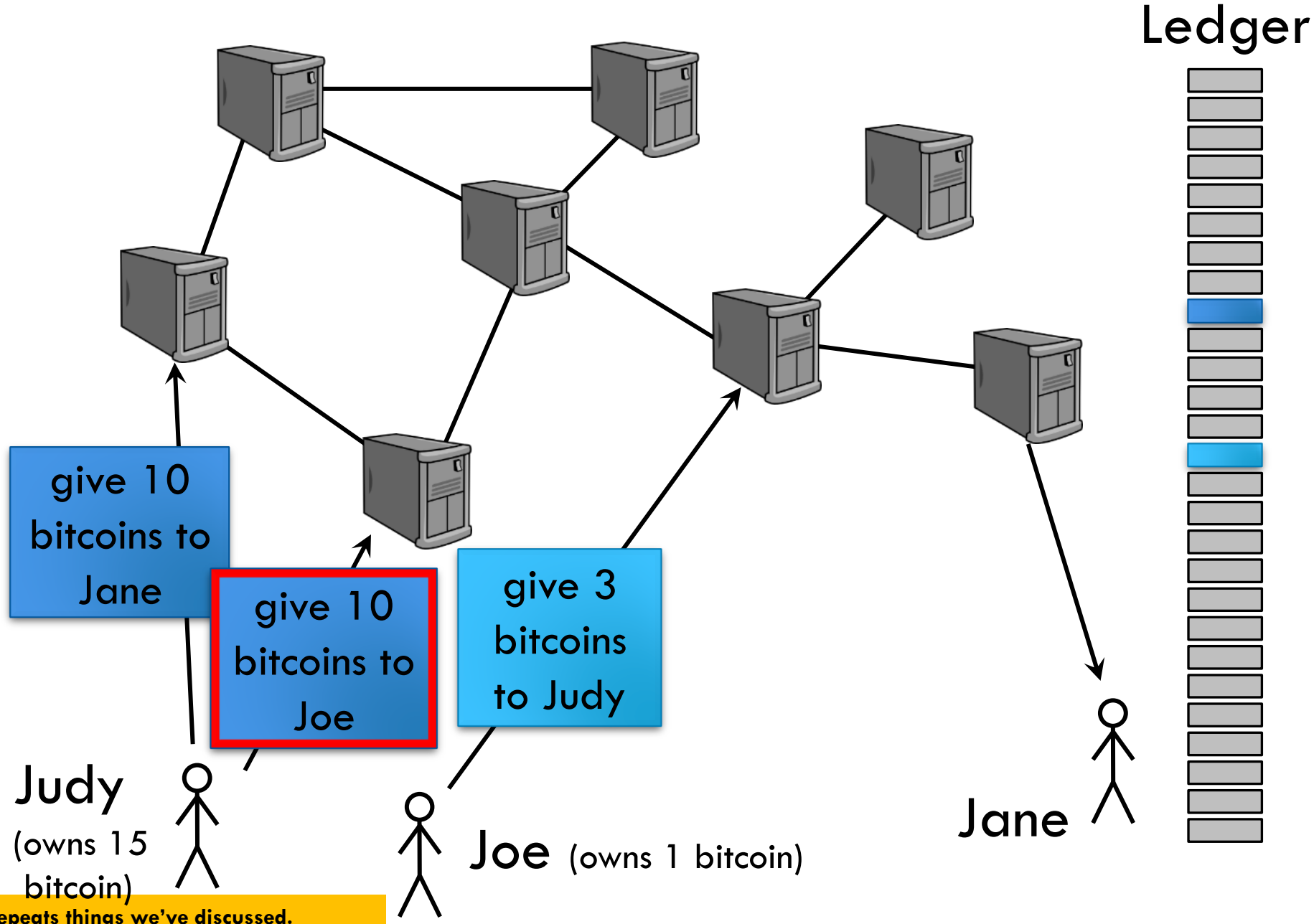
BLOCKCHAIN'S PROMISE

Promises

- Global currency
- Smart contracts
- Notarization
- *Accountability*
- …

BlockChain background: Repeats things we've discussed.

A REPLICATED LEDGER OF TRANSACTIONS



Ittay Eyal, May '16

BlockChain background: Repeats things we've discussed.

SMART CONTRACTS

Smart contracts are executable programs on the BlockChain, take input from the BlockChain, and produce output on the BlockChain

Main use: *automated escrow*, where disbursement depends on agreed upon conditions

Caution: Smart Contracts have been found to be prone to (very expensive) bugs

BlockChain background: Repeats things we've discussed.

POTENTIAL USE CASES

Killer app: cryptocurrencies

Other potential uses:

- Reduce opaqueness of supply chains
  - One "trustless" place for all transactions along the way
  - Improvements over paper-based systems and many disjoint databases
- Eliminate middlemen
  - Why does farmer make so little and consumer pay so much?
- Reduce fraud
  - India, Russia, Sweden, Georgia… are building blockchain-based land registries to fight "land fraud" and simplify international property transactions

BlockChain background: Repeats things we've discussed.

FOR THE FOOD SUPPLY CHAIN?

Supply chain management

- Walmart is building one for the food supply chain
  - Food safety: fast identification of tainted foods
  - Consumers are demanding more information about the products they buy (organic, fair trade, …)
- Simplify international transactions

Help farmers

- Want to know what happens to their products for fair pricing
- What products should they be producing?

Reduce food scandals

- illegal production, misrepresentation, loss and waste, …

BlockChain background: Repeats things we've discussed.

INDUSTRIAL UPTAKE?

ripe.io:

- A company that is building a "blockchain of food" with IoT interfaces

Walmart:

- partnered with IBM and Tsinghua to identify sources of contaminated products and speed up recall

But today's blockchain technology may not be appropriate for all use cases

- too dependent on availability of plentiful power, networking, and storage

DESIRED BLOCKCHAIN PROPERTIES

Performance:
- High Throughput, Low Latency
- Energy-Efficient

Security:
- Always available for reading (verifying) and appending
- Fair
- Tamperproof (Integrity)
- Possibly confidentiality as well

No Single Administrative Domain
- *no need to trust a single provider*

Open membership (or not)

OPEN MEMBERSHIP IS HARD

Traditional secure logs are based on voting

Members vote on which transactions to add to the log and in what order

Problem: "Sybil" or impersonation attacks

- a participant may try to vote multiple times
- with closed membership, cryptographic signatures can identify the source of a vote
- with open membership, anybody can create identities and that way vote many times

PERMISSIONLESS VS PERMISSIONED BLOCKCHAINS

	Permissionless	Permissioned
Approach	Competitive	Cooperative
Basic technique	Proof-of-Resource	Voting
Membership	Open	Closed
Energy-efficiency	Often terrible	Excellent
Transaction rate	At best hundreds / sec	Many thousands per second
Transaction latency	As high as many minutes	Less than a second

BITCOIN BLOCKCHAIN

Permissionless, open membership

Proof-of-Work

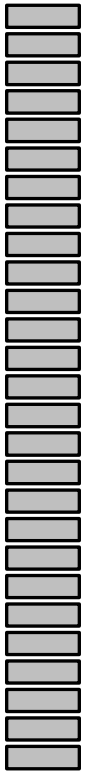
There are thousands of Bitcoin miners

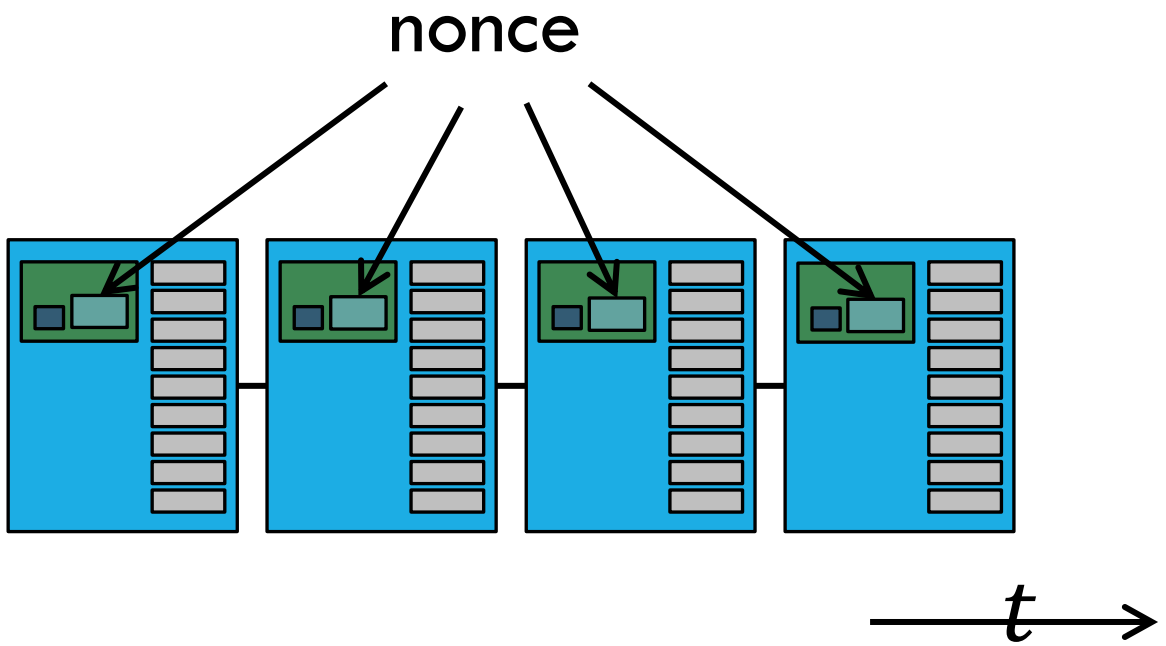
- they use ASIC hardware to compute SHA256 hashes
- use about more energy than the country of Denmark

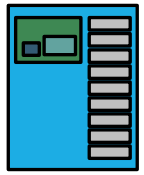
Overall rate is a few transactions per second

THE BLOCKCHAIN

Ledger

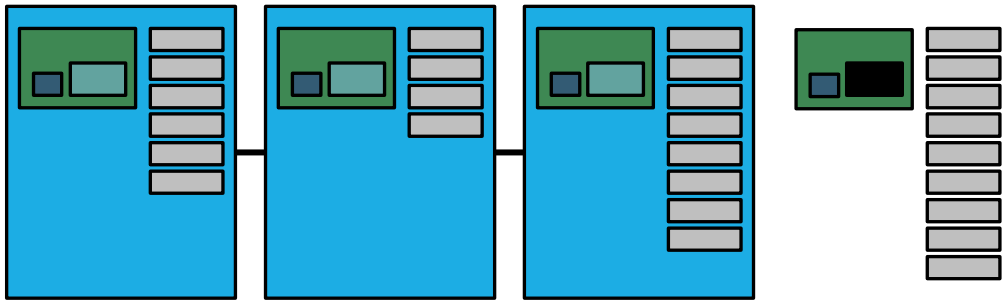




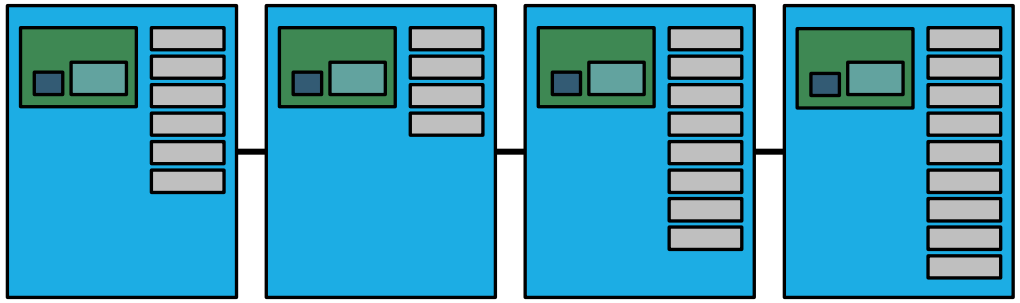
HASH() < ***target***

"cryptopuzzle"

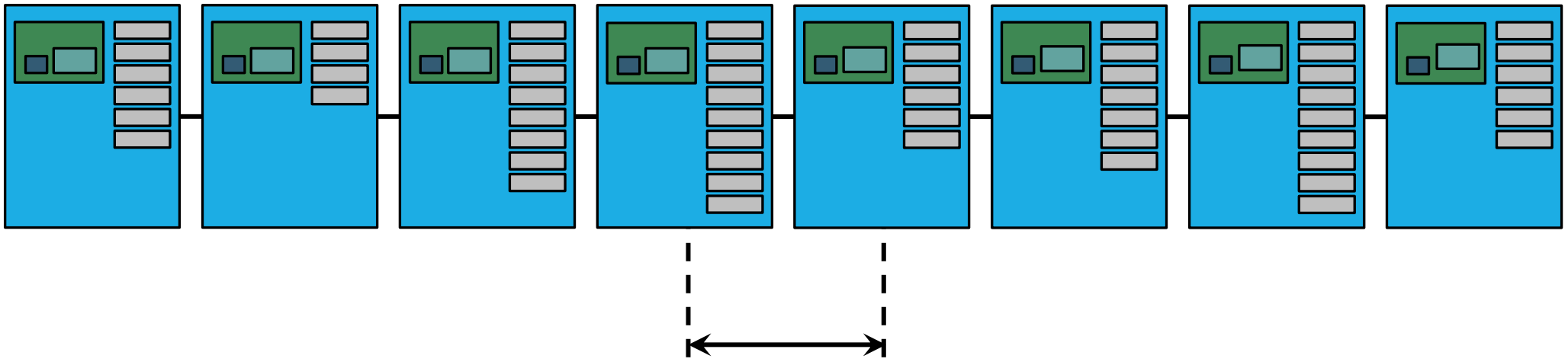
THE BLOCKCHAIN



THE BLOCKCHAIN



THE BLOCKCHAIN



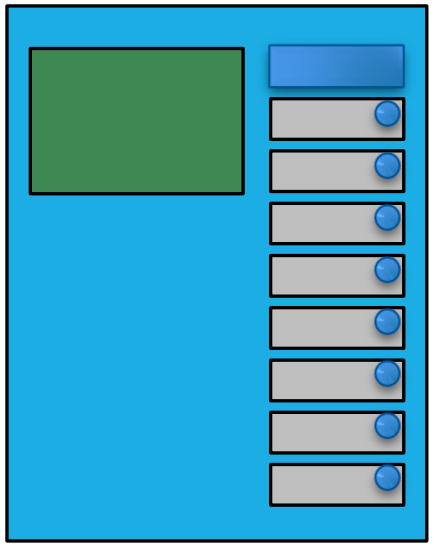
Exponentially distributed rate of new blocks, with constant mean interval

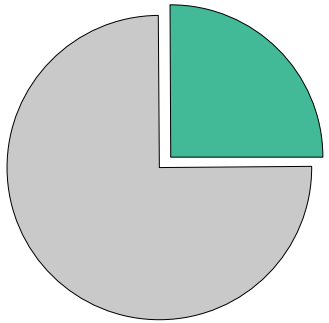
target automatically adjusted every 2016 blocks so that mean interval is **10 minutes**

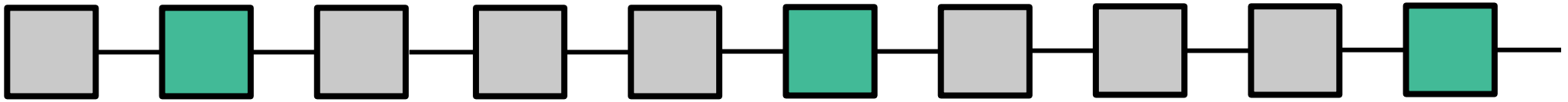
INCENTIVES FOR MINING

Prize:

- **"Minting"**
- **Transaction Fees**

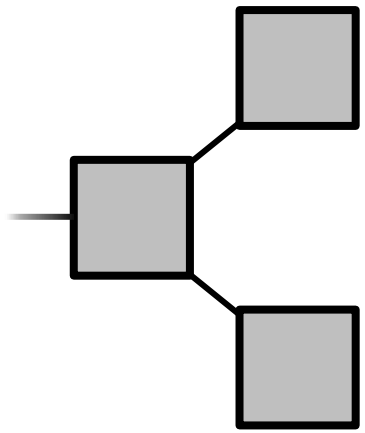






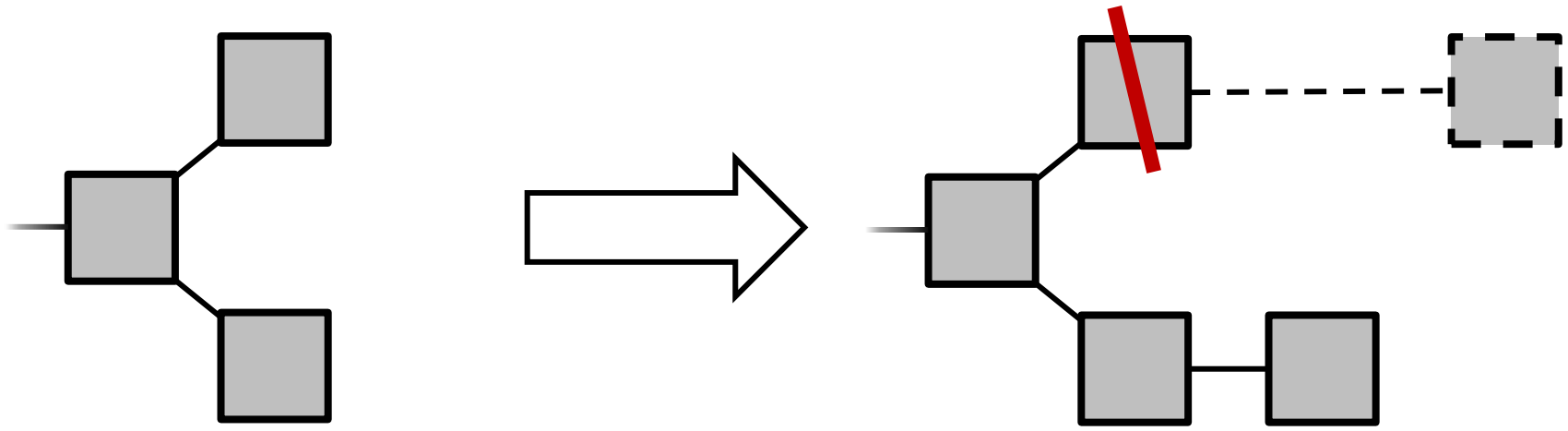
***Wins** proportional to computation power*

FORKS



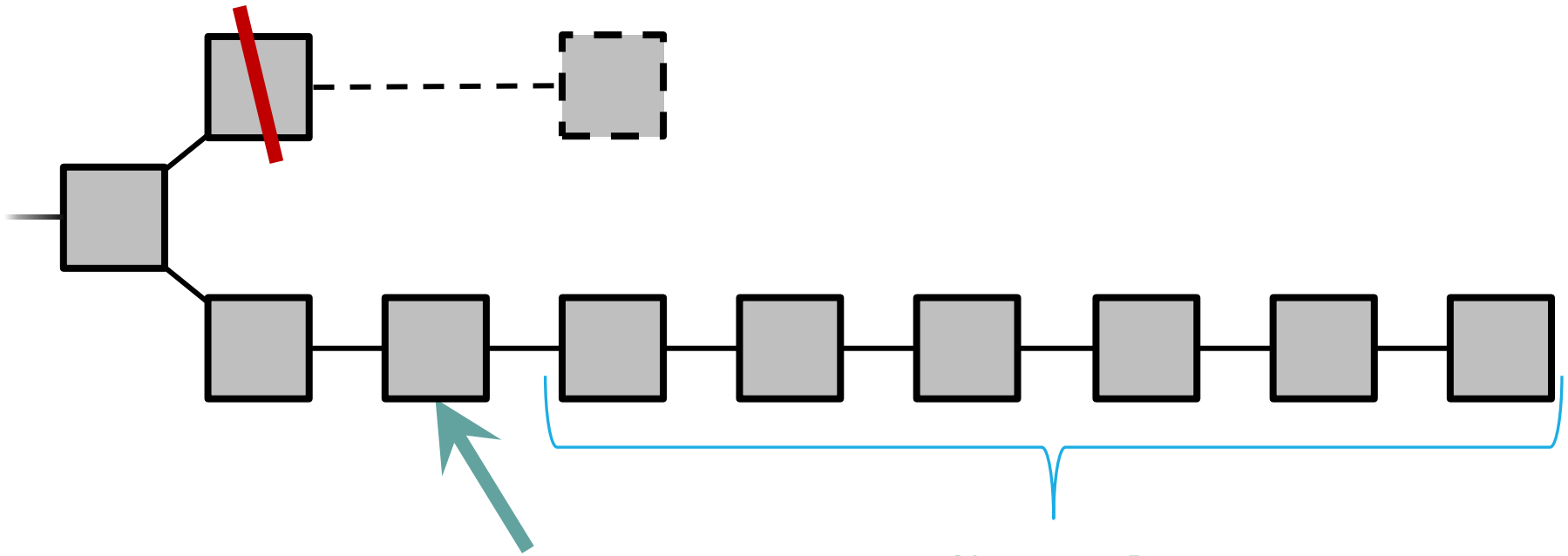
Two blocks “mined” at approximately the same time by two different miners

FORK RESOLUTION



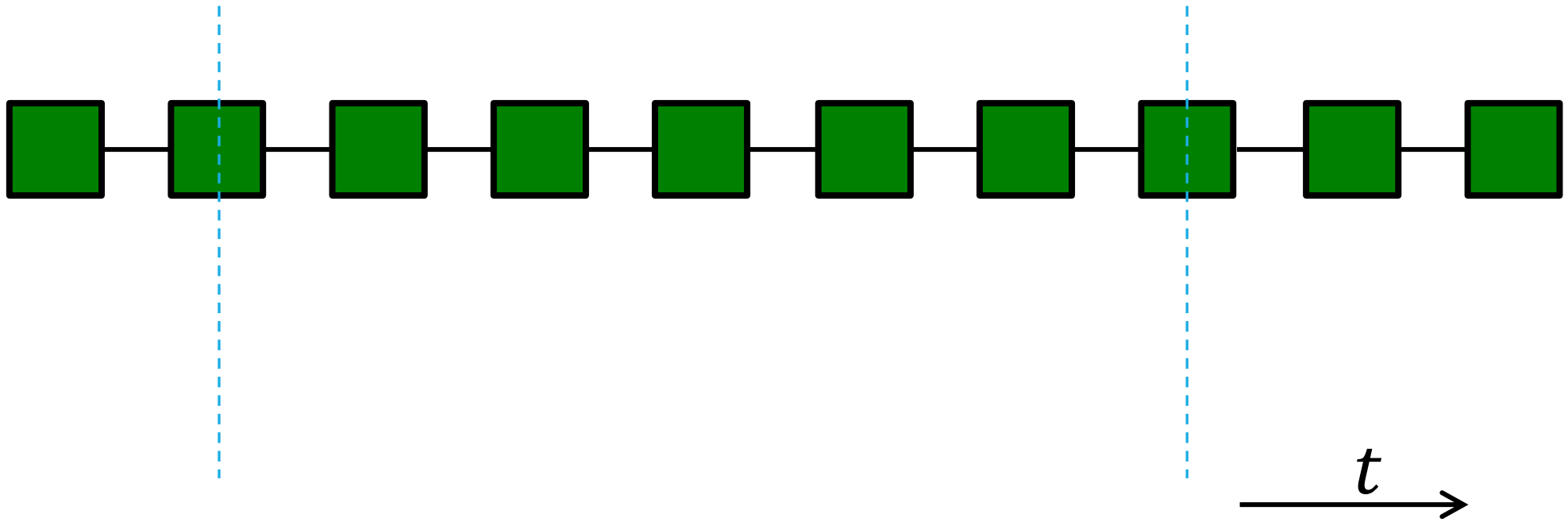
- **Longest** chain wins
- Transactions on short chain are reverted

FORK RESOLUTION

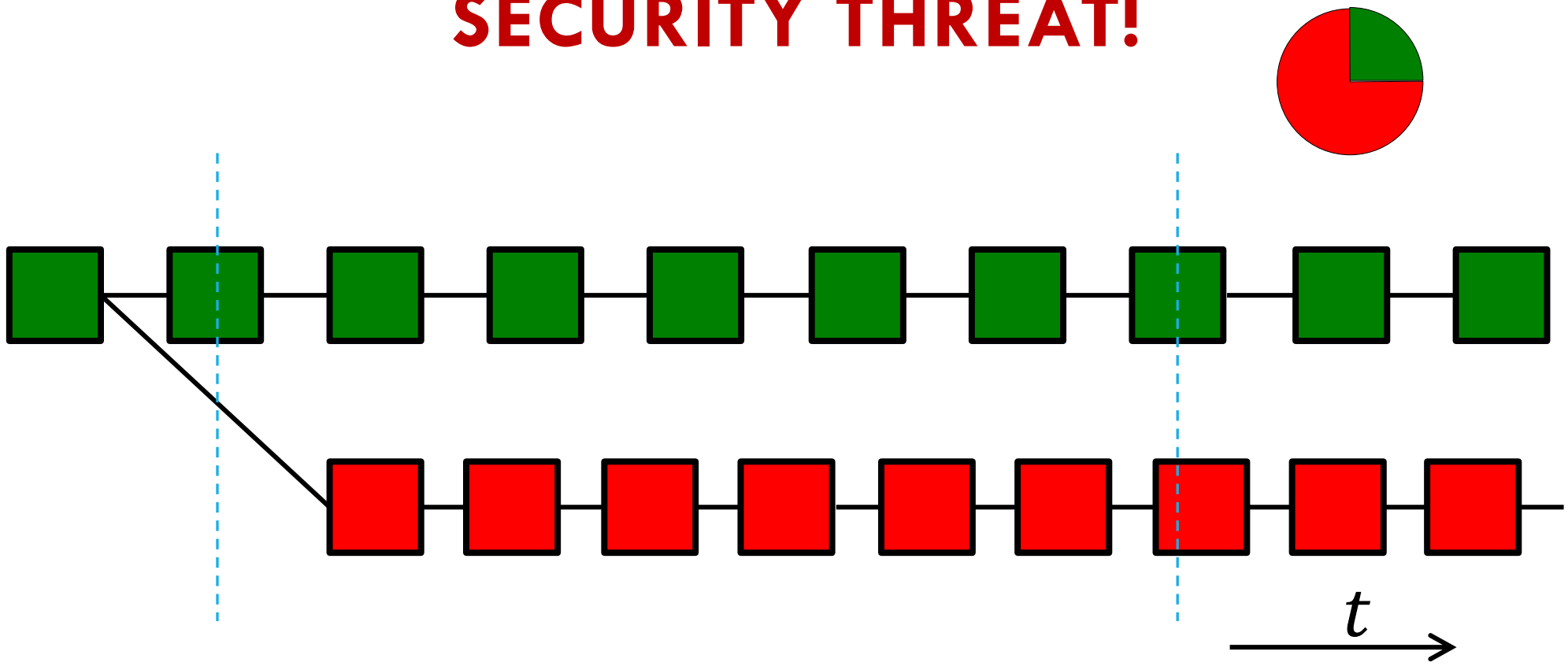


A transaction is **confirmed** when
it is **buried** "deep enough"
(typically 6 blocks – i.e., one hour)

SECURITY THREAT!

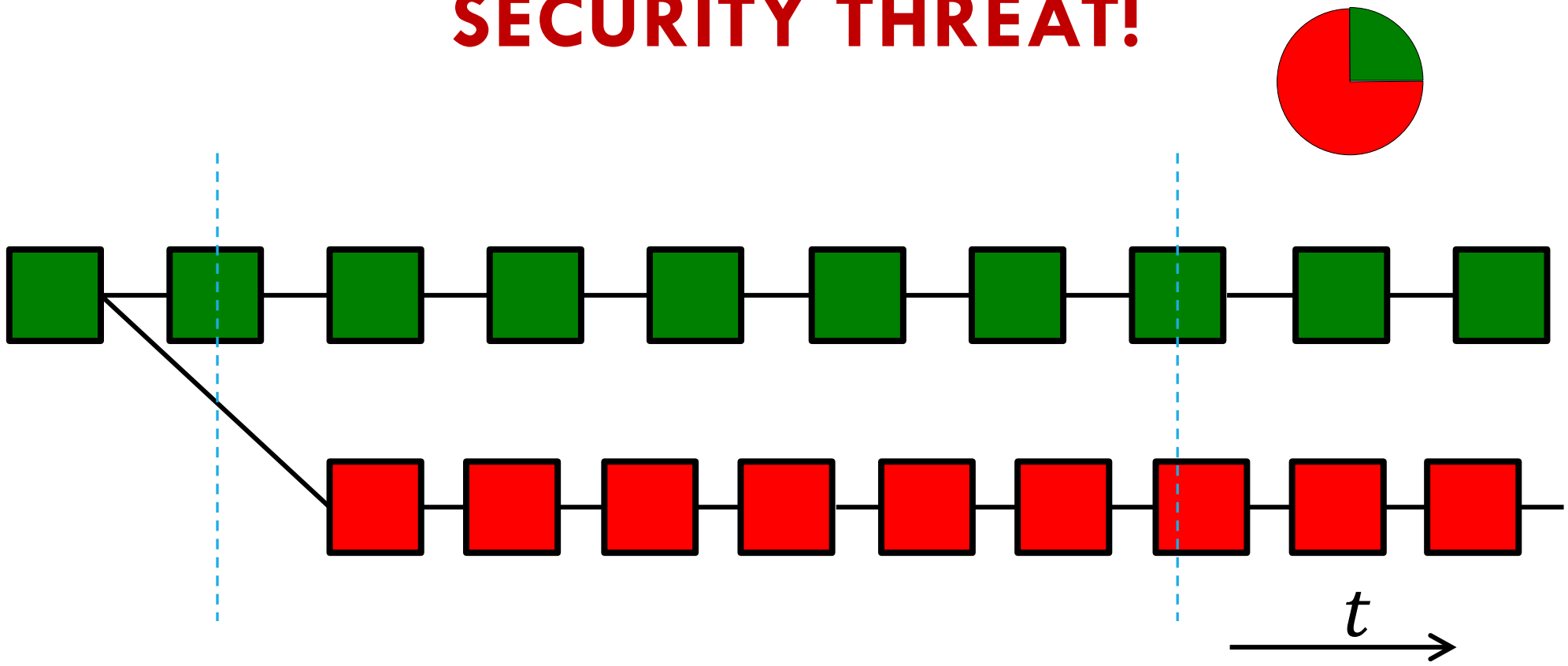


SECURITY THREAT!



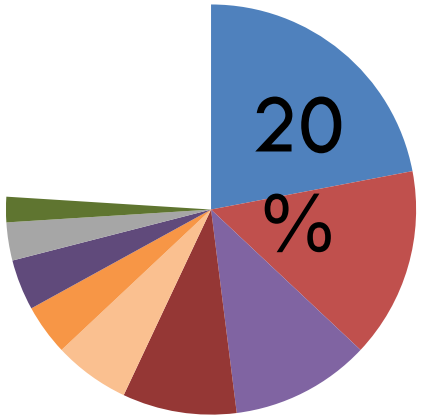
Threat: attacker outruns good miners

SECURITY THREAT!



Threat: attacker outruns good miners

➔ **Security Assumption:** *good miners own >.5 of the total compute power*



[blockchain.info, April 2015]

PERMISSIONLESS BLOCKCHAINS

Open membership, but inefficient

Vulnerable to 50% attacks

Examples include Bitcoin, Ethereum, IOTA

PERMISSIONED BLOCKCHAINS

Performance:

- High Throughput, Low Latency
- Energy-efficient

Security:

- No forks!

Closed membership

Examples include Ripple, Hyperledger

BLOCKCHAIN FOR THE FARM?

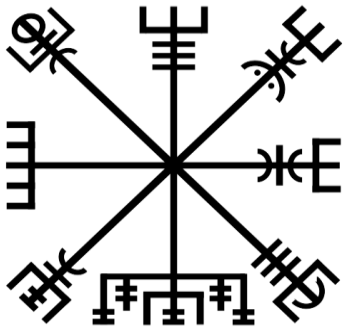
Blockchains require strong network connectivity and lots of storage

Permissionless blockchain are power-hungry

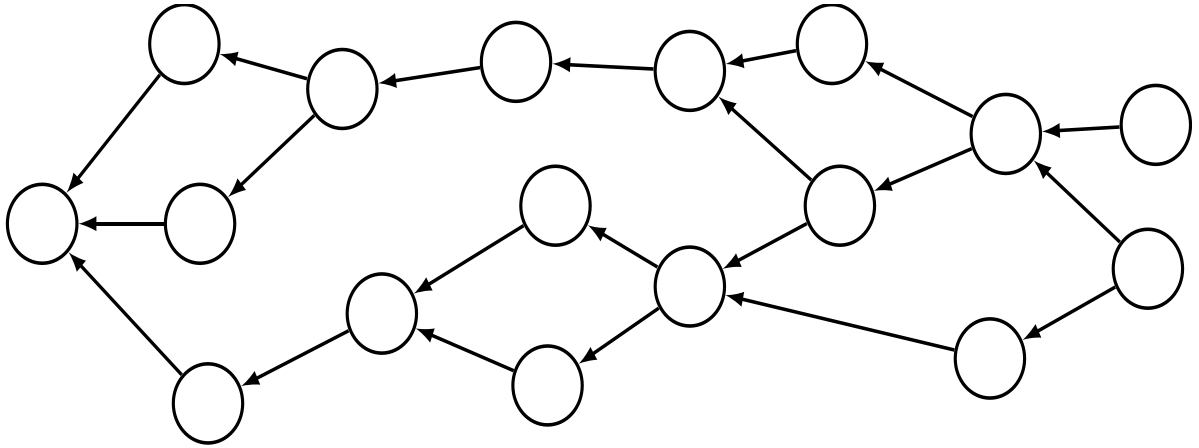
Sensors have limited resources

- Sensors for growing conditions, storage conditions, shipping conditions, …

*Blockchain for a farm will generate records in a decentralized way, and hence it *must* work in a network-partitioned or -challenged environment*



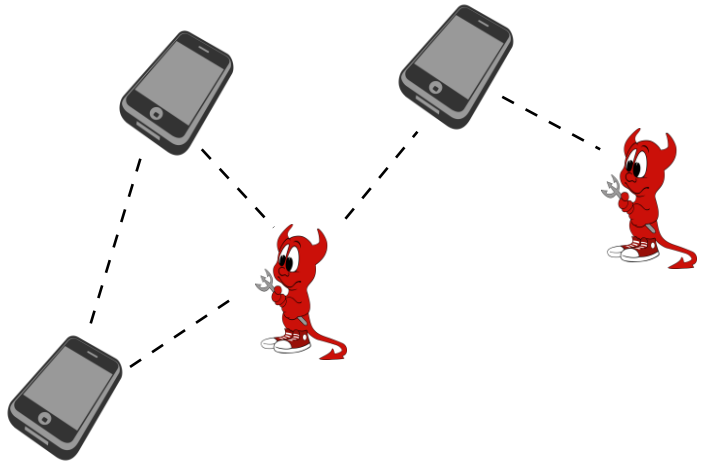
Vegvisir: tolerate branches

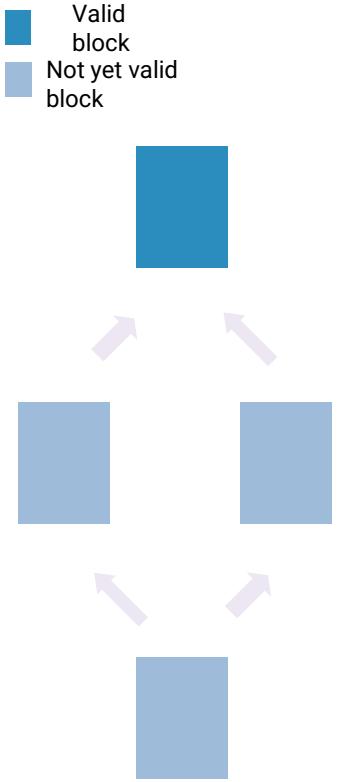


- The key innovation is to allow branching as a feature.
- Leads to DAG structure instead of linear blockchain
- Still maintains full causal history of events (respect's Lamport's $\rightarrow$)

Proof-of-Witness to persist blocks securely

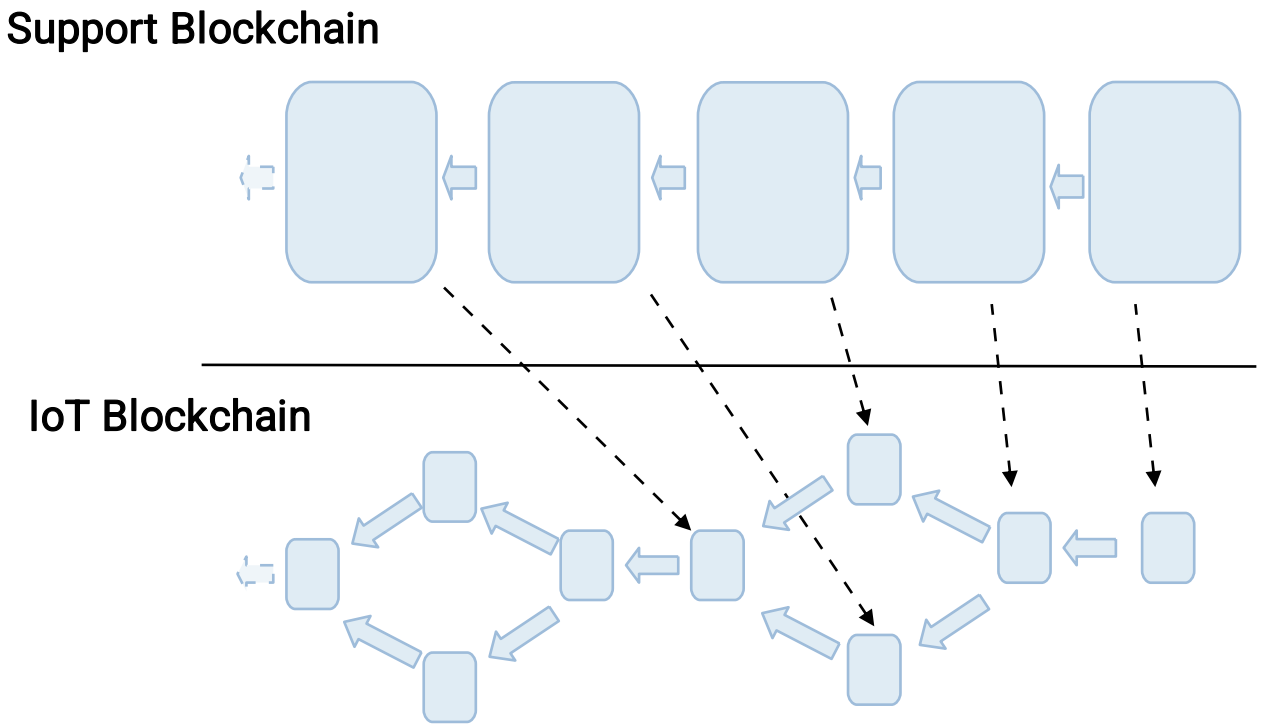
No more than *k* malicious nodes in any neighborhood





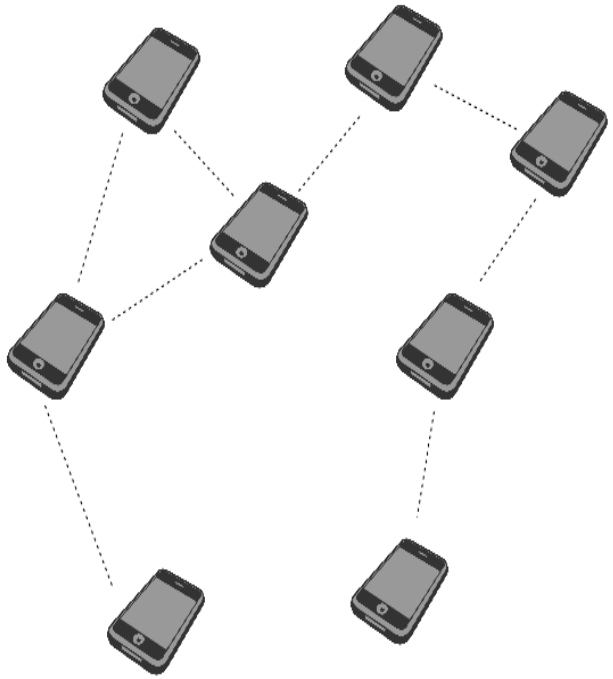
At least one copy of a valid block will survive if < *k* malicious peers

The Support Blockchain reduces sensor storage needs



Allows regular peers to discard old blocks when storage space is low

Blocks are *gossiped* over ad hoc network

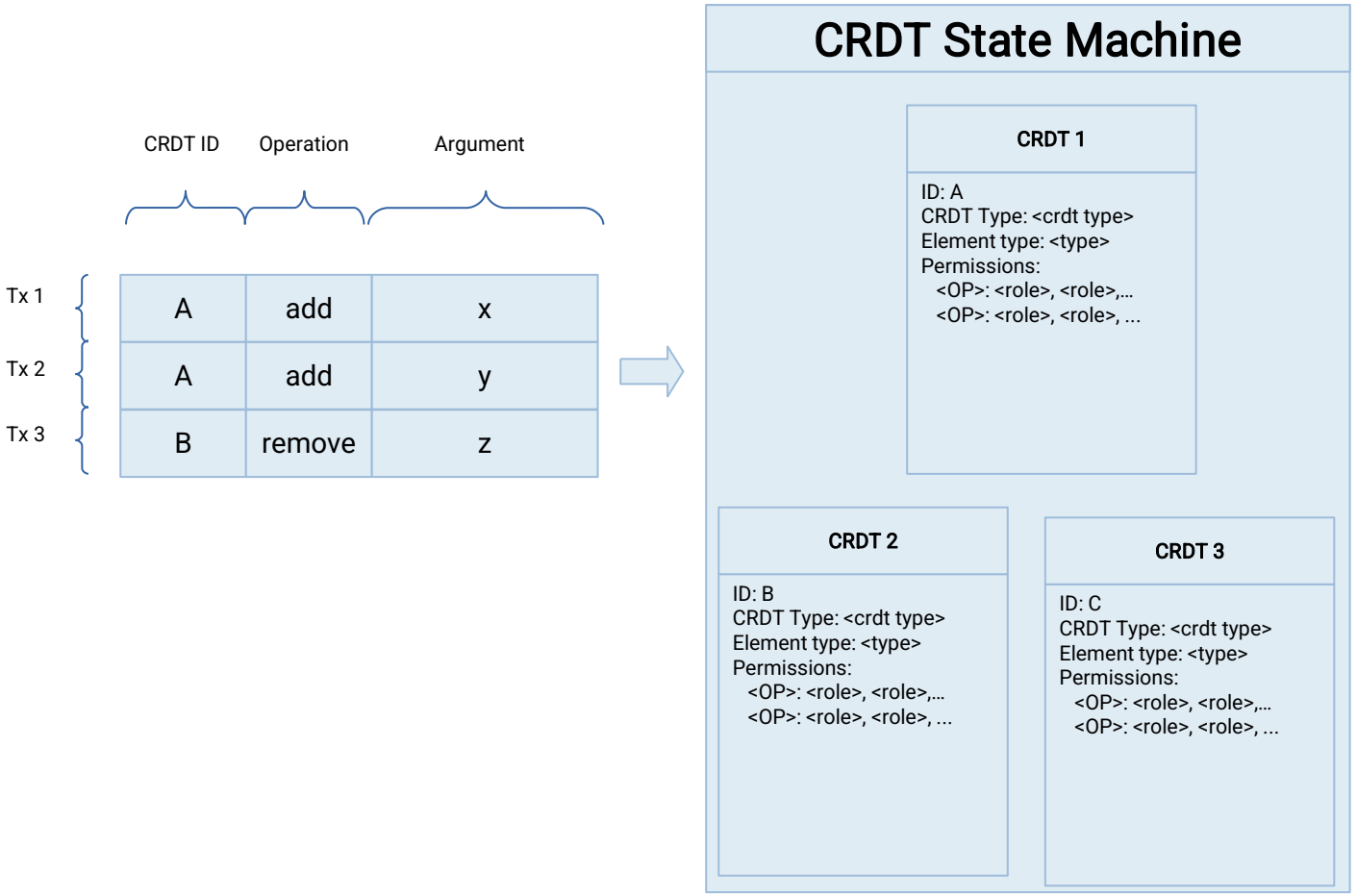


Heterogeneous, opportunistic networking

CRDTs for strong semantics in partitioned world

- *Conflict-Free Replicated Datatype*
- Updates must be associative, commutative, idempotent
- Replicas can be updated independently and concurrently
- Basic CRDTs form registers, counters, sets

Transactions manipulate CRDTs



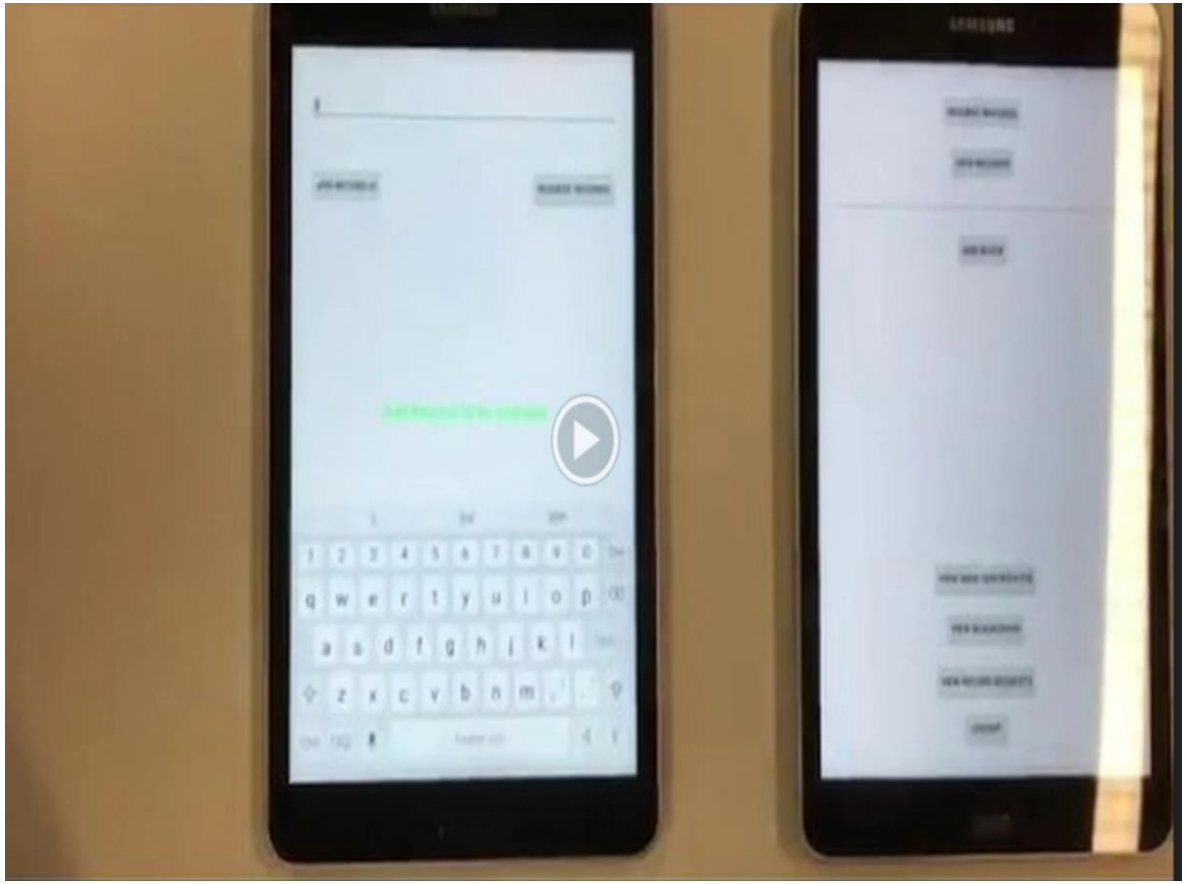
BUT SOME QUESTIONS REMAIN OPEN

How would Vegsivir handle "double spending" (same coin spent in both branches), or other kinds of semantic conflicts that might not involve coins?

- The actual meaning of the operation changes, or it becomes invalid.
- This could cascade to impact subsequent operations, too.
- We can't simply merge the chains and walk away…

Also, although smart farms have many sensors, Vegvisir lacks an answer to the issue of trust: we need the IoT hub to log enough information to know why we should trust a sensor, but this topic is out of scope for the paper.

DEMONSTRATION VIDEO



Proof of concept system

CONCLUSION

Exciting possibilities for blockchains in the food supply chain

But current blockchain designs may not be compatible with some deployment scenarios in the food supply chain

Vegvisir supports partitioned operation and has low power/networking/storage requirements