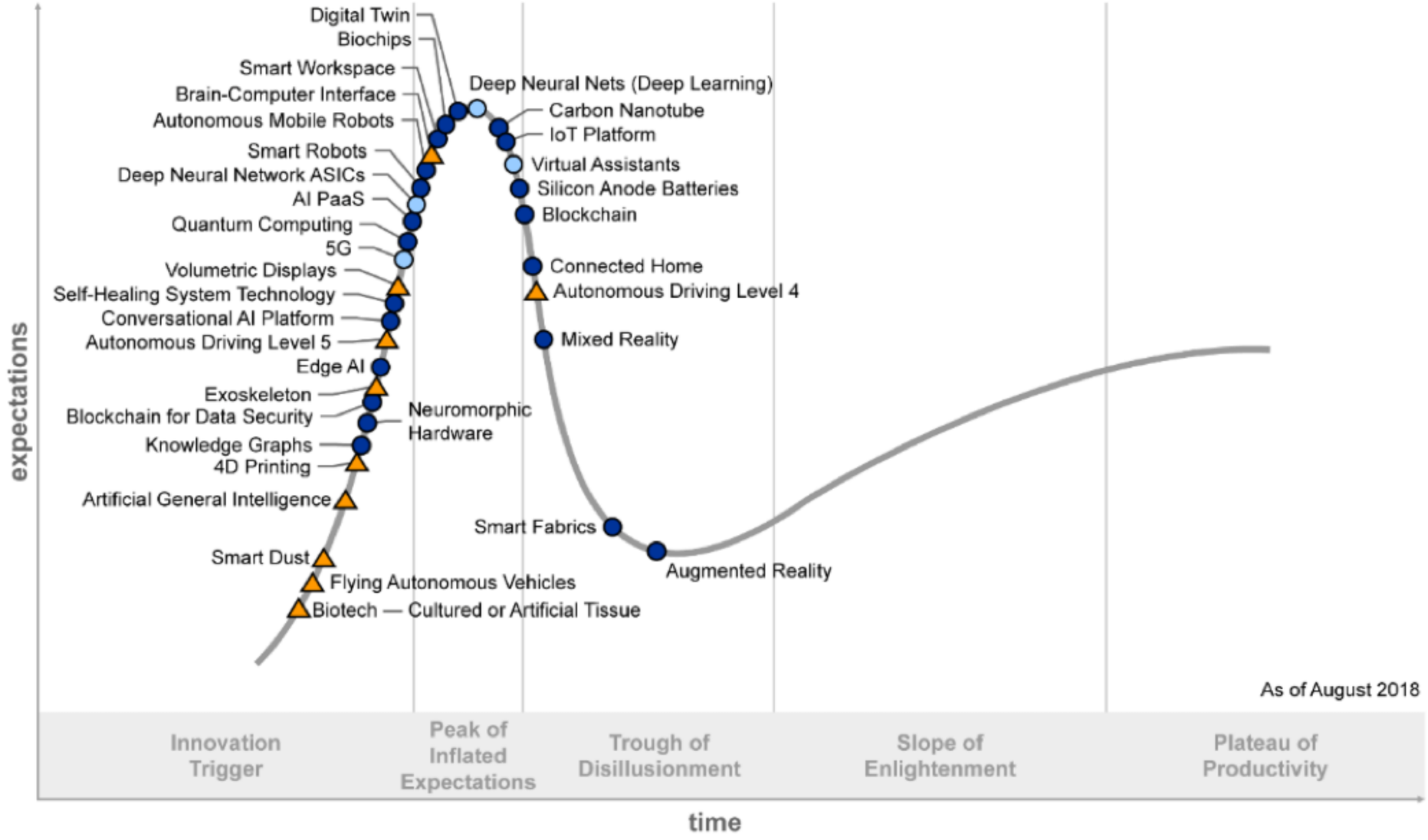
Blockchain Technologies: What is it, and what can it be used for?

Hein Meling

University of Stavanger

Gartner Hype Cycle Emerging Technologies



> *“Today, blockchain research is one of the more vibrant areas of computer science, with the potential of revolutionizing how our society deals with **trust**.*
>
> — Maurice Herlihy

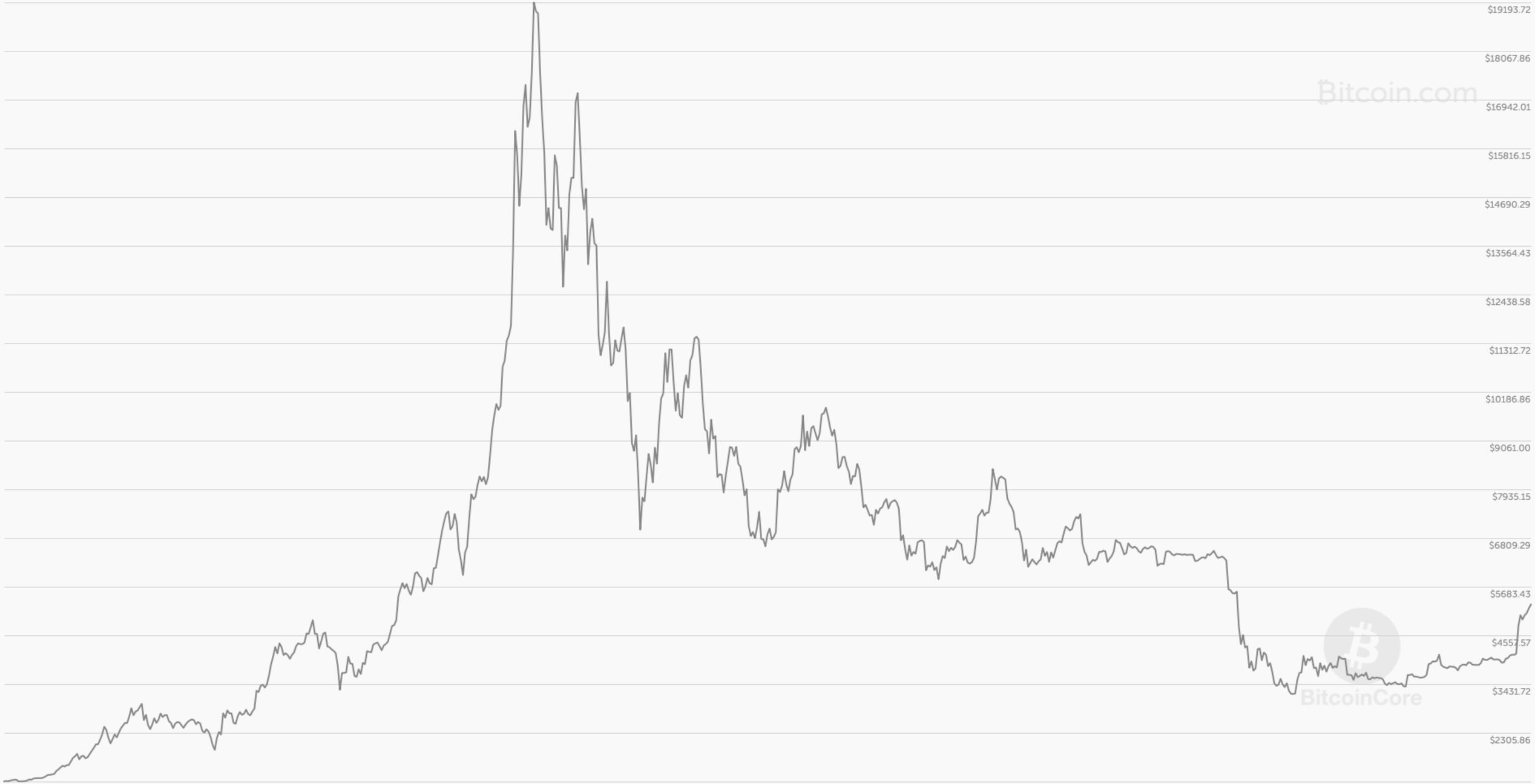
Bitcoin Preview

- Peer-to-peer network of miners
- Completely decentralized
- Transactions are recorded in a public distributed ledger called a *blockchain*
- Bitcoin was the first use of a blockchain to support consensus
- Nodes in the network *verify transactions* for inclusion in the blockchain

Bitcoin History

- White paper: Bitcoin: A Peer-to-Peer Electronic Cash System, October 2008
- Invented by Satoshi Nakamoto
- Open source implementation released January 2009
- Nakamoto active until mid-2010
- Nakamoto's “account”: roughly one million bitcoins worth more than 6 billion dollars (9/2018) or 19 billion (12/2017) at its peak

Bitcoin Core (BTC) Price



2017-04-07

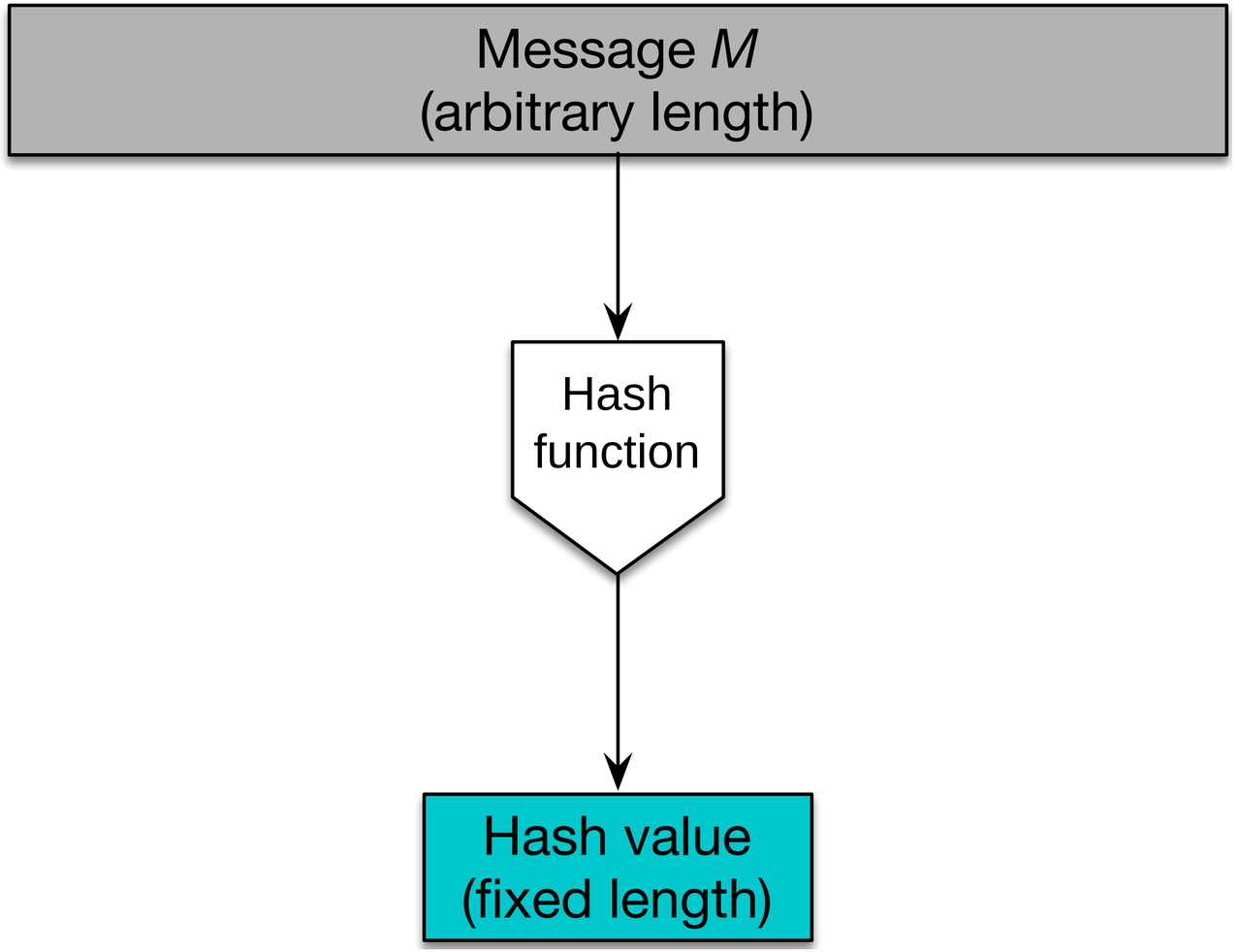
charts.bitcoin.com

2019-04-09

Some Technical Preliminaries

Hash Function

- Takes any string as input
- Fixed-size output
  (typically 256 bits)
- Efficiently computable



Properties of Hash Functions

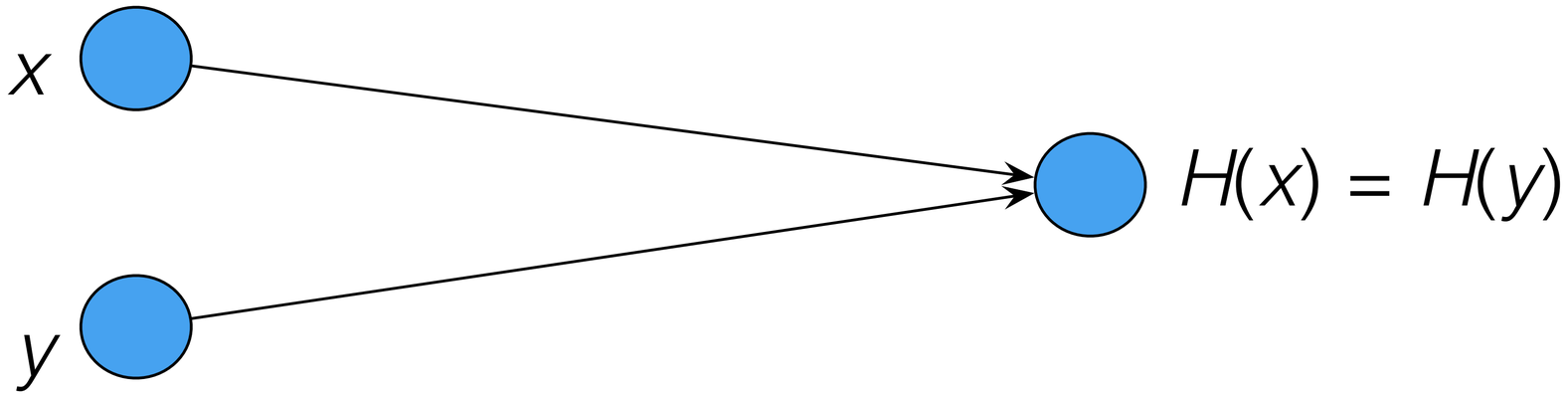
- Collision-free
- Hiding
- (Puzzle-friendly)

Easy

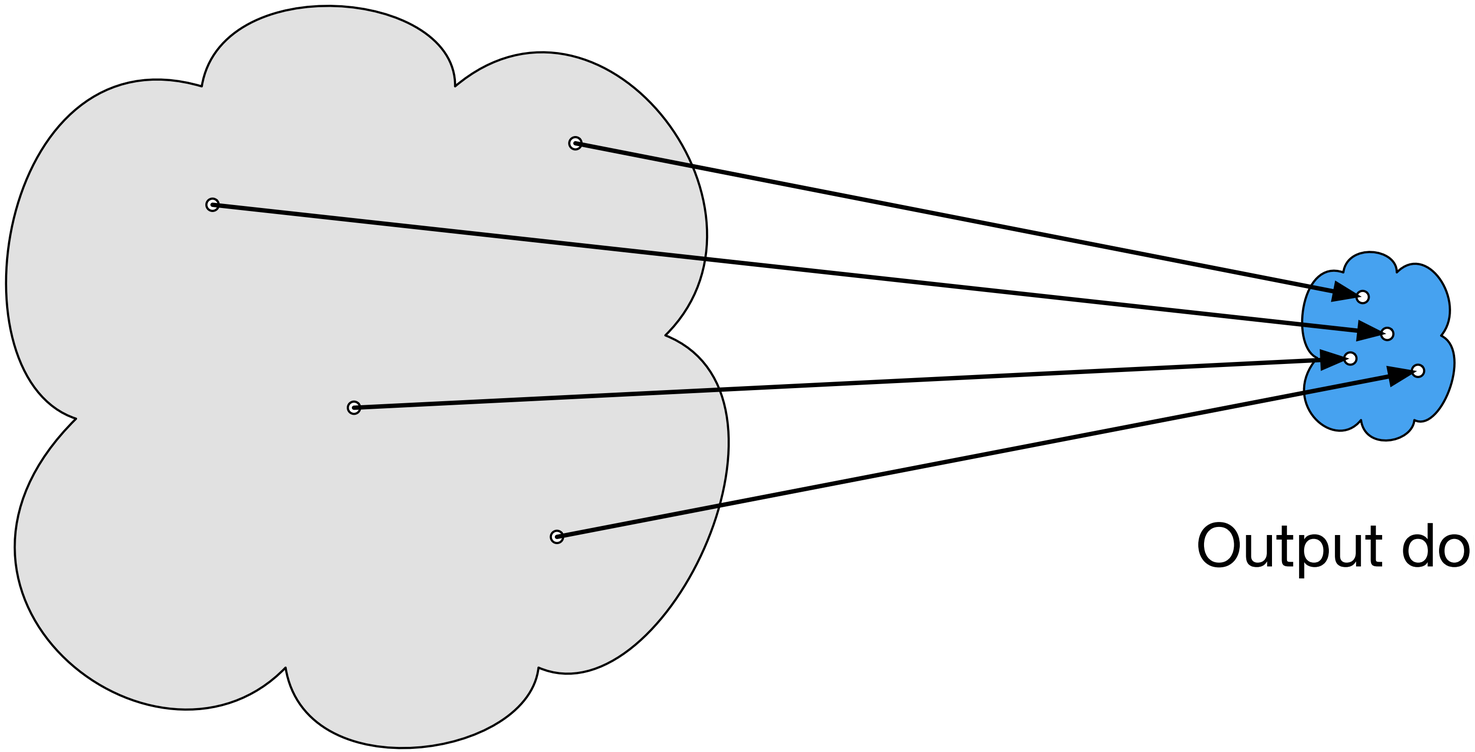
Hard

Property 1: Collision-free

- Difficult to find x and y such that:
  - $x \neq y$ and $H(x) = H(y)$



Collisions must exist…



Output domain

Input domain

… but can anyone find them?

Property 2: Hiding

- We want the following property:
  - Given $H(x)$, it is *infeasible* to find x
- Problem: if the input is deterministic, e.g.: H(“heads”) or H(“tail”)
  - Then it is easy to find x!

Add Non-determinism

- Pick some random fixed-length (256-bit) string *r*
- Then given $H(r \mid x)$, it is infeasible to find *x*
  - *r* is called a *nonce* (number used only once)

```
% echo "Hello my name is Hein Meling. My primary research interests are
fault tolerant and secure distributed computing and in making systems
autonomic. I'm interested in building fault tolerant systems that are
robust against a wide range of failures and attacks. In particular, I'm
interested in developing techniques and methods to test and evaluate the
robustness of systems. Together with my students, I'm developing
protocols, high-level programming abstractions, and middleware
environments that help alleviate development effort for large-scale
distributed systems that aim to tolerate failures and dynamism.
Currently, I'm focusing on technology for blockchains and Byzantine
fault tolerance." | sha256

4fce622354de5fdaa0ac16ae6d527639b171cc2753bcbb94913e101e14202fb1
```

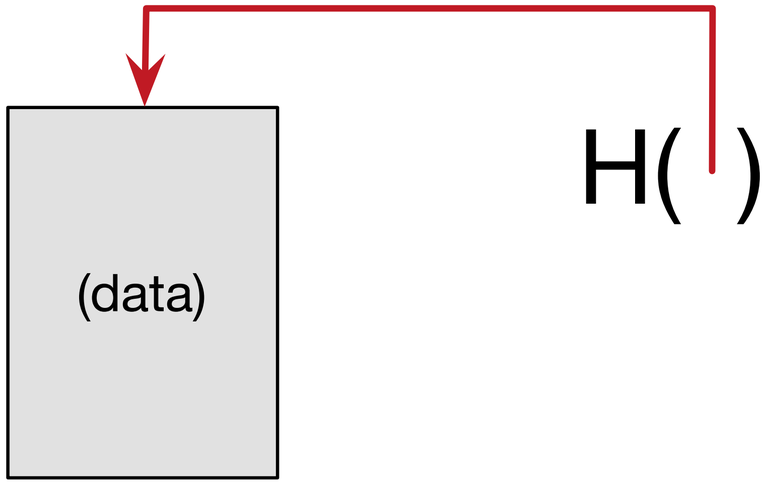
```
% echo "Hello my name is Hein Meling. My primary research interests are
fault tolerant and secure distributed computing and in making systems
autonomic. I'm interested in building fault tolerant systems that are
robust against a wide range of failures and attacks. In particular, I'm
interested in developing techniques and methods to test and evaluate the
robustness of systems. Together with my students, I'm developing
protocols, high-level programming abstractions, and middleware
environments that help alleviate development effort for large-scale
distributed systems that aim to tolerate failures and dynamism.
Currently, I'm focusing on technology for blockchains and Byzantine
fault tolerance*" | sha256

577f3142e42538b20c8bead720162acc1ab1b2065d30f648fc458a0b466898b7
```

Hash Pointers and Data Structures

Definition: Hash Pointer

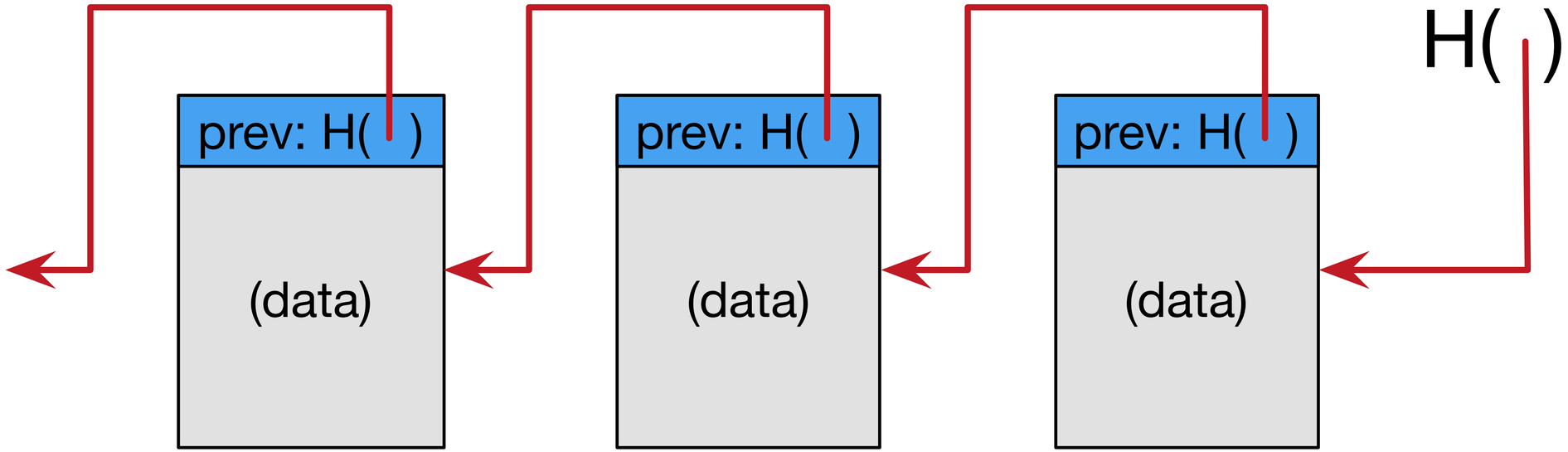
- A pointer to where some info is stored, and
- a cryptographic hash of the info



- If we have a hash pointer, we can
  - ask to get the info back, and
  - verify that it hasn’t changed

Blockchain

A Blockchain is a linked list with hash pointers

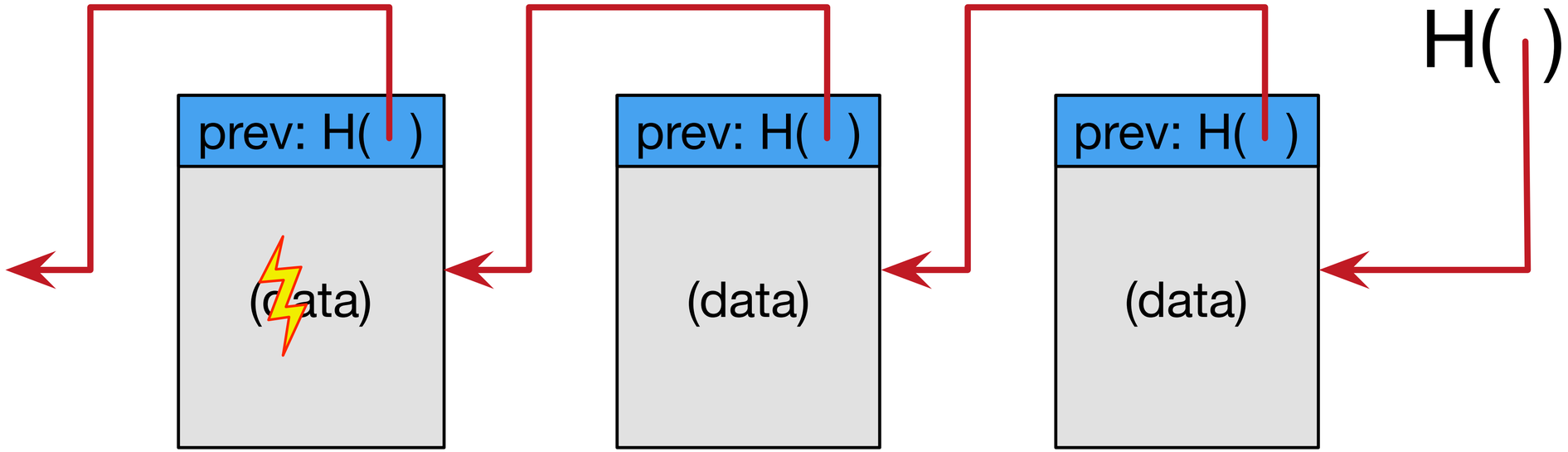


Tamper-evident Log

- Log data structure that appends data to the end of the log
- If somebody alters data appearing earlier in the log
  - We will detect the tampering!
  - Need to store H( ) safely!

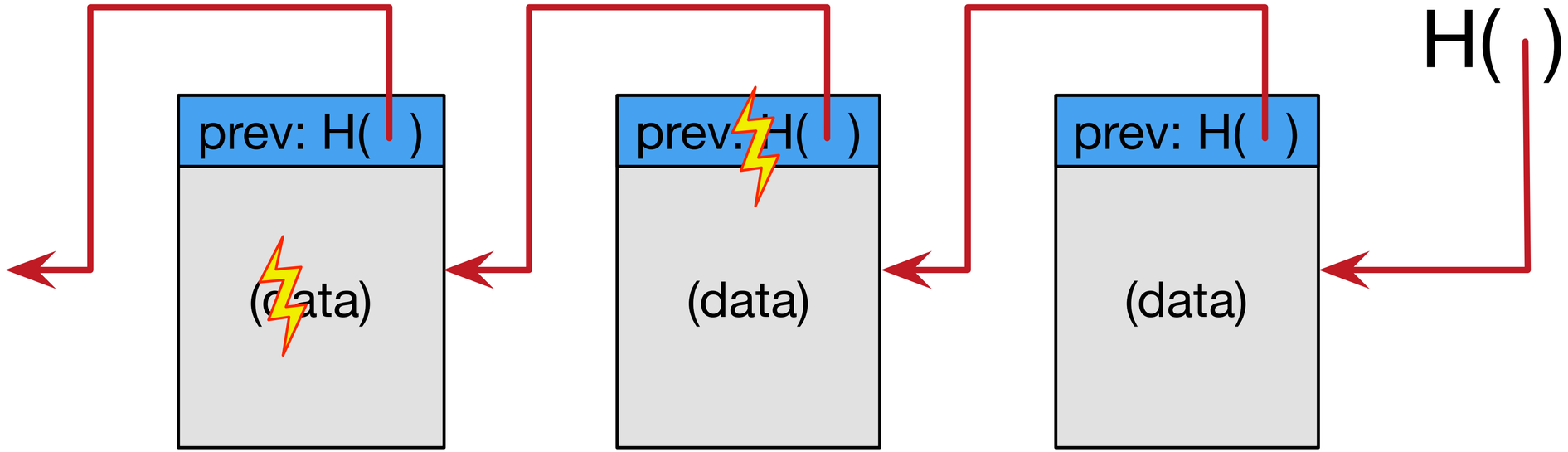
Blockchain

A Blockchain is a linked list with hash pointers



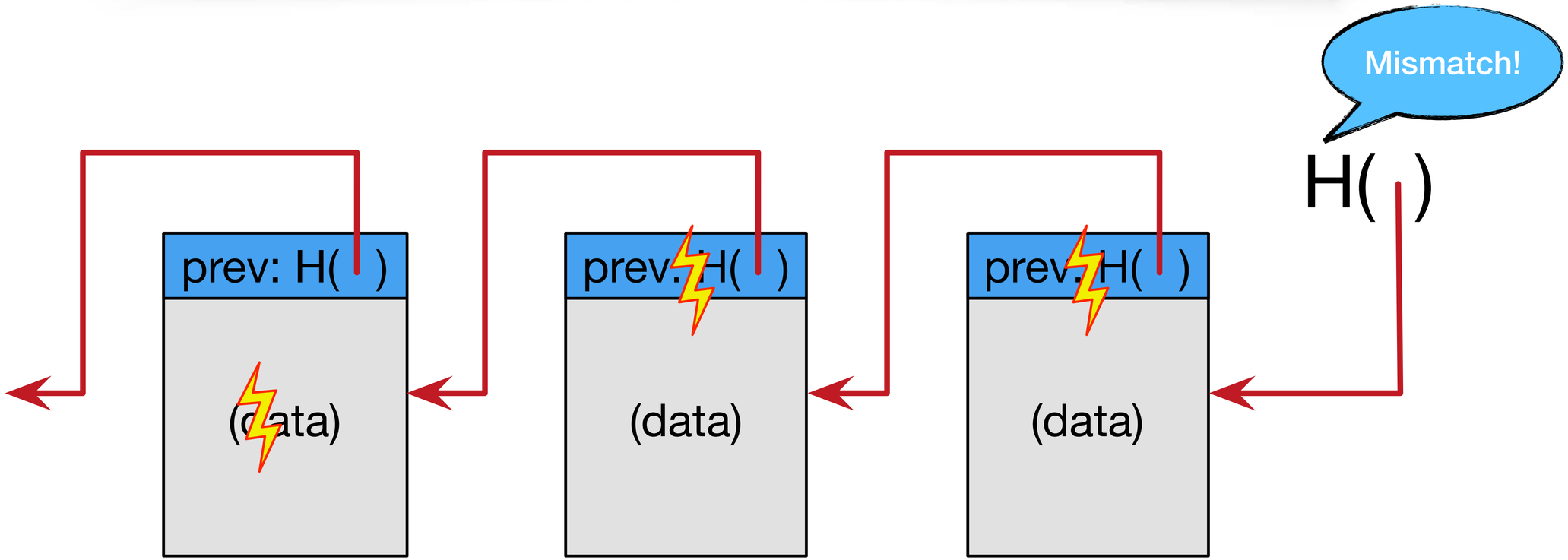
Blockchain

A Blockchain is a linked list with hash pointers



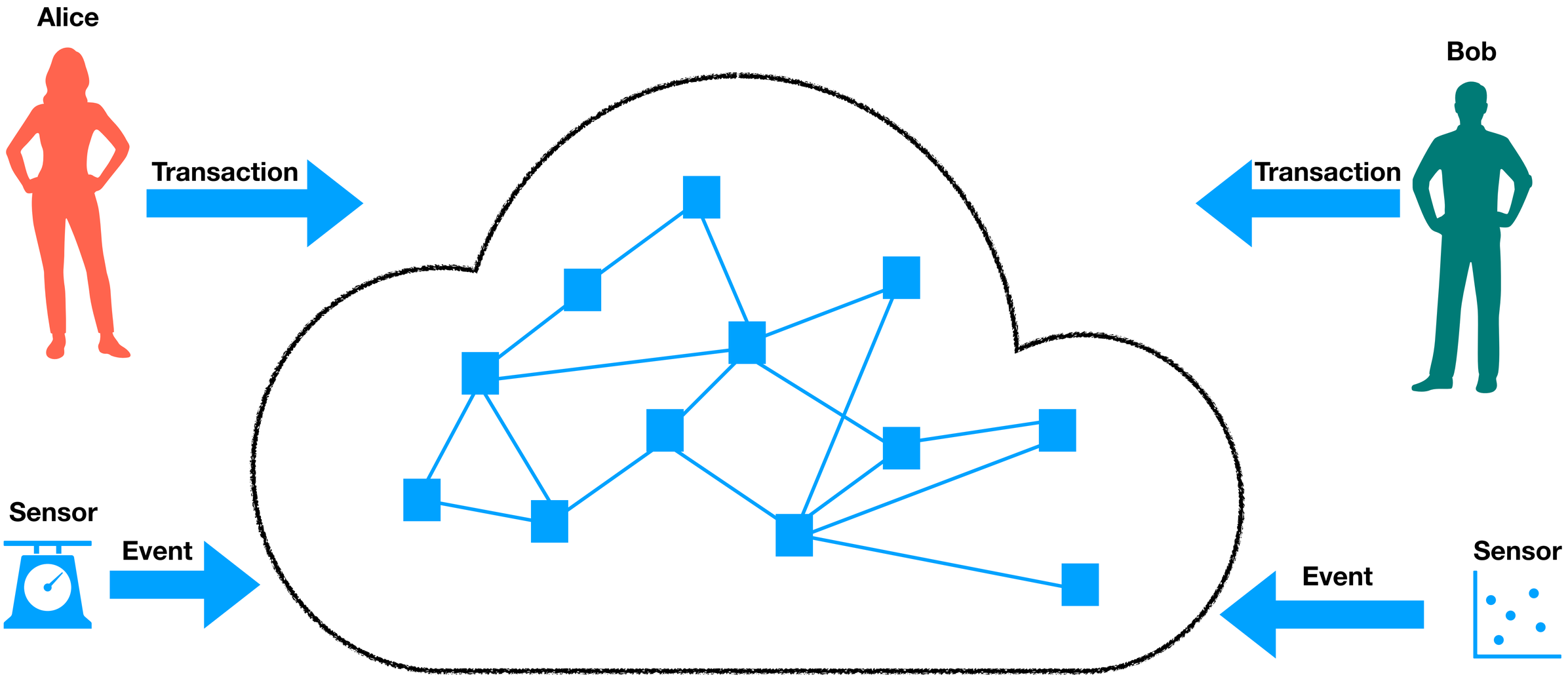
Blockchain

A Blockchain is a linked list with hash pointers

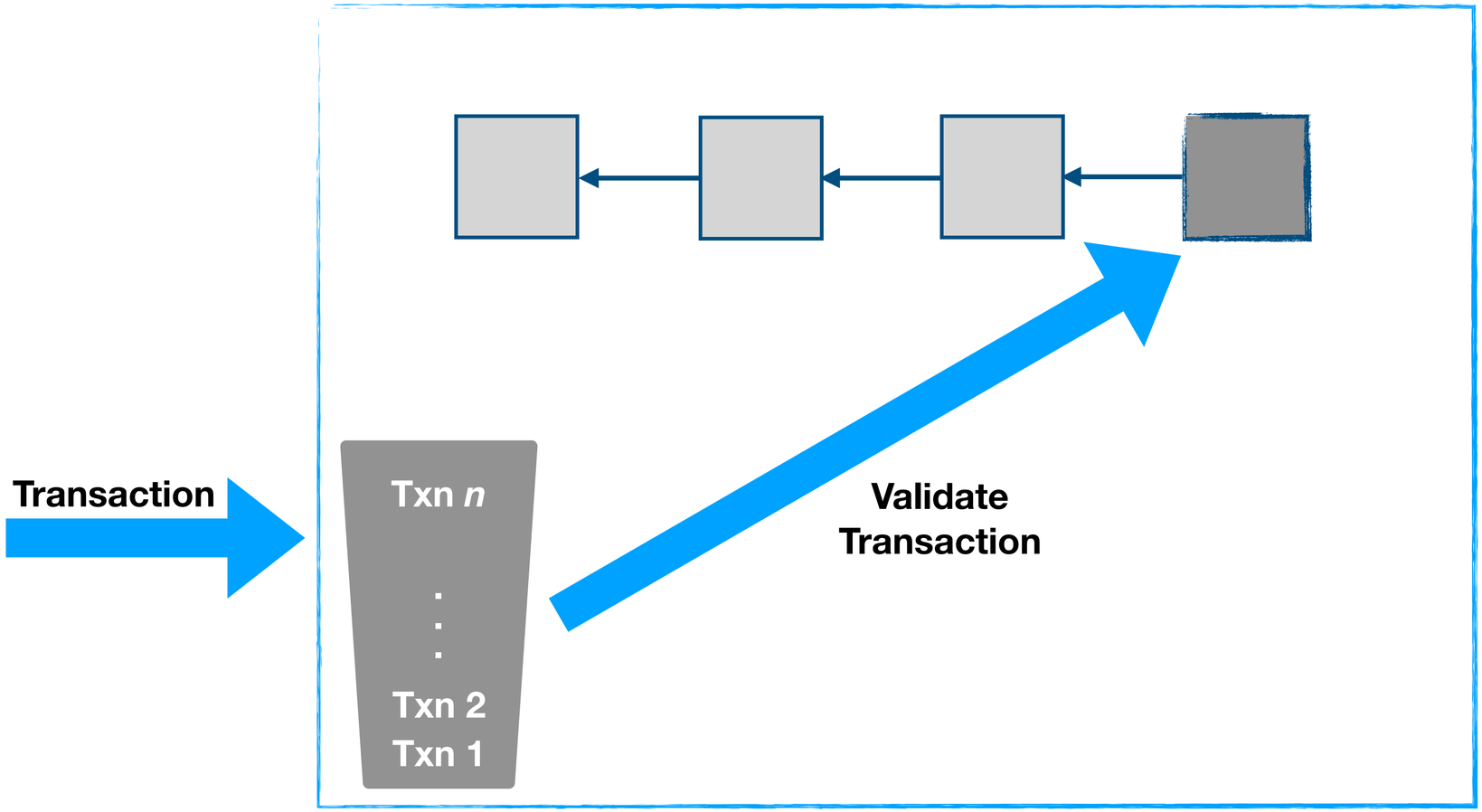


Where is the Chain stored?

Where is the Chain stored?

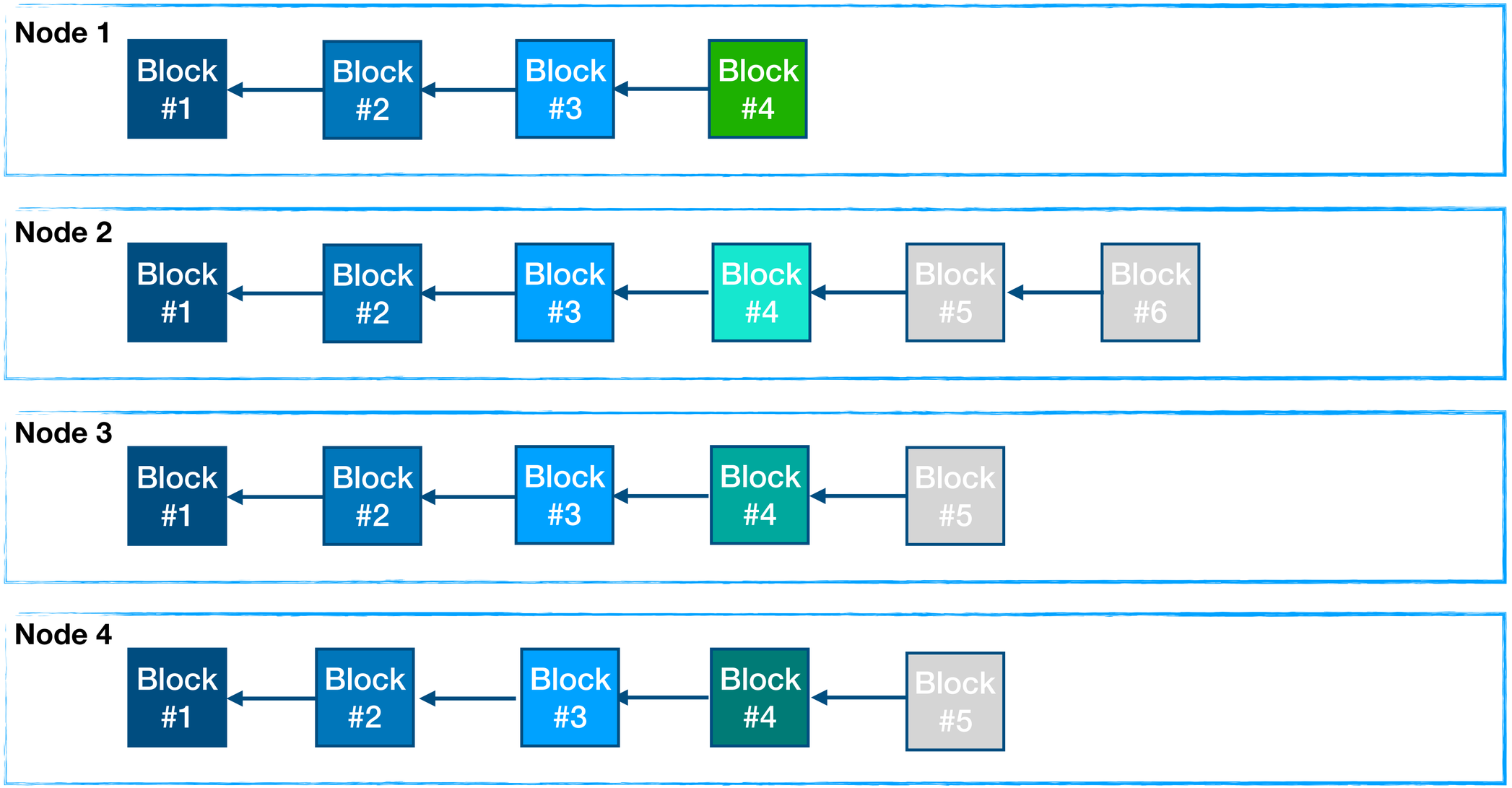


What's on each of the nodes?

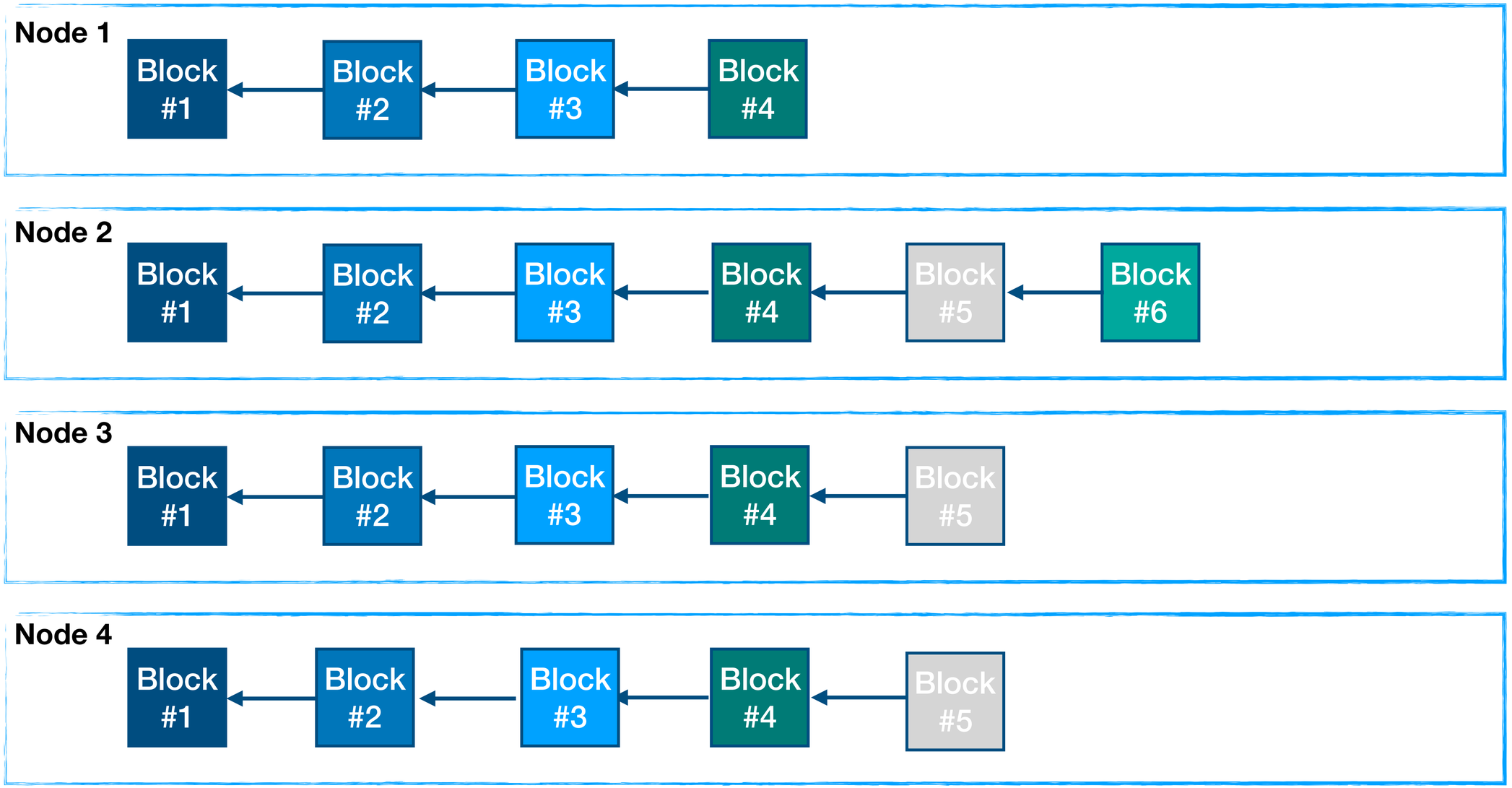


Problem:
Trust and Consistency

How can we avoid inconsistency?



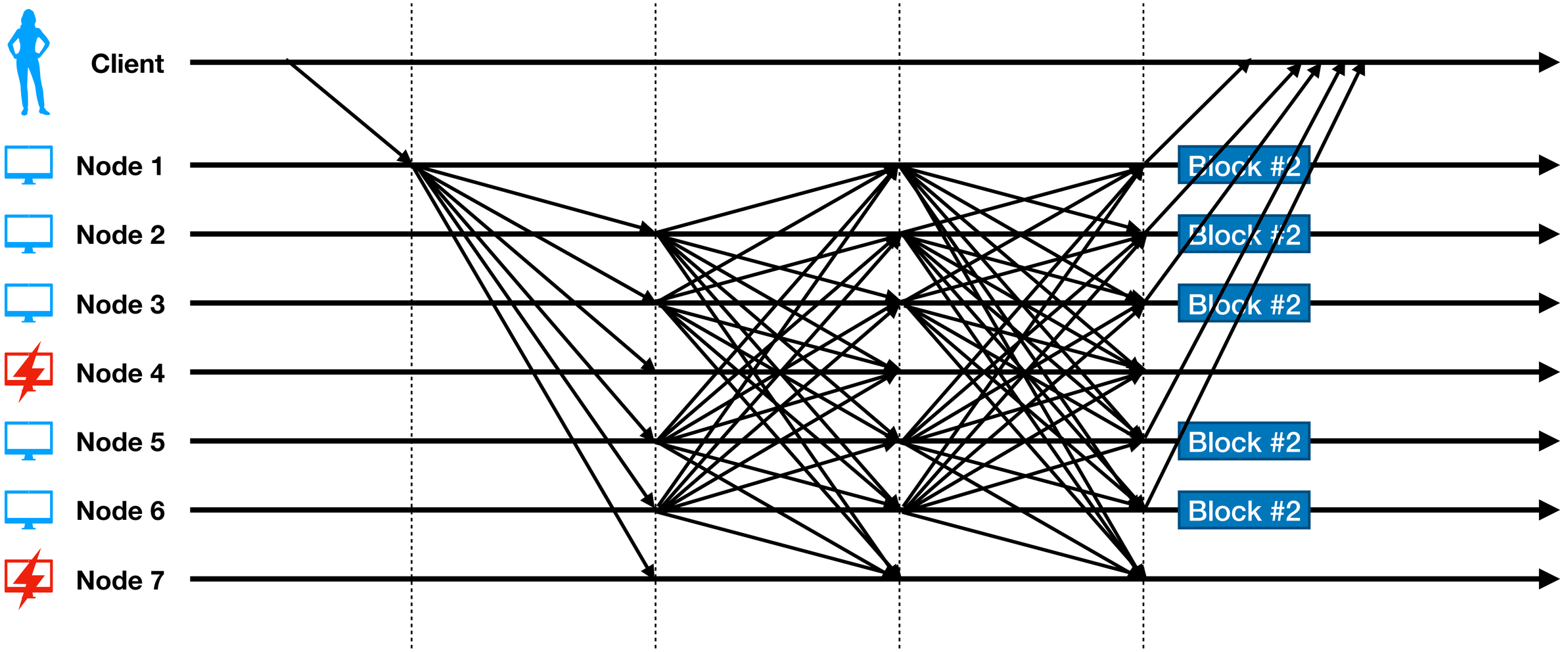
Consensus


Node 1
Block #1
Block #2
Block #3
Block #4
Node 2
Block #1
Block #2
Block #3
Block #4
Block #5
Block #6
Node 3
Block #1
Block #2
Block #3
Block #4
Block #5
Node 4
Block #1
Block #2
Block #3
Block #4
Block #5

Why consensus is hard?

- Nodes may crash
- Nodes may be malicious (or compromised)
- Network is imperfect
  - Not all pairs of nodes connected
  - Network failures
  - Latency is unpredictable

Classical Byzantine Consensus
Agree on a single block (voting)



Private Blockchains

- Byzantine consensus traditionally considered too expensive
  - Requires $3f+1$ nodes to tolerate f failures and many messages

Private Blockchains

- Byzantine consensus traditionally considered too expensive
  - Requires 3*f*+1 nodes to tolerate *f* failures and many messages
- Only *known* storage nodes can join the system
  - Otherwise, we would be susceptible to *Sybil attacks*
- Clients must also be *known* to have transactions accepted

Private Blockchains

- Byzantine consensus traditionally considered too expensive
  - Requires 3*f*+1 nodes to tolerate *f* failures and many messages
- Only *known* storage nodes can join the system
  - Otherwise, we would be susceptible to *Sybil attacks*
- Clients must also be *known* to have transactions accepted

HYPERLEDGER

Public Blockchains

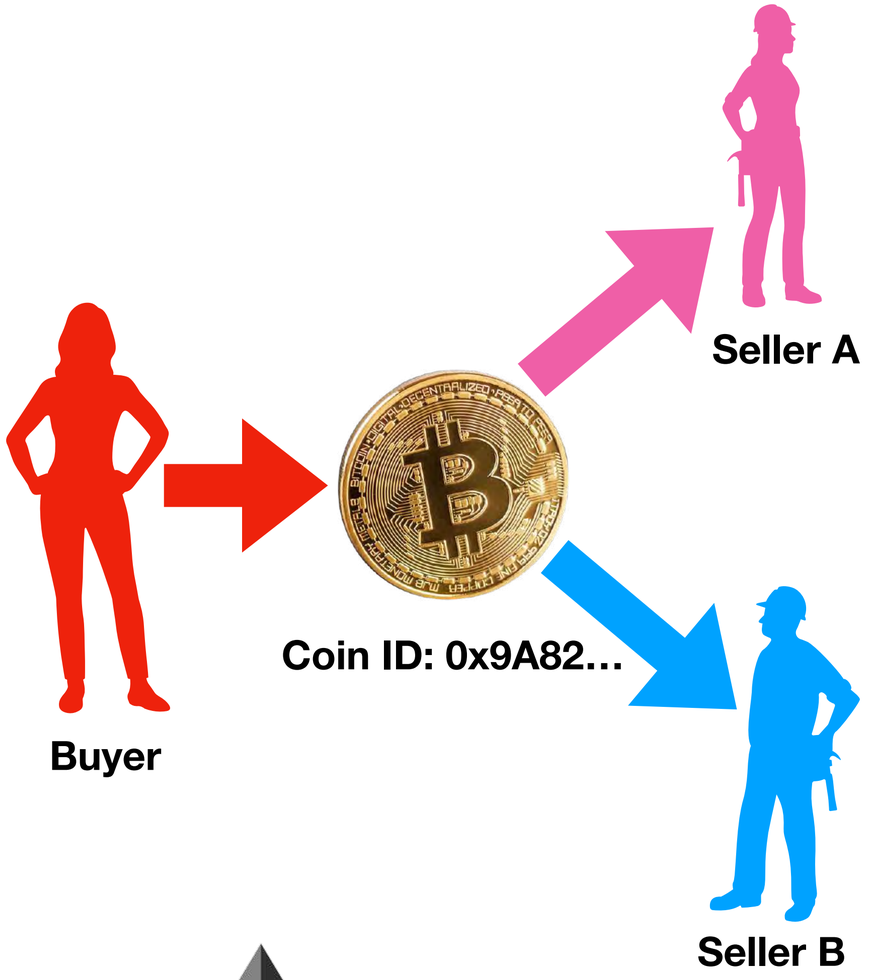
- Allow anyone (storage nodes) to join the system
- Incentivize node owners to
  - Deter creation of Sybils
  - Validate transactions to prevent *double spend attack*

ethereum

bitcoin

Public Blockchains

- Allow anyone (storage nodes) to join the system
- Incentivize node owners to
  - Deter creation of Sybils
  - Validate transactions to prevent *double spend attack*



ethereum

bitcoin

Public Blockchains

- Allow anyone to submit transactions
  - Anonymously
  - Real identity not directly connected your “blockchain” address
  - But observer can link together the activity of an address over time, and make inferences

ethereum

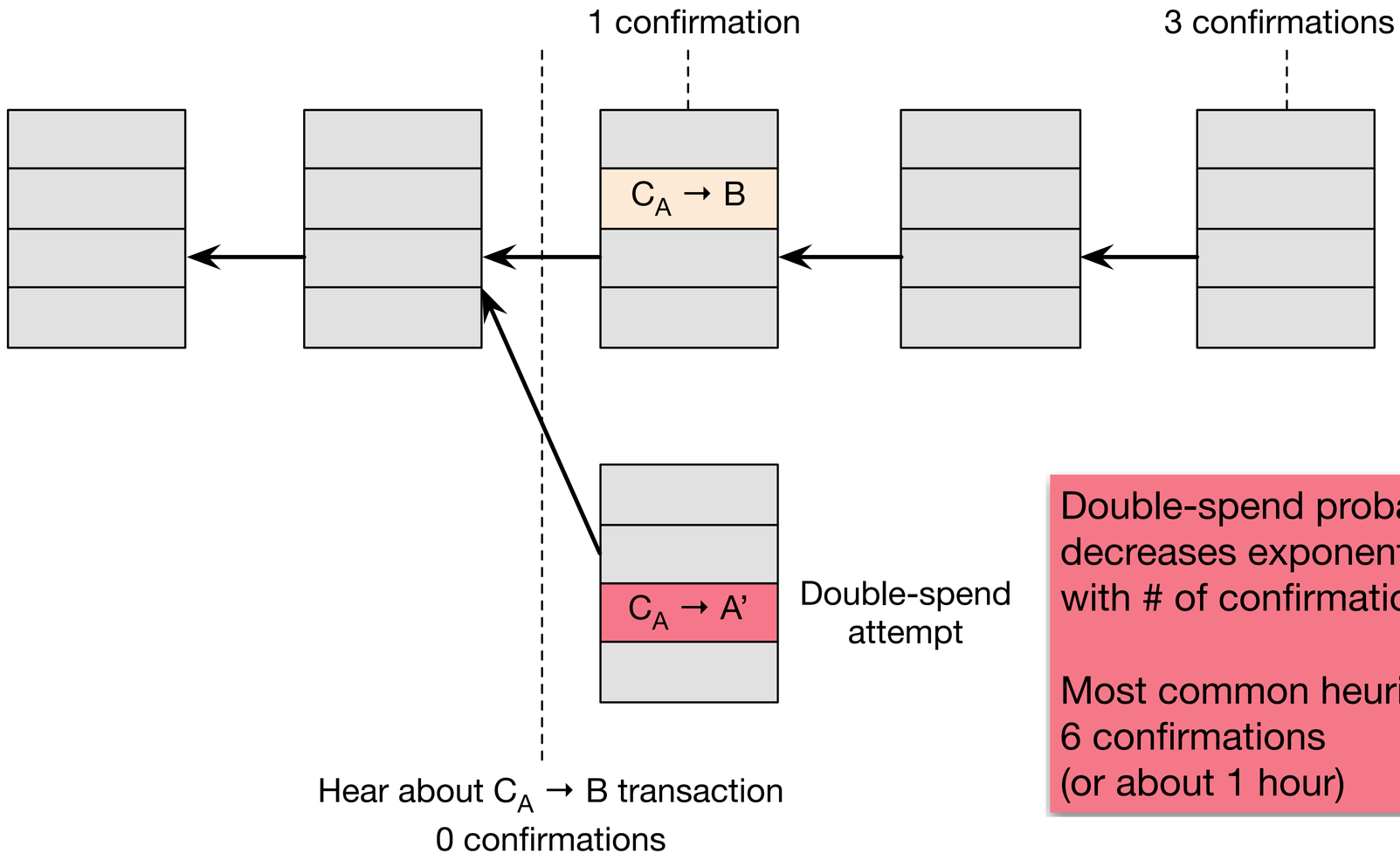
bitcoin

Implicit Consensus

1. New transactions are sent to all nodes
2. Each node collects new transactions into a block
3. In each round a random node broadcasts its block:
   proposing the next block in the chain
4. Other nodes implicitly accept/reject this block
   - by either extending it or
   - ignoring it and extending chain from earlier block
5. Every block contains hash of the block it extends

What can a Malicious Node do?

From Bob the merchant's point of view

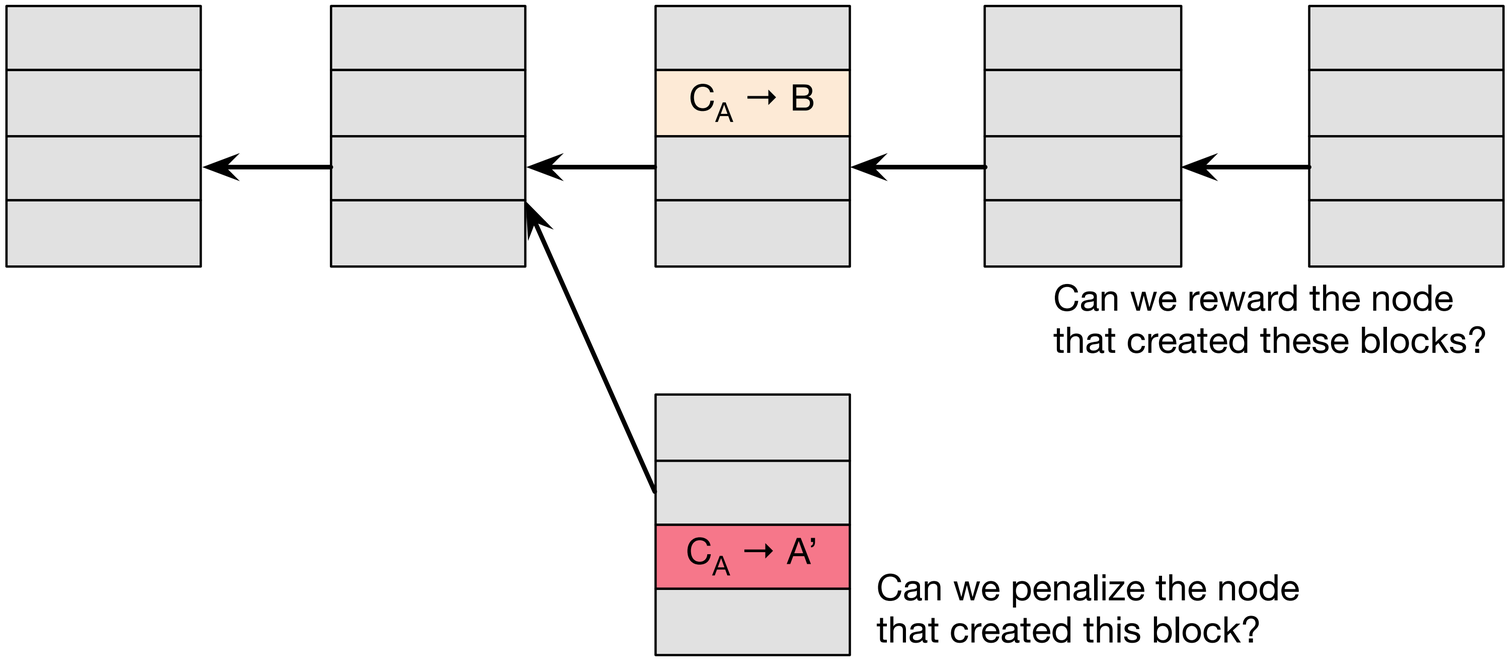


Double-spend probability decreases exponentially with # of confirmations

Most common heuristic:
6 confirmations
(or about 1 hour)

Assumption of Honesty is Problematic

Can we give nodes <u>incentives</u> for behaving honestly?

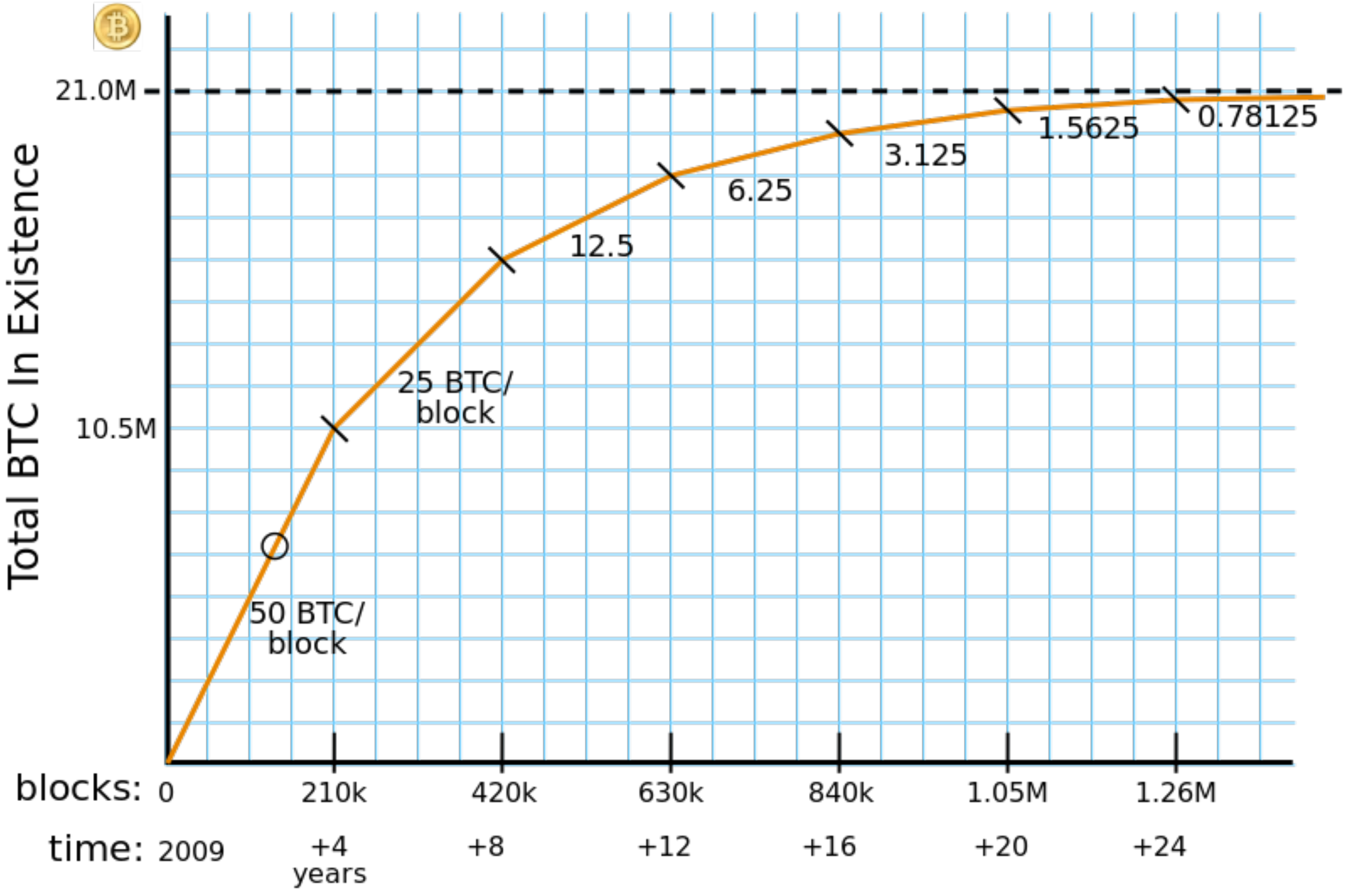


Everything so far is just a distributed consensus protocol

But now we utilize the fact that the currency has value

Incentive 1: Block Reward

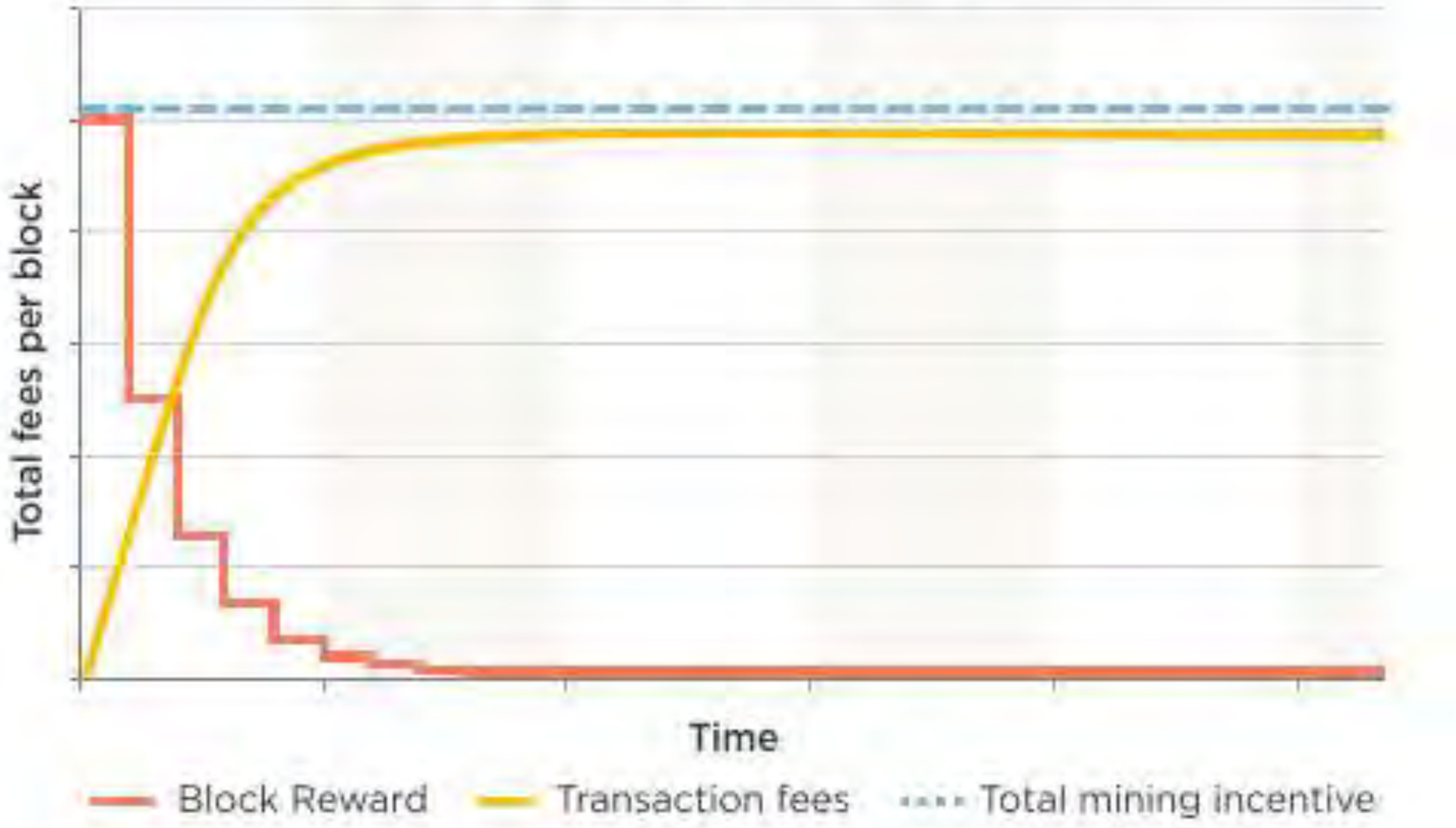
- Creator of block gets to
  - Add a CoinCreation transaction in the block
  - Choose recipient address of this transaction
- Block reward is fixed, but halves every 4 years:
  - originally 50 BTC
  - currently 12.5 BTC
- *Block creator gets to "collect" the reward only if the block ends up on long-term consensus branch!*


21.0M
10.5M
Total BTC In Existence
50 BTC/ block
25 BTC/ block
12.5
6.25
3.125
1.5625
0.78125
blocks: 0
210k
420k
630k
840k
1.05M
1.26M
time: 2009
+4 years
+8
+12
+16
+20
+24

Incentive 2: Transaction fees

- Creator of transaction can choose to make output value less than input value
- Remainder is a transaction fee and goes to the block creator
- Purely voluntary, like a tip
  - But if you don’t leave a tip it is unlikely your transaction will be processed

TRANSACTION FEES ARE MEANT TO REPLACE BLOCK REWARDS




— Bitcoin - Avg. Transaction Fee
Avg. Transaction Fee, USD
60
55
50
45
40
35
30
25
20
15
10
5
0
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019

Remaining Problems

1. How to pick a random node?
2. How to avoid a free-for-all due to rewards?
3. How to prevent Sybil attacks?

Proof-of-Work

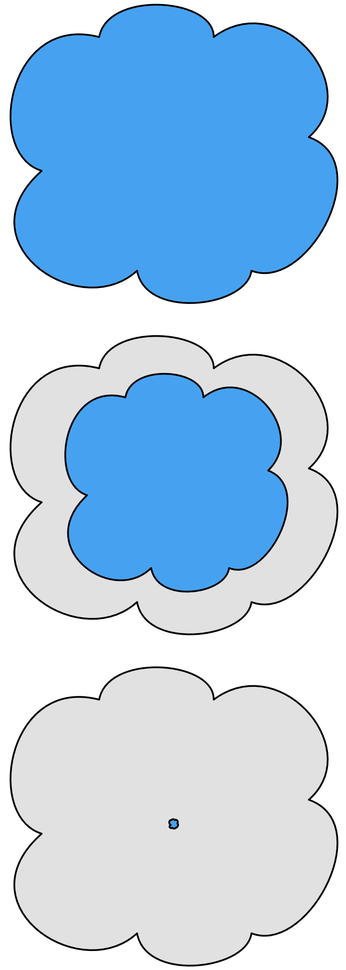
- To approximate selecting a random node:
  - Select nodes in proportion to a resource that no one can monopolize (we hope)
- Equivalent views of proof-of-work
  - Select nodes in proportion to computing power
  - Let nodes compete for right to create block
  - Make it pointless to create new identities

Search Puzzle with Hash Function

- Given a puzzle k and a target set Y, try to find a solution x, such that:

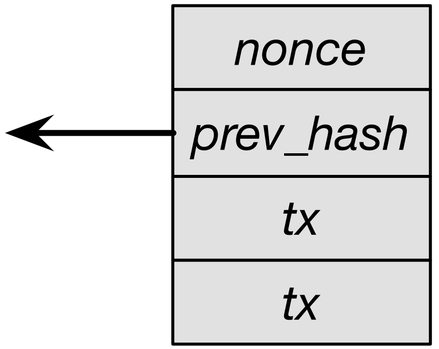
$$H(k \mid x) \in Y$$

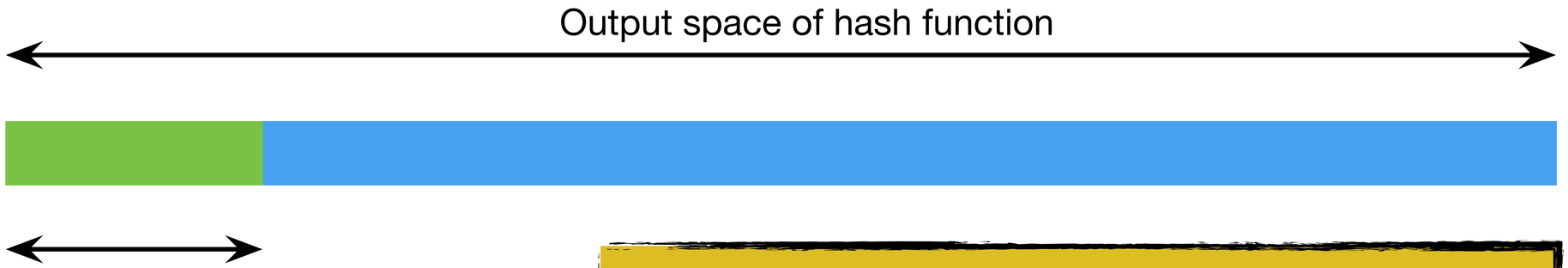
- For n-bit hash function, we have 2^n possible outputs
- Can adjust puzzle difficulty by varying the target set Y



Hash Puzzles

To create a block, find *nonce* such that
H(*nonce* | *prev_hash* | *tx* | … | *tx*)
is very small





If hash function is secure:
only way to succeed is to try
enough nonces until you get lucky

CPU

GPU

FPGA

ASIC

gold pan

sluice box

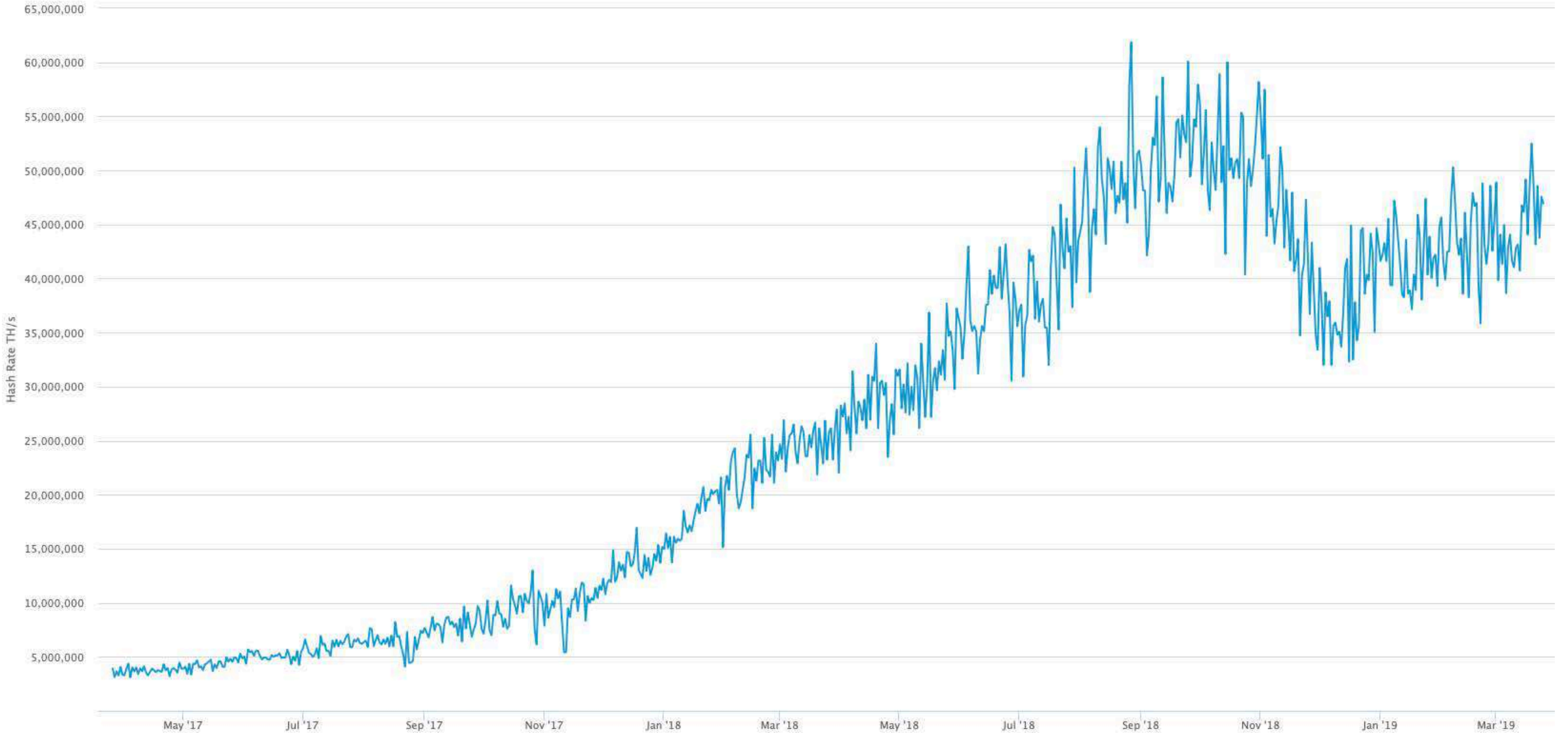
placer mining

pit mining

Hash Rate

The estimated number of tera hashes per second (trillions of hashes per second) the Bitcoin network is performing.

Source: blockchain.com



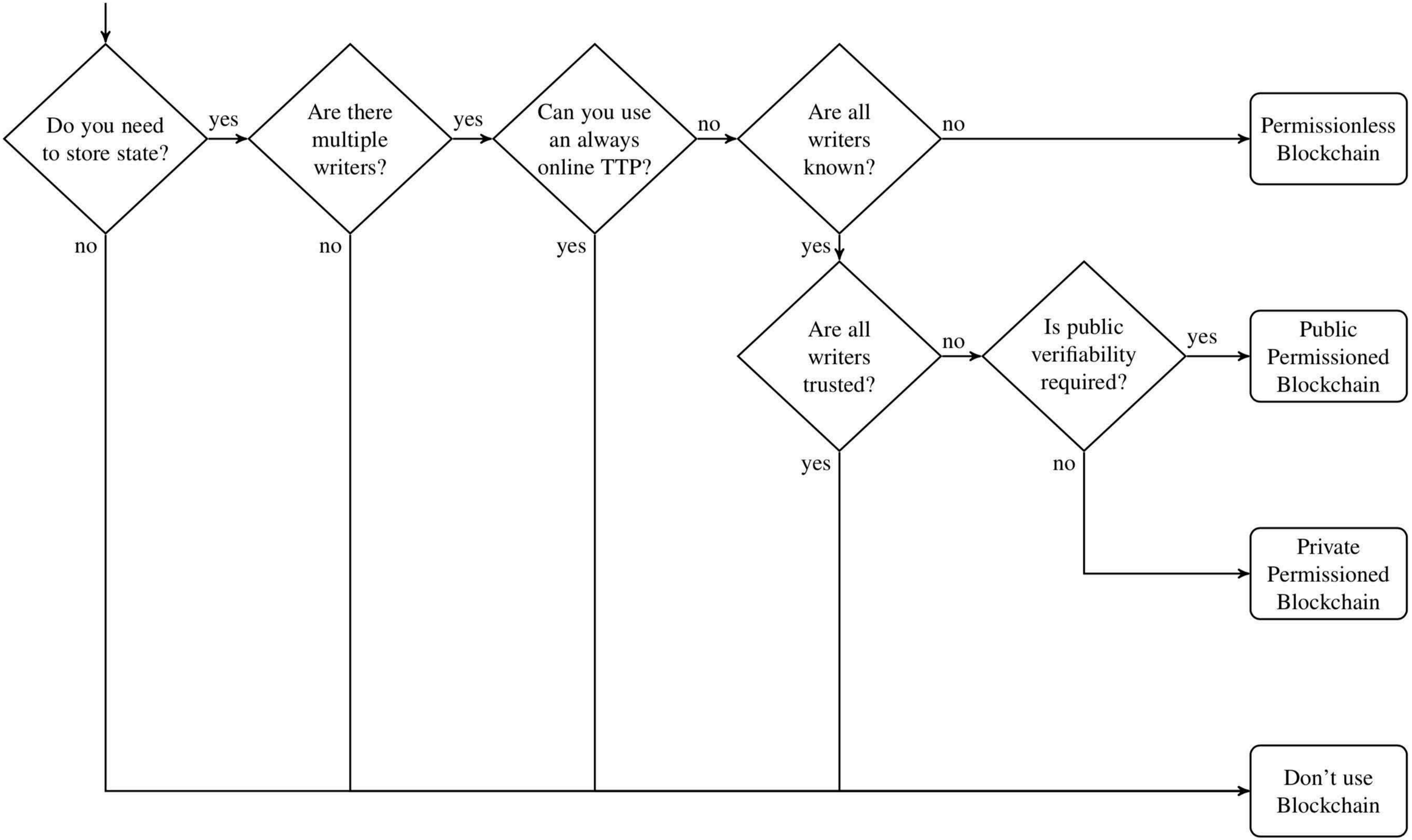
When does it make sense to use a Blockchain?

When to use a Blockchain?

When multiple *mutually mistrusting entities* want to interact and change the blockchain, and are not willing to trust a third party.

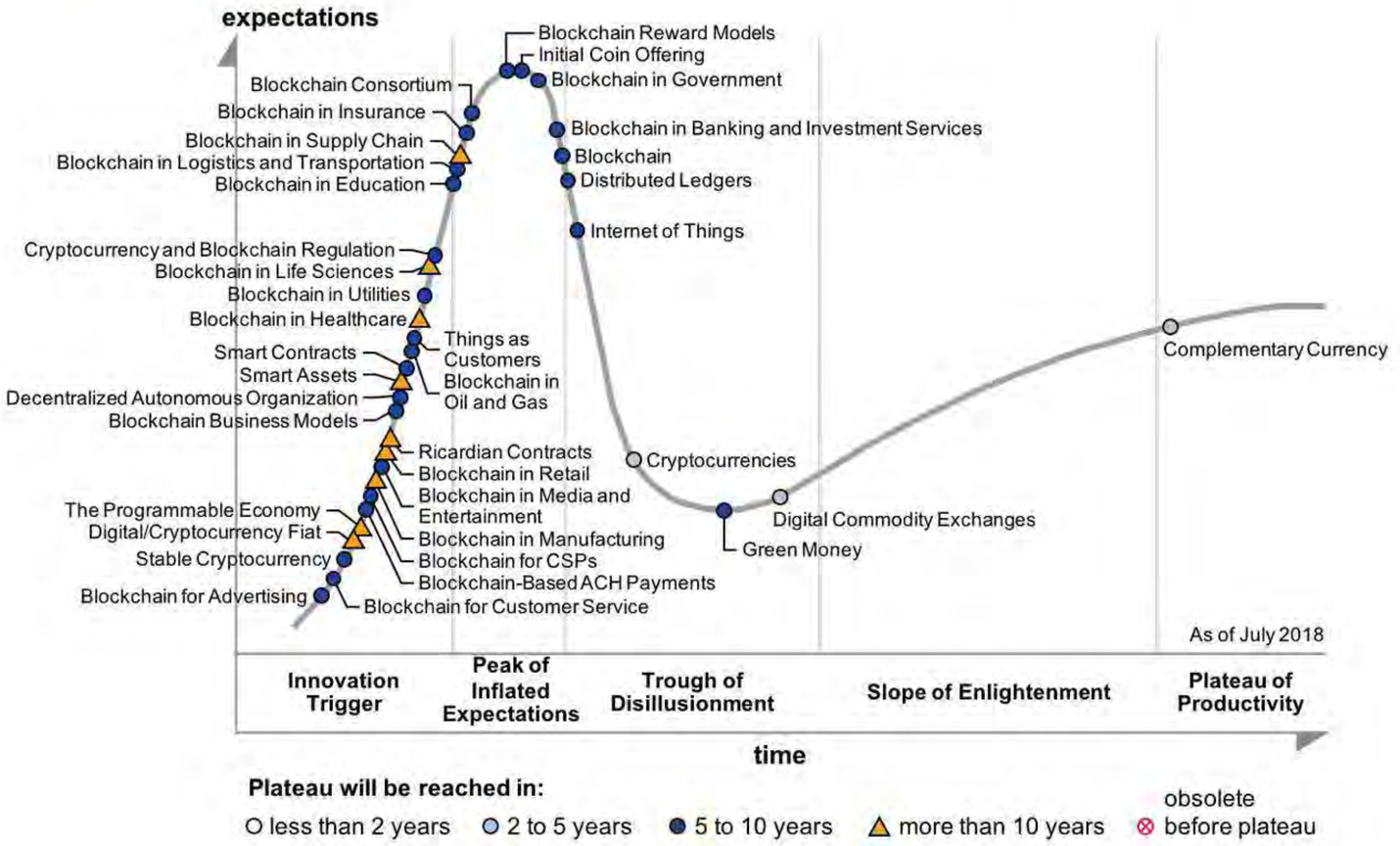
Blockchain Classification

	Public	Private
Access (Transparency)	Open read/write (transparent)	Permissioned read and/or write (private/closed)
Identity (Privacy)	Anonymous Pseudonymous	Know identities
Speed (tps)	Slow	Fast
Scalability	Poor	Decent
Trust	Trust-free	Trusted nodes
Security	Proof-of-Work Proof-of-Stake++	Pre-approved participant nodes
Energy Use	Extremely high	Low


Do you need to store state?
yes
Are there multiple writers?
yes
Can you use an always online TTP?
no
Are all writers known?
no
Permissionless Blockchain
no
no
yes
yes
Are all writers trusted?
no
Is public verifiability required?
yes
Public Permissioned Blockchain
yes
no
Private Permissioned Blockchain
Don’t use Blockchain

Use Cases

Hype Cycle for Blockchain Business, 2018

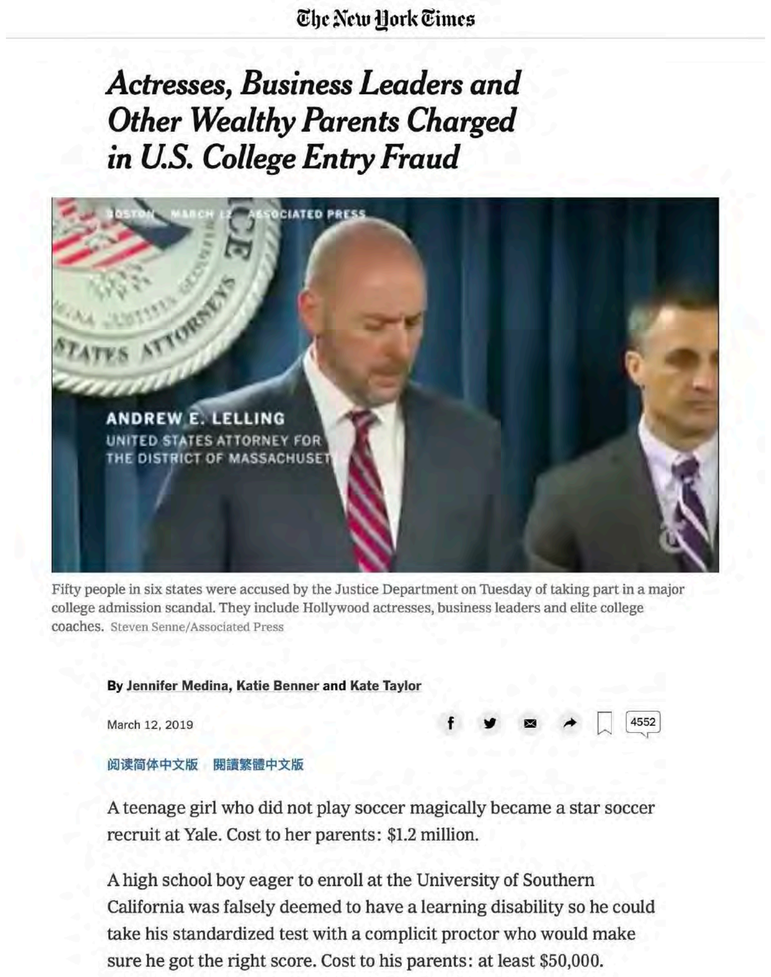


gartner.com/SmarterWithGartner

Gartner®

Some Envisioned Use Cases

- Distributed file storage
  - Similar to Dropbox
- Supply chain management
- Interbank and international payments
- Healthcare
- Ownership and royalty distribution
- Recording of degree certificates



The New York Times

Actresses, Business Leaders and Other Wealthy Parents Charged in U.S. College Entry Fraud

Fifty people in six states were accused by the Justice Department on Tuesday of taking part in a major college admission scandal. They include Hollywood actresses, business leaders and elite college coaches. Steven Senne/Associated Press

By Jennifer Medina, Katie Benner and Kate Taylor

March 12, 2019

4552

阅读简体中文版 開讀繁體中文版

A teenage girl who did not play soccer magically became a star soccer recruit at Yale. Cost to her parents: $1.2 million.

A high school boy eager to enroll at the University of Southern California was falsely deemed to have a learning disability so he could take his standardized test with a complicit proctor who would make sure he got the right score. Cost to his parents: at least $50,000.

Supply Chain Management

Supply chain

Blockchain can enable manufacturers to improve track and trace of components and finished goods across a supply chain and enable new applications around anticounterfeiting, supplier and purchaser financing, or management of supply chain disruptions and recalls.

The Blockchain Oracle

An oracle is an agent that finds and verifies real-world occurrences and submits this information to a blockchain.

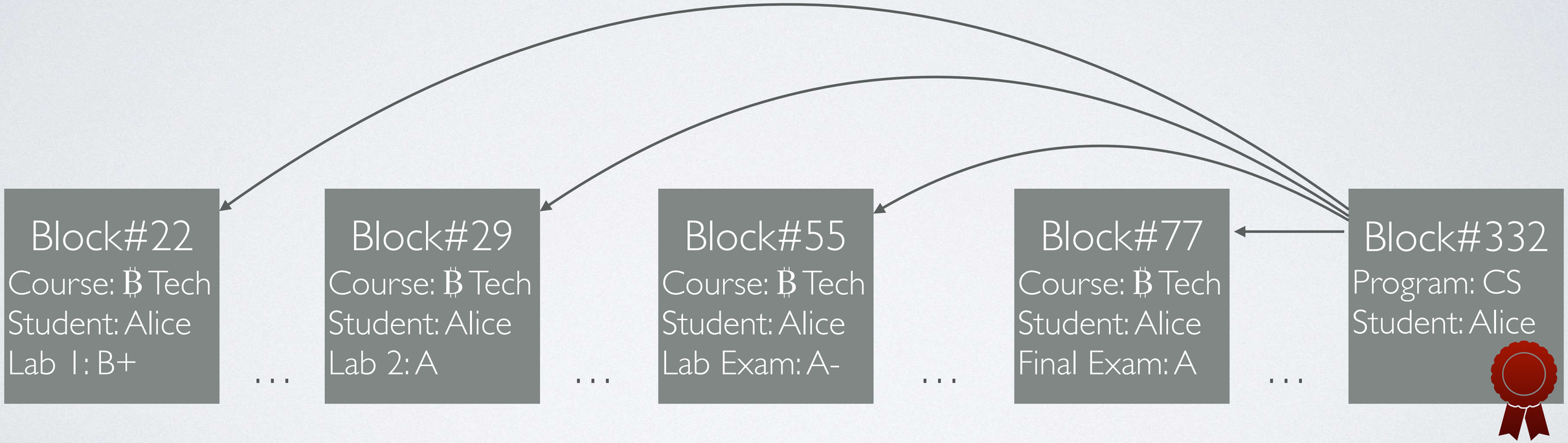
MOTIVATION

- Student admission process
- Admissions committees
- Refugees — long-term storage

PREVENTING FAKE CERTIFICATES WITH LEDGER-BASED RECORDING

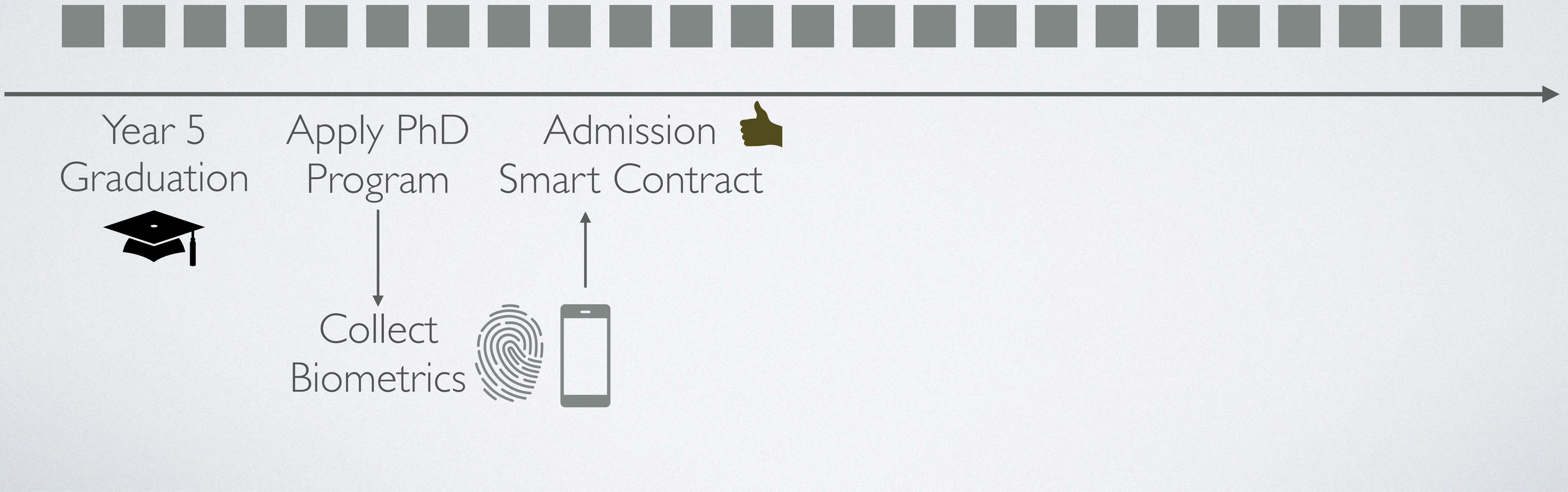
- Record evaluation events when they happen
  - Record individual exams, or even individual assignments
  - Include meta data about exam/assignments
- Create a final degree certificate linking together evaluation events

PREVENTING FAKE CERTIFICATES WITH LEDGER-BASED RECORDING



PREVENTING FAKE CERTIFICATES WITH LEDGER-BASED RECORDING

VERIFYING CERTIFICATES



PERMANENT STORAGE OF DEGREE CERTIFICATES

- Students can carry/store their degree certificate
  - But cannot modify it
- Also want to store degree certificate permanently (that is for a life-time)






Resilient Systems Lab

Thanks!

bbchain.no

@heinmeling

hein.meling@uis.no