Bitcoin, blockchain, etc

Bo Waggoner
UPenn theory lunch 2017-06-16

Forewarning: this will probably contain mistakes

Agenda

- Philosophy
- What is Bitcoin (and what’s a blockchain)
- Interesting issues
- History
- What is Ethereum (and what’s a smart contract)

A perspective on “fiat money”

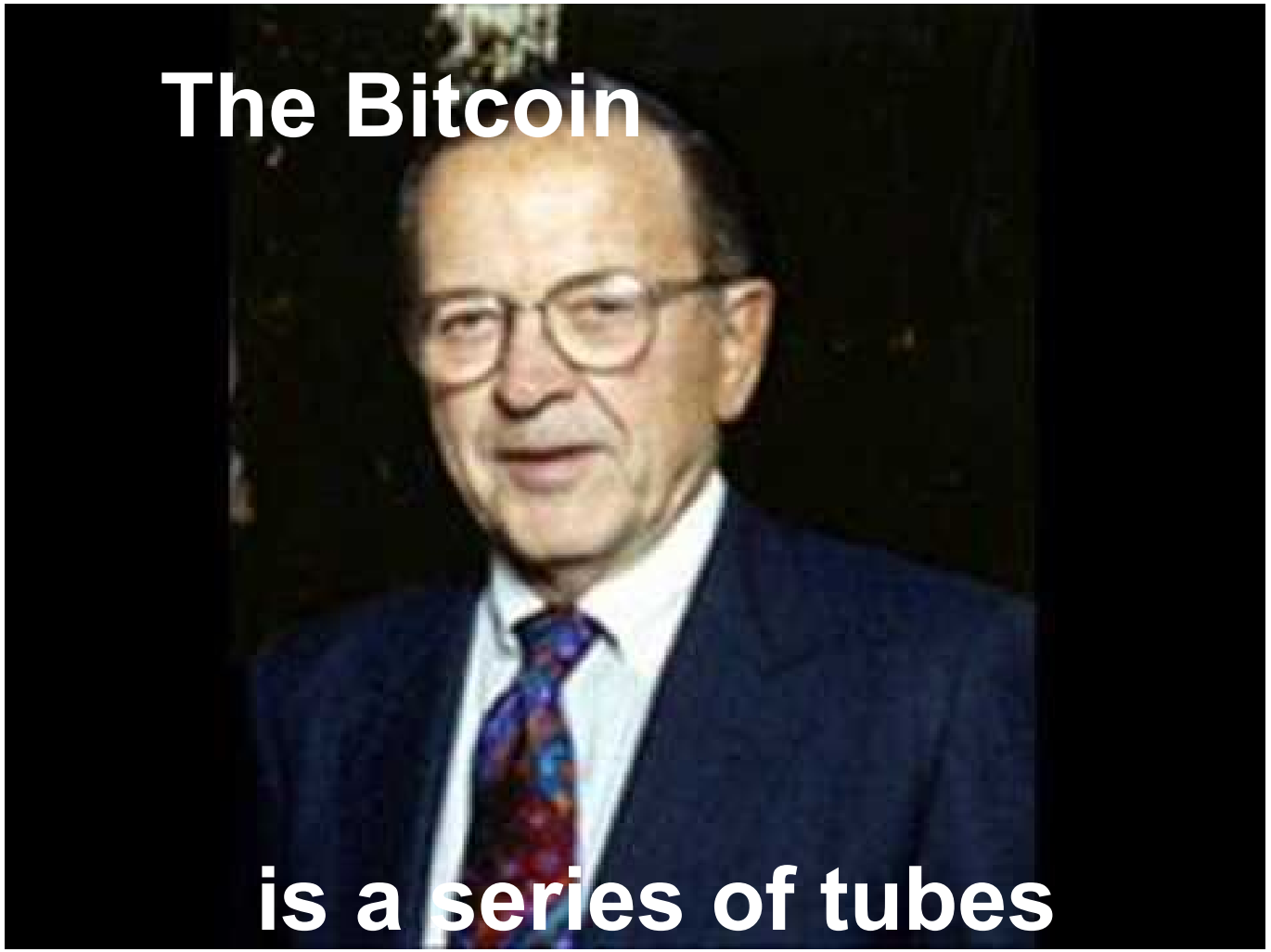
- Money is a reputation system; it links transactions over time.
- The most general such system: Given entire history, determine if a current transaction is “legal” and if so, the new state of the system.
- Cash is an incredibly simple / compressed protocol.
  - Enabled by physical, non-counterfeitable tokens; or digitally by trusted authorities.
  - (The government printing new money is just part of the protocol.)
- Without physical tokens, fiat money can still be implemented: use a more complex/memory-intensive **public data structure**. (list all previous transactions)

Challenges in implementing digital cash

1. Forgery, impersonation (easy)
    a. Each “coin” is owned by some public key. Only the associated private key can “spend it”: digitally sign a transfer to some other public key.

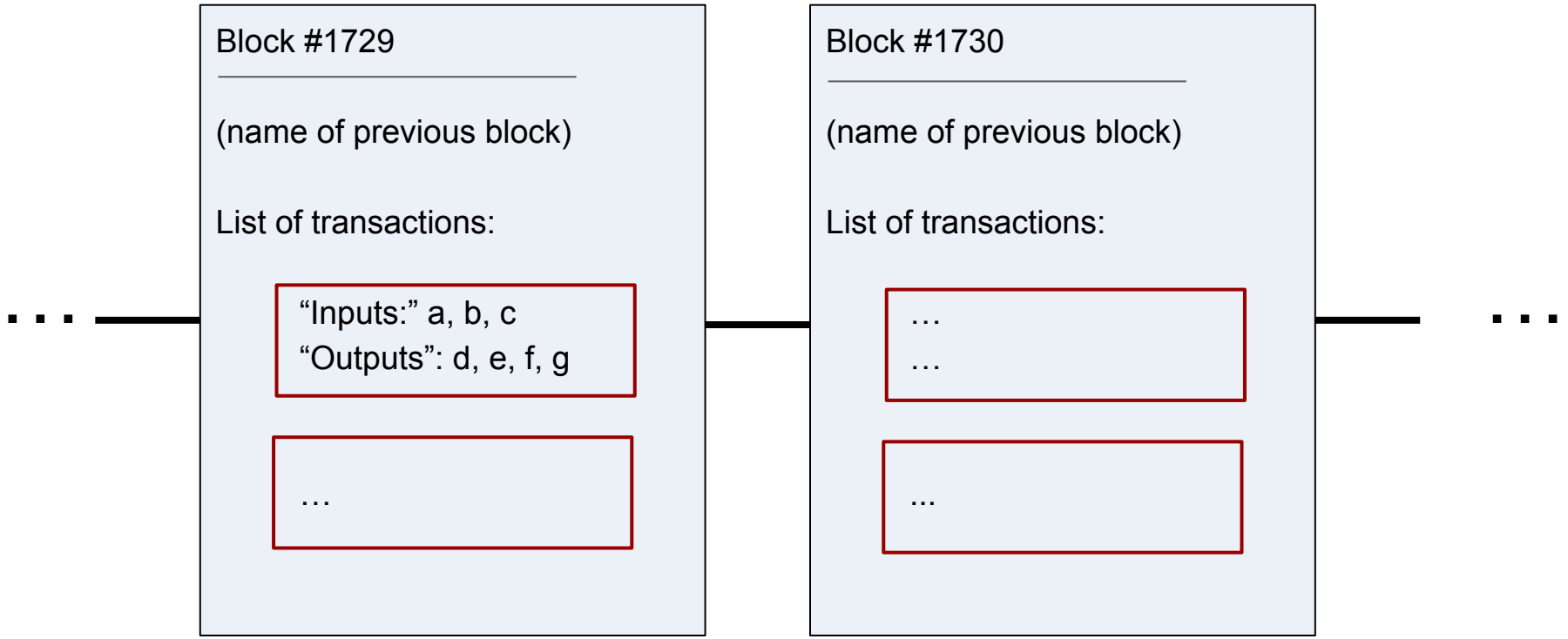
2. Digital consensus (hard)
    a. If this were all taking place on one computer, we’d be done.
    b. The **blockchain** aims to implement it without a central trusted authority.

The Bitcoin protocol

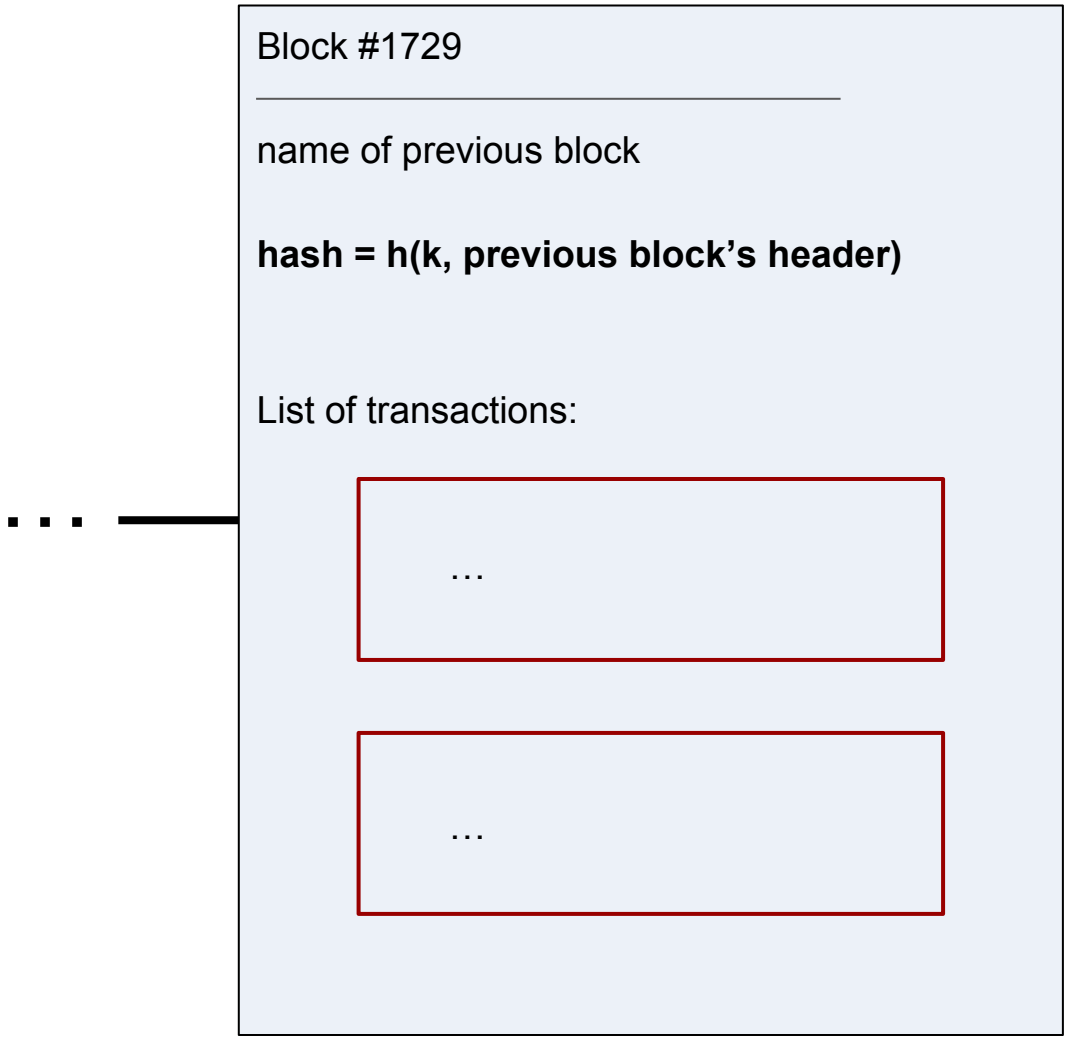


Bitcoin, simplified

A transaction is e.g. “5 coins from Alice to Bob”.
A “block” has a set of transactions. The blockchain is a chain of … yeah.
We all keep a copy of the blockchain, representing a consensus history.



Inside a block



A block is only valid if **hash** is smaller than some predefined goal.

Miner's problem: Find **k** so that **hash <=** goal.

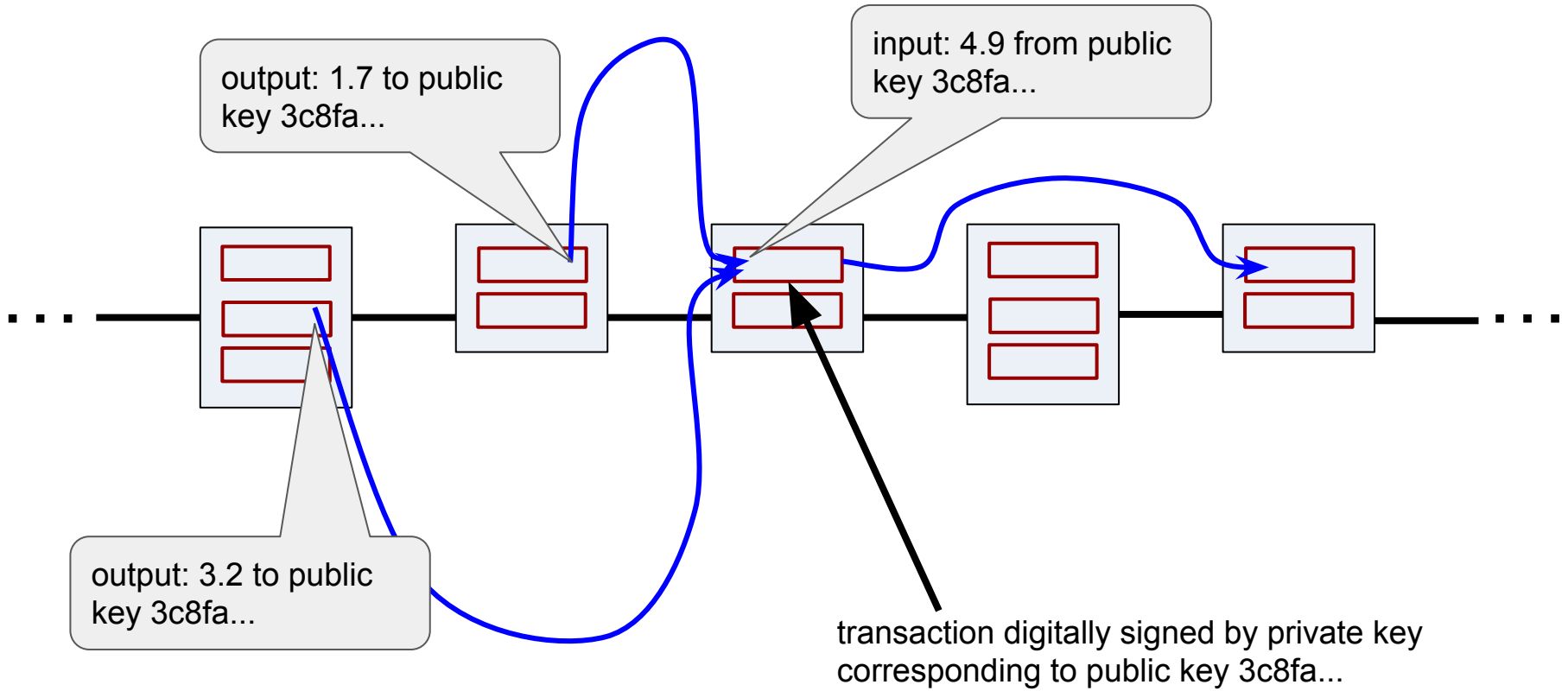
Once such a **k** is found, miner creates and announce the new block (including in it any transactions* she wants).

Anyone can publicly announce a transaction* to be included in the next block (or hopefully soon), including "transaction fees" the miner can award themself.

Over time, "goal" decreases, making this problem harder.

*if transaction's digital signature doesn't match up, it's invalid.

“Cashflow”



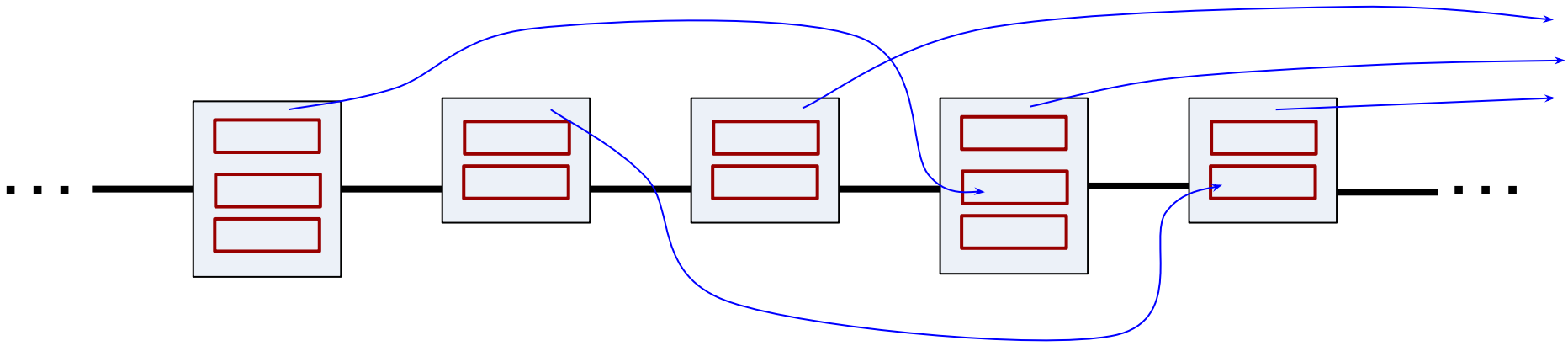
Note: Can use a different key pair for every transaction (recommended), or keep the same one.

Creation of new coins

Each block has one “output” with no “inputs”: brand-new coins.

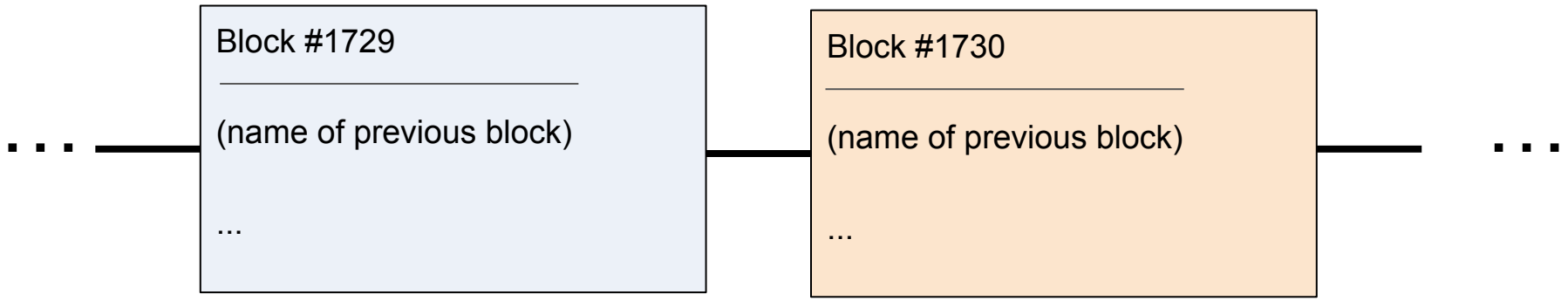
The person creating the block chooses where they go (presumably self).

The number created halves over time; total is capped.



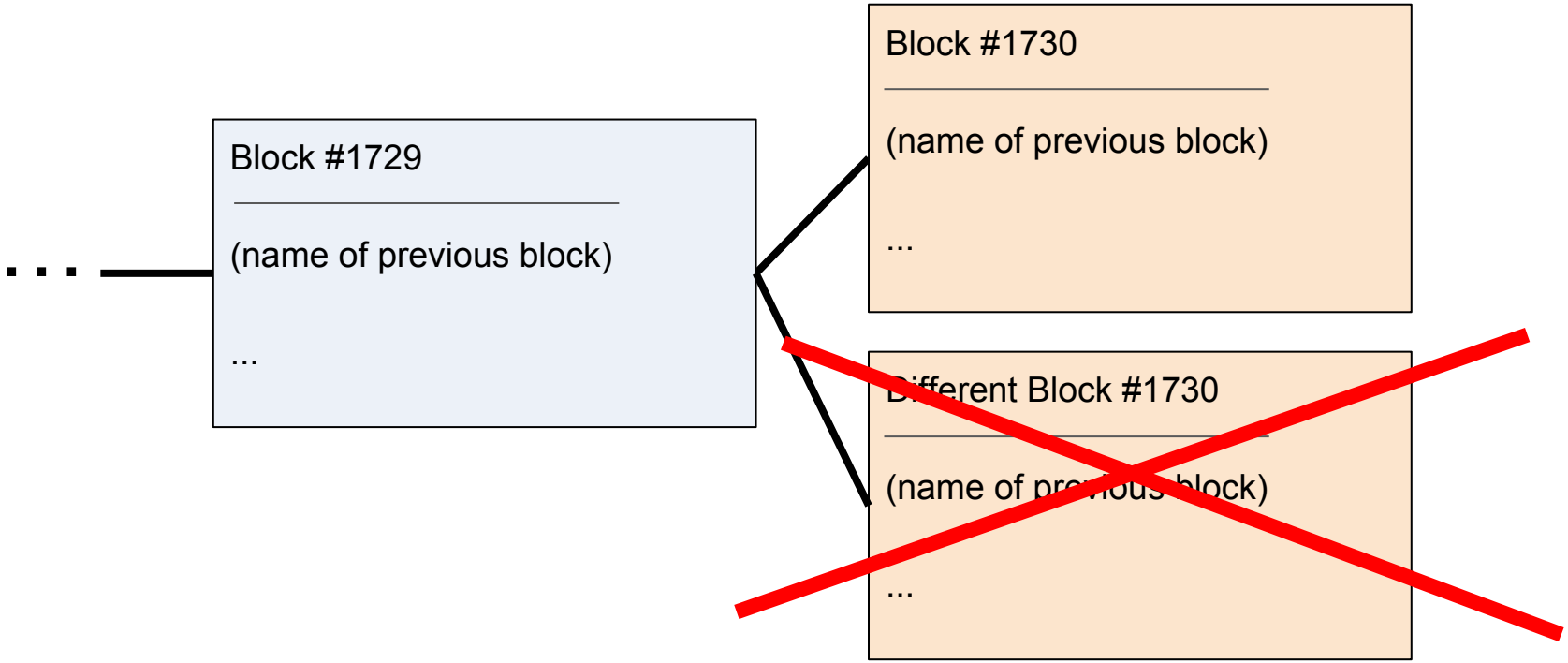
Distributed consensus algorithm

- Every node keeps a copy of the whole blockchain.
  - (Note: “Merkle tree” is used to quickly check/verify, rather than re-reading 100+ GB each time)
- When someone else announces a new block, check that it is “valid” and add it to to the chain.



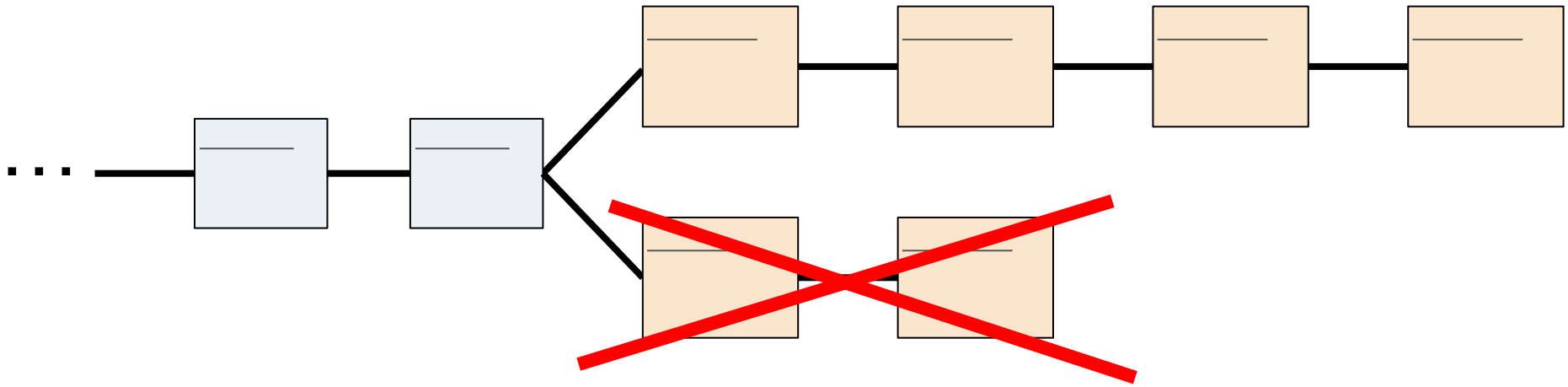
Distributed consensus algorithm cont'd

- If you see multiple new blocks, pick only one (preferably the older)



Distributed consensus algorithm cont'd

- More generally, if you see different versions of the chain, pick whichever version is longer.



Aside: Blockchains in general

The blockchain is a **modular** invention separate from Bitcoin.

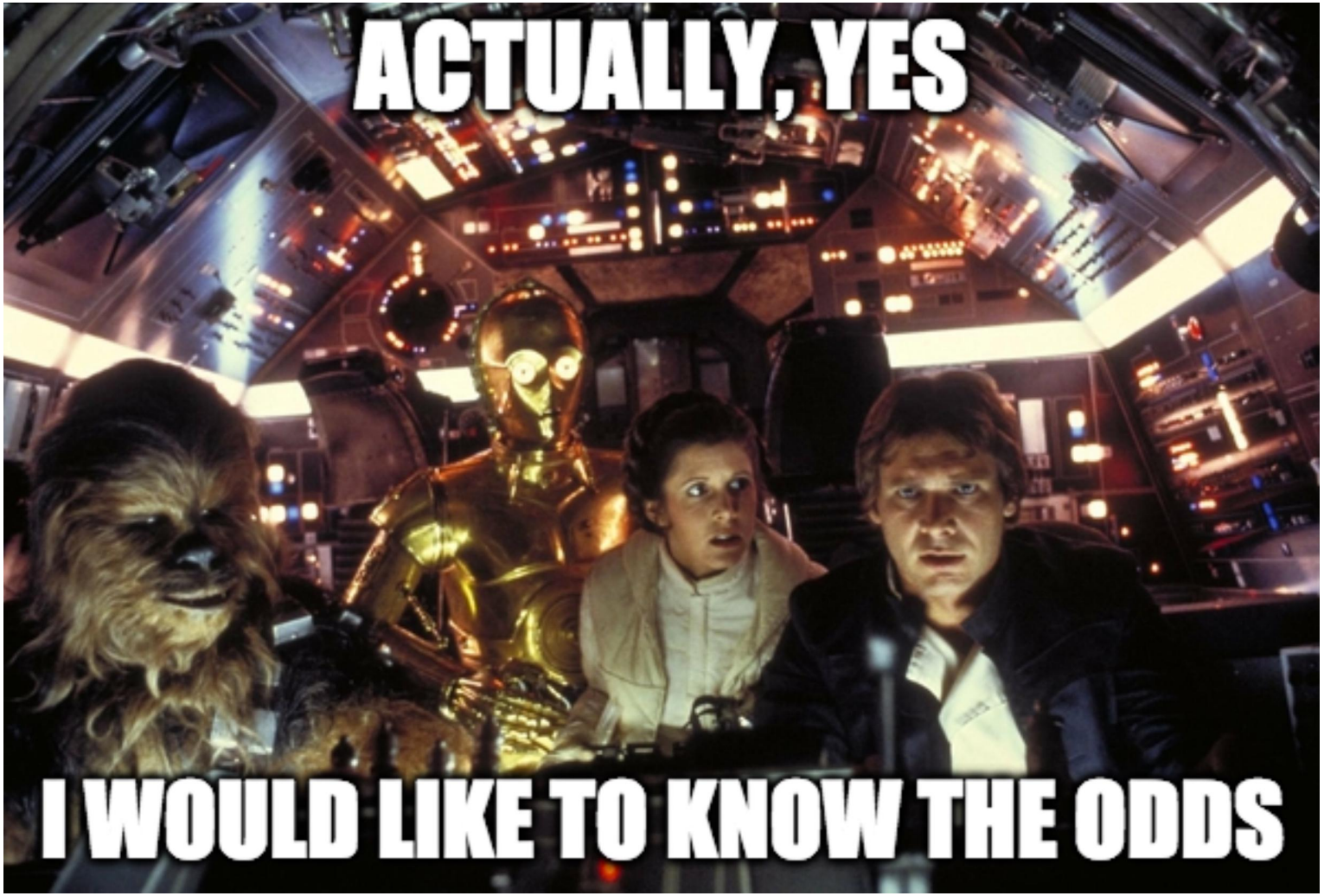
The stuff in the blocks did not have to be a list of transactions - it could have been any **dynamic data structure**.

(But you need incentives for people to mine new blocks…)

Should we put _____ on the blockchain?
Use your head! Does ____ need a **distributed, dynamic, consensus** algorithm?

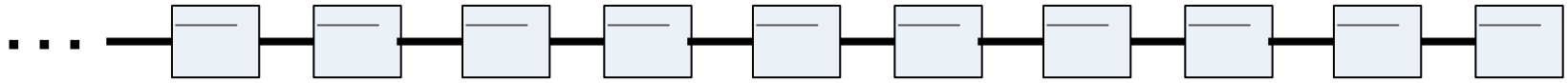
Example: my company wants to internally track history of ____.
You can probably track history of ____ on a single trusted computer with a database. What would a blockchain do other than waste energy and storage?

Possible problems



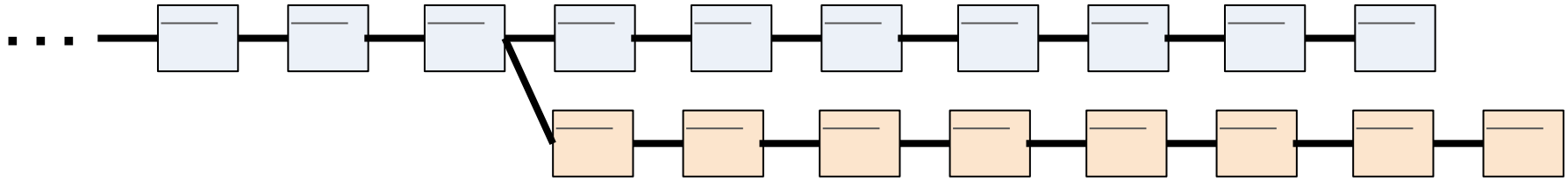
Musings on the blockchain (so to speak)

- You can never be 100% certain that a spent coin is “spent”
  - e.g. with 51% of the mining power, someone can create the longest chain at will.
  - In theory, they can reverse/cancel as many transactions as they want.



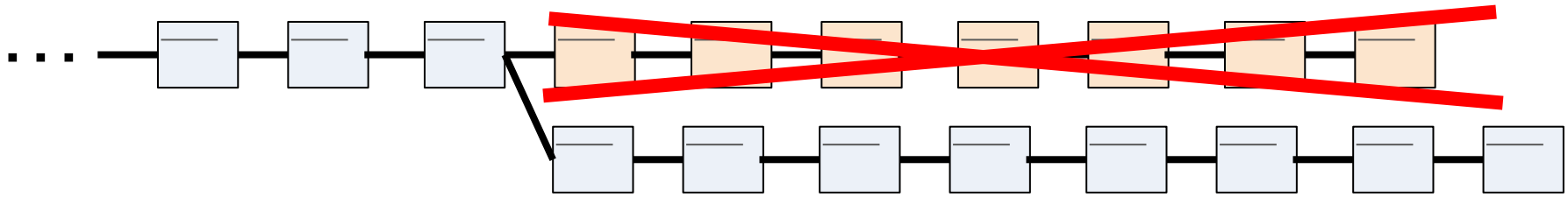
Musings on the blockchain (so to speak)

- You can never be 100% certain that a spent coin is “spent”
  - e.g. with 51% of the mining power, someone can create the longest chain at will.
  - In theory, they can reverse/cancel as many transactions as they want.



Musings on the blockchain (so to speak)

- You can never be 100% certain that a spent coin is “spent”
  - e.g. with 51% of the mining power, someone can create the longest chain at will.
  - In theory, they can reverse/cancel as many transactions as they want.



- Currently: 10 mins / block on average.
  - The sooner you find out about a block, the sooner you start mining the next one.
  - (If you create one, maybe don’t tell everyone else immediately.)
  - If most bitcoins are mined in big datacenters in China, you’re best off mining them in your own big datacenter next door, not seconds away in the US.

- “Block size:” limited # of transactions fit in a block. Competition for space ⇒ larger transaction fees. (Issues: http://www.cs.princeton.edu/~smattw/CKWN-CCS16.pdf)

“Proof of work” vs “Proof of stake”

- Above: “proof of work” protocol for creating new blocks:
  - find the hash key, get the bitcoin and create the new block.
  - Global result: lots of “wasted” energy spent simultaneously inverting hash functions

- Alternative that burns much less energy: “Proof of stake”
  - Algorithmically designate someone or small set of people to create the next block, e.g. with probability proportional to number of coins owned.
  - Lots of incentive issues arise, ad-hoc solutions suggested
  - **Can we make the incentives work? Open question.**

History


…, and this is crazy
but here's my cryptocurrency
so mine it, maybe?
VEVO

Abbreviated history of Bitcoin

- **October 2008:** “Satoshi Nakamoto" publishes whitepaper (https://bitcoin.org/bitcoin.pdf). True identity remains unknown.
- **January 2009:** Bitcoin launched, adopted. Nakamoto has not spent any coins since then, is currently estimated to hold ~$2.8 billion worth of Bitcoin.
- **August 2010:** Major bug exploited in bitcoin to create billions of bitcoins. The protocol is updated to fix the bug and officially "forked" to cancel those transactions.
- **September 2013:** US gov't shuts down "Silk Road", online drug marketplace, seizing 174,000 bitcoins valued at that time at $120 million.
- **February 2014:** Mt. Gox, the largest bitcoin exchange which handled over 70% of all bitcoin transactions, suspends trading, files for bankruptcy, and announces that 850,000 bitcoins were stolen ($450 million at that time), with most remaining "stolen" to this day.
- **Present day (mid 2017):** ~ $40 billion worth of BTC total, estimated 3-6 million unique users; accepted by Paypal, Microsoft, Dell, many minor vendors.

Ethereum - briefly and probably incorrectly

Ethereum (as I understand it)

- Idea: **use blockchain to run “consensus” computations.**
- Each block can contain financial transactions (the “currency” is named Ether) … or **code**.
- Every node maintains a copy of the Ethereum Virtual Machine.
  See a new valid block → execute its code.
- New code can “pass messages” to existing code.
- Example: one simple smart contract is an escrow: Waits until it receives a message that enough time has passed, then releases the money to the appropriate public key.

Ethereum history

Abbreviated history of **Ethereum**

- **late 2013:** idea for Ethereum proposed by Vitalik Buterin, Russian programmer/ researcher (non-academic).
- **mid-2014:** Development and coding, supported by crowdfunding. Founders Vitalik Buterin, Gavin Wood and Jeffrey Wilcke.
- **July 2015:** Ethereum goes live.
- **May 2016:** "The DAO" (Decentralized Autonomous Organization) raises “$150 million” in crowdfunding. It is implemented "on" the Ethereum blockchain with open-source code.
- **June 2016:** User extracts 1/3 of the money in the DAO exploiting unintentional behavior in the code.
- **July 2016:** Ethereum community decides to "fork" Ethereum: "Ethereum Classic" recognizes the legality of the heist, while the main current fork of Ethereum reverses all of those transactions, "stealing back" the money from the exploiter.
- **Late 2016:** Ethereum decides to "hard fork" two more times to deal with new attacks.
- **Present day:** Graphics cards becoming scarce / price hikes due to Ethereum mining. Uses/platforms built on top of Ethereum proliferating.

Possible research questions - Bo’s picks

- On a blockchain, we can implement more complex “reputation” mechanisms than cash. Should we?
- (Big one in practice) Can we get the incentives right for “proof of stake” or some other alternative to “proof of work”?
- PL: develop sane, formally verified scripting languages for Ethereum (or similar platforms), so that the contract you write is the one you actually want.
- Possibly deep: At what rate should new currency be printed? Why??
  In Bitcoin, can transaction fees replace mining new coins?
- What’s so great about blockchains, and what alternatives could there be?
  Other distributed “consensus” data structures?