



Advanced Network Security

Cryptocurrencies & Blockchains

Amir Mahdi Sadeghzadeh, Ph.D.



Cryptocurrencies & Blockchains

Murtuza Jadliwala

murtuza.jadliwala@utsa.edu



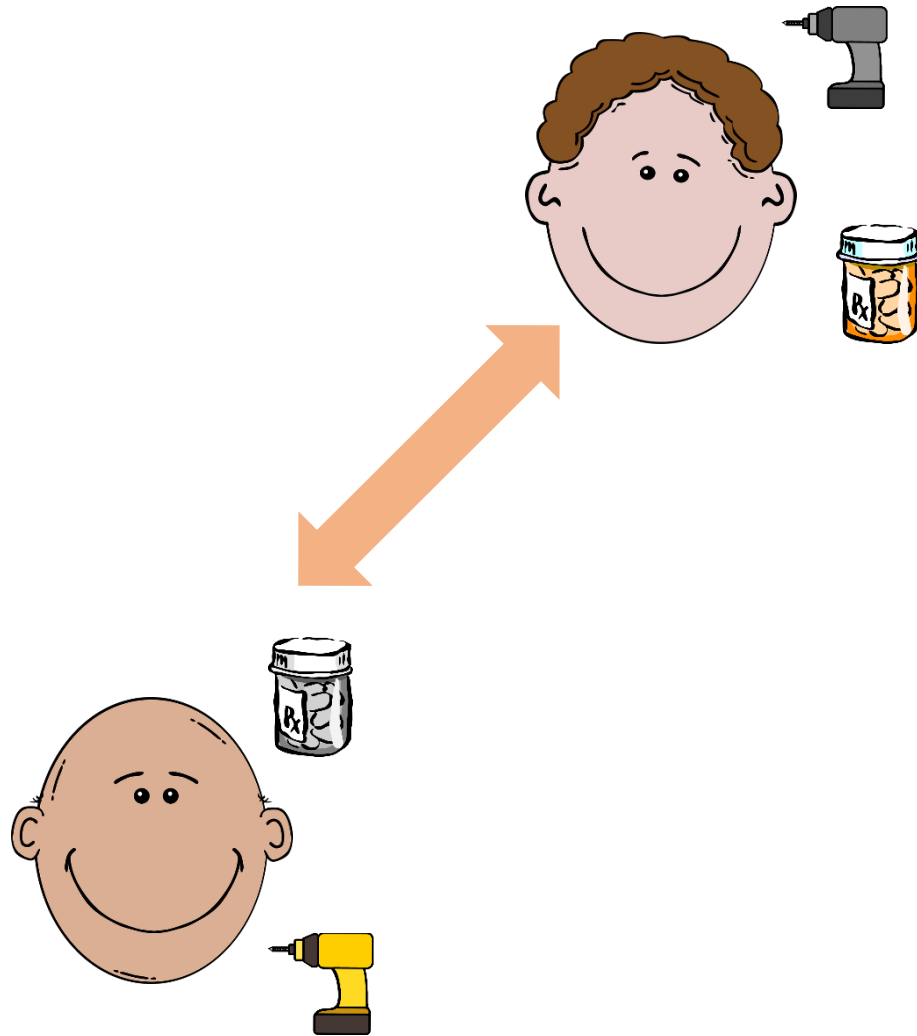
Traditional Currencies

1. Barter

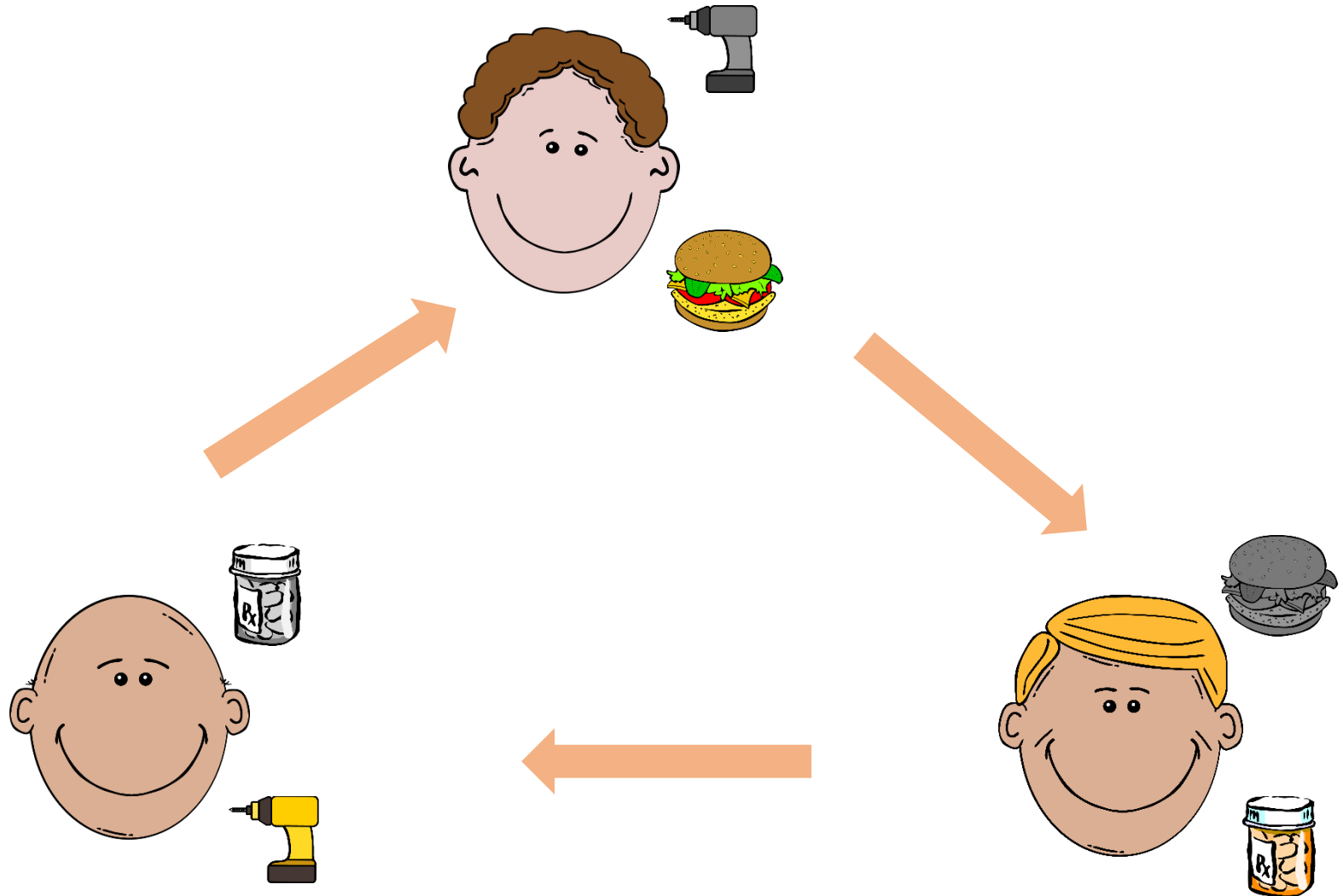
2. Credit

3. Cash

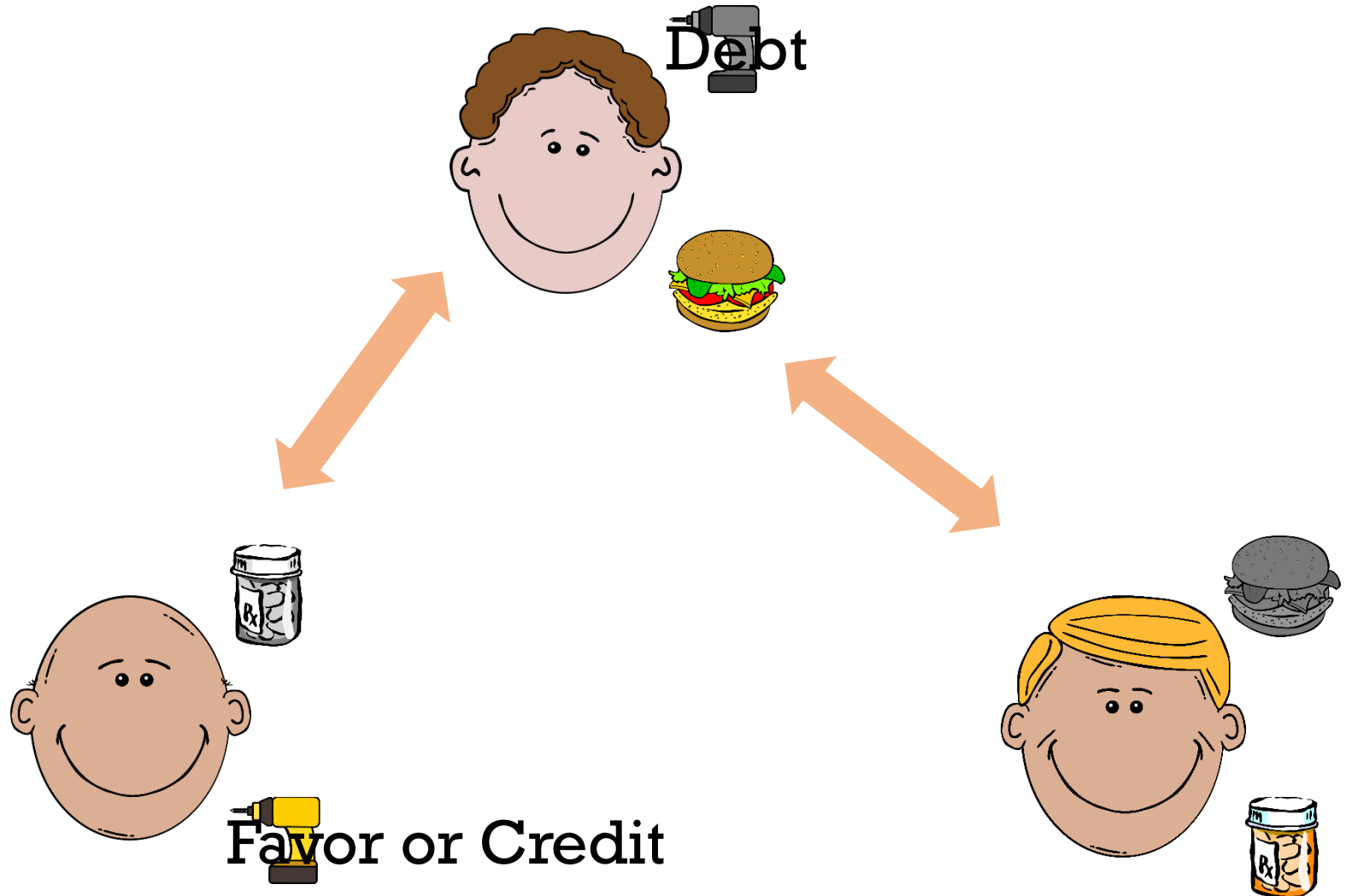
Barter



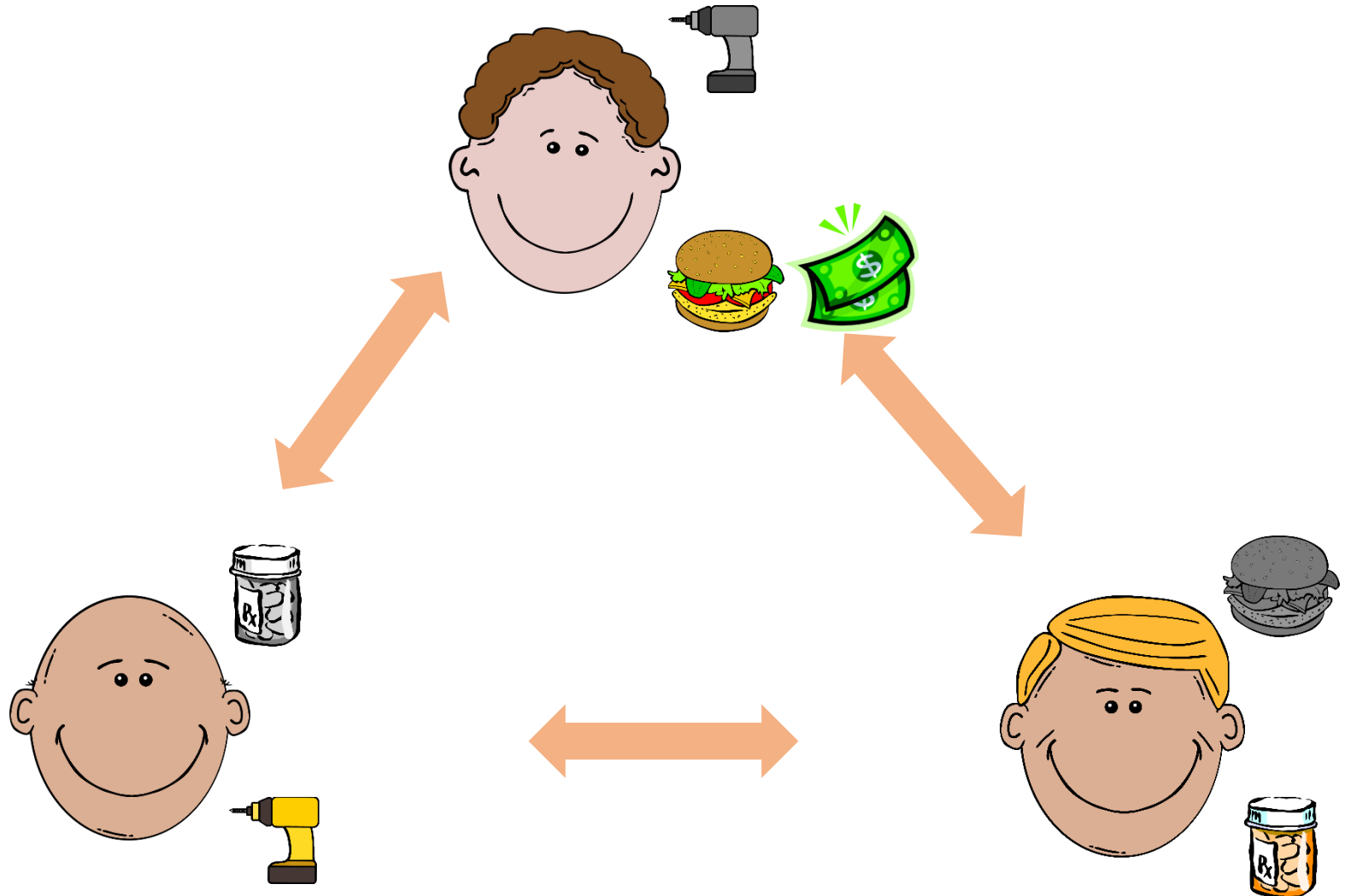
Barter



Credit



Cash



Cash vs Credit

- Cash requires initial allocation, but allows fine-grained valuation of products.
- Credit acquires risk.
- When cash and credit are combined?
 - Cash allow credit to be quantified, for example, how much a person owes another?

Digital Currency

Traditional Financial Tools for the Digital Realm



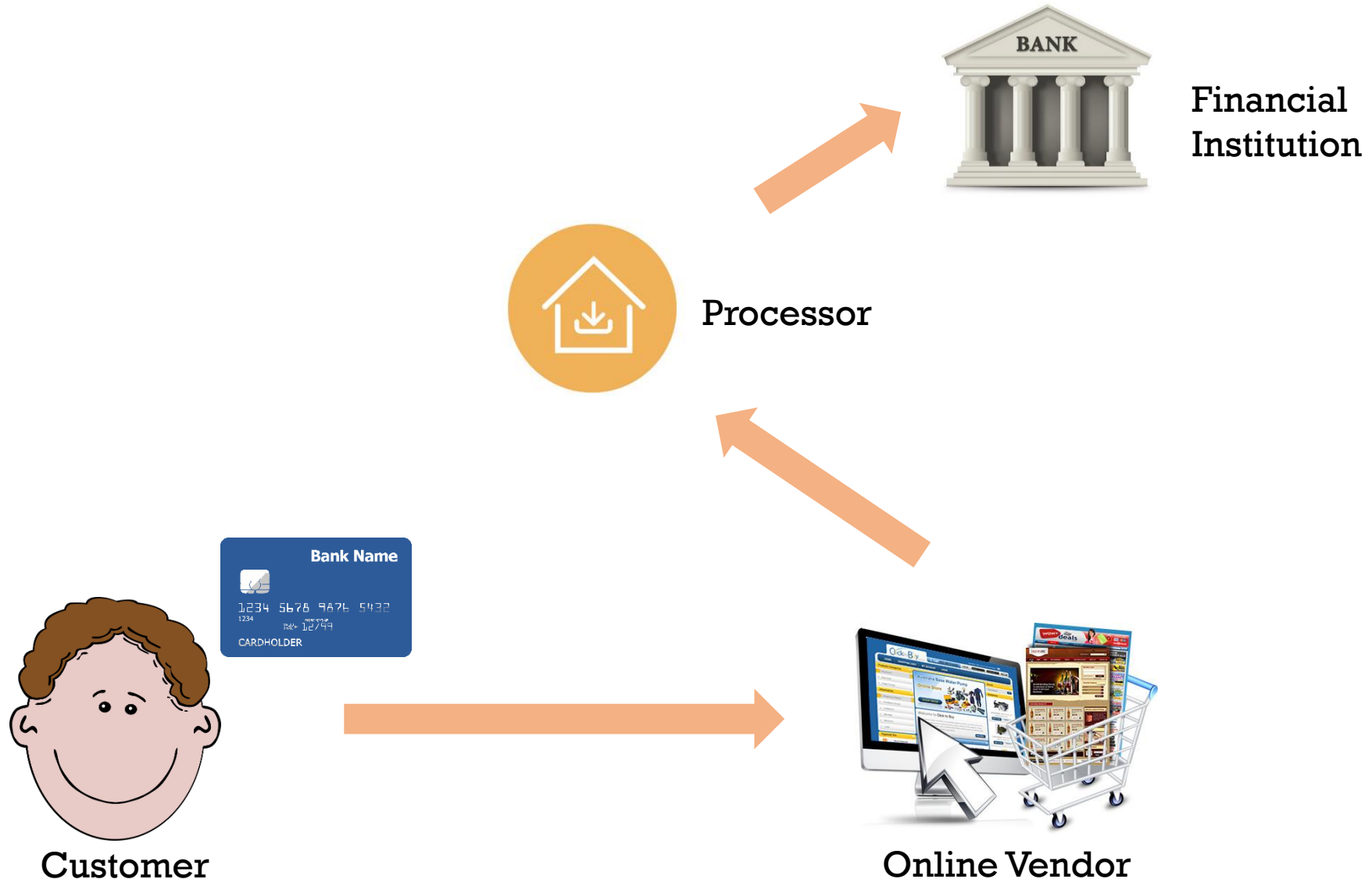
Digital Credit

- First Virtual (1994)
- CyberCash
- iKP (IBM)
- SEPP
- STT (MS & Visa)
- SET (1996)
- Paypal
-

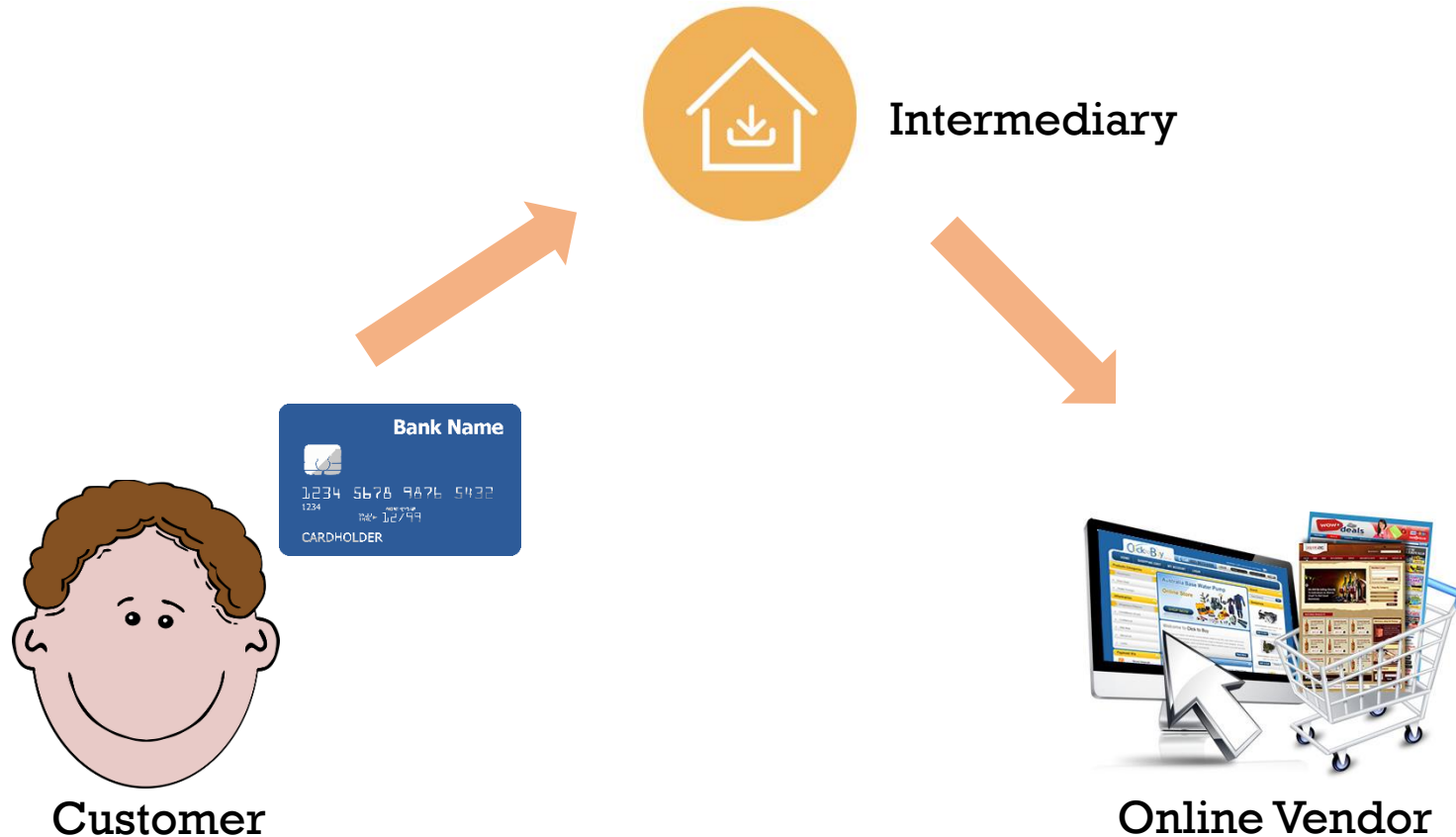
Digital Cash

- Chaum (1983)
- Chaum, Fiat and Noar (1988)
- Digicash (1989-1988)
- MagicMoney
- Lucre
- HINDE
- MONDEX
-

Digital Credit Architecture 1



Digital Credit Architecture 2



Advantages: Hides customer credit card data from online vendors.

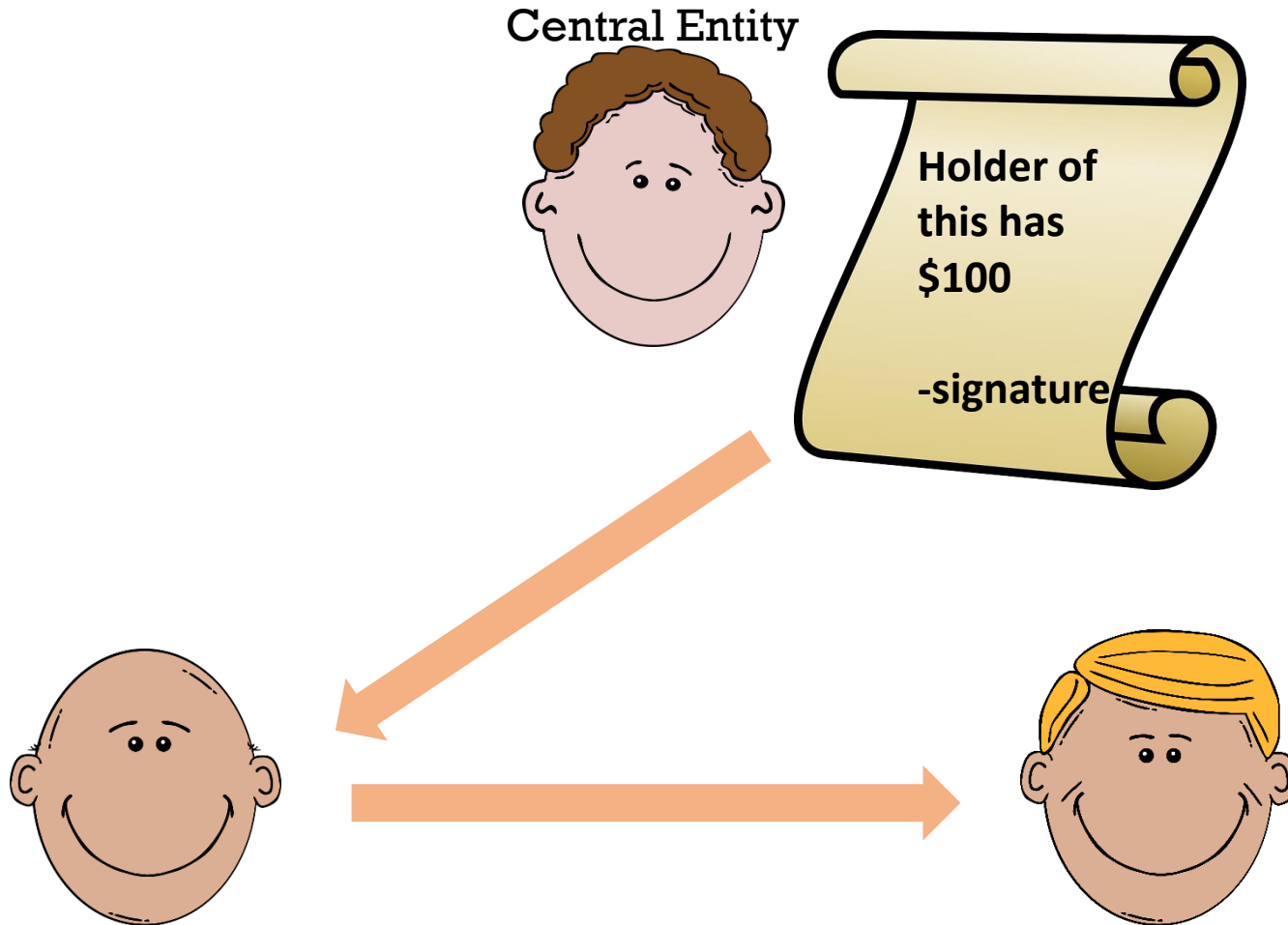
Disadvantages: Requires redirection, enrollment, etc.

Digital Cash

- In parallel, there has been a lot of research in cash-like systems.
- Ideal Requirements:
 - Higher anonymity (similar to traditional fiat cash).
 - Offline transactions.

First Proposal for eCash

- David Chaum (1983)



Problems with Chaum's scheme?

Copying and double spending is easy!

1. First attempt to fix: Introduce *serial numbers*.
 - Shortcoming: **Traceability**!
2. Second attempt to fix: use *Blind Signatures*.
 - Shortcoming: Requires a **centralized entity** that records and maintains all transactions!

Drawbacks

- Drawbacks of current digital currency systems:
 1. Most require a centralized trusted entity.
 2. Some require specialized hardware.
 3. Some require complex/specialized cryptographic techniques.
 4. Others do not provide enough privacy/anonymity.

Motivation

- How can we design a new form of digital currency that
 - does not require a centralized entity, and,
 - does not require a specialized hardware, and,
 - does not require complex cryptography, and,
 - provides decent anonymity?

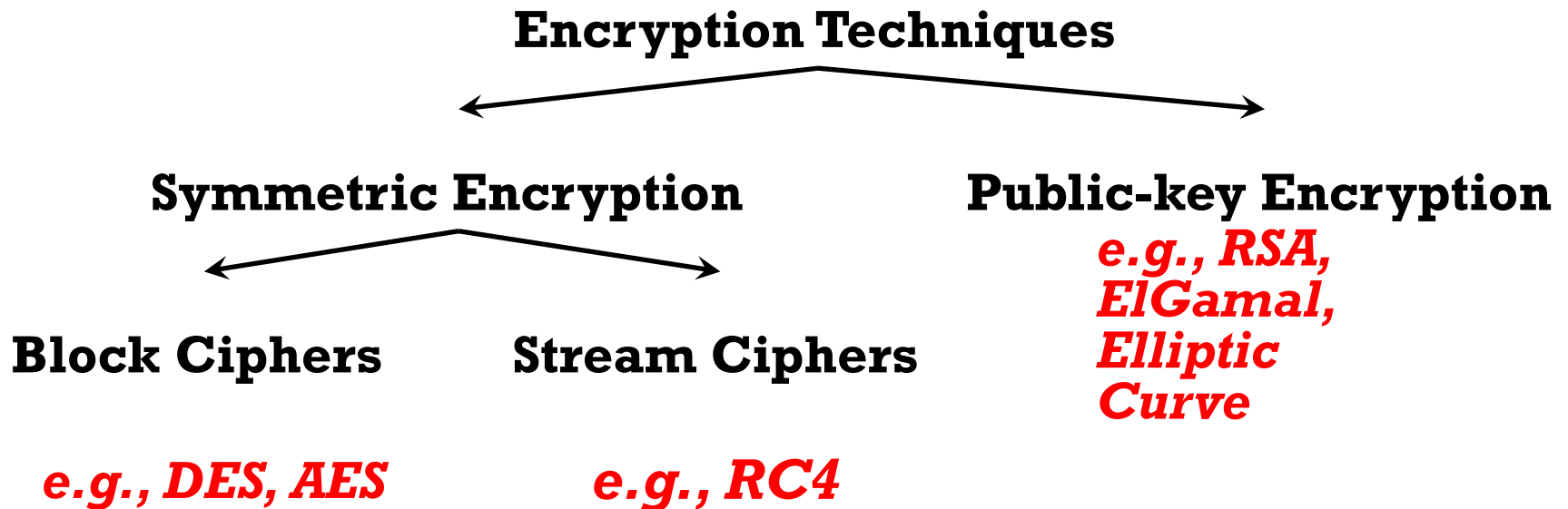
This was the main motivation that led to the development of Bitcoin!

Talk Outline

- Crypto Background
- Bitcoin Details
- What's Next

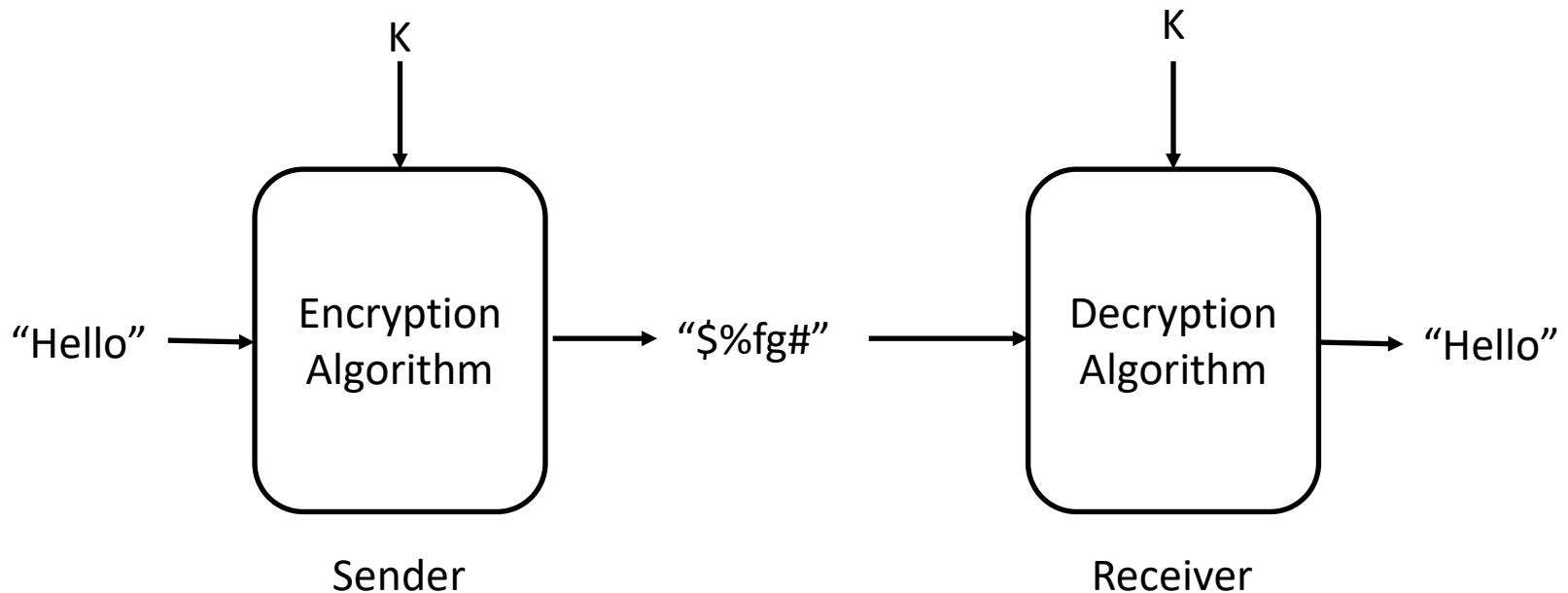
Encryption

Process of transforming information (a.k.a plaintext) into something that is unintelligible (a.k.a ciphertext) to everyone except authorized receivers.



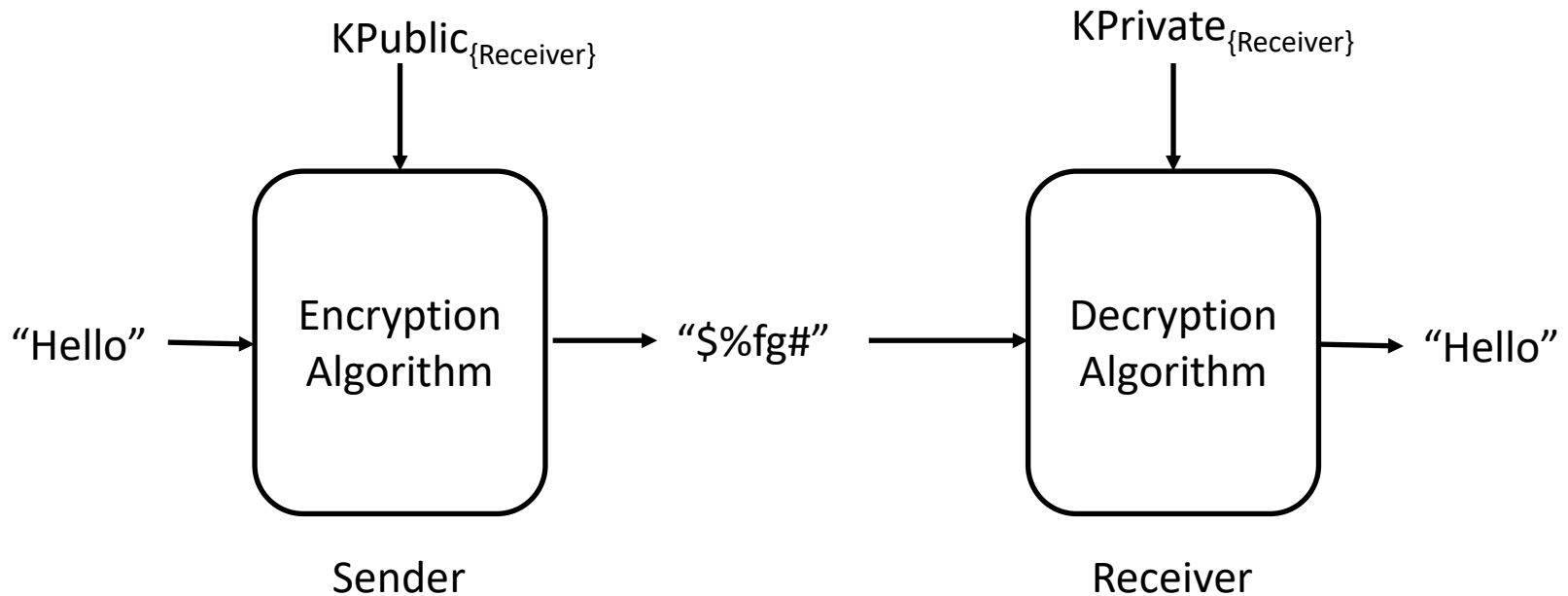
Symmetric Encryption

Algorithm uses the same key for encryption and decryption - also referred to as single-key encryption.

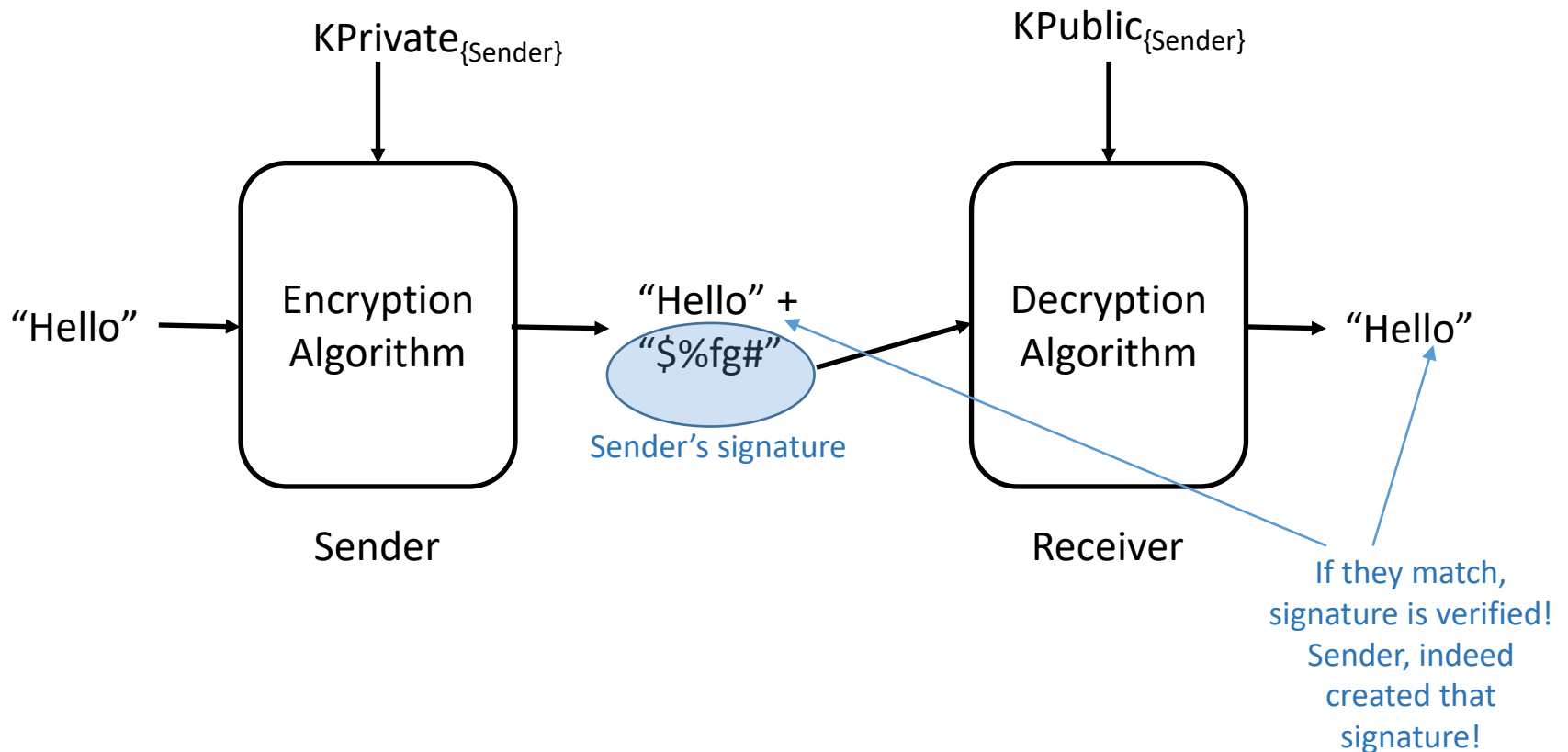


Public-Key Encryption

- **Asymmetric** - uses two separate keys:
 - Public key is made public for others to use.
 - Private key is secret and is never released.



Digital Signatures



Why?

Only sender knows his/her private key → Only sender can create signature, anyone can verify!

Digital Signature Properties

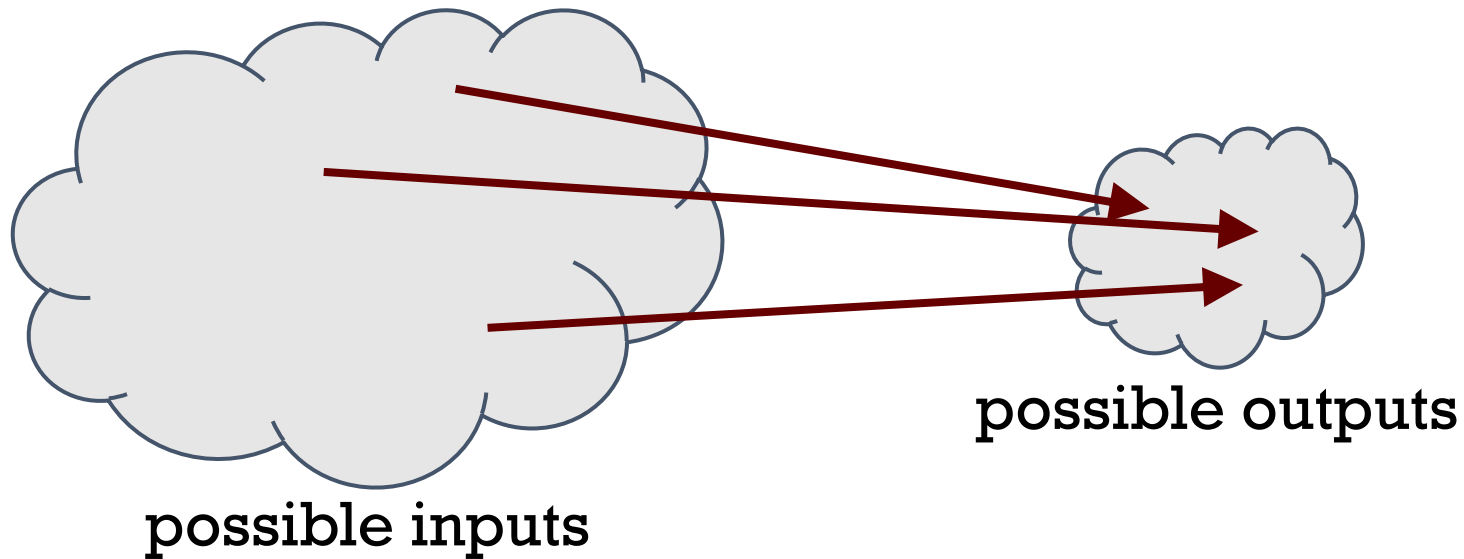
Same as properties we need from handwritten signatures:

1. **Security:** only you can sign as yourself, but anyone can verify that your signature was indeed made by you.
2. **Unforgeability:** signature tied to a particular document - can't be cut-and-pasted to another document.

Key Enabler

- How to create and maintain an append-only, immutable record or ledger of transactions
 - in a *distributed fashion* without requiring any centralized entity, and,
 - without requiring any *specialized hardware*, and,
 - without requiring *complex cryptography*, and,
 - such that it provides *user anonymity*?
- Result: Bitcoin P2P network that maintains transactions on the Blockchain!

Hash Functions



All hash functions satisfy the following properties:

1. Inputs can be any size (not-fixed).
2. Outputs are fixed-size (output size \leq input size).
3. Efficiently computable.

Cryptographic Hash Functions

Satisfy the following additional security properties:

1. **Collision Resistance**: Infeasible to find x and y such that $x \neq y$ and $H(x) = H(y)$
2. **Puzzle-friendliness**: Given a y such that $H(k \parallel x) = y$, and k is random and known, it is infeasible to find x .

Hash Function Applications

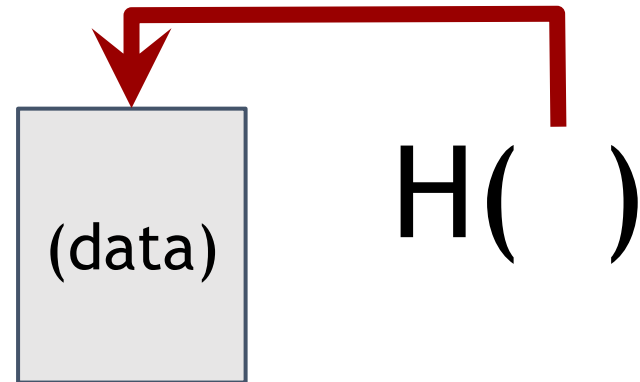
1. **Message digest:** Verify integrity of data (i.e., whether the data under question has changed).
2. **Commitments:** Commit to a value, reveal it later (analogous to sealing something in an envelope)
3. **Search Puzzles:**
Given: A random “puzzle ID” id and a target set Y :
Objective: Try to find a “solution” x such that $H(id \parallel x) \in Y$.

Puzzle-friendly property implies that no solving strategy is much better than trying random values of x .

Hash Pointers

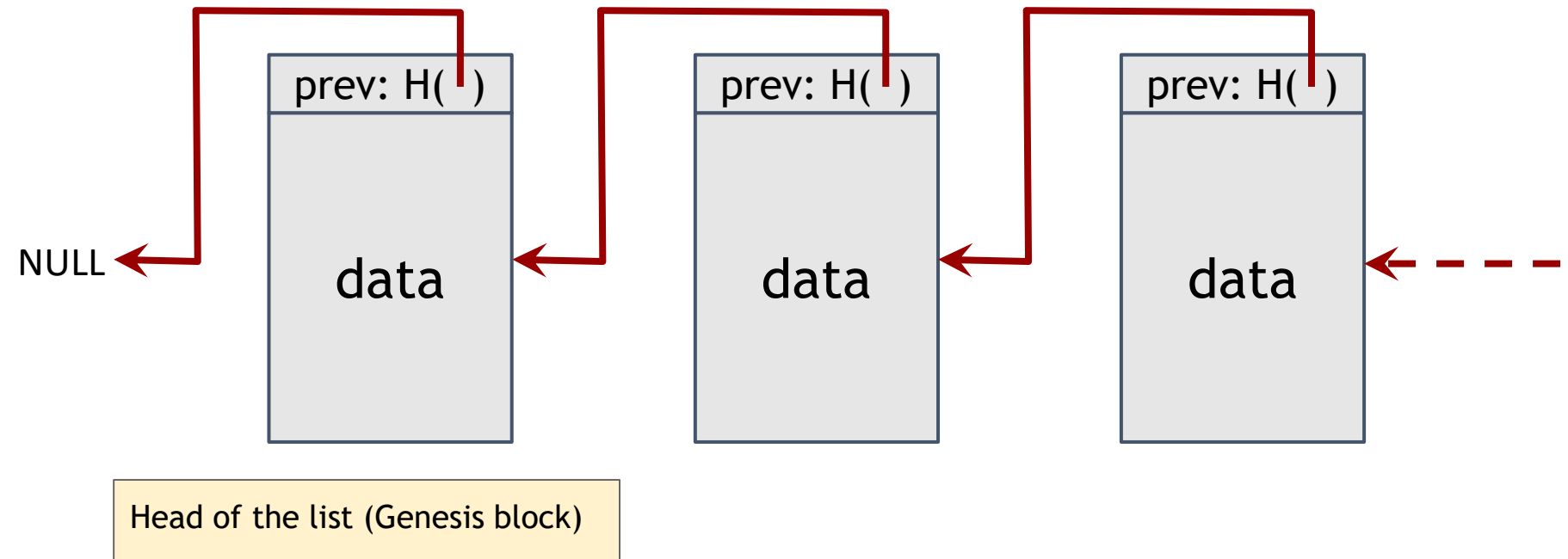
- What is a Hash pointer?
 1. Pointer to where some info/data is stored, and
 2. (Cryptographic) hash of the info.
- What can you do with a hash pointer?
 - Retrieve or get back the info/data.
 - Verify that the info/data hasn't changed.
 - What else?

Use hash pointers to construct data structures such as blockchains!

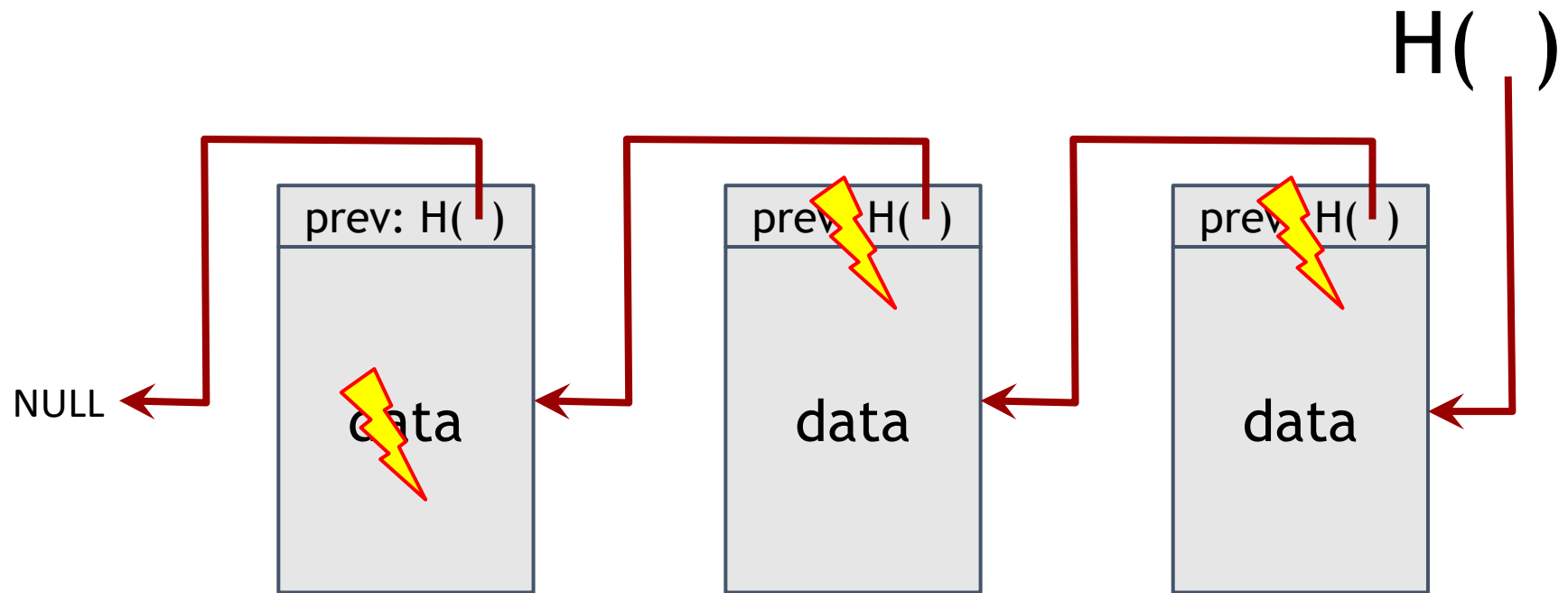


Blockchains

- What is a Blockchain?
 - Linked or ordered list of hash pointers and data blocks.
- What is it used for?
 - Tamper-evident log or register



Tamper-evident Log



Talk Outline

- **Crypto Background**
- Bitcoin Details
- What's Next

What is Bitcoin?

Digital cash or financial instrument:

- Proposed in 2009 by an anonymous author under pen name “**Satoshi Nakamoto**” on the cypherpunks mailing list.
- Is managed in a completely distributed manner.
 - No central authority or government controls Bitcoins.
- Can be (and is) used for online and other transactions and to settle debts.
- Can be (and is) exchanged for other fiat currency.
 - By means of Bitcoin exchanges.
- Can be (and is) traded as other fiat currency.
 - It is what gives Bitcoins its value!

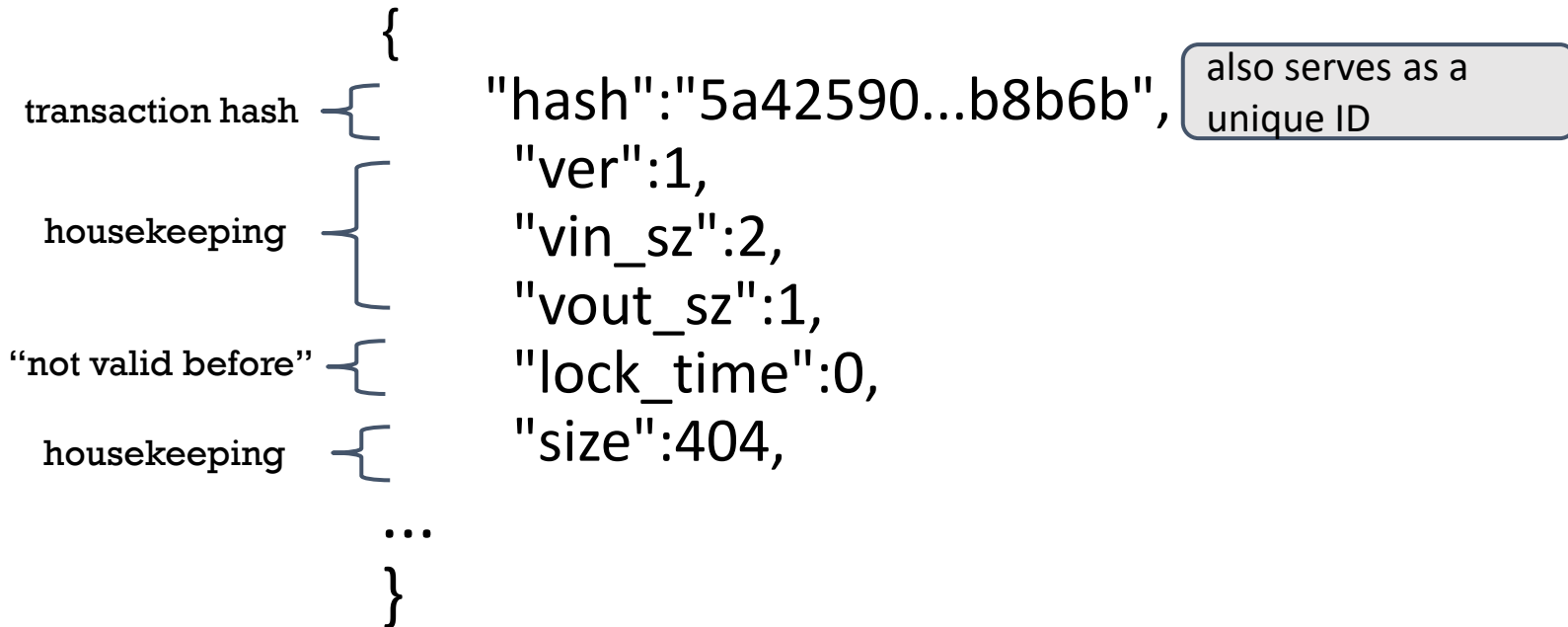
Bitcoin Summary

- A purely distributed system that records and maintains an immutable and consistent ledger (a.k.a. Block chain) of transactions.
- Three Important Aspects of Bitcoins:
 1. Data structures → *what is stored in these ledgers?*
 2. Bitcoin peer-to-peer network → *who maintains these ledgers?*
 3. Consensus → *how is the consistency and immutability of these ledgers maintained?*

A Bitcoin Transaction



Transaction Metadata



Transaction Inputs

```
    "in":[
      {
        "prev_out":{
          "hash":"3be4...80260",
          "n":0
        },
        "scriptSig":"30440....3f3a4ce81"
      },
      ...
    ],
```

previous transaction {

signature {

(more inputs) {

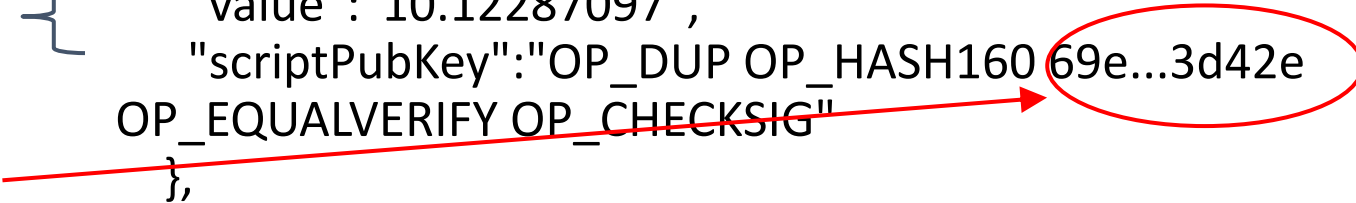
Transaction Outputs

```
"out":[
  {
    "value":"10.12287097",
    "scriptPubKey":"OP_DUP OP_HASH160 69e...3d42e
OP_EQUALVERIFY OP_CHECKSIG"
  },
  ...
]
```

output value {

recipient address?? {

(more outputs) {



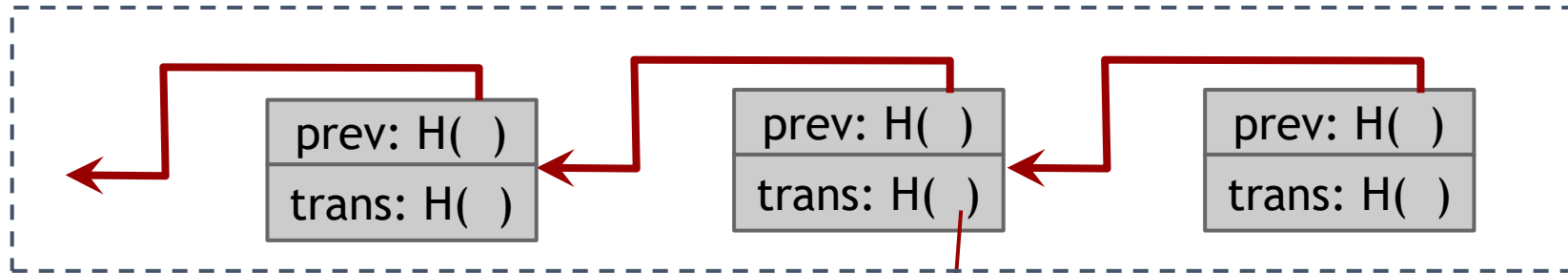
Sum of all output values less than or equal to sum of all input values!
If sum of all output values less than sum of all input values, then difference goes to miner as a transaction fee

Bitcoin Blocks

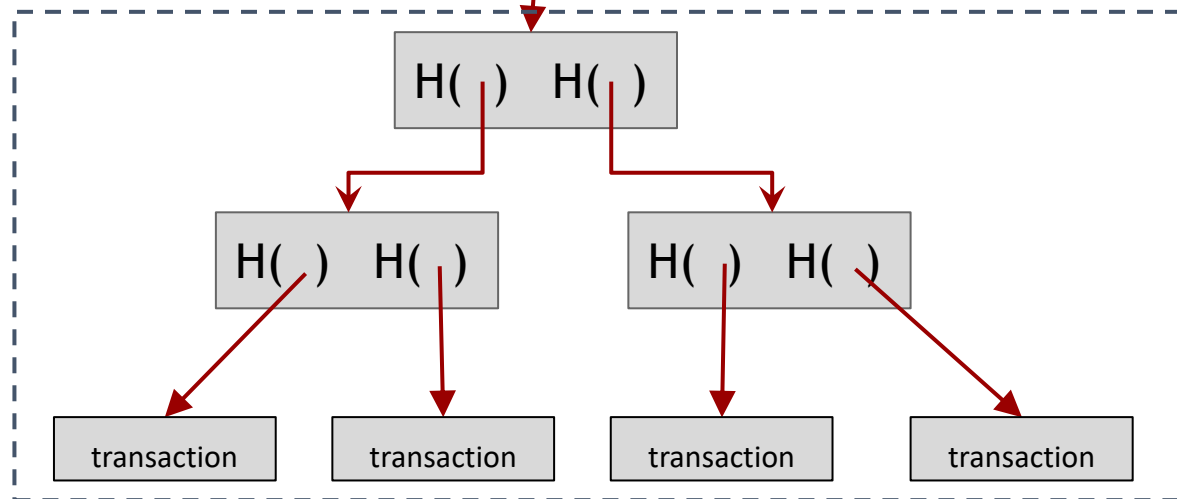
- In a Bitcoin system, multiple transactions are bundled together in blocks.
 - Rather than recording individual transactions into the ledger (or Blockchain), the system records blocks
- Why bundle transactions together?
 - Efficiency!

Bitcoin Block Structure

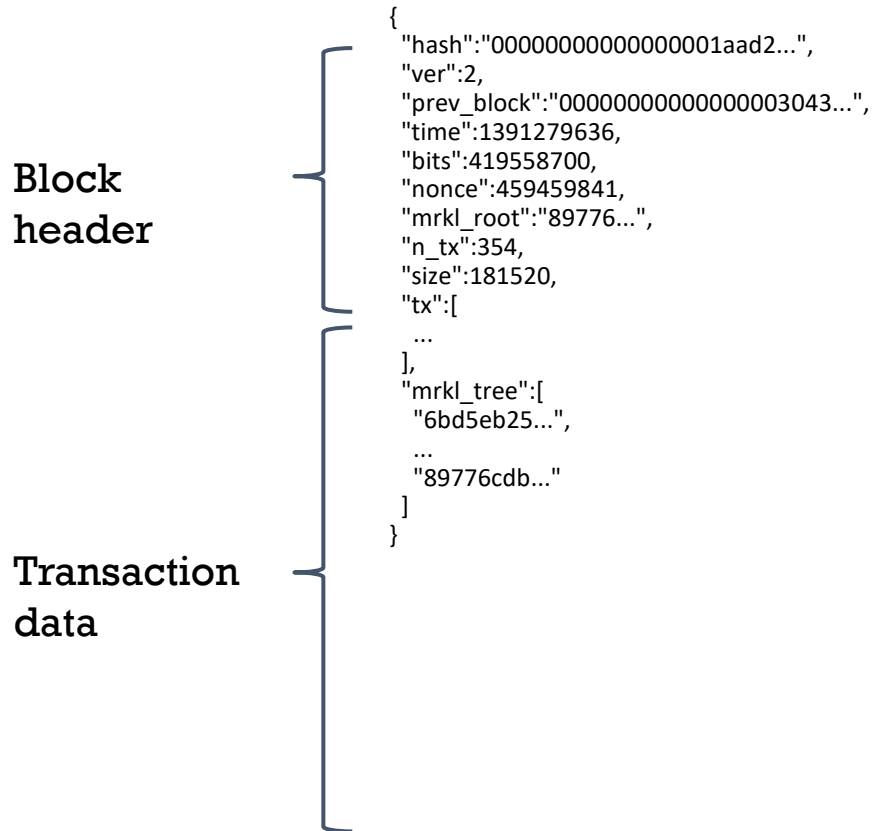
Hash chain of blocks



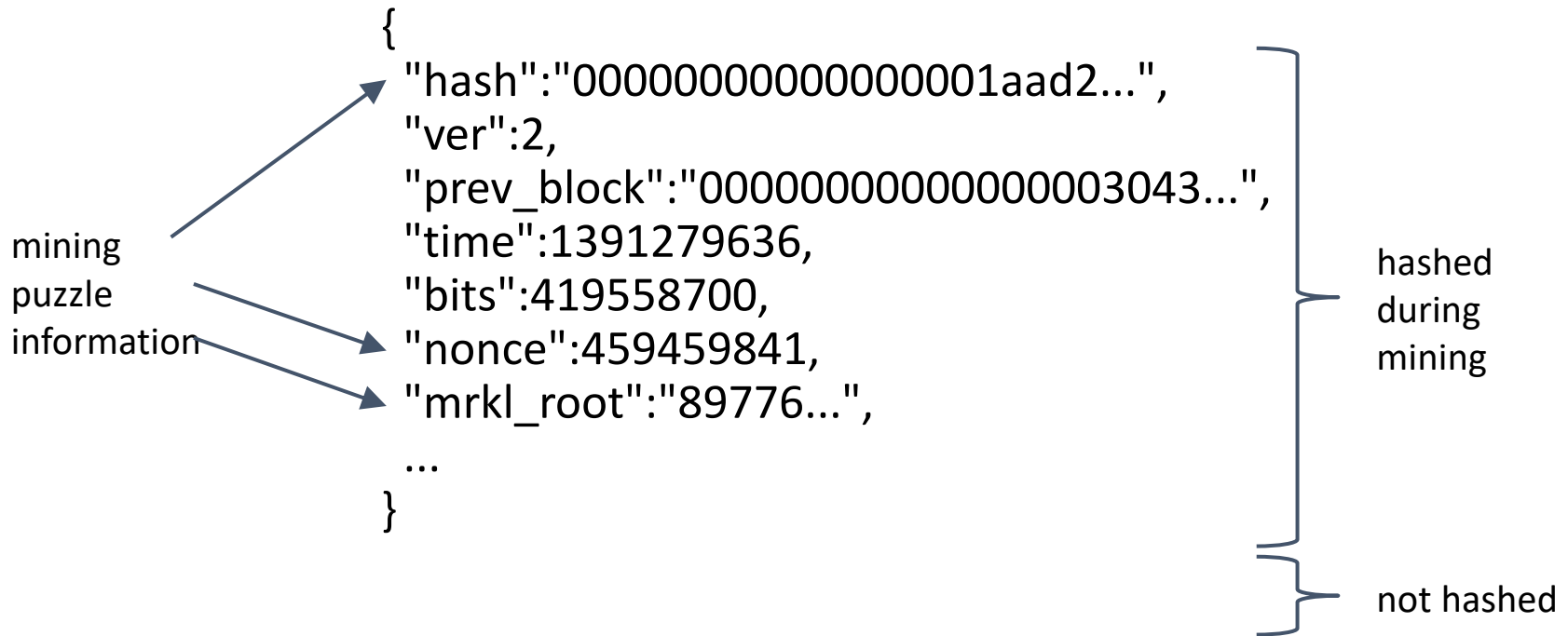
Hash tree (Merkle tree)
of transactions in each
block



A Bitcoin block



A Bitcoin block header



See for yourself!

Transaction View information about a bitcoin transaction

151b750d1f13e76d84e82b34b12688811b23a8e3119a1cba4b4810f9b0ef408d

1KryFUt9tXHvaoCYTNPbqpWPJKQ717YmL5




1KvdrQ3oGqMAiDTMEYCcdDSnVaGNW2YZh
1KryFUt9tXHvaoCYTNPbqpWPJKQ717YmL5

1.0194 BTC
3.458 BTC

9 Confirmations

4.4774 BTC

Summary

Size	257 (bytes)
Received Time	2014-08-05 01:55:25
Included In Blocks	314018 (2014-08-05 02:00:40 +5 minutes)
Confirmations	9 Confirmations
Relayed by IP 	Blockchain.info
Visualize	View Tree Chart

Inputs and Outputs

Total Input	4.4775 BTC
Total Output	4.4774 BTC
Fees	0.0001 BTC
Estimated BTC Transacted	1.0194 BTC
Scripts	Show scripts & coinbase

blockchain.info (and many other sites)

Bitcoin Summary

- A purely distributed system that records and maintains an immutable and consistent ledger (a.k.a Blockchain) of transactions.
- Three Important Aspects of Bitcoins :
 - Data structures.
 - Bitcoin peer-to-peer network.
 - Consensus.

Bitcoin P2P network

- Nodes run a Bitcoin reference (or other) client on TCP port 8333 implementing an ad-hoc communication protocol.
- Nodes typically:
 - Create transactions
 - Forward transactions
 - Validate transactions
 - Add transaction blocks onto the Blockchain

More on this later!
- Ad-hoc network has random topology – no centralized coordinating service or authority
- All nodes are equal – however two types of nodes typically found:
 - Fully validating nodes
 - Thin clients or SPV nodes
- New nodes can join any time - forget non-responding nodes after 3 hours

How big is the Bitcoin network?

- Impossible to measure exactly.
- Estimates - up to 1M new IP addresses/month.
(2015)

Fully-validating nodes

- Permanently connected
- **Store entire block chain**
- Hear and forward every node/transaction
- Only about 5-10k “full nodes”
 - Permanently connected
 - Fully-validate

Thin/SPV clients

SPV – Simplified Payment Verification (e.g., Wallet nodes)

Idea: Don't store everything

- Store block headers – verify the puzzle was solved correctly, but cannot verify every transaction in each block!
- Validate only those transactions that affect them → By requesting transactions as needed
 - To verify incoming payment
 - Trust fully-validating nodes

1000x cost savings! Requires only a few **tens of Megabytes (compare to tens of Gigabytes needed for fully validating nodes)**

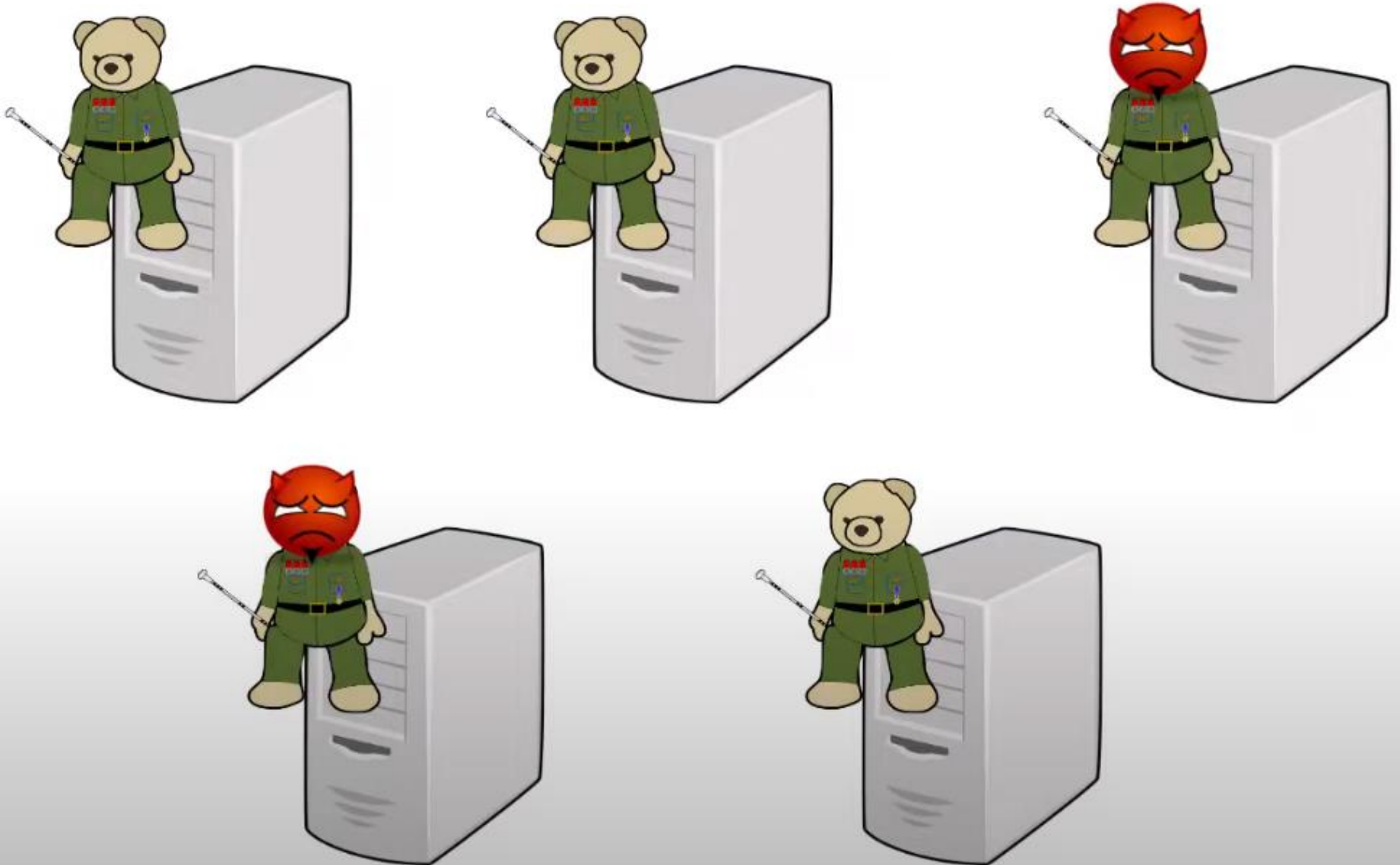
Bitcoin Summary

- A purely distributed system that records and maintains an immutable and consistent ledger (a.k.a Blockchain) of transactions.
- Three Important Aspects of Bitcoins
 - Data structures.
 - Bitcoin peer-to-peer network.
 - Consensus.

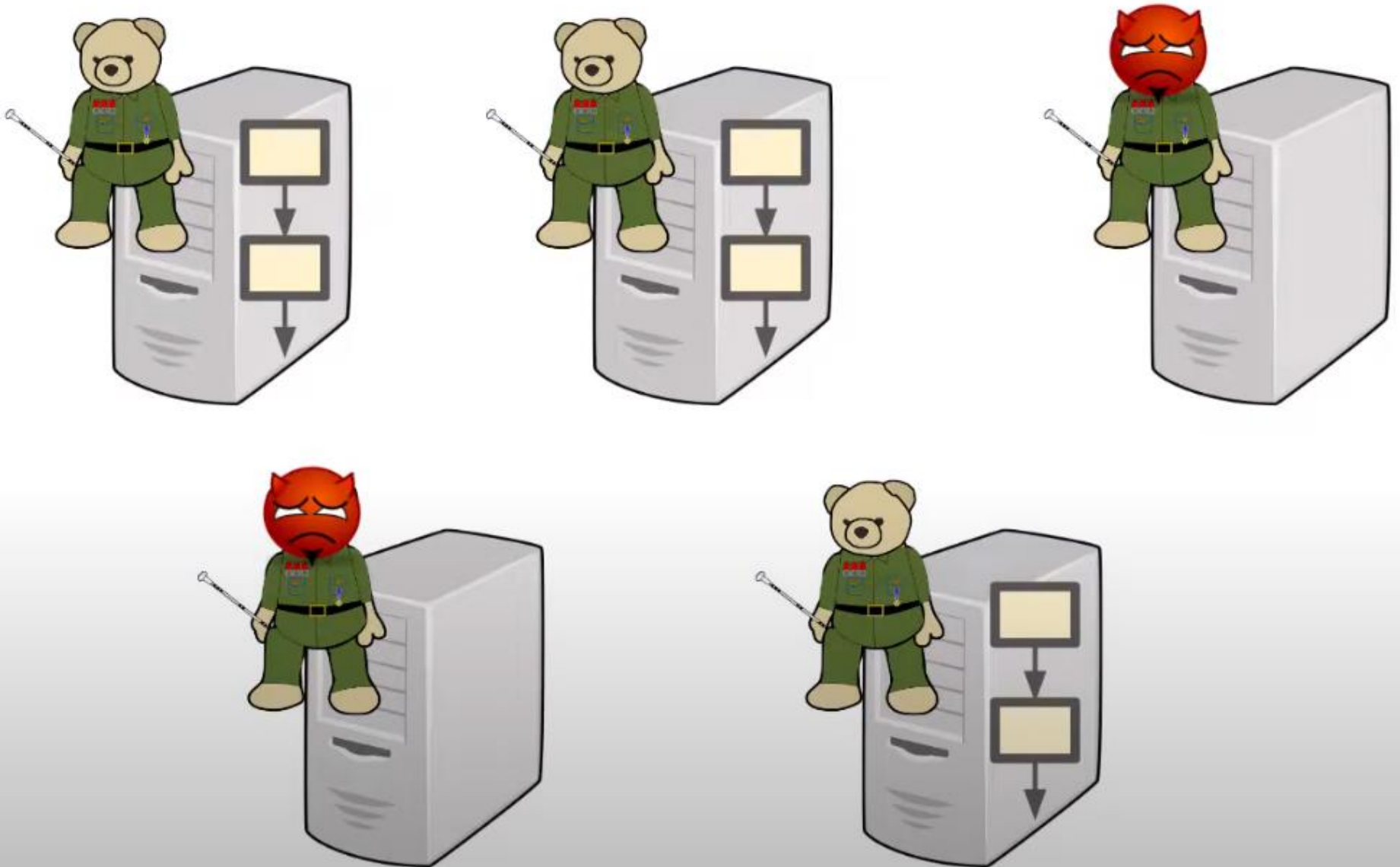
What is Consensus?



What is Consensus?



What is Consensus?



Bitcoin Consensus

- Bitcoin Consensus – most important functionality of the Bitcoin P2P network.
 - What do Bitcoin nodes need to reach a consensus on?
 - Which transactions were broadcast on the network.
 - Order in which these transactions occurred.
 - Transactions are valid (output ≤ input and not double spent).
- Result of the consensus protocol: **Consistent, valid and immutable global transaction ledger.**

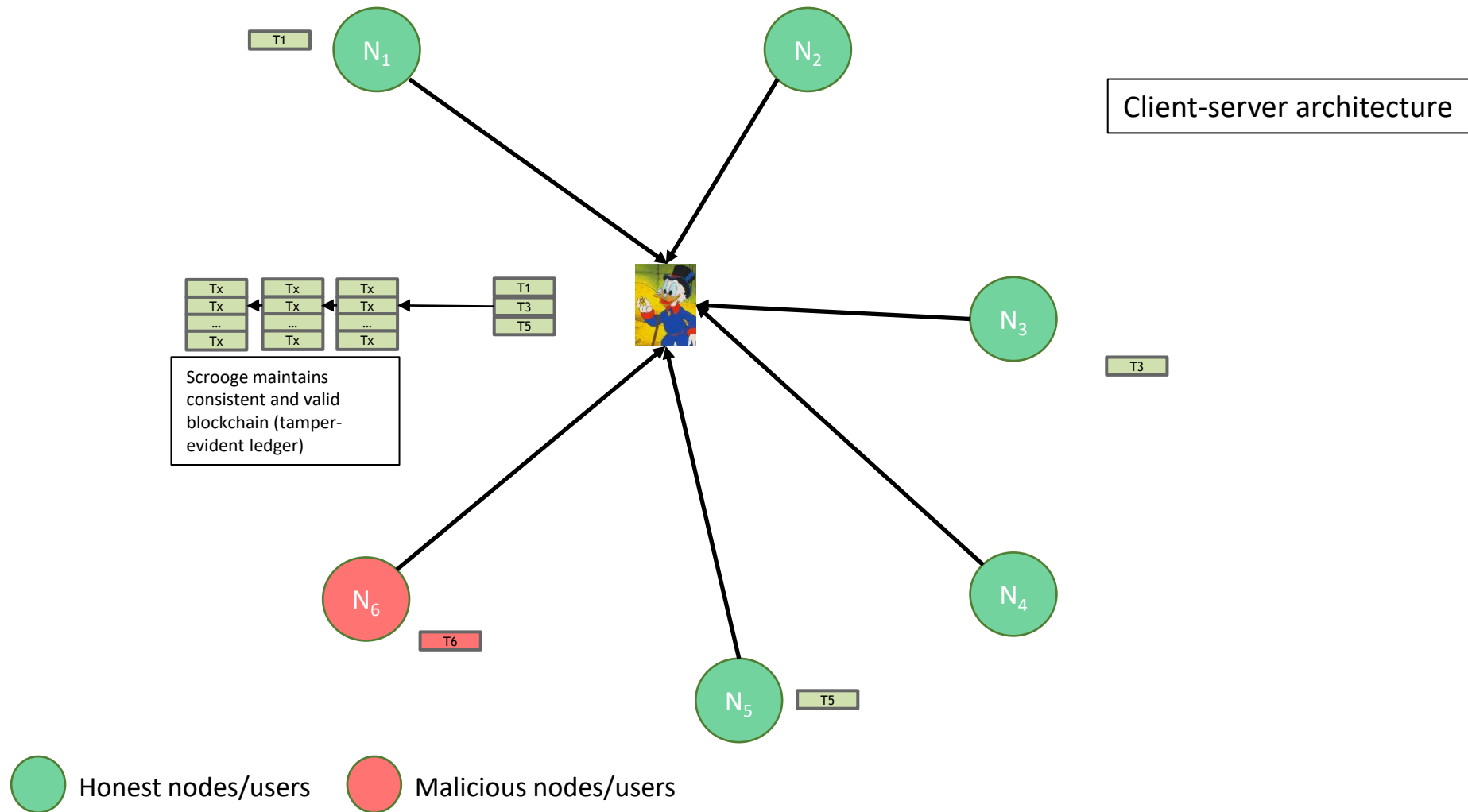
Bitcoin Consensus

- Bitcoin is a **public** blockchain
 - Any machine can join or leave at any time
 - Virtual machines let an attacker create a unlimited number of machines

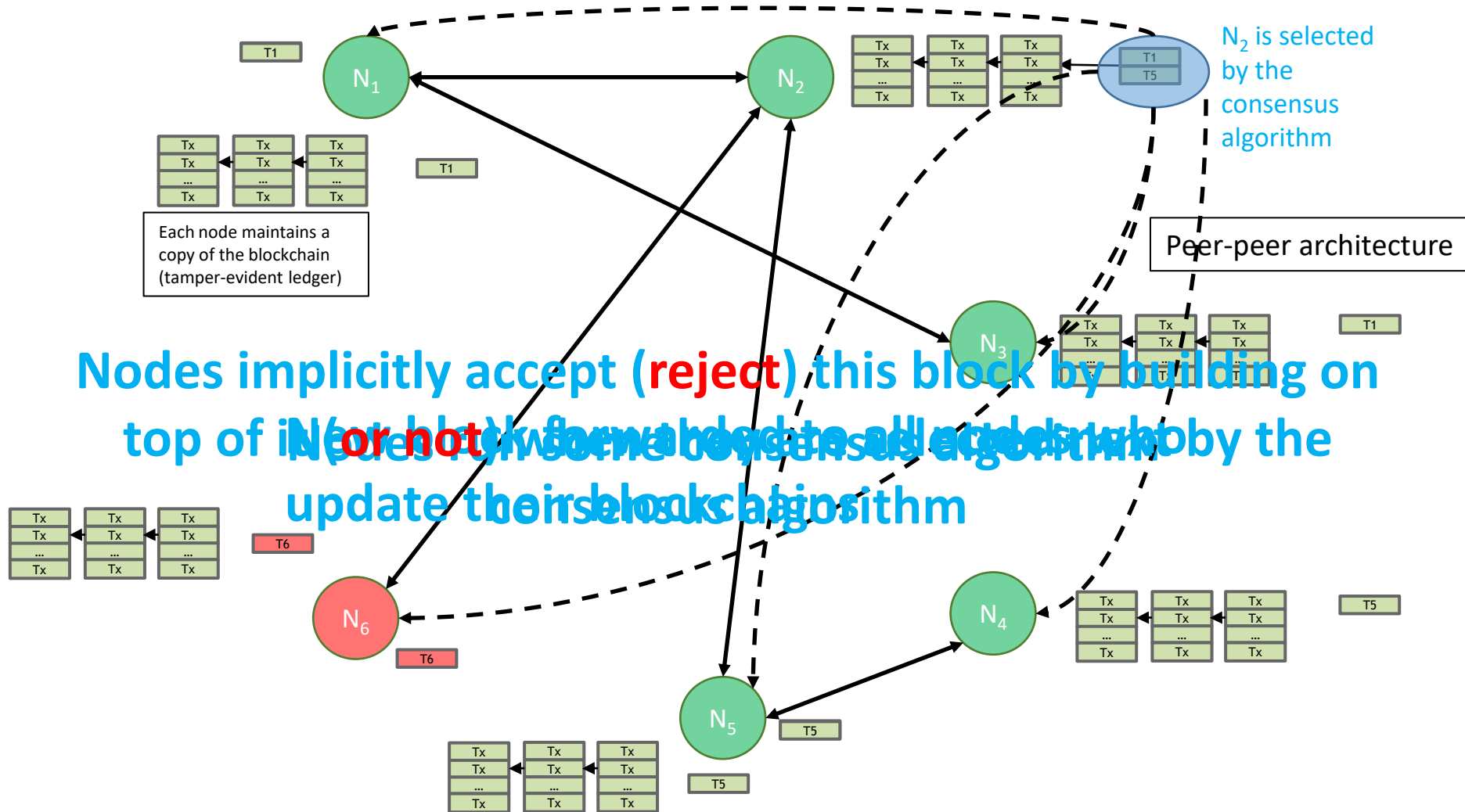
Requirements

- New blocks eventually replicated to all “good” nodes
- Newer blocks point to this new block
- Nodes can join or leave at any time
 - While maintaining consensus
 - Without causing deadlock
- Network partition tolerant
 - Each half forms a consensus
 - When partition heals, so does the consensus

How Centralized Consensus Works



How Bitcoin Consensus Works



How consensus could work in Bitcoin

At any given time (in the bitcoin peer-to-peer network):

- All nodes have a sequence of blocks of transactions (called, ledger or block chain) they've reached consensus on
- Each node has a set of outstanding transactions it's heard about (but not yet included in the block chain)
 - For these transactions consensus has not yet happened
 - Each node may have a slightly different outstanding transaction pool

Consensus algorithm (simplified)

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

Why consensus in Bitcoin is hard?

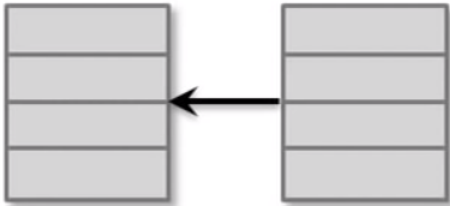
1. Nodes may **crash** or become **offline**.
2. Peer-to-peer **network is imperfect**.
 - Not all pairs of nodes connected (and may participate).
 - Faults in network.
 - Latency.



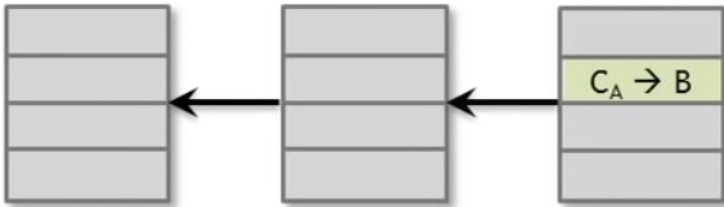
No notion of global time → constraints the set of consensus algorithms that can be used

3. Nodes may be **malicious**.

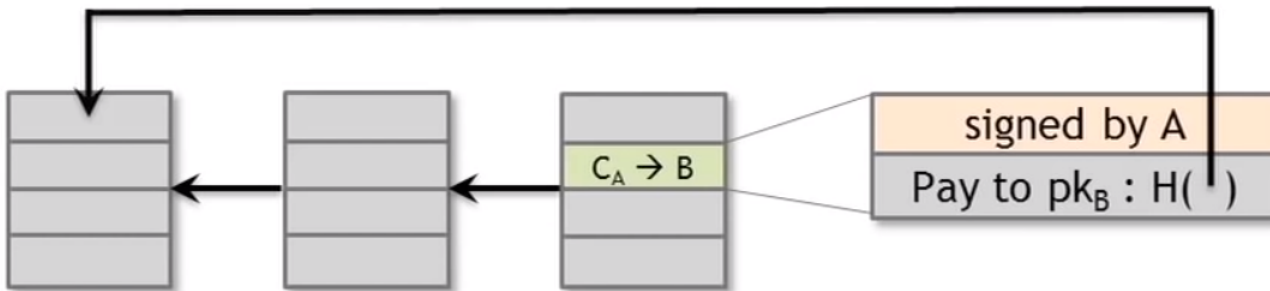
What can a malicious node do?



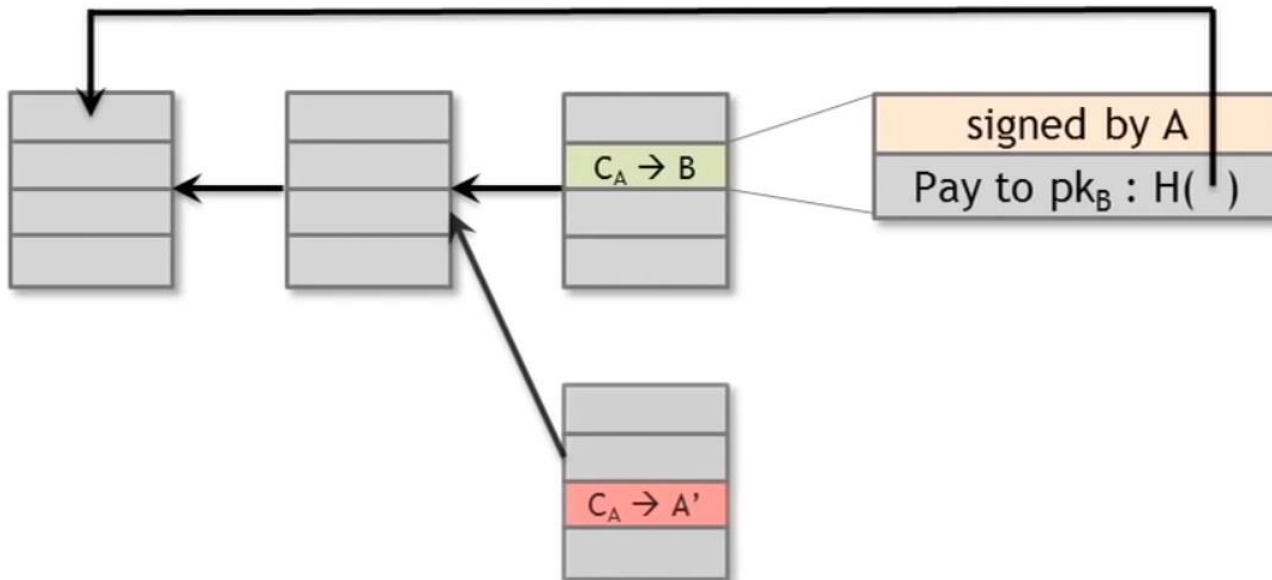
What can a malicious node do?



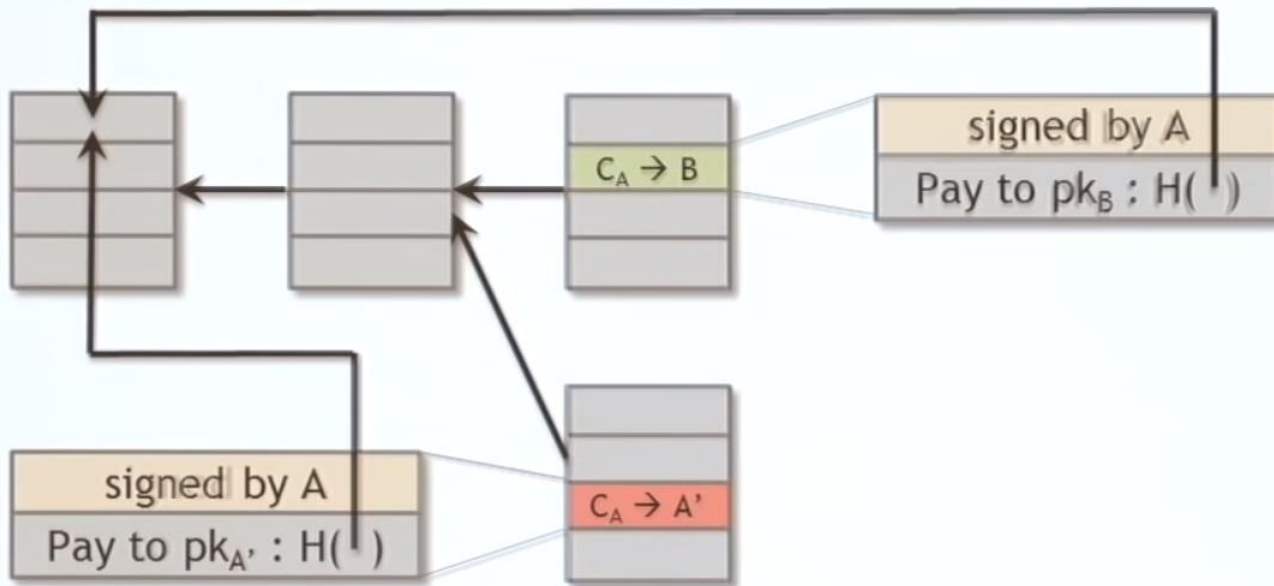
What can a malicious node do?



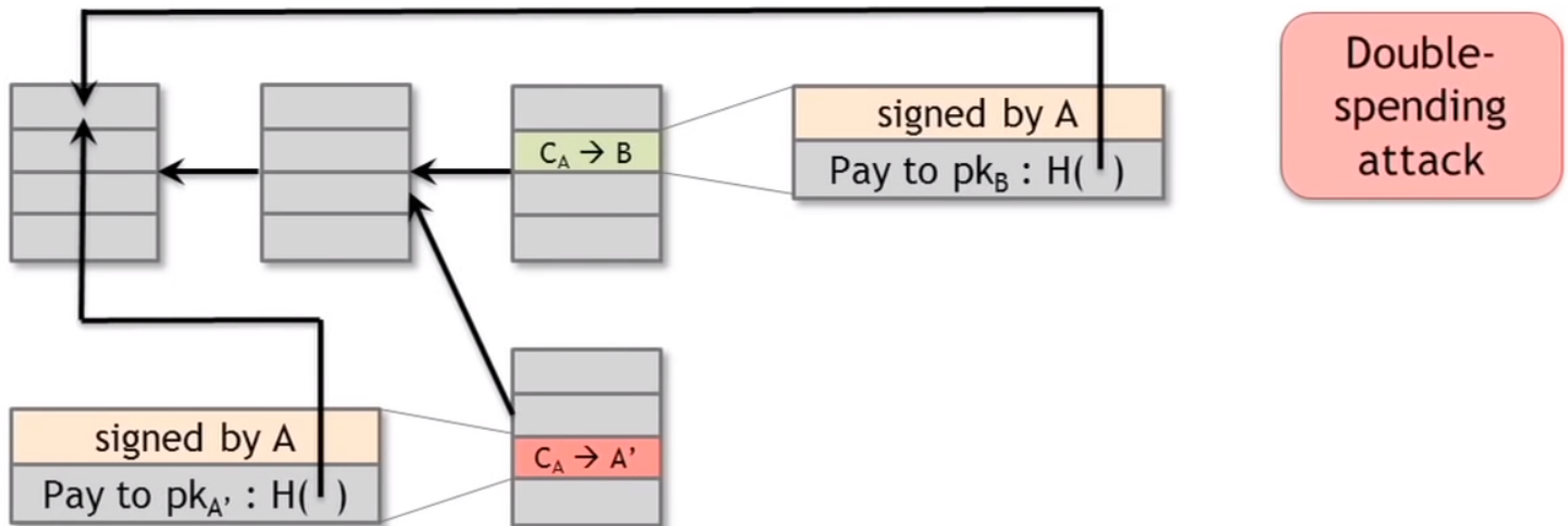
What can a malicious node do?



What can a malicious node do?

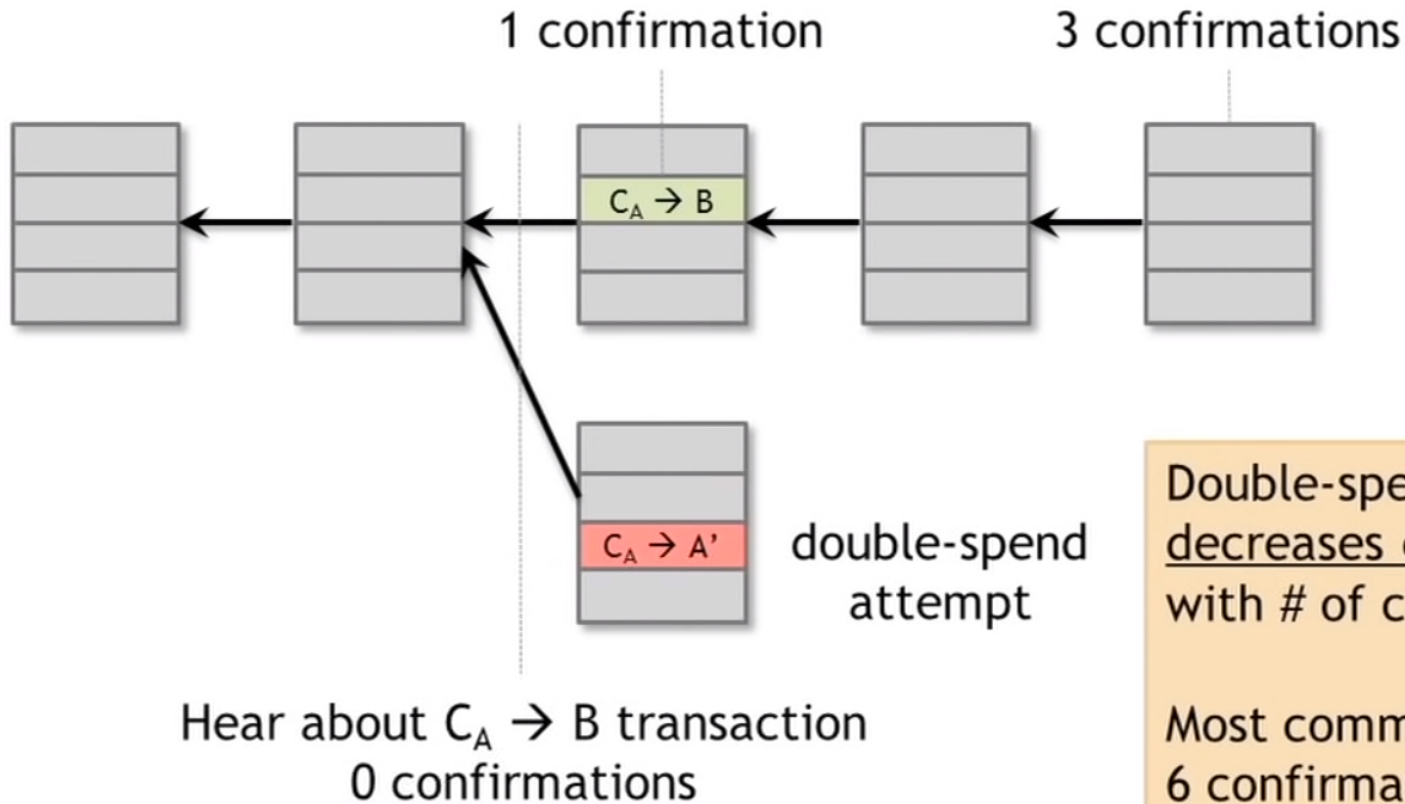


What can a malicious node do?



Honest nodes will extend the longest valid branch

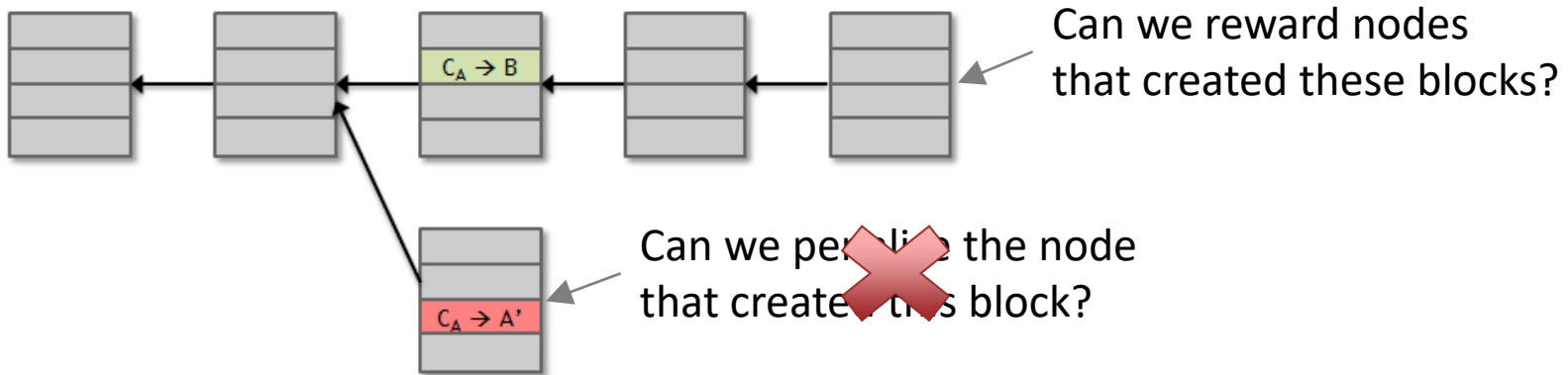
From Bob's point of view



Double-spend probability
decreases exponentially
with # of confirmations

Most common heuristic:
6 confirmations

Assumption of Honesty is Problematic



In other words, can we give nodes incentives for behaving honestly?

✓ We can utilize the fact that Bitcoin (the currency) has value to achieve distributed consensus!

Incentive 1: Block Reward

Creator of block gets to

- include special coin-creation transaction in the block.
- choose recipient address of this transaction.

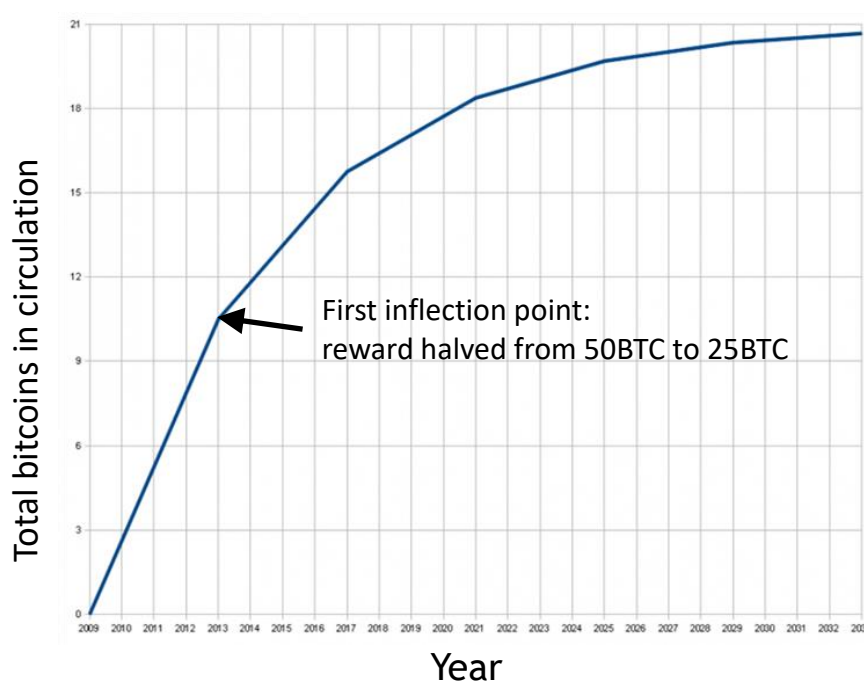
Value is fixed: currently **12.5 BTC**, halves every 210,000 blocks created (or every 4 years at the current rate of block creation).

- We are now in the third period – first period block reward was 50 BTC.
- Reward drops to 6.25 BTC on 24th May 2020, 16:11:39 (est).

Block creator gets to “collect” the reward only if the block ends up on long-term consensus branch!

- Subtle but powerful trick: Incentivizes nodes to behave in way that will get other nodes to extend their block.

There's a finite supply of bitcoins



Block reward is how
new bitcoins are created

Runs out in 2040. No new
bitcoins unless rules change

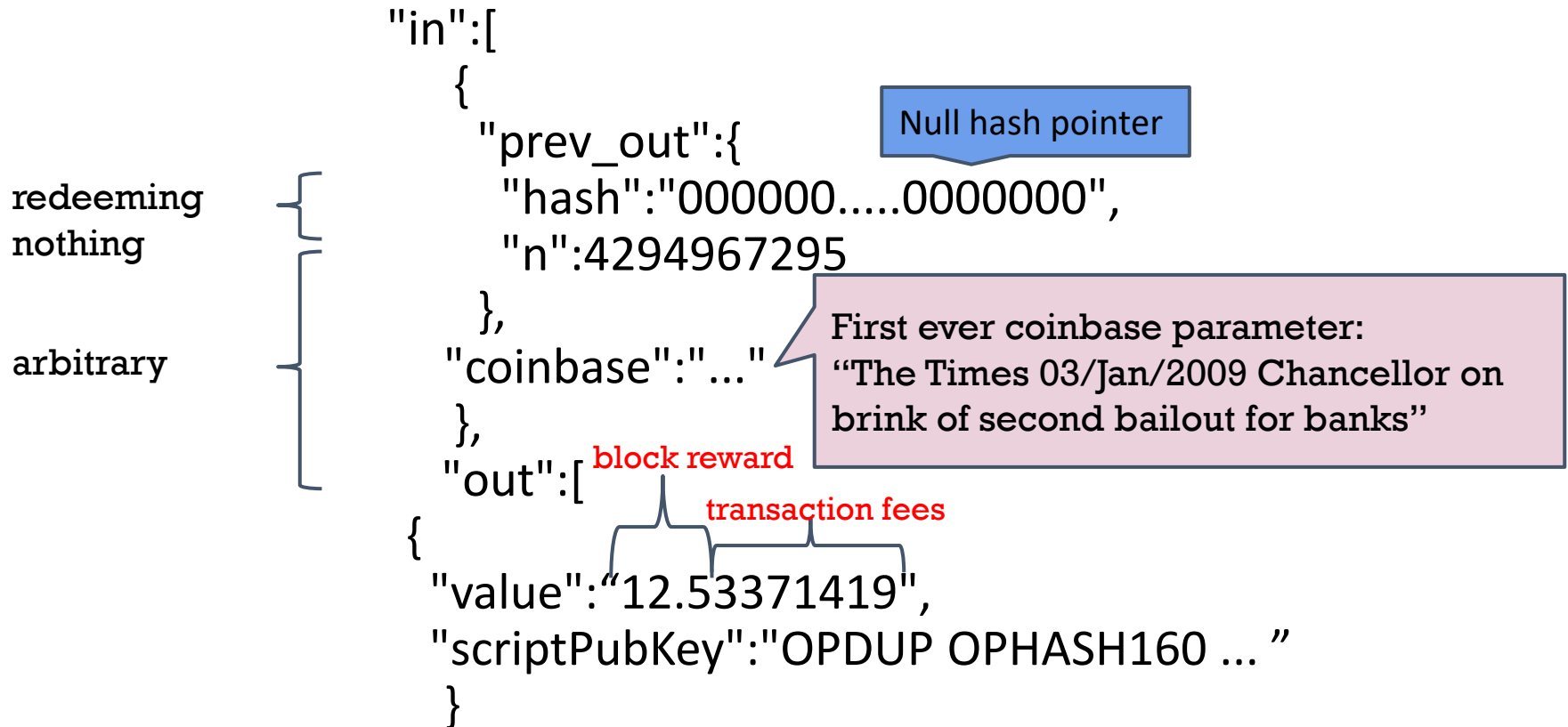
**Does that mean that after 2040,
nodes will no longer have
incentive to behave honestly?**

Not really!

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 block creator.
- Purely **voluntary**, like a tip.
 - But system will evolve, and will become mandatory, as Block rewards run out.

A Coinbase Transaction



Remaining Problems

1. How to pick a **random** node?
2. How to avoid a **free-for-all** system due to rewards?
 - Everybody may want to run a bitcoin node in order to get free rewards.
3. How to prevent **Sybil** attacks?
 - An adversary may create a large number of Sybil nodes to subvert the consensus process.

Solution: Mining using Proof-of-Work (PoW).

Proof of Work (PoW)

To approximate selecting a random node: *select nodes in **proportion** to a resource that no one can monopolize (we hope):*

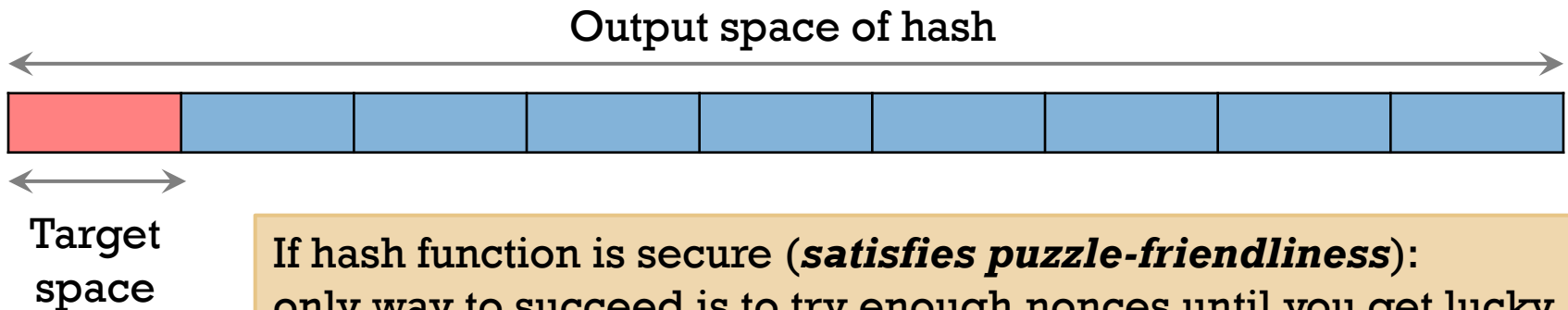
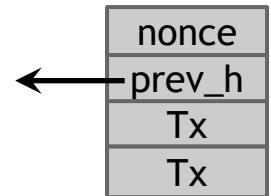
- In proportion to computing power: **proof-of-work** (*Used in Bitcoins*).
- In proportion to ownership of the currency: **proof-of-stake** (*Not used in Bitcoins – but a legitimate model used in other cryptocurrencies*).

Hash Puzzles

To create block, find nonce s.t.

$H(\text{nonce} \parallel \text{prev_hash} \parallel \text{tx} \parallel \dots \parallel \text{tx})$ is very small.

In other words, $H(\text{nonce} \parallel \text{prev_hash} \parallel \text{tx} \parallel \dots \parallel \text{tx}) < \text{target}$.



Mining Bitcoins in 6 Easy Steps

1. Join the network, listen for transactions.
 - a. Validate all proposed transactions.
2. Listen for new blocks, maintain blockchain.
 - a. When a new block is proposed, validate it.
3. Assemble a new valid block.
4. Find the nonce to make your block valid.
5. Hope everybody accepts your new block.
6. Profit!

Useful to
Bitcoin
network

Incentivize
miners to
do above

Advantage of a PoW System?

- It completely does away with the problem of magically picking a random node (to propose a block).
- Nodes independently compete by attempting to solve hash puzzles.
 - Once in a while, one (randomly) will succeed and propose the next block.
 - Result: Completely decentralized system → No one gets to decide which node proposes the next block.
- Other advantages:
 - Not a free-for-all system → Nodes need to work to get paid.
 - Creating new (Sybil) identities is useless without creating new computing power (to solve PoW) to go along with it!

Evolution of Mining

2009  2018



CPU



GPU



FPGA



ASIC



Gold pan



Sluice box



Placer mining



Pit mining

How to Transact in Bitcoin?

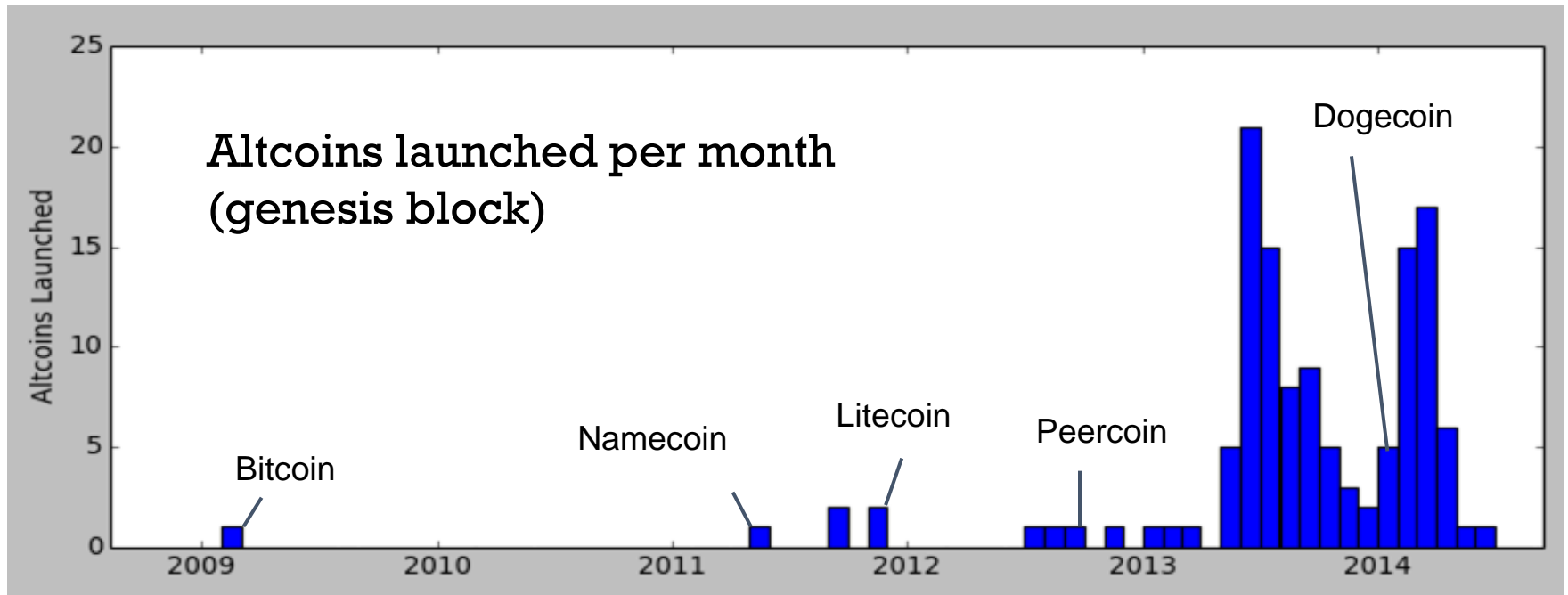
To spend a Bitcoin, you need to know:

- Some info from the **public blockchain**,
and
- The owner's **secret signing key**

So it's all about key management.

Bitcoin is Not Alone!

As of 2015, 50-500 altcoins launched!



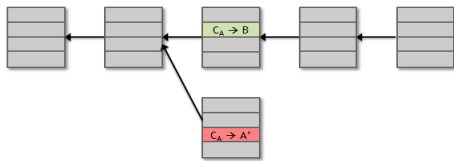
Data from mapofcoins.com

Reasons for Altcoins

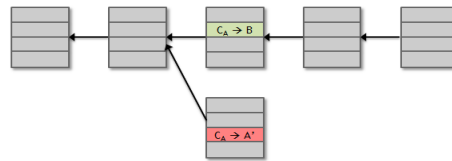
- Better (or different) security.
 - Mining puzzle.
- Contract/platform features.
- Different parameters and monetary policy.
 - Inflation.
 - Inter block time.
- Community or common interest support.

Bitcoin's Blockchain Platform

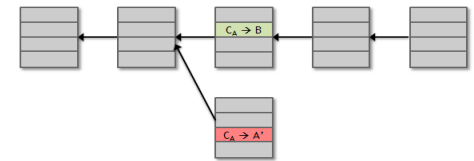
- Works only for Bitcoin!
- **How to implement a distributed application with a slightly different logic/requirement?**
 - Create a new blockchain to support the application!
- Result:



Bitcoin



Litecoin



Dogecoin

Question: Can we build a single blockchain that supports multiple distributed applications?

Smart Contract Model in Ethereum

- Notion of accounts:
 1. Externally Owned Accounts (governed by users).
 2. Contract Accounts (governed by contracts or code).
- Smart Contract: A program that lives on the Blockchain (forever).
 - Written in Solidity – a high-level programming language used by Ethereum.
- **Anyone** can create a contract and upload it.
 - Pay a small fee, done by means of a special “transaction”.
- **Users** send “specially crafted” transactions to execute these contracts.
- **Miners** agree on order of transactions and actually execute contracts using a PoW-based consensus.
- Ethereum clients run a special virtual machine, called EVM, which executes the smart contracts.

Two types of Blockchain Applications

- **Public Blockchains:**

- Blockchain participants are not authenticated.
- Anyone can participate (also referred to as permission-less blockchains).
- Example: Bitcoin, Ethereum.

- **Private Blockchains:**

- Blockchain participants are authenticated.
- Only authenticated nodes can participate (also referred to as permissioned blockchains).
- Example: Hyperledger framework by IBM.

Moving Forward – Main Innovations

- **Incentives**: Why should users participate?
- **Scalability**: How to increase number of transactions per second?
- **Space**: How to reduce the size of the Blockchain?
- **Applications in other domains/businesses**: For authentication, integrity, record-keeping, etc.
 - IoT
 - Supply Chain
 - Financial Services
 - Spectrum Management
 -