Blockchains & Cryptocurrencies

Towards Consensus & Bitcoin

Instructor: Abhishek Jain
Johns Hopkins University - Spring 2021

*Many slides based on NBFMG

Today

- We're going to look at "warmup" cryptocurrencies
- We will then start talking about "consensus": what it is, why is it important?
- This is all in preparation for next time, when we'll actually talk about Bitcoin

Review: digital currency problems

- **Double spending**
  - To capture double spending, it seems that we need an online (networked) party that must be trusted
- **Authentication / Authentication**
  - How do I prove that I am the owner of currency & thus authorized to transact with it?
- **Origin/Issuance**
  - How is new currency created?

"Warmup" approach

- Let's start with a centralized approach:
  - There will be a central trusted party that maintains a *public ledger*
  - This centralized party can also create ("mint") new currency and assign it to be owned by users
    - However, authentication/ownership problem will be solved using digital signatures
- Later, we will see how the "role" of the central party can be implemented using a distributed protocol

GoofyCoin

Goofy can create new coins

signed by pk_{Goofy}

CreateCoin [uniqueCoinID]

A coin's owner can spend it.

signed by pk_{Goofy}
Pay to pk_{Alice} : H( )

signed by pk_{Goofy}
CreateCoin [uniqueCoinID]

The recipient can pass on the coin again.

signed by pk_{Alice}

Pay to pk_{Bob} : H( )

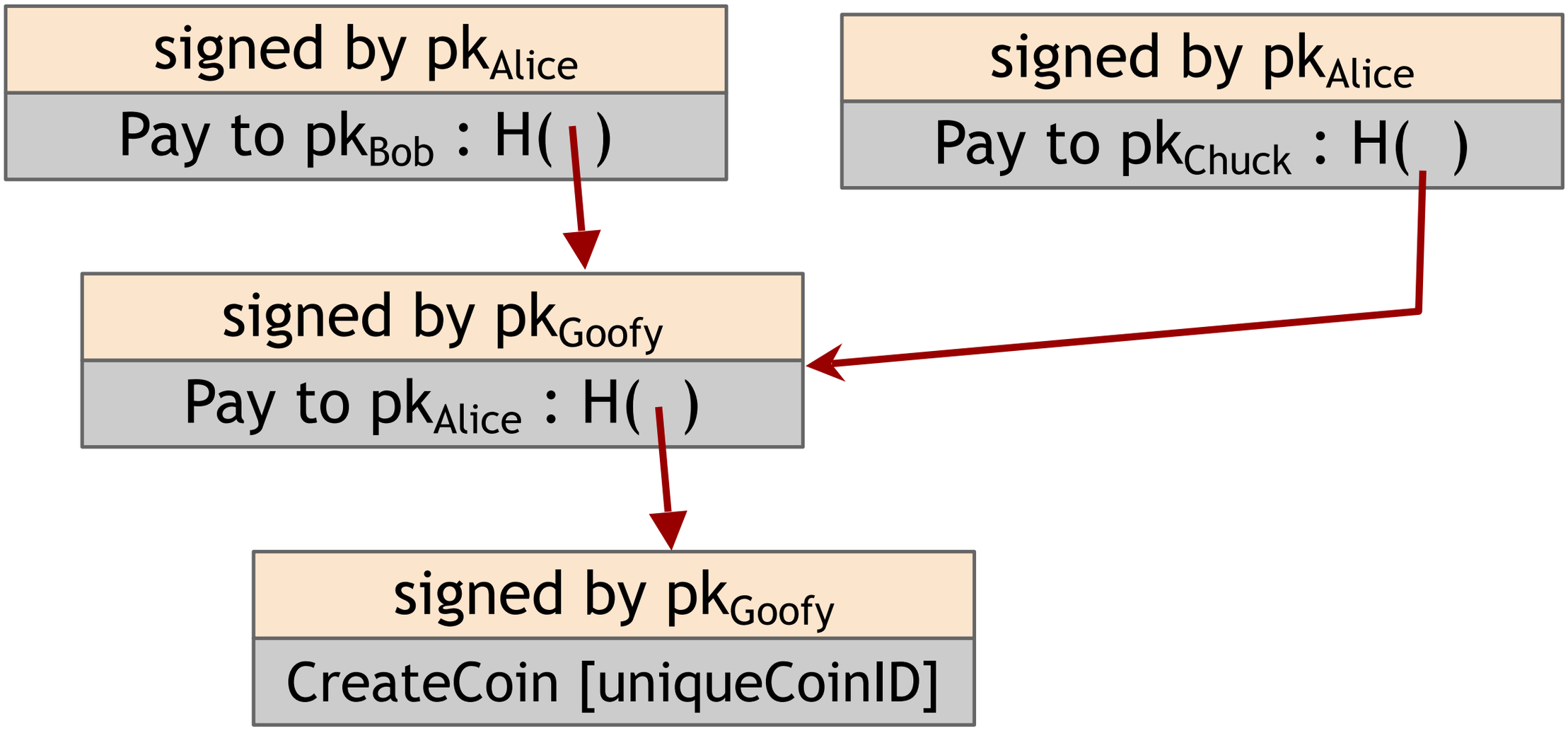
signed by pk_{Goofy}

Pay to pk_{Alice} : H( )

signed by pk_{Goofy}

CreateCoin [uniqueCoinID]

double-spending attack



This is the main design challenge in digital currency

How do we solve this?

- Simplest answer: send all transactions to an atomic, append-only centralized ledger
- Have the ledger provide a definite ordering for transactions
    - If two transactions conflict, simply disallow the later one
- No TX is valid unless the ledger has "approved" and ordered it

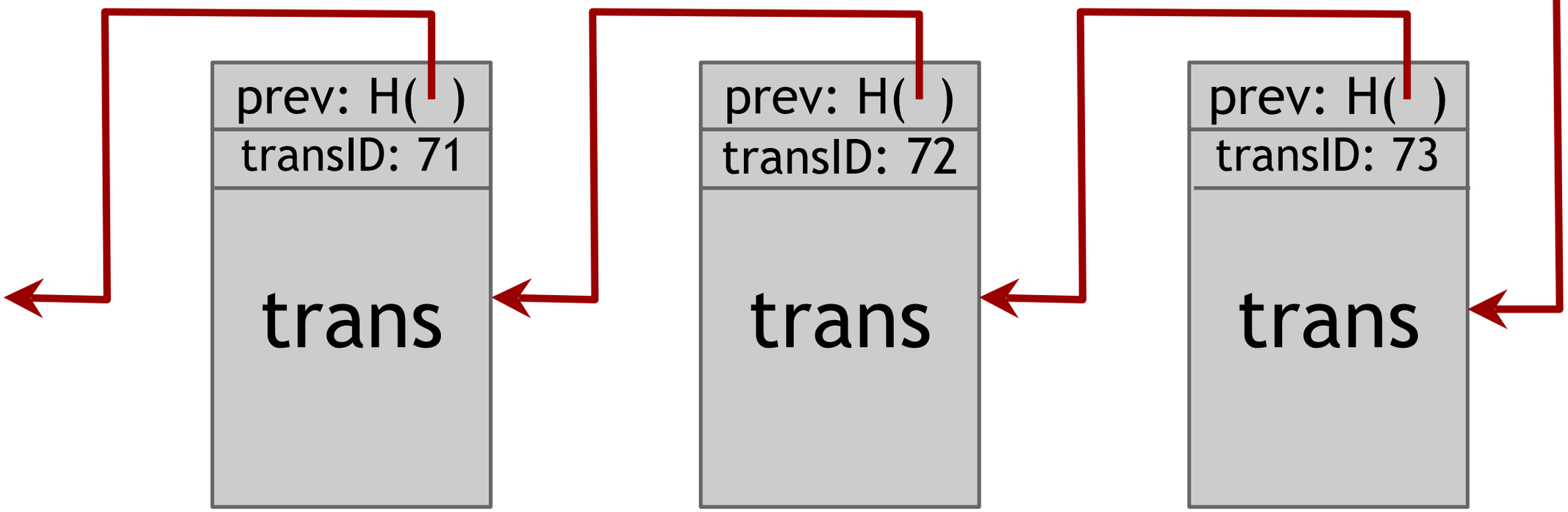
ScroogeCoin

Scrooge publishes a history of all transactions in an “append-only” ledger

H( )

Sig

Implement the ledger using a block chain, signed by Scrooge



optimization: put multiple transactions in the same block

CreateCoins transaction creates new coins

Valid, because I said so.

transID: 73 type:CreateCoins

coins created

num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

coinID 73(0)

coinID 73(1)

coinID 73(2)

signature

CreateCoins transaction creates new coins

Valid, because I said so.

transID: 73	type:CreateCoins	
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

These are public keys!

coinID 73(0)

coinID 73(1)

coinID 73(2)

signature

PayCoins transaction consumes (and destroys) some coins, and creates new coins of the same total value

transID: 73 type:PayCoins

consumed coinIDs:
68(1), 42(0), 72(3)

coins created

num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

signatures

Valid if:
-- consumed coins valid,
-- not already consumed,
-- total value out = total value in, and
-- signed by owners of all consumed coins

One signature for each consumed coin

Immutable coins

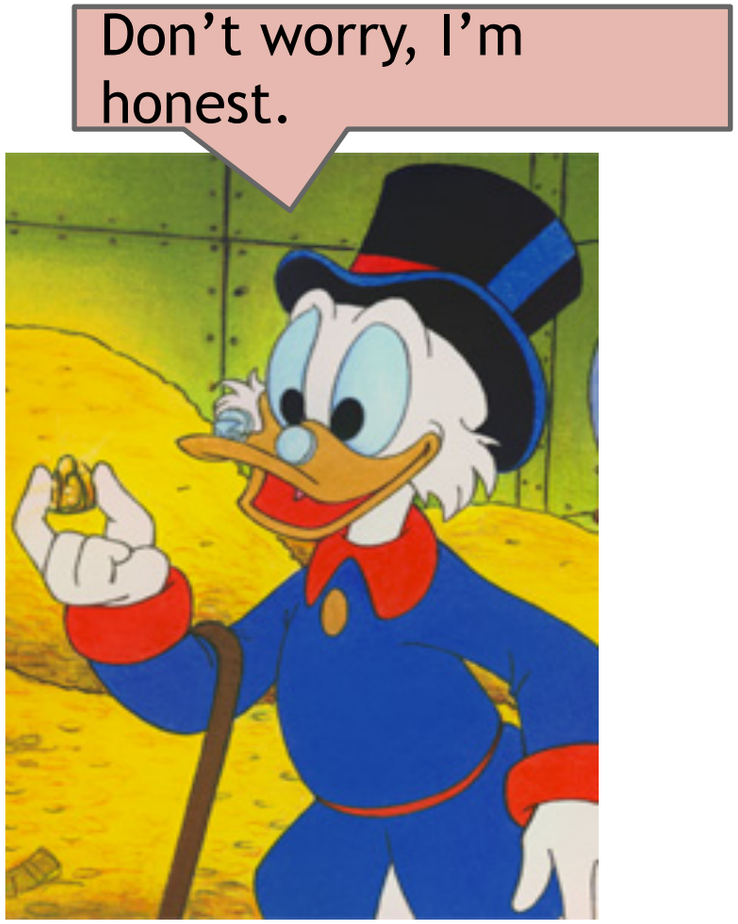
Coins can't be transferred, subdivided, or combined.

But: you can get the same effect by using transactions

to subdivide: create new transaction

consume your coin

pay out two new coins to yourself



Crucial question:

Can we descroogify the currency, and operate without any central, trusted party?

Crucial question:

Can we descroogify the currency, and operate without any central, trusted party?

Related question:

Why do we <u>need</u> to do this?

Centralization vs. Decentralization

- **Competing paradigms that underlie many technologies**

- Decentralized != Distributed
  (as in distributed system) but we'll often use them as synonyms

Aspects of decentralization in Bitcoin

1. Who maintains the ledger?
2. Who has authority over which transactions are valid?
3. Who creates (and obtains) new bitcoins?
4. Who determines how the rules change?
5. How do these coins acquire monetary value?

Aspects of decentralization in Bitcoin

Peer-to-peer network:

open to anyone, low barrier to entry

high node churn (nodes can come and go)

Mining:

open to anyone, but inevitable concentration of power

often seen as undesirable

Updates to software:

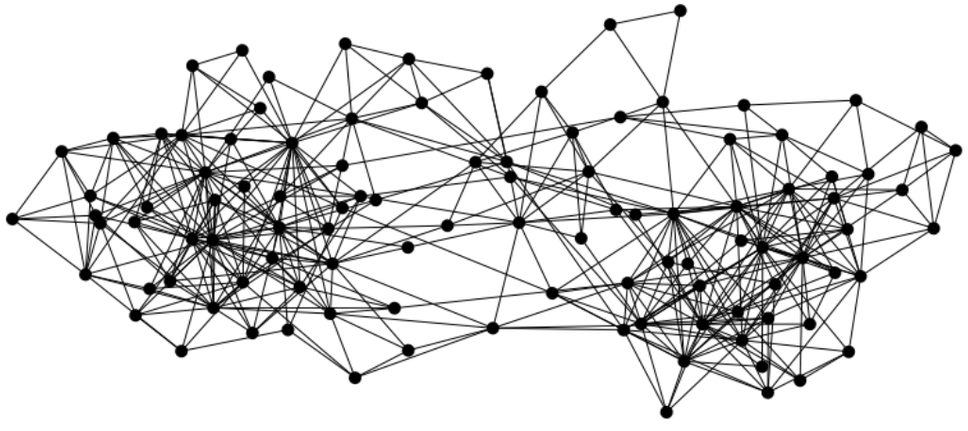
core developers trusted by community, have great power

Bitcoin is a peer-to-peer system

When Alice wants to pay Bob:
she broadcasts the transaction to all Bitcoin nodes

signed by Alice

Pay to pk_{Bob} : H( )



Note: Bob's computer is not in the picture

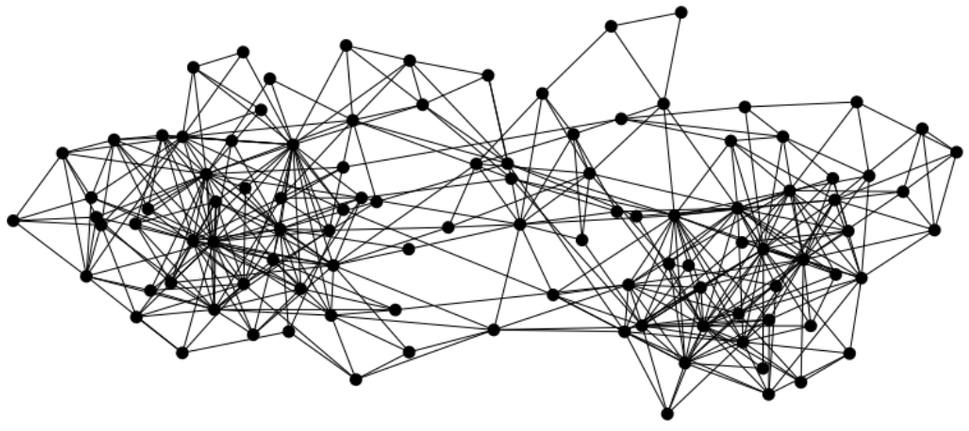
Bitcoin is a peer-to-peer system

This network is a fill/flood style P2P network:
all nodes perform basic validation, then relay
to their peers

This introduces bootstrapping, spam and DoS problems, which are dealt with through "seeders" and "reputation" scores

signed by Alice

Pay to pk_{Bob} : H( )

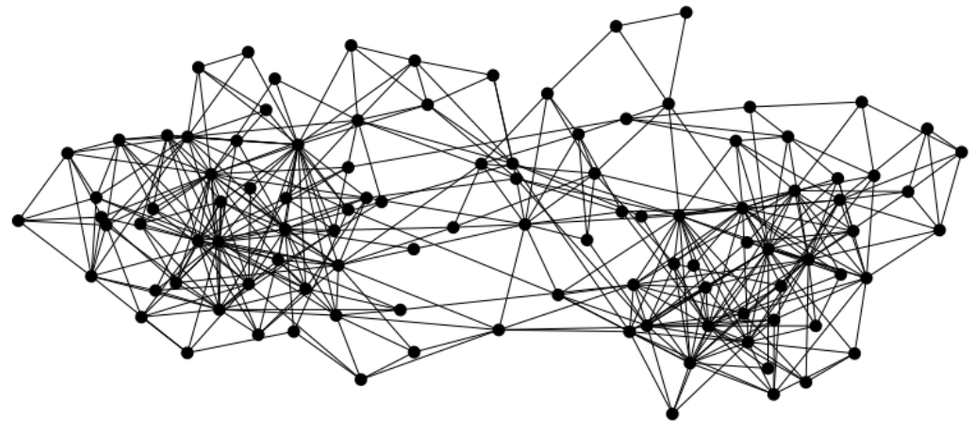


Why aren't we done here?

Why can't we just trust this system to eliminate invalid blocks, and give everyone a robust view of the Tx history?

signed by Alice

Pay to pk_{Bob} : H( )



Bitcoin's key challenge

Key technical challenge of decentralized e-cash: distributed consensus

or: how do all of these nodes agree on an ordered history of transactions?

Distributed consensus

Defining distributed consensus

The protocol terminates and all honest nodes decide on the same **value**

This value must have been proposed by some honest node

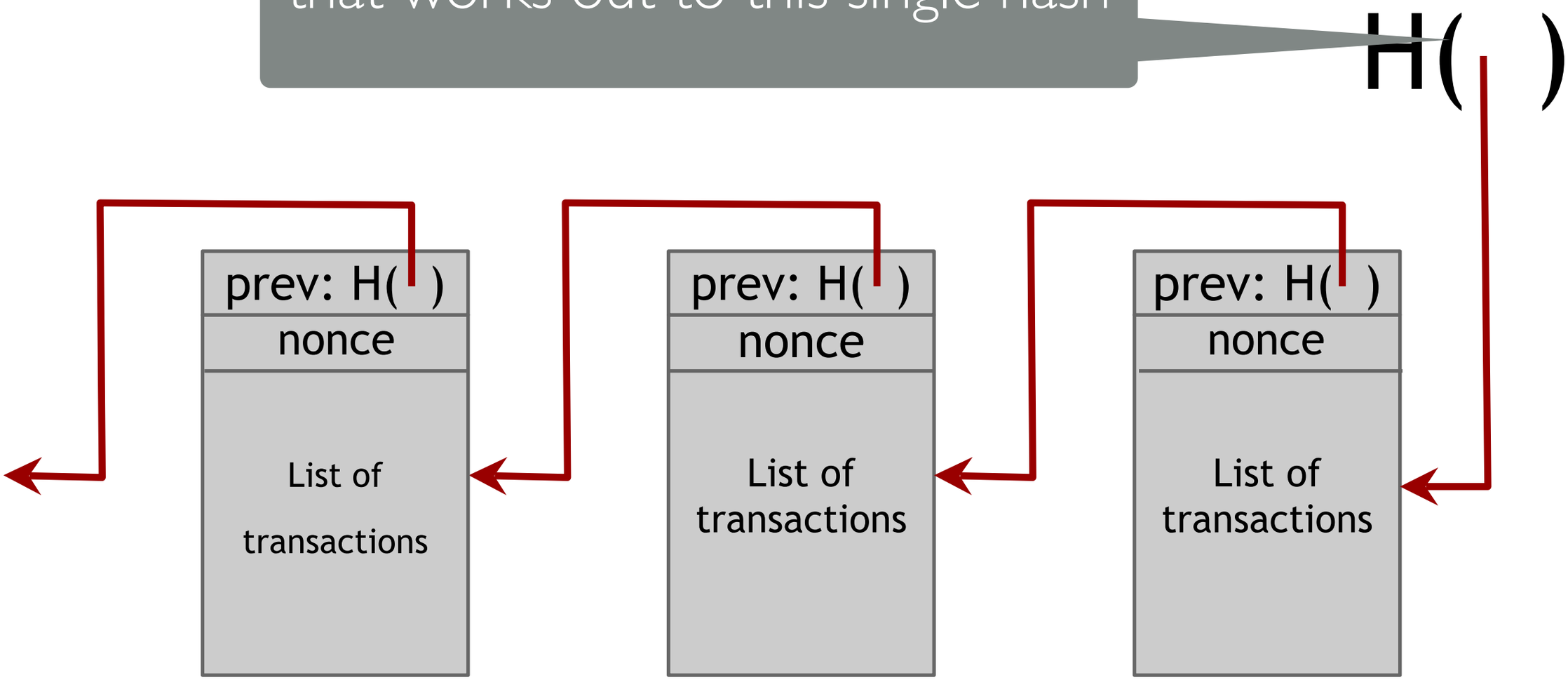
Defining distributed consensus

Q: What is this "value" in Bitcoin?

The protocol terminates and all honest nodes decide on the same **value**

This value must have been proposed by some honest node

A: In Bitcoin, the value we want to agree on is the current state of the ledger. If we use a blockchain, that works out to this single hash

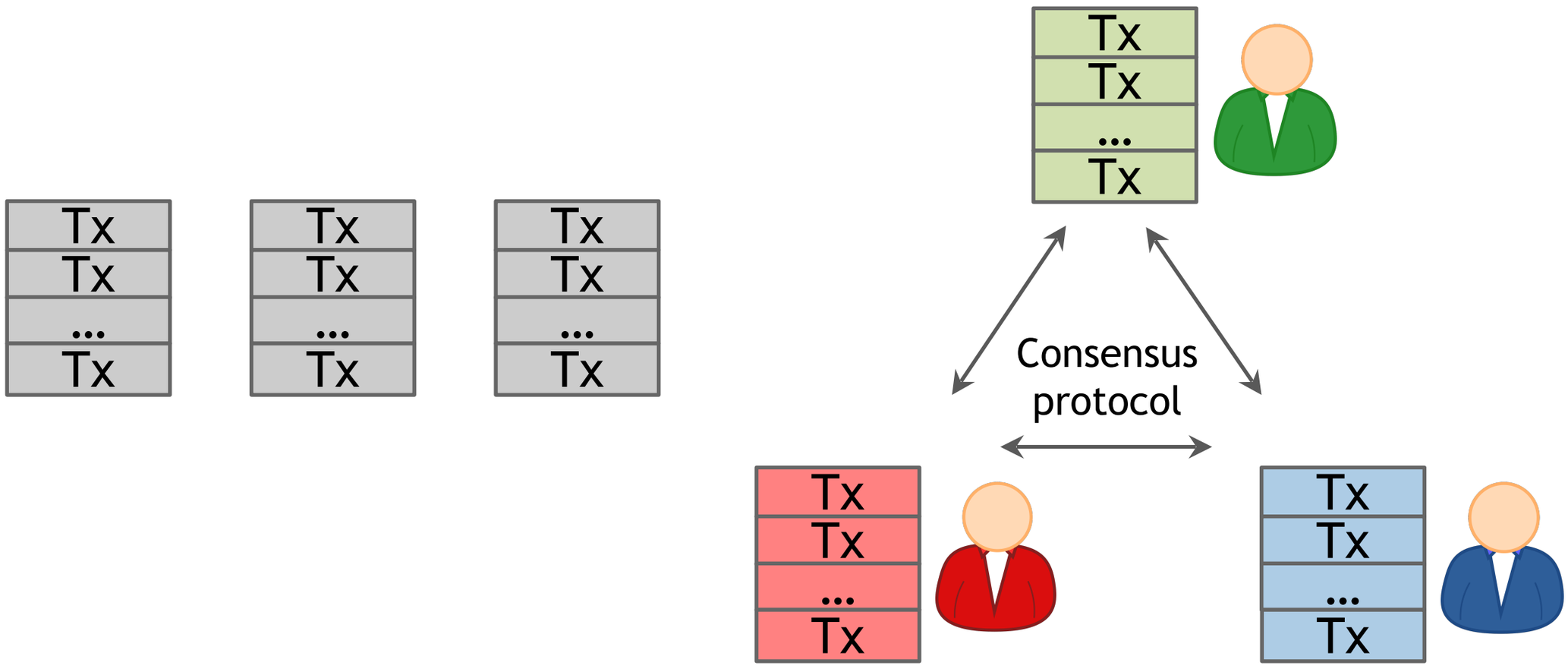


How consensus could work in Bitcoin

At any given time:

- All nodes have a sequence of blocks of transactions they've reached consensus on
- (Blocks are also distributed via p2p network)
- Each node has a set of outstanding transactions it's heard about

How consensus could work in Bitcoin



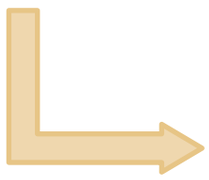
OK to select any valid block, even if proposed by only one node

Why consensus is hard

Nodes may crash
Nodes may be malicious

Network is imperfect

- Not all pairs of nodes connected
- Faults in network ("partitioning")
- Latency



No notion of global time

Many impossibility results

- Impossible without 2/3 honest majority [Pease, Shostak, Lamport'80]

- Impossible with a single faulty node, in the fully asynchronous setting, with deterministic nodes [Fischer-Lynch-Paterson'85]

Some positive results

Example: Paxos [Lamport]

Never produces inconsistent result,
but can (rarely) get stuck

Understanding impossibility results

These results say more about the model than about the problem

The models were developed to study systems like distributed databases

Bitcoin consensus: theory & practice

- Bitcoin consensus: initially, seemed to work better in practice than in theory

- Theory has been steadily catching up to explain why Bitcoin consensus works [e.g., Garay-Kiayias-Leonardos'15,Pass-Shelat-Shi'17,Garay-Kiayias-Leonardos'17,…]

- Theory is important, can help predict unforeseen attacks