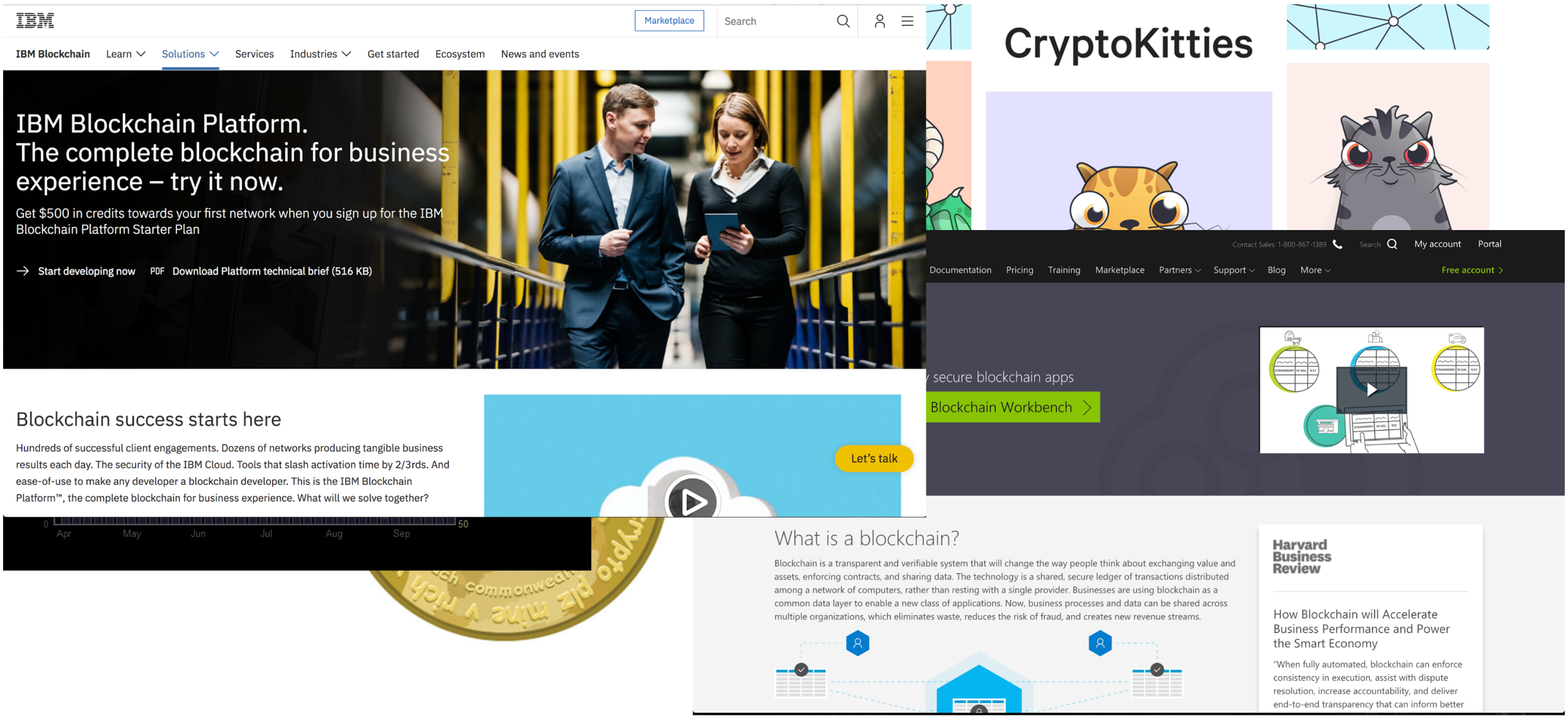
Bitcoin
Basic Concepts

Based on slides by Ariel Procaccia, Alex Psomas and Aviv Zohar

mixed and matched…

CRYPTOCURRENCIES



CRYPTOCURRENCY LOGIC

- Bitcoin was worth $200 in May 2014
- $215 in May 2015
- $450 in May 2016
- $1000 in May 2017
- $9000 in May 2018
- $10,361 in Feb 2020

THE PLAN

- Basics of Bitcoin
- Incentive Issues

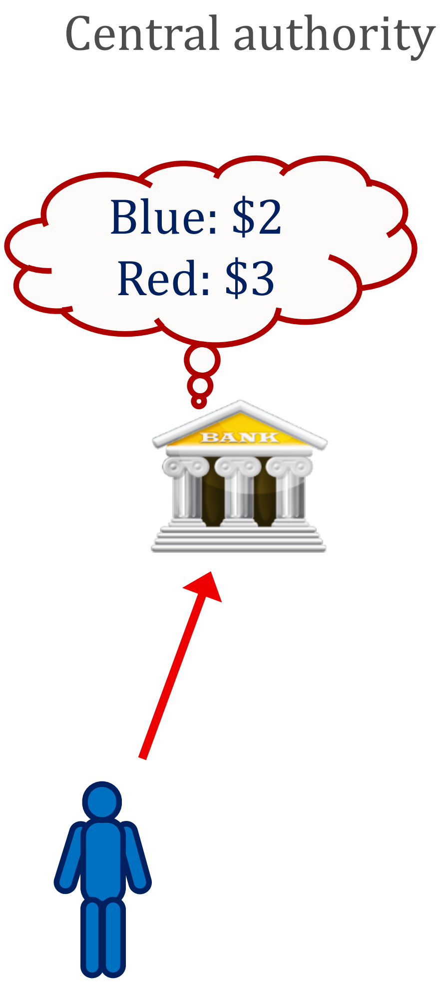
Bitcoin: A Distributed electronic currency.

Invented by Satoshi Nakamoto (2008)

FEATURES OF BITCOIN

- Purely digital
- Allows payments to be sent almost instantaneously
- Extremely low fees
- Anonymous like cash
- Bitcoin addresses (equivalent of accounts) free
- Decentralized protocol
- Supply limited

Concern: bits easily replicated. How to avoid double spending?



LEDGER

From:	To:	$$$
	Arvind	200
	Mira	200
....	....	....
Mira	Alex	50
Arvind	Anna	20
Anna	Jacob	100
....	....	....

When you send money to someone, new entry added to ledger

Invalidates transaction if there are insufficient funds.

HOW BITCOIN WORKS: MAINTAINING A LEDGER

From:	To:	$$$
	Arvind	200
	Mira	200
….	….	….
Mira	Alex	50
Arvind	Anna	20
Anna	Jacob	100
….	….	….

Ledger is public

Anyone can add lines to it.

PROBLEM #1: AUTHORIZING TRANSACTIONS

- What if someone (Alex) tries to move money to their account without the owner's (Mira) authorization?

- **Fix: Digital Signatures!**

From:	To:	$$$
	Arvind	200
	Mira	200
….	….	….
Mira	Alex	50
Arvind	Anna	20
Anna	Jacob	100
….	….	….
Mira	Alex	150
Mira	Alex	150

PROBLEM #1: AUTHORIZING TRANSACTIONS

From:	To:	$$$	Signed
....	....	....	
....	....	....	
Anna	Jacob	100	Anna's signature
Mira	Alex	150	Mira's signature
Mira	Alex	150	
....	....	....	

BASIC CRYPTOGRAPHY: SIGNATURES

- Problem: I want to cryptographically sign a document
  - Only I should be able to sign it (**unforgeability**), but everyone should be able to check that my signature is valid
- Solution: Public key cryptography
- I have a private key p_1
  - Only I know p_1
- I have a public key p_2
  - Everyone knows p_2
- Functionality:
  - $Sign(doc, p_1) =$ signed doc (only I can do this)
  - $Verify(signed\ doc, p_2, doc) \in \{Valid, Invalid\}$ (everyone can do this)

PROBLEM #1: AUTHORIZING TRANSACTIONS

Each transaction initiated by Mira has a unique identifier (sequence #)

From:	To:	$$$	Signed
....	....	....	
....	....	....	
Anna	Jacob	100	Anna’s signature
Mira	Alex	150	Mira’s signature
Mira	Alex	150	
....	....	....	

Sign(| Mira | Alex | 150 | , Mira’s private key) = Mira’s signature

Verify(signature, Mira’s public key, | Mira | Alex | 150 |) ∈ { Valid, Not Valid }

PROBLEM #2: SPENDING MONEY YOU DON'T HAVE

What if someone (George) tries to spend money they don't have?

From:	To:	$$$	Signed
….	….	….	
….	….	….	
George	Matt	1000	George' sign.
George	Jane	1000	George' sign.
George	Arvind	1000	George' sign.
….	….	….	

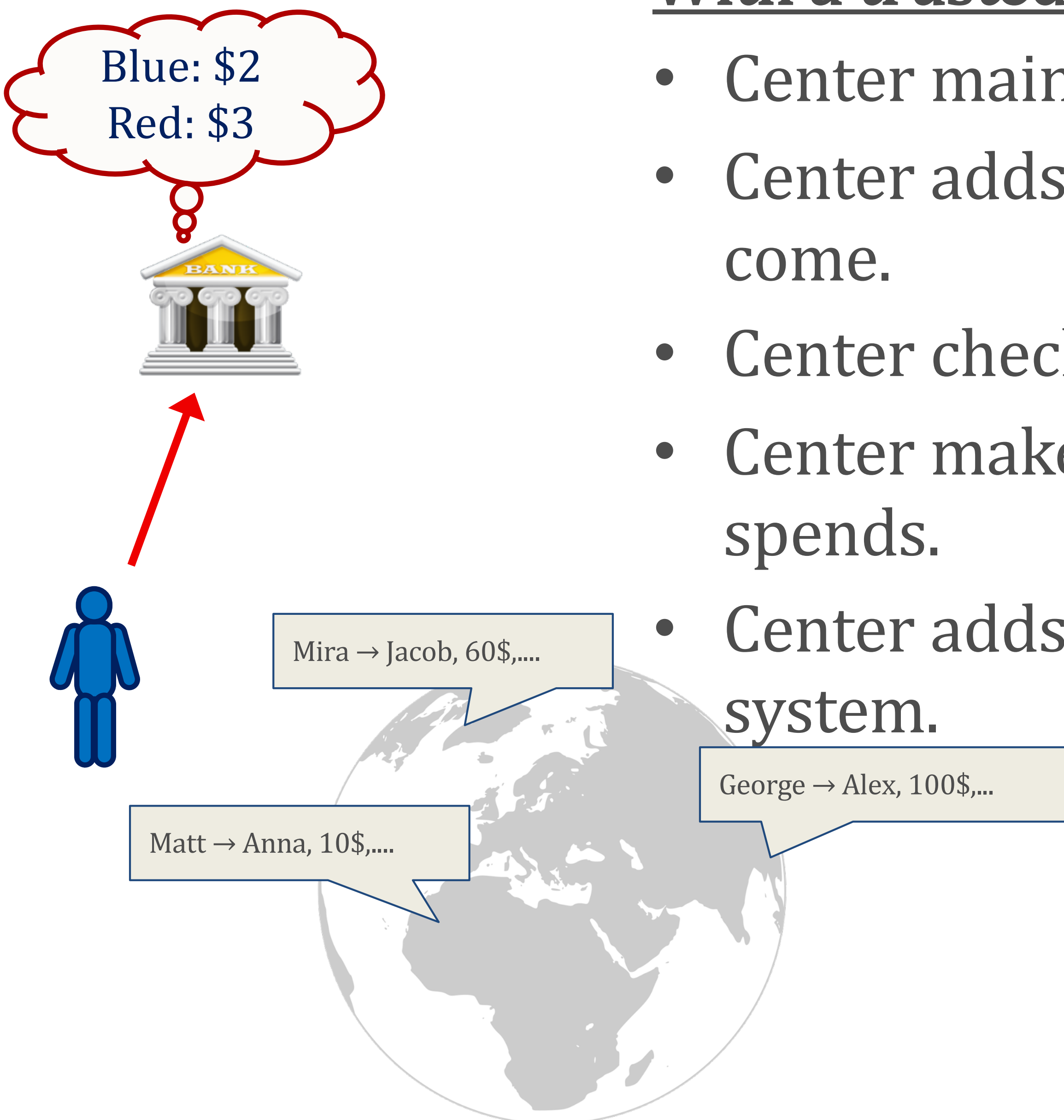
PROBLEM #2: SPENDING MONEY YOU DON'T HAVE

- Fix: Scan past transactions and check flow of money.

	From:	To:	$$$	Input	Signed
#123	Alex	George	100	#51	Alex's sign.
….	….	….	….	….	
#256	Matt	George	900	#100	Matt's sign.
….	….	….	….	….	….
#1100	George	Arvind	1000	#123, #256	George' sign.
….	….	….	….		

Make sure this money wasn't spent in this interval

HOW TO DECENTRALIZE?



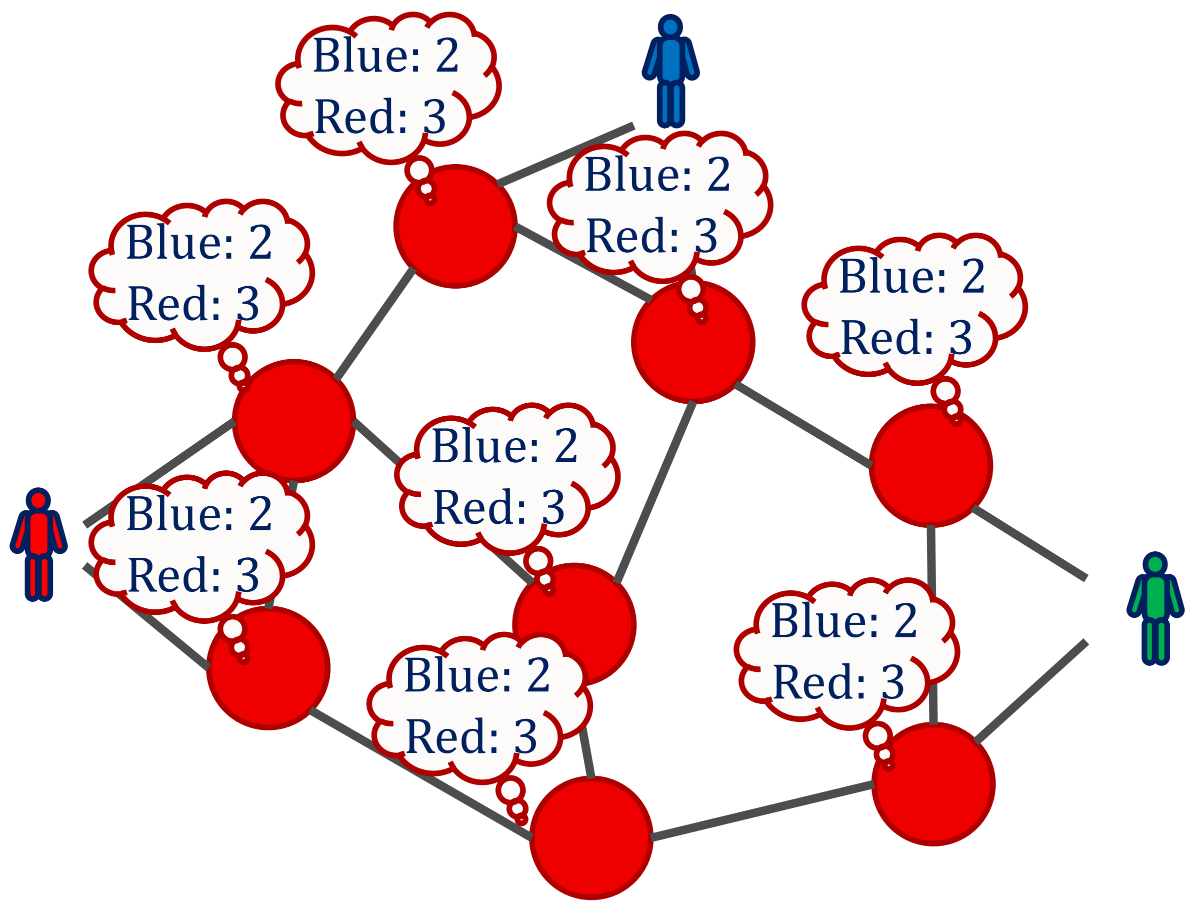
With a trusted center

- Center maintains a single ledger
- Center adds transactions as they come.
- Center checks validity.
- Center makes sure no one double spends.
- Center adds new people to the system.

Bitcoin replaces centralized intermediary with decentralized P2P system of “Bitcoin miners”, each with copy of entire ledger.

Central authority

Distributed P2P system



TRANSACTIONS

- When someone wants to transfer money to somone else, they send the transaction to **everyone in the network.**
  - Sender (identified by public key)
  - Receiver. (identified by public key)
  - Amount of BTC to be transferred from sender to receiver
  - Proof of ownership (pointer to previous transactions that verify sufficient funds)
  - Transaction fee, paid by sender to authorizer of transaction
  - Signature

Transaction is **valid** if

- Signature is valid
- Sender owns the BTC being transferred.

Each miner checks validity and "adds to ledger".

PROBLEM #3: DECENTRALIZATION

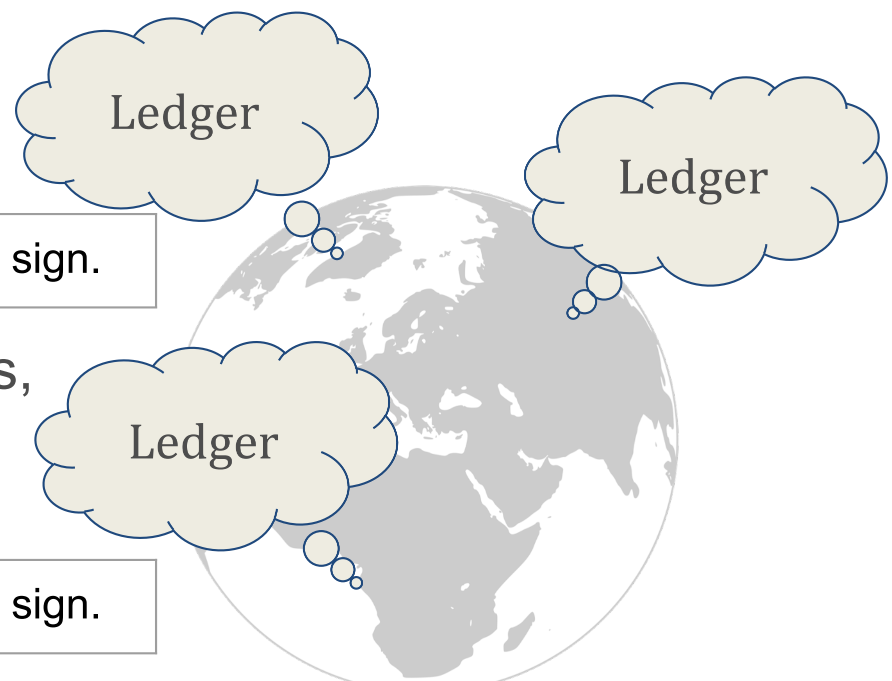
- How do we make sure that everybody has the same view of history?
- Need a protocol for how to accept/reject transactions, and in what order, so that everyone is confident of consistency of the ledger.

Scenario: Alex wants to buy a car from Matt.

Alex	Matt	10000$	Input = #127	Alex's sign.

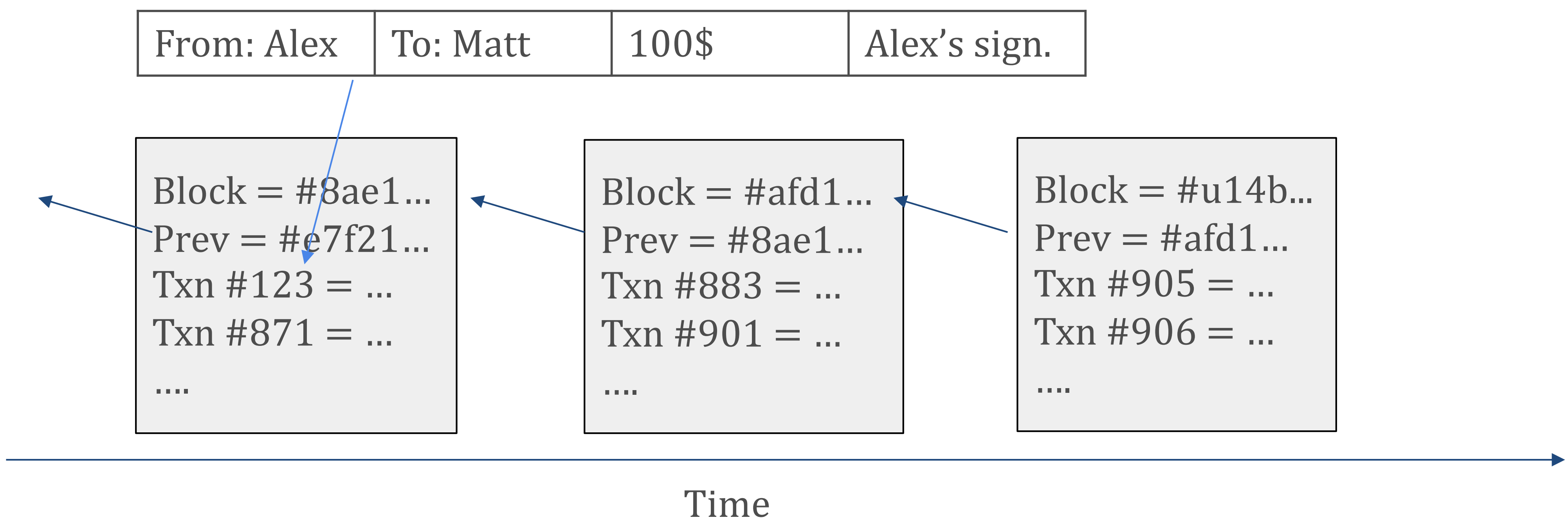
As soon as Matt gives Alex the keys, broadcast:

Alex	Jacob	10000$	Input = #127	Alex's sign.



LEDGER STORED IN BLOCKCHAIN

- Blockchain is sequence of **blocks** ordered in time.
- A block contains confirmed/valid transactions
- Each block contains a pointer to its predecessor
- Each block contains cryptographic hash of its predecessor

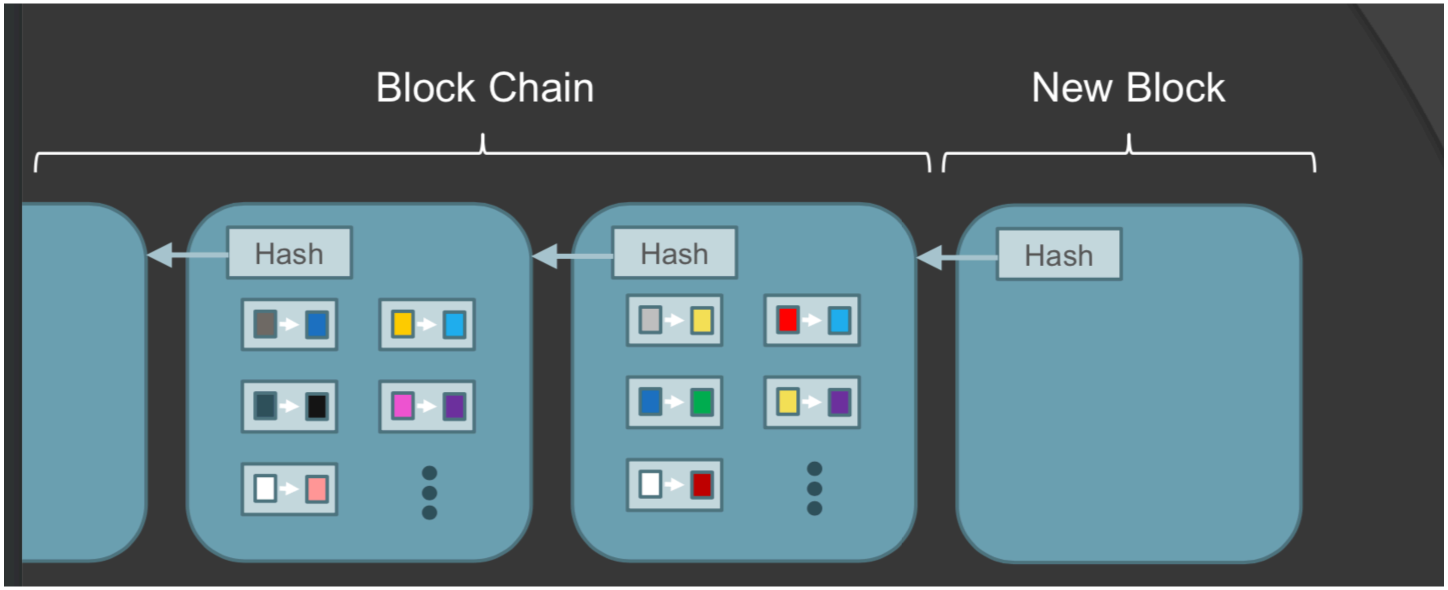


CRYPTOGRAPHIC HASH FUNCTIONS

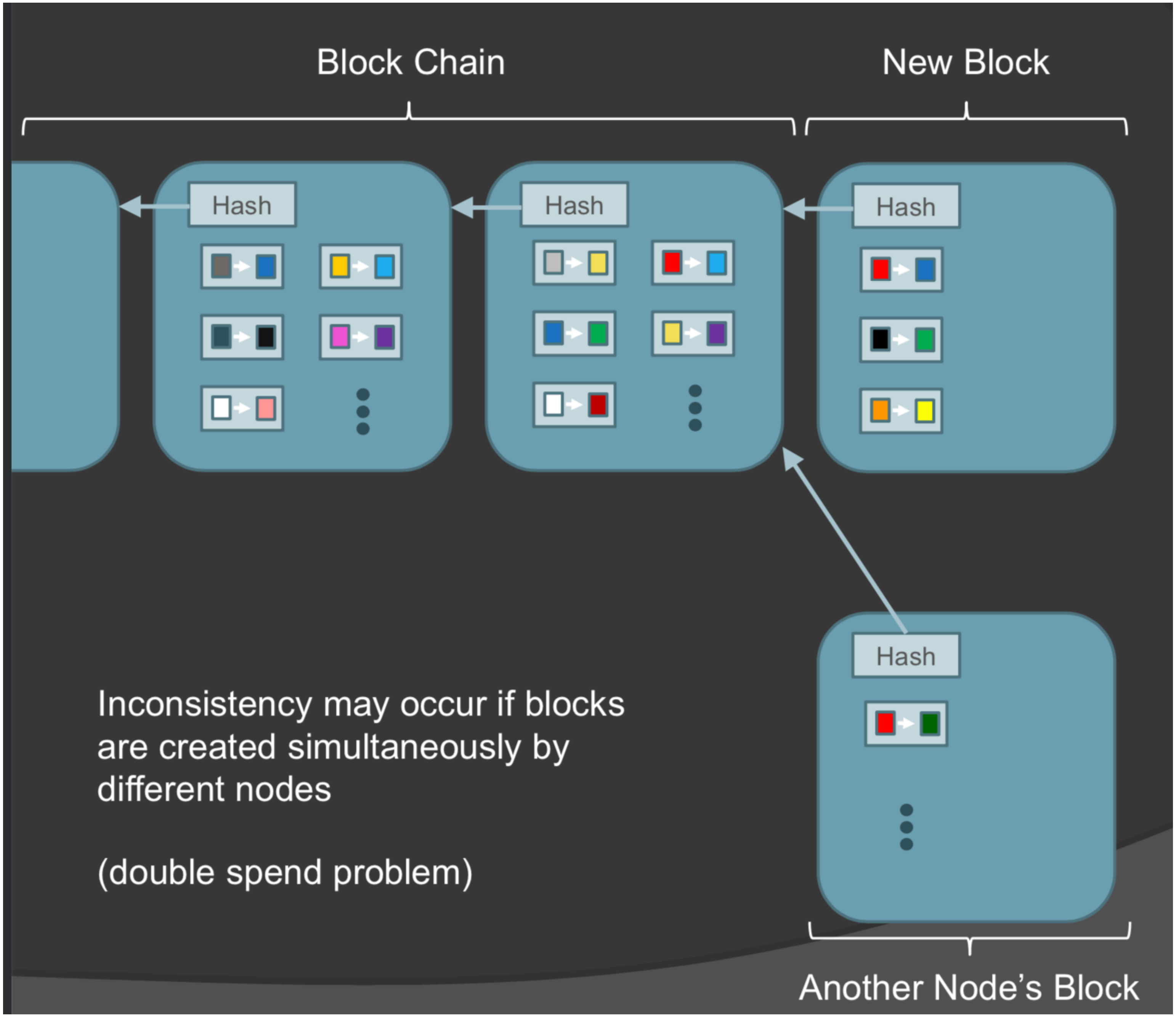
- Input:
  - String of any size
- Output:
  - Fixed size output (say 256-bits)
- Property #1: Efficiently computable
  - In fact linear time
- Property #2: Collision resistant
  - Basically impossible (computationally) to find a collision: inputs x and y that map to the same output $H(x) = H(y)$
  - Note: collisions exist. We ask that they are hard to find.

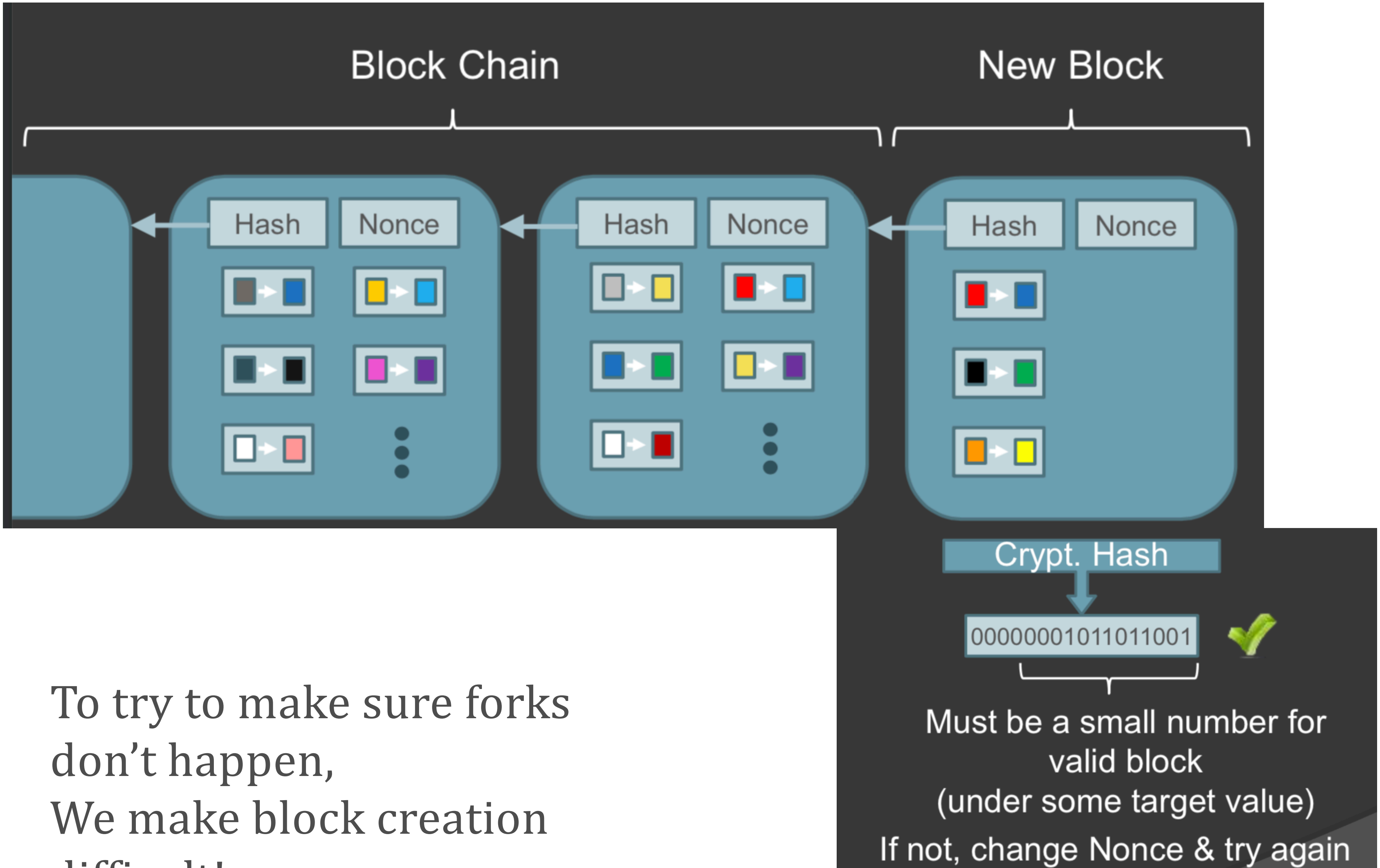
BASIC CRYPTOGRAPHY 1: HASH FUNCTIONS

- Property #3: Hiding
  - Looks random.
  - Slightly change input and hash changes completely and unpredictably.
  - If a value x is chosen from a sufficiently big set, then given $H(x)$ it is hard to find x
  - If goal is to find input x that gives particular output H(x), nothing better than guessing and checking (we believe).



- Blocks contain batch of transactions
- Each block contains a cryptographic hash of prev block, "proving" it was created later.
- Can read ledger from start to finish to "follow the money"
- Each node (miner) tries to grow the chain with recent transactions that they have heard about
  - Create a block with recent consistent transactions
  - Send to peers


Block Chain
New Block
Hash
Hash
Hash
Hash
Inconsistency may occur if blocks are created simultaneously by different nodes
(double spend problem)
Another Node's Block



To try to make sure forks don’t happen,
We make block creation difficult!

Nonce: a bunch of bits that can be set arbitrarily.

Key: Miners compete to create blocks.

PROOF OF WORK

Miners compete to solve a **“crypto puzzle”**

Goal: The cryptographic hash of the entire text of a block plus an additional number (the nonce) must be in a certain range

SHA256 (Block =…
Txn #905 = …
Txn #906 = …
….
nonce) = 0000000000000000….00b39d9ca51f07fef3429ae15.

A bunch of zeros

Why do we call this a ``proof of work”?

CRYPTOGRAPHIC HASH FUNCTIONS

- Recall, cryptographic hash functions are "hiding".

SHA256 (Block =…
 Txn #905 = …
 Txn #906 = …
 ….
 nonce) = 000000000000000….00b39d9ca51f07fef3429ae15.

- No faster way of finding such a nonce than just trying random strings.

PROOF OF WORK

Miners compete to solve a **"crypto puzzle"**

Goal: The cryptographic hash of the entire text of a block plus an additional number (the nonce) must be in a certain range

SHA256 (

Block =...
Txn #905 = ...
Txn #906 = ...
....
nonce

) = 0x00000000000000….00b39d9ca51f07fef3429ae15.

A bunch of zeros

This means that a miner's chance of solving the puzzle first is proportional to that miner's **computational power!**

SOURCE OF TRUST IN BITCOIN

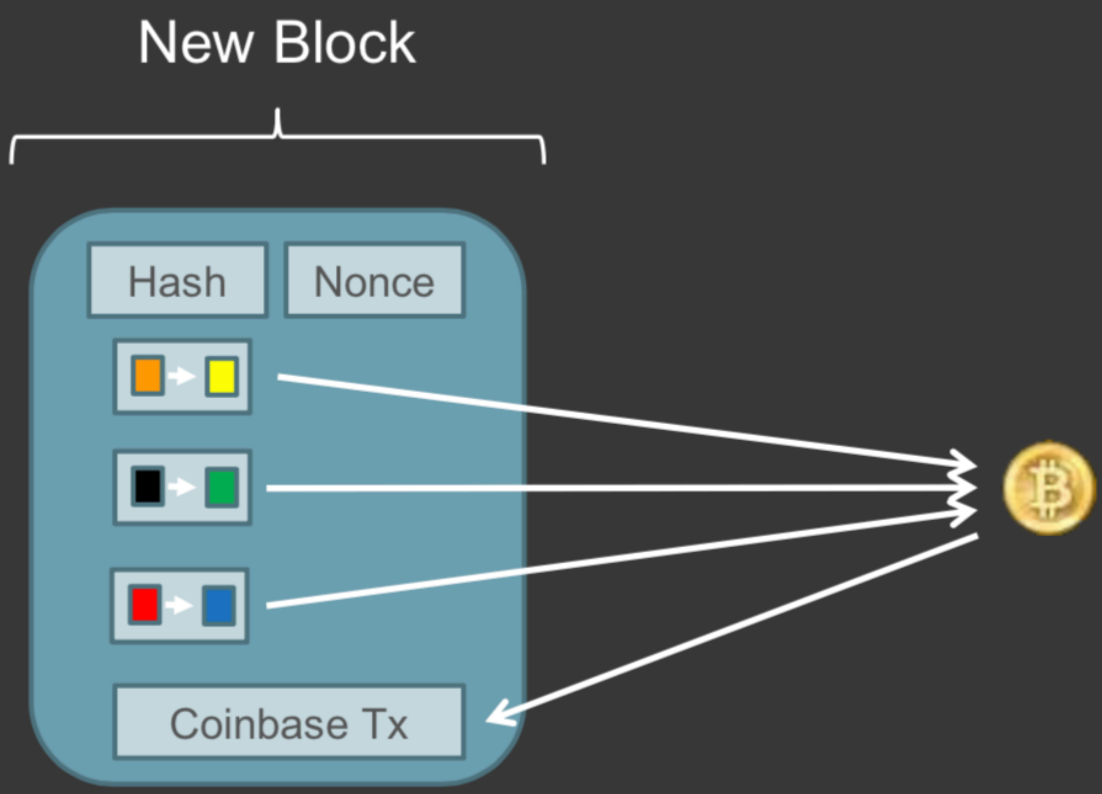
- Trust version of ledger that has the most computational work going into it.
- Creating a “bad” ledger would require infeasible amount of computation.

WHY DO THEY DO IT?

Block creators are rewarded in two ways:

- Block reward: add a special transaction giving the miner a certain number of (new) bitcoins. Was 12.5 Bitcoin per block a few weeks ago. Now it's 6.25.
- Transaction fees: "tips" from the participants of the transaction to the miner, if the transaction is included in the new block.

To encourage nodes to authorize transactions:



Reward the authorizer with fees from each transaction (+ newly minted money)

Block creation is known as "Mining"

Block size is limited (currently to 1MB)
Transactions will compete to enter – highest fee first.
(An auction!)

CONSENSUS

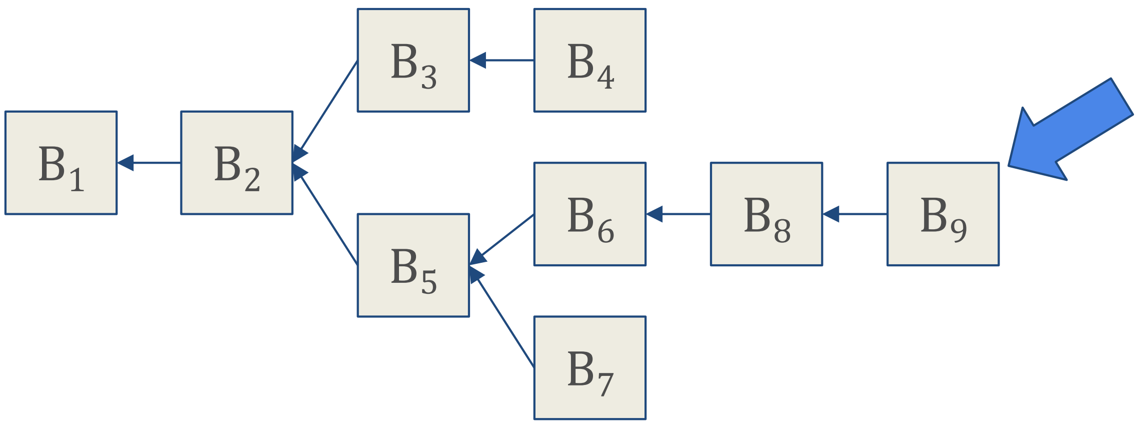
- Blocks rarely created, so few conflicts (forks)
  - i.e., likely that new block propogated everywhere before next one produced.
- Therefore, there won't be conflicting histories.

FORKS

- If two miners discover valid blocks at around the same time, there will be a fork in the blockchain.
- Need a mechanism for choosing one:
  - So that everybody knows which transactions have been authorized
  - So Bitcoin miners know which block they should be trying to extend.

BITCOIN PROTOCOL SAYS:

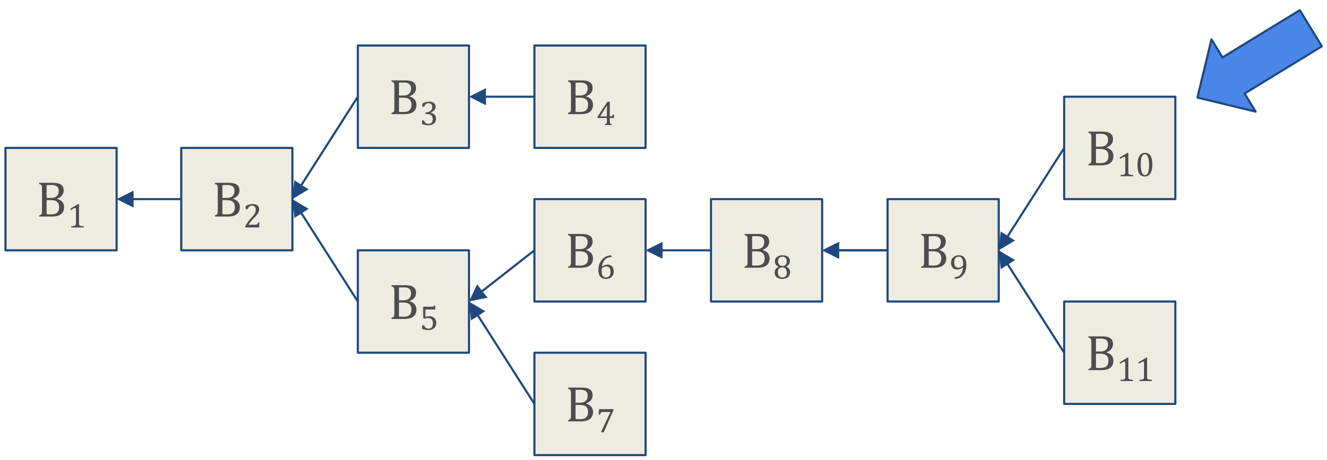
- The network so far:



- Users should regard **longest chain** as valid blockchain, breaking ties in favor of what user hears about first

BRANCHES

- The network so far:



- More than one block is solved at the same time
- Which block should a miner try to extend?

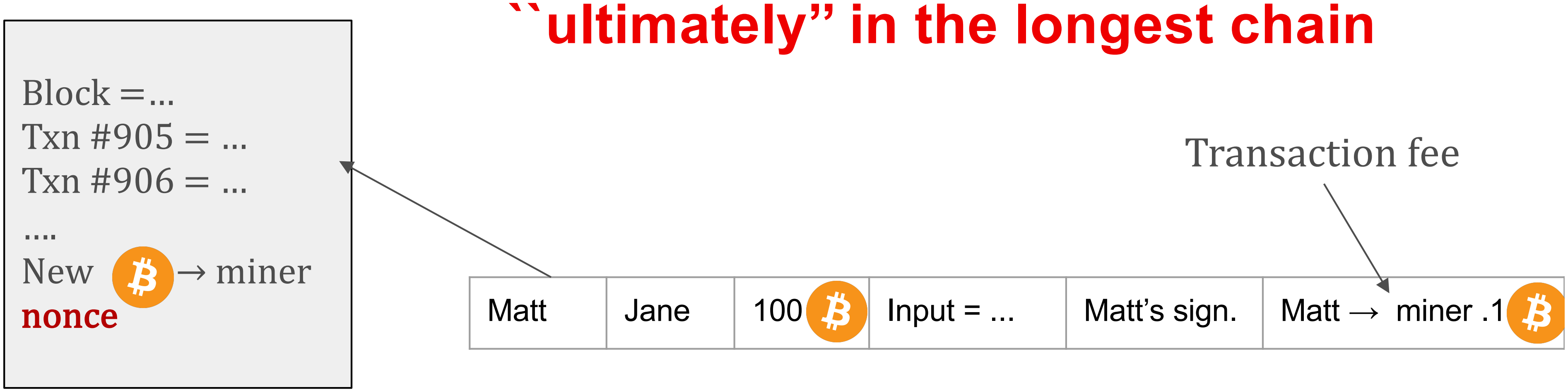
The first one you hear about

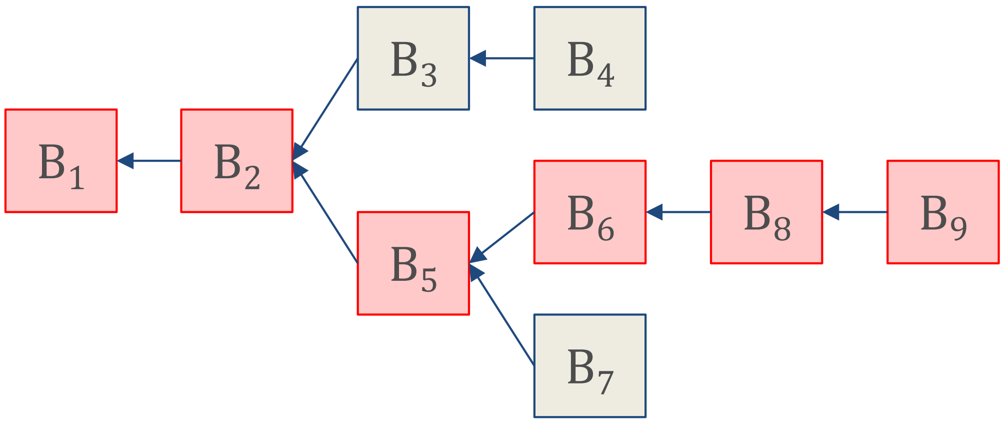
WHY DO THEY DO IT?

Block creators are rewarded in two ways:

- Block reward: add a special transaction giving the miner a certain number of (new) bitcoins. Currently 12.5 Bitcoin per block.
- Transaction fees: "tips" from the participants of the transaction to the miner, if the transaction is included in the new block.

These rewards are "real" only if the block is in the "true" history, i.e. this block is ``ultimately" in the longest chain





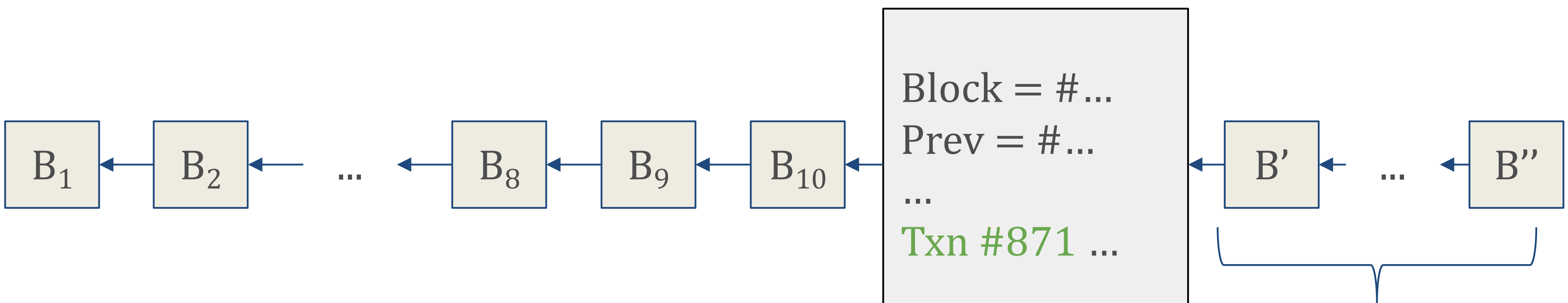
Only the red blocks are considered valid.

OTHER DETAILS

- The number of leading zeros gets adjusted every 2016 blocks so that a block gets created every ~10 minutes
- The block reward is scheduled to be halved every 4 years **and it just happened**!
  - Eventually all rewards will come from transaction fees

RECAP

View of someone who wants to make a transaction



Txn=#871	George	Anna	1 ₿	....

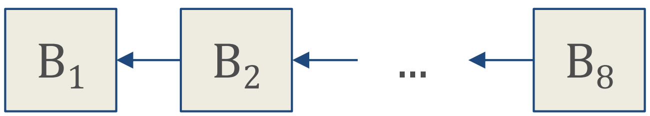
Wait a few blocks until you can say that the transaction is confirmed

WHY?

Want some assurance that this block will be on the longest chain in the long run!

PROOF OF WORK: RECAP

View of a miner



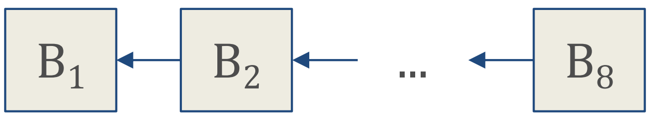
SHA256 (

Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
nonce

) = 0x0b39d9ca51f07fef3429ae15…

PROOF OF WORK: RECAP

View of a miner



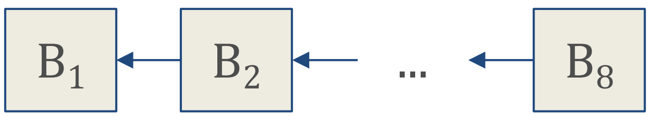
SHA256 (

Block = #8ae1...
Prev = B_8
....
Txn #123 = ...
....
nonce'

) = 0x000000ef34244s1jd99a533g...

PROOF OF WORK: RECAP

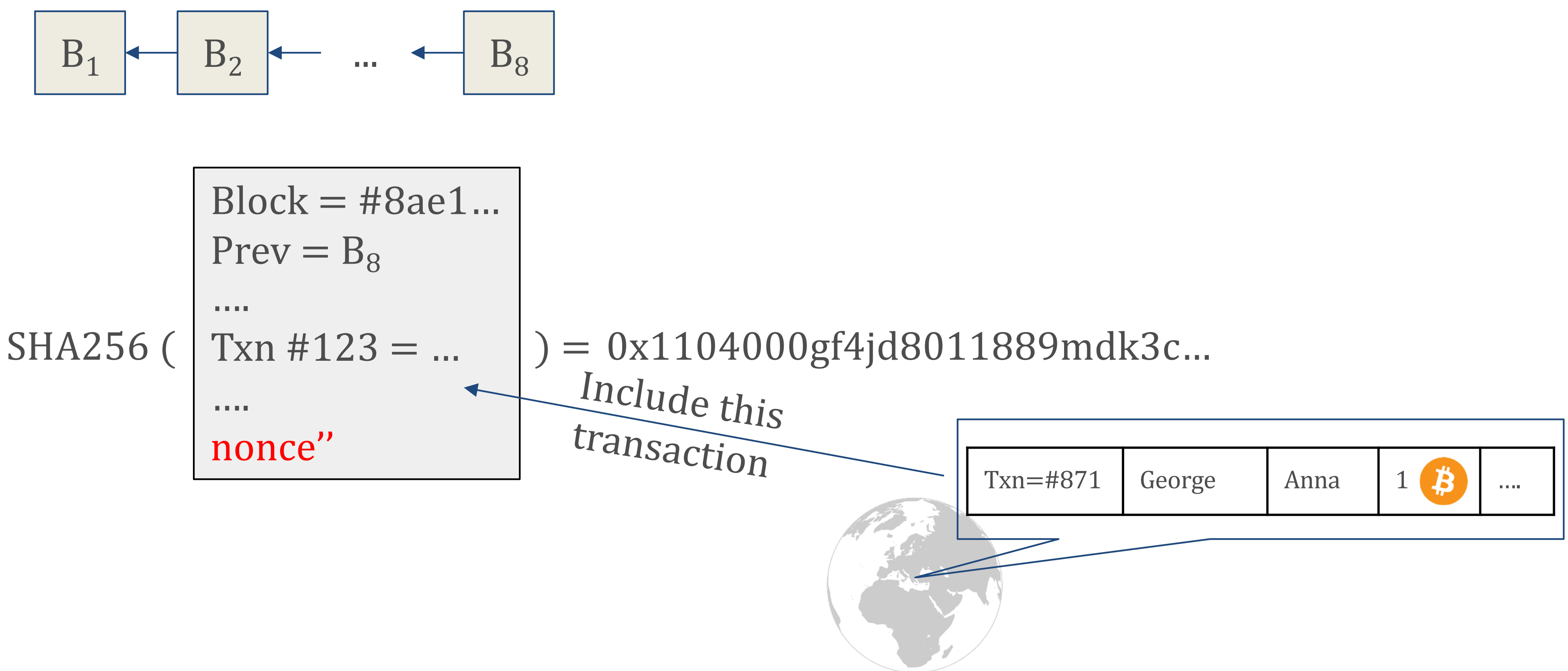
View of a miner



SHA256 (
Block = #8ae1…
Prev = B_8
….
Txn #123 = …
….
nonce''
) = 0x1104000gf4jd8011889mdk3c…

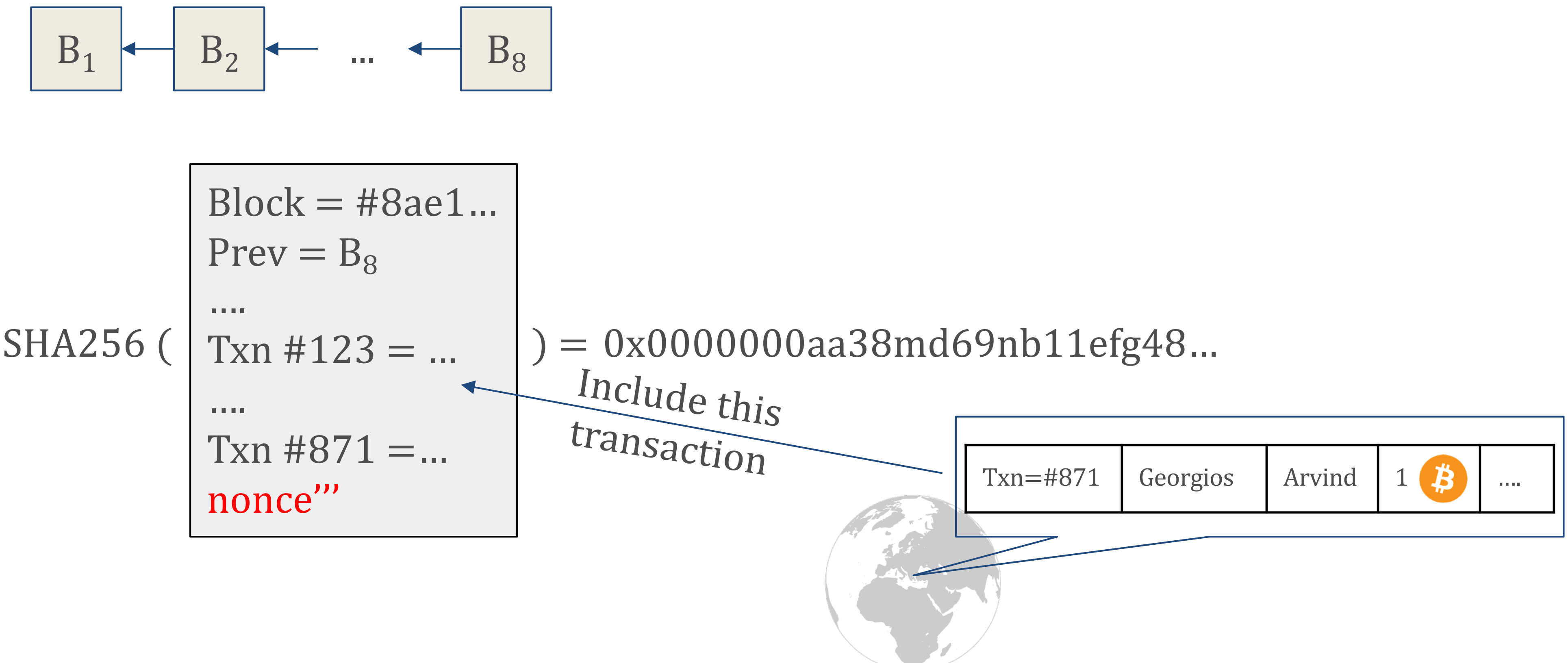
PROOF OF WORK: RECAP

View of a miner



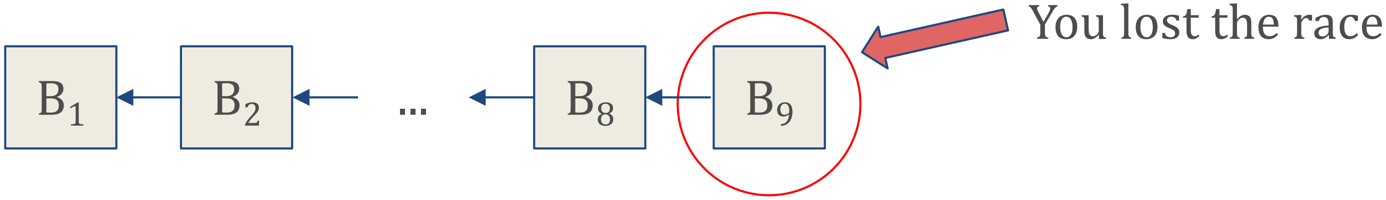
PROOF OF WORK: RECAP

View of a miner



PROOF OF WORK: RECAP

View of a miner



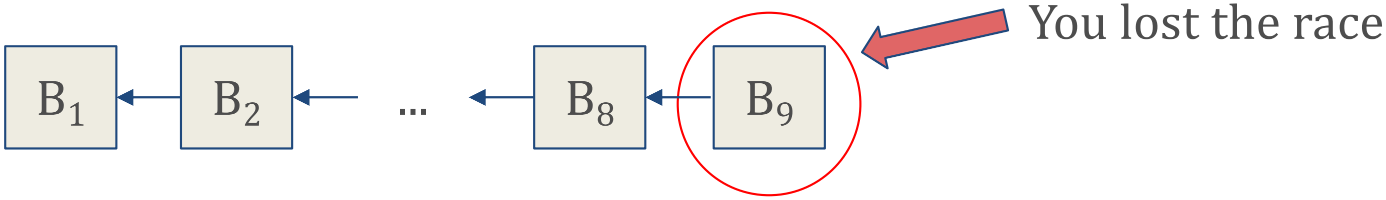
SHA256 (

Block = #8ae1...
Prev = B_8
....
Txn #123 = ...
....
Txn #871 =...
nonce''

) = 0x0000000aa38md69nb11efg48...

PROOF OF WORK: RECAP

View of a miner



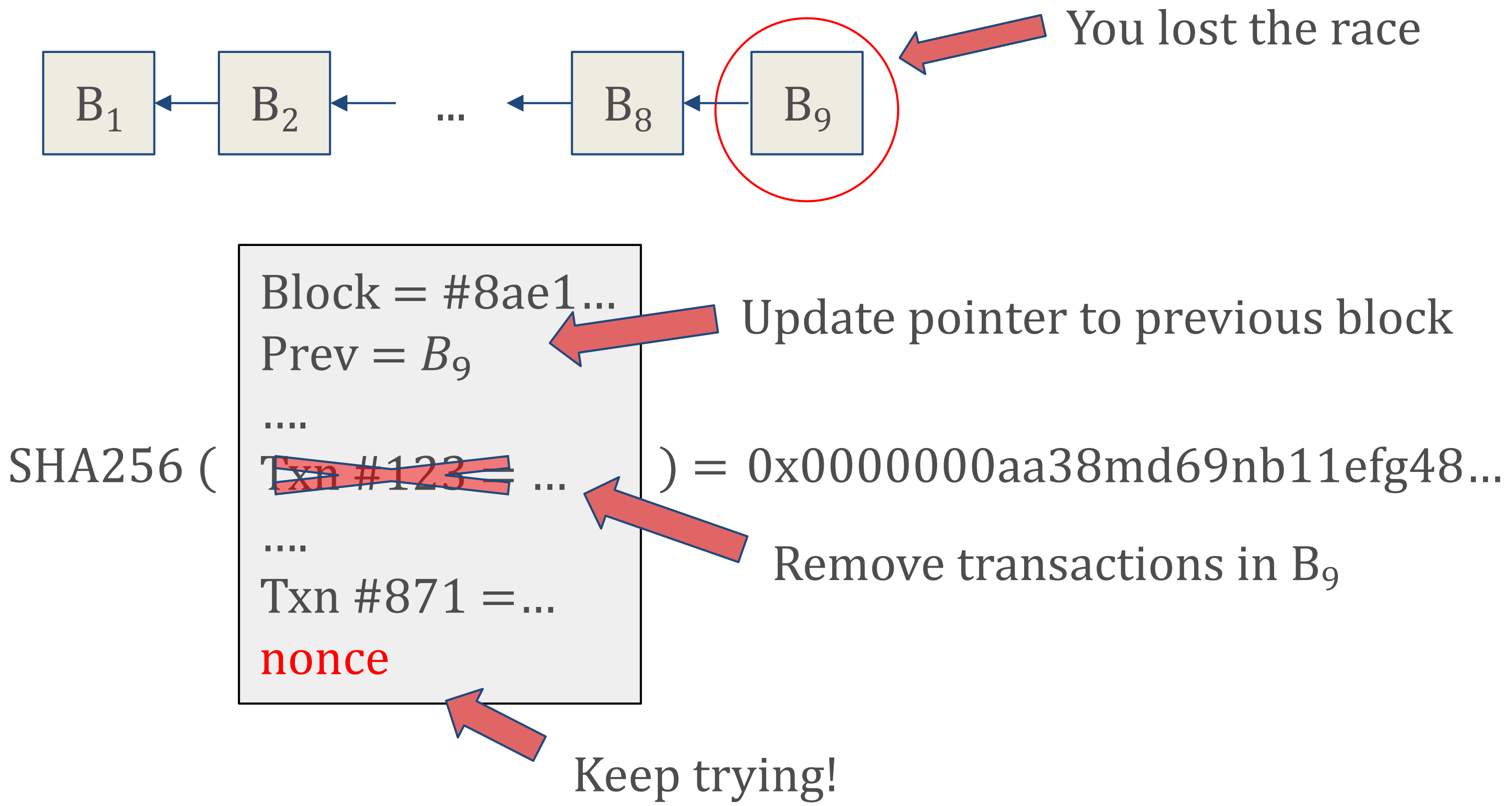
SHA256 (Block = #8ae1...
Prev = B_9
....
~~Txn #123 =~~ ...
....
Txn #871 =...
nonce'') = 0x0000000aa38md69nb11efg48...

Update pointer to previous block

Remove transactions in B_9

PROOF OF WORK: RECAP

View of a miner



RECAP OF BITCOIN

- **Transactions:** At any time, any buyer b can generate a transaction to pay d BTC to seller s.
- **Block:** A block consists of
  - A set of transactions
  - A cryptographic hash of the previous block (pointer to previous block
  - An ID of the miner for this block
  - A nonce.
- A set of properly signed transactions is **valid** if no account ever overspent its limit.
- A block is valid if
  - It points to a valid block.
  - All transactions on the chain to B are valid.
  - SHA256(nonce|| info in block) has k leading zeros.

RECAP OF BITCOIN II

- **Mining:** the process of extending the blockchain from some block B.
- Longest Chain Protocol (for miners):
  - Choose B to be the block furthest from the root, tie-breaking in favor of the first block you heard about.
  - Include all valid transactions you've heard about.
  - As soon as valid block created, announce it to the network.
- Miners are paid for creating valid blocks with freshly minted Bitcoins and with transaction fees.
- Difficulty of the puzzle is adjusted every 2016 blocks with the objective of making it so that a block takes 10 minutes to make in expectation.

KEY IDEA

- Trust the ledger that has the most "computational work" put into it.

- Ensure that fraudulent transactions/conflicting ledgers would require an infeasible amount of computation to create.

BITCOIN

- Is a mechanism.

- Question for us: are there beneficial deviations that can help a miner earn more than his fair share of rewards?