Introduction to Bitcoin

Mohsen Lesani

Original slides adopted from Maurice Herlihy

Abstraction: Distributed Ledger

Cash				
Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		$ 7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 8, 20X3	Cash sale	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000

Abstraction: Distributed Ledger

Cash

Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		$ 7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 8, 20X3	Cash sale	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000

Append-only list of events

Abstraction: Distributed Ledger

Cash

Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 8, 20X3	Cash sale	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000

Append-only list of events

Not just financial

Abstraction: Distributed Ledger

Cash

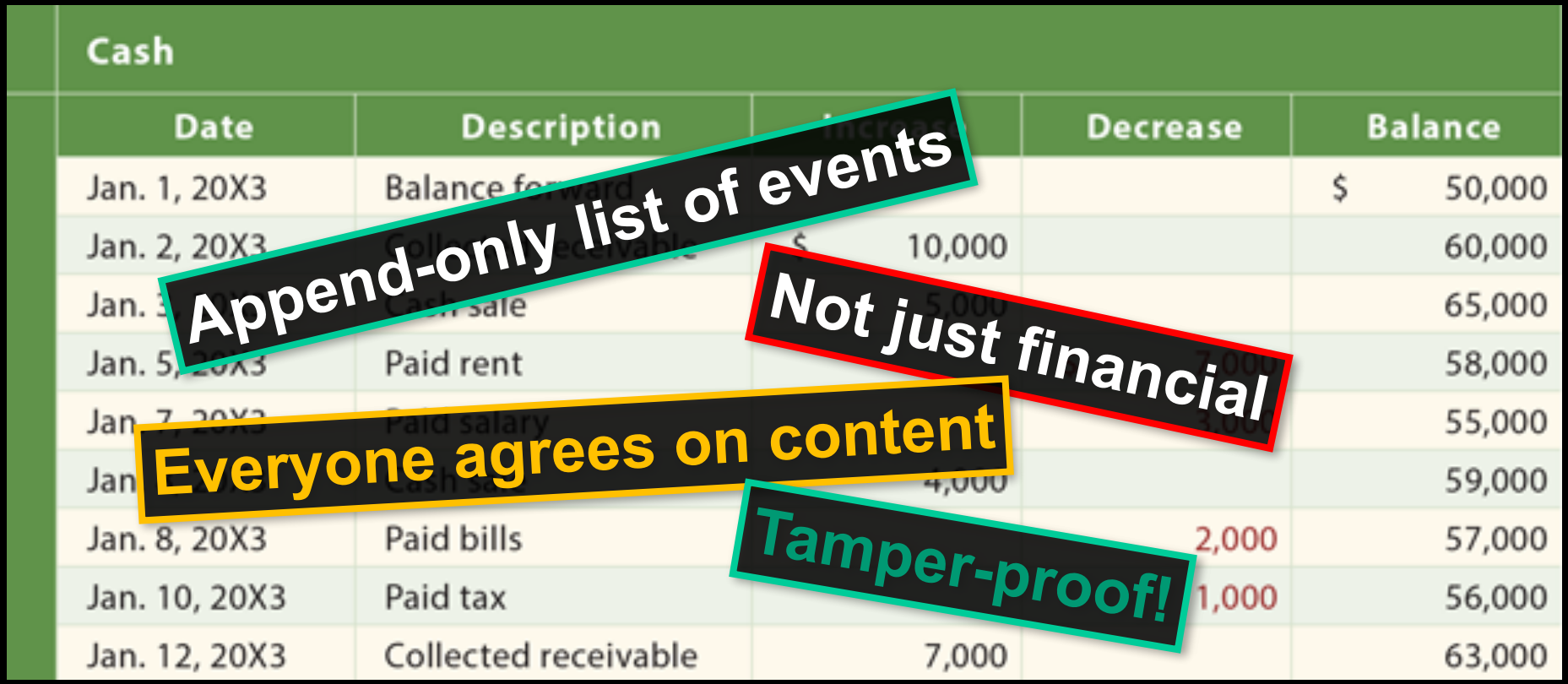
Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	[illegible]	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. [illegible]	[illegible]	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000

Append-only list of events

Not just financial

Everyone agrees on content

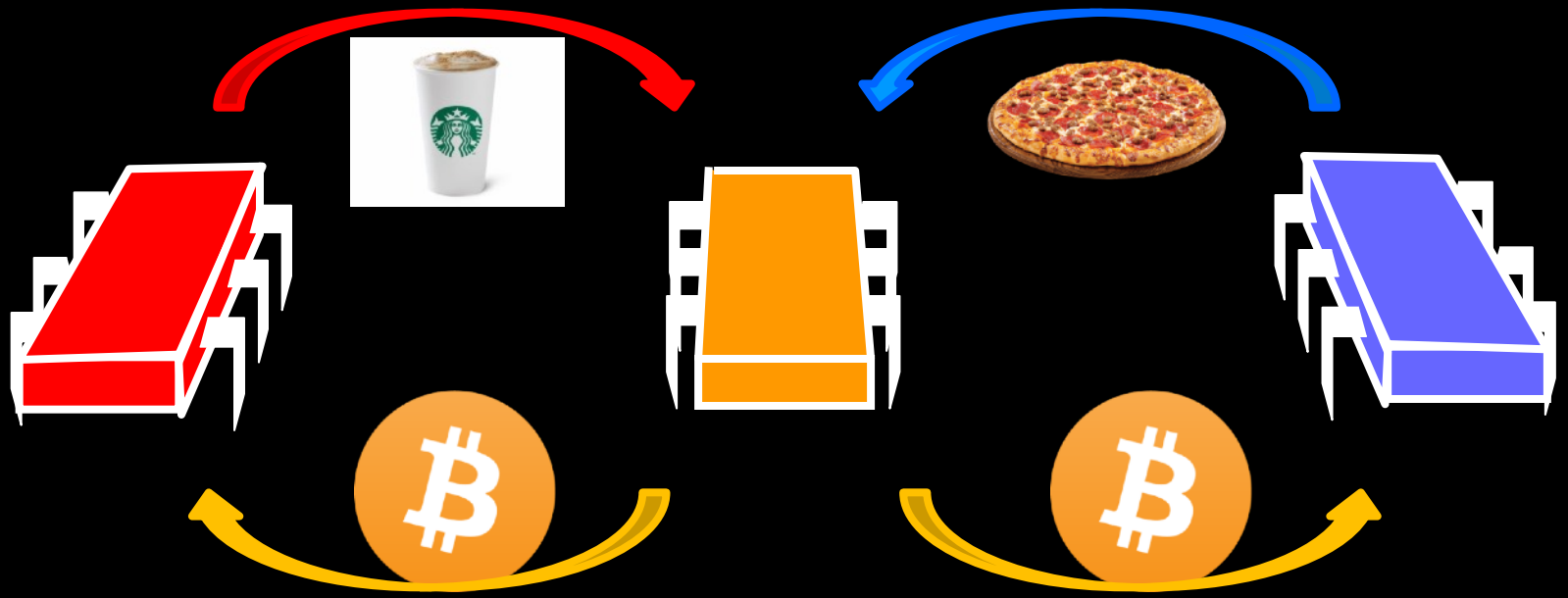
Abstraction: Distributed Ledger



Cash

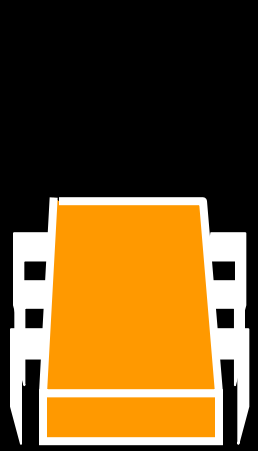
Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 7, 20X3	Cash sale	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000

Problem: Double Spending

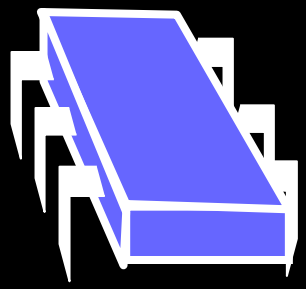


Old-School Solution

Nakamoto Solution



Cash				
Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		$ 7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 8, 20X3	Cash sale	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000



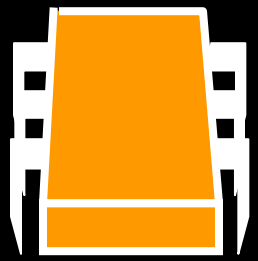
Public ledger

dbea25daf536

dbea25daf536

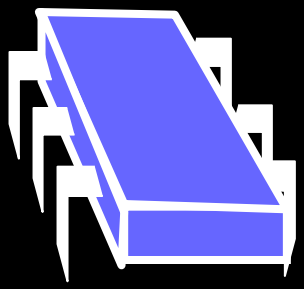
Nakamoto Solution

decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the



Cash

Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		$ 7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 8, 20X3	Cash sale	4,000		59,000
Jan. 8, 20X3	Paid bills		2,000	57,000
Jan. 10, 20X3	Paid tax		1,000	56,000
Jan. 12, 20X3	Collected receivable	7,000		63,000



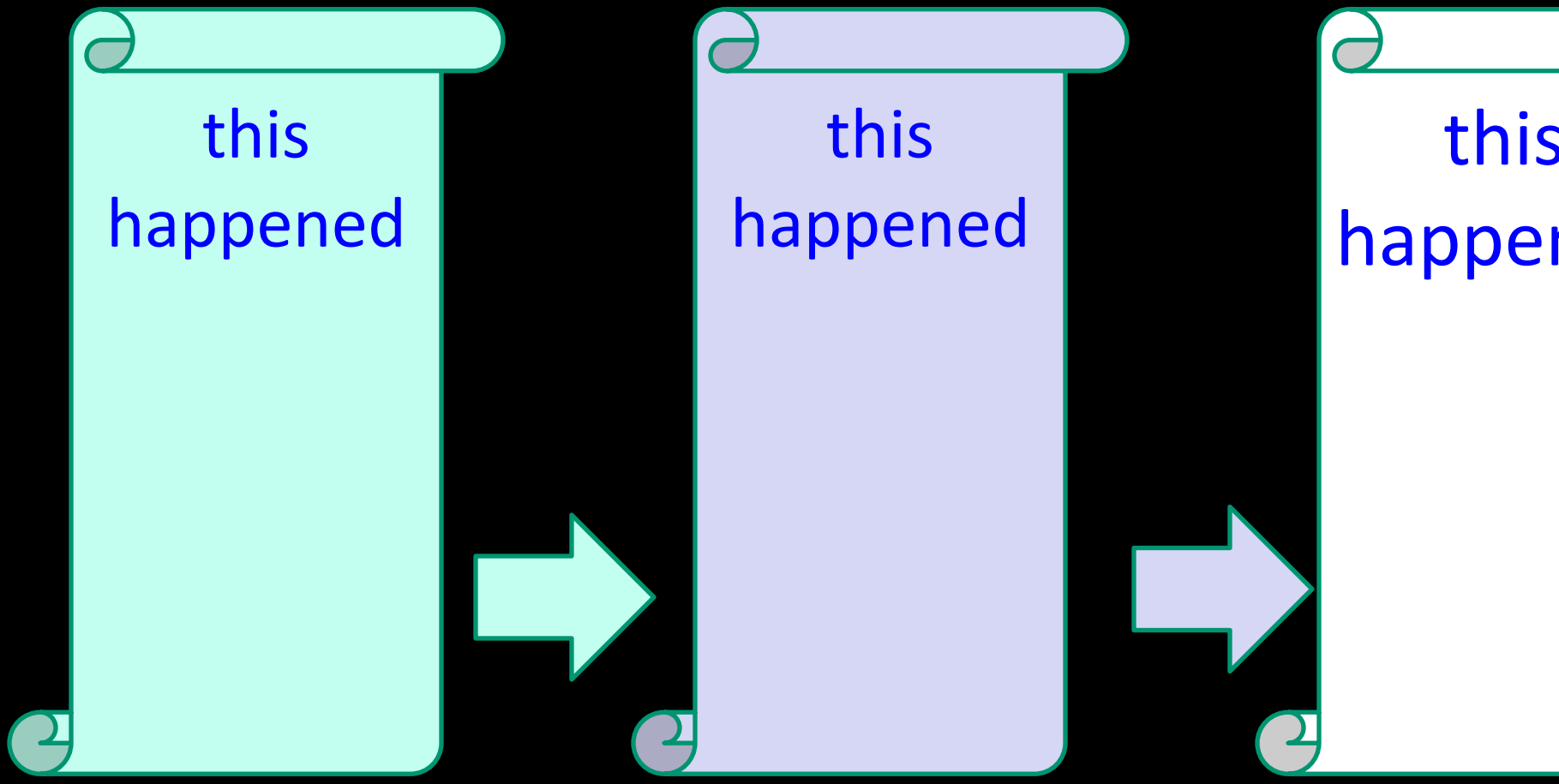
Public ledger

dbea25daf536

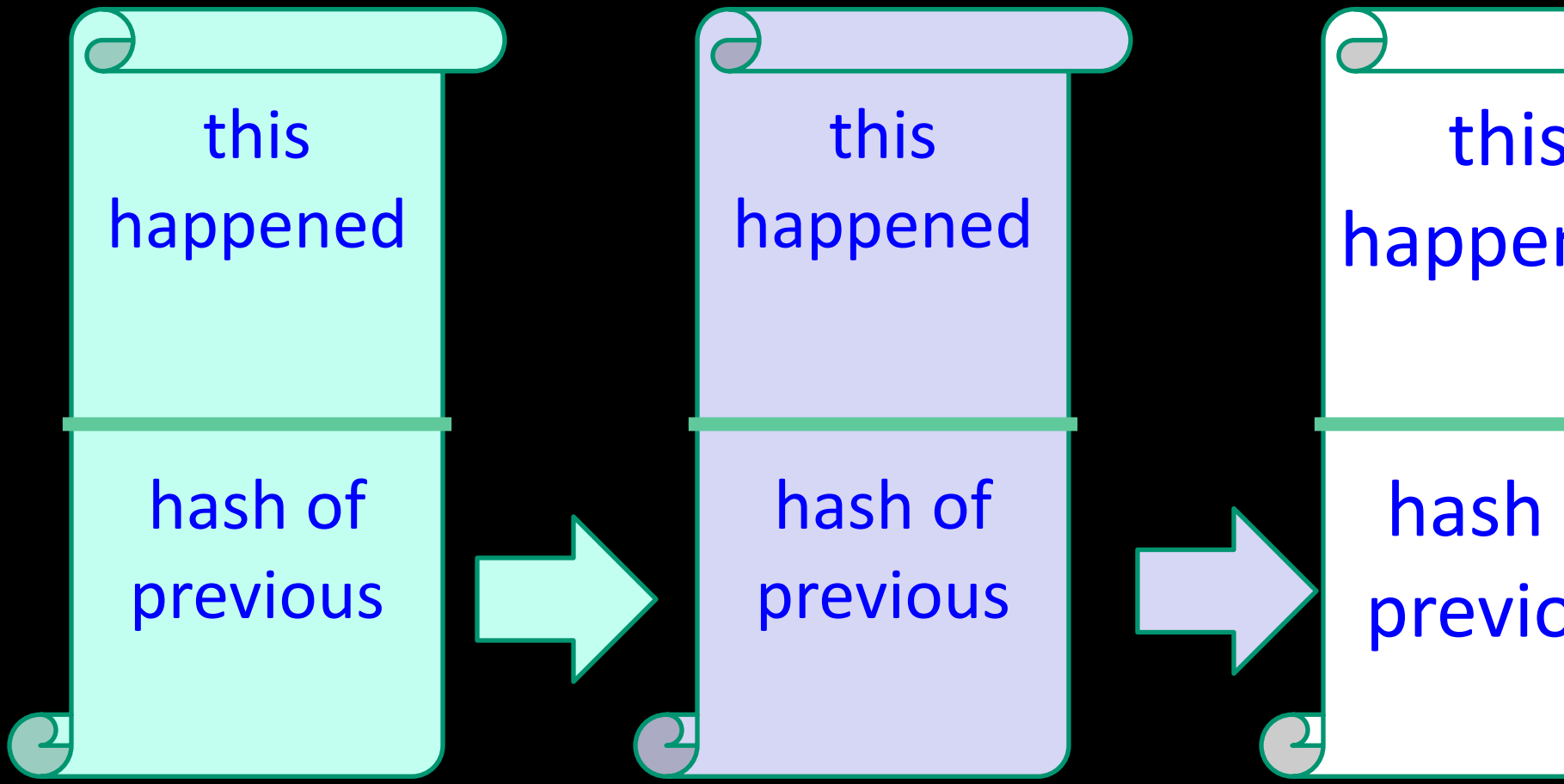
dbea25daf536

What is this Blockchain of which you speak?

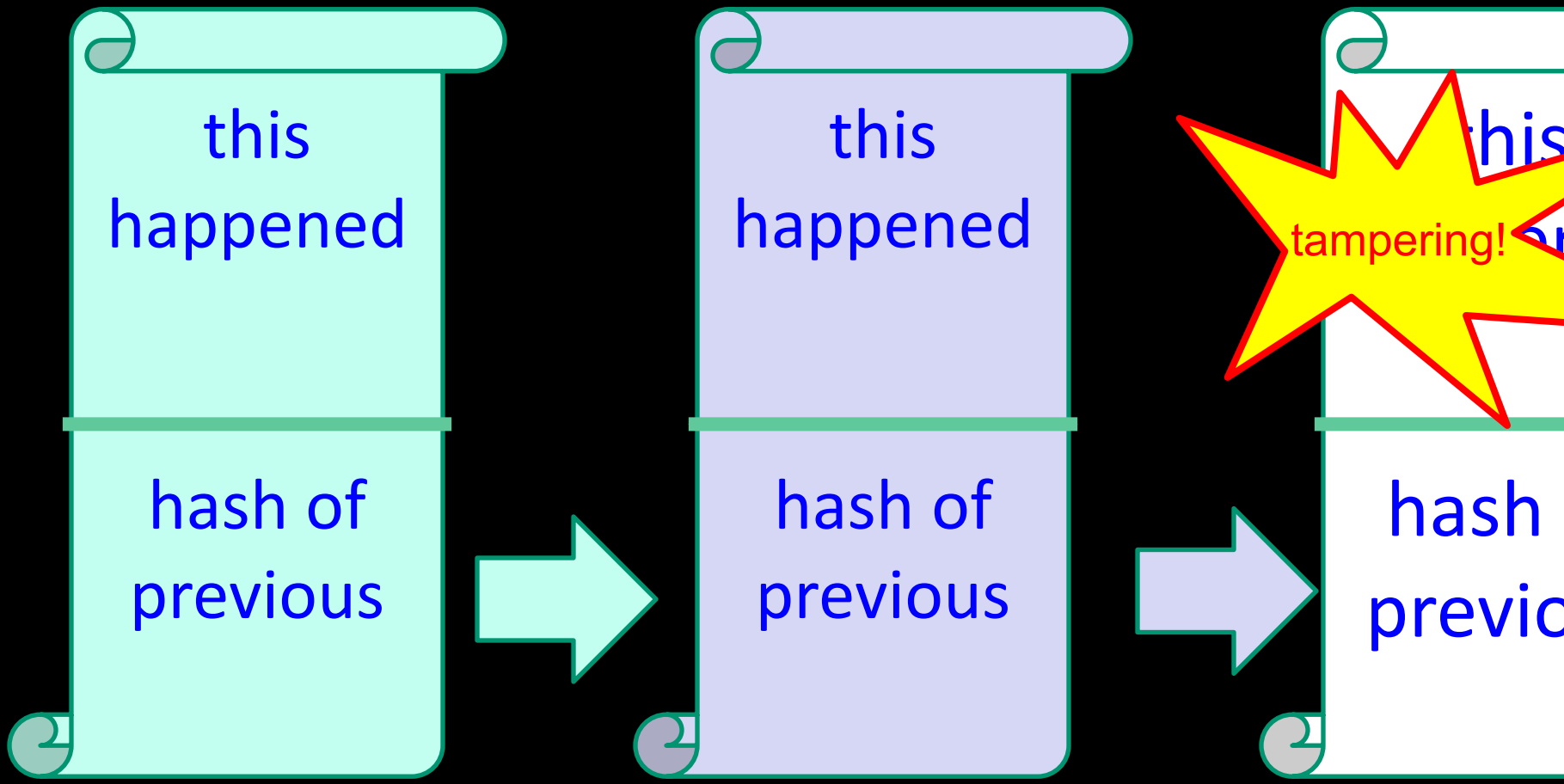
Literally



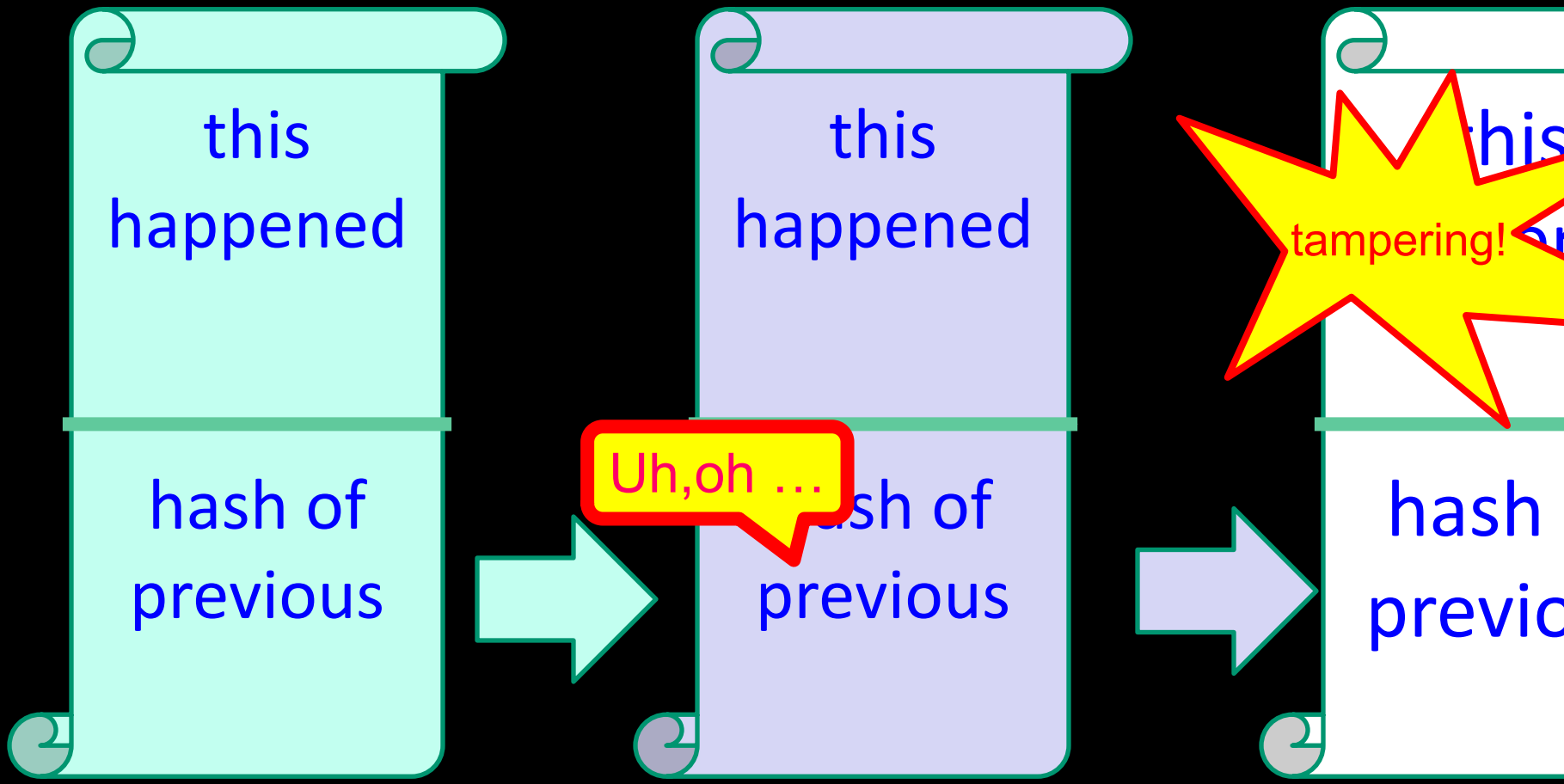
More Literally



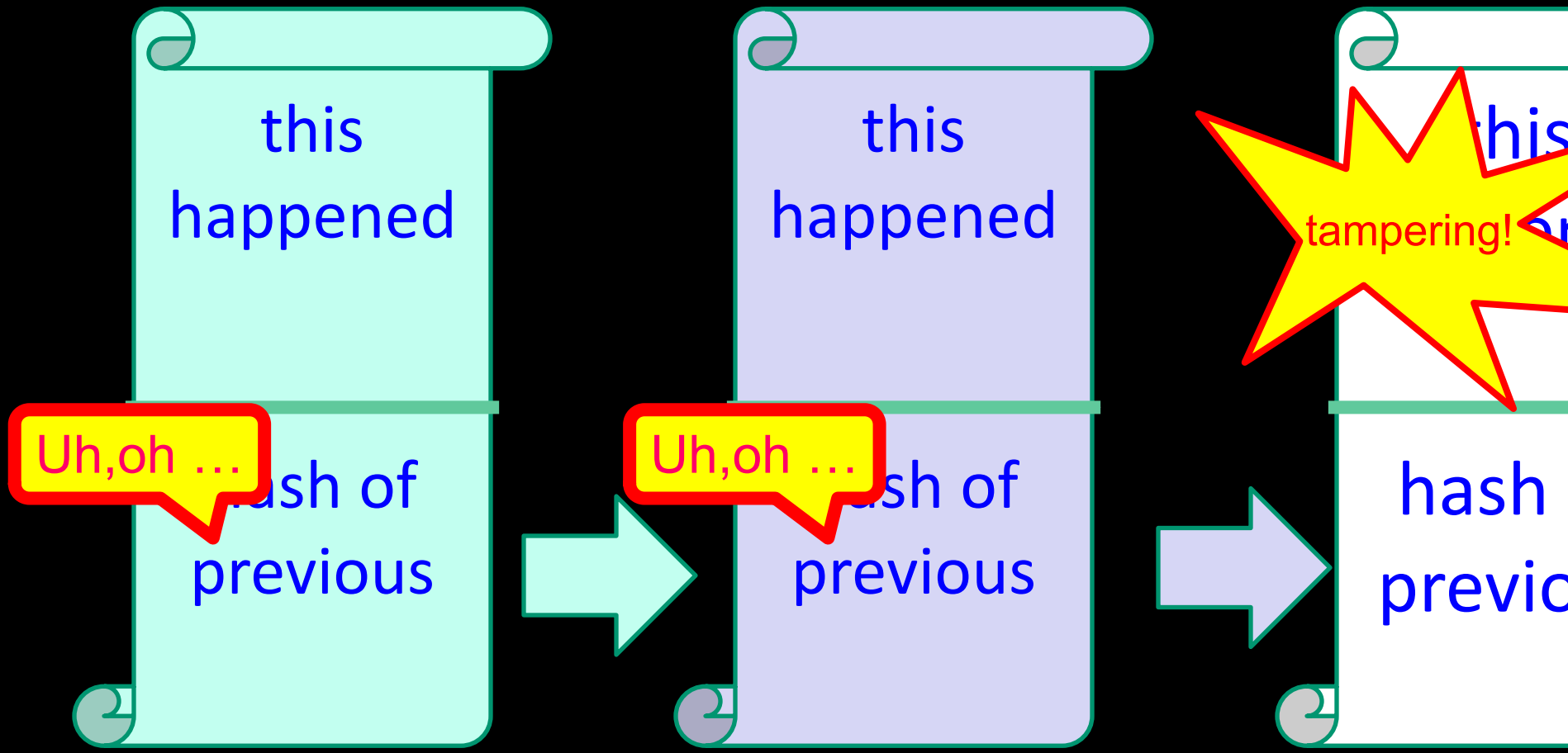
More Literally



More Literally



More Literally



MOTHERBOARD

BLOCKCHAINS | By Daniel Oberhaus | Aug 27 2018, 4:19pm

The World's Oldest Blockchain Has Been Hiding in the New York Times Since 1995

This really gives a new meaning to the "paper

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South Street
Morristown, N.J. 07960-1910

Abstract

The prospect of a world in which all text, audio, picture, and video documents are in digital form on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Hash document

Bellcore
445 South Street
Morristown, N.J. 07960-1910

Abstract

The prospect of a world in which all text, audio, picture, and video documents are in digital form on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South St.

Hash document

Cryptographic seal from Timestamp & Hash

Abstract

The prospect of a world in which all text, audio, picture, and video documents are in digital form on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South St

Hash document

Cryptographic seal from Timestamp & Hash

Store seals on server …

dig… on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South St

Hash document

Cryptographic seal from Timestamp & Hash

Store seals on server …

Every day, publish hash of seals to …

dig… …on easily modifiable media raises the issue of how to … audio, picture, and video documents are in
was created or last changed. Th… …en a document
pro… …e medium. We
tha… …documents so
the … …forward-date his document, even with
… time-stamping service. Our procedures maintain complete privacy of the
documents themselves, and require no record-keeping by the time-stamping service.

The New York Times

NOTICES & LOST AND FOUND

(5100-5102)

Universal Registry Entries:
Zone 2 -

dS8492cgVOFAoP9kvE1XzMOrQ
HgEwzkVbVafNvIkUz99qvq8/ME
p5v9EFSG8XxzMBaIGQQ==

Zone 3 -

JnFCg+HCmvhj8GmmUP7VZna71
NgZup/RfuKUQNzCHWXMuqLK
durxHQV5pSHLqBGPRIv+mg==

These base64-encoded values represent the combined fingerprints of all digital records notarized by Surety between 2009-06-03Z 2009-06-09Z.

www.surety.com 571-748-5800

The New York Times

NOTICES &
LOST AND
FOUND

Vitalik Non-giver of Ether
@VitalikButerin

Following

Replying to @ofnumbers @ittaia @ittayeyal

The more realistic attack vector would be to make fake newspapers with a different chain of hashes and circulate them more widely. Still very difficult though :)

4:48 AM - 27 Aug 2018

9 Retweets **55** Likes

8 9 55

The Bitcoin Protocol (simplified)

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Introduction

...net has come to rely almost exclusively on financial institutions serving as ...lectronic payments. While the system works well enough for ...the inherent weaknesses of the trust based model. ...possible, since financial institutions cannot ...transaction costs, limiting the ...small casual transactions, ...nts for non-

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Illustrate ideas using the original paper

Introduction

Precursors

Precursors

Pricing via Processing or Combatting Junk Mail*

Cynthia Dwork *

Moni Naor †

Abstract

We present a computational technique for combatting junk mail, in particular, and controlling access to a shared resource, in general. The main idea is to require a user to compute a moderately hard, but not intractable, function in order to gain access to the resource, thus preventing frivolous use. To this end we suggest *pricing functions*, based on, respectively, extracting square roots modulo a prime, the Fiat-Shamir signature scheme, and the Ong-

Precursors

Hashcash - A Denial of Service Counter-Measure

Adam Back
e-mail: adam@cypherspace.org

1st August 2002

Abstract

Hashcash was originally proposed as a mechanism to throttle systematic abuse of un-metered internet resources such as email, and anonymous remailers in May 1997. Five years on, this paper captures in one place the various applications, improvements suggested and related subsequent publications, and describes initial experience from experiments using hashcash.

The *hashcash* CPU cost-function computes a token which can be used as a proof-of-work. Interactive and non-interactive variants of cost-functions can be constructed which can be used in situations where the server can issue a challenge (connection oriented interactive protocol), and where it can not (where the communication is store–and–forward, or packet oriented) respectively.

Key Words: hashcash, cost-functions

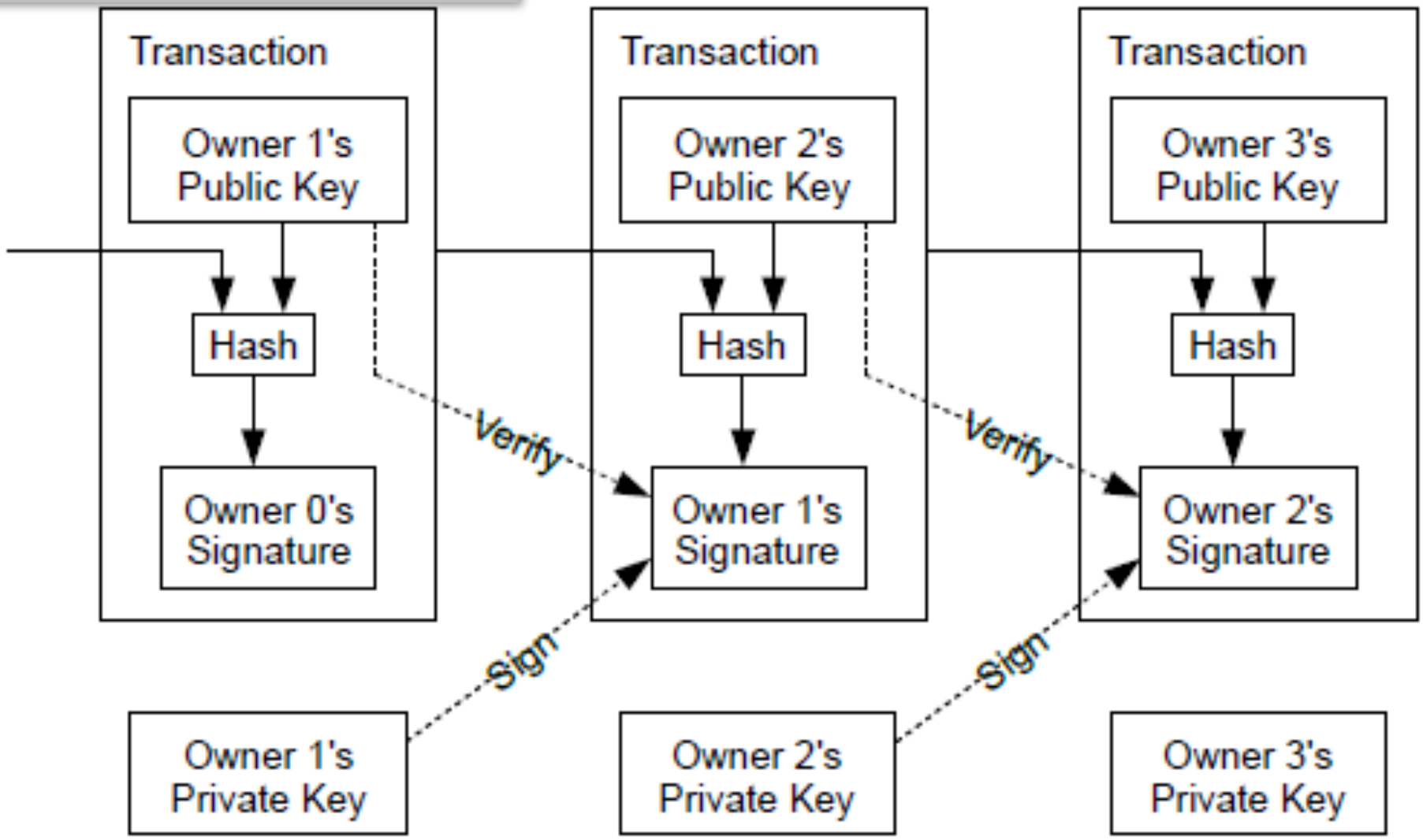
1 Introduction

Hashcash [1] was originally proposed as a mechanism to throttle systematic abuse of un-metered internet resources
mail, and anonymous remailers in May 1997. Five years on, this paper captures in one place the various
ments suggested and related subsequent publications, and describes initial experience from ex-

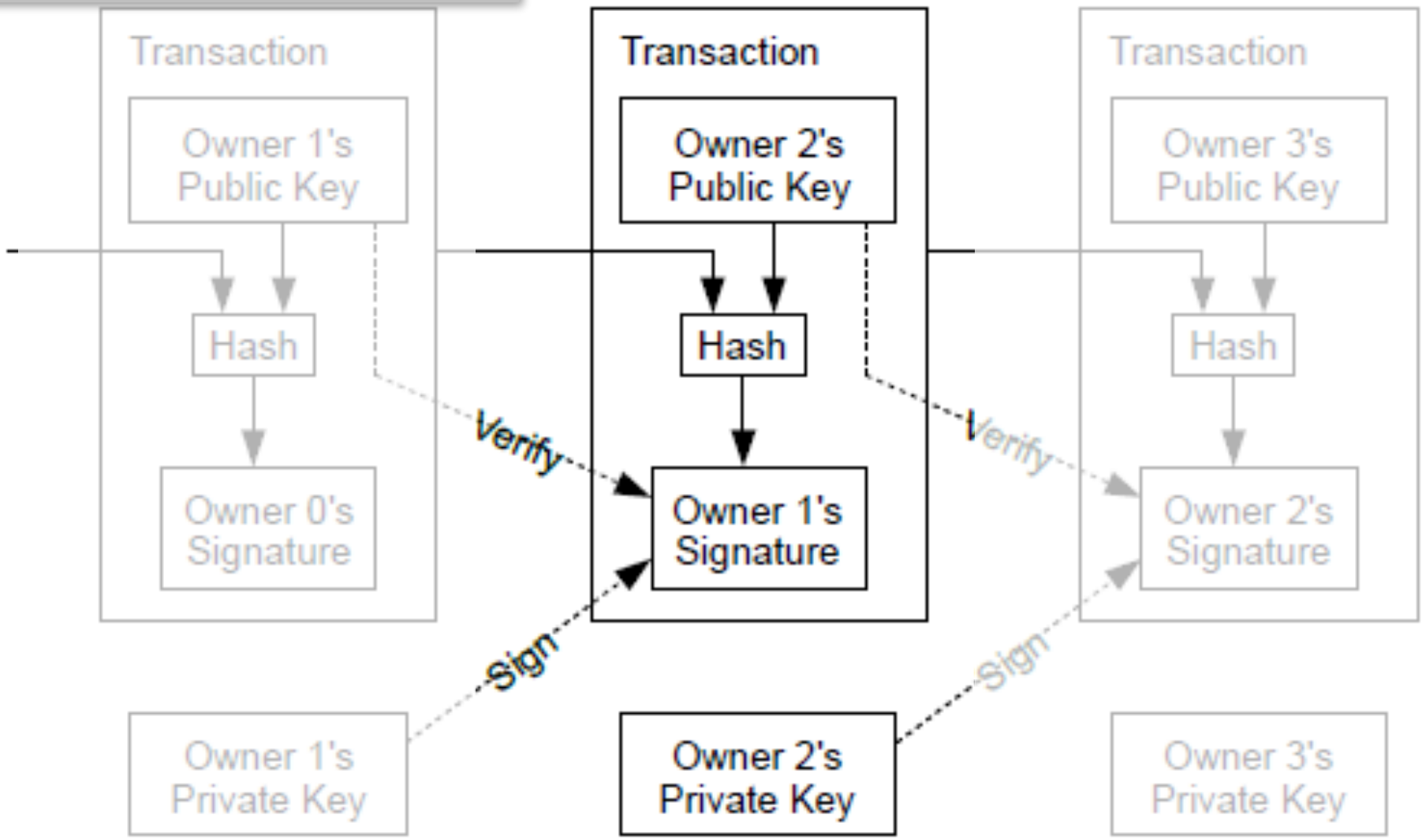
a token which can be used as a proof-of-work. Interactive and non-
which can be used in situations where the server can issue
ot (where the communication is store–and-

nd Naor in [2] wh
for co

How it works

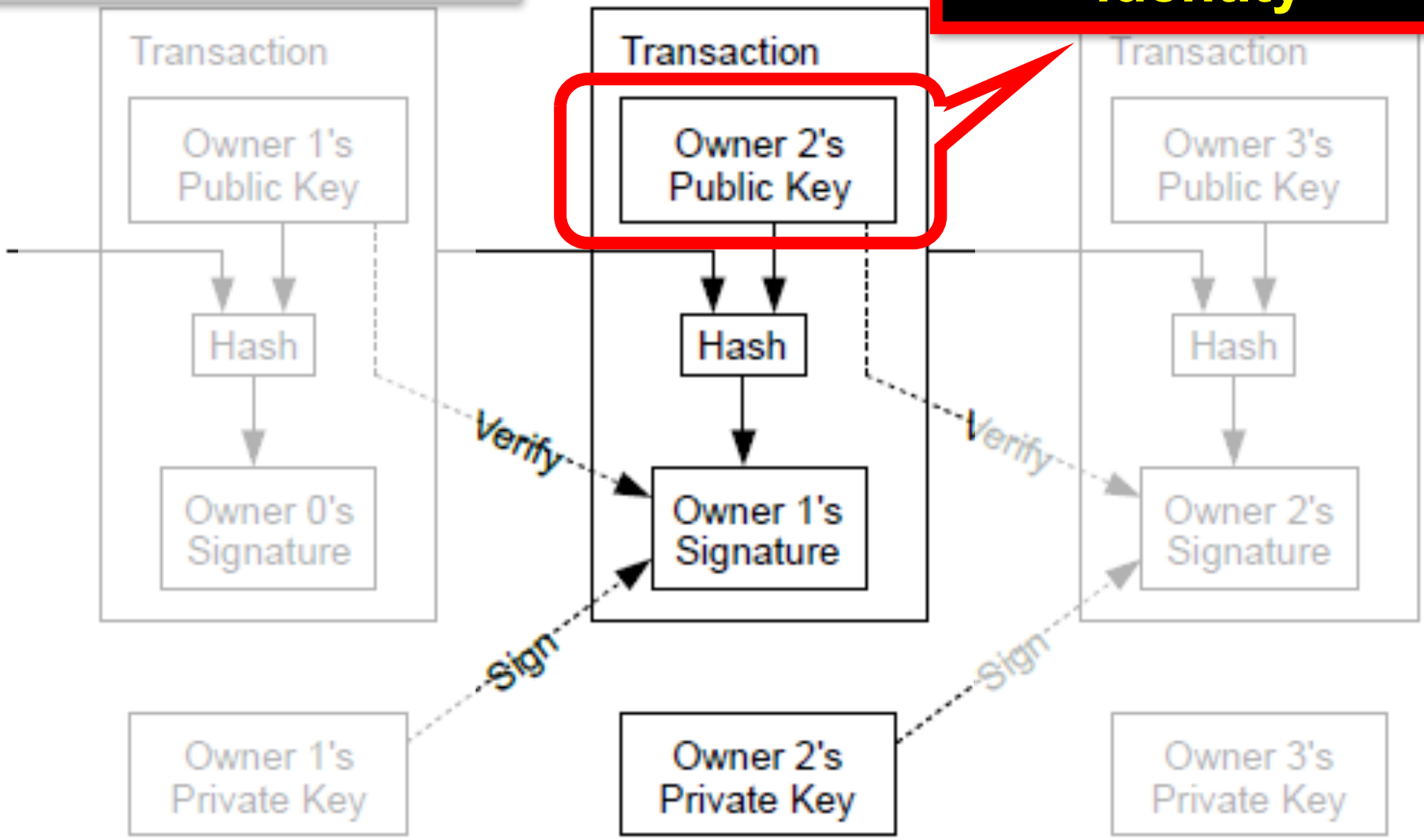

Transaction
Owner 1's Public Key
Hash
Owner 0's Signature
Transaction
Owner 2's Public Key
Hash
Owner 1's Signature
Transaction
Owner 3's Public Key
Hash
Owner 2's Signature
Verify
Verify
Sign
Sign
Owner 1's Private Key
Owner 2's Private Key
Owner 3's Private Key

How it works


Transaction
Owner 1's Public Key
Hash
Owner 0's Signature
Transaction
Owner 2's Public Key
Hash
Owner 1's Signature
Transaction
Owner 3's Public Key
Hash
Owner 2's Signature
Verify
Verify
Sign
Sign
Owner 1's Private Key
Owner 2's Private Key
Owner 3's Private Key

How it works

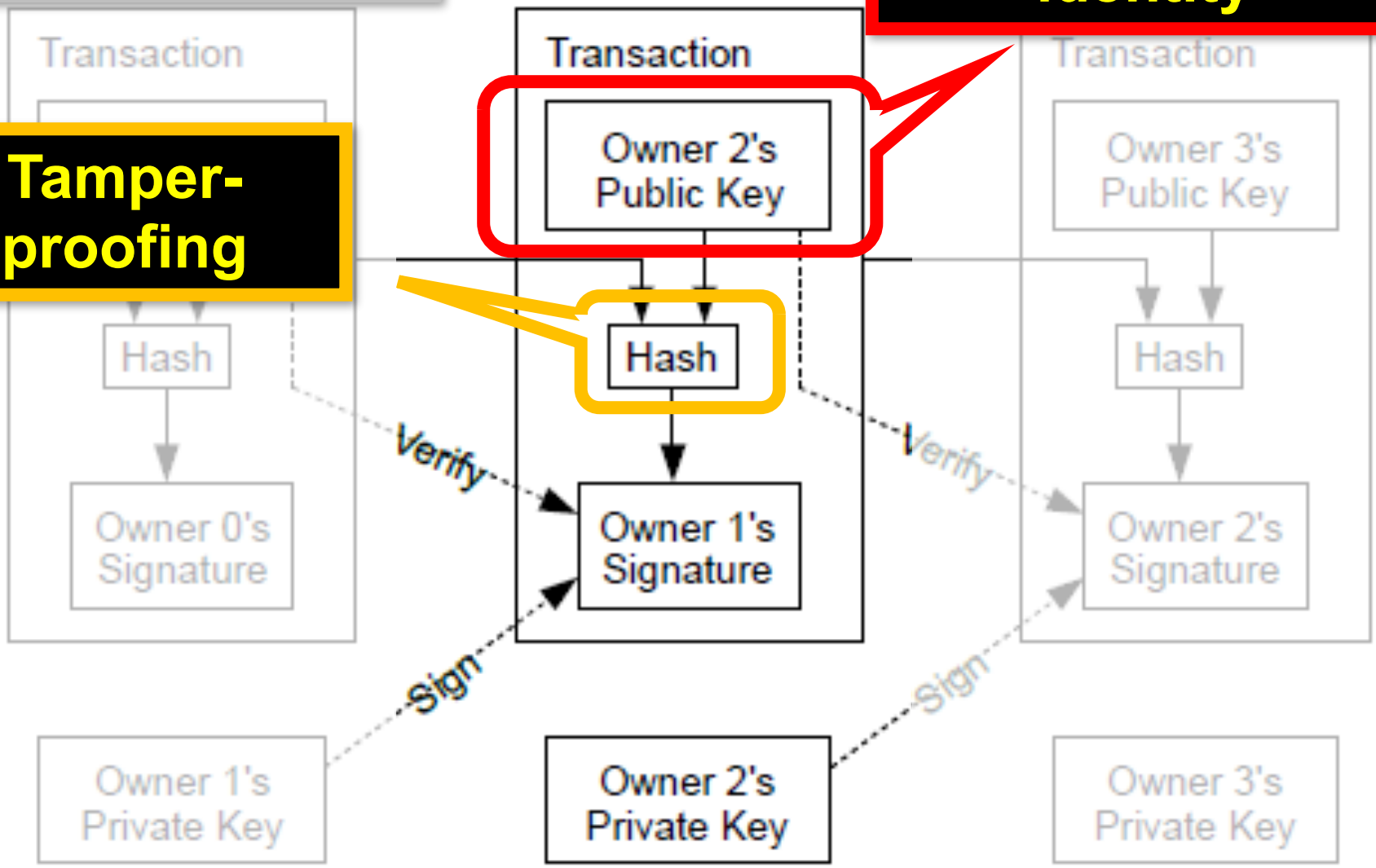
New owner's identity



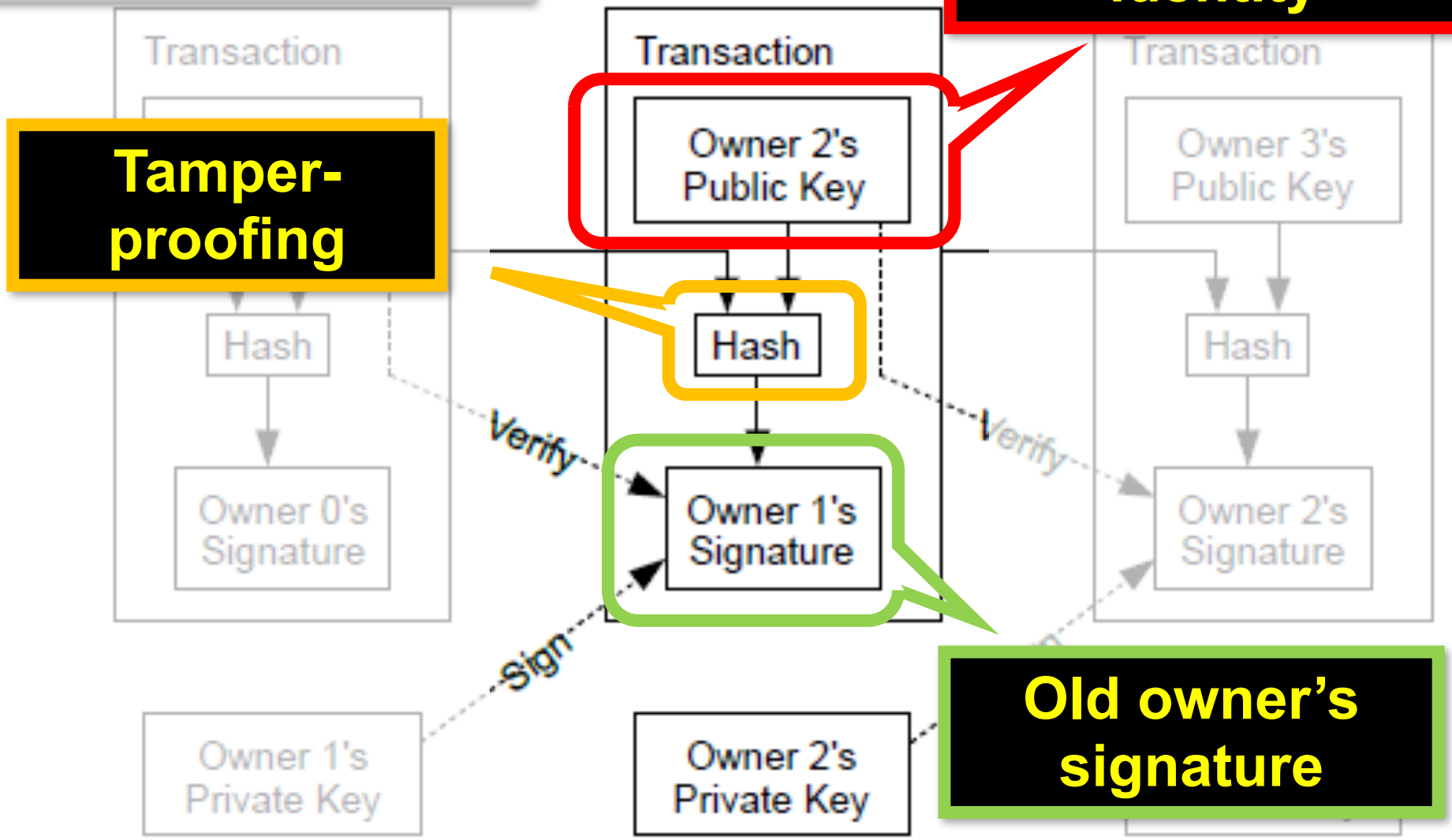
How it works

New owner's identity

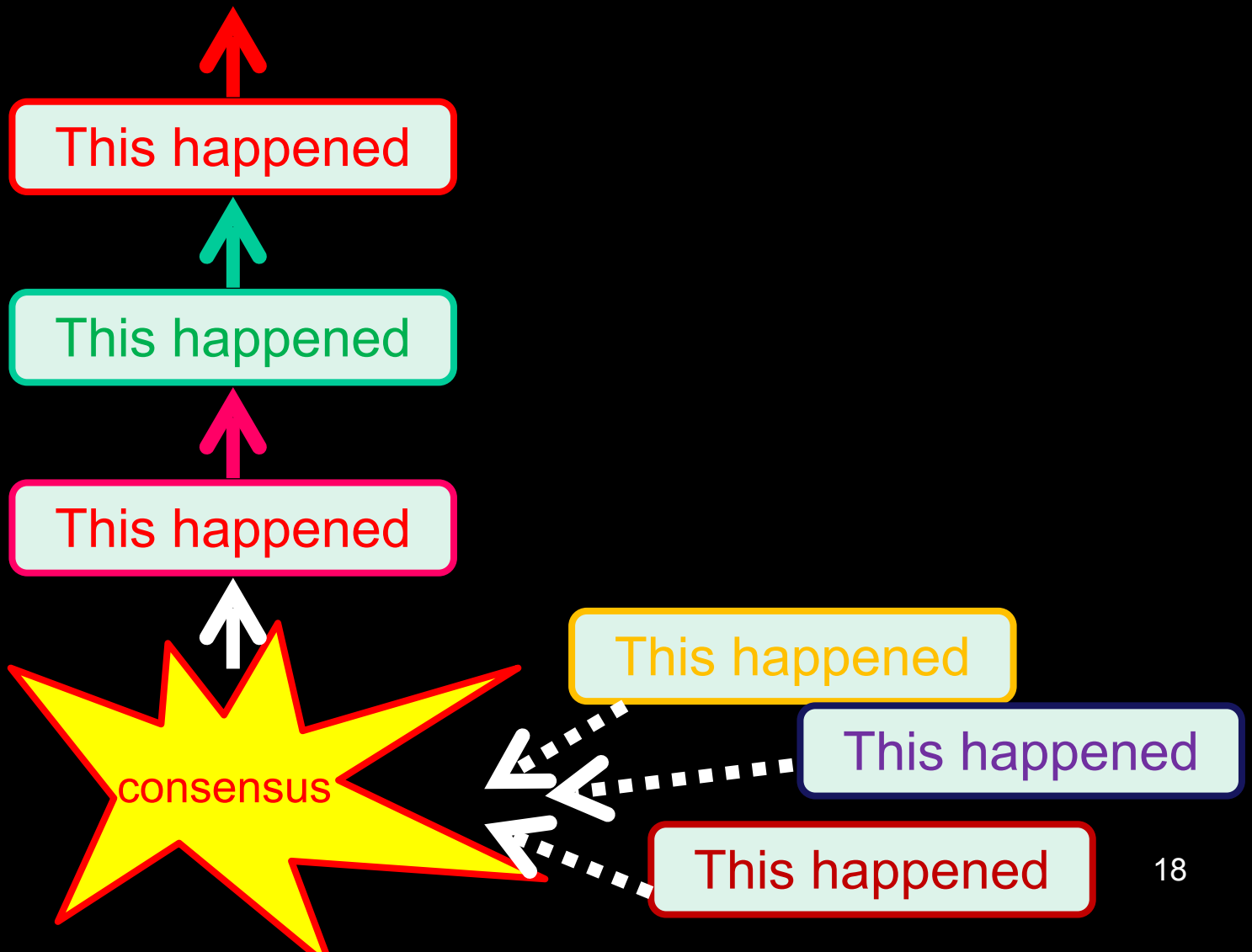
Tamper-proofing



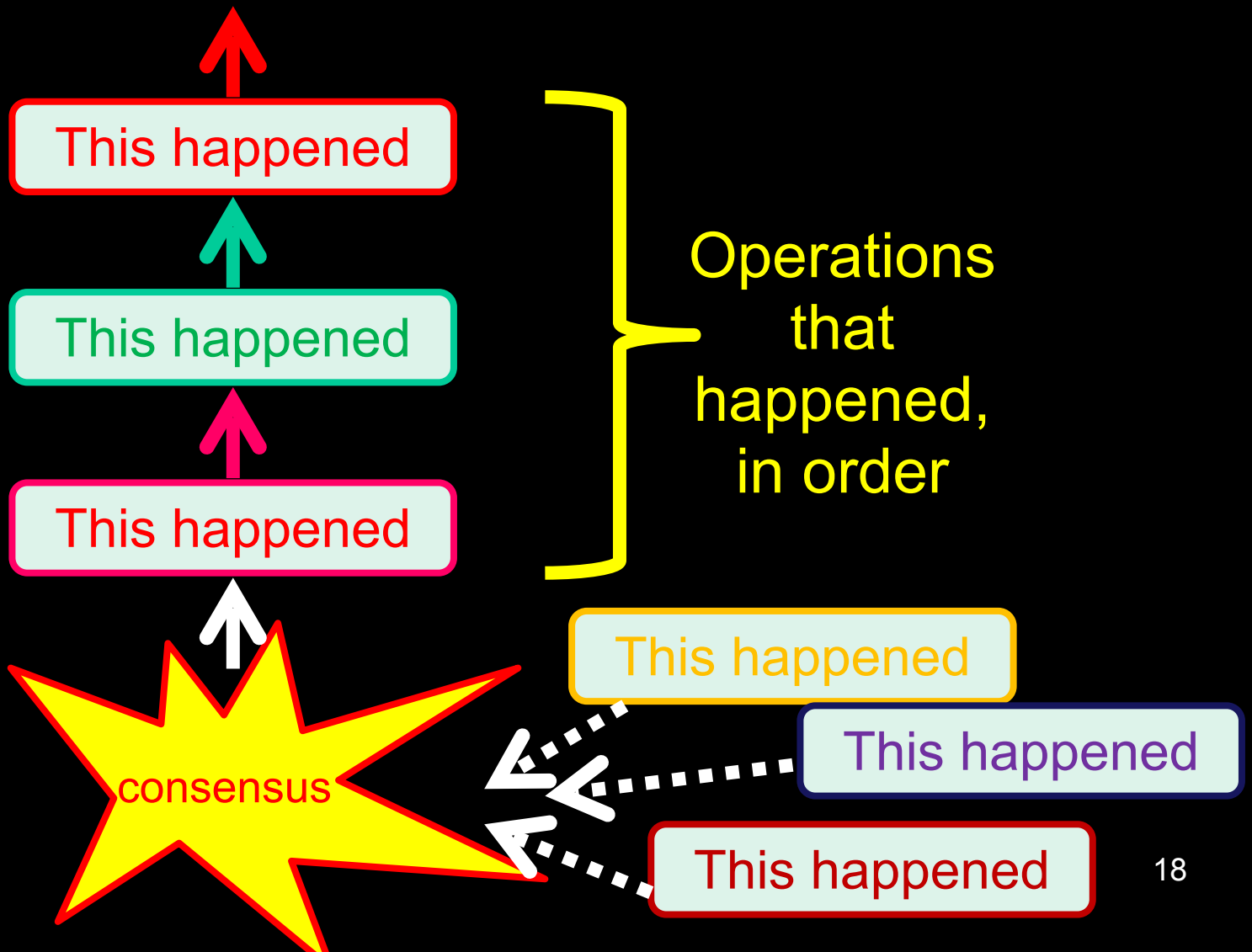
How it works


New owner's identity
Tamper-proofing
Old owner's signature
Transaction
Owner 2's Public Key
Hash
Owner 1's Signature
Transaction
Owner 3's Public Key
Hash
Owner 2's Signature
Transaction
Hash
Owner 0's Signature
Verify
Verify
Sign
Owner 1's Private Key
Owner 2's Private Key

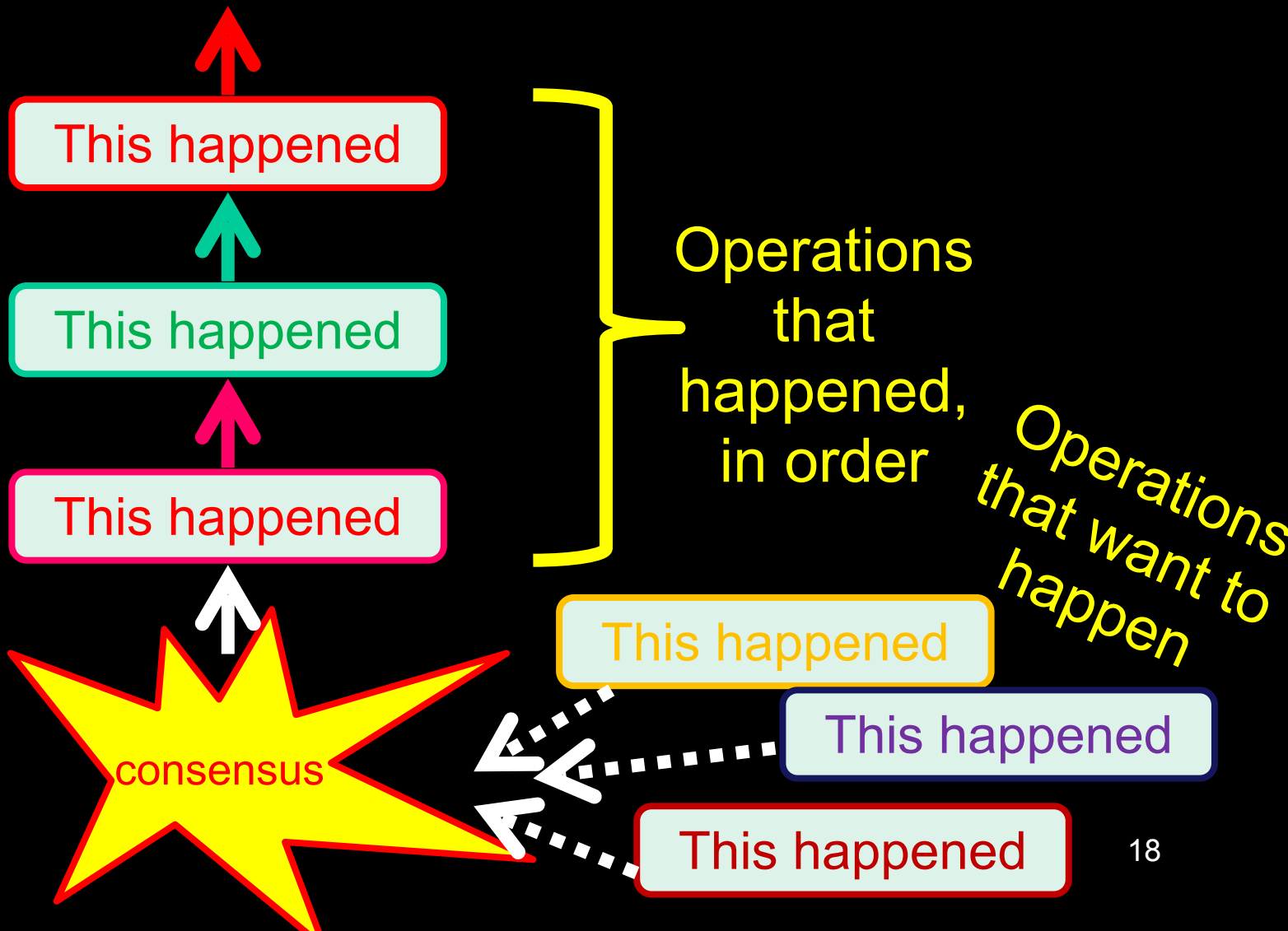
Blockchain Construction (simplified)



Blockchain Construction (simplified)



Blockchain Construction (simplified)

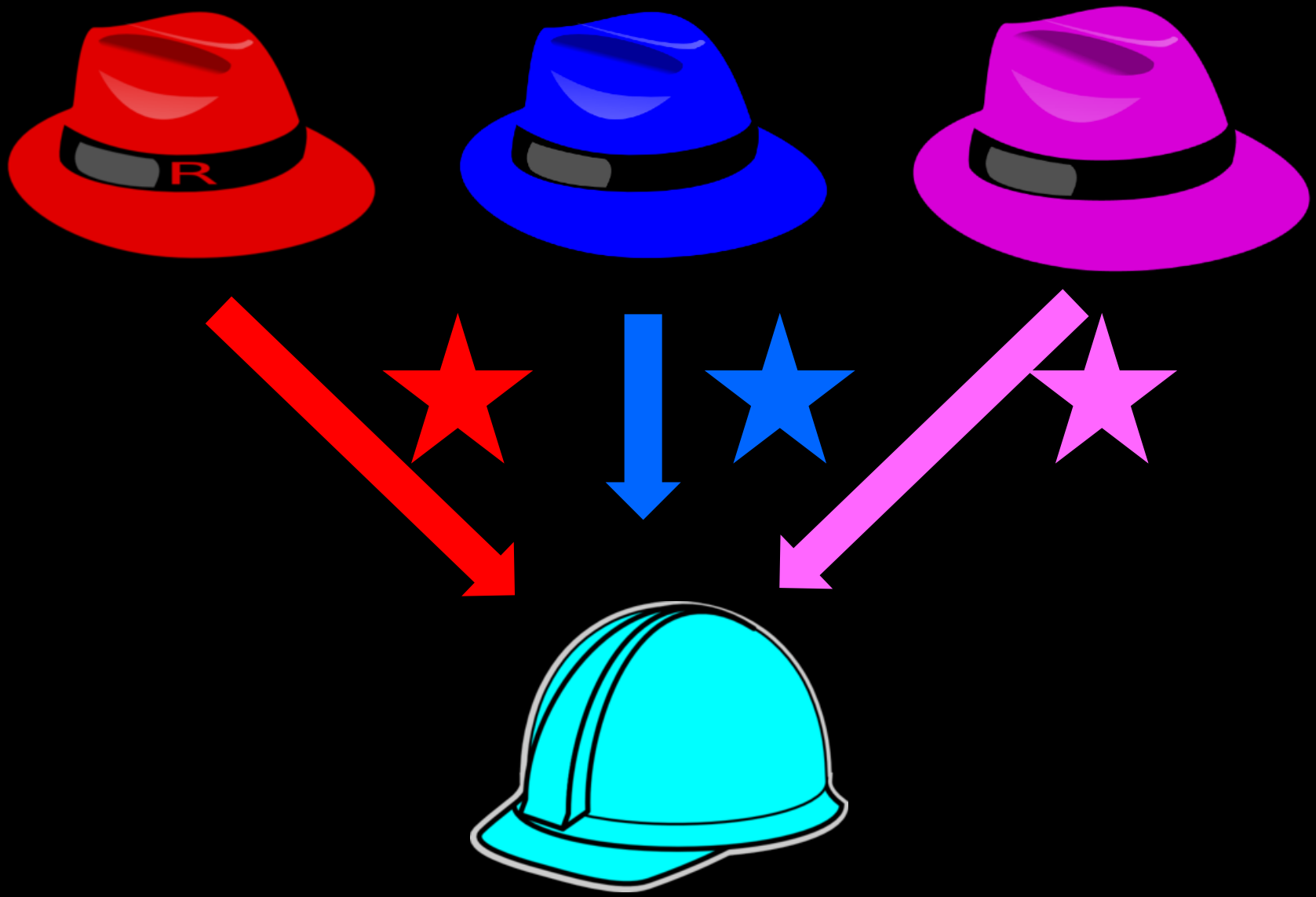


Clients ....

Clients ....

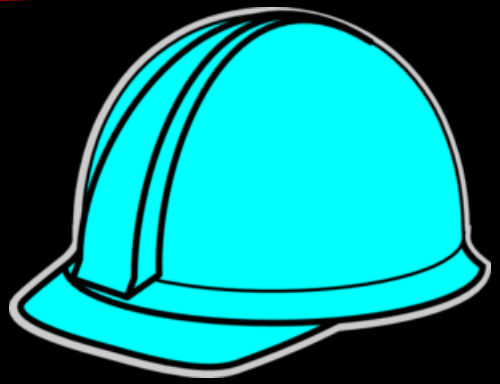
Miners ...

Clients ....



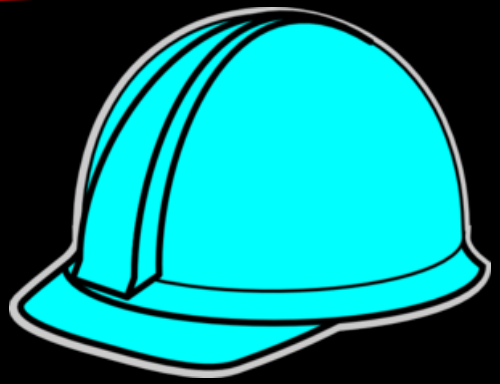
Clients ....

send transactions …



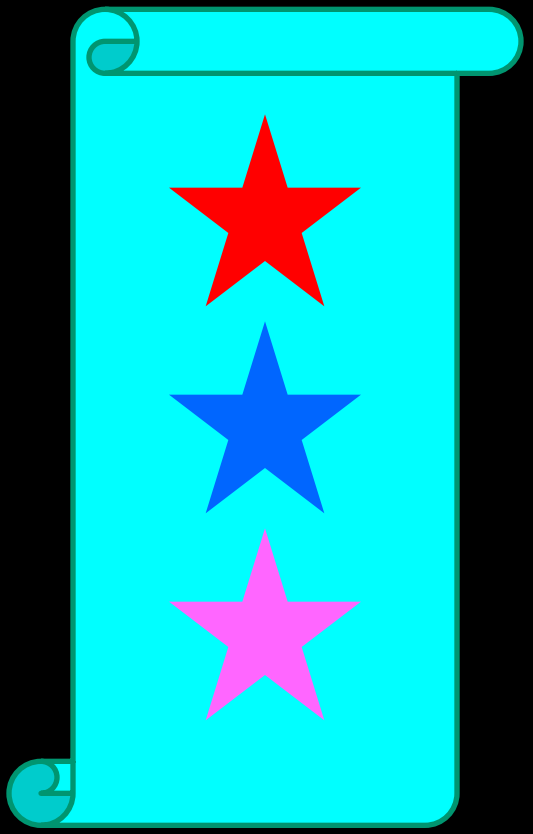
Clients ....

send transactions …

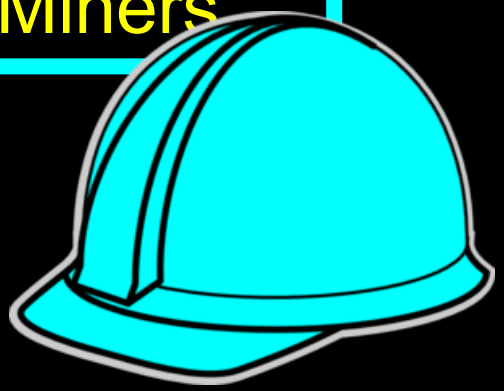


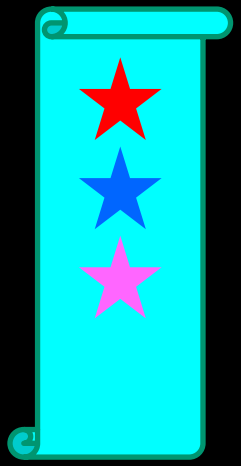
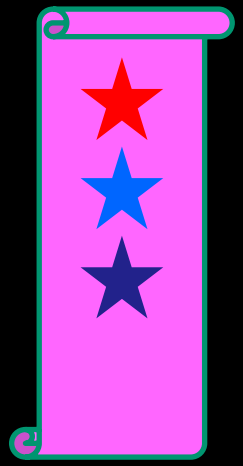
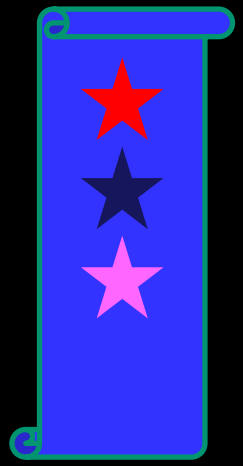
to miners.

Miner batches transactions in blocks

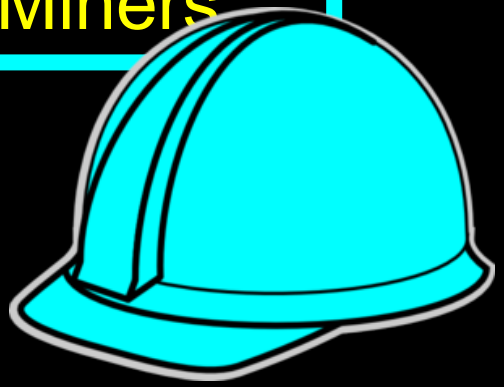


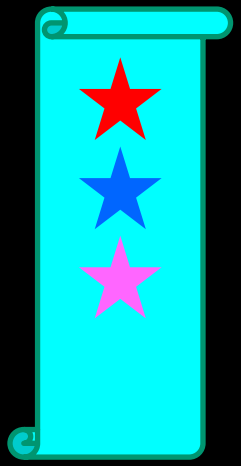
Miners



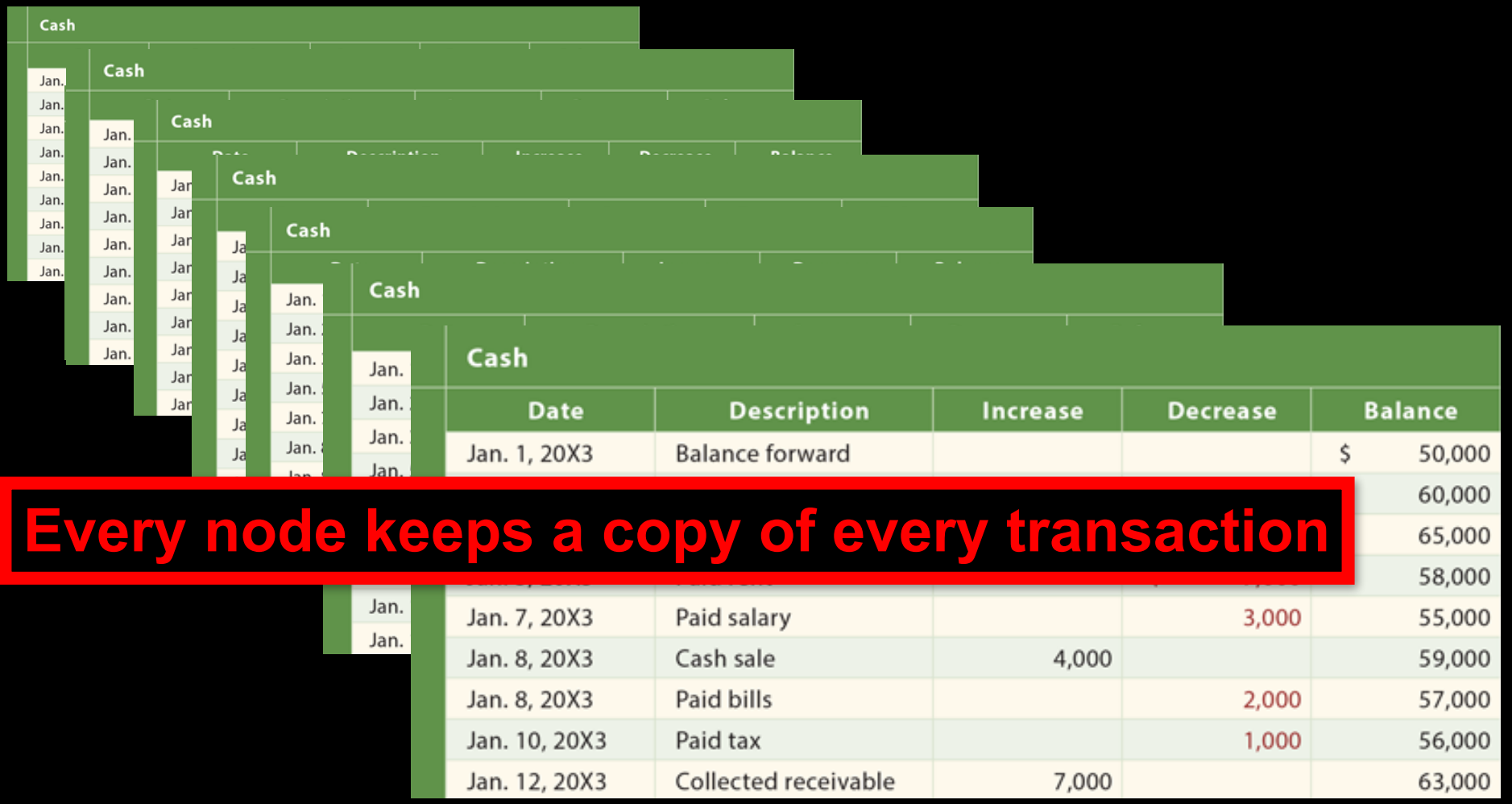




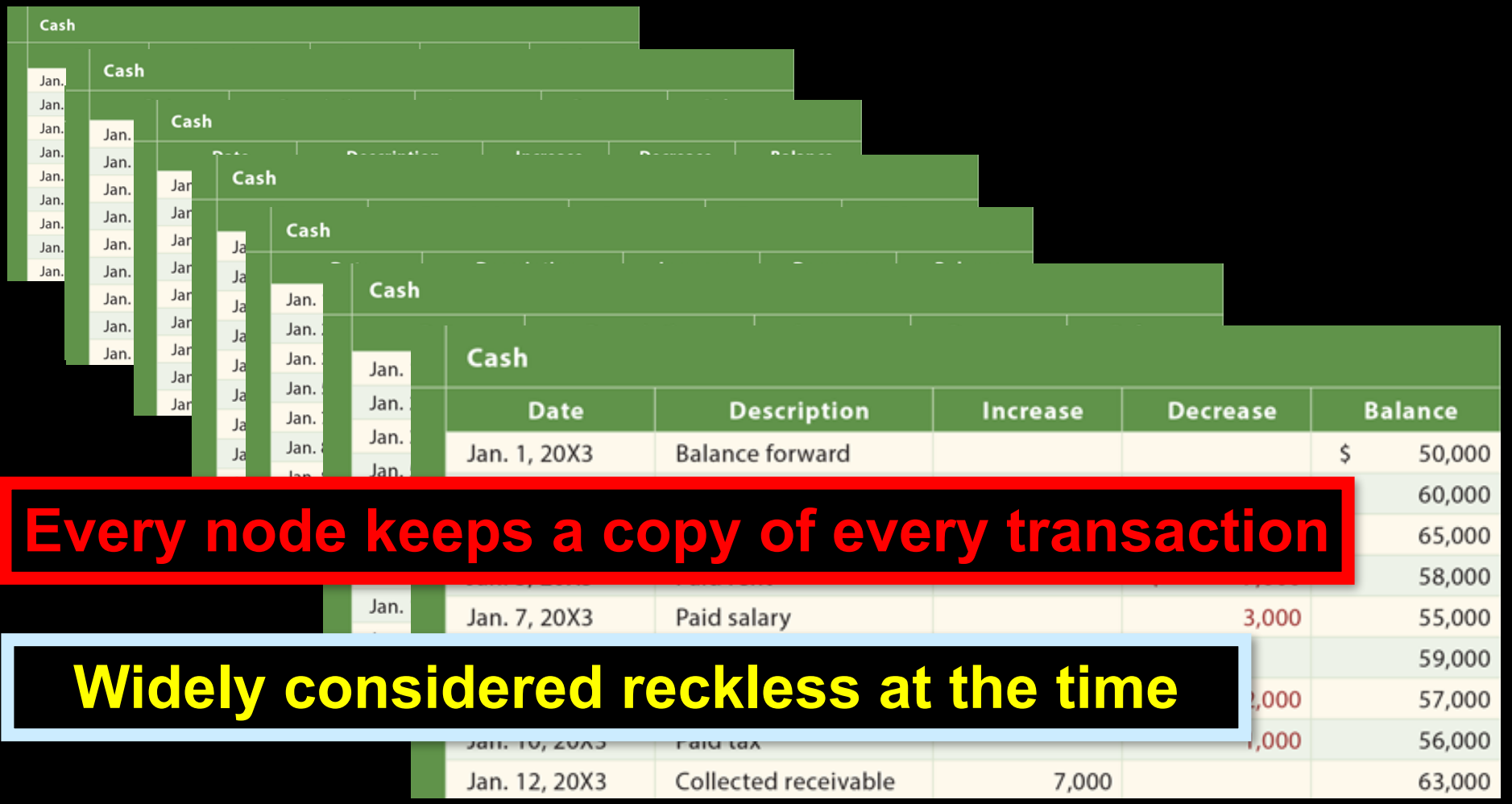
Miners

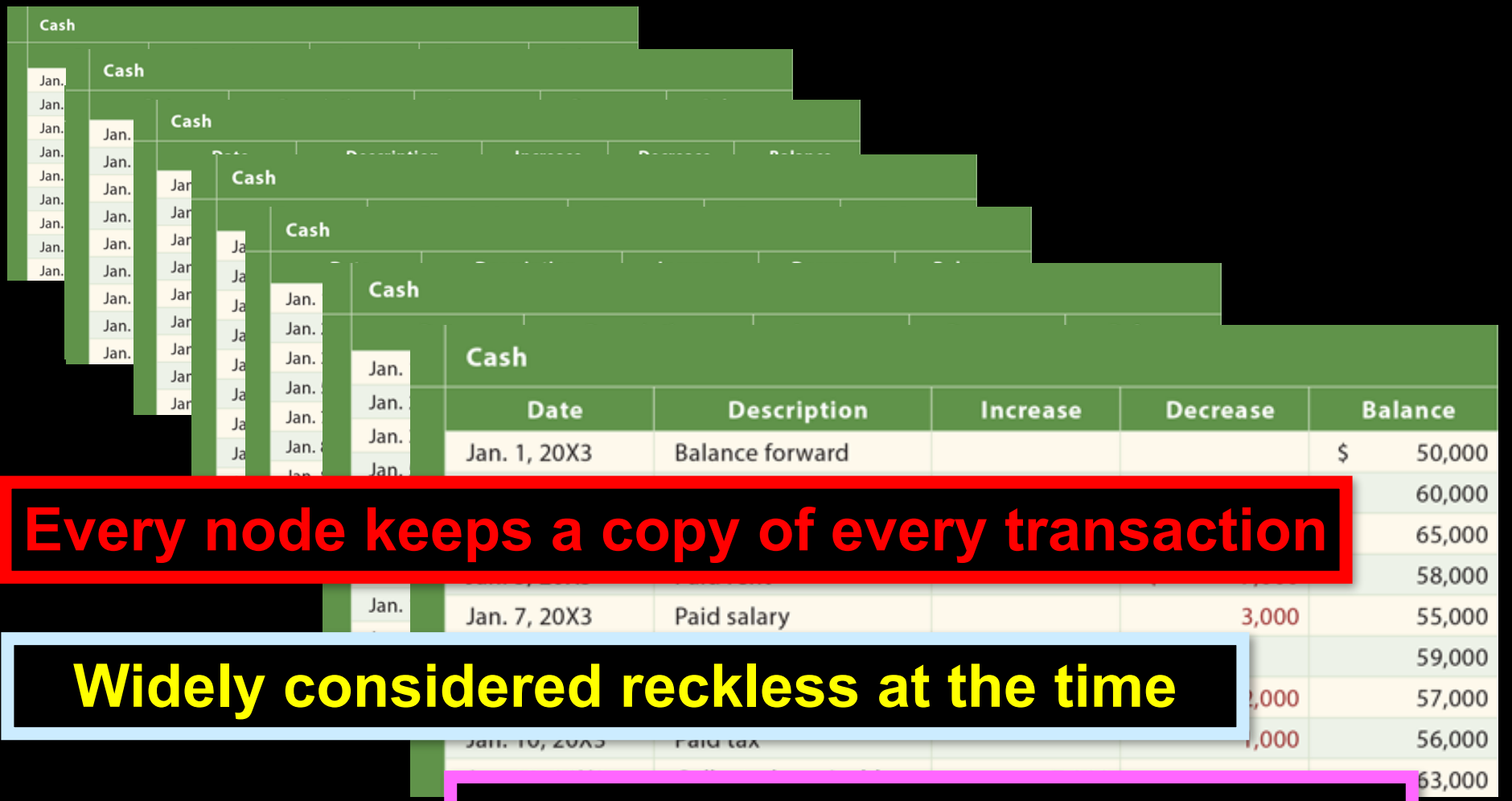


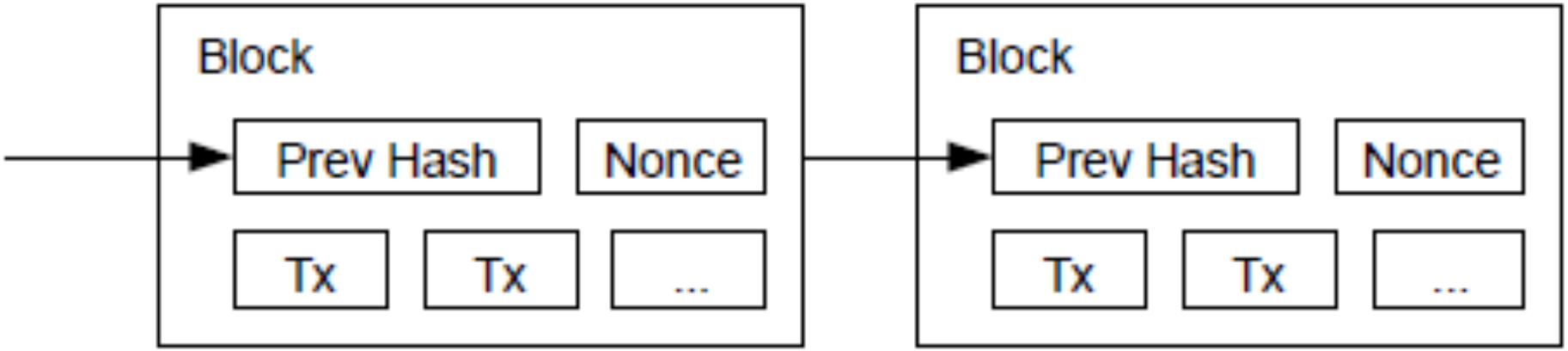


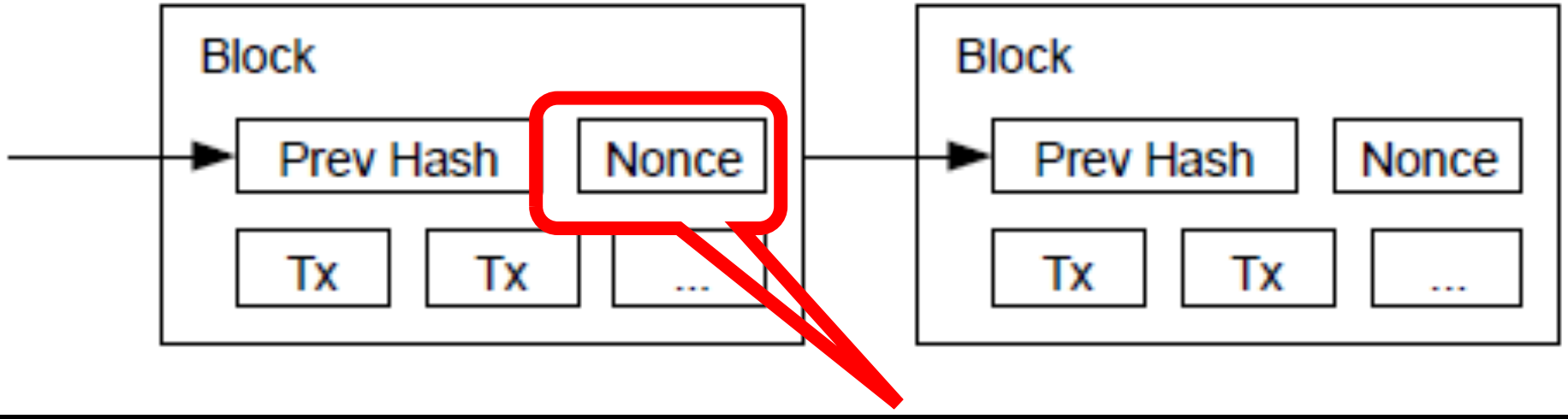
do consensus to pick one block …


Cash
Cash
Cash
Cash
Cash
Cash
Cash
Date
Description
Increase
Decrease
Balance
Jan. 1, 20X3
Balance forward
$ 50,000
60,000
65,000
58,000
Jan. 7, 20X3
Paid salary
3,000
55,000
Jan. 8, 20X3
Cash sale
4,000
59,000
Jan. 8, 20X3
Paid bills
2,000
57,000
Jan. 10, 20X3
Paid tax
1,000
56,000
Jan. 12, 20X3
Collected receivable
7,000
63,000
Every node keeps a copy of every transaction

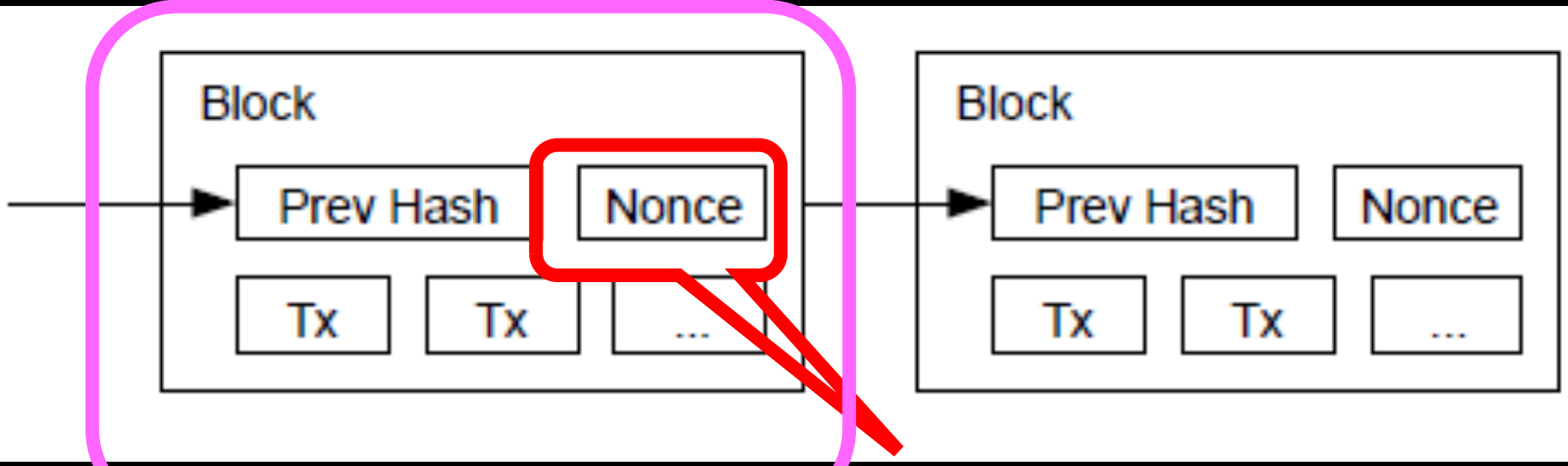

Cash
Date
Description
Increase
Decrease
Balance
Jan. 1, 20X3
Balance forward
$ 50,000
60,000
65,000
58,000
Jan. 7, 20X3
Paid salary
3,000
55,000
59,000
57,000
56,000
Jan. 12, 20X3
Collected receivable
7,000
63,000
Every node keeps a copy of every transaction
Widely considered reckless at the time


Cash
Cash
Cash
Cash
Cash
Cash
Cash
Date
Description
Increase
Decrease
Balance
Jan. 1, 20X3
Balance forward
$ 50,000
60,000
65,000
58,000
Jan. 7, 20X3
Paid salary
3,000
55,000
59,000
57,000
56,000
63,000
Every node keeps a copy of every transaction
Widely considered reckless at the time
Still a scalability issue


Block
Prev Hash
Nonce
Tx
Tx
...
Block
Prev Hash
Nonce
Tx
Tx
...



Find a value to put here …



Find a value to put here …

To give this hash *k* leading 0s

Nakamoto Consensus in One Line

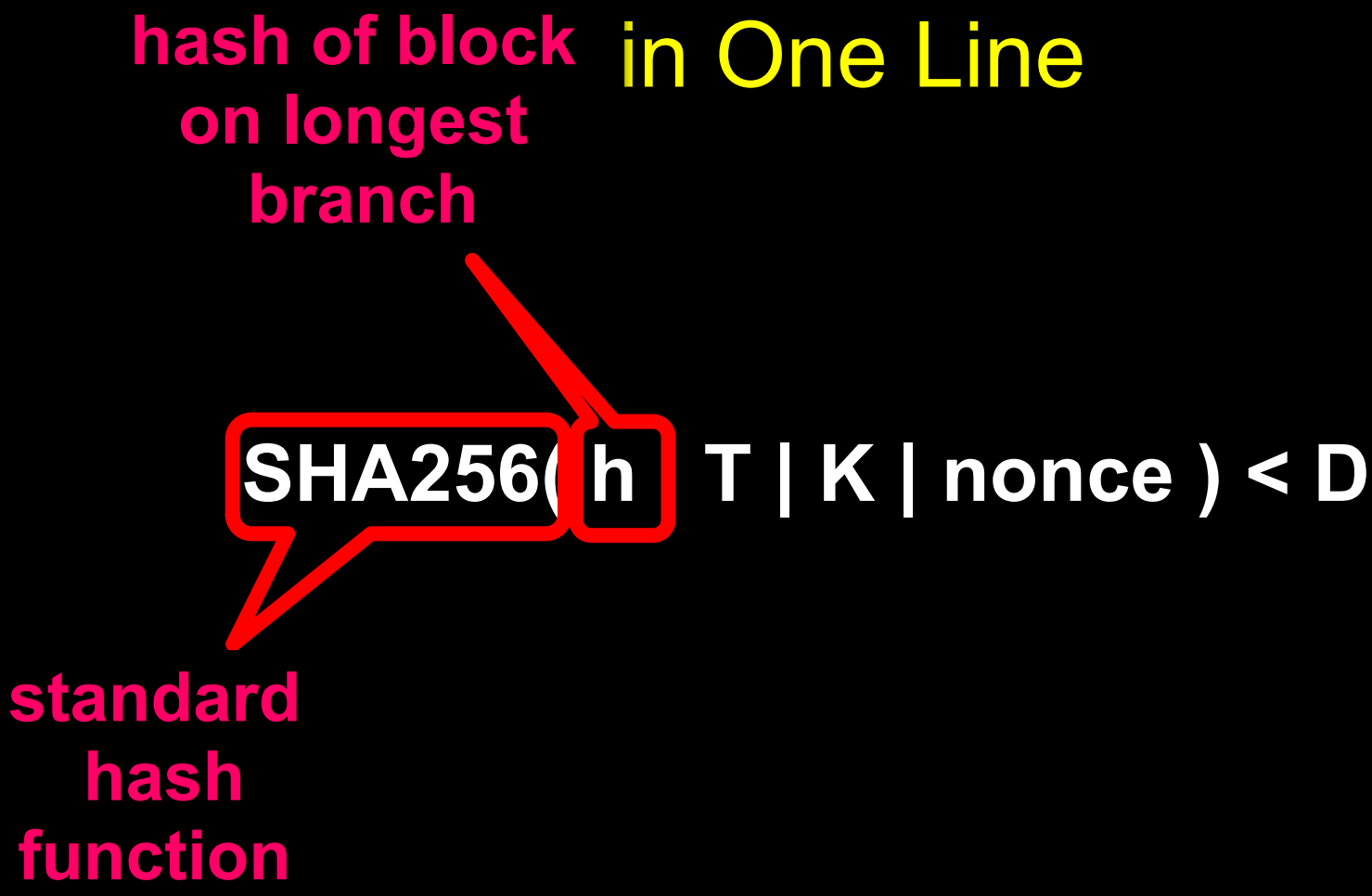
SHA256(h | T | K | nonce) < D

Nakamoto Consensus in One Line

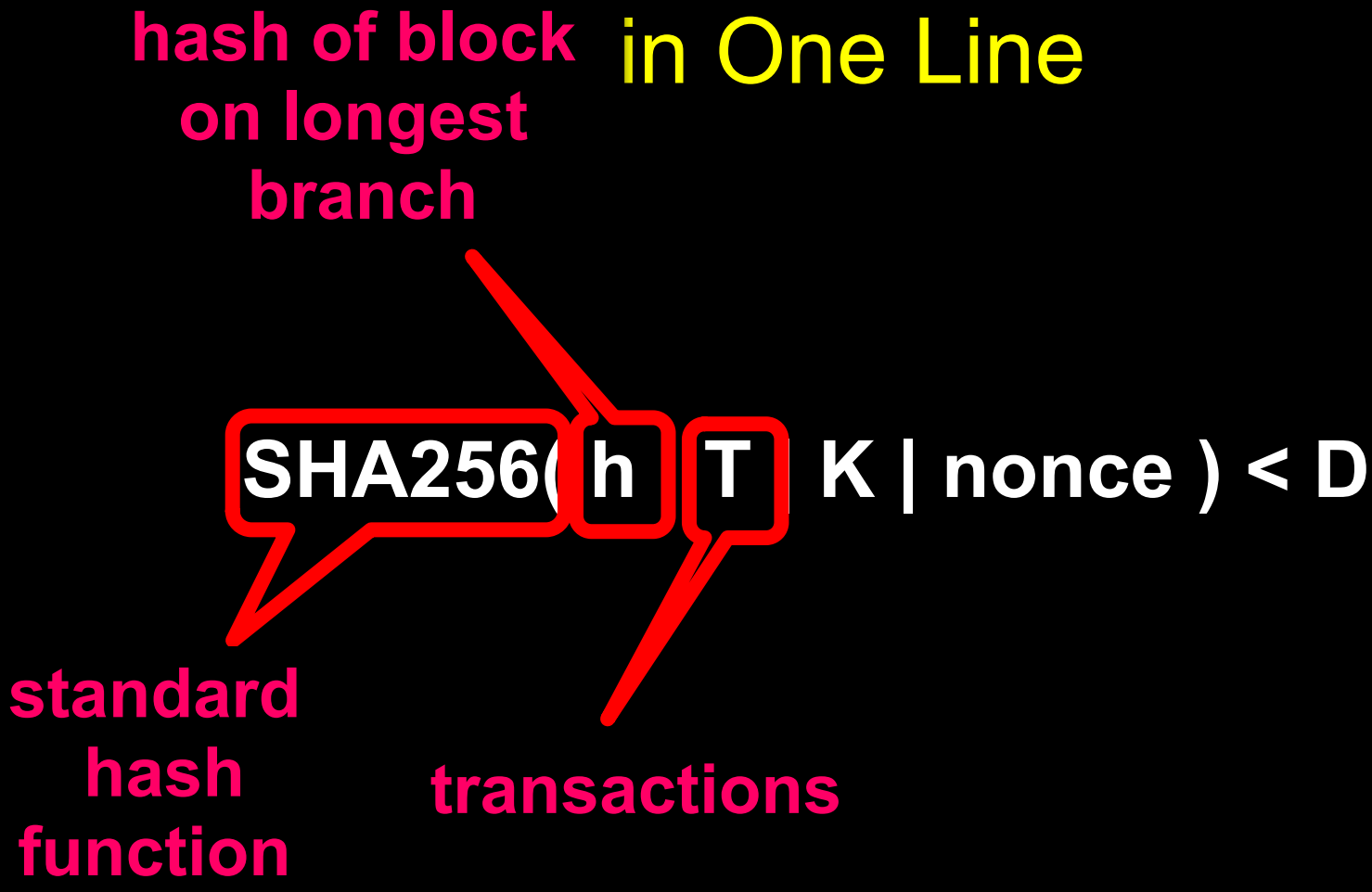
SHA256(h | T | K | nonce) < D

standard hash function

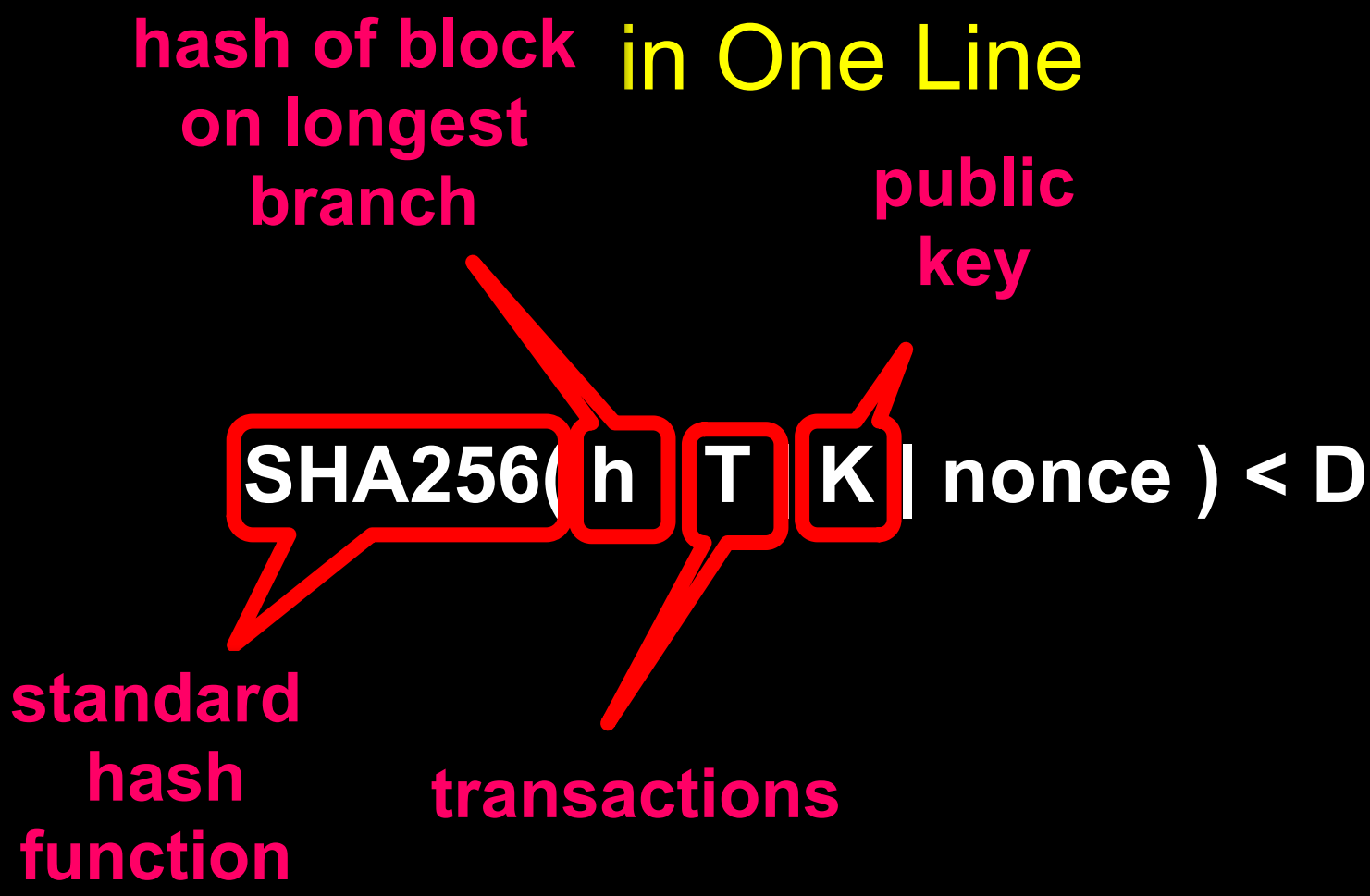
Nakamoto Consensus in One Line



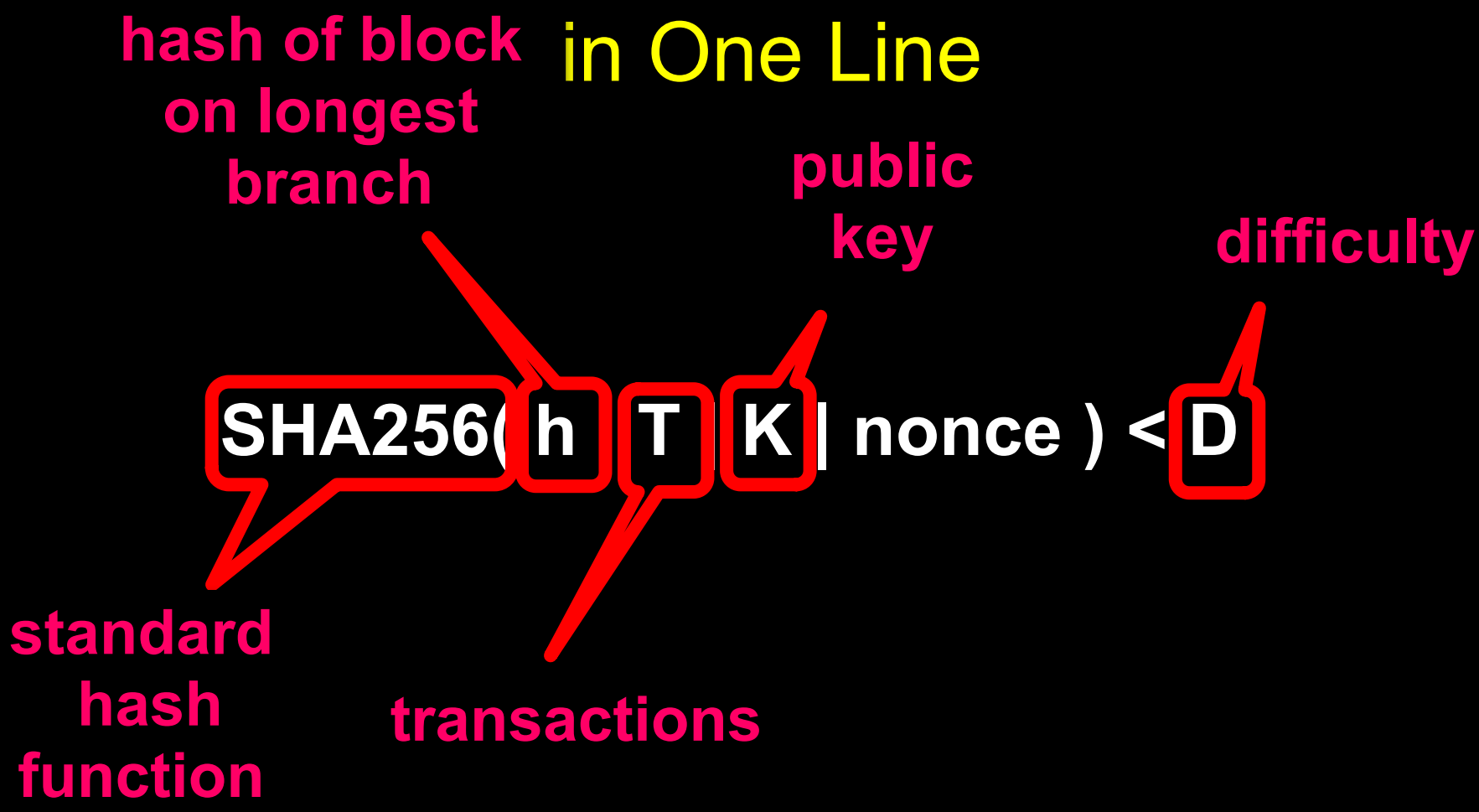
Nakamoto Consensus in One Line



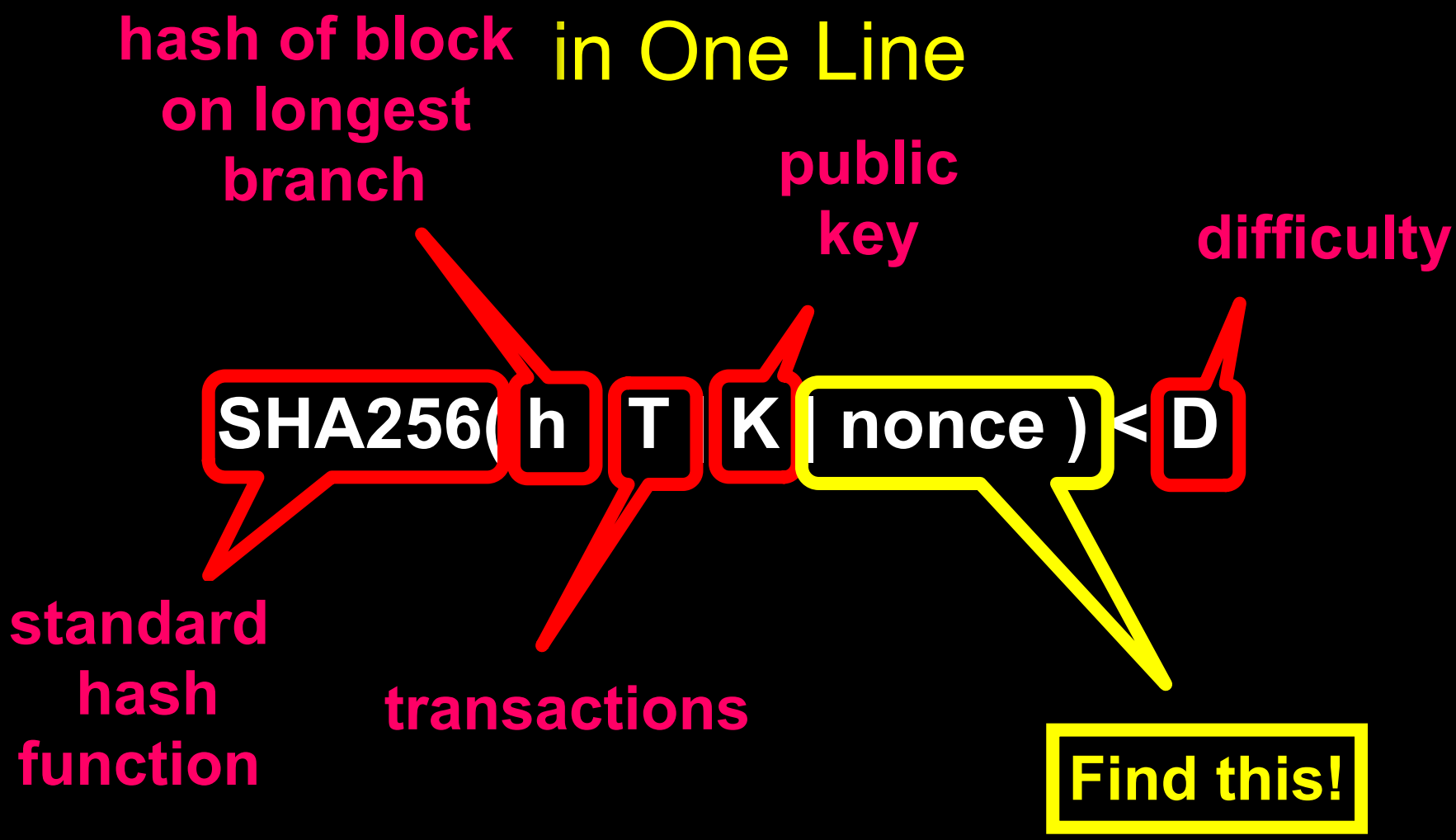
Nakamoto Consensus in One Line



Nakamoto Consensus in One Line



Nakamoto Consensus in One Line



Nakamoto Consensus in One Line

SHA256(h | T | K | nonce) < D

No formal proof of collision-resistence
Possible quantum attacks?
Ethereum uses similar, not same hash

Nakamoto Consensus in One Line

SHA256(h | T | K | nonce) < D

Tamper-proofing

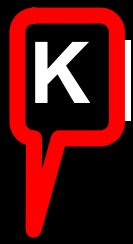
Nakamoto Consensus in One Line

SHA256(h | T | K | nonce) < D

Actually hash of txn "Merkel tree" root
Constant size
Too expensive to hash txns themselves

Nakamoto Consensus in One Line

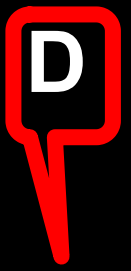
SHA256(h | T | K | nonce) < D



Pay “coinbase” reward to this address

Nakamoto Consensus in One Line

SHA256(h | T | K | nonce) < D



Too easy? frequent forks, finality slow
Too hard? slow progress, low throughput
Adjusted dynamically: ~1 block / 10 min

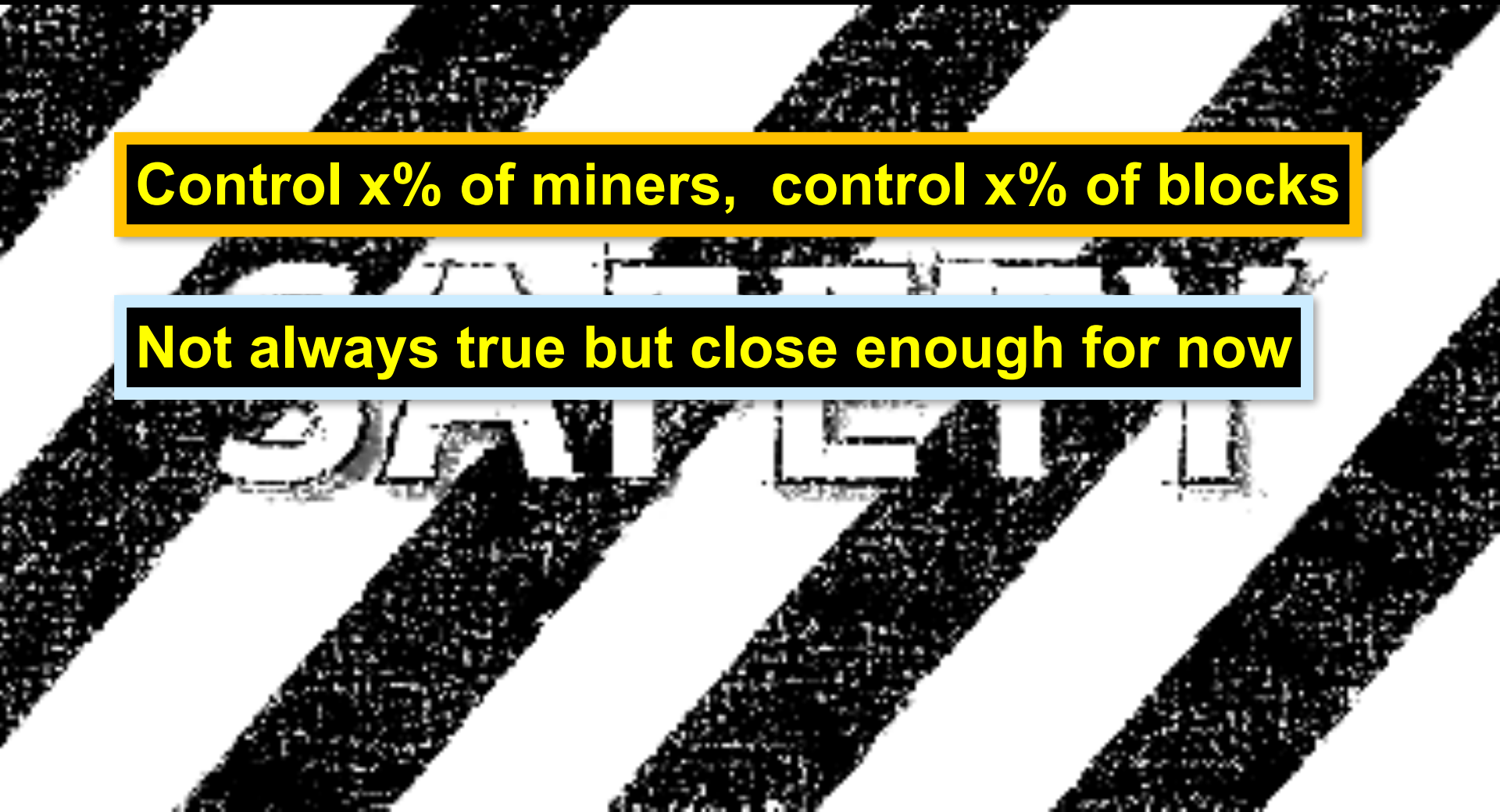
Nakamoto Consensus in One Line

Chain Property

Chain Property

Control x% of miners, control x% of blocks

Chain Property

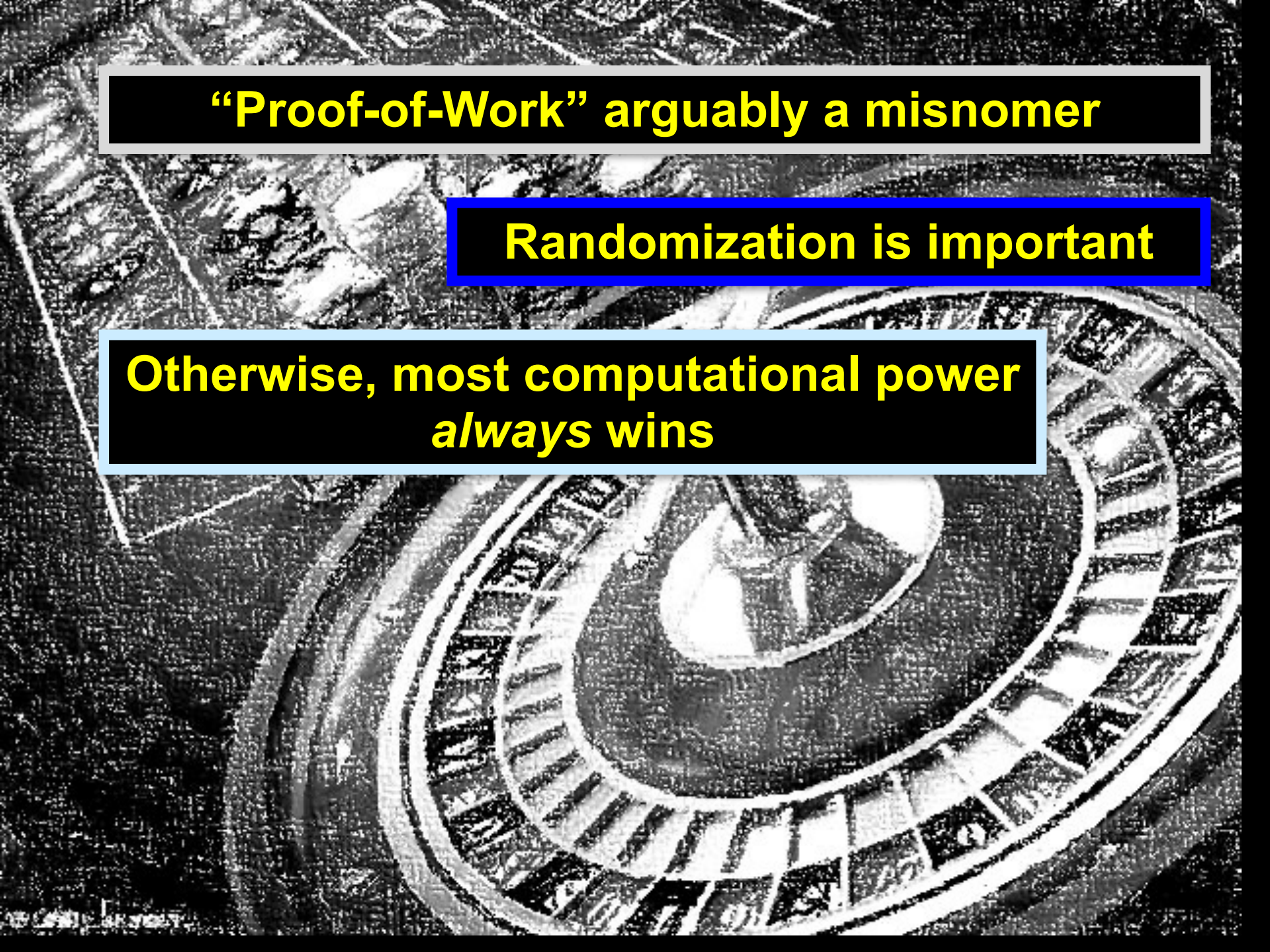


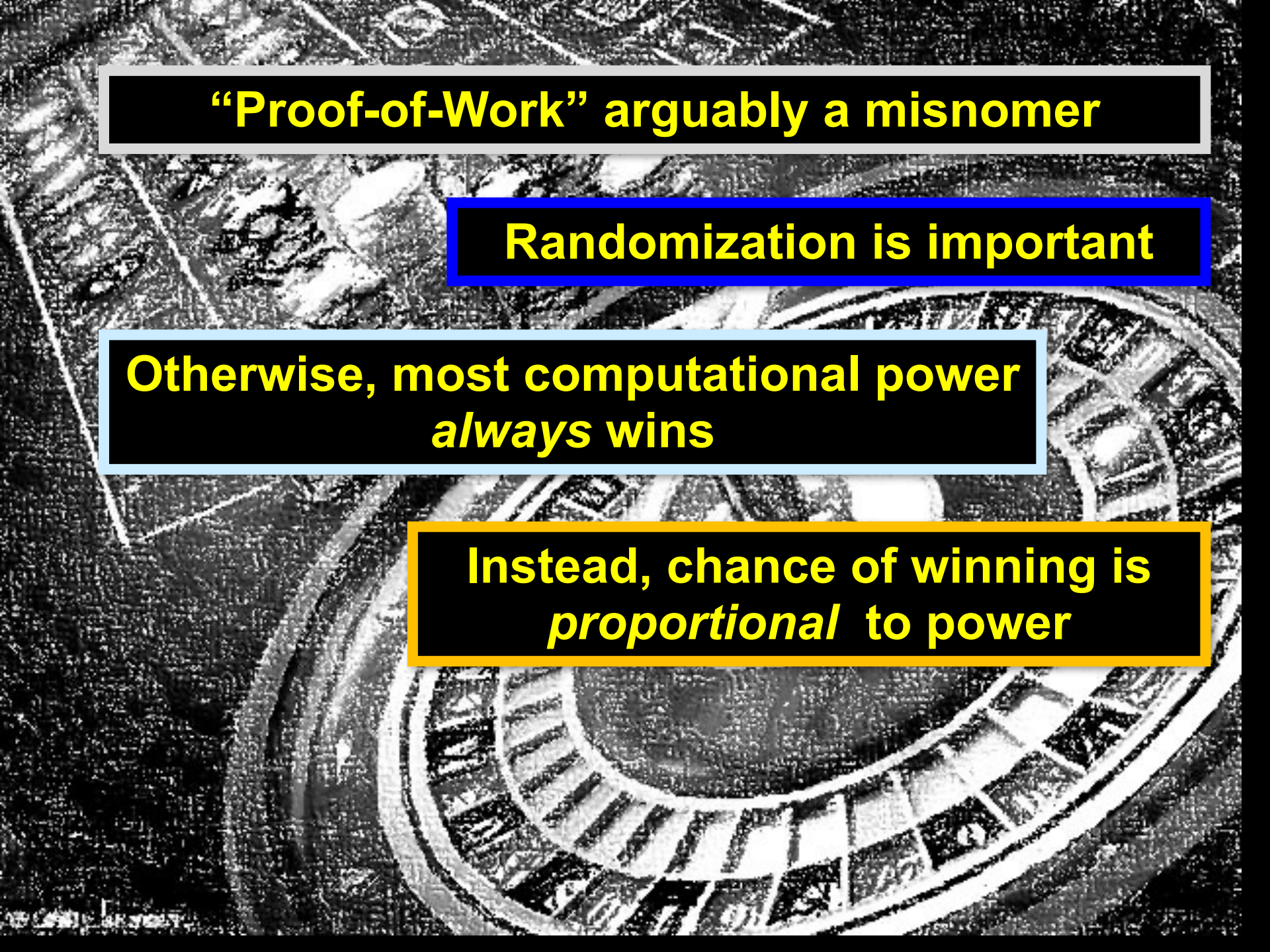
Control x% of miners, control x% of blocks

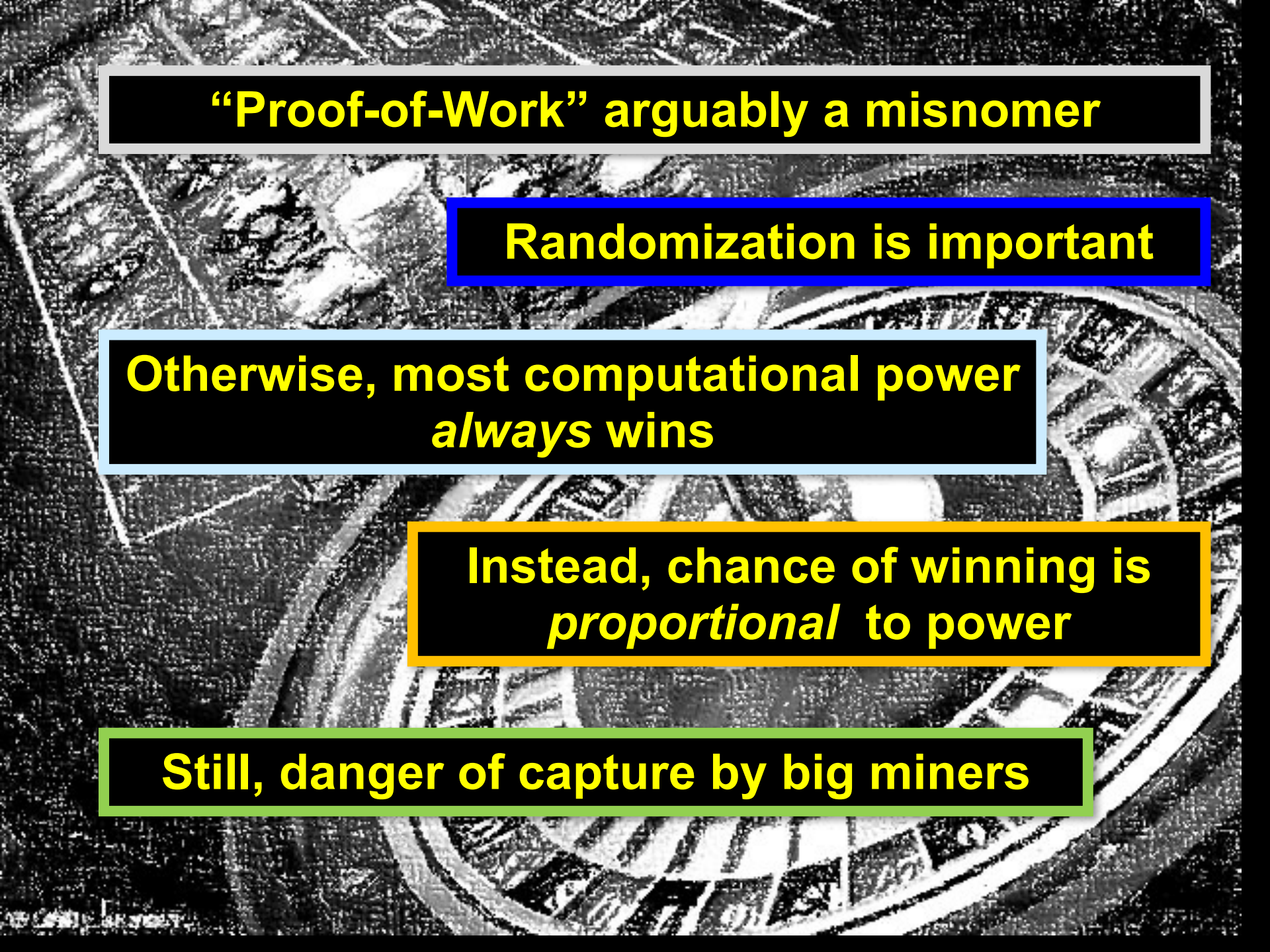
Not always true but close enough for now

“Proof-of-Work” arguably a misnomer


“Proof-of-Work” arguably a misnomer
Randomization is important


"Proof-of-Work" arguably a misnomer
Randomization is important
Otherwise, most computational power *always* wins


"Proof-of-Work" arguably a misnomer
Randomization is important
Otherwise, most computational power *always* wins
Instead, chance of winning is *proportional* to power


"Proof-of-Work" arguably a misnomer
Randomization is important
Otherwise, most computational power *always* wins
Instead, chance of winning is *proportional* to power
Still, danger of capture by big miners

Longest Chain Rule

in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to

Honest miners build on longest chain …

Longest Chain Rule

in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to

Honest miners build on longest chain …

Dishonest miners would have to out-compute all honest miners

Calculation

p = probability an honest node finds the next block

Back of the envelope calculation

q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

Calculation

p = probability an honest node finds the next block

Back of the envelope calculation

q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} (q$$

How likely dishonest miner can overtake honest miner to reverse transaction?

Calculation

p = probability an honest node finds the next block

Back of the envelope calculation

q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} (q$$

How likely dishonest miner can overtake honest miner to reverse transaction?

Exponentially small

Calculation

p = probability an honest node finds the next block

Back of the envelope calculation

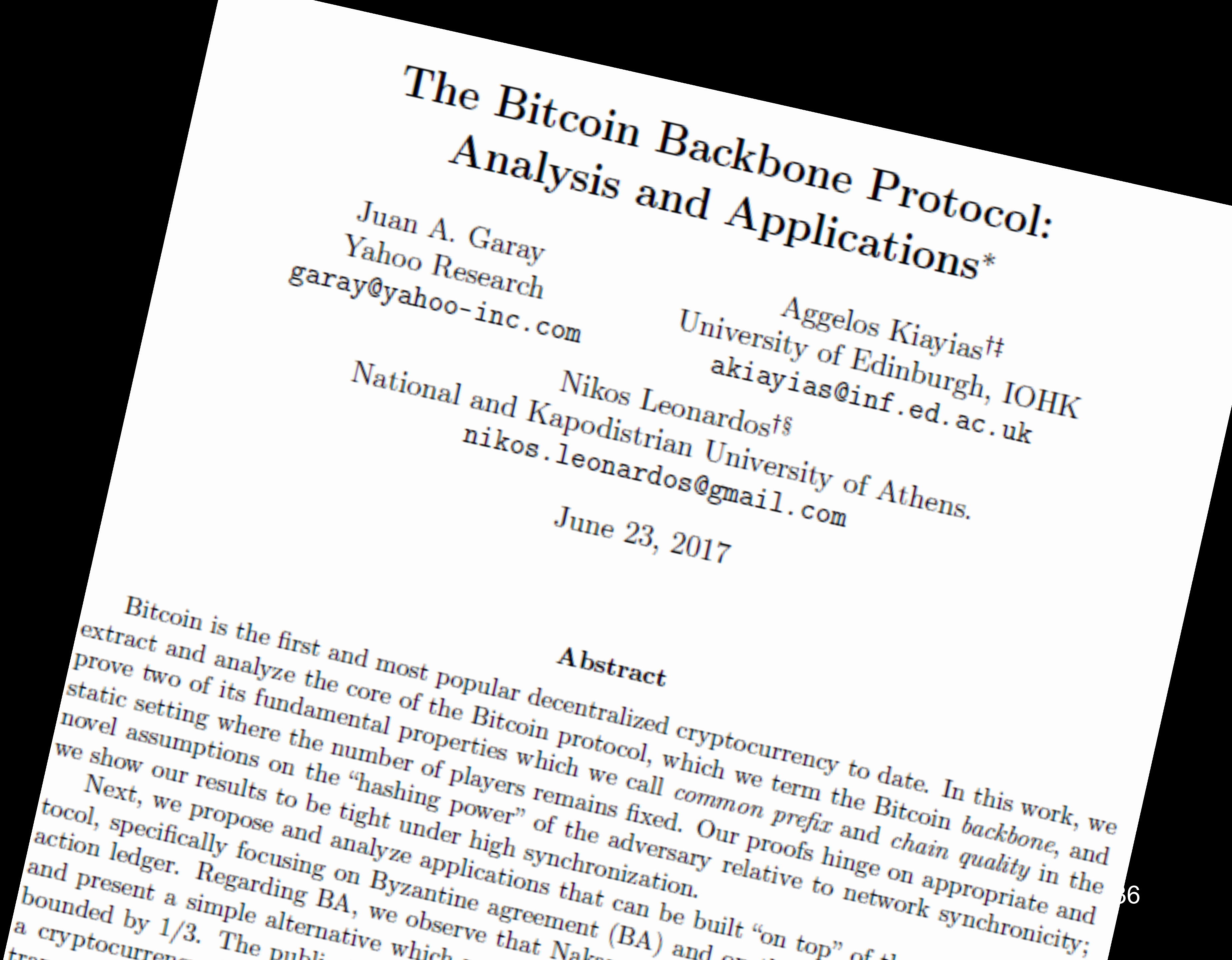
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} (q$$

How likely dishonest miner can overtake honest miner to reverse transaction?

Exponentially small

Calculation naïve but probably mostly right



The Bitcoin Backbone Protocol: Analysis and Applications*

Juan A. Garay
Yahoo Research
garay@yahoo-inc.com

Aggelos Kiayias†‡
University of Edinburgh, IOHK
akiayias@inf.ed.ac.uk

Nikos Leonardos†§
National and Kapodistrian University of Athens.
nikos.leonardos@gmail.com

June 23, 2017

Abstract

Bitcoin is the first and most popular decentralized cryptocurrency to date. In this work, we extract and analyze the core of the Bitcoin protocol, which we term the Bitcoin *backbone*, and prove two of its fundamental properties which we call *common prefix* and *chain quality* in the static setting where the number of players remains fixed. Our proofs hinge on appropriate and novel assumptions on the "hashing power" of the adversary relative to network synchronicity; we show our results to be tight under high synchronization.

Next, we propose and analyze applications that can be built "on top" of th
tocol, specifically focusing on Byzantine agreement (BA) and on th
action ledger. Regarding BA, we observe that Nak
and present a simple alternative which
bounded by 1/3. The publi
a cryptocurren

The Bitcoin Backbone Protocol: Analysis and Applications*

Juan A. Garay
Yahoo Research
garay@yaho[illegible]

[illegible] Leonardos†§
[illegible] and Kapodistrian University of Athens.
nikos.leonardos@gmail.com

June 23, 2017

Here is a more complete calculation …

Abstract

Bitcoin is the first and most popular decentralized cryptocurrency to date. In this work, we extract and analyze the core of the Bitcoin protocol, which we term the Bitcoin *backbone*, and prove two of its fundamental properties which we call *common prefix* and *chain quality* in the static setting where the number of players remains fixed. Our proofs hinge on appropriate and novel assumptions on the "hashing power" of the adversary relative to network synchronicity; we show our results to be tight under high synchronization.

Next, we propose and analyze applications that can be built "on top" of t[illegible] tocol, specifically focusing on Byzantine agreement (BA) and [illegible] action ledger. Regarding BA, we observe that Nak[illegible] and present a simple alternative which [illegible] bounded by 1/3. The publi[illegible] a cryptocurrency [illegible]

The Bitcoin Backbone Protocol: Analysis and Applications*

Juan A. Garay
Yahoo Research
garay@yaho

Here is a more complete calculation …

…os Leonardos†§
…dr and Kapodistrian Univ…
nikos.leo…
…inf.ed.ac.uk
…, IOHK

Lots of Chernoff bounds …

Abstract

Bitcoin is the first and most popular decentralized cryptocurrency to date. In this work, we extract and analyze the core of the Bitcoin protocol, which we term the Bitcoin *backbone*, and prove two of its fundamental properties which we call *common prefix* and *chain quality* in the static setting where the number of players remains fixed. Our proofs hinge on appropriate and novel assumptions on the "hashing power" of the adversary relative to network synchronicity; we show our results to be tight under high synchronization.

Next, we propose and analyze applications that can be built "on top" of the Bitcoin backbone protocol, specifically focusing on Byzantine agreement (BA) and …tocol, specifically focusing on Byzantine agreement (BA) and …action ledger. Regarding BA, we observe that Nak… and present a simple alternative which … bounded by 1/3. The publi… a cryptocurrency …

The Bitcoin Backbone Protocol: Analysis and Applications*

Juan A. Garay
Yahoo Research
garay@yaho

Here is a more complete calculation …

Lots of Chernoff bounds …

more precise statements of correctness

…centralized cryptocurrency to date. In this work, we
…damental properties which we call *common prefix* and *chain quality* in the
…setting where the number of players remains fixed. Our proofs hinge on appropriate and
novel assumptions on the "hashing power" of the adversary relative to network synchronicity;
we show our results to be tight under high synchronization.
Next, we propose and analyze applications that can be built "on top" of
tocol, specifically focusing on Byzantine agreement (BA) and
action ledger. Regarding BA, we observe that
and present a simple alternative which
bounded by 1/3. The
a cryptocurrency

The Bitcoin Backbone Protocol:
Analysis and Applications*

Juan A. Garay
Yahoo Research
garay@yaho

Here is a more complete calculation …

Lots of Chernoff bounds …

more precise statements of correctness

more precise bounds on hashing power

Crime doesn't Pay

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to

Suppose dishonest party acquires lots of hashing power …

Crime doesn't Pay

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to

Suppose dishonest party acquires lots of hashing power …

Unlimited double spending?

Crime doesn't Pay

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to

Suppose dishonest party acquires lots of hashing power …

Unlimited double spending?

Or collect all the rewards?

Crime doesn't Pay

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to

Suppose dishonest party acquires lots of hashing power …

Unlimited double spending?

Or collect all the rewards?

Vandalism destroys coin values!

Limited Throughput is Feature, not Bug

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

Limited Throughput is Feature, not Bug

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

Generated blocks should have time to propagate; otherwise, forks increase

Limited Throughput is Feature, not Bug

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

Generated blocks should have time to propagate; otherwise, forks increase

Number of blocks/time kept approximately constant

Limited Throughput is Feature, not Bug

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

Generated blocks should have time to propagate; otherwise, forks increase

Number of blocks/time kept approximately constant

By varying PoW difficulty

Limited Throughput is Feature, not Bug

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

Generated blocks should have time to propagate; otherwise, forks increase

Number of blocks/time kept approximately constant

By varying PoW difficulty

Limited scalability becomes a problem as Bitcoin becomes successful

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Clients send transactions to miners

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Clients send transactions to miners

On the Bitcoin P2P layer

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Clients send transactions to miners

On the Bitcoin P2P layer

Rumor: mining cartels use faster side-channels

The steps to run the network are as follows:

2) Each node collects new transactions into a block.

4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Miners assemble transactions into blocks

The steps to run the network are as follows:

2) Each node collects new transactions into a block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Miners assemble transactions into blocks

Economy of scale: single transaction too expensive

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Miners race to do Proof of Work

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Miners race to do Proof of Work

Today, consumes lots of energy

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Miners race to do Proof of Work

Today, consumes lots of energy

Cartels with access to cheap power and ASICs control most of hashing power

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

If multiple winners at the same time …

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

If multiple winners at the same time …

the blockchain forks …

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

If multiple winners at the same time …

the blockchain forks …

Result: high latency because need to wait until your transaction deep enough in chain

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Sanity check: malformed txns rejected

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Sanity check: malformed txns rejected

Incentive for miners to behave …

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Sanity check: malformed txns rejected

Incentive for miners to behave …

Double spending filter

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Successors build on recent well-formed blocks

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Successors build on recent well-formed blocks

Pick longest chain if there is a fork

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Successors build on recent well-formed blocks

Pick longest chain if there is a fork

Break ties arbitrarily

Where's my Money?

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them

You mine a block, you get paid

Where's my Money?

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them

You mine a block, you get paid

How new bitcoins are generated

Where's my Money?

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them

You mine a block, you get paid

How new bitcoins are generated

"Coinbase" transaction

Where's my Money?

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered

Customers can include transaction fee

Where's my Money?

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered

Customers can include transaction fee

Higher fees buy lower latency?

Deflationary

the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

Limit on number of BTC ever minted

Deflationary

the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

Limit on number of BTC ever minted

Fear of inflation?

Deflationary

the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

Limit on number of BTC ever minted

Fear of inflation?

But not deflation?

Deflationary

the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

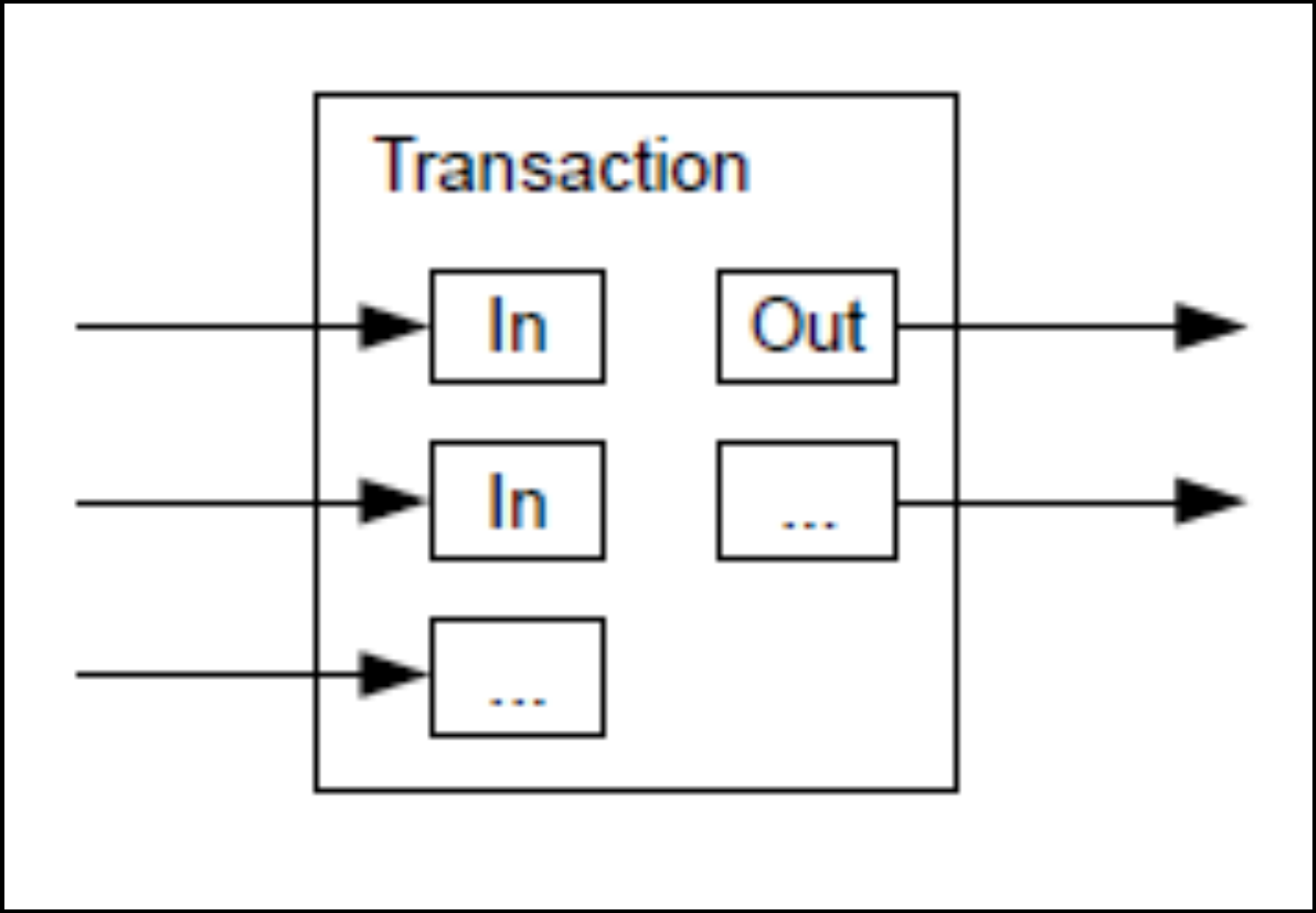
Limit on number of BTC ever minted

Fear of inflation?

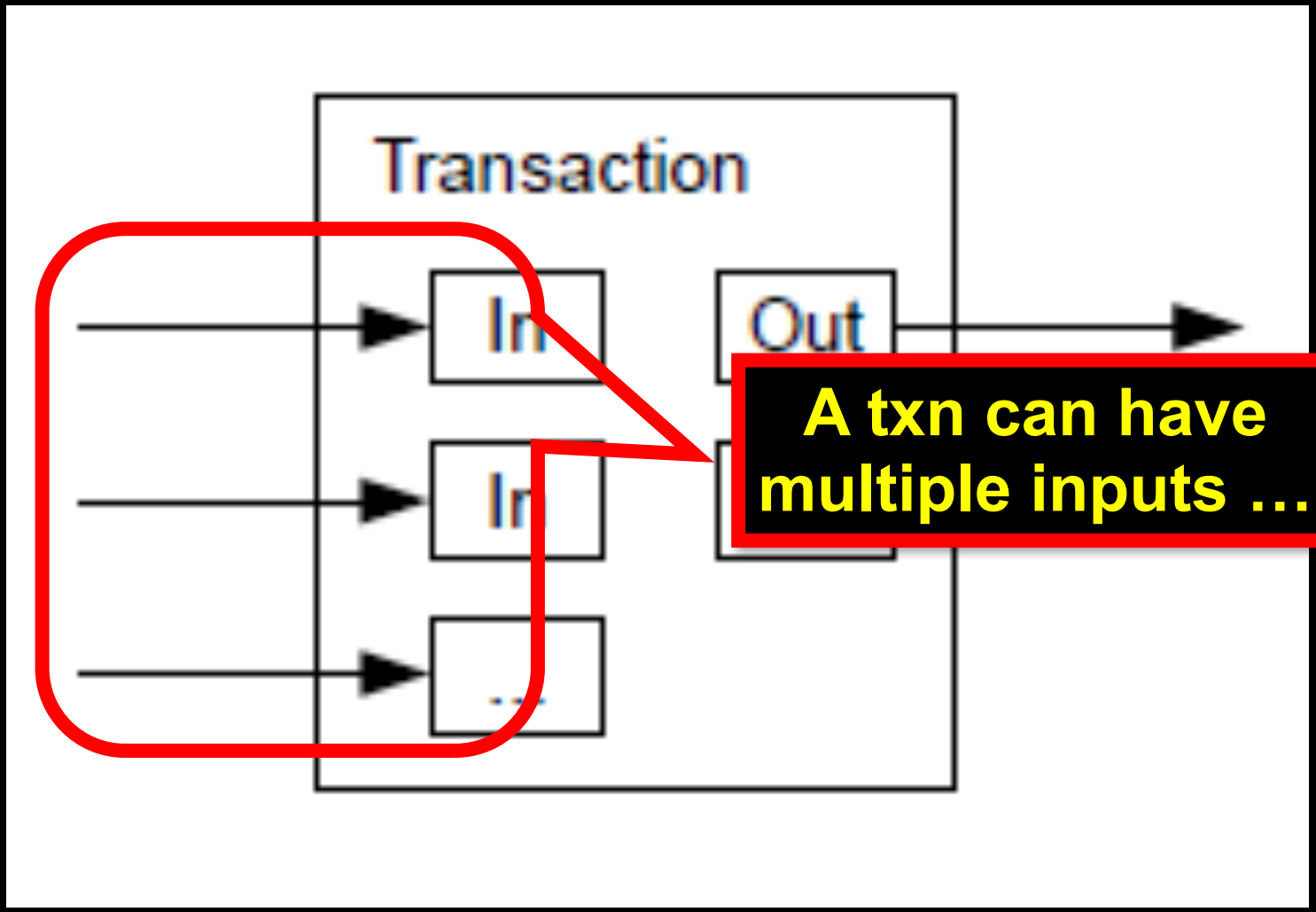
But not deflation?

Deflation implies inflated fees

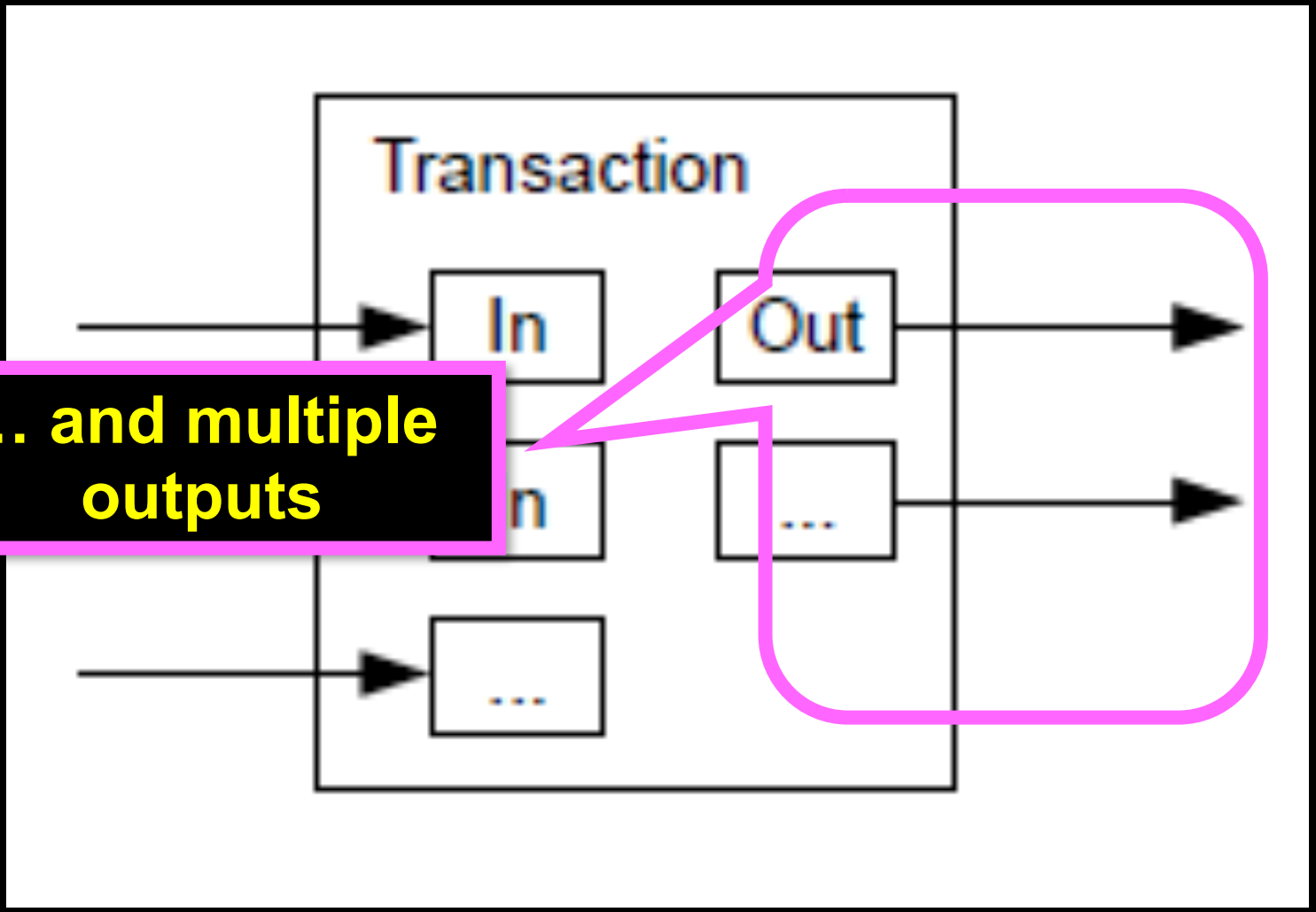
UTXO (unspent transaction output) Model



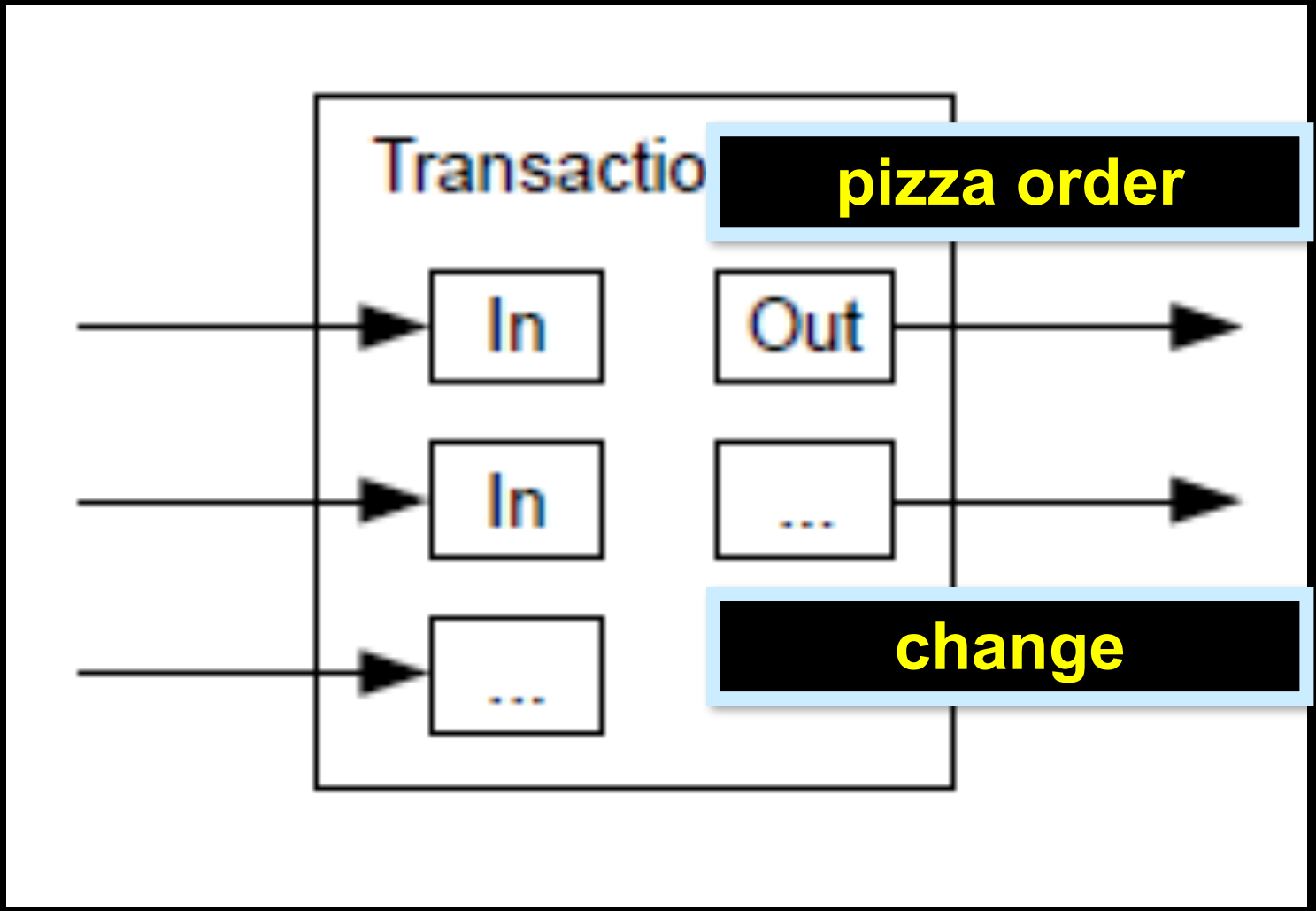
UTXO Model



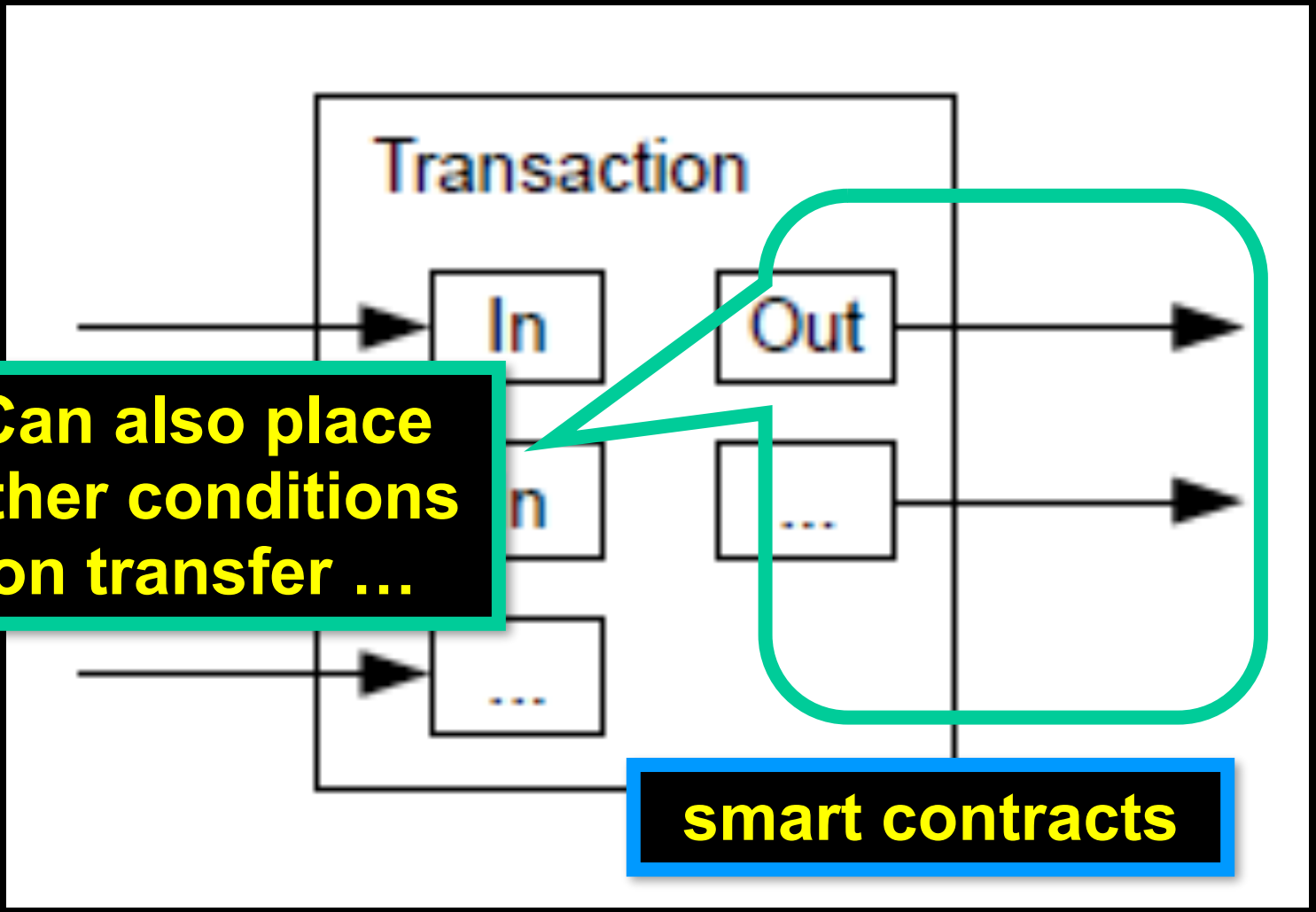
UTXO Model



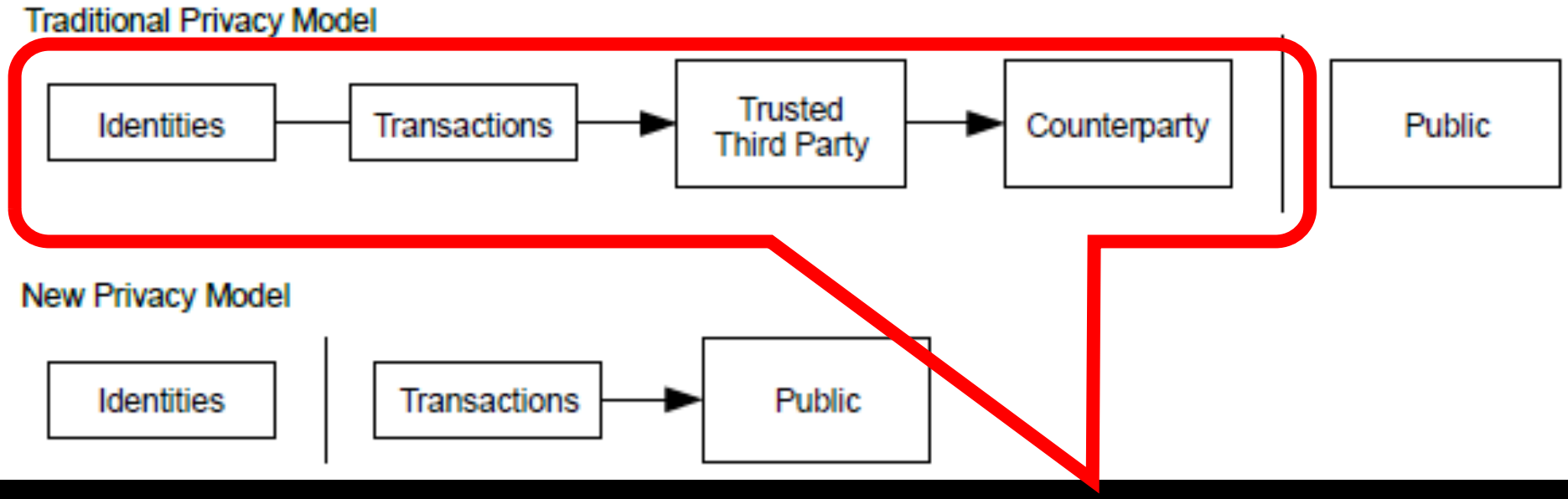
UTXO Model



UTXO Model


Transaction
In
Out
n
...
...
Can also place other conditions on transfer …
smart contracts

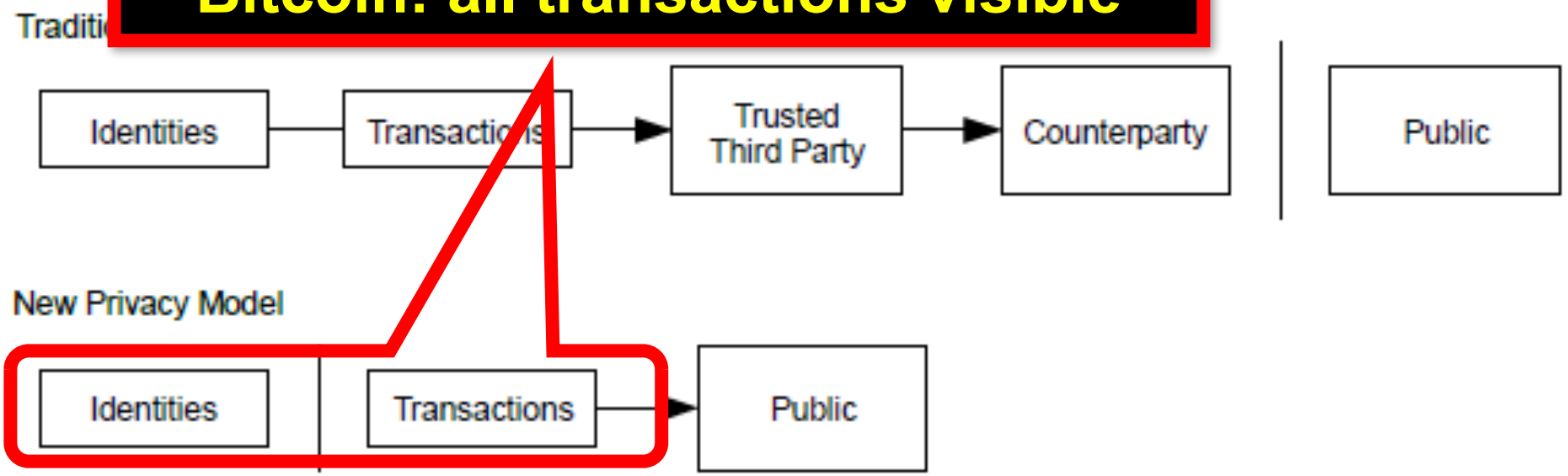
Privacy



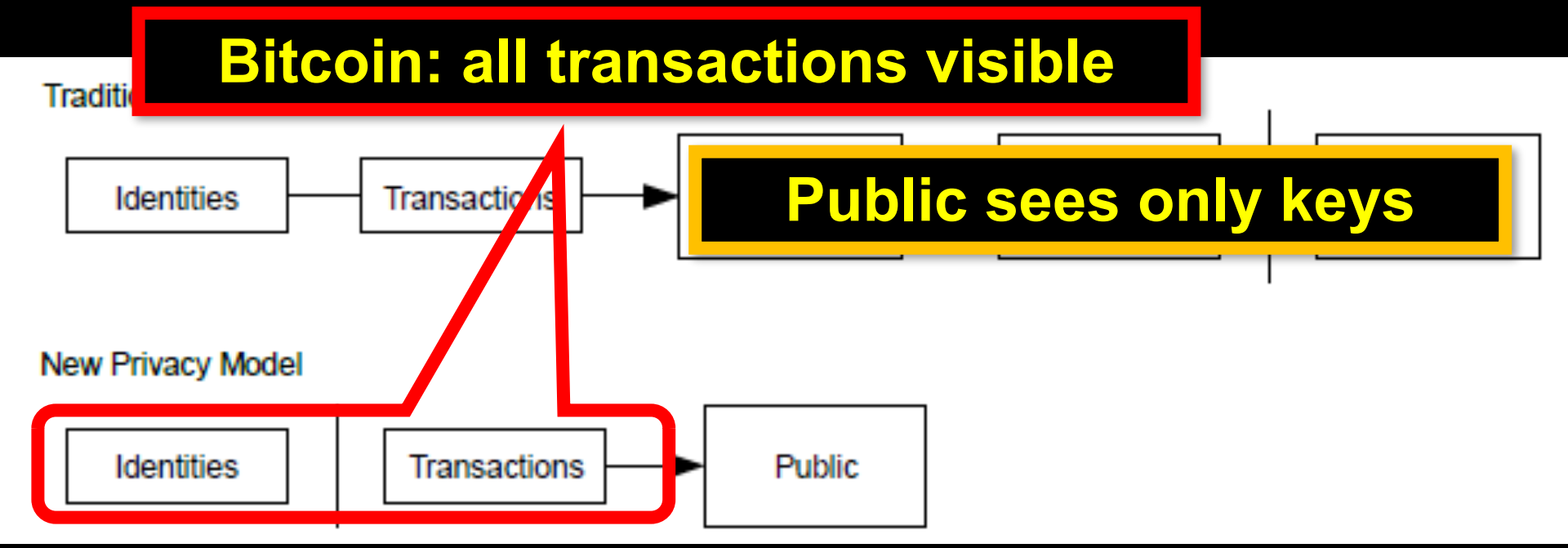
Old-school: Trusted 3rd party keeps IDs and transactions secret

Privacy

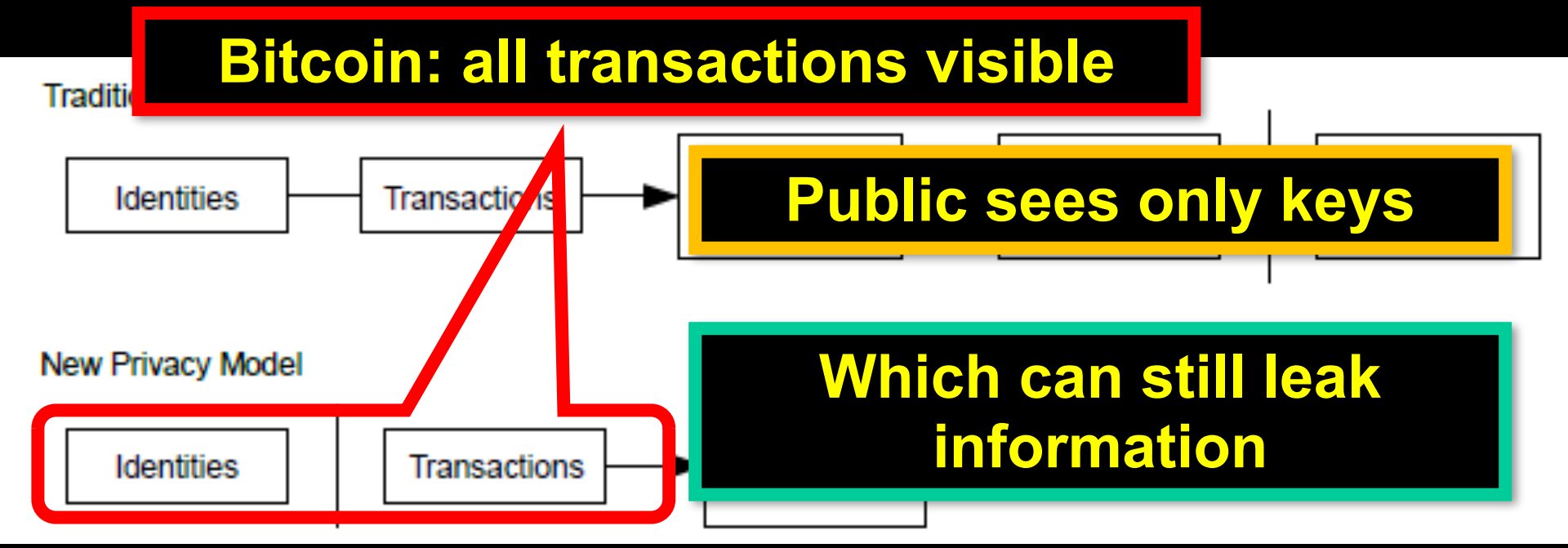
Bitcoin: all transactions visible



Privacy

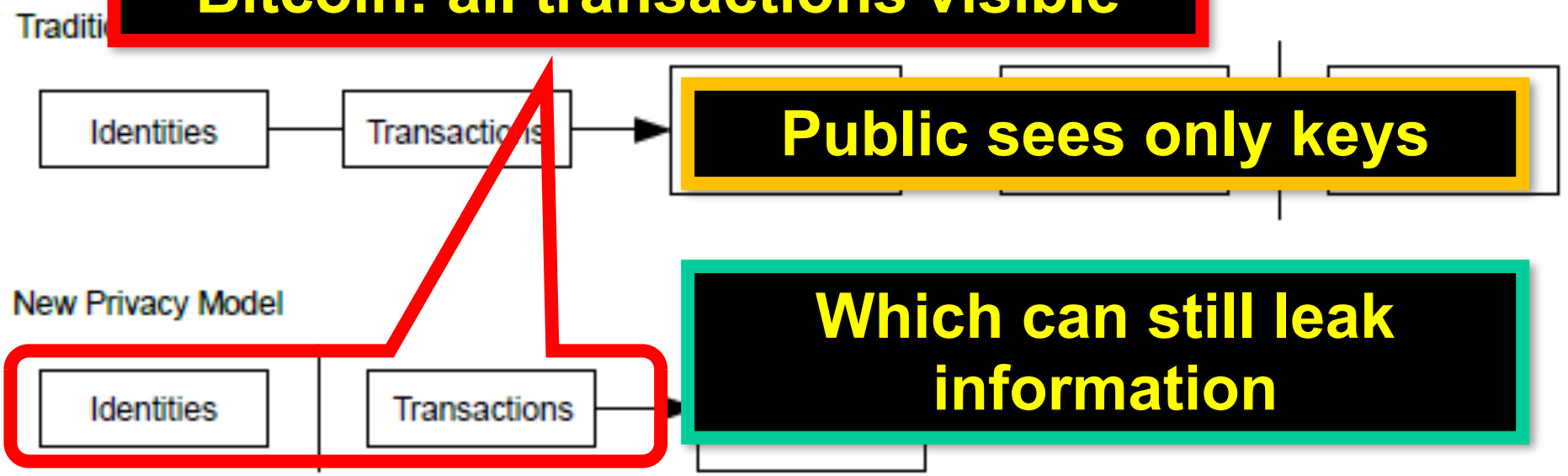

Bitcoin: all transactions visible
Identities
Transactions
Public sees only keys
New Privacy Model
Identities
Transactions
Public

Privacy


Bitcoin: all transactions visible
Identities
Transactions
Public sees only keys
New Privacy Model
Identities
Transactions
Which can still leak information

Privacy

Bitcoin: all transactions visible



Public sees only keys

Which can still leak information

“pseudonymous”

“Decentralized Trust Infrastructure”

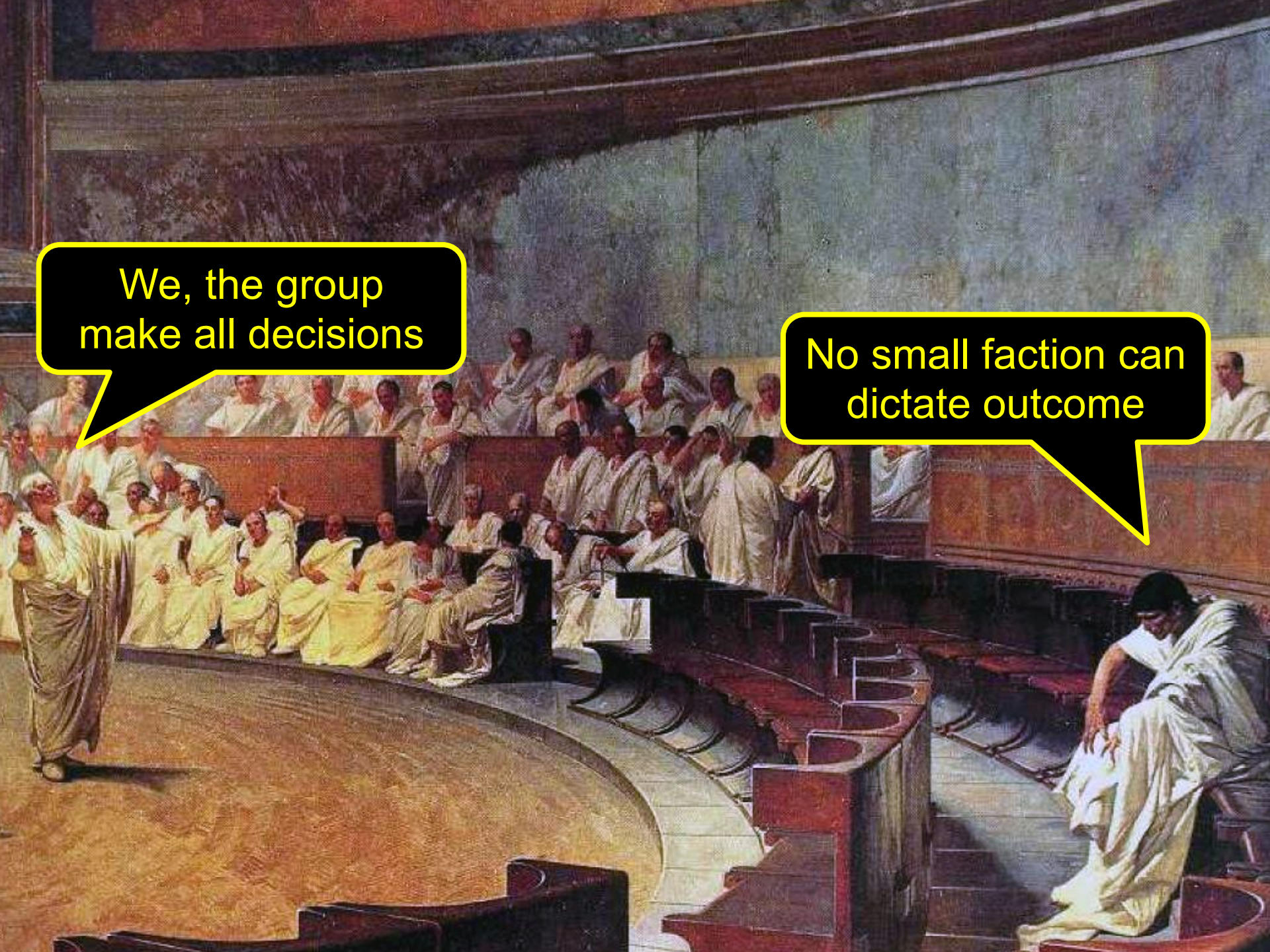
Centralized

Decentralized

Centralized Trust

Centralized Trust

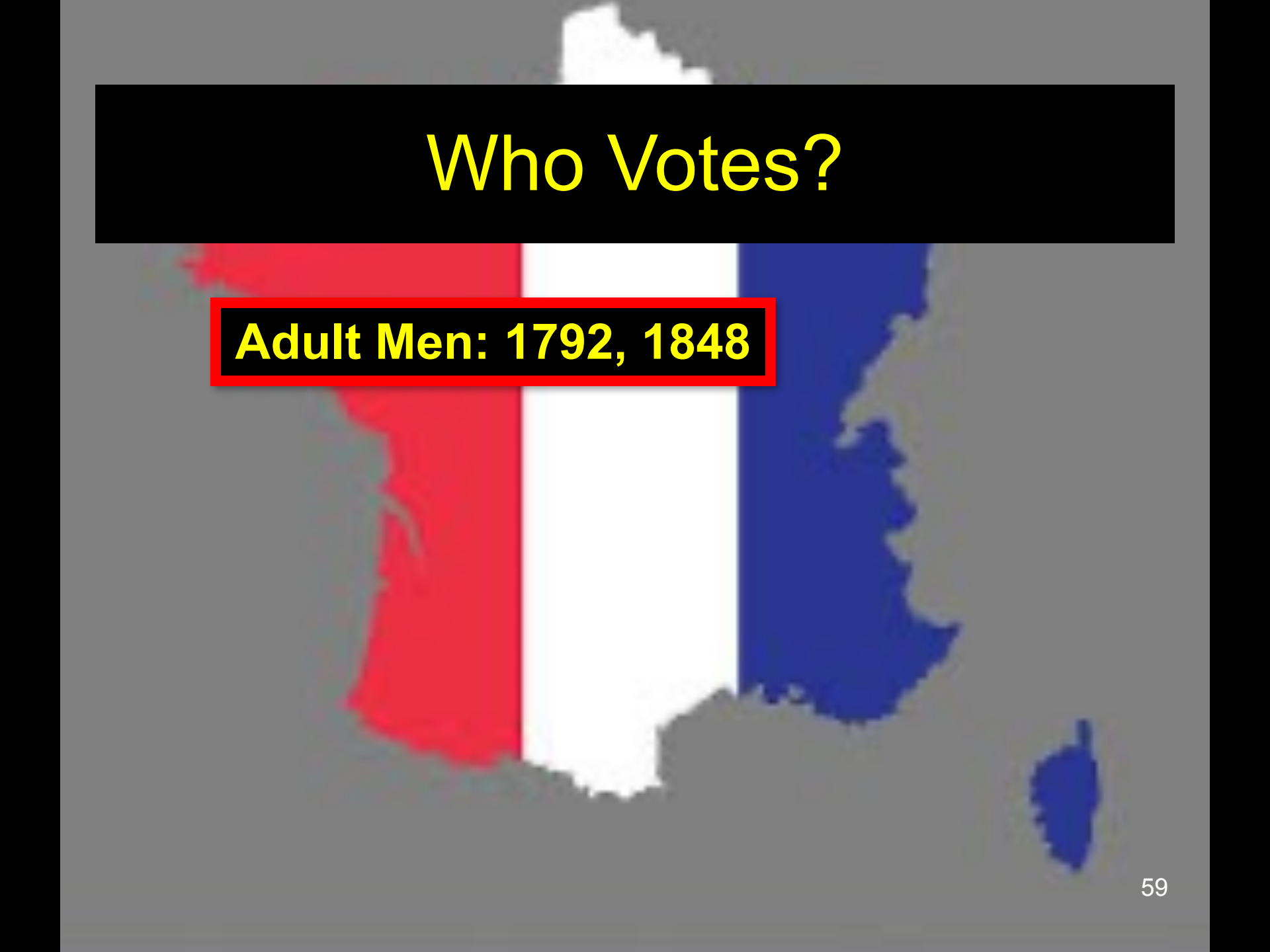

We, the group
make all decisions


We, the group make all decisions
No small faction can dictate outcome

Who Votes?

Who Votes?

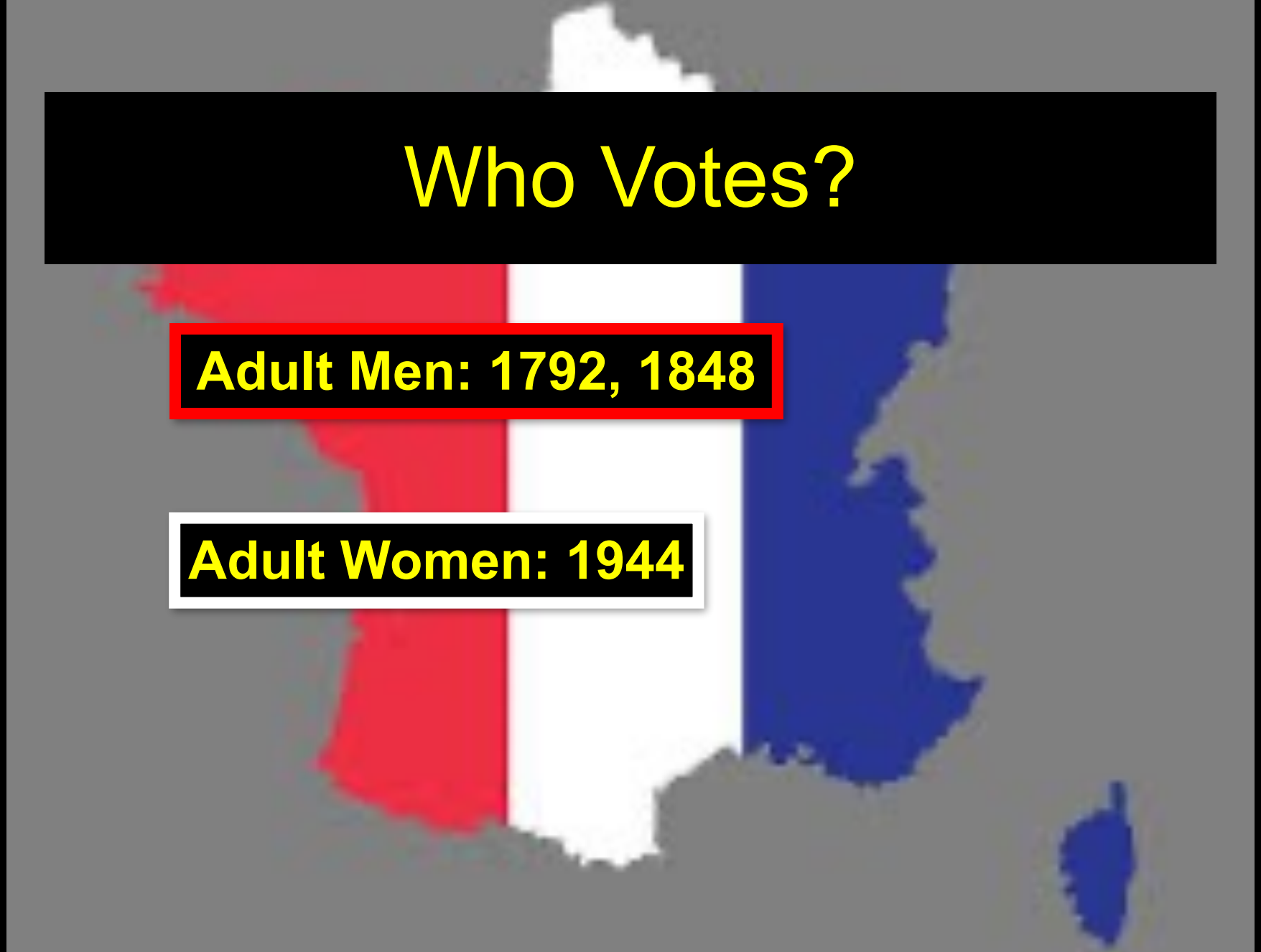
Adult Men: 1792, 1848



Who Votes?

Adult Men: 1792, 1848

Adult Women: 1944



Who Votes?

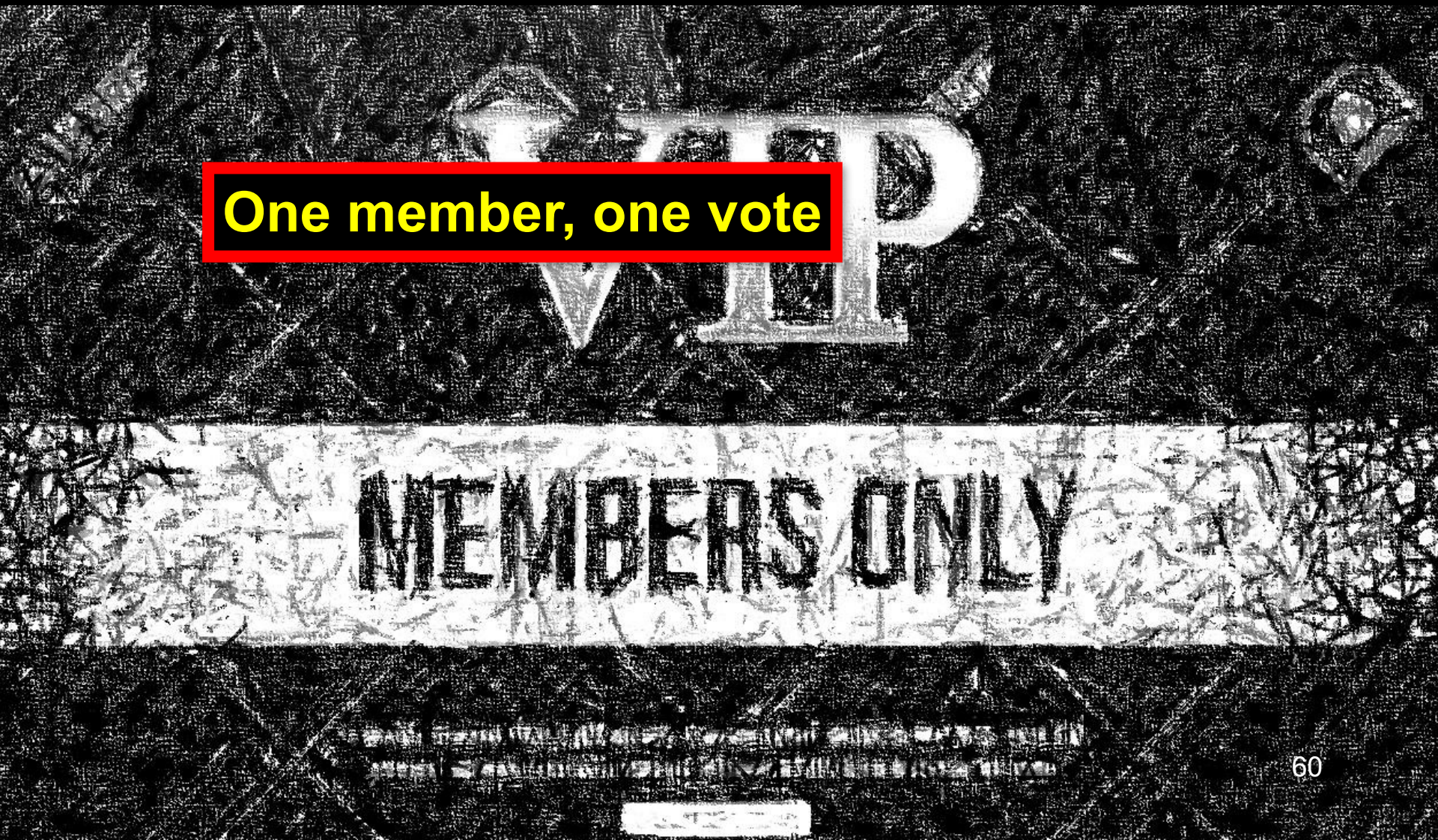
Adult Men: 1792, 1848

Adult Women: 1944

Only citizens …

Proof of Membership

Proof of Membership



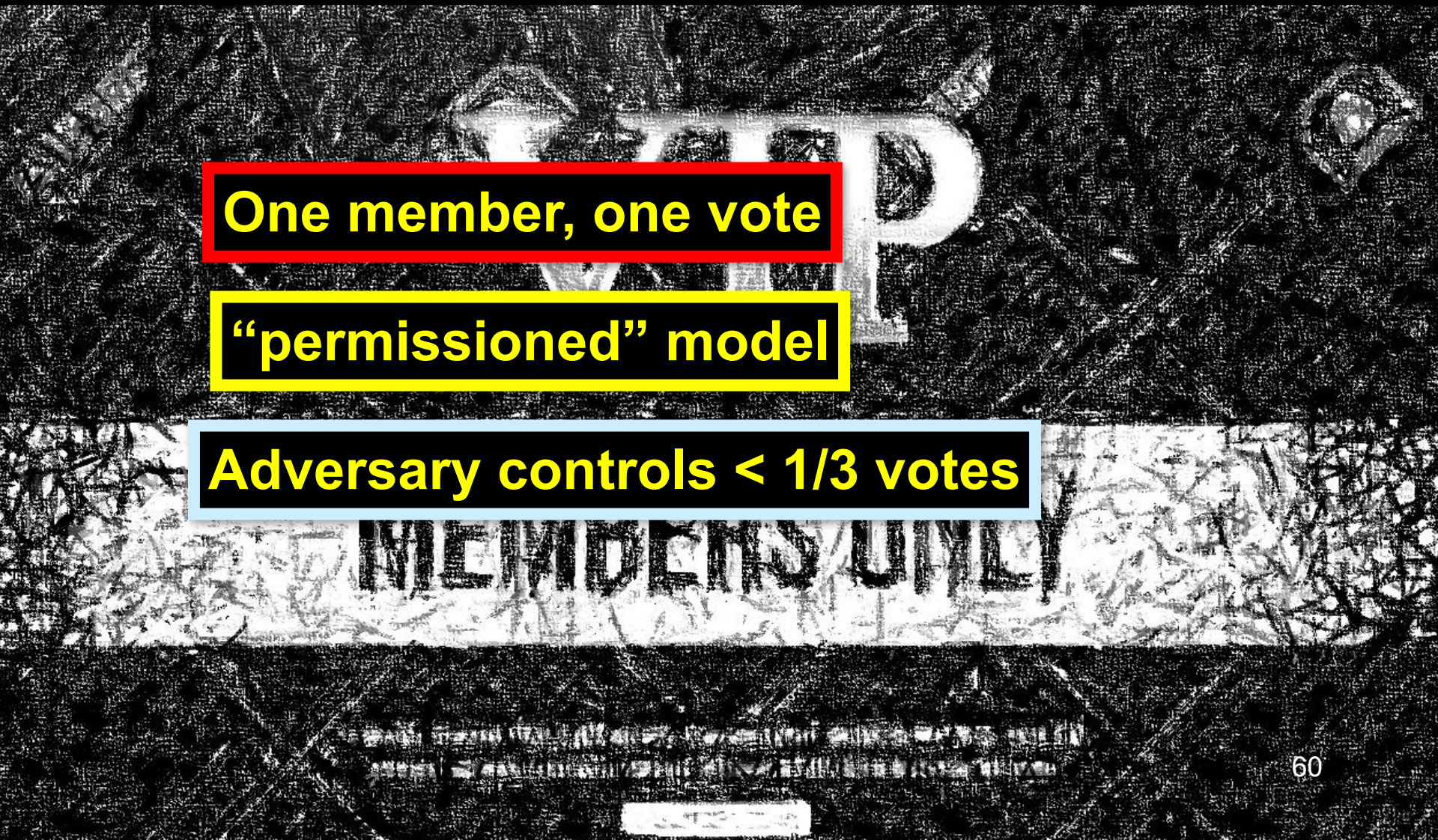
One member, one vote

Proof of Membership

One member, one vote

“permissioned” model

Proof of Membership

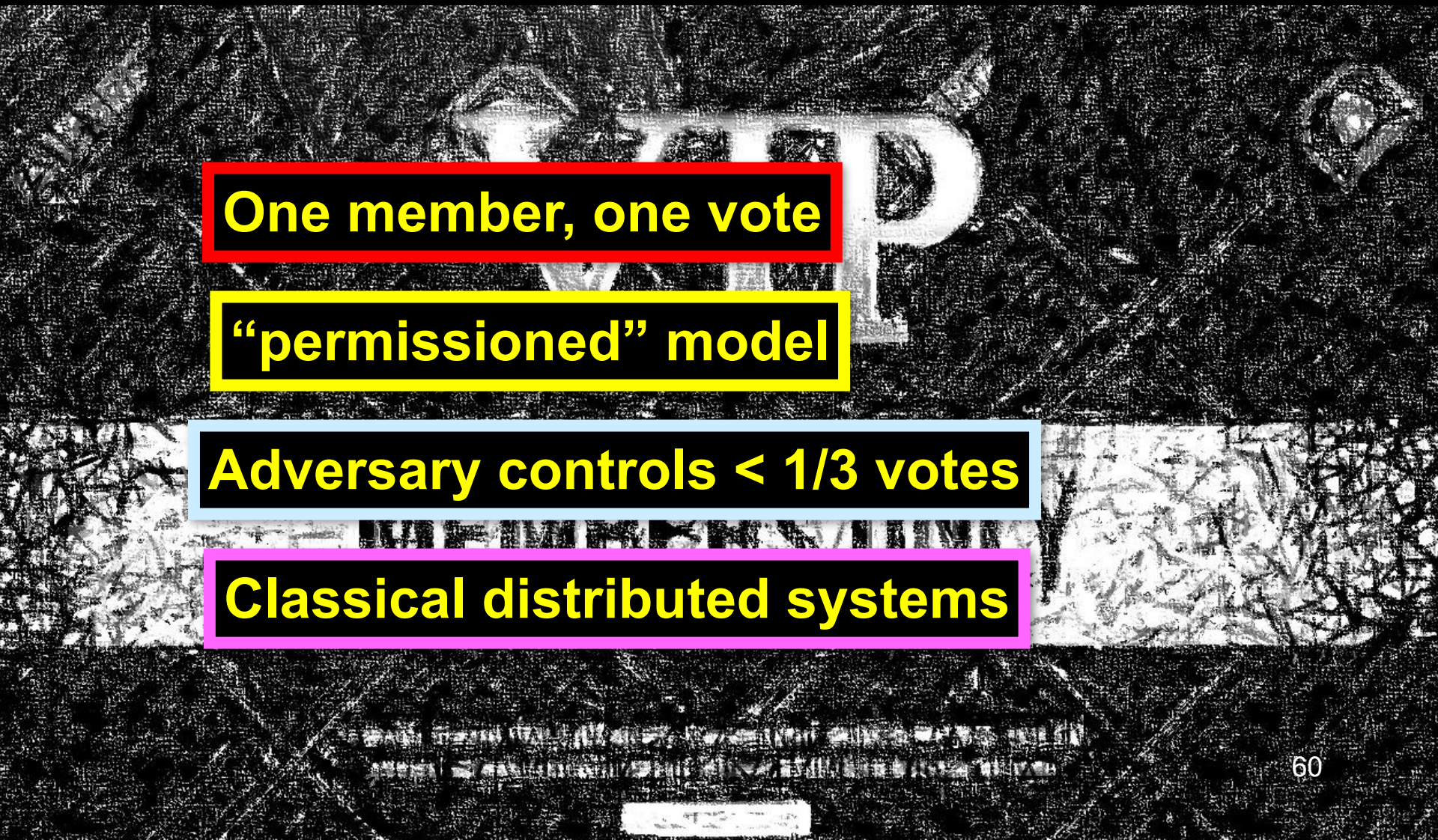


One member, one vote

“permissioned” model

Adversary controls < 1/3 votes

Proof of Membership



One member, one vote

“permissioned” model

Adversary controls < 1/3 votes

Classical distributed systems

making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority

making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority

Proof of Work

Proof of Work

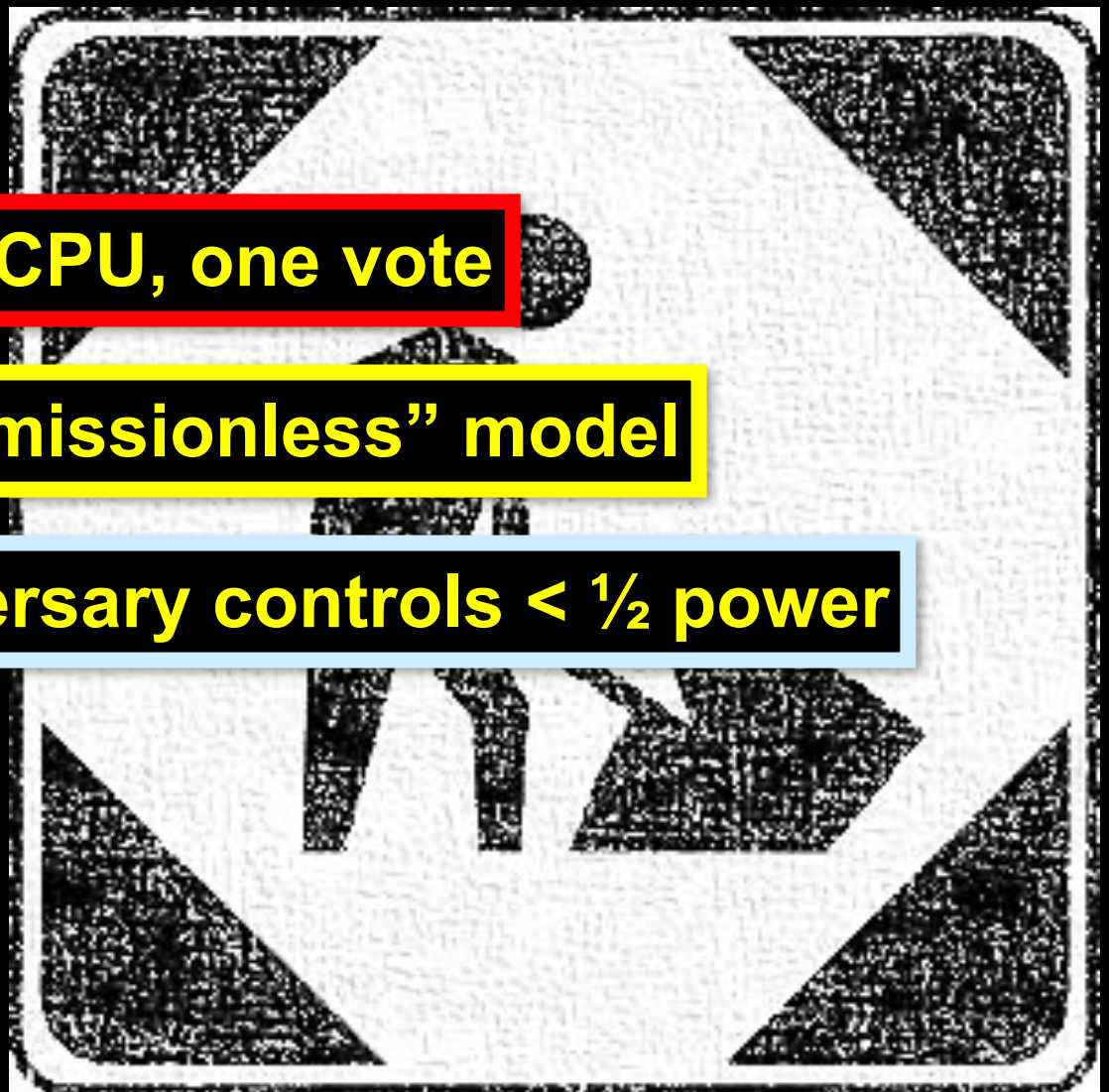
One CPU, one vote

Proof of Work

One CPU, one vote

“permissionless” model

Proof of Work

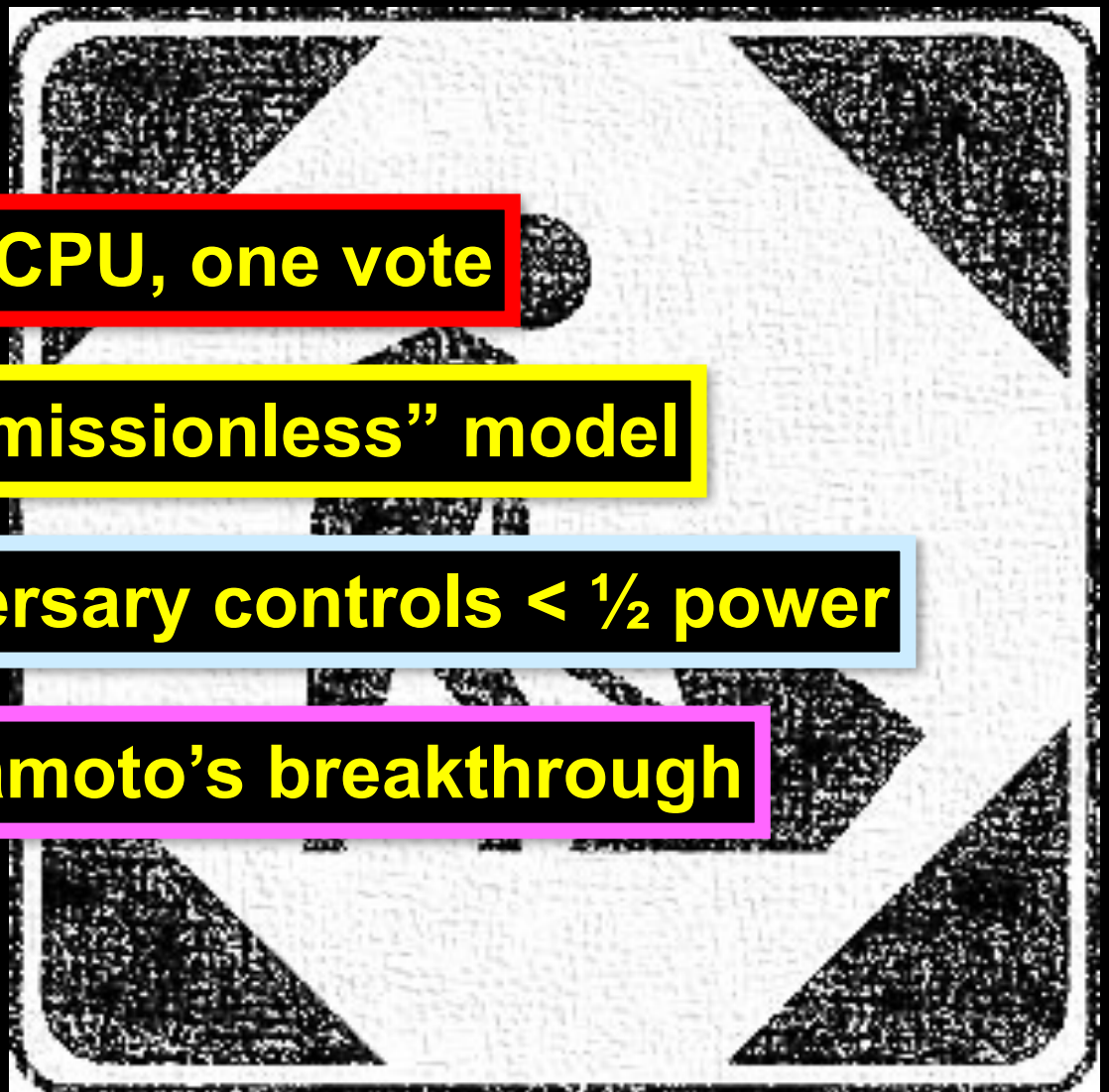


One CPU, one vote

“permissionless” model

Adversary controls < ½ power

Proof of Work



One CPU, one vote

“permissionless” model

Adversary controls < ½ power

Nakamoto’s breakthrough

Bitcoin Adversary

"The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes."

S. Nakamoto

Popular Culture

Sillicon Valley Season 5 Finale Episode – 51 % Attack

by Rohit Kukreja | May 21, 2018 | Bitcoin, Cryptocurrency, Cryptocurrency News

Fake Dentralization

Decentralization in Bitcoin and Ethereum Networks

Adem Efe Gencer[1,2], Soumya Basu[1,2], Ittay Eyal[1,3], Robbert van Renesse[1,2], and Emin Gün Sirer[1,2]

[1] Initiative for Cryptocurrencies and Contracts (IC3)
[2] Computer Science Department, Cornell University
[3] Electrical Engineering Department, Technion

Abstract. Blockchain-based cryptocurrencies have demonstrated how to securely implement traditionally centralized systems, such as currencies, in a decentralized fashion. However, there have been few measurement studies on the level of decentralization they achieve in practice. We present a measurement study on various decentralization metrics of two of the leading cryptocurrencies with the largest market capitalization and user base, Bitcoin and Ethereum. We investigate the extent of decentralization by measuring the network resources of nodes and the interconnection among them, the protocol requirements affecting the operation of nodes, and the robustness of the two systems against
In particular, we adapted existing internet
used the Falcon Relay Network
our data. We

Fake Dentralization

Decentralization in Bitcoin and Ethereum Networks

Adem Efe Gencer[1,2]

Bitcoin (& Ethereum) are pretty centralized

Gün Sirer[1,2], Robbert van Renesse[1,2],

[1] Initiative for Cryptocurrencies and Contracts (IC3)
[2] Computer Science Department, Cornell University
[3] Electrical Engineering Department, Technion

Abstract. Blockchain-based cryptocurrencies have demonstrated how to securely implement traditionally centralized systems, such as currencies, in a decentralized fashion. However, there have been few measurement studies on the level of decentralization they achieve in practice. We present a measurement study on various decentralization metrics of two of the leading cryptocurrencies with the largest market capitalization and user base, Bitcoin and Ethereum. We investigate the extent of decentralization by measuring the network resources of nodes and the interconnection among them, the protocol requirements affecting the operation of nodes, and the robustness of the two systems against In particular, we adapted existing internet used the Falcon Relay Network our data. We

Fake Dentralization

Decentralization in Bitcoin and Ethereum Networks

Adem Efe Gencer[1,2]

Bitcoin (& Ethereum) are pretty centralized

Initiative for Cryptoeconomics and Blockchain Technologies

Emin Gün Sirer[1,2], Robbert van Renesse[1,2],

Top 4 Bitcoin Miners > 53% power

Department, Technion

Abstract. Blockchain-based cryptocurrencies have demonstrated how to securely implement traditionally centralized systems, such as currencies, in a decentralized fashion. However, there have been few measurement studies on the level of decentralization they achieve in practice. We present a measurement study on various decentralization metrics of two of the leading cryptocurrencies with the largest market capitalization and user base, Bitcoin and Ethereum. We investigate the extent of decentralization by measuring the network resources of nodes and the interconnection among them, the protocol requirements affecting the operation of nodes, and the robustness of the two systems against In particular, we adapted existing internet measurement used the Falcon Relay Network our data. We

Fake Dentralization

Decentralization in Bitcoin and Ethereum Networks

Adem Efe Gencer[1,2]
Initiative for Crypto
Gün Sirer[1,2]
Robbert van Renesse[1,2],
partment, Technion

Bitcoin (& Ethereum) are pretty centralized

Top 4 Bitcoin Miners > 53% power

Top 3 Ethereum miners > 61% power

Abstract

demonstrated how
centralized systems, such as curren-
fashion. However, there have been few measure-
studies on the level of decentralization they achieve in practice.
We present a measurement study on various decentralization metrics of
two of the leading cryptocurrencies with the largest market capitaliza-
tion and user base, Bitcoin and Ethereum. We investigate the extent of
decentralization by measuring the network resources of nodes and the
interconnection among them, the protocol requirements affecting the op-
eration of nodes, and the robustness of the two systems
In particular, we adapted existing internet
used the Falcon Relay
our data. We

Fake Dentralization

Decentralization in Bitcoin and Ethereum Networks

Adem Efe Gencer[1,2] ... Sirer[1,2] ... Robbert van Renesse[1,2],

[2] ... Initiative for Crypto... ... partment, Technion ... versity

Abstract ...

Bitcoin (& Ethereum) are pretty centralized

Top 4 Bitcoin Miners > 53% power

Top 3 Ethereum miners > 61% power

15 Bitcoin, 11 Ethereum miners > 90%

... studies on the level of ... fashion. However, ... centralized systems, such ... demonstrated how

We present a ... the largest market capitaliza-

... Bitcoin and Ethereum. We investigate the extent of interconnection by measuring the network resources of nodes and the eration of nodes, and the robustness of the two systems affecting the op- In particular, we adapted existing internet ... used the Falcon Relay Network ... our data. We ...

PoW Encourages Centralization

TECH INDUSTRY

This coal power plant is being reopened for blockchain mining

The now shuttered coal-fired power station on Australia's east coast will offer cheap power prices to blockchain operators.

BY CLAIRE REILLY | APRIL 11, 2018 12:27 AM PDT

PoW Encourages Centralization

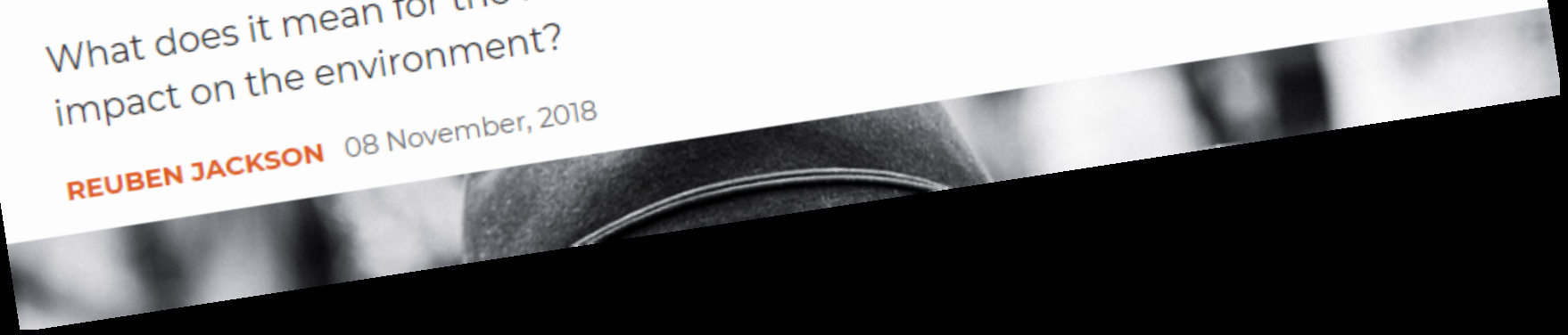
Wasteful

Wasteful

Bitcoin mining uses as much energy as mining for gold, study finds

What does it mean for the future of the cryptocurrency movement and its impact on the environment?

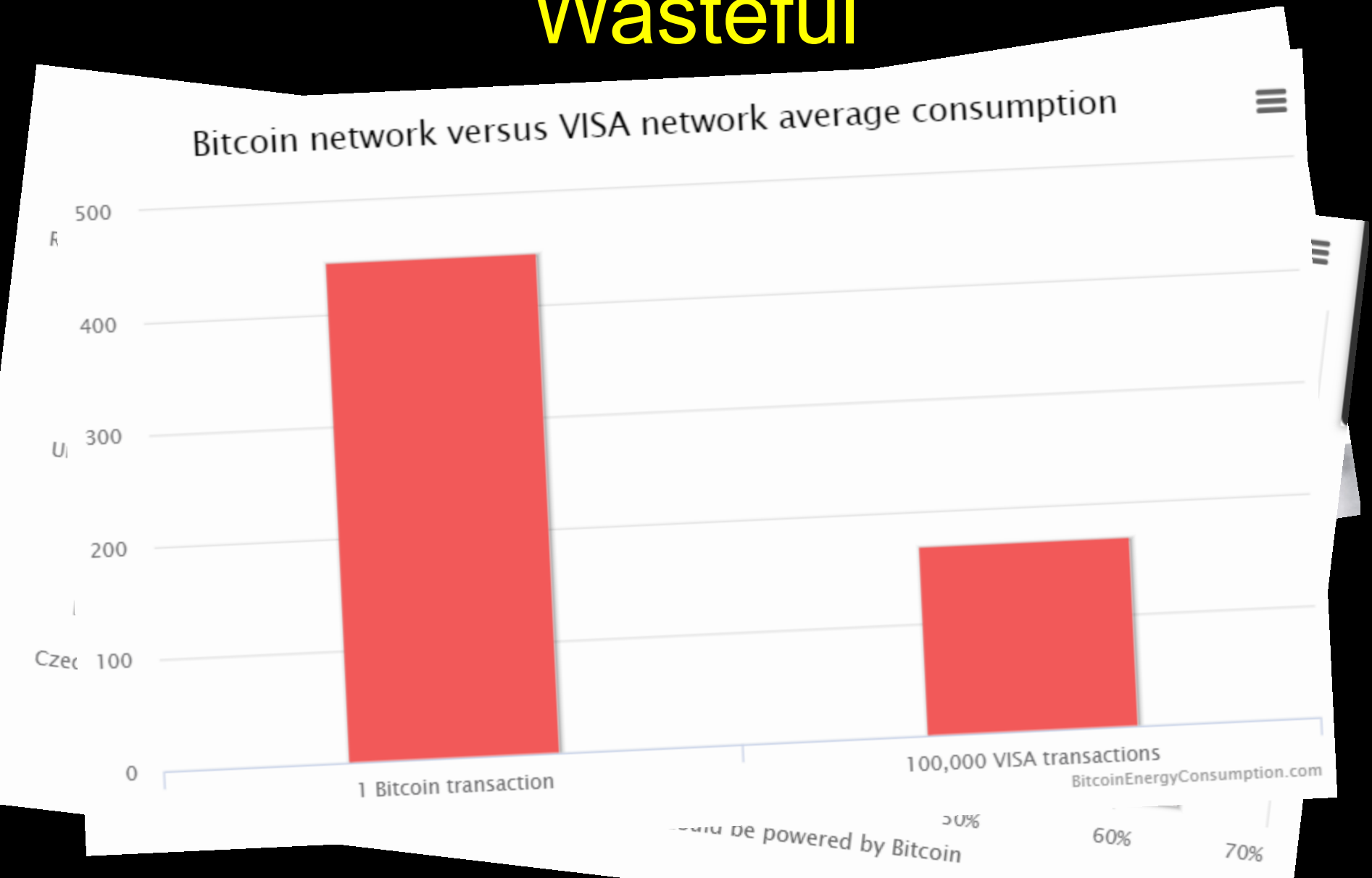
REUBEN JACKSON 08 November, 2018



Wasteful


Bitcoin Energy Consumption Relative to Several Countries
United States
Russian Federation
Canada
Germany
France
United Kingdom
Italy
Australia
Netherlands
Czech Republic
0%
10%
20%
30%
40%
50%
60%
70%
Percentage that could be powered by Bitcoin

Wasteful

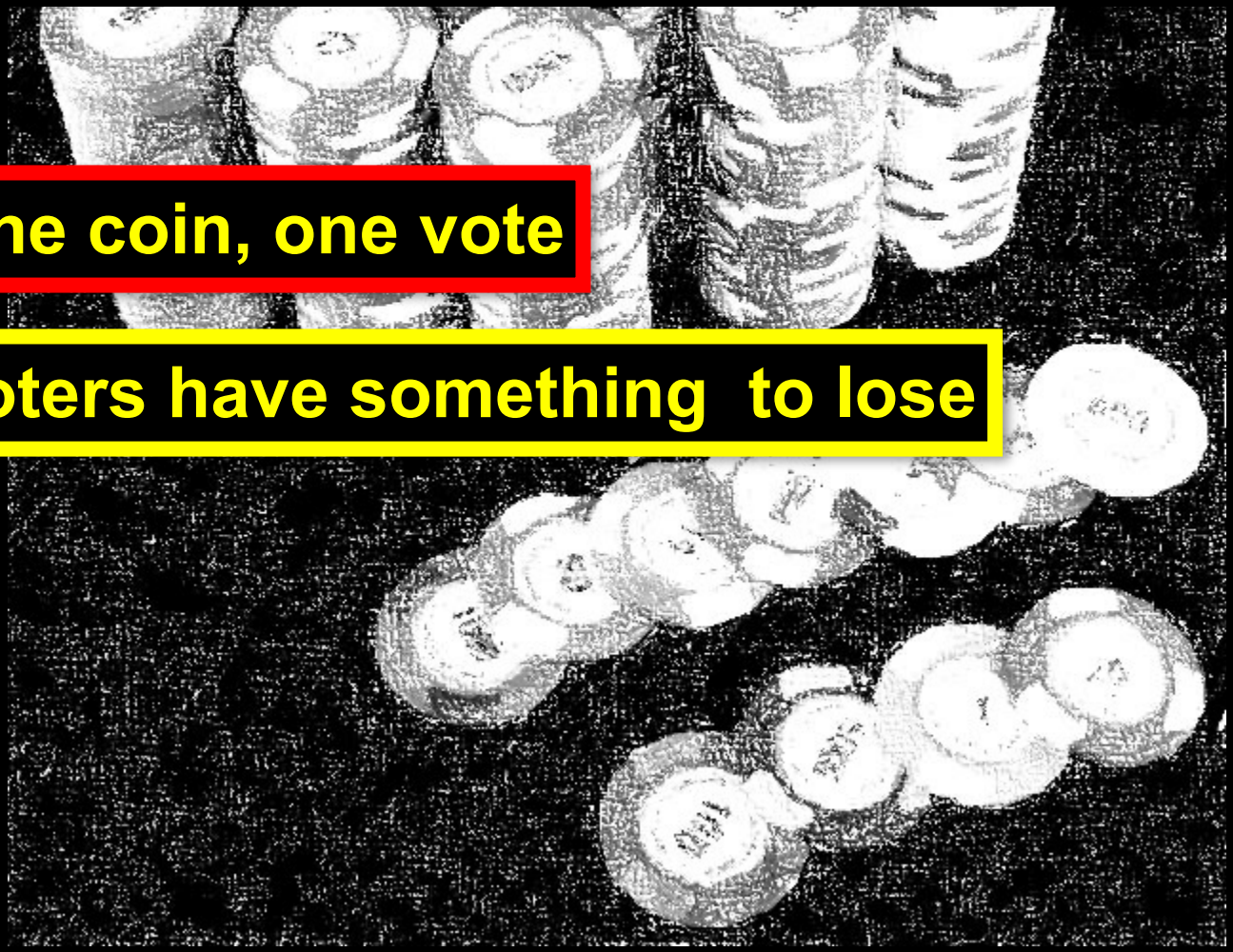

Bitcoin network versus VISA network average consumption
500
400
300
200
100
0
1 Bitcoin transaction
100,000 VISA transactions
BitcoinEnergyConsumption.com
50%
60%
70%

Proof of Stake

Proof of Stake

One coin, one vote

Proof of Stake



One coin, one vote

Voters have something to lose

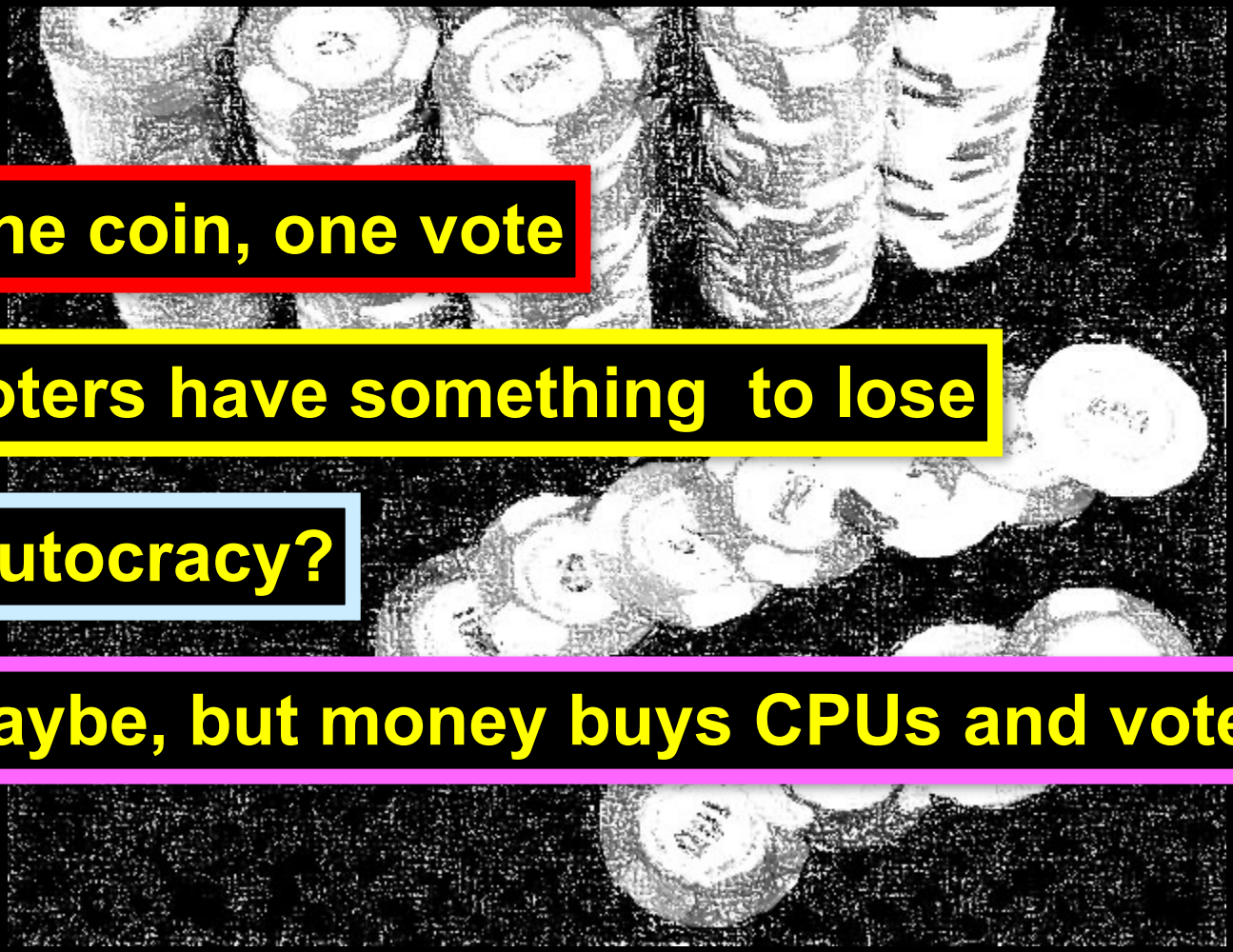
Proof of Stake

One coin, one vote

Voters have something to lose

Plutocracy?

Proof of Stake



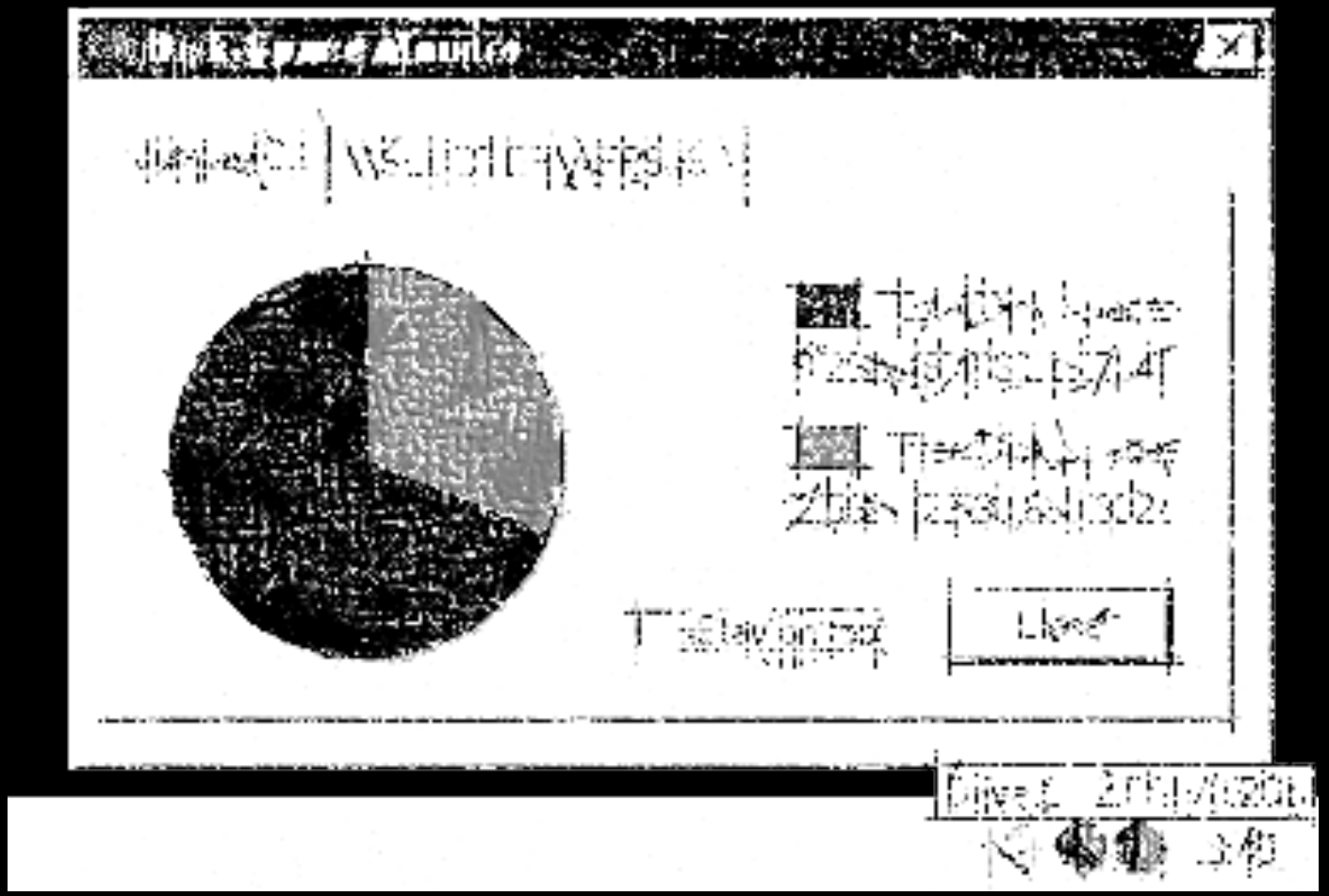
One coin, one vote

Voters have something to lose

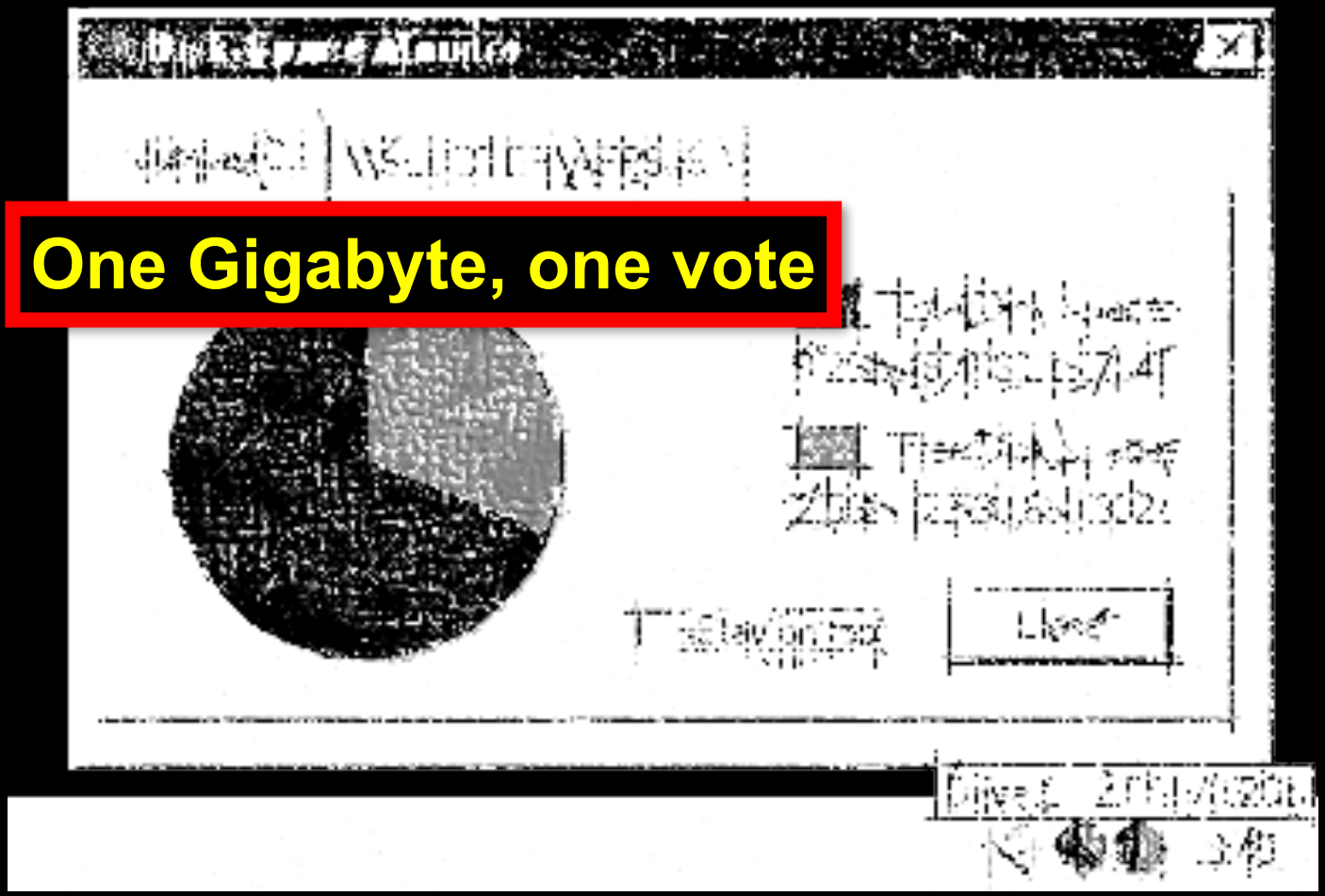
Plutocracy?

Maybe, but money buys CPUs and votes …

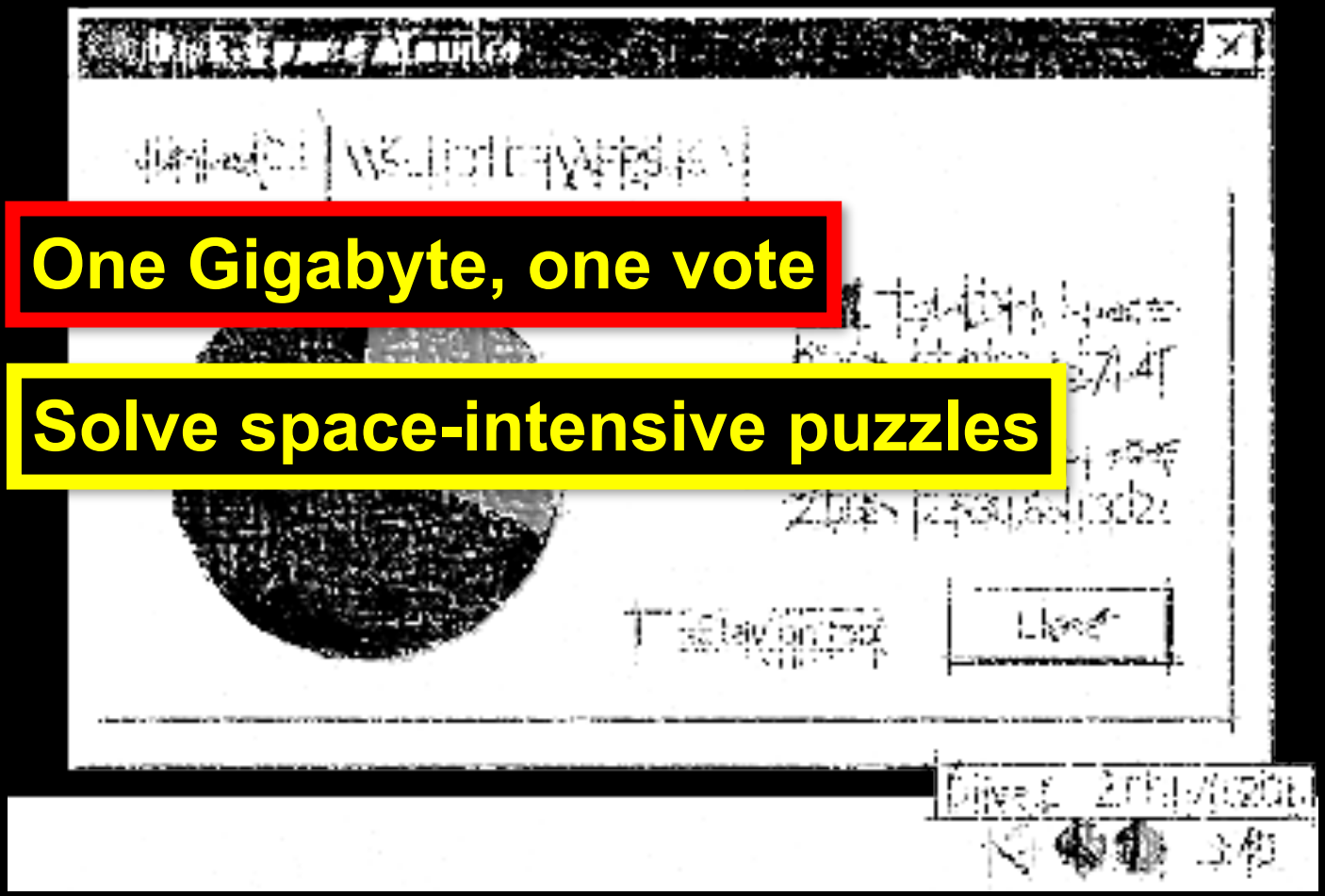
Proof of Space



Proof of Space



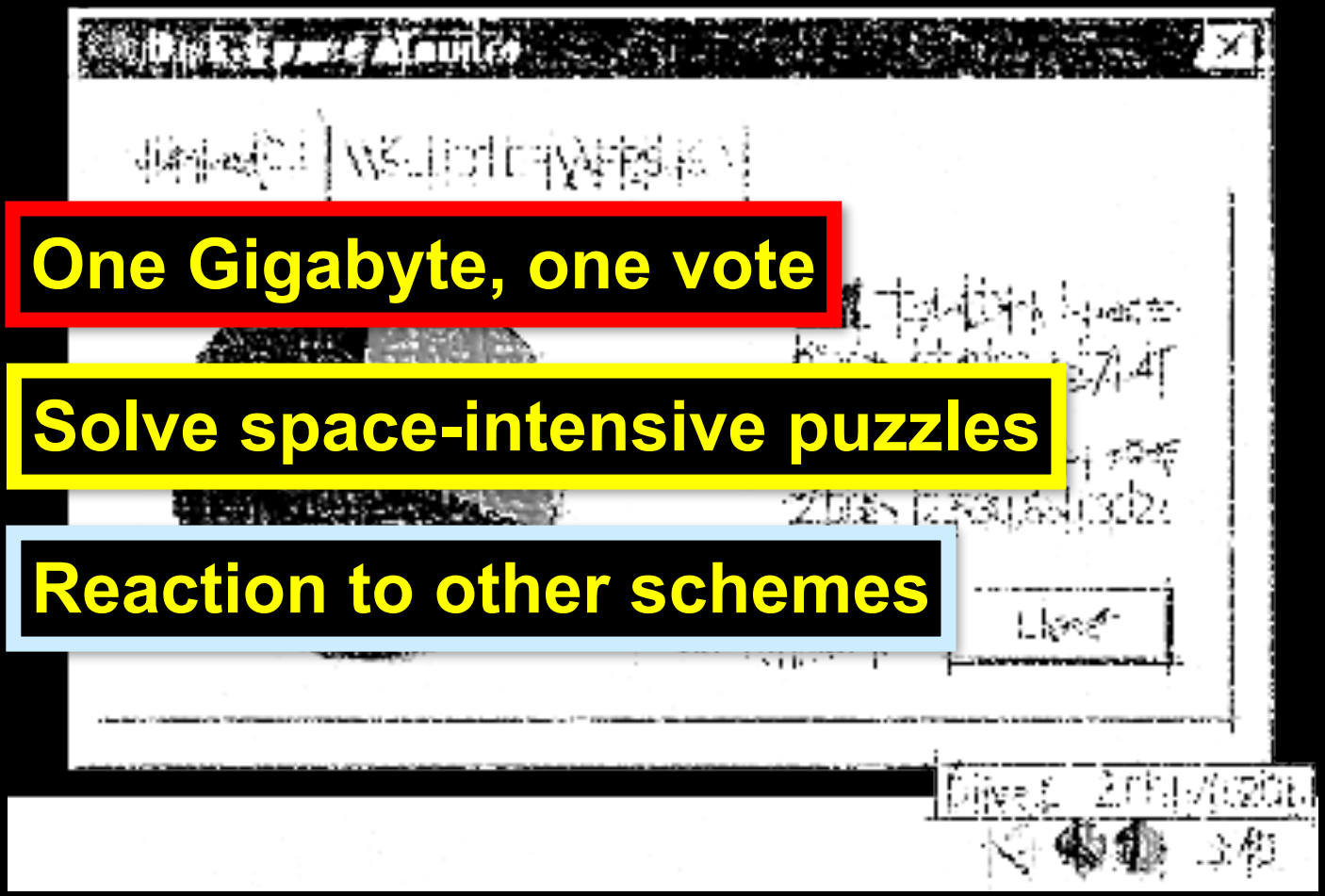
Proof of Space



One Gigabyte, one vote

Solve space-intensive puzzles

Proof of Space

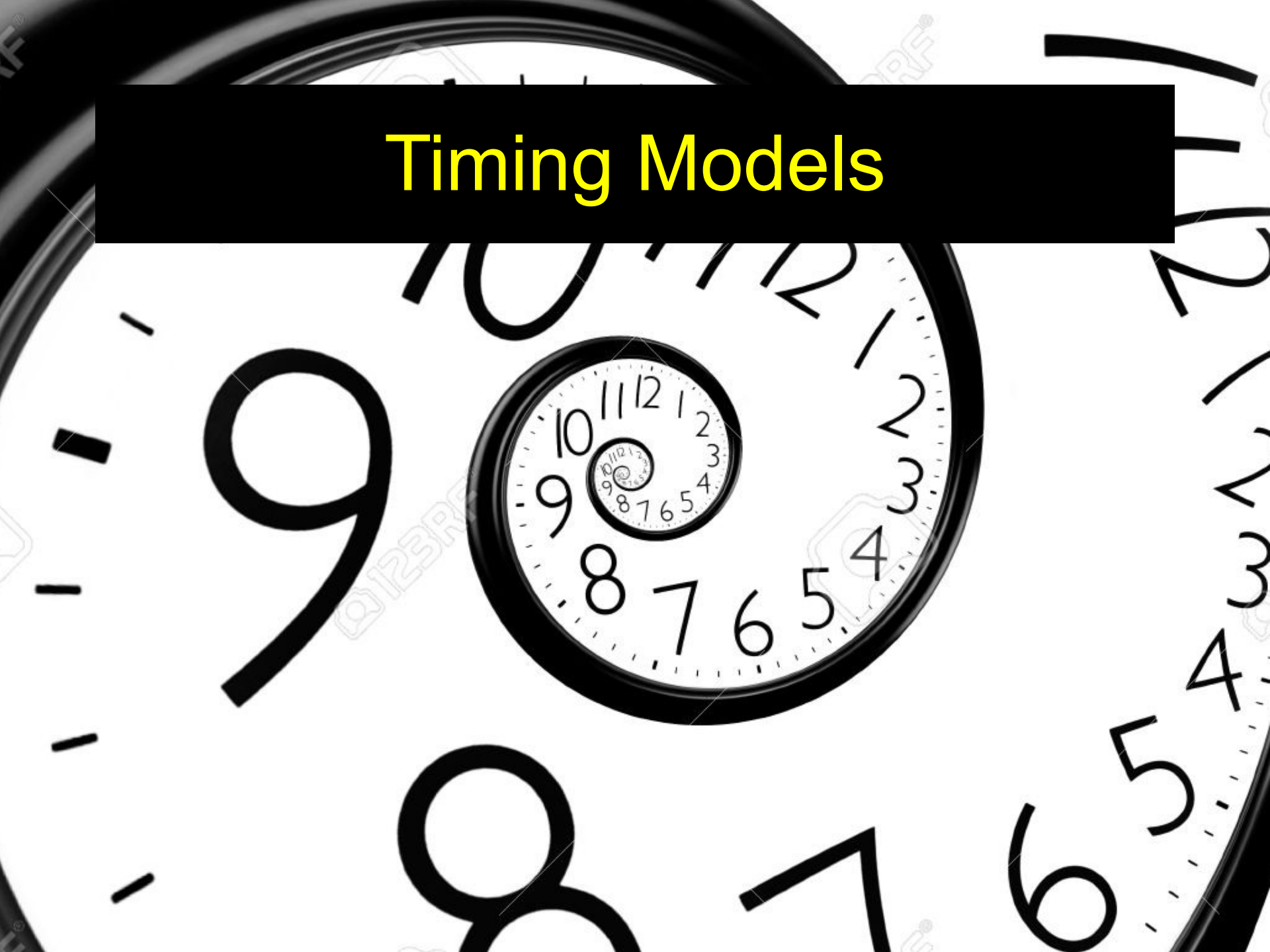


One Gigabyte, one vote

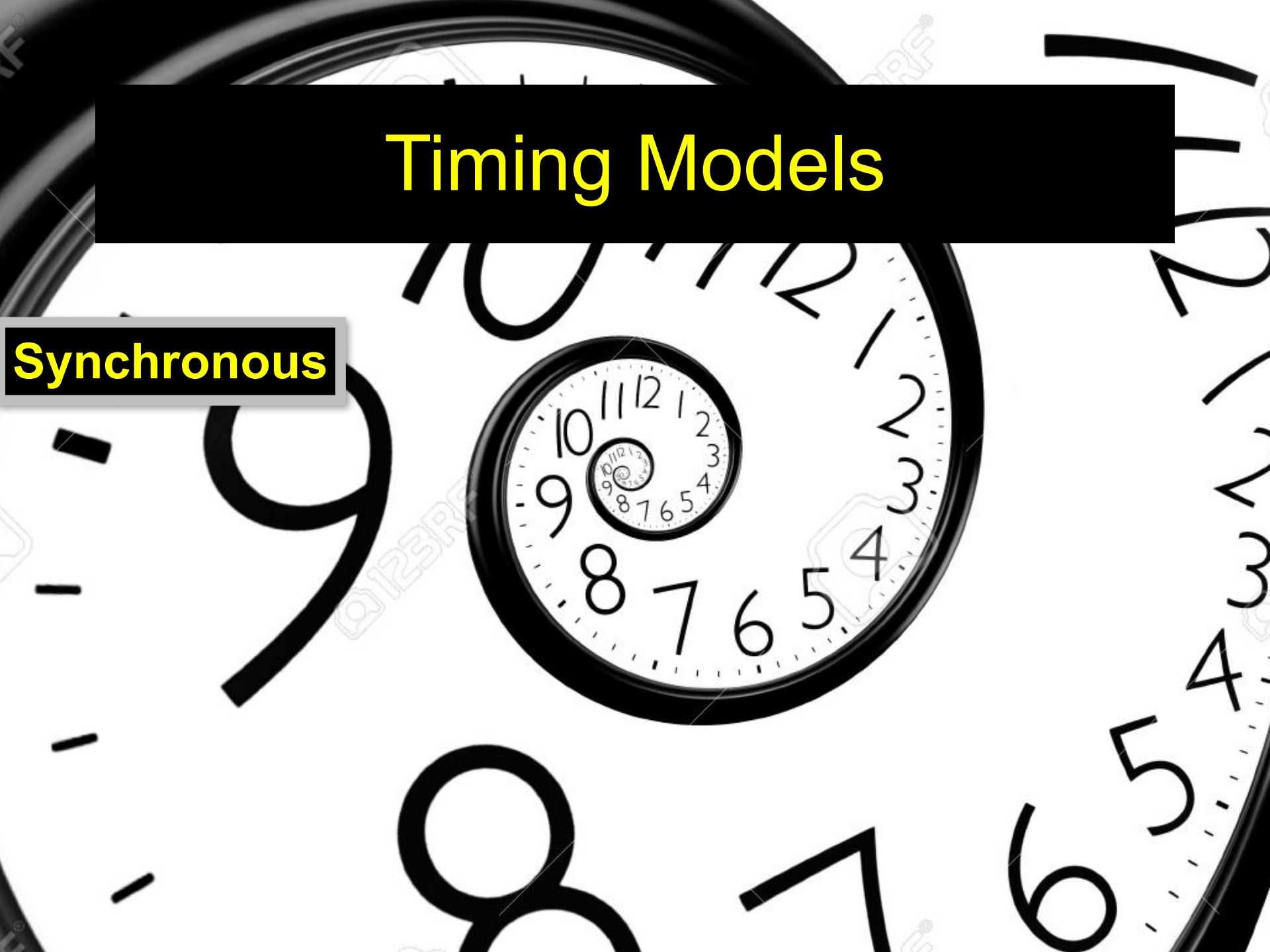
Solve space-intensive puzzles

Reaction to other schemes

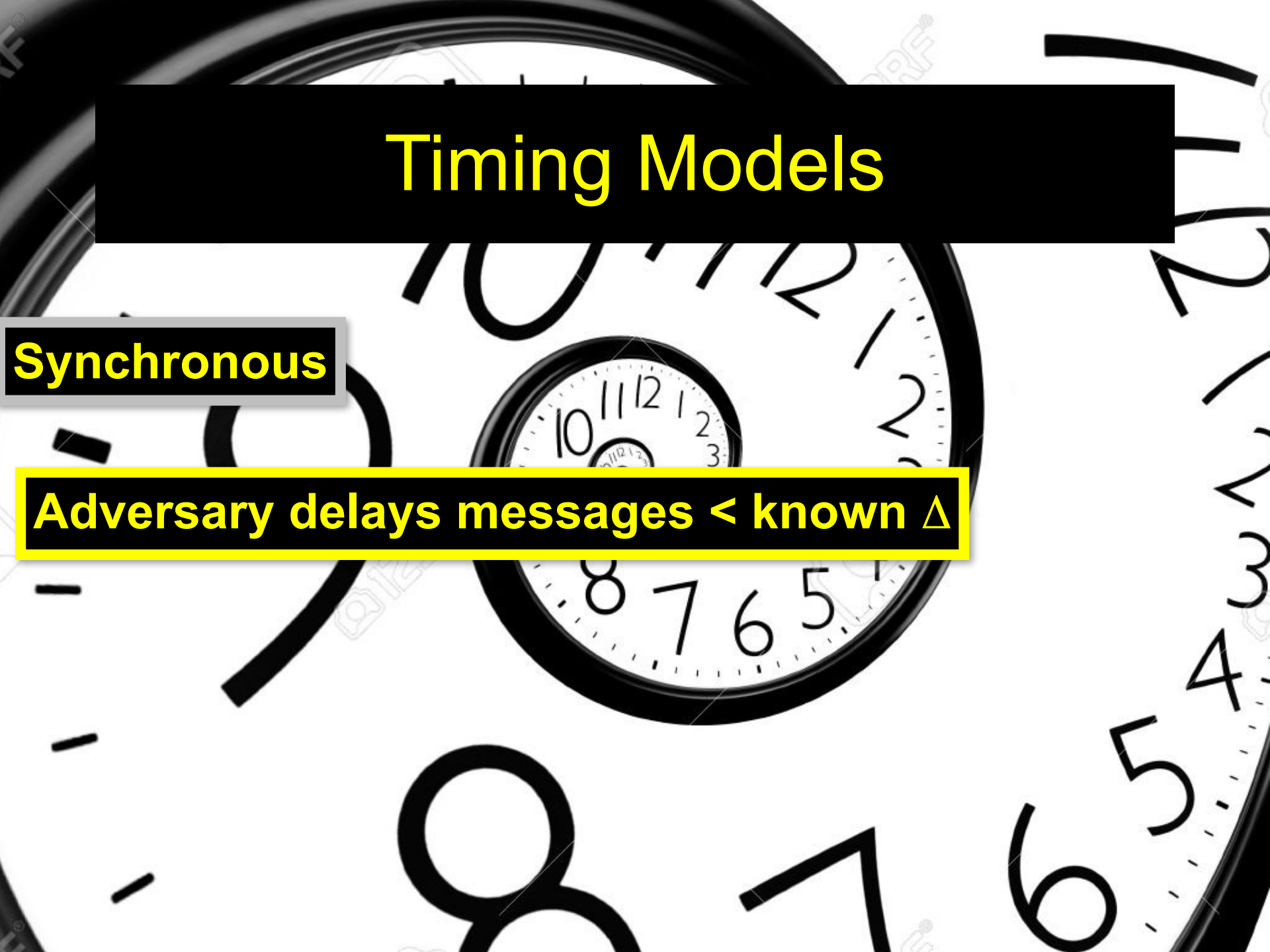
Timing Models



Timing Models



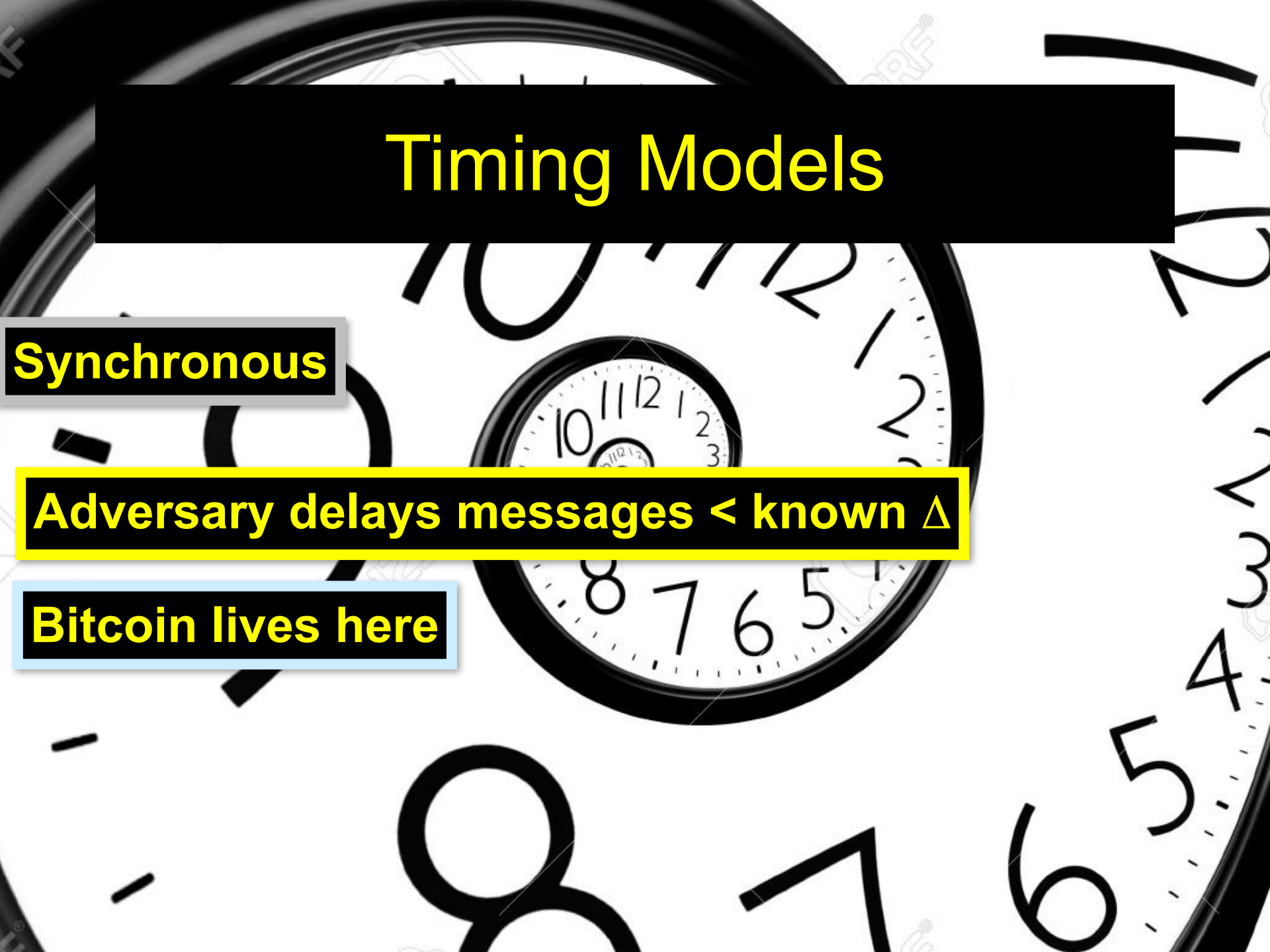
Synchronous



Timing Models

Synchronous

Adversary delays messages < known Δ

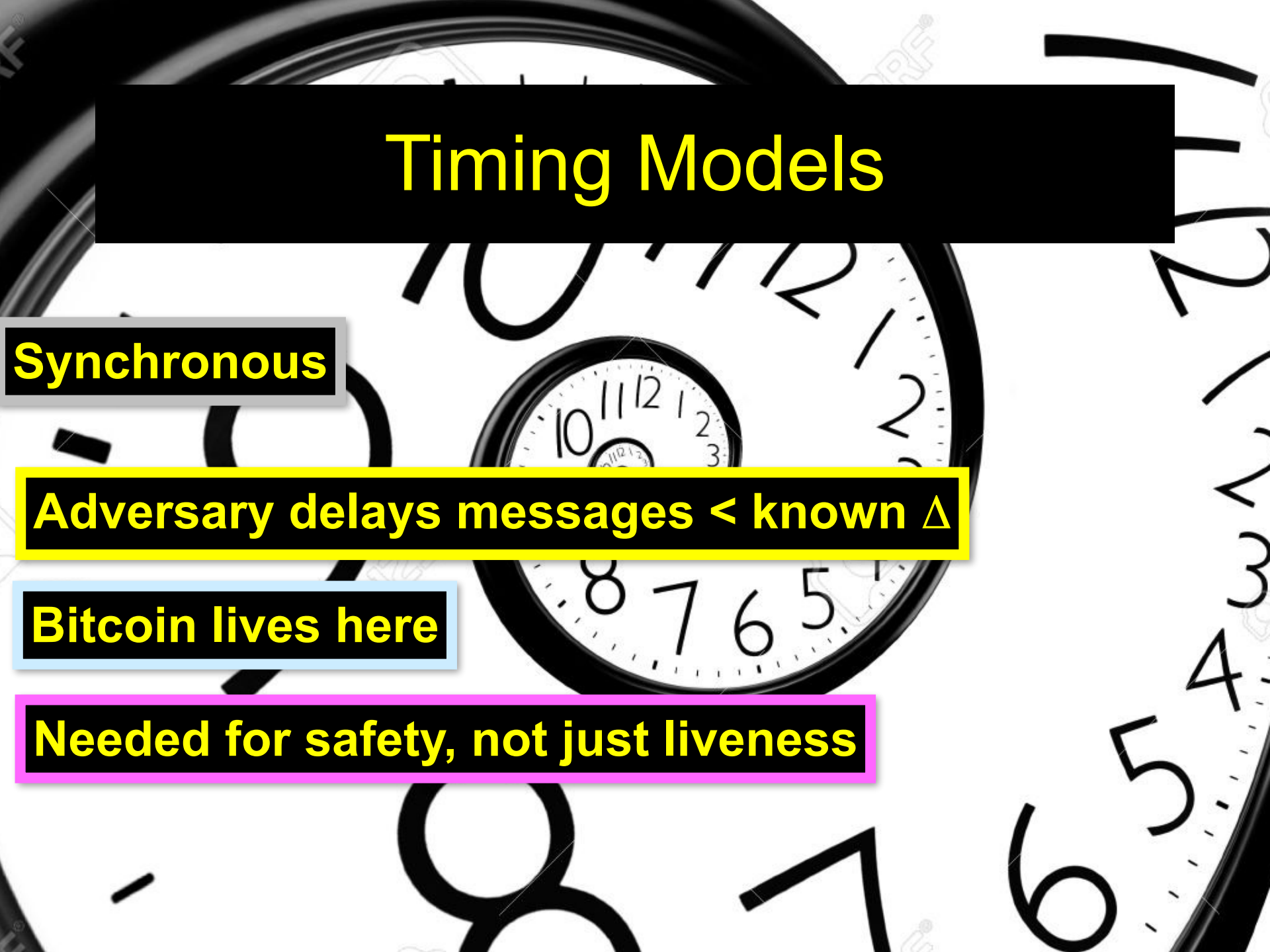


Timing Models

Synchronous

Adversary delays messages < known Δ

Bitcoin lives here



Timing Models

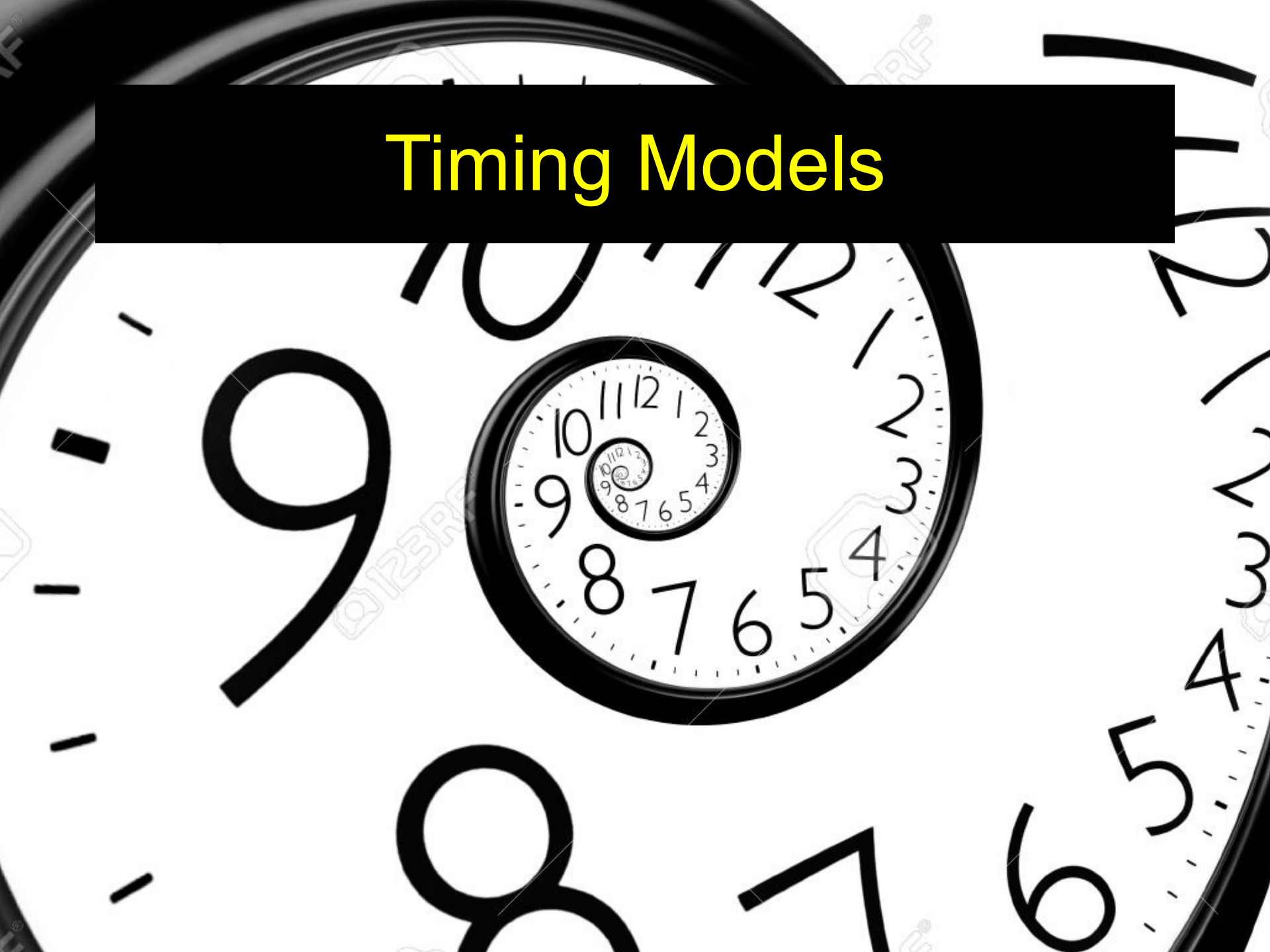
Synchronous

Adversary delays messages < known Δ

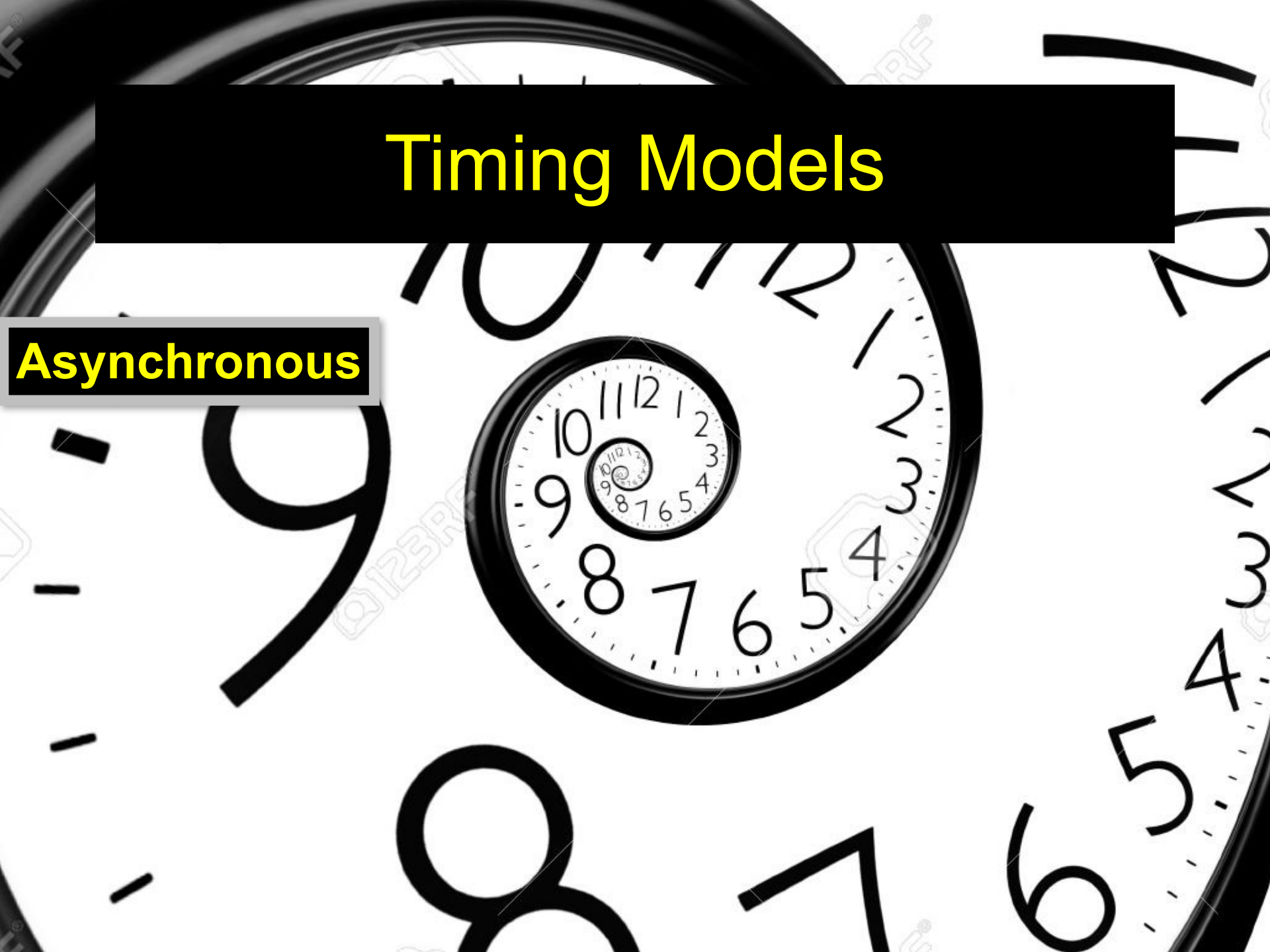
Bitcoin lives here

Needed for safety, not just liveness

Timing Models



Timing Models

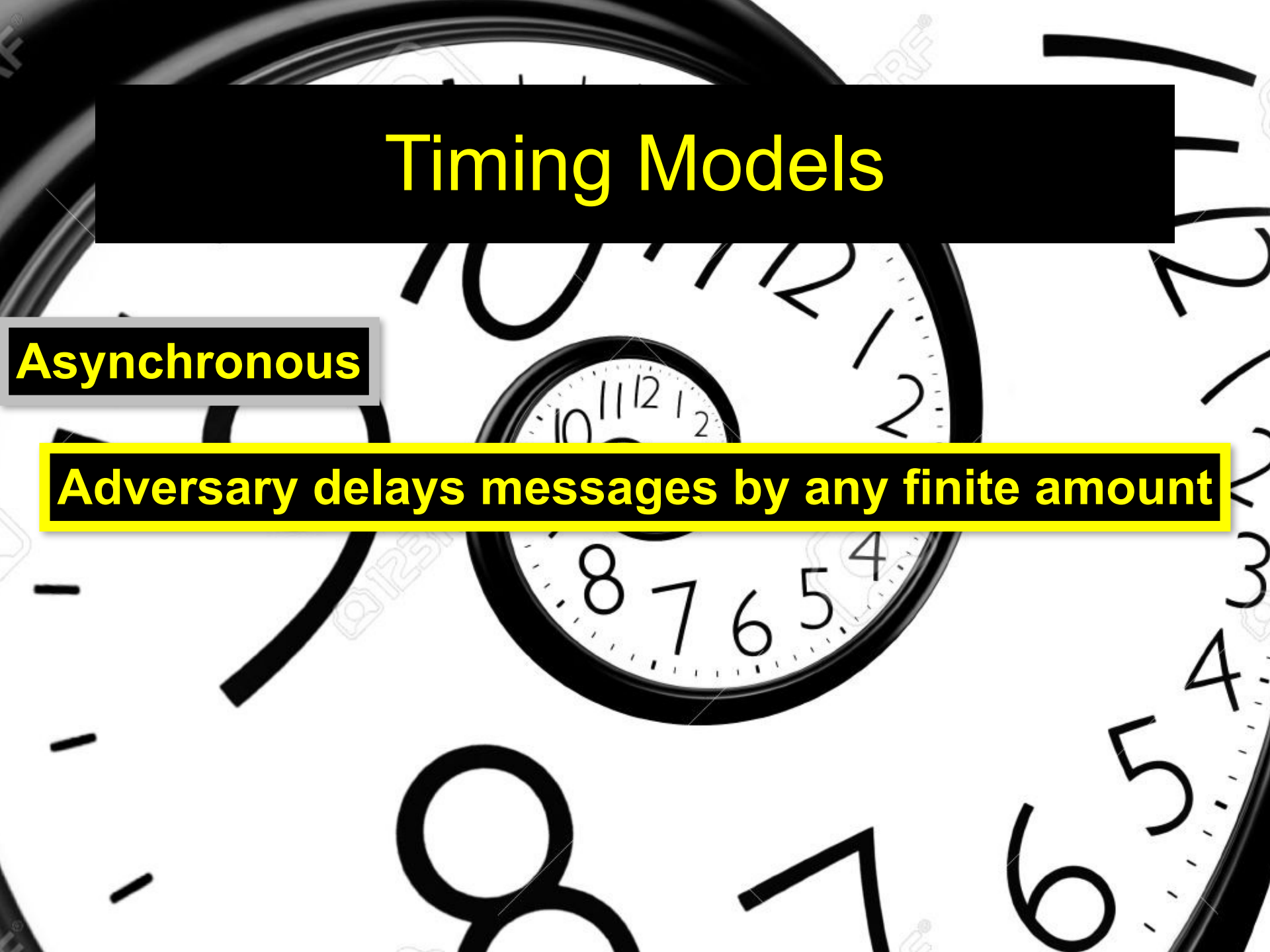


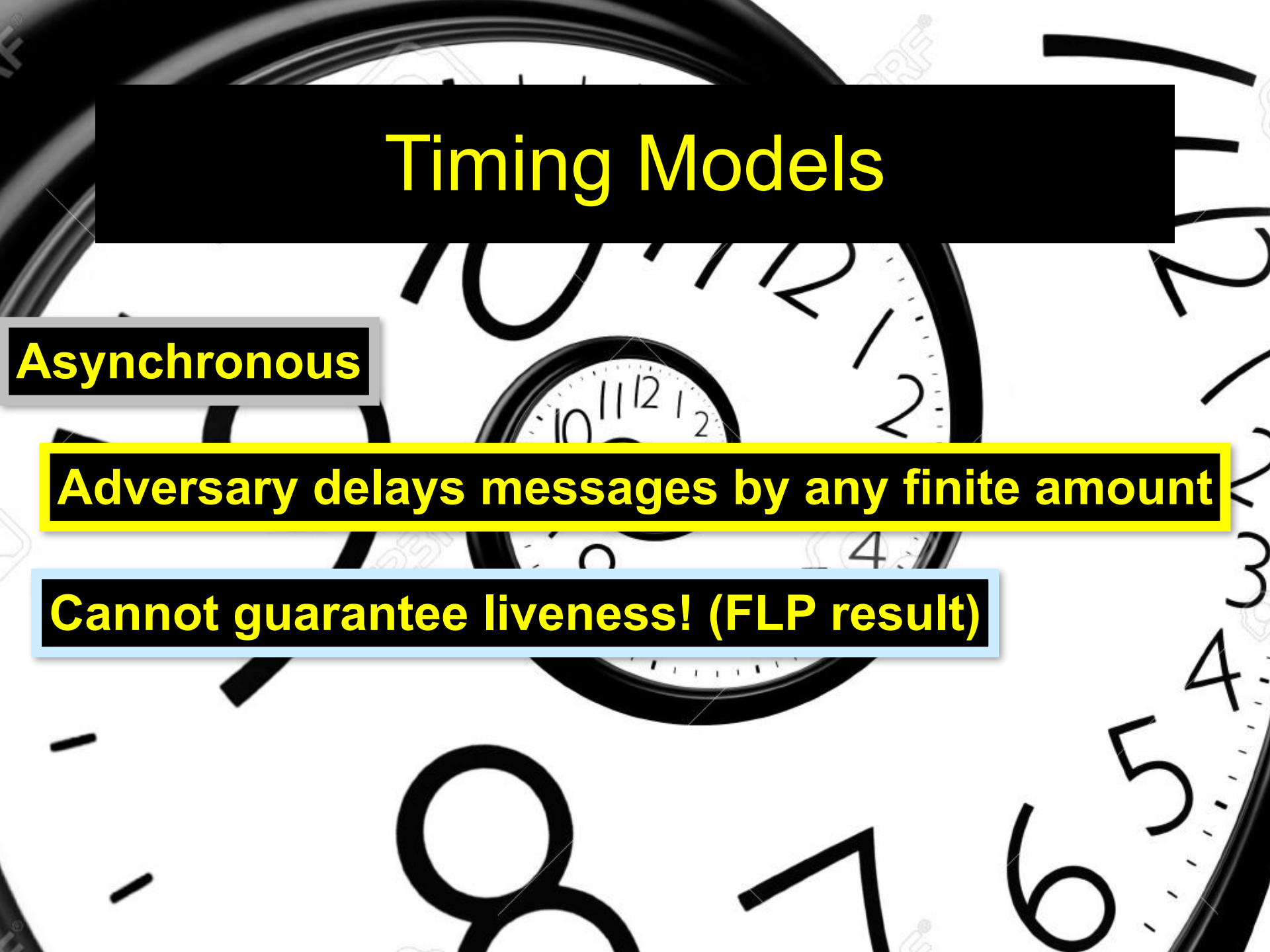
Asynchronous

Timing Models

Asynchronous

Adversary delays messages by any finite amount





Timing Models

Asynchronous

Adversary delays messages by any finite amount

Cannot guarantee liveness! (FLP result)

Timing Models

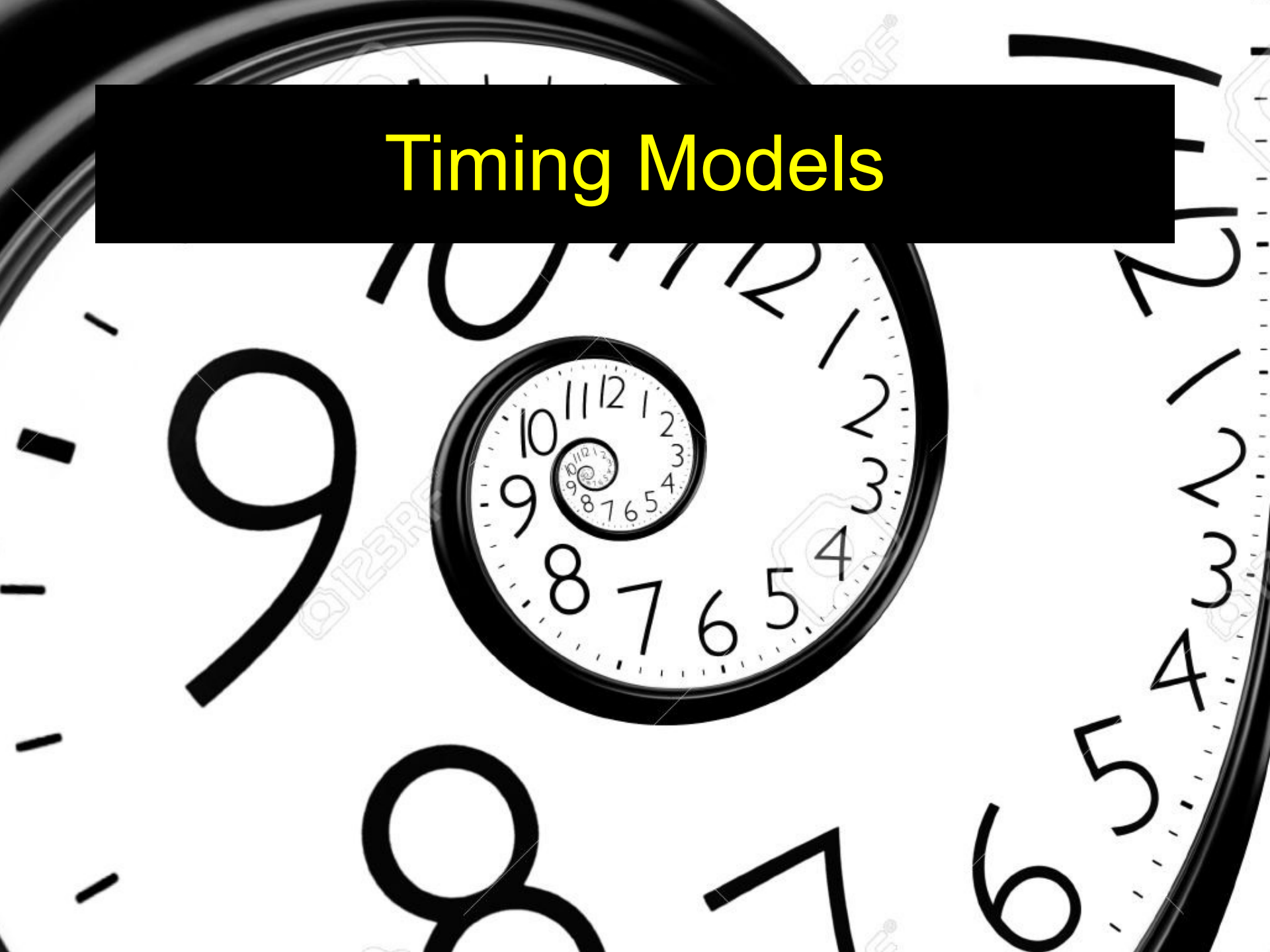
Asynchronous

Adversary delays messages by any finite amount

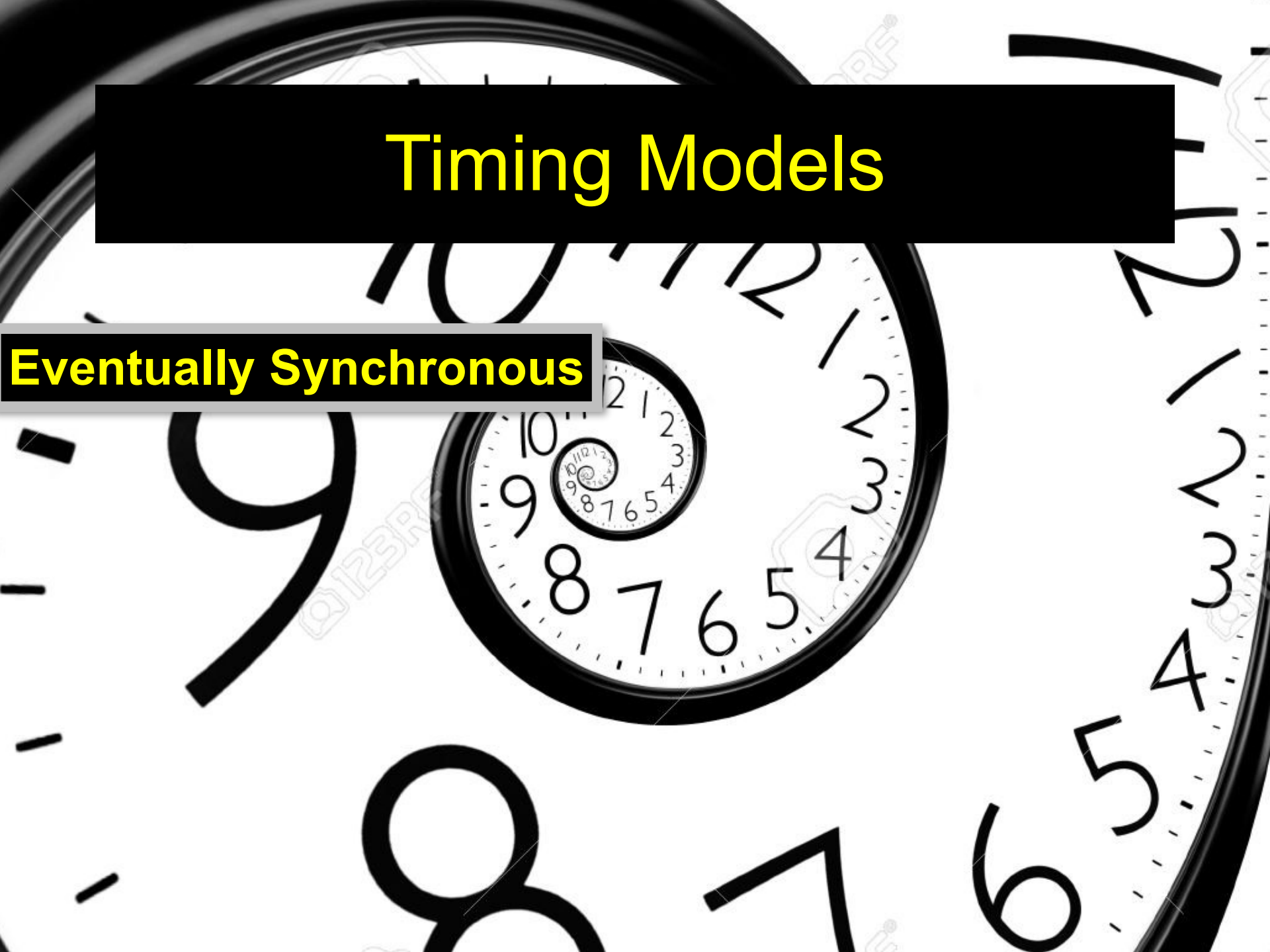
Cannot guarantee liveness! (FLP result)

Rrandomization for expected termination

Timing Models



Timing Models

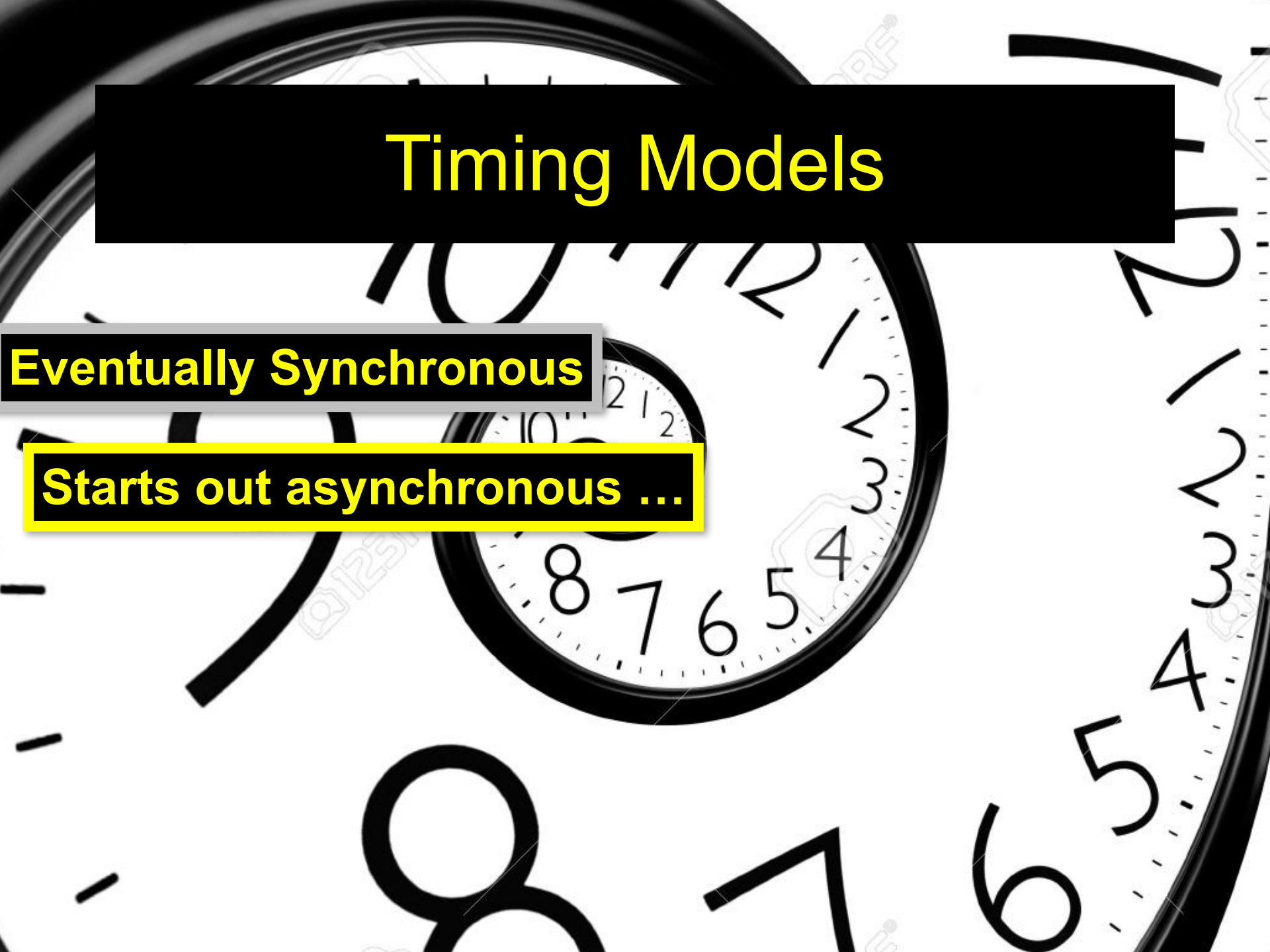


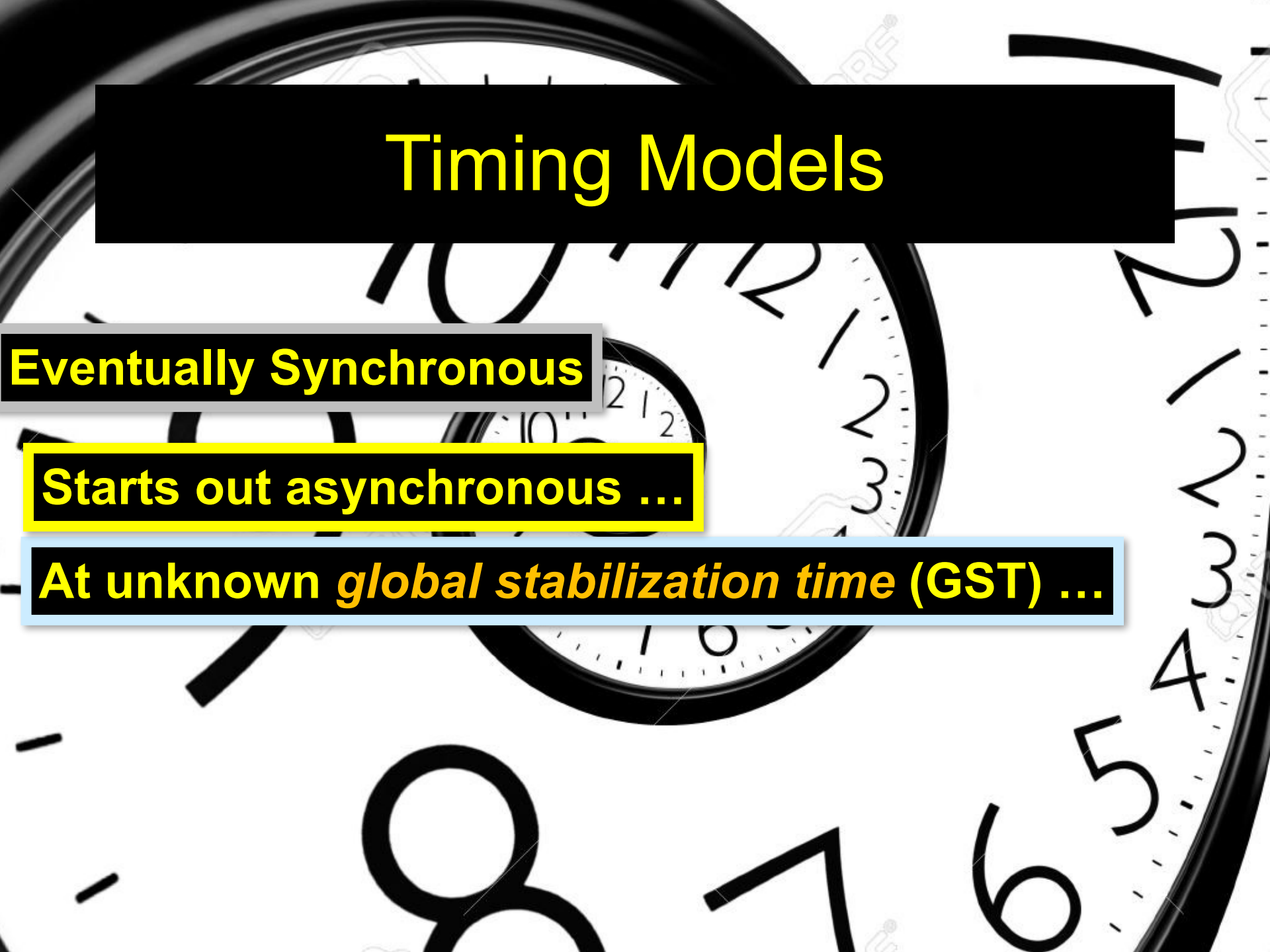
Eventually Synchronous

Timing Models

Eventually Synchronous

Starts out asynchronous …



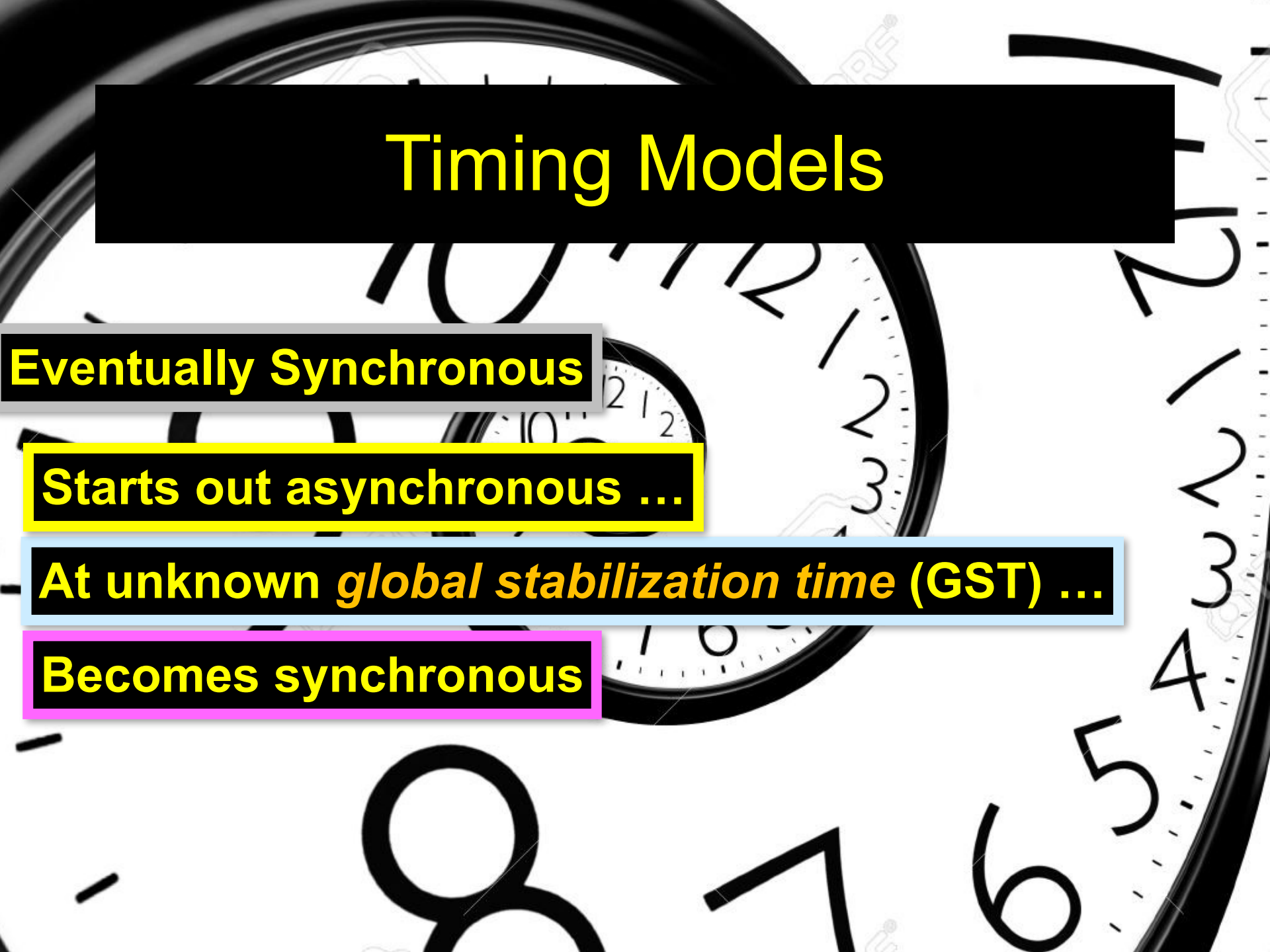


Timing Models

Eventually Synchronous

Starts out asynchronous …

At unknown *global stabilization time* (GST) …



Timing Models

Eventually Synchronous

Starts out asynchronous …

At unknown *global stabilization time* (GST) …

Becomes synchronous

Timing Models

Eventually Synchronous

Starts out asynchronous …

At unknown *global stabilization time* (GST) …

Becomes synchronous

Byzantantine Fault-tolerant (BFT) protocols live here

Timing Models

Eventually Synchronous

Starts out asynchronous …

At unknown *global stabilization time* (GST) …

Becomes synchronous

Byzantantine Fault-tolerant (BFT) protocols live here

Need: safety during asynchronous, liveness during synchronous